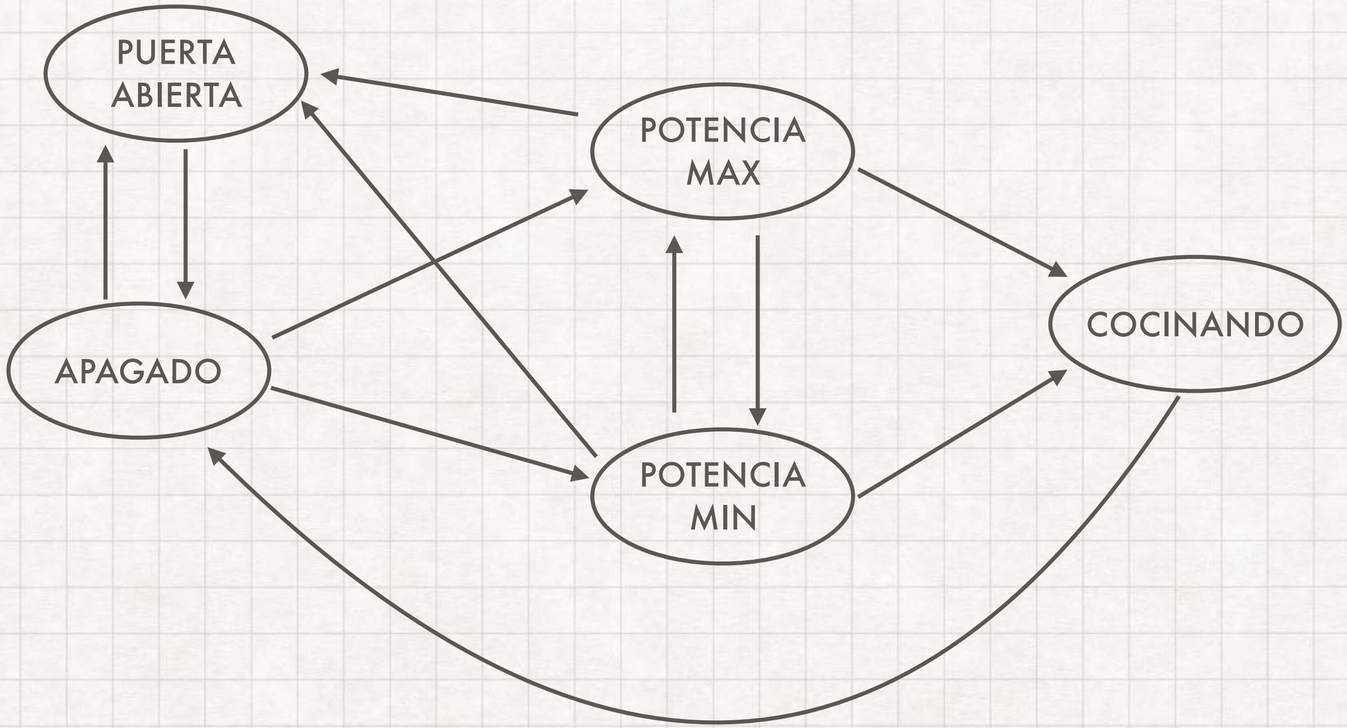
STATE PATTERN

MAQUINA MICROONDAS



- Solo tiene dos potencias
- Cuando se abre la puerta, se va al estado inicial

Diagrama de estados de una máquina microondas bastante limitada. Las flechas muestran las transiciones permitidas

```
public class Microondas {

  private Estado estado = Estado.APAGADO;

  private enum Estado {
    APAGADO, PUERTA_ABIERTA, POTENCIA_MAX, POTENCIA_MIN, COCINANDO
  }

  public void setPotenciaMin() {
    switch (estado) {
    case APAGADO:
    case POTENCIA_MAX:
      estado = Estado.POTENCIA_MIN;
      System.out.println("fijada potencia mínima");
      break;
    default:
      System.out.println("No se puede fijar potencia desde " + estado);
    }
  }
```

“

Allow an object to alter its behavior when its internal state changes. The object will appear to change its class.

— *GoF*

”

```
public class Microondas  {

  private Estado estado = Estado.APAGADO;

  private enum Estado  {
          ....
   }


     ...
}
```

Estado inicial

```
private enum Estado  {
  APAGADO {
      ...
  }, ... ;

  public Estado setPotenciaMin() {
    System.out.println("No se permite operacion desde " + this);
    return this;
  }
  public Estado setPotenciaMax() {
    System.out.println("No se permite operacion desde " + this);
    return this;
  }
    ....
  public Estado encender() {
    System.out.println("No se permite operacion desde " + this);
    return this;
  }
  public Estado apagar() {
    System.out.println("No se permite operacion desde " + this);
    return this;
  }
}
```

Por defecto:
no acepta las transiciones

Cada transición:

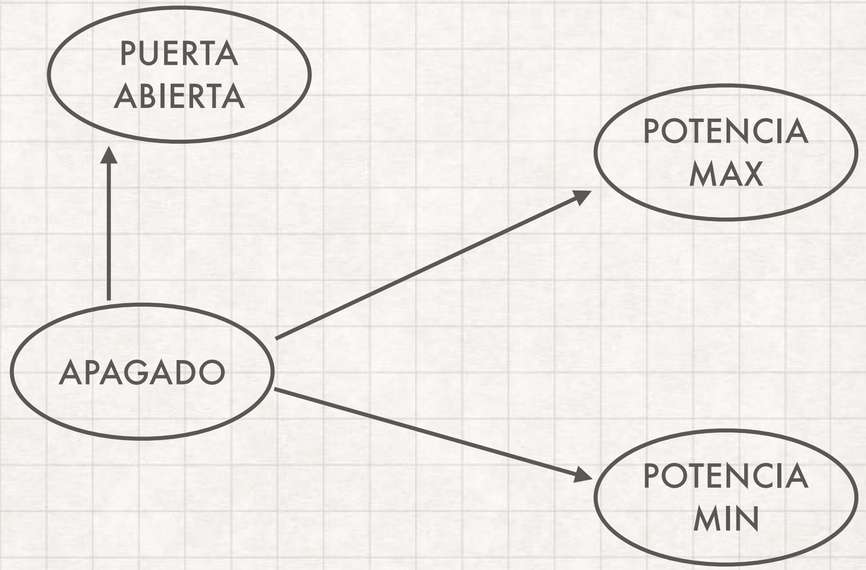
- realiza una acción
- - devuelve el siguiente estado

ESTADO: APAGADO

```
    APAGADO {
      public Estado setPotenciaMin() {
        System.out.println("fijada potencia mínima");
        return Estado.POTENCIA_MIN;
      }

      public Estado setPotenciaMax() {
        System.out.println("fijada potencia máxima");
        return Estado.POTENCIA_MIN;
      }
      public Estado abrirPuerta() {
        System.out.println("abriendo puerta");
        return Estado.PUERTA_ABIERTA;
      }

    }, PUERTA_ABIERTA {
```



```
public class Microondas  {

  private Estado estado = Estado.APAGADO;

  private enum Estado  {
      ...
  }

  public void setPotenciaMin() {
    estado = estado.setPotenciaMin();
  }

  public void setPotenciaMax() {
    estado = estado.setPotenciaMax();
  }
...

  public void apagar() {
     estado = estado.apagar();
  }
}
```

Es un implementación sencilla, pero limitada.

Por ejemplo:

- Todo el estado es representable con un enum
- Las transiciones no necesitan acceder a ningún método del microondas

Una implementación más general usaría un objeto con una referencia al microondas para cada estado.

STATE PATTERN