

**Propuesta de Proyecto de Big Data**

**Empresa XYZ**

# Indice

Contenido

[Indice 2](#_Toc41404518)

[Objetivos 3](#_Toc41404519)

[Exposición de la información 8](#_Toc41404520)

[Recursos y Planificación 9](#_Toc41404521)

[Plazos 10](#_Toc41404522)

[Procesamiento de datos 10](#_Toc41404523)

[Generación de reportes para el usuario 10](#_Toc41404524)

[Entregables del Proyecto 11](#_Toc41404525)

[Propuesta Económica 12](#_Toc41404526)

[Anexos 1](#_Toc41404527)

# Objetivos

El objetivo de este informe es proponer un diseño de arquitectura que pueda obtener los datos generados por los sitios web (compras, clics, etc.), información de redes sociales, entre otros y disponibilizar a científicos de datos, quien generará un modelo de recomendación de productos, que será desplegado por el usuario cuando esté navegando por alguno de los sitios en “tiempo real”.

Para ello, en el presente informe se entregan las bases, tiempos y alcances para el desarrollo de este sistema, separando en tres puntos que requiere este proyecto: Propuesta de Procesamiento de datos, Roles, Propuesta de Visualización Interactiva, Plazos, Entregables del Proyecto, Consideraciones para la prestación del Servicio y Propuesta económica.

La arquitectura expuesta tendrá un diseño escalable, permitiendo cualquier modificación, sin alterar su estructura. Además, contará con seguridad en el acceso de los datos y estará disponible en tiempo real.

El sistema cuenta con al menos tres roles de usuario, el primero el administrador del sistema, segundo el Cientista de datos y finalmente gerente de la empresa.

El proyecto

Partimos de la base para este trabajo, de que teníamos que dar un orden a la ejecución de las tareas con el fin de generar las prioridades. En este sentido tenemos en cuenta de que existe una gran cantidad de datos históricos los cuáles son muy convenientes para segmentar nuestros clientes y generar mejores campañas de marketing que es el objetivo final de este proyecto. En un inicio llegamos a pensar que podríamos hacer una arquitectura híbrida en donde servicios de la nube podían conectarse a datos históricos on-premise. Sin embargo, considerando que era mucha información creímos mejor subirla a la nube y analizarla allí; siendo que además la puesta en marcha de este proyecto en algún momento va a generar datos históricos. Acabamos determinando que el flujo de trabajo debería entonces tener el siguiente orden:

1. Generar arquitectura en la nube
2. Generar arquitectura de respaldo on Premise
3. Modelar los repositorios de información
4. Carga inicial de datos históricos a la nube
5. Programación de funcionalidades
6. Diseño de los reportes de usuario
7. Marcha blanca
8. Puesta en producción
9. Implementación de políticas de purga y políticas de respaldos de la data fría

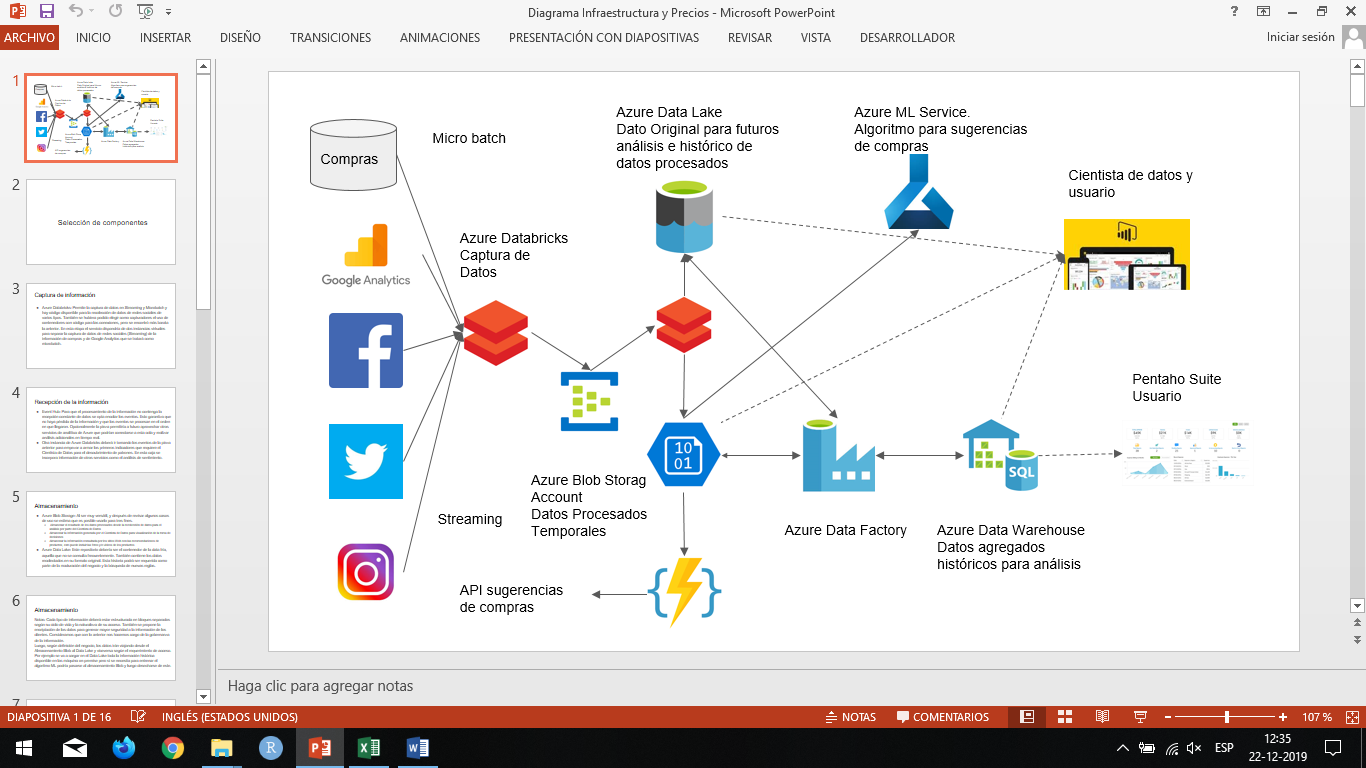
Diseño

Esta sección abordará el diseño de las arquitecturas y los componentes que las integran además del criterio asumido para cada selección.

Dado que la mayoría de las fuentes de información en este proyecto provienen de la internet se pensó era conveniente abordar la recolección de datos en tiempo real. Esto además nos daba la ventaja de ir ajustando los resultados de manera rápida y así generar un mejor servicio. Esto es una arquitectura Kappa

Teniendo en cuenta esto a continuación mostramos el gráfico de la arquitectura en la nube y sus componentes; hay que tener en cuenta que las orientaciones de las flecas que se muestra no es al azar, sino que tiene que ver con el recorrido que hacen los datos entre los componentes:

**Arquitectura en la Nube Azure**



Componentes y criterios de selección:

* ***Azure Databricks***: Permite la captura de datos en Streaming y Microbatch y hay código disponible para la recolección de datos de redes sociales de varios tipos. También se hubiera podido elegir como capturadores el uso de contenedores con código para las conexiones, pero se encontró más barata la anterior. En esta etapa el servicio dispondría de dos instancias virtuales para separar la captura de datos de redes sociales (Streaming) de la información de compras y de Google Analytics que se tratará como microbatch. Otra instancia de Azure Databricks deberá ir tomando los eventos de la pieza anterior para empezar a armar los primeros indicadores que requiere el Cientista de Datos para el descubrimiento de patrones. En esta caja se incorpora información de otros servicios como el análisis de sentimiento.
* ***Event Hub***: Para que el procesamiento de la información no contenga la recepción constante de datos se opta encolar los eventos. Esto garantiza que no haya pérdida de la información y que los eventos se procesan en el orden en que llegaron. Opcionalmente la pieza permitiría a futuro aprovechar otros servicios de analítica de Azure que podrían conectarse a esta cola y realizar análisis adicionales en tiempo real.
* ***Azure Blob Storage***: Al ser muy versátil, y después de revisar algunos casos de uso se estima que es posible usarlo para tres fines.
  + Almacenar el resultado de los datos procesados desde la recolección de datos para el análisis por parte del Cientista de Datos.
  + Almacenar la Información generada por el Cientista de Datos para visualización de la toma de decisiones.
  + Almacenar la Información consultada por los sitios Web con las recomendaciones de productos, esto puede incluir las fotos y/o videos de los productos.
* ***Azure Data Lake***: Este repositorio debería ser el contenedor de la data fría, aquella que no se consulta frecuentemente. También contiene los datos recolectados en su formato original. Esta historia podrá ser requerida como parte de la maduración del negocio y la búsqueda de nuevas reglas.

Notas: Cada tipo de información deberá estar estructurada en bloques separados (dígase un sistema de carpetas) según su ciclo de vida y la naturaleza de su acceso. También se propone la encriptación de los datos para generar mayor seguridad a la información de los clientes. Consideramos que con lo anterior nos hacemos cargo de la gobernanza de la información.

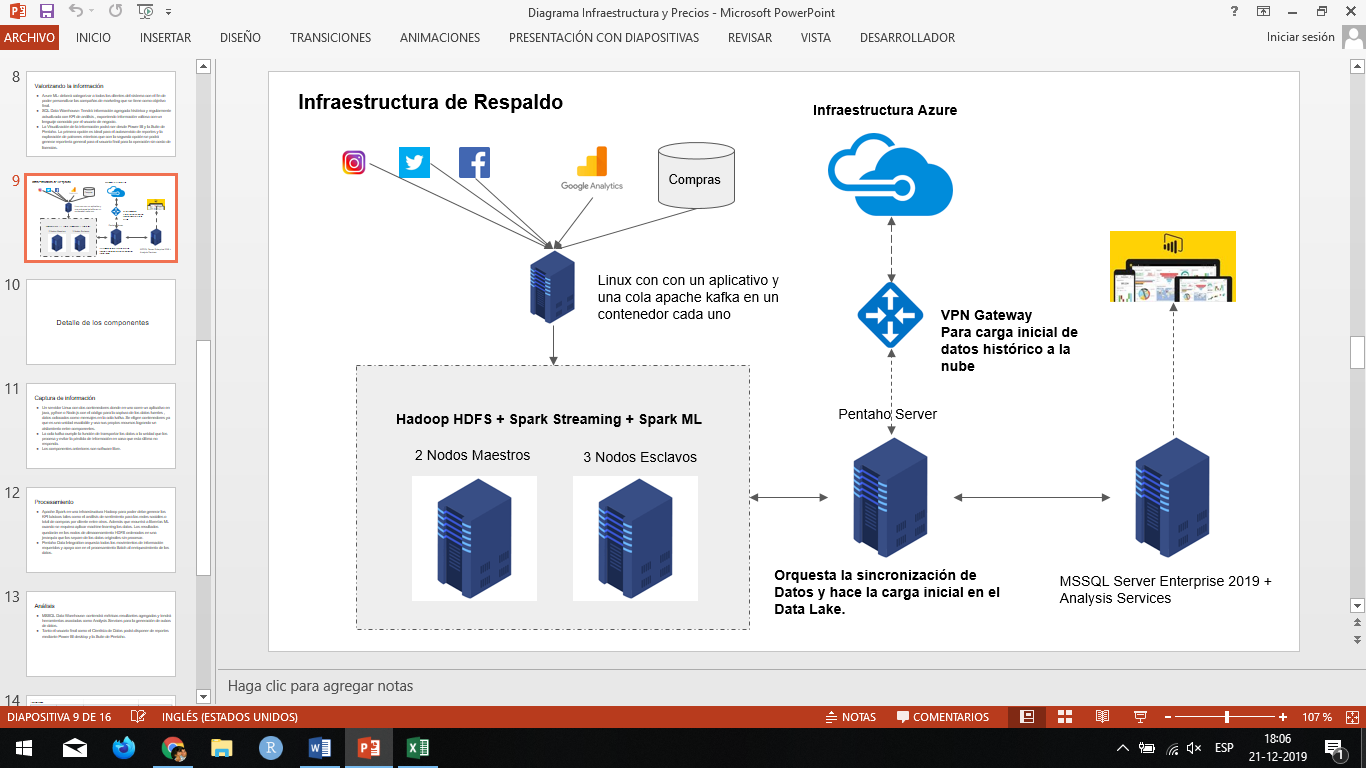
Luego, según definición del negocio, los datos irán viajando desde el Almacenamiento Blob al Data Lake y viceversa según el requerimiento de acceso. Por ejemplo, se va a cargar en el Data Lake toda la información histórica disponible en las máquinas on-premise pero si se necesita para entrenar el algoritmo ML podría pasarse al almacenamiento Blob y luego desecharse de este.

* ***Data Factory***: Este es el servicio ETL de Azure y será el orquestador del flujo de los datos, haciendo los movimientos y procesamiento necesarios para que la información esté donde se requiere y de la manera óptima. Algunas de las tareas de la pieza son:
  + Mediante procesos batch moverá la información que se convierta en datos fríos (históricos) al Data Lake y purgará el origen para hacer uso eficiente del espacio.
  + Podrá transformar los datos disponibles para generar la información de la manera en que sería más fácil para la visualización del cliente.
  + Realizará la carga inicial del sistema desde la información que disponible en la arquitectura on-premise.
  + Llevará la información mínima necesaria desde la infraestructura cloud al sistema on-premise para su funcionamiento como backup.
* ***Azure ML***: categorizará a todos los clientes del sistema con el fin de poder personalizar las campañas de marketing que se tiene como objetivo final.
* SQL Data Warehouse: Tendrá información agregada histórica y regularmente actualizada con KPI de análisis, exponiendo información valiosa con un lenguaje conocido por el usuario de negocio.
* La Visualización de la información podrá ser desde Power BI y la Suite de Pentaho. La primera opción es ideal para el autoservicio de reportes y la exploración de patrones mientras que con la segunda opción se podrá generar reportaría general para el usuario final para la operación sin costo de licencias.

**Arquitectura on-premise**

Esta sección está dedicada a la arquitectura de respaldo que se tendrá para el hipotético caso en que falle lo que se ha hecho en la nube. Se partió de la base de la posibilidad de aprovechar hardware que se encontraba sin utilizar en la empresa, esto sumado a la elección de todo el software libre posible; se pudo generar un desarrollo con bajo costo de licenciamiento para así no salirse del presupuesto asignado. Solo se cree conveniente, por ser una solución bien conocida por los usuarios finales, la opción de comprar una licencia de Microsoft SQL Server que viene con un motor de formularios y un servicio OLAP incluidos útil para el análisis de datos. Igualmente nos aprovechamos de que Microsoft soporta dicho servicio sobre Linux desde la versión 2017 y excluimos la necesidad de adquirir licencia para el sistema operativo de Windows Server.

A continuación, el diagrama de la arquitectura de respaldo y sus componentes:



* Un servidor Linux con dos contenedores donde en uno corre un aplicativo en java, python o Node.js con el código para la captura de los datos fuentes, datos colocados como mensajes en una cola Apache kafka. Se eligen servicios en contenedores, ya que es una unidad escalable y usa sus propios recursos logrando un aislamiento entre componentes.
* La cola kafka cumple la función de transportar los datos a la unidad que los procesa y evitar la pérdida de información en caso que esta última no responda.
* Apache Spark en una infraestructura Hadoop para generar los KPI básicos tales como el análisis de sentimiento para las redes sociales o total de compras por cliente entre otros. Además, que se recurrirá a librerías ML cuando se requiera aplicar Machine Learning a los datos. Los resultados quedarán en los nodos de almacenamiento HDFS ordenados en un modelo donde los datos procesados y los datos originales recolectados sin procesar se encuentren separados, igualmente y de manera separada, se deberá almacenar toda información que sea considerada datos fríos (que se consulten con poca frecuencia).
* Pentaho Data Integration orquesta todos los movimientos de información requeridos desde el Hadoop hacia la instancia SQL, o dentro del mismo HDFS en proceso batch cuando sea requerido, además que puede generar indicadores post-proceso útiles para el cientista de datos o los usuarios finales.
* MSSQL Data Warehouse: contendrá métricas resultantes agregadas y tendrá herramientas asociadas como Analysis Services y para la generación de cubos OLAP que permitan varias vistas de las métricas y el descubrimiento de patrones.
* Tanto el usuario final como el Cientista de Datos podrá disponer de reportes mediante Power BI desktop y la Suite de Pentaho además del servicio de Reporting Services que viene incluido con la licencia de MS SQL. Todos con distintas complejidades y distinto público objetivo.
* En el gráfico de arquitectura se puede observar un componente muy importante llamado VPN Gateway. Este es el nodo que comunica la infraestructura de la empresa con el desarrollo en la nube por donde pasarán los flujos de datos en ambos sentidos haciendo parecer que todo se encuentra en la misma RED, por lo mismo deberá tener configurada un buen control seguridad y accesos estrictamente necesarios.

## Exposición de la información

La exposición de los datos para su consumo por otros sistemas (los datos de sugerencias de compras para los portales de la empresa) será a través de API Rest, tanto para la infraestructura en la nube mediante servicios Lambda como para la infraestructura de respaldo usando lenguaje Node.JS en uno o varios contenedores. Estas interfaces son estándares fáciles de consumir, que abstraen la complejidad de cómo está almacenada la información. Se le deberá adicionar alguna capa de seguridad para un acceso controlado, existen muchos servicios de software libre disponibles para este fin; también se sugiere colocar un certificado de comunicación https de los que seguramente ya tiene la empresa para sus propios portales.

# Recursos y Planificación

Para cumplir con nuestro objetivo debemos contar con diversidad de profesionales, los cuales ayudaran en la implementación y puesta en marcha del proyecto. Los valores representados están en pesos chilenos.

* Jefe de Proyecto - $1.700.000
* Cientista de Datos (ya están disponibles y no entran en el presupuesto)
* Ingeniero de Operaciones para armar infraestructura - $1.800.000
* Ingeniero de Datos Senior - $2.400.000



# Plazos

Se estima que es posible realizar la migración completa en un plazo de 10 meses, desglosada de la siguiente forma:

## Procesamiento de datos

La primera parte, el procesamiento de datos, se descompone en los siguientes plazos:

* Recolección de información: 2 meses.
* Diseño de algoritmos: 3 meses.
* Diseño de API Rest de consumo de datos: 1 mes.

## Generación de reportes para el usuario

La visualización interactiva, la cual estará disponible a través de una plataforma Web, será desarrollada en un plazo de 4 meses, descompuesto en lo siguiente:

* Desarrollo Web: 3 meses.
* Etapa de pruebas: 1 mes.

# Entregables del Proyecto

Los entregables definidos en función de los objetivos del proyecto son los siguientes:

**Servicio**

* Servicio para análisis de comportamiento de clientes y generación de propuestas de productos para obtener compras.
* Reportes de KPI de funcionamiento de la plataforma.
* Reportes de KPI de resultados de la campaña.
* Interfaces API REST para las respuestas de las propuestas de compras a los clientes.

# Propuesta Económica

La oferta económica por este proyecto, considerando los recursos requeridos para su ejecución, se presenta en la siguiente tabla.

|  |  |
| --- | --- |
| Item | Valor UF |
| Desarrollo de Plataforma (6 meses) | **1350 UF** |
| Plataforma Interactiva (4 meses) | **900 UF** |

**Nota 1**: Los valores son presentados en UF y están exentos de IVA por la naturaleza del servicio.

**Detalle de aproximación de costos para servicios en la nube**

|  |  |  |  |
| --- | --- | --- | --- |
| **Service type** | **Region** | **Description** | **Estimated Cost** |
| Azure Databricks | East US 2 | Data Engineering Workload, Standard Tier, 2 D3V2 (4 vCPU(s), 14 GB RAM), 1 year reserved, 1.5 DBU x 730 Hours | $380,58 |
| Azure Databricks | East US 2 | Solo para recepción de datos | $292,98 |
| Event Hubs | East US 2 | Encolamiento de peticiones | $43,83 |
| Azure Data Lake Storage Gen1 | East US 2 | Commitment: 10 Overage GB, 100 Read Transactions, 100 Write Transactions | $2.906,00 |
| Storage Accounts | East US 2 | Block Blob Storage, Blob Storage, LRS Redundancy, Hot Access Tier, 6 TB Capacity - Pay as you go, 500,000 Write operations, 500,000 List and Create Container Operations, 500,000 Read operations, 100,000 Archive High Priority Read, 1 Other operations. 1,000 GB Data Retrieval, 1,000 GB Archive High Priority Retrieval, 1,000 GB Data Write | $925,05 |
| Azure Functions | East US 2 | 128 MB memory, 100 milliseconds execution time, 10,000 executions/mo | $0,00 |
| Azure SQL Database | East US 2 | Single Database, vCore Purchase Model, General Purpose Tier, Serverless, Gen 5, 4 Billed vCores, 4,000 GB Storage, 4000 GB Backup Storage | $1.682,09 |
| VPN Gateway | East US 2 | Tipo VpnGw5 hasta 10GB/seg. Gigabytes de transferencias asociados a la recepcion diaria de datos | $3.100,00 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Support** | | $0,00 | |
| **Licensing Program** | | **Microsoft Online Services Agreement** | |
| **Monthly Total** | | **$9.286,69** | |
| **Annual Total** | | **$111.440,28** | |
| **Software On Premise** |  | |  |
| MS SQL Server 2019 Standar | Soportado sobre Ubuntu 16 o superior, no requiere costo de licencia en SO. Costo una sola vez | | $14.872,00 |

# Anexos

**Carta Gantt**

