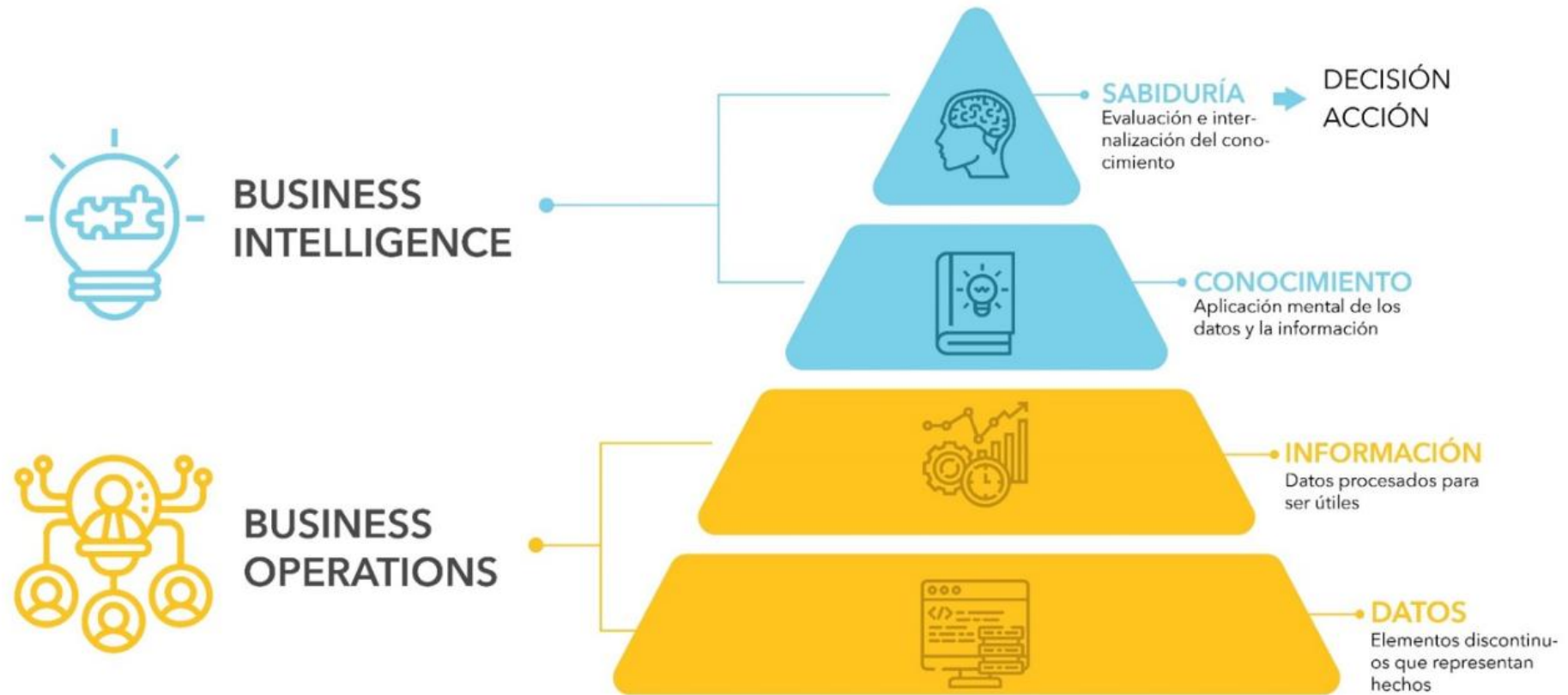


POCESO DE DEFINICIÓN DE INDICADORES

¿De dónde viene la necesidad de generar indicadores? La pirámide de la información



La pirámide de la información



Conceptos de gestión estratégica

La estrategia implica planificar cómo una organización o un individuo alcanzarán sus objetivos, en definitiva, se trata de asegurar la supervivencia y prosperidad de la empresa.

La toma de decisiones a nivel estratégico hace referencia al modo en el que la empresa despliega sus recursos y capacidades en el entorno con el fin de alcanzar sus objetivos.

Los elementos principales de la estrategia de éxito son: objetivos sencillos coherentes y a largo plazo, implantación eficaz, conocimiento profundo del entorno competitivo y valoración objetiva de recursos.



Estrategia dentro de una organización

Cuando queremos trazar un cambio en nuestra organización solemos hablar de 3 planes:

- Plan director: declaración de intenciones a largo plazo, no tiene una construcción a menos de 5 años.

Ejemplo: transformación digital en el sistema educativo de España de 2025 a 2030

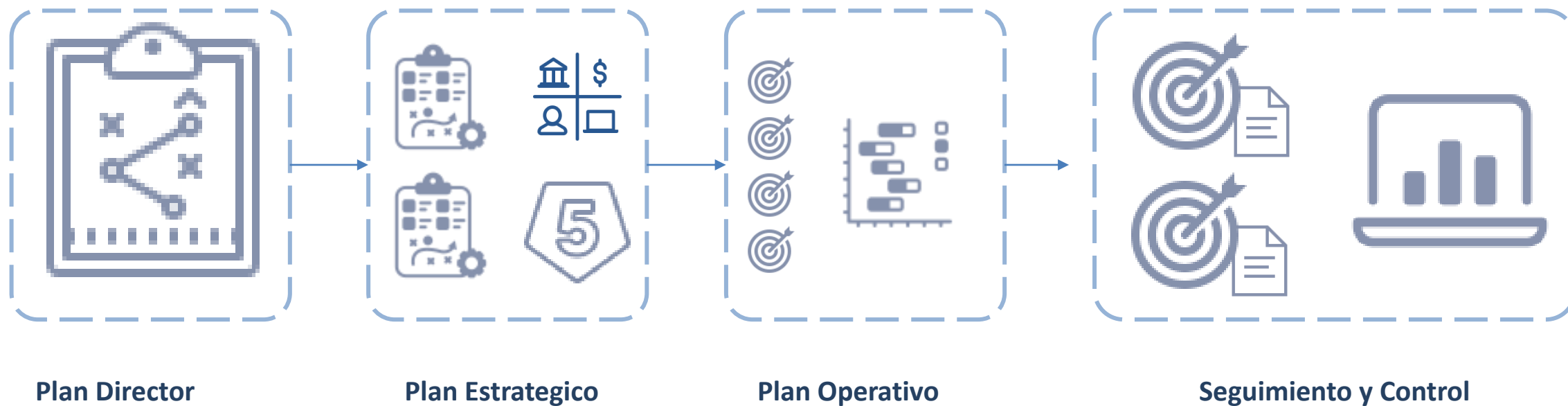
- Plan estratégico: tratamos de dibujar como voy a llevar a cabo la estrategia, quienes están involucrados, cuando tienen que estar hechas dichas acciones, etc.

Ejemplo: Desarrollo de estrategias específicas, como la implementación de plataformas educativas en línea y la capacitación de docentes en tecnologías digitales, para cumplir con los objetivos del plan director de transformación digital en la educación.

- Plan operativo: planes prácticos, llegando al mayor de los detalles con el trabajo.

Ejemplo: Contratación de especialistas en tecnología educativa, diseño de programas de formación docente en el uso de herramientas digitales y asignación de presupuestos para la adquisición de tecnología educativa en el próximo año.

¿Qué papel juegan las herramientas de Business Intelligence como Power BI?





Indicador o KPI

Magnitud usada para medir o comparar resultados obtenidos durante la ejecución de una actividad de negocio, proyecto o programa.

¿Cómo debe ser un indicador?

Útil

Se debe ser concreto a la hora de definir indicadores. Que estén vinculados con el éxito de negocio. Mejor pocos y bien.

Existente

Debemos ser capaces de poder extraer la información para el cálculo de alguno de nuestros sistemas de información.

Disponible

Fácilmente obtenible por todas las áreas de negocio relacionadas.

Actualizable

Se elige el periodo de actualización en función de los requerimientos. Se suele desear actualizaciones periódicas

Preciso

Se debe asegurar la calidad del dato para que la información sea veraz

Propósito

Debe tener un propósito concreto y ser usable.

Principio SMART

KPIs SMART

S

Específicas

Specific

M

Medibles

Measurable

A

Alcanzable

Attainable

R

Relevantes

Relevants

T

Definidos
en el tiempo

Time based

Principio SMART





Cálculos

Operaciones que realizamos sobre nuestros datos con el objetivo de resumirlos y poder obtener mejores conclusiones

¿Por qué es importante usar cálculos?

Síntesis de datos

Permiten condensar grandes conjuntos de datos para obtener información clave de manera concisa.

Toma de decisiones

Esenciales para respaldar decisiones informadas al proporcionar métricas clave para evaluar el rendimiento.

Identificación de patrones

Facilitan la detección de patrones y relaciones en los datos que podrían pasar desapercibidos.

Validación de datos

Ayudan a validar hipótesis al proporcionar pruebas cuantitativas

Presentación de Informes

Son clave para comunicar de manera clara y efectiva los resultados del análisis de datos.

Optimización de recursos

Permiten evaluar el rendimiento y la eficacia, conduciendo a mejoras operativas y ahorro de recursos.

CREACIÓN DE MEDIDAS DAX

¿Cómo realizo cálculos en Power BI?

Medidas implícitas:

Se generan cuando incluimos un campo numérico a un objeto visual. Los campos numéricos se agregan automáticamente en formato suma, recuento... Podemos elegir agregaciones diferentes.

Medidas explícitas:

Se crean mediante el uso de funciones DAX que permiten definir columnas o medidas calculadas



¿Qué es DAX?

Se trata de un lenguaje de formulación que admite funciones, operadores y constantes que se pueden utilizar en una fórmula o expresión para calcular y devolver valores. Ayuda a crear información nueva a partir de la que se encuentra ya en el modelo de datos.

Permite añadir medidas y columnas calculadas

Sintaxis intuitiva similar a Excel

Dentro de Power BI, podemos crear dos tipos de elementos con DAX:

- Columnas calculadas
- Medidas



Diferencia entre una medida y una columna calculada

Columnas calculadas

La fórmula se evalúa para cada fila de la tabla y devuelve un valor único. Esa nueva columna se almacena en la tabla.

Ingresos = Precio x Cantidad

Cliente	Precio	Cantidad	Ingresos
Rubén	22	3	66
Pablo	23	8	184
Juan	25	12	300
Andrés	28	6	168
Diana	34	2	68
María	15	7	105

Medidas

Las medidas son agrupaciones que resumen los datos, se recalculan en función de la información observada

Total Cantidad = SUM(Cantidad)

Cliente	Precio	Cantidad
Rubén	22	3
Pablo	23	8
Juan	25	12
Andrés	28	6
Diana	34	2
María	15	7

$\Sigma = 38$

Columnas Calculadas

- No se puede referenciar una celda concreta de la tabla como en Excel
- Los cálculos serán visibles en la vista de datos
- Las columnas calculadas son afectadas por el contexto de fila
- Funcionalidad muy similar a Power Query
- **No se usan para agrupar datos sino para generar columnas nuevas**

Medidas

- No se puede referenciar una celda concreta de la tabla como en Excel
- Las medidas no son visibles en las tablas. Solo se pueden ver al ser añadidas como una visualización gráfica, en tablas, visuales, matrices...
- Las medidas se basan en el **contexto de filtro**. Cada medida se recalcula cuando se cambian los filtros del informe que haya alrededor
- **Se usan cuando necesitas agrupar información aplicando una operación**



Sintaxis de una medida DAX

Cada tipo de cálculo en DAX, se define por su nombre, seguido del símbolo igual (=) y después una fórmula de DAX:

- Nombre de la medida
- Nombre de la función aplicada
- Referencia a la tabla a usar
- Nombre del campo a usar

Total Quantity = SUM(Ventas[Importe])

- * Para referenciar campos de tabla, se indica el nombre de la columna usando también la referencia de tabla.
- * Para referenciar medidas no se incluye el nombre de la tabla, porque las medidas se pueden mover de sitio

Tipos de funciones principales en DAX

- Funciones de fecha y hora
- Funciones de filtro
- Funciones financieras
- Funciones lógicas
- Funciones matemáticas y trigonométricas



DAX: FUNCIONES DE TIEMPO



Funciones temporales (I)

DAY/MONTH/YEAR (*fecha*):

Devuelve el día del mes, mes del año, o año, de una fecha introducida. El campo debe ser tipo fecha. Ejemplos:

```
Día = DAY( Calendario[Fecha] )  
Mes = MONTH( Calendario[Fecha] )  
Año = YEAR( Calendario[Fecha] )
```

HOURL/MINUTE/SECOND(*Datetime*):

Devuelve la hora, minuto o segundo de una columna introducida. El campo debe ser de tipo datetime. Ejemplos:

```
Hora = HOUR( Calendario[FechaTime] )  
Minuto = MINUTE( Calendario[FechaTime] )  
Segundo = SECOND( Calendario[FechaTime] )
```

Funciones temporales (II)

TODAY/NOW():

Devuelve la fecha o la hora exacta del momento en el que se está ejecutando el comando DAX

`DíaDeHoy = TODAY()`

`HoraActual = NOW()`

WEEKDAY(*fecha*):

Devuelve un número del 1 al 7 que identifica el día de la semana de una fecha. De forma predeterminada, el día está comprendido entre 1 (domingo) y 7 (sábado).

`DíaDeLasSemana = WEEKDAY(Calendario[Fecha])`

Funciones temporales (III)

WEEKNUM(*fecha*):

Devuelve el número de semana del año de la fecha especificada

`SemanaDeIAño = WEEKNUM(Calendario[Fecha])`

ENDOFMONTH(*fecha*):

Devuelve la última fecha del mes del contexto actual para la columna de fechas especificada. Ejemplo:

`FinalDeMes = ENDOFMONTH(Calendario[Fecha])`

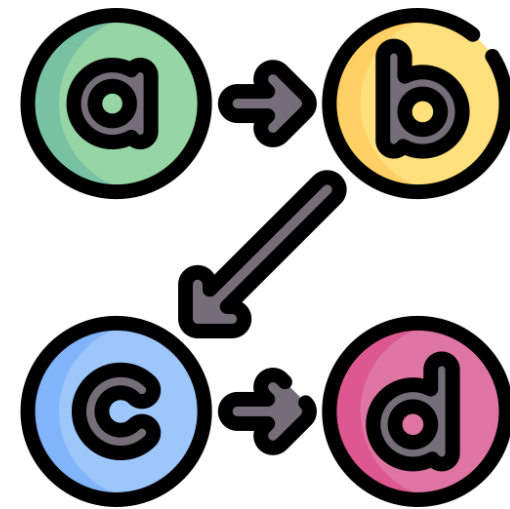
Funciones temporales (IV)

DATEDIFF(*fecha1*, *fecha2*, *intervalo*):

Devuelve diferencia entre dos fechas, eligiendo el intervalo.

Diff = DATEDIFF([Fecha1], [Fecha2], Day)

DAX: FUNCIONES LÓGICAS



Operadores de comparación

Estas funciones van a hacer uso de los diferentes operadores de comparación, para generar un nuevo resultado en forma de medida o de columna adicional.

Los operadores básicos en DAX son:

Operadores de comparación	Significado	Ejemplo
=	Igual a	[Region] = "USA"
==	Estrictamente igual a	[Region] == "USA"
>	Mayor que	[Sales Date] > "Jan 2009"
<	Menor que	[Sales Date] < "Jan 1 2009"
>=	Mayor o igual que	[Amount] >= 20000
<=	Menor o igual que	[Amount] <= 100
<>	No igual a	[Region] <> "USA"

Funciones condicionales (I)

IF(prueba_lógica, valor_true, valor_false):

Comprueba una condición y devuelve un valor cuando es “True”; en caso contrario, devuelve un segundo valor.

Ejemplos:

*ColumnaCondional =
IF (Calendario[DiaSemana] < 6, “Laborable”)*

Día Nombre ▾	DiaSemana ▾
Lunes	1
Martes	2
Miercoles	3
Jueves	4
Viernes	5
Sabado	6
Domingo	7

ColumnaCondional ▾
Laborable
Laborable
Laborable
Laborable
Laborable

Funciones condicionales (II)

IF(prueba_lógica, valor_true, valor_false):

Comprueba una condición y devuelve un valor cuando es “True”; en caso contrario, devuelve un segundo valor.

Ejemplos:

```
ColumnaCondicional =  
IF (Calendario[DiaSemana] < 6  
    ,“Laborable”  
    ,“Festivo”)
```

Día Nombre ▾	DiaSemana ▾
Lunes	1
Martes	2
Miercoles	3
Jueves	4
Viernes	5
Sabado	6
Domingo	7

ColumnaCondicional ▾
Laborable
Laborable
Laborable
Laborable
Laborable
Festivo
Festivo

Operadores lógicos

Los operadores lógicos pueden ser usados dentro de una función *IF* de DAX para concatenar varias condiciones dentro de una misma sentencia:

Operador de texto	Significado	Ejemplos
&& (doble Y comercial)	Crea una condición AND entre dos expresiones que tienen ambas un resultado booleano. Si ambas expresiones devuelven TRUE, la combinación de las expresiones también devuelve TRUE; de lo contrario, la combinación devuelve FALSE.	<code>(([Region] = "France") && ([BikeBuyer] = "yes"))</code>
(doble barra vertical)	Crea una condición OR entre dos expresiones lógicas. Si alguna de las expresiones devuelve TRUE, el resultado es TRUE; solo cuando ambas expresiones son FALSE, el resultado es FALSE.	<code>((([Region] = "France") ([BikeBuyer] = "yes"))</code>
IN	Crea una condición OR lógica entre cada fila que se compara con una tabla. Nota: En la sintaxis del constructor de tabla se emplean llaves.	<code>'Product'[Color] IN { "Red", "Blue", "Black" }</code>

Funciones condicionales (II)

IF(prueba_lógica, valor_true, valor_false):

Comprueba una condición y devuelve un valor cuando es “True”; en caso contrario, devuelve un segundo valor.

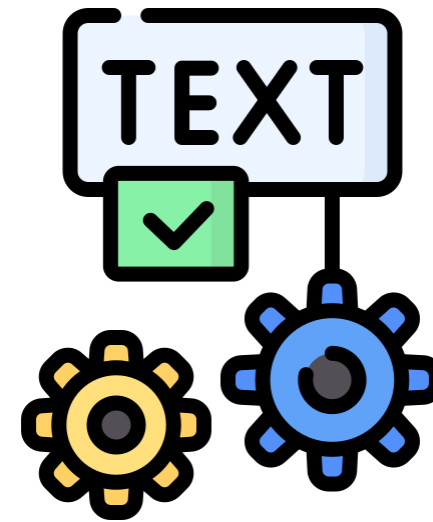
Ejemplos:

```
Categoría =  
IF (  
Tabla[Precio] < 25 && Tabla[Cantidad] >= 8  
    ,“Económico y demandado”  
    ,“No definido”  
)
```

Cliente	Precio	Cantidad
Rubén	22	3
Pablo	23	8
Juan	25	12
Andrés	28	6
Diana	34	2
María	15	7

Categoría
No definido
Económico y demandado
No definido
No definido
No definido
Económico y demandado

DAX: FUNCIONES DE TEXTO



Funciones de texto

UPPER/LOWER (*Texto*):

Convierte las letras de una cadena de texto a mayúsculas / minúsculas. Ejemplos:

```
TextoMayus = UPPER ( Clientes[Nombre] )  
TextoMinus = LOWER ( Clientes[Apellidos] )
```

SUBSTITUTE (*Columna, TextoViejo, TextoNuevo*)

Reemplaza texto existente por otro nuevo en una cadena de texto. Ejemplo:

```
ModificacionDpto =  
SUBSTITUTE( Empresa[Dpto], "IT", "IT & Data")
```


Funciones de texto

LEN():

Devuelve el nº de caracteres de un string. Ejemplo:

LEN(Texto)

LEFT/MID/RIGHT():

Devuelve un numero de caracteres desde el inicio, medio o fin de un texto. Ejemplo:

LEFT/RIGHT(Texto, N° Caracteres)

DAX: FUNCIONES DE AGREGACIÓN



Operaciones matemáticas (I)

SUM (Número):

Suma todos los números de una columna. Ejemplos:

`BeneficioTotal = SUM (Ventas[Beneficio])`

AVERAGE (Número):

Devuelve el promedio (media aritmética) de todos los números de una columna. Ejemplos:

`PromedioCantidad = AVG (Ventas[Cantidad])`

Operaciones matemáticas (II)

MAX (Número, [Número2])

Devuelve el valor mayor de una columna, o entre dos expresiones escalares. Ejemplos:

CantidadMax = MAX(Ventas[Cantidad])

CantidadMax = MAX(Ventas[Cantidad], 5)

MIN (Número, [Número2])

Devuelve el valor menor de una columna, o entre dos expresiones

CantidadMin = MIN(Ventas[Cantidad])

CantidadMin = MIN(Ventas[Cantidad], 5)

Operaciones matemáticas (III)

DIVIDE (Numerador, Denominador, Alternativa)

Realiza la división y devuelve el resultado alternativo o BLANK() si se divide entre 0. Ejemplos:

% Beneficio =

DIVIDE (Ventas[Costes], Ventas[Precio], 0)

DAX: FUNCIONES DE CONTEO



Conteo de registros (I)

`COUNT (Columna):`

Cuenta el número de filas de la columna especificada que contienen valores que no están en blanco. Ejemplos:

```
NumVentas = COUNT( Ventas[ID] )
```

`COUNTROWS (Columna):`

La función COUNTROWS cuenta el número de filas de la tabla especificada o de una tabla definida por una expresión. Ejemplos:

```
NumVentas = COUNTROWS( Ventas )
```

Conteo de registros (II)

DISTINCTCOUNT (*Columna*):

Cuenta el número de valores distintos de una columna. Ejemplos:

```
NumClientes = DISTINCTCOUNT ( Clientes[ID] )
```


DAX: FUNCIÓN CALCULATE



Uso básico de Calculate (I)

CALCULATE (Expresión, Filtro1, Filtro2...):

Evalúa una expresión en un contexto de filtro modificado. Ejemplos:

```
VentasBicicleta =  
CALCULATE(  
SUM(Ventas[Beneficio], Producto = "Bicicleta")  
)
```

Uso básico de Calculate (II)

CALCULATE (Expresión, Filtro1, Filtro2...):

Evalúa una expresión en un contexto de filtro modificado. Ejemplos:

```
VentasBicicleta =  
CALCULATE(  
SUM(Ventas[Beneficio],  
Ventas[Prd]="Bicicleta" && Ventas[Col]= "Rojo"  
)
```

Uso básico de Calculate (III)

CALCULATE (Expresión, Filtro1, Filtro2...):

Evalúa una expresión en un contexto de filtro modificado. Ejemplos:

```
VentasMultiproducto =  
CALCULATE(  
SUM(Ventas[Beneficio],  
Ventas[Cantidad] >= 2  
)
```

Uso básico de Calculate (IV)

CALCULATE (Expresión, Filtro1, Filtro2...):

Evalúa una expresión en un contexto de filtro modificado. Ejemplos:

```
VentasMultiproducto =  
CALCULATE(  
SUM(Ventas[Beneficio],  
Ventas[Fecha] >= "03-03-2023"  
)
```

Uso básico de Calculate (V)

DATEADD (Fecha, N°Intervalos, TipoIntervalo):

Se suele utilizar junto a la función Calculate. Devuelve una tabla que contiene una columna de fechas que se han desplazado hacia delante o hacia atrás en el tiempo según el número especificado de intervalos desde las fechas del contexto actual. Ejemplos:

```
VentasAñoAnterior =  
CALCULATE(  
SUM(Ventas[Beneficio],  
DATEADD( Ventas[Fecha], -1, YEAR ))
```