

Informe de Practica 01

Tema: Array

Nota

Integrantes	Escuela	Asignatura
Juan Sergio Zeballos Perez José Carlos Huaranca Condori Roni Companocca Checco Jeremy Joshua Perez Huamani Jorge Gabriel Llerena Huanca	Escuela Profesional de Ingeniería de Sistemas	Semestre: II Código:

Practica	Tema	Duración
01	Array	02 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2023 - B	Del 29 Septiembre 2023	Al 04 Octubre 2023

1. TAREA

- Práctica 01: Elaborar un proyecto utilizando git. donde se elabore un sistema para ingresar datos de alumnos universitarios. (Clase Student) El sistema debe almacenar los estudiantes en un Array. (Considerar leer archivos CSV). Implemente el algoritmo de ordenamiento por Inserción(Iterativo-Cuadrático) para ordenar el arreglo de estudiantes por diferentes parámetros. Ejemplo: Por apellido, paterno. Descubra cuál es el tiempo que se demora en las ejecuciones. Explique cualquier otro algoritmo de ordenamiento de complejidad logarítmica. e implemente el ordenamiento utilizando los mismo parámetros anteriores. Grafique los resultados de las simulaciones realizadas considerando como unidad de medida los nanosegundos. Desde n=1 alumno hasta n=N alumnos. Luego, para el arreglo ordenado implemente el algoritmo de búsqueda binaria iterativo/recursivo y grafique los resultados de sus simulaciones.

2. EQUIPOS, MATERIALES Y TEMAS UTILIZADOS

- Sistema Operativo Windows
- OpenJDK 64-Bits 17.0.7.
- Git 2.39.2.
- Cuenta en GitHub con el correo institucional.

3. URL DE REPOSITORIO GITHUB

- URL para el Repositorio GitHub.
- <https://github.com/JuanSergioZeballos>
- URL para la practica 01 en el Repositorio GitHub.
- <https://github.com/JuanSergioZeballos/Practica01>

4. EJERCICIO

4.1. Clase Ejercicio.java

```
import java.io.*;
import java.util.Scanner;

public class Ejercicio {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Student[] students = null;

        try {
            BufferedReader reader = new BufferedReader(new FileReader("Datos.csv"));
            String line = null;
            int count = 0;

            // Contar las lineas en el archivo para determinar el tamaño del array
            while ((line = reader.readLine()) != null) {
                count++;
            }

            // Reiniciar el lector para volver al principio del archivo
            reader.close();
            reader = new BufferedReader(new FileReader("Datos.csv"));

            students = new Student[count];
            int index = 0;

            while ((line = reader.readLine()) != null) {
                String[] parts = line.split(";");
                Student estudiante = new Student(
                    parts[0], parts[1], parts[2], parts[3], parts[4],
                    Integer.parseInt(parts[5]), parts[6], parts[7]
                );
                students[index] = estudiante;
                index++;
            }

            reader.close();
        } catch (IOException e) {
            e.printStackTrace();
        }

        if (students != null) {
```

```
        for (int i = 0; i < students.length; i++) {
            System.out.println(students[i].toString());
        }

        System.out.println("=====");
        System.out.println("=====");
        System.out.println("=====");
        System.out.println("Ingrese el nombre del alumno a buscar: ");
        String nombre = sc.next();
        int pos = busquedaSecuencialNombre(students, nombre);

        if (pos != -1) {
            System.out.println(students[pos].toString());
        } else {
            System.out.println("No encontrado");
        }
    }

    // Ordenar estudiantes por edad usando el algoritmo de ordenamiento de burbuja
    ordenarPorEdad(students);

    // Imprimir estudiantes ordenados por edad
    System.out.println("Estudiantes ordenados por edad:");
    for (int i = 0; i < students.length; i++) {
        System.out.println(students[i].toString());
    }
}

public static int busquedaSecuencialNombre(Student[] students, String s) {
    for (int i = 0; i < students.length; i++) {
        if (s.equals(students[i].getName())) {
            return i;
        }
    }
    return -1;
}

public static void ordenarPorEdad(Student[] students) {
    int n = students.length;
    boolean intercambio;
    do {
        intercambio = false;
        for (int i = 0; i < n - 1; i++) {
            if (students[i].getEdad() > students[i + 1].getEdad()) {
                // Intercambiar estudiantes
                Student temp = students[i];
                students[i] = students[i + 1];
                students[i + 1] = temp;
                intercambio = true;
            }
        }
    } while (intercambio);
}
```

4.2. clase Student.java

```
// Creamos la clase Student, para guardar los datos de los estudiantes
public class Student {
    private String Name;
    private String Materno;
    private String Paterno;
    private String Gender;
    private String CUI;
    private String Status;
    private String Email;
    private String Edad;

    // Constructor
    public Student(String CUI, String Email, String Name, String Paterno, String
        Materno, String Edad, String Status, String Gender) {
        this.Name = Name;
        this.Materno = Materno;
        this.Paterno = Paterno;
        this.Gender = Gender;
        this.CUI = CUI;
        this.Status = Status;
        this.Email = Email;
        this.Edad = Edad;
    }

    // Getters
    public String getName() {
        return Name;
    }

    public String getMaterno() {
        return Materno;
    }

    public String getPaterno() {
        return Paterno;
    }

    public String getGender() {
        return Gender;
    }

    public String getCUI() {
        return CUI;
    }

    public String getStatus() {
        return Status;
    }

    public String getEmail() {
        return Email;
    }

    public String getEdad() {
        return Edad;
    }
}
```

```
// Setters
public void setName(String Name) {
    this.Name = Name;
}

public void setMaterno(String Materno) {
    this.Materno = Materno;
}

public void setPaterno(String Paterno) {
    this.Paterno = Paterno;
}

public void setGender(String Gender) {
    this.Gender = Gender;
}

public void setCUI(String CUI) {
    this.CUI = CUI;
}

public void setStatus(String Status) {
    this.Status = Status;
}

public void setEmail(String Email) {
    this.Email = Email;
}

public void setEdad(String Edad) {
    this.Edad = Edad;
}

public String toString() {
    return "Nombre: " + Name +
        "Apellido Paterno: " + Paterno +
        "Apellido Materno: " + Materno +
        "Gnero: " + Gender +
        "CUI: " + CUI +
        "Estado: " + Status +
        "Email: " + Email +
        "Edad: " + Edad;
}
}
```

5. REFERENCIAS

- M. Aedo, “Fundamentos de Programación 2 - Tópicos de Programación Orientada a Objetos”, Primera Edición, 2021, Editorial UNSA.
- <https://github.com/rescobedoq/programacion.git>
- J. Dean, “Introduction to programming with Java: A Problem Solving Approach”, Third Edition, 2021, McGraw-Hill.

- C. T. Wu, "An Introduction to Object-Oriented Programming with Java", Fifth Edition, 2010, McGraw-Hill.
- P. Deitel, "Java How to Program", Eleventh Edition, 2017, Prentice Hall.