

Practica 1. Validar hora.

1. Objetivo

- Poner en práctica algunos de los conceptos estudiados en los primeros temas del módulo: sentencia if-else, métodos de la clase String, operadores de comparación, etc

2. Introducción

Se quiere realizar un programa que nos diga si una hora que introduce el usuario por teclado (hora y minutos) es correcta o incorrecta.

Ejemplo 1:

Introduzca la hora: 20:33

La hora 20:33 es correcta

Ejemplo 2:

Introduzca la hora: 25:33

La hora 25:33 es incorrecta

Además, se dará la opción al usuario de introducir la hora de forma incompleta y el programa se encargará de interpretar qué hora pretende el usuario expresar.

Ejemplo 3:

Introduzca la hora: 1

La hora 01:00 es correcta

Ejemplo 4:

Introduzca la hora: 12

La hora 12:00 es correcta

Ejemplo 5:

Introduzca la hora: 25

La hora 02:50 es correcta

3. El problema.

Se pedirá al usuario que introduzca la hora (hora y minutos), se leerá desde teclado usando el método `nextLine()` de la clase `Scanner` y se almacenará en una variable de tipo `String`.

Supongamos de aquí en adelante que esa variable que contiene lo introducido por el usuario se llama **texto**

A continuación habrá que mostrarle la hora que ha introducido con formato hh:mm (dos dígitos para la hora y dos dígitos para los minutos), y si la hora es correcta o no, tal y como se ha indicado en los ejemplos. Para ello:

- Si **texto** contiene el carácter de dos puntos (:), asumiremos que hasta los ":" está la hora y a partir de ellos , los minutos.
 - Habrá que separar la hora (hasta los :) de los minutos (a partir de los :). Para ello tendrás que utilizar los métodos *indexOf* y *substring*.
 - Una vez separados la hora y minutos habrá que convertirlos a int. Para ello habrá que utilizar el método `Integer.parseInt()`, que convierte un texto a entero.
 - Por último habrá que comprobar si la hora es correcta (entre 0 y 23) y los minutos son correctos (entre 0 y 59), y mostrar al usuario el correspondiente mensaje.
 - **Nota:** Ten en cuenta que aunque **texto** contenga los dos puntos (:), puede que no contenga nada a la izquierda o nada a la derecha. Por ejemplo, las ":23" serían las 12:23, o las "13:" serían las 13:00.
- Si **texto** contiene el carácter de dos puntos (:), interpretaremos cual es la hora y minutos que trata de introducir el usuario:
 - Si ha introducido un único carácter, asumimos que es la hora y completaremos con 0 minutos. P.ej: "7" son las 07:00
 - Si ha introducido dos caracteres
 - Si el valor de lo introducido es una hora correcta (entre 0 y 23), asumiremos que ha introducido la hora y completaremos con 0 minutos. P.ej: "17" son las 17:00
 - Si supera el valor de una hora correcta, el primer carácter será la hora y el segundo carácter (multiplicado por 10) los minutos. Por ejemplo: "25" son las 02:50, "28" son las 02:80
 - Si ha introducido tres caracteres, el primer carácter será la hora y los dos últimos los minutos. P. ej: "112" son las 01:12
 - Si ha introducido 4 caracteres, los dos primeros son la hora y los dos últimos los minutos. P. ej: "1224" son las 12:24
 - **Nota:** Supondremos que el usuario nunca introducirá un texto con más de 5 caracteres, y también que no introducirá caracteres que no sean los dígitos numéricos o el símbolo de dos puntos.

4. La solución.

Abre el entorno de desarrollo y sigue los siguientes pasos:

1. Crea el proyecto *practica1Hora.ApellidoNombre* (sustituyendo “ApellidoNombre” por tu primer apellido y nombre).
2. Crea en él el paquete *practica*
3. Crea la clase *ValidarHora* dentro del paquete *practica* y desarrolla el método *main* con las indicaciones dadas en el apartado anterior.
4. Una vez finalizado, prueba el programa todo lo exhaustivamente que puedas para asegurarte de que está bien.

5. Entrega y calificación

1. Localiza la carpeta del proyecto en tu workspace y comprímela en un archivo. La carpeta se debería llamar *practica1Hora.ApellidoNombre* (sustituyendo “ApellidoNombre” por tu primer apellido y nombre). Si por algún motivo ese no fuera el nombre, renómbrala.
2. Sube el archivo comprimido al aula virtual: Junto al enunciado de la práctica encontrarás un apartado que te permitirá subir el archivo comprimido.
3. En la calificación se valorará:
 - Que el programa funcione correctamente, según se detalla en el enunciado.
 - Que los identificadores del proyecto, paquete y clase coincidan con los indicados en el enunciado (apartado 4) y el archivo entregado, con lo que se indica en el apartado 5.1
 - Que los identificadores de variables tengan nombre adecuados y sigan la notación camelCase (cuando corresponda).
 - Que el código fuente esté correctamente indentado.
 - Que no haya sentencias if-else redundantes o innecesarias, del tipo

```
if (a > 0) {...}
else if (a <= 0) {...}
```
 - El uso de comentarios que puedan resultar útiles para clarificar el código.