**PRE – WORK**

En el mundo moderno, los datos son el recurso más valioso. Ya sea que estés gestionando un negocio, trabajando en marketing o simplemente tratando de comprender tu mundo, los datos pueden ofrecerte valiosas perspectivas.

Pero ¿cómo podemos convertir esos datos en información significativa y visualmente impactante? La respuesta está en Power BI.

* **Transforma los Datos en Historias Visuales**: Power BI te permite transformar datos crudos en visualizaciones atractivas y comprensibles. Puedes crear gráficos, tablas y mapas interactivos para comunicar tus ideas de manera efectiva.
* **Toma Decisiones Basadas en Datos:** Al analizar datos en tiempo real, puedes tomar decisiones informadas y estratégicas. Power BI te ayuda a identificar patrones, tendencias y oportunidades, lo que te permite anticipar cambios y actuar proactivamente.
* **Conecta Múltiples Fuentes de Datos:** Power BI puede integrarse con diversas fuentes de datos, incluidas bases de datos, servicios en la nube y archivos locales. Esto significa que puedes analizar datos dispersos y obtener una visión completa de tu negocio o proyecto.
* **Colaboración y Compartir Información:** Power BI te permite colaborar en tiempo real con colegas y compartir informes interactivos y paneles de control. La colaboración se vuelve fácil, lo que facilita el trabajo en equipo y la toma de decisiones conjunta.

1. **DICCIONARIO DE CONCEPTOS IT**
2. **IDE**

**¿Qué es el IDE?:** IDE es la abreviatura del inglés de Integrated Development Environment, o lo que es lo mismo, un entorno de desarrollo integrado.

**¿Para qué usamos el IDE?:** El IDE es una de las herramientas principales de cualquier programador, ya que integra diferentes funciones en una sola interfaz, que facilitan y agilizan el proceso de programación.

El IDE es una herramienta indispensable en el mundo de la programación. Es un software que reúne varias funciones que facilitan el proceso de programación. A continuación, se presentan algunas de las características que se pueden encontrar en un IDE:

* **Editor de código integrado**: permite escribir código de manera sencilla y eficiente con características como resaltado de sintaxis, corrección automática, autocompletado, entre otros.
* **Depurador integrado:** es una herramienta que permite detectar y solucionar errores en el código de manera más sencilla.
* **Gestor de proyectos integrado:** facilita la organización y gestión de los archivos y recursos que se utilizan en un proyecto de programación.
* **Compilador integrado:** permite compilar el código de manera más sencilla y rápida.
* **Integración con sistemas de control de versiones:** facilita el trabajo en equipo y el control de cambios en el código fuente.
* **Asistencia en la documentación**: permite la creación y edición de documentación, como comentarios en el código, manuales de usuario, entre otros.
* **Integración con librerías y frameworks**: permite utilizar librerías y frameworks de manera más sencilla y eficiente. En resumen, el IDE es una herramienta que permite a los programadores trabajar de manera más eficiente y productiva, al reunir varias funciones en una sola interfaz de usuario.

1. **BASES DE DATOS**

* ¿Qué son realmente las bases de datos?

Una base de datos es un sistema diseñado para almacenar, organizar y recuperar o consultar grandes cantidades de información de manera eficiente.

El formato más habitual es una estructura de filas y columnas, organizadas en tablas, que permiten aumentar la eficacia del procesamiento y consulta de datos.

* ¿Qué podemos hacer con una base de datos y cómo?

Se puede acceder, gestionar, modificar, actualizar, añadir, eliminar, controlar y organizar los datos.

Para poder realizar estas acciones se requiere un lenguaje específico para escribir y consultar estos datos.

* Pero si son tablas... ¿por qué no usar una hoja de cálculo?

Aunque ambas son modos de almacenaje y procesado de datos, las **hojas de cálculo** son perfectas para un **único usuario** o un pequeño número de usuarios que no requieran una manipulación de datos compleja o gran cantidad de datos.

Las **bases de datos** nos permiten tener recopilaciones más grandes de **información** o de forma **masiva**, además de permitir el acceso a **muchos usuarios de forma simultánea** y consultar los datos de forma más rápida.

**TIPOS DE BASES DE DATOS**

**1.Bases de datos relacionales:** Estas bases de datos utilizan tablas para organizar los datos en filas y columnas.

Para realizar las operaciones sobre los datos (crear un registro, modificar, añadir una tabla, etc.) se utiliza un lenguaje de consulta estructurado que denominamos SQL o Structured Query Lenguage.

Además, establecen relaciones entre las tablas para evitar la redundancia y mantener la integridad de los datos.

Para el manejo de bases de datos relacionales existen diferentes tecnologías como MySQL, PostreSQL, SQL Server, Oracle o SQLite.

Como ventajas encontramos:

* Estructura organizada y fácil de entender.
* Soporte para transacciones ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad).
* Relaciones establecidas para evitar la redundancia.

Y como desventajas:

* Pueden volverse complejas en estructuras de datos grandes.
* No son ideales para datos no estructurados o semi-estructurados.

**2.Bases de datos NoSQL o no estructuradas:** Estas bases de datos no siguen un esquema fijo como las bases de datos relacionales. Pueden manejar datos no estructurados o semi-estructurados.

Existen diferentes tecnologías para bases de datos no relacionales, como MongoDB, Cassandra o Redis.

Como ventajas encontramos:

* No siguen un esquema fijo como las bases de datos relacionales.
* Pueden manejar datos no estructurados o semi-estructurados.
* Escalan horizontalmente, lo que significa que se pueden agregar más servidores para manejar grandes volúmenes de datos.

Como desventajas encontramos:

* Falta de estandarización en comparación con las bases de datos relacionales.
* Algunas operaciones pueden ser más complejas que en las bases de datos relacionales.

**3.Bases de datos en memoria**: Las bases de datos en memoria se caracterizan porque almacenan los datos en la memoria principal (RAM) en lugar de en discos. Esto permite un acceso ultra rápido a los datos debido a la velocidad de la memoria.

Entre las tecnologías de este tipo de bases de datos encontramos Redis o Memcached. Aunque Redis también se clasifica como NoSQL, pero es ampliamente utilizado como base de datos en memoria.

Como ventajas encontramos:

* Acceso extremadamente rápido a los datos.
* Ideal para aplicaciones que requieren alta velocidad y baja latencia (el tiempo que tarda en completarse una operación). Esto es especialmente útil para aplicaciones que requieren respuestas instantáneas, como sistemas de tiempo real y aplicaciones web interactivas.
* Las bases de datos en memoria son ideales para datos temporales que no necesitan ser almacenados a largo plazo, como datos de sesión en aplicaciones web.

Y como desventajas:

* Los datos se pierden si el sistema se apaga ya que no se almacenan permanentemente en disco.
* Los costos de almacenamiento en memoria son generalmente más altos.

**4.Bases de datos de grafos:** Están diseñadas para manejar relaciones complejas y conexiones entre datos. Utilizan nodos y relaciones para representar datos y sus interconexiones.

Las tecnologías principales para gestionar este tipo de base de datos son Neo4j o ArangoDB.

Como ventajas encontramos:

* Ideales para representar relaciones complejas y consultas relacionadas.
* Rendimiento rápido para consultas que involucran relaciones complejas.

Y como desventajas:

* No son eficientes para datos que no tienen relaciones significativas.
* No son tan comunes como las bases de datos relacionales o NoSQL, por lo que puede haber menos recursos y soporte disponibles.

**5.Bases de datos de series temporales:** Estas bases de datos están optimizadas para almacenar y recuperar datos organizados por marcas de tiempo. Son comunes en aplicaciones que gestionan grandes volúmenes de datos de series temporales como sensores, registros de eventos, etc.

Entre las tecnologías para estas bases de datos están InfluxDB o OpenTSDB.

Como ventajas encontramos:

* Eficientes para almacenar grandes volúmenes de datos de series temporales.
* Ofrecen funcionalidades específicas para análisis y agregación de datos temporales.

Y como desventajas:

* No son tan versátiles como las bases de datos relacionales o NoSQL para otros tipos de datos.

1. **SQL (Estructured Query Language)**

SQL (Structured Query Language) es un lenguaje de programación diseñado para gestionar bases de datos relacionales. Se utiliza para crear, modificar, gestionar y consultar bases de datos. Es el "idioma" que permite a las personas comunicarse con las bases de datos y realizar diversas operaciones en los datos almacenados en ellas.

**Las principales características de SQL son:**

* Lenguaje Declarativo: SQL es un lenguaje declarativo, lo que significa que los usuarios le dicen al sistema qué resultados desean, y el sistema se encarga de cómo obtener esos resultados, sin necesidad de especificar los pasos detallados. SQL te permite hacer preguntas a una base de datos. Puedes preguntar cosas como "¿Cuántos productos tenemos en stock?" o "¿Cuál es el nombre del cliente con el ID número 123?".
* Independencia del sistema: SQL es independiente del sistema operativo y de la arquitectura del hardware. Puede ejecutarse en diferentes sistemas, lo que hace que sea portátil
* Estructurado y eficiente: SQL permite definir la estructura de los datos y realizar operaciones de manera eficiente, incluso en grandes conjuntos de datos. SQL tiene una sintaxis estructurada que se parece mucho al inglés. Esto hace que sea relativamente fácil de aprender y entender para las personas, incluso si no son expertas en programación.
* Manejo de relaciones: SQL permite establecer relaciones entre diferentes tablas de una base de datos relacional. Estas relaciones aseguran la integridad de los datos y evitan la redundancia. Está diseñado específicamente para bases de datos relaciones, que son como hojas de cálculo gigantes con tablas que están conectadas entre sí.

A través de SQL podemos realizar consultas en las bases de datos para buscar o recuperar datos específicos, insertar nuevos datos, actualizarlos o modificarlos, así como eliminarlos. También nos permite hacer consultas cruzando los datos de diferentes tablas a través de diferentes relaciones.

Las consultas son instrucciones escritas en SQL para realizar operaciones en los datos. Las consultas pueden ser complejas, pero también ser tan simples como preguntas “¿Dónde está el dato que tiene ese nombre?

1. **SERVER**

**¿Qué es un servidor o server?**

Es un aparato informático que **almacena, distribuye y suministra información.** Los servidores funcionan basándose en el modelo “cliente-servidor”.

* El cliente puede ser tanto un ordenador como una aplicación que requiere información del servidor para funcionar.
* Un servidor ofrecerá la información demandada por el cliente siempre y cuando el cliente esté autorizado.
* Los servidores pueden ser **físicos o virtuales.**

Un servidor es una aplicación o un dispositivo que proporciona servicios a otras aplicaciones o dispositivos. En el mundo de la programación, los servidores se utilizan para varias tareas, tales como:

* Almacenamiento y gestión de datos: los servidores se utilizan para almacenar y gestionar grandes cantidades de datos, para que puedan ser accedidos y utilizados por otras aplicaciones o dispositivos.
* Hosting de sitios web: los servidores web alojan los sitios web, permitiendo a los usuarios acceder a ellos a través de Internet.
* Procesamiento de solicitudes: los servidores procesan las solicitudes enviadas por las aplicaciones o dispositivos y proporcionan respuestas.
* Compartir recursos: los servidores pueden ser utilizados para compartir recursos, como impresoras o archivos, entre diferentes dispositivos.

En general, los servidores son esenciales en el mundo de la programación, ya que permiten que las aplicaciones y dispositivos se comuniquen y compartan información de manera eficiente

**Algunas de las características principales de un servidor son:**

* Potencia de procesamiento: Los servidores suelen tener procesadores potentes para manejas múltiples solicitudes simultáneamente.
* Almacenamiento: Tienen grandes capacidades de almacenamiento para guardar datos, archivos y aplicaciones que serán compartidos con los clientes.
* Conectividad: Los servidores están diseñados con múltiples interfaces para conectarse eficientemente con otros dispositivos en la red.
* Estabilidad: Los servidores están optimizados para funcionar de forma continua y estable, a menudo funcionando 24/7 para garantizar la disponibilidad de los servicios.
* Seguridad: Tienen medidas de seguridad avanzadas para proteger los datos y prevenir accesos no autorizados.

Los servidores tendrán diferentes características en función de su uso. Por ejemplo, encontramos servidores web, de correo electrónico, de bases de datos, de aplicaciones o de Porxy y VPN. Los servidores de Proxy y VPN facilitan la navegación segura y anónima en internet, redirigiendo el tráfico a través de conexiones seguras.

En resumen, un servidor es una computadora especializada que proporciona servicios y recursos a otras computadoras en una red. Su función es esencial para el funcionamiento de internet y muchas aplicaciones en línea que usamos todos los días.

1. **API (Server y DB)**

API es la abreviatura del inglés **Application Programming Interface**, o dicho de otra manera, el **intermediario** entre el usuario y diferentes programas de software, que nos permite la comunicación con ellos, permitiendo intercambiar información y funcionalidad.

*Ejemplo: Si estamos en un restaurante, nosotros le pedimos la comida al camarero (API), con el pedido él se iría a la cocina que sería el SERVER. El cliente no se comunica con el Server, es el API quien traslada la solicitud del cliente. El Server (la cocina), se comunicaría con la base de datos (Supermercado). La base de datos reporta los datos de nuevo al server, que es quien transforma esos datos y lo vuelve a reportar a la API para que éste le de al cliente lo que ha solicitado.*

*CLIENTE 🡪 API 🡪 SERVER🡪 BASE DE DATOS 🡪 SERVER 🡪 API 🡪 CLIENTE*

1. **CONTEXTO DE DATA**
2. **¿QUÉ ES BUSINESS INTELLIGENCE?**

**Business intelligence (BI)** se refiere al conjunto de tecnologías, procesos y herramientas que permiten a las empresas recopilar, almacenar, analizar y transformar datos en información significativa y útil para la toma de decisiones empresariales.

En términos sencillos, BI es el proceso de convertir datos en conocimiento para ayudar a las empresas a tomar decisiones informadas.

Dentro del ciclo de vida del dato, que es el proceso completo de gestión de datos desde su creación hasta su eliminación, Business Intelligence ocupa principalmente la fase de “Análisis y Visualización”.

Rectángulo

El contenido generado por IA puede ser incorrecto.

Los datos, han debido extraerse, transformarse y cargarse previamente en un entorno para su consumo.

Esta es la etapa del ciclo de vida del dato que denominamos **ETL (Extract, Transform and Load)**, en la que se extraen y unifican los datos de todas las fuentes de origen.

Una vez transformados los datos se guardan en lo que llamamos **almacenes de datos o data warehouse,** que es una base de datos optimizada para el análisis y la presentación de informes, donde se almacenan los datos consolidados y limpios de diversas fuentes desde donde consumirán los datos los científicos de datos y analistas de datos para su análisis, reporting, data mining o análisis predictivos.

En la fase de **visualización y presentación de datos,** que es la que nos ocupa, los datos recopilados son analizados, procesados y presentados de manera que los usuarios puedan entenderlos fácilmente.

Esto implica la creación de informes, paneles de control, gráficos y visualizaciones interactivas que permiten a los usuarios hacer preguntas especificas sobre los datos y obtener respuestas significativas para tomar decisiones empresariales fundamentadas.

1. **DATOS, INFORMACIÓN Y CONOCIMIENTO**

Vivimos en un mundo rodeados de datos que generamos nosotros mismos. Cada vez que hacemos una compra online, navegamos por internet, usamos el cajero de un banco o pagamos gasolina, estamos generando datos.

Sin embargo, los datos en sí mismos no nos dan la información que necesitamos para extraer **conclusiones** y, por tanto, tomar **decisiones.** De ahí la importación de la existencia de la analítica del datos, que nos permite extraer **conocimiento** de éstos en última instancia para poder tomar la decisión correcta en el momento adecuado.

1. **TIPOS DE ANALITICA**

La analítica del dato nos plantea la primera pregunta. ¿Qué tipo de problemática es la que quiero resolver? Para dar respuesta a esta pregunta, nos apoyamos en **distintos de analítica** que nos ayudan a entender mejor la realidad y/o qué acciones llevar a cabo.

Encontramos **tres tipos de analítica** y cada una de ellas responde a una pregunta de nuestra realidad.

* **La analítica descriptiva** responde a la pregunta **¿Qué ha ocurrido?** Es decir, utiliza datos del pasado que analizamos mediante distintos tipos de gráficos como pastel, barra o histogramas para detectar patrones que respondan a esa pregunta y refleje lo que ha ocurrido en un periodo temporal en el pasado.
* **La analítica prescriptiva** responde a la pregunta **¿Qué debemos hacer?** Es decir, utiliza datos del pasado que analizamos con técnicas distintas, como análisis de grados, simulaciones, procesamiento complejo de eventos o redes neuronales. Aquí es donde entraría en juego la ciencia de datos e Inteligencia Artificial (por ejemplo, los motores de recomendación de productos, machine learning…)

Ambas utilizan las mismas fuentes de datos, del pasado, pero utilizan distintas técnicas de análisis que responden a preguntas totalmente distintas del negocio.

* **La analítica predictiva** responde a la pregunta **¿Qué ocurrirá en el futuro?** Es decir, predecir qué ocurrirá en el futuro para tomar una decisión en el presente.

Aunque tanto la analítica descriptiva como la prescriptiva utilizan datos del pasado, la forma en la que se realizan los análisis son diferentes ya que responden a dos preguntas distintas.

1. **MEDIDAS, DIMENSIONES Y KPI’s**

En el contexto de Business Intelligence (BI), las medidas, dimensiones y KPI’s son conceptos fundamentales que ayudan a comprender y analizar datos de manera significativa.

* **MEDIDAS:** Son valores cuantificables y específicos que se pueden analizar. Son los datos numéricos sobre los que se realiza algún tipo de cálculo, como suma, promedio o conteo. Las medidas son las base para realizar análisis y tomar decisiones.

*En un conjunto de datos de ventas, la medida podría ser la “Cantidad Vendida” o el “Total de Ventas”.*

* **DIMENSIONES:** Son los atributos o categorías que proporcionan contexto a las medidas. Son las maneras en que organizamos, filtramos y analizamos los datos. Las dimensiones son descriptivas y no numéricas. Ayudan a responder preguntas como “¿Qué?”, “¿Dónde?”, “¿Cuándo?*” y ¿Cómo?”.*

Comprender las dimensiones permite organizar y analizar las medidas de manera mas significativa, proporcionando insights sobre diferentes aspectos del negocio.

*En el mismo conjunto de datos de ventas, las dimensiones podrían ser “Producto”, “Región Geográfica”, “Periodo de Tiempo” o “Cana de Ventas”. Estas dimensiones proporcionan contexto a las medidas de ventas y permiten analizar las ventas según diferentes criterios*

* **KPIs (Indicadores Clave de Rendimiento):** Son medidas específicas que se utilizan para evaluar el rendimiento de una organización o un proceso en relación con sus objetivos. Los KPIs ayudan a determinar si una empresa está alcanzando sus metas estratégicas.

*Para una empresa de comercio electrónico, un KPI podría ser la “Tasa de Conversión”, que mide la proporción de visitantes del sitio web que realizan una compra. Si el objetivo es aumentar las ventas en línea, la empresa podría establecer la meta de aumentar la tasa de conversión del 2% al 3%. La “Tasa de Conversión” es un KPI crucial para medir el rendimiento y el progreso hacia este objetivo específico.*

La comprensión profunda de estos tres aspectos es esencial para transformar los datos en información significativa, que a su vez impulsa la toma de decisiones fundamentadas y la mejora continua de las operaciones empresariales.

1. **MODELADO DE DATOS**

El **modelado de datos** en el contexto de Business Intelligence (BI) es el **proceso de diseño y estructuración de los datos** para facilitar su almacenamiento, gestión, análisis y presentación.

Es fundamental para garantizar que los datos se utilicen de manera efectiva y se traduzcan en información valiosa para la toma de decisiones.

Existen diferentes tipos de modelado de datos:

* **Modelado dimensional:** Se centra en las **dimensiones y medidas.** Utiliza tablas de hechos para las medidas y tablas de dimensiones para describir las características de los datos (como productos, clientes y fechas).

Es comúnmente utilizado en almacenes de datos y es ideal para análisis y creación de informes.

* **Modelado relacional:** Se basa en el modelo **relacional de las bases de datos**, donde los datos se organizan en tablas con relaciones entre ellas. Es útil para sistemas de transacciones y puede implicar el uso de claves primarias y externas para definir las relaciones entre las tablas.
* **Modelado multidimensional:** Se enfoca en el **análisis de datos en múltiples dimensiones.** Es especialmente útil para explorar grandes volúmenes de datos desde múltiples perspectivas.

Para abordar de forma exitosa el modelado de datos tendremos que comprender las necesidades y requerimientos de los usuarios finales, lo que ayudará a determinar qué datos son importantes y como deben estructurarse.

A continuación, recopilaremos los datos necesarios, de una o varias fuentes, que limpiaremos según sea necesario de acuerdo con los requisitos y pasaremos a realizar el diseño del modelo.

En el diseño del modelo estructuraremos los datos en tablas y definiremos las relaciones entre ellas. En el modelo dimensional, esto implica identificar las tablas de hechos y las tablas de dimensiones, así como establecer las claves primarias y foráneas.

Es de vital importancia un modelo de datos correcto, ya que tiene un alto impacto en el rendimiento de nuestra aplicación de análisis de datos.

Un modelo bien diseñado facilita el análisis, mejora el rendimiento y contribuye significativamente al existo de las iniciativas de BI en cualquier organización.

Una clave primaria es un atributo o conjunto de atributos en una tabla de base de datos que identifica de forma única cada fila en esa tabla. En otras palabras, es un valor único para cada registro en la tabla y se utiliza para garantizar que no haya duplicados ni registros nulos en esa tabla.

Tabla

El contenido generado por IA puede ser incorrecto.

*En una tabla de “Clientes”, el numero de identificación del cliente podría ser la clave primaria. Cada número de identificación del cliente en esa tabla sería único para un cliente específico*.

Una clave foránea es un atributo en una tabla que se relaciona con la clave primaria de otra tabla, lo que permite la creación de relaciones y consultas complejas entre las tablas de una base de datos.

*Diagrama

El contenido generado por IA puede ser incorrecto.*

*Supongamos que tenemos dos tablas, una para “Pedidos” y otra para “Clientes”.*

*En la tabla de “Pedidos”, el número de identificación del cliente seria una clave foránea. Esta clave foránea establece una relación entre los pedidos y los clientes, permitiendo que cada pedido esté asociado a un cliente especifico mediante su número de identificación único en la tabla de “Clientes”.*

1. **ROLES**

Debido a la magnitud de los proyectos de Data & Analytics encontramos **roles muy segmentados** con tareas definidas. Entender esto te ayudará a desenvolverte mejor y saber de antemano qué papel desempeñas en un proyecto de Data & Analytics.

*Diagrama

El contenido generado por IA puede ser incorrecto.*

Agrupamos los roles en tres grandes grupos según su función:

1. **Roles técnicos**

* Arquitecto de datos

Es la persona que diseña y construye el sistema total. Es la persona más técnica del equipo. Es un rol que toma decisiones tales como, la tecnología que se va a emplear, cómo se va a emplear, las practicas para que el resto de las personas ejecuten y programen de la misma forma.

*Trabaja dentro de los bloques de: Transformación de datos, Data Hub y BI & Analítica.*

* Ingeniero de datos

Es uno de los roles más demandados en los proyectos ya que es dónde más trabajo hay. Son los encargados de sacar información de los sistemas origen (Fuentes de datos) y lo transforma en información (Data Hub). Es un programador.

*Trabaja dentro de los bloques de: Transformación de datos y Data Hub.*

* Ingeniero de plataforma

Es la figura encargada de mantener todo el diseño y todo el sistema que haya diseñado el arquitecto de datos. Simplifica el trabajo de los ingenieros de datos. *Trabaja dentro de los bloques de: Trasformación de datos, Data Hub, BI & Análitica.*

1. **Roles Data & Analytics**

* Científico de datos

Su misión principal es hacer un software que prédica el futuro (Analítica predictiva) o que prescriba el futuro (Analítica Prescriptiva). Es un rol muy difícil de encontrar porque tiene que ser experto en estadística, tiene que ser técnico y tiene que conocer perfectamente el negocio donde trabaja.

Diagrama, Diagrama de Venn

El contenido generado por IA puede ser incorrecto.

* Analista de negocio

Trabaja haciendo analítica descriptiva. Tiene una pregunta y tiene que buscar la respuesta. La pregunta más importante: cómo va el negocio. Es la persona que consume los datos que ha transformado el ingeniero de datos generando información (Data Hub) y que se conecta a través de las herramientas de visualización tipo PBI y con su conocimiento del negocio es capaz de realizar la pregunta correcta para responder a lo que le están pidiendo sus interlocutores: managers, directores, personas del comité. Esta persona da soporte al científico de datos sobre el conocimiento del negocio.

1. **Roles Híbridos**

* Data Steward

Es la persona que se encarga desde el punto de vista de gobierno del dato de fijar y velar porque todos los criterios sean comunes.

* CDO (Chief Digital Officer)

Es el director del departamento de Data Analytics. Es el que define la estrategia. Es un perfil de gestión y responde por todo el equipo.

* DPO (Data Protection Officer)

Esta figura normalmente es un abogado. Es la persona que vela y que responde por las brechas de seguridad de datos. Es la encargada de firmar que toda la información recolectada por el ingeniero de datos y que posteriormente va a transformar es exactamente la estricta y puede justificarla de forma legal para ser consumida por un científico de datos o un analista de negocio.

1. **TECNOLOGÍA VS LENGUAJE**

El **lenguaje** de programación es el idioma con el que nos comunicamos con una máquina y nos permite darle instrucciones. Algunos **tipos de lenguaje** comunes usados en el análisis de datos son:

* SQL para consulta de bases de datos
* Python para manipulación de datos, análisis estadístico y machine learning
* Otros lenguajes como Java o C++

La **tecnología** son las herramientas y plataformas utilizadas, en el caso del análisis de dato, para almacenar, procesar y analizar datos. Algunos **tipos de tecnología** habituales en el análisis de datos son:

* Bases de datos como MySQL, PostgreSQL, Oracle
* Sistemas de procesamiento de big data, como Hadoop y Spark
* Plataformas de análisis como Tableau o PowerBI
* Plataformas de machine learning como TensorFlow o PyTorch

El lenguaje y la tecnología trabajar juntos para permitir un análisis de datos eficaz.

1. **GOBIERNO DEL DATO**

Todos sabemos que, aunque los datos constituyen información objetiva, la interpretación de éstos puede transformarse en subjetiva o sesgada.

Para evitarlo, necesitamos tener una **definición común** de qué son los datos y mantener una **trazabilidad** de cómo hemos obtenido el dato. De ahí surge la necesidad del **Gobierno del dato que regule** y dicte unas directrices que aseguren la **veracidad del dato.**

El gobierno de dato trata de poner **criterios comunes** en la empresa para que se mida la información de la misma forma y bajo los mismos criterios. También tiene la función y la responsabilidad de velar por la calidad del dato (asegurarse de que toda la recopilación de datos es correcta para que la información que se ha sacado sea veraz).

También se encargan de poner las herramientas necesarias para que la recopilación de datos no contenga errores.

1. **PRIVACIDAD & DATA PROTECTION**

¿Cómo se gestiona la recogida de datos y cuándo y cómo pueden usarse de forma que no vulnere mis derechos como individuo?

La **privacidad del dato** responde a esta pregunta estableciendo un marco legal que determina estas condiciones. Hay que tener en cuenta que no se aplica de la misma manera en todo el mundo, ni su aplicación es siempre perfecta.

1. **CONCEPTOS PROGRAMACION**
2. **VARIABLES & TIPOS DE DATOS**

En programación, una variable es un **espacio de memoria reservado para almacenar** un valor o una **información que puede ser modificada durante la ejecución del programa**. Es decir, es un **contenedor que permite guardar datos** temporalmente, para que puedan ser **utilizados en diferentes partes del programa.**

**Las variables** se utilizan para **almacenar** diferentes tipos de **datos**, como números, texto, booleanos, entre otros. Las variables tienen un **nombre que las identifica y permiten acceder a su contenido.** Este **nombre** debe ser **único y descriptivo,** para facilitar la compresión del código.

Las variables son fundamentales en la programación, ya que permiten crear programas dinámicos y adaptables. Al utilizar variables, es posible realizar operaciones matemáticas, concatenar cadenas de texto, tomar decisiones condicionales, entre otras acciones que hacen que el programa se comporte de diferentes maneras según las necesidades del usuario.

Tipos de datos:

Los tipos de datos son la forma en la que los programas representan la información. A continuación, se describen los tipos de datos más comunes:

* Números enteros: Se utilizan para representar números enteros. Pueden ser positivos o negativos, y tienen un tamaño determinado según el lenguaje de programación
* Números de punto flotante: Se utilizan para representar números con decimales. Puedes ser positivos o negativos, y tienen un tamaño determinado según el lengua de programación.
* Cadenas de texto: Se utilizan para representar texto. Pueden contener letras, numero y otros caracteres, y su longitud puede variar según el lenguaje de programación.
* Booleanos: Se utilizan para representar valores de verdad. Pueden ser verdadero o falso.
* Arreglos /Arrays: Se utilizan para representar conjuntos de datos. Pueden contener números, texto, o cualquier otro tipo de dato, y su longitud puede variar según el lenguaje de programación
* Objetos: Se utilizan para representar entidades o conceptos. Pueden contener atributos y métodos, y su estructura y funcionalidad depende del lenguaje de programación utilizado.

1. **OPERADORES**

Los operadores son símbolos o palabras clave que nos permiten realizar operaciones con distintos valores.

Algunos tipos de operaciones que podemos hacer son:

* **Aritméticas**: Son los utilizados para realizar operaciones matemáticas como suma, resta, multiplicación, división, modulo y exponenciación. (+,-,X,/….)
* **Comparación:** Son utilizados para comparar dos valores y devolver verdadero o falso como resultado. Mayor que (>), menor que (<), igual a (==), mayo o igual que (>=) y menor o igual que (<=)
* **Lógicos**: Son los utilizados para realizar operaciones lógicas (NOT, OR, AND). Estos operadores se utilizan principalmente en las estructuras de control de flujo
* **Asignación**: Son utilizados para asignar un calor a una variable. (=)
* **Incremento o Decremento**: Son utilizados para incrementar o disminuir el valor de una variable en una cantidad determinada. (++, --)
* **Operadores Bit a Bit**: Son utilizados para realizar operaciones a nivel de bits en los valores. Ejemplos de estos operadores son el operados AND bit a bit (&), el operador OR BIT A BIT (|) y el operador XOR bit a bit (^).

1. **CONDICIONALES**

Los condicionales son **fragmentos de código** que permiten **establecer una acción** que se llevará a cabo o no **en función de si se cumple una condición dada**. Es decir, una forma de establecer distintos caminos en nuestro código dadas unas condiciones.

Algunas de las **características** de los condicionales son:

* Permiten evaluar una o varias condiciones y ejecutar diferentes acciones según el resultado
* Pueden ser simples o complejos, dependiendo de la cantidad de condiciones que se evalúen.
* Son esenciales para la lógica y el control de flujo en la mayoría de los programas.
* Ayudan a optimizar el código, evitando la ejecución de instrucciones innecesarias.

Tipos de condicionales:

* **If-Then** 🡪 Este tipo de condicional evalúa una condición y ejecuta una acción si se cumple. Si la condición no se cumple, la acción no se ejecuta.
* **If-Then-Else** 🡪 Este tipo de condicional evalúa una condición y ejecuta una acción si se cumple. Si la condición no se cumple, se ejecuta otra acción.
* **Switch** 🡪 Este tipo de condicional evalúa una variable o expresión y ejecuta diferentes acciones según el valor. Es una alternativa al uso de múltiples condicionales if-the-else
* **Temario** 🡪 Este tipo de condicional es una forma abreviada de escribir una instrucción if-the-else en una sola línea.
* **Try-Catch** 🡪 Este tipo de condicional se utiliza para manejar excepciones o errores que puedan ocurrir durante la ejecución del programa. El bloque try contiene el código que puede generar la excepción y el bloque catch se encarga de manejarla.

1. **BUCLES**

Los bucles o loops son una **estructura** que nos permite **repetir una acción** o bloque de códigos **varias veces**. Es una herramienta esencial para **automatizar tareas** y procesar grandes cantidades de datos de manera eficiente.

Algunas **características** de los bucles son:

* Iteración: Los bucles permiten repetir una serie de instrucción un número determinado de veces hasta que se cumpla una determinada condición.
* Condiciones de finalización: Los bucles pueden ser diseñados para terminar cuando se cumpla una condición especifica, como alcanzar un valor específico o cuando una variable alcanza cierto valor.
* Control de flujo: Los bucles permiten controlar el flujo de ejecución de un programa, permitiendo el salto a diferentes secciones de código según ciertas condiciones.

Existen diferentes tipos de bucles, entre los cuales destacan:

* **Bucle while** 🡪 Es una estructura de control que permite repetir un bloque de código siempre y cuando se cumpla una determinada condición.
* **Bucle for** 🡪 Es una estructura de control que permite repetir un bloque de código un número determinado de veces. Este tipo de bucle se utiliza principalmente cuando se conoce la cantidad exacta de interacciones que se deben realizar.
* **Bucle do-while** 🡪 Es una estructura de control similar al bucle while, con la diferencia de que se ejecuta al menos una vez antes de comprobar la condición de finalización.

1. **FUNCIONES**

En el mundo de la programación, las funciones son **bloques de código** que se pueden llamar en cualquier momento para **realizar una tarea especifica**. Estas funciones pueden tomar argumentos como entrada y devolver un resultado como salida. Algunos de los conceptos clave asociados con las funciones en programación incluyen:

* **Modularidad:** Capacidad de dividir un programa en funciones más pequeñas y manejables.
* **Reutilización de código:** La capacidad de utilizar la misma función en diferentes partes de un programa o en diferentes programas.
* **Abstracción:** La capacidad de ocultar los detalles de implementación de una función y proporcionar una interfaz simple para su uso.

Además, las funciones pueden ser definidas por el usuario o pueden estar predefinidas en un lenguaje de programación especifico. Pueden ser llamadas dentro de otros programas, lo que permite la creación de programas más complejos y la automatización de tareas repetitivas.

En resumen, las funciones son una herramienta fundamental en el mundo de la programación que permite la creación de programas más modulares, reutilizables y abstractos.