

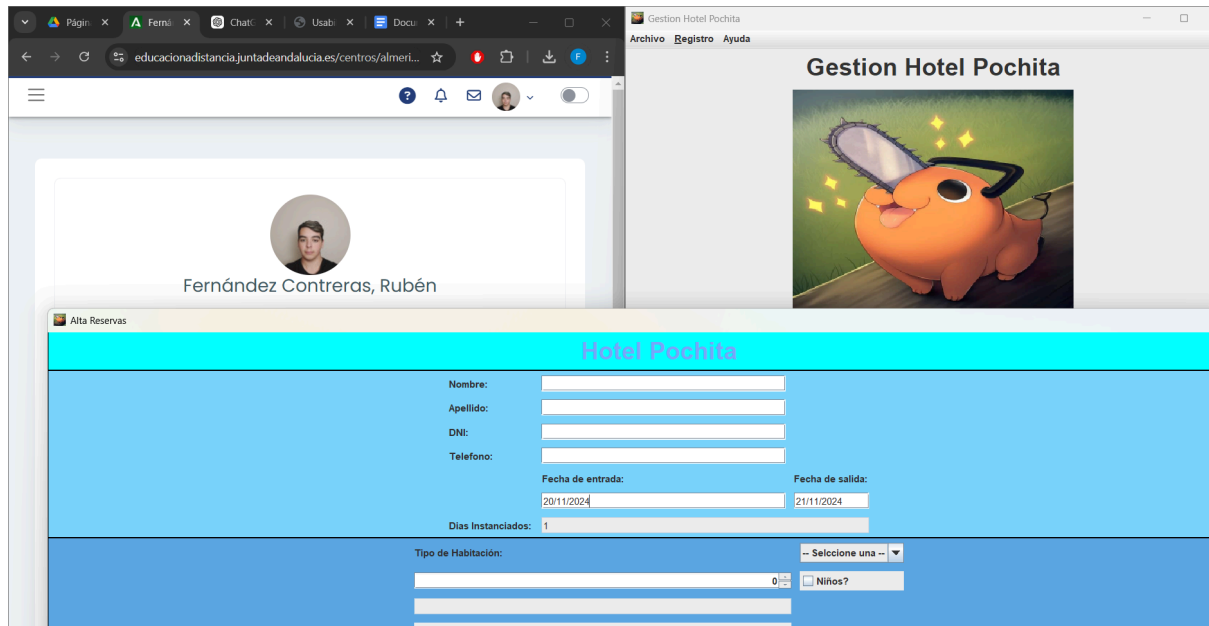
## Indice:

<b>1. Visibilidad del estado del sistema</b>	<b>2</b>
<b>2. Coincidencia entre el sistema y el mundo real</b>	<b>2</b>
<b>3. Control y libertad del usuario</b>	<b>3</b>
<b>4.Consistencia y estándares.</b>	<b>3</b>
<b>5. Prevención de errores.</b>	<b>4</b>
<b>6. Reconocer en lugar de recordar.</b>	<b>4</b>
<b>7. Flexibilidad y eficiencia de uso.</b>	<b>4</b>
<b>8. Estética y diseño minimalista.</b>	<b>5</b>
<b>9. Ayuda al usuario a reconocer, diagnosticar y recuperarse de los errores.</b>	<b>5</b>
<b>10. Ayuda y documentación.</b>	<b>6</b>

# 1. Visibilidad del estado del sistema

Este principio consiste en mantener a los usuarios actualizados de todas las acciones que hacen sobre la aplicación.

Cuando el usuario selecciona la fecha de entrada o salida se actualiza un campo que indica los días entre estas 2 fechas. Este comportamiento indica al usuario en todo momento cuál es el estado de su instancia.

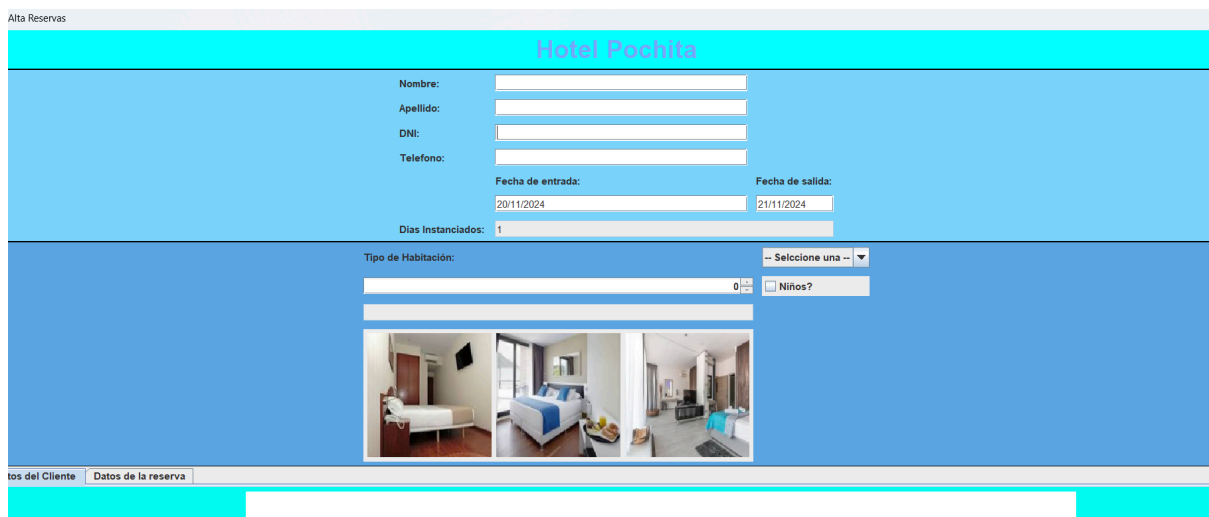


## 2. Coincidencia entre el sistema y el mundo real

Este principio consiste en que haya una cohesión entre el mundo real y el del sistema.

Este se puede ver representado por el simple diseño de la aplicación.

Como se observa en la siguiente captura hay palabras de uso cotidiano como “hotel” y “Fecha de salida” que dan a entender que estamos en un contexto relacionado con una reserva de hotel.

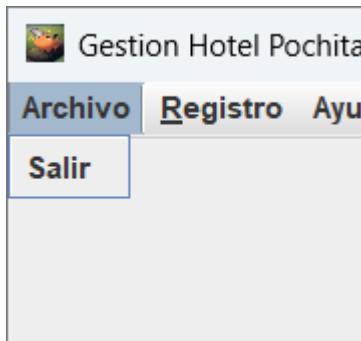


### 3. Control y libertad del usuario

Este principio consiste en que cualquier persona sea capaz de poder entender para qué sirve la aplicación.

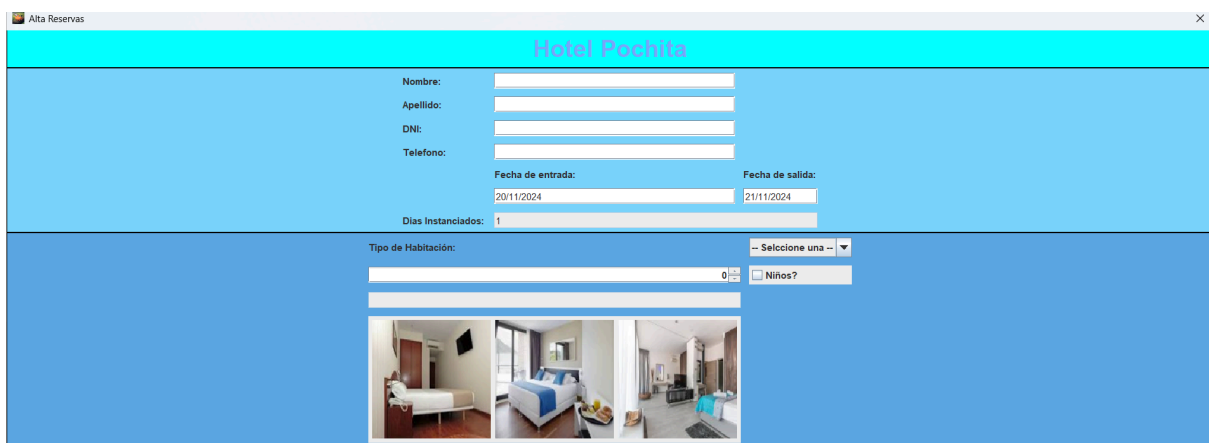
Gracias al uso de palabras simples en la interfaz el usuario es capaz de interpretar qué está pasando en cada acción que hace.

Un ejemplo es la siguiente opción del menú que te propone salir de la aplicación.



### 4. Consistencia y estándares.

Este principio consiste en el simple hecho de que haya unión entre todos los elementos de la interfaz, es decir, que si tienes un panel que tiene el fondo azul, de repente no puede haber uno seguido que sea amarillo porque choca mucho a la vista.



## 5. Prevención de errores.

El diseño de la aplicación debe de estar pensado para evitar que el usuario cometa errores involuntarios.

Un ejemplo sería la siguiente captura que le indica al usuario que le hace falta para tener el campo “teléfono” y “DNI” para que estén correctos.



The screenshot shows a web form titled "Hotel Pochita" with a light blue background. The form contains the following fields and messages:

- Nombre:** A text input field.
- Apellido:** A text input field.
- DNI:** A text input field containing "sadasdasd". To its right, a message reads: "Se requiere 8 números y una letra al final para el Dni."
- Telefono:** A text input field containing "asdasd". To its right, a message reads: "Se requiere 9 números para el teléfono."
- Fecha de entrada:** A text input field containing "20/11/2024".
- Fecha de salida:** A text input field containing "21/11/2024".
- Dias Instanciados:** A text input field containing "1".

## 6. Reconocer en lugar de recordar.

Este principio nos dice que muchas veces una imagen vale más que mil palabras, pues en este caso es así.

Gracias a la implementación de iconos dentro de los botones el usuario es capaz de captar a la primera lo que hace cada botón por el simple hecho de reconocer la imagen.



## 7. Flexibilidad y eficiencia de uso.

No todo el mundo es un experto en atajos o en internet hablando en general.

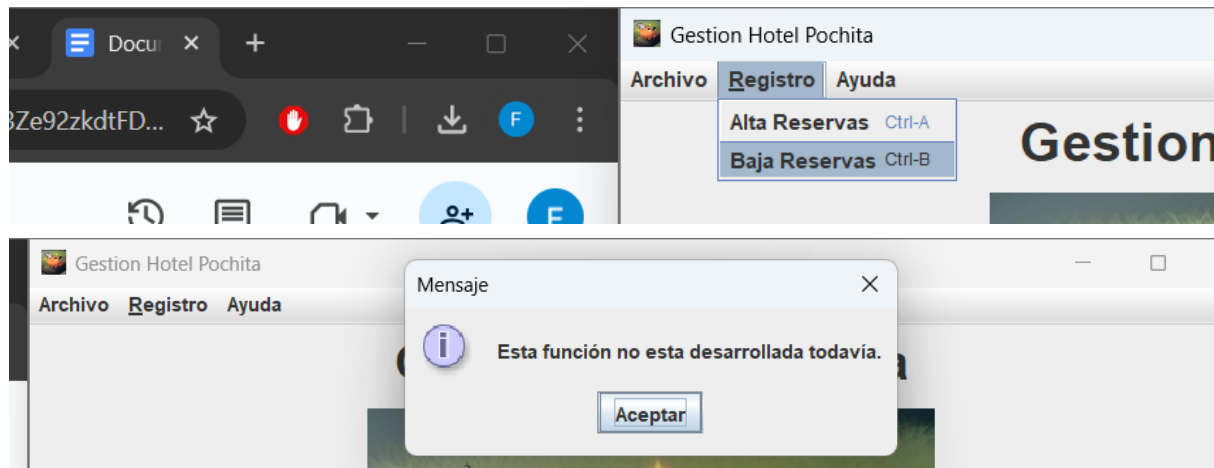
Es por eso que este principio nos habla de que da igual que experiencia tengas, pues si no eres capaz de usar atajos la interfaz proporciona un texto en el menú que te da un indicio de que uso tiene ese apartado como es el caso de la siguiente captura.



## 8. Estética y diseño minimalista.

En una aplicación es innecesario dejar elementos al usuario que no tengan ningún uso. Es por eso que todo lo que haya en la interfaz tiene que tener un fin ya sea decorativo o recogida de datos.

En la siguiente captura se podrá observar que hay una opción que no esta implementada pero se ha solucionado gracias a un mensaje informativo.



## 9. Ayuda al usuario a reconocer, diagnosticar y recuperarse de los errores.

Este principio va de la mano del caso anterior y del principio 5.

Es importante que el usuario sepa en todo momento que está haciendo un buen uso de la aplicación sin llegar a cometer ningún error que pueda provocar la aplicación.

Gracias a una funcionalidad de swing existe una clase llamada Máscaras (Hablado mal y pronto) que consiste en ponerle un filtro a algún campo de texto. Gracias a esto el usuario es incapaz de poner letras en el teléfono.

```
1  ...
2
3  /** The mascara telefono. */
4  MaskFormatter mascaraTelefono;
5
6  /**
```

## 10. Ayuda y documentación.

Siempre viene bien algo de ayuda extra por si se necesita consultar algo.

Es por eso que existe un “método” (no se que se considera) llamado javadoc que te hace un informe sobre todo el código del programa.

> swing_c_p01_parte1_Fernandez	45
> swing_c_p01_parte2_Fernandez	46
▼ swing_c_p2_FernandezContrera	47
> JRE System Library [JavaSE-2	48
▼ src	49
▼ resources	50
doble.jpg	51
guardar.jpg	52
imprimir.png	53
mas.png	54
menos.png	55
nuevo.png	56
pochita.jpg	57
simple.jpg	58
suite.jpg	59
▼ swing_c_p2_FernandezCo	60
> MainClass.java	61
> PanelDatosHabitacion	62
> PanelDatosPersonales	63
> PanelExtraNiños.java	64
> PanelInformacion.java	
> PanelTitulo.java	
> PanelTituloDialogo.java	
> VentanaDialogo.java	
> VentanaPrincipal.java	
> module-info.java	
> doc	
> swing_p01_FernandezContreras	

Console  
MainClass