

Tarea DI03

Usabilidad

Ejercicio 2

Contenido

EJERCICIO 2 Propuesta de mejora de la interfaz	2
1.- Objetivos de mejora.....	2
2.- Rediseño y aplicación práctica.....	3
3.- Justificación de las mejoras.....	4
A) Argumenta por qué cada modificación mejora la experiencia del usuario.....	4
B) Relaciona tus decisiones con los principios de Nielsen aplicados en el ejercicio anterior.....	4
C) Opcional. Incluye un pequeño esquema o boceto mostrando cómo quedaría la interfaz mejorada.....	5
4.- Implementa las mejoras propuestas:.....	6
A) Implementa esas mejoras que has propuesto. Si quieres, puedes realizarlas en el mismo proyecto que has descargado.....	6

EJERCICIO 2 Propuesta de mejora de la interfaz

1.- Objetivos de mejora

A) *Selecciona, en tu opinión, los 3 fallos más relevantes identificados en el ejercicio anterior.*

B) *Define para cada uno un objetivo de mejora claro (por ejemplo: "Mejorar la visibilidad del estado del sistema al guardar un registro").*

A continuación, se indican los tres fallos elegidos, así como los objetivos de mejora:

1. Inconsistencia en la nomenclatura y lenguaje (Heurística 2^a y 4^a)

Fallo detectado: El uso de etiquetas contradictorias, por ejemplo, en el panel CLIENTE se encuentran etiquetas como "Cliente#" o "Apell." que no representan claramente los datos que se debe incorporar, por otra parte, los mensajes de error aparecen "DNI" en lugar de "Cliente#" que para el caso se debería sustituir por "Código" en ambos casos, sumado a la mezcla de idiomas en la propiedad text de los botones ("Add" frente a "Aceptar" o "Salir").

Objetivo de mejora: Establecer una consistencia terminológica y seleccionar el mismo idioma (castellano) en toda la interfaz para reducir la confusión y la carga cognitiva del usuario.

2. Dispersión funcional → Organización espacial (Heurística 7^a y 8^a)

Fallo detectado: Los controles de entrada del panel PRESUPUESTO (ComboBox, Spinner e imagen) se encuentran desvinculados físicamente de la tabla de resultados, rompiendo el flujo lógico de trabajo Z-Pattern.

Objetivo de mejora: Implementar una agrupación lógica (lectura normal de izquierda a derecha y de arriba a abajo, Z-Pattern) que facilite el flujo de tareas y minimice el movimiento ocular del usuario.

3. Falta de ayuda preventiva y feedback descriptivo (Heurística 5^a y 10^a)

Fallo detectado: Los campos de texto no ofrecen indicaciones de formato antes de escribir, y los mensajes de error son genéricos, indicando el fallo, pero no la solución.

Objetivo de mejora: Incorporar mensajes de ayuda y de diagnóstico claros que permitan al usuario solucionar errores de forma autónoma.

2.- Rediseño y aplicación práctica

A) Describe qué cambios realizarías sobre los componentes implicados

Distribución de paneles y campos.

Etiquetado de botones y menús.

Colores, iconos, fuentes y espaciado.

Uso de diálogos de confirmación y mensajes informativos.

B) Los cambios que propongas no deben alterar la lógica de negocio del proyecto.

1. Distribución de paneles y campos:

Alineación de componentes (Enrejillado): Reorganizar el panel CLIENTE utilizando una estructura de filas y columnas alineadas. Las etiquetas se situarán siempre a la izquierda de cada TextField para reducir el salto visual y facilitar la trazabilidad de los datos a registrar.

Proporcionalidad visual: Ajustar el ancho físico de los TextFields según la longitud del dato, es decir, un campo corto para Cliente# (5 dígitos) y un campo más extendido para Dirección y Apell. (máximo 30 caracteres).

Flujo visual de lectura tradicional en Z: Sobre el panel PRESUPUESTO, situar los selectores (ComboBox y Spinner) y el botón Add (Aregar) en la parte superior izquierda, la imagen de previsualización justo debajo y la tabla ocupando toda la mitad derecha del panel.

2. Etiquetado de Botones y Menús

Unificación del lenguaje: Sustituir el texto del botón "Add" por "Aregar" para mantener la consistencia con el resto de la interfaz en castellano.

Nomenclatura estandarizada: Cambiar la etiqueta "Cliente#" por "Código" y modificar igualmente el texto del mensaje de error "DNI" por "Código", cumpliendo la heurística de consistencia y estándares. Y también, cambiaría el texto "Aceptar" del botón por "Guardar".

Mnémicos: Añadir atajos de teclado en el menú y botones principales (ej: Alt+S para Salir y Alt+A para Agregar) para mejorar la eficiencia de los usuarios familiarizados con la aplicación.

3. Colores, iconos, fuentes y espaciado

Iconografía funcional: Activar la propiedad contentAreaFilled para resaltar los botones respecto al fondo de la interfaz, también sería correcto incorporar iconos universales en los botones (un signo + para agregar, un disco para guardar y una puerta de salida para salir) para facilitar el reconocimiento rápido de cada acción.

Jerarquía tipográfica: Mantener la fuente Segoe UI, pero en los títulos de los paneles CLIENTE y PRESUPUESTO, activar la propiedad bold para visualizarlas en negrita y aumentar el tamaño para diferenciar claramente las secciones (paneles) de los datos de entrada.

Espaciado: Incrementar el margen interno (padding) entre los campos de texto y las etiquetas para evitar el ruido visual y reducir la carga cognitiva, pero que a la vez se vea todo homogéneo y estructurado en filas y columnas imaginarias (enrejillado de componentes).

4. Uso de diálogos y mensajes informativos

Notas emergentes preventivas en tiempo real (toolTipText): Inicializar la propiedad toolTipText en cada componente (sobre todo en los campos de texto) para indicar el formato esperado (ej: "Introduzca 9 dígitos para el teléfono") antes de que el usuario cometa el error.

Mensajes de error descriptivos: Redactar diálogos de error para que no solo indiquen dónde está el fallo, sino cómo solucionarlo (ej: "Error: El nombre no puede contener números").

Consistencia en confirmaciones: mantener el JOptionPane de confirmación al salir, asegurando que la opción por defecto no sea "Sí" para evitar cierres accidentales tras pulsar ENTER, establecer no por defecto.

Flujo de registro post-guardado: Es necesario renombrar el botón "Aceptar" por "Guardar". Asimismo, se debe corregir el comportamiento actual: tras un registro exitoso, el programa no debe finalizar, sino ofrecer al usuario la posibilidad de continuar con el registro siguiente o replicar la lógica existente en el botón "Salir".

3.- Justificación de las mejoras

A) Argumenta por qué cada modificación mejora la experiencia del usuario

1. Unificación de nomenclatura y lenguaje

Al eliminar la mezcla de idiomas ("Add" vs "Aceptar") y modificar las etiquetas "Cliente#" y "Apell." por "Código" y "Apellidos" respectivamente, al igual que ocurre con el mensaje de error que lanza "DNI" cuando tendría que apuntar a "Código", también, sustituir el texto del botón "Aceptar" por "Guardar", de esta manera se elimina la ambigüedad y la interpretación del usuario que puede generar errores de entrada de datos.

2. Reorganización espacial (Patrón en Z)

Agrupar los elementos por su relación lógica (Entrada → Acción → Resultado) mejorando el flujo visual. Al colocar el comboBox selector de trabajos, el spinner selector de cantidad y el botón "Agregar" de izquierda a derecha, se mejora la trazabilidad del aplicativo y la consecución del registro de los datos de entrada en el presupuesto. Luego, quedaría el componente JTable a la derecha para terminar con el flujo visual normal cumpliendo con el patrón Z de lectura tradicional y occidental.

3. Implementación de ayuda contextual (toolTipText)

La propiedad toolTipText actúa como una guía, en lugar de esperar a que el usuario cometa un error y reciba un mensaje de advertencia, el programa le informa de antemano sobre el formato requerido o la cadena de caracteres permitida como dato, de esta manera se estaría aumentando la eficacia del proceso.

B) Relaciona tus decisiones con los principios de Nielsen aplicados en el ejercicio anterior.

1. Unificación de nomenclatura y lenguaje

Se cumple la 4^a Heurística (Consistencia y estándares), que dicta que el usuario no debe preguntarse si diferentes palabras significan lo mismo. También aplica la 2^a Heurística (Adecuación entre el sistema y el mundo real) al hablar el lenguaje del usuario (Castellano).

2. Reorganización espacial (Pattern Z)

Se basa en la Jerarquía Visual y Organización Espacial, asegurando un flujo de lectura natural que minimiza la carga cognitiva del usuario, aplicando así dos principios de Jakob Nielsen 7^a Heurística (Flexibilidad y eficiencia de uso) y la 8^a Heurística (Estética y diseño minimalista).

3. Implementación de ayuda contextual (toolTipText)

Se identifica con la 10^a Heurística (Ayuda y documentación) y la 5^a Heurística (Prevención de errores), priorizando que el error nunca llegue a producirse.

C) Opcional. Incluye un pequeño esquema o boceto mostrando cómo quedaría la interfaz mejorada.

ACCIONES	SALIR						
CLIENTE Código <input type="text"/> Nombre <input type="text"/> Fecha <input type="text"/> Apellidos <input type="text"/> Teléfono <input type="text"/> Dirección <input type="text"/>							
PRESUPUESTO <div style="display: flex; justify-content: space-between;"> <input type="text"/> <input type="button" value="▼"/> <input type="button" value="▲"/> <input type="button" value="Agregar"/> </div> <div style="display: flex; align-items: center;"> <div style="flex: 1; padding: 10px; margin-right: 20px;"> IMAGEN labor a presupuestar </div> <table border="1" style="border-collapse: collapse; width: 30%;"> <thead> <tr> <th style="padding: 5px;">Trabajo</th> <th style="padding: 5px;">Cantidad</th> <th style="padding: 5px;">Precio</th> </tr> </thead> <tbody> <tr> <td style="height: 100px; padding: 5px;"></td> <td style="height: 100px; padding: 5px;"></td> <td style="height: 100px; padding: 5px;"></td> </tr> </tbody> </table> </div>		Trabajo	Cantidad	Precio			
Trabajo	Cantidad	Precio					
<input type="button" value="GUARDAR"/>	<input type="button" value="SALIR"/>						

4.- Implementa las mejoras propuestas:

A) Implementa esas mejoras que has propuesto. Si quieres, puedes realizarlas en el mismo proyecto que has descargado.

CLIENTE

Código	11233	Nombre	Nombre del cliente
Fecha	dd/mm/aaaa	Apellidos	Apellidos del cliente
Teléfono	678112233	Dirección	Dirección del cliente

PRESUPUESTO

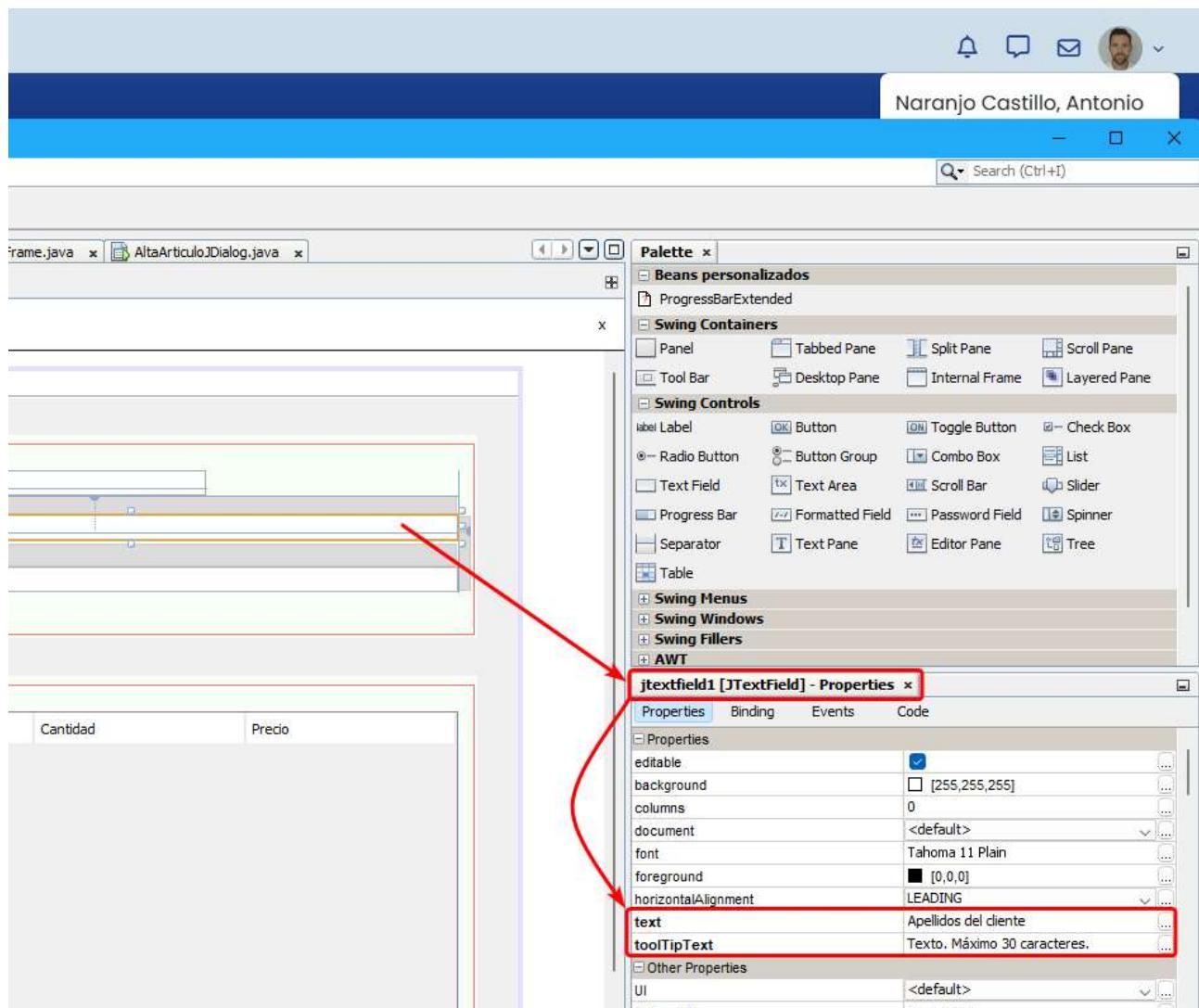
Instalación de Plato de ducha	1	Agregar						
<table border="1"> <thead> <tr> <th>Trabajo</th> <th>Cantidad</th> <th>Precio</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> </tr> </tbody> </table>			Trabajo	Cantidad	Precio			
Trabajo	Cantidad	Precio						

Guardar Salir

En el panel CLIENTE se reorganizan todas las etiquetas y campos de texto, incluso se renombran para expresar un mismo idioma (castellano), así como para la identificación correcta de cada componente, por otra parte, se dimensionan los anchos de los campos de texto guardando cierta relación con el tamaño máximo de la cadena de caracteres que deberán recoger como dato de entrada.

En cuanto al panel PRESUPUESTO, se reorganizan los componentes para facilitar una lectura normal de los mismos siguiendo el criterio Z-Pattern o lectura clásica tradicional de occidente, previamente se renombra el botón "add" por "Agregar"

Por último, el panel de salida, donde el usuario determina si guardar el presupuesto o salir del programa. Igualmente, se renombra el botón Aceptar por Guardar.



En los campos de texto se han alimentado las propiedades **text** y **toolTipText** para que el usuario pueda interpretar correctamente los datos de entrada que se requieren y las condiciones que deben cumplir para evitar errores al guardar, asegurando así la persistencia correcta de los registros.

Para cada textField se ha tenido en cuenta lo indicado en el enunciado de la tarea:

Código: Numérico. Máximo 5 caracteres.

Nombre: Texto. Máximo 20 caracteres.

Apellidos: Texto. Máximo 30 caracteres.

Dirección: Alfanumérico. Máximo 30 caracteres.

Teléfono: Numérico. Máximo 9 caracteres.

Fecha: Formato fecha: dd/mm/aaaa.

The screenshot shows a software interface with two main sections: 'CLIENTE' and 'PRESUPUESTO'.

CLIENTE Section:

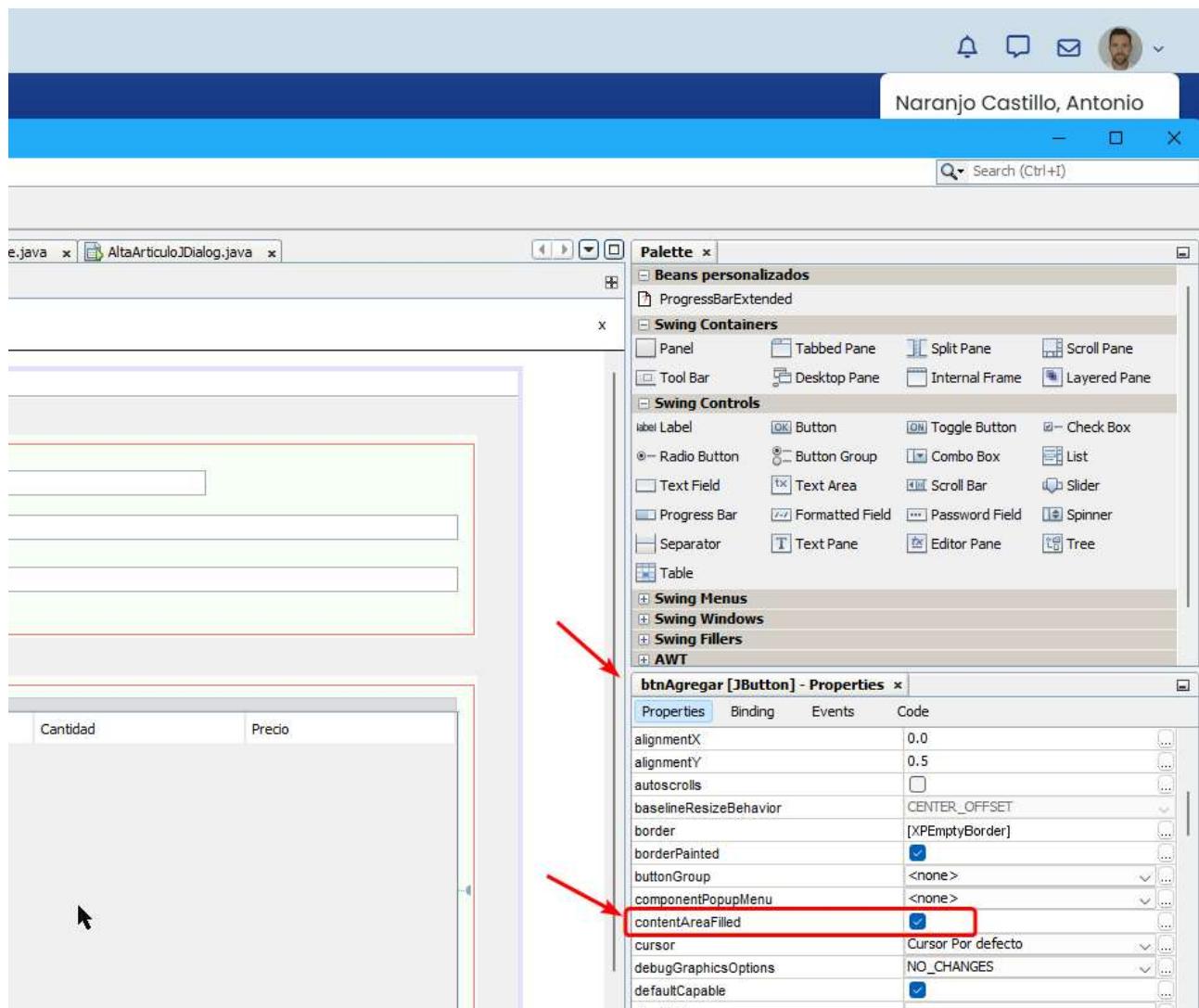
- Código: 11233
- Nombre: Nombre del cliente
- Fecha: dd/mm/aaaa
- Apellidos: Apellidos del cliente
- Teléfono: 678112233
- Dirección: Dirección del cliente I
- Alfanumérico. Máximo 30 caracteres. 1

PRESUPUESTO Section:

- Instalación de Plato de ducha
- 1 2
- Trabajo Cantidad Precio
-

3

También se cambia de color de fondo del botón "Agregar", se establece un color azul claro, de esta manera se identifica claramente la acción que debe acometer el usuario, y, también en el panel inferior, se resaltan los botones "Guardar" y "Salir" respecto al fondo del panel, para igualmente identificar visualmente las acciones que puede llevar a cabo el usuario. Para ello, se activa la propiedad contentAreaFilled en todos los botones.



```

461
462
463     private void btnGuardarActionPerformed(java.awt.event.ActionEvent evt) {
464         // TODO add your handling code here:
465         String errores="";
466         String codigo=txtCodigo.getText().toUpperCase();
467         String nombre=txtNombre.getText().toUpperCase();
468         String apellidos=jTextField1.getText().toUpperCase();
469         String direccion=txtDireccion.getText().toUpperCase();
470         String telefono=txtFecha.getText();
471
472
473
474         if (!codigo.matches("\\d*") || codigo.isEmpty()) errores+="\n-Error en Código";
475         if (!nombre.matches("[A-Z]*[\\s[A-Z]*]")) || nombre.isEmpty()) errores+="\n-Error en Nombre";
476         if (!apellidos.matches("[A-Z]*[\\s[A-Z]*]")) || apellidos.isEmpty()) errores+="\n-Error en Apellidos";
477         if (!direccion.matches("[A-Z]*[\\s[A-Z]*][0-9]*") || direccion.isEmpty()) errores+="\n-Error en Dirección";
478         if (!telefono.matches("[0-9]*") || telefono.isEmpty()) errores+="\n-Error en Teléfono";
479
480
481
482         if(errores.isEmpty()){
483             System.out.println("Registro Guardado");
484             JOptionPane.showMessageDialog(null,"Registro guardado.");
485             limpiarFormulario();
486             //System.exit(0); 1
487         }
488         else{
489             System.out.println("Errores: " + errores);
490             JOptionPane.showMessageDialog(null,errores);
491         }
492     }
493 
```

Por otra parte, se decide modificar el código del método que se lleva a cabo tras pulsar el botón Guardar, dejando comentado la salida del programa `System.exit(0)`, ya que no tiene sentido salir del programa tras guardar, o al menos tal y como se ha interpretado la funcionalidad del programa. Aunque también es cierto que se podría haber renombrado el texto del botón "Aceptar" por "Guardar y Salir", pero como ya existe un botón para "Salir" del programa, se ha decidido que el botón "Aceptar" (Guardar) solo ejecute el guardado de los datos.

También se modifica el texto de salida de los errores para modificar "DNI" por "Código" tal y como se muestra en la imagen.

Para finalizar, cabe destacar la siguiente mejora y la aclaración de porqué no se ha llevado a cabo: También se podría haber introducido un campo más en la componente JTable denominado IMPORTE, de tal manera que se muestre el producto de los campos CANTIDAD y PRECIO, quedando un presupuesto más transparente de cara al usuario, incluso una etiqueta bajo dicho campo IMPORTE que recogiera la suma total de todos los importes de cada línea. Esta última mejora no se ha implementado dando por hecho que la tarea se centra en los componentes que muestra la interfaz, pero no en unos nuevos añadidos, ya que una aplicación nunca se puede dar por concluida y siempre pueden surgir aspectos de mejoras para tener en cuenta en futuras actualizaciones. Digamos que se ha interpretado que la tarea se centra en esta exclusiva versión del aplicativo y no en futuras versiones.