**BOTÓN KYC**

**Objetos:**

* EvaluarSolicitudHP.aspx
* EvaluarSolicitudHP.aspx.cs

Primero realiza el llamado de la función “btnKYC\_Click” el cual realiza las siguientes funciones:

* Invoca a la función “CargaFlagAML()”, el cual invoca a la función “ObtenerSwitchAML\_543()” que realiza las siguientes operaciones:
* Inicializa: string valor = "0"; int flagPilotoAgencia = 0;
* Obtiene fecha de la solicitud, si esta diferente de vacío convierte a formato fecha, sino obtiene la fecha de hoy.
* Obtiene los datos seleccionados de los combos “codProducto, codSubProducto, codTipoEvaluacion”.
* Obtiene el código de agencia para convertirlo en numérico, en caso sea vacío será 0.
* Realiza la siguiente evaluación:

“codProducto > 0 && codSubProducto > 0 && codTipoEvaluacion > 0”, si cumple realiza las siguientes evaluaciones:

* Obtiene el llamado de la función “TipoFormulario(codProducto)”, el cual llama a la función “\_presenter.TipoFormulario\_Obtener(objProducto)” que se encuentra en el objeto “EvaluarSolicitudHPPresenter.cs”, el cual invoca a la función “\_controller.TipoFormulario\_Obtener(objProducto)” que se encuentra en el objeto “ISolcredController.cs”, invoca a la función “TipoFormulario\_Obtener(objProducto)” que se encuentra en el objeto “solhipoController.cs”, invoca a la función “ObtenerTipoFormulario(objProducto)” que se encuentra en el objeto “ISolicitud.cs”, invoca a la función “ObtenerTipoFormulario(objProducto)” que se encuentra en el objeto “Solicitud.cs”, invoca a la función “TipoFormularioDAO.Obtener(objProducto)” que se encuentra en el objeto “TipoFormularioDAO.cs”, el cual realiza un llamado al procedimiento “USP\_TIPOFORMULARIO” enviándole el parámetro “@CODIGOPRODUCTO" objProducto.CodigoProducto”, el procedimiento le devuelve el campo “InternalList” que es el campo que recibe la primera función “tipoFormulario”.
* Crea la entidad “obj, oFormulario” que se encuentra en “com.scotiabank.sts.business.entities.seguridad.STSF996\_1”
* Crea la lista “lstFormulario” que se encuentra en “com.scotiabank.sts.business.entities.seguridad.STSF996\_1”
* obj.Codigo = 543;
* lstFormulario = this.Master.ListaItemsSTSF996(obj);
* Valida si p.Codigo == tipoFormulario && p.Nro1 == 1
* Valida if (oFormulario == null) { return valor; }}
* Crea entidad oPiloto “com.scotiabank.sts.business.entities.adminpol.STSF996”
* Crea Lista lstPiloto “com.scotiabank.sts.business.entities.adminpol.STSF996”
* oPiloto.TCTABLA = 550;
* oPiloto.TCARGUM = 0;
* Obtiene la lista “lstPiloto” invocando a la función “this.presenter.Cargar\_Guia996(oPiloto)” que se encuentra en el objeto “EvaluarSolicitudHPPresenter.cs”, invoca a la función “\_controller.Cargar\_Guia996(oRequest)” que se encuentra en el objeto “IsolhipoController.cs”, invoca a la función “obj.Cargar\_Guia996(oSTSF996)” que se encuentra en el objeto “IAdminPoISTSF999.cs”, invoca a la función “Cargar\_Guia996(STSF996 oSTSF996)” que se encuentra en el objeto “AdminPoISTSF999.cs”, invoca a la función “STSF999DAO.Cargar\_Guia996(request)” que se encuentra en el objeto “STSF999DAO.cs”, el cual realiza el llamado del procedimiento “SP\_STSF996” enviando los parámetros “@TCTABLA=oTabla.TCTABLA; @TCARGUM=oTabla.TCARGUM; @Tipo=04” el cual devuelve un listado de la entidad “STSF996.cs”, el listado será en “lstPiloto”.
* Realiza un foreach con el listado “lstPiloto” para realizar un substring(21,1) al campo “TCDESCRIP” y guardarlo en el campo “flagPilotoAgencia”
* Luego se realiza un switch,

Si es 0

“flagPilotoAgencia>=DateTime.Parse(oFormulario.Abreviatura), el valor es 1

Si es 1

Crea dos entidad “objA y oAgencia” y una lista “lstAgencia”, referencia “com.scotiabank.sts.business.entities.seguridad.STSF996\_1”

objA.Codigo = 550;

lstAgencia = this.Master.ListaItemsSTSF996(objA);

oAgencia = lstAgencia.Find(delegate(com.scotiabank.sts.business.entities.seguridad.STSF996\_1 o) { return o.Codigo == codAgencia && o.Nro1 == 1; });

if (oAgencia != null && fechaSolicitud >= DateTime.Parse(oAgencia.Abreviatura))

{

valor = "1";

}

break

* Si el valor es igual a 1, Verifica si el producto, subproducto y tipo de evaluación está configurado en el SQL, realizando el llamado a la funcion “\_presenter.configuracionAMLExistente(codProducto, codSubProducto, codTipoEvaluacion)” y devuelve el flag “flagAMLExistente == 0” entonces el valor es 0
* bool flagCiiu = true;
* Evalúa si el valor = 1, realiza la siguiente operación :
* Crea la siguiente entidad “obj” y el listado “lstObj”, referencia “com.scotiabank.sts.business.entities.seguridad.STSF996\_1”
* obj = lstObj.Find(delegate(com.scotiabank.sts.business.entities.seguridad.STSF996\_1 \_Detalle) { return \_Detalle.Codigo == 6; });}
* if (obj != null) { hdfSwitchCiiu4.Value = obj.Nro1.ToString().Trim(); }
* Graba el Resultado del Switch Ciiu v.4

long nroSolicitud = txtNroSolicitud.Text.Trim() != string.Empty ? long.Parse(txtNroSolicitud.Text.Trim()) : 0;

* GrabaCampoSolicitudAML(185, hdfSwitchCiiu4.Value, 0, “”)

Obtiene los siguiente campos seleccionados: producto,subproducto, tipoevaluacion, tipodocumento y numero de documento.

Valida si los campos “producto, subproducto, tipoEvaluacion, tipoDocumento” son mayores que cero y nroDocumento sea diferente de vacio, realiza el grabado de los siguiente campos “producto, subProducto, tipoEvaluacion, tipoDocumento, nroDocumento, idCampo, valorCampo, nroItem, idSTS, nroSolicitud” llamando a la

funcion “this.\_presenter.GrabaDatosSolicitud(campos arriba)” que se encuentra en el objeto “EvaluarSolicitudHPPresenter.cs”,

invoca a la función “\_controller.GrabaDatosSolicitud(Producto, SubProducto, TipoEvaluacion, TipoDocumento, NroDocumento, idCampo, valorCampo, nroItem, idSTS, nroSolicitud)” que se encuentra en el objeto “IsolhipoController.cs”,

invoca a la funcion “GrabaDatosSolicitud(int Producto, int SubProducto, int TipoEvaluacion, int TipoDocumento, string NroDocumento, int idCampo, string valorCampo, int nroItem, string idSTS, long nroSolicitud)” que se encuentra “solhipoController\_aml.cs”,

invoca a la función “oCampo.GrabaDatosSolicitud(Producto, SubProducto, TipoEvaluacion, TipoDocumento, NroDocumento, idCampo, valorCampo, nroItem, idSTS, nroSolicitud)” que se encuentra en “ICampo.cs”,

invoca a la función “GrabaDatosSolicitud(int Producto, int SubProducto, int TipoEvaluacion, int TipoDocumento, string NroDocumento, int idCampo, string valorCampo, int nroItem, string idSTS, long nroSolicitud)” que se encuentra en el objeto “Campo.cs”,

invoca a la funcion “CampoDAO.GrabaDatosSolicitud(Producto, SubProducto, TipoEvaluacion, TipoDocumento, NroDocumento, idCampo, valorCampo, nroItem, idSTS, nroSolicitud)” que se encuentra en el objeto “CampoDAO.cs”, el cual invoca al un procedimiento “usp\_AML\_MntCampoSolicitud” envianlo los parámetros

Accion", 2

Producto", Producto

SubProducto", SubProducto

TipoEvaluacion", TipoEvaluacion

TipoDocumento", TipoDocumento

NroDocumento", NroDocumento

idCampo", idCampo

valorCampo", valorCampo

nroItem", nroItem

idSTS", idSTS

nroSolicitud", nroSolicitud

el procedimiento retorna el valor 0.

* Evalúa si hdfSwitchCiiu4.Value != "1" entonces flagCiiu = false;
* Valida si (hdfSwitchCiiu4.Value == "1" && hdfHabilitarCtrls.Value == "1")
* long NumeroSolicitud = 0;
* Evalúa txtNroSolicitud.Text != string.Empty entonces NumeroSolicitud = long.Parse(txtNroSolicitud.Text.Trim());
* Evalúa NumeroSolicitud == 0 entonces
* int ciiu3 = Ciiu3Actividad(); //Ciiu de Cliente
* Graba los siguientes campos

GrabaCampoSolicitudAML(170, ciiu3 > 0 ? ciiu3.ToString() : string.Empty, 0, "");//Ciiu Versión 3 - Cliente

GrabaCampoSolicitudAML(183, hdfCiiu3Padron.Value.ToString(), 0, "");//Ciiu Versión 3 - Padrón

GrabaCampoSolicitudAML(171, txtCodCiiu4.Text.Trim(), 0, "");//Ciiu Versión 4 - Cliente //15-02-2019 Alfredo

GrabaPreguntaSolicitudAML(5, hdfCiiu4Obligado.Value == "1" ? "Si" : "No", "Pregunta3\_9");//Sujeto Obligado//Ciiu Versión 4 - SujetoObligado

GrabaCampoSolicitudAML(184, hdfCiiu4Sensible.Value.ToString() == string.Empty ? "0" : hdfCiiu4Sensible.Value.ToString().Replace(".", ""), 0, "");//Ciiu Versión 4 - Actividad Sensible

//Cliente con Negocio

Evalúa

SI (ddlTipoClienteAML.SelectedValue == "2")

GrabaPreguntaSolicitudAML(3, "Si", "Pregunta3");//Negocio Propio

GrabaCampoSolicitudAML(25, txtRuc.Text, 0, "Pregunta3\_1"); //RUC - Persona

GrabaCampoSolicitudAML(33, txtRazonSocial.Text, 0, "Pregunta3\_2"); //Nombre del Negocio

GrabaCampoSolicitudAML(34, txtDireccionNegocio.Text, 0, "Pregunta3\_3"); //Dirección del Negocio

GrabaCampoSolicitudAML(26, txtCiiu4.Text.Trim(), 0, "Pregunta3\_4"); //CIIU - Negocio Propio (Descripción)

SINO

GrabaPreguntaSolicitudAML(3, "No", "Pregunta3");//Negocio Propio GrabaCampoSolicitudAML(25, string.Empty, 0, "Pregunta3\_1"); //RUC - Persona

GrabaCampoSolicitudAML(33, string.Empty, 0, "Pregunta3\_2"); //Nombre del Negocio

GrabaCampoSolicitudAML(34, string.Empty, 0, "Pregunta3\_3"); //Dirección del Negocio

GrabaCampoSolicitudAML(26, string.Empty, 0, "Pregunta3\_4"); //CIIU - Negocio Propio (Descripción)

* Actualiza Ciiu en AS400 en caso exista Solicitud

Evalúa si NumeroSolicitud > 0

ActualizaSancionGesintelTitular(NumeroSolicitud);

GrabaCamposAML\_SQL();

GrabarCampoAML();

GrabarPreguntasAML();

GrabarSQLCamposAML();

* Evalua si (this.hdfFlagAML.Value.Trim() == "1")

Entonces abre popup con preguntas

ScriptManager.RegisterStartupScript(this, this.GetType(), Guid.NewGuid().ToString(), "abrirPreguntasAML();", true);

**Una vez el realizado todo el flujo anterior muestra el popup “PreguntasAML.aspx”, en el popup se muestra las preguntas y los botones “Aceptar y Cerrar”.**

**Botón “Aceptar”:**

Invoca función “btnAceptar\_Click” que se encuentra en el objeto “EvaluarSolicitudHP.aspx.cs”, realiza la siguiente operación:

* Si (txtDireccionExt.Text != "" && ddlPaisExt.SelectedIndex != 0) entonces realiza el llamado a las siguientes funciones
* GrabaCampoSolicitudAML(13, this.ddlTipoResidencia.SelectedValue, 0, string.Empty);
* GrabaCampoSolicitudAML(27, this.txtDireccionExt.Text, 0, string.Empty);
* GrabaCampoSolicitudAML(29, this.ddlPaisExt.SelectedValue, 0, string.Empty);
* txtDirExt.Text = txtDireccionExt.Text + " - " + ddlPaisExt.SelectedItem.Text;

**Botón “Cerrar”:**