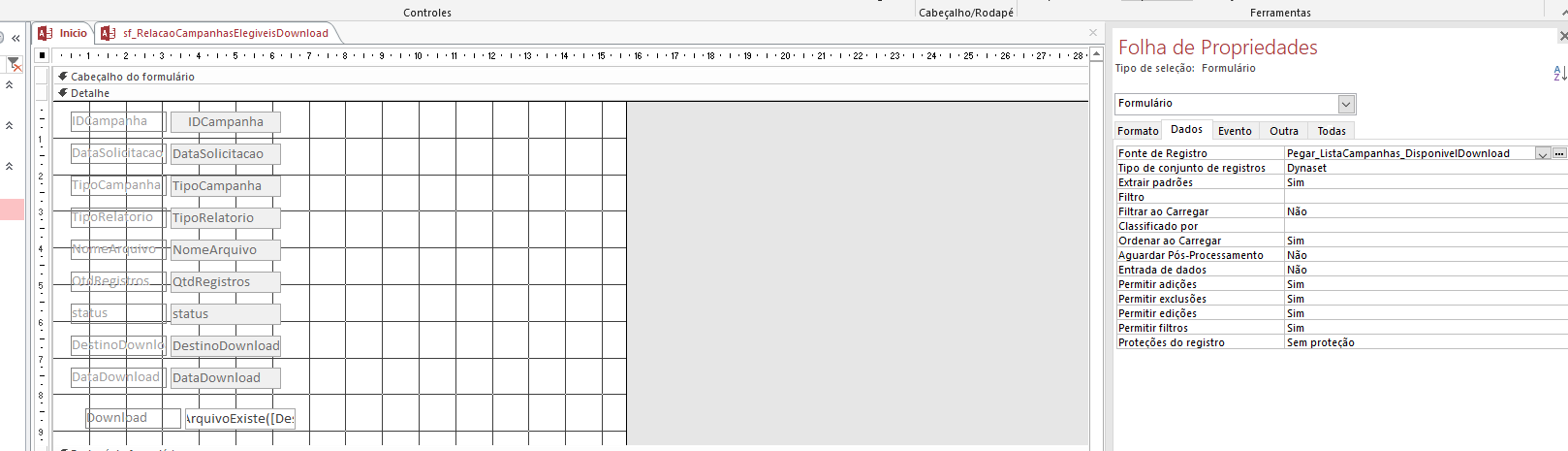
Atualizações que devem ser realizadas na versão da Rede

## UI – Interface

* Criar um subform "sf\_RelacaoCampanhasElegiveisDownload" que irá listar os downloads
* Apontar o subform para a consulta "Pegar\_ListaCampanhas\_DisponivelDownload"
* Acrescentar os campos da consulta conforme abaixo :

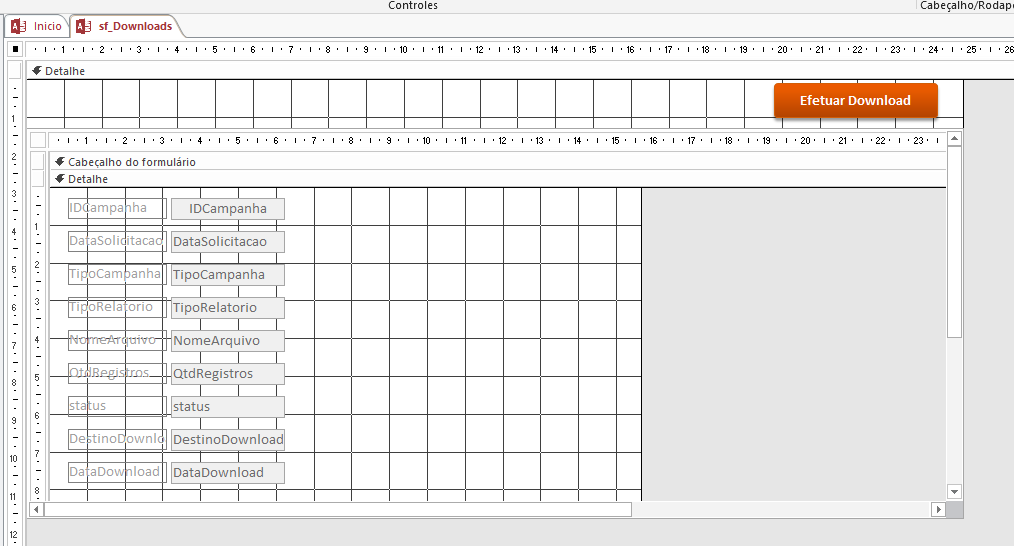


* Criar consulta “Pegar\_ListaCampanhas\_DisponivelDownload” com o seguinte código :

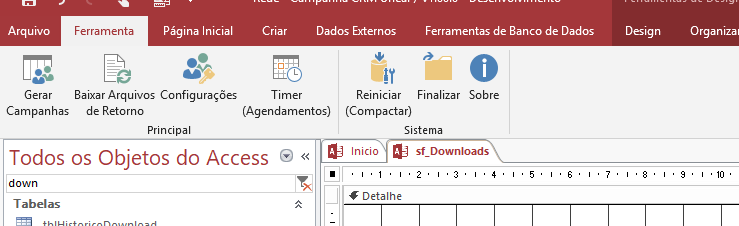
*SELECT b.\**

*FROM tblHistoricoDownload AS b;*

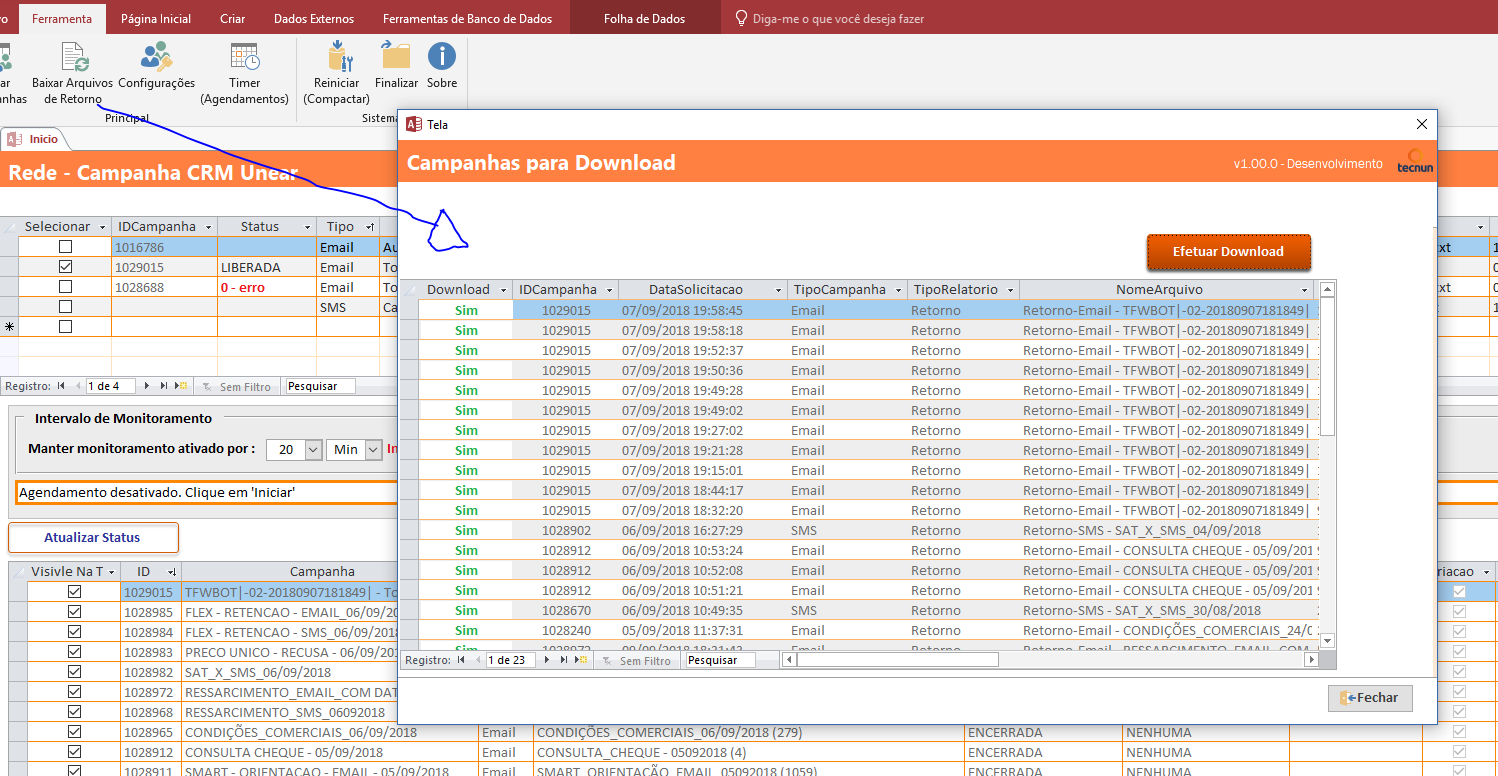
* Criar o sub form “sf\_Downloads” e apontar como fonte o form acima com a seguinte aparência :



* Modificar o botão da Ribbon “Baixar Do” para abrir o form com a lista de downloads



Após essas modificações o form ficará assim :



## Código - VBA

**Modulo : mBL\_CRMUnear**

Criar/Alterar as seguintes funções :

**BaixarArquivosRetorno()/ CRIAR**

**ObtemStatusDownloadParaCampanha()/ CRIAR**

**ArquivoExiste()/ CRIAR**

**BaixarListaCampanhas**()**/ ALTERAR**

**ValidaParametros() / ALTERAR**

**Codigo:**

Sub **BaixarArquivosRetorno**()

'Tratamento de Erro

'---------------------------------------------------------------------------------------

1 On Error GoTo BaixarArquivosRetorno\_Error

Dim lngErrorNumber As Long, strErrorMessagem As String:

Const cstr\_ProcedureName As String = "cBL\_Unear.BaixarArquivosRetorno()"

'---------------------------------------------------------------------------------------

Dim rs As Object

Dim qtdBaixou As Integer

Dim cUnear As New cBL\_Unear

Dim bRetorno, strArquivoLog As String

If MsgBox("Deseja efetuar o download dos arquivos da lista ?", vbQuestion + vbYesNo) = vbNo Then Exit Sub

strArquivoLog = PegaEndereco\_Programa & "\Log\Downloads.txt"

cUnear.LogFile = strArquivoLog

showMessage "Preparando para baixar os arquivos..."

Call RegistraStatus(Mensagem:="Download de Arquivos - Log Iniciado............", LogTxt:=strArquivoLog, Progresso:=True, IncluirDataHoraLog:=False)

Call RegistraStatus(Mensagem:="Data/Hora : " & VBA.Now(), LogTxt:=strArquivoLog, Progresso:=False, IncluirDataHoraLog:=False)

Call RegistraStatus(Mensagem:="Acessando o site............", LogTxt:=strArquivoLog, Progresso:=False, IncluirDataHoraLog:=False)

'Acessa o site e garante que a lista de campanhas esteja selecionada

Call cUnear.AcessarSite

Call cUnear.Entrar(Unear\_Usuario(), Unear\_Senha())

Inicializar\_Globais

Call RegistraStatus(Mensagem:="Carregando a lista de downloads disponivel............", LogTxt:=strArquivoLog, Progresso:=False, IncluirDataHoraLog:=False)

cUnear.ObtemListaDeArquivosDeCampanhas (0)

Call RegistraStatus(Mensagem:="Iniciando o download dos arquivos............", LogTxt:=strArquivoLog, Progresso:=False, IncluirDataHoraLog:=False)

Set rs = Conexao.PegarRS("Pegar\_ListaCampanhas\_DisponivelDownload")

showMessage "Efetuando o download..."

Do While Not rs.EOF

'Invoca a função de download dos arquivos

bRetorno = cUnear.BaixarArquivoDeRetorno(rs!IDCampanha.value, \_

rs!OrigemDownload.value, \_

rs!NomeArquivo.value)

If bRetorno(0) = 1 Then

Call AtualizaDadosCampanha(rs!IDCampanha.value, "Download", -1)

qtdBaixou = qtdBaixou + 1

End If

rs.MoveNext

Loop

Call RegistraStatus(Mensagem:="Download dos arquivos foi concluído...........", LogTxt:=strArquivoLog, Progresso:=True, IncluirDataHoraLog:=False)

showMessage "Download concluído !" & vbNewLine & vbNewLine & "Qtd De Arquivos Baixados : " & qtdBaixou, "Download", Fim, , , , Icone\_Sucesso

Fim:

9 On Error GoTo 0

10 Exit Sub

BaixarArquivosRetorno\_Error:

11 If VBA.Err <> 0 Then

12 lngErrorNumber = VBA.Err.Number: strErrorMessagem = VBA.Err.Description

13 Call VBA.Err.Raise(VBA.Err.Number, cstr\_ProcedureName, VBA.vbNewLine & cstr\_ProcedureName & " Em" & VBA.IIf(VBA.Erl() <> 0, " > Linha (" & VBA.Erl() & ")", "") & " > " & VBA.Err.Description)

14 End If

GoTo Fim:

'Debug Mode

15 Resume

End Sub

Function **ObtemStatusDownloadParaCampanha**(intCampanha As Long)

ObtemStatusDownloadParaCampanha = ArquivoExiste(Nz(DLookup("DestinoDownload", "tblHistoricoDownload", "IDCampanha = " & intCampanha), "--"))

End Function

Function **ArquivoExiste**(arquivo As String) As Boolean

On Error Resume Next

ArquivoExiste = VBA.Dir(arquivo) <> ""

End Function

Sub **BaixarListaCampanhas**(Optional vArgs As Variant, Optional bShowMsg As Boolean = True, Optional sTimerID As String)

Dim rsListaDisparo As Object

Dim strSelecao As String, strCampanha As String

Dim bRetorno

Dim intID As Long

Dim intSegundos As Integer

'Tratamento de Erro

'---------------------------------------------------------------------------------------

1 On Error GoTo AtualizarStatusDaLista\_Error

Dim lngErrorNumber As Long, strErrorMessagem As String:

Const cstr\_ProcedureName As String = "mBL\_CRMUnear.BaixarListaCampanhas()"

'---------------------------------------------------------------------------------------

intSegundos = 2

2 Inicializar\_Globais

Dim strStatus As String

Dim strTempTabela As String

3 If Not cUnear.BrowserEstaAberto Then

If bShowMsg Then MsgBox "É necessário que o navegador esteja aberto e com o Sistema Unear logado !", vbExclamation

GoTo Fim

6 End If

If Not cUnear.CRM\_Conectado Then

If bShowMsg Then MsgBox "É necessário Conectado ao Unear !", vbExclamation

GoTo Fim

End If

If bShowMsg Then showMessage "Aguarde, Selecionando a lista atualizada."

cUnear.AcessarListaDeCampanhas

FecharProgresso

strTempTabela = cUnear.DownloadLista("Campanha", "bsDataTable")

8 Set rsListaDisparo = CurrentDb.OpenRecordset("SELECT a.\* FROM [" & strTempTabela & "] AS a WHERE 1=1 OR a.Campanha Like 'TFWBOT\*';")

9 Do While Not rsListaDisparo.EOF

VBA.DoEvents

10 If bShowMsg Then Call showMessage("Recuperando lista atualizada das campanhas '" & intID & "'", "Unear")

11 intID = rsListaDisparo!ID.value

12 strStatus = rsListaDisparo!status.value

13 Call AtualizaDadosCampanha(intID, "Status", strStatus)

Call AtualizaDadosCampanha(intID, "Campanha", rsListaDisparo!Campanha.value)

Call AtualizaDadosCampanha(intID, "TipoCampanha", rsListaDisparo!TIPO.value)

Call AtualizaDadosCampanha(intID, "SelecaoUsuarios", rsListaDisparo![Seleção de Usuários].value)

Call AtualizaDadosCampanha(intID, "Criacao", -1)

Call AtualizaDadosCampanha(intID, "Selecionar", -1)

Call AtualizaDadosCampanha(intID, "VisivelNaTela", -1)

14 Select Case strStatus

Case "ATIVA"

15 Call AtualizaDadosCampanha(intID, "AcaoPendente", "EXECUTAR")

16 Case "CADASTRADA"

17 Call AtualizaDadosCampanha(intID, "AcaoPendente", "TESTAR")

18 Case "SMS TESTANDO", "TESTANDO"

19 Call AtualizaDadosCampanha(intID, "AcaoPendente", "EM TESTE")

20 Case "SMS LIBERADA", "GERANDO SMS (0%)"

21 Call AtualizaDadosCampanha(intID, "AcaoPendente", strStatus & " (EM PROCESSAMENTO)")

22 Case "ENCERRADA"

23 Call AtualizaDadosCampanha(intID, "AcaoPendente", "NENHUMA")

Call AtualizaDadosCampanha(intID, "Inicio", rsListaDisparo!Início.value)

Call AtualizaDadosCampanha(intID, "Fim", rsListaDisparo!Fim.value)

Call AtualizaDadosCampanha(intID, "Download", VBA.CInt(ObtemStatusDownloadParaCampanha(intID)) \* -1)

24 Case Else

25 Call AtualizaDadosCampanha(intID, "AcaoPendente", strStatus)

26 End Select

27 If strStatus = "ENCERRADA" Then

28 Call AtualizaDadosCampanha(intID, "Execucao", -1)

29 Call AtualizaDadosCampanha(intID, "Teste", -1)

30 End If

31 rsListaDisparo.MoveNext

32 Loop

rsListaDisparo.Close

Set rsListaDisparo = Nothing

Call ExcluiTabela(strTempTabela)

33 If bShowMsg Then showMessage "Status atualizados com sucesso !", , Fim, , , , Icone\_Sucesso, intSegundos

Fim:

34 On Error GoTo 0

35 Exit Sub

AtualizarStatusDaLista\_Error:

36 If Err <> 0 Then

37 lngErrorNumber = VBA.Err.Number: strErrorMessagem = VBA.Err.Description

38 Call VBA.Err.Raise(VBA.Err.Number, cstr\_ProcedureName, VBA.vbNewLine & cstr\_ProcedureName & " Em" & VBA.IIf(VBA.Erl() <> 0, " > Linha (" & VBA.Erl() & ")", "") & " > " & VBA.Err.Description)

39 End If

GoTo Fim:

40 Resume 'Debug Mode

End Sub

Function **ValidaParametros**(rs As Object) As Collection

'----------------------------------------------------------------------------------------------------

On Error GoTo TratarErro: Const cstr\_ProcedureName As String = "TFWCliente.mBL\_CRMUnear.ValidaParametros"

'------------------------------------------------------------------------------------------------

Dim cF As New Collection

Dim i As Integer

Dim vInfoClick As Variant

Dim strArquivoSelecao As String

For i = 0 To rs.Fields.count - 1

'Preenchimento

Select Case rs.Fields(i).Name

Case "TipoCampanha", "Campanha", "ArquivoSelecaoUsuarios", "ModeloEmail", "Assunto"

If rs.Fields(i).Name = "ModeloEmail" And rs.Fields("TipoCampanha").value = "SMS" Then

Else

If Access.Nz(rs.Fields(i).value) = "" Then cF.Add rs.Fields(i).Name & " - Preenchimento obrigatório !"

If rs.Fields(i).Name = "ArquivoSelecaoUsuarios" Then

strArquivoSelecao = PegarArquivoLista(Access.Nz(rs!ArquivoSelecaoUsuarios.value), VBA.Date())

If Not FileExists(strArquivoSelecao) Then

cF.Add rs.Fields(i).Name & " - Arquivo não localizado ! > " & Nz(rs!ArquivoSelecaoUsuarios.value)

End If

End If

End If

Case Else

End Select

Next i

vInfoClick = cUnear.BuscaInfoPonteito("MenuOpcoes"): If VBA.IsArray(vInfoClick) Then If vInfoClick(0) = 0 Then cF.Add "MenuOpcoes - Posição do Clique não configurado!"

vInfoClick = cUnear.BuscaInfoPonteito("MenuSelecoes"): If VBA.IsArray(vInfoClick) Then If vInfoClick(0) = 0 Then cF.Add "MenuSelecoes - Posição do Clique não configurado!"

vInfoClick = cUnear.BuscaInfoPonteito("BotaoUploadSelecao"): If VBA.IsArray(vInfoClick) Then If vInfoClick(0) = 0 Then cF.Add "BotaoUploadSelecao - Posição do Clique não configurado!"

vInfoClick = cUnear.BuscaInfoPonteito("SelecaoModeloEmail\_Editar"): If VBA.IsArray(vInfoClick) Then If vInfoClick(0) = 0 Then cF.Add "SelecaoModeloEmail\_Editar - Posição do Clique não configurado!"

vInfoClick = cUnear.BuscaInfoPonteito("SelecaoModeloEmail\_Selecionar"): If VBA.IsArray(vInfoClick) Then If vInfoClick(0) = 0 Then cF.Add "SelecaoModeloEmail\_Selecionar - Posição do Clique não configurado!"

vInfoClick = cUnear.BuscaInfoPonteito("OpcaoSMS"): If VBA.IsArray(vInfoClick) Then If vInfoClick(0) = 0 Then cF.Add "OpcaoSMS - Posição do Clique não configurado!"

Set ValidaParametros = cF

On Error GoTo 0

Fim:

Exit Function

TratarErro:

If VBA.Err <> 0 Then Call VBA.Err.Raise(VBA.Rnd(), cstr\_ProcedureName & " - " & VBA.Erl() & ">" & VBA.Err.Description & VBA.vbCrLf)

GoTo Fim:

Resume

End Function

**Modulo : cBL\_Unear**

Criar/Alterar as seguintes funções :

**ObtemListaDeArquivosDeCampanhas()/ CRIAR**

**BaixarArquivoDeRetorno()/ CRIAR**

**CriarTabelaDownloadLog()/ CRIAR**

**DownloadLista**()**/ ALTERAR**

**ValidaParametros() / ALTERAR**

'---------------------------------------------------------------------------------------

' PROCEDIMENTO : cBL\_Unear.DownloadArquivosDeCampanhas()

' TIPO : Function

' DATA/HORA : 07/09/2018 19:00

' CONSULTOR : TECNUN - Adelson Rosendo Marques da Silva (adelson@tecnun.com.br)

' DESCRIÇÃO : Acessa e direciona para download dos arquivos

'---------------------------------------------------------------------------------------

Function **ObtemListaDeArquivosDeCampanhas**(strCampanhaSelecionada As String)

'Tratamento de Erro

'---------------------------------------------------------------------------------------

1 On Error GoTo BaixarArquivosRetorno\_Error

Dim lngErrorNumber As Long, strErrorMessagem As String:

Const cstr\_ProcedureName As String = "cBL\_Unear.DownloadArquivosDeCampanhas()"

'---------------------------------------------------------------------------------------

Dim Tabela As Object

Dim i As Integer, y As Integer

Dim Campanha As Object

Dim strCampanha As String

Dim strNomeArquivo As String

Dim strLinkDownload As String

Dim qtdRegistro As String

Dim strDataSolic As String

Dim qtdEncontradas As Integer

Dim qtdBaixadas As Integer

Call EnviaLogStatus(VBA.String(120, ".")): Call EnviaLogStatus("INICIANDO > " & cstr\_ProcedureName): Call EnviaLogStatus(VBA.String(120, "."))

Call EnviaLogStatus("Seleciona a primeira campanha e clica no botao Lista de Downloads")

vRetorno = Array(-1, "Nada executado")

'Se tiver na tela de campanhas,

If cUnear.CRM\_TelaIndex = 2 Then

'Solicita o arquivo de retorno

'Call ProcessarAcaoMenuCampanha("downloadResponse", strCampanhaSelecionada)

'Acessa a lista de arquivos

Call ProcessarAcaoMenuCampanha("listdownloads", strCampanhaSelecionada)

Else

'Se ja tive o ID campaignDownloadList, indica que a tela de download ja esta ativa

If IEApp.ControlExists("campaignDownloadList") Then

Debug.Print "Tela de Download ja ativa"

End If

End If

Call EnviaLogStatus("Aguarda o carregamento da tela")

'Aguarda pelo ID, que esta na pagina de downloads

IEApp.WaitFor "campaignDownloadList", STOP\_WHEN\_FIND\_ID

Call AguardaCarregamentoTabela

'Garante q a tabela exista

Call CriarTabelaDownloadLog

Dim paginas As Collection

Dim pg As HTMLLIElement

Dim page As cIE\_Control

Dim qtdPaginas As Integer

Dim iPg As Integer

Dim bNovo As Boolean

Call EnviaLogStatus("Efetua a leitura da tabela...")

'Efetua a leitura da tabela linha a linha do que esta disponivel

Set Tabela = IEApp.Control("bsDataTable")

'Recupera todos os elementos que representa os links de cada pagina disponivel

Set paginas = IEApp.DOMSelect("page-number")

qtdPaginas = paginas.count

Call EnviaLogStatus(qtdPaginas & " - Paginas ")

For iPg = 1 To qtdPaginas

'A pesquisa deve ser refeita para atualizar a instancia dos objetos

Set paginas = IEApp.DOMSelect("page-number")

Set pg = paginas(iPg).element

'Para a pagina ativa nao precisa clicar

If Not pg.ClassName Like "\*active" Then

Call pg.Click

'Aguarda 2 segundos para garantir o carregamento da pagina

win.Wait 2000

End If

Call EnviaLogStatus("Pagina " & iPg & " Ativada !")

'Cada Linha

For i = 0 To Tabela.tBodies.item(0).Rows.Length - 1

strDataSolic = Tabela.tBodies.item(0).Rows(i).Cells(0).innerText

strCampanha = Tabela.tBodies.item(0).Rows(i).Cells(3).innerText

strNomeArquivo = Tabela.tBodies.item(0).Rows(i).Cells(4).innerText

'If strCampanha = strCampanhaSelecionada Then

qtdEncontradas = qtdEncontradas + 1

If Tabela.tBodies.item(0).Rows(i).Cells(4).Children.Length > 0 Then

strLinkDownload = Tabela.tBodies.item(0).Rows(i).Cells(4).Children(0).href

Else

strLinkDownload = "#Arquivo Nao disponivel para download ainda"

End If

Debug.Print "Solicitando o download de : Campanha : " & strCampanha & " : Arquivo > " & strLinkDownload

'Efetua o downloa e salva na tabela de log de downloads

With CurrentDb.OpenRecordset("SELECT \* FROM tblHistoricoDownload WHERE IDCampanha = " & strCampanha & " and OrigemDownload = '" & strLinkDownload & "'")

bNovo = False

If .EOF Then

.addNew

bNovo = True

Else

.edit

End If

.Fields("Pagina").value = iPg

.Fields("Linha").value = i + 1

.Fields("IDCampanha").value = VBA.CLng(strCampanha)

.Fields("TipoCampanha").value = Tabela.tBodies.item(0).Rows(i).Cells(1).innerText

.Fields("TipoRelatorio").value = Tabela.tBodies.item(0).Rows(i).Cells(2).innerText

.Fields("NomeArquivo").value = .Fields("TipoRelatorio").value & "-" & .Fields("TipoCampanha").value & " - " & strNomeArquivo

.Fields("OrigemDownload").value = strLinkDownload

.Fields("DataSolicitacao").value = VBA.CDate(strDataSolic)

If bNovo Then

.Fields("status").value = "Novo"

Else

.Fields("status").value = "Ja atualizou"

End If

.Fields("QtdRegistros").value = Tabela.tBodies.item(0).Rows(i).Cells(6).innerText

.Update

Call EnviaLogStatus("...Solicitando o download de : Campanha : " & strCampanha & " : Arquivo > " & strLinkDownload)

End With

Next i

Next iPg

vRetorno = Array(-1, qtdEncontradas, qtdBaixadas)

Call EnviaLogStatus("Lista atualizada !")

Fim:

ObtemListaDeArquivosDeCampanhas = vRetorno

On Error GoTo 0

10 Exit Function

BaixarArquivosRetorno\_Error:

11 If VBA.Err <> 0 Then

12 lngErrorNumber = VBA.Err.Number: strErrorMessagem = VBA.Err.Description

13 Call VBA.Err.Raise(VBA.Err.Number, cstr\_ProcedureName, VBA.vbNewLine & cstr\_ProcedureName & " Em" & VBA.IIf(VBA.Erl() <> 0, " > Linha (" & VBA.Erl() & ")", "") & " > " & VBA.Err.Description)

14 End If

GoTo Fim:

'Debug Mode

15 Resume

End Function

'---------------------------------------------------------------------------------------

' PROCEDIMENTO : cBL\_Unear.BaixarArquivoDeRetorno()

' TIPO : Function

' DATA/HORA : 07/09/2018 19:00

' CONSULTOR : TECNUN - Adelson Rosendo Marques da Silva (adelson@tecnun.com.br)

' DESCRIÇÃO : Efetua o download da campanha e salva no diretorio

'---------------------------------------------------------------------------------------

Function **BaixarArquivoDeRetorno**(Campanha, linkDownload As String, NomeArquivo As String) As Variant

'Tratamento de Erro

'---------------------------------------------------------------------------------------

1 On Error GoTo BaixarArquivoDeRetorno\_Error

Dim lngErrorNumber As Long, strErrorMessagem As String:

Const cstr\_ProcedureName As String = "cBL\_Unear.BaixarArquivosRetorno()"

'---------------------------------------------------------------------------------------

Dim strPastaDownload As String

Dim strNomeArquivo As String

Dim strArquivoDestino As String

Dim status As String

Dim strNomeInterno As String

strNomeInterno = VBA.StrReverse(VBA.Left(VBA.StrReverse(linkDownload), VBA.InStr(VBA.StrReverse(linkDownload), "/") - 1))

strNomeArquivo = Campanha & " - " & NomeArquivo & "\_" & strNomeInterno '.csv"

strPastaDownload = PegaEndereco\_Relatorios()

strArquivoDestino = strPastaDownload & "\" & strNomeArquivo

Call EnviaLogStatus(VBA.String(120, ".")): Call EnviaLogStatus("INICIANDO > " & cstr\_ProcedureName): Call EnviaLogStatus(VBA.String(120, "."))

Call EnviaLogStatus("Preparando para o download do arquivos..")

Call EnviaLogStatus(" Campanha :........." & Campanha)

Call EnviaLogStatus(" Link de Origem : .." & linkDownload)

Call EnviaLogStatus(" Destino: .........." & strArquivoDestino)

Dim ie As New cTFW\_WebUtil

Call ie.Download(linkDownload, API, strArquivoDestino, True)

Call EnviaLogStatus("Download concluído !")

If VBA.Dir(strArquivoDestino) <> "" Then

status = "1"

Else

status = "0"

End If

Call EnviaLogStatus("Salvando log..")

With CurrentDb.OpenRecordset("SELECT \* FROM tblHistoricoDownload WHERE IDCampanha = " & Campanha & " and OrigemDownload = '" & linkDownload & "'")

.addNew

If .EOF Then .addNew Else .edit

.Fields("DestinoDownload").value = strArquivoDestino

.Fields("status").value = VBA.IIf(status = "1", "Download efetuado", "Erro no download")

.Fields("DataDownload").value = VBA.Now()

.Update

End With

vRetorno = Array(1, 0)

BaixarArquivoDeRetorno = vRetorno

Call EnviaLogStatus("Download do arquivo finalizado !")

Fim:

On Error GoTo 0

10 Exit Function

BaixarArquivoDeRetorno\_Error:

11 If VBA.Err <> 0 Then

12 lngErrorNumber = VBA.Err.Number: strErrorMessagem = VBA.Err.Description

13 Call VBA.Err.Raise(VBA.Err.Number, cstr\_ProcedureName, VBA.vbNewLine & cstr\_ProcedureName & " Em" & VBA.IIf(VBA.Erl() <> 0, " > Linha (" & VBA.Erl() & ")", "") & " > " & VBA.Err.Description)

14 End If

GoTo Fim:

'Debug Mode

15 Resume

End Function

'Cria a tabela de downloads, caso nao exista

Public Sub **CriarTabelaDownloadLog**()

Dim tb As Object, pSQLCreate As String

10 On Error Resume Next

20 Set tb = CurrentDb.TableDefs("tblHistoricoDownload")

30 If tb Is Nothing Then

pSQLCreate = "CREATE TABLE tblHistoricoDownload ( "

pSQLCreate = pSQLCreate & VBA.vbNewLine & " Pagina LONG"

pSQLCreate = pSQLCreate & VBA.vbNewLine & " ,Linha LONG"

pSQLCreate = pSQLCreate & VBA.vbNewLine & " ,IDCampanha LONG"

pSQLCreate = pSQLCreate & VBA.vbNewLine & " ,DataSolicitacao DateTime"

pSQLCreate = pSQLCreate & VBA.vbNewLine & " ,TipoCampanha TEXT"

pSQLCreate = pSQLCreate & VBA.vbNewLine & " ,TipoRelatorio TEXT"

pSQLCreate = pSQLCreate & VBA.vbNewLine & " ,OrigemDownload TEXT"

pSQLCreate = pSQLCreate & VBA.vbNewLine & " ,NomeArquivo TEXT"

pSQLCreate = pSQLCreate & VBA.vbNewLine & " ,DestinoDownload TEXT"

pSQLCreate = pSQLCreate & VBA.vbNewLine & " ,QtdRegistros INT"

pSQLCreate = pSQLCreate & VBA.vbNewLine & " ,status TEXT"

pSQLCreate = pSQLCreate & VBA.vbNewLine & " ,DataDownload DateTime"

pSQLCreate = pSQLCreate & VBA.vbNewLine & ")"

Call CurrentDb.Execute(pSQLCreate)

220 End If

End Sub

'---------------------------------------------------------------------------------------

' PROCEDIMENTO : cBL\_Unear.DownloadLista()

' TIPO : Function

' DATA/HORA : 08/08/2017 13:09

' CONSULTOR : TECNUN - Adelson Rosendo Marques da Silva (adelson@tecnun.com.br)

' DESCRIÇÃO : Baixa o conteudo da lista atual para uma tabela local

'---------------------------------------------------------------------------------------

Public Function **DownloadLista**(tipoTabela As String, idListaHTML As String) As String

Dim sTabelaNome As String, strTableName As String, strCampos As String

Dim Tabela As Object

Dim i As Integer, y As Integer

Dim strSQLTable As String

Dim rs As Object

'Tratamento de Erro

'---------------------------------------------------------------------------------------

1 On Error GoTo DownloadLista\_Error

Dim lngErrorNumber As Long, strErrorMessagem As String:

Const cstr\_ProcedureName As String = "cBL\_Unear.DownloadLista()"

'---------------------------------------------------------------------------------------

2 Do While Not IEApp.ControlExists(idListaHTML): VBA.DoEvents: Loop

3 Set Tabela = IEApp.Control("bsDataTable")

4 strTableName = tipoTabela & "\_" & Tabela.ID & "\_" & VBA.Format(Now, "yyyymmdd\_hhnnss")

5 strSQLTable = "CREATE TABLE " & strTableName & "("

6 strCampos = ""

7 For i = 0 To Tabela.tHead.Rows.item(0).Cells.Length - 1

8 strCampos = strCampos & VBA.vbNewLine & " ,[" & Tabela.tHead.Rows.item(0).Cells(i).innerText & "] TEXT"

9 Next i

10 strSQLTable = strSQLTable & VBA.Mid(strCampos, 7) & VBA.vbNewLine & ")"

11 CurrentDb.Execute strSQLTable

Dim paginas As Collection, qtdPaginas As Integer

Dim pg As Object, iPg As Integer

Set paginas = IEApp.DOMSelect("page-number")

qtdPaginas = paginas.count

For iPg = 1 To qtdPaginas

'-############ SELEÇÃO DA PAGINA ----------------

'A pesquisa deve ser refeita para atualizar a instancia dos objetos

Set paginas = IEApp.DOMSelect("page-number")

Set pg = paginas(iPg).element

'Para a pagina ativa nao precisa clicar

If Not pg.ClassName Like "\*active" Then

Call pg.Click

'Aguarda 2 segundos para garantir o carregamento da pagina

win.Wait 2000

End If

'-############ EFETUA A LEITURA DO CONTEUDO DA TABELA NA PAGINA ATUAL ----------------

'-############ EFETUA A LEITURA DO CONTEUDO DA TABELA NA PAGINA ATUAL ----------------

Set Tabela = IEApp.Control("bsDataTable")

12 Set rs = CurrentDb.OpenRecordset(strTableName)

13 For i = 0 To Tabela.tBodies.item(0).Rows.Length - 1

14 rs.addNew

15 For y = 0 To Tabela.tBodies.item(0).Rows(i).Cells.Length - 1

16 rs(y).value = Tabela.tBodies.item(0).Rows(i).Cells(y).innerText

17 Next y

18 rs.Update

19 Next i

'-############ EFETUA A LEITURA DO CONTEUDO DA TABELA NA PAGINA ATUAL ----------------

Next iPg

20 DownloadLista = strTableName

Fim:

21 On Error GoTo 0

22 Exit Function

DownloadLista\_Error:

23 If Err <> 0 Then

24 lngErrorNumber = VBA.Err.Number: strErrorMessagem = VBA.Err.Description

25 Call VBA.Err.Raise(VBA.Err.Number, cstr\_ProcedureName, VBA.vbNewLine & cstr\_ProcedureName & " Em" & VBA.IIf(VBA.Erl() <> 0, " > Linha (" & VBA.Erl() & ")", "") & " > " & VBA.Err.Description)

26 End If

GoTo Fim:

27 Resume 'Debug Mode

End Function

**Modulo : cIE\_Tab**

Criar/Alterar as seguintes funções :

**DOMSelect()/ CRIAR**

**Selector()/ CRIAR**

Public Function DOMSelect(Selector As String) As VBA.Collection

Dim cElements As New Collection

Dim oParent As Object

Dim element As Object

If oParent Is Nothing Then Set oParent = Document.DocumentElement

If VBA.InStr(Selector, "#") > 0 Then

Set colElement = Document.getElementById(VBA.Replace(Selector, "#", ""))

Else

Set colElement = Document.getElementsByClassName(Selector)

End If

If Not colElement Is Nothing Then

If colElement.Length > 0 Then

For Each element In colElement

Call addNewControl(cElements, element)

Next element

End If

End If

Set DOMSelect = cElements

End Function

Public Function Selector(pSelector As String)

Dim cElements As New Collection

Dim oParent As Object

Dim element As Object

If oParent Is Nothing Then Set oParent = Document.DocumentElement

If VBA.InStr(Selector, "#") > 0 Then

Set colElement = Document.querySelector(Selector)

Else

Set colElement = Document.querySelectorAll(Selector)

End If

If Not colElement Is Nothing Then

If colElement.Length > 0 Then

For Each element In colElement

Call addNewControl(cElements, element)

Next element

End If

End If

Set Selector = cElements

End Function

**Modulo : cTFW\_WebUtil**

Criar/Alterar as seguintes funções :

**Download()/ ALTERA**

'---------------------------------------------------------------------------------------

' PROCEDIMENTO : cTFW\_WebUtil.Download()

' TIPO : Function

' DATA/HORA : 30/09/2016 13:44

' CONSULTOR : TECNUN - Adelson Rosendo Marques da Silva (adelson@tecnun.com.br)

' DESCRIÇÃO : Rotina otimizada para efetuar o download por meio de dois metodos :

' 1 - Por meio de API : Nesse modo é possivel verificar byte a byte

' que esta dendo baixado.

' 2 - Por meio de solicitação HTTP : Nesse modo é possivel tratar o

' retorno de varias formas

'---------------------------------------------------------------------------------------

'

' + Historico de Revisão

' \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

' Versão Data/Hora Autor Descriçao

'---------------------------------------------------------------------------------------

' \* 1.00 30/09/2016 13:44

' \* 2.00 09/09/2018 21:44 - Incluindo tratamento para caracteres invalidos em nome de arquivos

'---------------------------------------------------------------------------------------

Function Download(ByVal Url As String, \_

Optional metodo As eDownloadMethod = eDownloadMethod.API, \_

Optional destFilePath As String, \_

Optional ByVal bDebug As Boolean)

Dim sFileName As String

Dim oStream As Object

Dim NomeArquivo As String

Dim headers As Collection

Dim RetDownload As Variant

'---------------------------------------------------------------------------------------

1 On Error GoTo Download\_Error

Dim lngErrorNumber As Long, strErrorMessagem As String

2 Dim dtSartRunProc As Date: dtSartRunProc = VBA.Time

Const cstr\_ProcedureName As String = "Function cTFW\_WebUtil.Download()"

'---------------------------------------------------------------------------------------

'Trata caracteres que nao podem ter no nome do arquivo

destFilePath = VBA.Replace(destFilePath, "/", "\_")

destFilePath = VBA.Replace(destFilePath, "\*", "\_")

destFilePath = VBA.Replace(destFilePath, "?", "\_")

destFilePath = VBA.Replace(destFilePath, "<", "\_")

destFilePath = VBA.Replace(destFilePath, ">", "\_")

destFilePath = VBA.Replace(destFilePath, "|", "\_")

3 Select Case metodo

Case eDownloadMethod.API

4 Download = downloadViaApi(Url, bDebug, destFilePath)

5 Case eDownloadMethod.HTTP\_Request

'--------------------------------------------------------------------------------------------------------------------------------

'Executa a chamada do HTTP que realiza o download do arquivo solicitado

'--------------------------------------------------------------------------------------------------------------------------------

6 RetDownload = CreateHTTPSession(pMethod:="GET", pAction:=Url, pResponseType:=ResponseResultBinary, bAssync:=True, bDebugPrint:=bDebug)

'--------------------------------------------------------------------------------------------------------------------------------

'Trata o retorno do Download

'--------------------------------------------------------------------------------------------------------------------------------

If RetDownload(0) <> 0 Then

8 If destFilePath = "" Then

9 Set headers = RetDownload(2)

10 Call RaiseEventPrint("Tamanho : " & formatFileSize(getItemColletion(headers, "Content-Length")), bDebug)

11 Call RaiseEventPrint("Tipo de Arquivo : " & getItemColletion(headers, "Content-Type"), bDebug)

12 NomeArquivo = getItemColletion(headers, "Content-Disposition")

'Configura o nome do arquivo de destino

13 If NomeArquivo = "" Then

14 NomeArquivo = Right$(Url, Len(Url) - VBA.InStrRev(Url, "/"))

15 Else

16 NomeArquivo = VBA.Replace(VBA.CStr(VBA.Split(VBA.CStr(VBA.Split(VBA.LCase(NomeArquivo), ";")(1)), "filename=")(1)), VBA.Chr(34), "")

17 End If

18 If NomeArquivo <> "" Then

19 sFileName = Environ("Temp") & "\" & NomeArquivo

20 Else

21 sFileName = Environ("Temp") & "\Download" & VBA.Format(VBA.Now, "hhmmssmmddyy") & ".tmp"

22 End If

23 Else

24 sFileName = destFilePath

25 End If

'Descarrega o conteudo do Stream (bytes) que foi baixado e salva no arquivo

Call salvarStreamBytes(RetDownload(1), sFileName)

End If

Download = sFileName

35 End Select

Fim:

36 On Error GoTo 0

37 Exit Function

Download\_Error:

38 If VBA.Err <> 0 Then

39 lngErrorNumber = VBA.Err.Number: strErrorMessagem = VBA.Err.Description

Debug.Print cstr\_ProcedureName, "Linha : " & VBA.Erl() & " - " & strErrorMessagem

41 End If

GoTo Fim:

'Debug Mode

42 Resume

End Function