

Tarea online 04

Título de la tarea: Utilizando cadenas de caracteres y arrays.

Unidad: 04

Ciclo formativo y módulo: DAM/DAW, Programación.

Curso académico: 2025/26

¿Qué contenidos o resultados de aprendizaje trabajaremos?

En esta tarea se trabaja el resultado de aprendizaje *RA06: "Escribe programas que manipulen información seleccionando y utilizando tipos avanzados de datos"*

Concretamente, se tratan los siguientes criterios de evaluación:

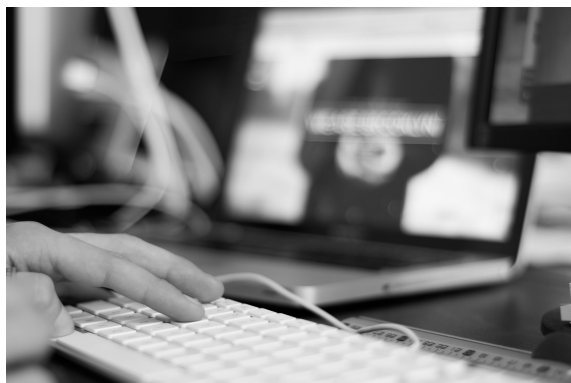
- a. Se han escrito programas que utilicen matrices (arrays).
- h. Se han utilizado expresiones regulares en la búsqueda de patrones en cadenas de texto.

Para ello, se tratan los siguientes contenidos:

- Cadenas de caracteres.
- Arrays.
- Arrays multidimensionales.
- Expresiones regulares

1.- Descripción de la tarea

Caso práctico



[Pixabay](#), [Empresa Informática](#) (Pixabay License)

Ada está repartiendo trabajo entre los distintos componentes del equipo de **Juan**, y a **María** le ha tocado programar varias aplicaciones en las cuales va a tener que usar arrays y expresiones regulares.

Ada, viendo los problemas de seguridad de la empresa ha encargado a **María** que realice una aplicación en Java para la testear y dar una verificación de nivel de seguridad para las contraseñas que utilizan.

—**María**: Vaya lío con los patrones y las expresiones regulares!!

— Por otra parte, **Juan**, ¿tú sabes trabajar con arrays multidimensionales en Java?
... — pregunta **Ada**.

— Buff, he trabajado un poco con ellos... — le contesta **Juan**, un tanto desconcertado.

—Nos han encargado realizar un juego para un programa de televisión de Canal Sur, llamado el "Campo de Regalos" **Juan**, tú eres una máquina y te encargarás de ello.
— Le comenta **Ada**.

—No os preocupéis que estoy preparando una aplicación para ver qué días festivos nos podemos pedir los días de asuntos propios para hacer puente. —
Comenta **Ada** para animar.

¿Qué te pedimos que hagas?

La tarea consiste en resolver **un ejercicio** en el que tendrás que implementar un programa donde pondrás en práctica los conocimientos, técnicas y herramientas vistos en esta unidad (**cadenas**, **arrays** y **expresiones regulares**), junto con todo lo que has ido aprendiendo en unidades anteriores. Antes de empezar a trabajar con el ejercicio, **lee bien el enunciado**, así como los apartados **"Información de interés"** y **"Evaluación de la tarea"**. Es importante que revises esos dos apartados antes de empezar a resolver el ejercicio y antes de llevar a cabo la entrega para asegurarte de que **cumples con todo lo que ha pedido**. Revisa bien el apartado de

evaluación y asegúrate de que estás entregando lo que se te pide y de la forma que se te pide (tipos y formato de los documentos, uno o varios proyectos, clases o archivos Java, etc.).

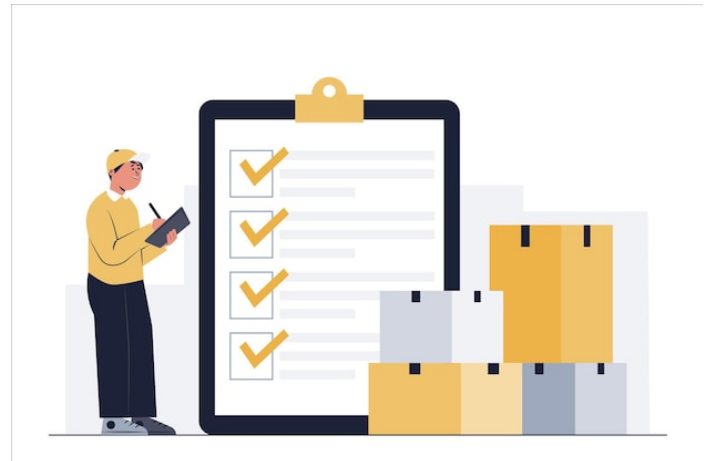
1.1.- Ejercicio 1: Inventario dispositivos

Vamos a realizar un programa en Java que gestione el inventario de dispositivos informáticos de una tienda.

Toda la información deberá almacenarse en una matriz bidimensional, donde cada fila representará un dispositivo y cada columna cada uno de sus atributos.

Registro inicial de dispositivos:

- Solicitar cuántos dispositivos se desea registrar. Deberá seguir solicitándolo hasta que se inserte un número positivo.
- Para cada dispositivo, pedir:
 - Código del dispositivo (DISP-XXXX, donde X serán números. El programa deberá validar mediante una expresión regular que el código completo es correcto. No pueden existir dos códigos iguales).
 - Nombre del dispositivo.
 - Tipo (Ordenador, Monitor, Router, Impresora).
 - Estado (Disponibile, Reparación, Vendido).
 - Precio, almacenado como texto pero que será necesario convertir a decimal al realizar operaciones.



[Freepick](#)

** Para facilitar la entrada de Tipo y Estado, solo será necesario indicar la inicial para cada opción (O para Ordenador, R para Router,...). Ver ejemplos de ejecución**

- Si el código ya existe, debe mostrar un mensaje indicándolo.
- Si alguno de los datos no es válido, el programa debe volver a solicitarlo.

Una vez introducidos los datos, el programa debe mostrar un **menú de opciones**, que permita realizar las siguientes operaciones:

Opción 1: Mostrar el inventario completo

- Mostrar en pantalla todos los dispositivos registrados en formato de tabla clara y legible.

Opción 2: Buscar precio de un dispositivo por su código

- Solicitar el código y mostrar el precio del dispositivo registrado cuyo código coincida exactamente con el introducido.

Opción 3: Salir

- Terminar el programa.

El programa debe cumplir con lo siguiente:

- Usar una matriz bidimensional para almacenar todas las filas y columnas de información del inventario.

- Debe validar el formato del código mediante una expresión regular.
- Debe verificar que el código introducido no existe ya en el inventario o esté duplicado.
- No debe permitir valores no válidos para el tipo y el estado del dispositivo.
- El precio debe ser un valor numérico.
- El menú deberá repetirse mientras que el usuario no seleccione la opción de salida.
- Los mensajes y resultados deben mostrarse de forma clara, estructurada y comprensible.
- Se ha de mostrar la salida formateada en aquellos casos donde sea adecuado (printf).
- El programa debe solicitar que vuelva a introducir los datos hasta hacerlo correctamente.

Resultado del Ejercicio

A continuación os mostramos ejemplos de ejecuciones del programa:

Al arrancar el programa deberemos indicar el número de dispositivos a introducir. (Se muestra también el registro de uno dispositivo correctamente).

```
run:
Ejercicio 1: Inventario de dispositivos

-----

¿Cuántos dispositivos desea registrar?.
3

--- Dispositivo 1 ---
Introduzca código (DISP-XXXX): DISP-1234
Introduzca el nombre del dispositivo: HP-335
Tipo (O->Ordenador/ R->Router/M->Monitor/I->Impresora): O
Estado (D->Disponible/R->Reparación/V->Vendido): D
Precio: 1290
```

El programa deberá indicarnos el mensaje de error cuando se inserte un código ya existente, y de repetir el dato hasta que se introduzca correctamente. Ejemplo de tratar de introducir un dispositivo con varios datos erróneos.

```

--- Dispositivo 2 ---
Introduzca código (DISP-XXXX): DISP-1234
--Código ya existente--
Introduzca código (DISP-XXXX): DISP-123
Introduzca código (DISP-XXXX): DISP-1235
Introduzca el nombre del dispositivo: ASUS-44
Tipo (O->Ordenador/ R->Router/M->Monitor/I->Impresora): L
Tipo (O->Ordenador/ R->Router/M->Monitor/I->Impresora): m
Estado (D->Disponible/R->Reparación/V->Vendido): h
Estado (D->Disponible/R->Reparación/V->Vendido): V
Precio: AAA
Error: Debes introducir un número válido.
Precio: 350

```

¿Cómo debe verse nuestro programa al pulsar la opción 1: Mostrar inventario completo.?

```

Menú de opciones:
1. Mostrar el inventario completo.
2. Buscar dispositivos por código.
3. Salir

Seleccione una opción: 1

--- INVENTARIO COMPLETO ---

CÓDIGO          | NOMBRE          | TIPO  | ESTADO | PRECIO
-----
DISP-1234       | HP-335          | O      | D      | 1290.0 €
DISP-1235       | ASUS-44         | M      | V      | 350.0 €
DISP-1236       | RT-4500         | R      | R      | 90.0 €

```

A la hora de seleccionar la opción 2, para conocer el precio de un dispositivo pueden darse estos casos:

```
Menú de opciones:
1. Mostrar el inventario completo.
2. Buscar dispositivos por código.
3. Salir

Seleccione una opción: 2
Introduzca el código del dispositivo a buscar su precio (DISP-XXXX): DISP-0000

--- No existe ningun dispositivo con ese código ---

Menú de opciones:
1. Mostrar el inventario completo.
2. Buscar dispositivos por código.
3. Salir

Seleccione una opción: 2
Introduzca el código del dispositivo a buscar su precio (DISP-XXXX): DISP-1235
PRECIO: 350.0 €
```

Por último, el menú se seguirá mostrando hasta que se pulsa la opción de salir:

```
Menú de opciones:
1. Mostrar el inventario completo.
2. Buscar dispositivos por código.
3. Salir

Seleccione una opción: 8
Opción no válida. Por favor, elige una opción correcta.

Menú de opciones:
1. Mostrar el inventario completo.
2. Buscar dispositivos por código.
3. Salir

Seleccione una opción: 3
Saliendo del programa...
BUILD SUCCESSFUL (total time: 3 minutes 0 seconds)
```

2.- Información de interés.

Recursos necesarios y recomendaciones

✓ Para la realización de la tarea tendrás que usar el proyecto base que os proporcionamos a continuación: [Proyecto base para la tarea 4.](#)

✓ Si al realizar operaciones sobre cadenas o arrays no se controlan apropiadamente los límites, puede saltar una excepción (`IndexOutOfBoundsException`, por ejemplo). Esas excepciones no deberían producirse y reflejarían un mal diseño del código. En el código no debe usarse un "try-catch" para evitar dichas excepciones, sino que debe corregirse adecuadamente para que no ocurran.



[mohamed Hassan](#) (Licencia Pixabay)

✓ Te recordamos una vez más que debes **indentar** tu código de forma adecuada (te lo llevamos indicando desde la primera tarea). Es algo que se tendrá en cuenta a la hora de evaluar la tarea. No olvides que NetBeans puede hacer esto por ti (selecciona el código a "formatear" o "embellecer" y pulsa `Alt+Mayús.+F`). El aspecto visual de la estructura del código que proporciona una buena indentación mejora muchísimo la **legibilidad**.

✓ Para construir tus **patrones** de coincidencia mediante **expresiones regulares**, te aconsejamos que repases los contenidos dedicados a expresiones regulares en la unidad. También puedes utilizar alguna de las siguientes páginas (o cualquier otra sobre ese tema) para verificar si tus expresiones regulares se comportan como lo que tú quieres que lo hagan:

- [Free Online Java Regular Expression Tester.](#)
- [RegExr.](#)

Indicaciones de entrega

Una vez completados todos los ejercicios de la tarea, el envío se realizará a través de la plataforma. Comprime la carpeta del proyecto NetBeans en un fichero .zip y nómbralo siguiendo las siguientes pautas:

Apellido1_Apellido2_Nombre_PROG04_Tarea

Recuerda que **tu proyecto NetBeans también debe llamarse así** (tanto la carpeta donde se encuentra el proyecto como el nombre del proyecto en NetBeans tienes que renombrarlos respecto al nombre original del proyecto base que vas a utilizar). **Si no lo haces, todos los proyectos se llamarán igual y la corrección de tu tarea podría confundirse con la de otra persona.**

Por tanto, **si no entregas tu tarea siguiendo estas reglas, se te devolverá para que lo hagas correctamente** .

3.- Evaluación de la tarea.

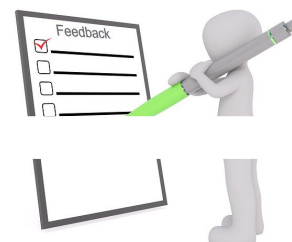
Criterios de evaluación implicados

Del RA6 (Escribe programas que manipulen información seleccionando y utilizando tipos avanzados de datos.):

- ✓ a) Se han escrito programas que utilicen arrays
- ✓ h) Se han utilizado expresiones regulares en la búsqueda de patrones en cadenas de texto.

¿Cómo valoramos y puntuamos tu tarea?

Cada uno de los CE cubiertos por esta tarea será evaluado a través de los criterios de corrección y puntuación indicados detalladamente en la rúbrica de la tarea que puedes consultar a continuación:



[Peggy_Marco](#) (Pixabay License)

ÍTEMS	PUNTUACIONES		
(RA6.a) EJERCICIO 1: Se declara y se instancia correctamente el array bidimensional.	No se declara ni se instancia correctamente el array bidimensional. 0 puntos	No se declara o no se instancia correctamente el array bidimensional. 0.5 puntos	Se declara y se instancia correctamente el array bidimensional. 1 punto
(RA6.a) EJERCICIO 1: Se rellena correctamente el array bidimensional con los dispositivos introducidos.	No se rellena correctamente el array bidimensional. 0 puntos	Se rellena correctamente el array de dispositivos pero no se controla adecuadamente si existen códigos duplicados. 1.25 puntos	Se rellena correctamente el array de dispositivos . 2.5 punto
(RA6.a) EJERCICIO 1: Se muestra correctamente el listado de dispositivos recorriendo el array bidimensional correctamente.	No se muestra correctamente el listado de dispositivos. 0 puntos	Se recorre el array bidimensional pero no se muestra de forma adecuada la información. 1 punto	Se muestra correctamente el listado de dispositivos con el formato especificado. 2 puntos
(RA6.a) EJERCICIO 1: Se realiza la búsqueda de un dispositivo de forma adecuada mostrando su precio si existe.	No se realiza correctamente la búsqueda de un dispositivo. 0 puntos	Se realiza la búsqueda del dispositivo pero no se muestra el precio o el mensaje de producto no encontrado, de forma adecuada. 1.25 puntos	Se realiza la búsqueda de un dispositivo de forma adecuada mostrando su precio si existe. 2.5 puntos

(RA6.h) EJERCICIO 1: Se han creado expresiones regulares correctas para controlar el formato del código del dispositivo.	No se han utilizado expresiones regulares en la búsqueda de patrones. 0 puntos	Se han creado expresiones regulares pero no son adecuadas para controlar el formato especificado para el código del dispositivo. 0.5 puntos	Se han creado expresiones regulares correctas para controlar el formato de código del dispositivo 1 punto
(RA6.h) EJERCICIO 1: Se han utilizado de forma correcta expresiones regulares para controlar la validez del formato del código del dispositivo.	No se utilizan correctamente las expresiones regulares creadas para controlar el formato del código del dispositivo. 0 puntos	Se utilizan correctamente las expresiones regulares creadas para controlar el formato del código del dispositivo. 1 punto	

Tanto en la tarea en general como en la actividad que tenga que desarrollar en particular, se tendrán en cuenta los siguientes puntos de control o elementos evaluables a la hora de corregir tu tarea:

- Se ha creado un único proyecto con todos los ejercicios en él utilizando **el entorno NetBeans 16 y Java Development Kit 17 (JDK)**.
- Los programas **compilan**.
- Se sigue la **estructura** propuesta de **declaración de variables, entrada de datos, procesamiento y salida de resultados**, tal y como se indica en la plantilla y el enunciado.
- Se declaran las **variables necesarias** con el **tipo adecuado** y con **nombres apropiados**.
- No se incluyen **elementos superfluos** ni **innecesarios**, ni se llevan a cabo **operaciones sin sentido**.
- Se declaran **identificadores** que con **nombres significativos o descriptivos** que representen de alguna manera la información que están almacenando para que **el código quede lo más claro, legible y autodocumentado posible**.
- Se llevan a cabo los **cálculos y comprobaciones necesarias** para llevar a cabo las tareas que se pide en cada actividad. Para ello habrá que construir **expresiones apropiadas** y hacer uso de los **operadores correctos**.
- El código de los programas está correctamente **indentado**.
- El código está apropiadamente **comentado**. Se observa una **corrección ortográfica y gramatical**, así como la **coherencia en las expresiones lingüísticas** en los comentarios incluidos.
- Los **identificadores** cumplen con el **convenio** sobre **"asignación de nombres a identificadores"** establecido para el lenguaje Java para constantes, variables, métodos, clases, *etc.*
- En los programas **se muestra la información esperada y correcta por pantalla en un formato apropiado**. Se observa una **corrección ortográfica y gramatical**, así como la coherencia en las expresiones lingüísticas, tanto en los textos de los mensajes que aparezcan en pantalla para pedir información de entrada al usuario como a la hora de mostrar resultados de salida. Se evitan mensajes de entrada de datos y/o salida de resultados inapropiados, descontextualizados, insuficientes o incorrectos.

Aparte de todos estas cuestiones de carácter "técnico", también se valorará en cada uno de los apartados:

- que la respuesta sea personal y no copiada de Internet o de otras fuentes;
- que sea coherente con el resto de respuestas y con los conceptos de la unidad;
- que muestre que se comprenden adecuadamente los conceptos tratados.

IMPORTANTE: Motivos de devolución de la tarea o de calificación mínima

La tarea obtendrá automáticamente una calificación de **cero** si:

- ✓ No se entrega **un único proyecto** que incluya los dos ejercicios que se piden en la tarea utilizando el IDE NetBeans.
- ✓ Alguno de los programas presenta **errores de compilación** impidiendo su ejecución y prueba.
- ✓ **No se respeta el proyecto base** que se entrega al alumno para realizar la tarea.
- ✓ Se utilizan **contenidos** en alguno de los ejercicios del proyecto **que no se han estudiado** en la unidad actual o anteriores a la misma.
- ✓ Se presenta un proyecto total o parcialmente copiado de otro alumno.

Recordatorio: Si una tarea obtiene calificación cero por alguno de los motivos previamente mencionados, será devuelta al alumno. Este tendrá derecho a una segunda entrega únicamente en los siguientes casos:

- Que se trate de la primera vez que presenta la tarea.
- Que la fecha de entrega inicial haya sido al menos una semana antes de la fecha límite establecida para dicha tarea.