



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

---

*Profesor: Guadalupe Lizeth Parrales Romay*

*Asignatura: Programación Orientada a Objetos*

*Grupo: 4*

*No. De práctica(s): 3*

*Integrante(s): Miguel Adán Vargas López*

*No. de Equipo de cómputo empleado: Ninguno*

*Semestre: 2021-2*

*Fecha de entrega: 03/10/21*

*Observaciones:* \_\_\_\_\_

**CALIFICACIÓN:** \_\_\_\_\_

# Introducción

Para realizar un programa, ya sea de un reto grande o chico, siempre hay que buscar la mejor solución y si se puede que sea simple de realizar, de implementar y de entender para demás programadores. Hoy en día, la comunidad de desarrolladores esta bien comunicada y tiene buena documentación de trabajos ya realizados y esto conlleva a que haya librerías, paquetes de códigos ya realizado que buscan simplificar tareas comunes, para evitar que tengamos que inventar todo desde cero. Esta practica busca realizar programas que involucre trabajar con diferentes clases para realizar la tarea.

## Desarrollo de práctica

2.- Desarrolle un programa en Java que implemente la aproximación a la función seno para un ángulo  $x$  en grados mediante un polinomio de Taylor.

Para este programa habrá tres funciones, la principal Main, una llamada factorial que regresa la factorial de su parámetro y otra llamada toSenX que calculará la función sen para la variable introducida por el usuario guardada en la variable grados. Por otro lado, en otra variable guardamos el variable de los grados convertidos a radianes para así comenzar a trabajar.

```
*/
import java.util.Scanner;

public class ej2{

    public static void main(String[] s_2){
        double gradosRadianes,resultado,grados;

        //lectura de grados
        Scanner valor = new Scanner(System.in);
        System.out.printf("Digite Valor Grados: ");
        grados = valor.nextDouble();
        gradosRadianes = (grados*3.1416)/180;
        resultado = toSenx(gradosRadianes);
        System.out.println("Sen(" + grados + ") = " + resultado);
    }

    public static int factorial(double n){
        int f = 1;
        for(int i = 1; i <= n; i++){
            f*=i;
        }
        return f;
    }
}
```

```

22     for(int i = 1; i <= n; i++){
23         f*=i;
24     }
25     return f;
26 }
27 public static double toSenx(double gradosRadianes){
28     int c = 1;
29     double resultado;
30     double sumaNegativa= 0.0f, sumaPositiva = 0.0f;
31     double potencia1 = 1;
32     double potencia2 = 3;
33     do{
34         sumaPositiva += Math.pow(gradosRadianes,potencia1)/factorial(potencia1);
35         potencia1 += 4;
36         sumaNegativa -= Math.pow(gradosRadianes,potencia2)/factorial(potencia2);
37         potencia2 += 4;
38         c++;
39     }while(c<4);
40     resultado = sumaNegativa + sumaPositiva;
41     return resultado;
42 }
43 }

```

Para la función toSenx nos basamos en la siguiente formula:

$$\text{sen } x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots + (-1)^n \frac{x^{2n+1}}{(2n+1)!} + \dots, \forall x \in \mathbb{R}$$

Donde según vemos, a cada termino va cambiando el signo, por lo cual una parte de la suma es negativa y otra positiva, acorde a las potencias, creamos dos variables que llevaran el control de esto, una inicializada en 1 y otra en 3, que son las potencias de los dos primeros términos de la serie, pero cada uno representa a diferente suma, para que llegue otro termino, el exponente se suma en 4 es por ello que dentro de la iteración le incrementamos en 4 a las variables usadas para esto('potencia1' y 'potencia2'), por último, una vez terminada la sumatoria solo quedaría sumar la parte positiva de la sumatoria con la negativa y la función se encargara de devolver dicho resultado.

Algunos ejemplos ejecutados en consola son:

```

C:\Users\adan_\Desktop\Tester\Java\Practica3>java ej2.java
Digite Valor Grados: 90
Sen(90.0) = 0.9999999437325973

C:\Users\adan_\Desktop\Tester\Java\Practica3>java ej2.java
Digite Valor Grados: 50
Sen(50.0) = 0.7660457548072163

```

3.- Investigue la clase Properties del paquete java.util y desarrolle un programa que al ejecutarse obtenga las características del sistema (equipo de cómputo) en el que se ejecuta y muestre en pantalla dichas características.

#### Clase properties

Es una clase representada por una tabla hash, donde su llave y valor son strings, puede ser útil para tener guardado en un FileStream las preferencias establecidas por un usuario, como si quiere tener la música activada, si dejó algunas configuraciones ya establecidas puedan ser cargadas fácilmente de la clase Properties

Para mostrar en consola el nombre del sistema operativo, su versión y su arquitectura, accedamos a través de la clase System y el método getProperty y los almacenamos en variables String para solo imprimirlos.

```
4 import java.util.Properties;
5
6 public class ej3{
7
8     public static void main(String[] s_2){
9         String sistemaOperativo = System.getProperty("os.name");
10        String userName = System.getProperty("user.name");
11        String sistemaOperativoVersion = System.getProperty("os.version");
12        String sistemaOperativoArch = System.getProperty("os.arch");
13
14        System.out.println("Sistema Operativo: " + sistemaOperativo);
15        System.out.println("Sistema Operativo Version: " + sistemaOperativoVersion);
16        System.out.println("Sistema Operativo Arquitectura: " + sistemaOperativoArch);
17        System.out.println("Usuario del sistema: " + userName);
18    }
19 }
20 }
```

```
C:\Users\adan_\Desktop\Tester\Java\Practica3>java ej3.java
Sistema Operativo: Windows 10
Sistema Operativo Version: 10.0
Sistema Operativo Arquitectura: amd64
Usuario del sistema: adan_
C:\Users\adan_\Desktop\Tester\Java\Practica3>
```

4.- Explique a detalle el funcionamiento de los programas ReferenciasWrappers.java y Fechas.java.

#### *ReferenciasWrappers.java*

*El objetivo de este programa es sumar dos cantidades, ya sea con datos de tipo entero o floats.*

```
9      System.out.println("El valor de x antes de la suma es: " + x);
10
11      suma(x,y);
12
13      System.out.println("El valor de x despues de la suma es: " + x);
14
15      Float alpha, beta;
16
17      //alpha = 3.456;
18      //beta = 64.368;
19
20      alpha = 3.456f;
21      beta = 64.368f;
22
23      System.out.println("\nEl valor de alpha antes de la suma es: " + alpha);
24
25      suma(alpha, beta);
26
27      System.out.println("El valor de alpha despues de la suma es: " + alpha);
28
29  }
30
31  public static void suma(Integer a, Integer b){
32      a += b;
33      System.out.println("El resultado de la suma es: " + a);
34  }
35
36  public static void suma(Float a, Float b){
37      a += b;
38      System.out.println("El resultado de la suma es: " + a);
39  }
40 }
```

Las dos funciones son similares en operación y algoritmo, pero tener las dos funciones nos sirven para evitar excepciones y que desde la parte del Main se pueda trabajar para obtener la suma de dos variables tanto int como float, asimismo, en ambos métodos, el valor de la suma busca que se vea almacenado en la variable que se pasa primero como argumento tomando en cuenta que sería (a += b) pero al ser dos tipos de datos primitivos no se guarda en la variable original.

El resultado en consola de este programa es el siguiente:

```
C:\Users\adan\Desktop\Tester\Java\Practica3>java Referenc
El valor de x antes de la suma es: 5
El resultado de la suma es: 31
El valor de x despues de la suma es: 5

El valor de alpha antes de la suma es: 3.456
El resultado de la suma es: 67.824
El valor de alpha despues de la suma es: 3.456
```

#### *Fechas.java*

Este programa busca ejemplificar 3 formas diferentes de obtener la fecha desde un programa de java, importando diferentes librerías en las que hay diferentes métodos y por lo cual diferentes formas para obtener la fecha actual.

Para esto se usa la clase `Date` y `Calendar` procedentes de `java.util` y la clase `LocalDate` de `java.time`. Antes de trabajar en cada caso en el programa imprimimos en consola para especificar a quien ejecute el programa, de cual clase de las que se trabajaron es la que se usa en esa parte del código.

Instanciamos un objeto de cada una de las clases y al hacer eso, podemos acceder a propiedades y métodos de esa clase, como en el caso de la clase `Date`, creamos una instancia y con el método `toString()` obtenemos dicha cadena y con la clase `SimpleDateFormat` la pasamos a un formato más familiar o conocido y lo imprimimos.

```
6  public class fechas{
7      public static void main(String[] args) {
8
9          System.out.println("Fechas con la clase Date:");
10         Date fecha = new Date();
11         System.out.println("La fecha es: " + fecha.toString());
12         SimpleDateFormat formato = new SimpleDateFormat("dd-MM-yyyy");
13         System.out.println("La fecha es: " + formato.format(fecha));
14
15         System.out.println("\nFechas con la clase Calendar:");
16
17         String miFecha = "Hoy es dia ";
18
19         Calendar calendario = Calendar.getInstance();
20
21         miFecha += calendario.get(Calendar.DAY_OF_MONTH) + " del mes ";
22         miFecha += calendario.get(Calendar.MONTH) + 1 + " de ";
23         miFecha += calendario.get(Calendar.YEAR);
24
25         System.out.println(miFecha);
26
27         System.out.println("\nFechas con la clase LocalDate:");
28
29         LocalDate hoy = LocalDate.now();
30         System.out.println(hoy);
31         System.out.println(hoy.plusWeeks(1));
32     }
33 }
```

Para la clase `Calendar`, igualmente creamos una instancia de esa clase y con un `get()` podemos acceder a valores respecto al tiempo tomando en cuenta a los atributos de la clase `Calendar`. Por último, la clase `LocalDate` tiene su instancia que al ser inicializada se ejecutó un método de la clase `LocalDate`, `now()` y por tanto la variable 'hoy' puede almacenar el tiempo actual y hasta realizar funciones como `plusWeeks` para tener almacenada en esa variable el tiempo actual más una semana ,solo es necesario imprimirlo en consola.

El resultado en consola de este programa es el siguiente:

```
C:\Users\adan\Desktop\Tester\Java\Practica3>java fechas.java
Fechas con la clase Date:
La fecha es: Sun Oct 03 20:34:45 CDT 2021
La fecha es: 03-10-2021

Fechas con la clase Calendar:
Hoy es día 3 del mes 10 de 2021

Fechas con la clase LocalDate:
2021-10-03
2021-10-10
```

## Conclusión

La practica de esta semana ya empieza a elevar el nivel de los algoritmos a realizar respecto a las pasadas y sobre todo deja en claro la importancia de conocer con que lenguaje trabajamos, que facilidades nos proporciona y también saber comprender, así como buscar la documentación disponible sobre el mismo para poder sacarle todas las ventajas posibles.

Ya que, según lo realizado, podemos ver que incluso para realizar una tarea simple como obtener la fecha actual, hay diferentes formas, en este caso vimos que podíamos importar tres clases distintas y cada una nos ayudaba a obtener este dato, pero de diferente forma, nos arrojaba diferentes formatos y también eran diferentes métodos los que había que usar de estas clases.

Al final queda en claro una frase que siempre esta presente, en trabajos, escuelas, deportes, etc. Pero que muchas veces pasamos por alto o no le creemos.

“Lo difícil, es hacer lo sencillo”

# Bibliografía

*Martín, Antonio*

***Programador Certificado Java 2.***

*Segunda Edición.*

*México*

*Alfaomega Grupo Editor, 2008*

*Sierra Katy, Bates Bert*

***SCJP Sun Certified Programmer for Java 6 Study Guide***

*Mc Graw Hill*

*Dean John, Dean Raymond.*

***Introducción a la programación con Java Primera Edición.***

*México*

*Mc Graw Hill, 2009*

Cortés, J. J., González, M. E., & Pinilla, V. D. (2019). *Métodos de descomposición LU: Crout y*

*Doolittle*. Plataforma Educativa Para Análisis Numérico,.

[https://www.ingenieria.unam.mx/pinilla/PE105117/pdfs/tema3/3-2\\_descomposicion\\_lu.pdf](https://www.ingenieria.unam.mx/pinilla/PE105117/pdfs/tema3/3-2_descomposicion_lu.pdf)

Oracle. (2021). *System Properties (The Java™ Tutorials > Essential Java Classes > The Platform Environment)*. Java Documentation.

<https://docs.oracle.com/javase/tutorial/essential/environment/sysprop.html>