

## Act 2.8.- Puente

### Main

```
public class Main {  
    public static void main(String[] args) {  
        Puente puente = new Puente();  
  
        Persona[] personas = new Persona[15];  
        for (byte i = 0; i < 15; i++) {  
            personas[i] = new Persona("Persona " + i, puente);  
        }  
  
        for (Persona persona : personas) {  
            persona.start();  
        }  
    }  
}
```

# Persona

```
public class Persona extends Thread{

    private final String nombre;
    private final Puente puente;
    private final int peso;

    public Persona(String nombre, Puente puente) {
        this.nombre = nombre;
        this.puente = puente;
        this.peso = generarPeso();
    }

    public String getNombre() {
        return nombre;
    }

    public int getPeso() {
        return peso;
    }

    private int generarPeso(){
        Random r = new Random();
        return r.nextInt(origin: 40, bound: 121);
    }

    @Override
    public void run() {

        try {
            puente.cruzarPuente(p: this);
        } catch (InterruptedException ex) {
            System.out.printf(format: "ERROR: %s\n", args: ex.getMessage());
        }

    }

}
```

# Puente

```
public class Puente {

    private int cantidadPersonas = 0;
    private int cantidadPesoActual = 0;

    public Puente() {
    }

    public synchronized void cruzarPuente(Persona p) throws InterruptedException {
        while (cantidadPersonas >= 3 || cantidadPesoActual + p.getPeso() > 200) {
            System.out.printf(format: "[%s] con peso %skg. Espera a cruzar el puente.\n", args: p.getNombre(), args: p.getPeso());
            p.wait();
        }

        cantidadPersonas++;
        cantidadPesoActual += p.getPeso();

        System.out.printf(format: "[%s] cruza el puente\n", args: p.getNombre());
        System.out.printf(format: "Personas en el puente: %d - Peso en el puente: %d\n", args: cantidadPersonas, args: cantidadPesoActual);

        int tiempo = new Random().nextInt(origin: 1000, bound: 50001);
        int tiempoMostrar = tiempo / 1000;
        System.out.printf(format: "%s%s me quedan %d segundos para terminar de cruzar.\n", args: "\u001B[31m", args: p.getNombre(), args: tiempoMostrar);
        Thread.sleep(millis: tiempo);

        cantidadPersonas--;
        cantidadPesoActual -= p.getPeso();
        System.out.printf(format: "[%s] con peso %skg. Termino de cruzar el puente.\n", args: p.getNombre(), args: p.getPeso());

        notifyAll();
    }

}
```

## Resultado

```
input < Puentes (run) x
run:
[Persona 0] cruza el puente
Personas en el puente: 1 - Peso en el puente: 66
Persona 0 me quedan 41 segundos para terminar de cruzar.
[Persona 0] con peso 66kg. Termino de cruzar el puente.
[Persona 14] cruza el puente
Personas en el puente: 1 - Peso en el puente: 79
Persona 14 me quedan 21 segundos para terminar de cruzar.
[Persona 14] con peso 79kg. Termino de cruzar el puente.
[Persona 13] cruza el puente
Personas en el puente: 1 - Peso en el puente: 72
Persona 13 me quedan 42 segundos para terminar de cruzar.
[Persona 13] con peso 72kg. Termino de cruzar el puente.
[Persona 12] cruza el puente
Personas en el puente: 1 - Peso en el puente: 62
Persona 12 me quedan 29 segundos para terminar de cruzar.
BUILD STOPPED (total time: 1 minute 59 seconds)
```