

## 1. INICIACIÓN A LA PREPARACIÓN DE DATOS INTELIGENTE CON POWER QUERY

- a. Fundamentos en Preparación de Datos y Teoría Esencial
  - i. Conceptos de emergencia
    1. Proceso ETL/ELT
    2. Consulta: Bloque de trabajo (tres perspectivas)
    3. ¿Qué es Power Query? y ¿Qué es el Lenguaje M?
    4. Objetivos de la preparación de datos
      - a. El formato tabular óptimo
      - b. Modelado de datos
    5. Plegado de Consultas (Folding/Unfolding)
    6. Repositorio de Power Query Zone
      - a. Explorando el repositorio
      - b. Utilización para obtener tabla de Feriados/Festivos
  - ii. Acerca de Power Query: Una tecnología Cross Platform
    1. Power Query de escritorio (Desktop)
      - a. Interfaz y aplicaciones que cuenta con P.Q
      - b. Características (Comunes y propias)
    2. Power Query en la nube (Online)
      - a. Interfaz y servicios que cuentan con P.Q
      - b. Características (Comunes y propias)
    3. Power Query de escritorio Vs Power Query en la nube
    4. Power Query como un todo
  - iii. Extracción de datos
    1. Conectores nativos
      - a. Excel, CSV, SQL, Web, Dataverse, ...
      - b. Sobre Flujos de Datos (Data flows)
        - i. Flujos de Datos de Power BI
        - ii. Flujos de Datos de Power Platform
        - iii. Consumir flujos de Datos en Excel
    2. Datos from "scratch"
    3. Sobre conectores personalizados

- iv. Limpieza de Datos de Primer Grado: Técnicas de Depuración
  - 1. Entendimiento de datos y buenas prácticas
    - a. Perfil de datos: utilización, consideraciones y precauciones
    - b. Buenas prácticas
  - 2. Transformación de datos con técnicas de depuración
    - a. Reducción
      - i. Exclusión de filas
      - ii. Promover encabezados
      - iii. Quitar Columnas
        - 1. Métodos estáticos
        - 2. Método dinámico con ***Table.RemoveEmptyColumns***
    - b. Formateo de Datos
      - i. División de columnas
        - 1. Métodos clásicos (delimitador, caracteres, ect.)
        - 2. Método por fila
        - 3. Métodos avanzados, ejemplo de división de nombres compuestos con ***Splitter.SplitByCompoundName***
      - ii. Reemplazar valores
      - iii. Transformación a nivel de columnas
        - 1. Tipo texto
        - 2. Tipo numérico
        - 3. Tipo fecha
      - iv. Asignación de nombres
      - v. Asignación de tipo de datos
      - vi. *El veneno de tipo cambiado automático*
    - c. Agregar Columnas (Entendimiento)
      - i. Columnas a partir de ejemplo
      - ii. Columna condicional
      - iii. Columna personalizada
      - iv. Invocación de función personalizada
      - v. Otras columnas
        - 1. Columna de índice
        - 2. Duplicar columna

3. Aplicaciones superiores con técnicas de depuración
  - a. Limpieza con la operación de rellenar
    - i. Implementación y variaciones
    - ii. Consideraciones (null y totalmente vacío)
  - b. Manipulación de datos compromiso (caso tabla comprimida)
  - c. División de columnas con múltiples criterios Dinámicos

## 2. LIMPIEZA DE DATOS DE SEGUNDO GRADO: TÉCNICAS DE ALTERACIÓN

- a. Introducción
  - i. Anulación de dinamización
  - ii. Columna dinámica
  - iii. Agrupación
- b. Anulación de dinamización (Unpivot) estudio detallado y mecánica
  - i. Anulación de dinamización de columnas (A.D.C)
  - ii. Anulación de dinamización de otras Columnas (A.D.O.C): **Table.UnpivotOtherColumns**
  - iii. Anulación de dinamización con columnas y filas vacías: A.D.C y A.D.O.C
  - iv. Anulación de dinamización con totales y subtotales A.D.C y A.D.O.C
  - v. Anulación de dinamización de columnas seleccionadas únicamente (A.D.S.C)
  - vi. *Anulación de dinamización y plegado de consultas*
- c. Transformaciones con Anulación de Dinamización
  - i. Anulación de dinamización con múltiples niveles
    1. Paso a Paso y Pseudo-Generalización NxM
    2. Función **Table.PivotNxMToTabular**
  - ii. Tablas abultadas con 2 Dimensiones (2D)
    1. Tablas abultadas 2d con son saltos
    2. Tablas abultadas con saltos
    3. Tablas abultadas con **Table.UnpivotKeepNulls**
- d. Columna dinámica **Table.Pivot**
  - i. Versión simplificada
  - ii. Versión extendida
  - iii. *Columna dinámica y plegado de consultas*
  - iv. Remover filas y columnas vacías con: *Método Unpivot/Pivot (No recomendado)*

- e. Transformación con Anulación de dinamización
  - i. Tabla apilada con nombres coincidentes
  - ii. Tabla apilada con nombres no coincidentes (Tabla de conversión)
  - iii. Sobre otros métodos
- f. Estrategias de Transformación con agrupación

### 3. CONSOLIDACIÓN DE DATOS

- a. Fundamentos
  - i. Anexar (Append)
  - ii. Combinar (Merge)
    - 1. Externa izquierda
    - 2. Externa derecha
    - 3. Externa completa
    - 4. Interna
    - 5. Anti izquierda
    - 6. Anti derecha
  - iii. Coincidencia difusa (combinación/grupación aproximada)
- b. Técnicas Básicas, Intermedias y Avanzadas de: Consolidación de Datos
  - i. Técnicas gracias a la generación de la CONSULTA DE MUESTRA AUTOMÁTICA
    - 1. Consolidación con nombre coincidentes
    - 2. Consolidación con nombre no coincidente síncronos (mismo orden)
    - 3. Consolidación con nombres no coincidentes asíncronos (orden diferente)
      - a. Consolidación por estandarización dado patrón
      - b. Consolidación por estandarización con método de la tabla de conversión
    - 4. ¡ADVERTENCIA!: Nombres de hojas iguales y cómo manejarlo manipulando la consulta de muestra.
  - ii. Técnicas por invocación de FUNCIÓN O EXPRESIÓN DEL LENGUAJE M
    - 1. Consolidación de n archivos que necesitan A.D.O.C
  - iii. Técnicas por generación de CONSULTA DE MUESTRA PROPIA
    - 1. Generación de consulta de muestra propia
      - a. Creación paso a paso (con receta)
      - b. Entrada "Seed" de tabla
    - 2. Consolidación con nombres no coincidentes asíncronos (orden diferente/patrón)

- iv. Técnica de consolidación por DESINTEGRACIÓN DE GRAN TABLA PLAN
  - 1. Desintegración con Agrupación por generación de errores
  - 2. Consolidación de n archivos que necesitan A.D.O.C
- v. El Mapa de Estrategias y Tácticas de Consolidación de Datos (Síntesis)

#### 4. INICIACIÓN AL LENGUAJE M Y ESTUDIO FORMAL

- a. Interiorización de la lógica del lenguaje M
  - i. Bloque de trabajo/creación y su anatomía
    - 1. Identificador
      - a. Identificadores simples
      - b. Identificadores compuestos
    - 2. Expresión
      - a. Operadores
      - b. Ambiente
    - 3. Bloque **let ... in**
      - a. Bloque de un solo paso
      - b. Bloque de múltiples pasos
      - c. Lógica e implementación de **let ... in** anidados
      - d. El bloque primordial (*identificador externo*)
  - ii. Otros bloques de creación
    - 1. El bloque **if ... then... else if ...**
  - iii. Clases de valores (*los ladrillos del lenguaje M*)
    - 1. Sobre valores primitivos
      - a. Valor numérico
        - i. Precisión de número (double y decimal)
        - ii. Funciones **Number.X** y ejercicios
        - iii. Valores numéricos especiales
      - b. Valores texto
        - i. Cadenas literales
        - ii. Secuencias de escape
        - iii. Interpolación de cadena ("String interpolation") con Text.Format
        - iv. Funciones **Text.X** y ejercicios

- c. Valores de fecha
  - i. Funciones *Date.X*
- d. Funciones de fecha y hora
  - i. Funciones *DateTime.X*
- e. Funciones de duración
  - i. *Duration.X* y ejercicios
- f. Funciones de Hora
  - i. *Time.X* y ejercicios
- iv. Cómo trabajar con operaciones entre fechas y horas correctamente
  - 1. Sumar y restar fechas
  - 2. Sumar y restar horas para obtener duración
- v. Valores de entrada del usuario (parámetros)
  - 1. Parámetros estáticos
    - a. Creación de un parámetro
    - b. Parámetro de un sólo valor
    - c. Parámetro de una lista
    - d. Parámetro de una consulta
    - e. Ejemplos y ejercicios
  - 2. Parámetros dinámicos
  - 3. Metadatos y parámetros detrás de cámaras

## 5. VALORES ESTRUCTURADOS Y CASOS DE APLICACIÓN

- a. Paradigma de programación del Lenguaje M
  - i. Lenguaje de programación funcional
  - ii. Funciones "mayoritariamente" puras (Sobre side effect y No side effect)
  - iii. Inmutabilidad (Sobre variables mutables e inmutables)
  - iv. Dinámicamente tipado
  - v. Ciudadanos de primera clase
  - vi. Funciones de orden superior (expresiones Lambda y funciones anónimas)
  - vii. Orden de evaluación
  - viii. Parcialmente perezoso (Leazy evaluation & Eager evaluation)
  - ix. Semántica de transmisión
  - x. Plegado de consultas

b. Estudio de valores estructurados

i. Valores Lista (*list*)

1. Sintaxis de inicialización
  - a. Definición y sintaxis
  - b. Su importancia en el lenguaje (*columnas*)
  - c. Lo que no puede hacer una lista
2. Creación de listas estáticas
  - a. Lista numérica
  - b. Lista de textos
  - c. Listas alfanuméricas
  - d. Listas de listas
  - e. Lista vacía significado y cuidados
3. Creación de listas dinámicas
  - a. Serie de número consecutivos
  - b. Serie de número con saltos
  - c. Serie del alfabeto en mayúscula, minúscula y combinaciones
  - d. Serie de valores de texto con salto
4. Operador de acceso
  - a. Operador de índice posicional (*Base de conteo de M*)
    - i. Acceso simple
    - ii. Acceso a múltiples niveles
  - b. Operador de protección
  - c. Guía de implementación
5. APLICACIÓN DE LOS VALORES LISTA EN EL MUNDO REAL
  - a. Tratamiento de datos faltantes en encuestas
    - i. Imputación de datos por constante con desde la interfaz
    - ii. Imputación de datos por promedio con *List.ReplaceValue* y *anidación en un sólo bloque M*
6. Funciones de lista y aplicaciones
  - a. Listado de funciones de lista y su papel en el lenguaje M
  - b. Dividir columna a filas y aplicación con *List.RemoveItems* y *List.RemoveMatchingItems*

7. Ejercicios
  - a. Lista vacía *List.IsEmpty*
  - b. Lista y operadores lógicos en M: *and*, *or* y *not*
8. Ciudadanos de primera clase y funciones de orden superior
  - a. Funciones de iteración
  - b. Expresión Lambda y funciones anónimas

## 9. STRANGER THINGS EN M: CAPÍTULO 1

- ii. Valores de registro (*record*)
  1. Sintaxis de inicialización
    - a. Definición y sintaxis
    - b. Su importancia en el lenguaje M (*filas*)
    - c. Lo que no puede hacer un registro
  2. Creación de registros
    - a. Registro de número
    - b. Registro de valores primitivos
    - c. Registro con identificadores "especiales"
    - d. Registro de valores estructurados
    - e. Registro de registros
    - f. Registro vacío y su significado
  3. Operador de Acceso
    - a. Operador de búsqueda
      - i. Acceso simple
      - ii. Acceso múltiples niveles
    - b. Operador de protección
    - c. Operador de proyección
    - d. Guía de implementación
  4. APLICACIÓN DE LOS VALORES DE REGISTRO EN EL MUNDO REAL
    - a. Extracción del valor mínimos de múltiples columnas
  5. Funciones de registro
    - a. Listado de funciones de registro
      - i. Ejercicio con *Record.ToList* y *Record.FieldValues*

## 6. STRANGER THINGS EN M: CAPÍTULO 2



- iii. Valores de tabla (**table**)
  - 1. Construcción de valores de tabla
    - a. Función de construcción **#table** sin tipo de dato
    - b. Función de construcción **#table** con tipo de datos
    - c. Construcción con **Table.FromColumns**
    - d. Construcción con **Table.FromRecords**
    - e. Construcción con **Table.FromRows**
    - f. Construcción con **Table.FromList**
    - g. Construcción con **Table.FromValue**
  - 2. Acceso a valores en una tabla
    - a. Acceso a columna
    - b. Acceso a fila
    - c. Acceso a una celda
    - d. Protección y acceso con sólo un parámetro
    - e. Guía de implementación
- c. Valor de función (**function**)
  - i. Elementos de una función personalizada
  - ii. Sintaxis y de una función personalizada
  - iii. Identificador para un valor de función
  - iv. Funciones personalizadas básicas
    - 1. Crecimiento año a año
    - 2. Resolución de distancias con múltiples argumentos
    - 3. Resolución horas de inicio y finalización con argumentos opcionales
  - v. Aplicaciones de funciones personalizadas
    - 1. Función para obtención de días festivos según región y años
    - 2. Función para obtención de latitudes y longitudes
- d. Profundización en Funciones de Iteración en M e **each**
  - i. Funciones de iteración en el lenguaje M
    - 1. FUNCIONES DE ITERACIÓN BASADO EN REGISTROS
      - a. Función **Table.SelectRows**
      - b. Función **Table.AddColumn**
      - c. Creación de funciones para argumentos de otras funciones

- d. Creación de funciones para argumentos en S.S: **each** y anónimas
  - e. Implementación de **each** anidados ("**SUMAR.SI.CONJUNTO**")
2. FUNCIONES DE ITERACIÓN BASADO EN LISTAS
- a. Función **Table.CombineColumns**
    - i. Funciones espaciales para argumentos de funciones
      - 1. *Funciones de Combinación*
      - 2. *Funciones de Comparación*
      - 3. *Funciones de Sustituto*
      - 4. *Funciones de Separación*
  - b. Función **Table.SplitColumn**
  - c. Función **Table.Pivot**
- e. APLICACIÓN DE LENGUAJE M A PREPARACIÓN DE DATOS EN EL MUNDO REAL
- i. **Anulación de Dinamización Avanzada con Lenguaje M**
    - 1. Resolución a la obtención de datos en celdas específicas para poblar columna
    - 2. Tabla abultada en 2D con saltos y con encabezados/rótulos NxM (Caso Angélica)
      - a. Técnicas de referenciación a celdas adyacentes
      - b. Aplicación de **if ... then ... else**
      - c. Aplicación de operador de protección
      - d. Resolución al formato
      - e. Resolución al formato con técnica de errores con
        - i. **try**
        - ii. **try... otherwise**
        - iii. **try... catch**
    - 3. Resolución de un formato DEBE/HABER en Contabilidad
      - a. Función **Table.Group**
      - b. Función **Table.AddIndexColumn**
      - c. Función **Table.NestedJoin**
        - i. Combinaciones y ajustes finales del formato
  - ii. **Transformaciones Avanzadas con Lenguaje M**
    - 1. Formato basado en primera fila en grupo si hay repetición (Caso Gregorio García)
    - 2. Generación de código tributarios tipo RFC (Caso Joseph Pinzón)
    - 3. Resolución de un Formato "**Mushroom**" apilado (Caso Miguel Cordero)