

[Todos mis módulos](#) / [Mis módulos](#) / [2022-2023](#) / [Desarrollo de aplicaciones multiplataforma](#) / [Programación](#)
/ [Estructura de un programa informático \(10%\)](#) / [PROG01 Tarea Evaluación 01. Realiza un programa en Java \(30%\)](#)

[Descripción](#)[Entrega](#)[Editar](#)[Ver entrega](#)

PROG01 Tarea Evaluación 01. Realiza un programa en Java (30%)

Límite de entrega: viernes, 30 de septiembre de 2022, 23:59

Ficheros requeridos: MoscaMora.java ([Descargar](#))

Tipo de trabajo: Individual

PROG01 TE01 REALIZA UN PROGRAMA EN JAVA (30%)

ENUNCIADO

El objetivo de esta tarea es realizar un programa que visualice por consola una canción acumulativa y realizar su autoevaluación. Una canción acumulativa es aquella cuyos versos se crean a partir de versos anteriores.

Para ello, tendrás que utilizar métodos y la sentencia `println`. En concreto usarás métodos para evitar redundancia de manera que solo tengas que escribir cada línea una vez. Por ejemplo, la línea "Y la mora en su moralito sola" aparece varias veces en la canción pero deberías utilizar un único `println` para mostrarla. No se pueden utilizar métodos que solo contenga una línea de código en su interior.

También deberás usar métodos para definir la estructura de la canción. Por ejemplo, podrías definir un método por cada estrofa de manera que los únicos `println` que aparezcan en el `main` si es que los hay, sean para introducir líneas en blanco.

Además, deberás incluir un comentario al principio con información básica (Nombre, fecha, módulo, UD, Tarea) y la descripción del programa, así como una explicación por cada uno de los métodos que crees.

La canción que se propone es la siguiente pero podéis elegir otra canción, poema, cuento... siempre que sea acumulativo y lo confirméis primero con nosotros. El objetivo es que se muestre tal y como aparece a continuación:

```
Estaba la mora en su lugar,  
Vino la mosca y le hizo mal,  
La mosca a la mora,  
Y la mora en su moralito sola.
```

```
Estaba la mosca en su lugar,  
Vino la araña y le hizo mal,  
La araña a la mosca,  
La mosca a la mora,  
Y la mora en su moralito sola.
```

```
Estaba la araña en su lugar,  
Vino el ratón y le hizo mal,  
El ratón a la araña,  
La araña a la mosca,  
La mosca a la mora,  
Y la mora en su moralito sola.
```

```
Estaba el ratón en su lugar,  
Vino el gato y le hizo mal,  
El gato al ratón,  
El ratón a la araña,  
La araña a la mosca,  
La mosca a la mora,  
Y la mora en su moralito sola.
```

```
Estaba el gato en su lugar,  
Vino el perro y le hizo mal,  
El perro al gato,  
El gato al ratón,  
El ratón a la araña,  
La araña a la mosca,  
La mosca a la mora,  
Y la mora en su moralito sola.
```

En este caso, se os da una primera versión del programa pero con errores. Una vez corregido y modificado para que cumpla con los requisitos exigidos, deberéis realizar su evaluación siguiendo la rúbrica definida en el apartado "Criterios de puntuación".

Autoevaluación

Debéis indicar la nota que creéis que os corresponde en cada apartado y explicar porqué, adjuntando pruebas de vuestro programa (pantallazos de la compilación, de la ejecución o del código del programa) que lo demuestren.

Para ello, podéis realizar un documento de Google Drive de hasta 4 páginas o un videotutorial de hasta 2 minutos. En la cabecera del programa se incluirá el enlace a la autoevaluación.

[AQUÍ](#) un ejemplo de autoevaluación que podéis tomar como referencia.

RECURSOS

Para realizar esta tarea puedes utilizar jGRASP o el editor VPL de Moodle que además te dirá si la salida del programa coincide con la canción sin fijarse en el código utilizado.

INDICACIONES DE DESARROLLO

Sigue el proceso seguido para descomponer un programa utilizando métodos.

Escribe primero el programa sin fijarte en la estructura.

A continuación, estructura el método main en bloques con sentido (por ejemplo estrofas) y crea los métodos necesarios.

Por último, identifica las redundancias y crea los métodos que te permitan eliminarlos.

Recuerda que cada línea se debe escribir una única vez aunque aparezca repetida en la canción y que no podemos usar métodos que contengan una única línea.

Da nombres significativos a métodos, variables y parámetros siguiendo además las reglas de Java. Utiliza correctamente la indentación y los espacios en blanco. Incluye cabeceras explicativas al principio del programa y de cada uno de los métodos, así como junto a cualquier línea de código que lo pueda necesitar.

Guía de estilo ([Referencia](#)):

Da nombres significativos a métodos.

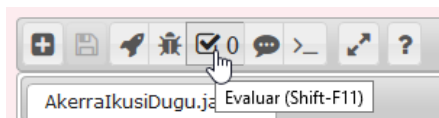
Utiliza correctamente la indentación y los espacios en blanco.

Incluye cabeceras (comentarios) explicativas al principio del programa y al principio de cada uno de los métodos, así como junto a cualquier línea de código que lo pueda necesitar.

INDICACIONES DE ENTREGA

El **programa** se escribirá directamente en el IDE proporcionado o se copiará en él una vez finalizado.

Una vez tengamos el programa en la plataforma, lo podemos ejecutar y ver que funciona (icono del cohete).



También debemos comprobar si la salida de nuestro programa coincide al 100% con la salida propuesta (icono Evaluar). En este caso comparará los 2 texto línea a línea y no tiene en cuenta los espacios en blanco pero sí los fallos de ortografía (tildes, mayúsculas...).

En la **cabecera**, además de la información que describe el programa, se añadirá el enlace a la evaluación realizada, un enlace a Google Drive o a Youtube según la plataforma a la que hayáis subido la autoevaluación.

CRITERIOS DE CALIFICACIÓN

La tarea se evaluará siguiendo la siguiente rúbrica:

Funcionalidad (60%).

Comentarios (15%).

Legibilidad (25%).

Con esta rubrica se calculará la nota de vuestro programa y se comparará con la que habéis indicado en la autoevaluación.

Si la diferencia es menor de un punto, se calculará la nota media entre las dos.

Si la diferencia es mayor, se utilizará directamente la calificación de la profesora.

En los casos en los que falte la autoevaluación, se utilizará la calificación de la profesora con una penalización de un punto.

Ficheros requeridos

MoscaMora.java

```
1 package principal;
2
3 public class MoscaMora {
4
5     public static void main(String[] args) {
6         System.out.println("Estaba la mora en su lugar,");
7         System.out.println("Vino la mosca y le hizo mal,");
8         System.out.println("La mosca a la mora,");
9         System.out.println("Y la mora en su moralito sola.");
10        System.out.println();
11        System.out.println("Estaba la mosca en su lugar,");
12        System.out.println("Vino la araña y le hizo mal,");
13        System.out.println("La araña a la mosca,");
14        System.out.println("La mosca a la mora,");
15        System.out.println("Y la mora en su moralito sola.");
16        System.out.println();
17        System.out.println("Estaba la araña en su lugar,");
18        System.out.println("Vino el ratón y le hizo mal,");
19        System.out.println("El ratón a la araña,");
20        System.out.println("La araña a la mosca,");
21        System.out.println("La mosca a la mora,");
22        System.out.println("Y la mora en su moralito sola.");
23        System.out.println();
24        System.out.println("Estaba el ratón en su lugar,");
25        System.out.println("Vino el gato y le hizo mal,");
26        System.out.println("El gato al ratón,");
27        System.out.println("El ratón a la araña,");
28        System.out.println("La araña a la mosca,");
29        System.out.println("La mosca a la mora,");
30        System.out.println("Y la mora en su moralito sola.");
31        System.out.println();
32        System.out.println("Estaba el gato en su lugar,");
33        System.out.println("Vino el perro y le hizo mal,");
34        System.out.println("El perro al gato,");
35        System.out.println("El gato al ratón,");
36        System.out.println("El ratón a la araña,");
37        System.out.println("La araña a la mosca,");
38        System.out.println("La mosca a la mora,");
39        System.out.println("Y la mora en su moralito sola.");
40        System.out.println();
41    }
42 }
43 }
44 }
```

[VPL](#)

Navega por la unidad

◀ PROG01 Tarea Aprendizaje 03. Descomposición de programas

PROG01 Tarea Aprendizaje 04. Datos primitivos, expresiones y variables ▶

Contacta con nosotros:

Dirección: Paseo de Ubarburu 39, Edificio EnerTIC of. 206 · Donostia San Sebastián

Telefono : 945 567 953

E-mail: info@birt.eus

Twitter: @Birt_LH

