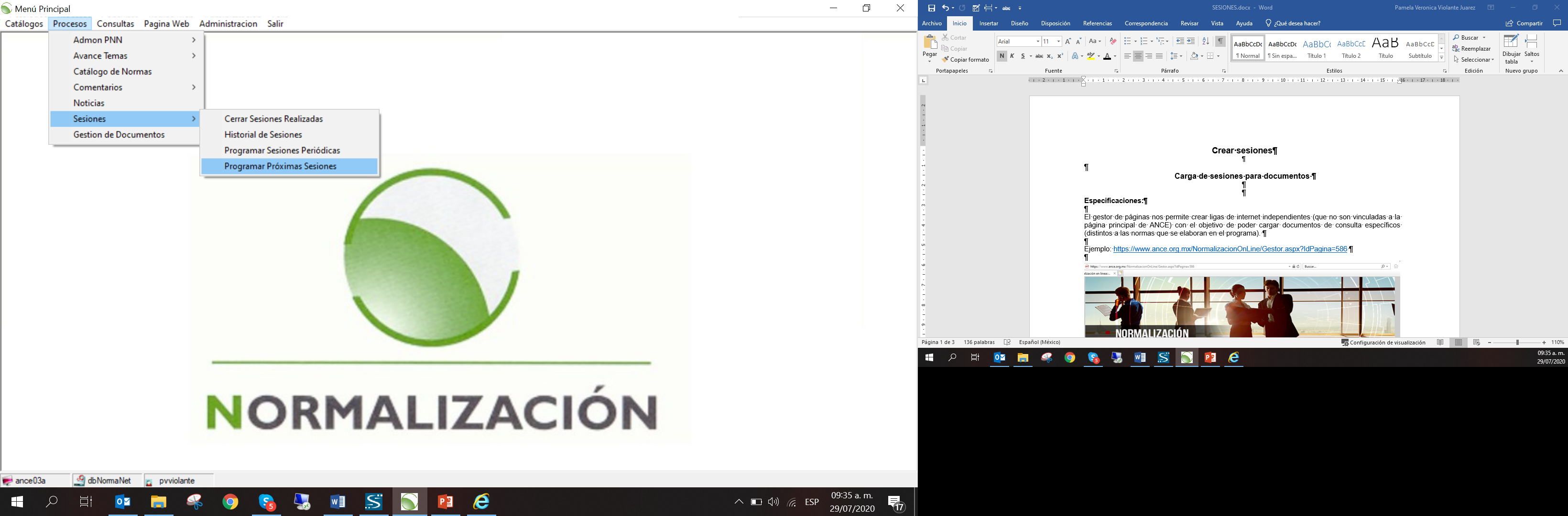
**SESIONES**

**Creación de sesiones y carga de documentos**

**Especificaciones:**

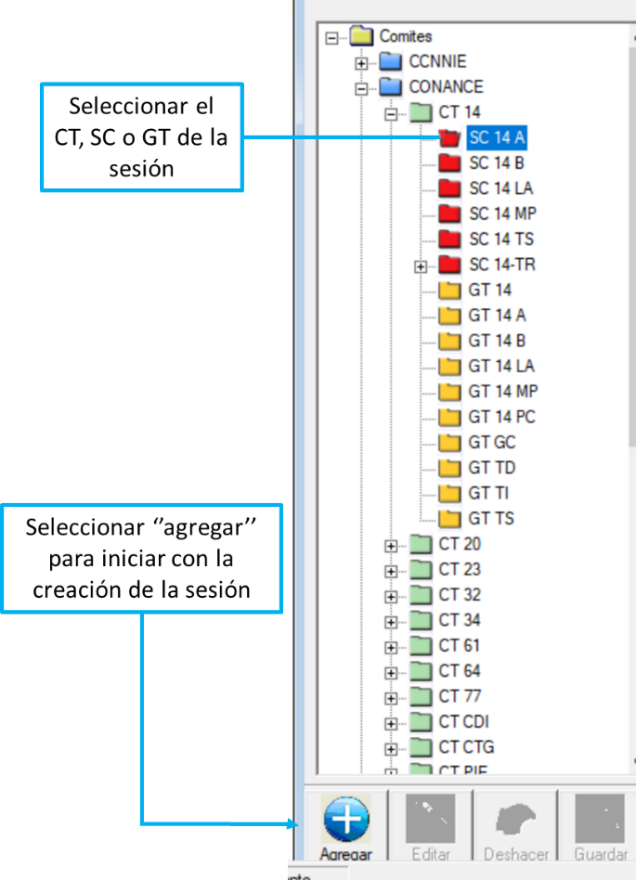
Una sesión en NORMANET corresponde a una sesión programada para CONANCE, un CT, SC o cualquiera de sus grupos de trabajo. Hoy en día lo único que permite es la descarga de los documentos que se verán durante la sesión, ya sea, de manera particular o en carpeta zip.

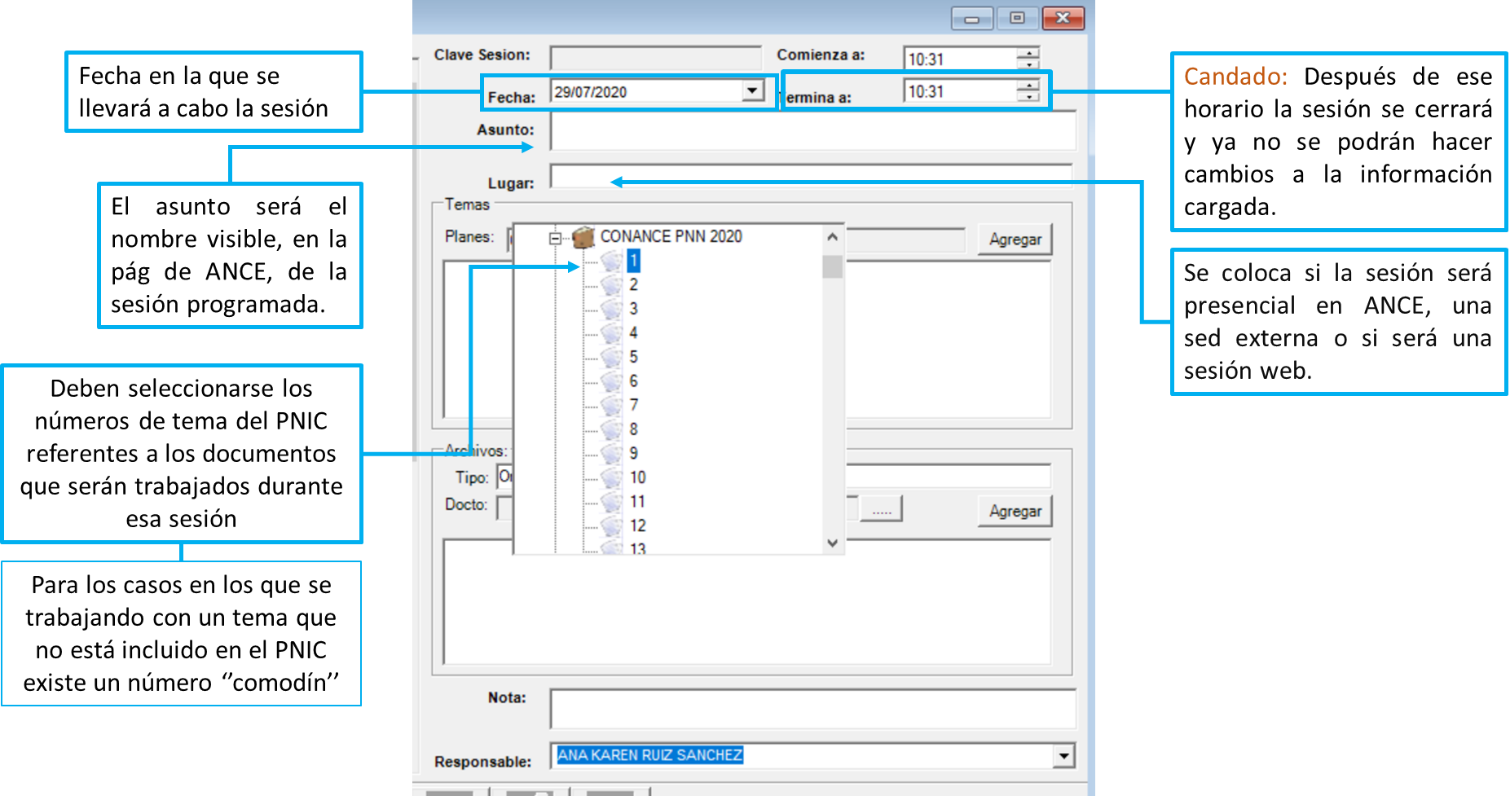
La carga se hace de la manera siguiente:



Seleccionar el Comité para el que se generará la sesión.

**MEJORA:** Actualmente hay varios Comités o grupos que no se ocupan, generalmente únicamente se crean sesiones para CONANCE, sus CT, SC y GT. Que puedan ocultarse los demás comités o que el responsable pueda seleccionar cuáles CTs quiere tener visibles, así no perderá tiempo en buscar el CT en donde se desea crear la sesión.



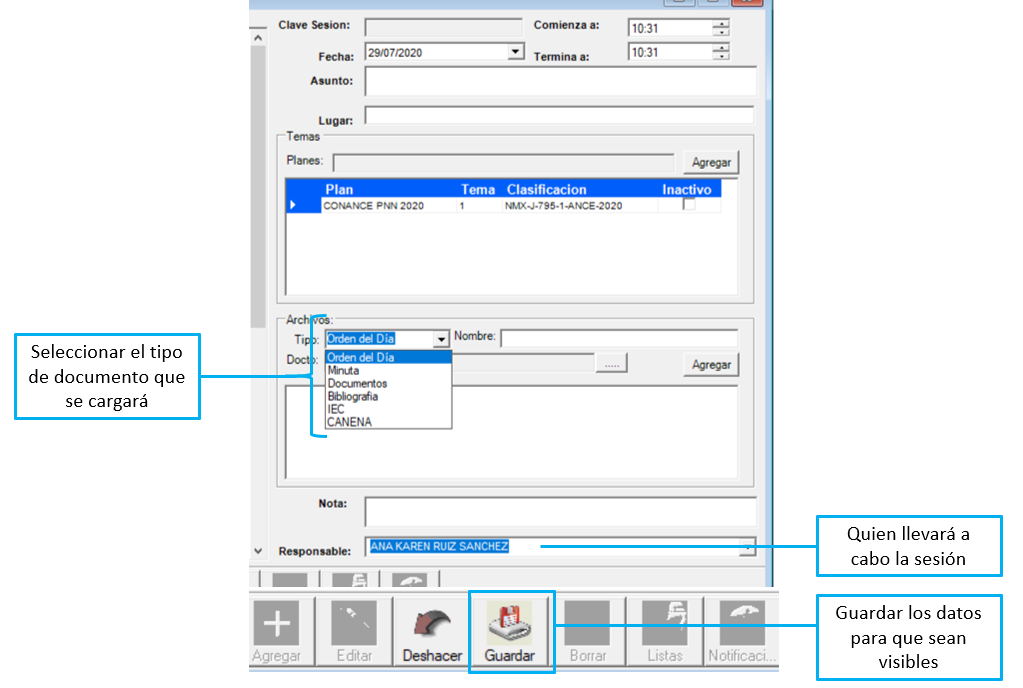


|  |  |
| --- | --- |
| Botones | Funcionalidad y/o negocio |
| Load del form | Arbol Comité  Ban = 9  ByVal var1 =’’,  ByVal Par1 =’’,  ByVal var2 =’’,  ByVal Par2 =’’,  ByVal var3 =’’,  ByVal Par3 =’’  "pawIndex"  cmd.Parameters.Add("@Ban", ban)  If Not Par1 = "" Then  cmd.Parameters.Add(var1, Par1)  End If  If Not Par2 = "" Then  cmd.Parameters.Add(var2, Par2)  End If  If Not Par3 = "" Then  cmd.Parameters.Add(var3, Par3)  End If  Crear Menu Comites  (dtComites, nodoCT = Nothing, nodoSC = Nothing, nodoGT = Nothing, idx = 0, ides = 1, indGT = False)  Recorrer los Comites  Obtener los comité técnico  **If nodoCT Is Nothing Then**  (2, "@idCM", nuevoNodo.Text.ToString, "", "", "", "")  Bandera = 2  "pawIndex"  cmd.Parameters.Add("@Ban", ban)  If Not Par1 = "" Then  cmd.Parameters.Add(var1, Par1)  End If  If Not Par2 = "" Then  cmd.Parameters.Add(var2, Par2)  End If  If Not Par3 = "" Then  cmd.Parameters.Add(var3, Par3)  End If  Obtener los Grupos de Trabajo  (4, "@idCM", nuevoNodo.Text.ToString, "@idCT", "NA", "@idSC", "NA")  "pawIndex"  cmd.CommandType = CommandType.StoredProcedure  cmd.Connection = cn  cmd.Parameters.Add("@Ban", ban)  If Not Par1 = "" Then  cmd.Parameters.Add(var1, Par1)  End If  If Not Par2 = "" Then  cmd.Parameters.Add(var2, Par2)  End If  If Not Par3 = "" Then  cmd.Parameters.Add(var3, Par3)  End If  Obtener las sesiones  Bandera = "s20"  Pertenece = nodoPadre.Text  "pawSesiones"  cmd.CommandType = CommandType.StoredProcedure  cmd.Connection = cn  cmd.Parameters.Add("@Id\_Sesion", \_Id\_Sesion)  cmd.Parameters.Add("@Id\_Comite", \_Id\_Comite)  cmd.Parameters.Add("@Id\_CT", \_Id\_CT)  cmd.Parameters.Add("@Id\_SC", \_Id\_SC)  cmd.Parameters.Add("@Id\_Grupo", \_Id\_Grupo)  cmd.Parameters.Add("@Fecha", \_Fecha)  cmd.Parameters.Add("@HoraI", \_HoraI)  cmd.Parameters.Add("@HoraT", \_HoraT)  cmd.Parameters.Add("@Asunto", \_Asunto)  cmd.Parameters.Add("@Lugar", \_Lugar)  cmd.Parameters.Add("@Responsable", \_Responsable)  cmd.Parameters.Add("@Notas", \_Notas)  cmd.Parameters.Add("@Status", \_Status)  cmd.Parameters.Add("@No\_Sol\_Sala", \_No\_Sol\_Sala)  cmd.Parameters.Add("@Pertenece", \_Pertenece)  cmd.Parameters.Add("@Bandera", \_Bandera)  **ElseIf indGT = True Then**  Obtener las sesiones  Bandera = "s20  Pertenece = nodoPadre.Text  "pawSesiones"  cmd.CommandType = CommandType.StoredProcedure  cmd.Connection = cn  cmd.Parameters.Add("@Id\_Sesion", \_Id\_Sesion)  cmd.Parameters.Add("@Id\_Comite", \_Id\_Comite)  cmd.Parameters.Add("@Id\_CT", \_Id\_CT)  cmd.Parameters.Add("@Id\_SC", \_Id\_SC)  cmd.Parameters.Add("@Id\_Grupo", \_Id\_Grupo)  cmd.Parameters.Add("@Fecha", \_Fecha)  cmd.Parameters.Add("@HoraI", \_HoraI)  cmd.Parameters.Add("@HoraT", \_HoraT)  cmd.Parameters.Add("@Asunto", \_Asunto)  cmd.Parameters.Add("@Lugar", \_Lugar)  cmd.Parameters.Add("@Responsable", \_Responsable)  cmd.Parameters.Add("@Notas", \_Notas)  cmd.Parameters.Add("@Status", \_Status)  cmd.Parameters.Add("@No\_Sol\_Sala", \_No\_Sol\_Sala)  cmd.Parameters.Add("@Pertenece", \_Pertenece)  cmd.Parameters.Add("@Bandera", \_Bandera)  **If nodoSC Is Nothing Then**  Obtener Sub Comite  (ban= 3, var1 ="@idCM", Part1 = nodoCT.Text, var2 = "@idCT", par2 = nuevoNodo.Text, var3= "", par3 = "")  "pawIndex"  cmd.CommandType = CommandType.StoredProcedure  cmd.Connection = cn  cmd.Parameters.Add("@Ban", ban)  If Not Par1 = "" Then  cmd.Parameters.Add(var1, Par1)  End If  If Not Par2 = "" Then  cmd.Parameters.Add(var2, Par2)  End If  If Not Par3 = "" Then  cmd.Parameters.Add(var3, Par3)  End If  Obtener grupos de trabajo  (ban= 4, var1 ="@idCM", Part1 = nodoCT.Text, var2 = "@idCT", par2 = nuevoNodo.Text, var3= "@idSC ", par3 = "NA")  "pawIndex"  cmd.CommandType = CommandType.StoredProcedure  cmd.Connection = cn  cmd.Parameters.Add("@Ban", ban)  If Not Par1 = "" Then  cmd.Parameters.Add(var1, Par1)  End If  If Not Par2 = "" Then  cmd.Parameters.Add(var2, Par2)  End If  If Not Par3 = "" Then  cmd.Parameters.Add(var3, Par3)  End If  **If nodoGT Is Nothing Then**  Obtener los comites  (ban= 4, var1 ="@idCM", Part1 = nodoCT.Text, var2 = "@idCT", par2 = nodoSC.Text.ToString, var3= "@idSC ", par3 = nuevoNodo.Text.ToString)  Agregar las sesiones  Bandera = "s20"  Pertenece = nodoPadre.Text  "pawSesiones"  cmd.CommandType = CommandType.StoredProcedure  cmd.Connection = cn  cmd.Parameters.Add("@Id\_Sesion", \_Id\_Sesion)  cmd.Parameters.Add("@Id\_Comite", \_Id\_Comite)  cmd.Parameters.Add("@Id\_CT", \_Id\_CT)  cmd.Parameters.Add("@Id\_SC", \_Id\_SC)  cmd.Parameters.Add("@Id\_Grupo", \_Id\_Grupo)  cmd.Parameters.Add("@Fecha", \_Fecha)  cmd.Parameters.Add("@HoraI", \_HoraI)  cmd.Parameters.Add("@HoraT", \_HoraT)  cmd.Parameters.Add("@Asunto", \_Asunto)  cmd.Parameters.Add("@Lugar", \_Lugar)  cmd.Parameters.Add("@Responsable", \_Responsable)  cmd.Parameters.Add("@Notas", \_Notas)  cmd.Parameters.Add("@Status", \_Status)  cmd.Parameters.Add("@No\_Sol\_Sala", \_No\_Sol\_Sala)  cmd.Parameters.Add("@Pertenece", \_Pertenece)  cmd.Parameters.Add("@Bandera", \_Bandera)  **Else**  Si no hay CT, GT y SC, solo agregar las sesiones  Agregar las sesiones  Bandera = "s20"  Pertenece = nodoPadre.Text  "pawSesiones"  cmd.CommandType = CommandType.StoredProcedure  cmd.Connection = cn  cmd.Parameters.Add("@Id\_Sesion", \_Id\_Sesion)  cmd.Parameters.Add("@Id\_Comite", \_Id\_Comite)  cmd.Parameters.Add("@Id\_CT", \_Id\_CT)  cmd.Parameters.Add("@Id\_SC", \_Id\_SC)  cmd.Parameters.Add("@Id\_Grupo", \_Id\_Grupo)  cmd.Parameters.Add("@Fecha", \_Fecha)  cmd.Parameters.Add("@HoraI", \_HoraI)  cmd.Parameters.Add("@HoraT", \_HoraT)  cmd.Parameters.Add("@Asunto", \_Asunto)  cmd.Parameters.Add("@Lugar", \_Lugar)  cmd.Parameters.Add("@Responsable", \_Responsable)  cmd.Parameters.Add("@Notas", \_Notas)  cmd.Parameters.Add("@Status", \_Status)  cmd.Parameters.Add("@No\_Sol\_Sala", \_No\_Sol\_Sala)  cmd.Parameters.Add("@Pertenece", \_Pertenece)  cmd.Parameters.Add("@Bandera", \_Bandera) |
| Agregar | Limpiar(txtClaveSesion, DTPkrhoraini, DTPkrhoratem, dtpFecha, TxtAsunto, TxtLugar, txtPlanes, txtDocumento, grdArchivos, TxtNotas)  Escritura(TxtAsunto, TxtLugar, TxtNotas)  Activar(DTPkrhoratem, DTPkrhoraini, dtpFecha, txtNombreDoc)  Activar(grdArchivos, grdTemas, cboTipo, cboResponsable, txtPlanes, cmdBuscar, cmdAgregarArchivos, cmdAgregarPlan)  Activar(tlbBotones.Buttons.Item(2), tlbBotones.Buttons.Item(3))  inactivar(tlbBotones.Buttons.Item(0), tlbBotones.Buttons.Item(1), tlbBotones.Buttons.Item(4), trvComites, tlbBotones.Buttons.Item(5), tlbBotones.Buttons.Item(6))  grdArchivos.DataSource = Nothing  grdTemas.DataSource = Nothing  grdArchivos.ContextMenu = cmDocumentos  grdTemas.ContextMenu = cmTemas |
| Editar | grdArchivos.ContextMenu = cmDocumentos  grdTemas.ContextMenu = cmTemas  Escritura(TxtAsunto, TxtLugar, TxtNotas)  Activar(DTPkrhoratem, DTPkrhoraini, dtpFecha, txtNombreDoc)  Activar(grdArchivos, grdTemas, cboTipo, cboResponsable, txtPlanes, cmdBuscar, cmdAgregarArchivos, cmdAgregarPlan)  Activar(tlbBotones.Buttons.Item(2), tlbBotones.Buttons.Item(3))  inactivar(tlbBotones.Buttons.Item(0), tlbBotones.Buttons.Item(1), tlbBotones.Buttons.Item(4), trvComites, tlbBotones.Buttons.Item(5), tlbBotones.Buttons.Item(6))  grdArchivos.ContextMenu = cmDocumentos grdTemas.ContextMenu = cmTemas |
| Deshacer | Ejecutar lo del Load |
| Guardar | If (DTPkrhoraini.Value > DTPkrhoratem.Value) Or (DTPkrhoraini.Value = DTPkrhoratem.Value) Then  MsgBox("la hora de inicio no puede ser mayor o igual a la hora de fin")  band = True  End If  If band = False And dt Is Nothing Then  MsgBox("No se ha seleccionado ningun tema")  band = True  End If  If band = False And Not dt Is Nothing Then  If dt.Rows.Count <= 0 Then  MsgBox("No se ha seleccionado ningun tema")  band = True  End If  End If  **SI ES NUEVO**  Obtener comité  Bandera = "s22"  If porSesion = True Then  objSesion.Id\_Comite = fullPhat(fullPhat.Length - 2)  Else  objSesion.Id\_Comite = trvComites.SelectedNode.Text  End If  "pawSesiones"  cmd.Parameters.Add("@Id\_Sesion", \_Id\_Sesion)  cmd.Parameters.Add("@Id\_Comite", \_Id\_Comite)  cmd.Parameters.Add("@Id\_CT", \_Id\_CT)  cmd.Parameters.Add("@Id\_SC", \_Id\_SC)  cmd.Parameters.Add("@Id\_Grupo", \_Id\_Grupo)  cmd.Parameters.Add("@Fecha", \_Fecha)  cmd.Parameters.Add("@HoraI", \_HoraI)  cmd.Parameters.Add("@HoraT", \_HoraT)  cmd.Parameters.Add("@Asunto", \_Asunto)  cmd.Parameters.Add("@Lugar", \_Lugar)  cmd.Parameters.Add("@Responsable", \_Responsable)  cmd.Parameters.Add("@Notas", \_Notas)  cmd.Parameters.Add("@Status", \_Status)  cmd.Parameters.Add("@No\_Sol\_Sala", \_No\_Sol\_Sala)  cmd.Parameters.Add("@Pertenece", \_Pertenece)  cmd.Parameters.Add("@Bandera", \_Bandera)  Llenar los campos  Comite = IIf(IsDBNull(dt.Rows(0).Item("ID\_Comite")), "", dt.Rows(0).Item("ID\_Comite"))  Ct = IIf(IsDBNull(dt.Rows(0).Item("ID\_CT")), "", dt.Rows(0).Item("ID\_CT"))  Sc = IIf(IsDBNull(dt.Rows(0).Item("Id\_SC")), "", dt.Rows(0).Item("Id\_SC"))  Gt = IIf(IsDBNull(dt.Rows(0).Item("ID\_Grupo")), "", dt.Rows(0).Item("ID\_Grupo"))  Bandera = "i1"  .Id\_Comite = Comite  .Id\_CT = Ct  .Id\_SC = Sc  .Id\_Grupo = Gt  .Fecha = Format(dtpFecha.Value, "dd/MM/yyyy")  .HoraI = Format(DTPkrhoraini.Value, "HH:mm:ss")  .HoraT = Format(DTPkrhoratem.Value, "HH:mm:ss")  .Asunto = TxtAsunto.Text  .Lugar = TxtLugar.Text  .Responsable = cboResponsable.SelectedValue  .Notas = TxtNotas.Text  .Status = 2  .No\_Sol\_Sala = Nothing  Ejecutar  "pawSesiones"  cmd.Parameters.Add("@Id\_Sesion", \_Id\_Sesion).Direction = ParameterDirection.InputOutput  cmd.Parameters.Add("@Id\_Comite", \_Id\_Comite)  cmd.Parameters.Add("@Id\_CT", \_Id\_CT)  cmd.Parameters.Add("@Id\_SC", \_Id\_SC)  cmd.Parameters.Add("@Id\_Grupo", \_Id\_Grupo)  cmd.Parameters.Add("@Fecha", \_Fecha)  cmd.Parameters.Add("@HoraI", \_HoraI)  cmd.Parameters.Add("@HoraT", \_HoraT)  cmd.Parameters.Add("@Asunto", \_Asunto)  cmd.Parameters.Add("@Lugar", \_Lugar)  cmd.Parameters.Add("@Responsable", \_Responsable)  cmd.Parameters.Add("@Notas", \_Notas)  cmd.Parameters.Add("@Status", \_Status)  cmd.Parameters.Add("@No\_Sol\_Sala", \_No\_Sol\_Sala)  cmd.Parameters.Add("@Pertenece", \_Pertenece)  cmd.Parameters.Add("@Bandera", \_Bandera)  Si se inserto correctamente  REM inserto temas  REM validar si viene de la base de datos o no (en caso de ser de la base se actualiza  REM de lo contrario se guarda  Id\_Plan = rows("Id\_Plan")  Id\_Tema = rows("Id\_Tema")  Inactivo = rows("Inactivo")  If rows("DeBase") = False Then  Bandera = "s4"  Id\_Plan = Id\_Plan  Id\_Tema = Id\_Tema  "uspSesionTema"  cmd.CommandType = CommandType.StoredProcedure  cmd.Connection = cn  cmd.Parameters.Add("@Id\_Sesion", \_Id\_Sesion)  cmd.Parameters.Add("@Id\_Plan", \_Id\_Plan)  cmd.Parameters.Add("@Id\_Tema", \_Id\_Tema)  cmd.Parameters.Add("@Inactivo", \_Inactivo)  cmd.Parameters.Add("@ref\_año", \_ref\_año)  cmd.Parameters.Add("@ref\_comite", \_ref\_comite)  cmd.Parameters.Add("@ref\_consecutivo", \_ref\_consecutivo)  cmd.Parameters.Add("@ref\_regreso", \_ref\_regreso)  cmd.Parameters.Add("@ref\_traspaso", \_ref\_traspaso)  cmd.Parameters.Add("@Bandera", \_Bandera)  Bandera = "i1"  Id\_Sesion = Id\_Sesion  .Id\_Plan = Id\_Plan  .Id\_Tema = Id\_Tema  .Inactivo = Inactivo  .ref\_año = IIf(IsDBNull(dt.Rows(0).Item("ref\_año")), "", dt.Rows(0).Item("ref\_año"))  .ref\_comite = IIf(IsDBNull(dt.Rows(0).Item("ref\_comite")), "", dt.Rows(0).Item("ref\_comite"))  .ref\_consecutivo = IIf(IsDBNull(dt.Rows(0).Item("ref\_consecutivo")), "", dt.Rows(0).Item("ref\_consecutivo"))  .ref\_regreso = IIf(IsDBNull(dt.Rows(0).Item("ref\_regreso")), "", dt.Rows(0).Item("ref\_regreso"))  .ref\_traspaso = IIf(IsDBNull(dt.Rows(0).Item("ref\_traspaso")), "", dt.Rows(0).Item("ref\_traspaso"))  "uspSesionTema"  cmd.Parameters.Add("@Id\_Sesion", \_Id\_Sesion)  cmd.Parameters.Add("@Id\_Plan", \_Id\_Plan)  cmd.Parameters.Add("@Id\_Tema", \_Id\_Tema)  cmd.Parameters.Add("@Inactivo", \_Inactivo)  cmd.Parameters.Add("@ref\_año", \_ref\_año)  cmd.Parameters.Add("@ref\_comite", \_ref\_comite)  cmd.Parameters.Add("@ref\_consecutivo", \_ref\_consecutivo)  cmd.Parameters.Add("@ref\_regreso", \_ref\_regreso)  cmd.Parameters.Add("@ref\_traspaso", \_ref\_traspaso)  cmd.Parameters.Add("@Bandera", \_Bandera)  SI viene de la base, se actualiza  Referencia = rows("Referencia")  Bandera = "u2"  .Id\_Sesion = Id\_Sesion  .Inactivo = Inactivo  .Referencia = Referencia  "uspSesionTema"  cmd.Parameters.Add("@Id\_Sesion", \_Id\_Sesion)  cmd.Parameters.Add("@Id\_Plan", \_Id\_Plan)  cmd.Parameters.Add("@Id\_Tema", \_Id\_Tema)  cmd.Parameters.Add("@Inactivo", \_Inactivo)  cmd.Parameters.Add("@ref\_año", \_ref\_año)  cmd.Parameters.Add("@ref\_comite", \_ref\_comite)  cmd.Parameters.Add("@ref\_consecutivo", \_ref\_consecutivo)  cmd.Parameters.Add("@ref\_regreso", \_ref\_regreso)  cmd.Parameters.Add("@ref\_traspaso", \_ref\_traspaso)  cmd.Parameters.Add("@Bandera", \_Bandera)  cmd.Parameters.Add("@Referencia", \_Referencia)  REM Iserto Documentos  sRutaDocto = rows("Docto")  id\_tipo = rows("id\_tipo\_doc")  NombreDocto = rows("NombreDelDocumento")  Inactivo = rows("Inactivo")  SI NO vienen de la base, se inserta  Bandera = "i1"  .Id\_Sesion = Id\_Sesion  .Docto = NombreArchivo  .Id\_Tipo = id\_tipo  .Inactivo = Inactivo  .NombreDelDocumento = NombreDocto  "uspSesionDoctos"  cmd.Parameters.Add("@Id\_Sesion", \_Id\_Sesion)  cmd.Parameters.Add("@Id\_Docto", \_Id\_Docto)  cmd.Parameters.Add("@Docto", \_Docto)  cmd.Parameters.Add("@Id\_Tipo", \_Id\_Tipo)  cmd.Parameters.Add("@Inactivo", \_Inactivo)  cmd.Parameters.Add("@NombreDelDocumento", \_NombreDelDocumento)  cmd.Parameters.Add("@Bandera", \_Bandera)  SI viene de la base, se actualiza  Id\_docto = rows("Id\_docto")  Bandera = "u2"  objSesionesDoctos.Id\_Docto = Id\_docto  objSesionesDoctos.Inactivo = inactivo  "uspSesionDoctos"  cmd.CommandType = CommandType.StoredProcedure  cmd.Connection = cn  cmd.Parameters.Add("@Id\_Sesion", \_Id\_Sesion)  cmd.Parameters.Add("@Id\_Docto", \_Id\_Docto)  cmd.Parameters.Add("@Docto", \_Docto)  cmd.Parameters.Add("@Id\_Tipo", \_Id\_Tipo)  cmd.Parameters.Add("@Inactivo", \_Inactivo)  cmd.Parameters.Add("@NombreDelDocumento", \_NombreDelDocumento)  cmd.Parameters.Add("@Bandera", \_Bandera)  **SI ES EDITAR**  Obtener comités  Bandera = "s22"  If porSesion = True Then  objSesion.Id\_Comite = fullPhat(fullPhat.Length - 2)  Else  objSesion.Id\_Comite = trvComites.SelectedNode.Text  End If  "pawSesiones"    cmd.Parameters.Add("@Id\_Sesion", \_Id\_Sesion)  cmd.Parameters.Add("@Id\_Comite", \_Id\_Comite)  cmd.Parameters.Add("@Id\_CT", \_Id\_CT)  cmd.Parameters.Add("@Id\_SC", \_Id\_SC)  cmd.Parameters.Add("@Id\_Grupo", \_Id\_Grupo)  cmd.Parameters.Add("@Fecha", \_Fecha)  cmd.Parameters.Add("@HoraI", \_HoraI)  cmd.Parameters.Add("@HoraT", \_HoraT)  cmd.Parameters.Add("@Asunto", \_Asunto)  cmd.Parameters.Add("@Lugar", \_Lugar)  cmd.Parameters.Add("@Responsable", \_Responsable)  cmd.Parameters.Add("@Notas", \_Notas)  cmd.Parameters.Add("@Status", \_Status)  cmd.Parameters.Add("@No\_Sol\_Sala", \_No\_Sol\_Sala)  cmd.Parameters.Add("@Pertenece", \_Pertenece)  cmd.Parameters.Add("@Bandera", \_Bandera)  .Bandera = "u1"  Id\_Comite = Comite  .Id\_CT = Ct  .Id\_SC = Sc  .Id\_Grupo = Gt  .Fecha = Format(dtpFecha.Value, "dd/MM/yyyy")  .HoraI = Format(DTPkrhoraini.Value, "HH:mm:ss")  .HoraT = Format(DTPkrhoratem.Value, "HH:mm:ss")  .Asunto = TxtAsunto.Text  .Lugar = TxtLugar.Text  .Responsable = cboResponsable.SelectedValue  .Notas = TxtNotas.Text  .Status = 2  .No\_Sol\_Sala = Nothing  .Id\_Sesion = trvComites.SelectedNode.Tag  "pawSesiones"  cmd.Parameters.Add("@Id\_Sesion", \_Id\_Sesion)  cmd.Parameters.Add("@Id\_Comite", \_Id\_Comite)  cmd.Parameters.Add("@Id\_CT", \_Id\_CT)  cmd.Parameters.Add("@Id\_SC", \_Id\_SC)  cmd.Parameters.Add("@Id\_Grupo", \_Id\_Grupo)  cmd.Parameters.Add("@Fecha", \_Fecha)  cmd.Parameters.Add("@HoraI", \_HoraI)  cmd.Parameters.Add("@HoraT", \_HoraT)  cmd.Parameters.Add("@Asunto", \_Asunto)  cmd.Parameters.Add("@Lugar", \_Lugar)  cmd.Parameters.Add("@Responsable", \_Responsable)  cmd.Parameters.Add("@Notas", \_Notas)  cmd.Parameters.Add("@Status", \_Status)  cmd.Parameters.Add("@No\_Sol\_Sala", \_No\_Sol\_Sala)  cmd.Parameters.Add("@Pertenece", \_Pertenece)  cmd.Parameters.Add("@Bandera", \_Bandera)  REM inserto temas  Bandera = "s4"  .Id\_Plan = Id\_Plan  .Id\_Tema = Id\_Tema  "uspSesionTema"  cmd.Parameters.Add("@Id\_Sesion", \_Id\_Sesion)  cmd.Parameters.Add("@Id\_Plan", \_Id\_Plan)  cmd.Parameters.Add("@Id\_Tema", \_Id\_Tema)  cmd.Parameters.Add("@Inactivo", \_Inactivo)  cmd.Parameters.Add("@ref\_año", \_ref\_año)  cmd.Parameters.Add("@ref\_comite", \_ref\_comite)  cmd.Parameters.Add("@ref\_consecutivo", \_ref\_consecutivo)  cmd.Parameters.Add("@ref\_regreso", \_ref\_regreso)  cmd.Parameters.Add("@ref\_traspaso", \_ref\_traspaso)  cmd.Parameters.Add("@Bandera", \_Bandera)  Bandera = "i1"  .Id\_Sesion = Id\_Sesion  .Id\_Plan = Id\_Plan  .Id\_Tema = Id\_Tema  .Inactivo = Inactivo  .ref\_año = IIf(IsDBNull(dt.Rows(0).Item("ref\_año")), "", dt.Rows(0).Item("ref\_año"))  .ref\_comite = IIf(IsDBNull(dt.Rows(0).Item("ref\_comite")), "", dt.Rows(0).Item("ref\_comite"))  .ref\_consecutivo = IIf(IsDBNull(dt.Rows(0).Item("ref\_consecutivo")), "", dt.Rows(0).Item("ref\_consecutivo"))  .ref\_regreso = IIf(IsDBNull(dt.Rows(0).Item("ref\_regreso")), "", dt.Rows(0).Item("ref\_regreso"))  .ref\_traspaso = IIf(IsDBNull(dt.Rows(0).Item("ref\_traspaso")), "", dt.Rows(0).Item("ref\_traspaso"))  "uspSesionTema"  cmd.Parameters.Add("@Id\_Sesion", \_Id\_Sesion)  cmd.Parameters.Add("@Id\_Plan", \_Id\_Plan)  cmd.Parameters.Add("@Id\_Tema", \_Id\_Tema)  cmd.Parameters.Add("@Inactivo", \_Inactivo)  cmd.Parameters.Add("@ref\_año", \_ref\_año)  cmd.Parameters.Add("@ref\_comite", \_ref\_comite)  cmd.Parameters.Add("@ref\_consecutivo", \_ref\_consecutivo)  cmd.Parameters.Add("@ref\_regreso", \_ref\_regreso)  cmd.Parameters.Add("@ref\_traspaso", \_ref\_traspaso)  cmd.Parameters.Add("@Bandera", \_Bandera)  REM actualizo  Referencia = rows("Referencia")  Bandera = "u2"  .Id\_Sesion = Id\_Sesion  .Inactivo = Inactivo  .Referencia = Referencia  "uspSesionTema"  cmd.Parameters.Add("@Id\_Sesion", \_Id\_Sesion)  cmd.Parameters.Add("@Id\_Plan", \_Id\_Plan)  cmd.Parameters.Add("@Id\_Tema", \_Id\_Tema)  cmd.Parameters.Add("@Inactivo", \_Inactivo)  cmd.Parameters.Add("@ref\_año", \_ref\_año)  cmd.Parameters.Add("@ref\_comite", \_ref\_comite)  cmd.Parameters.Add("@ref\_consecutivo", \_ref\_consecutivo)  cmd.Parameters.Add("@ref\_regreso", \_ref\_regreso)  cmd.Parameters.Add("@ref\_traspaso", \_ref\_traspaso)  cmd.Parameters.Add("@Bandera", \_Bandera)  cmd.Parameters.Add("@Referencia", \_Referencia)  REM Iserto Documentos  REM validar si viene de la base de datos o no (en caso de ser de la base se actualiza  REM de lo contrario se guarda  sRutaDocto = rows("Docto")  id\_tipo = rows("id\_tipo\_doc")  NombreDocto = rows("NombreDelDocumento")  Inactivo = rows("Inactivo")  Guardar documentos  sRutaDoctos = "\\ance-nas\doctosNormanet$\"  Ruta = Comite & "\" \_  & IIf(Ct = "NA", "", Ct & "\") \_  & IIf(Sc = "NA", "", Sc & "\") \_  & IIf(Gt = "NA", "", Gt & "\")  Validar si existe la carpeta  Directory.Exists(sRutaDoctos & Ruta) Then Directory.CreateDirectory(sRutaDoctos & Ruta)  NombreArchivo = "DocSes\_" & Format(Now, "dMyy\_hhmmss") & Now.Millisecond.ToString  ArchivoArr = Split(sRutaDocto, ".")  NombreArchivo &= "." & ArchivoArr(ArchivoArr.Length - 1)  Copiar los archivos a la ruta  Bandera = "i1"  .Id\_Sesion = Id\_Sesion  .Docto = NombreArchivo  .Id\_Tipo = id\_tipo  .Inactivo = Inactivo  .NombreDelDocumento = NombreDocto  "uspSesionDoctos"  cmd.Parameters.Add("@Id\_Sesion", \_Id\_Sesion)  cmd.Parameters.Add("@Id\_Docto", \_Id\_Docto)  cmd.Parameters.Add("@Docto", \_Docto)  cmd.Parameters.Add("@Id\_Tipo", \_Id\_Tipo)  cmd.Parameters.Add("@Inactivo", \_Inactivo)  cmd.Parameters.Add("@NombreDelDocumento", \_NombreDelDocumento)  cmd.Parameters.Add("@Bandera", \_Bandera)  Si viene de la base  Id\_docto = rows("Id\_docto")  Bandera = "u2"  .Id\_Docto = Id\_docto  .Inactivo = inactivo  "uspSesionDoctos"    cmd.Parameters.Add("@Id\_Sesion", \_Id\_Sesion)  cmd.Parameters.Add("@Id\_Docto", \_Id\_Docto)  cmd.Parameters.Add("@Docto", \_Docto)  cmd.Parameters.Add("@Id\_Tipo", \_Id\_Tipo)  cmd.Parameters.Add("@Inactivo", \_Inactivo)  cmd.Parameters.Add("@NombreDelDocumento", \_NombreDelDocumento)  cmd.Parameters.Add("@Bandera", \_Bandera) |
| Borrar | ¿Estas seguro de eliminar la sesion seleccionada?  Bandera = "u2"  objSesiones.Status = 3  objSesiones.Id\_Sesion = trvComites.SelectedNode.Tag  "pawSesiones"  cmd.Parameters.Add("@Id\_Sesion", \_Id\_Sesion)  cmd.Parameters.Add("@Id\_Comite", \_Id\_Comite)  cmd.Parameters.Add("@Id\_CT", \_Id\_CT)  cmd.Parameters.Add("@Id\_SC", \_Id\_SC)  cmd.Parameters.Add("@Id\_Grupo", \_Id\_Grupo)  cmd.Parameters.Add("@Fecha", \_Fecha)  cmd.Parameters.Add("@HoraI", \_HoraI)  cmd.Parameters.Add("@HoraT", \_HoraT)  cmd.Parameters.Add("@Asunto", \_Asunto)  cmd.Parameters.Add("@Lugar", \_Lugar)  cmd.Parameters.Add("@Responsable", \_Responsable)  cmd.Parameters.Add("@Notas", \_Notas)  cmd.Parameters.Add("@Status", \_Status)  cmd.Parameters.Add("@No\_Sol\_Sala", \_No\_Sol\_Sala)  cmd.Parameters.Add("@Pertenece", \_Pertenece)  cmd.Parameters.Add("@Bandera", \_Bandera)  Mensaje:  "Registro eliminado correctamente" |
| Listas | \*\*\*\*  Se manda a la pantalla de FrmListaAsistencia.vb pero no abre el fromulario |
| Notificación | |  |  | | --- | --- | | Botones | Negocio y/o funcionalidad | | Load del form | sPagina = "http://www.ance.org.mx/editorhtml/index.aspx"  .Bandera = "s2"  .IdSesion = IdSesion  "uspSesionesNotificaciones"  cmd.Parameters.Add("@IdSesionNotificacion", \_IdSesionNotificacion)  cmd.Parameters.Add("@IdSesion", \_IdSesion)  cmd.Parameters.Add("@Titulo", \_Titulo)  cmd.Parameters.Add("@Cuerpo", \_Cuerpo)  cmd.Parameters.Add("@Bandera", \_Bandera)  llenar los campos  \_IdSesionNotificacion = IIf(IsDBNull(dt.Rows(0).Item("IdSesionNotificacion")) = True, Nothing, dt.Rows(0).Item("IdSesionNotificacion"))  \_IdSesion = IIf(IsDBNull(dt.Rows(0).Item("IdSesion")) = True, Nothing, dt.Rows(0).Item("IdSesion"))  \_Titulo = IIf(IsDBNull(dt.Rows(0).Item("Titulo")) = True, Nothing, dt.Rows(0).Item("Titulo"))  \_Cuerpo = IIf(IsDBNull(dt.Rows(0).Item("Cuerpo")) = True, Nothing, dt.Rows(0).Item("Cuerpo"))  SI .Encontrado = True  IdAccion = eAccion.Editar  txtTitulo.Text = objSesionesNotificaciones.Titulo  txtCuerpo.Text = objSesionesNotificaciones.Cuerpo  Activar(tlbBotones.Buttons(1))  Si NO se encontró  IdAccion = eAccion.Nuevo  inactivar(tlbBotones.Buttons(1)) | | Guardar | Si guardar  Bandera = "i1"  .Cuerpo = txtCuerpo.Text  .IdSesion = IdSesion  .Titulo = txtTitulo.Text  "uspSesionesNotificaciones"  cmd.Parameters.Add("@IdSesionNotificacion", \_IdSesionNotificacion)  cmd.Parameters.Add("@IdSesion", \_IdSesion)  cmd.Parameters.Add("@Titulo", \_Titulo)  cmd.Parameters.Add("@Cuerpo", \_Cuerpo)  cmd.Parameters.Add("@Bandera", \_Bandera)  mensaje: "Contenido Agregado correctamente"  Activar(tlbBotones.Buttons(1))  Si editar  Bandera = "u2"  .Cuerpo = txtCuerpo.Text  .IdSesion = IdSesion  .Titulo = txtTitulo.Text  "uspSesionesNotificaciones"  cmd.Parameters.Add("@IdSesionNotificacion", \_IdSesionNotificacion)  cmd.Parameters.Add("@IdSesion", \_IdSesion)  cmd.Parameters.Add("@Titulo", \_Titulo)  cmd.Parameters.Add("@Cuerpo", \_Cuerpo)  cmd.Parameters.Add("@Bandera", \_Bandera)  mensaje:  "Contenido Agregado correctamente" | | Copiar | Copiar lo del campo contenido | | Pegar | Pegar lo del campo contenido | | Enviar correo | paramIdSesion.Key = "IdSesion"  paramIdSesion.Value = IdSesion  paramSesion.Key = "Sesion"  paramSesion.Value = txtNoSesion.Text  .Bandera = "s4"  objCorreo.Subject = "NOTIFICACIÓN A SESIÓN " & txtNoSesion.Text  objCorreo.sTo = [efren.tello@maplesistemas.com.mx](mailto:efren.tello@maplesistemas.com.mx)  "upsCorreosPersonalizados"  cmd.Parameters.Add("@asunto", \_Subject)  cmd.Parameters.Add("@Para", \_sTo)  cmd.Parameters.Add("@CC", \_scc)  cmd.Parameters.Add("@Perfil", "MailNor")  cmd.Parameters.Add("@xml", \_sXML)  cmd.Parameters.Add("@Bandera", \_Bandera)  Mensaje  "Correo Enviado Correctamente" | | Componenete WEB | |  |  | | --- | --- | | botones | Negocio y/o funcionalidad | | Load del form | sPagina = <http://www.ance.org.mx/editorhtml/index.aspx>  Mostrar la paguina de ANCE | | regresar | Cierra el formulario | | |

Considerar un módulo alterno que permita agregar temas a las sesiones, que sea de acceso restrictivo a la CATL, agregando justificación para poder agregar el tema.

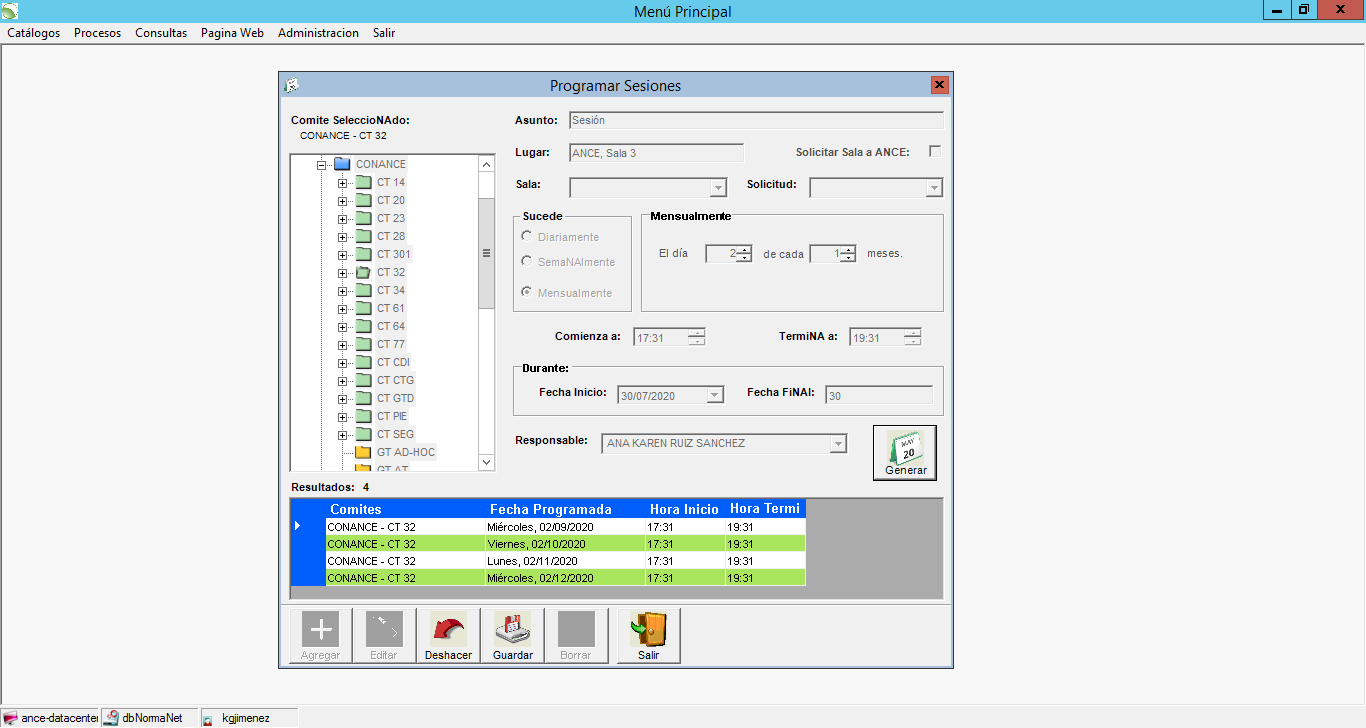
El candado para cerrar las sesiones se considera hasta la media noche de esa fecha.

**MEJORA:** El sistema debe arrojar por default el nombre del responsable (secretario del CT), con opción de seleccionar otro nombre (En el Directorio aparece quien es Secretario de cada CT).



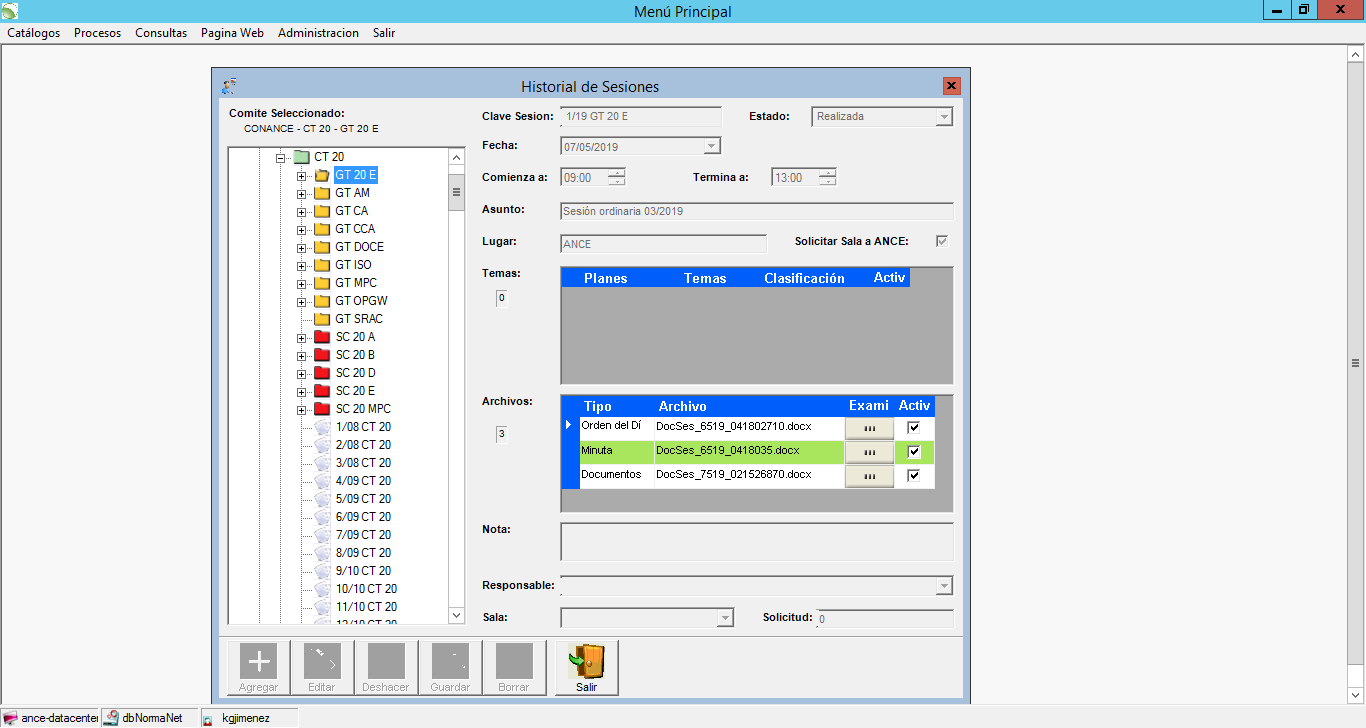
Se puede cargar archivo por archivo o un zip con todos los archivos de la sesión.

Debe aparecer un campo nuevo, en donde se pueda colocar el link, ID de reunión, contraseña e información de la sesión, misma que se visualizará en la plataforma para miembros (ver especificaciones para dicha plataforma). Que pueda imprimirse la lista de asistencia desde aquí, en el formato del SGC, FORNOR-P01.05 y que pueda enviarse la notificación a sesión desde aquí también hasta que se le dé click en el botón Notificación.



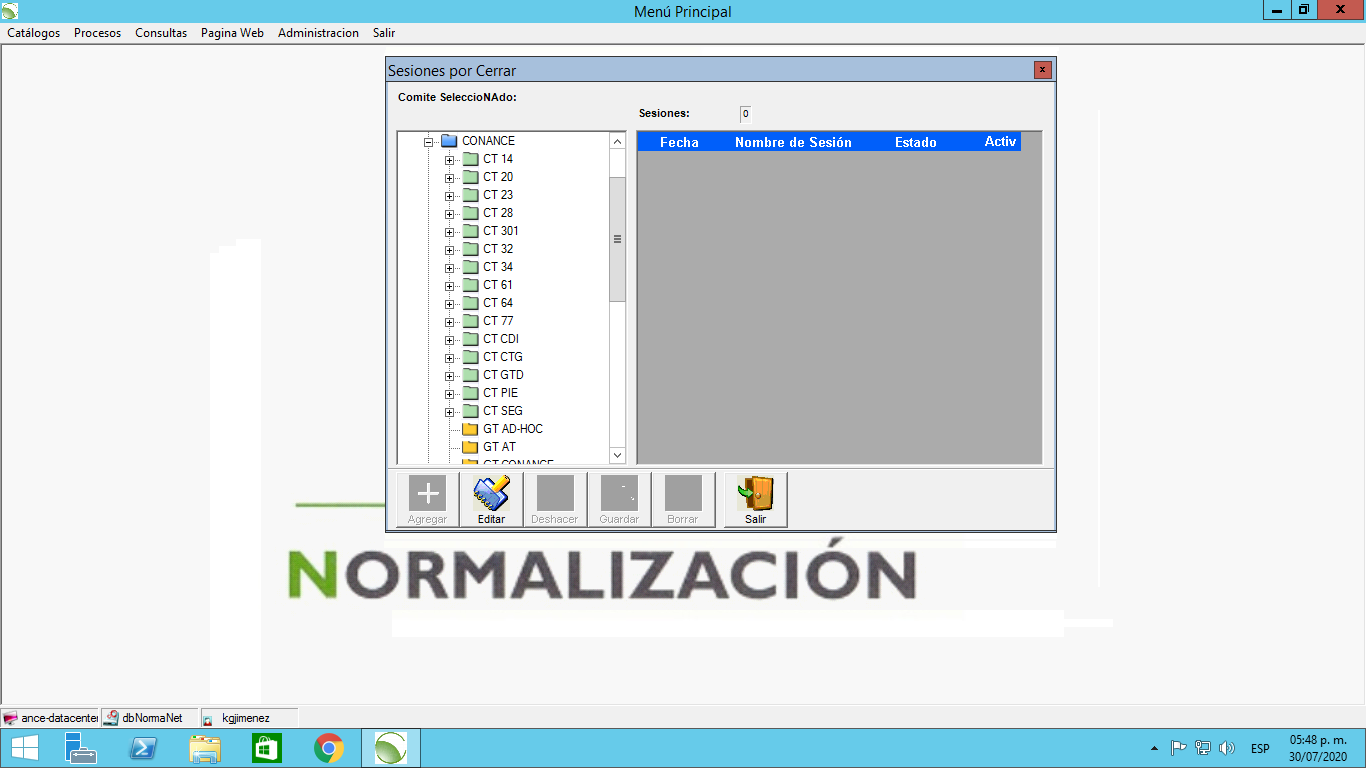
|  |  |
| --- | --- |
| Botones | Negocio y/o funcionalidad |
| Load del form | Inicializar las variables globales  min As DateTime = DateTime.Now  time As TimeSpan = New TimeSpan(min.TimeOfDay.Days, 7, 0, 0, 0)  min = DateTime.Today.Add(time)  DTPkrhoraini.MinDate = min  DTPkrhoraini.MaxDate = min.AddHours(16)  DTPkrhoratem.MinDate = DTPkrhoraini.MinDate  DTPkrhoratem.MaxDate = DTPkrhoraini.MaxDate  Llenar comiteTreeview  "SELECT ID\_Comite, Descripcion, Responsable FROM C\_Comite ORDER BY ID\_Comite ASC"  Para cada comité  'PARA LLENAR LOS COMITES TÉCNICOS  'se necesita que se envíe por que Comité se filtra  "SELECT ID\_Comite, ID\_CT, Descripcion, Objetivo, Responsable FROM c\_CT WHERE ID\_Comite = '" + sComite + "' ORDER BY ID\_CT ASC"  Por cada comité técnico, obtener los sub comité  'PARA LLENAR LOS SUB TÉCNICOS  'se necesita que se envíe por que Comité y CT se filtra  "SELECT ID\_Comite, ID\_CT, ID\_SC, Descripcion, Objetivo, Responsable FROM c\_SC WHERE ID\_Comite = '" + sComite + "' and ID\_CT='" + sCT + "' ORDER BY ID\_SC ASC"    Por cada sub comité  'PARA LLENAR LOS GRUPOS DE TRABAJO  'se necesita que se envíe por que Comité, CT y SC se filtra  "SELECT ID\_Comite, ID\_CT, ID\_SC, ID\_Grupo, Descripcion, Objetivo, Responsable FROM c\_Gpos\_Trabajos WHERE ID\_Comite = '" + sComite + "' and ID\_CT='" + sCT + "' and ID\_SC='" + sSC + "' ORDER BY ID\_Grupo ASC"  objempleados.Inicializa(Application.StartupPath & "\principal.ini", "comun", gUsuario, gPasswordSql)  Cargar combo responsable  .Bandera = 6  "sp\_C\_Emp\_Buscar"  cmd.Parameters.Add("@Bandera", \_Bandera)  cmd.Parameters.Add("@ID\_Usuario", \_Id\_usuario)  cmd.Parameters.Add("@ID\_Area", \_Id\_Area)  cmd.Parameters.Add("@Id\_Sistema", \_Id\_Sistema)  Cargar salas  Bandera = 1  "sp\_P\_Salas\_Buscar"  cmd.Parameters.Add("@Bandera", \_Bandera)  Cargar combo equipo  Bandera = 2  "sp\_P\_Salas\_Buscar"  cmd.Parameters.Add("@Bandera", \_Bandera)  RBnSucede1.Checked = True  INActivos(LblCada, Lblcada2, Lblcada3, Lblcada4, NUDFrec, NUDFrec2, CkBLun, CkBmartes, CkBMiercoles, CkBJueves, CkBViernes, CboResponsable, CboSalas, CboEquipo, TxtAsunto, TxtLugar, CkBxSalasANCE, tlbBotonera.Buttons(1), tlbBotonera.Buttons(3), tlbBotonera.Buttons(4), tlbBotonera.Buttons(5), tlbBotonera.Buttons(7), DTPkrhoraini, DTPkrhoratem, RBnSucede1, RBnSucede2, RBnSucede3, DTPKFecini, DTPKFecfin, cmdGenerar)  Activos(tlbBotonera.Buttons(1), tlbBotonera.Buttons(7)) |
| Agregar | Activos(LblCada, Lblcada2, Lblcada3, Lblcada4, NUDFrec, NUDFrec2, CkBLun, CkBmartes, CkBMiercoles, CkBJueves, CkBViernes, CboResponsable, TxtAsunto, TxtLugar, DTPkrhoraini, DTPkrhoratem, RBnSucede1, RBnSucede2, RBnSucede3, DTPKFecini, DTPKFecfin, tlbBotonera.Buttons(3), tlbBotonera.Buttons(4), tlbBotonera.Buttons(7))  Inactivos(tlbBotonera.Buttons(1), cmdGenerar, tvComites)  iEditar = "Agregar"  Si el árbol no tiene ningún comité  Mensaje: "¿Deseas crear un nuevo comite?"  Si es si. Se manda llamar el formulario de comité |
| Editar |  |
| Deshacer | Se manda llamar lo del Load |
| Guardar | Si es guardar  Validar los campos  valida("comites") And valida("dtpker") And valida("Horario") And valida("Asunto") And valida("Lugar") And valida("sucede")  Select Case array\_texto.Length ''''''''''''''''''''''''''''' LleNAdo de las propiedades De Comite  Case 1  .Id\_Comite = array\_texto(0)  .Id\_CT = "NA"  .Id\_SC = "NA"  .Id\_Grupo = "NA"  Case 2  .Id\_Comite = array\_texto(0)  .Id\_CT = array\_texto(1)  .Id\_SC = "NA"  .Id\_Grupo = "NA"  Case 3  .Id\_Comite = array\_texto(0)  .Id\_CT = array\_texto(1)  .Id\_SC = array\_texto(2)  .Id\_Grupo = "NA"  Case 4  .Id\_Comite = array\_texto(0)  .Id\_CT = array\_texto(1)  .Id\_SC = array\_texto(2)  .Id\_Grupo = array\_texto(3)  Para cada registro  Inserta datos en P\_sesion, nuevo registro  .Fecha = Format$(CDate(dr(0)), "dd/MM/yyyy")  .HoraI = Format(DTPkrhoraini.Value, "HH:mm")  .HoraT = Format(DTPkrhoratem.Value, "HH:mm")  .Asunto = TxtAsunto.Text()  .Lugar = TxtLugar.Text  .Responsable = CboResponsable.SelectedValue  .Notas = ""  .Status = 0  .Bandera = 6  "sp\_P\_Sesiones"  Se obtendrá una nueva sesion  sesion = .Id\_Sesion  If CkBxSalasANCE.CheckState = CheckState.Checked  Insertar salas  .F\_Solicitada = Format$(CDate(dr(0)), "dd/MM/yyyy")  .F\_Apartado = Format$(Now(), "dd/MM/yyyy")  .Id\_Area = objiniarray.IniGet(Application.StartupPath + "\Principal.ini", "Sistema", "id\_area")  .Id\_Propuesta = CboSalas.SelectedValue  .Id\_Responsable = CboResponsable.SelectedValue  .Id\_CoordiNAdor = CboResponsable.SelectedValue  .H\_Entrada = Format(DTPkrhoraini.Value, "HH:mm")  .H\_Salida = Format(DTPkrhoratem.Value, "HH:mm")  .No\_PersoNAs = 5  .Reunion = TxtAsunto.Text  .Servicio = ""  .Acomodo = "Acomodo Unico"  .Bandera = 1  "sp\_P\_Salas"  Se obtiene solicitud = .No\_Solicitud  Si, se seleccino un equipo  If CboEquipo.SelectedValue <> 0  .Equipo = CboEquipo.SelectedValue  .Bandera = 2  "sp\_P\_Salas" |
| Borrar | No se pueden eliminar |
| Salir | Salir del formulario |

**MEJORA:** La ventana “Historial de sesiones” actualmente no arroja mucha información. Debería aparecer el historial de todas las sesiones de ese CT, SC o GT que se han generado.



|  |  |
| --- | --- |
| Botones | Negocio y/o funcionalidad |
| Load del form | Dim min As DateTime = DateTime.Now  Dim time As TimeSpan = New TimeSpan(min.TimeOfDay.Days, 7, 0, 0, 0)  min = DateTime.Today.Add(time)  DTPkrhoraini.MinDate = min  DTPkrhoraini.MaxDate = min.AddHours(16)  DTPkrhoratem.MinDate = DTPkrhoraini.MinDate  DTPkrhoratem.MaxDate = DTPkrhoraini.MaxDate  Combo planes  .Bandera = 1  "sp\_Plan\_Buscar"  cmd.Parameters.Add("@Bandera", \_Bandera)  Combo temas  .Bandera = 11  .Id\_Plan = ""  "SP\_Programa\_Trabajo\_buscar"  cmd.Parameters.Add("@Bandera", \_Bandera)  Cargar combo tipo documentos para el grid  .Bandera = 6  "sp\_P\_Sesiones\_Buscar"  cmd.Parameters.Add("@Bandera", \_Bandera)  Llenar treeview comité  "SELECT ID\_Comite, Descripcion, Responsable FROM C\_Comite ORDER BY ID\_Comite ASC"  'PARA LLENAR LOS COMITES TÉCNICOS  'se necesita que se envíe por que Comité se filtra  "SELECT ID\_Comite, ID\_CT, Descripcion, Objetivo, Responsable FROM c\_CT WHERE ID\_Comite = '" + sComite + "' ORDER BY ID\_CT ASC"  'PARA LLENAR LOS SUB TÉCNICOS  'se necesita que se envíe por que Comité y CT se filtra  "SELECT ID\_Comite, ID\_CT, ID\_SC, Descripcion, Objetivo, Responsable FROM c\_SC WHERE ID\_Comite = '" + sComite + "' and ID\_CT='" + sCT + "' ORDER BY ID\_SC ASC"    'PARA LLENAR LOS GRUPOS DE TRABAJO  'se necesita que se envíe por que Comité, CT y SC se filtra  "SELECT ID\_Comite, ID\_CT, ID\_SC, ID\_Grupo, Descripcion, Objetivo, Responsable FROM c\_Gpos\_Trabajos WHERE ID\_Comite = '" + sComite + "' and ID\_CT='" + sCT + "' and ID\_SC='" + sSC + "' ORDER BY ID\_Grupo ASC"    Obtener las sesiones  (RegComite("ID\_Comite"), RegCT("ID\_CT"), RegSC("ID\_SC"), RegGT("ID\_Grupo"), "")  .Bandera = 9  .Id\_Comite = comite  .Id\_CT = CT  .Id\_SC = SC  .Id\_Grupo = grupo  "sp\_P\_Sesiones\_Buscar"  If grupo.Trim <> "NA" Then  comite = grupo.Trim  ElseIf SC.Trim <> "NA" Then  comite = SC.Trim  ElseIf CT.Trim <> "NA" Then  comite = CT.Trim  End If  combotable.Columns.Add("Numero")  iNumero = 0  Validar si se tiene sesión asignada  If iNumero = 0 And iEditar <> Nothing  Mensaje = "No tiene asignada ninguna Sesion" |
| Agregar |  |
| Editar |  |
| Deshacer |  |
| Guardar |  |
| Borrar |  |
| salir |  |

La ventana de “cerrar sesiones realizadas” no se utiliza, porque la sesión se cierra automáticamente después del término de la sesión y ya no puede modificarse. Sin embargo, NO hay que eliminar esta ventana.



|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |