Documentacion Hotel Printer-Boi

Vamos a mostrar el archivo main:

Main.java

```java
/*
 * Created on 9 dic. 2020
 * @author Carlos
 * @version 1.0
 */
package swing_c_p02_MartinezSanchezCarlosJose;

// TODO: Auto-generated Javadoc
/**
 * The Class Main.
 */
public class Main {

	/**
	 * The main method.
	 * Created on 9 dic. 2020
	 *
	 * @author Carlos
	 * @version 1.0
	 * @param args the arguments
	 */
	public static void main(String[] args) {

		@SuppressWarnings("unused")
		Ventana v = new Ventana();
	}

}

```

Ahora mostramos la Ventana:

*Ventana.java

```java
/*
 * Created on 9 dic. 2020
 * @author Carlos
 * @version 1.0
 */
package swing_c_p02_MartinezSanchezCarlosJose;
import java.awt.Image;


// TODO: Auto-generated Javadoc
/**
 * The Class Ventana.
 */
public class Ventana extends JFrame implements ActionListener, KeyListener{

    /** Declaracion de atributos. */
    private static final long serialVersionUID = 1L;

    /** The barra menu. */
    private JMenuBar barraMenu;

    /** The ayuda. */
    private JMenu archivo, registro, ayuda;;

    /** The acerca de. */
    private JMenuItem salir, altaReservas, bajaReservas, acercaDe;

    /** The bt baja reservas. */
    private JButton btAltaReservas, btBajaReservas;

    /**
     * Instantiates a new ventana.
     *
     * @author Carlos
     * Created on 9 dic. 2020
     * @version 1.0
     */
    public Ventana() {

        /**
         * Declaracion del titulo de la ventana y de las medidas de la misma
         */
        super("Gestion Hotel Printer-Boi");
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(400, 200);
        setLocationRelativeTo(null);
```

Ventana.java

```java
        setResizable(false);
        this.setLayout(null);

        Image logo= new ImageIcon(getClass().getResource("recursos/anagrama.png")).getImage();
        setIconImage(logo);
        //-----------------------------------------------------------------------------------------

        //Imagenes
        Image imgBtAlta = new ImageIcon(getClass().getResource("recursos/add.png")).getImage();
        ImageIcon imgAltaUps = new ImageIcon(imgBtAlta.getScaledInstance(50, 50, Image.SCALE_SMOOTH));

        Image imgBtBaja = new ImageIcon(getClass().getResource("recursos/delete.png")).getImage();
        ImageIcon imgBajaUps = new ImageIcon(imgBtBaja.getScaledInstance(50, 50, Image.SCALE_SMOOTH));
        //-----------------------------------------------------------------------------------------

        //Inicializamos
        barraMenu= new JMenuBar();

        archivo= new JMenu("Archivo");
        registro= new JMenu("Registro");
        ayuda= new JMenu("Ayuda");

        salir= new JMenuItem("Salir");
        altaReservas= new JMenuItem("Alta Reservas");
        bajaReservas= new JMenuItem("Baja Reservas");
        acercaDe= new JMenuItem("Acerca De...");

        btAltaReservas= new JButton(imgAltaUps);
        btBajaReservas= new JButton(imgBajaUps);
        //-----------------------------------------------------------------------------------------

        //Damos medidas a los elementos
        barraMenu.setSize(400,25);
        btAltaReservas.setBounds(0, 25, 200, 150);
        btBajaReservas.setBounds(200, 25, 200, 150);
        //-----------------------------------------------------------------------------------------

        //Añadios los listeners correspondientes
        salir.addActionListener(this);
        altaReservas.addActionListener(this);
        bajaReservas.addActionListener(this);
        acercaDe.addActionListener(this);

        btAltaReservas.addActionListener(this);
        btBajaReservas.addActionListener(this);

        registro.setMnemonic(KeyEvent.VK_R);
```

Ventana.java

```java
        registro.setMnemonic(KeyEvent.VK_R);
        btAltaReservas.setMnemonic(KeyEvent.VK_A);
        btBajaReservas.setMnemonic(KeyEvent.VK_B);
        //-----------------------------------------------------------------------------------------------

        //Añadimos los componentes a los elementos del menu
        archivo.add(salir);
        registro.add(altaReservas);
        registro.add(bajaReservas);
        ayuda.add(acercaDe);

        //Añadimos los elementos del menu al menu
        barraMenu.add(archivo);
        barraMenu.add(registro);
        barraMenu.add(ayuda);

        add(barraMenu);
        add(btAltaReservas);
        add(btBajaReservas);
        //-----------------------------------------------------------------------------------------------

        //Mostramos el panel
        this.setVisible(true);
    }

    /**
     * Metodo que devuelve Key pressed.
     *
     * @param e the e
     */
    @Override
    public void keyPressed(KeyEvent e) {

    }

    /**
     * Metodo que devuelve Key released.
     *
     * @param e the e
     */
    @Override
    public void keyReleased(KeyEvent e) {

    }

    /**
     * Metodo que devuelve Key typed.
     *
     * @param e the e
     */
    @Override
    public void keyTyped(KeyEvent e) {

    }

    /**
     * Metodo que devuelve Action performed.
     *
     * @param e the e
     */
    @Override
    public void actionPerformed(ActionEvent e) {
        if(e.getSource()==salir)
            System.exit(0);

        if(e.getSource()==btBajaReservas || e.getSource()==bajaReservas)
            JOptionPane.showMessageDialog(null, "Esta opcion aun no está desarrollada");

        if(e.getSource()==acercaDe)
            JOptionPane.showMessageDialog(null, "Progrmama encargado de gestionar las reservas del hotel Printer-Boi");

        if(e.getSource()==btAltaReservas || e.getSource()==altaReservas) {
            @SuppressWarnings("unused")
            Dialogo d= new Dialogo();
        }

    }

}
```

Ahora mostramos el Dialogo:

Dialogo.java

```java
/*
 * Created on 9 dic. 2020
 * @author Carlos
 * @version 1.0
 */
package swing_c_p02_MartinezSanchezCarlosJose;

import java.awt.BorderLayout;

// TODO: Auto-generated Javadoc
/**
 * The Class Dialogo.
 */
public class Dialogo extends JDialog implements ActionListener {

    /** The Constant serialVersionUID. */
    private static final long serialVersionUID = 1L;

    /** The horizontal. */
    int horizontal=Toolkit.getDefaultToolkit().getScreenSize().width;

    /** The vertical. */
    int vertical=Toolkit.getDefaultToolkit().getScreenSize().height;

    /** The panel 1. */
    private Panel_1 panel1;

    /** The panel 2. */
    private Panel_2 panel2;

    /** The panel 3. */
    private Panel_3 panel3;

    /** The panel 4. */
    private Panel_4 panel4;

    /** The bt guardar. */
    private JButton btImprimirDocumento, btNuevo, btGuardar;

    /** The datos. */
    private static ArrayList<String> datos=new ArrayList<String>();

    /**
     * Instantiates a new dialogo.
     *
     * @author Carlos
     * Created on 9 dic. 2020
```

Dialogo.java

```java
     * @author Carlos
     * Created on 9 dic. 2020
     * @version 1.0
     */
    public Dialogo() {
        setSize(Toolkit.getDefaultToolkit().getScreenSize()); // Se obtiene la resolucion usando Toolkit
        setIconImage(new ImageIcon(getClass().getResource("recursos/anagrama.png")).getImage()); // Se añade el icono al
                                                                                                  // panel
        setResizable(false); // Hacemos que no se pueda redimensionar la ventana
        setTitle("Alta reservas"); // Se añade el titulo de la ventana
        setLayout(new BorderLayout(2,2));
        // ------------------------------------------------------------------------------------------

        // Imagenes
        Image imgPrintDocument = new ImageIcon(getClass().getResource("recursos/printDocument.png")).getImage();
        ImageIcon imgPrintDocumentUps = new ImageIcon(imgPrintDocument.getScaledInstance(50, 50, Image.SCALE_SMOOTH));

        Image imgNew = new ImageIcon(getClass().getResource("recursos/new.png")).getImage();
        ImageIcon imgNewUps = new ImageIcon(imgNew.getScaledInstance(50, 50, Image.SCALE_SMOOTH));

        Image imgSave = new ImageIcon(getClass().getResource("recursos/save.png")).getImage();
        ImageIcon imgSaveUps = new ImageIcon(imgSave.getScaledInstance(50, 50, Image.SCALE_SMOOTH));
        // ------------------------------------------------------------------------------------------

        panel1 = new Panel_1(); // El panel 1

        panel2 = new Panel_2(); // El panel 2
        panel2.setPreferredSize(new Dimension((int)(horizontal/3), vertical)); //Ajustamos su tamaño a un tercio del eje X
        panel2.setBackground(Color.GRAY);

        panel3 = new Panel_3(); // El panel 3

        panel4 = new Panel_4(); // El panel 4
        panel4.setPreferredSize(new Dimension((int)(horizontal/3), vertical)); //Ajustamos su tamaño a un tercio del eje X
        panel4.setBackground(Color.GRAY);

        JPanel botones=new JPanel(); //Panel donde meteremos los botones

        //Incializamos los botones
        btImprimirDocumento = new JButton(imgPrintDocumentUps);
        btNuevo = new JButton(imgNewUps);
        btGuardar = new JButton(imgSaveUps);
        //Los metemos en el panel de los botones
        botones.add(btImprimirDocumento);
        botones.add(btNuevo);
        botones.add(btGuardar);

        //Añadimos los listeners
        btImprimirDocumento.addActionListener(this);
        btNuevo.addActionListener(this);
        btGuardar.addActionListener(this);

        //Los añadimos al panel de Dialogo
        getContentPane().add(panel1, BorderLayout.NORTH);
        getContentPane().add(panel2, BorderLayout.WEST);
        getContentPane().add(panel3, BorderLayout.CENTER);
        getContentPane().add(panel4, BorderLayout.EAST);
        getContentPane().add(botones, BorderLayout.SOUTH);

        setVisible(true);
    }

    /**
     * Metodo que Realiza acciones segun los datos introducidos.
     *
     * El boton de Imprimir captura 2 excepciones personalizadas que heredan de IllegalArgumentException
     * Se realiza asi para controlar que se introducen todos los campos necesarios.
     *
     * El boton Nuevo llama a los metodos reset de los 3 paneles
     * @param e the ActionEvent
     */
    @SuppressWarnings("static-access")
    @Override
    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == btImprimirDocumento) {

            datos.clear(); //Limpiamos el arrayList antes de añadir valores

            // Variables para el panel 2
            String nombre = "", dni = "", telefono = "", apellidos = "", fechaEntrada = "", fechaSalida = "", nDias = "";
            // Variables para el panel 3
            String tipoHabitacion = "", cantidadHabitacion = "", edadKids = "", importeTotal="";

            try {
                // Datos de cliente (Panel 2)
                panel2.setNDias(); //Se calculan los dias de la estancia

                //Obtenemos los datos de Clietnes
                nombre = panel2.getNombre();
                dni = panel2.getDni();
                apellidos = panel2.getApellidos();
                telefono = panel2.getTelefono();
```

```
153                 telefono = panel2.getTelefono();
154                 fechaEntrada = panel2.getFechaEntrada();
155                 fechaSalida = panel2.getFechaSalida();
156                 nDias = panel2.getNDias();
157                 // ----------------------------------------
158                 // Datos de la habitacion (Panel 3)
159                 panel3.calcularImporte();
160 
161                 //Obtenemos los datos de las Habitaciones
162                 tipoHabitacion = panel3.getTipoHabitacion();
163                 cantidadHabitacion = panel3.getNumHabitaciones();
164                 edadKids = panel3.getEdadKids();
165                 importeTotal=panel3.getImporteTotal();
166 
167 
168                 // ----------------------------------------
169 
170                 //Metemos todo en un arrayList para pasarlo al panel Toggle
171                 datos.add(nombre);
172                 datos.add(apellidos);
173                 datos.add(dni);
174                 datos.add(telefono);
175                 datos.add(fechaEntrada);
176                 datos.add(fechaSalida);
177                 datos.add(nDias);
178 
179                 datos.add(tipoHabitacion);
180                 datos.add(cantidadHabitacion);
181                 datos.add(edadKids);
182                 datos.add(importeTotal);
183 
184 
185                 //Llamamos al rellenar clientes del panel 4
186                 Panel_4.rellenarPanelClientes();
187 
188                 //Llamamos al rellenar habitaciones del panel 4
189                 Panel_4.rellenarPanelHabitacion();
190 
191             } catch (ClienteException ex) {
192                 return;
193             } catch (HabitacionException ex) {
194                 return;
195             }
196         }
197 
198         if(e.getSource()==btNuevo) {
199             panel2.resetClientes();
200             panel3.resetHabitacion();
201             panel4.resetVentanas();
202         }
203 
204         if(e.getSource()==btGuardar) {
205             JOptionPane.showMessageDialog(null, "Registro guardado correctamente");
206         }
207     }
208 
209     /**
210      * Gets the datos.
211      * Created on 9 dic. 2020
212      *
213      * @author Carlos
214      * @version 1.0
215      * @return the datos
216      */
217     public static ArrayList<String> getDatos(){
218         return datos;
219     }
220 
221 }
222 
```

Vamos a mostrar las clases de excepción:

ClienteException.java

```java
/*
 * Created on 04-dic-2020
 * @author Carlos
 * @version 1.0
 */
package Exceptions;

// TODO: Auto-generated Javadoc
/**
 * The Class ClienteException
 * Es una clase que hereda de IllegalArgumentException
 * Está destinada a ser lanzada cuando ocurra algun problema con la validacion de los datos de los clientes (panel 2).
 */
public class ClienteException extends IllegalArgumentException {

    /** The Constant serialVersionUID. */
    private static final long serialVersionUID = 1L;

}

```

HabitacionException.java

```java
/*
 * Created on 04-dic-2020
 * @author Carlos
 * @version 1.0
 */
package Exceptions;

// TODO: Auto-generated Javadoc
/**
 * The Class HabitacionException
 * Es una clase que hereda de IllegalArgumentException
 * Está destinada a ser lanzada cuando ocurra algun problema con la validacion de los datos de las habitaciones (panel 3).
 */
public class HabitacionException extends IllegalArgumentException {

    /** The Constant serialVersionUID. */
    private static final long serialVersionUID = 1L;

}

```

Ahora vamos a mostrar el Panel 1, el del titulo:

Panel_1.java

```java
/*
 * Created on 9 dic. 2020
 * @author Carlos
 * @version 1.0
 */
package Paneles;

import java.awt.Color;

// TODO: Auto-generated Javadoc
/**
 * The Class Panel_1.
 */
public class Panel_1 extends JPanel{

    /** The Constant serialVersionUID. */
    private static final long serialVersionUID = 1L;

    /** The titulo. */
    private JLabel titulo;

    /**
     * Instantiates a new panel 1.
     *
     * @author Carlos
     * Created on 9 dic. 2020
     * @version 1.0
     */
    public Panel_1() {
        setBackground(Color.PINK);
        setLayout(new FlowLayout());

        titulo= new JLabel("Hotel Printer-Boi");
        titulo.setFont(new Font("Arial",1,40));
        titulo.setForeground(Color.BLUE);


        add(titulo);

        setVisible(true);
    }

}
```

Ahora el Panel 2, el de los clientes:

Panel_2.java

```java
/*
 * Created on 04-dic-2020
 * @author Carlos
 * @version 1.0
 */
package Paneles;

import java.time.LocalDate;

// TODO: Auto-generated Javadoc
/**
 * The Class Panel_2.
 */
public class Panel_2 extends JPanel {

    /** The Constant serialVersionUID. */
    private static final long serialVersionUID = -8622274147906685389L;

    /** Los JLabel. */
    private JLabel nombre, dni, apellidos, telefono, fechaEntrada, fechaSalida, nDias;

    /** The JTextField. */
    private static JTextField tfNombre, tfDni, tfApellidos, tfTelefono, tfFechaEntrada, tfFechaSalida, tfNDias;

    /**
     * Instantiates el panel 2.
     *
     * @author Carlos Created on 04-dic-2020
     * @version 1.0
     */
    public Panel_2() {
        // ponemos el layout y la resolucion
        setLayout(null);


        // Incializamos los JLabel
        nombre = new JLabel("Nombre:");
        dni = new JLabel("DNI:");
        apellidos = new JLabel("Apellidos:");
        telefono = new JLabel("Telefono:");
        fechaEntrada = new JLabel("Fecha de Entrada:");
        fechaSalida = new JLabel("Fecha de Salida:");
        nDias = new JLabel("Numero de dias:");

        // Inicializamos los JTextField
        tfNombre = new JTextField();
        tfDni = new JTextField();
```

```
        tfNombre = new JTextField();
        tfDni = new JTextField();
        tfApellidos = new JTextField();
        tfTelefono = new JTextField();
        tfFechaEntrada = new JTextField(getFechaActual());
        tfFechaSalida = new JTextField(getFechaPosterior());
        tfNDias = new JTextField();

        tfNDias.setEditable(false);

        // Damos medidas
        nombre.setBounds(50, 275, 50, 50);
        tfNombre.setBounds(160, 290, 100, 25);
        dni.setBounds(50, 350, 50, 25);
        tfDni.setBounds(160, 350, 75, 25);
        apellidos.setBounds(50, 390, 75, 50);
        tfApellidos.setBounds(160, 405, 150, 25);
        telefono.setBounds(50, 455, 75, 25);
        tfTelefono.setBounds(160, 455, 100, 25);
        fechaEntrada.setBounds(50, 500, 110, 25);
        tfFechaEntrada.setBounds(160, 500, 80, 25);
        fechaSalida.setBounds(50, 550, 110, 25);
        tfFechaSalida.setBounds(160, 550, 80, 25);
        nDias.setBounds(50, 600, 100, 25);
        tfNDias.setBounds(160, 600, 50, 25);

        // Añadimos
        add(nombre);
        add(tfNombre);
        add(dni);
        add(tfDni);
        add(apellidos);
        add(tfApellidos);
        add(telefono);
        add(tfTelefono);
        add(fechaEntrada);
        add(tfFechaEntrada);
        add(fechaSalida);
        add(tfFechaSalida);
        add(nDias);
        add(tfNDias);

        // Mostramos el panel
        setVisible(true);
    }

    /**
```

```java
    /**
     * String to local date.
     * Convierte una cadena a LocalDate
     *
     * @param fecha the fecha
     * @return the local date
     */
    private static LocalDate stringToLocalDate(String fecha) {
        DateTimeFormatter formatter = DateTimeFormatter.ofPattern("dd/MM/yyyy");
        LocalDate localDate = LocalDate.parse(fecha, formatter);
        return localDate;
    }

    /**
     * Calcula los dias que hay entre la fecha de salida y la fecha de entrada
     *
     * @return the string
     */
    private static String calcularDias() {
        LocalDate entrada= stringToLocalDate(tfFechaEntrada.getText().trim());
        LocalDate salida= stringToLocalDate(tfFechaSalida.getText().trim());

        Long entradaDias=entrada.atStartOfDay(ZoneOffset.UTC).toInstant().toEpochMilli();
        Long salidaDias=salida.atStartOfDay(ZoneOffset.UTC).toInstant().toEpochMilli();

        if(salidaDias<entradaDias) {
            JOptionPane.showMessageDialog(null, "ERROR: LA FECHA DE SALIDA NO PUEDE SER MENOR A LA DE ENTRADA");
            throw new ClienteException();
        }

        Long diferencia=salidaDias-entradaDias;

        Long millisecondsToDays=TimeUnit.MILLISECONDS.toDays(diferencia);

        String dias=millisecondsToDays.toString();

        return dias;
    }

    /**
     * Gets the fecha actual.
     *
     * @author Carlos
     * @version 1.0
     * @return the fecha actual
     */
```

```java
     */
    public static String getFechaActual() {
        LocalDate ahora = LocalDate.now();
        DateTimeFormatter formateador = DateTimeFormatter.ofPattern("dd/MM/yyyy");
        return ahora.format(formateador);
    }

    /**
     * Gets the fecha posterior.
     *
     * @author Carlos
     * @version 1.0
     * @return the fecha posterior
     */
    public static String getFechaPosterior() {
        LocalDate diaSiguiente = LocalDate.now().plusDays(1);
        DateTimeFormatter formateador = DateTimeFormatter.ofPattern("dd/MM/yyyy");
        return diaSiguiente.format(formateador);
    }

    /**
     * Gets the nombre.
     *
     * @author Carlos
     * @version 1.0
     * @return the nombre
     */
    public String getNombre() {
        return tfNombre.getText().trim();
    }

    /**
     * Gets the dni.
     *
     * @author Carlos
     * @version 1.0
     * @return the dni
     */
    public String getDni() {
        if (tfDni.getText().trim().length() != 9) {
            JOptionPane.showMessageDialog(null, "ERROR: EL DNI DEBE DE TENER 9 ALFANUMERICOS");
            throw new ClienteException();
        }
        return tfDni.getText().trim();
    }

```

```java

    /**
     * Gets the apellidos.
     *
     * @author Carlos
     * @version 1.0
     * @return the apellidos
     */
    public String getApellidos() {
        return tfApellidos.getText().trim();
    }

    /**
     * Gets the telefono.
     *
     * @author Carlos
     * @version 1.0
     * @return the telefono
     */
    public String getTelefono() {
        final String PATRON = "[0-9]+";
        String telefono = tfTelefono.getText().trim();

        if (!telefono.matches(PATRON) || telefono.length() != 9) {
            JOptionPane.showMessageDialog(null, "ERROR: EL TELEFONO DEBE DE TENER 9 DIGITOS");
            throw new ClienteException();
        }

        return tfTelefono.getText().trim();
    }

    /**
     * Gets the fecha entrada.
     *
     * @author Carlos
     * @version 1.0
     * @return the fecha entrada
     */
    public String getFechaEntrada() {
        return tfFechaEntrada.getText().trim();
    }

    /**
     * Gets the fecha salida.
     *
     * @author Carlos
```

```
237         * @author Carlos
238         * @version 1.0
239         * @return the fecha salida
240         */
241        public String getFechaSalida() {
242            return tfFechaSalida.getText().trim();
243        }
244
245        /**
246         * Gets dias de estancia.
247         *
248         * @author Carlos
249         * @version 1.0
250         * @return the n dias
251         */
252        public static String getNDias() {
253            return tfNDias.getText().trim();
254        }
255
256        /**
257         * Metodo que modifica el campo de los dias de estancia
258         */
259        public static void setNDias() {
260            tfNDias.setText(calcularDias());
261        }
262
263        /**
264         * Metodo que realiza un reset de los campos de clientes.
265         */
266        public void resetClientes() {
267            tfNombre.setText("");
268            tfDni.setText("");
269            tfApellidos.setText("");
270            tfTelefono.setText("");
271            tfFechaEntrada.setText(getFechaActual());
272            tfFechaSalida.setText(getFechaPosterior());
273            tfNDias.setText("");
274
275            tfNombre.requestFocus();
276        }
277
```

Proseguimos con el Panel 3, el de las habitaciones:

Panel_3.java

```java
/*
 * Created on 9 dic. 2020
 * @author Carlos
 * @version 1.0
 */
package Paneles;

import java.awt.BorderLayout;

// TODO: Auto-generated Javadoc
/**
 * The Class Panel_3.
 */
public class Panel_3 extends JPanel implements ActionListener {

    /** The Constant serialVersionUID. */
    private static final long serialVersionUID = 1519237183299406495L;

    /** Los JLabel. */
    private JLabel tfTipoHabitacion, numHabitaciones, importeHabitacion;

    /** The lista. */
    private static JComboBox<String> lista;

    /** The sp habitaciones. */
    private JSpinner spHabitaciones;

    /** The panel. */
    private JPanel panel;

    /** The ninios. */
    private static JCheckBox ninios;

    /** The panel kids. */
    private Panel_ExtraKids panelKids;

    /** The panel imagenes. */
    @SuppressWarnings("unused")
    private Panel_Imagenes panelImagenes;

    /** The tf importe habitacion. */
    private static JTextField tfImporteHabitacion;

    private JButton botonImg;

```

```java
/**
 * Instantiates a new panel 3.
 *
 * @author Carlos
 * Created on 9 dic. 2020
 * @version 1.0
 */
// ponemos el layout y la resolucion
public Panel_3() {
    setLayout(null);
    // setSize(640, 480);

    // Array que contiene las distintas opciones del JComboBox
    String[] tipoHabitacion = { "Simple", "Doble", "Suite" };
    // ------------------------------------------------------------------------

    // Inicializamos el JComboBox
    lista = new JComboBox<String>(tipoHabitacion);

    // Incializamos los JLabel
    tfTipoHabitacion = new JLabel("Tipo de habitacion:");
    numHabitaciones = new JLabel("Numero de habitaciones:");
    importeHabitacion = new JLabel("Importe de la habitacion:");
    importeHabitacion.setForeground(Color.RED);

    // Inicializamos el JTextField
    tfImporteHabitacion = new JTextField();
    tfImporteHabitacion.setEditable(false);

    // Incializamos el JSlider
    spHabitaciones = new JSpinner(new SpinnerNumberModel(1, 1, 50, 1));

    // Lo añadimos a un panel para evitar que ocupe mucho espacio
    panel = new JPanel();
    panel.add(spHabitaciones);

    // Inicializamos el JCheckBox
    ninios = new JCheckBox("¿Niños?");

    // Inicializamos el panel que se abrirá en caso de pulsar el JCheckBox
    panelKids = new Panel_ExtraKids();

    botonImg=new JButton("Mostrar imagenes de las habitaciones");
    // ------------------------------------------------------------------------
```

```
101         // ------------------------------------------------------------------------------
102
103         // Añadimos los listeners
104         lista.addActionListener(this);
105         ninios.addActionListener(this);
106         botonImg.addActionListener(this);
107         // ------------------------------------------------------------------------------
108
109         // Damos posiciones Falta calibrar la posicion de los paneles de kids e imagenes
110         tfTipoHabitacion.setBounds(50, 350, 125, 50);
111         lista.setBounds(200, 362, 100, 25);
112
113         numHabitaciones.setBounds(50, 450, 150, 25);
114         spHabitaciones.setBounds(200, 450, 75, 25);
115
116         ninios.setBounds(50, 550, 75, 50);
117
118         panelKids.setBounds(50, 600, 250, 150);
119
120         importeHabitacion.setBounds(50, 800, 150, 25);
121         tfImporteHabitacion.setBounds(200, 800, 100, 25);
122
123         botonImg.setBounds(150, 100, 375, 50);
124         // ------------------------------------------------------------------------------
125
126         // Añadimos al panel
127         add(tfTipoHabitacion);
128         add(numHabitaciones);
129         add(lista);
130         add(spHabitaciones);
131         add(ninios);
132         add(panelKids);
133         add(importeHabitacion);
134         add(tfImporteHabitacion);
135         add(botonImg, BorderLayout.NORTH);
136         // ------------------------------------------------------------------------------
137
138         // Mostramos el panel
139         setVisible(true);
140     }
141
142     /**
143      * Metodo que calcula el importe de la habitacion.
144      * Realizando la llamada al setImporteTotal para mostrarlo en el JTextField
145      */
146     public static void calcularImporte() {
```

```java
     */
    public static String getFechaActual() {
        LocalDate ahora = LocalDate.now();
        DateTimeFormatter formateador = DateTimeFormatter.ofPattern("dd/MM/yyyy");
        return ahora.format(formateador);
    }

    /**
     * Gets the fecha posterior.
     *
     * @author Carlos
     * @version 1.0
     * @return the fecha posterior
     */
    public static String getFechaPosterior() {
        LocalDate diaSiguiente = LocalDate.now().plusDays(1);
        DateTimeFormatter formateador = DateTimeFormatter.ofPattern("dd/MM/yyyy");
        return diaSiguiente.format(formateador);
    }

    /**
     * Gets the nombre.
     *
     * @author Carlos
     * @version 1.0
     * @return the nombre
     */
    public String getNombre() {
        return tfNombre.getText().trim();
    }

    /**
     * Gets the dni.
     *
     * @author Carlos
     * @version 1.0
     * @return the dni
     */
    public String getDni() {
        if (tfDni.getText().trim().length() != 9) {
            JOptionPane.showMessageDialog(null, "ERROR: EL DNI DEBE DE TENER 9 ALFANUMERICOS");
            throw new ClienteException();
        }
        return tfDni.getText().trim();
    }

```

```java
     * Metodo que devuele el tipo de la habitacion.
     *
     * @author Carlos
     * @version 1.0
     * @return el tipo de la habitacion
     */
    public String getTipoHabitacion() {
        return lista.getSelectedItem().toString();
    }

    /**
     * Metodo que devuelve el numero de habitaciones.
     *
     * @author Carlos
     * @version 1.0
     * @return el numero de habitaciones
     */
    public String getNumHabitaciones() {
        return spHabitaciones.getValue().toString();
    }

    /**
     * Metodo que devuelve la edad en caso de que la casilla de niños esté activada
     * En caso contrario devuelve N, indicando que no está pulsada.
     *
     * @author Carlos
     * @version 1.0
     * @return la edad o N
     */
    public String getEdadKids() {
        if (ninios.isSelected() == true) {
            return panelKids.getEdad();
        }
        return "N";
    }

    /**
     * Sets the importe total.
     * Created on 9 dic. 2020
     *
     * @author Carlos
     * @version 1.0
     * @param importe the new importe total
     */
    // Metodo set para el JTextField del importe de la habitacion
    public static void setImporteTotal(int importe) {
        tfImporteHabitacion.setText(Integer.toString(importe) + " €");
    }

    /**
     * Gets the importe total.
     * Created on 9 dic. 2020
     *
     * @author Carlos
     * @version 1.0
     * @return the importe total
     */
    public String getImporteTotal() {
        return tfImporteHabitacion.getText();
    }

    /**
     * Metodo que resetea todo el panel.
     */
    public void resetHabitacion() {
        lista.setSelectedIndex(0);
        spHabitaciones.setValue(((SpinnerNumberModel) spHabitaciones.getModel()).getMinimum());
        panelKids.resetKids();
        importeHabitacion.setText("");
    }

}
```

Seguimos con el Panel 4, el que mostrará los datos:

Panel_4.java

```java
/*
 * Created on 9 dic. 2020
 * @author Carlos
 * @version 1.0
 */
package Paneles;

import java.awt.BorderLayout;

// TODO: Auto-generated Javadoc
/**
 * The Class Panel_4.
 */
public class Panel_4 extends JPanel {

    /** The Constant serialVersionUID. */
    private static final long serialVersionUID = -1740828945108575741L;

    /** The pestañas. */
    private JTabbedPane pestañas;

    /** The datos. */
    private static ArrayList<String> datos = Dialogo.getDatos();

    /** The area clientes. */
    private static JTextArea areaClientes;

    /** The area habitacion. */
    private static JTextArea areaHabitacion;


    /**
     * Instantiates a new tabbed panel.
     *
     * @author Carlos Created on 23-nov-2020
     * @version 1.0
     */
    public Panel_4() {
        setLayout(null);

        pestañas = new JTabbedPane();

        // Inicializamos el panel
        JPanel panel1 = new JPanel();
        JPanel panel2 = new JPanel();

        // Componentes del panel1------------------------
```

```
53
54          // Componentes del panel1------------------------
55          panel1.setLayout(new BorderLayout());
56
57          areaClientes=new JTextArea("");
58          areaClientes.setEditable(false);
59
60          panel1.add(areaClientes);
61
62          // Componentes del panel2------------------------
63
64          panel2.setLayout(new BorderLayout());
65
66          areaHabitacion=new JTextArea("");
67          areaHabitacion.setEditable(false);
68
69          panel2.add(areaHabitacion);
70
71          // Añadimos los paneles al JTabbedPane
72          pestañas.addTab("Datos Cliente", panel1);
73          pestañas.addTab("Datos Habitacion", panel2);
74
75          pestañas.setBounds(75, 100, 450, 500);
76
77          // Añadimos el JTabbedPane
78          add(pestañas);
79
80          setVisible(true);
81      }
82
83      /**
84       * Metodo que rellena el textArea con los datos indicados de Clientes.
85       */
86      public static void rellenarPanelClientes() {
87
88          areaClientes.setText("Nombre: "+datos.get(0)+"\n"
89                  + "Apellidos: "+datos.get(1)+"\n"
90                  + "DNI: "+datos.get(2)+"\n"
91                  + "Telefono: "+datos.get(3)+"\n"
92                  + "Fecha de entrada: "+datos.get(4)+"\n"
93                  + "Fecha de salida: "+datos.get(5)+"\n"
94                  + "Numero de dias: "+datos.get(6));
95      }
96
97      /**
98       * Metodo que rellena el textArea con los datos indicados de Habitaciones.
99       * Si la cadena de los niños es igual a N quiere decir que no hay y por eso se muestra el mensaje indicado
100      */
101     public static void rellenarPanelHabitacion() {
102         String cadenaKids = "No hay";
103
104         if (datos.get(9).trim() != "N") {
105             cadenaKids = datos.get(9);
106         }
107
108         areaHabitacion.setText("Tipo de habitacion: "+datos.get(7)+"\n"
109                 + "Cantidad de habitaciones: "+datos.get(8)+"\n"
110                 + "Edad de los niños: "+cadenaKids+"\n"
111                 + "Importe total: "+datos.get(10));
112
113     }
114
115     /**
116      * Metodo que vacía los dos textArea.
117      */
118     public void resetVentanas() {
119         areaClientes.setText("");
120         areaHabitacion.setText("");
121     }
122
123 }
124
```

Mostramos el panel de las Imagenes del panel 3:

Panel_Imagenes.java

```java
/*
 * Created on 9 dic. 2020
 * @author Carlos
 * @version 1.0
 */
package Paneles;

import java.awt.GridLayout;

// TODO: Auto-generated Javadoc
/**
 * The Class Panel_Imagenes.
 */
public class Panel_Imagenes extends JDialog {

    /** The Constant serialVersionUID. */
    private static final long serialVersionUID = 1L;

    /** The l img suite. */
    private JLabel lImgSimple, lImgDoble, lImgSuite;

    /** The suite. */
    private JLabel simple, doble, suite;

    /**
     * Instantiates a new panel imagenes.
     *
     * @author Carlos
     * Created on 9 dic. 2020
     * @version 1.0
     */
    public Panel_Imagenes() {
        setSize(640, 480);
        setLocationRelativeTo(null);
        setResizable(false);
        setLayout(new GridLayout(3, 2));

        //Imagenes
        Image imgSimple = new ImageIcon(getClass().getResource("resources/hotel1.jpg")).getImage();
        ImageIcon imgSimpleUps = new ImageIcon(imgSimple.getScaledInstance(200, 100, Image.SCALE_SMOOTH));

        Image imgDoble = new ImageIcon(getClass().getResource("resources/hotel2.jpg")).getImage();
        ImageIcon imgDobleUps = new ImageIcon(imgDoble.getScaledInstance(200, 100, Image.SCALE_SMOOTH));

        Image imgSuite = new ImageIcon(getClass().getResource("resources/hotel3.jpg")).getImage();
        ImageIcon imgSuiteUps = new ImageIcon(imgSuite.getScaledInstance(200, 100, Image.SCALE_SMOOTH));

        //-------------------------------------------------------------------

        //Incializamos los Jlabel con sus imagenes
        lImgSimple = new JLabel(imgSimpleUps);
        lImgDoble = new JLabel(imgDobleUps);
        lImgSuite = new JLabel(imgSuiteUps);

        //Incializamos los Jlabel con su texto
        simple = new JLabel("Habitacion Simple");
        doble = new JLabel("Habitacion Doble");
        suite = new JLabel("Habitacion Suite");

        //Añadimos todo al panel
        add(lImgSimple);
        add(simple);
        add(lImgDoble);
        add(doble);
        add(lImgSuite);
        add(suite);

        //Lo mostramos
        setVisible(true);
    }
}
```

Y terminamos el panel de los niños:

Panel_ExtraKids.java

```java
/*
 * Created on 08-dic-2020
 * @author Carlos
 * @version 1.0
 */
package Paneles;

import java.awt.Color;

// TODO: Auto-generated Javadoc
/**
 * The Class Panel_ExtraKids.
 */
public class Panel_ExtraKids extends JPanel implements FocusListener {

    /** The Constant serialVersionUID. */
    private static final long serialVersionUID = 1L;

    /** The l extras. */
    private JLabel lEdad, lExtras;

    /** The edad. */
    private JSpinner edad;

    /** The extras. */
    private JTextField extras;

    /**
     * Instantiates a new panel extra kids.
     *
     * @author Carlos
     * Created on 08-dic-2020
     * @version 1.0
     */
    public Panel_ExtraKids() {
        setLayout(null);
        setBackground(Color.GRAY);

        //Inicializamos los JLabel
        lEdad = new JLabel("Edad:");
        lExtras = new JLabel("Extras:");

        // Incializamos el JSlider
        edad = new JSpinner(new SpinnerNumberModel(0, 0, 14, 1));

        //Inicializamos el JTextField
        extras = new JTextField();
```

```
55          extras = new JTextField();
56          extras.setEditable(false);
57
58          //Añadimos el foco
59          extras.addFocusListener(this);
60
61
62          lEdad.setBounds(25, 20, 50, 25);
63          edad.setBounds(70, 20, 50, 25);
64          lExtras.setBounds(25, 75, 50, 25);
65          extras.setBounds(70, 75, 150, 25);
66
67          //Añadimos al panel
68          add(lEdad);
69          add(edad);
70          add(lExtras);
71          add(extras);
72          setVisible(false);
73      }
74
75
76      public String getEdad() {
77          String laEdad= edad.getValue().toString();
78          return laEdad;
79      }
80
81      /**
82       * Metodo que devuelve Focus gained.
83       *
84       * @param arg0 the arg 0
85       */
86      @Override
87      public void focusGained(FocusEvent arg0) {
88          int edadValor=(Integer) edad.getValue();
89
90          if(edadValor<=3)
91              extras.setText("Cuna");
92
93          if(edadValor>3 && edadValor<=10)
94              extras.setText("Cama supletoria pequeña");
95
96          if(edadValor>10 && edadValor<=14)
97              extras.setText("Cama supletoria normal");
98
99      }
```

```java

    /**
     * Metodo que devuelve Focus lost.
     *
     * @param arg0 the arg 0
     */
    @Override
    public void focusLost(FocusEvent arg0) {
    }


    public void resetKids() {
        edad.setValue(((SpinnerNumberModel) edad.getModel()).getMinimum());
        lExtras.setText("");
    }

}

```

Por ultimo muestro la distribución de paquetes: