



**Asignatura:** **Programación Orientada a objetos**

**Carrera:** **ingeniería en informática**

**Docente:** **MARIA JACINTA MARTINEZ CASTILLO**

**Nombre del tema:** **Entorno del desarrollo**

**Nombres:** **Kevin Alan Ortiz Flores**  
**Ángel de Jesús Contreras Herrera**

**Numero De control:** **21010207**

**21010180**

**Grupo:** **2a3B**

**Fecha de entrega:** **17 /febrero/ 2023**

## **Introducción**

En este reporte hablaremos de lo más básico de la programación, como instalar el IDE Eclipse o NetBeans así como su respectivo JDK para el funcionamiento del IDE. Así como también hablaremos de como se compone una clase y los métodos, dándonos a entender cual es su funcionamiento y en que momento usarlas y el uso de exportaciones de clase por que será algo que ocuparemos a futuro.

Conoceremos 3 tipos de menús, Menú despegable que en una ventana emergente nos saldrá como una lista despegable de todos los datos que le hayamos puesto para una mayor facilidad de selección, Menú de botones que en una ventana emergente nos saldrá las diferentes opciones que le hayamos puesto en forma de botones, pero no es muy recomendada cuando son mas de 3 opciones y el menú básico, que solo muestra el mensaje de texto con la lista de las opciones que existen, es poco recomendable por que el usuario deberá ingresar el texto de lo que desea realizar y si no lo escribe bien o escribe otra cosa nos podría marcar un error.

Veremos como tener organizado nuestros proyectos con diferentes paquetes y plantillas que usaremos a lo largo de nuestra carrera, para tener trabajos mas limpios y organizados dando una mejor apariencia y facilitando el entendimiento de los códigos.

Conoceremos la plantilla de entradas y salidas llamadas Tools, que nos facilitara y ahorrara el uso de un system.out o un JOptionPane.

Así como también crear un paquete para realizar todos los métodos que realicemos evitando hacer todo dentro del main(), aprenderemos diferentes métodos y formas de resolver problemas, como también aprendernos los rangos de cada tipo de variables y su correspondiente espacio de memoria que ocupa para razonar que tipo de variable es correcta o “mejor” para cada código

### **Competencia especifica:**

Conoce y aplica el modularidad en el desarrollo de programas para la optimización de los mismos y reutilización de código



## Marco teórico:

### 1.1 Estructura de un proyecto:

1. Directorio del proyecto: El nombre del directorio principal debe estar relacionado con el nombre del proyecto. Por ejemplo, si el nombre del proyecto es "MiAppJava", el nombre del directorio principal podría ser "proyecto-miappjava".
  2. Directorio src: Este directorio debe contener todas las fuentes del proyecto, es decir, archivos .java.
  3. Directorio bin: Este directorio contendrá los archivos compilados del proyecto, es decir, los archivos .class.
  4. Directorio lib: Este directorio contendrá todas las librerías externas utilizadas por el proyecto. Los archivos de librerías suelen estar en formato jar.
  5. Directorio doc: Este directorio contendrá toda la documentación relativa al proyecto.
  6. Otros directorios: Puede haber otros directorios, dependiendo del proyecto. Por ejemplo, un proyecto web puede contener un directorio "web", que contendrá archivos relacionados con la interfaz web.
1. En general, la estructura del proyecto puede variar dependiendo del entorno de desarrollo utilizado y las convenciones utilizadas en cada proyecto en particular.

## 1.2 Proceso de desarrollo de aplicaciones:

El proceso de desarrollo de aplicaciones Java normalmente sigue los siguientes pasos:

1. **Análisis y diseño:** Se debe realizar un análisis detallado de los requisitos del usuario y diseñar una solución adecuada. En esta etapa se definen los detalles de la aplicación, como las funcionalidades, el flujo de trabajo, la arquitectura, etc.
2. **Desarrollo de la aplicación:** Una vez definidos los requerimientos y el diseño, se comienza a desarrollar el código de la aplicación. Esto incluye la implementación de la lógica de negocio, la creación de interfaces de usuario, la integración con bases de datos y otros sistemas, etc.
3. **Pruebas:** Después de haber desarrollado la aplicación, es importante realizar pruebas exhaustivas para garantizar que funcione correctamente en diferentes escenarios. Esto incluye pruebas funcionales, de rendimiento, de seguridad, de compatibilidad, etc.
4. **Depuración y corrección de errores:** Después de las pruebas, es probable que se detecten errores o fallas en la aplicación, que deben ser depurados y corregidos antes de la implementación.
5. **Implementación:** Una vez que se ha probado y depurado la aplicación, se puede implementar en un entorno de producción. Esto puede ser en un servidor de aplicaciones, en la nube, o en otros dispositivos según sea necesario.
6. **Mantenimiento:** Finalmente, es importante realizar el mantenimiento de la aplicación para garantizar su correcto funcionamiento a largo plazo. Esto incluye la reparación de errores, la actualización de funciones y características, la mejora del rendimiento, la optimización de la seguridad, y otros aspectos similares.

### 1.2.1 Compilación:

La compilación es el proceso de transformar el código fuente en un programa ejecutable por la máquina.

### 1.2.2 Archivos que crea el entorno de desarrollo:

Durante el proceso de desarrollo, el entorno de desarrollo genera diversos archivos, como archivos de objeto, bibliotecas y ejecutables.

### **1.2.3 Depuración:**

Esto ayuda a garantizar que el programa funcione según lo previsto y que no haya problemas o errores que puedan provocar un mal funcionamiento o fallos en el programa. La depuración es una parte esencial del desarrollo de software, ya que los errores pueden ser difíciles de encontrar sin herramientas adecuadas, y esto puede provocar problemas para los usuarios finales. La depuración también ayuda a mejorar la calidad del código fuente, lo que a su vez mejora la eficiencia y fiabilidad del programa completo.

### **Material y Equipo:**

✓ Computadora Windows 8.1 pro con 4 GB de RAM y procesador Intel Pentium CPU J2900 2.41 GHz

✓ JDK 15.0.2 – NetBeans IDE 8.2 y Eclipse IDE 2021-09

✓ Videos del canal [Programación ATS](#)

### **Desarrollo de la practica:**

1. Descargar y entrar a nuestro IDE favorito para crear nuevos proyectos.
2. Crearemos un proyecto llamado Tema1POO
3. Con este proyecto haremos 3 tipos de paquetes llamados: Tools, Métodos y Test o Menú
4. En el paquete Tools crearemos una clase donde ira la plantilla de entradas y salidas que tendrá como nombre EntradaSalida
5. En el paquete TDA tendrá todas las clases que obtendrá métodos de cada ejercicio, así como también importaremos la clase EntradaSalida del paquete Tools
6. En el paquete Test o Menú es donde crearemos nuestro menú seleccionado y es donde llamaremos a todos los métodos creado para que el usuario decida que operación desea realizar

## Resultados:

```
1 package Metodos;
2
3 public class Conversion {
4     public static String decimalBinario(int val)
5     {
6         String cad="";
7         if(val>0)
8         {
9             while (val>0){
10                 if(val%2==0)
11                     cad += "0";
12                 else
13                     cad+="1";
14                 val= val/2;
15             }
16         } else if(val==0)
17             cad="";
18         else
19             cad="Imposible convertir";
20         return cad;
21     }
22 }
23
24
25
```

La clase contiene un método llamado "decimalBinario" que realiza la conversión de un valor entero en base decimal a su equivalente en base binaria.

```
Conversion.java x Ejercicio.java x
Source History
1
2 package Metodos;
3
4 import EntradaSalida.Tools;
5
6 public class Ejercicio {
7
8     public static boolean numArmstrong(int valor){
9
10         int aux=valor, sum=0;
11
12         while(aux>0){
13             sum+=Math.pow(aux%10, 3);
14             aux/=10;
15         }
16
17         return(sum== valor);
18     }
19     public static void sumaDigitos(int valor)
20     {
21         int suma=0;
22         while(valor!=0)
23         {
24             suma+=valor%10;
25             valor/=10;
26         }
27         Tools.imprimePantalla("Suma de digitos:"+suma);
28     }
29 }
30
```

El primer método, "numArmstrong", verifica si un valor entero dado es un número de Armstrong. Un número de Armstrong es aquel cuya suma de los cubos de sus dígitos es igual al valor original. El segundo método, "sumaDigitos", calcula la suma de los dígitos de un valor entero dado.

```
Ejercicio1.java x
Source History
1 package Metodos;
2
3 public class Ejercicio1 {
4     public static String cliente(String nCliente){
5
6         return(nCliente);
7     }
8
9     public static double montoAdeudo(double consumo){
10
11         double adeudo = 0;
12         if(consumo <100)
13             adeudo = consumo * 40;
14         else
15             if (consumo>=100 && consumo<500)
16                 adeudo = (consumo - 100) * 60 + 4000;
17             else
18                 if (consumo>=500 && consumo<=1000)
19                     adeudo = (consumo - 499) * 80 + 33940;
20                 else
21                     if (consumo>1000)
22                         adeudo = (consumo - 1000)* 100 + 113940;
23             if (adeudo > 600)
24                 adeudo = (adeudo* .02) + adeudo;
25             return(adeudo);
26         }
27     }
28 }
```

El primer método, "cliente", toma un nombre de cliente como parámetro y lo devuelve como resultado. El segundo método, "montoAdeudo", calcula el monto adeudado de acuerdo a un consumo de energía eléctrica dado.



```
Ejercicio2.java
Source History
1 package Metodos;
2
3 public class Ejercicio2 {
4     public static String cliente(String nEstudiante){
5
6         return(nEstudiante);
7     }
8
9     public static double calculaAdeudo(double prom, String cat){
10        double adeudo = 0;
11        int pago = 0;
12
13        if(cat.equalsIgnoreCase("A"))
14            pago = 1200;
15        else
16            if(cat.equalsIgnoreCase("B"))
17                pago = 1000;
18            else
19                if(cat.equalsIgnoreCase("C"))
20                    pago = 900;
21                else
22                    if(cat.equalsIgnoreCase("D"))
23                        pago = 600;
24        if(prom >80 && prom<=100)
25            adeudo = pago- (pago*.15);
26        else
27            if(prom >75 && prom<=80)
28                adeudo=pago - (pago*.08);
29
30        return(adeudo);
31    }
32 }
```

El primer método, "cliente", toma un nombre de estudiante como parámetro y lo devuelve como resultado. El segundo método, "calculaAdeudo", calcula el adeudo de un estudiante en función de su promedio y una categoría específica.

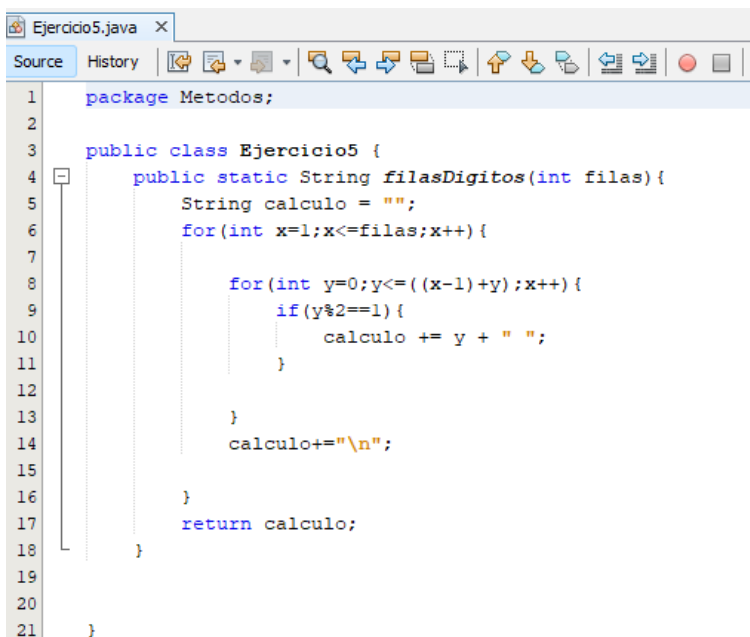
```
Ejercicio3.java X
Source History
1 package Metodos;
2
3 public class Ejercicio3 {
4     public static boolean calculaPerfecto(int numero){
5
6         boolean nPerfecto = false;
7         int suma = 0;
8         for (int i = 1; i < numero; i++) {
9             if (numero % i == 0) {
10                 suma = suma + i;
11             }
12         }
13         if(suma==numero)
14             nPerfecto = true;
15
16         return(nPerfecto);
17     }
18 }
19
```

El método se llama "calculaPerfecto" y toma un número como parámetro. El objetivo del método es determinar si el número dado es un número perfecto o no.

Un número perfecto es un número entero positivo que es igual a la suma de sus divisores propios positivos, excluyendo a sí mismo. Por ejemplo, 6 es un número perfecto porque sus divisores propios (1, 2 y 3) suman 6.

```
Ejercicio4.java X
Source History
1 package Metodos;
2
3 public class Ejercicio4 {
4     public static int multiplicacionRusa(int num1, int num2){
5         int multiplicacion=0;
6         while(num1!=0){
7             if(num1 % 2 != 0){
8                 multiplicacion = multiplicacion + num2;
9             }
10
11             num1 =num1 / 2;
12             num2 = num2 * 2;System.out.println(num1+" "+num2+"\n");
13         }
14         return multiplicacion;
15     }
16 }
17
```

El método se llama "multiplicacionRusa" y toma dos números enteros como parámetros: "num1" y "num2". El objetivo del método es realizar una multiplicación utilizando el algoritmo conocido como "multiplicación rusa" o "doble y suma".

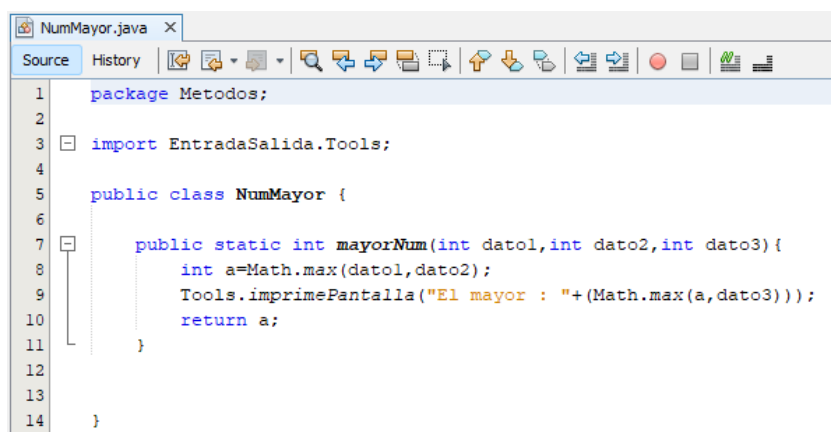


```
1 package Metodos;
2
3 public class Ejercicio5 {
4     public static String filasDigitos(int filas){
5         String calculo = "";
6         for(int x=1;x<=filas;x++){
7
8             for(int y=0;y<=((x-1)+y);x++){
9                 if(y%2==1){
10                     calculo += y + " ";
11                 }
12             }
13             calculo+="\n";
14         }
15         return calculo;
16     }
17 }
18
19
20
21 }
```

El método se llama "filasDigitos" y toma un número entero "filas" como parámetro. El objetivo del método es generar una serie de filas con dígitos impares, separados por espacios, en forma de un patrón triangular.

```
EjercicioClase.java x
Source History
1 package Metodos;
2
3 import static Metodos.frecuencia.imprimeFrecuencia;
4
5 public class EjercicioClase {
6     public static String cuentaVocales(String cadena){
7         byte a=0,e=0,i=0,o=0,u=0;
8         byte f=0;
9         String cad="";
10        while(f<cadena.length()){
11            switch(cadena.charAt(f)){
12
13                case 'A' :
14                case 'a' : a++;break;
15                case 'E' :
16                case 'e' : e++;break;
17                case 'I' :
18                case 'i' : i++;break;
19                case 'O' :
20                case 'o' : o++;break;
21                case 'U' :
22                case 'u' : u++;break;
23            }
24            f++;
25        }
26        cad="a="+ imprimeFrecuencia(a)+"\n"+"e="+ imprimeFrecuencia(e)+"\n"+
27            "i="+ imprimeFrecuencia(i)+"\n"+"o="+ imprimeFrecuencia(o)+"\n"+
28            "u="+ imprimeFrecuencia(u)+"\n";
29        return cad;
30    }
31 }
32
33 }
```

El método se llama "cuentaVocales" y toma una cadena de caracteres "cadena" como parámetro. El objetivo del método es contar la frecuencia de cada una de las vocales (a, e, i, o, u) en la cadena de caracteres y retornar una cadena con los resultados.



```
1 package Metodos;
2
3 import EntradaSalida.Tools;
4
5 public class NumMayor {
6
7     public static int mayorNum(int dato1,int dato2,int dato3){
8         int a=Math.max(dato1,dato2);
9         Tools.imprimePantalla("El mayor : "+(Math.max(a,dato3)));
10        return a;
11    }
12
13
14 }
```

El método se llama "mayorNum" y toma tres enteros como parámetros: "dato1", "dato2" y "dato3". El objetivo del método es determinar y mostrar en pantalla el mayor de los tres números utilizando la función "Math.max" de la clase "Math" en Java.

```
NumPrimo.java x
Source History
1 package Metodos;
2
3 import EntradaSalida.Tools;
4
5 public class NumPrimo {
6     public static void esPrimo(int numero)
7     {
8         boolean primo=false;
9         for(int i=2; i<numero; i++)
10             if(numero%i==0)
11                 primo=false;
12         else
13             primo=true;
14
15         if(primo)
16             Tools.imprimePantalla("Es un numero primo");
17         else
18             Tools.imprimePantalla("No es un numero primo");
19     }
20 }
21 }
```

El método se llama "esPrimo" y toma un entero llamado "numero" como parámetro. El objetivo del método es determinar si el número ingresado es primo o no.

```
Palindrome.java x
Source History
1 package Metodos;
2
3 public class Palindrome {
4     public static boolean esPalindromo(String palabra)
5     {
6         palabra = palabra.toLowerCase().replace(" ", "").replace(",","");
7         int cont = 0, inverso = palabra.length()-1;
8         boolean palindromo = false;
9
10        while ((cont<inverso) && (!palindromo)){
11            if (palabra.charAt(cont) == palabra.charAt(inverso)){
12                cont++;
13                inverso--;
14            } else {
15                palindromo = true;
16            }
17        }
18        return palindromo;
19    }
20 }
```

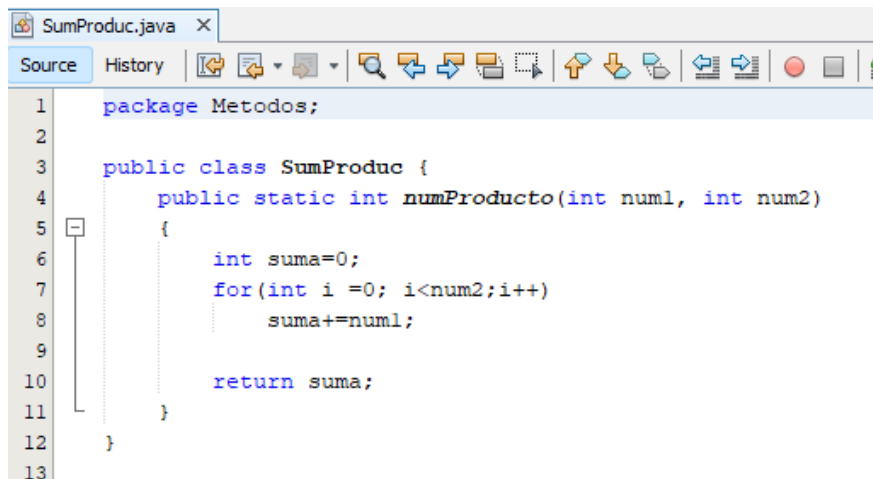
El método se llama "esPalindromo" y toma una cadena de caracteres llamada "palabra" como parámetro. El objetivo del método es determinar si la palabra ingresada es un palíndromo o no, y devuelve un valor booleano que indica si es un palíndromo o no.

```
Pintura.java x
Source History
1 package Metodos;
2
3 public class Pintura {
4     public static double Pintura(double base,double altura){
5         double m2=0;
6         double litros=0;
7
8         m2=base*altura;
9         litros=m2/3;
10
11         return (litros*67);
12
13     }
14 }
15
16
17
18 }
```

El método se llama "Pintura" y toma dos parámetros de tipo double: "base" y "altura", que representan las dimensiones de una superficie a pintar.

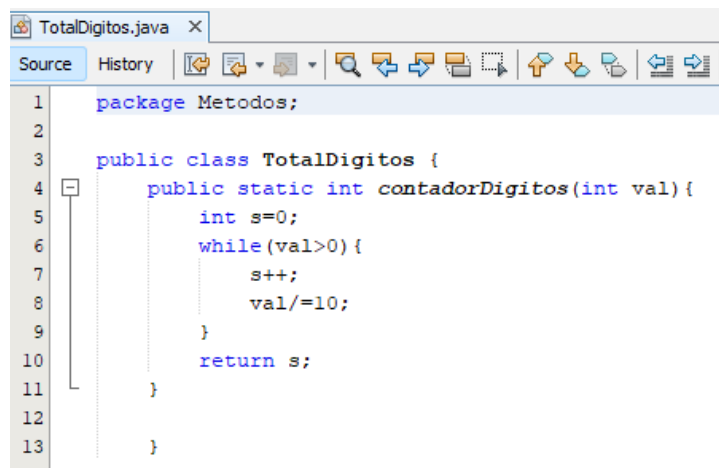
```
SumNum.java x
Source History
1 package Metodos;
2
3 import EntradaSalida.Tools;
4
5 public class SumNum {
6     public static void sumParImpar(int cont)
7     {
8         int i=1, sImpares=0, sPares=0;
9         while(i<=cont){
10             if(i%2==0)
11                 sPares+=i;
12             else
13                 sImpares+=i;
14             i++;
15         }
16         Tools.imprimePantalla("La suma de los numeros pares es de " + sPares +
17                               " y la suma de los numeros impares es de " + sImpares);
18     }
19 }
20 }
```

El método se llama "sumParImpar" y toma un parámetro de tipo entero llamado "cont", que representa el número máximo hasta el cual se deben sumar los números pares e impares.



```
1 package Metodos;
2
3 public class SumProduc {
4     public static int numProducto(int num1, int num2)
5     {
6         int suma=0;
7         for(int i =0; i<num2;i++)
8             suma+=num1;
9
10        return suma;
11    }
12 }
13
```

El método se llama "numProducto" y toma dos parámetros de tipo entero llamados "num1" y "num2", que representan dos números enteros que se multiplicarán para obtener el producto.



```
1 package Metodos;
2
3 public class TotalDigitos {
4     public static int contadorDigitos(int val){
5         int s=0;
6         while(val>0){
7             s++;
8             val/=10;
9         }
10        return s;
11    }
12 }
13
```



El método se llama "contadorDigitos" y toma un parámetro de tipo entero llamado "val", que representa el valor del cual se contarán los dígitos.

```
ValEnteros.java x
Source History
1 package Metodos;
2
3 import EntradaSalida.Tools;
4
5 public class ValEnteros {
6     public static void contNumeros(int num1, int num2, int num3)
7     {
8         if(num1==num2 && num2==num3)
9             Tools.imprimePantalla("Los tres numeros son iguales");
10        else if(num1==num2 && num3>num1)
11            Tools.imprimePantalla("Los primeros dos numeros son iguales y el numero " + num3 + " es el mayor");
12        else if(num1==num3 && num2>num1)
13            Tools.imprimePantalla("El primer y tercer numero son iguales y el numero " + num2 + " es el mayor");
14        else if(num3==num2 && num1>num2)
15            Tools.imprimePantalla("El segundo y tercer numero son iguales y el numero " + num1 + " es el numero mayor");
16        else if (num1>num2 && num2>num3)
17            Tools.imprimePantalla("El numero " + num1 + " es el mayor");
18        else if (num1<num2 && num2>num3)
19            Tools.imprimePantalla("El numero " + num2 + " es el mayor");
20        else if (num1<num2 && num2<num3)
21            Tools.imprimePantalla("El numero " + num3 + " es el mayor");
22    }
23 }
```

El método se llama "contNumeros" y toma tres parámetros de tipo entero: "num1", "num2" y "num3", que representan los tres números enteros que se compararán.

```
VentaCubre bocas.java X
Source History
1 package Metodos;
2
3 public class VentaCubre bocas {
4     public static int VenCubre bocas(int cantidad){
5         int costo=0;
6
7         if(cantidad>10)
8             costo=100;
9         else
10            if(cantidad>=5&&cantidad<=10)
11                costo=120;
12        else
13            if(cantidad<=4)
14                costo=150;
15
16        return (cantidad*costo);
17    }
18 }
19
20
--
```

El método se llama "VenCubre bocas" y toma un parámetro de tipo entero llamado "cantidad", que representa la cantidad de cubrebocas que se van a vender.

```
frecuencia.java X
Source History
1 package Metodos;
2
3 public class frecuencia {
4     public static String imprimeFrecuencia(byte n){
5         String cad=" ";
6         for(int i=1;i<=n;i++){
7             cad+="*";
8         }
9         return cad;
10    }
11 }
12
```

El método se llama "imprimeFrecuencia" y toma un parámetro de tipo byte llamado "n", que representa el número de veces que se imprimirá el carácter "\*" en la cadena resultante.

```
1
2 package Principal;
3 import javax.swing.*;
4 import EntradaSalida.Tools;
5 import Metodos.Conversion;
6 import Metodos.Ejercicio;
7 import Metodos.Ejercicio1;
8 import Metodos.Ejercicio2;
9 import Metodos.Ejercicio3;
10 import Metodos.Ejercicio4;
11 import Metodos.Ejercicio5;
12 import Metodos.EjercicioClase;
13 import Metodos.NumMayor;
14 import Metodos.TotalDigitos;
15 import Metodos.ValEnteros;
16 import Metodos.NumPrimo;
17 import Metodos.Palindrome;
18 import Metodos.SumNum;
19 import Metodos.SumProduc;
20
21
22 public class testEjercicio {
23
24     public static void main(String [] args){
25
26         String menu="Numero Armstrong,Suma digitos,Monto Pagar,Pagos Estudiante,"
27             + "Numero Perfecto,Multiplicacion Rusa,"
28             + "Lista Impar,Frecuencia,Numero Mayor,Palindrome,Conversion,"
29             + "Total Digitos,Val Enteros,Num Primo,Sum Num,Sum Produc,Salir";
30         menu3(menu);
31
32     }
```

```
33
34 public static String boton(String menu) {
35
36     String valores[] = menu.split(",");
37     int n;
38
39     n = JOptionPane.showOptionDialog(null, " Selecciona Dando Click",
40         "M E N U",
41         JOptionPane.NO_OPTION,
42         JOptionPane.QUESTION_MESSAGE, null,
43         valores, valores[0]);
44
45
46     return (valores[n]);
47
48 }
49 public static void menu3(String menu) {
50
51     String cliente = "", estudiante = "", impresion = "";
52     double cant=0;
53     int multi =0;
54     String sel= "";
55     String cad="";
56     int a;
57     int num1,num2;
58     int N,contador = 0;
59
60
61
62     do{
63         sel= boton(menu);
64         switch(sel){
65             case "Numero Armstrong": {boolean nArms = Ejercicio.numArmstrong
66                 (Tools.leerEntero("Ingresa el numero entero"));
67                 if(nArms)
68                     Tools.imprimePantalla("Es un numero Armsrong");
69                 else
70                     Tools.imprimePantalla("No es un numero Armstrong");
71                 }break;
72             case "Suma digitos": Ejercicio.sumaDigitos(Tools.leerEntero
73                 ("Ingresa el numero entero"));
74             break;
75             case "Monto Pagar": {
76                 cliente = Ejerciciol.cliente(Tools.leerString
77                     ("Ingresa el nombre completo del cliente a capturar"));
78                 cant = Ejerciciol.montoAdeudo(Tools.leerDouble
79                     ("Ingresa la cantidad de agua utilizada en metros cubicos"));
80                 Tools.imprimePantalla("El cliente : " + cliente +
81                     " tiene un adeudo de: " + cant + " pesos.");
82             }
```

```
82     }
83     break;
84     case "Pagos Estudiante":{
85         estudiante = Ejercicio2.cliente(Tools.leerString
86             ("Ingrese el nombre completo del estudiante"));
87         cant = Ejercicio2.calculaAdeudo(Tools.leerDouble
88             ("Ingrese el promedio del estudiante"),
89             Tools.leerString
90             ("Ingrese la categoria en la que se encuentra el estudiante"));
91         Tools.imprimePantalla("El estudiante : " + estudiante +
92             " tiene un adeudo de: " + cant + " pesos.");
93     }
94     ;
95     break;
96     case ",Numero Perfecto":
97         if(Ejercicio3.calculaPerfecto(Tools.leerEntero
98             ("Ingresa el numero a calcular")))
99             Tools.imprimePantalla("El numero es perfecto");
100        else Tools.imprimePantalla("El numero no es perfecto"); ;
101    break;
102    case "Multiplicacion Rusa":
103        multi = Ejercicio4.multiplicacionRusa(Tools.leerEntero
104            ("Ingresa el multiplicador"), Tools.leerEntero
105            ("Ingresa el multiplicando"));
106        Tools.imprimePantalla
107            ("El resultado de la multiplicacion ha sido : " + multi);
108    break;
109    case "Lista Impar":{
110
111        impresion = Ejercicio5.filasDigitos(Tools.leerEntero
112            ("Ingresa el numero de filas"));
113        Tools.imprimePantalla(impresion);
114    } ;
115
116    break;
```

```
117
118    case "Frecuencia":{
119        cad=EjercicioClase.cuentaVocales(Tools.leerString
120            ("Ingresa una cadena"));
121        Tools.imprimePantalla(cad);
122    }
123    ;
124
125    break;
126
127    case "Numero Mayor" : {
128
129        a= NumMayor.mayorNum(Tools.leerEntero("Ingrese dato1"),
130            Tools.leerEntero("Ingresa dato2"),Tools.leerEntero("Ingresa dato3"));
131    }
132
133    break;
134
135
```

```
135
136
137     case "Palindrome" : {
138         if(Palindrome.esPalindromo(Tools.leerString
139             ("Ingrese una palabra para saber si es palindromo")))
140             Tools.imprimePantalla("La palabra es palindromo");
141         else
142             Tools.imprimePantalla("La palabra no es un palindromo");
143     }
144     break;
145
146     case "Conversion" : {
147         Tools.imprimePantalla("El valor binario es: " +
148             Conversion.decimalBinario(Tools.leerEntero
149             ("Ingrese el valor que desea convertir a binario")));
150     }
151     break;
152
153     case "Total Digos" : {
154         Tools.imprimePantalla("El numero de digitos es: " +
155             TotalDigos.contadorDigitos(Tools.leerEntero
156             ("Ingrese un numero.")));
157     }
158     break;
159
160     case "Val Enteros" : {
161         ValEnteros.contNumeros(Tools.leerEntero
162             ("Ingrese el primer numero"),Tools.leerEntero
163             ("Ingrese el segundo numero"),Tools.leerEntero
164             ("Ingrese el tercer numero"));
165     }
166
167
168 }
```

```
169         case "Num Primo" : {
170             NumPrimo.esPrimo(Tools.leerEntero
171             ("Ingresa un numero para determinar si es un numero primo"));
172         }
173         case "Sum Num" : {
174             SumNum.sumParImpar(Tools.leerEntero("Ingresa un numero"));
175         }
176         case "Sum Produc" : {
177             Tools.imprimePantalla("El producto de los numeros es de: "+
178             SumProduc.numProducto(Tools.leerEntero
179             ("Ingresa el primero numero"),Tools.leerEntero
180             ("Ingresa el segundo numero")));
181         }
182         case "Salir": ;
183         break;
184     }
185     }while(!sel.equalsIgnoreCase("Salir"));
186 }
187 }
188 }
189 }
190 }
191 }
192 }
```

Una clase llamada "testEjercicio" con un método principal "main" que muestra un menú con varias opciones.

Las opciones del menú incluyen diferentes métodos de diferentes clases como "Ejercicio", "Ejercicio1", "Ejercicio2", "Ejercicio3", "Ejercicio4", "Ejercicio5", "EjercicioClase", "NumMayor", "Conversion", "TotalDigitos", "ValEnteros", "NumPrimo", "Palindrome", "SumNum" y "SumProduc", que contienen funcionalidades específicas implementadas en los métodos correspondientes.

## Conclusiones:

Aprendimos y repasamos lo básico de la programación en que momento usar cada tipo de variable, tener de forma ordenada los paquetes de un proyecto para un mayor entendimiento y sobre todo una platilla útil que se puede utilizar llamada Tools, así como también aprendimos 3 tipos de menús muy útiles que se pueden usar para los programas, pero el mas recomendado es el menú despegable ya que es más fácil y estético.

Aprendimos que no todo se hace dentro del main(), y aprendimos a como realizar diferentes métodos y como retonar los resultados, aprendimos importar clases ya que es muy útil cuando ocupas más de una clase y lo más importan aprendimos a realizar corrida de escritorio, esto es muy importante para entender el funcionamiento del código y verificar errores lógicos que pueda tener un programa a la hora de ejecutarlo

## Bibliografía:



- *Proceso de desarrollo de aplicaciones.* (s. f.). Scribd.

<https://es.scribd.com/document/480719592/1-2-Proceso-de-desarrollo-de-aplicaciones>

- Garro, A. (s. f.-a). Estructura de un programa | Java.

<https://www.arkaitzgarro.com/java/capitulo-2.html>