

# Tarea online para la Unidad DI01

---

Título de la tarea: Tarea online para la Unidad DI01

Unidad N° 1: Confección de interfaces de usuario

Ciclo formativo y módulo: Desarrollo de Aplicaciones Multiplataforma - Desarrollo de Interfaces

Curso académico: 2025/2026

## ¿Qué contenidos o resultados de aprendizaje trabajaremos?

**RA 1.-** Genera interfaces gráficos de usuario mediante editores visuales utilizando las funcionalidades del editor y adaptando el código generado.

# 1.- Descripción de la tarea



## Caso práctico



[oatsy40](#)

La empresa BK comienza el desarrollo de las interfaces para una aplicación de gestión de Tiendas de Informática y solicitan tu ayuda para completar las interfaces que necesitan.

## ¿Qué te pedimos que hagas?

Se desea implementar una interfaz que **para la introducción de artículos de informática** en su sistema de gestión de inventario. El nombre de la empresa es **"ATM Informática"**.

Necesitan un formulario que permita dar de alta **productos de distintas categorías** (ordenadores, periféricos, componentes, etc.) y registrar toda la información necesaria para su control de stock.

Debes implementar una interfaz en **NetBeans** que se utilice para **recoger los datos de los nuevos artículos informáticos** que comercializa la empresa **ATM Informática**.

Crea un **proyecto en NetBeans** con nombre: **Tarea01\_[Tus iniciales\_2025]**, que contenga:

- **Formulario principal (JFrame)**
- **Diálogo para alta de artículos (JDialog)**

Para ello, crea un **proyecto** en NetBeans cuyo nombre sea **"Tarea01\_[Tus iniciales\_2025]"**. Dicho proyecto se compondrá de **dos formularios**, uno principal (JFrame) y otro que actuará como diálogo (JDialog) para la recogida de los datos que a continuación se detallan.

### FORMULARIO PRINCIPAL

El formulario principal debe mostrar:

- **Menú principal** con opción **"Gestión de Inventario"** y subopciones:
  - **Añadir Artículo**
  - **Salir**
- **Imagen representativa** de la empresa (propia o de Internet como logotipo de la empresa).
- Un **botón "Añadir Artículo"** que abra el formulario para introducir un nuevo producto.

El formulario principal de la aplicación deberá mostrar el formulario para la gestión del inventario tanto empleando la opción del menú correspondiente como el botón **“Añadir Artículo”**.

### **FORMULARIO PARA LA INTRODUCCION DE NUEVO ARTÍCULO**

A este formulario se accederá **pulsando el botón del formulario principal o seleccionando la opción del menú "Añadir Artículo"**. En el mismo se solicitarán los siguientes datos:

Al pulsar el botón o la opción de menú **“Añadir Artículo”** se abrirá un diálogo (JDialog) que solicitará al menos los siguientes datos:

1. **Código de artículo:** alfanumérico, máximo 10 caracteres.
2. **Nombre del artículo:** texto, máximo 50 caracteres.
3. **Categoría:** lista desplegable con opciones como: *Ordenador, Portátil, Monitor, Impresora, Accesorio, Componente interno, Otro*.
4. **Precio unitario:** número con 2 decimales.
5. **Stock disponible:** número entero.
6. **Proveedor:** texto máximo 30 caracteres.
7. **Fecha de entrada:** formato dd/mm/aaa. JSpinner
8. **Garantía:** radio buttons para elegir entre *12 meses, 24 meses* o *Sin garantía*.
9. **Estado del artículo** (checkboxes): *Nuevo, Reacondicionado, Oferta especial* (pueden seleccionarse varias).
10. **Compatibilidad:** Indicar sistemas operativos o plataformas compatibles. Al menos una (checkboxes) de: *Windows, Ubuntu, Debian*.
11. **Descripción detallada:** área de texto multilínea (JTextArea) para información extra.

Además, el formulario incluirá:

- **Botón “Guardar”** que valide todos los campos (obligatorios y opcionales) antes de cerrar o limpiar el formulario.
- **Botón “Salir/Cancelar”** con verificación de confirmación (por ejemplo, un JOptionPane que pregunte si desea salir sin guardar).

#### **Requisitos:**

- **Validaciones:**
    - **Todos los campos obligatorios deben verificarse antes de guardar** (por ejemplo, que el precio sea numérico, que no falte el código, etc.).
    - **Mostrar mensajes de error adecuados mediante JOptionPane cuando haya campos incorrectos o vacíos.**
  - **Persistencia temporal:** No es necesario guardar en una base de datos, pero los datos introducidos deben poder mostrarse en consola o en un cuadro de diálogo para verificar que se han recogido correctamente.
1. **Fecha de entrada.** Utiliza un spinner al que deberás modificar el modelo. Formato de fecha: dd/mm/aaaa.

2. **Precio unitario:** tendrá un valor máximo de 10.000 €. No se podrán introducir valores superiores.
3. Botones de **"Guardar"** y **"Salir"**.
  1. El botón **"Guardar"** comprobará que no quede ningún dato por cumplimentar (en caso de que no se haya cumplimentado algún dato deberá advertirlo) y que los datos que haya en todos los componentes son correctos. **Los datos se consideran correctos si ninguno de los campos ha quedado en blanco y cumplen los requisitos y formatos pedidos** . En caso de que se cumpla todo lo señalado, mostrará el mensaje **"Registro guardado"** y, tras esto, la **interfaz debe volver a su estado original** (para ello, se deberán borrar los campos del interfaz establecer las opciones de los elementos seleccionables a su estado por defecto).
  2. El botón **"Salir"** deberá comprobar si se ha introducido algún campo de la interfaz
    1. Si no se ha introducido ningún dato, se cerrará el cuadro de diálogo y volverá al formulario principal sin preguntar.
    2. Si el usuario **ha introducido algún dato** se le preguntará **"Está seguro de Salir"**.
      1. Si el usuario responde **afirmativamente**, **se cerrará** el cuadro de diálogo y volverá al formulario principal.
      2. Si el usuario responde que **no desea salir**, **la interfaz se quedará como está**.

### **CONSIDERACIONES ADICIONALES SOBRE EL DESARROLLO DE LA INTERFAZ**

Además de añadir los componentes que necesites para darle funcionalidad deberás cumplir con los siguientes requisitos no funcionales:

- ✔ Debes cambiar el nombre y añadir un `ToolTipText` a los componentes.
- ✔ Modifica el formato de texto de las etiquetas y añade separadores y paneles con título para aportar vistosidad a la interfaz.
- ✔ Utiliza el modo de diseño libre para colocar los componentes en el diálogo. Cuida que la composición quede armónica. Para ello debes usar varios paneles para que los controles queden mejor distribuidos y organizados.
- ✔ Las interfaces (`JFrame` y `Jdialog`) se crearán en clases distintas y sus componentes deben poder ser visualizados en modo diseño de cada interfaz.

## 2.- Información de interés

### Recursos necesarios y recomendaciones

Para el diseño de la interfaz y el desarrollo de la aplicación se usará **NetBeans 8**.

**Se aconseja hacer un diseño previo de la interfaz** (si es necesario, hacer un boceto en papel) y estudiar dónde se van a posicionar los elementos de los formularios.

### Guía para realizar la tarea.

#### 1.Preparar el proyecto en NetBeans

- Sección: 3.1 NetBeans. Contenidos: Crear proyecto, abrir el diseñador visual, usar “Inspector” y ventana “Propiedades” para formularios JFrame y JDialog.
- Aplicación en la tarea: Punto de partida para Tarea0 y creación de las dos ventanas (principal y diálogo).

#### 2. Inserción y personalización de componentes

- Sección: 5.2 Modificación de propiedades. Contenidos: Cambiar nombre de variables. Añadir ToolTipText. Cambiar colores, fuentes y formato de texto. Ajustar propiedades específicas de cada componente, incluido el modelo del JSpinner.
- Sección: 5.4 Ubicación y alineamiento de los componentes. Contenidos: Uso del diseño libre, separadores, paneles con título y disposición armónica.
- Sección opcional (no se solicita en esta tarea pero sí se pedirá en posteriores: 5.5 Enlace de componentes a orígenes de datos. Contenidos: Enlazar datos a JComboBox o listas si se desea carga dinámica.

#### 3. Controles específicos y dónde consultarlos

Componente en la Tarea	Sección	Aplicación
JTextField código, nombre, proveedor.	5.2	Campos de texto:
JCheckBox Nuevo, Reacondicionado, Oferta.	5.2	Estados del artículo:
JRadioButton + ButtonGroup Sin garantía.	5.2	Garantía: 12/24 meses o
JComboBox Portátil, etc.	5.2 + 5.5	Categoría: Ordenador,
JSpinner (fecha de entrada) el modelo a SpinnerDateModel y configurar DateEditor con patrón dd/MM/yyyy.	5.2	Crear JSpinner, cambiar

JDialog  
artículos.

5.2

Formulario para alta de

JButton  
Salir, con validaciones.

5.2

Botones Guardar y

#### 4. Eventos y validaciones.

- Sección: 6. Asociación de acciones a eventos. Contenidos: Añadir ActionListener a botones (Guardar/Salir). Uso de JOptionPane para confirmación y mensajes de error. Validación de campos obligatorios antes de guardar.



## Recomendación

Empieza por 3.1 para crear el proyecto y los dos formularios. Dedicar más tiempo a 5.2, ya que aquí aprenderás a: configurar SpinnerDateModel y DateEditor del JSpinner para el formato dd/mm/yyyy. Ajustar tooltips y nombres de todos los componentes. Usa la sección 6 para programar la lógica de los botones Guardar y Salir con validación y mensajes de confirmación.



## Indicaciones de entrega

Una vez realizada la tarea tendrás que subir el proyecto a la plataforma. La estructura de archivos a entregar es como sigue:

- ✓ El directorio con el proyecto *NetBeans* que has realizado.

Si quieres hacer algunas observaciones al profesor o profesora, puedes incluirlas en un archivo llamado **leeme** situado en el directorio raíz del proyecto.

El envío se realizará a través de la plataforma de la forma establecida para ello, y el archivo se nombrará siguiendo las siguientes pautas:

**Tarea01\_[Tus iniciales\_2025]**

### 3.- Evaluación de la tarea

#### Criterios de evaluación implicados

RA1:

- 1.a) Se ha creado un interfaz gráfico utilizando los asistentes de un editor visual.
- 1.b) Se han utilizado las funciones del editor para ubicar los componentes del interfaz.
- 1.c) Se han modificado las propiedades de los componentes para adecuarlas a las necesidades de la aplicación.
- 1.d) Se han asociado a los eventos las acciones correspondientes.
- 1.e) Se han enlazado componentes a orígenes de datos.
- 1.f) Se ha analizado el código generado por el editor visual.
- 1.g) Se ha modificado el código generado por el editor visual.
- 1.h) Se ha desarrollado una aplicación que incluye el interfaz gráfico obtenido.

#### ¿Cómo valoramos y puntuamos tu tarea?

La tarea se valorará sobre un máximo de 10 puntos repartidos de la siguiente forma:

#### Rúbrica de la tarea

Criterios de evaluación	Elementos de la rúbrica		Puntuación
RA1.a	Creación de la interfaz (hasta 2 puntos)	✓ Creación de la interfaz empleando los componentes más adecuados:	Hasta 2 pts
RA1.b	Disribución de elementos (hasta 2 puntos)	✓ Distribución de elementos adecuada usando el modo de distribución libre y utilización de paneles para la organización de contenidos.	Hasta 2 pts

<b>RA1.c</b>	<b>Propiedades de los componentes (hasta 2 puntos)</b>	<p>✔ Modificación de las propiedades de los componentes para que se adapten a lo solicitado en la tarea (adaptación de componentes a formatos y valores que deben contener, fuentes de las etiquetas, etc...).</p>	<b>Hasta 2 pts</b>
<b>RA1.d,RA1.e,RA1.f,RA1.g</b>	<b>Eventos y Métodos (hasta 3 puntos)</b>	<p>✔ Añadir la gestión de eventos e implementación de los métodos necesarios para el funcionamiento adecuado de los componentes que interactúan con el usuario.</p>	<b>Hasta 3 pts</b>
<b>RA1.h</b>	<b>Funcionamiento (hasta 1 puntos)</b>	<p>✔ Funcionamiento adecuado de la aplicación (información al usuario ante posibles errores, interfaz amigable, diseño elaborado, etc).</p>	<b>Hasta 1 pto</b>



# Anexo. Licencia de recursos

**Nota:**

Para esta tarea no es necesario incluir en este anexo la cita de las licencias de ningún recurso. Este anexo incluye exclusivamente recursos cuya licencia obliga a la cita expresa y que, por la extensión de sus credenciales o por cualquier cuestión técnica, no se pueden citar en línea. El resto de recursos son de elaboración propia y no requieren cita expresa, quedando su licencia expresada en la de los propios materiales o citada en línea, en el lugar donde aparecen en los contenidos.

**Licencias de recursos utilizados en la tarea**

Recurso	Datos del recurso	Recurso	Datos del recurso
	Autoría: Licencia: Procedencia:		Autoría: Licencia: Procedencia:
	Autoría: Licencia: Procedencia:		Autoría: Licencia: Procedencia:
	Autoría: Licencia: Procedencia:		Autoría: Licencia: Procedencia:
	Autoría: Licencia: Procedencia:		Autoría: Licencia: Procedencia:
	Autoría: Licencia: Procedencia:		Autoría: Licencia: Procedencia:

Licencia: otras licencias de software libres