



---

# PRACTICA 8B

---

Facultad de ciencias de la computación  
Ing. En ciencias de la computación



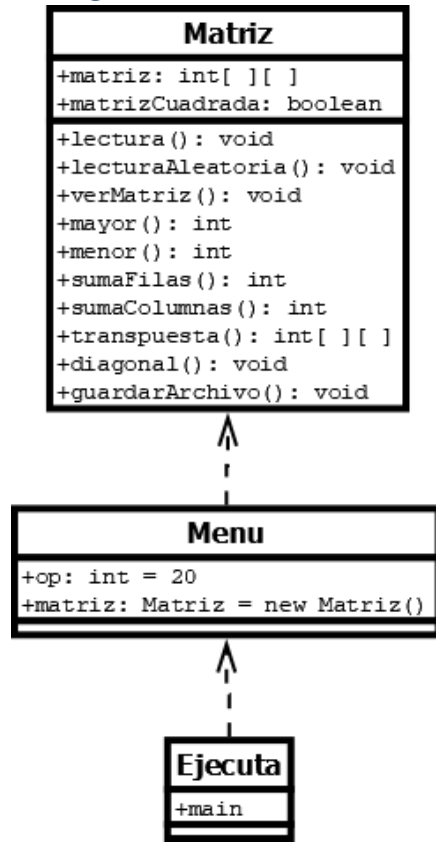
IRVYN XICALE CABRERA  
201963582

## Introducción

El objetivo de esta practica es implementar lo aprendido en el curso con las herencias clases abstractas e interfaces.

## Comida

## Diagrama ULM



## Código

Matriz.java

```
import java.io.*;
public class Matriz {
    // lectura
    InputStreamReader entrada = new InputStreamReader(System.in);
    BufferedReader flujoEntrada = new BufferedReader(entrada);
    String datoEntrada;

    // atributos
    int[][] matriz;
    boolean matrizCuadrada;
    int fila,columna;

    // metodos
    public void lectura(){
        boolean validacion = false;
        fila=0;
        columna=0;
        do{
            try{
                System.out.print("Ingresa el numero de filas de la matriz:");
                datoEntrada = flujoEntrada.readLine();
                fila=Integer.parseInt(datoEntrada);
                System.out.print("Ingresa el numero de columnas de la matriz:");
                datoEntrada = flujoEntrada.readLine();
                columna=Integer.parseInt(datoEntrada);
                if(fila==columna){
                    matrizCuadrada=true;
                }else
                    matrizCuadrada=false;
            }
        }
    }
}
```

```

        matriz= new int[fila][columna];
        // si la lectura lee todo bien validacion cambia a verdadero
        validacion = true;
        // si los datos obtenidos son menores a 0 la validacion cambia a false
        if(fila<1 && columna<1){
            validacion=false;
            System.out.println("la matriz no permite numeros menores que 0");
        }
    }catch(IOException error){
        System.err.println("Error " + error.getMessage());
    }
}while(validacion == false);
try{
    for(int i=0; i<fila; i++){
        for(int j=0; j<columna; j++){
            System.out.print("ingresa el dato [" + (i+1) + "] [" + (j+1) + "]:");
            datoEntrada = flujoEntrada.readLine();
            matriz[i][j] = Integer.parseInt(datoEntrada);
        }
    }
}catch(IOException error){
    System.err.println("Error " + error.getMessage());
}
System.out.println("");
}

public void lecturaAleatoria(){
    fila=0;
    columna=0;
    boolean validacion = false;

```

```

do{
    try{
        System.out.print("Ingresa el numero de filas de la matriz:");
        datoEntrada = flujoEntrada.readLine();
        fila = Integer.parseInt(datoEntrada);
        System.out.print("Ingresa el numero de columnas de la matriz:");
        datoEntrada = flujoEntrada.readLine();
        columna = Integer.parseInt(datoEntrada);
        matriz = new int[fila][columna];
        for(int i=0; i<fila; i++){
            for(int j=0; j<columna; j++){
                matriz[i][j] = (int)(Math.random()*100+1);
            }
        }
        if(fila == columna){
            matrizCuadrada=true;
        }else{
            matrizCuadrada = false;
            validacion=true;
        }catch(IOException error){
            System.err.println("Error " + error.getMessage());
        }
    }while(validacion==false);
    System.out.println("Matriz generada\n");
}

public void verMatriz(){
    for(int i=0;i<matriz.length;i++){
        for(int j=0;j<matriz[i].length;j++){
            System.out.print(matriz[i][j] + "\t");

```

```

    }
    System.out.println("");
}

public int[][] obtenerMatriz(){
    return matriz;
}

}

```

Menu.java

```

import java.io.*;
public class Menu {
    // lectura
    InputStreamReader entrada = new InputStreamReader(System.in);
    BufferedReader flujoEntrada = new BufferedReader(entrada);
    String datoEntrada;

    // atributos
    int op=20;
    boolean m1=false,m2=false;
    Matriz matriz1 = new Matriz();
    Matriz matriz2 = new Matriz();

    // metodos
    public void menu(){

```

```

do{
    try{
        System.out.println("\tMenu\n0.Salir\n1.Leer los elementos de una Matriz\n2. Generar aleatoria-
mente los números en la matriz\n3. Visualizar elementos de la matriz\n4. Sumar matrices\n5. Multiplicar matri-
ces\n6. Restar Matrices\n7. Verificar si la matriz es cuadrada\n8. Verificar si la matriz es triangular supe-
rior.\n9. Verificar si la matriz es identidad\n10. Obtenga el producto de la matriz por un escalar.");
        datoEntrada = flujoEntrada.readLine();
        op = Integer.parseInt(datoEntrada);
        switch(op){
            case 0:
                System.out.println("Gracias por usar este programa :)");
                break;
            case 1:
                System.out.println("");
                do{
                    System.out.println("cual matriz deseas ingresar\n1.-matriz1\n2.-matriz2");
                    datoEntrada=flujoEntrada.readLine();
                    op= Integer.parseInt(datoEntrada);
                    if (op==1){
                        matriz1.lectura();
                        m1=true;
                    }else if(op==2){
                        matriz2.lectura();
                        m2=true;
                    }

                }while (op!=1 && op!=2);
                System.out.println("");
                break;

```



```

case 2:
System.out.println("");
do{
    System.out.println("cual matriz deseas ingresar\n1.-matriz1\n2.-matriz2");
    datoEntrada=flujoEntrada.readLine();
    op= Integer.parseInt(datoEntrada);
    if (op==1){
        matriz1.lecturaAleatoria();
        m1=true;
    }else if(op==2){
        matriz2.lecturaAleatoria();
        m2=true;
    }

}while (op!=1 && op!=2);
System.out.println("");
break;

case 3:
System.out.println("");
if(m1 == true && m2 == true){
    do{
        System.out.println("cual matriz deseas ver\n1.-matriz1\n2.-matriz2");
        datoEntrada=flujoEntrada.readLine();
        op= Integer.parseInt(datoEntrada);
        if (op==1){
            matriz1.verMatriz();
            m1=true;
        }else if(op==2){
            matriz2.verMatriz();

```

```

        m2=true;
    }
    }while (op!=1 && op!=2);
}else
System.out.println("ingresa valores a las 2 matrizes");
System.out.println("");
break;

// sumar matrices
case 4:
int[][] suma;
System.out.println("");
if(m1 == true && m2 == true){
    if(matriz1.columna==matriz2.columna && matriz1.fila==matriz2.fila){
        suma = new int[matriz1.fila][matriz1.columna];
        for (int i = 0; i < matriz1.matriz.length; i++){
            for (int j = 0; i < matriz1.matriz[0].length; j++){
                suma[i][j]=matriz1.matriz[i][j] + matriz2.matriz[i][j];
            }
        }

        // imprime
        for(int i=0;i<matriz1.fila;i++){
            for(int j=0;i<matriz1.columna;j++){
                System.out.println(suma[i][j] + "  ");
            }
            System.out.println("");
        }
    }else
        System.out.println("el numero de filas o columnas no coincide entre las 2 matrices");
}
}

```

```

        System.out.println("ingresa valores a las 2 matrices");
        System.out.println("");
        break;

        // multiplicar matrices
        case 5:
            int[][] mul;
            int valor;
            System.out.println("");
            if(m1 == true && m2 == true){
                if(matriz1.columna==matriz2.fila){
                    mul = new int[matriz1.fila][matriz2.columna];
                    for(int i=0;i<matriz1.fila;i++){
                        for(int j=0;j<matriz1.columna;j++){
                            valor=0;
                            for(int k=0;k<matriz2.columna;k++){
                                valor = valor + matriz1.obtenerMatriz()[i][k] * matriz1.obtenerMa-
triz()[k][j];

                            }
                            mul[i][j]=valor;
                        }
                    }

                    // imprime
                    for(int i=0;i<matriz1.fila;i++){
                        for(int j=0;j<matriz1.columna;j++){
                            System.out.println(mul[i][j] + "    ");
                        }
                        System.out.println("");
                    }
                }else

```

```

        System.out.println("el numero de columnas de la primera matriz no coincide con las fi-
las de la segunda");
    }else
    System.out.println("ingresa valores a las 2 matrices");
    System.out.println("");
    break;

    // restar matrices
    case 6:
    int[][] resta;
    System.out.println("");
    if(m1 == true && m2 == true){
        if(matriz1.columna==matriz2.columna && matriz1.fila==matriz2.fila){
            resta = new int[matriz1.fila][matriz1.columna];
            for(int i=0;i<matriz1.fila;i++){
                for(int j=0;i<matriz1.columna;j++){
                    resta[i][j]=matriz1.obtenerMatriz()[i][j] - matriz2.obtenerMatriz()[i][j];
                }
            }

            // imprime
            for(int i=0;i<matriz1.fila;i++){
                for(int j=0;i<matriz1.columna;j++){
                    System.out.println(resta[i][j] + "   ");
                }
                System.out.println("");
            }
        }else
        System.out.println("el numero de filas o columnas no coincide entre las 2 matrices");
    }else
    System.out.println("ingresa valores a las 2 matrices");

```

```

System.out.println("");
break;

// matriz cuadrada
case 7:
System.out.println("");
if(m1 == true && m2 == true){
    do{
        System.out.println("cual matriz deseas ver si es cuadrada\n1.-matriz1\n2.-matriz2");
        datoEntrada=flujoEntrada.readLine();
        op= Integer.parseInt(datoEntrada);
        if (op==1){
            if(matriz1.matrizCuadrada==true){
                System.out.println("la matriz si es cuadrada");
            }else
                System.out.println("la matriz no es cuadrada");
        }else if(op==2){
            if(matriz2.matrizCuadrada==true){
                System.out.println("la matriz si es cuadrada");
            }else
                System.out.println("la matriz no es cuadrada");
        }
    }while (op!=1 && op!=2);
}else
    System.out.println("ingresa valores a las 2 matrices");
System.out.println("");
break;

// triangulo superior
case 8:
boolean trianguloSuperior=true;

```

```

        System.out.println("");
        if(m1 == true && m2 == true){
            do{
                System.out.println("cual matriz deseas ver si tiene triangulo superior\n1.-ma-
matriz1\n2.-matriz2");

                datoEntrada=flujoEntrada.readLine();
                op= Integer.parseInt(datoEntrada);
                if (op==1){
                    for(int i =0;i<matriz1.fila;i++){
                        for(int j=0;j<matriz1.columna;j++){
                            if(i>j){
                                if (matriz1.matriz[i][j]!=0){
                                    trianguloSuperior=false;
                                }
                            }
                        }
                    }
                    if(trianguloSuperior==true){
                        System.out.println("la matriz si tiene triangulo superior");
                    }else
                        System.out.println("la matriz no tiene triangulo superior");
                }else if(op==2){
                    for(int i =0;i<matriz2.fila;i++){
                        for(int j=0;j<matriz2.columna;j++){
                            if(i>j){
                                if (matriz2.matriz[i][j]!=0){
                                    trianguloSuperior=false;
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}

```

```

        if(trianguloSuperior==true){
            System.out.println("la matriz si tiene triangulo superior");
        }else
            System.out.println("la matriz no tiene triangulo superior");
    }
    }while (op!=1 && op!=2);
}else
    System.out.println("ingresa valores a las 2 matrices");
    System.out.println("");
    break;

// identidad
case 9:
    boolean identidad = true;
    System.out.println("");
    if(m1 == true && m2 == true){
        do{
            System.out.println("cual matriz deseas ver si es identidad\n1.-matriz1\n2.-ma-
matriz2");

            datoEntrada=flujoEntrada.readLine();
            op= Integer.parseInt(datoEntrada);
            if (op==1){
                if(matriz1.matrizCuadrada==true){
                    for (int i=0;i<matriz1.fila;i++){
                        for (int j=0;j<matriz1.fila;j++){
                            if(i==j){
                                if(matriz1.matriz[i][j]!=1){
                                    identidad=false;
                                }
                            }else{
                                if(matriz1.matriz[i][j]!=0){

```

```

                                identidad=false;
                                }
                                }
                        }
                }
                }else
                identidad=false;
        }else if(op==2){
                if(matriz2.matrizCuadrada==true){
                        for (int i=0;i<matriz2.fila;i++){
                                for (int j=0;j<matriz2.fila;j++){
                                        if(i==j){
                                                if(matriz2.matriz[i][j]!=1){
                                                        identidad=false;
                                                }
                                                }else{
                                                        if(matriz2.matriz[i][j]!=0){
                                                                identidad=false;
                                                        }
                                                }
                                }
                        }
                }
                }else
                identidad=false;
        }
    }while (op!=1 && op!=2);
    if(identidad==true){
        System.out.println("es identidad");
    }else
        System.out.println("es identidad");
}

```



```

        System.out.println("");
        break;

        // matriz por escalar
        case 10:
            int escalar;
            int[][] escal;
            System.out.println("");
            if(m1 == true && m2 == true){
                do{
                    System.out.println("cual matriz deseas multiplicarle el escalar\n1.-matriz1\n2.-ma-
matriz2");

                    datoEntrada=flujoEntrada.readLine();
                    op= Integer.parseInt(datoEntrada);
                    if (op==1){
                        escal = new int[matriz1.fila][matriz2.columna];
                        System.out.println("ingresa el escalar");
                        datoEntrada = flujoEntrada.readLine();
                        escalar = Integer.parseInt(datoEntrada);
                        for (int i=0;i<matriz1.fila;i++){
                            for (int j=0;j<matriz1.columna;j++){
                                escal[i][j]=matriz1.matriz[i][j]*escalar;
                            }
                        }

                        // imprime
                        for(int i=0;i<matriz1.fila;i++){
                            for(int j=0;i<matriz1.columna;j++){
                                System.out.println(escal[i][j] + "   ");
                            }
                        }
                        System.out.println("");
                    }
                }
            }

```

```

    }
    }else if(op==2){
        escal = new int[matriz1.fila][matriz2.columna];
        System.out.println("ingresa el escalar");
        datoEntrada = flujoEntrada.readLine();
        escalar = Integer.parseInt(datoEntrada);
        for (int i=0;i<matriz2.fila;i++){
            for (int j=0;j<matriz2.fila;j++){
                escal[i][j]=matriz2.matriz[i][j]*escalar;
            }
        }

        // imprime
        for(int i=0;i<matriz2.fila;i++){
            for(int j=0;i<matriz2.columna;j++){
                System.out.println(escal[i][j] + " ");
            }
            System.out.println("");
        }
    }
    }while (op!=1 && op!=2);
    System.out.println("");
}

break;

default:
    System.out.println("Opcion invalida");
    op = 20;
}
}catch(IOException error)

```

```
        {  
            System.err.println("Error " + error.getMessage());  
        }  
    }while(op!=0);  
}  
}
```

Ejecuta.java

```
public class Ejecuta {  
    public static void main(String[] args){  
        Menu menu = new Menu();  
        menu.menu();  
    }  
}
```

## Conclusión

Esta practica nos ayudo a comprender el funcionamiento de las matrices.

## Hoja de firmas

### RÚBRICA HOJA DE EVALUACIÓN DE PRÁCTICAS

MATRICULA: 201963582 FECHA: 24/01/2021

NOMBRE: Xicale Cabrera Irvyn No. PRACTICA: 1 INDIVIDUAL ( x ) COLABORATIVA ( )

| CRITERIOS                              | Deficiente   | Suficiente  | Bueno  | Excelente   | Calificación Obtenida |        |
|--|--|---|--|---|-----------------------|--------|
|  | 1-5.9  | 6-7.9   | 8-9  | 9.1-10  | %                     | Puntos |
| <b>CONOCIMIENTO TEÓRICO</b><br>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. |                       |        |
| <b>EJECUCIÓN DE LA PRÁCTICA</b><br>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).                         |                       |        |

|   |  |   |  |  |  |  |
|---|--|---|--|--|--|--|
| <b>SOLUCIÓN DE LA PRÁCTICA</b><br><br><b>40%</b>                              | 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.                   |  |  |
| <b>ACTITUD DE APRENDER Y COLABORAR EN EQUIPO DE TRABAJO</b><br><br><b>10%</b> | 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 |  |  |
| <b>Total</b>  |  |   |  |  |  |  |

Firma del Alumn@

Vo. Bo. Docente

