

[Todos mis módulos](#) / [Mis módulos](#) / [2021-2022](#) / [Desarrollo de aplicaciones multiplataforma](#) / [Programación](#) / Estructuras de control (10%)
/ [PROG03 Tarea de evaluación 02. Realiza un programa en Java \(60%\)](#)

Descripción

[Ver entrega](#)

PROG03 Tarea de evaluación 02. Realiza un programa en Java (60%)

Límite de entrega: viernes, 3 de diciembre de 2021, 23:59

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

Tipo de trabajo: Individual

PROG03 TE02 REALIZA UN PROGRAMA EN JAVA (60%)

ENUNCIADO

El objetivo de esta tarea es realizar un programa que genere un número al azar del 1 al 100 y que dé pistas para que el usuario o usuaria lo adivine.

Para ello, tendremos que utilizar **estructuras condicionales y repetitivas**, mientras seguimos practicando el uso de **variables**, **bucles FOR y métodos con paso de parámetros y sentencia return** así como, las funcionalidades de las clases de las librerías de java.

En primer lugar, el programa mostrará la siguiente **introducción al programa**:

```
El programa genera un número aleatorio del 1 al 100
El usuario o usuaria debe adivinarlo
El programa le dará pistas cada vez que falle
```

A continuación, generará un número utilizando la clase Random y pedirá adivinarlo. Como pista dirá si el número a adivinar es mayor o menor que el introducido y seguirá pidiendo números hasta que se adivine.

En ese momento indicará cuántos intentos se han necesitado y preguntará si se quiere volver a jugar. Continuará jugando siempre que se escriba algo que empiece por "S", tanto en mayúsculas como en minúsculas. Por ejemplo: "Sí", "si", "SIEMPRE", "s", "seguro"... permitirán seguir jugando:

```
Adivina un número del 1 al 100
Adivina el número: 50
El número es menor.
Adivina el número: 25
El número es mayor.
Adivina el número: 37
El número es mayor.
Adivina el número: 44
El número es mayor.
Adivina el número: 47
El número es menor.
Adivina el número: 46
Has adivinado en 6 intentos.
¿Quieres jugar otra vez (S/N)? s
```

Todas las partidas funcionarán igual hasta que se elija acabar. En ese momento, el programa visualizará las estadísticas del juego. Mostrará el número total de partidas jugadas, el número total de intentos realizados en todas esas partidas, el número de intentos de media por partida y el número de intentos de la mejor partida. Es decir, la partida que menos intentos ha necesitado:

```
Adivina un número del 1 al 100
Adivina el número: 50
El número es mayor.
Adivina el número: 75
El número es menor.
Adivina el número: 62
El número es menor.
Adivina el número: 56
El número es menor.
Adivina el número: 53
El número es menor.
Adivina el número: 52
El número es menor.
Adivina el número: 51
Has adivinado en 7 intentos.
¿Quieres jugar otra vez (S/N)? n

Resultados del juego:
Partidas jugadas: 2
Intentos realizados: 13
Intentos por partida: 6,5
Mejor partida: 6
```

Ten en cuenta que si se adivina a la primera, el número de intentos será igual a 1:

```
Adivina un número del 1 al 100
Adivina el número: 44
Has adivinado en 1 intentos.
```

Para calcular las estadísticas, se debe asumir que ninguna partida necesitará más de 1000000 intentos.

Para conseguir la máxima nota, se deberán utilizar al menos los 2 siguientes métodos: `jugarPartida` y `mostrarEstadísticas`. El primero controlará una única partida, desde que se genera el número hasta que se adivina, mientras que el segundo método se encargará de calcular y visualizar las estadísticas una vez que se ha decidido finalizar el juego.

Para que el programa pase el test de la plataforma tiene que generar los mismos números aleatorios que se le indican en el test. Para eso, hay que utilizar la semilla 20 y crear el objeto de la clase `Random` en el método `jugarPartida`.

RECURSOS

Para realizar esta tarea puedes utilizar `JGRASP` o el editor `VPL` de Moodle que además te dirá si la salida coincide con la propuesta si utilizas la semilla 20 a la hora de crear el objeto de tipo `Random`.

INDICACIONES DE DESARROLLO

Realiza el programa por partes. Escribe el código poco a poco compilándolo y testeándolo frecuentemente.

Utiliza métodos, parámetros y `returns` para estructurar el programa y evitar redundancias. Para conseguir la máxima nota, se deberán utilizar al menos los 2 siguientes métodos: `jugarPartida` y `mostrarEstadísticas`. El primero controlará una única partida, desde que se genera el número hasta que se adivina, mientras que el segundo método se encargará de calcular y visualizar las estadísticas una vez que se ha decidido finalizar el juego.

En este caso, podrán existir `println()` en el `main`.

Existirá un único objeto de tipo `Scanner` que se creará en el `main`. Se podrá utilizar en cualquier otro punto del programa si se pasa como parámetro.

Empieza generando el número aleatorio y visualizándolo por pantalla para saber cuál es y si el programa está respondiendo bien.

A continuación céntrate en una partida, en pedir números y comprobar si el número que hay que adivinar es mayor o menor que el introducido.

Cuando una partida funcione, consigue repetirla siempre que el usuario o usuaria quiera continuar. Necesitarás utilizar algunos de los métodos de los objetos de tipo `String`.

Recuerda el algoritmo acumulador para poder obtener el número de intentos y todas las estadísticas del juego.

Debes utilizar el método [System.out.printf\(\)](#) para formatear correctamente los resultados obtenidos.

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. En la cabecera de los métodos indica los parámetros que utiliza y si devuelve algún dato.

INDICACIONES DE ENTREGA

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

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 el formato elegido

CRITERIOS DE CALIFICACIÓN. TOTAL 10 PUNTOS.

La tarea se evaluará siguiendo la siguiente rúbrica:

Funcionalidad (70%).

Comentarios (15%).

Legibilidad (15%).

Con ella, 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.

[VPL](#)

Navega por la unidad

◀ PROG03 Tarea de aprendizaje 05. Boolean

Ir a...

PROG03 Tarea Evaluativa 02 - Solución ►

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