



EXAMEN ECCAS

Facultad de ciencias de la computación
Ing. En ciencias de la computación
07/05/2021



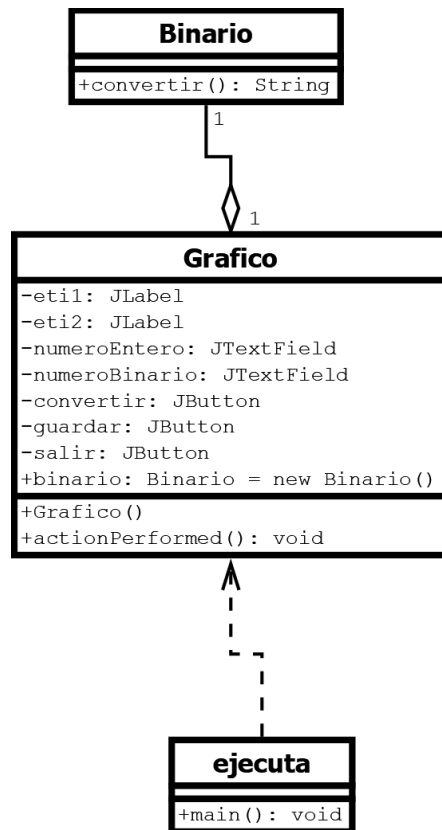
IRVYN XICALE CABRERA
201963582

Introducción

Examen para calificar nuestros conocimiento obtenidos en el semestre.

CODIGO EJERCICIO 1

DIAGRAMA



CODIGO

Ejecuta.java

```
// 201963582 irvyn xicale cabrera

public class ejecuta {
    public static void main(String[] args) {
        Grafico gui = new Grafico();
    }
}
```

Binario.java

```
// 201963582 irvyn xicale cabrera
public class Binario {
    public String convertir(int n) {
        if (n == 0) {
            return "" + n;
        } else {
            convertir(n / 2);
            System.out.print(n % 2);
            return "" + convertir(n/2) + n%2 ;
        }
    }
}
```

Grafico.java

```
import java.awt.event.*;
```

```
import java.io.FileWriter;
import java.io.IOException;

import javax.swing.*.*;

public class Grafico extends JFrame implements ActionListener {
    private JLabel eti1;
    private JLabel eti2;
    private JTextField numeroEntero;
    private JTextField numeroBinario;
    private JButton convertir;
    private JButton guardar;
    private JButton salir;

    Binario binario = new Binario();

    public Grafico() {
        this.setLayout(null);

        eti1 = new JLabel("Numero decimales:");
        eti1.setBounds(20, 20, 175, 20);

        numeroEntero = new JTextField();
        numeroEntero.setColumns(20);
        numeroEntero.setBounds(200, 20, 50, 20);

        eti2 = new JLabel("binario:");
        eti2.setBounds(20, 50, 70, 20);

        numeroBinario = new JTextField("numero en binario", 0);
        numeroBinario.setEditable(false);
    }
}
```

```
numeroBinario.setColumns(20);
numeroBinario.setBounds(70, 50, 250, 20);

convertir = new JButton("Convertir");
convertir.setBounds(20, 100, 175, 20);

guardar = new JButton("Guardar");
guardar.setBounds(20, 125, 175, 20);

salir = new JButton("salir");
salir.setBounds(20, 150, 175, 20);

this.add(eti1);
this.add(numeroEntero);
this.add(eti2);
this.add(numeroBinario);
this.add(convertir);
this.add(guardar);
this.add(salir);

convertir.addActionListener(this);
guardar.addActionListener(this);
salir.addActionListener(this);

this.setTitle("conversion de numeros");
this.setSize(600, 300);
this.setVisible(true);
this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}

@Override
```

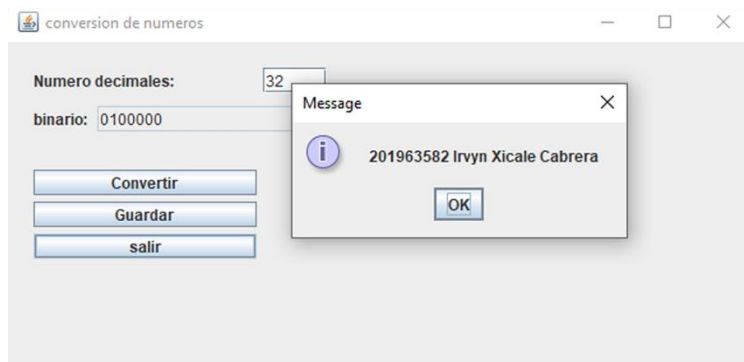
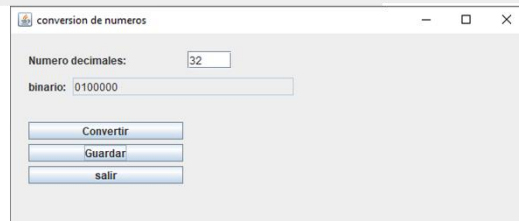
```

public void actionPerformed(ActionEvent e) {
    int datoEntero = 0;
    boolean negativo = false;
    if (e.getSource() == convertir) {
        try {
            datoEntero = Integer.parseInt(numeroEntero.getText());
        } catch (Exception error) {
            JOptionPane.showMessageDialog(null, "El capo esta vacio o tiene un valor incorrecto", "Error",JOptionPane.ERROR_MESSAGE);
        }
        if (datoEntero < 0) {
            datoEntero = datoEntero * -1;
            negativo = true;
        }

        // conversion
        String resul = binario.convertir(datoEntero);
        numeroBinario.setText(resul);
    }
    if (e.getSource() == salir) {
        JOptionPane.showMessageDialog(null, "\t 201963582 Irvyn Xicale Cabrera");
        System.exit(0);
        ;
    }
    if (e.getSource() == guardar) {
        try {
            FileWriter Registro = new FileWriter("resultado.txt", true);
            Registro.write(numeroEntero.getText() + " = " + numeroBinario.getText() + "\n");
            Registro.close();
        } catch (IOException er) {

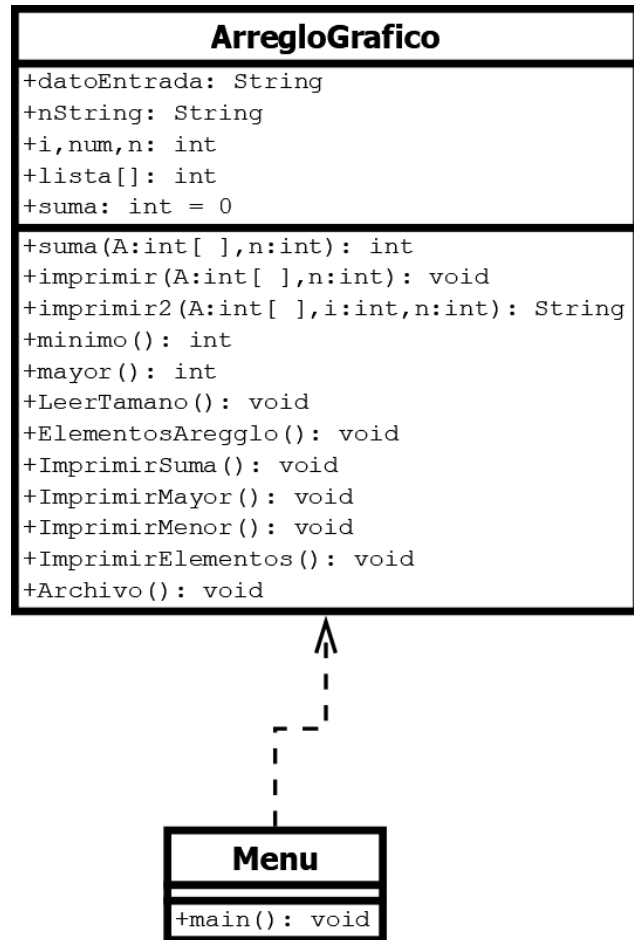
```

```
        JOptionPane.showMessageDialog(null, "no se pudo guardar el archivo \n" + er.getMes-
sage(), "Error",JOptionPane.ERROR_MESSAGE);
    }
}
}
```



CODIGO EJERCICIO 2

DIAGRAMA



CODIGO

ArregloGrafico.java

```
// Irvyn Xicale Cabrera
import java.io.*;
import javax.swing.JOptionPane;

public class ArregloGrafico {

    InputStreamReader entrada = new InputStreamReader(System.in);
    BufferedReader flujoEntrada = new BufferedReader(entrada);
    String datoEntrada;
    String nString;
    int i, num, n;
    int lista[];
    int suma = 0;

    public ArregloGrafico() {
    }

    public int suma(int[] A, int n) {
        if (n == 0) {
            return A[n];
        } else {
            return A[n] + suma(A, n - 1);
        }
    }

    public void imprimir(int[] A, int n) {
```

```

    if (n == 0) {
        System.out.println(A[n]);
    } else {
        System.out.println(A[n]);
        imprimir(A, n - 1);
    }
}

public String imprimir2(int[] A, int i, int n) {
    String cadena = "";
    if (n == 0) {
        cadena = cadena + " " + Integer.toString(A[i]);
        System.out.println(A[i]);
        return cadena;
    } else {
        System.out.println(A[i]);
        cadena = cadena + " " + Integer.toString(A[i]);
        return cadena + imprimir2(A, i + 1, n - 1);
    }
}

public int minimo(int[] A, int n, int min) {
    if (n == 0)
        return min;
    else {
        if (A[n] < min) {

            min = minimo(A, n - 1, A[n]);
        } else
            min = minimo(A, n - 1, min);
    }
}

```

```

        return min;
    }

    public int mayor(int[] A, int n, int max) {
        if (n == 0) {
            return max;
        } else {

            if (A[n] > max) {
                max = mayor(A, n - 1, A[n]);
            } else
                max = mayor(A, n - 1, max);
        }
        return max;
    }

    public void LeerTamano() {
        nString = (JOptionPane.showInputDialog(null, "Ingresa el numero de elementos para el arreglo: ",
            JOptionPane.PLAIN_MESSAGE));
        n = Integer.parseInt(nString);

        lista = new int[n];
        ArregloGrafico sum = new ArregloGrafico();
    }

    public void ElementosAregglo() {
        System.out.println("Ingresa los Valores arreglo A: ");
        for (i = 0; i < n; i++) {
            datoEntrada = JOptionPane.showInputDialog(null, "Ingresa el elemento " + (i + 1) + ":",
                JOptionPane.PLAIN_MESSAGE);
            num = Integer.parseInt(datoEntrada);
        }
    }

```

```

        System.out.println("Dato insertado: " + num);
        lista[i] = num;
    }
}

public void ImprimirSuma() {
    ArregloGrafico sum = new ArregloGrafico();
    JOptionPane.showMessageDialog(null, "El resultado de la suma es: " + sum.suma(lista, n - 1), "Resultado",
        JOptionPane.PLAIN_MESSAGE);
}

public void ImprimirMayor() {
    ArregloGrafico sum = new ArregloGrafico();
    JOptionPane.showMessageDialog(null, "\nEl numero mayor es: " + sum.mayor(lista, n - 1, lista[0]), "Resul-
tado",
        JOptionPane.PLAIN_MESSAGE);
}

public void ImprimirMenor() {
    ArregloGrafico sum = new ArregloGrafico();
    JOptionPane.showMessageDialog(null, "El numero menor es: " + sum.minimo(lista, n - 1, lista[0]), "Resul-
tado",
        JOptionPane.PLAIN_MESSAGE);
}

public void ImprimirElementos() {
    ArregloGrafico sum = new ArregloGrafico();
    JOptionPane.showMessageDialog(null, "Los numeros del arreglos son: " + sum.imprimir2(lista, 0, n - 1),
        "Resultado", JOptionPane.PLAIN_MESSAGE);
}

```

```

public void Archivo() {
    ArregloGrafico sum = new ArregloGrafico();
    try {
        FileWriter Registro = new FileWriter("Arreglo.txt");
        Registro.write("SUMA DEL ARREGLO: " + sum.suma(lista, n - 1) + "\nELEMENTO MINIMO: "
            + sum.minimo(lista, n - 1, lista[0]) + "\nELEMENTO MAXIMO: " + sum.mayor(lista, n - 1, lista[0])
            + "\nELEMENTOS DEL ARREGLO: " + sum.imprimir2(lista, 0, n - 1));
        Registro.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
}

```

Menu.java

```

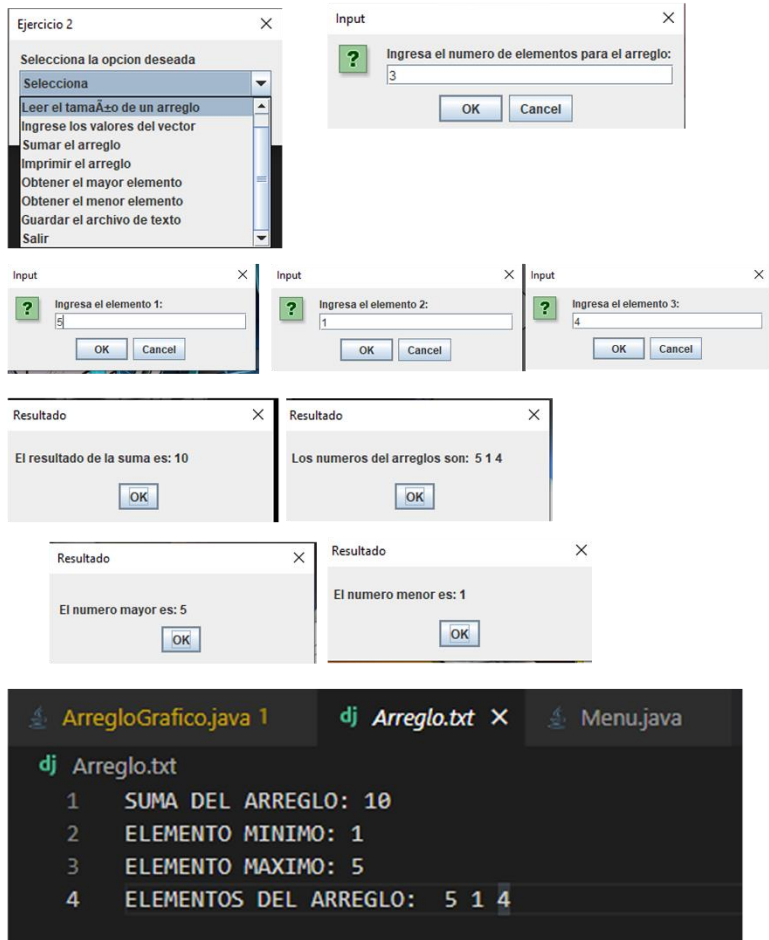
// Irvyn Xicale Cabrera
import javax.swing.JOptionPane;
public class Menu {
    public static void main(String[] args){
        ArregloGrafico Menu = new ArregloGrafico();
        String datos;
        do{
            datos = (JOptionPane.showInputDialog(null,"Selecciona la opcion deseada","Ejercicio 2",
                JOptionPane.PLAIN_MESSAGE,null,new Object[] { "Selecciona","Leer el tamaño de un arreglo", "In-
grese los valores del vector ",
                    "Sumar el arreglo ", "Imprimir el arreglo", "Obtener el mayor elemento", "Obtener el me-
nor elemento",
                    "Guardar el archivo de texto","Salir" },"Selecciona")).toString();

```

```

        switch(datos){
            case "Leer el tamaño de un arreglo":
                Menu.LeerTamano();
                break;
            case "Ingrese los valores del vector ":
                Menu.ElementosAregglo();
                break;
            case "Sumar el arreglo ":
                Menu.ImprimirSuma();
                break;
            case "Imprimir el arreglo":
                Menu.ImprimirElementos();
                break;
            case "Obtener el mayor elemento":
                Menu.ImprimirMayor();
                break;
            case "Obtener el menor elemento":
                Menu.ImprimirMenor();
                break;
            case "Guardar el archivo de texto":
                Menu.Archivo();
        }
        // JOptionPane.showMessageDialog(null, "Tu animal favorito es: "+datos, "Animales", JOptionPane.PLAIN_MESSAGE);
        } while (datos!="Salir");
    }}

```



se me cerro la página algunas veces cuando intentaba ingresar al examen

Conclusión

Esta practica nos ayudo a comprender la programación orientada a objetos

Hoja de firmas

RÚBRICA HOJA DE EVALUACIÓN DE PRÁCTICAS

MATRICULA: 201963582 FECHA: 07/04/2021

NOMBRE: Xicale Cabrera Irvyn No. PRACTICA: _____ INDIVIDUAL (x) COLABORATIVA ()

CRITERIOS	Deficiente	Suficiente	Bueno	Excelente	Calificación Obtenida	
	1-5.9	6-7.9	8-9	9.1-10	%	Puntos
CONOCIMIENTO TEÓRICO 20%	Conocimiento deficiente de los fundamentos teóricos de POO y no puede aplicarlos en el diseño de clases.	Conocimiento confuso de los fundamentos teóricos de POO y el diseño de las clases y relaciones es incompleto.	Conocimiento claro de los fundamentos teóricos POO pero requiere mejorar el modelado de las clases, sus métodos y sus relaciones entre clases.	Dominio del Conocimiento de los fundamentos teóricos POO y puede aplicarlos de forma completa en el modelado de las clases, métodos y todas las relaciones entre clase.		
EJECUCIÓN DE LA PRÁCTICA 30%	No puede realizar la práctica ya que desconoce el entorno de trabajo y desarrollo de la práctica en lenguaje UML y Java.	Realiza la práctica de forma incompleta ya que desconoce el entorno de trabajo del lenguaje UML y Java	Realiza la práctica pero requiere mejorar en el manejo del entorno de trabajo del lenguaje de programación(sintaxis y semántica)	Realiza la práctica de forma correcta y completa, demuestra dominio del entorno de trabajo del lenguaje de programación (sintaxis y semántica).		

SOLUCIÓN DE LA PRÁCTICA 40%	No puede generar las soluciones o programas a los problemas planteados ya que no posee el dominio teórico y práctico del modelado y el lenguaje de programación.	Propone soluciones confusas o programas incompletos a los problemas planteados, ya que carece del dominio del modelado y lenguaje de programación	Genera soluciones con poca profundidad y los programas no están orientados de acuerdo a los problemas solicitados, por lo cual no tiene un dominio profundo de la temática y del Lenguaje.	Genera soluciones con profundidad y los programas son correctos de acuerdo a los problemas planteados, por lo cual demuestra un dominio de la temática y del Lenguaje de Programación.		
ACTITUD DE APRENDER Y COLABORAR EN EQUIPO DE TRABAJO 10%	No posee una actitud proactiva para un aprendizaje autónomo y no le gusta participar y trabajar en equipo.	Posee una actitud propositiva para un aprendizaje autónomo, participa pero no le gusta trabajar en equipo.	Posee una actitud propositiva logrando un aprendizaje autónomo, colaborativo, le gusta trabajar en equipo pero requiere mejorar su participación y portaciones de forma profunda.	Posee una actitud proactiva logrando un aprendizaje autónomo, participa con propuestas concretas y profundas, le gusta trabajar en equipo y asume su responsabilidad dentro para lograr éxito del equipo		
Total						

Firma del Alumn@

Vo. Bo. Docente

