

Algoritmos de prueba para el analizador sintáctico

Algoritmo Marck Murillo

```
(println "Ingrese su nombre: ")
(def nombre (read-line))
(println "Ingrese su edad: ")
(def edad (read-line))
(def nombresecreto (str (subs nombre 2) edad) )
(println "tu Nombre secreto es: " nombresecreto)
(println "el cual se deletrea: " (seq nombresecreto))
```

Ejecución:

```
Prueba Murillo
clojure > (println "Ingrese su nombre: ")
nil
clojure > (def nombre (read-line))
variable_expression: stament
clojure > (println "Ingrese su edad: ")
nil
clojure > (def edad (read-line))
variable_expression: stament
clojure > (def nombresecreto (str (subs nombre 2) edad) )
yacc: Syntax error at line 1, token=ID
None
clojure > (println "tu Nombre secreto es: " nombresecreto)
nil
clojure > (println "el cual se deletrea: " (seq nombresecreto))
yacc: Syntax error at line 1, token=FUNCTION_SEQ
None
>>>
```

Algoritmo Henriquez

```
def prueba():
    print("Prueba Operaciones")
    linea = "(+ (* (- 8 4) 2) 2)"
    print("clojure > " + linea)
    result = parser.parse(linea)
```

```
print(result)
linea = "(>= 2 2)"
print("clojure > " + linea)
result = parser.parse(linea)
print(result)
linea = "(def variable true)"
print("clojure > " + linea)
result = parser.parse(linea)
print(result)
linea = "(and true false)"
print("clojure > " + linea)
result = parser.parse(linea)
print(result)
```

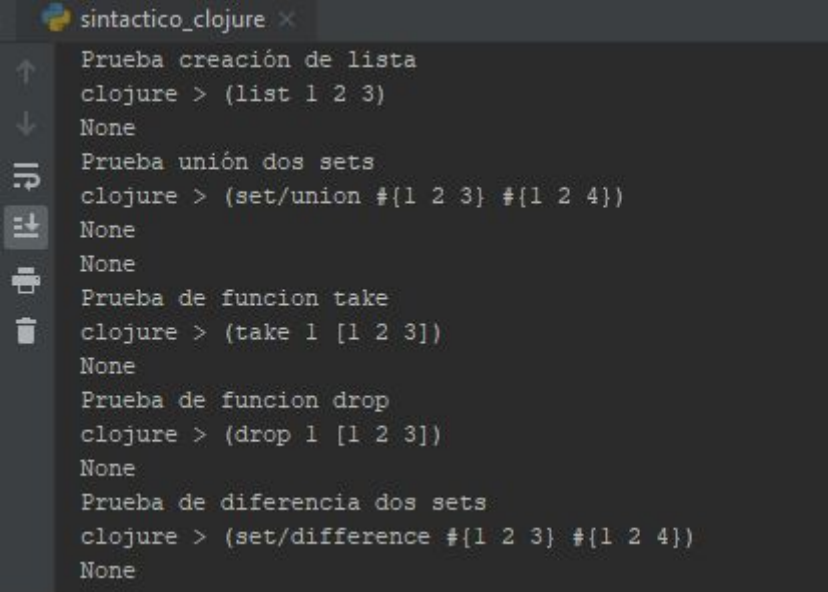
Ejecución

```
Prueba Operaciones
clojure > (+ (* (- 8 4) 2) 2)
10
clojure > (>= 2 2)
True
clojure > (def variable true)
variable_expression: BOOLEAN
clojure > (and true false)
false
PS C:\Users\HP\OneDrive - Escuela Superior Politécnica del Litoral\
proyecto-Clojure> █
```

Algoritmo Franklin Ordoñez

```
(list 1 2 3)
(set/union '(1 2 3) '(1 2 4))
(take 1 [1 2 3])
(drop 1 [1 2 3])
(set/difference #{1 2 3} #{1 2 4})
```

Ejecución



```
sintactico_clojure x
Prueba creación de lista
clojure > (list 1 2 3)
None
Prueba unión dos sets
clojure > (set/union #{1 2 3} #{1 2 4})
None
None
Prueba de funcion take
clojure > (take 1 [1 2 3])
None
Prueba de funcion drop
clojure > (drop 1 [1 2 3])
None
Prueba de diferencia dos sets
clojure > (set/difference #{1 2 3} #{1 2 4})
None
```