



Programación Orientada a Objetos

Ingeniería Informática

Entorno de desarrollo

Integrantes:

Gutiérrez Montaña Juan Luis 22010601

Hernández Heredia Kevin 22010603

Grupo:

2a3B

Fecha de entrega:

17/febrero/2023

Introducción:

En esta primera unidad se ha trabajado con programación orientada a objetos, una metodología de programación que permite la creación de programas complejos mediante la definición de clases y objetos.

En las presentes prácticas se abordarán temas fundamentales para el desarrollo de aplicaciones en Java, centrándose en la estructura de un proyecto, el proceso de desarrollo y la depuración de aplicaciones.

Los aprendizajes adquiridos en estas prácticas serán esenciales para el desarrollo de aplicaciones de calidad en Java.

Competencia específica:

Conoce y aplica la estructura de proyectos en un entorno de desarrollo para la solución de problemas.

Marco teórico:

La estructura de un proyecto en Java se refiere a cómo se organizan los archivos y carpetas que conforman una aplicación. Una aplicación en Java puede constar de varios archivos, cada uno con un propósito específico. Para trabajar de manera eficiente en un proyecto de Java, es importante comprender la estructura de los archivos y carpetas, y cómo se relacionan entre sí.

El proceso de desarrollo de aplicaciones en Java implica varios pasos, uno de los cuales es la compilación. La compilación es el proceso de convertir el código fuente de Java en código ejecutable, que puede ser entendido y ejecutado por la máquina virtual de Java. Durante el proceso de compilación, se generan varios archivos que son necesarios para que la aplicación se ejecute correctamente.

Los entornos de desarrollo integrados (IDE, por sus siglas en inglés) de Java, como Eclipse o NetBeans, crean automáticamente algunos de estos archivos al compilar el código. Estos archivos incluyen los archivos .class, que contienen el código ejecutable, y los archivos .jar, que son bibliotecas de clases reutilizables que pueden ser compartidas entre diferentes aplicaciones.

Por último, la depuración es un proceso importante en el desarrollo de aplicaciones en Java. La depuración se utiliza para encontrar y corregir errores en el código fuente, lo que ayuda a garantizar que la aplicación funcione correctamente. Durante el proceso de depuración, se utilizan herramientas especiales, como el depurador de Java, para identificar y solucionar los errores en el código.

Material y Equipo:

- ✓ Computadora
- ✓ Eclipse IDE for Java Developers – 2023 – 03
- ✓ Java SE Development kit 19.0.2

Desarrollo de la practica:

A lo largo de la primera unidad se nos otorgaron diferentes problemas, los cuales teníamos que resolver con la programación orientada a objetos, para ello se nos comenzo a instruir de que manera realizarlos y distribuir nuestras maneras de solucionar esos problemas para su buen desempeño.

La implementación de cada problema se realizo en java, utilizando las herramientas proporcionadas por el lenguaje para la definicion de clases y métodos.

Para esto se comenzo creando un Proyecto en el cual se crearon 3 paquetes que serian los que contienen las clases para tener un mayor orden de estas. En el primer paquete se metieron los que serian entradas y salidas como el menu y herramientas que utilizaríamos, en el segundo paquete entraron lo que serian los metodos que resolverian los problemas para poder ser llamados y utilizados en la clase menu para probar su funcionamiento, en el ultimo paquete estan dos clases de prueba para un metodo y menu.

Resultados:

ToolsPanel

```

1 package EntradaSalida;
2
3 import javax.swing.JOptionPane;
4
5 public class ToolsPanel {
6     public static void imprimeSalida(String msje) {
7         JOptionPane.showMessageDialog(null, msje, "Salida de datos", JOptionPane.QUESTION_MESSAGE);
8     }
9     public static void salidaError(String msje) {
10        JOptionPane.showMessageDialog(null, msje, "", JOptionPane.ERROR_MESSAGE);
11    }
12    public static short leerShort(String msje) {
13        return (Short.parseShort(JOptionPane.showInputDialog(null, msje, "Dato de entrada", JOptionPane.INFORMATION_MESSAGE)));
14    }
15    public static byte leerByte(String msje) {
16        return (Byte.parseByte(JOptionPane.showInputDialog(null, msje, "Dato de entrada", JOptionPane.INFORMATION_MESSAGE)));
17    }
18    public static int leerInt(String msje) {
19        return (Integer.parseInt(JOptionPane.showInputDialog(null, msje, "Dato de entrada", JOptionPane.INFORMATION_MESSAGE)));
20    }
21    public static long leerLong(String msje) {
22        return (Long.parseLong(JOptionPane.showInputDialog(null, msje, "Dato de entrada", JOptionPane.INFORMATION_MESSAGE)));
23    }
24    public static float leerFloat(String msje) {
25        return (Float.parseFloat(JOptionPane.showInputDialog(null, msje, "Dato de entrada", JOptionPane.INFORMATION_MESSAGE)));
26    }
27    public static char leerChar(String msje) {
28        return (JOptionPane.showInputDialog(null, msje, "Dato de entrada", JOptionPane.INFORMATION_MESSAGE).charAt(0));
29    }
30    public static String leerString(String msje) {
31        return (JOptionPane.showInputDialog(null, msje, "Dato de entrada", JOptionPane.INFORMATION_MESSAGE));
32    }
33    public static int seguirSino() {
34        return JOptionPane.showConfirmDialog(null, "Deseas continuar", "Capturando datos", JOptionPane.YES_NO_OPTION);
35    }
36    public static int leerEntero(String msje) {
37        return (Integer.parseInt(JOptionPane.showInputDialog(null, msje, "Lectura int: ", JOptionPane.INFORMATION_MESSAGE)));
38    }
39    public static void imprimePantalla(String msje) {
40        JOptionPane.showMessageDialog(null, msje, "Salida", JOptionPane.INFORMATION_MESSAGE);
41    }
42 }

```

El código define una clase llamada "ToolsPanel" que contiene varios métodos estáticos que se pueden usar para interactuar con el usuario mediante ventanas emergentes (pop-up dialogs). Cada método tiene una tarea específica:

- "imprimeSalida" muestra un mensaje en una ventana emergente con un botón de pregunta (QUESTION_MESSAGE).
- "salidaError" muestra un mensaje de error en una ventana emergente con un icono de error (ERROR_MESSAGE).
- "leerShort", "leerByte", "leerInt", "leerLong", "leerFloat" y "leerChar" muestran un mensaje en una ventana emergente con un campo de entrada y un botón de confirmación. Estos métodos convierten la entrada del usuario en un tipo de datos específico (short, byte, int, long, float o char) y lo devuelven como resultado.
- "leerString" es similar a los métodos anteriores, pero devuelve una cadena de caracteres como resultado.

- "seguirSino" muestra un mensaje en una ventana emergente con dos botones (Sí y No) y devuelve el botón que el usuario seleccionó como resultado.
- "leerEntero" es similar a "leerInt", pero muestra un mensaje diferente en la ventana emergente.
- "imprimePantalla" muestra un mensaje en una ventana emergente con un icono de información (INFORMATION_MESSAGE).

Ejercicios

```

1 package Metodos;
2
3 import javax.swing.JOptionPane;
4
5
6
7 public class Ejercicios {
8     public static int numArmstrong(int valor) {
9         int c=0,a;
10        int n=valor;
11        while(n>0) {
12            a=n%10;
13            n=n/10;
14            c=c+(a*a*a);
15        }
16        return c;
17    }
18    public static String binario(int decimal) {
19        String binario=""; int mod; int div=decimal;
20        while (div>=1) {
21            mod=div%2;
22            div=div/2;
23            binario = mod+binario;
24        }
25        return binario;
26    }
27    public static int numDigitos(int valor) {
28        int contador=0; int num=valor;
29        while(num>0) {
30            contador=contador+1;
31            num=num/10;
32        }
33        return contador;
34    }
35    public static int sumapares(int valor) {
36        int sumapar=0;
37        for (int i=1;i<=valor;i++) {
38            if (i%2==0) {
39                sumapar=sumapar+i;
40            }
41        }
42        return sumapar;
43    }
44 }

```

```

44 public static int sumaImpares(int valor, int n) {
45     int sumaImpar=0;
46     for (int i=0; i<=valor; i++) {
47         if (i%2!=0 && i<=n) {
48             sumaImpar=sumaImpar+i;
49         }
50     }
51     return sumaImpar;
52 }
53 public static void sumaDigitos(int valor) {
54     int suma=0;
55     while(valor!=0) {
56         suma+=valor%10;
57         valor/=10;
58     }
59     ToolsPanel.imprimePantalla("Suma de digitos: "+suma);
60 }
61 public static int numPrimo(int valor) {
62     int contador=0;
63     if (valor==1) {
64         contador=0;
65     }
66     else {
67         for(int i=1; i<=valor; i++) {
68             if(valor%i==0) {
69                 contador++;
70             }
71         }
72     }
73     return contador;
74 }
75 public static int sumaDeProductos(int valor1, int valor2) {
76     int producto=0;
77     for(int i=1; i<=valor2; i++) {
78         producto=producto+valor1;
79     }
80     return producto;
81 }

```

```

82 public static String tresDatos(int valor1, int valor2, int valor3) {
83     String a="";
84     if (valor1 == valor2 && valor2 == valor3 && valor3 == valor1) {
85         a="Los enteros "+valor1+", "+valor2+" y "+valor3+" son iguales";
86     }
87     else if (valor1>valor2 && valor1>valor3) {
88         a="El entero "+valor1+" es mayor que los enteros "+valor2+" y "+valor3;
89     }
90     else if (valor1<valor2 && valor2>valor3) {
91         a="El entero "+valor2+" es mayor que los enteros "+valor1+" y "+valor3;
92     }
93     else if (valor1<valor3 && valor2<valor3) {
94         a="El entero "+valor3+" es mayor que los enteros "+valor1+" y "+valor2;
95     }
96     return a;
97 }
98 public static boolean numeroPerfecto(int num) {
99     int sum=0;
100     for(int i=1; i<num; i++) {
101         if(num%i==0) {
102             sum+=i;
103         }
104     }
105     return(sum==num);
106 }

```

```

107 public static double consumoDeAgua(double aguaC) {
108     double pagar=0; double monto=0;
109     if (aguaC<100) {
110         pagar=aguaC*40;
111     }
112     else if(aguaC>=100 && aguaC<=499) {
113         pagar=aguaC*60;
114     }
115     else if(aguaC>=500 && aguaC<=600) {
116         pagar=aguaC*80;
117     }
118     else if(aguaC>600 && aguaC<=1000) {
119         monto=(aguaC*80)*0.02;
120         pagar=(aguaC*80)+monto;
121     }
122     else if(aguaC>1000) {
123         monto=(aguaC*100)*0.02;
124         pagar=(aguaC*100)+monto;
125     }
126     else {
127         JOptionPane.showMessageDialog(null, "Solo se pueden usar numeros enteros");
128     }
129     return pagar;
130 }
131 public static String pagosUniversidad(String nombre, String cat, double prom) {
132     double pago=0,des=0;
133     cat=cat.toUpperCase();
134     switch(cat) {
135         case "A":
136             pago=1200;
137             if(prom>80 && prom<=100) {
138                 des=1200*0.15;
139                 pago=1200-des;
140             }
141             else if(prom>75 && prom<=80) {
142                 des=1200*0.15;
143                 pago=1200-des;
144             }
145             break;

```

```

146         case "B":
147             pago=1000;
148             if(prom>80 && prom<=100) {
149                 des=1000*0.15;
150                 pago=1000-des;
151             }
152             else if(prom>75 && prom<=80) {
153                 des=1000*0.15;
154                 pago=1000-des;
155             }
156             break;
157         case "C":
158             pago=900;
159             if(prom>80 && prom<=100) {
160                 des=900*0.15;
161                 pago=900-des;
162             }
163             else if(prom>75 && prom<=80) {
164                 des=900*0.15;
165                 pago=900-des;
166             }
167             break;
168         case "D":
169             pago=600;
170             if(prom>80 && prom<=100) {
171                 des=600*0.15;
172                 pago=600-des;
173             }
174             else if(prom>75 && prom<=80) {
175                 des=600*0.15;
176                 pago=600-des;
177             }
178             break;
179     }
180     return "Nombre:"+nombre+"\n"+"Pago sin descuento"+des+"\n El pago total es: "+pago;
181 }

```

```

182 public static void Triangulo(int filas) {
183     String cadena=" ";
184     for(byte j=0;j<filas;j++) {
185         for(byte i=0;i<=j;i++) {
186             cadena+=2*i+1+" ";
187         }
188         cadena+="\n";
189     }
190     ToolsPanel.imprimePantalla(cadena);
191 }
192 public static int multiplicacionRusa(int mul,int multi) {
193     int suma=0;
194     while(mul!=1) {
195         if(mul%2==1) {
196             suma+=multi;
197         }
198         mul/=2;
199         multi*=2;
200         System.out.println(mul+" "+multi+"\n");
201     }
202     return multi+suma;
203 }
204 public static String imprimeFrecuencia(byte n) {
205     String cad=" ";
206     for(int i=1;i<=n;i++) {
207         cad+="*";
208     }
209     return cad;
210 }

```

```

211 public static void cuentaVocales(String cadena) {
212     byte a=0,e=0,i=0,o=0,u=0;
213     byte f=0; String cad=" ";
214     while(f<cadena.length()) {
215         switch(cadena.charAt(f)) {
216             case 'A':a++;break;
217             case 'a':a++;break;
218             case 'E':e++;break;
219             case 'e':e++;break;
220             case 'I':i++;break;
221             case 'i':i++;break;
222             case 'O':o++;break;
223             case 'o':o++;break;
224             case 'U':u++;break;
225             case 'u':u++;break;
226         }
227         f++;
228     }
229     cad="a= "+imprimeFrecuencia(a)+"\n"+
230         "e= "+imprimeFrecuencia(e)+"\n"+
231         "i= "+imprimeFrecuencia(i)+"\n"+
232         "o= "+imprimeFrecuencia(o)+"\n"+
233         "u= "+imprimeFrecuencia(u);
234     ToolsPanel.imprimePantalla("Frecuencia de vocales\n"+cad);
235 }
236
237 public static void mayorNum(int dato1,int dato2,int dato3) {
238     int a =Math.max(dato1,dato2);
239     ToolsPanel.imprimePantalla("El número mayor es: "+(Math.max(a, dato3)));
240 }
241 public static boolean palindrome(String cadena) {
242     cadena=cadena.toLowerCase();
243     int i, j=cadena.length()-1;
244     for (i=0;i<=j;i++) {
245         for (j=cadena.length()-1;j>=i;j--) {
246             if(cadena.charAt(i)!=cadena.charAt(j))
247                 return false;
248         }
249     }
250     return true;
251 }
252 }

```


Esta es una clase llamada "Ejercicios" que contiene varios métodos estáticos que realizan diferentes cálculos y devuelven un valor o una cadena. Aquí hay una breve descripción de cada método:

"numArmstrong" toma un valor entero y calcula el número de Armstrong correspondiente, que es un número en el que la suma de los cubos de sus dígitos es igual al número en sí mismo. El método utiliza una estructura de bucle while para descomponer el número en sus dígitos individuales y luego elevar cada dígito al cubo, sumándolos al final. Finalmente, devuelve el número de Armstrong como un entero.

"binario" toma un valor entero y devuelve su representación binaria en forma de una cadena de caracteres. El método utiliza una estructura de bucle while para dividir el número entre 2 y obtener los residuos sucesivamente, mientras que la parte entera de la división se divide nuevamente entre 2. El residuo obtenido se concatena en el lado izquierdo de la cadena de salida "binario", que eventualmente representa el valor binario completo del número de entrada.

"numDigitos" toma un valor entero y cuenta el número de dígitos que contiene el número. El método utiliza una estructura de bucle while para dividir sucesivamente el número entre 10 y contar los cocientes obtenidos hasta que el número se reduce a cero. Finalmente, devuelve la cantidad de dígitos en el número como un entero.

"sumapares" toma un valor entero y devuelve la suma de todos los números pares entre 1 y el valor proporcionado. El método utiliza una estructura de bucle for que recorre todos los números enteros entre 1 y el valor proporcionado. Cada número se evalúa para determinar si es par o no, y si lo es, se suma al resultado total "sumapar". Al final, el método devuelve la suma total de todos los números pares como un entero.

"sumaimpares" toma dos valores enteros y devuelve la suma de los números impares entre 1 y el valor1, incluyendo solo los primeros n números impares. El método utiliza una estructura de bucle for que recorre todos los números enteros entre 1 y el valor proporcionado. Cada número se evalúa para determinar si es impar

y si es menor o igual a n. Si cumple ambas condiciones, se suma al resultado total "sumaImpar". Al final, el método devuelve la suma total de los números impares como un entero.

"sumaDigitos" toma un valor entero y calcula la suma de todos sus dígitos. El método utiliza una estructura de bucle while para dividir sucesivamente el número entre 10 y sumar los residuos obtenidos, que representan los dígitos individuales del número. Finalmente, el método imprime la suma de los dígitos en la pantalla.

"numPrimo" toma un valor entero y devuelve el número de divisores que tiene, lo que determina si es un número primo o no. El método utiliza una estructura de bucle for que recorre todos los enteros entre 1 y el valor proporcionado. Cada número se evalúa para determinar si es un divisor del número de entrada. Si es un divisor, se incrementa el contador. Si el contador finalmente es igual a 2, el número se considera primo y el método devuelve el valor 2. De lo contrario, el método devuelve el número de divisores encontrados.

"sumaDeProductos" toma dos valores enteros y devuelve la suma de la multiplicación de los valores1 por valor2. El método utiliza una estructura de bucle for que recorre todos los enteros entre 1 y el valor2. En cada iteración, el valor1 se suma al producto total, que es el resultado final devuelto como un entero.

"tresDatos" recibe tres números enteros y devuelve una cadena de texto que describe su relación en términos de cuál es el mayor o si son iguales.

"numeroPerfecto" recibe un número entero y devuelve un valor booleano que indica si es un número perfecto o no. Un número es perfecto si la suma de sus divisores propios (números que dividen al número sin dejar residuo, excluyendo al número mismo) es igual al número.

"consumoDeAgua" recibe una cantidad de consumo de agua y devuelve el monto a pagar por la misma, según una escala de tarifas que depende de la cantidad de agua consumida.

“pagosUniversidad” recibe un nombre, una categoría de estudiante y un promedio, y devuelve una cadena de texto que indica el pago a realizar por el estudiante según la categoría y un posible descuento por promedio.

“Triangulo” Este método imprime un triángulo de números. Toma un parámetro filas que indica la cantidad de filas que debe tener el triángulo. Luego utiliza dos bucles for anidados para generar la cadena de números que conforman el triángulo. La variable cadena es donde se va concatenando cada número. El método imprimePantalla es llamado desde la clase ToolsPanel para imprimir la cadena en la pantalla.

“multiplicacionRusa” Este método implementa el algoritmo de la multiplicación rusa. Toma dos parámetros mul y multi que representan los dos números que se van a multiplicar. El método utiliza un bucle while para iterar mientras mul no sea igual a 1. Dentro del bucle, si mul es impar se suma multi a suma. Luego se divide mul entre 2 y se multiplica multi por 2. Finalmente, se devuelve la suma de multi y suma.

“imprimeFrecuencia” Este método genera una cadena de asteriscos que se utiliza para imprimir la frecuencia de cada vocal en el método cuentaVocales. Toma un parámetro n que indica la cantidad de veces que se debe imprimir el asterisco. Luego utiliza un bucle for para concatenar asteriscos en la cadena cad y finalmente devuelve la cadena.

“cuentaVocales” Este método cuenta la cantidad de apariciones de cada vocal en una cadena de texto y muestra la frecuencia de cada vocal en la pantalla. Toma un parámetro cadena que es la cadena de texto a analizar. El método utiliza un bucle while para recorrer cada carácter de la cadena y un switch para contar la cantidad de apariciones de cada vocal. Luego utiliza el método imprimeFrecuencia para generar la cadena de asteriscos correspondiente a cada vocal y finalmente imprime la frecuencia de cada vocal en la pantalla utilizando el método imprimePantalla.

“mayorNum” Este método encuentra el número mayor entre tres números y lo muestra en la pantalla. Toma tres parámetros dato1, dato2 y dato3 que representan los números a comparar. El método utiliza el método Math.max para encontrar el

número mayor y luego lo imprime en la pantalla utilizando el método `imprimePantalla`.

“palindrome” Este método verifica si una cadena de texto es un palíndromo. Toma un parámetro cadena que es la cadena de texto a verificar. El método utiliza dos variables `i` y `j` para recorrer la cadena desde ambos extremos hacia el medio. Si encuentra algún carácter que no coincida entre los extremos, devuelve `false`. Si logra recorrer toda la cadena sin encontrar diferencias, devuelve `true`. Cabe destacar que este método no tiene en cuenta espacios ni signos de puntuación. Además, convierte la cadena a minúsculas para ignorar las diferencias entre mayúsculas y minúsculas.

TestEjercicios:

```
1 package Principal;
2
3 import Metodos.Ejercicios;
4 import EntradaSalida.ToolsPanel;
5
6 public class TestEjercicios {
7     public static void main(String[] args) {
8         // TODO Auto-generated method stub
9         Ejercicios.sumaDigitos(ToolsPanel.LeerEntero("Escribe un valor entero"));
10    }
11 }
```

La clase `TestEjercicios` tiene un método principal (`main`) que es el punto de entrada del programa. Dentro del método `main`, se llama a un método llamado `sumaDigitos` de una clase llamada `Ejercicios` y se le pasa un argumento.

El argumento se obtiene a través de la llamada a otro método llamado `leerEntero` de una clase llamada `ToolsPanel`, que espera que el usuario ingrese un valor entero y devuelve ese valor.

Menu3

```

1 package Principal;
2
3 import javax.swing.JOptionPane;
4
5 public class menu3 {
6     public static String Desplegable(String menu) {
7         String valores[]=menu.split(",");
8         String res=(String)JOptionPane.showInputDialog(null,"M E N U","SELECCIONA TU ENTIDAD: ",
9             JOptionPane.QUESTION_MESSAGE,null,valores,valores[0]);
10        return res;
11    }
12    public static void menu_3(String menu) {
13        String sel="";
14        do {
15            sel=Desplegable(menu);
16            switch(sel){
17                case "Agregar":JOptionPane.showMessageDialog(null, "Caso Agregar");break;
18                case "Imprimir":JOptionPane.showMessageDialog(null, "Caso Imprimir");break;
19                case "Salir":JOptionPane.showMessageDialog(null, "Caso Salir");break;
20            }
21        } while(!sel.equalsIgnoreCase("Salir"));
22    }
23    public static void main(String[] args) {
24        menu_3("Agregar,Imprimir,Salir");
25    }
26 }
27 }

```

La clase "menu3" tiene tres métodos estáticos: "Desplegable", "menu_3" y "main".

El método "Desplegable" recibe una cadena de texto llamada "menu" como argumento y la divide en un arreglo de cadenas utilizando la coma como separador. Luego muestra un cuadro de diálogo de entrada de opción múltiple (JOptionPane) que muestra el texto "M E N U" y presenta al usuario las opciones disponibles en el arreglo de cadenas "valores", que son las opciones del menú. La opción predeterminada es la primera opción en el arreglo "valores[0]". El método devuelve la opción seleccionada por el usuario.

El método "menu_3" utiliza un ciclo "do-while" para mostrar el menú y leer la selección del usuario utilizando el método "Desplegable". Luego, utiliza una declaración "switch" para ejecutar la acción correspondiente a la opción seleccionada por el usuario. En este caso, si el usuario selecciona "Agregar", se muestra un cuadro de diálogo que indica "Caso Agregar"; si selecciona "Imprimir", se muestra un cuadro de diálogo que indica "Caso Imprimir"; y si selecciona "Salir", se muestra un cuadro de diálogo que indica "Caso Salir". El ciclo continúa hasta que el usuario seleccione la opción "Salir", momento en el cual el ciclo termina y el programa también.

El método "main" llama al método "menu_3" y le pasa como argumento la cadena de texto "Agregar,Imprimir,Salir", que son las opciones del menú que se mostrarán

al usuario. Por lo tanto, este programa muestra un menú desplegable al usuario y le permite seleccionar una opción. Dependiendo de la opción seleccionada, se mostrará un mensaje en una ventana emergente, y el programa continuará ejecutándose hasta que el usuario seleccione "Salir".

Menu

```
1 package EntradaSalida;
2
3 import javax.swing.JOptionPane;
4
5
6 public class Menu {
7     public static String Desplegable(String menu) {
8         String valores[]=menu.split(",");
9         String res=(String)JOptionPane.showInputDialog(null,"M E N U","SELECCIONA TU ENTIDAD: ",
10             JOptionPane.QUESTION_MESSAGE,null,valores,valores[0]);
11         return res;
12     }
13     public static void menu_3(String menu) {
14         String sel="";
15         do {
16             sel=Desplegable(menu);
17             switch(sel){
18                 case "Armstrong":
19                     int valorArmstrong=Integer.parseInt(JOptionPane.showInputDialog("Introduce un Entero:"));
20                     if(valorArmstrong==Ejercicios.numArmstrong(valorArmstrong))
21                         JOptionPane.showMessageDialog(null, "Es un numero Armstrong");
22                     else
23                         JOptionPane.showMessageDialog(null, "No es un numero Armstrong");
24                     break;
25                 case "Decimales a Binario":
26                     int valorDecimal = Integer.parseInt(JOptionPane.showInputDialog("Digite un entero: "));
27                     JOptionPane.showMessageDialog(null, Ejercicios.binario(valorDecimal));
28                     break;
29                 case "Contar Digitos":
30                     int valorDigitos = Integer.parseInt(JOptionPane.showInputDialog("Digite un entero:"));
31                     JOptionPane.showMessageDialog(null, "Numero de digitos "+Ejercicios.numDigitos(valorDigitos));
32                     break;
33                 case "Suma de pares":
34                     int primvalor= Integer.parseInt(JOptionPane.showInputDialog("Introduce el primer entero: "));
35                     int limiteN = Integer.parseInt(JOptionPane.showInputDialog("Introduce el limite de la suma de los impares: "));
36                     JOptionPane.showMessageDialog(null, "El entero: "+primvalor+. La suma de sus pares es: "+Ejercicios.sumapares(primvalor)+
37                         " y la suma de sus impares con el limite de "+limiteN+ es "+Ejercicios.sumaImpares(primvalor, limiteN));
38                     break;
39                 case "Numeros Primos":
40                     int ent = Integer.parseInt(JOptionPane.showInputDialog("Introduce un entero: "));
41                     if(Ejercicios.numPrimo(ent)==2) {
42                         JOptionPane.showMessageDialog(null, "El entero "+ent+" es un numero primo.");
43                     }
44                     else {
45                         JOptionPane.showMessageDialog(null, "El entero "+ent+" no es un numero primo.");
46                     }
47                     break;
48                 case "Producto 2 numeros":
49                     int entero1 = Integer.parseInt(JOptionPane.showInputDialog("Digita el primer entero: "));
50                     int entero2 = Integer.parseInt(JOptionPane.showInputDialog("Digita el segundo entero: "));
51                     JOptionPane.showMessageDialog(null, "El producto de los enteros "+entero1+ " y "+entero2+
52                         " es "+Ejercicios.sumaDeProductos(entero1, entero2));
53                     break;
54                 case "Dato mayor":
55                     int val1=ToolsPanel.leerEntero("Dame el primer numero: ");
56                     int val2=ToolsPanel.leerEntero("Dame el segundo numero: ");
57                     int val3=ToolsPanel.leerEntero("Dame el tercer numero: ");
58                     ToolsPanel.imprimePantalla(Ejercicios.tresDatos(val1, val2, val3));
59                     break;
60                 case "Suma digitos":
61                     Ejercicios.sumaDigitos(ToolsPanel.leerEntero("Escribe un valor entero"));
62                     break;
63                 case "Numero Perfecto":
64                     int valor=Integer.parseInt(JOptionPane.showInputDialog("Introduce un entero: "));
65                     JOptionPane.showMessageDialog(null, Ejercicios.numeroPerfecto(valor)+"Si es un numero perfecto":"No es un numero perfecto");
66                     break;
67                 case "Escasez de agua":
68                     double valorAgua=Double.parseDouble(JOptionPane.showInputDialog("Introduce el consumo de agua en metros cúbicos"));
69                     String nombre=JOptionPane.showInputDialog("Nombre del cosumidor:");
70                     JOptionPane.showMessageDialog(null, "El consumidor "+nombre+" debera pagar "+Ejercicios.consumoDeAgua(valorAgua));
71                     break;
72                 case "Pagos Estudiantes":
73                     String nombres=JOptionPane.showInputDialog("Escribe un nombre");
74                     String cat=JOptionPane.showInputDialog("Introduce la categoria");
75                     double prom=Double.parseDouble(JOptionPane.showInputDialog("Introduce tu promedio"));
76                     JOptionPane.showMessageDialog(null, "El estudiante: "+nombres+" debera pagar un monto de "+
77                         Ejercicios.pagosUniversidad(nombres, cat, prom)+" para la pensión");
78                     break;
79                 case "Multiplicación Rusa":
80                     int mul=Integer.parseInt(JOptionPane.showInputDialog("Escribe el divisor: "));
81                     int multi=Integer.parseInt(JOptionPane.showInputDialog("Escribe el multiplicador: "));
82                     JOptionPane.showMessageDialog(null, Ejercicios.multiplicacionRusa(mul,multi));
83                     break;
84                 case "Imprime Frecuencias":
85                     byte n=Byte.parseByte(JOptionPane.showInputDialog("Introduce un numero:"));
86                     JOptionPane.showMessageDialog(null, Ejercicios.imprimeFrecuencia(n));
87                     break;
88                 case "Cuenta Vocales":
89                     String cadena=JOptionPane.showInputDialog("Dame la cadena:");
90                     Ejercicios.cuentaVocales(cadena);
91                     break;
92             }
93         } while (sel != "Salir");
94     }
95 }
```

```
46
47     }
48     break;
49     case "Producto 2 numeros":
50         int entero1 = Integer.parseInt(JOptionPane.showInputDialog("Digita el primer entero: "));
51         int entero2 = Integer.parseInt(JOptionPane.showInputDialog("Digita el segundo entero: "));
52         JOptionPane.showMessageDialog(null, "El producto de los enteros "+entero1+ " y "+entero2+
53             " es "+Ejercicios.sumaDeProductos(entero1, entero2));
54         break;
55     case "Dato mayor":
56         int val1=ToolsPanel.leerEntero("Dame el primer numero: ");
57         int val2=ToolsPanel.leerEntero("Dame el segundo numero: ");
58         int val3=ToolsPanel.leerEntero("Dame el tercer numero: ");
59         ToolsPanel.imprimePantalla(Ejercicios.tresDatos(val1, val2, val3));
60         break;
61     case "Suma digitos":
62         Ejercicios.sumaDigitos(ToolsPanel.leerEntero("Escribe un valor entero"));
63         break;
64     case "Numero Perfecto":
65         int valor=Integer.parseInt(JOptionPane.showInputDialog("Introduce un entero: "));
66         JOptionPane.showMessageDialog(null, Ejercicios.numeroPerfecto(valor)+"Si es un numero perfecto":"No es un numero perfecto");
67         break;
68     case "Escasez de agua":
69         double valorAgua=Double.parseDouble(JOptionPane.showInputDialog("Introduce el consumo de agua en metros cúbicos"));
70         String nombre=JOptionPane.showInputDialog("Nombre del cosumidor:");
71         JOptionPane.showMessageDialog(null, "El consumidor "+nombre+" debera pagar "+Ejercicios.consumoDeAgua(valorAgua));
72         break;
73     case "Pagos Estudiantes":
74         String nombres=JOptionPane.showInputDialog("Escribe un nombre");
75         String cat=JOptionPane.showInputDialog("Introduce la categoria");
76         double prom=Double.parseDouble(JOptionPane.showInputDialog("Introduce tu promedio"));
77         JOptionPane.showMessageDialog(null, "El estudiante: "+nombres+" debera pagar un monto de "+
78             Ejercicios.pagosUniversidad(nombres, cat, prom)+" para la pensión");
79         break;
80     case "Multiplicación Rusa":
81         int mul=Integer.parseInt(JOptionPane.showInputDialog("Escribe el divisor: "));
82         int multi=Integer.parseInt(JOptionPane.showInputDialog("Escribe el multiplicador: "));
83         JOptionPane.showMessageDialog(null, Ejercicios.multiplicacionRusa(mul,multi));
84         break;
85     case "Imprime Frecuencias":
86         byte n=Byte.parseByte(JOptionPane.showInputDialog("Introduce un numero:"));
87         JOptionPane.showMessageDialog(null, Ejercicios.imprimeFrecuencia(n));
88         break;
89     case "Cuenta Vocales":
90         String cadena=JOptionPane.showInputDialog("Dame la cadena:");
91         Ejercicios.cuentaVocales(cadena);
92         break;
93     }
94 } while (sel != "Salir");
95 }
```

```

67         case "Escasez de agua":
68             double valorAgua=Double.parseDouble(JOptionPane.showInputDialog("Introduce el consumo de agua en metros cúbicos"));
69             String nombre=JOptionPane.showInputDialog("Nombre del consumidor:");
70             JOptionPane.showMessageDialog(null, "El consumidor "+nombre+" debera pagar "+Ejercicios.consumoDeAgua(valorAgua));
71             break;
72         case "Pagos Estudiantes":
73             String nombreEs=JOptionPane.showInputDialog("Escribe un nombre");
74             String cat=JOptionPane.showInputDialog("Introduce la categoria");
75             double prom=Double.parseDouble(JOptionPane.showInputDialog("Introduce tu promedio"));
76             JOptionPane.showMessageDialog(null, "El estudiante: "+nombreEs+" debera pagar un monto de "+
77                 Ejercicios.pagosUniversidad(nombreEs, cat, prom)+" para la pensión");
78             break;
79         case "Multiplicación Rusa":
80             int mul=Integer.parseInt(JOptionPane.showInputDialog("Escribe el divisor: "));
81             int multi=Integer.parseInt(JOptionPane.showInputDialog("Escribe el multiplicador: "));
82             JOptionPane.showMessageDialog(null, Ejercicios.multiplicacionRusa(mul,multi));
83             break;
84         case "Imprime Frecuencias":
85             byte n=Byte.parseByte(JOptionPane.showInputDialog("Introduce un numero:"));
86             JOptionPane.showMessageDialog(null, Ejercicios.imprimeFrecuencia(n));
87             break;
88         case "Cuenta Vocales":
89             String cadena=JOptionPane.showInputDialog("Dame la cadena:");
90             Ejercicios.cuentaVocales(cadena);
91             break;
92         case "Salir":
93             JOptionPane.showMessageDialog(null, "Hasta luego, que tenga un buen día");
94             break;
95         default: {ToolsPanel.salidaError(sel);
96             }
97     }
98 }
99 while(!sel.equalsIgnoreCase("Salir"));
100 }
101 public static void main(String[] args) {
102     menu_3("Armstrong,Decimales a Binario,"
103         + "Contar Digos,Suma de pares,"
104         + "Numeros Primos,Producto 2 numeros,"
105         + "Suma digitos,Numero Perfecto,"
106         + "Escasez de agua,Pagos Estudiantes,"
107         + "Multiplicación Rusa,Imprime Frecuencias,"
108         + "Cuenta Vocales,Salir");
109 }
110 }

```

Esta clase está definiendo un menú con varias opciones, donde cada opción está asociada con una funcionalidad específica proporcionada por algún método en la misma clase. El menú se muestra mediante un cuadro de diálogo JOptionPane con una lista desplegable de opciones. Una vez que se selecciona una opción, se llama al método correspondiente para realizar la funcionalidad asociada. El menú sigue mostrándose hasta que el usuario selecciona la opción "Salir" (Salir).

El método Desplegable toma un argumento de cadena que contiene la lista de opciones separadas por comas para el menú y devuelve la opción seleccionada por el usuario.

El método menu_3 es responsable de mostrar el menú, recibir la entrada del usuario y llamar al método apropiado según la selección del usuario. Utiliza un do-whilebucle para mantener el menú en ejecución hasta que el usuario decida salir.

Los diversos métodos asociados a las opciones del menú realizan las siguientes funcionalidades:

numArmstrong: comprueba si un entero dado es un número de Armstrong

binario: convierte un entero decimal a binario

numDigitos: cuenta el número de dígitos en un entero

sumapares y sumaimpares: calcula la suma de números pares y la suma de números impares hasta un límite dado, respectivamente

numPrimo: comprueba si un entero dado es un número primo

sumaDeProductos: calcula el producto de dos enteros dados

tresDatos: encuentra el mayor de tres enteros dados

sumaDigitos: calcula la suma de los dígitos de un entero dado

numeroPerfecto: comprueba si un entero dado es un número perfecto

consumoDeAgua: calcula la factura del agua para un consumo dado de agua

pagosUniversidad: calcula las tarifas universitarias para un estudiante en función de su categoría y promedio de calificaciones

multiplicacionRusa: realiza la multiplicación de dos números enteros utilizando el método campesino ruso

imprimeFrecuencia: imprime la frecuencia de ocurrencia de cada dígito en un entero dado

cuentaVocales: cuenta el número de vocales en una cadena dada

Conclusiones:

Finalmente para cada problema planteado se busco su solucion con lo aprendido en el aula y en fuentes externas, para cada solucion se le hicieron sus respectivas pruebas y atravez de la depuracion se solucionaron aquellos problemas ya sea en tiempo de ejecucion, sintaxis o de logica.

En particular, las practicas desarrolladas en esta primera unidad han sido de gran utilidad para comprender la estructura y el proceso de desarrollo de aplicaciones.

Bibliografia:

ING. INFORMATICA TEMA 1 POO. (2023). Padlet.

<https://padlet.com/mtzcastillo2023/ing-informatica-tema-1-poo-h4fb110wakh25lze>

TEMARIO. (2023). Padlet. <https://padlet.com/mtzcastillo2023/ing-informatica-encuadre-poo-2nnwhmi63h3nizu7/wish/2456404770>

en, P. (2023). Programación ATS [YouTube Video]. In *YouTube*.

<https://www.youtube.com/@ProgramacionATS>

de, C. (2023). pildorasinformaticas [YouTube Video]. In *YouTube*.

<https://www.youtube.com/@pildorasinformaticas>

KevinDCCsHeOs. (2023, April 12). *KevinDCCsHeOs/Proyecto1*. GitHub.

<https://github.com/KevinDCCsHeOs/Proyecto1>