Manual del Programador

Banco de Leche Humana

Uiversidad de El Salvador FMOcc

Diseño de Sistemas II

Indice

INTRODUCCIÓN

Los bancos de leche son un servicio especializado, para recolectar, almacenar y purificar la leche materna bajo un proceso de pasteurización; permitiendo así alimentar a los bebes prematuros o que nacen con bajo peso. El sistema BLH, es una solución para el control y manejo de los datos recolectados del Banco de Leche Materna del Hospital San Juan de Dios de la ciudad de Santa Ana, El Salvador.

El sistema BLH SYSTEM, fue desarrollado bajo el entorno de programación NetBeans usando el JDK 7, empleando como gestor de bases de datos (SQL SERVER 2008), a la vez auxiliado del SQL SERVER MANAGEMENT STUDIO y para la interfaz se usó Swing, apoyado de librerías como Koala Layout, Jcalendar, JXL API, entre otras

El propósito de este manual del programador, es dar a conocer al lector todos los listados del programa realizado. Para ello, se tratará de forma amena y concisa un repaso de todas las unidades, ejecutables entre otros, con el fin de que el usuario del conjunto pueda modificar a su gusto, algunos de los valores y parámetros de las funciones expuestas.

OBJETIVOS

Objetivo General

- Facilitar al programador la modificación del BLH SYSTEM, mediante un manual que describa el código contenido en el software.

Objetivo Específico

- Exponer al programador las herramientas necesarias para la optimización del software BLH SYSTEM.

- Describir de forma general el funcionamiento del software, permitiendo así el mantenimiento del BLH SYSTEM

JUSTIFICACIÓN

El siguiente manual de programador, se hace con el propósito de dar a conocer de manera general, el código del software BLH SYSTEM; esto con el fin de facilitar su modificación así como mejoras en el sistema; por lo que se pretende explicar de modo sencillo, el funcionamiento del software BLH SYSTEM.

GENERALIDADES

Las siguientes secciones, pretenden dar a conocer el funcionamiento de la interfaz del sistema BLH SYSTEM, así como de los errores, ausencias y futuras modificaciones y/o mejorías. Básicamente, la primera mejora que se requiere se presenta del campo estético; aunque funcional, la interfaz es visualmente satisfactoria.

EL PROGRAMA PRINCIPAL

Este conjugará el conjunto de los archivos expuestos anteriormente, y se compilará conteniendo todos ellos, como resultado obtendremos un archivo .jar, y este archivo será el ejecutable de nuestra aplicación.

DESCRIPCIÓN DE LA UNIDAD

La unidad aquí presentada, contiene las llamadas de aplicación al tratamiento de tipo de error, correspondiente a cualquier instrucción en tiempo de ejecución. Así el mismo contiene los procedimientos (y/o funciones) que se utilizan.

VARIABLES GLOBALES

Variables a nivel de clase.

VARIABLES LOCALES

Son las que se declaran y se utilizan en cada procedimiento y/o función.

REQUERIMIENTOS DEL SOFTWARE BLH SYSTEM

Requerimientos Hardware del sistema BLH System

- 512 MB de memoria RAM
- 800 MHZ de procesador
- 1 GB de espacio en disco duro
- 32 MB de video

Requerimiento software del sistema BLH System

- Sistema Operativo como plataforma (Windows, Linux)
- JDK v1.7
- SQL Server 2008
- Entorno de desarrollo (Netbeans o Eclipse para java)

Bibliotecas y complementos requeridos del sistema

- Koala layout
- JCalendar
- Xls (Librería para soportes de Excel)
- Jasper Report (Librería para soportes de reportes)
- SQL server 2008 sqljdbc4 (Librería para soporte de bases de datos con SQL Server 2008)

REQUISITOS ESPECÍFICOS

En este apartado se presentan los requisitos funcionales que deberán ser satisfechos por el sistema. Todos los requisitos aquí expuestos son ESENCIALES. Estos requisitos se han especificado teniendo en cuenta, entre otros, el criterio de testabilidad, dado un requisito, debería ser fácilmente demostrable si es satisfecho o no por el sistema.

Requisitos funcionales

Acceso a biblioteca

La interfaz de usuario podrá acceder a toda la biblioteca de un cliente.

Búsquedas sobre la biblioteca

La interfaz de usuario permitirá realizar búsquedas de descripciones, para lo cual se pueden especificar total o parcialmente las descripciones a buscar.

Refinamiento

La interfaz de usuario, tras efectuar una búsqueda y mostrar los resultados al usuario, permitirá refinar la búsqueda, es decir, reescribirla a partir de la actual haciéndola más restrictiva.

Acceso a recursos

Los resultados de la búsqueda incluirán mecanismos que permitirán al usuario solicitar la realización de una copia local del recurso referenciado por la descripción.

Interfaces de usuario

Requisito general de la interfaz de usuario

La interfaz de usuario diseñada deberá tener alto grado de usabilidad y ser sencilla, cumpliendo todos los requisitos especificados.

Portabilidad

El sistema está diseñado para llevar a diferentes plataformas, ya que el lenguaje de programación java es multiplataforma el sistema podrá ser llevado a muchos sistemas operativos.

ESTRUCTURA Y DISEÑO DEL SOFTWARE

- BLH
    - Paquetes de Fuentes
        - Instancia
            - InstanceOf.java
            - InstanceOfConsult.java
            - InstanceOfDonacion.java
            - InstanceOfGeneralidades.java
            - InstanceOfTransport.java
        - Operaciones
            - OperacionesExcel.java
            - OperacionesInforme.java
            - OperacionesReporte.java
        - Acceso Datos
            - Conexión.java
            - Operaciones.java
            - OperacionesConsulta.java
            - OperacionesDonacion.java
            - OperacionesDonantNiños.java

- OperacionesUsuario.java

- Clases
  - Calculos.java
  - Consulta.java
  - Control.java
  - Donacion.java
  - Donante.java
  - Encargado.java
  - Informe.java
  - InformeProduccion.java
  - Medico.java
  - Motorista.java
  - Paciente.java
  - PropiedadPaciente.java
  - PropiedadesPaciente.java
  - Tabla.java
  - Usuario.java
  - Vehiculo.java
  - VisitaDonacion.java
- Imágenes
  - Imágenes usadas en el sistema
- Imágenes fondo
  - Imágenes usadas en el sistema
- Reportes
  - Actividades.jasper
  - Actividades.jrxml
  - ConsultaPaciente.jasper
  - ConsultaPaciente.jrxml
  - InfoLab.jasper
  - InfoLab.jrxml
  - InfoProd.jasper
  - InfoProd.jrxml
  - ListaMedicos.jasper
  - ListaMedicos.jrxml
  - ListaMotorista.jasper
  - ListaMotorista.jrxml
- Vistas
  - AcercaDe.java
  - AdminPassword.java
  - AdministrarEncargado.java
  - AdministrarMedic.java
  - AdministrarMotorista.java
  - AdministrarUsuarios.java
  - AdministrarVehiculos.java
  - BuscarNino.java

- Consult.java
- ControlPacientes.java
- InfoConsulta.java
- InformeLaboratorista.java
- InformeProduccion.java
- ListaPacientes.java
- Logueo.java
- NuevaDonacion.java
- OtrosDatos.java
- RecoleccionDomiciliar.java
- RecoleccionLeche.java
- RutaRecDomiciliar.java
- SeleccionActividad.java
- NuevoUsuario.java

- Bibliotecas
  - Driver SQL JDBC sqljdbc4
  - Excel jxl
  - Jasper Reports
  - Jcalendar
  - Koala Layout
  - JDK 1.7

VARIABLES LOCALES, GLOBALES Y METODOS

Al correr el sistema por primera vez, se mostrara el formulario de identificación de usuarios, pero para entender esto se necesita conocer el funcionamiento de los métodos de acceso a datos y de operaciones.

- Paquete de fuentes
  - Instancia
    - Instanceof.java

```
public AdminPassword administrador = new AdminPassword();
public SeleccionActividad actividad = new SeleccionActividad();
public RecoleccionLeche donante = new RecoleccionLeche();
public RutaRecDomiciliar rutRecDom = new RutaRecDomiciliar();
```

Esta clase contiene estas variables globales que simplemente hacen referencia a los formularios construidos, esto ayuda a reducir la construcción de formularios, dentro de otros formularios.

- InstanceOfConsult.java

```
public ControlPaciente pte = new ControlPaciente();
public Consult consulta = new Consult();
public BuscarNino buscar = new BuscarNino();
public RecoleccionLeche recL = new RecoleccionLeche();
public AdministrarMedic Admed = new AdministrarMedic();
public AdministrarEncargado encargado = new AdministrarEncargado();
```

Esta clase contiene estas variables globales que simplemente hacen referencia a los formularios construidos, esto ayuda a reducir la construcción de formularios, dentro de otros formularios, pero en este caso se trabaja en el área de consultas.

- InstanceOfDonacion.java

```
public NuevaDonacion newDonacion = new NuevaDonacion();
```

Esta clase contiene esta variable global que simplemente hace referencia al formulario construido, esto ayuda a reducir la construcción de este formulario, dentro de otros formularios, pero en este caso se trabaja en el área de donaciones.

- InstanceOfGeneralidades.java

```
public AcercaDe acercaDe = new AcercaDe();
public nuevoUsuario usuario = new nuevoUsuario();
public AdministrarUsuarios admonUser = new AdministrarUsuarios();
public AdministrarMotoristas admMotoristas = new AdministrarMotoristas();
public AdministrarVehiculos admVehiculos = new AdministrarVehiculos();
```

Esta clase contiene estas variables globales que simplemente hacen referencia a los formularios construidos, esto ayuda a reducir la construcción de formularios, dentro de otros formularios.

- InstanceOfTransport.java

```
public RecoleccionDomiciliar transporte = new RecoleccionDomiciliar();
public RutaRecDomiciliar ruta = new RutaRecDomiciliar();
public AdministrarMotoristas adminMoto = new AdministrarMotoristas();
```

Esta clase contiene estas variables globales que simplemente hacen referencia a los formularios construidos, esto ayuda a reducir la construcción de formularios, dentro de otros formularios, pero en este caso se trabaja en el área de transporte.

- Operaciones
  - OperacionesExcel.java

Variables globales:

Conexion con = new Conexion();

Esta variable sirve para conectarnos a la base de datos.

Métodos:

- public void crearExcel(String ruta) throws IOException, WriteException.

Este método recibe una variable local, la cual especifica la ruta donde será almacenado el archivo doc.xls, además arroja las excepciones IOException y WriteException.

Funciona de la siguiente manera

```
Mientras (tuplas . next())
        Crear celdas con info de donantes
```

- public String obtenerRutaExcel()

Este método obtiene la ruta de un archivo.xls

- public void exportarExcelLab(String ruta, JTable tablita) throws IOException, WriteException

Este método recibe dos variables locales, una específica la ruta donde será almacenado el archivo doc.xls, y la otra la tabla con la cual será construido el archivo además arroja las excepciones IOException y WriteException, todo esto para el informe del laboratorista.

- public void exportarExcelProd(String ruta, JTable tablita) throws IOException, WriteException

Este método recibe dos variables locales, una específica la ruta donde será almacenado el archivo doc.xls, y la otra la tabla con la cual será construido el archivo además arroja las excepciones IOException y WriteException, todo esto para el informe de producción.

- public void exportarExcelCons(String ruta, JTable tablita)

Este método recibe dos variables locales, una específica la ruta donde será almacenado el archivo doc.xls, y la otra la tabla con la cual será construido el archivo además arroja las excepciones IOException y WriteException, todo esto para el informe de consultas de pacientes.

- OperacionesInforme.java

Variables Globales:

Conexion con = new Conexion();

Esta variable sirve para conectarnos a la base de datos.

Métodos:

- public void almacenarInforme(Informe info)

Este método recibe un objeto informe, el cual es sirve para almacenar los datos del laboratorista, en la base de datos, retorna void.

- public void eliminarInforme(Informe info)

Este método recibe un objeto informe, el cual es sirve para eliminar los datos del laboratorista, retorna void.

- public void listarInformeLab(JTable tablita)

Este método recibe un objeto jtable en la cual se listará el informe del laboratorista, retorna void

- public void listarInformeProd(JTable tablita)

Este método recibe un objeto jtable en la cual se listará el informe de producción, retorna void

- public String obtenerRutaExcel()

Este método obtiene la ruta de un archivo.xls

- public void almacenarInformeProd(InformeProduccion info)

Este método recibe un objeto informeProduccion, el cual es sirve para almacenar los datos de la producción en la base de datos, retorna void.

- public void eliminarInformeProd(InformeProduccion info)

Este método recibe un objeto informeProduccion, el cual es sirve para eliminar los datos de la producción en la base de datos, retorna void.

- OperacionesReporte.java

Variables Globales:

- Conexion con = new Conexion();

Esta variable sirve para conectarnos a la base de datos.

- OperacionesDonantNinos opDN = new OperacionesDonantNinos();

Se construye una instancia de la clase operaciones Donantes

Métodos:

- public void runReporte(String nombreReporte, String imagen)

Este método recibe 2 variables locales String, una es el nombre el reporte y la otra es la imagen que va a contener, este método se encarga de correr los reportes generados, retorna void.

- public void runReportePaciente(String nombreReporte, String imagen, String car)

Este método recibe 3 variables locales String, una es el nombre el reporte, otra es la imagen que va a contener, y la otra es el carnet del paciente, este método se encarga de correr los reportes generados de los pacientes, en base al carnet, retorna void.

- public String getRuta()

Este método obtiene la ruta actual de un archivo, retorna String

- Acceso Datos
  - Conexión

Variables Globales:

- Connection conexion = null;
- int puerto1 = 49205;

Métodos:

- public Connection conectar()

Este método retorna un objeto Connection, nos permite conectarnos a la base de datos del sistema usando un puerto mediante el driver para slq server 2008

- public void desconectar()

Este método cierra la conexión existente

  - Operaciones

Variables Globales:

- Conexion con = new Conexion();

Permite conectarse a la base de datos

Métodos:

- public void almacenarMedico(Medico doc)

Este método recibe un objeto medico, el cual es almacenado en la base de datos

- public void eliminarMedico(Medico doc)

Este método recibe un objeto medico, el cual es eliminado en la base de datos

- public void listarMedicos(JTable tabla)

Este método recibe un objeto jtable en la cual son listados los médicos existentes retorna void

Funciona de la siguiente manera:

Mientras (si encuentre medico)
Llenar tabla con medicos
Cerrar conexión de base de datos

- public void almacenarMotorista(Motorista moto)

Este método recibe un objeto Motorista, el cual es almacenado en la base de datos

- public void eliminarMotorista(Motorista moto)

Este método recibe un objeto Motorista, el cual es eliminado de la base de datos

- public void listarMotoristas(JTable tabla)

Este método recibe un objeto jtable en la cual son listados los motoristas existentes retorna void

Funciona de la siguiente manera:

Mientras (si encuentre motorista)
    Llenar tabla con motoristas
Cerrar conexión de base de datos

- public void almacenarVehiculo(Vehiculo veh)

Este método recibe un objeto Vehículo, el cual es almacenado en la base de datos

- public void eliminarVehiculo(Vehiculo veh)

Este método recibe un objeto Vehículo, el cual es eliminado de la base de datos

- public void listarVehiculo(JTable tabla)

Este método recibe un objeto jtable en la cual son listados los vehículos existentes retorna void

Funciona de la siguiente manera:

Mientras (se encuentre vehículos)
    Llenar tabla con vehículos
Cerrar conexión de base de datos

- public void almacenarRuta(VisitaDonacion visita, DefaultTableModel trayectos)

Este método recibe 2 objetos, uno Visita Donación y otro default table model, se encarga de almacenar una nueva ruta, retorna void

- public void listarRutas(JTable table)

Este método recibe un objeto jtable en la cual son listadas las rutas existentes retorna void

Funciona de la siguiente manera:

Mientras (si encuentra rutas)
Llenar tabla con rutas
Cerrar conexión de base de datos

- public void almacenarEncargado(Encargado encargado)

Este método recibe un objeto Encargado, el cual es almacenado en la base de datos

- public void listarEncargados(JTable tabla)

Este método recibe un objeto jtable en la cual son listados los encargados existentes retorna void

Funciona de la siguiente manera:
Mientras (si encuentra encargados)
Llenar tabla con encargados
Cerrar conexión de base de datos

- public void eliminarEncargado(Encargado encargado)

Este método recibe un objeto Encargado, el cual es eliminado de la base de datos

- public void listarPacientesFiltrados(JTable tabla, String campo, String value)

Este método recibe 3 variables locales, y sirve para buscar pacientes en la base de datos, los cuales son cargados en el parámetro tabla

- public void eliminarRuta(int id)

Este metodo recibe como parametro el id de la ruta, y en base a éste elimina la ruta de la base de datos

- public void modificarRuta(int idRuta, JTable tabla, JTextField medico, JTextField motorista, JTextField vehiculo, JCalendarCombo calendar)

Este método recibe 6 variables locales, y permite modificar rutas en base al id de la ruta

- public void modificarRutas(JTable tabla)

Este método recibe como parámetro un jtable, en el cual se listan las rutas modificadas

- public void eliminarRutas(int id)

Este metodo elimina una ruta de la base de datos, usando el id de ésta

- public void BuscarRutas(JTable table, String campo, String value) {

Este método busca las rutas en base al value en el campo respective y las carga en el objeto jtable

- OperacionesConsulta

Variables Globales:

- Conexion con = new Conexion();

Métodos:

- public void almacenarConsulta(Consulta consulta)

Este método recibe un objeto consulta, el cual es almacenado en la base de datos

- public void listarConsulta(JTable tabla, String carnet)

Este metodo recibe dos parametros, extrae las consultas de los pacientes en base al carnet de estos, para luego listarlos en la tabla.

- public void eliminarPropiedadPaciente(String carnet)

Este método recibe un parametro String, para eliminar las propiedades de un paciente de la base de datos en base a su carnet.

- public void eliminarConsultas(String carnet)

Este método elimina las consultas de la base de datos del paciente en base a su carnet

- public void eliminarConsulta(String carnet, String fecha, String medico, String peso, String longitud, String pc)

  Este método elimina las consultas de la base de datos del paciente en base a su carnet, a la fecha, al médico, al peso, a la longitud, y al perímetro cefálico de éste.

- public void listarPacientes(JTable tabla)

Este método recibe un objeto jtable en la cual son listados los pacientes existentes retorna void

Funciona de la siguiente manera:

Mientras (si encuentra pacientes)
Llenar tabla con pacientes
Cerrar conexión de base de datos

- public DefaultTableModel buscarNombres(String nombre, String apellido)

Este método busca los pacientes en base al nombre y apellido de éstos, retorna un default table model

- public DefaultTableModel buscarNombre(String campo, String dato)

Este método busca los pacientes en base al dato y campo de la base de datos, retorna un default table model

- public DefaultTableModel buscarFecha(Date de, Date hasta)

Este método busca los pacientes en base a un rango de su fecha de nacimientos en la base de datos, retorna un default table model

- public void BuscarConsulta(JTable tabla, String campo, String value, String carnet)

Este método busca las consultas en la base de datos de los pacientes en base al carnet de estos, para luego listarlos en el objeto jtable

- public void almacenarPropiedad(PropiedadPaciente prop, String carnet)

Este método recibe un objeto PropiedadPaciente y el carnet del paciente, el cual es almacenado en la base de datos.

- public PropiedadPaciente extraerPropiedad(String carnet)

Este método estrae las propiedades de un paciente en base a su carnet, retorna un objeto propiedad paciente.

- public boolean verificarPaciente(String carnet)

Este método verifica un paciente en la base de datos en base a su carnet, retorna true si el paciente existe, false si el paciente no existe

- public void actualizarPropiedades(String carnet, PropiedadPaciente p)

Este método actualiza las propiedades del paciente en base al carnet de este, recibe como parámetros la propiedad paciente a modificar y el carnet del paciente

- public void actualizarConsulta(Consulta c, Consulta mod)

Este método actualiza la consulta del paciente, recibe como parámetro dos objetos consulta

- public String getNombrePaciente(String nit)

Este método retorna un string con el nombre del paciente, este es extraído en base al carnet de éste.

- public void BuscarEncargado(JTable tabla, String campo, String value)

Este método busca los encargados en base al value en el campo de la base de datos de la tabla encargado, para luego listarlos en el objeto jtable

- Operaciones Donaciones

Variables globales:

- Conexion con = new Conexion();

Nos permite conectarnos a la base de datos

- DefaultTableModel tableModel = new DefaultTableModel()

Metodos:

- public void listarDonantes(JTable tabla)

Este método recibe un objeto table en el que lista los donantes, retorna void

Funciona de la siguiente manera:

Mientras (si encuentra donantes)
Llenar tabla con donantes
Cerrar conexión de base de datos

- public Donante listarInfoDonante(String documento)

Este método retorna un donante y recibe como variable local un documento, con el cual se hace referencia al donante que se quiere.

- public void almacenarInfoDonante(Donante donante)

Este método recibe un objeto donante, el cual es almacenado en la base de datos.

- public void eliminarDonante(String doc)

Este método recibe una variable string, con la cual se elimina un donante de la base de datos.

- public void almacenarDonacion(Donacion donacion, String idRutaDom)

Este método recibe como parámetro un objeto donación y una variable local string para almacenar una donación en la base de datos, retorna un void

- public void listarDonanciones(JTable tabla, String doc)

Este método recibe como parámetro un objeto jtable y una variable local string, y lista las donaciones en la tabla en base a la variable string

- public void eliminarDonacion(String doc, String fechD)

Este método recibe como parámetro 2 variables string para eliminar una donación en la base de datos, retorna un void

- public void buscarDonante(JTable tabla, String condicion, String aBuscar)

Este método busca los donantes en base a la condición en el campo aBuscar y las carga en el objeto jtable

- OperacionesDonantNinos

Variables Globales:

- Conexion con = new Conexion();

Nos permite conectarnos a la base de datos

Métodos:

- public void listarNinos(JTable tabla)

Este método recibe como parámetro un objeto jtable y basicamente lista los niños en la lista, retorna void.

- public Paciente listarInfoNino(String carnet)

Este método recibe como parámetro una variable string y básicamente lista la información de los niños, retorna un paciente.

- public PropiedadesPaciente listarPropeNino(String carnet)

Este método recibe como parámetro una variable string y básicamente lista las propiedades de los niños, retorna un Propiedadpaciente.

- public String EncargadoPaciente(String carnet)

Este método recibe como parámetro una variable string y básicamente lista el encargado del niño, retorna un Encargado

- public void almacenarInfoNino(Paciente paciente, String encar, PropiedadesPaciente propiedades).

Este método recibe como parámetro un objeto paciente, un objeto propiedadPaciente y una variable local string para almacenar la información de un niño en la base de datos, retorna un void

- public void eliminarNino(String carnet)

Este método recibe como parámetro una variable local string para eliminar la información de un niño en la base de datos, retorna un void

- public String obtenerCarnetPaciente(String valor)

Este método recibe como parámetro una variable local string y en base a ésta obtienes el carnet del paciente, retorna String

- public void actualizarPaciente(Paciente p, PropiedadesPaciente pro)

Este método recibe como parámetro un objeto paciente y un objeto propiedadPaciente y simplemente acualiza la información de los pacientes, retorna void

- Operaciones Usuarios

Variables Globales:

- Conexion con = new Conexion();

Nos permite conectarnos a la base de datos

Métodos:

- public void ActualizarUsuarios(String newUser, String pass, String viejoUs, String viejoPass)

Este método recibe 4 variables locales string, y basicamente actualiza los usuarios con sus contraseñas.

- public Boolean isLogged(String user, String pass)

Este método recibe 2 variables locales y permite verificar si un usuario esta logueado retorna true de ser verdadero y false de ser falso

- public Boolean isSaved(String user, String pass)

Este método recibe 2 variables locales y permite verificar si un usuario existe retorna true de ser verdadero y false de ser falso

- public void eliminarUsuario(String Id)

Este método recibe una variable local String, con la cual eliminar un usuario de la base de datos

- public void listarUsuarios(JTable tabla)

Este método recibe un objeto table en el que lista los usuarios, retorna void

Funciona de la siguiente manera:

Mientras (si encuentra donantes)
Llenar tabla con donantes
Cerrar conexión de base de datos

- public void nuevoUsuario(String username, String password, Boolean isAdm)

Este método recibe 3 variables locales, con las cuales crea un nuevo usuario en la base de datos, retorna un void

- public void verificar()

Este metodo permite verificar los usuarios existentes en la base de datos

- public void setEstado(String usuario, String estado)

Este método recibe 2 variables locales, con las cuales se verifican las operaciones que el usuario a realizado, marcadas por una hora determinada, retorna void

- public String usuarioUp()

Verifica si un usuario esta de alta actualmente

- public Boolean isAdmin()

Verifica si un usuario es administrador o no, retorna true de ser verdadero y false de ser falso

- Clases
  - Calculos

Variables globales:

- DefaultComboBoxModel model = null;

Construye un modelo de combobox ;

Métodos:

- public void Municipios(int depto, JComboBox combo)

Este método recibe dos variables locales, nos sirve para listar los municipios en una lista desplegable.

Función:

```
Switch(depto){

    Case n:
```

Llenar lista desplegable

Break;

- public String calcularDias(java.sql.Date fecha)

Este método recibe una variable local Date, y retorna una String, simplemente calcula la edad del paciente.

- Consulta

Variables globales:

```
private Date fecha;
private String diagnostico;
private double racionAsignada;
private int JVPM;
private String carnet;
private double peso;
private double estatura;
private String perimetroCefalico;
private String  condicionSalud;
```

Éstas variables nos ayudan a identificar una consulta

Constructores

Implícito
- public Consulta()

Inicializa todas las variables a un valor por defecto

Explícito
- public Consulta(Date fecha, String Carnet, double peso, double estatura, String perCefalico, int jpvm, String condicionSalud,double racionAsignada, String Diagnostico)

Inicializa las variables con el valor contenido en los parámetros

Métodos
- getters()

Extrae el valor de cualquier variable en un momento determinado

- setters()

Cambia el valor de cualquier variable en un momento determinado

- Control

Variables globales:

```
private String fechaHora;
   private Double estatura;
   private Double peso;
   private Double perCefalico;
```

Éstas variables nos ayudan a identificar el control

Constructores

Implícito
- public Control()

Inicializa todas las variables a un valor por defecto

Explícito
- public Control(String fechHora, Double estatura, Double peso, Double perCefalico)

Inicializa las variables con el valor contenido en los parámetros

Métodos
- getters()

Extrae el valor de cualquier variable en un momento determinado

- setters()

Cambia el valor de cualquier variable en un momento determinado

- Donación

Variables globales:

private String fechaDon;
private String tipo;
private String racAceptada;
private String racDescartada;
private String documento
private String estatura;
private String peso;

Éstas variables nos ayudan a identificar una donación

Constructores

Implícito

- public Donación()

Inicializa todas las variables a un valor por defecto

Explícito

- public Donacion(String fechaDon, String tipo, String racAcep, String racDesc, String Documento, String estatura, String peso)

Inicializa las variables con el valor contenido en los parámetros

Métodos

- getters()

Extrae el valor de cualquier variable en un momento determinado

- setters()

Cambia el valor de cualquier variable en un momento determinado

- Donante

Variables globales:

private String documento;
private String nombre;
private String apellido;
private String fechaNac;
private String direccion;
private String peso;
private String estatura;
private String tel;
private String semGest;
private String fechaParto;
private String VDRL;

private String HbsAg;
private String HIV;
private String transSangui;
private String tabaquismo;
private String etilismo;
private String aptaDonar;
private String fechaObtencion;

Éstas variables nos ayudan a identificar un donante

Constructores

Implícito

- public Donante()

Inicializa todas las variables a un valor por defecto

Explícito

- public Donante(String documento, String nombre, String apellido, String fechaNac, String direccion, String peso, String estatura, String tel, String semGest, String fechaParto, String vdrl, String hbsag, String hiv, String transSan, String tabaq, String etil, String apta, String fechaObt)

Inicializa las variables con el valor contenido en los parámetros

Métodos

- getters()

Extrae el valor de cualquier variable en un momento determinado

- setters()

Cambia el valor de cualquier variable en un momento determinado

- Encargado

Variables globales:

private String nit;
private String dui;
private String nombre;
private String apellido;
private String parentesco;
private String direccion;
private String telefono;

Éstas variables nos ayudan a identificar un Encargado

Constructores

Implícito
- Public Encargado()

Inicializa todas las variables a un valor por defecto

Explícito
- public Encargado(String nit, String dui, String nombre, String apellido, String parentesco, String direccion, String tel)

Inicializa las variables con el valor contenido en los parámetros

Métodos
- getters()

Extrae el valor de cualquier variable en un momento determinado

- setters()

Cambia el valor de cualquier variable en un momento determinado

- Informe

Variables globales:

```
private String fecha;
private int muestrasA;
private int preN;
private double prePor;
private int ausN;
private double ausPor;
```

Éstas variables nos ayudan a identificar un Informe

Constructores

Implícito
- Public Informe()

Inicializa todas las variables a un valor por defecto

Explícito
- public Informe(String fecha, int muestrasA, int preN, double prePor, int ausN, double ausPor)

Inicializa las variables con el valor contenido en los parámetros

Métodos
- getters()

Extrae el valor de cualquier variable en un momento determinado

- setters()

Cambia el valor de cualquier variable en un momento determinado

- InformeProducción

Variables globales:

```
private String IdProduccion;
private int AtenIndivisual;
private int AtenGrupo;
private int VisDom;
private int TotalAten;
private double Recolectado;
private double Distribuido;
private int DonantesP;
private int ReceptoresP;
private double Microbiologia;
private double FisQui;
private double Crematocrito;
private double AcidezDornic;
private double TotAnaFQ;
private double TotGenAna;
```

Éstas variables nos ayudan a identificar un InformeProducción

Constructores

Implícito

- Public InformeProduccion()

Inicializa todas las variables a un valor por defecto

Explícito

- public InformeProduccion(String IdProduccion, int AtenIndividual, int AtenGrupo, int VisDom, int TotalAten, double Recolectado, double Distribuido, int DonantesP, int ReceptoresP, double Microbriologia, double FisQui, double Crematocrito, double AcidezDornic, double TotAnaFQ, double TotGenAna)

Inicializa las variables con el valor contenido en los parámetros

Métodos

- getters()

Extrae el valor de cualquier variable en un momento determinado

- setters()

Cambia el valor de cualquier variable en un momento determinado

- Médico

Variables globales:

```
private int jvpm;
private String nombre;
```

private String apellido;
private String especialidad;

Éstas variables nos ayudan a identificar un medico

Constructores

Implícito
- Public Medico()

Inicializa todas las variables a un valor por defecto

Explícito
- public Medico(int jvpm, String nombre, String apellido, String especialidad)

Inicializa las variables con el valor contenido en los parámetros

Métodos
- getters()

Extrae el valor de cualquier variable en un momento determinado

- setters()

Cambia el valor de cualquier variable en un momento determinado

- Motorista

Variables globales:

private String nombre;
private String apellido;
private String dui;
Éstas variables nos ayudan a identificar un motorista

Constructores

Implícito
- Public Motorista()

Inicializa todas las variables a un valor por defecto

Explícito
- public Motorista(String nom, String ape, String d)

Inicializa las variables con el valor contenido en los parámetros

Métodos
- getters()

Extrae el valor de cualquier variable en un momento determinado

- setters()

Cambia el valor de cualquier variable en un momento determinado

- Paciente

Variables globales:

private String carnet;
private String nombre;
private String apellido;
private Date fechaNac;
private String sexo;
private String semGest;
private String notas;
private String tipo;

Éstas variables nos ayudan a identificar un paciente

Constructores

Implícito
- Public Paciente()

Inicializa todas las variables a un valor por defecto

Explícito
- public Paciente(String carnet, String nombre, String apellido, Date fechaNac, String sexo, String semGest, String notas, String tipo)

Inicializa las variables con el valor contenido en los parámetros

Métodos
- getters()

Extrae el valor de cualquier variable en un momento determinado

- setters()

Cambia el valor de cualquier variable en un momento determinado

- PropiedadPaciente

Variables globales:

private Date FechaIncorporacion;
private double pesoNacer;
private double pesoIncorporarse;
private double estaturaNacer;
private double estaturaIncorporarse;
private double pCefalicoNacer;
private double pCefalicoIncorporarse;

Éstas variables nos ayudan a identificar una propiedad paciente

Constructores

Implícito

- Public PropiedadPaciente()

Inicializa todas las variables a un valor por defecto

Explícito

- public PropiedadPaciente(Date fecha, double pesoNac, double pesoInc, double estNac, double estInc, double pCefNac, double pCefInc)

Inicializa las variables con el valor contenido en los parámetros

Métodos

- getters()

Extrae el valor de cualquier variable en un momento determinado

- setters()

Cambia el valor de cualquier variable en un momento determinado

- PropiedadesPaciente

Variables globales:

private String parto;
private String edadMaterna;
private String gravidez;
private String patologia;
private String municipio;
private String departamento;

Éstas variables nos ayudan a identificar una propiedad paciente

Constructores

Implícito

- Public PropiedadesPaciente()

Inicializa todas las variables a un valor por defecto

Explícito

- public PropiedadesPaciente(String parto, String edad, String gravidez, String patologia, String municipio, String departament)

Inicializa las variables con el valor contenido en los parámetros

Métodos

- getters()

Extrae el valor de cualquier variable en un momento determinado

- setters()

Cambia el valor de cualquier variable en un momento determinado

- Tabla

Variables Globales:

- private Conexion con = new Conexion();

Nos permite conectarnos a la base de datos

Métodos

- public DefaultTableModel listarPacientes()

Este metodo retorna un defaulttablemodel y basicamente lista los pacientes

- Usuario

Variables globales:

private String usuario;
private String password;
private boolean admin;
Éstas variables nos ayudan a identificar una Usuario

Constructores

Implícito
- Public usuario()

Inicializa todas las variables a un valor por defecto

Explícito
- public Usuario(String name, String pass, boolean admin)

Inicializa las variables con el valor contenido en los parámetros

Métodos
- getters()

Extrae el valor de cualquier variable en un momento determinado

- setters()

Cambia el valor de cualquier variable en un momento determinado

- Vehiculos

Variables globales:

private String placa;
private String tipo;
Éstas variables nos ayudan a identificar un Vehiculo

Constructores

Implícito
- Public Vehiculos()

Inicializa todas las variables a un valor por defecto

Explícito
- public Vehiculo(String pl, String tp)

Inicializa las variables con el valor contenido en los parámetros

Métodos
- getters()

Extrae el valor de cualquier variable en un momento determinado

- setters()

Cambia el valor de cualquier variable en un momento determinado

- VisitaDonación

Variables globales:

private Date fecha;
private String hora;
private String kilometraje;
private String LugarSalida;
private String LugarLlegada;
private String distancia;
private String galones;
private String jvpm;
private String IdVehiculo;
private String DuiMotorista;

Éstas variables nos ayudan a identificar una Visita Donación

Constructores

Implícito
- Public VisitaDOnacion()

Inicializa todas las variables a un valor por defecto

Explícito
- public VisitaDonacion(String dist, String LSalida, String Lllegada, String km, Date fecha, String hora, String gal, String jvpm, String idVehiculo, String duiMot)

Inicializa las variables con el valor contenido en los parámetros

Métodos
- getters()

Extrae el valor de cualquier variable en un momento determinado

- setters()

Cambia el valor de cualquier variable en un momento determinado

- Imágenes

Se muestran las imágenes que se utilizan en el sistema

- Imágenes fondos

Se muestran las imágenes que se utilizan en el sistema

- Reportes
  - Actividades.jrxml

Este reporte nos muestra las actividades realizadas por los usuarios realizados
Hace lo siguiente

- SELECT * FROM actividades WHERE usuario = nombre

Muestra los campos (id, fecha, hora, usuario, actividad)

  - Control pacientes

Este reporte nos muestra las consultas de los pacientes
Hace lo siguiente:

- SELECT * FROM consulta WHERE idPa ciente = idpaciente

Muestra los campos (carnet, nombre, apellidos, sexo, fecha de nacimiento, jvpm, fecha, peso, estatura, perimetro cefalico, condicion de salud)

  - InfoLab

Este reporte nos muestra la información de laboratorio
Hace lo siguiente:

- SELECT * FROM infoLab WHERE idinfoLab = infolab

Muestra los campos (fecha, muetras realizadas, presencia, ausencia)

  - ListarMedicos

Este reporte nos muestra la información de los medicos
Hace lo siguiente:

- SELECT * FROM medico WHERE idMedico = idmedico

Muestra los campos (jvpm, nombres, apellidos, especialidad)

  - ListarMotoristas

Este reporte nos muestra la información de los motoristas
Hace lo siguiente:

- SELECT * FROM Motoristas WHERE idmotorista = idmotorista

Muestra los campos (id, nombres, apellidos, dui)

- Vistas

- Acercade.java

Constructor

- public AcercaDe()

Construye los componentes swing del formulario
Main

- public void run()

Ejecuta el constructor para correr el formulario

- AdminPassword

Variables Globales:
OperacionesUsuarios op = new OperacionesUsuarios();
String usuario;
String Contraseña;
boolean admin = true;

Constructor:

- public AdminPassword()

Construye los componentes swing del formulario
Métodos:

- public void limpiarGUI()

Limpia los campos de la interfaz

- public boolean verificarDatos()

Comprueba si los campos contienen valores, retorna true de ser verdadero, false de lo contrario

- private void jpfRepitaContraseñaFocusLost(java.awt.event.FocusEvent evt)

Verifica si el campo repita contraseña coincide con el campo contraseña

- private void jbLimpiarActionPerformed(java.awt.event.ActionEvent evt)

Limpia los campos del formulario

- private void jbAceptarActionPerformed(java.awt.event.ActionEvent evt)

Modifica los valores de los usuarios en la base de datos

- private void jbSalirActionPerformed(java.awt.event.ActionEvent evt)

Cierra el formulario

Main

- public void run()

Ejecuta el constructor para correr el formulario

- Administrar encargado

Variables Globales:
Operaciones op = new Operaciones();

ControlPaciente ptes = new ControlPaciente();
OperacionesConsulta opc = new OperacionesConsulta();
OperacionesUsuarios opU = new OperacionesUsuarios();
String usu = "";
Constructor:

- public AdministrarEncargado()

Construye los componentes swing del formulario
Métodos:

- private void jbNuevoEncargadoActionPerformed(java.awt.event.ActionEvent evt)

Crea una nueva fila en la tabla encargado

- private void jbGuardarEncargadoActionPerformed(java.awt.event.ActionEvent evt)

Guarda los encargados en la base de datos

- private void jbAgregarActionPerformed(java.awt.event.ActionEvent evt)

Agrega un encargado al lugar de destino

- private void jbEliminarEncargadoActionPerformed(java.awt.event.ActionEvent evt)

Elimina el encargado seleccionado de la base de datos, al mismo tiempo refresca la tabla

- private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt)

Cierra el formulario

- private void formWindowClosed(java.awt.event.WindowEvent evt)

Cierra el formulario

- private void jbBuscarActionPerformed(java.awt.event.ActionEvent evt)

Busca los encargados en la base de datos, en base a un parámetro

- private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt)

retorna al formulario selección actividad
Main

- public void run()

Ejecuta el constructor para correr el formulario

- AdministrarMedic

Variables Globales:
Operaciones op = new Operaciones();
OperacionesReportes opr = new OperacionesReportes();
OperacionesUsuarios opU = new OperacionesUsuarios();
String usu = "";
public int filaAnterior = 0;
RutaRecDomiciliar recoleccion = new RutaRecDomiciliar();

static public String carnet = "";
static public String fechaNac = "";
static public double peso = 0.0;
static public double estatura = 0.0;
static public double pCefalico = 0.0;
static public String notas = "";
static public String conSalud = "";
static public String racion = "";

Constructor:

- public AdministrarMedic()

Construye los componentes swing del formulario

Métodos:

- private void jbAgregarMedicoActionPerformed(java.awt.event.ActionEvent evt)

Crea una nueva fila en la tabla medico

- private void jbGuardarActionPerformed(java.awt.event.ActionEvent evt)

Guarda los médicos en la base de datos

- private void jbAgregarActionPerformed(java.awt.event.ActionEvent evt)

Agrega un medico al lugar de destino

- public void comoSalir()

Cierra el formulario, y continúa en ejecución los formularios que estaban corriendo

- public void comoSalir2()

Cierra el formulario, y continúa en ejecución los formularios que estaban corriendo

- private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt)

Cierra el formulario

- private void formWindowClosed(java.awt.event.WindowEvent evt)

Cierra el formulario

- private void jbEliminarActionPerformed(java.awt.event.ActionEvent evt)

Elimina un médico de la base de datos, y actualiza la tabla de médicos

- private void jmiImprimirActionPerformed(java.awt.event.ActionEvent evt)

Imprime los médicos contenidos en la tabla medicos

- private void jtMedicosKeyPressed(java.awt.event.KeyEvent evt)
- private void jtMedicosKeyReleased(java.awt.event.KeyEvent evt)

Almacena los médicos en la base de datos

- private void jtMedicosMouseClicked(java.awt.event.MouseEvent evt)

Main

- public void run()

Ejecuta el constructor para correr el formulario

- Administrar Motoristas

Variables Globales:
public int filaAnterior = 0;
OperacionesUsuarios opU = new OperacionesUsuarios();
Operaciones op = new Operaciones();
OperacionesReportes opr = new OperacionesReportes();
RutaRecDomiciliar recoleccionDomiciliar = new RutaRecDomiciliar();
String usu;

Constructor:

- public AdministrarMotoristas()

Construye los componentes swing del formulario
Métodos:

- private void jbEliminarActionPerformed(java.awt.event.ActionEvent evt)

Elimina un motorista de la base de datos, y actualiza la tabla de motoristas

- private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt)

Cierra el formulario

- private void jbAgregarMotoristaActionPerformed(java.awt.event.ActionEvent evt)

Almacena un nuevo motorista en la base de datos

- private void formWindowClosed(java.awt.event.WindowEvent evt)

Cierra el formulario

- public void comoSalir()

Cierra el formulario, y continúa en ejecución los formularios que estaban corriendo

- public void comoSalir2()

Cierra el formulario, y continúa en ejecución los formularios que estaban corriendo

- private void jbNuevoMotoristaActionPerformed(java.awt.event.ActionEvent evt)

Crea una nueva fila en la tabla motorista

- private void jbModificar2ActionPerformed(java.awt.event.ActionEvent evt)

Modifica un motorista en la base de datos

- private void jmiImprimirActionPerformed(java.awt.event.ActionEvent evt)

Imprime los motoristas contenidos en la tabla motorista

- private void jtMotoristasKeyPressed(java.awt.event.KeyEvent evt)
- private void jtMotoristasKeyReleased(java.awt.event.KeyEvent evt)

Almacena los motoristas en la base de datos

- private void jtMotoristasMouseClicked(java.awt.event.MouseEvent evt)

Main

- public void run()

Ejecuta el constructor para correr el formulario

- Administrar usuarios

Variables Globales:
OperacionesUsuarios op = new OperacionesUsuarios();

Constructor:

- public AdministrarUsuarios()

Construye los componentes swing del formulario

Métodos:

- private void jbNuevoUsuarioActionPerformed(java.awt.event.ActionEvent evt)

Abre el formulario de nuevo usuario

- private void jbModificarActionPerformed(java.awt.event.ActionEvent evt)

Modifica un usuario

- private void jbEliminarActionPerformed(java.awt.event.ActionEvent evt)

Eliminar un usuario de la base de datos

Main

- public void run()

Ejecuta el constructor para correr el formulario

- Administrar Vehículos

Variables Globales:
public int filaAnterior = 0;
Operaciones op = new Operaciones();
OperacionesUsuarios opU = new OperacionesUsuarios();

RutaRecDomiciliar recoleccionDomiciliar = new RutaRecDomiciliar();
Constructor:

- public AdministrarVehiculos()

Construye los componentes swing del formulario

Métodos:

- private void jbNuevoVehiculoActionPerformed(java.awt.event.ActionEvent evt)

Agrega una nueva fila a la tabla vehículos

- private void jbModificar2ActionPerformed(java.awt.event.ActionEvent evt)

Modifica los vehículos en la tabla vehículos

- private void jtVehiculosKeyPressed(java.awt.event.KeyEvent evt)

- private void jtVehiculosKeyReleased(java.awt.event.KeyEvent evt)

Almacena un vehículo en la base de datos

- private void jbEliminarActionPerformed(java.awt.event.ActionEvent evt)

Elimina un vehículo de la base de datos

- private void jbAgregarActionPerformed(java.awt.event.ActionEvent evt)

Agregar un vehículo, al lugar de destino especificado

- private void formWindowClosed(java.awt.event.WindowEvent evt)

Cierra el formulario

- public void comoSalir()

Cierra el formulario

- Public void comoSalir2()

Cierra el formulario
Main

- public void run()

Ejecuta el constructor para correr el formulario

- Buscar Niño

Variables Globales:
OperacionesConsulta op = new OperacionesConsulta();
OperacionesDonantNinos opDN = new OperacionesDonantNinos();
String nombre = "";
String apellido = "";

String sexo = "";
String carnet = "";
String fecha = "";
Date fechade = null;
Date fechaHasta = null;
DefaultTableModel model = new DefaultTableModel();Constructor:

- Public BuscarNino()

Construye los componentes swing del formulario

Métodos:

- private void jrbIgualAActionPerformed(java.awt.event.ActionEvent evt)

Elimina la selección de los grupos de radio buttons

- private void jrbSeleccionarActionPerformed(java.awt.event.ActionEvent evt)

Selecciona un nuevo radio Button

- private void jrbTodosCarnetActionPerformed(java.awt.event.ActionEvent evt)

Selecciona un nuevo radio button

- private void jrbTodosFNacActionPerformed(java.awt.event.ActionEvent evt)

Selecciona un nuevo radio buton

- private void jbSalirActionPerformed(java.awt.event.ActionEvent evt)

Cierra el formulario

- private void formWindowClosed(java.awt.event.WindowEvent evt)

Cierra el formulario

- private void jtfNombreFocusLost(java.awt.event.FocusEvent evt)

Asigna un valor a la variable nombre

- private void jtfApellidosFocusLost(java.awt.event.FocusEvent evt)

Asigna un valor a la variable apellido

- public void comoSalir()

Cierra el formulario

- private void jbAceptarActionPerformed(java.awt.event.ActionEvent evt)

Busca en la base de datos el valor del campo

- private void jrbSexoTodosActionPerformed(java.awt.event.ActionEvent evt)

Selecciona el radio buton

- private void jrbSexoFemeninoActionPerformed(java.awt.event.ActionEvent evt)

Selecciona el radio button sexo

- private void jtfNCarnetBuscarFocusLost(java.awt.event.FocusEvent evt)

Asigna un valor a la variable carnet

- private void jccDeDateChanged(org.freixas.jcalendar.DateEvent evt)

Asigna un valor a la variable fechade

- private void jccHastaDateChanged(org.freixas.jcalendar.DateEvent evt)

Asigna un valor a la variable fechahasta

- private void jbLimpiarActionPerformed(java.awt.event.ActionEvent evt)

Limpia los campos de la interfaz grafica

Main

- public void run()

Ejecuta el constructor para correr el formulario

- Consult

Variables Globales:

```
static String fechaNac = "";
Date fecha = null;
String diagnostico;
double racionAsignada;
int JVPM;
static String carnet = "";
double peso;
double estatura;
String perimetroCefalico;
String condSalud;
OperacionesConsulta op = new OperacionesConsulta();
```

Constructor:

- Public Consult()

Construye los componentes swing del formulario

Métodos:

- public void limpiarGui()

Limpia los campos de l formulario

- private void jrbOtraActionPerformed(java.awt.event.ActionEvent evt)

Habilita el campo de texto “otra enfermedad”

- private void jrbNiñoSanoActionPerformed(java.awt.event.ActionEvent evt)

Asigna valor a la variable condSalud

- private void jrbIRAActionPerformed(java.awt.event.ActionEvent evt)

Asigna valor a la variable condSalud

- private void jrbEDAActionPerformed(java.awt.event.ActionEvent evt)

Asigna valor a la variable condSalud

- private void formWindowClosed(java.awt.event.WindowEvent evt)

Cierra el formulario

- private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt)

Limpia los campos de la interfaz

- private void jtfRacionAsignadaKeyTyped(java.awt.event.KeyEvent evt)

- private void jsPesoStateChanged(javax.swing.event.ChangeEvent evt)

Asigna valor a la variable peso

- private void jsEstaturaStateChanged(javax.swing.event.ChangeEvent evt)

Asigna valor a la variable estatura

- private void jsPerimetroCefalicoStateChanged(javax.swing.event.ChangeEvent evt)

Asigna valor a la variable perímetro cefálico

- private void jtfMedicoFocusLost(java.awt.event.FocusEvent evt)

Asigna valor a la variable JVPM

- private void jtfMedicoKeyTyped(java.awt.event.KeyEvent evt)

- private void jtfOtraEnfermedadFocusLost(java.awt.event.FocusEvent evt)

Asigna valor a la variable a la variable condSalud

- private void jtfRacionAsignadaFocusLost(java.awt.event.FocusEvent evt)

Asigna valor a la variable racionAsig nada

- private void jtfDiagnosticoFocusLost(java.awt.event.FocusEvent evt)

Asignar valor a la variable diagnostico

- private void jbAceptarActionPerformed(java.awt.event.ActionEvent evt)

Almacena la consulta en la base de datos

- private void jsPesoFocusLost(java.awt.event.FocusEvent evt)

Asigna valor a la variable peso

- private void jsEstaturaFocusLost(java.awt.event.FocusEvent evt)

Asigna valor a la variable estatura

- private void jsPerimetroCefalicoFocusLost(java.awt.event.FocusEvent evt)

Asigna valor a la variable perímetro cefálico

Main

- public void run()

Ejecuta el constructor para correr el formulario

- ControlPaciente

Variables Globales:

```
OperacionesReportes opr = new OperacionesReportes();
OperacionesConsulta op = new OperacionesConsulta();
OperacionesDonantNinos opDN = new OperacionesDonantNinos();
OperacionesInformes opi = new OperacionesInformes();
OperacionesExcel opx = new OperacionesExcel();
Calculos calculos = new Calculos();
OperacionesUsuarios opU = new OperacionesUsuarios();
String usu = "";
public String nombre = "";
public String apellido = "";
public String sexo = "";
public String carnet = "";
public String semGes = "";
public Date fechaNa = null;
public String patologias = "";
public String municipio = "";
public String departamento = "";
public String edadMaterna = "";
public String gravidez = "";
public String parto = "";
public String tipo = "";
static public String Encargado = "";
public String Notas = "";
Paciente pte;
PropiedadesPaciente propiedades;
BuscarNino buscar = new BuscarNino();
```

String value = "";

Constructor:

- public ControlPaciente()

Construye los componentes de la interfaz

Métodos:

- public void limpiarHabilitarValores()

Limpia los campos de la interfaz

- public void calcularEdad(Date fecha)

Calcula la edad de los pacientes

- private void jbAlmacenarActionPerformed(java.awt.event.ActionEvent evt)

Almacena un nuevo paciente en la base de datos

- private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt)

Limpia los valores de los campos de la interfaz

- private void jrbMasculinoActionPerformed(java.awt.event.ActionEvent evt)

Asigna un valor a la variable sexo

- private void jbReporteActionPerformed(java.awt.event.ActionEvent evt)

Muestra el reporte de los pacientes

- private void jbNuevaConsultaActionPerformed(java.awt.event.ActionEvent evt)

Prepara los campos de la interfaz para una nueva consulta

- private void jbAgregarEncargadoActionPerformed(java.awt.event.ActionEvent evt)

Agrega un encargado

- private void jbAbrirNinoActionPerformed(java.awt.event.ActionEvent evt)

Despliega la información de un paciente en la interfaz

- private void jbNuevoNinoActionPerformed(java.awt.event.ActionEvent evt)

Prepara los campos de la interfaz para agregar un nuevo paciente

- private void jbEliminarNinoActionPerformed(java.awt.event.ActionEvent evt)

Elimina un paciente de la base de datos

- private void jbBuscarNinoActionPerformed(java.awt.event.ActionEvent evt)

Abre el formulario buscar paciente

- Private void jbExcelActionPerformed(java.awt.event.ActionEvent evt)

Exporta los paciente a excel

- private void jcfechaNacDateChanged(org.freixas.jcalendar.DateEvent evt)

Asigna un valor a la variable fechaNa

- private void jsSemGestacionStateChanged(javax.swing.event.ChangeEvent evt

Asigna un valor a la variable semanas de gestación

- private void jtfEncargadoFocusLost(java.awt.event.FocusEvent evt)

Asigna valor a la variable encargado

- private void jtfNombresFocusLost(java.awt.event.FocusEvent evt)

Asigna valor a la variable nombre

- private void jtfApellidosFocusLost(java.awt.event.FocusEvent evt)

Asigna valor a la variable apellidos

- private void jrbFemeninoActionPerformed(java.awt.event.ActionEvent evt)

Asigna valor a la variable sexo

- private void jtfTarjetaFocusLost(java.awt.event.FocusEvent evt)

Asigna valor a la variable carnet

- private void jtfNotasFocusLost(java.awt.event.FocusEvent evt)

Asigna valor a la variable notas

- private void formWindowClosed(java.awt.event.WindowEvent evt)

Cierra el formulario

- private void jcbDepartamentoItemStateChanged(java.awt.event.ItemEvent evt)

Cambia el estado del combobox departamentos

- private void jcbPatologiasItemStateChanged(java.awt.event.ItemEvent evt)

Cambia el estado del combobox patologías

- private void jtfPatologiasFocusLost(java.awt.event.FocusEvent evt)

Asigna valor a la variable patologias

- private void jbOtrosDatosActionPerformed(java.awt.event.ActionEvent evt)

Abre el formulario Otros Datos

- private void jbAbrirConsultaActionPerformed(java.awt.event.ActionEvent evt)

Abre la consulta seleccionada y despliega los datos en el formulario Consulta

- private void jcbPartoItemStateChanged(java.awt.event.ItemEvent evt)

Asigna valor a la variable parto

- private void jsEdadMaternaStateChanged(javax.swing.eventchangeEvent evt)

Asigna valor a la variable edad materna

- private void jcbMunicipioItemStateChanged(java.awt.event.ItemEvent evt)

Cambia el estado del combobox Municipios

- private void jbEliminarConsultaActionPerformed(java.awt.event.ActionEvent evt)

Elimina una consulta de la base de datos, en base al carnet del paciente

- private void jbBuscarConsultaActionPerformed(java.awt.event.ActionEvent evt)

Busca una consulta en la base de datos en base a la fecha y al médico que la realizo

- private void jtfPalabraBuscarFocusLost(java.awt.event.FocusEvent evt)

Asigna valor a la variable value

- private void jcbTipoNiñoItemStateChanged(java.awt.event.ItemEvent evt)

Asigna valor a la variable tipo

- private void jsGravidezFocusLost(java.awt.event.FocusEvent evt)

Asigna valor a la variable gravidez

- private void jsGravidezStateChanged(javax.swing.event.ChangeEvent evt)

Asigna valor a la variable gravidez

- private void jbModificarActionPerformed(java.awt.event.ActionEvent evt)

Modifica los datos de un paciente

- private void jsSemGestacionFocusLost(java.awt.event.FocusEvent evt)

Asigna valor a la variable semanas de gestacion

- private void jsEdadMaternaFocusLost(java.awt.event.FocusEvent evt)

Asigna valor a la variable edad materna

- private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt)

Abre el formulario Listapacientes

Main

- public void run()

Ejecuta el constructor para correr el formulario

- InfoConsult

Variables Globales:
double peso;
double estatura;
String pCefalico;

String Notas;
double racion;
Date fecha;
int JVPM;
String carnet;
String condicionSalud;
OperacionesConsulta op = new OperacionesConsulta();
static Consulta ori = null;
Consulta mod;

Constructor:

- PublicInfo Consult()

Construye los componentes swing del formulario

Métodos:

- private void jbAceptarActionPerformed(java.awt.event.ActionEvent evt)

Modifica el valor de la consulta en la base de datos

- private void jsPesoStateChanged(javax.swing.event.ChangeEvent evt)

Asigna un valor a la variable peso

- private void jsPesoFocusLost(java.awt.event.FocusEvent evt)

Asigna un valor a la variable peso

- private void jsEstaturaStateChanged(javax.swing.event.ChangeEvent evt)

Asigna valor a la variable estatura

- private void jsEstaturaFocusLost(java.awt.event.FocusEvent evt)

Asigna valor a la variable estatura

- private void jsPerimetroCefalicoStateChanged(javax.swing.event.ChangeEvent evt)

Asigna valor a la variable perímetro cefálico

- private void jsPerimetroCefalicoFocusLost(java.awt.event.FocusEvent evt)

Asigna valor a la variable perímetro cefálico

- private void jtfRacionAsignadaFocusLost(java.awt.event.FocusEvent evt)

Asigna valor a la variable ración

- private void jtfDiagnosticoFocusLost(java.awt.event.FocusEvent evt)

Asigna valor a la variable Notas

- private void formWindowClosed(java.awt.event.WindowEvent evt)

Cierra el formulario

Main

- public void run()

Ejecuta el constructor para correr el formulario

- InformeLaboratorista

Variables Globales:
Operaciones op = new Operaciones();
OperacionesUsuarios opU = new OperacionesUsuarios();
OperacionesReportes opr = new OperacionesReportes();
OperacionesInformes opi = new OperacionesInformes();
OperacionesExcel opx = new OperacionesExcel();
String usu;
public int filaAnterior = 0;
RutaRecDomiciliar recoleccion = new RutaRecDomiciliar();

Constructor:

- public InformeLaboratorista()

Construye los componentes swing del formulario
Métodos:

- public void comoSalir()

Cierra el formulario

- private void jbNuevoInformeActionPerformed(java.awt.event.ActionEvent evt)

Agrega una nueva fila a la tabla laboratorista

- private void jbModificarActionPerformed(java.awt.event.ActionEvent evt)

Modifica los valores contenidos en la tabla

- private void jbEliminarActionPerformed(java.awt.event.ActionEvent evt)

Elimina la información del laboratorista de la base de datos

- private void jbCancelarActionPerformed(java.awt.event.ActionEvent evt

Cierra el formulario

- private void formWindowClosed(java.awt.event.WindowEvent evt)

Cierra el formulario

- private void jmiImprimirActionPerformed(java.awt.event.ActionEvent evt)

Imprime la información del laboratorista

- private void jtInformeKeyReleased(java.awt.event.KeyEvent evt)

Almacena la información del laboratorista en la base de datos

- private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt

Exporta los datos del laboratorista a Excel

- private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt)

Cierra el formulario

Main

- public void run()

Ejecuta el constructor para correr el formulario

- InformeProduccion

Variables Globales:
Operaciones op = new Operaciones();
OperacionesUsuarios opU = new OperacionesUsuarios();
OperacionesExcel opx = new OperacionesExcel();
OperacionesReportes opr = new OperacionesReportes();
OperacionesInformes opi = new OperacionesInformes();
String usu;
public int filaAnterior = 0;
RutaRecDomiciliar recoleccion = new RutaRecDomiciliar();

Constructor:

- public InformeProduccion()

Construye los componentes swing del formulario
Métodos:

- private void formWindowClosed(java.awt.event.WindowEvent evt)

Cierra el formulario

- private void jbNuevoInforme1ActionPerformed(java.awt.event.ActionEvent evt)

Agrega una nueva fila a la tabla de producción

- private void jbModificar1ActionPerformed(java.awt.event.ActionEvent evt)

Modifica el valor de la tabla producción

- private void jbEliminar1ActionPerformed(java.awt.event.ActionEvent evt)

Elimina la información de producción de la base de datos

- private void jbCancelar1ActionPerformed(java.awt.event.ActionEvent evt)

Cierra el formulario

- private void jbExcelActionPerformed(java.awt.event.ActionEvent evt)

Exporta la información de producción a Excel.

Main

- public void run()

Ejecuta el constructor para correr el formulario

- Lista Pacientes

Variables Globales:
Operaciones op = new Operaciones();
OperacionesConsulta opc = new OperacionesConsulta();
OperacionesInformes opi = new OperacionesInformes();

Constructor:

- public ListaPacientes()

Construye los componentes swing del formulario
Métodos:

- public void exportarExcel(String ruta)

Exporta los pacientes listados a excel

- private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)

Exporta los pacientes listados a Excel

- private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)

Busca los pacientes en la base de datos en base a los parametros introducidos

Main

- public void run()

Ejecuta el constructor para correr el formulario

- Logueo

Variables Globales:
Usuario usuario = new Usuario();
OperacionesUsuarios op = new OperacionesUsuarios();
SeleccionActividad act = new SeleccionActividad();

Constructor:

- public Logueo()

Construye los componentes swing del formulario
Métodos:

- public Image getIconImage()

Cambia la imagen del icono

- Public void ingresarDatos(String user, String pass)

Verifica si el usuario existe en la base de datos

- private void jbAceptarActionPerformed(java.awt.event.ActionEvent evt)

Ingresa los datos para validación
Main

- public void run()

Ejecuta el constructor para correr el formulario

- Nueva Donación

Variables Globales:
OperacionesDonacion opD = new OperacionesDonacion();
String fechaDon = "";
String racAceptada = "";
String racDescartada = "";
String tipoDon = "";

String idRuta = "";
String peso = "";
String estatura = "";
String telefono = "";
String direccion = "";
String documento = "";

Constructor:

- public NuevaDonacion()

Construye los componentes swing del formulario

Métodos:

- public void fecha()

Calcula la fecha de donación

- private void formWindowClosed(java.awt.event.WindowEvent evt)

Cierra el formulario

- private void jbIdRutaActionPerformed(java.awt.event.ActionEvent evt)

Coloca el indice oculto en 1

- private void jsPesoMouseWheelMoved(java.awt.event.MouseWheelEvent evt)

Cambia el valor del jsPeso

- private void jsPesoStateChanged(javax.swing.event.ChangeEvent evt)

Asigna valor a la variable peso

- private void jsEstaturaMouseWheelMoved(java.awt.event.MouseWheelEvent evt)

Cambia el valor del jsEstatura

- private void jsEstaturaStateChanged(javax.swing.event.ChangeEvent evt)

Asigna valor a la variable estatura

- private void jbCancelar1ActionPerformed(java.awt.event.ActionEvent evt)

Cierra el formulario

- private void jbAceptarActionPerformed(java.awt.event.ActionEvent evt)

Almacena la donacion en la base de datos

- private void jtfRacionAceptadaFocusLost(java.awt.event.FocusEvent evt)

Asigna valor a la variable racAceptada

- private void jcFechaDonacionDateChanged(org.freixas.jcalendar.DateEvent evt)

Ejecuta el metodo fecha

- private void jrbLocalActionPerformed(java.awt.event.ActionEvent evt)

Asigna valor a la variable tipoDon

- private void jrbDomiciliarActionPerformed(java.awt.event.ActionEvent evt)

Asigna valor a la variable tipoDon

- private void jtfRacionDescartadaFocusLost(java.awt.event.FocusEvent evt)

Asigna valor a la variable racDescartada

Main

- public void run()

Ejecuta el constructor para correr el formulario

- Otros datos

Variables Globales:
OperacionesDonacion opD = new OperacionesDonacion();
String fechaDon = "";
String racAceptada = "";
String racDescartada = "";
String tipoDon = "";
String idRuta = "";
String peso = "";
String estatura = "";
String telefono = "";
String direccion = "";
String documento = "";

Constructor:
- public OtrosDatos()

Construye los componentes swing del formulario
Métodos:
- public void fecha()

Calcula la fecha de donación
- private void formWindowClosed(java.awt.event.WindowEvent evt)

Cierra el formulario
- private void jbIdRutaActionPerformed(java.awt.event.ActionEvent evt)

Coloca el indice oculto en 1
- private void jsPesoMouseWheelMoved(java.awt.event.MouseWheelEvent evt)

Cambia el valor del jsPeso
- private void jsPesoStateChanged(javax.swing.event.ChangeEvent evt)

Asigna valor a la variable peso
- private void jsEstaturaMouseWheelMoved(java.awt.event.MouseWheelEvent evt)

Cambia el valor del jsEstatura
- private void jsEstaturaStateChanged(javax.swing.event.ChangeEvent evt)

Asigna valor a la variable estatura
- private void jbCancelar1ActionPerformed(java.awt.event.ActionEvent evt)

Cierra el formulario
- private void jbAceptarActionPerformed(java.awt.event.ActionEvent evt)

Almacena la donacion en la base de datos
- private void jtfRacionAceptadaFocusLost(java.awt.event.FocusEvent evt)

Asigna valor a la variable racAceptada
- private void jcFechaDonacionDateChanged(org.freixas.jcalendar.DateEvent evt)

Ejecuta el metodo fecha
- private void jrbLocalActionPerformed(java.awt.event.ActionEvent evt)

Asigna valor a la variable tipoDon
- private void jrbDomiciliarActionPerformed(java.awt.event.ActionEvent evt)

Asigna valor a la variable tipoDon
- private void jtfRacionDescartadaFocusLost(java.awt.event.FocusEvent evt)

Asigna valor a la variable racDescartada

Main

- public void run()

Ejecuta el constructor para correr el formulario

- Recoleccion Domiciliar

Variables Globales:

Operaciones op = new Operaciones();String fechaDon = "";

Constructor:

- public RecoleccionDomiciliar()

Construye los componentes swing del formulario e invoca al método op.listarRutas(jtRutas); para listar las recolecciones automáticamente.

Métodos:

- public Image getIconImage()

Importa el ícono del formulario

- public void comoSalir()

Cierra el formulario en base a un criterio específico

- private void jbNuevaRutaActionPerformed(java.awt.event.ActionEvent evt)

Agrega una nueva ruta

- private void formWindowClosed(java.awt.event.WindowEvent evt)

Invoca al método comoSalir()

- private void jbAnyadirActionPerformed(java.awt.event.ActionEvent evt))

Agrega una ruta.

- private void jbSalirActionPerformed(java.awt.event.ActionEvent evt)

Salir del formulario

- private void jbEliminarActionPerformed(java.awt.event.ActionEvent evt)

Elimina una ruta

- private void jbModificarRutaActionPerformed(java.awt.event.ActionEvent evt)

Modificar una ruta

- private void jbBuscarActionPerformed(java.awt.event.ActionEvent evt)

Buscar una ruta

- private void main()

Corre el formulario

- Recoleccion Leche

Variables Globales

OperacionesUsuarios opU = new OperacionesUsuarios();

OperacionesDonacion opD = new OperacionesDonacion();

OperacionesExcel opX = new OperacionesExcel();

OperacionesReportes opr = new OperacionesReportes();

String usu;

String nombre = "";

String apellido = "";

```
String documento = "";
Date fechaNac = null;
String peso = "";
String estatura = "";
String telefono = "";
String direccion = "";
String semGest = "";
Date fechaParto = null;
String vdrl = "";
String hbsag = "";
String hiv = "";
String transSang = "";
String tabaq = "";
String etilis = "";
String aptaDon = "";
Date fechaObtDatos = null;
```

Métodos:

- public RecoleccionLeche()

Constructor del formulario

- private void jbAbrirDonacionActionPerformed(java.awt.event.ActionEvent evt)

Abre una donación

- private void formWindowClosed(java.awt.event.WindowEvent evt)

Cierra el formulario

- private void jbEliminarDonanteActionPerformed(java.awt.event.ActionEvent evt)

Elimina una donante

- private void jbNuevaDonanteActionPerformed(java.awt.event.ActionEvent evt)

Agrega una nueva donante

- private void jbNuevaDonacionActionPerformed(java.awt.event.ActionEvent evt)

Agrega una nueva donación

- public void nuevaTabla()

Establece una table en blanco

- public void limpiarHabilitarValores(Boolean enabled)

Permite que los campos esten habilitados para su edición

- public Date enviarFecha(String fechaSQL)

Envia una fecha formateada a la base de datos.

- public void enviarDatos()

Envia los datos a la base de datos

- private void jbAbrirInfoDonanteActionPerformed(java.awt.event.ActionEvent evt)

Abrir la información de un donante

- public void asigancion()

Iguala los valores de las variables a la de los campos del formulario.

- public void asigancion2()

Iguala los valores de las variables a la de los campos del formulario.

- private void jbEliminarDonacionActionPerformed(java.awt.event.ActionEvent evt)

Borra una donación.

- private void jbBuscarDonanteActionPerformed(java.awt.event.ActionEvent evt)

Busca una donante en específico

- private void jbGuardarActionPerformed(java.awt.event.ActionEvent evt)

Guarda los cambios hechos en la donante.

- private void jbModificarActionPerformed(java.awt.event.ActionEvent evt)

Modifica los valores de una donante

- private void jbPrintActionPerformed(java.awt.event.ActionEvent evt)

Imprime en reporte

- Ruta Recoleccion Domiciliar

Variables Globales

```
static public String motorista = "";
static public String vehiculo = "";
static public String medico = "";
public int filaAnterior = 0;
Operaciones op = new Operaciones();
```

Métodos:

- public RutaRecDomiciliar()

Constructor

- private void jbAnyadirMotoristaActionPerformed(java.awt.event.ActionEvent evt)

Agrega un motorista

- private void jbAnyadirVehActionPerformed(java.awt.event.ActionEvent evt)

Agrega un vehiculo

- private void jbAnyadirMedicoActionPerformed(java.awt.event.ActionEvent evt)

Agrega un medico

- private void jbGuardarActionPerformed(java.awt.event.ActionEvent evt)

Almacenas las rutas

- private void jbAgregarTrayectoriaActionPerformed(java.awt.event.ActionEvent evt)

Agregar rutas

- private void jbModificarActionPerformed(java.awt.event.ActionEvent evt)

Invoca al metodo para modificarRutas()

- private void jbEliminarActionPerformed(java.awt.event.ActionEvent evt)

Eliminar ruta.

- Selección Domiciliar

Variables Globales

```
String usu = "w-out";
OperacionesUsuarios opU = new OperacionesUsuarios();
OperacionesReportes opr = new OperacionesReportes();
```

Métodos (principales)

- private void jlDonacionMouseClicked(java.awt.event.MouseEvent evt)

Abre el formulario de Donaciones

- private void jlConsultasMouseClicked(java.awt.event.MouseEvent evt)

Abre el formulario Consultas

- private void jlRecDomiciliarMouseClicked(java.awt.event.MouseEvent evt)

Abre el formulario Recoleccion Domiciliar

- private void formWindowClosing(java.awt.event.WindowEvent evt)

Cierre de la ventana y redirección al logueo.

- Nuevo Usuario

Variables Globales

```
OperacionesUsuarios opU = new OperacionesUsuarios();
Boolean adm;
```

Métodos

- private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)

Agregar un Nuevo usuario al sistema

- private void jButton2ActionPerformed(java.awt.event.ActionEvent evt)

Limpia todos los campos