**Sistema Artec**

**Resumen de actividades desarrolladas**

1. Lenguaje de programación:C#
2. Desarrollo en 5 capas: Entidades, BLL, DAL, GUI y Servicios
3. Acceso único a la Base de Datos a través de la capa de servicios
4. La clase de acceso a la BD es reutilizable y permite a otras capas iniciar y finalizar una transacción
5. Mapeador resuelve mediante reflection
6. Servicios:
   * Persistencia y recupero de datos
   * Backup y Restore BD
   * Idioma
   * Login implementando patrón Singleton
   * Encriptación y aplicación de hash
   * (Bitácora)
   * (Digitos Verificadores)
   * (Manejo de excepciones)
   * (Autorización mediante Familia Patente)
7. Interfaces y lógica programada:
   * Login (Autenticación)
   * Crear Solicitud
   * Backup y Restore
   * Crear Proveedor

Descartes:

26/06/2017

1. Las compras de software son únicamente de licencias permanentes, porque las que son por suscripción requieren una erogación de gastos continua, y eso conlleva a la elevación del pedido hacia la procuradora y el trámite de adquisición lo realiza el área de administración (se iría a una cuasi Licitación)
   * Eliminé fechasus y fehafinsusc de inventario
   * Eliminé IdTipoLicencia de Bien
   * Elimine relación Bien con Tipolicencia
   * Elimine Tabla TipoLicencia
2. Adquisicion
   * Elimine NroExpediente de Adquisicion
   * Eliminé NroOrdenCompra de Adquisicion
3. Tabla AsigDetalle
   * Eliminé TiempoAsignación de AsigDetalle porque no se contemplan los prestamos
4. Tabla TipoAsignación, la eliminé porque solo hay un tipo de asignación (ya que no se contemplan prestamos)
   * Eliminé IdTipoAsignacion en AsigDetalle
5. Idioma
   * Eliminé la columna IdFormulario de Etiqueta
   * Modifique la clave primaria en etiqueta quitando de la misma IdFormulario
   * Eliminé la tabla Formulario
6. Store [dbo].[CotizacionTraerPorSolicitud]
   * Modifiqué el select

Estaba: --select Coti.IdCotizacion, Coti.MontoCotizado, Coti.FechaCotizacion, Prov.IdProveedor, Prov.AliasProv, Coti.IdPartidaDetalle, Coti.IdPartida, Det.IdSolicitudDetalle, Det.IdSolicitud

Quedó: select Coti.IdCotizacion, Coti.MontoCotizado, Coti.FechaCotizacion, Prov.IdProveedor, Prov.AliasProv, Det.IdSolicitudDetalle, Det.IdSolicitud

1. Porque pinchaba cuando quería agregar una cotización a un detalle que tenía cero cotizaciones:
   * Agregue lo resaltado en frmsolicitudmodificar:

frmCotizaciones UnFrmCotizaciones = new frmCotizaciones(unaSolicitud.unosDetallesSolicitud[e.RowIndex].unasCotizaciones, unDetSolic);

* + Agregue lo resaltado (y quite lo comentado) en frmCotizaciones:

unaCotiz.unDetalleAsociado.IdSolicitud = unDetSolic.IdSolicitud;//unasCotizaciones[0].unDetalleAsociado.IdSolicitud;

unaCotiz.unDetalleAsociado.IdSolicitudDetalle = unDetSolic.IdSolicitudDetalle;//unasCotizaciones[0].unDetalleAsociado.IdSolicitudDetalle;

1. Se eliminó table TipoAdquisicion y referencia desde Adquisición
   * Modifique AdquisicionCrear en consecuencia
2. Agregue campo IdPartidaDetalle en tabla Inventario, para poder relacionar el inventario, con la compra y la solicitud
3. Agregar prop Bien en categoría para poder acceder al tipo de bien desde el detalle de solic, creo q ya no}
4. Eliminar IdAdquisicion de tabla PartidaDetalle (posiblemente no lo use nunca)
5. Quitar el agente relacionado al software
6. Quitar la tabla depósito, porque no hay varios depósitos, además se compra lo solicitado y se lo entrega, no se stockea, lo de consultar stock era para cuando se entrega en forma parcial algo
7. IdInventario quedo como pk de Inventario y hay un índice unique entre IdBienEspecif y SerieKey en Inventario
8. Elimine columna IdBien en AsigDetalle, porque la pk de inventario ya no contiene a IdBien
9. Quitar el campo Observacion de AsigDetalle (poner solo uno en asignación quizas)
10. ~~//FALTA QUE Ponga en estado "Compradas" las cosas de un SolicDetalle~~

~~//Para hacerlo se puede consultar de nuevo la cantidad comprada y comparar con la cantidad solicitada~~

~~//Todo dentro del negocio o dal.. El código donde empzaria esto esa en frmregistrar línea 66~~

1. Al registrar bienes, solo me tiene que permitir registrar los de un mismo SolicDetalle (sino el proveedor no concuerda con todos los datalles)
2. Seguir en DALAsignacion línea 58, lo de poner como finalizado un solicDetalle
3. ~~Al registrar bienes, todos aparecen con el mismo numero de serie, REVISAR~~
4. Fijarme en DALAsignacion línea 59 que con el METODO CIERRO LA CONEXION A LA BD Y DPS EN LA SEGUNDA VUELTA DEL FOR EACH PINCHA PORQ ESTA CERRADA LA BD
5. Agregue esta línea al final del store “InventarioHardTraerListosParaAsignar” para que traiga solo los inventarios que aun no fueron entregados (sino traía todos los inventarios que están dentro de un SolicDetalle que no está finalizado aún):
   * and inv.IdEstadoInventario = 1--Disponible
6. ~~Evlauar quitar la línea “and sdet.IdEstadoSolicDetalle = @IdEstadoSolicDetalle—Adquirido”, del store “InventarioHardTraerListosParaAsignar” para permitir asignar bienes sin que todos los inventarios de un mismo SolicDetalle hayan sido adquiridos~~
7. Al registrar un inventario me vuelve a hacer update del estadoSolicDetalle a “Adquirido” en caso de que se haya comprado todo (y aunque ya esté finalizado ese SolicDetalle porq se entregó todo), fijarme en frmRegistrar en el load, para que solo cargue los bienes que restan por entregar y no todos como está ahora q te los muestra y te pone la cantidad comprada

GUARDA:Para arreglarlo modifique mal, pero arece q sigue andando todo, la línea 114 de frmbienregistrar

Antes estaba esto: LisAUXDetalles = unosDetallesBienes.Select(x => new HLPDetallesAdquisicion() { DescripCategoria = x.unaCategoria.DescripCategoria, Cantidad = x.Cantidad, IdCategoria = x.unaCategoria.IdCategoria, IdSolicitudDetalle = x.IdSolicitudDetalle }).ToList();

EN LINEA 26 BLLAsignacion, poner que solo lo haga cuando ..asdf.asd



En store CategoriaDetBienesTraerPorIdPartida, modifique la línea:

--AND SolDet.IdEstadoSolicDetalle != 2--Distinto de Finalizado

Hacia:

AND sdet.IdEstadoSolicDetalle < 3--Que el estado del detalle no sea Adquirido ni Entregado

1. Quite relacion entre partidaDetalle y Adquisicion y eliminé campo IdAquisicion de tabla PartidaDetalle

Agregué tabla RelPDetAdq para tener un N a N entre PartidaDetalle y Adquisicion

1. .Agregué campo Costo en tabla Adquisicion
2. Asociar eventos a las funciones en solicmodificar y a lo último poner todas las validaciones comentadas (esta en proceso)
3. Empezar con el agregardetalle en modificarsolic
4. Al eliminar detalles el IdDetalle queda mal (en ModificarSolic)
5. Cargar el objeto que contiene el bien seleccionado, al hacer click en un detalle

31.1 Usar el store “InventariosTraerListosParaAsignar” para restringir que no se puedan asignar bienes sino hay inventarios listos para asignar

1. Agregar inv.costo en consulta “InventariosTraerListosParaAsignar”
2. No me muestra los agentes al seleccionar un software en detallessolic
   * Revisar store “SolicDetallesTraerPorNroSolicitud”, el problema viene en que en el load de frmsolicmodificar no cargo los agentesasociados al detalle que tiene un software

**EL PROBLEMA ESTABA PRINCIPALMENTE EN QUE NO GUARDO LOS AGENTES EN la tabla RelSolDetalleAgente. RESOLVER ESTO, lo hice en la línea 733**

* + **Agregue el store SolicDetallesTraerAgentesAsociados para traer los agentes dentro de unaSolicitud.unosDetallesSolicitud[LaCorrespondiente].unosAgentes cuando hago click en un detalle y es un software**
  + **Agregue el store “RelSolDetalleAgenteAgregar” para poder guardar los agentes asociados a un software**
  + **Ya asocia un nuevo agente a un detalle que ya tiene el software (yendo por el agregar nuevo detalle, falta verificar yendo por el de modificar). El problema ahora está en que al hacer click en el detalle del software, no me muestra todos los agentes, es decir solo me muestra los q ya estaban y no los q inclui recién.**
  + **Ahora no muestra los agentes al hacer click por primera vez (sin agregar un agente antes) en un detalle de software (Aclaro que si agrego un soft, al menos desde nuevodetalle, me muestra los agentes asociados correctamente). SOLUCION: Lo solucione con las líneas 337-344**
  + **Al estar viendo el detalle de un software y ver los agentes asociados… y al agregarle un agente, me borra de la lista de la grilla los agentes ya asociados y coloca el ingresado recién, seguro q lo arreglo como los puntos anteriores**

**Solucion: Linea 343, agregue … = unosAgentesAsociados =… para que se guarde en esa lista los agentes asociados actuales**

1. **Arreglar las cosas en frmSolicCrear que fueron mejoradas en frmSolicModificar**
   * **Agregar soft, etc**
2. **Fijarme que la línea 267 de DALSolicitud funcione(lo del if)**
3. **Agregar el campo SolDetalleUID y ponerlo como PK de la tabla SolicDetalle; esto es para poder modificar y eliminar solicDetalles y que no se me generen inconsistencias.**
   * **Corregir todo en el programa**
   * **Corregir todo en los store**
4. **Mejorar el servicio Mail, para que los datos estén en la BD y encriptado o con hash la contraseña del remitente**
5. **Servicio Mail:** 
   * **Cargar los datos del host al iniciar el sistema (en el login), ahora está en el evento btnSolicitar de cotizaciones**
   * **Se coloco en bd la dirección de mail general para enviar los mails**
   * **Colocar en el destinatario el mail del proveedor seleccionado, etc**
   * **Colocar en el frm de solicitar cotizaciones el textbox con el cuerpo del msj pre-seteado para que se pueda modificar si así se quisiese**
   * **Puse un thread para que al enviar mails se pueda cerrar la ventana de las cotizaciones y no se bloquee ahí**
6. **Corregir Idioma para que las consultas a la bd se hagan desde DAL.Servicios.ServicioIdioma**
   * **Quedaría Framework.servicios.ServicioIdioma con lo mismo, pero quitando la parte de consultar a la bd. Y quedaría “EtiquetasTraerTodosPorIdioma (int elIdioma)” dentro de dal.servicios.servicioIdioma, y desde ahí se llamaría a framework.servicios.Traducir()**
   * **CambiarIdioma cambiaría en que el idioma queda relacionado con el usuario, hay que crear un campo en el usuario para relacionar el ultimo idioma. Y se modificaría toda la función cambiarIdioma**
7. **Realizar el modificar idioma en frmPrincipal con:**

//Replico la traducción a TODOS LOS FORMULARIOS ABIERTOS ESTO HAY QUE USARLO AL CAMBIAR EL COMBOBOX DEL IDIOMA

ServicioIdioma.CambiarIdioma(this.FindForm(), (int)cboIdioma.SelectedValue);

foreach (Control unForm in Application.OpenForms)

{

ServicioIdioma.Traducir(unForm, ServicioLogin.GetLoginUnico().UsuarioLogueado.IdiomaUsuarioActual);

}

1. Antes cambiar en bd el idioma actual del empleado: Esto se hace en la función cambiarIdioma dentro de ServicioIdioma
2. **Realizar el cbobox en Principal para modiicar el idioma por default, la función ya está hecha en ServicioIdioma**
3. **Tener cuidado con el brake línea 57 en ServicioIdioma, lo puse para que corte al encontrar coincidencia entre el nombre del control y la etiqueta. Funciona bien**
4. **frmPartidaSolicitar, empezado para que funcione desde Principal directamente; ya busca las solicitudes, hay q hacer que cargue los detalles y cotizaciones; en línea 535 frmpartidasolicitar hay que cargar los detalles dentro de unaSolicitud, porq en dicha variable no están esos datos.**

**Hay que revisar la carga de las cotizaciones, porque pincha al hacer click en un detalle (muy posiblemente sea el orden de las columnas dentro de la grilla porq tira una excepci´n que dice algo así) Hecho, pero probar si todo funciona bien. FIJARME que tenga detalles y cotizaciones cargadas la solicitud, sino tiene, no cargar los Solicdetalles directamente. Que solo traiga los detalles que están pendientes de compra, y que si no tiene nada la solic de detalles pendientes directamente msj diciendo que no tiene detalles pendientes. Esto anda, pero falta el msj de cuando ya esta comprado o finalizado lo del detalle**

1. **Usar -res.First().NombreDependencia- para colocar en unaDep la dep seleccionada al escribirla a mano sin seleccionar del cbo. Y dps llamar a lo de cargar los agentes responsables**
2. **El idioma ingles quita la columna eliminar en frmmodificar**
3. **En frmSolicitarPartida Cuando hago click en un detalle y/o cotización, dps no puedo modificar los textbox de solicitud y dependencia**
4. **Ya puse que al agregar más de dos cotizaciones el solicdetalle se ponga en estado cotizado, pero falta que se actualice la vista de la grilla en frmmodificar (ver si hay que actualizar el objeto además de la grilla).**

**NO ACTUALIZA BIEN EL STRING DE LA GRILLA**

1. **FRMModificar: Corroborar al intentar eliminar un detalle, que no tenga partidas asociadas**
2. **Agregar el nombre de los formularios en idioma y filtrar la consulta a la bd y etiquetascompartidas por formulario, para optimizar los tiempos de carga del idioma**
3. **---------------------------------------------------------**
4. **Estoy haciendo el punto 36**
   * **Agregue UIDSolicDetalle**
   * **Agregue det.UIDSolicDetalle en select de store SolicDetallesTraerPorNroSolicitud**
   * **Agregue prop IUDSolicDetalle en SolicDetalle**
   * **Agregue línea 72 unDet.UIDSolicDetalle = (int)row["UIDSolicDetalle"]; en MapearSolicDetalles**
   * **Agregue línea 226 grillaDetalles.Columns["UIDSolicDetalle"].Visible = false; para que no se vea el UIDSolicDetalle en la grilla de frmModificarSolicitud**

**Todo Parece seguir funcionando**

* + **Agregue det.UIDSolicDetalle en select de store CotizacionTraerPorSolicitud**
  + **Cuando se elimina un detalle, listo hay q eliminar todo en la bd y registrar todo de nuevo**
  + **Cree Store SolicDetalleDeletePorSolicitud para eliminar todos los detalles asociados a una solicitud**
  + **Cree store SolicitudModificar para modificar los datos “directos” de la solicitud**
  + **Puse on delete cascade en foreign key a solicdetalle desde relcotsoldet**
  + **Puse on delete cascade en foreign key a solicdetalle desde relSolDetalleAgente**
  + **Pincha porque se cierra la transacción en:**

**if (GestorCategoria.TipoBienTraerTipoBienPorIdCategoria(item.unaCategoria.IdCategoria).IdTipoBien == (int)TipoBien.EnumTipoBien.Soft)**

**Lo más correcto sería corregir que el ejecutardataset no tenga integrada la transacción**

**Pero por ahora puse un store que pregunte por scalar y trae un int**

**INCONSISTENCIAS en las cotizaciones, porq al eliminar todo, no las vuelve a cargar. Porque falta agregar el registr de cotizaciones en SolicitudModificarConDetallesEliminados. Fijarme el vinculo que queda entre las cotiz y los solicDetalles, ya que las cotiz se guardan sin confirmación. Quitar el registro en bd de las cotiz en frmcotzaciones Linea 146 (aparentemente funciona) ~~y la actualización desde bd en frmsolicmodificar Linea 395~~; guardar todo al hacer click en modificarSolicitud (las cotiz se ven desde el btnModifSOlic con la modificación de la línea 146).**

* + **Colocar la eliminación de los detalles dentro de SolicitudModificarConDetallesEliminados, sino elimina los detalles antes de intentar guardar la modificación, y si la misma falla, quedan los detalles en blanco.**
  + **Fijarme que se elimine y cree nuevamente el ResSolDetCotiz (Lineas 326-331)**
  + **Fijarme ídem con ResSolDetAgente**
  + **Fijarme de eliminar tmb las cotizaciones en bd, para regenerarlas dps (Lineas 326-331)**
  + **Idem con la rel de agentes asociados a un detalle**
  + **Pincha al eliminar un detalle de software (que ya estaba), y se quiere agregar de nuevo el software y con el mismo usuario (dice que el agente ya se encuentra asociado al software), fijarme el objeto unosAgentesAsociados**

1. **En frmPartidaAsociar Traer las partidas asociadas a la solicitud seleccionada, y dps según la partida seleccionada traer los detalles de la misma. Fijarme que se borren los campos y grillas al buscar de nuevo**
2. **Impedir el inicio de rendicionCrear si la partida no está asociada o no hay partida creada, y si no tiene adquisiciones**
3. **Traer desde la consulta todos los datos, el inventario, el bien y del solicDetalle directamente para poder guardar la marca y modelo en HLPAsigs y que venga filtrado ya con el detalle q sí tiene algo para asignar**

**Linea 68 empeiza esto.**

**Lo Solucione con un “mediador” de objetos inventario**

**Cuidado porque modifiqué el store invtraerlistosparaasignarporDetalle, quizás tenga que revertirlo**

1. **En frmRegistrarBien, al confirmar no está actualizando el campo “comprado” en la grilla de los detalles de la partida. Pincha en el código y con la base**
   * **Store “InventarioAdquiridoCantPorPartDetalle” no devuelve nada, REVISAR ESTO PRIMERO. Linea 73 frmBienRegistrar. Le comente la línea “--and sdet.IdEstadoSolicDetalle != 2--Distinto a finalizado**
2. **Pincha el frmAsignar en InventariosTraerListosParaAsignar**

**Fijarme a q store apunta inventarioSoftTraerListosParaAsignar**

* + **Comente la liena en el store:**

**--and sdet.IdSolicitudDetalle = @IdSolicitudDetalle COMENTADO 30/10/2017 (AL IGUAL QUE EN LOS PARAMETROS)**

1. **Fijarme que en la asignación se contemple lo de los agentes**
2. **Fijarme si está creado el documento del remito**
3. **Trabajando con la modiicación de la partida ya creada** 
   * **Cuidado con el IDPartida, no se si lo envío cuando ya existe una a PartidaModificar**
   * **MODIF PARTIDA: SOLO SE MODIFICAN LAS COTIZACIONES ASOCIADAS, EN CASO DE QUERER ELIMINAR UN ITEM, HAY QUE ELIMINAR EL SOLICDETALLE MAS Q LA PARTIDADETALLE**
4. **Agregar validación de que al modificar partida no este asociada a una partida ya entregada**
5. **Impedir que se modifique una partida si ya hay un NroPartida asociado**

**Commit 12/11/2017 “Fix varios” (para volver atrás si es que no funciona todo lo siguiente)**

1. **EN frmPartidaSolicitar mostrar sólo los detalles cotizados y que no estén en una partida**
   1. **Esto hace que tenga que crear un buscarPartida, para llegar a modificarPartida,**
   2. **poner para agregar cotizaciones (quizás sería mejor mostrar las cotizaciones de los detalles y tildar las que fueron asociadas a la partida), poner tmb para eliminar cotizaciones, NO hacer lo de tildar**
   3. **El 2 no se hace, mejor agregar una grilla con las cotizaciones no asociadas y poner botón para agregar a las asociadas~~, además agregar para poder cargar nuevas cotizaciones.~~**
   4. **Codigo para modificar el MontoSolic en la BD, línea 169 frmpartidaModificar**
2. **Crear Mapear Idioma**
3. **Borrar DescripTipoDependencia, arrojaba excepciones por el mapeador automático**
   * **Elimino las RelCotizPartDetalle**

**delete from RelCotizPartDetalle**

**WHERE IdCotizacion = 1190**

* + **Elimino el partDetalle**

**delete from PartidaDetalle**

**where IdPartidaDetalle = 2**

**and IdPartida = 1072**

* + **Esto funcionaba porque IdPartidaDetalle era PK**

**update PartidaDetalle**

**set IdPartidaDetalle = (IdPartidaDetalle-1)**

**where IdPartidaDetalle > 2**

1. **Cree un brunch nuevo “ConClavesUID” y ArtecDiploma ahora apunta y sincroniza con ese brunch.**

**Descargue el brunch master en D:\DocumentosDescargas\uni\Diploma\ArtecDiploma\_master para no andar cambiando de brunch en la misma carpeta, pero no funciona la conexión con la bd, REVISARLO**

1. **A su vez ya modifique la primaryKey de PartidaDetalle (IdPartida, UIDPartidaDetalle)**
2. **Esto funciona ahora que le cambié la PK a PartidaDetalle por UIDPartidaDetalle**
   * **Elimino las RelCotizPartDetalle**

**delete from RelCotizPartDetalle**

**WHERE IdCotizacion in (<1190,xx,xx>)**

* + **Elimino el partDetalle**

**delete from PartidaDetalle**

**where IdPartidaDetalle = 2**

**and IdPartida = <1072>**

* + **Esto funciona ahora que le cambié**

**update PartidaDetalle**

**set IdPartidaDetalle = (IdPartidaDetalle-1)**

**where IdPartidaDetalle > 2**

**and IdPartida = <1074>**

1. **Puse como PK en RelCotizPartDetalle (IdCotizacion, IdPartida, IUDPartidaDetalle)**

****

1. **Modificando** PartidaCrear **en DALPartida**
   * **Linea 79. Ahora: new SqlParameter("@UIDPartidaDetalle", item.IdPartidaDetalle)**
   * **Le agregue Select Scope identity() a store [PartidaDetalleCrearSinCotiz] para poder usarlo en las relaciones con las cotizaciones**
   * **Lineas 70 y 71 quedaron así:**

**var ResultadoUIDPDet = (decimal)FRAMEWORK.Persistencia.MotorBD.EjecutarScalar(CommandType.StoredProcedure, "PartidaDetalleCrearSinCotiz", parametersPartidaDetalles);**

**int IDDevueltoUIDPDet = Decimal.ToInt32(ResultadoUIDPDet);**

* + **Linea 80 quedó así:**

**new SqlParameter("@UIDPartidaDetalle", IDDevueltoUIDPDet)**

* + **Se modificó store [RelCotizPartDetalleCrear] cambiando el idpartidadetalle por UIDPartidaDetalle**
  + **En store [PartidaDetalleTraerTodosPorNroPart] agregue en el select el campo UIDPartidaDetalle**
  + **Agregue al mapeador** MapearPartidaDetalles **para que agregue el UIDPartidaDetalle**
  + **Modifique el store [CotizacionTraerPorIdPartidaDetalle] para que en vez reemplazando el idpartidadetalle por UIDPartidaDetalle**

**ON Rel.IdPartida = Det.IdPartida AND Rel.UIDPartidaDetalle = Det.UIDPartidaDetalle**

* + **Modifique todo de unManagerCotizacion.CotizacionTraerPorIdPartidaDetalle para que sea UIDPartidaDetalle, hasta los nombres de las funciones y el store**

1. **REVISAR CotizacionAsociarConPartidaDetalle. Listo**
2. **Modifique store [SolicDetallePartidaDetalleAsociacionTraer] para q traiga UIDPartidaDetalle tmb**
3. **Revisar registrarBien y generarrendicion**
4. **No se actualiza el estado en frmmodificar de los detallesSolic cuando creo una partida**
5. **Para que no se arrojen excepciones por no haber datos en la consulta: verifico el count de rows del ds antes de llamar al mapeador**

**List<Partida> ListaPartidas = new List<Partida>();**

**if (ds.Tables[0].Rows.Count > 0)**

**{**

**ListaPartidas = MapearPartidas(ds);**

**}**

**return ListaPartidas;**

1. **Gestión de Excepciones:**
   * **En el throw enviar el msj de la exc a la bitácora y hacer un ScopeIdentity**
   * **En el throw retornar para la función de donde se ejecutó el mismo, el ID de la excepción grabada**
   * **En la gui, leer los datos de la bitácora según el id devuelto, y emitir un msj al usuario**
   * Modifiqué tabla inventario cambie nombre a de IdPartidaDetalle a UIDPartidaDetalle, agregue la referencia hacia PartidaDetalle
   * Modifique store [InventarioAdquiridoCantPorPartDetalle] porque comparaba con idPartDetalle como PK, le puse UIDPartidaDetalle
   * Modificados InventarioHardTraerListosParaAsignar tmb tiene lo indicado en el anterior punto, ídem [InventariosTraerListosParaAsignarPorSolicDetalle] y [InventariosTraerListosParaAsignar]
   * Modifique store PartidaDetalleUIDPorIdCategoriaIdPartida para que devuelva UIDPartidaDetalle en vez de IdPartidaDetalle, e hice los cambios de nombre tmb en la app
   * Modifiqué el store [InventarioHardCrear] para cambiando IdPartidaDetalle por UIDPartidaDetalle e hice el cambio tmb en la DALInventarioHard de la app
   * Idem anterior con [InventarioSoftCrear]
   * Modifique el mapeador MapearInventario porque tenia idpartidadetalle
   * Fijarme que el alta de inventario de soft usa otro store (distinta la consulta, no solo el tipoBien) que el de hard, PUSE PARA PROBAR InventarioHardTraerListosParaAsignar, pero hay que ver si hay q cambiarlo. ANDA
   * ~~Crear la lógica y store para guardar en la relación RelPDetAdq las partidasDetalles ya adquiridas~~
2. Hacer que la relación RelPDetAqd sea una rel entre partidaDetalle, Adquis e Inventario; por ende modificar el alta de los inventarios
   * Modifique store: [InventarioAdquiridoCantPorPartDetalle]

Para usar la interrelacion con relpdetadq y dejar de usar directamente desde inventario a partidadetalle

**select sdet.IdSolicitudDetalle, count(inv.IdInventario) as Comprado from Inventario inv**

**INNER JOIN RelPDetAdq pdetadq**

**ON inv.IdInventario = pdetadq.IdInventario**

**INNER JOIN PartidaDetalle pdet**

**ON pdetadq.IdPartida = pdet.IdPartida and pdetadq.UIDPartidaDetalle = pdet.UIDPartidaDetalle**

**LEFT JOIN SolicDetalle sdet**

**ON pdet.IdSolicitud = sdet.IdSolicitud and pdet.IdSolicitudDetalle = sdet.IdSolicitudDetalle**

**WHERE pdet.IdPartida = @IdPartida2**

**--and sdet.IdEstadoSolicDetalle != 2--Distinto a finalizado 30/10/2017 Comente esto para probar**

**group by sdet.IdSolicitudDetalle**

* + Modifique store [InventarioHardCrear] quitando idadq, uidpartdetalle y idpartida
  + Cree store RelPDetAdqCrear
  + Modifique dalInventarioHard para que se apegue a los nuevos store
  + Modifique store [InventarioSoftCrear] para q funcione la app con relpdetadqcrear en soft tmb
  + Elimine en la tabla Inventario los campos IdAdqsuisicion, UIDPartidaDetalle y IdPartida, que ya no los uso porq existe RelPdetAdq
  + Le agregue scope\_identity a stores inventarioHardCrear e InventarioSoftCrear y modifiqué la app para que tome los ids del scope y los envie como parámetro al store RelPdetAdqCrear

06/12/2017

* + Modifique store [InventarioHardTraerListosParaAsignar] agregando inner join con relpdetadq
  + Modifique store [InventariosTraerListosParaAsignar] agregando relpdetadq
  + Modifique store [AdquisicionesConBienesPorIdPartida]
  + ESTE STORE DEVUELVE QUE EL COSTO ES NULL, corregido, - Ahora se guarda el costo al dar de alta un inventario

1. Que no me permita dar de alta inventarios, si todavía no fue asociada la partida,
   * Agregue en BD nuevo estado “Partida” y corri todo uno para arriba desde “comprar”
   * Agregue en clase EstadoSolicDetalle el nuevo estado dentro de EnumEstadoSolicDetalle
   * Lineas 92 DALPartida, coloque el estado Partida
   * Colocar estado solicDetalle en Comprar cuando se asocia la partida
   * Modifique store [CategoriaDetBienesTraerPorIdPartida] porque sino me traía igual los SolicDetalles que no estaban para comprar todavia
2. Imprimir un retiro de caja si al comprar se pasa de lo otorgado en la partida asociada, falta la traducción a ingles
3. Revisar línea 249 frmPartidaModificar, el if. ->Queda así el if
4. ABM dependencia TRAYENDO LOS CARGOS lin 50 frmDependenciaModificar
   * No borrar, solo poner como inhabilitada
   * Select All con habilitadas
   * Cambiar al responsable
   * A parte, para que al escribir a mano la fiscalía me busque los datos, y que cuando a mano borre y no elija ninguna fiscalía me borre los datos:

private void txtDependencia\_TextChanged(object sender, EventArgs e)

{

DepSeleccionada = null;

txtTipoDep.Clear();

GrillaAgentes.DataSource = null;

---------------------------------------------

if (res.Count == 1 && string.Equals(res.First().NombreDependencia, txtDependencia.Text))

{

cboDep.Visible = false;

cboDep.DroppedDown = false;

cboDep.DataSource = res;

cboDep.SelectedIndex = 0;

cboDep\_SelectionChangeCommitted(cboDep, new EventArgs());

1. Mapeador TipoDependencia y Dependencia
2. ABM agentes
   * Cambiarlos de Fiscalía, fijarme que no afecte las solicitudes pendientes
   * No delete
3. ABM usuarios
   * No delete
4. Eliminar el estado de solicDetalle en el miniformulario de modifSolicitud porque el Estado es seteado automáticamente
5. ABM Proveedor
6. Optimizar form envio cotizaciones a proveedores
7. Parametrizar el envío de mails
8. Agregar el adjunto a las cotizaciones
9. SolicitarPartida desde frmModificarSolic, que deje crear una partida cuando hay al menos un SolicDetalle con todas las cotiz
   * REVISAR lo mismo en frmPartidaSolicitar, porque ahora se traba ahí
10. Permitir eliminar una partida (en buscar partidas) si no tiene una asociación
11. Controlar si la asociación de la partida con una real, se elimina al ingresar espacios en blanco (en partidaAsociar)
12. Buscar Adquisiciones, BM Adqsui
13. Buscar Bienes/Inventarios, BM Inventarios
14. ABM Bienes, ABM Modelo y Marca
15. Mejorar form frmRegistrarBienes
16. Buscar Rendiciones, Bajas (las modif se hacen directamente, revisar)
17. ABM Politicas
18. Grabación de adjuntos

You need to create an event handler for the user control that is raised when an event from within the user control is fired. This will allow you to bubble the event up the chain so you can handle the event from the form.

When clicking Button1 on the UserControl, i'll fire Button1\_Click which triggers UserControl\_ButtonClick on the form:

User control:

public event EventHandler ButtonClick;

protected void Button1\_Click(object sender, EventArgs e)

{

//bubble the event up to the parent

if (this.ButtonClick!= null)

this.ButtonClick(this, e);

}

Form:

UserControl1.ButtonClick += new EventHandler(UserControl\_ButtonClick);

protected void UserControl\_ButtonClick(object sender, EventArgs e)

{

//handle the event

}

O sino::

***User Control***

public event EventHandler StatusUpdated;

private void FunctionThatRaisesEvent()

{

//Null check makes sure the main page is attached to the event

if (this.StatusUpdated != null)

this.StatusUpdated(this, new EventArgs());

}

***Main Page/Form***

public void MyApp()

{

//USERCONTROL = your control with the StatusUpdated event

this.USERCONTROL.StatusUpdated += new EventHandler(MyEventHandlerFunction\_StatusUpdated);

}

public void MyEventHandlerFunction\_StatusUpdated(object sender, EventArgs e)

{

//your code here

}