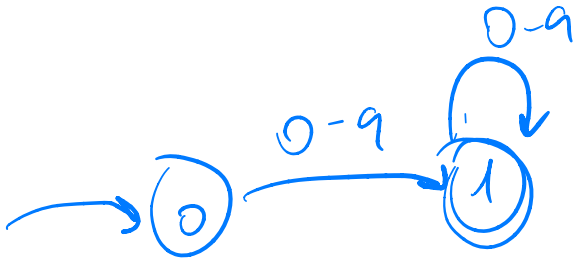


Crear los autómatas finitos:

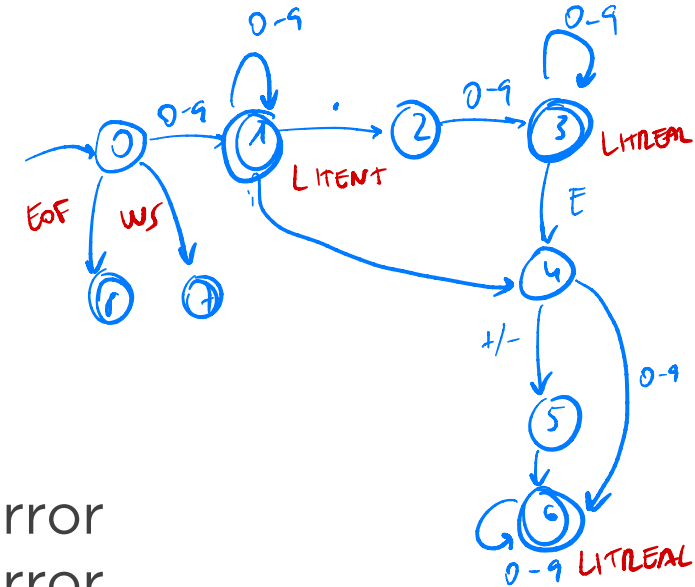
## Literales Enteros

24  
8  
3245



## Literales Reales

24.5  
24E10  
24.5E+10  
24.5E-6



24. // Error  
.5 // Error  
24E // Error  
24E- // Error  
E10 // Error

CONTRASEÑA zip = tabla

	0-9	.	E	+/-	WS	EOF	Otro
0	1				7	8	
1	1	2	4				
2	3						
3	3		4				
4	6			5			
5	6						
6	6						
7					7		
8							

```

int transiciones[][] = {
    { 1, ERROR, ERROR, ERROR, 7, 8, ERROR },
    { 1, 2, 4, ERROR, ERROR, ERROR, ERROR },
    { 3, ERROR, ERROR, ERROR, ERROR, ERROR, ERROR },
    { 3, ERROR, 4, ERROR, ERROR, ERROR, ERROR },
    { 6, ERROR, ERROR, 5, ERROR, ERROR, ERROR },
    { 6, ERROR, ERROR, ERROR, ERROR, ERROR, ERROR },
    { 6, ERROR, ERROR, ERROR, ERROR, ERROR, ERROR },
    { ERROR, ERROR, ERROR, ERROR, 7, ERROR, ERROR },
    { ERROR, ERROR, ERROR, ERROR, ERROR, ERROR, ERROR },
};

```

↓ forma de agrupar  
los caracteres por columnas

	0-9	.	E	+/-	WS	EOF	Otro
0	1				7	8	
1	1	2	4				
2	3						
3	3		4				
4	6			5			
5	6						
6	6						
7					7		
8							

```
int getColumna(int carácter) {  
    if (Character.isDigit(carácter))  
        return 0;  
    if (carácter == '.')  
        return 1;  
    if (carácter == 'E')  
        return 2;  
    if (carácter == '+' || carácter == '-')  
        return 3;  
    if (Character.isWhitespace(carácter))  
        return 4;  
    if (carácter == -1)  
        return 5;  
    return 6;  
}
```

```
public Token nextToken() throws IOException {
```

```
    int estadoActual = 0;
```

```
    while (true) {
```

```
        int columna = getColumna(getChar());
```

*↓ aquí solo puede haber un estado o un error*

```
        int siguiente = transiciones[estadoActual][columna];
```

```
        if (siguiente != Error) {
```

```
            estadoActual = siguiente;
```

```
            readNextC();
```

```
        } else {
```

```
            if (tokens[estadoActual] == No-FINAL) {
```

```
                print("Error");
```

```
                readNextC();
```

```
                estadoActual = 0;
```

```
            } else if (tokens[estado] == IGNORAR) {
```

```
                estadoActual = 0;
```

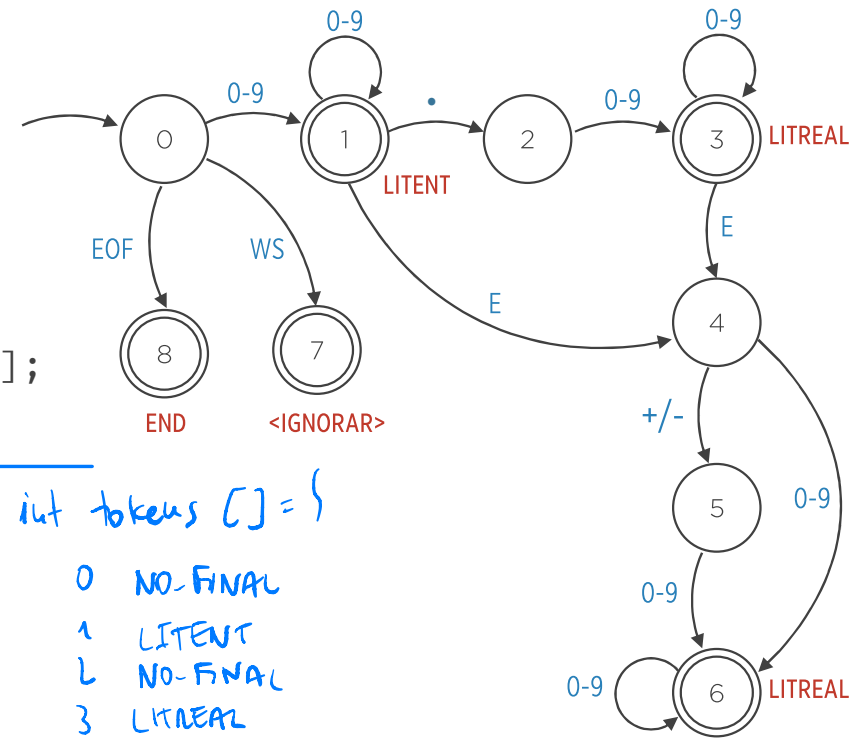
```
            } else {
```

```
                return new Token(tokens[estado], "no hacer");
```

```
            }
```

```
        }
```

```
    }
```



```
int tokens[] = {
```

```
    0 NO-FINAL
```

```
    1 LITENT
```

```
    2 NO-FINAL
```

```
    3 LITREAL
```

```
    4 NO-FINAL
```

```
    5 NO-FINAL
```

```
    6 LITREAL
```

```
    7 IGNORAR
```

```
    8 END
```

```
return new Token(...);
```

```
};
```

```
public static final int END = 0;
```

```
public static final int LITENT = 257;
```

```
public static final int LITREAL = 258;
```