

Modelos Estocásticos - T18

¿Quieres cambiar la puerta?

Jorge Aurelio Morales Manrique
C.C. 1010075711
jomorales@unal.edu.co

Universidad Nacional de Colombia
Abril 5 de 2021

5. Manual de usuario

En la carpeta adjunta con el nombre “Código Fuente”, se encuentra un archivo con código fuente en java **MontyHall.java**, el cual realiza la simulación para un número n ingresado por el usuario. A continuación se describe la lista de pasos para ejecutar el programa en sistemas operativos Windows y Linux:

1. Descargar el archivo **MontyHall.java** y guardarlo en el directorio de preferencia.
2. Abrir una consola de comandos. En Linux realizar la siguiente combinación de teclas (**Ctrl+Alt+T**). En Windows realizar la combinación (**Windows+R**) en el campo de texto emergente digitar **cmd** y posteriormente dar click en aceptar, como se muestra en la siguiente imagen:

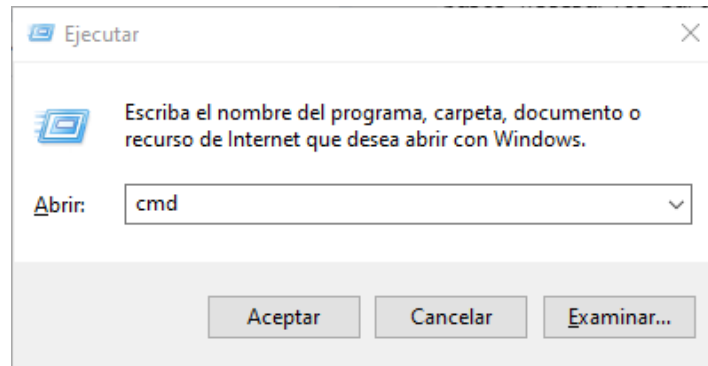


Figura 1: Abrir consola Windows.

3. Una vez abierta la consola cambiar la ruta a la ubicación del archivo haciendo uso del comando **cd**, el cual es igual tanto en Windows como en Linux.

4. Ejecutar los siguientes comandos en el orden que se muestra a continuación (tanto en Windows como en Linux)

```
javac MontyHall.java  
java MontyHall
```

5. Ingresar el valor de n . El programa comenzará la simulación e imprimirá el resultado.

6. Manual técnico

La estructura interna del programa se describe a continuación:

Variables globales

- **Variable `wins_changing`.** Inicializada en 0 guarda el número de victorias al cambiar de puerta.
- **Variable `wins_not_changing`.** Inicializada en 0 guarda el número de victorias sin cambiar de puerta.

```
// Variables que almacenan el número de victorias cambiando  
// y sin cambiar de puerta  
private static int wins_changing = 0;  
private static int wins_not_changing = 0;
```

Figura 2: Variables globales.

Función `simulate`

Genera dos números aleatorios correspondientes a la puerta ganadora y la puerta seleccionada por el usuario. En caso de que la puerta ganadora y la puerta seleccionada sean iguales, se aumenta el contador de victorias sin cambiar de puerta, en el otro caso se aumenta el contador de victorias cambiando la puerta.

```
// Método que representa las diferentes rondas
private static void simulate() {
    Random random = new Random();
    int win_door = random.nextInt( bound: 3);
    int selection = random.nextInt( bound: 3);
    if (selection == win_door) {
        wins_not_changing++;
    } else {
        wins_changing++;
    }
}
```

Figura 3: Variables globales.

Función principal (main)

Realiza la lectura del número n y comienza la simulación, haciendo uso de la función **simulate**. Una vez termina realiza el calculo del porcentaje de victorias cambiando la puerta y sin cambiar la puerta e imprime los resultados en la consola.

```
public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    System.out.print("Ingrese el número de simulaciones: ");
    int n = scanner.nextInt();
    for (int i = 0; i < n; i++) {
        simulate();
    }
    double wc_percent = (double) wins_changing / n;
    double wnc_percent = (double) wins_not_changing / n;
    System.out.println("Porcentaje de victorias cambiando: " + wc_percent);
    System.out.println("Porcentaje de victorias sin cambiar: " + wnc_percent);
}
```

Figura 4: Función principal (main).