~\Desktop\Nivel 5\trabalho nível 5.py

```python
import time

nome_arquivo = 'my note'


dados = """Perna Braço mao orelha"""


with open(nome_arquivo, 'w') as arquivo:
    arquivo.write(dados)

print(f"Arquivo '{nome_arquivo}' criado e dados escritos com sucesso.")




def bubble_sort(arr):
    n = len(arr)
    for i in range(n):
        for j in range(0, n-i-1):
            if arr[j] > arr[j+1]:
                arr[j], arr[j+1] = arr[j+1], arr[j]


def selection_sort(arr):
    n = len(arr)
    for i in range(n):
        min_idx = i
        for j in range(i+1, n):
            if arr[j] < arr[min_idx]:
                min_idx = j
        arr[i], arr[min_idx] = arr[min_idx], arr[i]


def read_words_from_file(file_path):
    words = []
    try:
        with open(file_path, 'r') as file:
            for line in file:
                words.extend(line.split())
        if not words:
            print("O arquivo está vazio ou não contém palavras.")
    except FileNotFoundError:
        print(f"Arquivo '{file_path}' não encontrado.")
    return words


def measure_time(sort_function, arr):
    start_time = time.time()
    sort_function(arr)
    end_time = time.time()
    return end_time - start_time


def main():
    file_path = 'my note'
    words = read_words_from_file(file_path)

    if not words:
        return


    words_bubble_sort = words.copy()
    bubble_sort_time = measure_time(bubble_sort, words_bubble_sort)
    print(f"Bubble Sort - Tempo de execução: {bubble_sort_time:.6f} segundos")


    words_selection_sort = words.copy()
    selection_sort_time = measure_time(selection_sort, words_selection_sort)
    print(f"Selection Sort - Tempo de execução: {selection_sort_time:.6f} segundos")


    words_native_sort = words.copy()
    start_time = time.time()
    words_native_sort.sort()
    end_time = time.time()
    native_sort_time = end_time - start_time
    print(f"Sort Nativo - Tempo de execução: {native_sort_time:.6f} segundos")


```

```python
    print(f"Exemplo de palavras ordenadas (nativo): {words_native_sort[:10]}...")


    output_file_path = 'sorted_words.txt'
    try:
        with open(output_file_path, 'w') as file:
            file.write('\n'.join(words_native_sort))
        print(f"Palavras ordenadas salvas em {output_file_path}")
    except Exception as e:
        print(f"Erro ao salvar o arquivo: {e}")

if __name__ == "__main__":
    main()

```