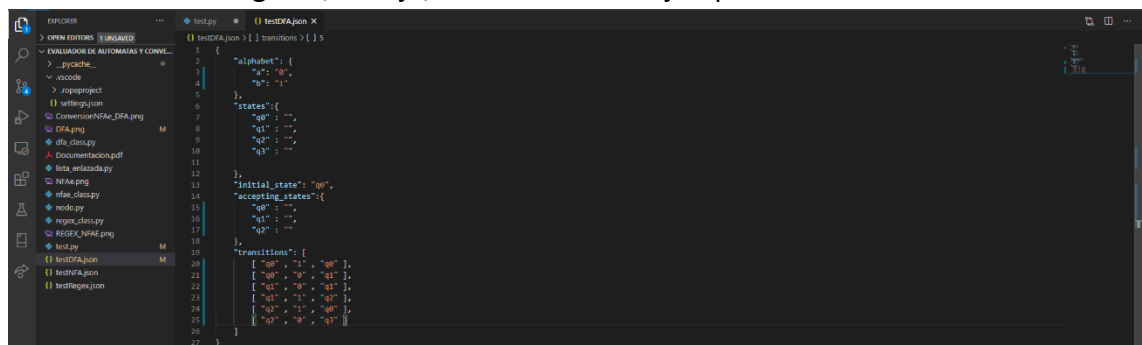
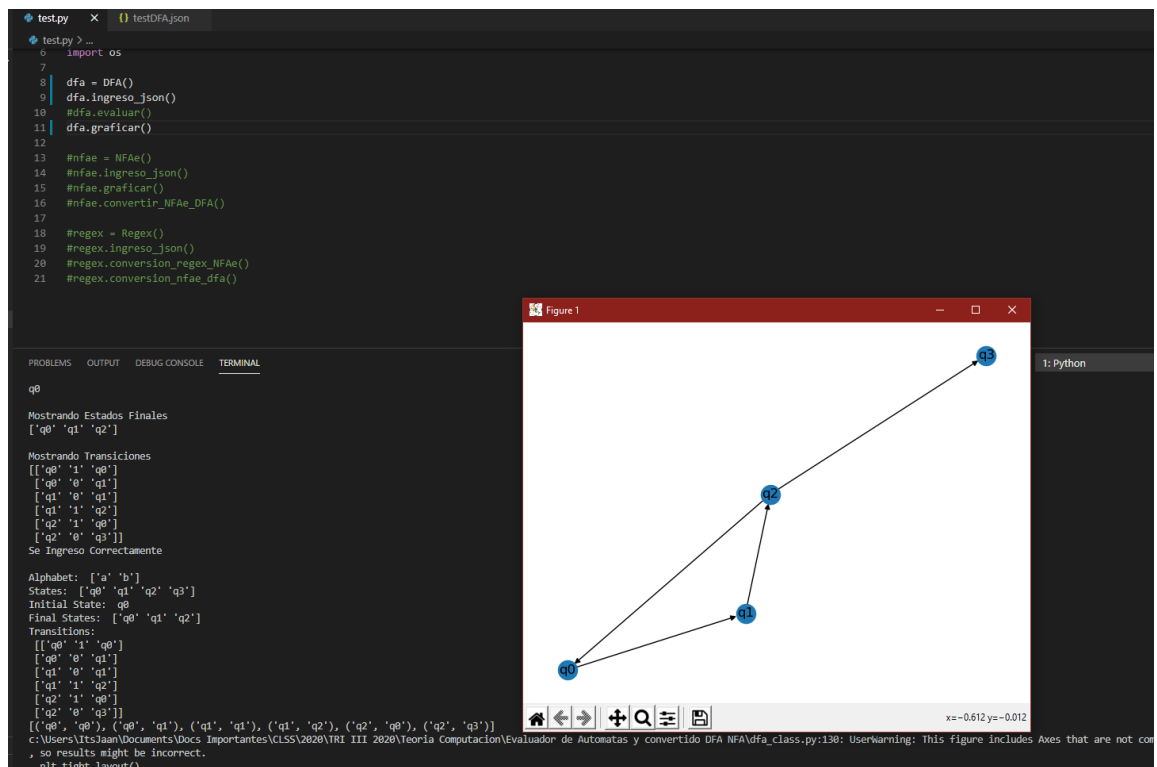


2.1 Ingresar un autómata para el siguiente lenguaje: {w | w no contiene la subcadena "010"}

2.1.1 Screenshots: Ingreso, Dibujo, evaluación de 5 ejemplos



```
1 {
2   "alphabet": {
3     "a": "0",
4     "b": "1"
5   },
6   "states": {
7     "q0": "...",
8     "q1": "...",
9     "q2": "...",
10    "q3": "...",
11  },
12  "initial_state": "q0",
13  "accepting_states": {
14    "q0": "...",
15    "q1": "...",
16    "q2": "...",
17    "q3": "...",
18  },
19  "transitions": [
20    [ "q0", "1", "q0" ],
21    [ "q0", "0", "q1" ],
22    [ "q1", "0", "q1" ],
23    [ "q1", "1", "q2" ],
24    [ "q2", "1", "q0" ],
25    [ "q2", "0", "q3" ],
26  ]
27 }
```



```
test.py > ...
1 import os
2
3 dfa = DFA()
4 dfa.ingreso_json()
5 #dfa.evaluar()
6 dfa.graficar()
7
8 #nfae = NFae()
9 #nfae.ingreso_json()
10 #nfae.graficar()
11 #nfae.convertir_NFAe_DFA()
12
13 #regex = Regex()
14 #regex.ingreso_json()
15 #regex.conversion_regex_NFAe()
16 #regex.conversion_nfae_dfa()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

q0

Mostrando Estados Finales

```
['q0' 'q1' 'q2']
```

Mostrando Transiciones

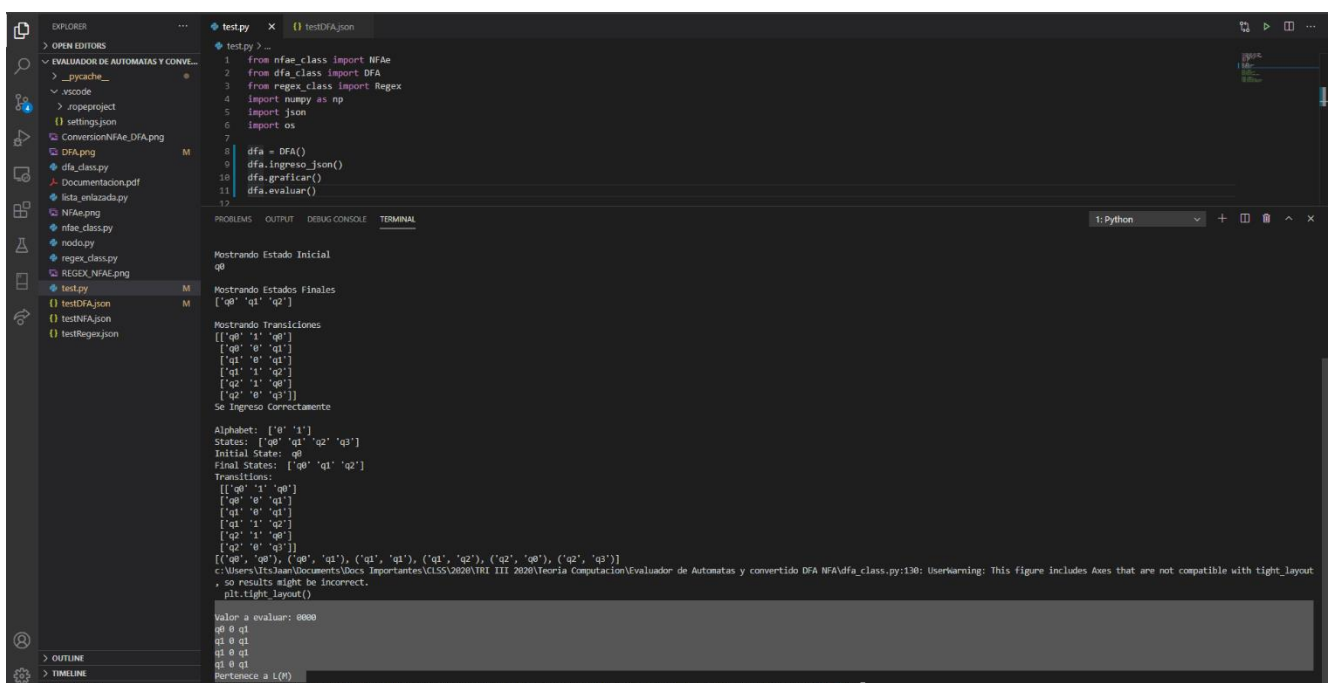
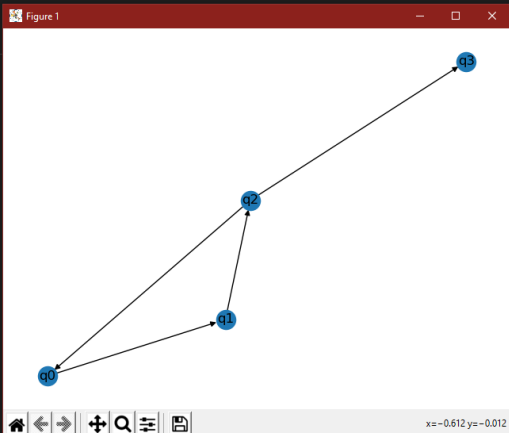
```
[["q0" "1" "q0"]
 ["q0" "0" "q1"]
 ["q1" "0" "q1"]
 ["q1" "1" "q2"]
 ["q2" "1" "q0"]
 ["q2" "0" "q3"]]
```

Se Ingreso Correctamente

Alphabet: ['a' 'b']
States: ['q0' 'q1' 'q2' 'q3']
Initial State: q0
Final States: ['q0' 'q1' 'q2']
Transitions:
[["q0" "1" "q0"]
 ["q0" "0" "q1"]
 ["q1" "0" "q1"]
 ["q1" "1" "q2"]
 ["q2" "1" "q0"]
 ["q2" "0" "q3"]]

[('q0', 'q0'), ('q0', 'q1'), ('q1', 'q1'), ('q1', 'q2'), ('q2', 'q0'), ('q2', 'q3')]

c:\Users\ItsJaan\Documents\Docs Importantes\CLSS\2020\TRI III 2020\Teoria Computacion\Evaluador de Automatas y convertido DFA NFA\dfa_class.py:130: UserWarning: This figure includes Axes that are not compatible with tight_layout, so results might be incorrect.
plt.tight_layout()



```
test.py > ...
1 from nfae_class import NFae
2 from dfa_class import DFA
3 from regex_class import Regex
4 import numpy as np
5 import json
6 import os
7
8 dfa = DFA()
9 dfa.ingreso_json()
10 dfa.graficar()
11 dfa.evaluar()
12
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Mostrando Estado Inicial

```
q0
```

Mostrando Estados Finales

```
['q0' 'q1' 'q2']
```

Mostrando Transiciones

```
[["q0" "1" "q0"]
 ["q0" "0" "q1"]
 ["q1" "0" "q1"]
 ["q1" "1" "q2"]
 ["q2" "1" "q0"]
 ["q2" "0" "q3"]]
```

Se Ingreso Correctamente

Alphabet: ['0' '1']
States: ['q0' 'q1' 'q2' 'q3']
Initial State: q0
Final States: ['q0' 'q1' 'q2']
Transitions:
[["q0" "1" "q0"]
 ["q0" "0" "q1"]
 ["q1" "0" "q1"]
 ["q1" "1" "q2"]
 ["q2" "1" "q0"]
 ["q2" "0" "q3"]]

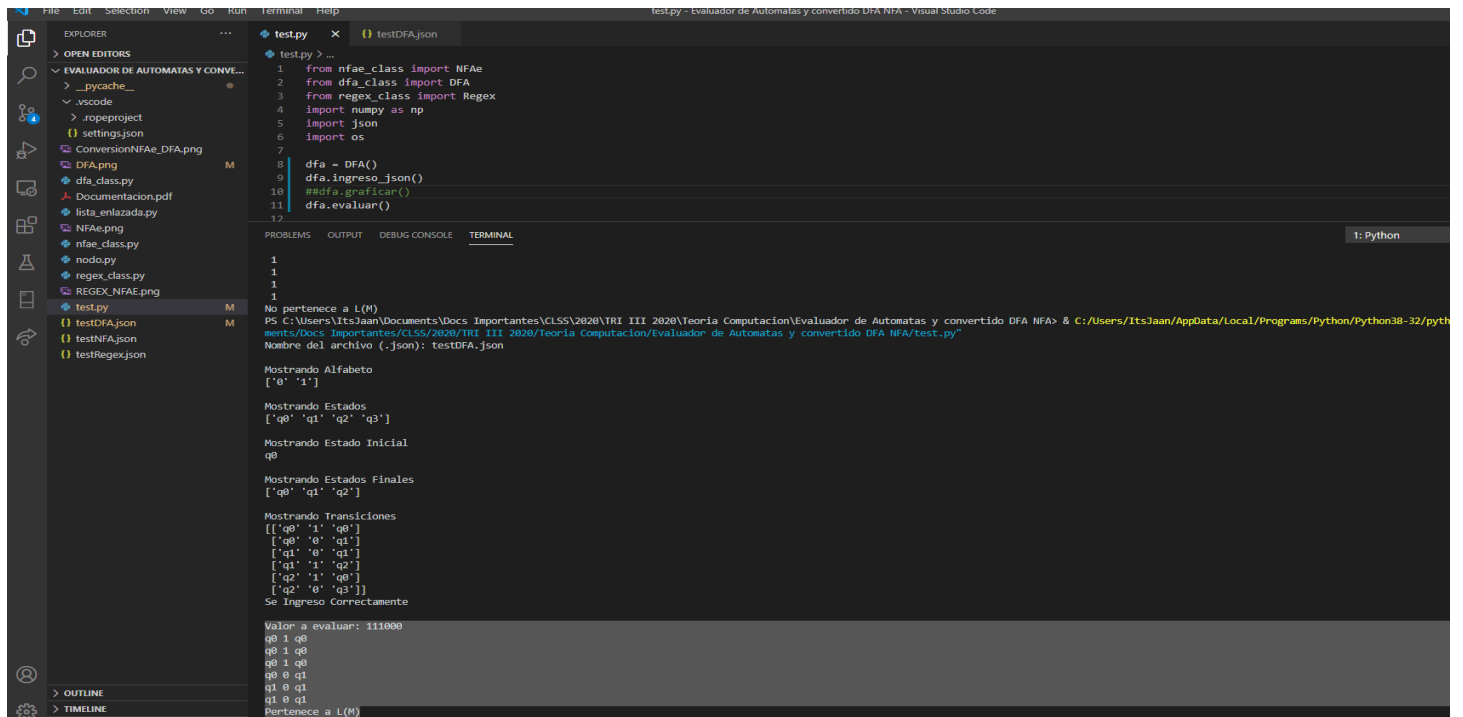
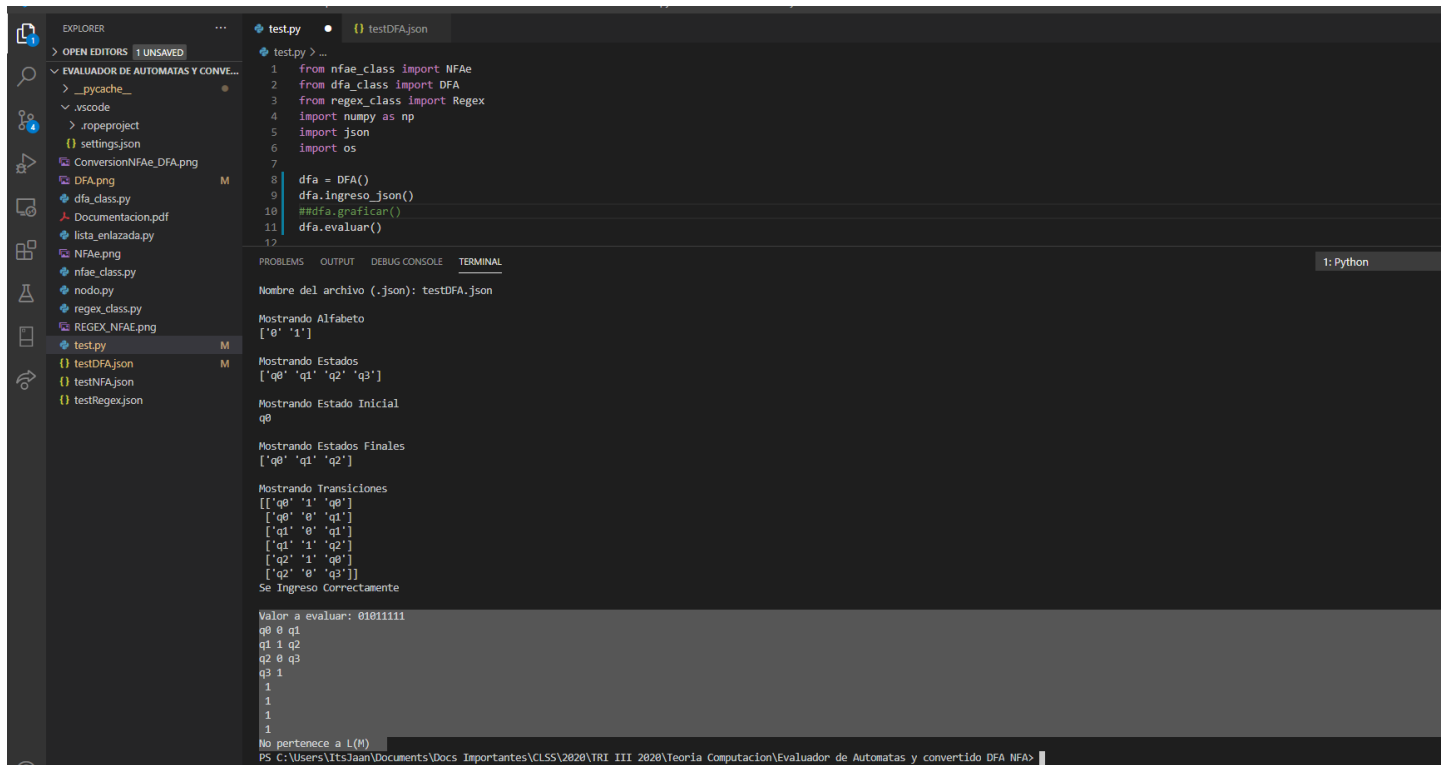
[('q0', 'q0'), ('q0', 'q1'), ('q1', 'q1'), ('q1', 'q2'), ('q2', 'q0'), ('q2', 'q3')]

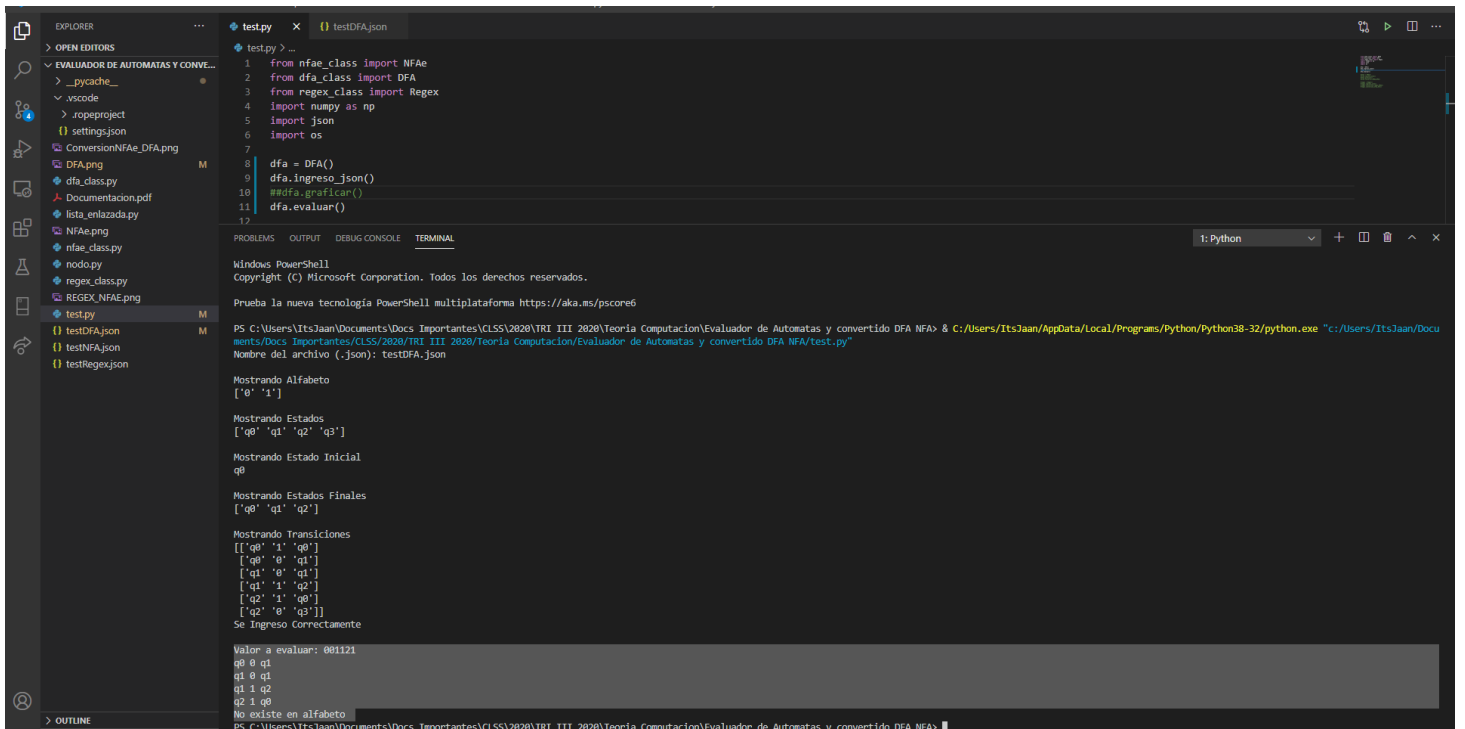
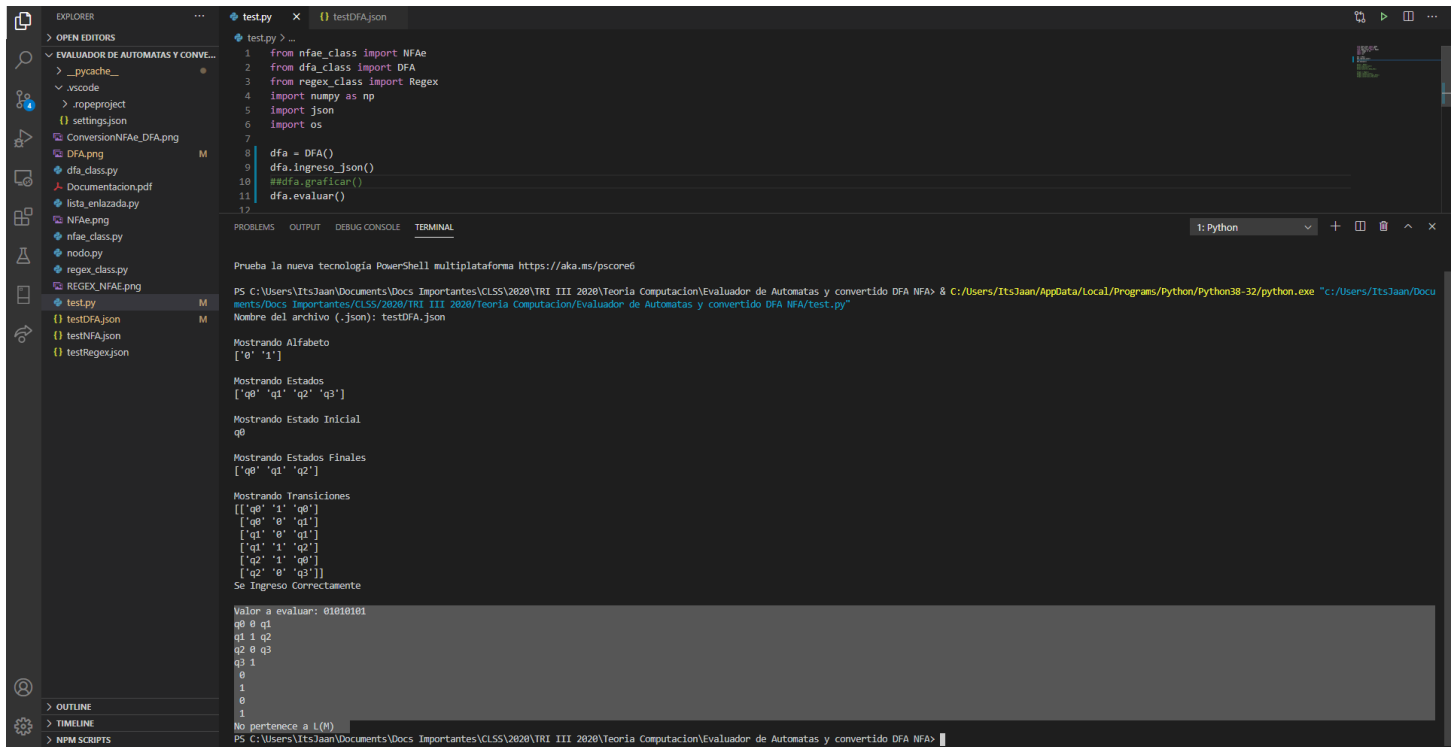
c:\Users\ItsJaan\Documents\Docs Importantes\CLSS\2020\TRI III 2020\Teoria Computacion\Evaluador de Automatas y convertido DFA NFA\dfa_class.py:130: UserWarning: This figure includes Axes that are not compatible with tight_layout, so results might be incorrect.
plt.tight_layout()

valor a evaluar: 0000

```
q0 0 q1
q1 0 q1
q1 0 q1
q3 0 q1
Resultado a L(0)
```

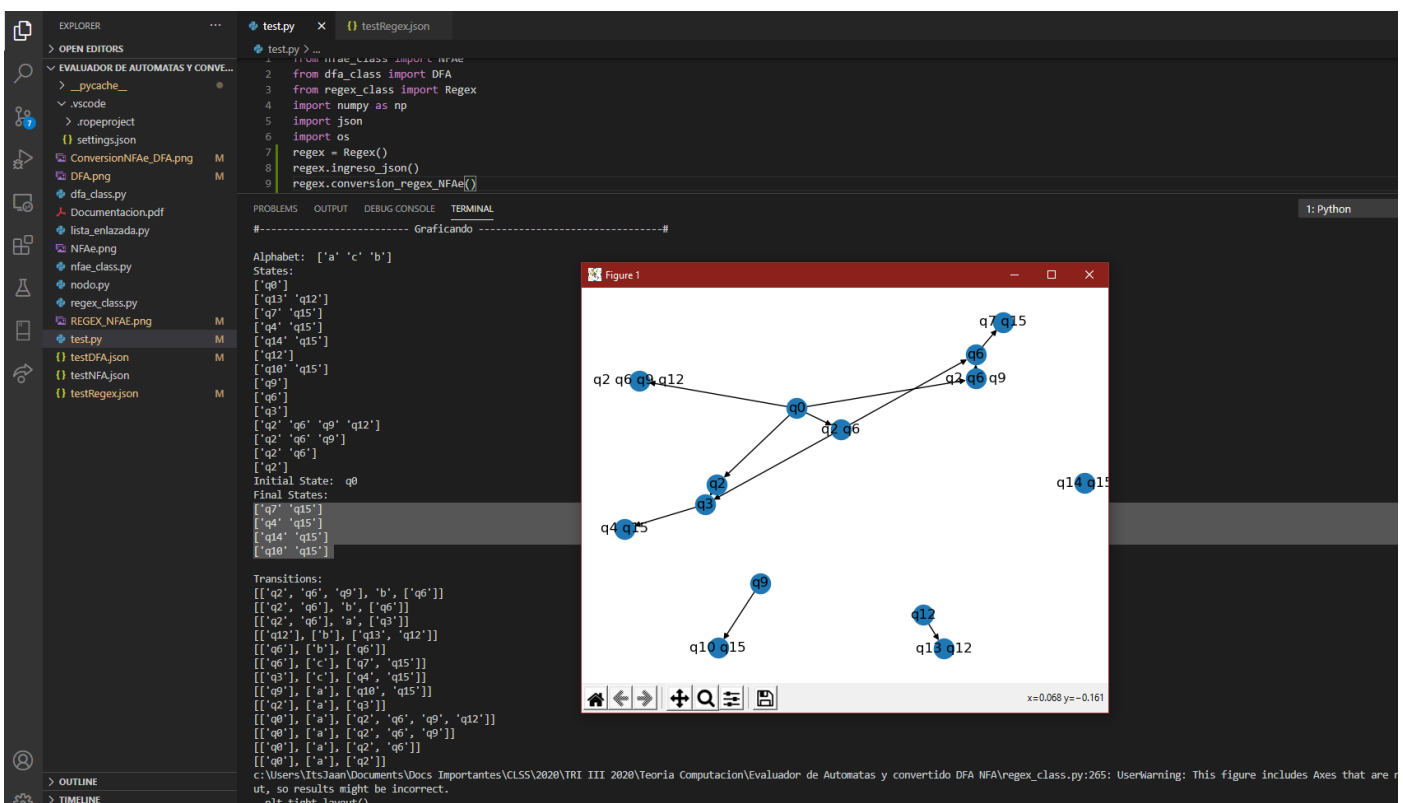
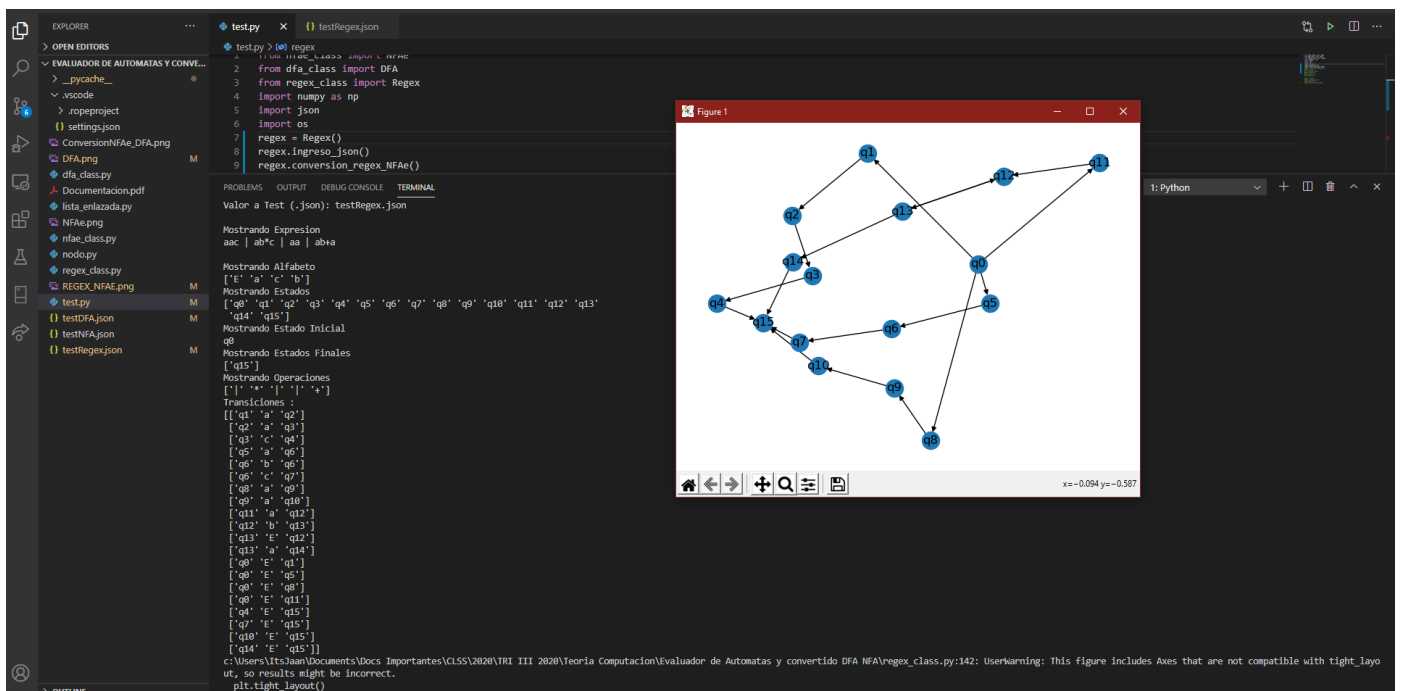
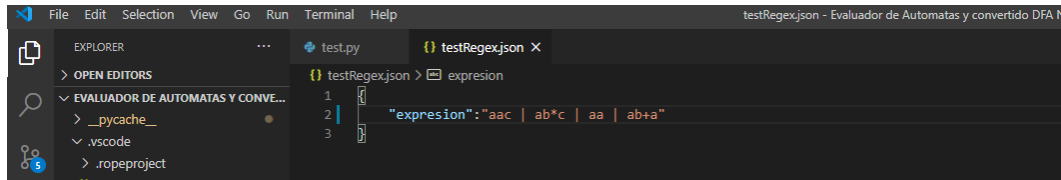
PS C:\Users\ItsJaan\Documents\Docs Importantes\CLSS\2020\TRI III 2020\Teoria Computacion\Evaluador de Automatas y convertido DFA NFA>

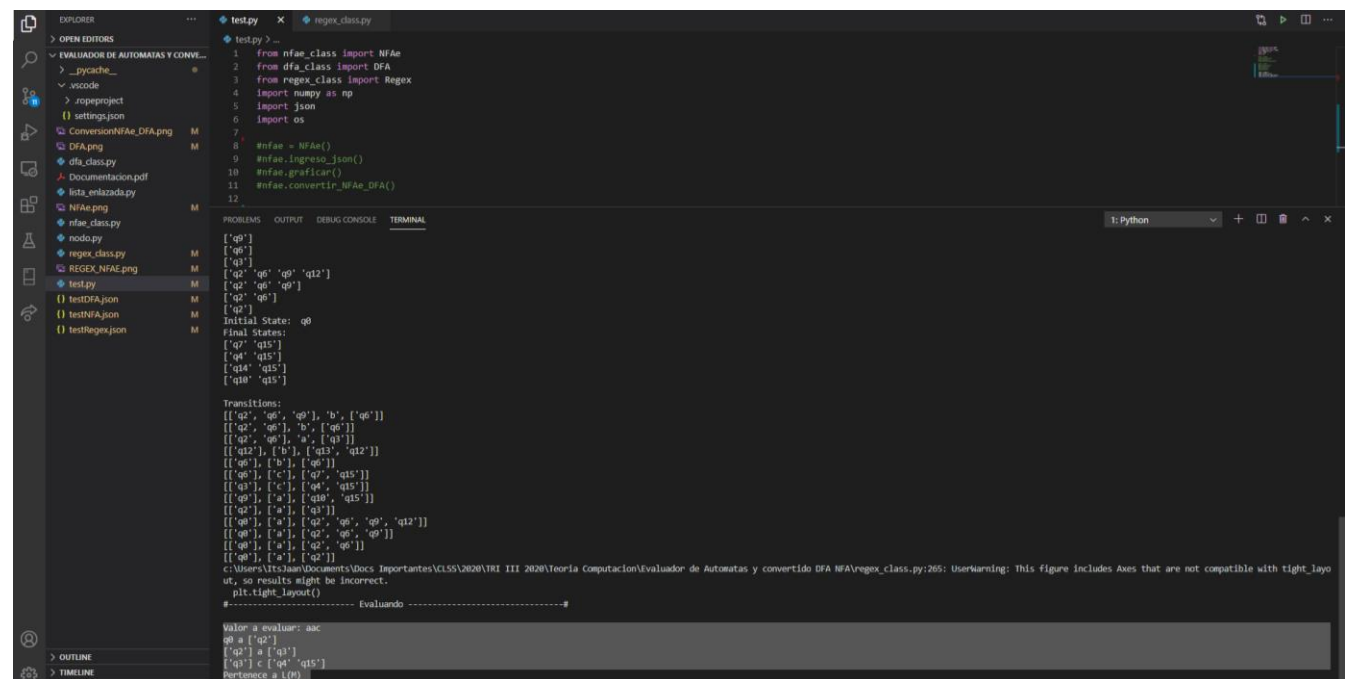
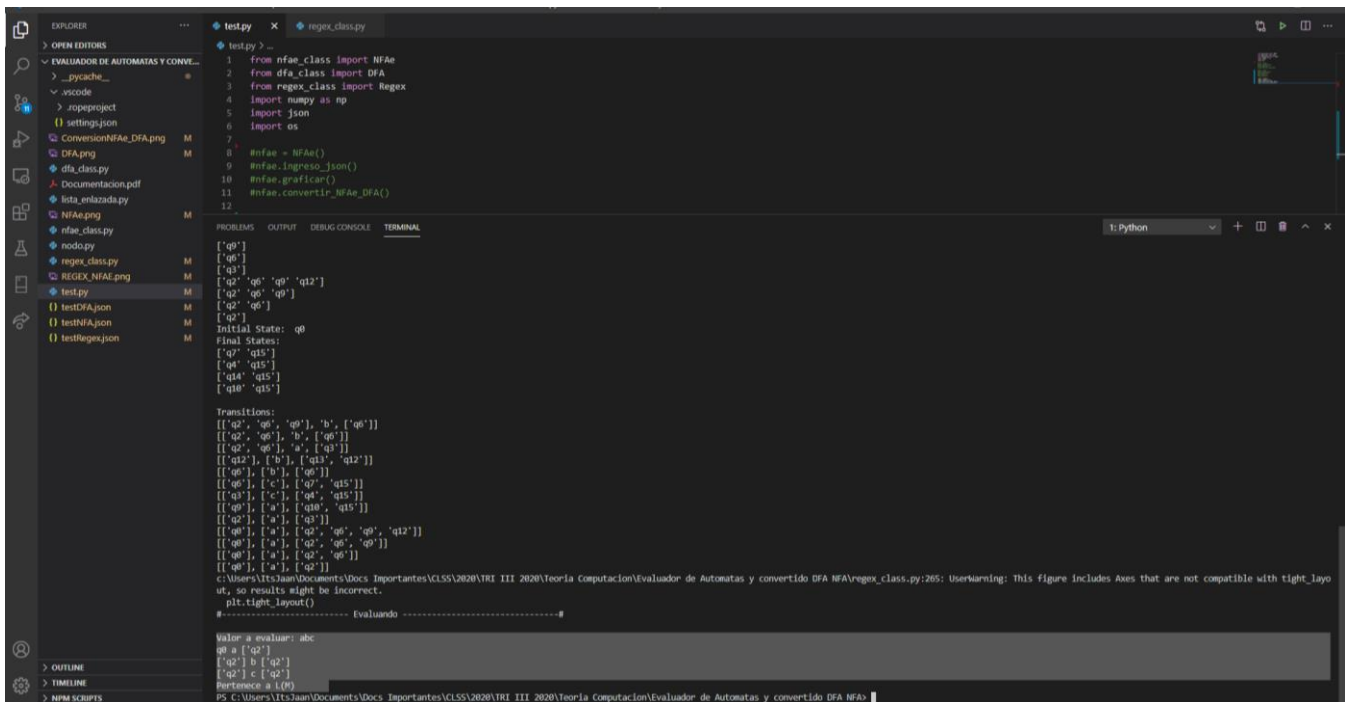


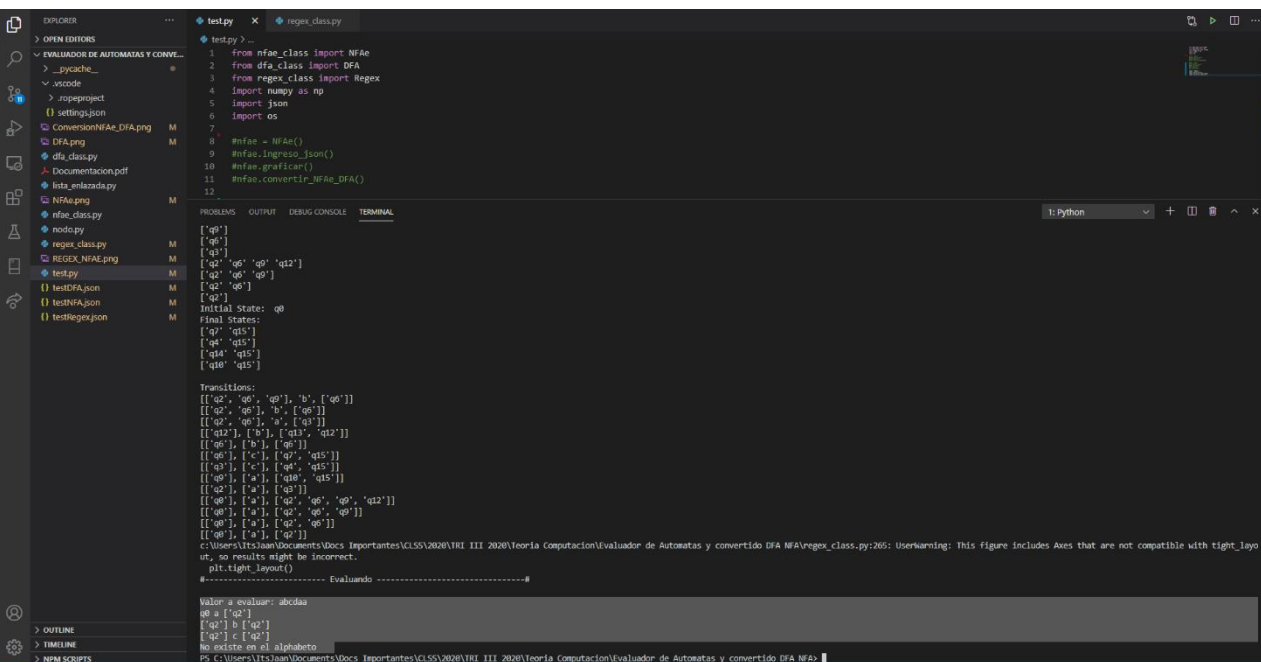
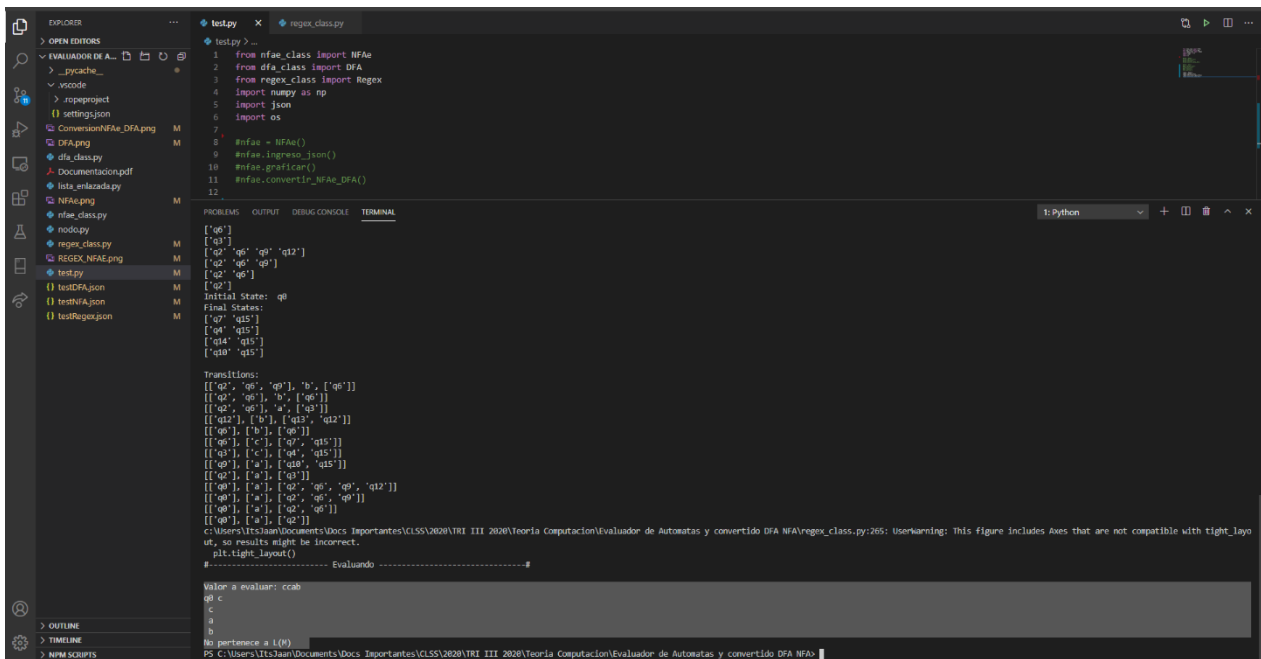
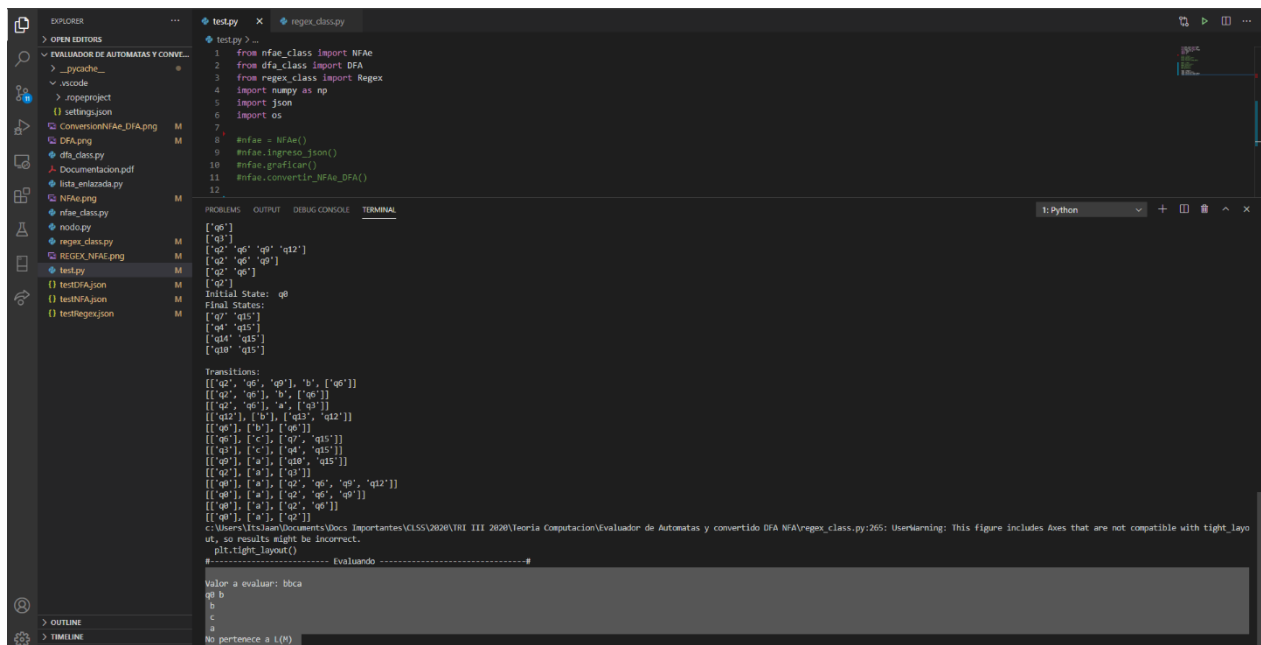


2.2 aac | ab*c | aa | ab+a

2.2.1 Ingresar, presentar (ER, NFAE, DFA), Evaluar 5 ejemplos







2.3 Ingresar, presentar (NFAE, DFA), Evaluar 5 ejemplos

```
EXPLORER
> OPEN EDITORS
  EVALUADOR DE AUTOMATAS Y CONVE...
    > _pycache_
    > .vscode
    > .ropeproject
    {} settings.json
    ConversionNFAe_DFA.png M
    DFA.png M
    dfa_class.py
    Documentacion.pdf
    lista_enlazada.py
    NFAe.png M
    nfae_class.py
    nodo.py
    regex_class.py
    REGEX_NFAE.png M
    test.py
    {} testDFA.json M
    {} testNFA.json M
    {} testRegex.json M

test.py
{} testNFA.json > [ ] transitions > [ ] 11
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36

{} testNFA.json
{
  "alphabet": {
    "E": "E",
    "a": "a",
    "b": "b"
  },
  "states": {
    "q0": "",
    "q1": "",
    "q2": "",
    "q3": "",
    "q4": "",
    "q5": "",
    "q6": "",
    "q7": ""
  },
  "initial_state": "q0",
  "accepting_states": {
    "q7": ""
  },
  "transitions": [
    [ "q0", "E", "q1"],
    [ "q0", "E", "q4"],
    [ "q1", "a", "q2"],
    [ "q2", "a", "q2"],
    [ "q2", "b", "q2"],
    [ "q2", "a", "q3"],
    [ "q3", "E", "q7"],
    [ "q4", "a", "q5"],
    [ "q5", "a", "q5"],
    [ "q5", "b", "q5"],
    [ "q5", "a", "q6"],
    [ "q6", "E", "q7"]
  ]
}
```

test.py

```
1 from nfae_class import NFAe
2 from dfa_class import DFA
3 from regex_class import Regex
4 import numpy as np
5 import json
6 import os
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Valor a Test (-,json): testNFA.json

Mostrando Alfabeto
['E' 'a' 'b']

Mostrando Estados
['q0' 'q1' 'q2' 'q3' 'q4' 'q5' 'q6' 'q7']

Mostrando Estado Inicial
q0

Mostrando Estados Finales
['q7']

Mostrando Transiciones
[[['q0' 'E' 'q1'],
['q0' 'E' 'q4'],
['q1' 'a' 'q2'],
['q2' 'a' 'q2'],
['q2' 'b' 'q2'],
['q2' 'a' 'q3'],
['q3' 'E' 'q7'],
['q4' 'a' 'q5'],
['q5' 'a' 'q5'],
['q5' 'b' 'q5'],
['q5' 'a' 'q6'],
['q6' 'E' 'q7']]]

Alphabet: ['E' 'a' 'b']
States: ['q0' 'q1' 'q2' 'q3' 'q4' 'q5' 'q6' 'q7']
Initial State: q0
Final States: ['q7']
Transitions:
[[['q0' 'E' 'q1'],
['q0' 'E' 'q4'],
['q1' 'a' 'q2'],
['q2' 'a' 'q2'],
['q2' 'b' 'q2'],
['q2' 'a' 'q3'],
['q3' 'E' 'q7'],
['q4' 'a' 'q5'],
['q5' 'a' 'q5'],
['q5' 'b' 'q5'],
['q5' 'a' 'q6'],
['q6' 'E' 'q7']]]

Figure 1

x=0.850 y=-0.024

c:\Users\lts\jaan\Documents\Docs Importantes\CLSS\2020\TRI III 2020\Teoria Computacion\Evaluador de Automatas y convertido DFA NFA\nfae_class.py:254: UserWarning: This figure includes Axes that are not compatible with tight layout


```
1 from nfae_class import NFae
2 from dfa_class import DFA
3 from regex_class import Regex
4 import numpy as np
5 import json
6 import os
7
8 nfae = NFae()
9 nfae.ingreso_json()
10 #nfae.graficar()
11 nfae.convertir_NFAe_DFA()
12
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

1: Python

[[['q0'], ['a'], ['q2']]]

#----- Graficando -----#

Alphabet: ['a' 'b']

States:

['q0']

['q5' 'q6' 'q7']

['q5']

['q2' 'q3' 'q7']

['q2' 'q5']

['q2']

Initial State: q0

Final States:

['q5' 'q6' 'q7']

['q2' 'q3' 'q7']

Transitions:

[[['q5', 'q6', 'q7'], 'b', ['q5']]]

[[['q2', 'q3', 'q7'], 'b', ['q2']]]

[[['q2', 'q5'], 'b', ['q2', 'q5']]]

[[['q5'], 'b'], ['q5']]]

[[['q2'], 'b'], ['q2']]]

[[['q5'], ['a'], ['q5', 'q6', 'q7']]]

[[['q2'], ['a'], ['q2', 'q3', 'q7']]]

[[['q0'], ['a'], ['q2', 'q5']]]

[[['q0'], ['a'], ['q2']]]

c:\Users\ItsJaen\Documents\Docs Importantes\CLSS\2020\TRI III 2020\Teoria Computacion\Evaluador de Automatas y convertido DFA NFA\nfae_class.py:189: UserWarning: This figure includes Axes that are not compatible with tight_layout, so results might be incorrect.

plt.tight_layout()

#----- Evaluando -----#

Valor a evaluar: abbb

q0 a [\'q2\']

['q2'] b [\'q2\']

['q2'] b [\'q2\']

['q2'] b [\'q2\']

['q2'] b [\'q2\']

No pertenece a L(H)

PS C:\Users\ItsJaen\Documents\Docs Importantes\CLSS\2020\TRI III 2020\Teoria Computacion\Evaluador de Automatas y convertido DFA NFA>

```
1 from nfae_class import NFae
2 from dfa_class import DFA
3 from regex_class import Regex
4 import numpy as np
5 import json
6 import os
7
8 nfae = NFae()
9 nfae.ingreso_json()
10 #nfae.graficar()
11 nfae.convertir_NFAe_DFA()
12
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

1: Python

[[['q0'], ['a'], ['q2']]]

#----- Graficando -----#

Alphabet: ['a' 'b']

States:

['q0']

['q5' 'q6' 'q7']

['q5']

['q2' 'q3' 'q7']

['q2' 'q5']

['q2']

Initial State: q0

Final States:

['q5' 'q6' 'q7']

['q2' 'q3' 'q7']

Transitions:

[[['q5', 'q6', 'q7'], 'b', ['q5']]]

[[['q2', 'q3', 'q7'], 'b', ['q2']]]

[[['q2', 'q5'], 'b', ['q2', 'q5']]]

[[['q5'], 'b'], ['q5']]]

[[['q2'], 'b'], ['q2']]]

[[['q5'], ['a'], ['q5', 'q6', 'q7']]]

[[['q2'], ['a'], ['q2', 'q3', 'q7']]]

[[['q0'], ['a'], ['q2', 'q5']]]

[[['q0'], ['a'], ['q2']]]

c:\Users\ItsJaen\Documents\Docs Importantes\CLSS\2020\TRI III 2020\Teoria Computacion\Evaluador de Automatas y convertido DFA NFA\nfae_class.py:189: UserWarning: This figure includes Axes that are not compatible with tight_layout, so results might be incorrect.

plt.tight_layout()

#----- Evaluando -----#

Valor a evaluar: bbbaa

q0 b

b

b

a

a

No pertenece a L(H)

PS C:\Users\ItsJaen\Documents\Docs Importantes\CLSS\2020\TRI III 2020\Teoria Computacion\Evaluador de Automatas y convertido DFA NFA>

