

**Benemérita Universidad Autónoma  
de Puebla.**

**Facultad en Ciencias de la  
Computación.**

**Licenciatura en Ciencias de la  
Computación.**

**Irvyn Xicale Cabrera**

**201963582**

**Practica No. 12**

**Excepciones**

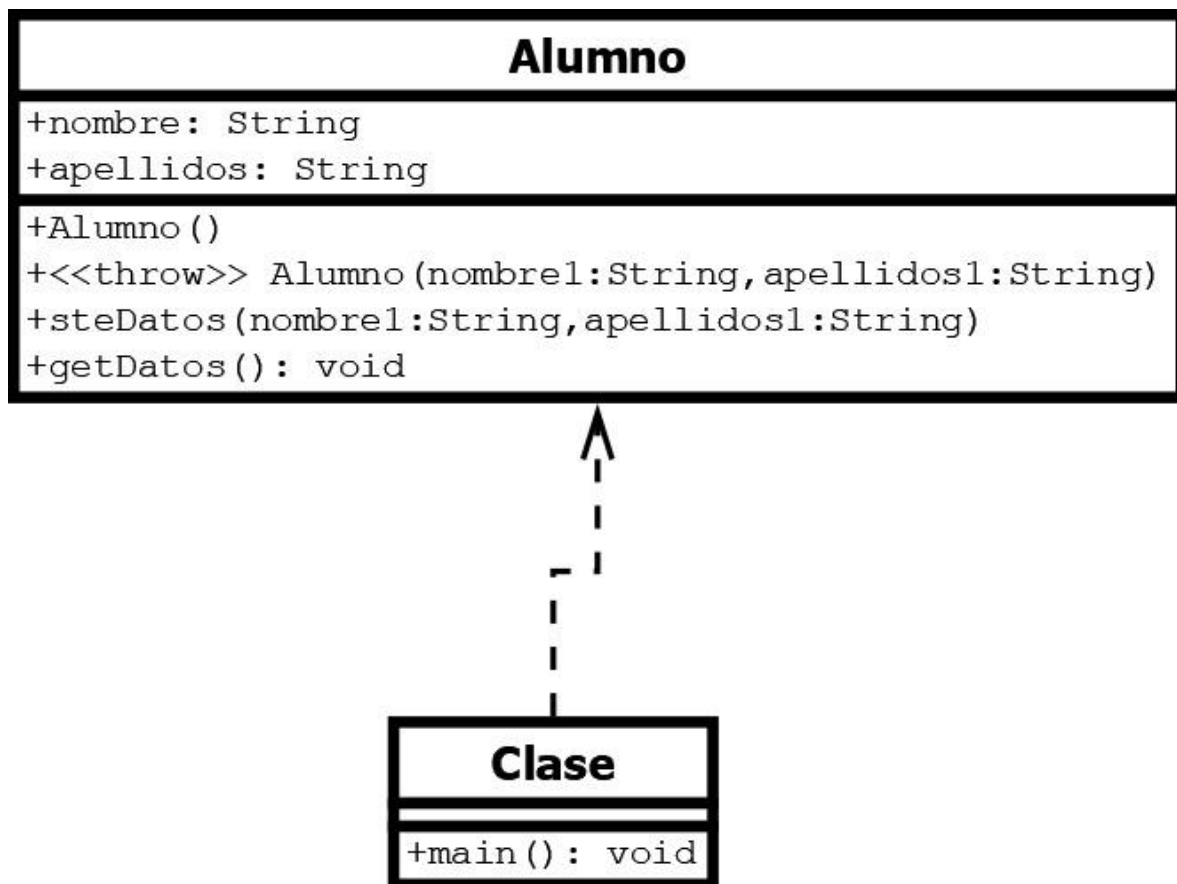
**Docente: Carmen Cerón Garnica**

## INTRODUCCIÓN.

Las excepciones son una ocurrencia de una situación indeseable que puede ser detectada durante la ejecución de un programa. Por ejemplo, una división entre cero, leer datos no válidos, tratar de abrir un archivo que no existe es una excepción, también acceder a un elemento de un arreglo fuera de su índice permitido.

## PROGRAMA ALUMNO

### DIAGRAMA UML



## CÓDIGO DEL PROGRAMA

### PROGRAMA: ALUMNO

```
public class Alumno
{
    String nombre;
```

```

String apellidos;

public Alumno()
{
    nombre=null;
    apellidos=null;
}

public Alumno(String nombre1, String apellidos1) throws Exception
{
    if(nombre1 == null || apellidos1 == null || "".equals(nombre1) ||
"".equals(apellidos1))
        throw new Exception("Argumentos no válidos");
    //si el constructor no tiene los dos atributos lanza la excepción el objeto
    //no se crea el objeto

    else
        System.out.println("Se construyó el objeto");
        nombre=nombre1;
        apellidos=apellidos1;
}

public void setDatos(String nombre1, String apellidos1)
{
    nombre=nombre1;
    apellidos=apellidos1;
}

public void getDatos()
{
    System.out.println ("=====Los datos del objeto =====");
    System.out.println ("Nombre: " +nombre);
    System.out.println ("Apellido: " +apellidos);
}
}

```

## PROGRAMA: EJECUTAALUMNO

```
import java.io.*;

public class EjecutaAlumno
{
    public static void main(String args[])
    {
        InputStreamReader entrada = new InputStreamReader(System.in);
        BufferedReader flujoEntrada = new BufferedReader(entrada);
        String varnombre;
        String varapellido;

        try
        {
            System.out.println("Ingrese el nombre: ");
            varnombre = flujoEntrada.readLine();
            System.out.println("Ingrese el apellido: ");
            varapellido = flujoEntrada.readLine();
            Alumno alum= new Alumno(varnombre,varapellido);
            alum.getDatos();
        }
        catch(Exception e)
        {
            System.out.println("No se pudo crear el objeto y llama el mensaje de la
Exception:"+ e.getMessage());
        }
        finally
        {
            System.out.println("Instrucciones a ejecutar finalmente tanto si se producen
anomalias como si no");
        }
    }
}
```

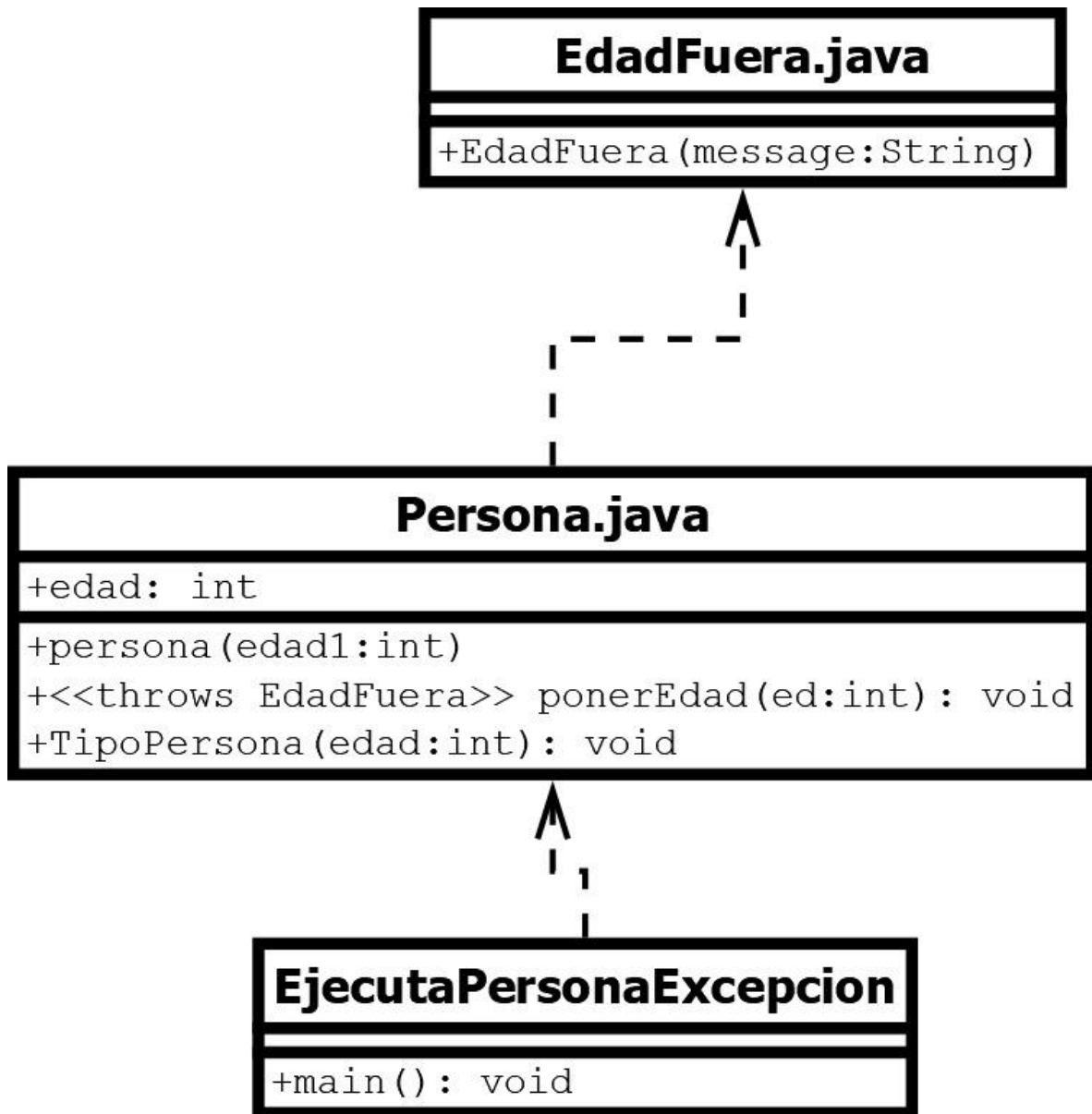
}

## CORRIDA DEL PROGRAMA

```
Ingrese el nombre:  
Ingrese el apellido:  
No se pudo crear el objeto y llama el mensaje de la Exception:Argumentos no validos  
Instrucciones a ejecutar finalmente tanto si se producen anomalias como si no  
PS C:\Users\irvyn\OneDrive\Pictures\paginaweb\practica12>
```

## PROGRAMA EDADFUERA

## DIAGRAMA UML



## PROGRAMA: PERSONA

```
public class Persona {
    int edad;
    public Persona()
    {
    }
}
```

```

public Persona(int edad1)
{
    edad=edad1;
}

public void ponerEdad (int ed) throws EdadFuera
{
    if ((ed < 0) || (ed > 130))
        throw new EdadFuera("Demasiado joven o demasiado viejo");
    else
        edad = ed;
}

public void TipoPersona(int edad)
{
    if (edad<=12)
        System.out.println("Eres un niño, juega y diviértete mucho");
    if ((edad>12) && (edad<18))
        System.out.println("Eres adolescente, come frutas y verduras");
    if ((edad>=18) && (edad<30))
        System.out.println("Eres Joven, sigue con tus sueños y proyectos ");
    if ((edad>=30) && (edad<60))
        System.out.println("Eres Joven Adulto productivo, sigue adelante, tu puedes");
    if ((edad>=60) && (edad<80))
        System.out.println("Eres Adulto Mayor, te has convertido en una gran persona");
    if ((edad>=80) && (edad<130))
        System.out.println("Eres Anciano, dejaste un gran legado");
}
}

```

## PROGRAMA: EDADFUERA

```

public class EdadFuera extends java.lang.Exception {
    /**
     * Creates a new instance of <code>EdadFuera</code>.
     */
}

```

```

public EdadFuera()
{
}

/**
 * Constructs an instance of <code>EdadFuera</code> with the specified
 * detail message.
 *
 * @param message.
 */
public EdadFuera(String message) {
    super(message);
}
}

```

## PROGRAMA: EJECUTAPERSONAEXCEPCION

```

import java.util.*;
import java.io.*;
public class EjecutaPersonaExcepcion
{
    public static void main(String args[])throws IOException
    {
        InputStreamReader entrada = new InputStreamReader(System.in);
        BufferedReader flujoEntrada = new BufferedReader(entrada);
        String datoEntrada;
        int varedad=0;

        Persona persona1= new Persona();
        try
        {

```



```

        System.out.println("Dame la edad de la persona");
        datoEntrada = flujoEntrada.readLine();
        varedad = Integer.parseInt(datoEntrada);
        persona1.ponerEdad(varedad);
        persona1.TipoPersona(varedad);
    }
    catch (NumberFormatException e1 ) //Excepción sólo para ese tipo
de error
    {
        System.out.println("Formato no valido deben ser numeros "+
e1.getMessage());
        e1.printStackTrace();// seguimiento del error
    }
    catch( EdadFuera e2)
    {
        System.out.println("No es valida la edad:"+ e2.getMessage());
    }
    finally
    {
        System.out.println("Instrucciones a ejecutar finalmente tanto si
se producen anomalias como si no");
    }

} //fin main

} //fin clase

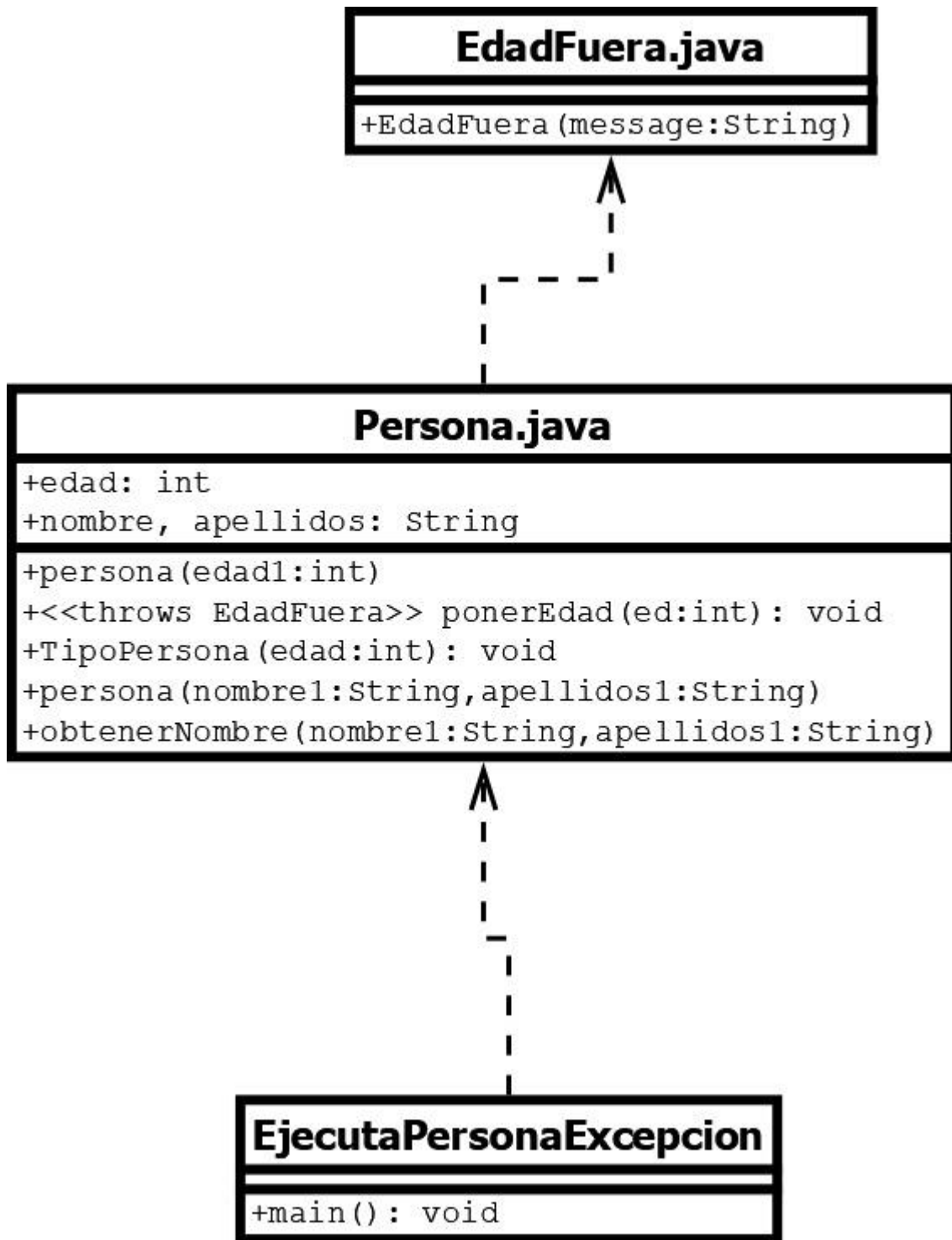
```

## **CORRIDA DEL PROGRAMA**

```
Dame la edad de la persona
500
No es valida la edad:Demasiado joven o demasiado viejo
Dame la edad de la persona
-200
No es valida la edad:Demasiado joven o demasiado viejo
Dame la edad de la persona
20
hola juan perrez de 20 años
Eres Joven, sigue con tus sueños y proyectos
PS C:\Users\irvyn\OneDrive\Pictures\paginaweb\practica12\combinado> |
```

## **PROGRAMA UNION DE AMBOS PROGRAMAS**

### **DIAGRAMA UML**



## PROGRAMA: PERSONA

```
public class Persona {
```

```

int edad;
String nombre;
    String apellidos;

public Persona() {
    nombre=null;
        apellidos=null;
    }

public Persona(String nombre1, String apellidos1) throws Exception
    {
        if(nombre1 == null || apellidos1 == null || "".equals(nombre1) ||
"".equals(apellidos1))
            throw new Exception("Argumentos no validos");
            //si el constructor no tiene los dos atributos lanza la excepciï½n el
objeto
            //no se crea el objeto

        else
            System.out.println("Se construyo el objeto");
            nombre=nombre1;
            apellidos=apellidos1;
        }

public Persona(int edad1){
    edad = edad1;
}

public void ponerEdad(int ed) throws EdadFuera {
    if ((ed < 0) || (ed > 130))
        throw new EdadFuera("Demasiado joven o demasiado viejo");
}

```

```

    else
        edad = ed;
}

public void TipoPersona(int edad) {
    if (edad <= 12)
        System.out.println("Eres un niño, juega y diviértete mucho");
    if ((edad > 12) && (edad < 18))
        System.out.println("Eres adolescente, come frutas y verduras");
    if ((edad >= 18) && (edad < 30))
        System.out.println("Eres Joven, sigue con tus sueños y proyectos ");
    if ((edad >= 30) && (edad < 60))
        System.out.println("Eres Joven Adulto productivo, sigue adelante, tu
puedes");
    if ((edad >= 60) && (edad < 80))
        System.out.println("Eres Adulto Mayor, te has convertido en una gran
persona");
    if ((edad >= 80) && (edad < 130))
        System.out.println("Eres Anciano, dejaste un gran legado");
}

```

// nombre y apellido

```

public void obtenerNombre(String nombre1, String apellidos1)
{
    nombre=nombre1;
    apellidos=apellidos1;
}
}

```

## PROGRAMA: EDADFUERA

```
public class EdadFuera extends java.lang.Exception {  
    /**  
     * Creates a new instance of <code>EdadFuera</code>.  
     */  
    public EdadFuera() {  
    }  
  
    /**  
     * Constructs an instance of <code>EdadFuera</code> with the specified detail  
     * message.  
     *  
     * @param message.  
     */  
    public EdadFuera(String message) {  
        super(message);  
    }  
}
```

## PROGRAMA: EDADFUERA

```
import java.util.*;  
import java.io.*;  
  
public class EjecutaPersonaExcepcion {  
    public static void main(String args[]) throws IOException {  
        InputStreamReader entrada = new InputStreamReader(System.in);  
        BufferedReader flujoEntrada = new BufferedReader(entrada);
```

```

String datoEntrada;
int varedad = 0;
String varnombre;
String varapellido;
boolean lecturaNombreCompleto= false;

Persona persona1 = new Persona();
do{
    try
    {
        System.out.println("Ingrese el nombre: ");
        varnombre = flujoEntrada.readLine();
        System.out.println("Ingrese el apellido: ");
        varapellido = flujoEntrada.readLine();
        persona1 = new Persona(varnombre,varapellido);
        lecturaNombreCompleto=true;
    }
    catch(Exception e)
    {
        System.out.println("No se pudo crear el objeto y llama el
mensaje de la Exception:"+ e.getMessage());
    }
}while(lecturaNombreCompleto==false);

do{
    try {
        System.out.println("Dame la edad de la persona");
        datoEntrada = flujoEntrada.readLine();
        varedad = Integer.parseInt(datoEntrada);
        persona1.ponerEdad(varedad);
    }

```

```

        } catch (NumberFormatException e1) // Excepci3n si lo
para ese tipo de error
    {
        System.out.println("Formato no valido deben ser
numeros " + e1.getMessage());
        e1.printStackTrace();// seguimiento del error
    } catch (EdadFuera e2) {
        System.out.println("No es valida la edad:" +
e2.getMessage());
    }
}while(varedad < 0 || varedad > 130);

    System.out.println("hola " + persona1.nombre + " " +
persona1.apellidos + " de " + persona1.edad + " años");
    persona1.TipoPersona(varedad);

} // fin main

} // fin clase

```

## **CORRIDA DEL PROGRAMA**



```
Ingrese el nombre:
Ingrese el apellido:
No se pudo crear el objeto y llama el mensaje de la Exception:Argumentos no validos
Ingrese el nombre:
Irvyn
Ingrese el apellido:
Xicale Cabrera
Se construyo el objeto
Dame la edad de la persona
-200
No es valida la edad:Demasiado joven o demasiado viejo
Dame la edad de la persona
21
hola Irvyn Xicale Cabrera de 21 años
Eres Joven, sigue con tus sueños y proyectos
PS C:\Users\irvyn\OneDrive\Pictures\paginaweb\practica12\combinado>
```

¿Qué es una excepción en Java?= son errores que se generan en tiempo de ejecución

¿Cuáles son los tipos de excepción?= de excepción y error

¿Cómo se trata una excepción y un error?= en las excepciones generalmente el programador debe proporcionar el código que las trate o gestione pero en los errores no se pueden tratar

¿Cómo puede crear una excepción un programador para validar un dato?=por redefinición de la clase o por herencia.

## Conclusión

Comprendemos mejor la creación y uso de las excepciones para nuestros programas.

**PROGRAMACIÓN II**  
**RÚBRICA HOJA DE EVALUACIÓN DE PRÁCTICAS**

**MATRICULA:** 201963582

**FECHA:** 19 de Abril de 2021

**NOMBRE:** Xicale Cabrera Irvyn

**No. PRACTICA:** 12

**INDIVIDUAL ( x ) COLABORATIVA ( )**

CRITERIOS	Deficiente	Suficiente	Bueno	Excelente	Calificación Obtenida	
	1-5-9	6-7-9	8-9	9-1-10	%	Puntos
<b>CONOCIMIENTO TEORICO</b> 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 PRACTICA</b> 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 PRACTICA</b> 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.		
<b>ACTITUD DE APRENDER Y COLABORAR EN EQUIPO DE TRABAJO</b> 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		
<b>Total</b>						

  
Firma del Alumn@

  
Vo. Bo. Docente

201963582 Irvyn Xicale Cabrera