

## ACTIVIDAD No. 4:

### Diseño de Interfaz de Usuario

|                              |
|------------------------------|
| Estudiante                   |
| <b>MATEO RODRIGUEZ LOPEZ</b> |
| Programa académico           |
| Ingeniería de Software       |

|                              |
|------------------------------|
| Asignatura                   |
| Algoritmos y Programación II |

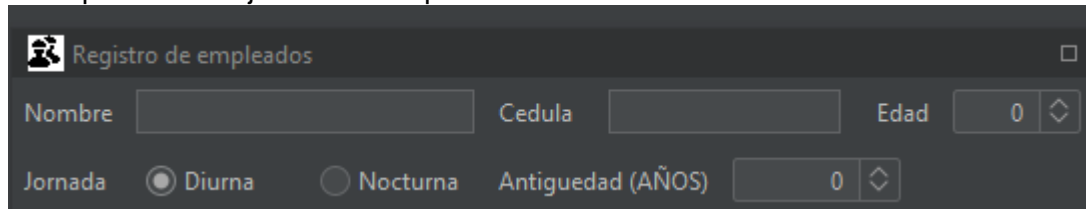
### ETAPA DE TRANSFERENCIA

En esta etapa se abordarán uno de los conceptos más relevantes y de suma importancia en la programación, como lo es el diseño avanzado de interfaces de usuario, el manejo de formularios, de gráficos y la creación de clases base para procedimientos de impresión, del cual se requiere que el estudiante tenga dominio en el momento de aplicar estos fundamentos, para que los pueda poner en práctica, cuando realice un programas haciendo uso del manejo de interfaces, gráficos, clases base e interfaces con mejores diseños.

### ACTIVIDAD

1. En esa actividad el estudiante debe diseñar y codificar un programa para el registro de inventario de los productos que se han vendido en la tienda de Compensar, para llevar este historial se deben agregar los datos en el componente de jTable. Además del registro de los productos, también de requiere de otra interfaz para el registro de los empleados que se tienen en la tienda.

Los datos que se le solicitará para el **registro de empleados** que se mostrará por medio del jTable será: El nombre del empleado, número de identificación, edad, jornada en la que trabaja (diurno o nocturno), tiempo que lleva trabajando en Compensar (en años) y según el tiempo mostrarle los descuentos o beneficios que tiene por ser trabajador de Compensar.



The screenshot shows a Java Swing window titled "Registro de empleados" with a standard Mac OS X title bar (red, yellow, and green buttons). The window contains a form with the following fields and controls:

- Nombre:** A text input field.
- Cedula:** A text input field.
- Edad:** A text input field containing the value "0", followed by a spin button (up and down arrows).
- Jornada:** Two radio buttons labeled "Diurna" (which is selected) and "Nocturna".
- Antigüedad (AÑOS):** A text input field containing the value "0", followed by a spin button (up and down arrows).

**Registro de empleados**

Nombre  Cedula  Edad

Jornada ☐ Diurna ☒ Nocturna Antigüedad (AÑO...)

| Nombre            | Id         | Edad    | Jornada  | Antigue... | Dcto Tie... | Dcto Re... |
|-------------------|------------|---------|----------|------------|-------------|------------|
| cristian pava     | 363124545  | 35 Años | Nocturna | 23 Año(s)  | 50%         | 60%        |
| manuel pontilla   | 48815123   | 45 Años | Diurna   | 3 Año(s)   | 30%         | 30%        |
| andrea torres ... | 1083900116 | 29 Años | Diurna   | 6 Año(s)   | 50%         | 60%        |

Los datos que se solicitarán para el **registro del inventario** serán: nombre del producto, tipo de producto (Aseo, papelería, víveres, producto para mascotas, otros), número de unidades, valor unitario. Además de estos datos que se deben mostrar en el jTable, se debe mostrar el valor del IVA y el valor total, estos dos últimos datos se deben calcular según el tipo de producto

**Registro de inventarios**

Nombre  Vlr Unitario  Unidades

Tipo

```
this.jTable2.setModel(modeloTablaInventarios);
modeloTablaInventarios.addColumn("Nombre");
modeloTablaInventarios.addColumn("Tipo");
modeloTablaInventarios.addColumn("Unidades");
modeloTablaInventarios.addColumn("Valor Un");
modeloTablaInventarios.addColumn("IVA");
modeloTablaInventarios.addColumn("Total");
```

**Registro de inventarios**

Nombre  Vlr Unitario  Unidades

Tipo

| Nombre          | Tipo      | Unidades | Valor Un | IVA  | Total     |
|-----------------|-----------|----------|----------|------|-----------|
| comida perro    | Mascotas  | 50000    | 110      | 16.0 | 4620000.0 |
| arena gato      | Mascotas  | 35000    | 10       | 16.0 | 294000.0  |
| cepillo dientes | Aseo      | 750      | 100      | 19.0 | 60750.0   |
| jabon cuerpo    | Aseo      | 100      | 100      | 19.0 | 8100.0    |
| sacapunta       | Papeleria | 22       | 33       | 9.0  | 660.66    |
| borrador        | Papeleria | 22       | 55       | 9.0  | 1101.1    |
| lapicero        | Papeleria | 500      | 2        | 9.0  | 910.0     |
| azucar          | Viveres   | 500      | 201      | 15.0 | 85425.0   |
| pasta           | Viveres   | 1800     | 201      | 15.0 | 307530.0  |
| arroz           | Viveres   | 1500     | 201      | 15.0 | 256275.0  |



```
if (checkInvEmpty() == false) {  
    Inventario inv = new Inventario(txtInvNombre.getText(), this.comboInvCat  
        Integer.parseInt(this.spinnerInvValor.getValue().toString()),  
        Integer.parseInt(this.spinnerInvUnds.getValue().toString()));  
  
    switch (this.comboInvCategoria.getSelectedIndex()) {  
        case 0:  
            this.contAseo += inv.GetUNDS();  
            break;  
        case 1:  
            this.contPapeleria += inv.GetUNDS();  
            break;  
        case 2:  
            this.contViveres += inv.GetUNDS();  
            break;  
        case 3:  
            this.conMascotas += inv.GetUNDS();  
            break;  
        case 5:  
            this.conOtros += inv.GetUNDS();  
            break;  
        default:  
            break;  
    }  
  
    this.modeloTablaInventarios.insertRow(0, new Object[]{  
        inv.GetNombre(),  
        inv.GetTipo(),  
        inv.GetVlrUNI(),  
        inv.GetUNDS(),  
        inv.CalcularIVA(),  
        inv.CalcularTotal()  
    });  
}
```

A continuación, se tienen las siguientes **condiciones** para el **registro de empleados**:

- Si el empleado lleva trabajando menos de un año, los beneficios que tiene son: 15% de descuento por compras en la tienda de Compensar y 20% en centros recreacionales.
- Si el empleado lleva trabajando entre 1 y 5 años, los beneficios que se les da son: 30% de descuento por compras en la tienda de Compensar y 30% en centros recreacionales.
- Si el empleado lleva trabajando más de 6 años, los beneficios que tiene son: 50% de descuento por compras en la tienda de Compensar y 60% en centros recreacionales de Compensar.



```

public void CalcularDescuentos() {
    if (antiguedad < 1) {
        dctoTienda = 15;
        dctoRecreacion = 20;
    } else if (antiguedad >= 1 && antiguedad < 6) {
        dctoTienda = 30;
        dctoRecreacion = 30;
    } else if (antiguedad >= 6) {
        dctoTienda = 50;
        dctoRecreacion = 60;
    }
}
}

```

|    | Antigue... | Dcto Tie... | Dcto Re... |
|----|------------|-------------|------------|
| na | 23 Año(s)  | 50%         | 60%        |
|    | 3 Año(s)   | 30%         | 30%        |
|    | 6 Año(s)   | 50%         | 60%        |

A continuación, se tienen las siguientes **condiciones** para el **registro del inventario**:

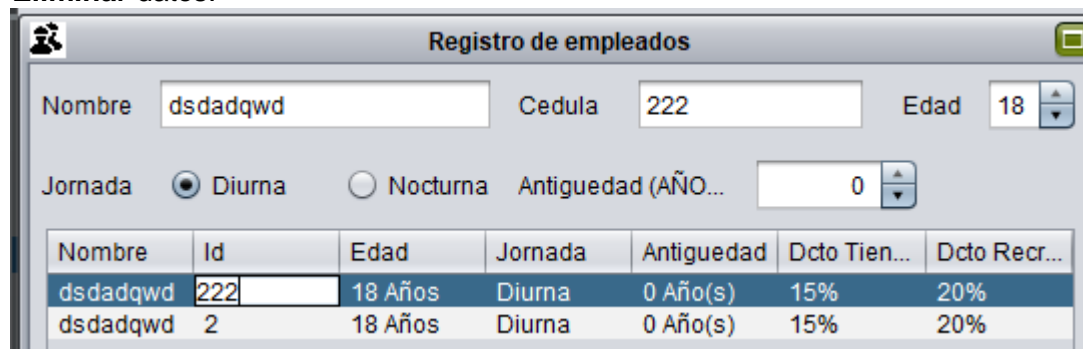
- Para aplicar el valor del IVA se debe tener que tipo de producto es, si el producto es de Aseo, se aplica un 19%, si es de papelería, se aplica el 9%, si son víveres, se aplica el 15%, si son productos para mascotas, se aplica el 16% y si son otros, se le aplica en 10%.

| Valor Un | IVA  | Total     |
|----------|------|-----------|
| 110      | 16.0 | 4620000.0 |
| 10       | 16.0 | 294000.0  |
| 100      | 19.0 | 60750.0   |
| 100      | 19.0 | 8100.0    |
| 33       | 9.0  | 660.66    |
| 55       | 9.0  | 1101.1    |
| 2        | 9.0  | 910.0     |
| 201      | 15.0 | 85425.0   |
| 201      | 15.0 | 307530.0  |
| 201      | 15.0 | 256275.0  |

```
public double CalcularIVA() {
    int valorIva = 0;

    switch (this.tipoProducto) {
        case 0:
            valorIva = 19;
            break;
        case 1:
            valorIva = 9;
            break;
        case 2:
            valorIva = 15;
            break;
        case 3:
            valorIva = 16;
            break;
        case 5:
            valorIva = 10;
            break;
        default:
            valorIva = 0;
            break;
    }
}
```

Para seleccionar la creación de los empleados y de los productos se deben crear por cada uno, una clase base, donde tendremos los atributos correspondientes, los métodos con parámetros para recibir los datos desde la interfaz y nuestros métodos Getter para enviar y retornar los valores y resultados según las condiciones, para guardarlos en el jTable, de igual manera se deben tener las opciones **de Editar y Eliminar** datos.



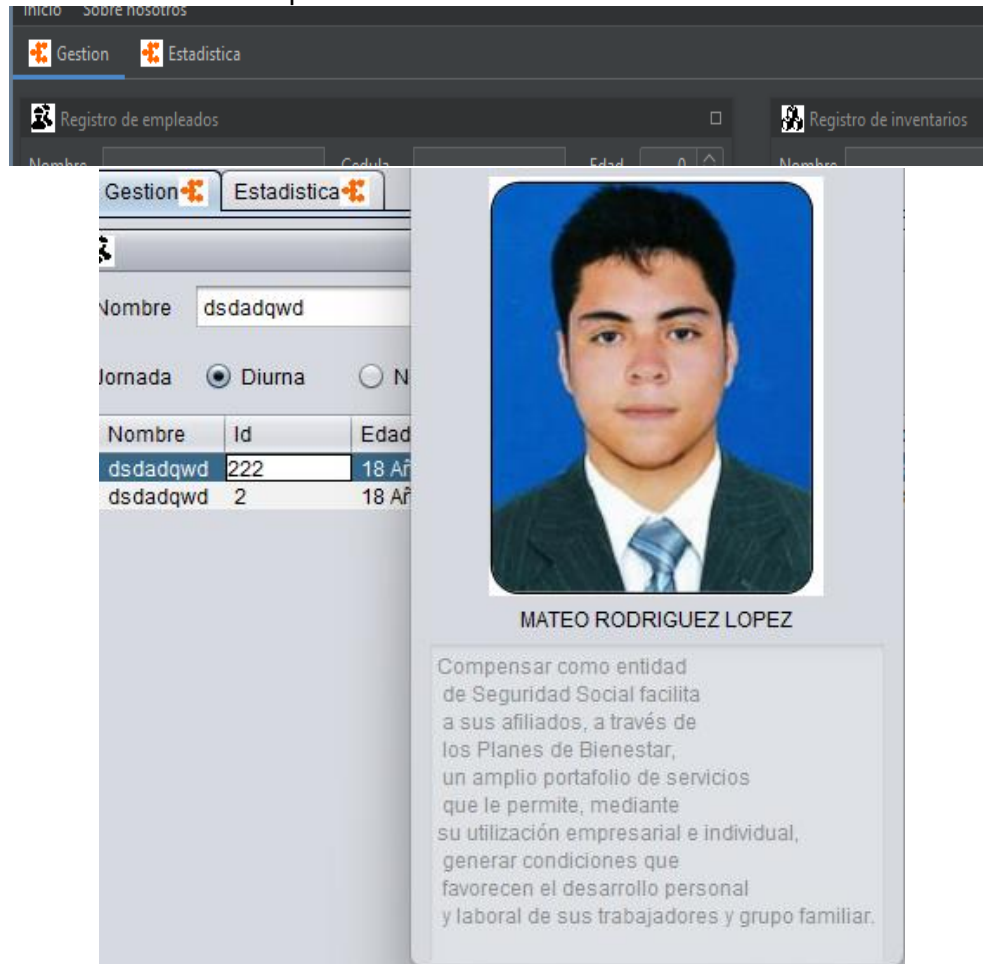
| Nombre   | Id  | Edad    | Jornada | Antigüedad | Dcto Tien... | Dcto Recr... |
|----------|-----|---------|---------|------------|--------------|--------------|
| dsdadqwd | 222 | 18 Años | Diurna  | 0 Año(s)   | 15%          | 20%          |
| dsdadqwd | 2   | 18 Años | Diurna  | 0 Año(s)   | 15%          | 20%          |

Eliminar Registro

Limpiar tabla

```
private void btnEmpLimpiarActionPerformed(java.awt.event.ActionEvent evt) {
    DefaultTableModel dtm = (DefaultTableModel) jTable2.getModel();
    dtm.setRowCount(0);
    this.contAseo = 0;
    this.contPapeleria = 0;
    this.contViveres = 0;
    this.conMascotas = 0;
    this.conOtros = 0;
}
```

Cada interfaz debe tener el icono de la universidad y para la interfaz principal se deben manejar el **Desktop Pane**, junto con los componentes **de menús (Menu Bar, menú, Menu Item)**, y un menú que se llamara sobre nosotros y colocaran información relevante sobre la persona realizó la actividad, una foto del estudiante, e información sobre Compensar.



**Como resultado del ejercicio de deben guardar siguientes datos en los Jtable**

- Para la **interfaz de empleados** se debe mostrar en la tabla: nombre del empleado, número de identificación, edad, jornada, tiempo que lleva trabajando en Compensar y los beneficios que tiene

```

public class Empleado {

    private String identificacion;
    private String nombre;
    private int edad;
    private boolean jornada;
    private int antiguedad;
    private int dctoTienda;
    private int dctoRecreacion;

    public Empleado(String nombre, int edad, String identificacion, boolean jornada, int antiguedad) {

        this.identificacion = identificacion;
        this.nombre = nombre;
        this.edad = edad;
        this.jornada = jornada;
        this.antiguedad = antiguedad;
        CalcularDescuentos();
    }
}

```

Registro de empleados

Nombre  Cedula  Edad

Jornada ☐ Diurna ☒ Nocturna Antigüedad (AÑO...)

| Nombre            | Id         | Edad    | Jornada  | Antigüedad | Dcto Tie... | Dcto Re... |
|-------------------|------------|---------|----------|------------|-------------|------------|
| cristian pava     | 363124545  | 35 Años | Nocturna | 23 Año(s)  | 50%         | 60%        |
| manuel pontilla   | 48815123   | 45 Años | Diurna   | 3 Año(s)   | 30%         | 30%        |
| andrea torres ... | 1083900116 | 29 Años | Diurna   | 6 Año(s)   | 50%         | 60%        |

Registrar empleado Eliminar Registro Limpiar tabla

- Para la **interfaz de inventario** se debe mostrar en la tabla: nombre del producto, tipo de producto, número de unidades, valor unitario, valor del IVA y el valor total.

```

public class Inventario {

    private String nombreProducto;
    private int tipoProducto;
    private int valorUnitario;
    private int nroUnidades;
    private int vlcIVA;

    public Inventario(String nombreProducto, int tipoProducto, int valorUnitario, int nroUnidades) {

        this.nombreProducto = nombreProducto;
        this.tipoProducto = tipoProducto;
        this.valorUnitario = valorUnitario;
        this.nroUnidades = nroUnidades;
    }
}

```



Registro de inventarios

Nombre  Vir Unitario  Unidades

Tipo

| Nombre          | Tipo      | Unidades | Valor Un | IVA  | Total     |
|-----------------|-----------|----------|----------|------|-----------|
| comida perro    | Mascotas  | 50000    | 110      | 16.0 | 4620000.0 |
| arena gato      | Mascotas  | 35000    | 10       | 16.0 | 294000.0  |
| cepillo dientes | Aseo      | 750      | 100      | 19.0 | 60750.0   |
| jabon cuerpo    | Aseo      | 100      | 100      | 19.0 | 8100.0    |
| sacapunta       | Papeleria | 22       | 33       | 9.0  | 660.66    |
| borrador        | Papeleria | 22       | 55       | 9.0  | 1101.1    |
| lapicero        | Papeleria | 500      | 2        | 9.0  | 910.0     |
| azucar          | Viveres   | 500      | 201      | 15.0 | 85425.0   |
| pasta           | Viveres   | 1800     | 201      | 15.0 | 307530.0  |
| arroz           | Viveres   | 1500     | 201      | 15.0 | 256275.0  |

Registrar inventario Eliminar Registro Limpiar tabla

Así mismo, se deberá mostrar una **grafico** de pastel para los datos de **empleados**, en el cual represente el número de empleados que esta en la **jornada diurna y nocturna**. Por parte de **inventario de productos**, el **grafico** debe representar los productos vendidos por cada uno de los **tipos de productos (Aseo, papelería, víveres, producto para mascotas, otros)**.





```
public void paint(Graphics g) {
    try {
        super.paint(g);
        if (banderaInventario == true) {

            int total = contAseo + contPapeleria + contViveres + conMascotas + conOtros;
            int gradosAseo = contAseo * 360 / total;
            int gradosPapeleria = contPapeleria * 360 / total;
            int gradosViveres = contViveres * 360 / total;
            int gradosMascotas = conMascotas * 360 / total;
            int gradosOtros = conOtros * 360 / total;

            g.setColor(Color.PINK);
            g.fillArc(125, 180, 200, 200, 0, gradosAseo);
            g.fillRect(380, 135, 20, 20);
            g.drawString("Aseo " + (contAseo * 100) / total + "%", 415, 150);
            g.setColor(Color.GREEN);
            g.fillArc(125, 180, 200, 200, gradosAseo, gradosPapeleria);
            g.fillRect(380, 165, 20, 20);
            g.drawString("Papeleria " + (contPapeleria * 100) / total + "%", 415, 180);
            g.setColor(Color.ORANGE);
            g.fillArc(125, 180, 200, 200, gradosAseo + gradosPapeleria, gradosViveres);
            g.fillRect(380, 195, 20, 20);
            g.drawString("Viveres " + (contViveres * 100) / total + "%", 415, 210);
            g.setColor(Color.BLUE);
            g.fillArc(125, 180, 200, 200, gradosAseo + gradosPapeleria + gradosViveres, gradosMascotas);
            g.fillRect(380, 225, 20, 20);
            g.drawString("Mascotas " + (conMascotas * 100) / total + "%", 415, 240);
            g.setColor(Color.CYAN);
            g.fillArc(125, 180, 200, 200, gradosAseo + gradosPapeleria + gradosViveres + gradosMascotas, gradosOtros);
            g.fillRect(380, 255, 20, 20);
            g.drawString("Otros " + (conOtros * 100) / total + "%", 415, 270);

        } else if (banderaJornada == true) {

            int total = contDiurna + contNocturna;
            int gradosDiurna = contDiurna * 360 / total;
            int gradosNocturna = contNocturna * 360 / total;
            g.setColor(Color.YELLOW);
            g.fillArc(125, 180, 200, 200, 0, gradosDiurna);
            g.fillRect(380, 135, 20, 20);
            g.drawString("Diurna " + (contDiurna * 100) / total + "%", 415, 150);
            g.setColor(Color.BLUE);
            g.fillArc(125, 180, 200, 200, gradosDiurna, gradosNocturna);
            g.fillRect(380, 165, 20, 20);
            g.drawString("Nocturna " + (contNocturna * 100) / total + "%", 415, 180);

        }
    } catch (Exception e) {
```

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

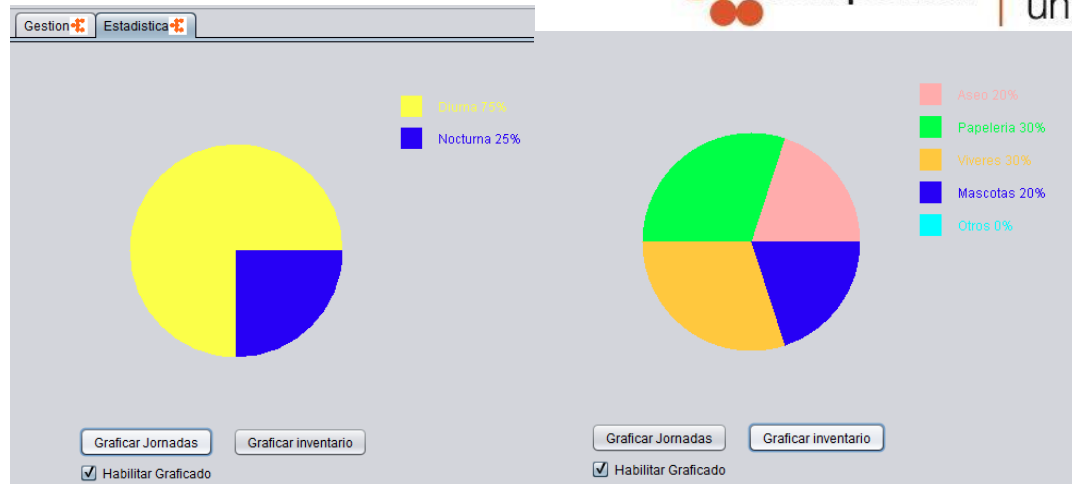
    datearGraficoJornadas(this.contDiurna, this.contNocturna);
    banderaInventario = false;
    banderaJornada = true;
    repaint();

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

    datearGraficoInventario(this.contAseo, this.contPapeleria, this.contViveres, this.conMascotas, this.conOtros);
    banderaInventario = true;
    banderaJornada = false;
    repaint();

}
```



Para esta actividad se debe saber manejar interfaces, la creación de gráficos, tablas, clases base, aplicación de métodos para solicitar los datos, otro para guardar los datos; hacer uso de vectores y matrices, manejo de excepciones y aplicación de estructuras condicionales y cíclicas, creación de gráficos, clases base y tablas.

### Trabajo individual

Esta actividad le permitirá al estudiante a poner en práctica los elementos que se vieron en la Unidad 3, para que pueda realizar un análisis de las diferentes problemáticas, haciendo uso de las buenas prácticas de programación como lo son: el manejo de formularios, agregar gráficos y la creación de clases base. A continuación, el estudiante deberá entregar:

1. Un documento donde se evidencie el diseño y codificación de la solución propuesta para el ejercicio, tanto la interfaz como el código realizado para toda la aplicación, con la explicación de cada bloque, método de código realizado; se debe realizar la captura de pantalla del código realizado, de cada una de las interfaces diseñadas y los resultados de los datos de salida de cada una de ellas, del ejercicio propuesto.

#### Tener en cuenta para el desarrollo de la actividad:

1. Realizar una análisis y proceso de forma individual de las temáticas abarcadas en esta actividad, como el manejo de formularios, agregar gráficos y la creación de clases base.
2. Leer la unidad 3: Diseño de Interfaz de Usuario
3. Subir las evidencias del trabajo autónomo en la plataforma LMS del curso
4. Resolver el cuestionario formativo de la tercera Unidad

**Nota:** Para este trabajo, el estudiante debe realizarlo de forma autónoma e individual.

## Desarrollo

1. Se anexan pantallazos de referencia del código y del funcionamiento del programa
2. El código se encuentra en el siguiente repositorio de GitHub para su revisión detallado

<https://github.com/THEMATEO135/ActividadAP2--final>

[Descargar el código aquí](#)

3. El binario ejecutable se puede descargar del siguiente enlace para su respectivo Testing

<https://github.com/THEMATEO135/ActividadAP2--final/releases>