

Clases de Valores en el Lenguaje M

Valores Primitivos, Valores Estructurados y Valores Abstractos

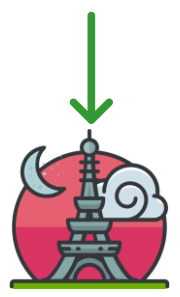
Clases de Valores en M



Valores Primitivos

Valores Atómicos

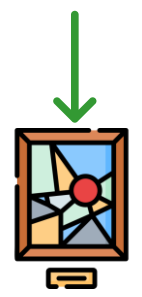
- Numéricos (**number**)
- Fecha (**date**)
- Hora (**time**)
- Fecha y Hora (**datetime**)
- F,H y Z (**datetimezone**)
- Duración (**duration**)
- Texto (**text**)
- Nulo (**null**)
- Binarios (**binary**)
- Lógico (**logical**)



Valores Estructurados

Valores Compuestos

- Lista (**list**)
- Registro (**record**)
- Table (**table**)



Valores Abstractos

Valores Compuestos

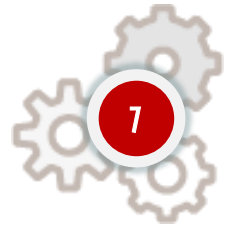
- Funciones (**function**)
- Tipos (**types**)



ELEMENTOS DE VALORES

A cada valor se les asocia un conjunto de elementos para poder trabajar con ellos

1. Una sintaxis “literal”
2. Un conjunto de valores
3. Un conjunto de operadores
4. Un valor intrínseco



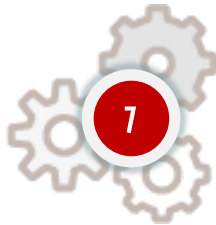
Clases de Valores en el Lenguaje M

Valores Primitivos, Valores Estructurados y Valores Abstractos

Literales y Sintaxis de Inicialización		
Tipo	Literal	Función
Primitivos	Nulo	null
	Lógico	true false
	Numérico	0 1 -1 1.5 2.3e-5
	Hora	#time (09, 15, 00)
	Fecha	#date (2019, 04, 27)
	Fecha y Hora	#datetime (2019, 04, 27, 09,15,00)
	Fecha, Hora y Zona	#datetime (2019, 04, 27, 09,15,00, 00)
	Duración	#duration (0, 1, 30, 0)
	Texto	"Escuela de Inteligencia"
	Binario	#binary("AQID")
Estructurados	Lista	{ 1, 2, 3 }
	Registro (Record)	[X = 1 , Y = 2]
	Tabla	#table({"X","Y"},{{0,1},{1,0}})
	Función	(x) => x + 1
	Tipo	type { number } type table [A = any, B = text]

Clases de Valores en el Lenguaje M


Valores Primitivos, Valores Estructurados y Valores Abstractos



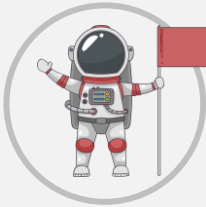
Familia de Funciones Nativas	
Tipo	Familia de Funciones
Nulo	
Lógico	= Logical.X
Numérico	= Number.X
Hora	= Time.X
Fecha	= Date.X
Fecha y Hora	= DateTime.X
Fecha, Hora y Zona	= DateTimeZone.X
Duración	= Duration.X
Texto	= Text.X
Binario	= Binary.X
Lista	= List.X
Registro (Record)	= Record.X
Tabla	= Table.X
Función	= Function.X
Tipo	= Type.X

Documentación


- 1



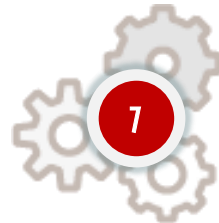
MSN Online: En la siguiente web: <https://docs.microsoft.com/en-us/powerquery-m/table-columnnames> encontramos la documentación online de Microsoft de las funciones M.
- 2



Función #shared: La función #shared retorna un registro con todos los valores del Power Query actual, esto incluye no sólo nuestras consultas, sino todas las funciones disponibles nativas y personalizadas.
- 3



Intellisense: El propio menú de autocompletado y resaltado de sintaxis nos brindará la lista filtrada según unas letras, donde mediante el tooltip nos podemos hacer una buena idea inicial de las funciones.



Funciones en Lenguaje M Que Iteran

La Syntax Sugar each la definición de Función Personalizada

=

Table.SelectRows (

Tabla como valor **table** ,

i ITERADOR

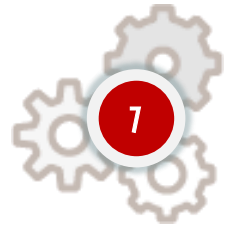
Condición como valor **function**

(FilaEnAlteración como valor **record**) como valor **logical**

i EXPRESIÓN M EN FILA

)

como valor **table**



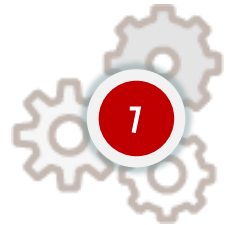
Funciones en Lenguaje M Que Iteran

La Syntax Sugar each la definición de Función Personalizada

= **Table.SelectRows** (t , (x as record) => x[Precio] > 50)

Libro	Año	Precio
Tablas Dinámicas la Quinta Dimensión	2015	40
El ADN de Power Pivot	2016	60
Inteligencia de Negocios con Excel y Power BI	2018	70
El ADN de Power Query	2019	60
Power BI Reportes e Informes	2020	70

40 > 50 ✖



Funciones en Lenguaje M Que Iteran

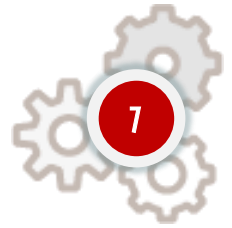
La Syntax Sugar each la definición de Función Personalizada

= **Table.SelectRows** (t , (x as record) => x[Precio] > 50)

Libro	Año	Precio
Tablas Dinámicas la Quinta Dimensión	2015	40
El ADN de Power Pivot	2016	60
Inteligencia de Negocios con Excel y Power BI	2018	70
El ADN de Power Query	2019	60
Power BI Reportes e Informes	2020	70

40 > 50 ✖

60 > 50 ✔



Funciones en Lenguaje M Que Iteran

La Syntax Sugar each la definición de Función Personalizada

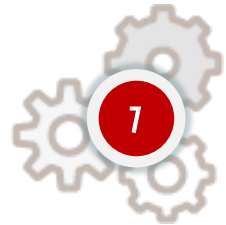
= **Table.SelectRows** (t , (x as record) => x[Precio] > 50)

Libro	Año	Precio
Tablas Dinámicas la Quinta Dimensión	2015	40
El ADN de Power Pivot	2016	60
Inteligencia de Negocios con Excel y Power BI	2018	70
El ADN de Power Query	2019	60
Power BI Reportes e Informes	2020	70

40 > 50 ✖

60 > 50 ✔

70 > 50 ✔

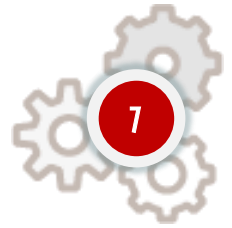


Funciones en Lenguaje M Que Iteran

La Syntax Sugar each la definición de Función Personalizada

= **Table.SelectRows** (t , (x **as record**) => x[Precio] > 50)

Libro	Año	Precio	
Tablas Dinámicas la Quinta Dimensión	2015	40	40 > 50 ✗
El ADN de Power Pivot	2016	60	60 > 50 ✓
Inteligencia de Negocios con Excel y Power BI	2018	70	70 > 50 ✓
El ADN de Power Query	2019	60	60 > 50 ✓
Power BI Reportes e Informes	2020	70	



Funciones en Lenguaje M Que Iteran

La Syntax Sugar each la definición de Función Personalizada

= **Table.SelectRows** (t , (x as record) => x[Precio] > 50)

Libro	Año	Precio		
Tablas Dinámicas la Quinta Dimensión	2015	40	40 > 50	✗
El ADN de Power Pivot	2016	60	60 > 50	✓
Inteligencia de Negocios con Excel y Power BI	2018	70	70 > 50	✓
El ADN de Power Query	2019	60	60 > 50	✓
Power BI Reportes e Informes	2020	70	70 > 50	✓

Funciones en Lenguaje M Que Iteran

La Syntax Sugar each la definición de Función Personalizada

let

```
Logcail_EsMayorA50 =
  ( Registro as record ) as logical => Registro[Precio] >= 50 ,
```

```
Logical_EsMayorA50yMenorA70 =
  ( R as record ) as logical =>
```

let

```
    s1 = R[Precio] > 50 ,
    s2 = R[Precio] < 60 ,
    s3 = s1 and s2
```

in

```
    s3
```

in

```
Logical_EsMayorA50yNenorA60
```

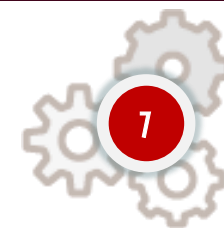
/*

Para que se una función válida para Table.SelectRows debe ser un función que recibe un registro (record) y retorna un valor booleano (logical)

Por otra parte, como la función puede constar más de un paso, entonces, cuando se así, no olvidar que varios pasos de trabajo se indican en un let ... in

*/





Funciones en Lenguaje M Que Iteran

La Syntax Sugar each la definición de Función Personalizada

```
= Table.SelectRows ( t, (x as record) => x[Precio] > 50 )
```

Libro	Año	Precio
Tablas Dinámicas la Quinta Dimensión	2015	40
El ADN de Power Pivot	2016	60
Inteligencia de Negocios con Excel y Power BI	2018	70
El ADN de Power Query	2019	60
Power BI Reportes e Informes	2020	70

40 > 50 ✗

60 > 50 ✓

70 > 50 ✓

60 > 50 ✓

70 > 50 ✓

Nota de Creación de Función

Nótese que al crear la función directamente dentro del argumento función (segundo argumento) de Table.SelectRows no existe la necesidad de señalar un identificador, puesto que el ambiente del argumento ya sabe para que será utilizando.

Es por ello que se dice que la definición e funciones dentro argumento de función en funciones son anónimas, por que o tiene identificador.

Si la función tiene varios de transformación el let ... In es inevitable así se encuentre en una función anónima

La Syntax Sugar: each





Funciones en Lenguaje M Que Iteran

Función Anónima each para syntax sugar de función personalizada

```
MyFuncion =  
let  
  Operacion = ( x ) => x + 1  
in  
  Operacion
```

```
MyFuncion =  
let  
  Operacion = ( _ ) => _ + 1  
in  
  Operacion
```

```
MyFuncion =  
let  
  Operacion = each _ + 1  
in  
  Operacion
```

```
MyFuncion =  
each _ + 1
```

Funciones en Lenguaje M Que Iteran

Función Anónima each para syntax sugar de función personalizada

La Syntax Sugar **each**



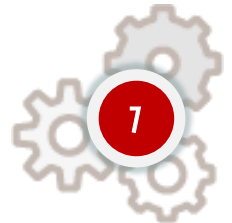
La palabra reservada **each** es una expresión en syntax sugar para definición o construcción de una función personalizada.



- 1 La expresión **each** no es un iterador, la iteración lo dicta la función en particular, ejemplo: `Table.SelectRows`
- 2 La función **each** se puede implementar por fuera de argumentos de funciones que reciben funciones.
- 3 El identificador `_` se puede omitir cuando se llama a un registro a cuando se llama fila de una tabla.

Algunas Funciones de Iteración en M





Funciones en Lenguaje M Que Iteran

La Syntax Sugar each la definición de Función Personalizada

=

Table.AddColumn (

Tabla como valor **table** ,

i ITERADOR

NombreDeNuevaColumna como valor **text** ,

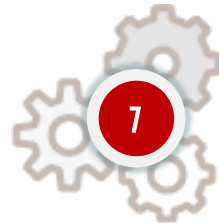
GeneradorDeColumna como valor **function** ,

(IteraciónActual como valor **record**) como valor **any**

[TipoDeColumna] como valor **type** ,

)

como valor **table**



Funciones en Lenguaje M Que Iteran

La Syntax Sugar each la definición de Función Personalizada



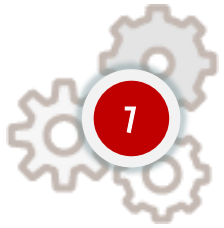
Let's Hack!

Ejercicio I — Table.AddColumn:

Sin apoyarse de la interfaz, y escribir todo desde cero en el editor avanzado: un nuevo paso de transformación que cree una nueva columna en la table t, la cual incremente el valor del precio en un 3% si es un libro ya publicado de lo contrario que devuelva el valor definido sin cambio.

Funciones en Lenguaje M Que Iteran

La Syntax Sugar each la definición de Función Personalizada



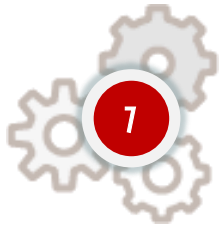
Solución B: Ejercicio 1



```
let
    tl =
        Table.AddColumn (
            t ,
            "Precio Nuevo" ,
            ( x ) =>
                if x[Publicado] then
                    x[Precio] * 1.03
                else
                    x[Precio]
            )
        )
in
    tl
```

Funciones en Lenguaje M Que Iteran

La Syntax Sugar each la definición de Función Personalizada



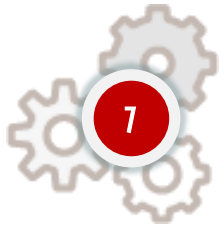
Solución A: Ejercicio 1



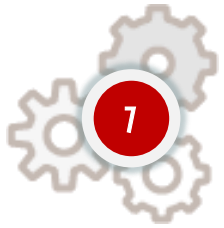
```
let
    tl =
        Table.AddColumn (
            t ,
            "Precio Nuevo" ,
            each
                if [Publicado] then
                    [Precio] * 1.03
                else
                    [Precio]
            )
in
    tl
```

Funciones en Lenguaje M Que Iteran

Función Anónima eacha para syntax sugar de función personalizada



**No todas las funciones de iteración del
Lenguaje M se comportan igual**



Funciones en Lenguaje M Que Iteran

La Syntax Sugar each la definición de Función Personalizada

=

Table.CombineColumns (

Tabla como valor **table** ,

i ITERADOR

Columns como valor **list** ,

i TRANSFORMACIÓN A...

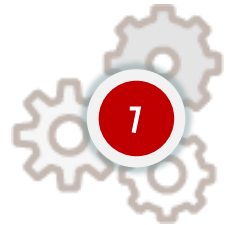
Combinador como valor **function** ,

(IteraciónActual como valor **list**) como valor **any**

NombreDeColumna como valor **text** ,

)

como valor **table**

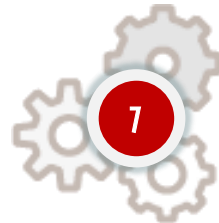


Funciones en Lenguaje M Que Iteran

La Syntax Sugar each la definición de Función Personalizada

= **Table.CombineColumns** (t , {"Libro" , "Año" } , (x as list) => x{1} & x{1} , "Nueva Columna")

Libro	Año	Precio
Tablas Dinámicas la Quinta Dimensión	2015	40
El ADN de Power Pivot	2016	60
Inteligencia de Negocios con Excel y Power BI	2018	70
El ADN de Power Query	2019	60
Power BI Reportes e Informes	2020	70

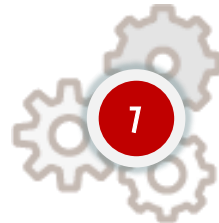


Funciones en Lenguaje M Que Iteran

La Syntax Sugar each la definición de Función Personalizada

`= Table.CombineColumns (t , {"Libro" , "Año"} , (x as list) => x{1} & x{1} , "Nueva Columna")`

Libro	Año	Precio
Tablas Dinámicas la Quinta Dimensión	2015	40
El ADN de Power Pivot	2016	60
Inteligencia de Negocios con Excel y Power BI	2018	70
El ADN de Power Query	2019	60
Power BI Reportes e Informes	2020	70

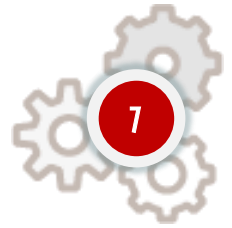


Funciones en Lenguaje M Que Iteran

La Syntax Sugar each la definición de Función Personalizada

`= Table.CombineColumns (t , {"Libro" , "Año"} , (x as list) => x{1} & x{1} , "Nueva Columna")`

Libro	Año	Precio
Tablas Dinámicas la Quinta Dimensión	2015	40
El ADN de Power Pivot	2016	60
Inteligencia de Negocios con Excel y Power BI	2018	70
El ADN de Power Query	2019	60
Power BI Reportes e Informes	2020	70

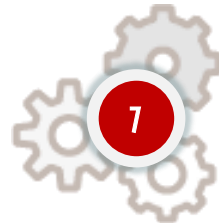


Funciones en Lenguaje M Que Iteran

La Syntax Sugar each la definición de Función Personalizada

= **Table.CombineColumns** (t , {"Libro" , "Año"} , (x as list) => x{1} & x{1} , "Nueva Columna")

Libro	Año	Precio
Tablas Dinámicas la Quinta Dimensión	2015	40
El ADN de Power Pivot	2016	60
Inteligencia de Negocios con Excel y Power BI	2018	70
El ADN de Power Query	2019	60
Power BI Reportes e Informes	2020	70

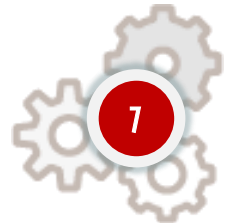


Funciones en Lenguaje M Que Iteran

La Syntax Sugar each la definición de Función Personalizada

`= Table.CombineColumns (t , {"Libro" , "Año"} , (x as list) => x{1} & x{1} , "Nueva Columna")`

Libro	Año	Precio
Tablas Dinámicas la Quinta Dimensión	2015	40
El ADN de Power Pivot	2016	60
Inteligencia de Negocios con Excel y Power BI	2018	70
El ADN de Power Query	2019	60
Power BI Reportes e Informes	2020	70



Funciones en Lenguaje M Que Iteran

La Syntax Sugar each la definición de Función Personalizada

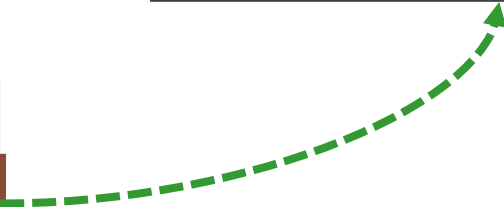
```
= Table.CombineColumns ( t , {"Libro" , "Año"} , (x as list) => x{1} & x{1} , "Nueva Columna" )
```

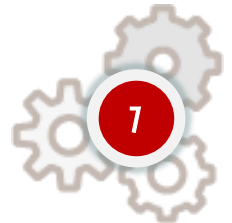


Libro	Año	Precio
Tablas Dinámicas la Quinta Dimensión	2015	40
El ADN de Power Pivot	2016	60
Inteligencia de Negocios con Excel y Power BI	2018	70
El ADN de Power Query	2019	60
Power BI Reportes e Informes	2020	70

X =

List
Tablas Dinámicas la Quinta Dimensión
2015





Funciones en Lenguaje M Que Iteran

La Syntax Sugar each la definición de Función Personalizada

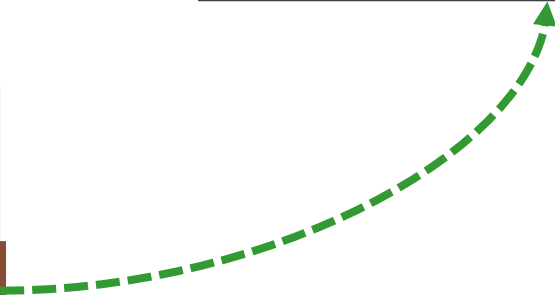
```
= Table.CombineColumns ( t , {"Libro" , "Año"} , (x as list) => x{1} & x{0} , "Nueva Columna" )
```



Libro	Año	Precio
Tablas Dinámicas la Quinta Dimensión	2015	40
El ADN de Power Pivot	2016	60
Inteligencia de Negocios con Excel y Power BI	2018	70
El ADN de Power Query	2019	60
Power BI Reportes e Informes	2020	70

X =

List
El ADN de Power Pivot
2016



Funciones en Lenguaje M Que Iteran

La Syntax Sugar each la definición de Función Personalizada

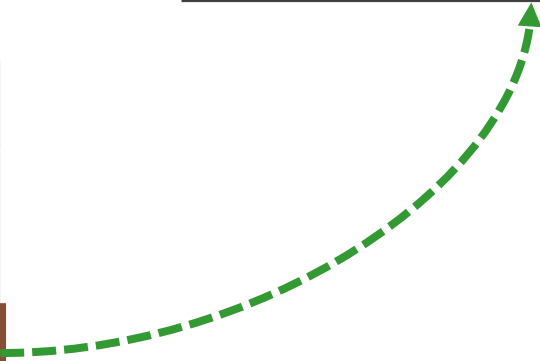
```
= Table.CombineColumns ( t , {"Libro" , "Año"} , (x as list) => x{1} & x{1} , "Nueva Columna" )
```

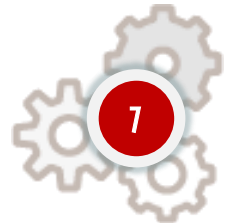


Libro	Año	Precio
Tablas Dinámicas la Quinta Dimensión	2015	40
El ADN de Power Pivot	2016	60
Inteligencia de Negocios con Excel y Power BI	2018	70
El ADN de Power Query	2019	60
Power BI Reportes e Informes	2020	70

X =

List
Int.de Neg. Con Excel y Power BI
2018





Funciones en Lenguaje M Que Iteran

La Syntax Sugar each la definición de Función Personalizada

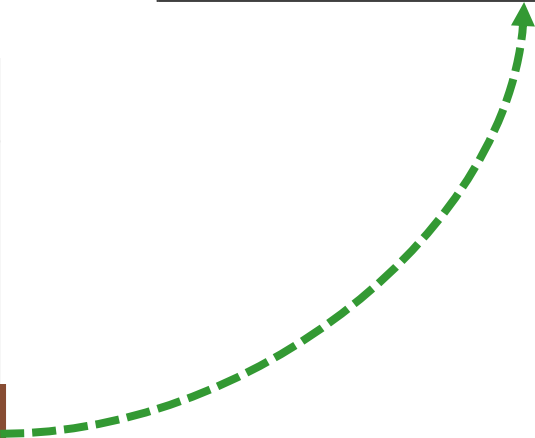
`= Table.CombineColumns (t , {"Libro" , "Año"} , (x as list) => x{1} & x{1} , "Nueva Columna")`

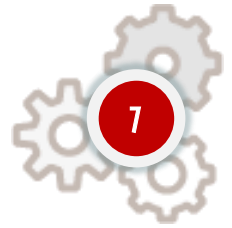


Libro	Año	Precio
Tablas Dinámicas la Quinta Dimensión	2015	40
El ADN de Power Pivot	2016	60
Inteligencia de Negocios con Excel y Power BI	2018	70
El ADN de Power Query	2019	60
Power BI Reportes e Informes	2020	70

X =

List
El ADN de Power Query
2019





Funciones en Lenguaje M Que Iteran

La Syntax Sugar each la definición de Función Personalizada

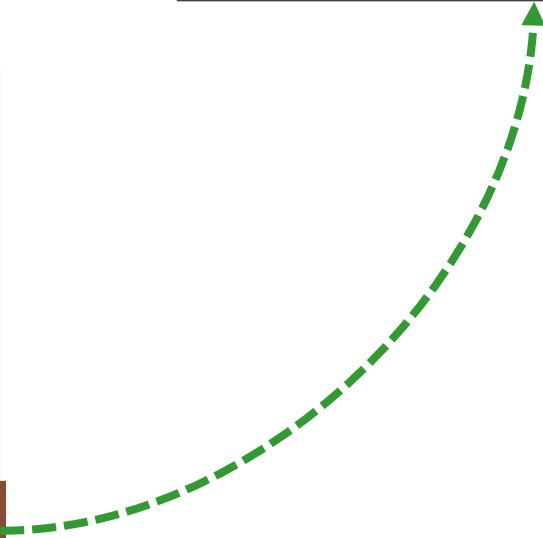
= **Table.CombineColumns** (t , {"Libro" , "Año"} , (x as list) => x{1} & x{1} , "Nueva Columna")

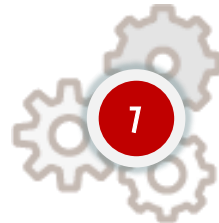


Libro	Año	Precio
Tablas Dinámicas la Quinta Dimensión	2015	40
El ADN de Power Pivot	2016	60
Inteligencia de Negocios con Excel y Power BI	2018	70
El ADN de Power Query	2019	60
Power BI Reportes e Informes	2020	70

X =

List
Power BI para Reportes e Informes
2020





Funciones en Lenguaje M Que Iteran

La Syntax Sugar each la definición de Función Personalizada



Let's Hack!

Ejercicio I — Table.CombineColumn:

Combinar las columnas Libro y Precio de la siguiente forma: 60: Power BI para Reportes e Informes, etc, es decir, en al combinación de los valores deben quedar invertidos



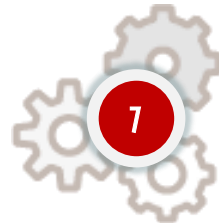
Funciones en Lenguaje M Que Iteran

La Syntax Sugar each la definición de Función Personalizada

Solución A: Ejercicio 1



```
let
  tl =
    Table.CombineColumns(
      t ,
      { "Libro" , "Precio" } ,
      each
        Text.From( _{l} ) & " : " & _{0} ,
        "Precio y Libro"
    )
in
  tl
```



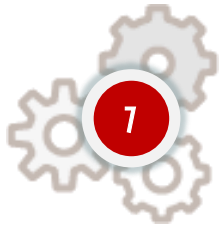
Funciones en Lenguaje M Que Iteran

Funciones de la Categoría Combiner

Función	Descripción
Combiner.CombineTextByDelimiter	Retorna una función que combina una lista de textos en una sola cadena de textos utilizando el delimitador especificado.
Combiner.CombineTextByEachDelimiter	Retorna una función que combina una lista de textos en una sola cadena de textos utilizando cada delimitador especificado. Es ve reflejado en la combinación de tres a más cadenas de texto.
Combiner.CombineTextByLengths	Retorna una función que combina una lista de textos en una sola cadena de textos utilizando una longitud determinada.
Combiner.CombineTextByPositions	Retorna una función que combina una lista de textos en una sola cadena de textos utilizando las posiciones especificadas.
Combiner.CombineTextByRanges	Retorna una función que combina una lista de textos en una sola cadena de textos utilizando las longitudes y posiciones especificadas.

Funciones en Lenguaje M Que Iteran

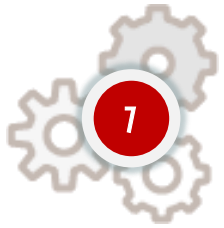
Funciones de la Categoría Combiner



Solución B: Ejercicio I



```
let
    tl =
        Table.CombineColumns (
            t ,
            { "Libro" , "Precio" } ,
            Combiner.CombineTextByDelimiter ( ":" )
        )
in
    tl
```



Funciones en Lenguaje M Que Iteran

Funciones de la Categoría Splitter

=

Table.SplitColumn (

Tabla como valor **table** ,

i ITERADOR

ColumnaFuente como valor **text** ,

i TRANSFORMACIÓN A...

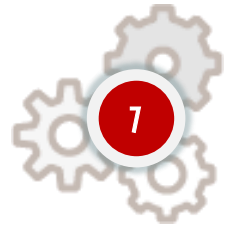
Divisor como valor **function** ,

(IteraciónActual como valor **list**) como valor **any**

NombreDeColumna como valor **text** ,

)

como valor **table**



Funciones en Lenguaje M Que Iteran

Funciones de la Categoría Splitter



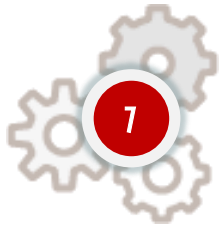
Let's Hack!

Ejercicio I — Table.SplitColumn:

Dividir la columna Libro y Precio del ejercicio: Ejercicio I — Table.CombineColumns por el carácter “:”, sin apoyo de la interfaz directamente de la interfaz.

Funciones en Lenguaje M Que Iteran

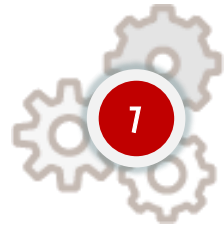
Funciones de la Categoría Splitter



Solución A: Ejercicio 1



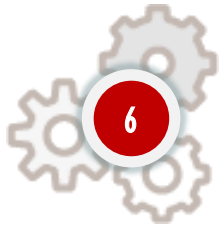
```
let
  t1 =
    Table.CombineColumns (
      t ,
      { "Libro" , "Precio" } ,
      Combiner.CombineTextByDelimiter ( ":" )
    ) ,
  t2 =
    Table.SplitColumn (
      t1
      "Libro y Precio" ,
      ( x ) =>
        Text.Split ( x , ":" )
    )
in
  t2
```



Funciones en Lenguaje M Que Iteran

Funciones de la Categoría Splitter

Función	Función
Splitter.SplitByNothing	Splitter.SplitTextByAnyDelimiter
Splitter.SplitTextByCharacterTransition	Splitter.SplitTextByDelimiter
Splitter.SplitTextByEachDelimiter	Splitter.SplitTextByLengths
Splitter.SplitTextByPositions	Splitter.SplitTextByRanges



Funciones en Lenguaje M Que Iteran

Función Anónima eacha para syntax sugar de función personalizada

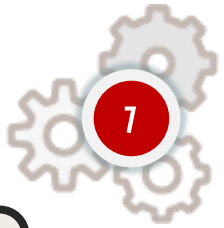
Función del Lenguaje M que Itera en Grupos

Columna Dinámica (Pivot)



Table.Pivot (Tabla , ListaDeElementos , Atributo , Valor , FunciónDeAgregación)

TABLA LISTA TEXTO TEXTO FUNCIÓN



Funciones en Lenguaje M Que Iteran

Función Anónima each para syntax sugar de función personalizada

=

Table.Pivot (

Tabla como valor **table** ,

i ITERADOR

ValoresPivot como valor **list** ,

NombreDeColumnaPivot como valor **text** ,

NombreDeColumnaDeValores como valor **text** ,

FunciónDeAgregación como valor **function** ,

(**GrupoPivot** como valor **list**) como valor **any**

)

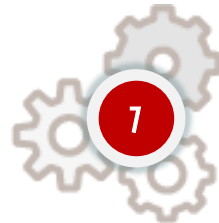
como valor **table**

Integración de Datos con Diversas Topologías de Tabla

Fibonacci:

0, 1, 1, 2, 3, 5, 8, 13, ...

$$\frac{a+b}{a} = \frac{b}{a} \approx 1,618$$



Funciones en Lenguaje M Que Iteran

Funciones de la Categoría Splitter



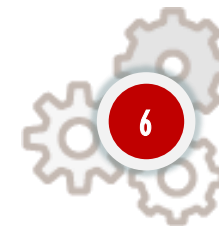
Let's Hack!

Ejemplo — Stacked Table:

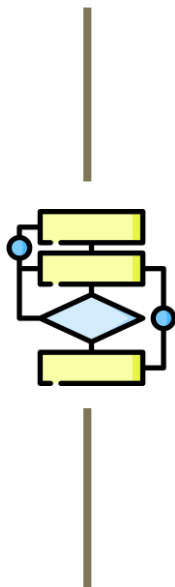
Crear la función personalizada para transformar una tabla con el formato apilado (Stacked Table) donde todos los nombres de campos son coincidentes, y luego utilizarla para integrar k tablas con el formato apilado en el archivo de Excel: 5 - StackedToTabular de K Tablas

Limpieza de Datos de 2º – Técnicas de Dinamización por Transposición

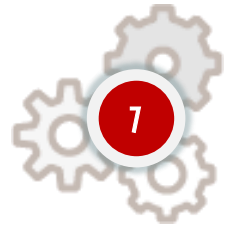
Stacked Table: Generalización Campos Coincidentes



¡Algoritmo!



1. Agregar Columna de Índice
2. Columna dinámica con columna 1 como dinámica y columna 2 como valor
3. Rellenar arriba de la columna tres en adelante
4. Remover null de la columna número dos.
5. Remover columna de índice
6. Aplicar tipo de datos a todas las columnas



Funciones en Lenguaje M Que Iteran

Funciones de la Categoría Splitter



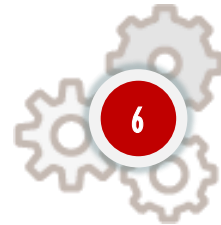
Let's Hack!

Ejercicio I —Pivot Table NxM:

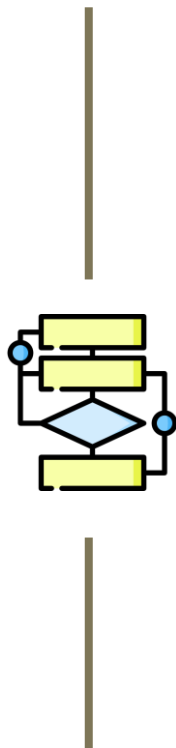
Crear la función personalizada para transformar una tabla con el formato de tabla dinámica Nx,M (Pivot Table) donde todos, y luego utilizarla para integrar k tablas con el formato de tabla dinámica en el archivo de Excel: *8 - PivotNxMToTabular de K Tablas.*

Limpieza de Datos de 2º – Técnicas de Dinamización por Transposición

Formato Pivot Table NxM – Pseudo-Generalización



¡Algoritmo!



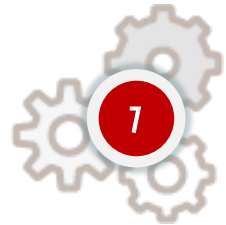
1. Rellenar abajo n-1 columnas
2. Combinar la n primeras columnas
3. Transponer
4. Rellenar abajo m-1 primeras columnas
5. Promover encabezado
6. Anular dinamización de otras columnas con m seleccionadas
7. Dividir columna Atributo por delimitador asociado
8. Renombras Columnas

Parámetros en Power Query *La Consulta y Función de Muestra*

Fibonacci:

0, 1, 1, 2, 3, 5, 8, 13, ...

$$\frac{a+b}{a} = \frac{b}{a} \approx 1,618$$



Parámetros y Metadatos en el Lenguaje M

Tipo especial de identificador y Datos sobre un Dato

Meta datos

Los meta datos son datos acerca de un valor, y en el lenguaje M esos datos asociados se representan con un valor de tipo registro, denominado registro de meta dato. Todo valor en M tiene meta datos, si no ha sido especificado dicho registro simplemente esta vacío.



```
let
    a = "El ADN de Power Query" meta [Precio=60, publicacion=2019, edicion=1]
in a
```

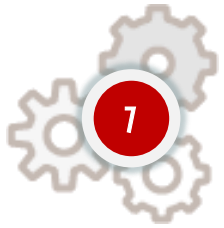


```
a = "El ADN de Power Query" meta [Precio=60, publicacion=2019, edicion=1]
```

Acceder a un valor de los metadatos se debe hacer con la función: **Value.Metadata**



```
let
    a = "El ADN de Power Query" meta [Precio=60, publicacion=2019, edicion=1]
in
    Value.Metadata ( a ) [Precio]
```

Parámetros y Metadatos en el Lenguaje M

Tipo especial de identificador y Datos sobre un Dato

Parámetro

Los parámetros en el lenguaje M son un tipo especial de identificador que tiene como fin dar facilidad en la inserción de valores estáticos de forma amigable mediante cuadro de diálogo o similar si la necesidad de un enter.



```
MuestraBinaria meta [IsParameterQuery=true, BinaryIdentifier=MuestraBinaria, Type="Binary", IsParameterQueryRequired=true]
```

Cuando se crea una función a partir de un parámetro estas se retroalimentan entre si automáticamente.



```
MuestraTabla meta [IsParameterQuery=true, IsParameterQueryRequired=false]
```