qwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxmoricvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmoriqwertyumoriiopasdfghjklzxcvbnmoriqwertyumoriopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmoriqwertyumoriopasdfghjklzxcvbinmqwertyumoriopasdfghjklzxewtyucvbnmoriqwertyumoripsdfghjklztopxcmoriqwertyuopasdfghjklzixcvbnmoriertyumoriopasdhjiklzxcvbnmoriqwertyumorimoriopasdfghjklzxcvdasfgfhbnmtwreorirtyumoriopasdfghjklzxcvbnmoriqwertyumoriopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnm



|  |
| --- |
| Actividad 5: ejercicio de hilos  2º Desarrollo de Aplicaciones Multiplataforma Programación de servicios y procesos  19/11/2021  Martínez Díez, Ángel Mori |

Contenido

[1. Abre el archivo de apoyo "Multitarea e hilos en Java..." introduce y ejecuta el ejercicio de las cajeras. 3](#_Toc88254095)

[a. Original 3](#_Toc88254096)

[Código 3](#_Toc88254097)

[Salida 4](#_Toc88254098)

[*b.* *Thread* 4](#_Toc88254099)

[Código 4](#_Toc88254100)

[Salida 5](#_Toc88254101)

[*c.* *Runnable* 6](#_Toc88254102)

[Código 6](#_Toc88254103)

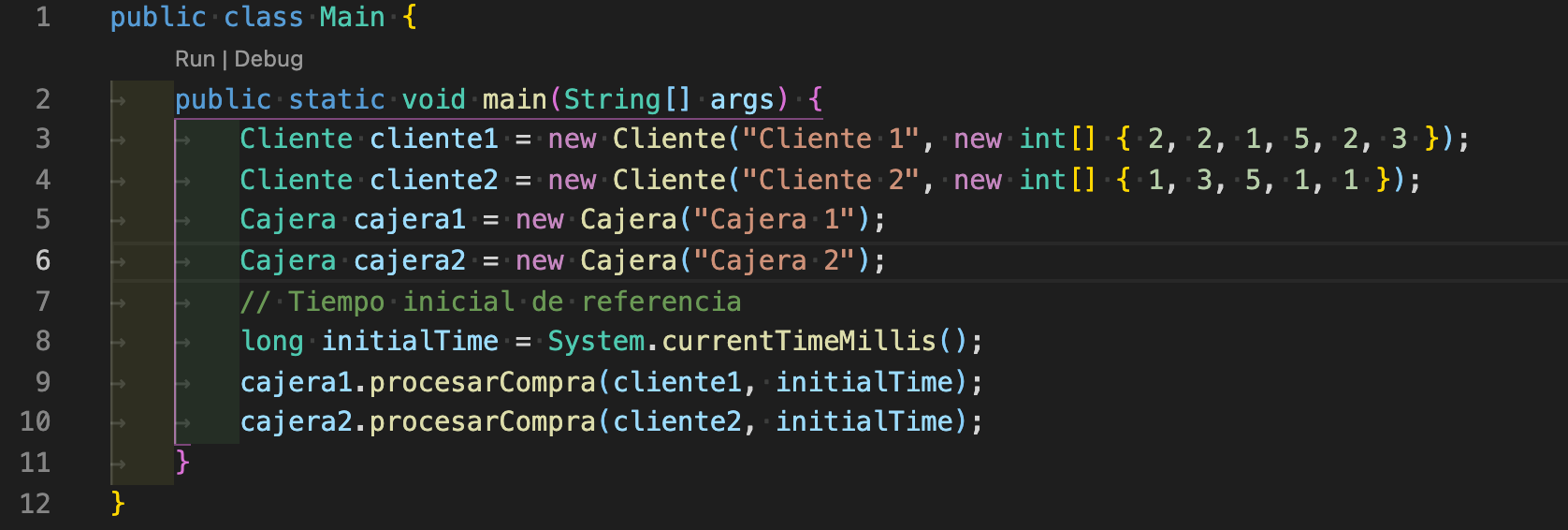
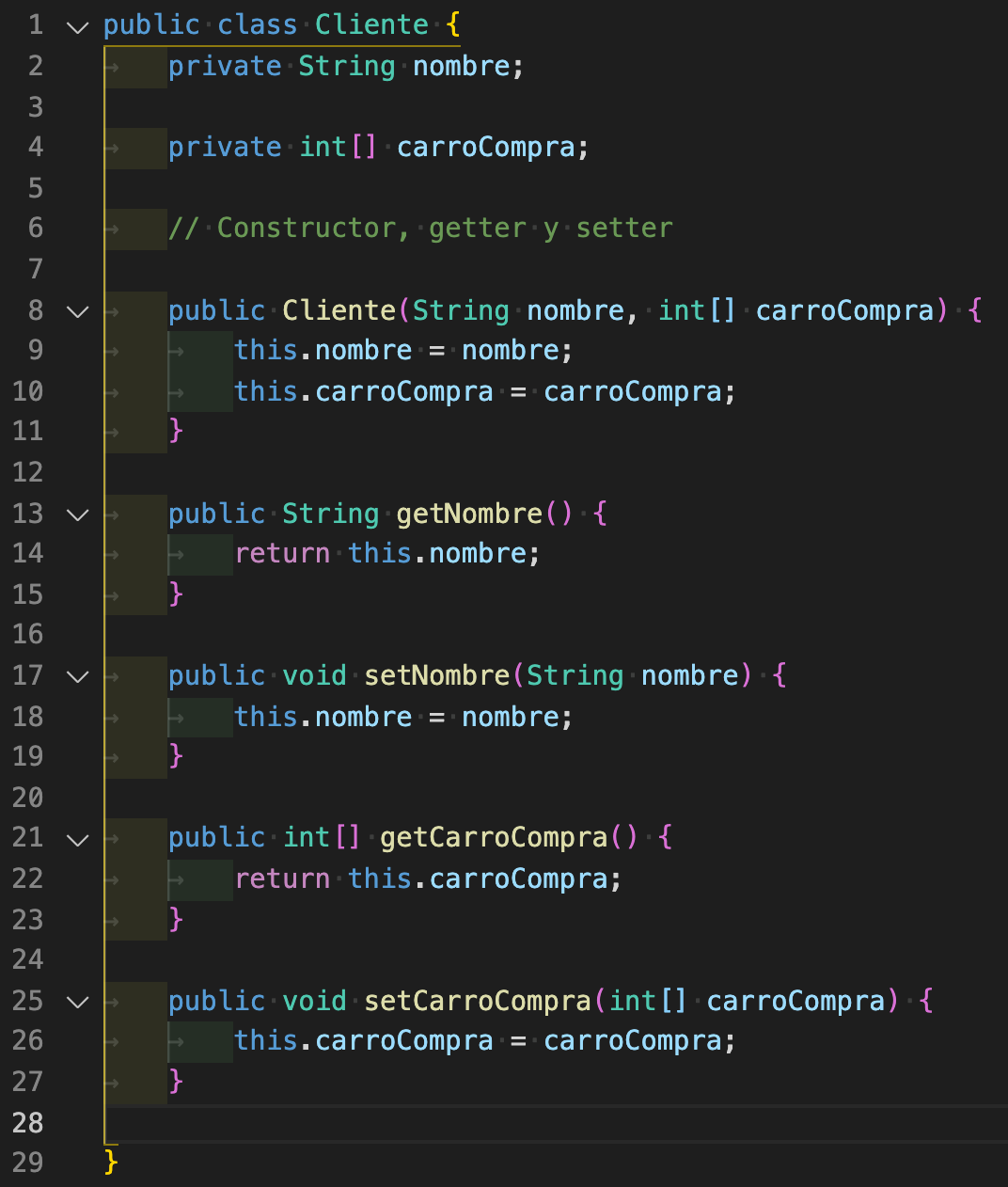
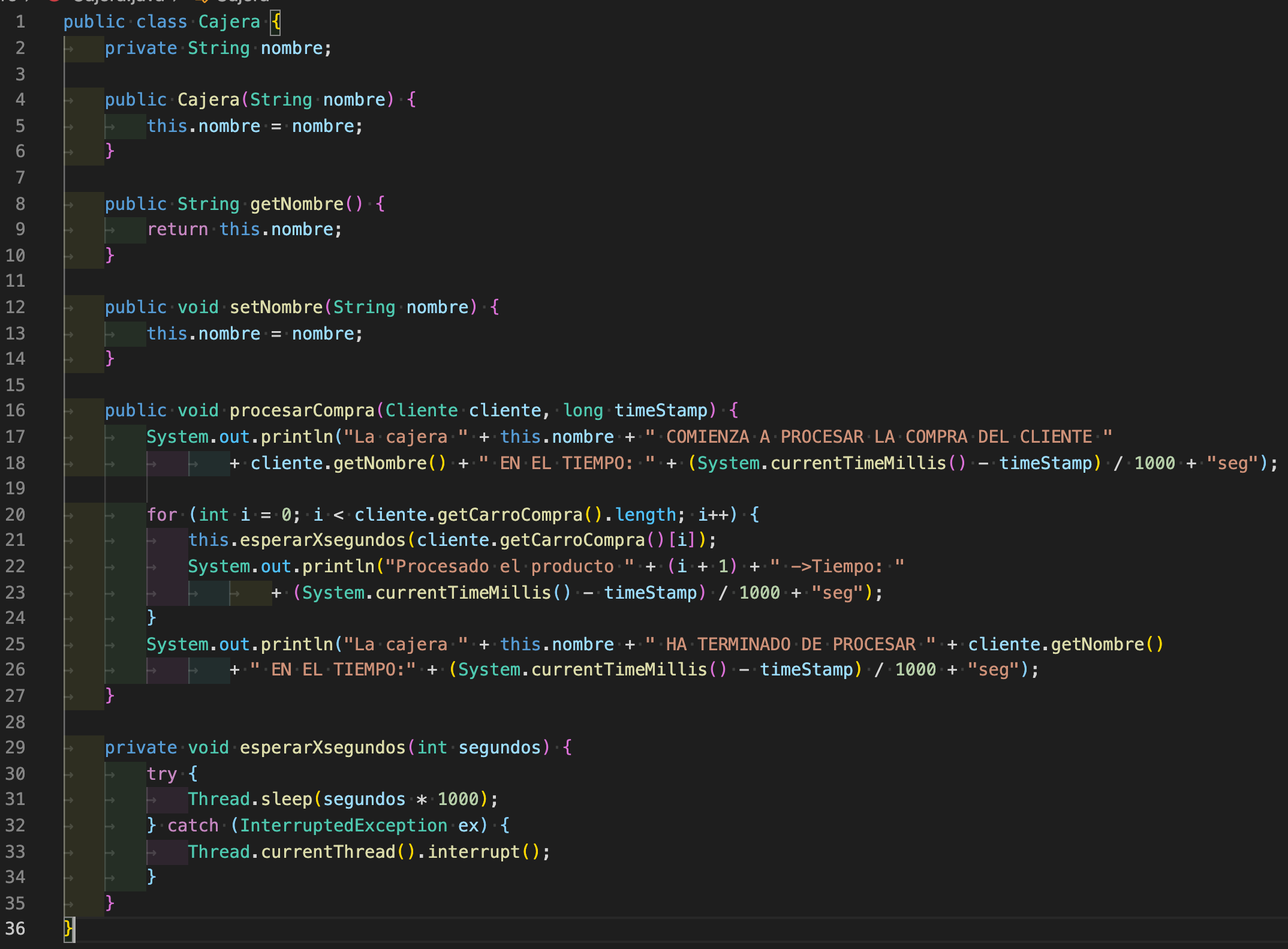
[Salida 6](#_Toc88254104)

[2. Abre el archivo "último hilos" y realiza uno de los ejercicios, el que tú quieras. 6](#_Toc88254105)

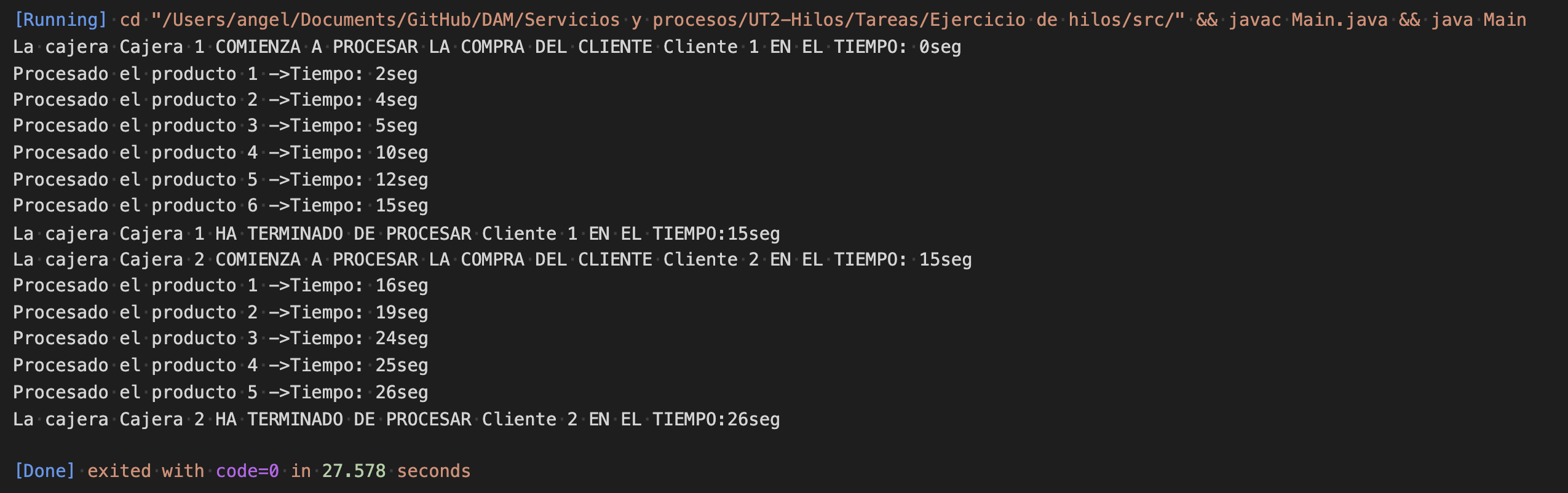
# Abre el archivo de apoyo "Multitarea e hilos en Java..." introduce y ejecuta el ejercicio de las cajeras.

## Original

### Código



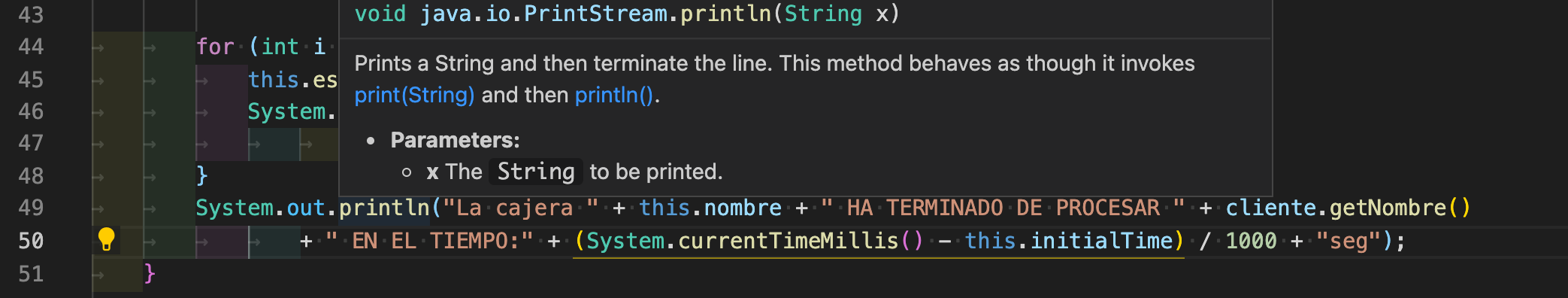
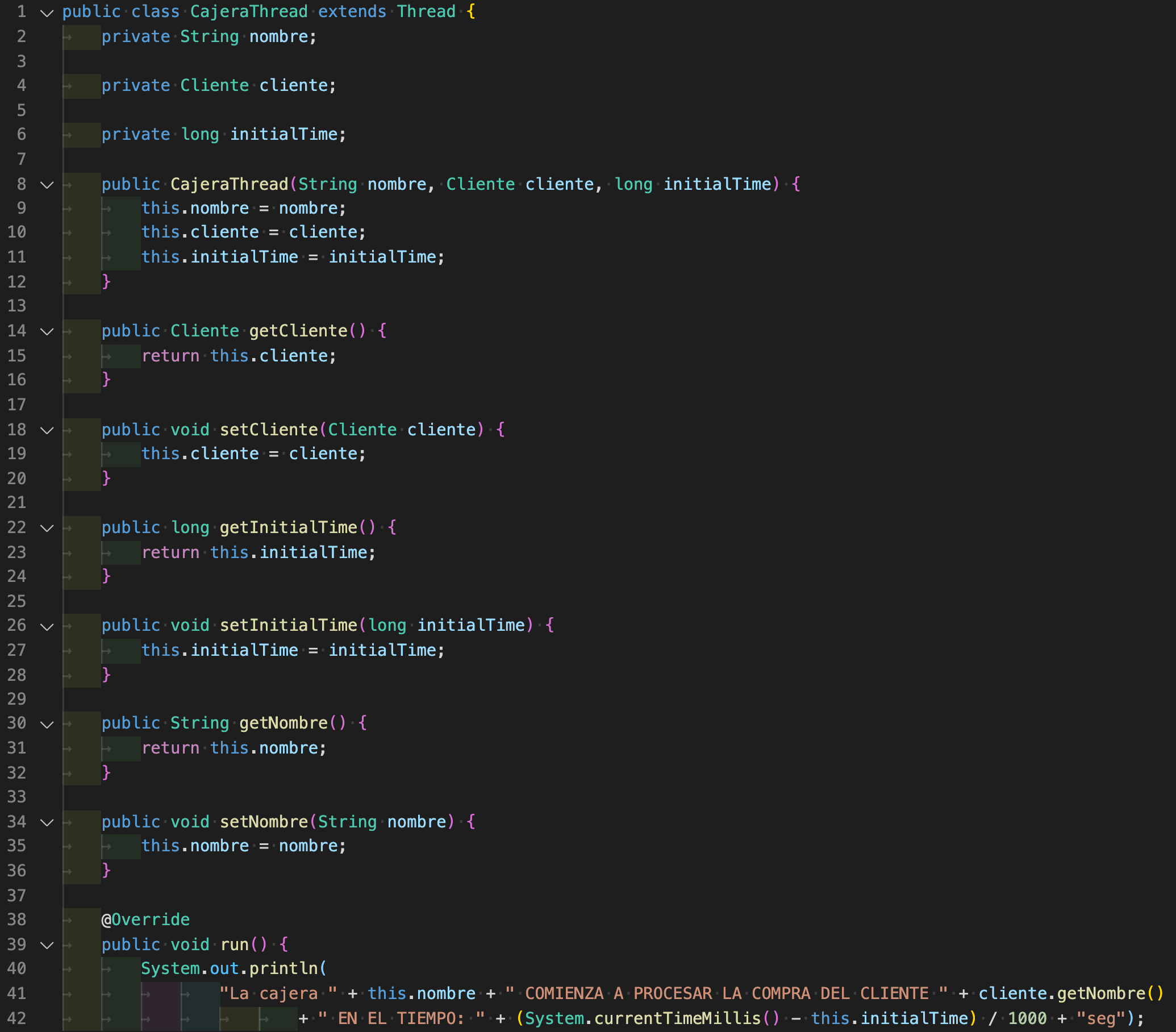
### Salida

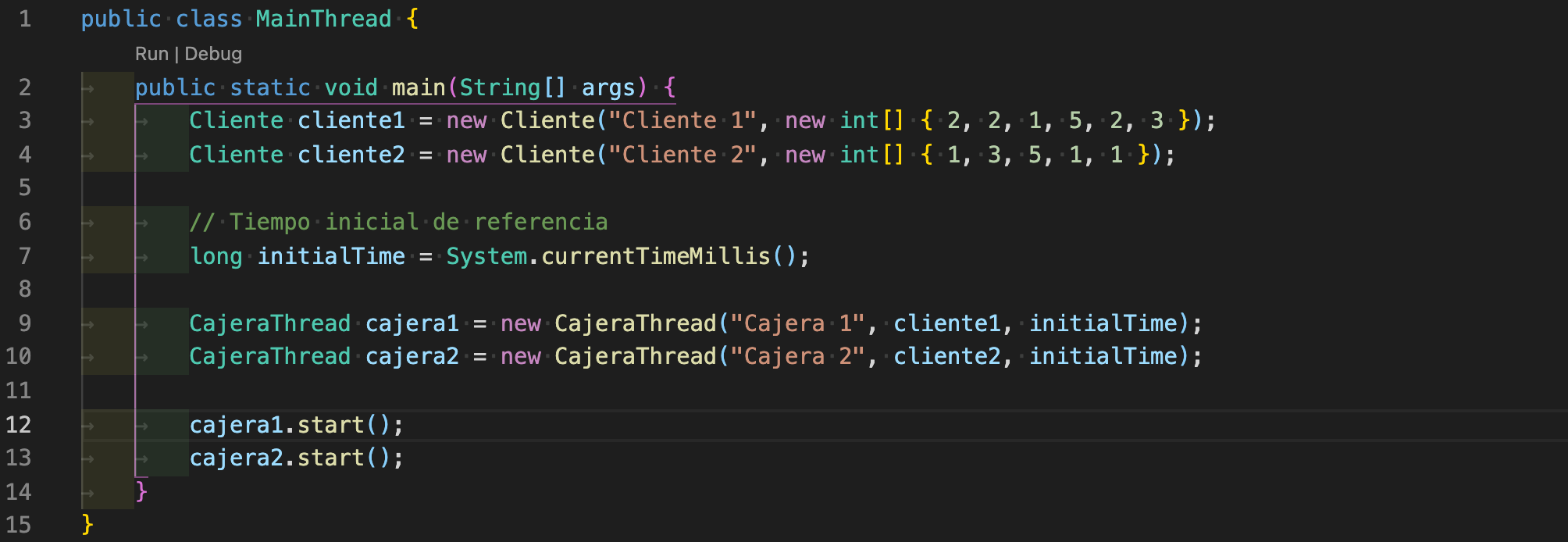
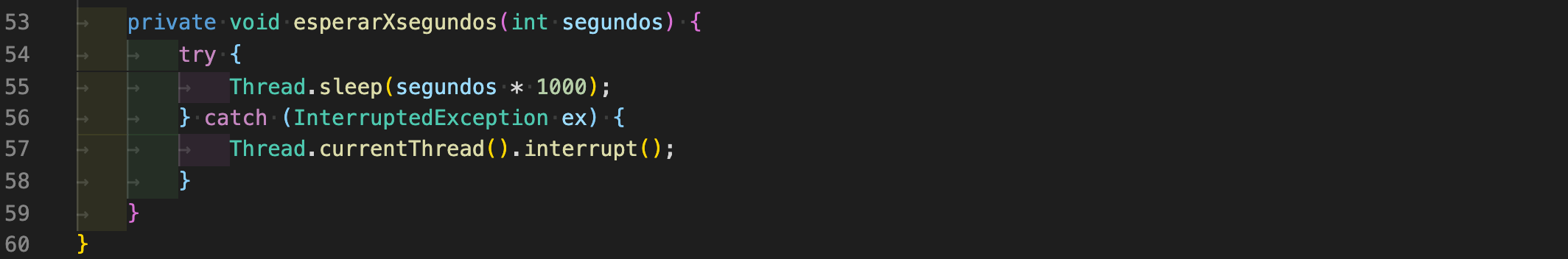


## *Thread*

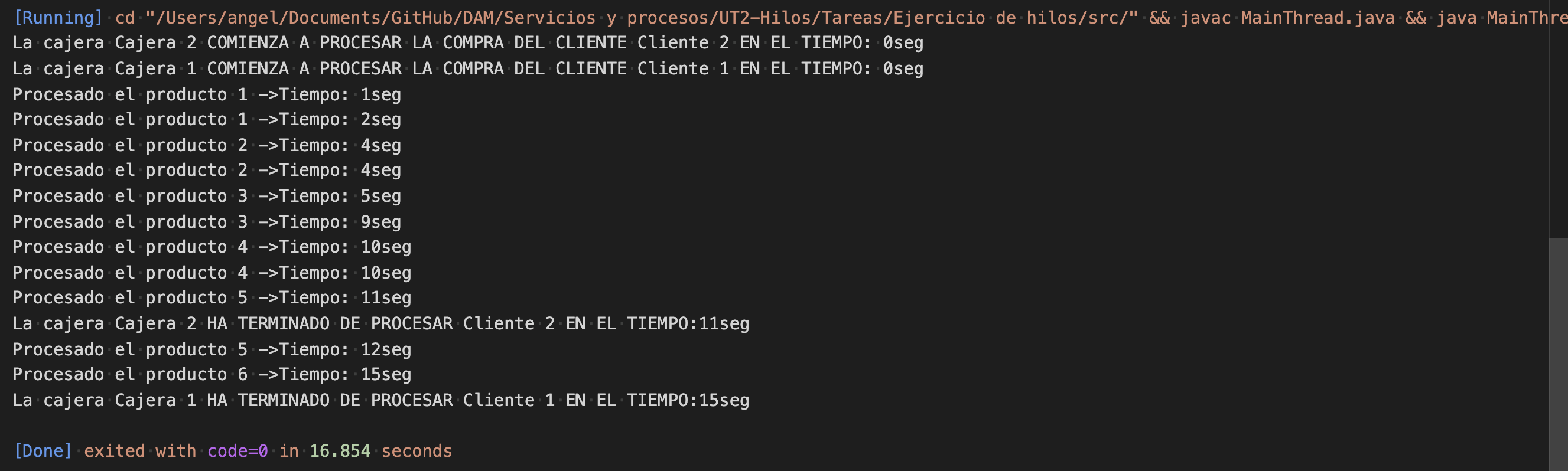
### Código

Cliente queda como en el original.



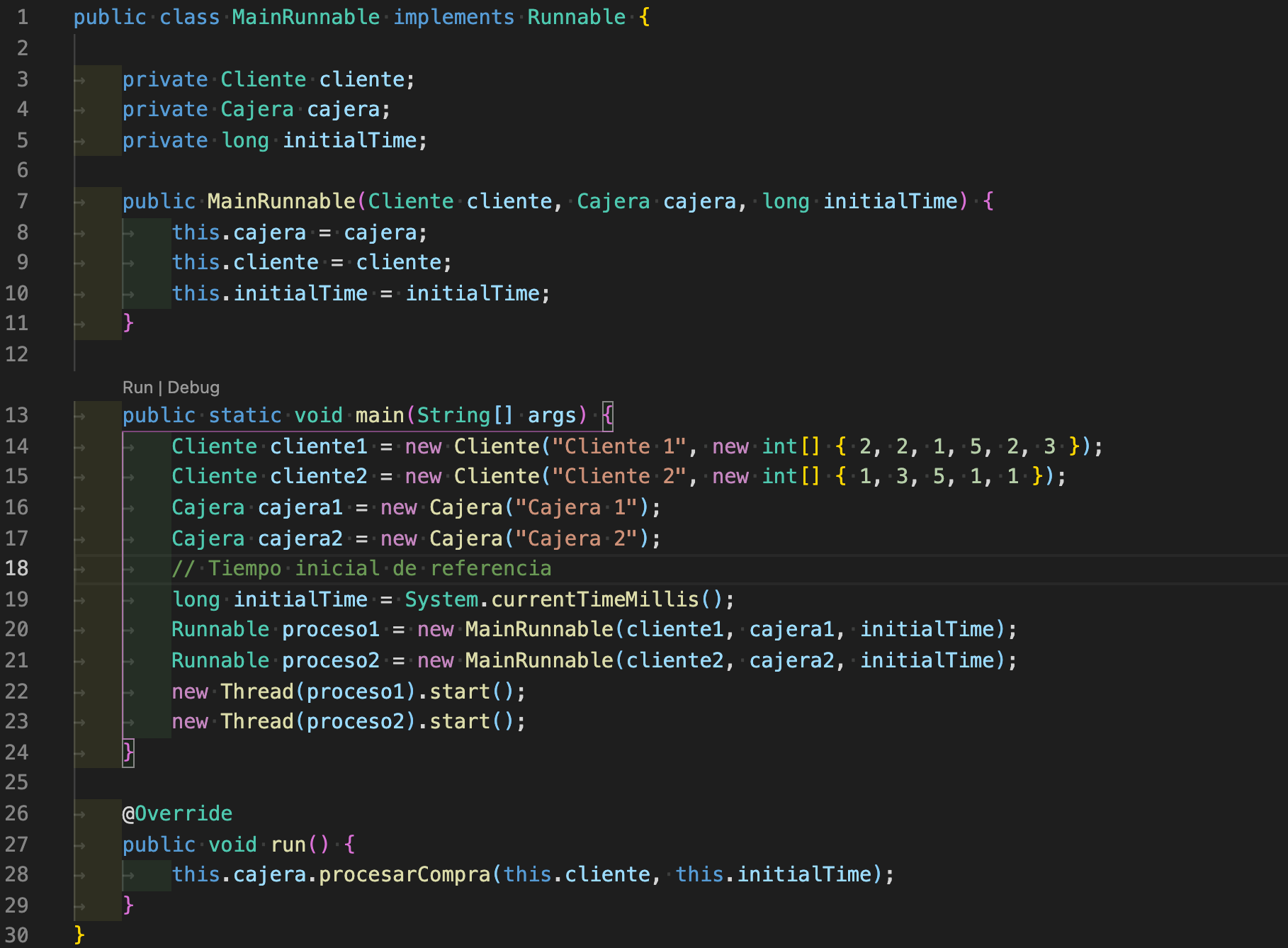


### Salida

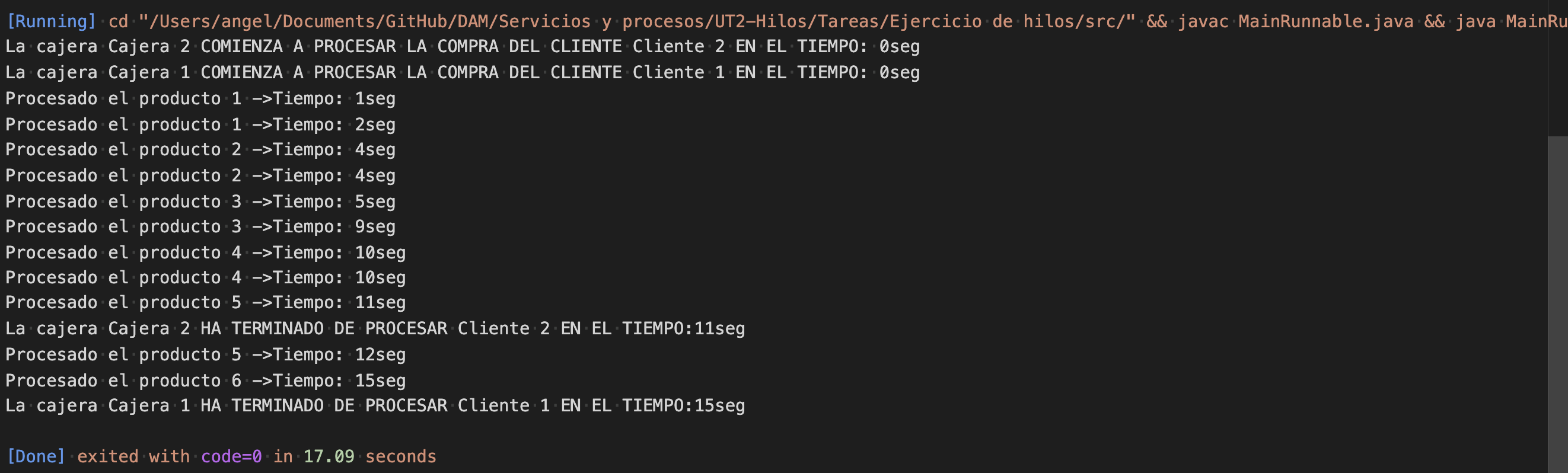


## *Runnable*

### Código

Cajera y cliente quedan como en el original

### Salida



# Abre el archivo "último hilos" y realiza uno de los ejercicios, el que tú quieras.

import java.util.Random;

class Resultados {

public static int ganancias;

public static long tiempo\_espera;

public static int clientes\_atendidos;

}

class Caja {

private static final int MAX\_TIME = 1000;

private boolean ocupada;

public Caja() {

this.ocupada = false;

}

public boolean ocupada() {

return ocupada;

}

synchronized public void atender(int pago) throws InterruptedException {

ocupada = true;

int tiempo\_atencion = new Random().nextInt(MAX\_TIME);

Thread.sleep(tiempo\_atencion);

Resultados.ganancias += pago;

Resultados.clientes\_atendidos++;

ocupada = false;

}

}

class Cola {

class Nodo {

int cliente;

Nodo sig;

}

Nodo raiz, fondo;

Caja cajas[];

int N;

public Cola(int N) {

raiz = null;

fondo = null;

this.N = N;

cajas = new Caja[N];

for (int i = 0; i < N; i++) {

cajas[i] = new Caja();

}

}

private boolean vacia() {

if (raiz == null)

return true;

else

return false;

}

private int cajaLibre() {

int i = 0;

while (i < N) {

if (!cajas[i].ocupada()) {

break;

}

i++;

}

return i;

}

synchronized public int esperar(int id\_cliente) throws InterruptedException {

int caja\_id;

Nodo nuevo;

nuevo = new Nodo();

nuevo.cliente = id\_cliente;

nuevo.sig = null;

if (vacia()) {

raiz = nuevo;

fondo = nuevo;

} else {

fondo.sig = nuevo;

fondo = nuevo;

}

// Esperar hasta el turno

while (((caja\_id = cajaLibre()) == N) || (raiz.cliente != id\_cliente)) {

// Me bloqueo hasta que sea mi turno

wait();

}

// Salgo de la cola

raiz = raiz.sig;

return caja\_id;

}

public void atender(int id\_caja, int pago) throws InterruptedException {

cajas[id\_caja].atender(pago);

}

synchronized public void finalizar\_compra() throws InterruptedException {

notify();

}

}

class Cliente extends Thread {

private static final int MAX\_DELAY = 2000;

private static final int MAX\_COST = 100;

private int id;

private Cola cola;

Cliente(int id, Cola cola) {

this.id = id;

this.cola = cola;

}

public void run() {

try {

int numero\_caja;

System.out.println("Cliente " + id + " realizando compra");

Thread.sleep(new Random().nextInt(MAX\_DELAY));

long s = System.currentTimeMillis();

numero\_caja = cola.esperar(id);

cola.atender(numero\_caja, new Random().nextInt(MAX\_COST));

System.out.println("Cliente " + id + " atendido en caja " + numero\_caja);

cola.finalizar\_compra();

System.out.println("Cliente " + id + " finalizando");

long espera = System.currentTimeMillis() - s;

Resultados.tiempo\_espera += espera;

System.out.println("Cliente " + id + " saliendo despues de esperar " + espera);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}

public class ModernSuperMarket {

public static void main(String[] args) throws InterruptedException {

int N = Integer.parseInt(args[0]);

Cola cola = new Cola(N);

int M = Integer.parseInt(args[1]);

Cliente clientes[] = new Cliente[M];

for (int i = 0; i < M; i++) {

clientes[i] = new Cliente(i, cola);

clientes[i].start();

}

try {

for (int i = 0; i < M; i++) {

clientes[i].join();

}

} catch (InterruptedException ex) {

System.out.println("Hilo principal interrumpido.");

}

System.out.println("Supermercado cerrado.");

System.out.println("Ganancias: " + Resultados.ganancias);

System.out.println("Tiempo medio de espera: " + (Resultados.tiempo\_espera / Resultados.clientes\_atendidos));

}

}

Se obtienen resultados diferentes con las mismas condiciones de ejecución, ya que hay factores aleatorios:

