

GOI ESKOLA
POLITEKNIKOA

ESCUELA
POLITÉCNICA
SUPERIOR



MEDIATOR – PATRONES DE DISEÑO

2.CURSO DE GRADO DE INGENIERÍA INFORMÁTICA

ANÁLISIS Y DISEÑO DEL SOFTWARE

AUTOR: Aitor Landa

En Arrasate, a 5 de junio de 2019

Tabla de contenido

Tabla de contenido.....	1
Tabla de Ilustraciones	2
1. INTRODUCCIÓN	3
2. USO DEL PATRÓN MEDIATOR.....	4
3. CÓDIGO	11

Tabla de Ilustraciones

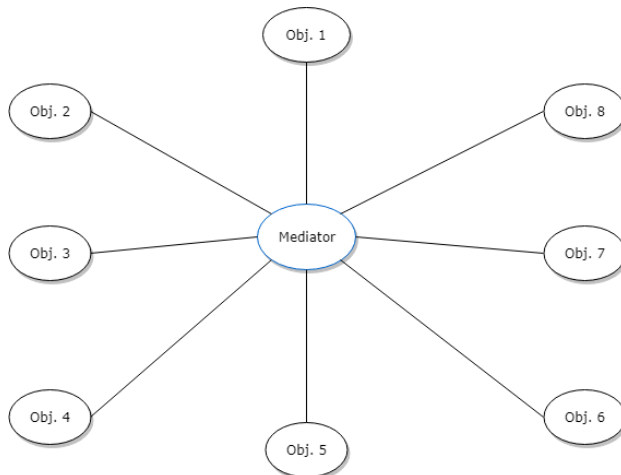
1.1 EJEMPLO GRÁFICO DE INTERACCIÓN DE OBJETOS CON MEDIATOR.....	3
1.2 EJEMPLO GRÁFICO DE INTERACCIÓN DE OBJETOS SIN MEDIATOR	3
2.1 DIAGRAMA CASOS DE USO	4
2.2 VISUALIZARBUSQUEDA SIN DATOS ADICIONALES	8
2.3 VISUALIZARBUSQUEDA CON DATOS ADICIONALES	9
2.4 DIAGRAMA DE CLASES	9
3.1 IMPLEMENTACIÓN DE MEDIADOR EN VISUALIZARBUSQUEDA.JAVA	11
3.2 CLASE DEL MEDIADOR MEDIATOR.JAVA	12
3.3 CLASE ABSTRACTA BUSQUEDA.JAVA.....	12
3.4 CLASE BUSQUEDAACTIVIDAD 1/2	13
3.5 CLASE BUSQUEDAACTIVIDAD 2/2	14
3.6 CLASE BUSQUEDAPISO 1/2.....	14
3.7 CLASE BUSQUEDAPISO 2/2.....	15
3.8 CLASE BUSQUEDAPERSONA 1/2	16
3.9 CLASE BUSQUEDAPERSONA 2/2	16

1. INTRODUCCIÓN

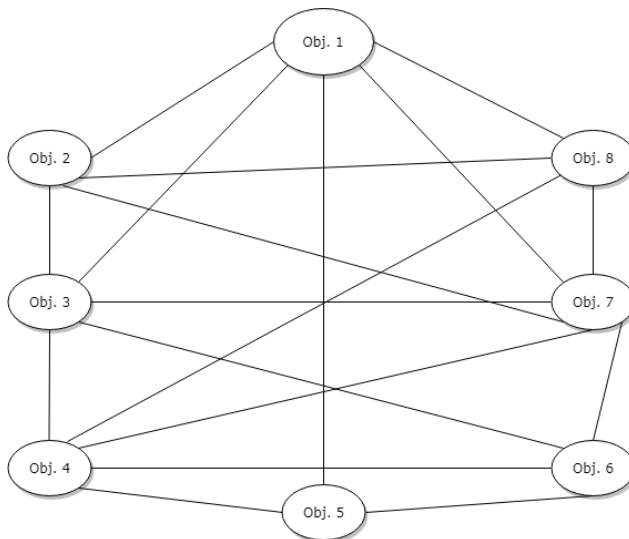
El patrón de diseño *Mediator*, el cual se encuentra en el subgrupo de patrones de comportamiento, consiste en la creación de un objeto para el control y la gestión de las relaciones mutuas de un conjunto de objetos. Con ello se consigue que los elementos del conjunto de objetos no tengan que conocerse e interactuar directamente entre sí.

Mediator pretende evitar que el código sea propenso a errores y de mala calidad, dado que la programación orientada a objetos suele acarrear la interacción entre diferentes objetos, puede ocurrir que un objeto deba conocer a todos los demás objetos de la aplicación, de modo que si uno de ellos fallara podría generar un fallo en cadena.

Con el fin de fomentar la modularidad, y la reutilización de objetos se implementa el patrón de diseño *Mediator*.



1.1 Ejemplo gráfico de interacción de objetos con Mediator

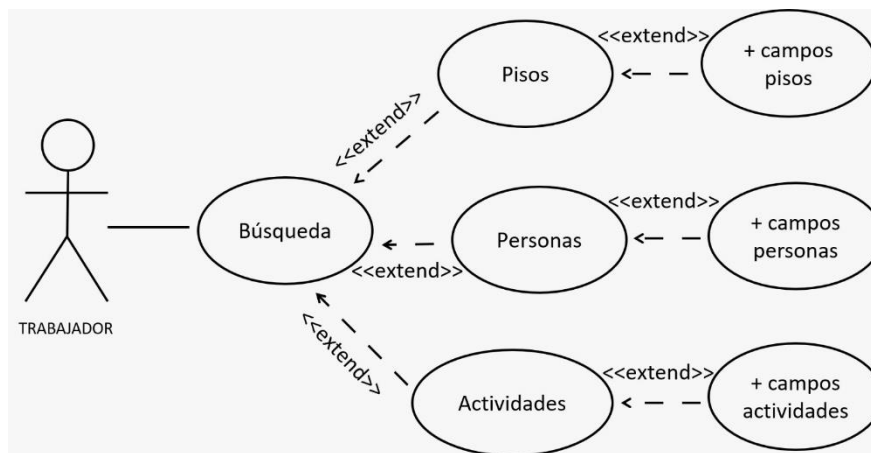


1.2 Ejemplo gráfico de interacción de objetos sin Mediator

2. USO DEL PATRÓN MEDIATOR

En la aplicación desarrollada se ha implementado *Mediator* para el control de la interfaz de *Búsqueda*. Nuestro programa permite buscar en la base de datos diferentes objetos de la aplicación (*Persona*, *Piso*, *Actividad*), y en dicha interfaz se distinguen 2 paneles (*JPanel*) dinámicos que alternan los datos de diferentes objetos mediante los cuales se hacen consultas a la base de datos. Estos paneles varían según la opción marcada a través de los botones superiores de la pantalla, el objeto mediator se encarga de cargar, cambiar, ocultar y visibilizar estos paneles de manera que se adapten a cada opción.

El planteamiento del código a implementar se ha obtenido en la fase de análisis donde se ha obtenido el siguiente caso de uso.



2.1 Diagrama casos de uso

Cuyos escenarios se describen a continuación:

Código	C.U.1.1
Nombre:	Más campos piso
Descripción:	Amplía la búsqueda mediante datos de los Pisos
Actor:	Empleado/Voluntario
Curso normal	Curso alternativo
1. Mostrar datos adicionales pisos.	1.1. Reduce búsqueda pisos

Código:	C.U.2.1
Nombre:	Más datos personas
Descripción:	Amplía los datos para la búsqueda Personas
Actor:	Empleado/Voluntario
Curso normal	Curso alternativo
1. Mostrar datos adicionales de búsqueda de personas	1.1. Reduce búsqueda personas

Código:	C.U.3.1
Nombre:	Más datos actividad
Descripción:	Amplía la búsqueda de datos de la Actividad
Actor:	Empleado/Voluntario
Curso normal	Curso alternativo
1. Mostrar datos adicionales de búsqueda de actividades	1.1. Reduce búsqueda actividades

Código	C.U.1.2
Nombre:	Búsqueda Pisos
Descripción:	Permite buscar pisos tras rellenar unos datos en un interfaz gráfica
Actor:	Empleado/Voluntario
Curso normal	Curso alternativo

1. Mostrar datos básicos del piso	2.1 Datos erróneos. Mostrar mensaje de error.
2. Comprobar datos	4.1 Búsqueda fallida. Mostrar mensaje de error. Vaciar campos datos
3. Extiende C.U.1.1 + Campos pisos	
4. Realizar búsqueda	
5. Fin del caso de uso	

Código:	C.U.2.2
Nombre:	Búsqueda Personas
Descripción:	Permite buscar personas tras rellenar una serie de campos en una interfaz gráfica.
Actor:	Empleado/Voluntario
Curso normal	Curso alternativo
1. Mostrar datos básicos de la persona.	2.1 Datos erróneos. Mostrar mensaje de error.
2. Comprobar datos	4.1 Búsqueda fallida. Mostrar mensaje de error. Vaciar campos datos
3. Extiende C.U.2.1 + Campos personas	
4. Realizar búsqueda	
5. Fin del caso de uso	

Código:	C.U.3.2
Nombre:	Búsqueda Actividades
Descripción:	Permite buscar actividades una vez rellenados diferentes datos
Actor:	Empleado/Voluntario
Curso normal	Curso alternativo
1. Mostrar datos básicos de la actividad.	2.1 Datos erróneos. Mostrar mensaje de error.
2. Comprobar datos	4.1 Búsqueda fallida. Mostrar mensaje de error. Vaciar campos datos
3. Extiende C.U.2.1 + Campos actividades	
4. Realizar búsqueda	
5. Fin del caso de uso	

Código:	C.U.1
Nombre:	Búsqueda
Descripción:	Interfaz para realizar búsquedas en la base de datos.
Actor:	Empleado/Voluntario
Curso normal	Curso alternativo
1. Iniciar pantalla de búsqueda.	
2. Extiende C.U.1.2 Búsqueda personas	

3. Extiende C.U.2.2 Búsqueda actividades	
4. Extiende C.U.3.2 Búsqueda pisos	
5. Fin del caso de uso	

El objeto Mediator, contiene los siguientes métodos:

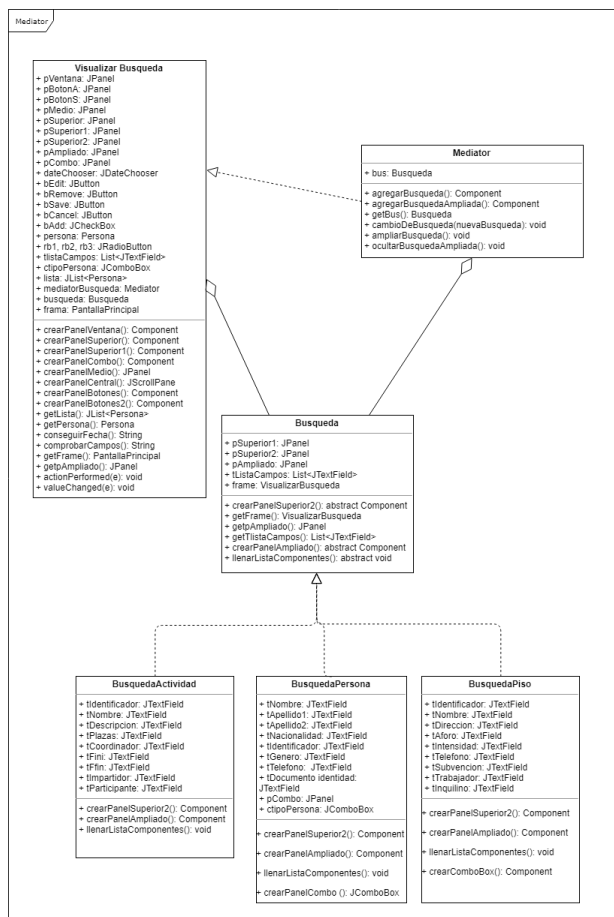
- agregarBusqueda: permite crear el respectivo panel de búsqueda para el objeto seleccionado.
- agregarBuquedaAmpliada: crea un panel de datos adicional para el objeto seleccionado.
- cambioDeBusqueda: cambia el objeto que busca (*Persona*, *Piso*, *Actividad*).
- ampliarBusqueda: hace visible el panel de datos adicionales.
- ocultarBusquedaAmpliada: oculta el panel de datos adicionales.

Para ampliar las opciones de los datos mediante los cuales se hacen las búsquedas, se emplea un *checkbox* que al clicar sobre él visibiliza el panel de datos adicionales (ampliados), y al desclicarlo se oculta el panel.

2.2 VisualizarBusqueda sin datos adicionales

2.3 VisualizarBusqueda con datos adicionales

En la siguiente ilustración se expone el diagrama de las clases que tienen relación con el patrón de diseño *Mediator*.



2.4 Diagrama de clases

Este diagrama incluye las clases siguientes:

- *Busqueda* es una clase abstracta que incluye los elementos comunes a todos los controles gráficos.
- *BusquedaActividad*, *BusquedaPersona*, *BusquedaPiso* son las subclases concretas de *Busqueda* que implementan los métodos *crearPanelSuperior2*, *crearPanelAmpliado*, *llenarListaComponentes*.
- *Mediator* es la clase que realiza la función de mediador.

3. CÓDIGO

A continuación se adjuntan imágenes del código donde ha sido implementado el patrón de diseño *Mediator* y las clases relacionadas.

```
@Override
public void actionPerformed(ActionEvent e) {
    if (e.getActionCommand().equals(BUSQUEDA)) {
        if (rb1.isSelected()) {
            try {
                switch ((String)ctipoPersona.getSelectedItem()) {
                    case "BENEFICIARIO":
                        lista.setModel((ListModel<Persona>) DAOBeneficiario.buscarPersona(comprobarCampos());
                        break;
                    default:
                        lista.setModel((ListModel<Persona>) DAOTrabajador.buscarTrabajador(comprobarCampos());
                        break;
                }
            } catch (SQLException e1) {
                e1.printStackTrace();
            }
        }
    }

    if (e.getActionCommand().equals("Cancel")) {
        this.dispose();
    }

    if (e.getActionCommand().equals("PERSONA") || e.getActionCommand().equals("PISO") || e.getActionCommand().equals("ACT")) {
        bAdd.setSelected(false);
        mediatorBusqueda.ocultarBusquedaAmpliada();
        mediatorBusqueda.cambioDeBusqueda(e.getActionCommand());
        pSuperior.remove(pSuperior2);
        pSuperior.remove(pAmpliado);
        pSuperior2 = (JPanel) mediatorBusqueda.agregarBusqueda();
        pAmpliado = (JPanel) mediatorBusqueda.agregarBusquedaAmpliada();
        pSuperior.add(pSuperior2);
        pSuperior.add(pAmpliado);
        pSuperior.revalidate();
    }

    if (e.getActionCommand().equals("add")) {
        if (!lbAdd.isSelected()) {
            mediatorBusqueda.ocultarBusquedaAmpliada();
        } else {
            mediatorBusqueda.ampliarBusqueda();
        }
    }
}
```

3.1 Implementación de mediador en VisualizarBusqueda.java

```

Mediator.java
package vistas;
2
3*import java.awt.Component;
9
10 public class Mediator {
11     Busqueda bus;
12
13     public Mediator(Busqueda busqueda) {
14         bus = busqueda;
15     }
16
17     public Component agregarBusqueda() {
18         return bus.crearPanelSuperior2();
19     }
20
21     public Component agregarBusquedaAmpliada() {
22         return bus.crearPanelAmpliado();
23     }
24
25     public Busqueda getBus() {
26         return bus;
27     }
28
29     public void cambioDeBusqueda(String busquedaNueva) {
30         switch (busquedaNueva) {
31             case "PISO":
32                 bus = new BusquedaPiso(bus.getFrame());
33                 break;
34             case "ACT":
35                 bus = new BusquedaActividad(bus.getFrame());
36                 break;
37             case "PERSONA":
38                 bus = new BusquedaPersona(bus.getFrame());
39                 break;
40         }
41     }
42
43     public void ampliarBusqueda() {
44         bus.getpAmpliado().setVisible(true);
45     }
46
47     public void ocultarBusquedaAmpliada() {
48         bus.getpAmpliado().setVisible(false);
49     }
50 }
51

```

3.2 Clase del mediador Mediator.java

```

Busqueda.java
1 package vistas;
2
3*import java.awt.Component;
7
8 public abstract class Busqueda extends Component{
9
10     public JPanel pSuperior1, pAmpliado;
11     public JPanel pSuperior2;
12
13     List<JTextField> tlistaCampos;
14     VisualizarBusqueda frame;
15
16     public Busqueda(VisualizarBusqueda visualizarBusqueda) {
17         frame = visualizarBusqueda;
18     }
19
20     public abstract Component crearPanelSuperior2();
21
22     public VisualizarBusqueda getFrame() {
23         return frame;
24     }
25
26     public JPanel getpAmpliado() {
27         return pAmpliado;
28     }
29
30     public List<JTextField> getTlistaCampos() {
31         return tlistaCampos;
32     }
33
34     public abstract Component crearPanelAmpliado();
35
36     public abstract void llenarListaComponentes();
37 }
38

```

3.3 Clase abstracta Busqueda.java

```

1 package vistas;
2
3 import java.awt.Component;
12
13 public class BusquedaActividad extends Busqueda{
14     JTextField tIdentificador, tNombre, tDescripcion, tPlazas, tCoordinador;
15     JTextField tFin, tFfin, tImpartidor, tParticipante;
16 public BusquedaActividad(VisualizarBusqueda frame) {
17     super(frame);
18     crearPanelSuperior2();
19     crearPanelAmpliado();
20 }
21
22 @Override
23 public Component crearPanelSuperior2() {
24     pSuperior2 = new JPanel(new GridLayout(2,6));
25     JLabel lId = new JLabel("Identificador");
26     tIdentificador = new JTextField(40);
27     tIdentificador.setName("identificador");
28     JLabel lNombre = new JLabel("Nombre");
29     tNombre = new JTextField(40);
30     tNombre.setName("nombre");
31     JLabel lDescripcion = new JLabel("Descripción");
32     tDescripcion = new JTextField(40);
33     tDescripcion.setName("descripcion");
34     JLabel lPlazas = new JLabel("Plazas");
35     tPlazas = new JTextField(40);
36     tPlazas.setName("plazas");
37     JLabel lCoordinador = new JLabel("Coordinador");
38     tCoordinador = new JTextField(40);
39     tCoordinador.setName("coordinador");
40     JLabel lEspacio = new JLabel("");
41
42     pSuperior2.add(lId);
43     pSuperior2.add(tIdentificador);
44     pSuperior2.add(Box.createHorizontalGLue());
45     pSuperior2.add(lNombre);
46     pSuperior2.add(tNombre);
47     pSuperior2.add(Box.createHorizontalGLue());
48     pSuperior2.add(lDescripcion);
49     pSuperior2.add(tDescripcion);
50     pSuperior2.add(lPlazas);
51     pSuperior2.add(tPlazas);
52     pSuperior2.add(Box.createHorizontalGLue());
53     pSuperior2.add(lCoordinador);
54     pSuperior2.add(tCoordinador);
55 }

```

3.4 Clase BusquedaActividad 1/2

```

@Override
public Component crearPanelAmpliado() {
    pAmpliado = new JPanel(new GridLayout(2,4));
    JLabel lFini = new JLabel("Fecha Inicio");
    tFini = new JTextField(40);
    tFini.setName("fini");
    JLabel lFfin = new JLabel("Fecha Fin");
    tFfin = new JTextField(40);
    tFfin.setName("ffin");
    JLabel lImpartidor = new JLabel("Impartidor");
    tImpartidor = new JTextField(40);
    tImpartidor.setName("impartidor");
    JLabel lParticipante = new JLabel("Participante");
    tParticipante = new JTextField(40);
    tParticipante.setName("participante");

    pAmpliado.add(lFini);
    pAmpliado.add(tFini);
    pAmpliado.add(Box.createHorizontalGLue());
    pAmpliado.add(lFfin);
    pAmpliado.add(tFfin);
    pAmpliado.add(Box.createHorizontalGLue());
    pAmpliado.add(lImpartidor);
    pAmpliado.add(tImpartidor);
    pAmpliado.add(Box.createHorizontalGLue());
    pAmpliado.add(lParticipante);
    pAmpliado.add(tParticipante);
    pAmpliado.setVisible(false);
    return pAmpliado;
}

@Override
public void llenarListaComponentes() {
    tlistaCampos = new ArrayList<>();
    tlistaCampos.add(tIdentificador);
    tlistaCampos.add(tNombre);
    tlistaCampos.add(tDescripcion);
    tlistaCampos.add(tPlazas);
    tlistaCampos.add(tCoordinador);
    tlistaCampos.add(tParticipante);
    tlistaCampos.add(tCoordinador);
}
}

```

3.5 Clase BusquedaActividad 2/2

```

public class BusquedaPiso extends Busqueda{
    JTextField tIdentificador, tNombre, tDireccion, tAforo, tIntensidad, tTelefono, tSubvencion, tTrabajador, tInquilino;
    JComboBox<String> cTipo;
    final static String[] tiposPisos = {"HOMBRES", "MUJERES", "MIXTO"};

    public BusquedaPiso(VisualizarBusqueda frame) {
        super(frame);
        crearPanelSuperior2();
        crearPanelAmpliado();
    }

    @Override
    public Component crearPanelSuperior2() {
        pSuperior2 = new JPanel(new GridLayout(2,6));
        JLabel lIdentificado = new JLabel("Identificador");
        tIdentificador = new JTextField(40);
        tIdentificador.setName("id");
        JLabel lNombre = new JLabel("Nombre");
        tNombre = new JTextField(40);
        tNombre.setName("nombrePrograma");
        JLabel lDireccion = new JLabel("Dirección");
        tDireccion = new JTextField(40);
        tDireccion.setName("direccion");
        JLabel lAforo = new JLabel("Aforo");
        tAforo = new JTextField(40);
        tAforo.setName("aforo");
        JLabel lTipo = new JLabel("Tipo");
        JLabel lIntensidad = new JLabel("Intensidad");
        tIntensidad = new JTextField(40);
        tIntensidad.setName("intensidad");

        pSuperior2.add(lIdentificado);
        pSuperior2.add(tIdentificador);
        pSuperior2.add(Box.createHorizontalGLue());
        pSuperior2.add(lNombre);
        pSuperior2.add(tNombre);
        pSuperior2.add(Box.createHorizontalGLue());
        pSuperior2.add(lDireccion);
        pSuperior2.add(tDireccion);
        pSuperior2.add(lAforo);
        pSuperior2.add(tAforo);
        pSuperior2.add(Box.createHorizontalGLue());
        pSuperior2.add(lTipo);
        pSuperior2.add(crearComboBox());
        pSuperior2.add(Box.createHorizontalGLue());
    }
}

```

3.6 Clase BusquedaPiso 1/2


```

        pSuperior2.add(Box.createHorizontalGlue());
        pSuperior2.add(lTipo);
        pSuperior2.add(crearComboBox());
        pSuperior2.add(Box.createHorizontalGlue());
        pSuperior2.add(lIntensidad);
        pSuperior2.add(tIntensidad);
        return pSuperior2;
    }

    private Component crearComboBox() {
        cTipo = new JComboBox<>(tiposPisos);
        cTipo.setSelectedItem(null);
        return cTipo;
    }

    @Override
    public Component crearPanelAmpliado() {
        pAmpliado = new JPanel(new GridLayout(2,6));
        JLabel lTelefono = new JLabel("Telefono");
        tTelefono = new JTextField(40);
        tTelefono.setName("telefono");
        JLabel lSubvencion = new JLabel("Subvencion");
        tSubvencion = new JTextField(40);
        tSubvencion.setName("subvencion");
        JLabel lInquilino = new JLabel("Inquilino");
        tInquilino = new JTextField(40);
        tInquilino.setName("inquilino");
        JLabel lEspacio = new JLabel("");

        pAmpliado.add(lTelefono);
        pAmpliado.add(tTelefono);
        pAmpliado.add(Box.createHorizontalGlue());
        pAmpliado.add(lSubvencion);
        pAmpliado.add(tSubvencion);
        pAmpliado.add(Box.createHorizontalGlue());
        pAmpliado.add(lInquilino);
        pAmpliado.add(tInquilino);
        pAmpliado.add(Box.createHorizontalGlue());
        pAmpliado.add(lEspacio);
        pAmpliado.add(Box.createHorizontalGlue());
        pAmpliado.add(lEspacio);
        pAmpliado.setVisible(false);
        return pAmpliado;
    }

    @Override
    public void llenarListaComponentes() {
        tlistaCampos = new ArrayList<>();
        tlistaCampos.add(tDireccion);
        tlistaCampos.add(tIdentificador);
        tlistaCampos.add(tIntensidad);
        tlistaCampos.add(tNombre);
        tlistaCampos.add(tSubvencion);
        tlistaCampos.add(tTelefono);
        tlistaCampos.add(tInquilino);
        tlistaCampos.add(tTrabajador);
        tlistaCampos.add(tAforo);
    }
}

```

3.7 Clase BusquedaPiso 2/2

```

package vistas;

import java.awt.Color;

public class BusquedaPersona extends Busqueda{

    public JTextField tNombre, tApellido1, tApellido2, tNacionalidad, tIdentificador, tGenero, tTelefono, tDocumentoIdentidad;
    public JPanel pCombo;
    private JComboBox<String> ctipoPersona;
    public BusquedaPersona(VisualizarBusqueda visualizarBusqueda) {
        super(visualizarBusqueda);
        crearPanelSuperior2();
        crearPanelAmpliado();
    }

    @Override
    public Component crearPanelSuperior2() {
        pSuperior2 = new JPanel(new GridLayout(2, 6));
        JLabel lNombre = new JLabel("Nombre");
        tNombre = new JTextField(40);
        tNombre.setName("nombre");
        JLabel lApellido1 = new JLabel("Apellido 1");
        tApellido1 = new JTextField(40);
        tApellido1.setName("apellido1");
        JLabel lApellido2 = new JLabel("Apellido 2");
        tApellido2 = new JTextField(40);
        tApellido2.setName("apellido2");
        JLabel lNacionalidad = new JLabel("Nacionalidad");
        tNacionalidad = new JTextField(40);
        tNacionalidad.setName("nacionalidad");
        JLabel lId = new JLabel("Identificador");
        tIdentificador = new JTextField(40);
        tIdentificador.setName("identificador");
        JLabel lComboBox = new JLabel("Tipo");

        pSuperior2.add(lId);
        pSuperior2.add(tIdentificador);
        pSuperior2.add(Box.createHorizontalGlue());
        pSuperior2.add(lNombre);
        pSuperior2.add(tNombre);
        pSuperior2.add(Box.createHorizontalGlue());
        pSuperior2.add(lApellido1);
        pSuperior2.add(tApellido1);
        pSuperior2.add(lApellido2);
        pSuperior2.add(tApellido2);
        pSuperior2.add(Box.createHorizontalGlue());
        pSuperior2.add(lNacionalidad);
        pSuperior2.add(tNacionalidad);
        pSuperior2.add(Box.createHorizontalGlue());
        pSuperior2.add(lComboBox);
        pSuperior2.add(crearPanelCombo());
        return pSuperior2;
    }
}

```

3.8 Clase BusquedaPersona 1/2

```

@Override
public Component crearPanelAmpliado() {
    pAmpliado = new JPanel(new GridLayout(2,4));
    JLabel lGenero = new JLabel("Genero");
    tGenero = new JTextField(40);
    tGenero.setName("genero");
    JLabel lTelefono = new JLabel("Telefono");
    tTelefono = new JTextField(40);
    tTelefono.setName("telefono");
    JLabel lDocumentoIdentidad = new JLabel("Documento Identidad");
    tDocumentoIdentidad = new JTextField(40);
    tDocumentoIdentidad.setName("docIdentidad");
    JLabel lEspacio = new JLabel("");

    pAmpliado.add(lTelefono);
    pAmpliado.add(tTelefono);
    pAmpliado.add(Box.createHorizontalGlue());
    pAmpliado.add(lGenero);
    pAmpliado.add(tGenero);
    pAmpliado.add(Box.createHorizontalGlue());
    pAmpliado.add(lDocumentoIdentidad);
    pAmpliado.add(tDocumentoIdentidad);
    pAmpliado.add(Box.createHorizontalGlue());
    pAmpliado.add(lEspacio);
    pAmpliado.add(Box.createHorizontalGlue());
    pAmpliado.add(lEspacio);
    pAmpliado.setVisible(false);
    return pAmpliado;
}

public Component crearPanelCombo() {

    ctipoPersona = new JComboBox<String>();

    ctipoPersona.setModel(new DefaultComboBoxModel<String>{Trabajador.tipos});
    ctipoPersona.setSelectedItem(null);
    ctipoPersona.setBackground(Color.WHITE);
    pCombo = new JPanel(new GridLayout(1,2));
    pCombo.add(ctipoPersona);

    return pCombo;
}

@Override
public void llenarListaComponentes() {
    tlistaCampos = new ArrayList<>();
    tlistaCampos.add(tIdentificador);
    tlistaCampos.add(tNombre);
    tlistaCampos.add(tApellido1);
    tlistaCampos.add(tApellido2);
    tlistaCampos.add(tNacionalidad);
    tlistaCampos.add(tDocumentoIdentidad);
    tlistaCampos.add(tGenero);
    tlistaCampos.add(tTelefono);
}
}

```

3.9 Clase BusquedaPersona 2/2