



Nombre y Apellidos:

Instrucciones. La tarea constará de diferentes puntos que crearan un modelo de clases y un programa que desarrollaremos en la unidad siguiente.

Abrimos la tarea de tema 3. Vamos a añadirle procesado de cadenas y validación con Expresiones regulares. Finalmente llevaremos los objetos a XML.

Resultados de aprendizaje

RAE 6. Escribe programas que manipulen información, seleccionando y utilizando tipos avanzados de datos.

- a) Se han escrito programas que utilicen arrays.
- b) Se han reconocido las librerías de clases relacionadas con tipos de datos avanzados.
- g) Se han utilizado expresiones regulares en la búsqueda de patrones en cadenas de texto.

Detalles de la tarea de esta unidad.

Enunciado.

En la unidad viste las características fundamentales de la programación orientada a objetos y estudiado los conceptos más importantes de este modelo de programación. Aprendiste a crear y manipular objetos, qué son los métodos y cómo se definen los parámetros.

Ahora es momento de usar **las operaciones con cadenas para formatear la salida, y de crear validaciones con expresiones regulares.**

También tendrás que realizar lecturas por teclado y visualizaciones por pantalla.

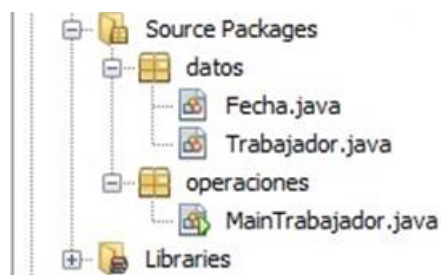
Crea un proyecto Java de nombre **Tarea5NombreApellidos:**

- Copia los paquetes de nombres **datos** y **operaciones**.
- Copia el paquete **datos** las clases **Fecha** y **Trabajador**.
- Copia el paquete **operaciones** la clase **MainTrabajador** para probar las clases anteriores.

El proyecto tiene que tener la siguiente estructura (el nombre será el tuyo):



Nombre y Apellidos:



1º) Modifica la clase Java de nombre **MainTrabajador** con su método **main()** para probar las clases **FeFecha** y **Trabajador**. Usa **StringFormat** y la clase **String** para producir el siguiente resultado de salida . (3,5 puntos).

Indicadores 54,55,66,67

- (0,75 puntos) Se deberá introducir por teclado el nombre, la edad, la categoría, la antigüedad y la fecha de alta (día, mes y año):

```

Introduce el nombre: Alicia
Introduce la categoría (0 a 3):4
Introduce la antigüedad (0 a 2):2
Introduce la edad: 40
FECHA DE ALTA:
Introduce el año: 1998
Introduce el día: 10
Introduce el mes:4
  
```

- (1,25 puntos) Después deberás mostrar la siguiente salida en la consola, donde se muestre si el mes, la categoría y la antigüedad son correctos o no, el nombre de la categoría, nombre de la antigüedad y todos los datos del trabajador, este es el formato de la información de salida:

```

=====
==== CONTROL DE DATOS DE ENTRADA:
    ¿Es correcto el mes:? true
        El nombre del mes es: Mayo
    ¿Es correcta la categoría:? true
        Nombre de la categoría: DIRECTIVO
    ¿Es correcta la antigüedad:?
true        Nombre de la antigüedad:
EXPERTO DATOS DEL TRABAJADOR:
    Alicia, edad: 40
    Categoría: DIRECTIVO
    Antigüedad: EXPERTO
    Fecha de Alta: 10 del mes de Mayo de 1998
  
```



Nombre y Apellidos:

Sueldo: 1571.2

=====
=====

EJEMPLO DE EJECUCIÓN CUANDO LA CATEGORÍA NO ES CORRECTA:

Introduce el nombre: Alicia

Introduce la edad: 40

Introduce la categoría (0 a 3): 5

Introduce la antigüedad (0 a 2): 2

FECHA DE ALTA:

Introduce el día: 10

Introduce el mes: 5

Introduce el año: 1998

=====
=====

==== CONTROL DE DATOS DE ENTRADA:

¿Es correcto el mes:? true

El nombre del mes es: Mayo

¿Es correcta la categoría:? false

Nombre de la categoría: CATEGORÍA ERRÓNEA

¿Es correcta la antigüedad:?

true Nombre de la antigüedad:

EXPERTO DATOS DEL TRABAJADOR:

Alicia, edad: 40

Categoría: CATEGORÍA ERRÓNEA

Antigüedad: EXPERTO

Fecha de Alta: 10 del mes de Mayo de 1998

Sueldo: 1207.0

=====
=====

EJEMPLO DE EJECUCIÓN CUANDO LA ANTIGÜEDAD NO ES CORRECTA:

Introduce el nombre: Alicia

Introduce la edad: 40

Introduce la categoría (0 a 3):

2 Introduce la antigüedad (0 a
2): 4 FECHA DE ALTA:

Introduce el día: 10

Introduce el mes: 5

Introduce el año: 1998

=====
=====

==== CONTROL DE DATOS DE ENTRADA:

¿Es correcto el mes:? true

El nombre del mes es: Mayo



Nombre y Apellidos:

```

¿Es correcta la categoría:? true
Nombre de la categoría: DIRECTIVO
¿Es correcta la antigüedad:? false
Nombre de la antigüedad:
ANTIGÜEDAD ERRÓNEA DATOS DEL
TRABAJADOR:
Alicia, edad: 40
Categoría: DIRECTIVO
Antigüedad: ANTIGÜEDAD ERRÓNEA
Fecha de Alta: 10 del mes de Mayo de 1998
Sueldo: 971.2
=====
=====

```

EJEMPLO DE EJECUCIÓN CUANDO LA ANTIGÜEDAD Y LA CATEGORÍA NO SON CORRECTAS:

```

Introduce el nombre: Ana
Introduce la edad: 40
Introduce la categoría (0 a 3):
5
Introduce la antigüedad (0 a
2): 4 FECHA DE ALTA:
Introduce el día: 10
Introduce el mes: 5
Introduce el año: 1998
=====
===== CONTROL DE DATOS DE ENTRADA:
¿Es correcto el mes:? true
El nombre del mes es: Mayo
¿Es correcta la categoría:? false
Nombre de la categoría: CATEGORÍA ERRÓNEA
¿Es correcta la antigüedad:? false
Nombre de la antigüedad:
ANTIGÜEDAD ERRÓNEA DATOS DEL
TRABAJADOR:
Ana, edad: 40
Categoría: CATEGORÍA ERRÓNEA
Antigüedad: ANTIGÜEDAD ERRÓNEA
Fecha de Alta: 10 del mes de Mayo de 1998
Sueldo: 607.0
=====

```

EJEMPLO DE EJECUCIÓN CUANDO EL MES NO ES CORRECTO:

```

Introduce el nombre: Alicia

```



Nombre y Apellidos:

```
Introduce la edad: 40
Introduce la categoría (0 a 3):
2 Introduce la antigüedad (0 a
2): 2
FECHA DE ALTA:
    Introduce el día: 10
    Introduce el mes: 5
    Introduce el año: 1998
=====
==== CONTROL DE DATOS DE ENTRADA:
    ¿Es correcto el mes:? false
    El nombre del mes es: ERROR
    ¿Es correcta la categoría:? true
    Nombre de la categoría: DIRECTIVO
    ¿Es correcta la antigüedad:?
true    Nombre de la antigüedad:
EXPERTO DATOS DEL TRABAJADOR:
    Alicia, edad: 40
    Categoría: DIRECTIVO
    Antigüedad: EXPERTO
    Fecha de Alta: 10 del mes de ERROR de 1998
    Sueldo: 1571.2
=====
=====
```

2º) Crea un **paquete validación** y una clase **Validaciones.java**. (4 puntos).

Indicadores 54,55, 60, 66 y 67

a) En la clase Validaciones, crea los métodos estáticos

validaNombre(String nombre): usa una expresión regular para que el nombre empiece obligatoriamente por letra mayúscula.

```
Introduce el nombre: Carlos
```

```
Error introduzca el nombre con la primera letra
mayuscula
```

validaFecha(String fecha) : el formato de fecha introducido por pantalla será ahora.



Nombre y Apellidos:

FECHA DE ALTA: Escribela en formato dd/mm/aaaa.

03/11/2002.

En caso de error pedirá que entrega la fecha correctamente

FECHA DE ALTA: Escribela en formato dd/mm/aaaa.

3/11/2002.

Error escriba la fecha correctamente en formato dd/mm/aaaa.

Fecha separarFecha(String): con funciones de cadena separa los tres números y devuelve un tipo Fecha correctamente rellenado.

validaCategoria(int categoría): con una expresión regular se comprueba que la categoría se insertó correctamente. Si no es así pide repetir la inserccion

Introduce la categoría (0 a 3):4

Error introduzca la categoría correcta (0 a 3):

validaAntigüedad(int antigüedad): con una expresión regular se comprueba que la categoría se insertó correctamente. Si no es así pide repetir la inserccion

Introduce la antigüedad (0 a 2):5

Error introduzca la antigüedad correcta (0 a 3):

Ejercicio 3. Manejo de XML. (2,5 puntos).

Añadir un paquete XML y una clase transformarXML.java



Nombre y Apellidos:

Transformar xml tendrá dos métodos.

String trabajadorAXML(Trabajador t): Transformará el trabajador a XML.

<Trabajador>

<Nombre> Carlos </Nombre>

<Edad> 40</Edad>

<Antigüedad>1</Antigüedad>

<Categoria> 3</Categoria>

<FechaAlta> 12/01/2000</FechaAlta>

</Trabajador>

,

Trabajador xmlATrabajador(String xml): Transformará el XML a trabajador.