

Act 2.4.- Adivinar número

Clase AdivinarNumero

```
public class AdivinarNumero {  
  
    public static void main(String[] args) {  
        Juego juego = new Juego(numeroMaximo: 51);  
        ArrayList<Jugador> arr = new ArrayList<Jugador>();  
  
        System.out.printf(format: "Numero de jugadores 5\nRango de numeros: Entre 1 y 50\nNumero a adivinar: %d\n", args, juego.getNumeroAdivinar());  
  
        byte i = 1;  
        boolean juegoTerminado = false;  
  
        do {  
            Jugador j = new Jugador(numero: i, juego, "\u001B[3" + i + "m");  
            arr.add(j);  
            j.start();  
  
            try {  
                j.join();  
                if (j.isAcertado()) {  
                    juegoTerminado = true;  
                }  
            } catch (InterruptedException ex) {  
                System.out.printf(format: "ERROR: %s\n", args, ex.getMessage());  
            }  
  
            if (i == 5) {  
                i = 1;  
            } else {  
                i++;  
            }  
        } while (!juegoTerminado && i <= 5);  
    }  
}
```

Clase Juego

```
public class Juego {  
  
    private final int numeroAdivinar;  
    private final int numeroMaximo;  
  
    public Juego(int numeroMaximo) {  
        Random r = new Random();  
        this.numeroAdivinar = r.nextInt(origin: 1, numeroMaximo + 1);  
        this.numeroMaximo = numeroMaximo;  
    }  
  
    public int getNumeroMaximo() {  
        return numeroMaximo;  
    }  
  
    public int getNumeroAdivinar() {  
        return numeroAdivinar;  
    }  
  
    public synchronized boolean adivinarNumero(String color, int idJugador, int numero) {  
        if (this.numeroAdivinar != numero) {  
            System.out.printf(format: "%sJugador %d dice: %d\n", args: color, args: idJugador, args: numero);  
        } else {  
            System.out.printf(format: "%sJugador %d dice: %d\n", args: color, args: idJugador, args: numero);  
            System.out.printf(format: "%sJugador %d gana!!!\n", args: color, args: idJugador);  
            return true;  
        }  
        return false;  
    }  
}
```

Clase Jugador

```
5
6 public class Jugador extends Thread{
7
8     private final int numero;
9     private final Juego juego;
10    private final String color;
11    private boolean acertado;
12
13    public Jugador(int numero, Juego juego, String color) {
14        this.numero = numero;
15        this.juego = juego;
16        this.color = color;
17    }
18
19    public boolean isAcertado() {
20        return acertado;
21    }
22
23    @Override
24    public void run() {
25        try {
26            Random r = new Random();
27            acertado = juego.adivinarNumero(color, idJugador: numero, numero: r.nextInt(origin: 1, juego.getNumeroMaximo()+1));
28            Thread.sleep((long) (r.nextInt(origin: 100, bound: 501)));
29        } catch (InterruptedException ex) {
30            System.out.printf(format: "ERROR: %s\n", args: ex.getMessage());
31        }
32    }
33
34 }
35
```

Resultado

```
Numero de jugadores 5
Rango de numeros: Entre 1 y 50
Numero a adivinar: 24
Jugador 1 dice: 46
Jugador 2 dice: 20
Jugador 3 dice: 27
Jugador 4 dice: 17
Jugador 5 dice: 30
Jugador 1 dice: 21
Jugador 2 dice: 51
Jugador 3 dice: 16
Jugador 4 dice: 29
Jugador 5 dice: 47
Jugador 1 dice: 24
Jugador 1 gana!!!
BUILD SUCCESSFUL (total time: 4 seconds)
```