Proyecto 01

Abner Iván García Alegría 21285

Oscar Esteban Donis Martínez 21610

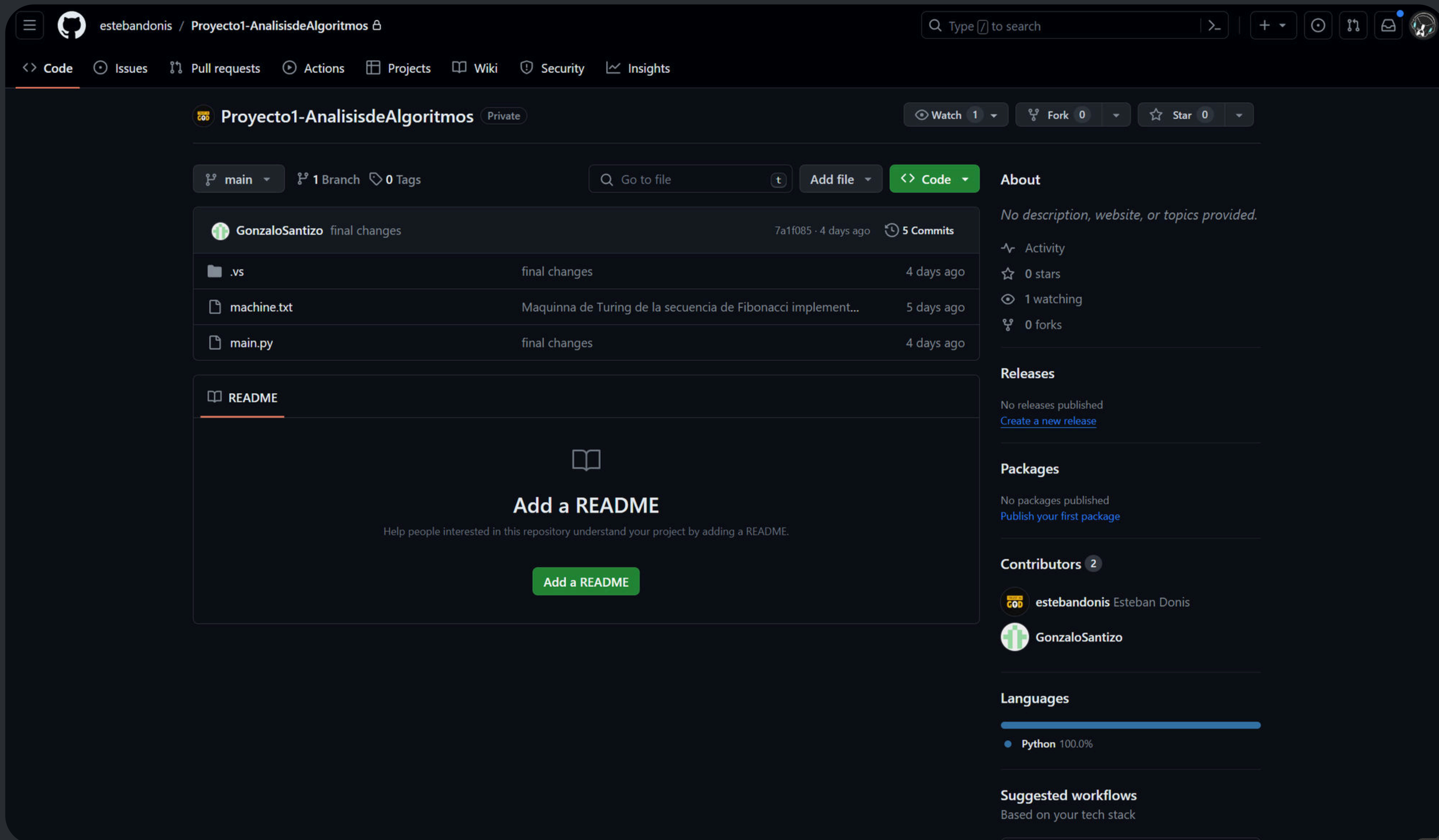
Astrid Marie Glauser Oliva 21299

Gonzalo Enrique Santizo vega 21504

Jose Daniel Gomez Cabrera 21429

Repositorio de Github

https://github.com/estebandonis/Proyecto1-AnalisisdeAlgoritmos



machine.txt

```
ESTADOS = {q0, q1, q2, q3, q4, q5, q6, q7, q8, q9, q10, q11, q12, q13, q14, q15, q16, q17, q18, q19, q20, q21, q22, q23, q24, q25, q26, q27, q28, q29, q30, q31}
ALFABETO DE ENTRADA = {|}
ALFABETO DE CINTA = {|, 1, X, Y, B}
FUNCION DE TRANSICION = {
    (q0, B) → (q0, B, R),
    (q0, |) → (q1, B, R),
    (q1, B) → (q5, |, R),
    (q1, |) → (q2, |, R),
    (q2, B) → (q4, B, R),
    (q2, |) → (q3, |, R),
    (q3, |) → (q3, |, R),
    (q3, X) → (q3, X, R),
    (q3, B) → (q6, X, R),
    (q6, B) → (q7, |, L),
    (q7, X) → (q8, X, L),
    (q8, |) → (q8, |, L),
    (q8, B) → (q9, B, R),
    (q8, X) → (q17, X, L),
    (q9, |) → (q3, B, R),
     ...
```

main.py

```python
def main():
    # Leemos los datos del archivo
    estados, alfabeto_entrada, alfabeto_cinta, func_transition, estado_inicial, sim_blanco, estados_aceptacion = read_file()

    # Definimos la secuencia de Fibonacci como entrada
    input_sequence = int(input("Ingrese el numero a averiguar en la secuencia de Fibonacci: "))
    numCinta = ""
    i = 0
    while i < input_sequence:
        numCinta += '|'
        i += 1
    # Convertimos la entrada en una cinta con un número definido de blanks
    blank_number = 100
    w = list(sim_blanco * blank_number + numCinta + sim_blanco * (blank_number*4))

    # Le pasamos los datos de la máquina de turing a la función para ejecutarla
    maquina_turing(estados, alfabeto_entrada, alfabeto_cinta, func_transition, estado_inicial, sim_blanco, estados_aceptacion, w, blank_number)
    print("Número ingresado: ", input_sequence)

if __name__ == "__main__":
    start_time = time.time()  # Start measuring time
    main()  # Call the main function
    end_time = time.time()  # Stop measuring time
    execution_time = end_time - start_time  # Calculate execution time
    print("Execution time:", execution_time, "seconds")
```

Entrada	Tiempo de ejecución en segundos
5	0.78
6	0.85
7	1.33
8	1.48
9	2.33
10	4.71
11	10.31
12	26.37

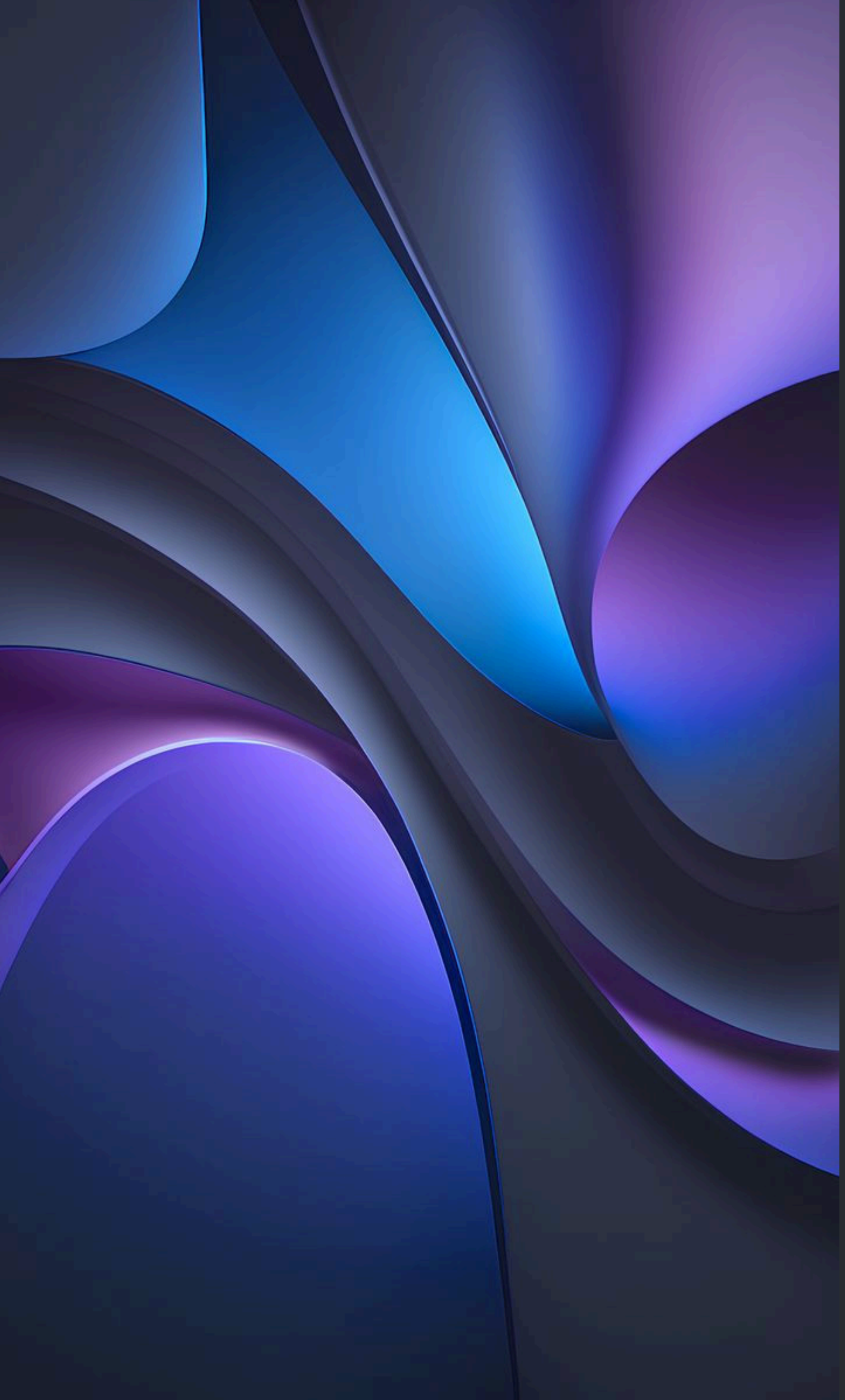
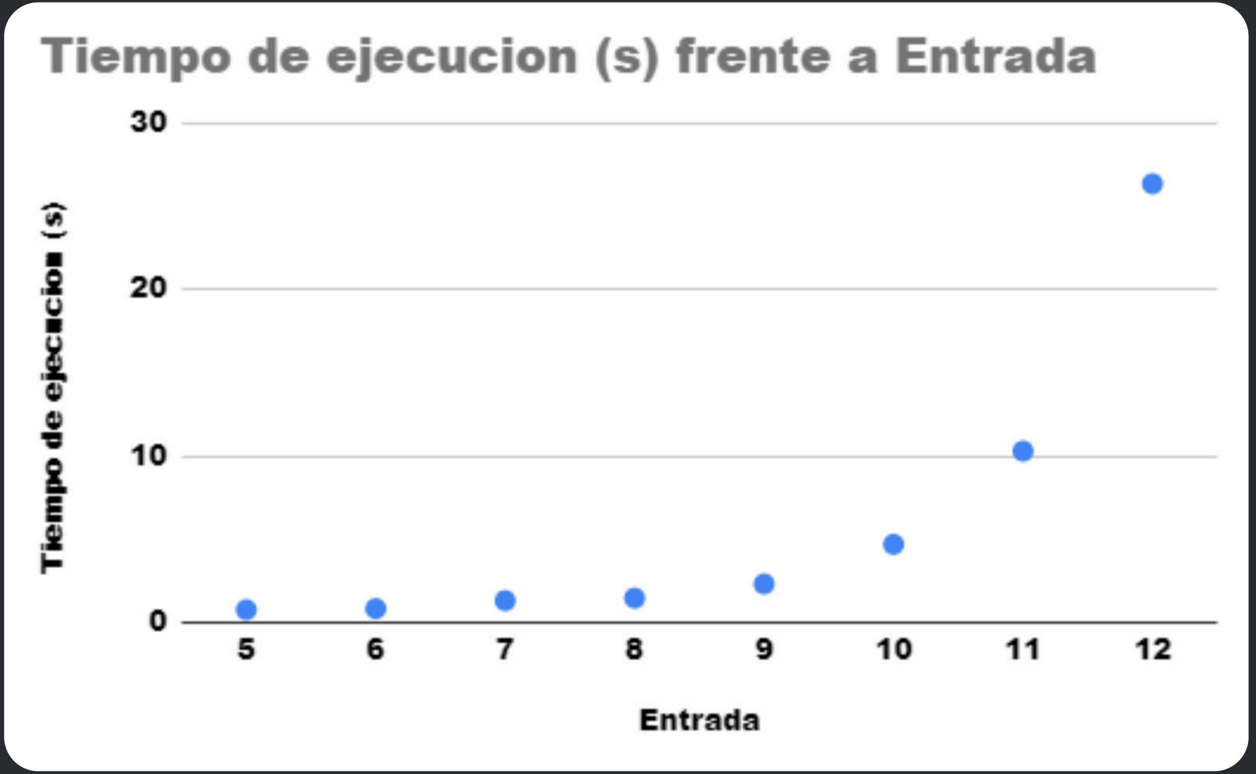
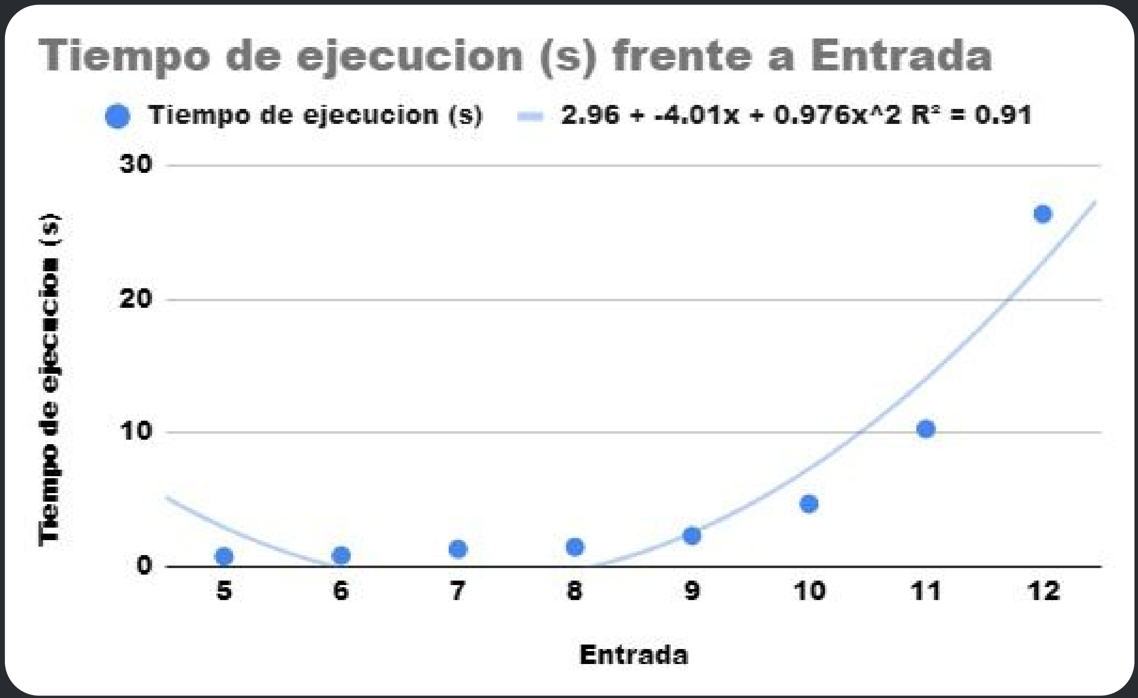


Diagrama de Dispersión



El algoritmo clásico para generar la secuencia de Fibonacci es iterativo, sumando los dos números anteriores de la secuencia para obtener el siguiente. A medida que la secuencia avanza, el tiempo necesario para calcular cada nuevo término aumenta, debido al crecimiento rápido de los números en la secuencia. Esto puede resultar en un crecimiento que parece exponencial en una gráfica que compare la entrada (el índice n de la secuencia) con el tiempo de cálculo necesario.

Regresión polinomial que se ajuste mejor a los datos



Tiempo de ejecución

$$2.96 - 4.01x + 0.976x^2$$

Donde "x" representa algún valor o parámetro relacionado con la entrada o el tamaño del problema. La ecuación muestra cómo el tiempo de ejecución varía a medida que "x" cambia.

$$R^2 = 0.91$$

El coeficiente R2 es una medida de cuán bien se ajusta el modelo a los datos reales. Un valor de R2 de 0.91 indica que el modelo explica aproximadamente el 91% de la variabilidad en los datos observados. En otras palabras, el modelo tiene un buen ajuste a los datos en este caso particular.


Verificamos si el número de palitos es menor a 3
Verificamos si la cantidad de palitos restantes es menor a 2.
Si la cantidad de palitos restantes es 1, entonces sumamos todas las cantidades, eliminándolas de la cinta a su vez.