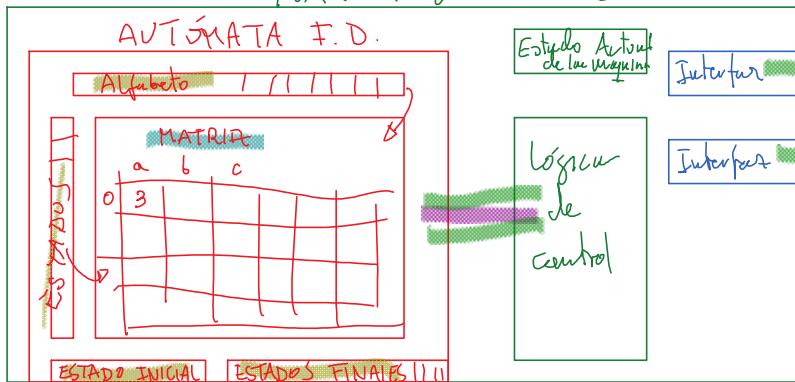
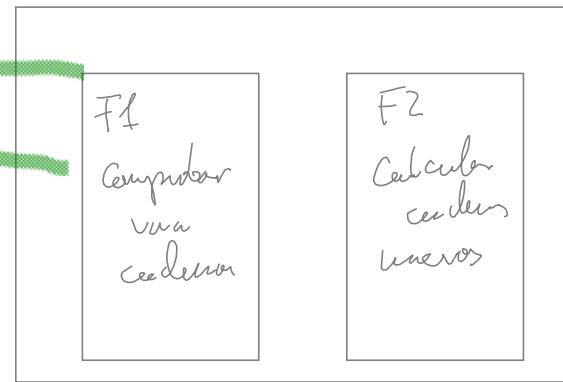


# MAQUINA DE ESTADOS



# PROGRAMA



AFD

Clade

Estruuras de Datos	char character
Alfabeto	<u>List&lt;Character&gt;</u>
Estados	<u>List&lt;Integer&gt;</u>
Estado Inicial	int ↪ Integer ↪
Estados Finales	<u>List&lt;Integer&gt;</u>



vord Jwainiliza Mutniz () !!

Para cada estado de Estados  
    Matriz add (estado, new HashMap <Character, Integer> ())  
    Repetic

void cargarMatriz();

```
matriz.get(estudo).add(caracter, estudo_de_sulto),
```

```
matiz.get(0).add('a', 3);
```

longer return

```
bool isFinal () {  
    return ATD.isFinal (EstadoActual);  
}  
  
bool competirCadena (cadena) {  
    = Inicializar();  
    |  
    ^
```

```
    Initialize( ) {  
        EstadoActual = AFD.getEstadoInicial();  
    }
```

```
    acepta (caracter) {  
        estado tmp = AFD.getSigienteEstado(EstadoActual, caracter)  
        if(estado tmp != null)  
            EstadoActual = estado tmp  
        else
```

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

<sup>Introducir()</sup>  
    Para cada carácter en cadena  
    try{this.acepta(caracter);  
        if(this.isFinal())  
            return true;  
    }{catch(... -  
        return false;

} {he completado  
la cadena de  
entrada}

else{  
    entraActual = entraTemp;

    throw Exception;  
}

endif