

Tarea DI03

Usabilidad

Ejercicio 1

Contenido

EJERCICIO 1 Análisis de usabilidad de la interfaz	2
1.- Revisión general del diseño actual.....	2
A) Examina la distribución de los elementos (formularios, botones, tablas, menús, etiquetas, etc.).....	2
B) Evalúa el uso del color, tipografía, iconografía y alineación de los campos en relación con los estándares de diseño coherente.....	2
C) Determina si la interfaz comunica correctamente la jerarquía de la información (qué es más importante, qué está disponible, qué requiere acción).....	3
2.- Aplicación de las heurísticas de Jakob Nielsen.....	4
3.- Valoración de accesibilidad y ergonomía visual:.....	5
A) Determina si los textos son legibles (contraste, tamaño, tipo de fuente, alineación).	5
B) Comprueba si el diseño sería fácilmente interpretable por usuarios con distintos niveles de experiencia.....	6
C) Evalúa si los mensajes de error, confirmación o validación son visibles, comprensibles y coherentes con la acción del usuario.....	6
4.- Conclusión del análisis:.....	7
A) Resume las principales debilidades detectadas.....	7
B) Explica por qué esas carencias afectan a la usabilidad general del proyecto.....	7
C) Justifica, con base teórica, qué heurísticas o pautas de diseño deberían cumplirse para mejorar la experiencia.....	7

EJERCICIO 1 Análisis de usabilidad de la interfaz

1.- Revisión general del diseño actual

A) Examina la distribución de los elementos (formularios, botones, tablas, menús, etiquetas, etc.).

La interfaz se divide en dos paneles o contenedores principales CLIENTE y PRESUPUESTO con dimensiones asimétricas, con anchos y altos distintos rompiendo la armonía visual del aplicativo. En la parte inferior aparecen por un lado dos botones ACEPTAR y SALIR ligeramente desplazados hacia la izquierda, y, por otro lado, existen un botón **add** acompañado de un Spinner desplazados ligeramente hacia la derecha y fuera del panel PRESUPUESTO al cual se intuyen en primera instancia que deberían pertenecer.

El componente panel CLIENTE dispone de varias etiquetas y campos de texto colocados casi al azar sin guardar ningún tipo de correlación entre ellos. Por otra parte, no existen toolTipText definidos y tampoco se encuentra definida la variable text de cada uno de los campos de texto.

El componente panel PRESUPUESTOS dispone de una tabla de tres campos (Trabajo, Cantidad, Precio), más una etiqueta que representa la imagen de la labor a desempeñar, y un ComboBox que almacena la lista de trabajos disponibles a presupuestar. Aunque guardan un mayor orden y proporcionalidad de dimensiones comparándolo con el panel anterior, la ubicación de cada uno de los componentes no es la más adecuada desde un punto de vista visual.

En todos los casos los elementos se encuentran aparentemente dispersos sin ningún criterio afín, lo que genera dificultad en el flujo visual de la aplicación.

B) Evalúa el uso del color, tipografía, iconografía y alineación de los campos en relación con los estándares de diseño coherente.

La interfaz presenta no más de cuatro o cinco colores, por lo que, resulta adecuado para no romper su limitación de uso como herramienta de comunicación, el uso de más de siete colores puede llevar a confusión. Por otra parte, se han seleccionado colores que garantizan un alto contraste generando una clara diferenciación visual, el color negro para la fuente del tipo de letra y colores como el amarillo claro o el blanco y gris para el fondo. No se observa un uso funcional del color para resaltar información crítica como por ejemplo el botón **add** (se considera una de las acciones críticas que el usuario debe acometer).

Se ha empleado el tipo de letra Segoe UI. Es la tipográfica estándar del sistema operativo para renderizar controles de usuario (botones, etiquetas y otros componentes). Es una fuente de tipo Sans Serif (sin remates), diseñada específicamente para mejorar la legibilidad en pantallas digitales, lo cual es un estándar en el diseño de interfaces de escritorio. También se ha empleado de manera uniforme en toda la interfaz para no generar distorsión en el contenido.

Se han asignado diferentes tamaños y resaltes en negrita para el título de los paneles componentes CLIENTE y PRESUPUESTO, lo cual resulta adecuado para que el usuario identifique rápidamente titulares frente a bloques de texto.

La redacción de algunas etiquetas es inconsistente ("Apell." frente a "Nombre"), lo que dificulta que el usuario identifique rápidamente la relación entre los campos de texto.

Solo se dispone de una imagen representativa del tipo de trabajo a presupuestar, lo cual puede resultar idóneo para no generar dispersión, aunque puede resultar algo monótona al predominar el texto claramente en toda la frame principal del aplicativo.

Los componentes del primer panel CLIENTE no se encuentran correctamente alienados, presentando una falta importante de correlación entre ellos. Por otra parte, la longitud de los campos de texto no representa el tamaño máximo previsto que la aplicación pudiera recibir como datos de entrada. La disposición de componentes no es limpia y están colocados prácticamente al azar, aunque la agrupación es lógica para este panel CLIENTE.

Para el panel PRESUPUESTO existe una correcta alineación de componentes, pero faltan el botón add y el Spinner que se encuentra fuera de éste en la parte inferior de la interfaz. Por otra parte, no existe un flujo natural en la disposición de sus componentes, ni un patrón de lectura lógico (como el Z-pattern).

Conjuntamente, no existe simetría vertical y horizontal de sus componentes lo cual genera un inadecuado flujo visual.

C) Determina si la interfaz comunica correctamente la jerarquía de la información (qué es más importante, qué está disponible, qué requiere acción).

La distribución y flujo es deficiente; los campos de texto están dispuestos al azar, rompiendo el flujo de lectura lógico (Z-pattern) recomendado para descubrir la información en orden.

El uso de contraste tipográfico mediante el uso de negrita en los títulos de los paneles ("CLIENTE", "PRESUPUESTO") es correcto, ayuda a establecer niveles de relevancia y a entender la relación entre bloques.

Falla en el uso de espacios en blanco, ya que existen desproporciones en los tamaños de los componentes que generan desorden visual.

No respeta la agrupación lógica, pues elementos que realizan funciones similares (como el botón Add y el Spinner selector de cantidad) están separados espacialmente del panel de presupuesto al que pertenecen.

Los botones no son totalmente claros; el uso de "Add" (inglés) rompe la consistencia del lenguaje empleado (castellano) y "Aceptar" es una nomenclatura genérica que no describe fielmente la acción de guardar el pedido. Por otra parte, los botones ACEPTAR y SALIR no están situados en la esquina inferior derecha, es decir, no guardan concordancia con la naturalización de escaneo visual.

Carece de mensajes toolTipsText que indiquen al usuario cómo introducir los datos correctamente.

2.- Aplicación de las heurísticas de Jakob Nielsen

- A) El principio evaluado (por ejemplo, "Visibilidad del estado del sistema").
- B) El componente afectado (por ejemplo, botón "Guardar", tabla de pedidos, menú superior...).
- C) La observación o problema detectado.
- D) Una justificación breve del impacto sobre la experiencia del usuario.

1. Consistencia y estándares (H4)

Componente afectado: Botón "Add", etiqueta "Apell." y navegación general.

Observación detectada: Se detecta una falta de consistencia del lenguaje empleado, se mezclan idiomas (botón "Add" en inglés frente a "Aceptar" en castellano) y utilizar abreviaturas no estandarizadas ("Apell.").

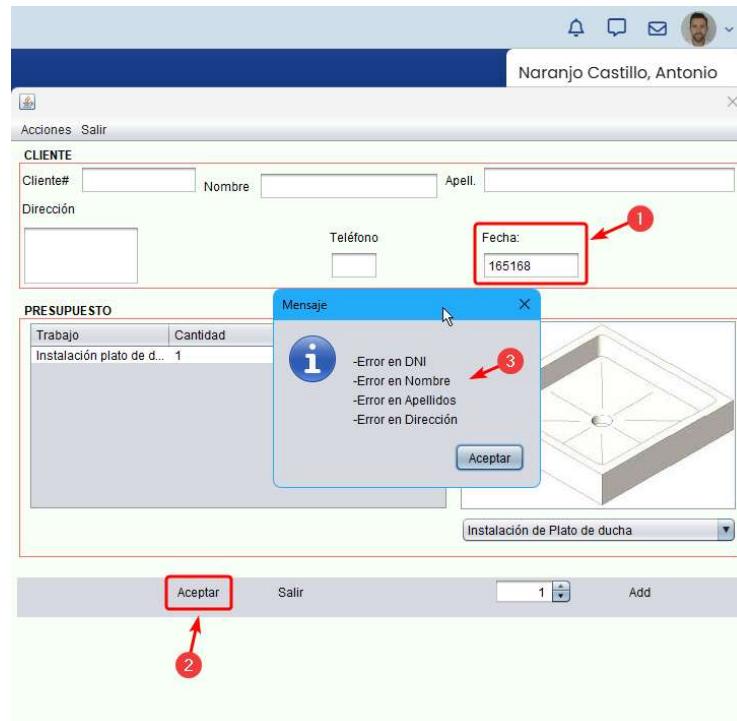
Justificación: La inconsistencia lingüística y de nomenclatura aumenta la carga cognitiva y genera interpretación en el usuario que puede dar lugar a error.

2. Prevención de errores (H5)

Componente afectado: Campo de texto "Fecha".

Observación detectada: El sistema solicita un formato específico (dd/mm/aaaa) en un campo de texto libre, sin ofrecer avisos mientras se escribe sobre si los datos son válidos o no.

Justificación: Es mejor prevenir el error antes de que ocurra que generar mensajes una vez se produzca; un diseño que no valida la entrada en tiempo real induce al usuario al error.



Nota: lo mismo ocurre con el campo de texto Teléfono, solo debería admitir caracteres numéricos (dígitos) cuando no es así. Admite cualquier carácter alfanumérico.

3. Adecuación entre el sistema y el mundo real (H2)

Componente afectado: Panel PRESUPUESTO (reordenación de componentes).

Observación detectada: La disposición actual obliga al usuario a realizar movimientos oculares de ida y vuelta (la tabla a la izquierda, la imagen arriba a la derecha y los controles debajo). Se propone una reordenación siguiendo el Z-pattern: situar el ComboBox y la Imagen en la parte superior izquierda (entrada), el Spinner y el botón Add a la derecha del ComboBox (acción), y la Tabla ocupando toda la parte derecha del panel (resultado), siguiendo una flecha imaginaria de izquierda a derecha.

Justificación: El sistema debe seguir un flujo de lectura natural. Al organizar los elementos de "Entrada → Acción → Resultado", se respeta el orden lógico de escritura y pensamiento de las personas, no crea fatiga visual y el tiempo de aprendizaje es reducido.

4. Ayuda y documentación (H10) - ToolTipText y Prompts

Componente afectado: Todos los TextFields del formulario de cliente.

Observación detectada: Los campos de texto carecen de la propiedad ToolTipText (información emergente) y de texto informativo inicial (propiedad text). El usuario no tiene indicaciones sobre si debe introducir solo números, el formato de la fecha o la longitud máxima de caracteres permitida.

Justificación: Es necesario que la ayuda sea fácil de encontrar y útil. Incluir descripciones emergentes que detallen las reglas de validación permite al usuario resolver dudas de forma inmediata. Hay que evitar que el usuario interprete puesto que será el principio de los errores que pueda cometer.

5. Estética y diseño minimalista (H8)

Componente afectado: Panel "CLIENTE" y disposición de TextFields.

Observación detectada: Existe contenido disperso y campos con dimensiones desproporcionadas que no contribuyen a simplificar la interfaz ni a eliminar el contenido irrelevante.

Justificación: El diseño debe ser minimalista para que el usuario solo se fije en lo realmente importante, la interfaz presenta un desorden visual importante que afecta al rendimiento del usuario.

3.- Valoración de accesibilidad y ergonomía visual:

A) Determina si los textos son legibles (contraste, tamaño, tipo de fuente, alineación).

Se emplea el tipo de fuente Segoe UI (Sans Serif) diseñada específicamente para pantallas, lo que facilita la lectura rápida de caracteres individuales, y, por lo tanto, una tipografía adecuada en entornos de gestión de datos.

El texto negro sobre fondo gris claro, amarillo o blanco ofrece un contraste alto, cumpliendo con los estándares básicos de accesibilidad.

Los tamaños de los textos son adecuados para una aplicación de escritorio estándar.

La alineación irregular de las etiquetas (unas arriba y otras al lado) y la disposición de los TextFields sin un eje vertical común rompen la continuidad del escaneo visual. El usuario no puede leer el formulario de forma secuencial o en Z, sino que debe "saltar" visualmente, aumentando la fatiga ocular y el tiempo de procesamiento.

B) Comprueba si el diseño sería fácilmente interpretable por usuarios con distintos niveles de experiencia.

Usuarios con baja experiencia: La separación entre el selector de trabajos (ComboBox) y el botón de añadir (Add) fuera del panel de presupuesto rompe la secuencia mental o visual "seleccionar -> añadir", lo que puede causar que el usuario se sienta perdido en los primeros usos.

Usuarios con alta experiencia: El diseño no muestra atajos de teclado (ej: Alt+S para Salir o Alt+G para Guardar) ni un orden de tabulación lógico evidente. Un usuario experto busca eficiencia y rapidez, si tiene que usar el ratón para cada pequeña acción, la interfaz se vuelve lenta a largo plazo.

C) Evalúa si los mensajes de error, confirmación o validación son visibles, comprensibles y coherentes con la acción del usuario.

1. Mensajes de Validación y Error (Acción: Aceptar con datos incorrectos)

Al pulsar el botón "Aceptar" con datos que no cumplen las restricciones del proyecto (como el teléfono con caracteres alfabéticos), el sistema despliega una lista de errores.

Su visibilidad es muy alta al interrumpir el flujo hasta que el usuario pulsa "Aceptar".

La compresión es deficiente. El mensaje simplemente aporta una lista de errores sin explicar la causa (ej. El teléfono solo admite números) ni cómo solucionarlo.

No es coherente ya que existe una inconsistencia de nomenclatura. El mensaje cita un "Error en DNI", pero el campo correspondiente en la interfaz se etiqueta como "Cliente#". Esto confunde al usuario al buscar un campo que no existe visualmente, incumpliendo la 4^a Heurística (Consistencia y estándares), en ambos casos se debería sustituir por "Código".

2. Mensajes de Éxito (Acción: Aceptar con datos válidos)

Cuando los campos cumplen las restricciones de longitud y tipo de datos, aparece el mensaje "Registro guardado".

El mensaje es claro y directo. Cumple con la 1^a Heurística (Visibilidad del estado del sistema) al confirmar que la acción de guardado se ha completado correctamente.

Igualmente, también es coherente con la acción del usuario, la aplicación se cierra tras el registro de los datos.

3. Mensajes de Confirmación (Acción: Salir)

Al intentar abandonar la aplicación, se lanza un cuadro de diálogo de decisión.

Altamente visual ya que se utiliza un ícono de interrogación que ayuda a identificar que es una confirmación de seguridad.

Muy comprensible porque el texto "¿Quiere salir sin guardar los cambios?" es ideal, informa claramente de las consecuencias (pérdida de persistencia de los datos introducidos hasta el momento).

Muy coherente, cumple perfectamente con la 3^a Heurística (Libertad y control por el usuario), permitiendo rectificar una pulsación accidental en el botón "Salir".

4.- Conclusión del análisis:

A) Resume las principales debilidades detectadas.

Desorden de componentes: Falta de alineación en los campos del texto del panel CLIENTE y dispersión de controles esenciales para las acciones a realizar por el usuario, como el botón Add o el spinner selector de cantidad fuera del panel PRESUPUESTO.

Inconsistencia en la nomenclatura: Discrepancia entre etiquetas de la interfaz ("Cliente#") y mensajes de error ("DNI"), además del uso de diferentes idiomas ("Add" vs "Aceptar").

Falta de ayuda y documentación (textos emergentes): Los mensajes de error son reactivos y poco descriptivos, indicando qué falló, pero no cómo solucionarlo. Faltaría completarlos indicando para cada caso el motivo del error de manera clara y concisa, pero sobre todo de manera positiva, constructiva y fuera de culpa por parte del usuario.

B) Explica por qué esas carencias afectan a la usabilidad general del proyecto.

Las carencias comentadas afectan negativamente sobre la experiencia del usuario incrementando el esfuerzo mental que el usuario debe realizar para el uso del aplicativo. El usuario no puede predecir el comportamiento del programa ni trabajar de forma fluida, lo que eleva la probabilidad de cometer un error en la entrada de datos. La falta de jerarquía visual obliga al usuario a aprender la interfaz mediante ensayo y error en lugar de usarla de forma intuitiva.

C) Justifica, con base teórica, qué heurísticas o pautas de diseño deberían cumplirse para mejorar la experiencia

Heurística 4ª (Consistencia y estándares): Unificar el lenguaje y los estilos para que el sistema sea predecible.

Heurística 2ª (Adecuación entre el sistema y el mundo real): Organizar el panel PRESUPUESTO siguiendo un Z-pattern, flujo de lectura natural, y, ajustar el ancho de los campos a la longitud real del dato en los textField del panel CLIENTE. En definitiva, establecer una jerarquía de la información llevando a cabo agrupaciones lógicas dentro de un mismo panel para que los botones de acción estén junto a los elementos que modifican.

Heurística 9ª (Ayudar a los usuarios a reconocer, diagnosticar y solucionar los errores) y Heurística 10ª (Ayuda y documentación): Definir e iniciar propiedades como text y toolTipText en los textFields para guiar al usuario sin dar lugar a malas interpretaciones que desemboquen en errores de registro.