# JUEGO DE ADIVINAR NUMERO

## By Facundo Nahuel Britrez – Desarrollo Web

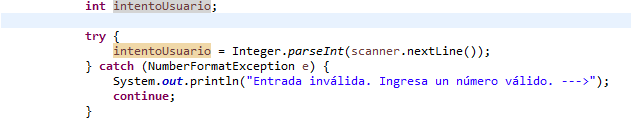


Para generar un número aleatorio en Java, se pueden utilizar dos métodos principales: Math.random() y la clase java.util.Random. Math.random() es una función estática que devuelve un valor double entre 0.0 (inclusive) y 1.0 (exclusivo). La clase Random permite generar números aleatorios de diferentes tipos y rangos, además de ofrecer opciones de inicialización.

Como solo necesitaba enteros y quería un código legible utilice la clase random.



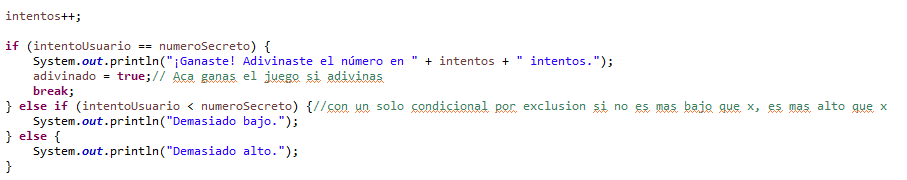
Inicio las variables el “adivinado” es falso porque aun no se adivina para esta instancia del código y es importante que sea falso.



Lo descubrí por accidente que se puede hacer una declaración de variable y sumarle el scanner en un try catch y no rompe el código, pero lo importante es que atajo el error de entrada de caracteres cuando es para ingresar números

No condicione que si o si sea entre 1 y 100

Porque la verdad es que da igual el numero será o muy alto o muy bajo así que me parece bien dejarlo así.



Acá la decisión de poner a mitad el aumento de intentos y no al final es simplemente porque ponerlo antes del try catch, es que haría que aumente, aquí la “Documentación”

*“¿Ocurre una excepción? Se verifica si se lanzó una excepción dentro del bloque try.*

*Si no ocurre ninguna excepción: El flujo continúa con el código después del bloque try (si lo hay) dentro del bucle, o con la siguiente iteración si es la última.*

***Si ocurre una excepción: Se salta inmediatamente al bloque catch correspondiente.***

*Bloque catch: Se ejecuta el código del bloque catch para manejar la excepción. Este bloque puede registrar el error, intentar una recuperación, o realizar cualquier otra acción necesaria.*

*Fin del bloque catch: Una vez que el bloque catch se ha completado, el flujo continúa:*

*Si es la última iteración: Se sale del bucle y continúa con el código después del bucle.*

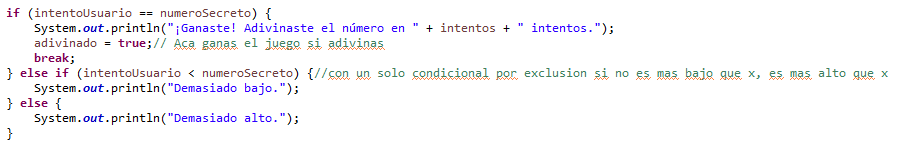
***Si no es la última iteración: Se regresa al inicio del bucle (a la siguiente iteración).”***

**En resumen, vuelve al inicio del bloque y si se ponen a escribir letras en todas las oportunidades perderían.**

Pero al estar después del try catch no pierde el intento



En estructuras como if, while, o for, la condición se evalúa como true o false, y el bloque de código asociado se ejecuta solo si la condición es true, por lo tanto, al agregar “!” se está diciendo que si no es verdadero (distinto de verdadero) se ejecuta el mensaje de que perdiste el juego



El echo de poner verdadero en el booleano hace que el código finalice exitosamente y aparezca el mensaje



FIN.

Gracias por leer :3