

***Manejo de Listas Protocolares
Banco Caroní***

Documentación Técnica

Manual II

(Descripción de Programas)

Junio, 2008

INTRODUCCIÓN

En este documento se describen los programas desarrollados para el Sistema de Manejo de Listas Protocolares del Banco Caroní. Fueron escritos 5 Programas para reportes y especiales, 4 menus y programas de arranque, 6 programas a ser incluidos en el rec.htm (post, fórmula, pre,etc.) y 4 programas para varias funciones.

1. Programas de Arranque y Menús

/**

* Program:

* Objective:

* Description:

momento, si no es Responsable

*

pendiente es diseccionado

*

* @query_string

* @post

* @get

* @author

* @called_by:

* @calls:

*/

Relacionados/arranque.php

Direccionar a los usuarios al iniciar sesion.

Se obtiene el perfil que esta logeado en ese

o si lo es pero no tiene Relacionados

al programa menu.htm.

N/A

N/A

N/A

Darwin Escalona

menu.htm

menu.htm y listarsup.php

//Obtener el Perfil del "Logeado" . Si el mismo es diferente a 99 (Responsable) lo redirecciona al programa menu.htm

//Crear Consulta de aquellos responsables que tengan Relacionados sin clasificar pendientes en el campo RELMLink1_AGE que almacena a las personas que haya corrido el post de REL

//Si no tiene Relacionados pendientes redirecciona el Responsable al programa menu.htm

//Si el Responsable tiene Relacionados pendientes se genera una lista con sus códigos y se muestran en un listarsup.

/**

* Program: **Relacionados/menu.htm**

* Objective: Mostrar un menú de trabajo para los Responsables de las listas protocolares (Dueños)

* Description: determinar el perfil y usuario que está “logueado” para desplegar el menú según estas condiciones.

* @query_string None

* @post None

* @get None

* @author Darwin Escalona

* @called_by: Enlace Especial A y arranque.php

* @calls: ingresorelacionados.htm, arranque.php y listarsup.php

*/

//determinar el perfil y usuario que está “logueado” para desplegar el menú según estas condiciones.

/**

* Program: **Relacionados/menuMercadeo.htm**
* Objective: Mostrar un menú de trabajo para las personas de mercadeo.
* Description: Despliega el menú con las distintas opciones de reportes que se muestran
* mediante listarsup y otros programas.
* @query_string None
* @post None
* @get None
* @author Darwin Escalona
* @called_by: Enlace especial D
* @calls: reporteMercadeo.php, reporteSClasificacion.php, ingresorelacionados.htm y
listarsup.php
*/

//Consultar los códigos de los relacionados con mas de un cargo.

// Reporte de relacionados con mas de un cargo no se establecen condiciones solo se lista los
códigos de la consulta anterior

//Relacionados pendientes por ser borrados de la lista protocolar

//Reporte con datos básicos de la plantilla del BC

//Reporte con datos básicos de todos los relacionados.

/**

* Program: **Entregable/MostrarMenu.php**

* Objective: Desplegar menú para administración de entregables.

* Description: Pendiente por aprobación, mostrar solo al confirmar, se estima que en la 2da etapa del proyecto

* @query_string None

* @post None

* @get None

* @author Darwin Escalona

* @called_by: Enlace Especial E

* @calls: Listarsup.php y reporteEntregable.php

*/

//Despliegue de menú

//Reporte Básico de Seguimiento a entregables

1. Programas para Reportes Especiales.

/**

* Program: **Entregable/reporteEntregable.php**

* Objective: Mostrar una lista desplegable para que el usuario seleccione el entregable a consultar.

* Description: Al seleccionar algún entregable se genera automáticamente un listarsup con los seguimientos

* enlazados al mismo.

* @query_string None

* @post None

* @get None

* @author Darwin Escalona

* @called_by: Enlace Especial E

* @calls: Listarsup.php

*/

//generar lista desplegable con todos los eventos que están en la base de datos.

/**

* Program: **Relacionados/reporteMercadeo.php**
* Objective: Desplegar un menú de opciones donde el usuario podrá elegir entre ver todas las listas
* protocolos agrupadas por dueño, o ver solo la de un dueño.
* Description: Se muestra un botón mostrar todos donde podrán verse todas las clasificaciones de las
diferentes listas
* protocolos agrupadas por Responsable además muestra una lista desplegable con los
nombres de los Responsables registrados en caso de querer consultar solo la lista de
alguno en particular.
* @query_string None
* @post None
* @get None
* @author Darwin Escalona
* @called_by: Submenú de enlace especial D
* @calls: Listarsup
*/

//Llamado a listarsup para mostrar las coincidencias

//Mostrar un select con los nombre de los distintos usuarios registrados en la tabla AGE

/**

* Program: **Relacionados/reporteSClasificacion.php**
* Objective: Desplegar un menú de opciones donde el usuario podrá elegir entre ver todos los relacionados sin clasificaciones agrupados por dueño, o ver solo los de un dueño.
* Description: Se muestra un botón mostrar todos donde podrán verse todas las clasificaciones de los relacionados sin clasificaciones
* agrupados por Responsable además muestra una lista desplegable con los nombres
* de los Responsables registrados en caso de querer consultar solo los de alguno en particular.
* @query_string None
* @post None
* @get None
* @author Darwin Escalona
* @called_by: Submenú de enlace especial D
* @calls: Listarsup
*/

//Llamado a listarsup para mostrar las coincidencias

//Mostrar un select con los nombre de los distintos usuarios registrados en la tabla AGE

/**

* Program: **Relacionados/ExtendidoREL.php**

* Objective: Mostrar los detalles, cargos y clasificaciones de los relacionados a manera de resumen.

* Description: Se leen los datos básicos del relacionado en cuestión y se imprimen en una tabla, por otro lado se leen sus cargos y dependiendo si es Responsable o no le muestra solo el que este le haya dado si lo tiene y si no le permite agregar uno o mas por ultimo se leen las clasificaciones y al igual que en el anterior dependiendo si es Responsable o no le muestra solo el que este le haya dado si lo tiene y si no le permite agregar uno.

* @query_string \$Codigo el código del relacionado al cual se le hará el extendido.

* @post None

* @get None

* @author Darwin Escalona

* @called_by: Listarsup.php

* @calls: None

*/

//Recupera los datos básicos a ser mostrados en el encabezado del Extendido.

//Determinar las condiciones dependiendo el usuario como se mostraran los cargos

//Determinar las condiciones como se mostraran las clasificaciones dependiendo el usuario y si ya había echo una clasificación.

/**

* Program: **Relacionados/ConsultaRelacionados.php**
* Objective: Mostrar las Clasificaciones del relacionados y actualizar la fecha de ultima
* consulta del responsable en la Tabla AGE.
* Description: Reporte modificable con los relacionados a los que el le ha dado una clasificación.
* (Aquí podrá modificar su clasificación)
* @query_string None
* @post None
* @get None
* @author Darwin Escalona
* @called_by: Enlace Especial B
* @calls: Listarsup.php
*/

//Actualización de la fecha de consulta en la tabla AGE

2. Programas incluidos en el rec.htm

```
/**
```

```
* Program:      Relacionados/CLAformula.php
* Objective:    Verificar si hubo cambios para armar el histórico, consultar cuales son las
*               clasificaciones y concatenarlos separados por coma para ser almacenados en el
*               campo RELBloque6.
* Description:  Programa Fórmula de la Tabla: CLA. Este programa realiza lo siguiente:
*               •En la tabla CLASIFICACIONES:
*                   A. Actualizar el campo: Historial de Cambios. Se debe grabar, la fecha del
*                   cambio, el usuario y el cambio realizado. Se concatena con la información
*                   que ya tenía el campo anteriormente.
*               •En la tabla RELACIONADOS:
*                   A. Buscar para ese Relacionado, todos las relaciones que tiene en la tabla
*                   CLASIFICACION, concatenarlos separados por coma y almacenar el
*                   resultado en el campo RELBloque6
* @query_string  None
* @post          None
* @get           None
* @author        Darwin Escalona
* @called_by:    rec.php y listarsup editables de CLASIFICACIONES
* @calls:        None
*/
```

// GrabarHistoriaREC(\$campo,\$nombre) Función para verificar los campos modificados por medio de rec.php recibe como parámetros el campo a evaluar con y el nombre externo del mismo

// Pendiente() Arma el enlace múltiple ahora sin el responsable que esta clasificando en ese momento al relacionado. Retorna una cadena con los responsables que aun lo tienen pendiente.

```
// if(strstr($_SERVER['PHP_SELF'], "rec.php")) Verifica si se esta en rec.php o listarsup
//Código del relacionado que se esta modificando
//Consultar los registros de CLA donde se encuentre el relacionado
//Recorrer los registros de el relacionado para determinar si esta activo, las clasificaciones que
//posee
//Recorrer los registros de el relacionado para determinar las clasificaciones que posee
//Verificar si el relacionado esta activo para tomar en cuenta sus motivos y clasificaciones
// if($OperType=="Update") Determinar si se esta haciendo una actualización para llamar a la
//GrabarHistoriaREC que devuelve el resumen de la operación realizada
//Se lleva la variable $Cambios con los posibles cambios realizados, si se encontraron cambios
//colocar un rastro y grabarlo en base de datos
//Se unifica el arreglo de las relaciones y las convierte en una cadena para ser almacenada en el
//campo RELBloque6 y almacenar los responsables pendientes.
//Si no se esta en rec.php es decir si se esta en listarsup y arma el histórico
//Verificar si el relacionado esta activo para tomar en cuenta sus motivos y clasificaciones
//Se unifica el arreglo de las relaciones y las convierte en una cadena para ser almacenada en el
```

campo RELBloque6 y almacenar los responsables pendientes.

/**

* Program: **Relacionados/PostCRG.php**

* Objective: Llenar el campo de estados en la tabla de relacionados con una cadena de todos los estados separados por coma.

* Description: Selecciona todos los cargos que coincidan con el relacionado, y al recorrerlos se

* almacenan los estados separados por comas y se actualiza el campo de estados con los nuevos valores

* @query_string None

* @post None

* @get None

* @author Darwin Escalona

* @called_by: rec.htm de la tabla CARGOS

* @calls: None

*/

//Obtener código del relacionado que se está trabajando.

//Seleccionar todos los cargos del Relacionado

//Recorrer los registros para armar la lista de los estados separada por coma

//Actualizar el campo de estados con el nuevo valor obtenido.

/**

* Program: **Relacionados/PostENT.php**

* Objective: Establecer las condiciones para listar los posible relacionados a ser asignados al entregable.

* Description: Despliega un formulario donde el usuario establecerá las condiciones para los relacionados que

* quiere enlazar con el entregable, estos le son mostrado en un listarsup con checkbox para que seleccione los que considere apropiados.

* @query_string None

* @post None

* @get None

* @author Darwin Escalona

* @called_by: rec.htm de la tabla ENTREGABLES

* @calls: Listarsup.php, utilitiesasesor.php, calendar.css, calendar.js, styleFormato.css

*/

JS:

//Cambiar el formato y los separadores de la fecha para que sea compatible con Mysql

//Llenar las condiciones de búsqueda de acuerdo a los ítems seleccionados en el formulario.

//Condiciones de las listas multiselección

//Condiciones extras.

//Asignación de las condiciones al hidden correspondiente de los parámetros de listarsup

//Pase de la Fecha por URL

//Función que obtiene los campos seleccionados en una lista multiselección y los retorna en forma de condición

PHP:

//Construir select de estados a partir de los valores correspondientes a esta lista en la tabla list

//Construir select de categorías a partir de los valores correspondientes a esta lista en la tabla list

//Checkbox para condicionar los motivos para listar a los relacionados

/**

* Program: **Relacionados/PostREL.php**
* Objective: Verificar si ya existen clasificaciones de el dueño para el relacionado en cuestión de
* manera que se la muestre, y sino le permita agregar una.
* Description: Chequear la operación que se esta haciendo, si es nuevo registro o no tiene clasificación
* para ese dueño permite agregar de lo contrario muestra la que encontró en forma editable.
* @query_string \$OperType,\$XXXCodigo,\$CookieAGENTE,\$EEE,\$COKO,\$Dominio
* @post None
* @get None
* @author Darwin Escalona
* @called_by: rec.php de la tabla relacionados
* @calls: rec.htm de CLA.
*/

//Esta función agrega al enlace múltiple los códigos de los AGE-Responsables que corren el Post ya que esto da por entendido que tienen un interés y se los mostrara al hacer logging sino les da una clasificación

//Variable que almacena el código del relacionado

//Si se esta agregando un nuevo registro enviar directamente al rec de CLA para que incluya su clasificación

//Si se esta Actualizando el relacionado se hace una consulta de las clasificaciones del mismo que coincidan con el dueño si la tiene abre el respectivo rec para editarla sino las tiene la abre para incluir

//Consulta de la clasificación que coincida con el Relacionado y el Usuario

//Si existe alguna clasificación de el usuario se la muestra para editar sino abre el rec para agregar.

/**

* Program: **Relacionados/PreCRG.php**
* Objective: Establecer solo una dirección de envió, (la última en ser marcada como tal)
* Description: Verificar que entre todos los Registros de un Relacionado, sólo 1 de ellos
* tengo el checkbox de Usar para Envíos, marcado. Si al ejecutarse el programa,
* este campo ha sido marcado, se debe buscar si algún registro lo tenía también
* marcado, si es así, éste se desmarca y se deja marcado el actual.
* @query_string Parámetro recibido en el query string
* @post None
* @get None
* @author Darwin Escalona
* @called_by: rec.htm de CARGOS
* @calls: None
*/

//Recuperar el nuevo valor del checkbox usar como dirección de envió y el código del relacionado

//Seleccionar aquellos registros de cargos del relacionado

//Recorrer los registros del relacionado

//Si esta marcado ese checkbox desmarcar

/**

* Program: **Relacionados/RELformula.php**

* Objective: Componer el nombre del Contacto, como la unión del campo Nombre y el campo Apellido, y el campo de histórico.

* Description: Verifica que tenga un segundo nombre si es así lo incluye luego del primero sino se imprime una coma seguido del mismo

* @query_string None

* @post None

* @get None

* @author Darwin Escalona

* @called_by: rec.php y listarsup.php

* @calls: None

*/

//Función para verificar los campos modificados por medio de rec.php recibe como parámetros el campo a evaluar con y el nombre externo del mismo

//Si hubo cambios construya el histórico y anéxelo al existente

//Determinar si existe el segundo nombre para as dejar un espacio luego del primero, sino coloca la coma inmediatamente después del primero

//Obtener los campos del registro

//Determinar si existe el segundo nombre para as dejar un espacio luego del primero, sino coloca la coma inmediatamente después del primero y guardar los cambios

3. Varios

/**

```

* Program: Relacionados/borrarUnir.php
* Objective: Borrar los cargos y clasificaciones de un relacionado y marcarlo como borrado.
* Description: Verifica si se esta haciendo un borrado para inicializar las respectivas variables
* una vez inicializadas las variables descompone el listado de códigos recibidos del
* listarsup en un arreglo y el mismo se recorre para ir eliminando las clasificaciones
* y los cargos de cada relacionado seleccionado, este es marcado como borrando
* y por ultimo se almacena en el Histórico de Relacionado un resumen de la
* información borrada.
* @query_string
* @post $_POST[SelectedRec] listado con los códigos de los relacionados seleccionados
* en el listarsup para ser borrados.
* @post $_POST[Record] código del registro que hereda los hijos del o los demás registros
* a ser borrados (Importante esta funcionalidad no esta operativa se deja como
* sugerencia para futuras mejoras)
* @get $_GET[Relacionado] lista de los relacionados a ser unificados (Importante esta
* funcionalidad no esta operativa se deja como sugerencia para futuras mejoras)
* @get $_GET[Borrar] Variable que indica si se entro en modo borrar o unificar.
* @author Darwin Escalona
* @called_by: listarsup.php
* @calls: None
*/

```

//Función que realiza el borrado de forma ascendente de un relacionado de las tablas CLA y CRG y marcándolo en REL como Borrado

/* Función para unificar dos o más Relacionados en uno (Importante: No esta funcional)

// Verifica que se esta haciendo un borrado para inicializar las variables.

/******Recorrer el arreglo para hacer las inserciones por relacionados a la tabla ENT******/

/**

* Program: **Relacionados/GrabarSeguimiento.php**
* Objective: Validar si no hay un seguimiento con la dupla relacionado-entregable para proceder a crearla y al final mostrar un reporte de los seguimientos que tiene el entregable.
* Description: Se recibe un listado de códigos de relacionados seleccionados en el listarsup y el código del entregable para proceder a guardar los seguimientos si no existe ya. Y posteriormente se muestra un listarsup donde se muestran todos los seguimientos del entregable.
* @query_string \$SelectedRec: lista de códigos de los relacionados seleccionados en el listarsup para vincularlos al entregable.
* @post None
* @get \$_GET[Fecha]: Fecha se entrega a fijada por el usuario que este creando el seguimiento
* \$_GET[Entregable]: Código del entregable del que se esta creando el seguimiento.
* @author Darwin Escalona
* @called_by: listarsup.php del post de entregables PostENT.php.
* @calls: Listarsup.php
*/

//Recorrer el arreglo para hacer las inserciones por relacionados a la tabla ENT

//Ejecutar consulta para saber si ya existe la dupla entregable-relacionado

//Si la variable no tiene valor es porque no existe la dupla

//Insertar en la tabla el nuevo registro.

//Al finalizar ejecute listarsup

/**

* Program: **Relacionados/IngresoRelacionados.php**
* Objective: Se ingresan los datos básicos del relacionado que se desea incluir para validar que no
* exista ya algún registro con datos similares, al personal de mercadeo también les eliminar
* aquellas personas que se muestren en la lista eso para eliminar duplicidad en la data.
* Description: Formulario para ingresar de datos básicos, los mismos son enviados al programa
* Validar.php.
* @query_string None
* @post None
* @get None
* @author Darwin Escalona
* @called_by: Submenús de enlace Especiales A y B.
* @calls: Validar.php
*/

/**

* Program: **Relacionados/Validar.php**

* Objective: Comparar para obtener las posibles coincidencias con los relacionados registrados y cargar

* un listarsup con estas para poder seleccionar la que corresponda editar los datos del relacionado o bien agregar uno nuevo.

* Description: Se realiza la comparación para obtener las posibles coincidencias, se almacenan los códigos de las mismas para ser pasadas a un listarsup que las mostrara. Se pasan los valores a validar a través de la URL a rec.htm.

* @query_string None

* @post \$_POST[Nombre1]:Primer nombre

* @post \$_POST[Nombre2]:Segundo nombre

* @post \$_POST[Apellido1]:Primer Apellido

* @post \$_POST[Apellido2]:Segundo Apellido

* @post \$_POST[Email]: Email

* @post \$_POST[Fijo]: Teléfono Fijo

* @post \$_POST[Movil]: Teléfono móvil

* @post \$_POST[Borrar].

* @get None

* @author Darwin Escalona

* @called_by: IngresoRelacionados.htm

* @calls: Listarsup.php

*/

//Inicialización de las Variables Recibidas

//Si alguno de los valores viene en blanco inicializar a ABCDEFGHIJKLMN para que no coincida con los registros MySQL

//Carga de las condiciones para que no incluir las que estén vacías

//Recorrido de la base de datos para tomar todos los códigos resultantes de la consulta

//Llamado a listarsup para mostrar las coincidencias