



## **Curso: Programación I - INFO-103.**

**Desarrollo de aplicaciones mediante arquitectura**

**POO (Programación Orientada a Objetos).**

**Estudiantes:**

**Saborio Rojas Rafael Alberto**

**Pérez Pérez Carlos Ernesto**

**Pérez Córdoba Freddy Junior**

**Docente : Melber Dalorso Cruz**

**diciembre 2021.**

## Índice

Índice	2
Resumen	3
Abstract	3
1. Introducción	4
2. Objetivos	4
2.1 Objetivo General	4
2.2 Objetivos específicos	4
4. Desarrollo del trabajo	5
4.1 Programa utilizado para el desarrollo	5
4.2 Paquetes Utilizados	5
4.3 Limitaciones	6
4.3.1 Clasificador de palabras	6
4.3.2 Palabras desordenadas.	8
4.3.3 Trivia Matemática.	10
5. Representación UML de cada clase	12
6. Conclusiones	12
7. Recomendaciones	13
8. Referencias	13

## **Resumen**

El presente proyecto final del curso de programación I, tiene el objetivo de poner en práctica los diversos conocimientos adquiridos durante el III cuatrimestre del 2021. En dicho trabajo, se comparan e investigan códigos en diversas fuentes confiables de internet, tales como; artículos relacionados al tema, revistas, artículos de periódicos etc. Sin embargo, el proyecto casi en su totalidad es realizado según lo ejecutado y aprendido en las clases del curso, utilizando los métodos y atributos explicados durante las clases del cuatrimestre.

En dicho trabajo, se realiza la creación de tres distintas aplicaciones, con el fin de resolver lo solicitado en los problemas dados por el docente.

**Palabras clave:** programación, aplicaciones, códigos, internet, métodos, atributos, problemas, paquetes, clases.

## **Abstract**

The present final project of the programming I course, has the objective of putting into practice the diverse knowledge acquired during the III quarter of 2021. In this work, codes are compared and investigated in several reliable internet sources, such as; articles related to the subject, magazines, newspaper articles, etc. However, the project is almost entirely carried out according to what was executed and learned in the classes of the course, using the methods and attributes explained during the classes of the term.

In this work, the creation of three different applications is carried out, in order to solve the problems requested by the teacher.

**Keywords:** programming, applications, codes, internet, methods, attributes, problems, paquetes, clases.

# **1. Introducción**

El presente proyecto, tiene como finalidad utilizar los múltiples conocimientos adquiridos en el curso de programación I, para así poder desarrollar diversas aplicaciones que permitan la resolución de los problemas dados. Para poder confeccionar las tres distintas aplicaciones, se utilizó el programa Apache netBeans, con el lenguaje de programación denominado JAVA JDK, con su versión 8.

## **2. Objetivos**

- Desarrollar las aplicaciones solicitadas.
- Investigar sobre diferentes paquetes para el desarrollo de las aplicaciones.
- Desarrollar de forma ordenada y simple la solución de cada solución de las aplicaciones solicitadas.

### **2.1 Objetivo General**

Desarrollar tres distintas aplicaciones capaces de lograr la resolución de los problemas asignados testeando así diferentes formas hasta conseguir el resultado exacto.

### **2.2 Objetivos específicos**

- **2.2.1** Desarrollar un programa, capaz de clasificar las palabras según la cantidad de letras, ordenarlas por cantidad de vocales y por cantidad de consonantes.
- **2.2.2** Ejecutar un programa capaz de revolver palabras preestablecidas dentro del sistema, presentando las al usuario para que las ordene mostrándole si el resultado es correcto o incorrecto.
- **2.2.3** Ejercer un programa que genere operaciones matemáticas aleatorias.

## **3. Formulación del problema**

Desarrollo de tres aplicaciones (clasificador de palabras, palabras desordenadas y trivia matemática) que estén involucradas una con la otra mediante el uso de clases en la plataforma de desarrollo de programación JAVA 8 JDK llamada NetBeans IDE 12.

## 4. Desarrollo del trabajo

### 4.1 Programa utilizado para el desarrollo

El programa utilizado, es Apache netBeans, mismo que como, Juliana Fantino, 2021 menciona en su revista, que Apache netBeans es un IDE (Integrated Development Environment), el cual es gratuito y de código abierto, se trata de un editor de código fuente con recursos, que permiten el automático llenado de código, este IDE ofrece un gran compilador y auto rellenado Juliana Fantino. (5/11/2021). ¿Qué es Netbeans? ¡Crea aplicaciones con Java a la velocidad de la luz!. crehana, 0, 1.



El lenguaje de desarrollo utilizado, Java mismo que Herbert Schildt dice que es una combinación de diferentes conceptos innovadores uniéndolos para así crear la mejor combinación, JAVA esta relacionado con el lenguaje C++ y el lenguaje C, JAVA se hereda de dos lenguajes del Lenguaje C, JAVA hereda su sintaxis, también muchas de las características que son orientadas a objetos fueron influenciadas por C++, Herbert Schildt.(2019). Java: The Complete Reference, Eleventh Edition.

**Fig 1.** Logo del programa utilizado (Apache netBeans)

### 4.2 Paquetes Utilizados

Los paquetes utilizados fueron los siguientes:

- java.util.Arrays.
- java.util.Collections.
- java.util.JOptionPane.

### **4.3 Limitaciones**

La utilización de otras bibliotecas que no han sido vistas en clase, lo cual representó la búsqueda muy detallada del funcionamiento de esas bibliotecas a la hora de aplicar los métodos en la generación de código.

Se debió realizar un análisis profundo sobre la utilización del método `math` con sus variables para el desarrollo del proyecto.

Al desarrollar la impresión de la clasificación de palabras se tuvieron problemas, para resolverlo se creó un método estándar de impresión.

#### **4.3.1 Clasificador de palabras**

Para dar inicio, se evidencia el problema de clasificador de palabras, mismo donde se le solicita a los usuarios ingresar tres palabras mediante `JOptionPane` (paquete de java), posteriormente por el mismo paquete, se le pide al usuario indicar cuál de las tres opciones desea seleccionar.

La opción número uno, le permitirá al usuario conocer el orden por cantidad de letras, la opción número dos le permitirá al usuario saber el orden por cantidad de vocales y por último, la opción tres, le permitirá al usuario ordenar las palabras por la cantidad de consonantes. Al seleccionar, la opción uno se va a ingresar a un ciclo `for` en el cual, se va a ir ordenando según la cantidad de letras que tenga, al finalización, de dicho proceso, se procede a imprimir un mensaje mediante `JOptionPane`, el cual dará el resultado. La opción número dos, va a constar de varios ciclos `for`, los cuales se van a encargar de ordenar las palabras por la cantidad de vocales, los cuales se van a imprimir nuevamente por el paquete de java (`JOptionPane`), por último la tercera opción, consta de varios ciclos `for` los cuales se encargan de ordenar las palabras por consonantes, cuyo resultado se imprimirá mediante el paquete `JOptionPane`.

## Clase:

### ClasificadorPalabras

```
-vocales: char[*] = {
    'a', 'e', 'i', 'o', 'u', 'á', 'é', 'í', 'ó', 'ú',
    'A', 'E', 'I', 'O', 'U', 'Á', 'É', 'Í', 'Ó', 'Ú'
}
-consonantes: char[*] = {
    'b', 'c', 'd', 'f', 'g', 'h', 'j', 'k', 'l', 'm', 'n', 'ñ', 'p', 'q', 'r', 's', 't', 'v', 'w', 'x', 'y', 'z',
    'B', 'C', 'D', 'F', 'G', 'H', 'J', 'K', 'L', 'M', 'N', 'Ñ', 'P', 'Q', 'R', 'S', 'T', 'V', 'W', 'X', 'Y', 'Z'
}
-contador: int[*]
-arrPalabras: String[*]
-palabrasIngresadas: String[*]
-arreglo: int[*]

+ClasificadorPalabras()
+llenadoPalabras()
+mostrarMenu()
+ordenadoPalabras(palabras: String, cantidades: int)
+ordenPalabras()
+ordenadoVocales()
+imprimirResultado(mensaje: String, palabras: String, cantidades: int)
+ordenConsonantes()
+ordenAlfabetico()
+ordenIngreso()
```

## Métodos:

```
public ClasificadorPalabras()
public void llenadoPalabras()
public void mostrarMenu()
public void ordenadoPalabras(String[] palabras, int[] cantidades)
public void ordenPalabras()
public void ordenadoVocales()
private void imprimirResultado(String mensaje, String[] palabras, int[] cantidades)
public void ordenConsonantes()
public void ordenAlfabetico()
public void ordenIngreso()
```

```
this.arreglo = new int[];
int ConsonantesNum = consonantes.length, auxContaConsonante;
int contConsonantes = 0;
```

### Atributos:

```
char[] consonantes
int[] contador;
String[] arrPalabras;
String[] palabrasIngresadas;
int[] arreglo;
String dato = "";
int opcion = 0;
String aux;
int i = 0;
String opciones = "";
int[] numVocales = new int[];
int VocalesNum = vocales.length, ConsonantesNum = consonantes.length;
int contVocales = 0;
String auxConsonante = "";
```

```
char[] vocales = {'a', 'e', 'i', 'o', 'u'};
char[] consonantes = {'b', 'c', 'd', 'f', 'g', 'h', 'j', 'k', 'l', 'm', 'n', 'p', 'q', 'r', 's', 't', 'v', 'w', 'x', 'y', 'z'};

public void ClasificadorPalabras() {
    String aux = "";
    String[] arrPalabras = new String[3];
    String[] arrPalabrasTam = new String[3];
    String[] arrConso = new String[3];

    arrPalabras[0] = JOptionPane.showInputDialog(null, "Ingrese la primera palabra: ");
    arrPalabrasTam[0] = arrPalabras[0];
    arrConso[0] = arrPalabras[0];

    arrPalabras[1] = JOptionPane.showInputDialog(null, "Ingrese la segunda palabra: ");
    arrPalabrasTam[1] = arrPalabras[1];
    arrConso[1] = arrPalabras[1];

    arrPalabras[2] = JOptionPane.showInputDialog(null, "Ingrese la tercera palabra: ");
    arrPalabrasTam[2] = arrPalabras[2];
    arrConso[2] = arrPalabras[2];

    String opc = "";
    opc = JOptionPane.showInputDialog(null, "\n1. Ordenar por cantidad de letras, \n2. Ordenar por cantidad de vocales, \n3. Ordenar por cantidad de consonantes,");

    if (opc.equalsIgnoreCase("1")) {
        boolean cambio = true;
        int j = 0;
        String tmp;
        while (cambio) {
            cambio = false;
            j++;
            for (int i = 0; i < arrPalabras.length - j; i++) {
                if (arrPalabras[i].length() > arrPalabras[i + 1].length()) {
                    tmp = arrPalabras[i];
                    arrPalabras[i] = arrPalabras[i + 1];
                    arrPalabras[i + 1] = tmp;
                }
            }
        }
    }
}
```

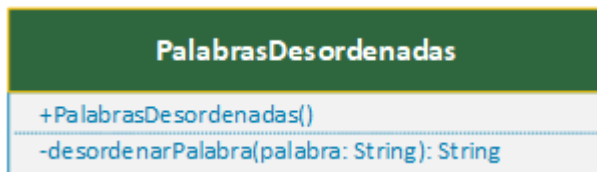
**Fig 2.** Demostración del código utilizado, para la creación de la aplicación.

### 4.3.2 Palabras desordenadas.

Seguidamente, para poder realizar este programa, se pre establecen 8 palabras, las cuales cuando el usuario inicia el programa, se procede a desordenar, y a empezar a mostrarle al usuario de forma incorrecta, de esta forma el usuario tiene que analizar la palabra desordenada que se le muestra mediante un mensaje de JOptionPane, una vez el usuario ingrese la palabra que cree correcta el sistema va a proceder a analizarla todo esto en un ciclo while, el cual determinará si es correcta la palabra ingresada o es incorrecta.



### Clase:



### Métodos:

`public void PalabrasDesordenadas()`

### Atributos:

```
String[] listaPalabras = "";
int i = 0;
int correctas = 0;
int incorrectas = 0;
int ramdonEntero = 0;
String palabra = "";
String palabraDesordenada = "";
String opc = "";
opc = "";
```

```
public void PalabrasDesordenadas() {
    String[] palabras = {"trailer", "mesa", "ventilador", "copa", "libro", "audifonos", "sofa", "vevedero"};
    int i = 4;
    int cor = 0;
    int incorr = 0;
    while (i >= 0) {
        int ramdonEntero = (int) Math.floor(Math.random() * (7 - 0 + 1) + 0);
        String palabra = palabras[ramdonEntero];
        String desPalabra = desordenarPalabra(palabra);
        String opcIngreso = "";
        JOptionPane.showMessageDialog(null, "Palabras desordenadas " + desPalabra);

        opcIngreso = JOptionPane.showInputDialog(null, "Ingrese la palabra correcta: ");

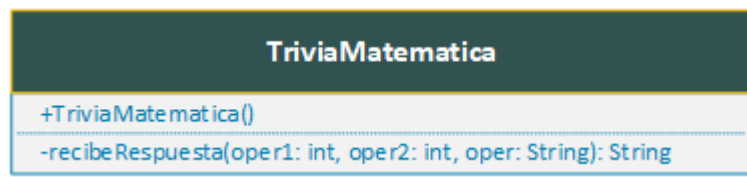
        if (opcIngreso.equalsIgnoreCase(palabra)) {
            System.out.println("Palabra Correcta");
            JOptionPane.showMessageDialog(null, "Palabras Correcta");
            cor++;
        } else {
            System.out.println("Palabra Incorrecta");
            JOptionPane.showMessageDialog(null, "Palabra Incorrecta");
            incorr++;
        }
        i--;
    }
    JOptionPane.showMessageDialog(null, "Palabras Coorrectas " + cor + "\n Palabras Incorrectas " + incorr);
}
```

**Fig 3.** Demostración del código utilizado, para la creación de la aplicación.

### 4.3.3 Trivia Matemática.

Por último, en la trivia matemática, al iniciar el programa va a generar operaciones aleatorias las cuales le va a presentar una operación al usuario mediante JOptionPane, la cual le tendrá que dar en el botón de aceptar, a continuación le mostrará un posible resultado y así sucesivamente durante 2 opciones más, posteriormente le solicitará al usuario que ingrese el resultado de la operación correcta y automáticamente el programa validará si el resultado de la operación es correcta, este proceso se repetirá durante 5 veces, al final mostrará un mensaje que dirá cuántas operaciones resolvió correctamente y le mostrará cuantas operaciones realizó incorrectamente.

#### Clase:



#### Métodos:

```
public void TriviaMatematica()
static String recibeRespuesta(int oper1, int oper2, String oper)
```

#### Atributos:

```
int correctas = 0;
int incorrectas = 0;
int i = 0;
String[] operEspeciales = {"+", "-", "*"};
String[] result;
String oper = "";
String operacion = "";
String resultCorrecto = "";
String resultIncorrecto = "";
String resultIncorrecto2 = "";
int contador = 0;
int res = 0;
```

```

public class TriviaMatematica {

    public void TriviaMatematica() {
        String[] operadores = {"+", "-", "*"};
        int cor = 0;
        int incor = 0;
        for (int i = 0; i < 5; i++) {
            String[] resp = new String[3];
            int operador1 = (int) Math.floor(Math.random() * (25 - 5 + 1) + 5);
            int operador2 = (int) Math.floor(Math.random() * (25 - 5 + 1) + 5);
            String operador = operadores[(int) Math.floor(Math.random() * (2 - 0 + 1) + 0)];
            String operacion = String.valueOf(operador1) + operador + String.valueOf(operador2);

            String respuestaCorrecta = dameRespuesta(operador1, operador2, operador);
            String respuestaIncorrecta = String.valueOf((int) Math.floor(Math.random() * ((Integer.valueOf(respuestaCorrecta) - 1) - (-35) + 1) + (-35)));
            String respuestaIncorrecta2 = String.valueOf((int) Math.floor(Math.random() * (900 - ((Integer.valueOf(respuestaCorrecta) - 1)) + 1) + ((Integer.valueOf(
            resp[0] = respuestaCorrecta;
            resp[1] = respuestaIncorrecta;
            resp[2] = respuestaIncorrecta2;
            System.out.println();
            JOptionPane.showMessageDialog(null, "Cual es el resultado de la operacion: " + operacion);
            if (i % 2 == 0) {

                for (int j = 0; j < resp.length; j++) {
                    JOptionPane.showMessageDialog(null, resp[j]);
                }

            } else {

                for (int j = resp.length - 1; j >= 0; j--) {
                    JOptionPane.showMessageDialog(null, resp[j]);
                }

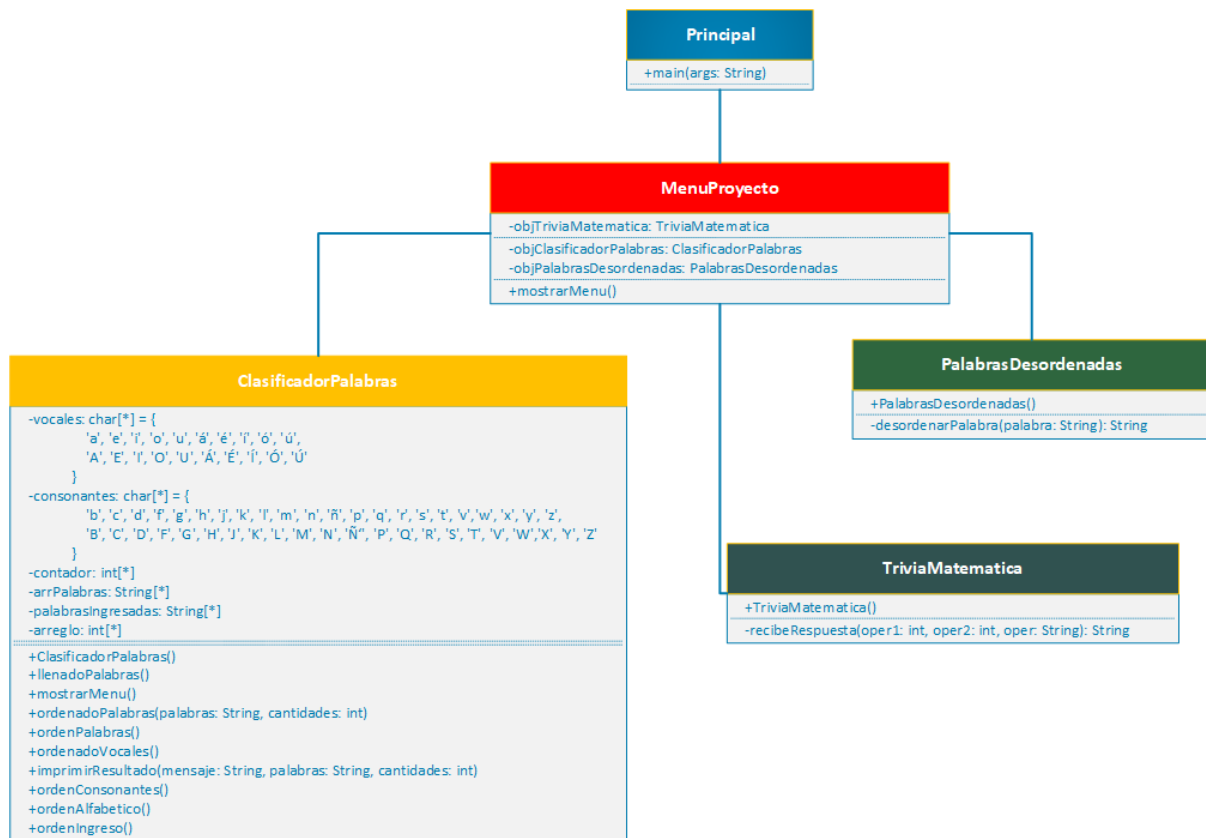
            }

            int res = 0;
            System.out.println("Ingrese el resultado de la operacion");
            res = Integer.valueOf(JOptionPane.showInputDialog(null, "Ingrese el resultado de la operacion"));
        }
    }
}

```

**Fig 4.** Demostración del código utilizado, para la creación de la aplicación.

## 5. Representación UML de proyecto



## 6. Conclusiones

Se logra desarrollar las tres aplicaciones mostradas en los problemas de investigación, las cuales eran desarrollar un programa capaz de clasificar las palabras según la cantidad de letras, las cuales se imprimen de forma ordenada por vocales o consonantes.

Se ejecuta un programa capaz de mostrar letras de formas aleatorias, previamente establecidas, reproduciendo el orden correcto o incorrecto al usuario de forma que enumera los fallos y aciertos.

Se desarrolló una aplicación que genera operaciones matemáticas aleatorias, representándose cada una en forma de trivia, para que el usuario ingrese el resultado correcto de la trivia demostrada.


Se ejecutó de forma exitosa la programación de las tres aplicaciones solicitadas en la formulación del problema.

## 7. Recomendaciones

Involucrarse en la investigación profunda de los métodos de desarrollo para la aplicación de lenguaje en JAVA.

Implementar diferentes paquetes y formas de realizar la programación no practicadas en clases.

## 8. Referencias

Fantino, J., 2021.  ¿Qué es Netbeans y para qué sirve? [2022]. [online] <https://www.crehana.com>. Available at: <<https://www.crehana.com/mx/blog/desarrollo-web/que-es-netbeans/>> [Accessed 5 November 2021].

Schildt, H., 2019. *Z-Library single sign on*. [online] B-ok.lat. Available at: <<https://b-ok.lat/ireader/3697965>> [Accessed 10 December 2021].