let

/\* A Tabela é onde possui uma coluna Data que deseja usar em seus cálculos ou visuais.

Exemplo. Tabela fVendas que possui a coluna Data Pedido Ficaria List.Min (fVendas[Data Pedido]).

Se Possuir mais de uma coluna ficaria List.Min (fVendas[Data Pedido] & fVendas[Data Vencimento]).

Criando uma lista de Datas únicas e Bufferizando >> Insira nesta linha a Culuna de datas do seu modelo \*/

Datas = List.Buffer(

List.Distinct(

teste[Data Inicial] & teste[Data Final]

)

), // listar as tabelas de datas das suas fatos

//Identifica a menor data

MenorData = List.Min(Datas),

//Identifica a maior data

MaiorData = List.Max(Datas),

//Retorna ao inicio do Ano da menor data

DataInicio = Date.StartOfYear(MenorData),

//Vai para fim do ANO da maior data

DataFim = Date.EndOfYear(MaiorData),

//Quantidade de dias da dCalendário

Dias = Duration.Days(DataFim - DataInicio) + 1,

//Cria a lista de datas

ListarDatas =

List.Dates(

DataInicio,

Dias,

#duration(1, 0, 0, 0)

),

//Identifica o dia de hoje

Hoje = Date.From(DateTime.LocalNow()),

//Identifica o MesAno do dia de hoje

Hoje\_MesAno = Date.Year(Hoje) \* 100 + Date.Month(Hoje),

// Definir os nomes de Colunas e seus Tipos

Tabela =

#table(

type table [

Data = date,

Ano = Int64.Type,

//colunas para ordenação e podem ser ocultadas

INTMes = Int64.Type,

INTAnoMes = Int64.Type,

INTTrimestre = Int64.Type,

INTDiaSemana = Int64.Type,

INTSemanaMes = Int64.Type,

//INTSemanaAno = Int64.Type,

//Colunas análises temporais

Mês = text,

MesAbre = text,

MesAno = text,

Trimestre = text,

Bimestre = text,

Semestre = text,

Semana = text,

//SemanaAno = text,

Inicio da Semana = date,

Fim da Semana = date,

NomeDia = text,

Mês Atual = text,

Até Ontem = text,

Ciclo13m = text

],

List.Transform(

ListarDatas,

each

{

//na oredem da lista acima defina o cálculo das colunas

\_, //Data

Date.Year(\_), //Ano

//Colunas INTEIROS para ordenação

Date.Month(\_), //INTMes

Date.Year(\_) \* 100 + Date.Month(\_), //INTAnoMes

Date.QuarterOfYear(\_), //INTTrimestre

Date.DayOfWeek(\_), //INTDiaSemana

Date.WeekOfMonth(\_), //INTSemanaMes

//Date.WeekofYear(\_), //INTSemanaAno

//Colunas

Text.Proper(Date.MonthName(\_)), //Mês

Text.Proper(Text.Start(Date.MonthName(\_), 3)), //MesAbre

Text.Proper(Text.Start(Date.MonthName(\_), 3)) & "-" & Text.End(Text.From(Date.Year(\_)), 2), //MesAno

Text.From(Date.QuarterOfYear(\_)) & "º Tri ", //Trimestre

Text.From(Number.RoundUp(Date.Month(\_) / 2, 0)) & "º Bim", //Bimestre

Text.From(Number.RoundUp(Date.Month(\_) / 6, 0)) & "º Sem", //Semestre

"s." & Text.From(Number.RoundUp(Date.WeekOfMonth(\_) / 2, 0)), //Semana do Mês

//"s." & Text.From(Number.RoundUp(Date.WeekOfYear(\_) / 2, 0)), //Semana do Ano

Date.StartOfWeek(\_),

Date.EndOfWeek(\_),

Date.DayOfWeekName(\_),

// Calcula o AnoMês Atual

if Date.Year(\_) \* 100 + Date.Month(\_) = Hoje\_MesAno

then "Atual"

else Text.Proper(Text.Start(Date.MonthName(\_), 3)) & "-" & Text.End(Text.From(Date.Year(\_)), 2),

// Calcula o dia até Ontem

if (\_) < Hoje

then "Até ontem"

else Text.BeforeDelimiter(Text.From(\_, "pt-BR"), "/"),

// Calcula ultimos meses 13 meses

if (\_) <= Date.EndOfMonth(Hoje)

then

if (\_) >= Date.StartOfMonth(Date.AddDays(Hoje,-365))

then "13 meses"

else "desconsiderar"

else "desconsiderar"

}

)

)

in

Tabela