

Práctica 2

Concurrencia

Objetivo

Practicar con el modelo de concurrencia de Java.

Actividad 1.

Compile, ejecute, analice y explique el siguiente código:

```
class Order {
    private static int i = 0;
    private int count = i++;
    public Order() {
        if(count == 10) {
            System.out.println("Se acabo la comida, cerramos");
            System.exit(0);
        }
    }
    public String toString() { return "orden " + count; }
}

class WaitPerson extends Thread {
    private Restaurant restaurant;
    public WaitPerson(Restaurant r) {
        restaurant = r;
        start();
    }
    public void run() {
        while(true) {
            while(restaurant.order == null)
                synchronized(this) {
                    try {
                        wait();
                    } catch(InterruptedException e) {
                        throw new RuntimeException(e);
                    }
                }
            System.out.println("El mesero obtuvo " + restaurant.order);
            restaurant.order = null;
        }
    }
}

class Chef extends Thread {
    private Restaurant restaurant;
    private WaitPerson waitPerson;
    public Chef(Restaurant r, WaitPerson w) {
        restaurant = r;
        waitPerson = w;
        start();
    }
    public void run() {
```

```
while(true) {
    if(restaurant.order == null) {
        restaurant.order = new Order();
        System.out.print("Orden lista! ");
        synchronized(waitPerson) {
            waitPerson.notify();
        }
    }
    try {
        sleep(100);
    } catch(InterruptedException e) {
        throw new RuntimeException(e);
    }
}

public class Restaurant {
    Order order;
    public static void main(String[] args) {
        Restaurant restaurant = new Restaurant();
        WaitPerson waitPerson = new WaitPerson(restaurant);
        Chef chef = new Chef(restaurant, waitPerson);
    }
}
```

Actividad 2.

Considere los siguientes objetos:

```
Runnable r1 = new Runnable() {
    public void run() {
        try {
            while (true) {
                System.out.println("Hola, mundo!");
                Thread.sleep(1000L);
            }
        } catch (InterruptedException iex) {}
    }
};

Runnable r2 = new Runnable() {
    public void run() {
        try {
            while (true) {
                System.out.println("Adiós, mundo cruel!");
                Thread.sleep(2000L);
            }
        } catch (InterruptedException iex) {}
    }
};
```

Cada uno de estos objetos imprime un mensaje periódicamente y de manera indefinida. Utilice el modelo de concurrencia de Java para que estas dos tareas se ejecuten de manera coordinada.