

TICS200: Tarea #3

Universidad Adolfo Ibáñez

Loreto Arriagada
loreto.arriagada.v@edu.uai.cl
15 de mayo de 2022

Objetivos

- Aprender a estructurar un programa con paradigma **orientado a objetos** en el lenguaje **Java**.
- Desarrollar un entregable funcional utilizando la herramienta **replit**.

1. Enunciado

Mejorar el algoritmo “adivina el número”, que se adjunta, en los siguientes aspectos que se señalan en los requerimientos funcionales y NO funcionales.

1.1. Requerimientos funcionales y NO funcionales

El programa deberá estar desarrollado en replit y en lenguaje Java con **paradigma orientado a objetos** y debe cumplir con los siguientes requerimientos:

1. Está construido en clase Main en paradigma imperativo. Se requiere pasarlo a paradigma orientado a objetos, para ello debe crear al menos las clases “NumeroOculto” y “Jugador”,
2. Validar que los datos ingresados estén entre 1 y 10 (Se recomienda leer un String con `System.console().readLine()` y pasarlo a entero con `parseInt()`), y
3. Luego de entregar los resultados de un juego, preguntar al usuario si desea seguir jugando para lo que debe ofrecer las opciones S o N. Si presiona S o s, debe repetir el ciclo de un juego y si presiona N o n, debe finalizar el programa. Así sucesivamente, iterar hasta que el jugador (usuario) seleccione N o n.

2. Sobre la entrega

- La tarea debe ser realizada de forma individual.
- Se debe entregar un archivo `main.java` y quienes deseen usar packages deben hacer entrega en un archivo `.zip` (El uso de packages es opcional, toda vez que subdividiendo en clases, se cumple con el requerimiento de pasarlo de imperativo a orientado a objetos).
- El plazo para entrega es el domingo 15 de mayo a las 23:59.

3. Pauta de Evaluación

La tarea será revisada utilizando la siguiente pauta de evaluación:

Concepto de evaluación	Descripción	Puntos	Descuento
Punto base	Punto base	1,0	No aplica
Aplicación del paradigma	Aplicar correctamente el paradigma orientado a objetos Para lenguaje Java evitar métodos imperativos	3,0	0.5 por cada desviación
Funcionalidades	Cumplimiento del requerimiento	1,5	0.1 por cada incumplimiento funcional 0.2 por cada incumplimiento de supuestos 0.2 por cada falla de validación o caída del programa
Lógica y lenguaje	Uso adecuado del lenguaje seleccionado y lógica empleada: - Optimización códigos fuentes - Uso de comentarios - Evitar vicios como while true - Excesivo uso de continue, break o return al medio de un método - Orden del código fuente - Uso adecuado de las estructuras de datos y sentencias del lenguaje	1,5	0.3 por cada desviación