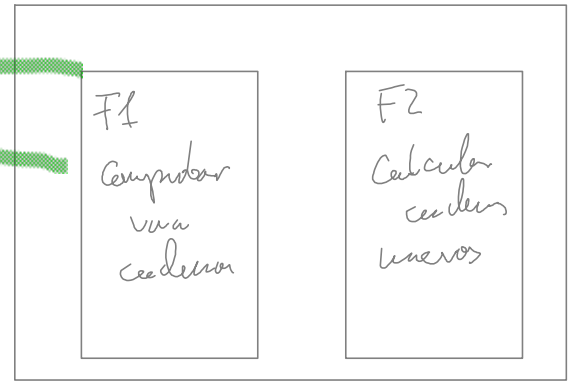
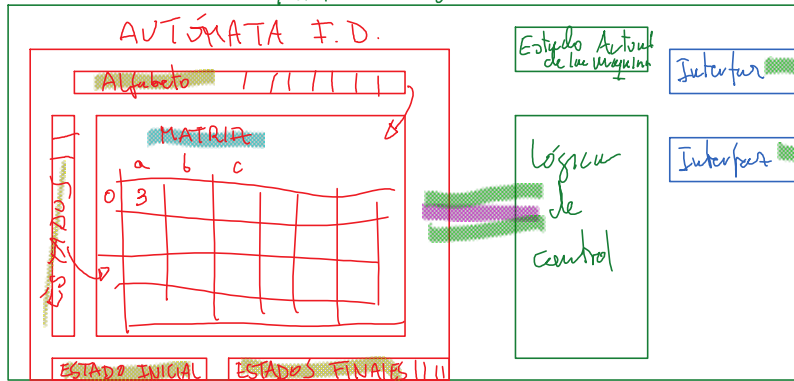


MAQUINA DE ESTADOS

PROGRAMA



AFD

Clase

Estructuras de Datos
 Alfabeto List<Character> char
 Estados List<Integer> int
 EstadoInicial Integer
 EstadosFinales List<Integer> Integer



void inicializaMatriz ();

Para cada estado de Estados

Matriz.add (estado, new HashMap<Character, Integer> ())

Repetir

}

void cargarMatriz ();

matriz.get (estado).add (caracter, estado.de_siguiente);

...

matriz.get (0).add ('a', 3);

...

}

Integer getEstadoInicial ()
 return EstadoInicial;

cargarAlfabeto () ?
 Alfabeto.add (new Character ('a'));

b
c
d
e

cargarEstados () {
 Estados.add (new Integer (1))

for x=0 to 18

Estados.add (new Integer (x))

}

establecerQi (p) {
 EstadoInicial = p;
 }

establecerQf () {
 EstadosFinales.add (new Integer (7))
 }

Integer getSiguienteEstado (estado, caracter)
 return matriz.get (estado).get (caracter);
 }

bool isFinal (estado) {
 return EstadosFinales.contains (estado);
 }

ME: ED EstadoActual integer
 AFD AutomataFD

Inicializar () {
 EstadoActual = AFD.getEstadoInicial ();
 }

bool isFinal () {
 return AFD.isFinal (EstadoActual).
 }

bool computaCadena (cadena) {
 return true;
 }

acepta (caracter) {
 estadoTemp = AFD.getSiguienteEstado (EstadoActual, caracter)
 if (estadoTemp != null)
 EstadoActual = estadoTemp
 else

```
bool completaCadena (cadena) {
```

```
    Inicializa()
```

```
    Para cada caracter en cadena
```

```
        try { this.acepta (caracter);
```

```
            if this.istFinal ()
```

```
                return true; }
```

```
    { catch ( ) {
```

```
        return false;
```

```
    } // he completado  
    la cadena de  
    entrada?
```

```
    if ( ... == ... ) {
```

```
        Estado Actual = estado temp
```

```
    else
```

```
        throw Exception
```

```
    endif
```

```
}
```