



Sistema para el Control de Activos Informáticos en la FACCI

UNIVERSIDAD LAICA ELOY ALFARO DE MANABÍ

FACULTAD DE CIENCIAS INFORMÁTICAS

INGENIERÍA EN SISTEMAS

MANUAL TÉCNICO

CURSO: 6to Nivel “A”

DOCENTE: Ing. Winther Molina

**Manta – Manabí – Ecuador
2016**

Tabla de contenido

TEMA	3
OBJETIVOS	3
Objetivo General:.....	3
Objetivo Específicos:.....	3
METODOLOGÍA DE DESARROLLO APLICADA.....	3
Modelo Propio de Calidad: Proceso Unificado Ágil	3
DICCIONARIO DE DATOS:.....	6
DIAGRAMA ENTIDAD - RELACIÓN	13
INTERFACES	14
Ingreso al Sistema	14
Ventana Principal del sistema:	15
Recuperación de clave:	16
Registro de Activo:	17
Búsqueda de Activos:	18
Búsqueda de activos sin responsables:	19
Registro de Documento de Asignación:.....	20
Registro de activos por responsable:	21
Gestión de Áreas:	22
Búsqueda de Recurso Humano:	23
Control de recursos Humanos:.....	24
Listado de Activos:	25
Registro de Solicitud de mantenimiento:	27
Control de Solicitudes de mantenimiento:.....	28
Registro Encargado de Mantenimiento:.....	29
Listado de entregas:	30
Registro de Entregas:	31
Configuración de cuenta:	32
Interfaz de Configuración de cuenta Inicial	33

TEMA

Sistema para el Control de activos informáticos fijos de la Facultad de Ciencias Informáticas de la Universidad Laica “Eloy Alfaro” de Manabí.

OBJETIVOS

Objetivo General:

- ✓ Desarrollar e Implementar un sistema para la gestión de los activos fijos informáticos de la Facultad de Ciencias Informáticas de la Universidad Laica “Eloy Alfaro” de Manabí.

Objetivo Específicos:

- ✓ Realizar un estudio de los procesos manuales que se aplican para el control de activos fijos informáticos de la Facultad de Ciencias Informáticas.
- ✓ Aplicar la metodología del modelo propio de calidad.
- ✓ Construir una herramienta tecnológica de escritorio que permita llevar el control, manejo y administración de los procedimientos de gestión que se realizan con los activos informáticos fijos de la Facultad de Ciencias Informáticas.
- ✓ Entregar toda la documentación respectiva del proyecto a los usuarios.

METOLOGÍA DE DESARROLLO APLICADA

Modelo Propio de Calidad: Proceso Unificado Ágil

Para la ejecución de los procesos de desarrollo del presente proyecto aplicamos un modelo propio de calidad basándonos en el Proceso Unificado Ágil (AUP, del inglés *Agile Unified Process*) el cual es una versión simplificada del Proceso Unificado Racional (*Rational Unified Process*, RUP), que describe una aproximación al desarrollo de aplicaciones que combina conceptos propios del proceso unificado tradicional con técnicas ágiles, con el objetivo de mejorar la productividad. En general, el Proceso Unificado Ágil supone un enfoque intermedio entre XP (*eXtreme Programming*) y el Proceso Unificado Racional. Por lo que este proceso adopta algunas de las técnicas ágiles de XP y otros

procesos ágiles, pero reteniendo parte de la formalidad del Proceso Unificado Racional.

Dicho proceso consta de cuatro fases que el proyecto atraviesa de forma secuencial:

1. **Iniciación.** El objetivo de esta fase es identificar el alcance inicial del proyecto, una arquitectura potencial para el sistema y obtener, si procede, financiación para el proyecto y la aceptación por parte de los promotores del sistema.
2. **Elaboración.** Mediante esta fase se pretende identificar y validar la arquitectura del sistema.
3. **Construcción.** El objetivo de esta fase consiste en construir software desde un punto de vista incremental basado en las prioridades de los participantes.
4. **Transición.** En esta fase se valida y despliega el sistema en el entorno de producción.

La siguiente figura muestra un esquema de estas cuatro fases incluyendo, además, los objetivos y tareas fundamentales de cada una.

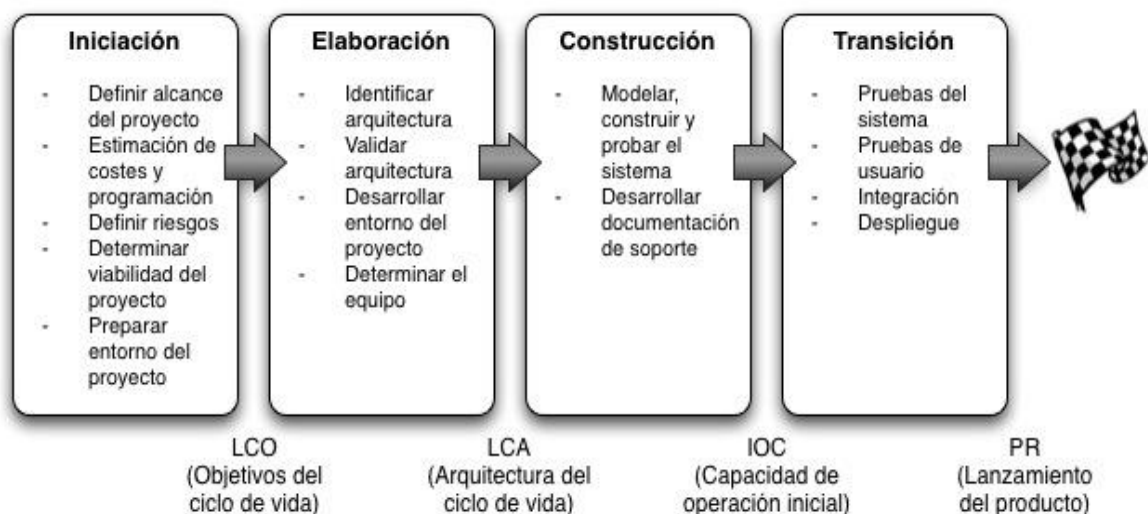


Figura 1. Esquema de las cuatro fases del Proceso Unificado Ágil.

En base a lo descrito anteriormente se ha definido el siguiente esquema donde se plantea los procesos que se deben aplicar para el desarrollo del proyecto software requerido:

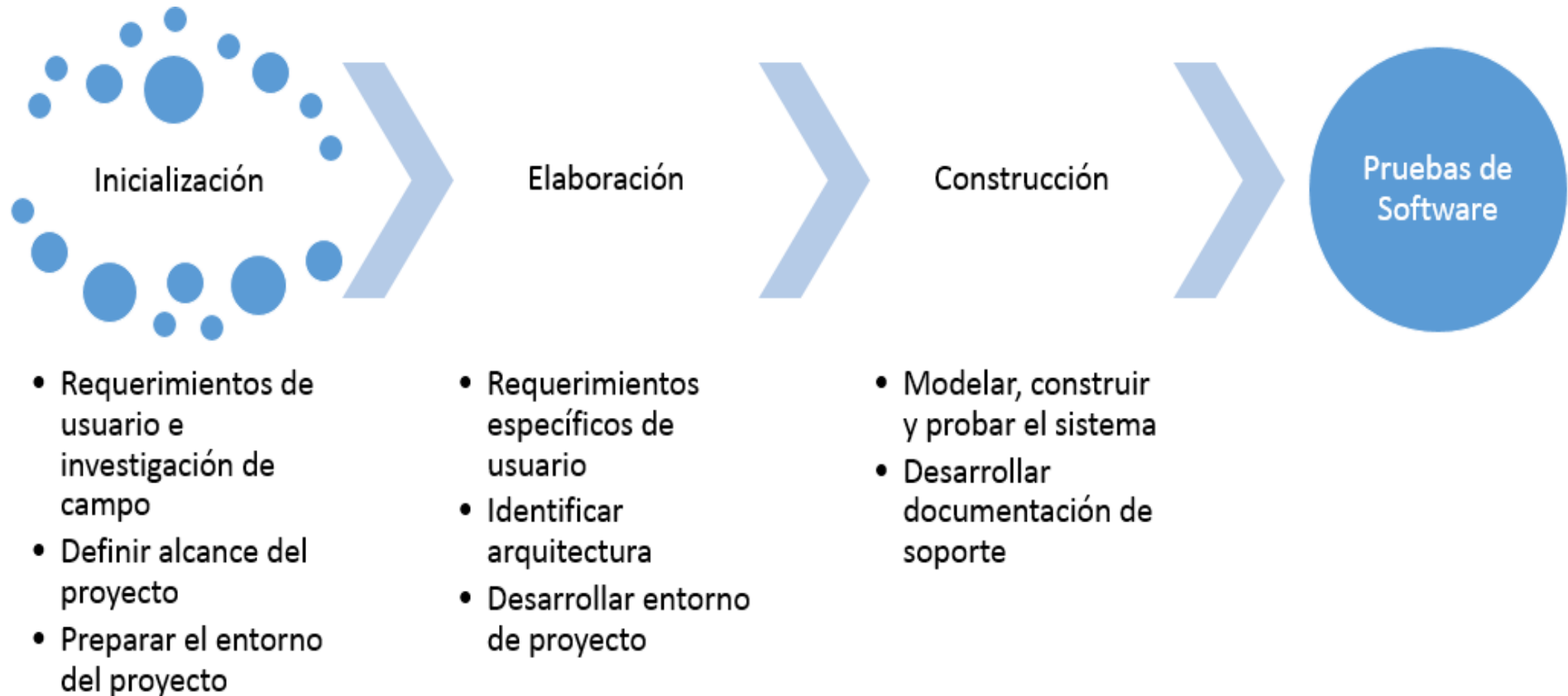


Figura 2. Esquema de los procesos de desarrollo aplicada como metodología en el proyecto software basado en el Proceso Unificado Ágil.

DICCIONARIO DE DATOS:

<i>Descripción tabla :</i>	Institución
<i>Nombre tabla:</i>	TMAEINSCON
<i>Tipo de tabla:</i>	Configuración
<i>Clave principal:</i>	id_institucion
<i>Clave Alterna:</i>	nombre_institucion

Definición De Campos De La Tabla

Descripción	Nombre campo	Tipo	Tamaño	Dominio
Código de institución	id_institucion	serial	-	0...n
Nombre de la institución	nombre_institucion	varchar	150	Caracteres
Dirección	direccion_institucion	varchar	200	Caracteres
Correo Institucional	correo_institucion	Varchar	150	Caracteres

Todos los campos obligatorios.

<i>Descripción tabla :</i>	Tipo Activo
<i>Nombre tabla:</i>	TMAETACCON
<i>Tipo de tabla:</i>	Maestra
<i>Clave principal:</i>	id_tipoActivo
<i>Clave Alterna:</i>	nombre_tipoActivo

Definición De Campos De La Tabla

Descripción	Nombre campo	Tipo	Tamaño	Dominio
Tipo de activo	id_tipoActivo	serial	-	0...n
Nombre del activo	nombre_tipoActivo	Varchar	50	*

Todos los campos son obligatorios.

Listado tipo de activos:

- | | | |
|------------------------------|-----------------------|----------------|
| 1. Cámara de vigilancia | 5. Impresora | 10. Servidores |
| 2. CPU | 6. Mouse | 11. Teclado |
| 3. Computadora de escritorio | 7. Pantalla o Monitor | 12. Otros |
| 4. Detector de Huellas | 8. Portátil | |
| | 9. Proyector | |

Descripción tabla :	Area
Nombre tabla:	TMAEARECON
Tipo de tabla:	Maestra
Clave principal:	id_area
Clave Alterna:	idResponsable_area

Definición De Campos De La Tabla

Descripción	Nombre campo	Tipo	Tamaño	Dominio
Código Area	id_area	integer	-	0 ... n
Tipo	tipo_area	varchar	20	1:Laboratorio 2:Administración 3:Docente
Descripción	descripcion_area	varchar	200	a...zñ, A...ZÑ, - , espacios , 0...9
Código encargado / Recurso Humano	idResponsable_area	integer	-	0...n

Todos los campos son obligatorios.

Descripción tabla :	Persona
Nombre tabla:	TMAEPERCON
Tipo de tabla:	Maestra
Clave principal:	id_persona
Clave Alterna:	cedula_persona

Definición De Campos De La Tabla

Descripción	Nombre campo	Tipo	Tamaño	Dominio
Id	id_persona	integer	-	0 ... n
Cedula	cedula_persona	integer	10	0 - 9999999999
Nombres	nombre_persona	varchar	200	a...zñ, A...ZÑ, espacio
Apellidos	apellido_persona	varchar	200	a...zñ, A...ZÑ, espacio
Correo	correo_persona	varchar	150	Caracteres
Fecha Registro	fechaRegistro_persona	date	Año 1980 – Año AAAA	Fecha: aaaa/mm/dd

Todos los campos son obligatorios.

Descripción tabla :	Recursos humanos			
Nombre tabla:	TMOVREHCON			
Tipo de tabla:	Movimiento			
Clave principal:	id_rrhh			
Clave Alterna:	id_persona			
Definición De Campos De La Tabla				
Descripción	Nombre campo	Tipo	Tamaño	Dominio
Id	id_rrhh	integer	-	0 ... n
Id persona	idPersona_rrhh	integer	-	0...n
Id institución	idInstitucion_rrhh	integer	-	0...n
Tipo	tipo_rrhh	Varchar	25	1:Docente 2:Administrativo 3:Mantenimiento 4:Técnico Institucional

Todos los campos son obligatorios.

Descripción tabla :	Usuario
Nombre tabla:	TMAEUSUCON
Tipo de tabla:	Maestra
Clave principal:	id_usuario
Clave Alterna:	nombre_usuario

Definición De Campos De La Tabla

Descripción	Nombre campo	Tipo	Tamaño	Dominio
Id	id_usuario	integer	-	0 ... n
Nombre de usuario	nombre_usuario	varchar	20	admin_global, admin, admin_mantenimi ento
Clave de usuario	clave_usuario	Varchar	16	Caracteres
Tipo de usuario	tipo_usuario	Integer	1-3	1:admin_global 2:admin 3:admin_manteni miento
Código Persona	id_persona	Integer	-	0...n

Todos los campos son obligatorios.

Descripción tabla :	Activo
Nombre tabla:	TMOVACTCON
Tipo de tabla:	Movimiento
Clave principal:	id_activo
Clave Alterna:	tipo_activo

Definición De Campos De La Tabla

Descripción	Nombre campo	Tipo	Tamaño	Dominio
Id	id_activo	integer	-	0 ... n
Fecha Registro	fechaRegistro_activo	Date	-	AAAA/MM/DD
Código manejado por la institución	codigoInternoInstitucional_activo	Varchar	200	A...Z, 0...9, -
Id Tipo de Activo/Tabla tipo de activo	idTipo_activo	integer	-	0...n
Fecha Compra	fechaCompra_activo	Date	Año 1980 – Año AAAA	AAAA/MM/DD
Marca	marca_activo	varchar	50	*
Modelo	modelo_activo	varchar	150	A...Z, 0..9
Serie	serie_activo	varchar	150	A...Z, 0..9
Costo	costo_activo	Decimal	7,2	0.50 – 99999999.99
Procesador	procesador_activo	varchar	50	*
Memoria	memoria_activo	varchar	50	*
Disco duro	discoDuro_activo	varchar	50	*
Estado	estado_activo	integer	0-2	0:Activo 1:Inactivo 2:Dado de baja
Fecha del ultimo mantenimiento	fechaUltimoMantenimiento_activo	date	Año 1980 – Año AAAA	aaa/mm/dd
Detalle	detalle_activo	Varchar	500	Caracteres
Código Documento de asignación de responsable	idDocAsignacion_activo	integer	-	0...n

Listado de marcas por activos NO VA EN LA BD.

- | | |
|------------------------------|-----------------------|
| 1. Cámara de vigilancia | 7. Pantalla o Monitor |
| a. Samsung | a. LG |
| b. Sony | b. HACER |
| c. EverFocus | c. HP |
| d. Avtech | d. LENOVO |
| e. Otros | e. BENQ |
| 2. CPU | f. Otros |
| a. HP | 8. Portátil |
| b. DELL | a. HP |
| c. TOSHIBA | b. DELL |
| d. ACER | c. TOSHIBA |
| e. Otros | d. HACER |
| 3. Computadora de escritorio | e. Otros |
| a. DELL | 9. Proyector |
| b. HP | a. EPSON |
| c. TOSHIBA | b. SONY |
| d. HACER | c. PANASONIC |
| e. Otros | d. INFOCUS |
| 4. Detector de Huellas | e. Otros |
| a. Averiguarlo | 10. Servidores |
| 5. Impresora | a. IBM |
| a. Epson | b. TOSHIBA |
| b. Canon | c. DELL |
| c. HP | d. COMPAQ |
| d. XEROX | e. Otros |
| e. Otros | 11. Teclado |
| 6. Mouse | a. DELL |
| a. Genius | b. Genius |
| b. Compac | c. DLINK |
| c. HP | d. Microsoft |
| d. Microsotf | e. Otros |
| e. Otros | 12. Otros |

Listado de Memoria, Disco Duro y RAM // No va en la BD

- | | |
|---------------|---------------|
| 1. Procesador | d. 1GB |
| a. Core i7 | |
| b. Core i5 | 3. Disco Duro |
| c. Core i3 | a. 160 GB |
| 2. Memoria | b. 250 GB |
| a. 8GB | c. 320 GB |
| b. 4GB | d. 500 GB |
| c. 2GB | e. 1 TB |

f. 1.5 TB
g. 2 TB

h. 2.5 TB
i. 3 TB

Descripción tabla :	Documento Asignación Activo
Nombre tabla:	TMOVDASCON
Tipo de tabla:	Movimiento
Clave principal:	id_asigActivo
Clave Alterna:	codigoInternoInstitucional_asigActivo

Definición De Campos De La Tabla

Descripción	Nombre campo	Tipo	Tamaño	Dominio
Id	id_asigActivo	integer	-	0 ... n
Código interno de oficio manejado por la institución	codigoOficio_asigActivo	varchar	200	A...Z, 0...9, -
Persona responsable de activo por área	idArea_asigActivo	integer	-	0...n

Descripción tabla :	Solicitud de Mantenimiento
Nombre tabla:	TMOVDOSCON
Tipo de tabla:	Movimiento
Clave principal:	id_docSolicitud
Clave Alterna:	codigoOficio_docSolicitud

Definición De Campos De La Tabla

Descripción	Nombre campo	Tipo	Tamaño	Dominio
Id	id_docSolicitud	serial	-	0...n
Código interno de oficio manejado por la institución	codigoOficio_docSolicitud	varchar	200	A...Z, 0...9, -
Fecha de solicitud	fecha_docSolicitud	date	Año 1980 – Año AAAA	Aaaa/mm/dd
Fecha entrega	fechaEntrega_docSolicitud	Date	Año 1980 – Año AAAA	AAA/MM/DD
Id persona que entrega activo	idEntregante_docSolicitud	Integer	-	0...n
Id persona que solicita el mantenimiento	idSolicitante_docSolicitud	integer	-	0...n
Tipo de solicitud de mantenimiento	tipoMantPS_docSolicitud	integer	0-1	0:Programado 1:Solicitud
Tipo de mantenimiento 3	tipoGe_mantActivo	Integer	0-1	0: general 1: Eq. Individual

Motivo de la solicitud	motivo_docSolicitud	Text	1000	Caracteres
Estado de solicitud	estado_docSolicitud	integer	0-3	0:Pendiente 1:En Proceso 2:Realizada 3:Entregada

Descripción tabla : Detalle Solicitud de Mantenimiento

Nombre tabla: TMOVDTSOCON

Tipo de tabla: Movimiento

Clave principal: id_detalleSolicitudMant

Clave Alterna: idDocSolicitud_detalleSolicitudMant

Definición De Campos De La Tabla

Descripción	Nombre campo	Tipo	Tamaño	Dominio
Id	id_detalleSolicitudMant	serial	-	0...n
Código de solicitud	idDocSolicitud_detalleSolicitudMant	Integer	-	0...n
Activos de solicitud	IdActivo_detalleSolicitudMant	Integer	-	0...n

Descripción tabla : Mantenimiento de Activo

Nombre tabla: TMOVMANCON

Tipo de tabla: Movimiento

Clave principal: id_mantActivo

Clave Alterna: idDocSolicitud_mantActivo

Definición De Campos De La Tabla

Descripción	Nombre campo	Tipo	Tamaño	Dominio
ID de mantenimiento	id_mantActivo	serial	-	0...n
Fecha inicio de mantenimiento	fechaInicio_mantActivo	Date	Año 1980-AAA	AAA/MM/DD
Fecha finalización de mantenimiento	fechaFin_mantActivo	Date	Año 1980-AAA	AAA/MM/DD
Código de solicitud de mantenimiento	idDocSolicitud_mantActivo	integer	-	0...n
Tipo mantenimiento 1	tipoPC_mantActivo	Integer	0-1	0:Preventivo 1:Correctivo
Tipo mantenimiento 2	tipoHS_mantActivo	Integer	0-1	0:Hardware 1:Software
Tipo mantenimiento 3	tipoGE_mantActivo	Integer	0-1	0: General 1: Eq. Individual
Actividades realizadas	actividades_mantActivo	Text	7000	Caracteres
Observaciones	observaciones_mantActivo	Text	1000	Carácter
Persona que realiza el mantenimiento	idPersonaRealiza_mantActivo	Integer	-	0...n

DIAGRAMA ENTIDAD - RELACIÓN

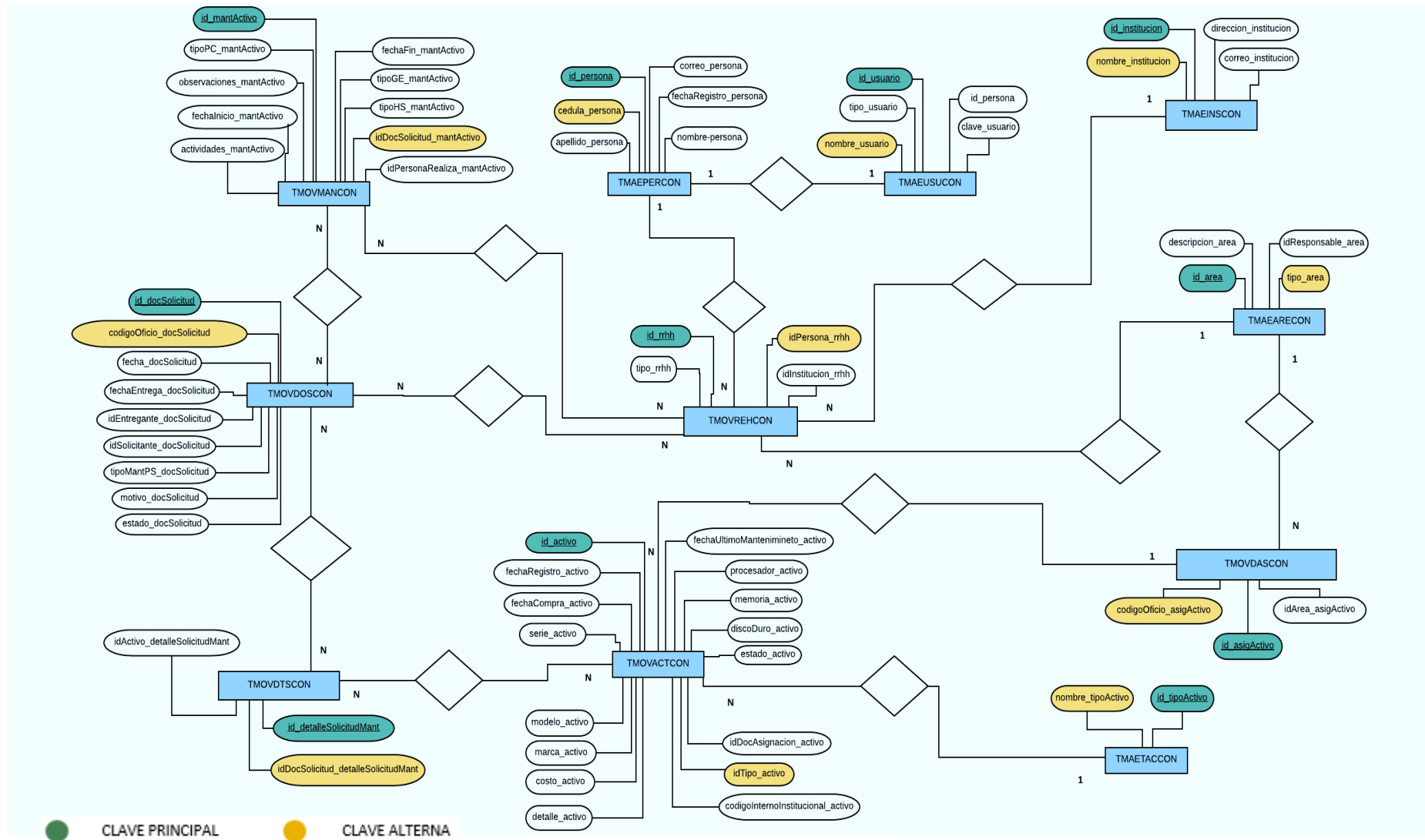
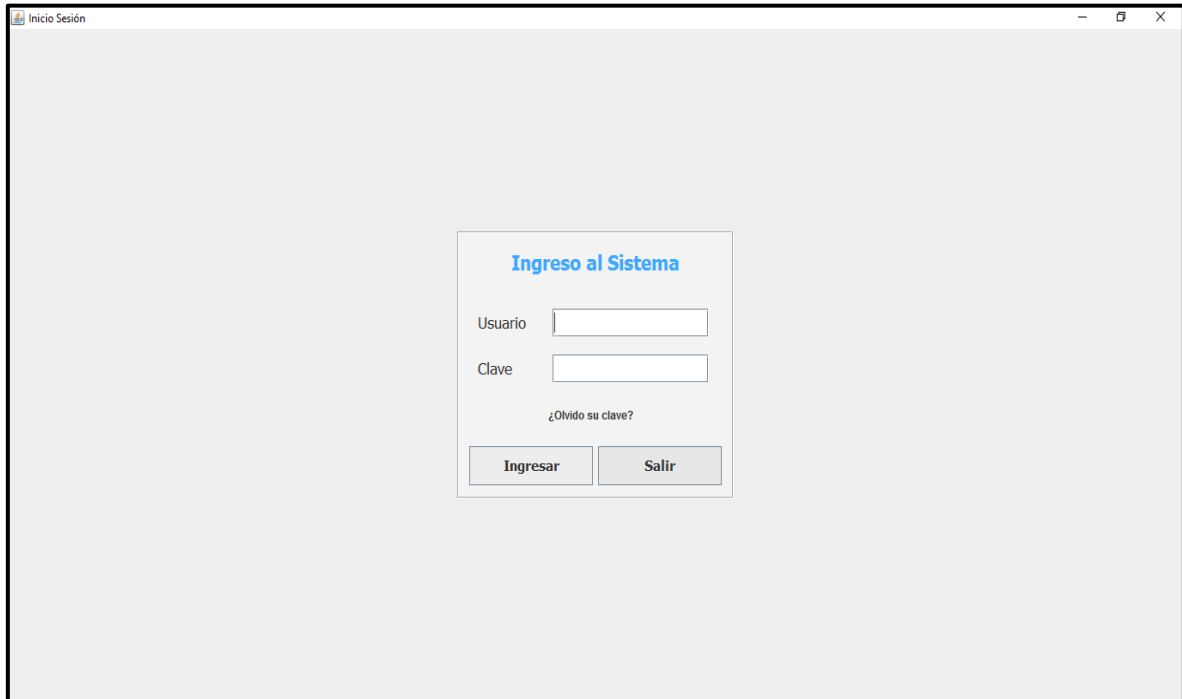


Figura 3. Diagrama Entidad-Relación

INTERFACES

Ingreso al Sistema



Código:

```
public Ingreso() {
    initComponents();
    this.getContentPane().setLayout (new GridBagLayout());
    limitar_caracteres();
}

public void limitar_caracteres(){
    RestrictedTextField limite_nombre = new RestrictedTextField(txt_nombre_usuario);
    limite_nombre.setLimit(20);
}

private void btn_ingresarActionPerformed(java.awt.event.ActionEvent evt) {
    dispose();
    new Principal().setVisible(true);
}

private void btn_cambiar_claveActionPerformed(java.awt.event.ActionEvent evt) {
    new CambiarClave(this, true).setVisible(true);
}

private void btn_salirActionPerformed(java.awt.event.ActionEvent evt) {
    System.exit(0);
}
```

```
public static void main(String args[]) {  
  
    try {  
        for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {  
            if ("Nimbus".equals(info.getName())) {  
                javax.swing.UIManager.setLookAndFeel(info.getClassName());  
                break;  
            }  
        }  
    } catch (ClassNotFoundException | InstantiationException |  
            IllegalAccessException | javax.swing.UnsupportedLookAndFeelException ex) {  
        java.util.logging.Logger.getLogger(Ingreso.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);  
    }  
}
```

Ventana Principal del sistema:



Código:

```
public Principal() {  
    initComponents();  
    setLocationRelativeTo(null);  
}  
  
private void formWindowStateChanged(java.awt.event.WindowEvent evt) {  
  
}  
  
private void btn_registrar_activoActionPerformed(java.awt.event.ActionEvent evt) {  
    new RegistroActivos(this, true).setVisible(true);  
}  
  
private void btn_control_entregasActionPerformed(java.awt.event.ActionEvent evt) {  
    new ControlEntrega(this, true).setVisible(true);  
}  
  
private void btn_cerrar_sesionActionPerformed(java.awt.event.ActionEvent evt) {  
    dispose();  
    new Ingreso().setVisible(true);  
}  
  
private void btn_asignar_activoActionPerformed(java.awt.event.ActionEvent evt) {  
    new AsignaciónActivos(this, true).setVisible(true);  
}
```

```
private void btn_activos_responsableActionPerformed(java.awt.event.ActionEvent evt) {  
    new BusquedaActivoResponsable( this, true).setVisible(true);  
}  
  
private void btn_gestion_areasActionPerformed(java.awt.event.ActionEvent evt) {  
    new GestionAreas( this, true).setVisible(true);  
}  
  
private void btn_registrar_rrhhActionPerformed(java.awt.event.ActionEvent evt) {  
    new Control_Recursos_Humanos( this, true).setVisible(true);  
}  
  
private void btn_solicitar_mantenimientoActionPerformed(java.awt.event.ActionEvent evt) {  
    new Solicitar_Mantenimiento( this, true).setVisible(true);  
}  
  
private void btn_actualizar_solicitudesActionPerformed(java.awt.event.ActionEvent evt) {  
    new Control_solicitudes_Mantenimiento( this, true).setVisible(true);  
}  
  
private void btn_mi_cuentaActionPerformed(java.awt.event.ActionEvent evt) {  
    new MiCuenta( this, true).setVisible(true);  
}  
  
private void btn_cambiar_claveActionPerformed(java.awt.event.ActionEvent evt) {  
    new CambiarClave( this, true).setVisible(true);  
}
```

Recuperación de clave:

Código:

```
public CambiarClave(java.awt.Frame parent, boolean modal) {  
    super(parent, modal);  
    initComponents();  
    this.setLocationRelativeTo(null);  
    setLocationRelativeTo(null);  
}  
  
private void btn_regresarActionPerformed(java.awt.event.ActionEvent evt) {  
    dispose();  
}  
  
private void txt_nuevaclave_usuarioActionPerformed(java.awt.event.ActionEvent evt) {  
}
```


Registro de Activo:

Código:

```
public RegistroActivos(java.awt.Frame parent, boolean modal) {
    super(parent, modal);
    initComponents();
    this.getContentPane().setLayout (new GridBagLayout());
    this.setLocationRelativeTo(null);
}

private void formMousePressed(java.awt.event.MouseEvent evt) {
}

private void formMouseClicked(java.awt.event.MouseEvent evt) {
}

private void formKeyPressed(java.awt.event.KeyEvent evt) {
}

private void combo_tipo_activoActionPerformed(java.awt.event.ActionEvent evt) {
}

private void combo_procesador_activoActionPerformed(java.awt.event.ActionEvent evt) {
}

private void combo_discoduro_activoActionPerformed(java.awt.event.ActionEvent evt) {
}

private void txt_serie_activoActionPerformed(java.awt.event.ActionEvent evt) {
}

private void btn_buscarActionPerformed(java.awt.event.ActionEvent evt) {
    new BusquedaActivo( this, true).setVisible(true);
}

private void btn_regresarActionPerformed(java.awt.event.ActionEvent evt) {
    dispose();
}
```

Búsqueda de Activos:

Código:

```
public BusquedaActivo(java.awt.Dialog parent, boolean modal) {
    super(parent, modal);
    initComponents();
    this.getContentPane().setLayout (new GridBagLayout());
    this.setLocationRelativeTo(null);
    tabla_activo.getTableHeader().setReorderingAllowed(false);
}

private void btn_nuevoActionPerformed(java.awt.event.ActionEvent evt) {
}

private void btn_regresarActionPerformed(java.awt.event.ActionEvent evt) {
    dispose();
}

private void
rbtn_codigoinstitucional_activoActionPerformed(java.awt.event.ActionEvent evt){
}

private void rbtn_tipoActionPerformed(java.awt.event.ActionEvent evt) {
}

private void rbtn_marca_activoActionPerformed(java.awt.event.ActionEvent evt) {
}

private void btn_darBajaActionPerformed(java.awt.event.ActionEvent evt) {
}
```

```
private void btn_modificarActionPerformed(java.awt.event.ActionEvent evt) {  
    }  
}
```

Búsqueda de activos sin responsables:

Código:

```
public AsignaciónActivos(java.awt.Frame parent, boolean modal) {  
    super(parent, modal);  
    initComponents();  
    panelBusquedaTipo.setVisible(false);  
    this.setLocationRelativeTo(null);  
    cargar();  
}  
  
Conexion conexion = new Conexion();  
  
public void cargar() {  
    if (conexion.crearConexion()) {  
        combo_tipo_activo.removeAllItems();  
        ArrayList<String> resultat;  
        ArrayList<String> ls = new ArrayList<>();  
        String sql="select nombre_tipoactivo from tmaetaccon";  
        ResultSet rs = conexion.ejecutarSQLSelect(sql);  
        try {  
            while(rs.next()) {  
                ls.add(rs.getString("nombre_tipoactivo"));  
            }  
        }  
    }  
}
```

```

        } catch (SQLException ex) {
        }
        resultat = ls;

        for (String resultat1 : resultat) {
            combo_tipo_activo.addItem(resultat1);
        }
    }
}

private void combo_tipo_activoMouseEntered(java.awt.event.MouseEvent evt) {
    if (!combo_tipo_activo.getSelectedItem().equals("Computador") ||
        !combo_tipo_activo.getSelectedItem().equals("Laptop") || !combo_tipo_activo.getSelectedItem().equals("CPU"))
        combo_procesador_activo.setEnabled(false);
        combo_memoria_activo.setEnabled(false);
        combo_discoduro_activo.setEnabled(false);
        rbtn_procesador_activo.setEnabled(false);
        rbtn_memoria_activo.setEnabled(false);
        rbtn_discoduro_activo.setEnabled(false);
    } else{
        combo_procesador_activo.setEnabled(true);
        combo_memoria_activo.setEnabled(true);
        combo_discoduro_activo.setEnabled(true);
        rbtn_procesador_activo.setEnabled(true);
        rbtn_memoria_activo.setEnabled(true);
        rbtn_discoduro_activo.setEnabled(true);
    }
}

```

Registro de Documento de Asignación:

Código:

```

public RegistroDocumentoAsignacion(java.awt.Dialog parent, boolean modal) {
    super(parent, modal);
    initComponents();
    this.getContentPane().setLayout(new GridBagLayout());
    this.setLocationRelativeTo(null);
    tabla_documentos_asignacion.getTableHeader().setReorderingAllowed(false);
}

private void btn_regresarActionPerformed(java.awt.event.ActionEvent evt) {
    dispose();
}

```

```

public static void main(String args[]) {

    try {
        for (javax.swing.UIManager.LookAndFeelInfo info
            : javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException | InstantiationException | IllegalAccessException
        | javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(RegistroDocumentoAsignacion.class.getName())
            .log(java.util.logging.Level.SEVERE, null, ex);
    }

    java.awt.EventQueue.invokeLater(() -> {
        RegistroDocumentoAsignacion dialog = new RegistroDocumentoAsignacion
            (new javax.swing.JDialog(), true);
        dialog.addWindowListener(new java.awt.event.WindowAdapter() {
            @Override
            public void windowClosing(java.awt.event.WindowEvent e) {
                System.exit(0);
            }
        });
        dialog.setVisible(true);
    });
}

```

Registro de activos por responsable:

Código:

```

public BusquedaActivoResponsable(java.awt.Frame parent, boolean modal) {
    super(parent, modal);
    initComponents();
    this.getContentPane().setLayout (new GridBagLayout());
    this.setLocationRelativeTo(null);
    tabla_activos_responsable.getTableHeader().setReorderingAllowed(false);
}

```

```
private void txt_responsableKeyTyped(java.awt.event.KeyEvent evt) {
    Character c = evt.getKeyChar();
    if(Character.isLetter(c) || c == KeyEvent.VK_SPACE) {
        evt.setKeyChar(c);
    }else{
        evt.consume();
    }
}

private void btn_actualizarActionPerformed(java.awt.event.ActionEvent evt) {
    new RegistroDocumentoAsignacion( this, true).setVisible(true);
}

private void btn_regresarActionPerformed(java.awt.event.ActionEvent evt) {
    dispose();
}

java.awt.EventQueue.invokeLater(() -> {
    BusquedaActivoResponsable dialog = new BusquedaActivoResponsable
        (new javax.swing.JFrame(), true);
    dialog.addWindowListener(new java.awt.event.WindowAdapter() {
        @Override
        public void windowClosing(java.awt.event.WindowEvent e) {
            System.exit(0);
        }
    });
    dialog.setVisible(true);
});
```

Gestión de Áreas:

Código:

```
public GestionAreas(java.awt.Frame parent, boolean modal) {
    super(parent, modal);
    initComponents();
    this.setLocationRelativeTo(null);
    this.getContentPane().setLayout(new GridBagLayout());
    table_area.getTableHeader().setReorderingAllowed(false);
}

private void btn_nuevoActionPerformed(java.awt.event.ActionEvent evt) {

}

private void btn_modificarActionPerformed(java.awt.event.ActionEvent evt) {

}

private void btn_regresarActionPerformed(java.awt.event.ActionEvent evt) {
    dispose();
}

private void btn_buscar_responsablActionPerformed(java.awt.event.ActionEvent evt) {
    new Buscar_Recurso_Humano(this, true).setVisible(true);
}

java.awt.EventQueue.invokeLater(() -> {
    GestionAreas dialog = new GestionAreas
        (new javax.swing.JFrame(), true);
    dialog.addWindowListener(new java.awt.event.WindowAdapter() {
        @Override
        public void windowClosing(java.awt.event.WindowEvent e) {
            System.exit(0);
        }
    });
    dialog.setVisible(true);
});
}
```

Búsqueda de Recurso Humano:

Búsqueda de Recurso Humano

☐ Cédula ☐ Nombres y Apellidos

Cédula	Nombres	Apellidos	Correo

[Search] [Delete]

Código:

```
public Buscar_Recurso_Humano(java.awt.Dialog parent, boolean modal) {
    super(parent, modal);
    initComponents();
    this.getContentPane().setLayout (new GridBagLayout());
    this.setLocationRelativeTo(null);
    tabla_persona.getTableHeader().setReorderingAllowed(false);
}

private void btn_regresarActionPerformed(java.awt.event.ActionEvent evt) {
    dispose();
}

private void btn_modificar2ActionPerformed(java.awt.event.ActionEvent evt) {
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    java.awt.EventQueue.invokeLater(() -> {
        Buscar_Recurso_Humano dialog = new Buscar_Recurso_Humano
            (new javax.swing.JDialog(), true);
        dialog.addWindowListener(new java.awt.event.WindowAdapter() {
            @Override
            public void windowClosing(java.awt.event.WindowEvent e) {
                System.exit(0);
            }
        });
        dialog.setVisible(true);
    });
}
```

Control de recursos Humanos:



Código:

```
public Control_Recursos_Humanos(java.awt.Frame parent, boolean modal) {
    super(parent, modal);
    initComponents();
    this.setLocationRelativeTo(null);
    this.getContentPane().setLayout (new GridBagLayout());
}

    java.awt.EventQueue.invokeLater(() -> {
        Control_Recursos_Humanos dialog = new Control_Recursos_Humanos
            (new javax.swing.JFrame(), true);
        dialog.addWindowListener(new java.awt.event.WindowAdapter() {
            @Override
            public void windowClosing(java.awt.event.WindowEvent e) {
                System.exit(0);
            }
        });
        dialog.setVisible(true);
    });
}
```

Listado de Activos:

Código:

```
public class Solicitar_Mantenimiento extends javax.swing.JDialog {
    DefaultTableModel model;
    Capa_Mensajes.Mensajes mensajes = new Capa_Mensajes.Mensajes();
    Capa_ConexionBD.Conexion conexion = new Capa_ConexionBD.Conexion();
}
```

```

public Solicitar_Mantenimiento(java.awt.Frame parent, boolean modal) {
    super(parent, modal);
    initComponents();
    this.setLocationRelativeTo(null);
    this.getContentPane().setLayout (new GridBagLayout());
}

public void combo_Tipo_Area() {
    combo_tipo_area.removeAllItems();
    combo_tipo_area.addItem("Laboratorio");
    combo_tipo_area.addItem("Administrativo");
    combo_tipo_area.addItem("Docentes");
}

public void Detalles_Busqueda() {
    conexion.crearConexion();
    String msj = "Manco";
    String[] var1 = new String[10];
    String[] var2 = {"Tipo", "Marca", "Procesador", "Memoria", "Disco Duro",
        "Modelo", "Serie", "Costo", "Fecha Compra", "Cod. Institucional"};
    model = new DefaultTableModel(null, var2);
    String sql = "select idTipo_activo,marca_activo,procesador_activo,"
        + "memoria_activo,discoduro_activo,modelo_activo,serie_activo,"
        + "costo_activo,fechacompra_activo,"
        + "codigointernoinstitucional_activo from tmovactcon";
    ResultSet rs = conexion.ejecutarSQLSelect(sql);
    try {
        while (rs.next()) {
            var1[0] = rs.getString(1);
            var1[1] = rs.getString(2);
            var1[2] = rs.getString(3);
            var1[3] = rs.getString(4);
            var1[4] = rs.getString(5);
            var1[5] = rs.getString(6);
            var1[6] = rs.getString(7);
            var1[7] = rs.getString(8);
            var1[8] = rs.getString(9);
            var1[9] = rs.getString(10);
            model.addRow(var1);
            tabla_lista_activos.setModel(model);
        }
    } catch (Exception ex) {
        mensajes.msg_advertencia(msj);
    }
}

private void btn_regresarActionPerformed(java.awt.event.ActionEvent evt) {
    dispose();
}

private void btn_asignarActionPerformed(java.awt.event.ActionEvent evt) {
    new Registro_Solicitud_Mantenimiento(this, true).setVisible(true);
}

private void combo_tipo_areaFocusGained(java.awt.event.FocusEvent evt) {
    if ("Seleccionar...".equals
        (combo_tipo_area.getSelectedItem().toString())) {
        combo_Tipo_Area();
    }
}

```

Registro de Solicitud de mantenimiento:

Solicitud de Mantenimiento

Detalles Búsqueda

Cédula Nombres y Apellidos

Código de oficio Fecha de Solicitud

Detalle

Tipo	Marca	Procesador	Memoria	Disco Duro	Modelo	Serie	Cod. Institución...

Código:

```
public Registro_Solicitud_Mantenimiento(java.awt.Dialog parent,
    boolean modal) {
    super(parent, modal);
    initComponents();
    this.setLocationRelativeTo(null);
    this.getContentPane().setLayout(new GridBagLayout());
    tabla_solicitud_mantenimiento.getTableHeader().setReorderingAllowed(
        false);
}

private void btn_regresarActionPerformed(java.awt.event.ActionEvent evt) {
    dispose();
}

java.awt.EventQueue.invokeLater(() -> {
    Registro_Solicitud_Mantenimiento dialog =
        new Registro_Solicitud_Mantenimiento
            (new javax.swing.JDialog(), true);
    dialog.addWindowListener(new java.awt.event.WindowAdapter() {
        @Override
        public void windowClosing(java.awt.event.WindowEvent e) {
            System.exit(0);
        }
    });
    dialog.setVisible(true);
});
```

Control de Solicitudes de mantenimiento:

The screenshot shows a web application titled "Control de Solicitudes de Mantenimiento". At the top, there are three buttons: "Nuevo", "Imprimir", and "Regresar". Below these, there are several input fields and dropdown menus for filtering requests. The fields include "Código Institucional", "Fecha de solicitud", "Estado", "Tipo de Área", "Descripción", and "RR.HH.". The "Estado" field has three radio button options: "Pendiente", "En Proceso", and "Realizado". A "Buscar" button with a magnifying glass icon is located to the right of the filters. Below the filters is a table with the following headers: "Código Interno", "Fecha Solicitud", "Cédula Solicitante", "Solicitante", and "Motivos". The table body is currently empty. To the right of the table, there is a vertical sidebar with five icons: a pencil (edit), a monitor (view), a person (user), a folder (list), and a trash can (delete).

Código:

```
public Control_solicitudes_Mantenimiento(java.awt.Frame parent,
    boolean modal) {
    super(parent, modal);
    initComponents();
    this.getContentPane().setLayout (new GridBagLayout());
    this.setLocationRelativeTo(null);
    tabla_solicitudes_mantenimiento.getTableHeader().setReorderingAllowed
        (false);
}

private void btn_buscarActionPerformed(java.awt.event.ActionEvent evt) {

}

private void btn_regresarActionPerformed(java.awt.event.ActionEvent evt) {
    dispose();
}

java.awt.EventQueue.invokeLater(() -> {
    Control_solicitudes_Mantenimiento dialog =
        new Control_solicitudes_Mantenimiento
            (new javax.swing.JFrame(), true);
    dialog.addWindowListener(new java.awt.event.WindowAdapter() {
        @Override
        public void windowClosing(java.awt.event.WindowEvent e) {
            System.exit(0);
        }
    });
    dialog.setVisible(true);
});
});
```

Registro Encargado de Mantenimiento:

Guardar Actualizar Regresar

Registro Encargado de Mantenimiento

Apellidos

Cédula	Nombres	Apellidos
--------	---------	-----------

Fecha de envío a Mantenimiento

Código:

```
public Registro_datos_encargado_mantenimiento(java.awt.Dialog parent,
        boolean modal) {
    super(parent, modal);
    initComponents();
    //this.getContentPane().setLayout (new GridBagLayout());
    this.setLocationRelativeTo(null);
    table_nombres_mantenimiento.getTableHeader().setReorderingAllowed
        (false);
}

private void btn_guardarActionPerformed(java.awt.event.ActionEvent evt) {

}

private void btn_regresarActionPerformed(java.awt.event.ActionEvent evt) {
    dispose();
}

java.awt.EventQueue.invokeLater(() -> {
    Registro_datos_encargado_mantenimiento dialog =
        new Registro_datos_encargado_mantenimiento
            (new javax.swing.JDialog(), true);
    dialog.addWindowListener(new java.awt.event.WindowAdapter() {
        @Override
        public void windowClosing(java.awt.event.WindowEvent e) {
            System.exit(0);
        }
    });
    dialog.setVisible(true);
});
```

Listado de entregas:

Código:

```
public ControlEntrega(java.awt.Frame parent, boolean modal) {
    super(parent, modal);
    initComponents();
    this.setLocationRelativeTo(null);
    tabla_solicitudes.getTableHeader().setReorderingAllowed(false);
}

private void rbtn_fecha_solicitudActionPerformed(java.awt.event.ActionEvent evt) {
}

private void btn_modificarActionPerformed(java.awt.event.ActionEvent evt) {
    new RegistroEntregaActivos(this, true).setVisible(true);
}

private void btn_buscar_responsablActionPerformed(java.awt.event.ActionEvent evt) {
    new Buscar_Recurso_Humano(this, true).setVisible(true);
}

private void btn_entregarActionPerformed(java.awt.event.ActionEvent evt) {
    new RegistroEntregaActivos(this, true).setVisible(true);
}

java.awt.EventQueue.invokeLater(() -> {
    ControlEntrega dialog = new ControlEntrega
        (new javax.swing.JFrame(), true);
    dialog.addWindowListener(new java.awt.event.WindowAdapter() {
        @Override
        public void windowClosing(java.awt.event.WindowEvent e) {
            System.exit(0);
        }
    });
});
```

```

    }
    });
    dialog.setVisible(true);
});

```

Registro de Entregas:

Códigos:

```

public RegistroEntregaActivos(java.awt.Dialog parent, boolean modal) {
    super(parent, modal);
    initComponents();
}

private void btn_buscar_personaActionPerformed(java.awt.event.ActionEvent evt) {
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
    dispose();
}

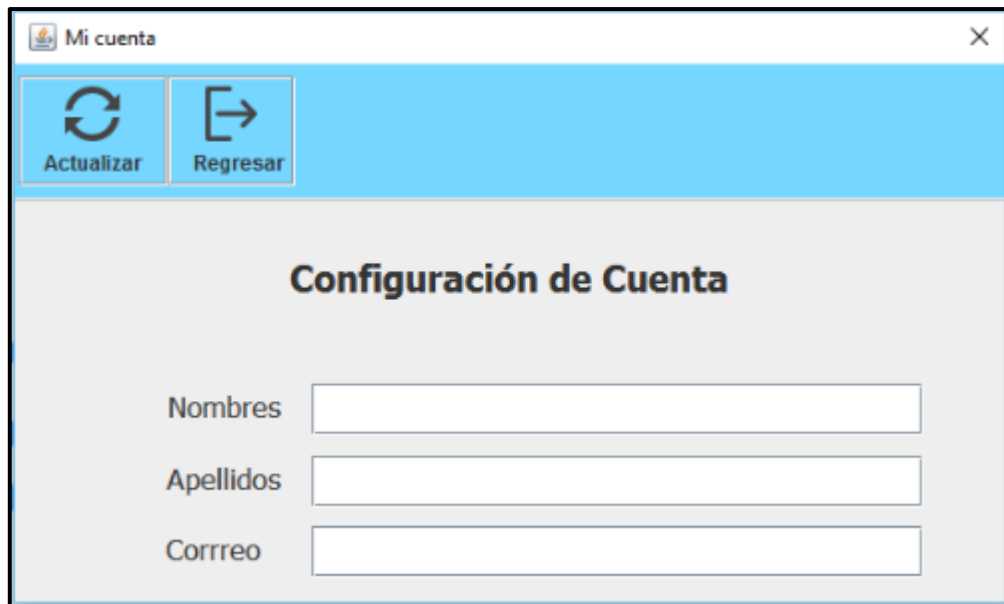
private void btn_buscar_responsablActionPerformed(java.awt.event.ActionEvent evt) {
    new Buscar_Recurso_Humano(this, true).setVisible(true);
}

java.awt.EventQueue.invokeLater(() -> {
    RegistroEntregaActivos dialog =
        new RegistroEntregaActivos
            (new javax.swing.JDialog(), true);
    dialog.addWindowListener(new java.awt.event.WindowAdapter() {
        @Override
        public void windowClosing(java.awt.event.WindowEvent e) {
            System.exit(0);
        }
    });
});

```

```
    }  
    });  
    dialog.setVisible(true);  
});
```

Configuración de cuenta:



Código:

```
public MiCuenta(java.awt.Frame parent, boolean modal) {  
    super(parent, modal);  
    initComponents();  
}  
  
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
    dispose();  
}  
  
java.awt.EventQueue.invokeLater(new Runnable() {  
    public void run() {  
        MiCuenta dialog = new MiCuenta(new javax.swing.JFrame(), true);  
        dialog.addWindowListener(new java.awt.event.WindowAdapter() {  
            @Override  
            public void windowClosing(java.awt.event.WindowEvent e) {  
                System.exit(0);  
            }  
        });  
        dialog.setVisible(true);  
    }  
});
```


Interfaz de Configuración de cuenta Inicial

Configuración Inicial

Hola, para comenzar ingrese los datos de la institución

Nombre

Correo

Direccion

Ingrese los datos del administrador

Correo:

Clave de acceso:

Confirmar clave:

Cancelar Finalizar

Código:

```
public Configuracion_Inicial() {  
    initComponents();  
}  
  
private void btn_finalizarActionPerformed(java.awt.event.ActionEvent evt) {  
}  
  
private void txt_nombre_institucionActionPerformed(java.awt.event.ActionEvent evt) {  
}  
  
private void txt_clave_usuarioActionPerformed(java.awt.event.ActionEvent evt) {  
}  
  
java.awt.EventQueue.invokeLater(new Runnable() {  
    public void run() {  
        new Configuracion_Inicial().setVisible(true);  
    }  
});
```