

# Documento sobre Usabilidad

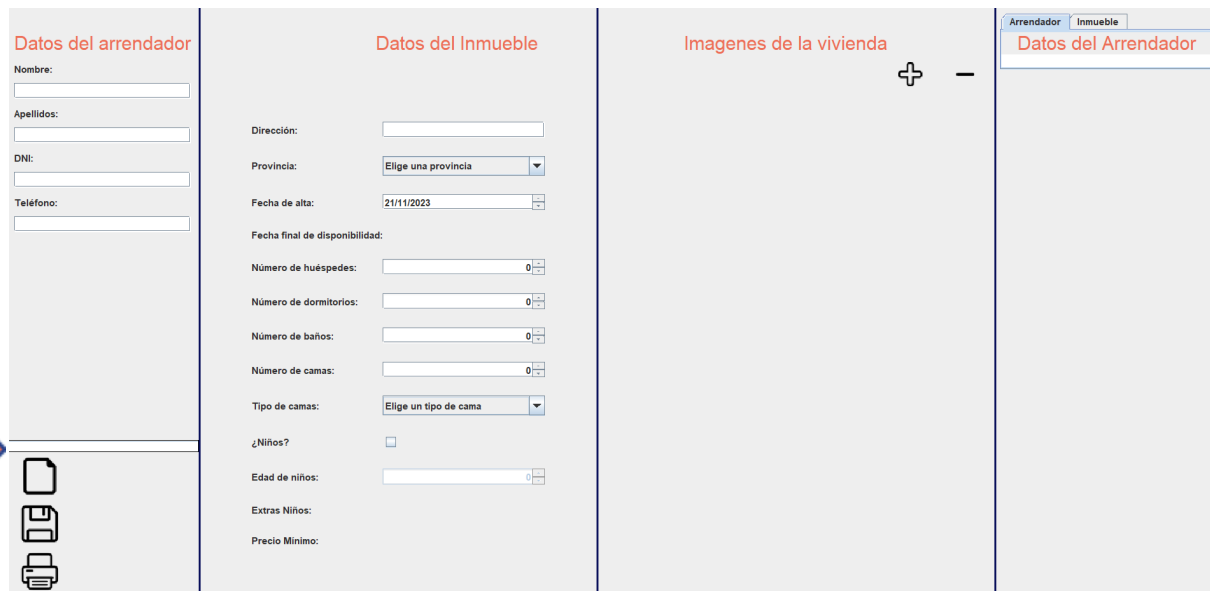
# Índice

<b>Sobre las 10 reglas heurísticas de usabilidad de Jacob Nielsen</b>	<b>2</b>
1- Visibilidad del estado del sistema	2
2- Coincidencia entre el sistema y el mundo real	3
3- Control y libertad del usuario	4
4- Consistencia y estándares	4
5- Prevención de errores	5
6- Reconocimiento en lugar de recuerdo	6
7- Flexibilidad y eficiencia de uso	6
8- Diseño estético y minimalista	6
9- Ayuda a los usuarios a reconocer, diagnosticar y recuperarse de los errores	7
10- Ayuda y documentación	8
<b>Sobre la elección de los colores</b>	<b>9</b>

# Sobre las 10 reglas heurísticas de usabilidad de Jacob Nielsen

## 1- Visibilidad del estado del sistema

En este caso, he creado un progressbar que irá aumentando en funcion de que se vayan rellenando los datos:

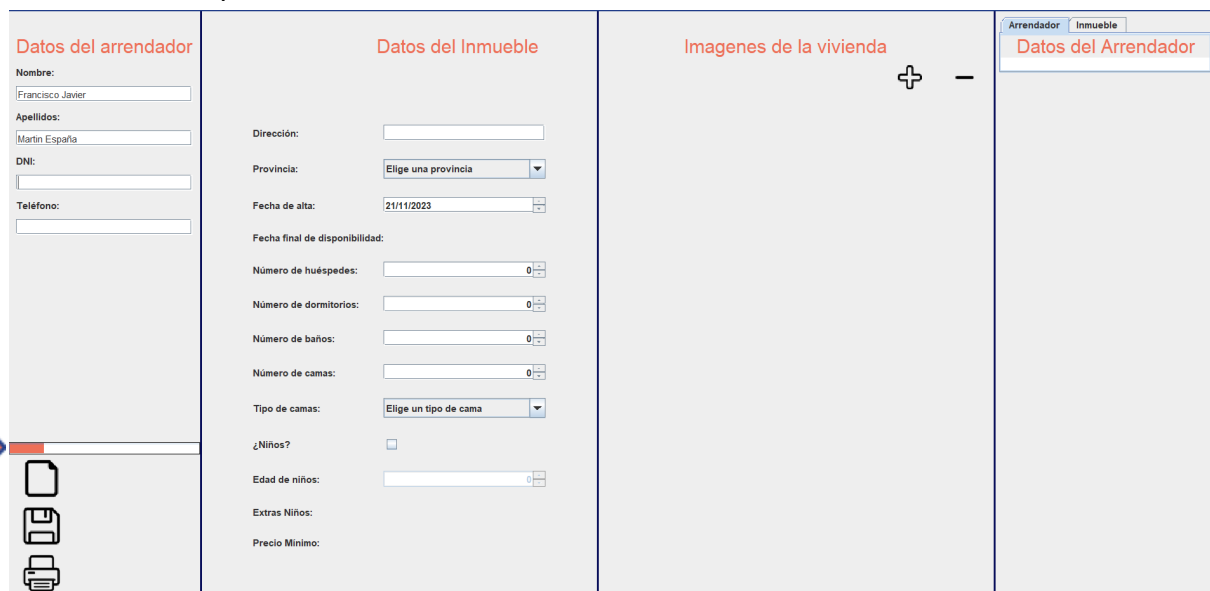


The form is divided into four main sections:

- Datos del arrendador:** Includes fields for Nombre, Apellidos, DNI, and Teléfono.
- Datos del Inmueble:** Includes fields for Dirección, Provincia (dropdown), Fecha de alta, Fecha final de disponibilidad, Número de huéspedes, Número de dormitorios, Número de baños, Número de camas, Tipo de camas (dropdown), ¿Niños?, Edad de niños, Extras Niños, and Precio Mínimo.
- Imagenes de la vivienda:** A large area for uploading images, indicated by a plus sign and minus sign.
- Datos del Arrendador:** A tabbed section with a sub-tab for 'Arrendador' and 'Inmueble'.

A red arrow points to the progress bar at the bottom left, which is currently empty.

Si escribimos campos, se ira rellenando.



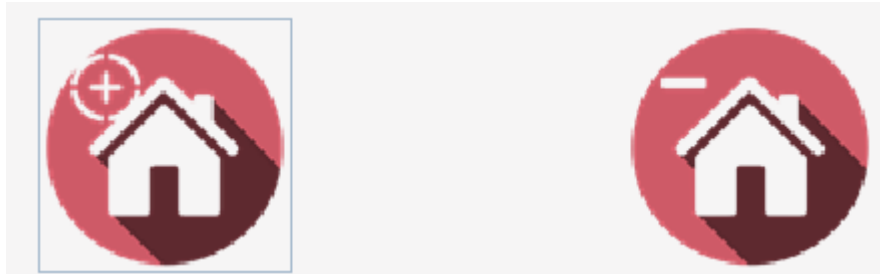
The form is identical to the previous one, but with some input fields filled:

- Datos del arrendador:** Nombre: Francisco Javier, Apellidos: Martin España.
- Datos del Inmueble:** Dirección: (empty), Provincia: (dropdown), Fecha de alta: 21/11/2023, Fecha final de disponibilidad: (empty), Número de huéspedes: 0, Número de dormitorios: 0, Número de baños: 0, Número de camas: 0, Tipo de camas: (dropdown), ¿Niños?: (checkbox), Edad de niños: (empty), Extras Niños: (empty), Precio Mínimo: (empty).

A red arrow points to the progress bar at the bottom left, which is now partially filled with red.

## 2- Coincidencia entre el sistema y el mundo real

En este caso he usado iconos bastantes reconocibles(creo).



Icono alta pisos (Uso una vivienda con un simbolo "+", creo que se entiende).

Icono bajapisos (Lo mismo pero al contrario, uso el "-").

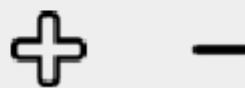


→Icono nuevo (uso una papelera para que se vea que es para "limpiar" el contenido).

→Icono guardar(uso un disquete, el icono bastante normalizado de guardar).

→Icono imprimir(uso el icono de la impresora).

### Imágenes de la vivienda



Los iconos + y - quizás no dan mucha informacion, pero se entiende que si se encuentra en el panel "Imágenes de la vivienda", serán para añadir o quitar imagenes.

### 3- Control y libertad del usuario

En mi programa, todo se puede deshacer, desde borrar el texto de un campo, hasta cambiar elementos de spinner, marcar o desmarcar los checkbox, incluso añadir o quitar imágenes.

### 4- Consistencia y estándares

Creo que todo el proyecto es bastante consistente, porque no hay ambigüedades, los campos están bien definidos y no debería haber lugar para ambigüedades.

## 5- Prevención de errores

Aparte de implementar unos maskformatters en los campos telefono y DNI, les añado unas expresiones regulares, para prevenir por ejemplo que, el numero de telefono empiece por "1". También preveo el error por ejemplo si el usuario quiere subir una imagen y no es una imagen.

→Usos de maskformatters

```
try {
    filtroTelefono = new MaskFormatter("#####");
    txtTelefono = new JFormattedTextField(filtroTelefono);
    txtTelefono.setFocusLostBehavior(JFormattedTextField.COMMIT);
    filtroDNI = new MaskFormatter("#####A");
    txtDNI = new JFormattedTextField(filtroDNI);
    txtDNI.setFocusLostBehavior(JFormattedTextField.COMMIT);
} catch (ParseException e) {

    System.out.println(e.getMessage());
}
```

→Usos de expresiones regulares para prevenir que se introduzcan valores que no corresponden (por ejemplo un nombre o apellido que empiece por numero).

```
String patronDNI = "\\d{8}[A-HJ-NP-TV-Za-hj-np-tv-z]";
String patronTelefono = "[6-9][0-9]{8}";
String patronNombreApellido = "^[a-zA-ZáéíóúÁÉÍÓÚñÑ\\s]*$";
```

→Usos de try catch que prevee el elemento que sea o no una imagen

```
if (fileChooser.showOpenDialog(null) == JFileChooser.APPROVE_OPTION) {
    try {
        // Leemos el archivo como imagen
        Image imagen = ImageIO.read(fileChooser.getSelectedFile());

        // Si la imagen no es nula (es una imagen)
        if (imagen != null) {
            // Creamos un icono con la imagen y le damos un tamaño
            ImageIcon icono = new ImageIcon(
                imagen.getScaledInstance(280, 210, java.awt.Image.SCALE_SMOOTH));

            // Creamos una etiqueta con el icono y la añadimos al panel y a la listaImágenes
            JLabel etiquetaImagen = new JLabel(icono);
            this.add(etiquetaImagen);
            listaImágenes.add(etiquetaImagen);

            // Actualizamos el panel
            this.revalidate();
            this.repaint();
        } else {
            // Si la imagen es nula, (no es una imagen) mostramos un mensaje de error
            JOptionPane.showMessageDialog(this, "El archivo seleccionado no es una imagen válida.", "Error",
                JOptionPane.ERROR_MESSAGE);
        }
    } catch (IOException ex) {
    }
}
```

→Todos estos elementos lanzan un JOptionPane con los correspondientes mensajes de error.

## 6- Reconocimiento en lugar de recuerdo

Todo está absolutamente a la vista, no hay nada escondido.

## 7- Flexibilidad y eficiencia de uso

En mi caso, por ejemplo solo tengo ciertos atajos, medianamente “visibles”, pero no he implementado mas controles para la eficiencia de uso, sin embargo, no he visto necesario de poner mas allá de los aceleradores.

## 8- Diseño estético y minimalista

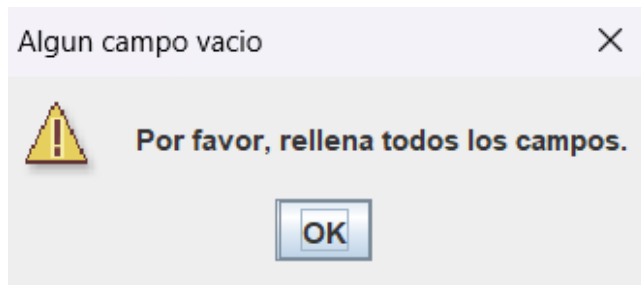
Los campos contienen la información minima que necesitamos, y bastante concisa, lo minimo para entenderlo vamos, y los botones, son imágenes.

Dirección	Nombre
Provincia	<input type="text"/>
Fecha de alta	Apellidos
Fecha final de disponibilidad	<input type="text"/>
Número de huéspedes	DNI
Número de dormitorios	<input type="text"/>
Número de baños	Teléfono
Número de camas	<input type="text"/>
Tipo de camas	
¿Niños?	
Edad de niños	
Extras Niños	
Precio Mínimo	

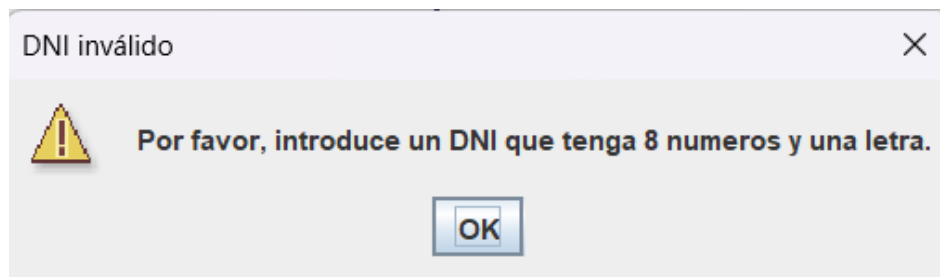
## 9- Ayuda a los usuarios a reconocer, diagnosticar y recuperarse de los errores

Los mensajes de error que posee mi proyecto, explica el error en cuestión por ejemplo:

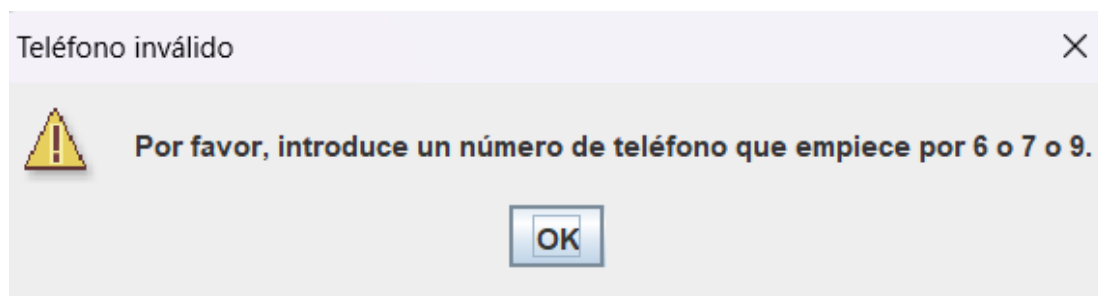
→Falta uno o mas campos



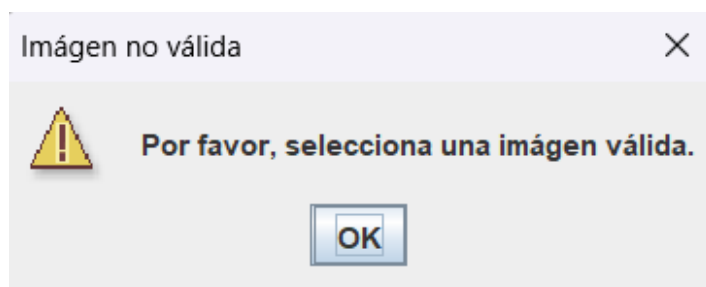
→El DNI no está correctamente escrito.



→EL número de teledono no esta correctamente escrito.



→No se ha seleccionado una imagen válida





## 10- Ayuda y documentación

Tengo un “acerca de”, el cual en teoría enlazaría con una web, o abriría un documento donde tendría toda esa documentación.



## Sobre la eleccion de los colores

He decidido usar una paleta con colores fríos, que creo que va bien para el diseño, me pareció suficiente para el diseño de esta aplicación. Tiene demasiados elementos, y casi todo texto, he querido dejar el color de la ventana tal cual y los colores azules como separadores casi siempre, el anaranjado este, para el texto mayormente.

