



UNIVERSIDAD TECNOLÓGICA DE PANAMÁ
FACULTAD EN INGENIERÍA DE SISTEMAS COMPUTACIONALES
LIC. INGENIERÍA EN SISTEMAS Y COMPUTACIÓN



HERRAMIENTAS DE PROGRAMACIÓN APLICADA II
"Proyecto #1" GRUPAL

FACILITADOR:
LUDIA GÓMEZ

INTEGRANTES:

CHACON, JEFFERSON 20-70-7314

GONZÁLEZ, JORGE 8-1002-2167

HIM, AMY 3-753-2253

RAMOS, DIEGO 8-1002-1456

GRUPO:
1IL121

I SEMESTRE FECHA DE ENTREGA:
06/15/23

Introducción

Este proyecto tiene como objetivo ayudar a los niños a prepararse para un emocionante concurso nacional de matemáticas. En este programa llamado los participantes podrán fortalecer sus habilidades matemáticas y poner a prueba su destreza en diversas operaciones.

En este proyecto, exploraremos el poder de los arreglos en Java y cómo pueden facilitar la resolución de problemas matemáticos. Cabe destacar que, los arreglos son estructuras de datos que nos permiten almacenar múltiples valores bajo un mismo nombre. Utilizaremos esta potente herramienta para crear un programa interactivo que ayudará a los niños a practicar las operaciones aritméticas básicas: suma, resta, multiplicación y división.

El programa les presentará a los participantes una serie de desafíos matemáticos, donde deberán responder preguntas en forma de operaciones aritméticas. Los niños podrán ingresar sus respuestas y el programa verificará si son correctas o incorrectas. Además, el programa llevará un registro de las respuestas correctas e incorrectas de cada participante, para que puedan hacer un seguimiento de su progreso y mejorar en las áreas que necesiten.

Durante el proyecto grupal, también aprenderemos cómo utilizar los arreglos en Java para organizar y almacenar los resultados de los participantes, lo que nos permitirá llevar un registro. Al finalizar este taller, los niños estarán preparados y confiados para enfrentar el concurso nacional de matemáticas. ¡Así que prepárense para sumergirse en el emocionante mundo de las operaciones aritméticas y desafiar sus habilidades matemáticas con nuestro programa para niños de 4 grado.

Informe

La clase "**Adivinanza**" proporciona métodos para acceder y manipular preguntas, respuestas y respuestas correctas del segundo juego de adivinanzas en donde el niño tiene 3 oportunidades para responder. Los métodos generarPregunta, generarRespuestas y generarRespCorrecta utilizan un valor aleatorio para seleccionar una pregunta y sus respuestas correspondientes, mientras que los métodos getPreguntas, getResp y getRespCorrecta devuelven los arreglos completos de preguntas, respuestas y respuestas correctas, respectivamente.

```
public class Adivinanza {
    private String preguntas[];
    private String resp[][];
    private int respCorrecta[];
    private int random;

    public void asignar (String preguntas[], String resp[][], int respCorrecta[], int random) {
        this.preguntas = preguntas;
        this.resp = resp;
        this.respCorrecta = respCorrecta;
        this.random = random;
    }

    public String generarPregunta() {
        return preguntas[random];
    }

    public String[] generarRespuestas() {
        return resp[random];
    }

    public int generarRespCorrecta() {
        return respCorrecta[random];
    }

    public String[] getPreguntas() {
        return preguntas;
    }

    public String[][] getResp() {
        return resp;
    }
}
```

la clase "**Ecuacion**" Esta clase tiene como objetivo generar ecuaciones con términos desconocidos para que los usuarios puedan resolverlas. El método cargarTerminos se utiliza en el programa para asignar los valores de los términos de la ecuación. El método obtenerRespuesta permite entonces obtener el valor de un término específico en el arreglo de términos. El método generarEcuacion construye una cadena que representa una ecuación con un término desconocido en la posición especificada. Por último, el método getTerminos devuelve el arreglo de términos completo.

```
import java.util.Random;

public class Ecuacion {
    private int[] terminos;

    Random random = new Random();
    public Ecuacion() {
        terminos = new int[3];
    }

    public void cargarTerminos(int num1, int num2, int resultado) {
        terminos[0] = num1;
        terminos[1] = num2;
        terminos[2] = resultado;
    }

    // tentativo
    public int obtenerRespuesta(int posicion, int[] terminos) {
        return terminos[posicion];
    }

    public String generarEcuacion(int posicion, char operador) {
        String ecuacion = "";
        switch(posicion) {
            case 0:
                ecuacion = "?" + " " + operador + " " + terminos[1] + " = " + terminos[2];
                break;
            case 1:
                ecuacion = terminos[0] + " " + operador + " " + "?" + " = " + terminos[2];
                break;
            case 2:
                ecuacion = terminos[0] + " " + operador + " " + terminos[1] + " = " + "?";
                break;
        }
    }
}
```

la clase "MundoM" proporciona métodos para asignar y verificar respuestas proporcionadas por el usuario, generar números aleatorios, generar operadores aleatorios, calcular resultados de operaciones aritméticas y obtener el valor de la respuesta proporcionada por el usuario. A continuación, se explica paso a paso lo que se hace en el programa:

- Se declara la clase "MundoM" con una variable de instancia respUs de tipo entero que almacena la respuesta proporcionada por el usuario.
- Se crea una instancia de la clase Random llamada random, que se utiliza para generar números aleatorios.
- El método asignar recibe un parámetro respUs de tipo entero y asigna su valor a la variable de instancia respUs.
- El método verificarRespuesta recibe un parámetro respCorrecta de tipo entero y compara el valor de respUs con respCorrecta. Si son iguales, devuelve true, indicando que la respuesta proporcionada por el usuario es correcta. En caso contrario, devuelve false.
- El método generarNum recibe dos parámetros min y max de tipo entero, que representan el rango mínimo y máximo para generar un número aleatorio. Utiliza el método nextInt de la instancia random para generar un número aleatorio dentro del rango especificado y lo devuelve.
- El método generarOperador recibe un parámetro OPERADORES de tipo arreglo de caracteres, que contiene los operadores matemáticos disponibles. Utiliza el método nextInt de la instancia random para seleccionar un índice aleatorio del arreglo OPERADORES y devuelve el carácter correspondiente.
- El método calcularResultado recibe tres parámetros num1, num2 y op de tipo entero y carácter, que representan los operandos y el operador de una operación aritmética. Utiliza una estructura de control switch para determinar qué operación se debe realizar. Calcula el resultado de la operación y lo asigna a la variable resultado. Finalmente, devuelve el valor de resultado.
- El método getRespUs devuelve el valor almacenado en la variable de instancia respUs.

```
public class MundoM {
    private int respUs;

    // datos de entrada
    // respuestas de los niños
    Random random = new Random();

    public void asignar(int respUs) {
        this.respUs = respUs;
    }

    public boolean verificarRespuesta(int respCorrecta) {
        return respUs == respCorrecta;
    }

    public int generarNum(int min, int max) {
        return random.nextInt(max - min + 1) + min;
    }

    public char generarOperador(char[] OPERADORES) {
        return OPERADORES[random.nextInt(OPERADORES.length)];
    }

    public int calcularResultado(int num1, int num2, char op) {
        int resultado = 0;
        switch (op) {
            case '+':
                resultado = num1 + num2;
                break;
        }
    }
}
```

```

    return OPERADORES[random.nextInt(OPERADORES.length)];
}

public int calcularResultado(int num1, int num2, char op) {
    int resultado = 0;
    switch (op) {
        case '+':
            resultado = num1 + num2;
            break;
        case '-':
            resultado = num1 - num2;
            break;
        case '/':
            resultado = num1 / num2;
            break;
        case '*':
            resultado = num1 * num2;
            break;
    }
    return resultado;
}

public int getRespUs() {
    return respUs;
}
}
```

Class Main

- Se importan las clases necesarias y se crea la clase Main.
- Dentro del método main, se crean objetos de las clases MundoM, Adivinanza y Ecuacion.
- Se crea un objeto de la clase Random para generar números aleatorios.
- Se cargan imágenes usando objetos de la clase ImageIcon.

```
6 public class Main {
7     Run|Debug
8     public static void main(String[] args) {
9         // Objeto de la clase
10        MundoM mates = new MundoM();
11        Adivinanza adi = new Adivinanza();
12        Ecuacion ecuacion = new Ecuacion();
13
14        // Objeto Random
15        Random rand = new Random();
16        int random;
17
18        // Imágenes
19        ImageIcon img_bien1 = new ImageIcon(filename:"./imgs/bien1.png");
20        ImageIcon img_bien2 = new ImageIcon(filename:"./imgs/bien2.jpg");
21        ImageIcon img_bien3 = new ImageIcon(filename:"./imgs/bien3.jpg");
22        ImageIcon img_mot1 = new ImageIcon(filename:"./imgs/motivador1.png");
23        ImageIcon img_mot2 = new ImageIcon(filename:"./imgs/motivador2.jpg");
24        ImageIcon imgs[] = {img_bien1, img_bien2, img_bien3, img_mot1, img_mot2};
25    }
26 }
```

- Se declaran y se inicializan variables que se utilizarán más adelante en el programa.
- Se muestra un mensaje de bienvenida al juego mediante el uso de JOptionPane.
- Se muestra un mensaje para comenzar con una adivinanza.
- Se inicia un bucle do-while para realizar la adivinanza antes de entrar al menú principal.
- Dentro del bucle, se comprueba si el jugador ha agotado los intentos. En caso afirmativo, se muestra un mensaje y se da la opción de intentarlo nuevamente o salir del juego.

```
// INICIANDO EL PROGRAMA
JOptionPane.showMessageDialog(parentComponent:null, message:"Bienvenido al Mundo de las Operaciones Matemáticas", title:"Mundo de las Operaciones Matemáticas", JOptionPane.INFORMATION_MESSAGE);
JOptionPane.showMessageDialog(parentComponent:null, message:"Comencemos con una adivinanza\n¡tú puedes!", title:"Mundo de las Operaciones Matemáticas", JOptionPane.INFORMATION_MESSAGE);

// ADIVINAZA ANTES DE ENTRAR AL MENÚ
do {
    if (contIntentos == 3) {
        JOptionPane.showMessageDialog(parentComponent:null, message:"Lo siento, se te acabaron los intentos.", title:"Mundo de las Operaciones Matemáticas", JOptionPane.INFORMATION_MESSAGE);
        opcion = JOptionPane.showConfirmDialog(parentComponent:null, message:"¿Deseas intentarlo otra vez?", tiempoTomado, JOptionPane.YES_NO_CANCEL_OPTION);
        // reiniciamos los intentos
        if (opcion == JOptionPane.YES_OPTION) {
            contIntentos = 0;
        } else {
            JOptionPane.showMessageDialog(parentComponent:null, message:"Gracias por jugar!", title:"Mundo de las Operaciones Matemáticas", JOptionPane.INFORMATION_MESSAGE);
            System.exit(status:0);
        }
    }
} while (opcion == JOptionPane.YES_OPTION);
// break;
```

- Se genera una adivinanza aleatoria utilizando el objeto adi y se muestra al jugador mediante un cuadro de diálogo JOptionPane.showMessageDialog.
- El jugador elige una respuesta y se compara con la respuesta correcta. Si la respuesta es correcta, se muestra un mensaje de felicitación; de lo contrario, se muestra un mensaje de error y se incrementa el contador de intentos.
- El bucle se repite mientras la respuesta sea incorrecta o el jugador elija volver a intentarlo.
- Después del bucle de la adivinanza, se muestra el menú principal del juego en otro bucle do-while.

```
// MENÚ
do {
    opcionS = JOptionPane.showInputDialog(parentComponent:null,
        "Elige una opción\n"
        + "1. Calculos Mentales\n"
        + "2. Adivina si puedes\n"
        + "3. Salir\n",
        title:"Bienvenido al Mundo de las Operaciones Matemáticas", JOptionPane.INFORMATION_MESSAGE, icon:null, selectionValue:1);

    try {
        opcion = Integer.parseInt(opcionS);
        if (opcion < 1 || opcion > 3) {
            bien = false;
            JOptionPane.showMessageDialog(parentComponent:null, message:"Lo siento, la opción que elegiste no está en el menú.", title:"Mundo de las Operaciones Matemáticas", JOptionPane.INFORMATION_MESSAGE);
        }
    } catch (NumberFormatException e) {
        bien = false;
        JOptionPane.showMessageDialog(parentComponent:null, message:"Lo siento, las opciones se eligen con números!\nVuelve a intentarlo.", title:"Mundo de las Operaciones Matemáticas", JOptionPane.INFORMATION_MESSAGE);
    }
} while (!bien);
```

- El jugador elige una opción del menú y se realiza una validación para asegurarse de que la opción elegida sea válida.
- Se ejecuta un bloque switch para realizar diferentes acciones según la opción seleccionada.
- Para la opción 1, se inicia el desafío de "Cálculos Mentales".
- Se muestra un mensaje de instrucciones y se inicia un bucle for para realizar 7 preguntas.
- Dentro del bucle, se generan números aleatorios y un operador para formar una ecuación aritmética.
- Se muestra la ecuación al jugador y se solicita la respuesta mediante un cuadro de diálogo JOptionPane.showInputDialog.

```
switch(opcion) {
    // Opcion 1
    case 1:
        // tiempo inicio del juego
        tiempoInicioJuego = System.currentTimeMillis();
        JOptionPane.showMessageDialog(parentComponent:null, "Calculos Mentales\n"
            + "Debes responder correctamente 7 preguntas\n"
            + "Tu puedes!\n",
            title:"Calculos Mentales", JOptionPane.PLAIN_MESSAGE, icon:null);
        for (i = 0; i < 7; i++) {
            num1 = mates.generarNum(mini:2, max:9);
            num2 = mates.generarNum(mini:2, max:9);
            op = mates.generarOperador(OPERADORES);
            resultado = mates.calcularResultado(num1, num2, op);

            do {
                // tiempo de inicio
                tiempoInicio = System.currentTimeMillis();
                bien = true;
                resp5 = JOptionPane.showInputDialog(parentComponent:null,
                    "Responde lo siguiente\n"
                    + num1 + " " + op + " " + num2 + " = ?\n",
                    "Calculos Mentales - Desafio " + (i+1) + "/" + 7,
                    JOptionPane.QUESTION_MESSAGE, icon:null, selectionValues:null, initialSelectionValue:null).toString());
                try {
                    resultado = Integer.parseInt(resp5);
                } catch (NumberFormatException e) {
                    JOptionPane.showMessageDialog(parentComponent:null, message:"El numero ingresado no es valido.\n\n");
                    bien = false;
                }
            } while (!bien);

            tiempoFinal = System.currentTimeMillis();
            // tiempo total
            tiempoTotal = tiempoFinal - tiempoInicioJuego;
            minutos = (tiempoTotal / 1000) / 60;
            segundos = (tiempoTotal / 1000) % 60;

            if (minutos != 0) {
                tiempoTomado = "Te tomaste un tiempo de: " + minutos + " minutos y " + segundos + " segundos";
            } else {
                tiempoTomado = "Te tomaste un tiempo de: " + segundos + " segundos";
            }

            JOptionPane.showMessageDialog(parentComponent:null, "Felicidades!\n"
                + "Has completado el desafio de Calculos Mentales\n"
                + tiempoTomado,
                title:"Calculos Mentales", JOptionPane.PLAIN_MESSAGE, icon:null);
            break;
        }
    }
}
```

- Se compara la respuesta del jugador con el resultado calculado y se muestra un mensaje indicando si la respuesta es correcta o incorrecta. También se calcula y muestra el tiempo tomado para responder la pregunta.
- El bucle se repite hasta que se hayan realizado las 7 preguntas.
- Se muestra un mensaje de felicitación por completar el desafío y se muestra el tiempo total tomado para completar el desafío.

```
        } catch (NumberFormatException e) {
            JOptionPane.showMessageDialog(parentComponent:null, message:"El numero ingresado no es valido.\n\n");
            bien = false;
        }
    } while (!bien);
}

tiempoFinal = System.currentTimeMillis();
// tiempo total
tiempoTotal = tiempoFinal - tiempoInicioJuego;
minutos = (tiempoTotal / 1000) / 60;
segundos = (tiempoTotal / 1000) % 60;

if (minutos != 0) {
    tiempoTomado = "Te tomaste un tiempo de: " + minutos + " minutos y " + segundos + " segundos";
} else {
    tiempoTomado = "Te tomaste un tiempo de: " + segundos + " segundos";
}

JOptionPane.showMessageDialog(parentComponent:null, "Felicidades!\n"
    + "Has completado el desafio de Calculos Mentales\n"
    + tiempoTomado,
    title:"Calculos Mentales", JOptionPane.PLAIN_MESSAGE, icon:null);
break;
}
```

- Para la opción 2, se inicia el desafío de "Adivina si puedes".
- Se establece el tiempo de inicio utilizando `System.currentTimeMillis()` para calcular el tiempo que toma completar el desafío.
- Se muestra un cuadro de diálogo de instrucciones utilizando `JOptionPane.showMessageDialog()` para explicar al jugador cómo jugar.

```
// Opción 2
case 2:
    // tiempo de inicio
    tiempoInicio = System.currentTimeMillis();
    JOptionPane.showMessageDialog(parentComponent:null, "Instrucciones\n"
    + "Debes llenar con el numero que falta\n"
    + "Por ejemplo:\n"
    + "20 - ? = 15\n"
    + "La respuesta es 5 porque 20-5=15\n", title:"Adivina si puedes", JOptionPane.PLAIN_MESSAGE, icon:null
```

- Se crea un arreglo llamado `terminos` con capacidad para almacenar 3 enteros.
- Se inicia un bucle `do-while` para realizar las adivinanzas. Este bucle se repetirá hasta que el jugador decida no continuar.
- Dentro del bucle, se establece la variable `bien` en `true` para controlar si la respuesta del jugador es correcta.
- Se reinicia el contador de intentos `contIntentos` a 0.
- Se inicia otro bucle `do-while` para generar ecuaciones y obtener una respuesta del jugador.
- Dentro de este bucle, se generan dos números aleatorios (`num1` y `num2`) y un operador (`op`) utilizando métodos de la clase `mates`.
- Se calcula el resultado de la ecuación utilizando el método `calcularResultado()` de la clase `mates`.
- Se verifica si el resultado es mayor a 100. Si es así, se repite el proceso de generación de números y operador hasta obtener un resultado válido.

```
// arreglo de trabajo
int[] terminos = new int[3];
do {
    bien = true;
    // reiniciamos contador de intentos
    contIntentos = 0;
    do {
        // generar los terminos
        num1 = mates.generarNum(min:0, max:99);
        num2 = mates.generarNum(min:0, max:99);
        // generar el operador
        op = mates.generarOperador(OPERADORES);
        // generar el resultado
        resultado = mates.calcularResultado(num1, num2, op);

    } while (resultado > 100);

    // guardar en el arreglo de trabajo los terminos
    terminos[0] = num1;
    terminos[1] = num2;
    terminos[2] = resultado;
```

- Los números generados y el resultado se guardan en el arreglo `terminos`.
- Se genera un índice aleatorio (`random`) para seleccionar uno de los términos de la ecuación que se ocultará al jugador.
- Se cargan los términos de la ecuación en la instancia de la clase `ecuacion` utilizando el método `cargarTerminos()`.
- El resultado de la ecuación se actualiza con el valor del término oculto (`resultado = terminos[random]`).

- Se inicia otro bucle do-while para obtener la respuesta del jugador y verificar si es correcta.
- Dentro de este bucle, se muestra la ecuación al jugador mediante un cuadro de diálogo JOptionPane.showInputDialog() y se solicita la respuesta.
- Se compara la respuesta del jugador con el resultado calculado. Si la respuesta es incorrecta, se muestra un mensaje de error y se incrementa el contador de intentos contIntentos.
- Si la respuesta es correcta, se calcula el tiempo total tomado para responder la pregunta y se muestra un mensaje de felicitación junto con el tiempo tomado.
- Al final del bucle, se muestra el tiempo total tomado para responder la pregunta.
- Después del bucle de la adivinanza, se muestra un mensaje con el número de respuestas correctas e incorrectas obtenidas durante el desafío.

```
// tiempo final
tiempoFinal = System.currentTimeMillis();
// tiempo total
tiempoTotal = tiempoFinal - tiempoInicio;
// tiempo en minutos
minutos = (tiempoTotal / 1000) / 60;
// tiempo en segundos
segundos = (tiempoTotal / 1000) % 60;

continuar = JOptionPane.showConfirmDialog(parentComponent:null, "Te tomaste un tiempo de " + minutos + " minutos y " + segundos + " segundos\n\n"
+ "¿Deseas realizar otra adivinanza?", "Continuar?", JOptionPane.YES_NO_OPTION, JOptionPane.QUESTION_MESSAGE, icon:null);
while (continuar == JOptionPane.YES_OPTION);

tionPane.showMessageDialog(parentComponent:null, "Sea cual sea el resultado lo hicistes muy bien\n"
"Obtuviste\n"
"Respuestas Buenas\n"
: " + contCorrectas + "\n"
"Respuestas Malas\n"
: " + contIncorrectas + "\n",
"le:"Adivina si puedes", JOptionPane.PLAIN_MESSAGE, icon:null);
ak;
```

- El bucle principal se repite si el jugador elige continuar con otro desafío (opcion == JOptionPane.YES_OPTION).
- Al final del programa, se muestra un mensaje de despedida.

```
// Opción 3
case 3:
    break;
default:
    JOptionPane.showMessageDialog(parentComponent:null, message:"Opcion no valida", title:"Mundo de las Matematicas", JOptionPane.ERROR_MESSAGE);
    break;
}

// preguntamos si desea continuar
if (opcion != 3)
    opcion = JOptionPane.showConfirmDialog(parentComponent:null, message:"¿Deseas realizar otro desafio?", title:"Continuar? - Mundo de las Matematicas", JOptionPane.YES_NO_OPTION, JOptionPane.QUESTION_MESSAGE, icon:null);

while (opcion == JOptionPane.YES_OPTION || opcion != 3);
OptionPane.showMessageDialog(parentComponent:null, message:"Gracias por jugar!\n", title:"Mundo de las Matematicas", JOptionPane.PLAIN_MESSAGE, icon:null);
```


Conclusión

En conclusión, a lo largo de este proyecto grupal hemos aplicado diversos conceptos y técnicas de programación aprendidos en el salón de clases, para resolver el problema de preparar a los niños para un concurso nacional de matemáticas. a través de la aplicación de los conceptos de entrada/salida, ciclos de repetición, manejo de excepciones y arreglos, hemos logrado desarrollar un programa en Java que cumple con los requerimientos del problema planteado. Este programa interactivo, orientado a objetos, ha brindado a los niños la oportunidad de practicar y fortalecer sus habilidades matemáticas mientras se preparan para el concurso nacional. La combinación de estas técnicas ha permitido una solución eficiente y robusta, brindando una experiencia de aprendizaje divertida y efectiva. Gracias a este proyecto logramos aprender y expandir nuestros conocimientos.