

Ejercicio 3:

```

3) while (true) {
    Fetch()           // fetch
    Decode()          // decodificación
    Execute()         // ejecución
    if I==1 AND INTR==1 {
        PUSH PSW      // [SP]=PSW, SP=SP-1
        PUSH PC       // [SP]=PC, SP=SP-1
        CLI           // I=0
        PC=[0x0000]   // llamado a la rutina
        INTA()        // se le indica al dispositivo
    }
}
// que atenderá un pedido

```

Ejercicio 4:

```

4) leída Consecutiva: CMP R0, 0x0000 // verifica si se debe seguir leyendo al
JE Fin // dispositivo
estado: lea MOV R2, [0xFFFF] // verifica si el dispositivo está ocupado
AND R2, 0x0000
JNE estado: lea
MOV R2, [0xFFFF] // escribe al dato a la dirección de memoria
AND R2, 0x00FF // máscara de R2
MOV [R1], R2
ADD R1, 0x0001 // R1 pasa a la dirección siguiente
SUB R0, 0x0001 // decrece R0
JMP leída Consecutiva
Fin: RET // la rutina termina

```