

Usabilidad y accesibilidad

1. Visibilidad del estado del sistema.....	3
2. Relación entre el sistema y el mundo real.....	4
3. Control y libertad del usuario.....	4
4. Consistencia y estándares.....	5
5. Prevención de errores.....	5
6. Reconocimiento antes que recuerdo.....	6
7. Flexibilidad y eficiencia de uso.....	6
8. Estética y diseño minimalista.....	7
9. Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de errores.....	8
10. Ayuda y documentación.....	9

1. Visibilidad del estado del sistema

Principio: Visibilidad: Explica al usuario cuál es el estado del sistema en cada momento, y manténlo informado de lo que está pasando.

- Justificación:

- Se cumple mediante la actualización en tiempo real de la etiqueta `labelPrecioMinimo`. Gracias a los `ChangeListener` y `ItemListener` añadidos a los `Spinners` y `ComboBoxes`, el usuario ve cómo cambia el precio final inmediatamente después de modificar el número de habitaciones, baños o tipo de cama, sin tener que esperar a pulsar un botón de "Calcular".
- El uso de `JTabbedPane` permite al usuario saber en qué sección del resumen se encuentra (Arrendador o Inmueble) mediante la pestaña activa.

Apartamentos Andarax

Datos del Arrendador

Datos del Inmueble

Estado de conservación:

Precio Mínimo Calculado: 0,00 €

Apartamentos Andarax

Datos del Arrendador

Datos del Inmueble

Estado de conservación:

Precio Mínimo Calculado: 72,00 €

Resumen de Datos

Arrendador Inmueble

2. Relación entre el sistema y el mundo real

Principio: Relación con la realidad: Utiliza un lenguaje familiar y apropiado para los usuarios a los que te diriges, y organiza la información con un orden natural y lógico.

- Justificación:
 - Uso de terminología inmobiliaria familiar: "Arrendador", "Fianza", "Alta Pisos", "Cuna", "Cama Supletoria".
 - Uso de metáforas visuales estándar en los botones: Icono de disquete para "Guardar", hoja en blanco para "Nuevo" e impresora para "Imprimir".
 - Lógica de fechas naturales: El código Calendar.add(Calendar.YEAR, 1) asegura que la fecha de fin sea coherente con un contrato anual respecto a la fecha de inicio.



3. Control y libertad del usuario

Principio: Control y libertad: Ofrece funciones de rehacer y deshacer que permitan al usuario tener el control de sus interacciones con libertad.

- Justificación:
 - Implementación del botón "Nuevo" que invoca al método limpiarFormulario(). Esto permite al usuario "deshacer" todo el llenado del formulario si se ha equivocado o quiere empezar de cero, reseteando campos, bordes rojos y variables internas.
 - Uso de setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE) en la ventana secundaria, permitiendo al usuario cerrar la ventana de alta sin cerrar la aplicación principal.



4. Consistencia y estándares

Principio: Consistencia y estándares: Establecer una convenciones lógicas y mantenerlas siempre (mismo lenguaje, mismo flujo de navegación)

- Justificación:
 - Uso de componentes estándar de Java Swing (JTextField, JComboBox, JSpinner) que se comportan como el usuario espera en cualquier aplicación de escritorio.
 - Uso de atajos de teclado estándar (Accelerators) definidos en la clase Menu: Ctrl+S para Guardar, Ctrl+N para Nuevo, Ctrl+Q para Salir. Esto respeta los estándares de la industria del software.
 - Diseño visual consistente mediante GridBagLayout y BorderFactory en todos los paneles.



5. Prevención de errores

Principio: Prevención de errores: Ayuda a los usuarios a evitar equivocarse antes de que cometan el error.

- Justificación:
 - Uso de JSpinner con SpinnerNumberModel y SpinnerDateModel: Evita físicamente que el usuario introduzca letras en campos numéricos o seleccione fechas imposibles (como el 32 de enero).
 - Uso de JComboBox para provincias y tipos de cama: Evita errores ortográficos al obligar a elegir de una lista predefinida.
 - Lógica de fechas restringida: El código ajusta automáticamente el mínimo de la fechaFin basándose en la fechaAlta, impidiendo seleccionar una fecha de fin anterior a la de inicio.
 - Ocultación de opciones irrelevantes: El panel de tipo de cama y detalles de niños permanece oculto (setVisible(false)) hasta que es necesario, evitando que el usuario rellene datos que no aplican.

6. Reconocimiento antes que recuerdo

Principio: Reconocimiento: Haz visible todo lo que sea posible, no esperes que los usuarios recuerden o memoricen información, muéstralas si es necesaria en el proceso, las instrucciones deben estar a la vista cuando sea necesario.

- Justificación:
 - Implementación extensiva de setToolTipText en el método configurarAccesibilidad. Al pasar el ratón por encima de "DNI" o "Teléfono", el sistema muestra el formato esperado ("8 números y una letra"), eliminando la carga de memoria del usuario.
 - Uso de Mnemonics (Alt + Letra) visibles en los menús y etiquetas (subrayado de la letra), indicando visualmente cómo interactuar con el teclado.
 - El panel de "Resumen" permite al usuario revisar los datos introducidos antes de confirmar, sin tener que recordar qué escribió en la pestaña anterior.

Datos del Arrendador

Nombre: _____

Apellidos: Introduce el nombre del arrendador

7. Flexibilidad y eficiencia de uso

Principio: Flexibilidad: Permite que el sistema pueda adaptarse a los usuarios frecuentes, diseña la realización de tareas avanzadas de manera fluida y eficiente.

- Justificación:
 - El sistema permite dos modos de operación:
 1. Novatos: Uso del ratón navegando por los menús y haciendo clic en los botones.
 2. Expertos: Uso de Accelerators (Ctrl+N para abrir alta) y Mnemonics (Alt+N para ir al nombre, Alt+G para guardar) definidos en el método configurarAccesibilidad y la clase Menu.
 - El botón "Guardar" está configurado como defaultButton (getRootPane().setDefaultButton(btnGuardar)), permitiendo enviar el formulario pulsando "Enter", agilizando el flujo.

```

JMenu archivo = new JMenu("Archivo");
archivo.setMnemonic(KeyEvent.VK_A);
JMenuItem salir = new JMenuItem("Salir");
salir.setMnemonic(KeyEvent.VK_S);
salir.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_Q, InputEvent.CTRL_DOWN_MASK));
salir.addActionListener(e -> System.exit(0));
archivo.add(salir);

JMenu registro = new JMenu("Registro");
registro.setMnemonic(KeyEvent.VK_R);
JMenuItem alta = new JMenuItem("Alta Pisos");
alta.setMnemonic(KeyEvent.VK_A);
alta.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_N, InputEvent.CTRL_DOWN_MASK));

JMenuItem baja = new JMenuItem("Baja Pisos");
baja.setMnemonic(KeyEvent.VK_B);
baja.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_B, InputEvent.CTRL_DOWN_MASK));

registro.add(alta);
registro.add(baja);

JMenu ayuda = new JMenu("Ayuda");
ayuda.setMnemonic(KeyEvent.VK_Y);

JMenuItem acercaDe = new JMenuItem("Acerca de...");
acercaDe.setMnemonic(KeyEvent.VK_C);
acercaDe.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_F1, 0));
acercaDe.addActionListener(e -> JOptionPane.showMessageDialog(ventana, mensaje));

```

8. Estética y diseño minimalista

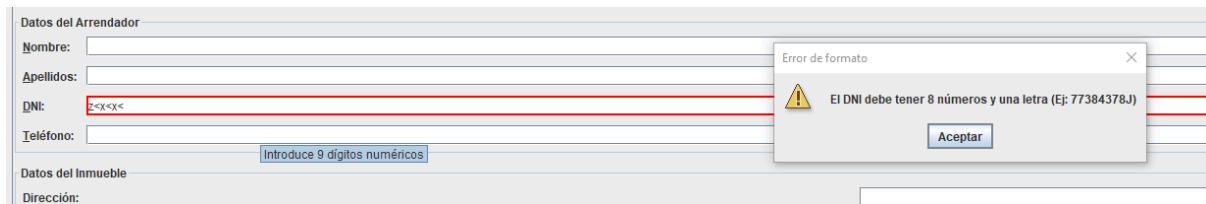
Principio: Estética y minimalismo: Muestra sólo lo necesario y relevante en cada situación, no distraigas al usuario con información extra poco relevante.

- Justificación:
 - Organización limpia mediante JTabbedPane para separar la entrada de datos del resumen, evitando saturar una sola pantalla.
 - Uso de TitledBorder ("Datos del Arrendador", "Datos del Inmueble") para agrupar visualmente la información relacionada, reduciendo la carga cognitiva.
 - La cabecera utiliza un diseño diferenciado (fondo azul claro) pero simple, destacando la identidad corporativa sin distraer del formulario.

9. Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de errores

Principio: Recuperarse de los errores: Ayuda a los usuarios a reconocer y corregir sus errores, indica siempre el problema concreto que está ocurriendo y sugiere soluciones constructivas.

- Justificación:
 - Implementación de la clase ExpresionRegularInputVerifier. Cuando el usuario introduce un DNI o teléfono con formato incorrecto, el sistema:
 1. Marca el borde del campo en rojo (LineBorder Color.RED) para localización rápida.
 2. Muestra un JOptionPane explicativo (ej: "El teléfono debe tener 9 dígitos numéricos").
 - El método validarCamposObligatorios genera un resumen de todos los errores encontrados antes de permitir guardar, listándolos claramente.



10. Ayuda y documentación

Principio: Ayuda y documentación: La información de ayuda debe ser breve, concisa, fácil de buscar y enfocada a las tareas del usuario.

- Justificació:
 - Existencia de un menú específico "Ayuda" en la barra de menú principal.
 - La opción "Acerca de..." proporciona información esencial sobre la versión y autoría.
 - Los Tooltips actúan como micro-ayuda contextual integrada en cada campo del formulario, guiando al usuario paso a paso sin necesidad de consultar un manual externo.

