



# Escuela Superior de Computo

## Paradigmas de programación

### Practica 1

#### Integrantes

Elizalde Hernández Alan

Toral Hernández Leonardo Javier

Solis lugo Mayra

Solares Velasco Arturo Misael

Reyes Ruiz Yoselyn Estefany  
3CV1

Repositorio:

<https://github.com/alaneh/paradigmas-escom-2024>



## Diseño de la solución

Hemos desglosado cada una de las especificaciones del problema en pequeñas funciones para un enfoque más claro y modular

### Función 'generar'

Esta función se encarga de escribir en un archivo de texto los números del 1 a 'n', dependiendo de la cantidad que se desee manejar. Por defecto, utiliza el valor mínimo de 'n' como 1000.

### Función 'shuffle'

Toma el archivo de texto generado previamente y reordena de forma aleatoria los números que contiene.

### Función 'leer'

Diseñada para la lectura de archivos de texto, permite pasar el contenido a arreglos permitiendo su manipulación.

### Función 'escribir'

Facilita la escritura o reescritura de archivos de texto para su uso posterior o para marcar la salida de datos.

### Función de búsqueda

Permite la búsqueda de un elemento específico en los archivos de texto.

### Función 'ordenar'

Esta función posibilita el reordenamiento del archivo de texto utilizando el orden aleatorio generado previamente.

### Cálculo del tiempo

Contamos el tiempo desde la función generar hasta el final de imprimir el arreglo ordenado en un archivo de texto.

### Función principal

Nos permite ordenar como se manejarán las funciones y que el cálculo de tiempo sea mas sencillo

Se explicará detalladamente cada una de las funciones con el lenguaje Python porque es más simplificado.

## Desarrollo del programa

### Función generar

En esta primera sección abrimos el archivo “numeros.txt” para proceder a llenar el archivo utilizando un ciclo for para llenar el archivo con los números de 1 a N en el archivo como una cadena de caracteres.

```
def generar():  
    with open("numeros.txt", "w") as archivo:  
        for i in range(1, 1001):  
            archivo.write(str(i) + "\n")
```

### Función shuffle

Python tiene una función para ordenar un arreglo de forma aleatoria integrada así que es con la que trabajamos para poder llenar el arreglo con números aleatorios.

```
random.shuffle(numeros)
```

### Función ‘leer’

```
def leer(filename):  
    numeros = []  
    with open(filename, "r") as archivo:  
        for linea in archivo:
```

```
        numeros.append(int(linea.strip()))
    return numeros
```

la función va a abrir un archivo, lo asigna al identificador archivo y se procede a iterar sobre cada línea del archivo se van agregando los números a la lista después de eliminar los espacios en blanco y después devuelve la lista de números en un arreglo.

## Función 'escribir'

Abre un archivo y escribe el contenido de un arreglo en cada una de las líneas del archivo y luego regresa un mensaje para saber que archivo fue el que se modificó, escribe cada numero seguido de un salto de línea.

```
def escribir(numeros, filename):
    with open(filename, "w") as archivo:
        for num in numeros:
            archivo.write(str(num) + "\n")
    print(f"Se escribió en el archivo '{filename}'")
```

## Función de búsqueda

La función recorrerá todo el arreglo y comparará la entrada que le dimos para saber si esta o no esta en el arreglo elemento a elemento

```
def busqueda_secuencial(numeros, dato):
    for i, num in enumerate(numeros):
        if num == dato:
            return True, i
    return False, -1
```

## Función de ordenamiento

Python ya incluye sort para ordenar así que ese será el algoritmo que se utilizara para el ordenamiento

```
numeros.sort()
```

**Función main:** Es la función principal del programa. Aquí se realizan las siguientes acciones:

- Se reserva memoria para un arreglo de 1000 enteros llamado numeros.
- Se llama a la función generar() para crear un archivo de texto llamado numeros.txt que contiene los números del 1 al 1000.
- Se llama a la función leer() para leer los números del archivo numeros.txt y cargarlos en el arreglo numeros.
- Se llama a la función shuffle() para aleatorizar el orden de los números en el arreglo numeros.
- Se llama a la función escribir() para escribir los números aleatorios en un archivo llamado numeros\_aleatorios.txt.
- Se solicita al usuario que ingrese un número para buscar en el arreglo.
- Se llama a la función busqueda\_secuencial() para buscar el número ingresado por el usuario en el arreglo numeros.
- Se llama a la función ordenar\_numeros() para ordenar los números en el arreglo numeros.
- Se llama a la función escribir() para escribir los números ordenados en un archivo llamado numeros\_ordenados.txt.
- Se libera la memoria del arreglo numeros.
- Se calcula y muestra el tiempo de ejecución del programa.

```
def main():
    inicio = time.time()

    generar()
    numeros = leer("numeros.txt")
    random.shuffle(numeros)
    escribir(numeros, "numeros_aleatorios.txt")
    print("Archivo de texto de números aleatorios generado")

    dato = int(input("Ingrese el número a buscar: "))
    encontrado, posicion = busqueda_secuencial(numeros, dato)
    if encontrado:
        print(f"El número {dato} fue encontrado en la posición {posicion}")
    else:
        print(f"El número {dato} no fue encontrado en el archivo")

    numeros.sort()
    escribir(numeros, "numeros_ordenados.txt")

    fin = time.time()
    tiempo_ejecucion = fin - inicio
    print(f"Tiempo de ejecución: {tiempo_ejecucion:.6f} segundos")
```

# Código en Python

```
Python
import random

import time

def generar():
    with open("numeros.txt", "w") as archivo:
        for i in range(1, 1001):
            archivo.write(str(i) + "\n")

def leer(filename):
    numeros = []
    with open(filename, "r") as archivo:
        for linea in archivo:
            numeros.append(int(linea.strip()))
    return numeros

def escribir(numeros, filename):
    with open(filename, "w") as archivo:
        for num in numeros:
            archivo.write(str(num) + "\n")
    print(f"Se escribió en el archivo '{filename}'")

# Función para búsqueda secuencial
def busqueda_secuencial(numeros, dato):
    for i, num in enumerate(numeros):
        if num == dato:
            return True, i
    return False, -1

def main():
    inicio = time.time()

    generar()
    numeros = leer("numeros.txt")
```

```

random.shuffle(numeros)
escribir(numeros, "numeros_aleatorios.txt")
print("Archivo de texto de números aleatorios generado")

dato = int(input("Ingrese el número a buscar: "))
encontrado, posicion = busqueda_secuencial(numeros, dato)
if encontrado:
    print(f"El número {dato} fue encontrado en la posición {posicion}")
else:
    print(f"El número {dato} no fue encontrado en el archivo")

numeros.sort()
escribir(numeros, "numeros_ordenados.txt")

fin = time.time()
tiempo_ejecucion = fin - inicio
print(f"Tiempo de ejecución: {tiempo_ejecucion:.6f} segundos")

```

```

if __name__ ==
    "__main__":
        main()

```

## Código en lenguaje C

```

C
#include <stdio.h>

#include <stdlib.h>
#include <time.h>
#include <string.h>

/* Prototipos */
void generar();
void leer(int numeros[], const char* filename);
void shuffle(int array[], int n);
void escribir(int numeros[], const char* filename);
int busqueda_secuencial(int *numeros, int tamaño, int dato, int *posicion);
void ordenar_numeros(int *numeros, int tamaño);

int main() {

```

```

int *numeros = (int *)malloc(1000 * sizeof(int));
int tamano = 1000;
int dato;
int posicion;

clock_t inicio, fin;
double tiempo;
inicio = clock();
generar();
leer(numeros, "numeros.txt");
shuffle(numeros, 1000);
escribir(numeros, "numeros_aleatorios.txt");
printf("%s\n", "Archivo de texto de numeros aleatorios generado");

printf("Ingrese el numero a buscar: ");
scanf("%d", &dato);

if (busqueda_secuencial(numeros, tamano, dato, &posicion)) {
    printf("El numero %d fue encontrado en la posicion %d\n", dato,
posicion);
} else {
    printf("El numero %d no fue encontrado en el archivo\n", dato);
}

ordenar_numeros(numeros, tamano);

escribir(numeros, "numeros_ordenados.txt");

free(numeros); /* libera la memoria del arreglo */

fin = clock();

tiempo = (double)(fin - inicio) / CLOCKS_PER_SEC;
printf("Tiempo de ejecucion: %.6f segundos\n", tiempo);

return 0;
}

void generar() {
    FILE *archivo;
    archivo = fopen("numeros.txt", "w");
    if (archivo == NULL) {
        printf("Error al abrir el archivo.");
        return;
    }
}

```



```

    for (int i = 1; i <= 1000; i++) {
        fprintf(archivo, "%d\n", i);
    }

    fclose(archivo);
}

void swap(int *a, int *b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}

void shuffle(int array[], int n) {
    srand(time(NULL));
    for (int i = n - 1; i > 0; i--) {
        int j = rand() % (i + 1);
        swap(&array[i], &array[j]);
    }
}

void leer(int numeros[], const char* filename) {
    FILE *archivo;
    archivo = fopen(filename, "r");
    if (archivo == NULL) {
        printf("Error al abrir el archivo.");
        return;
    }

    int i = 0;
    while (fscanf(archivo, "%d", &numeros[i]) != EOF && i < 1000) {
        i++;
    }

    fclose(archivo);
}

void escribir(int numeros[], const char* filename) {
    FILE *archivo;
    archivo = fopen(filename, "w");
    if (archivo == NULL) {
        printf("Error al abrir el archivo.");
        return;
    }
}

```

```
    for (int i = 0; i < 1000; i++) {
        fprintf(archivo, "%d\n", numeros[i]);
    }

    fclose(archivo);
    printf("Se escribio en el archivo '%s'\n", filename);
}

int busqueda_secuencial(int *numeros, int tamaño, int dato, int *posicion)
{
    for (int i = 0; i < tamaño; i++) {
        if (numeros[i] == dato) {
            *posicion = i;
            return 1; // Se encontró el dato
        }
    }
    return 0; // No se encontró el dato
}

void ordenar_numeros(int *numeros, int tamaño) {
    for (int i = 0; i < tamaño - 1; i++) {
        for (int j = 0; j < tamaño - i - 1; j++) {
            if (numeros[j] > numeros[j + 1]) {
                int temp = numeros[j];
                numeros[j] = numeros[j + 1];
                numeros[j + 1] = temp;
            }
        }
    }
}
```