

Clase: Programación Orientada a Objetos
Alumno: Santiago Daniel Zamora Solís
Número de cuenta: 217924-0
Practica : 9

MainFile

```
public class MainFile {  
  
    public static void main(String args[]) throws Exception {  
  
        DecimalFormat df = new DecimalFormat( pattern: "###.####");  
        double desvEstandar = 1f, media = 0f, x = 0f;  
        // Creas el scanner del input  
        Scanner scanner = new Scanner(System.in);  
  
        try{  
            System.out.println("Ingrese la media");  
            media = scanner.nextDouble();  
  
            System.out.println("Ingrese la desviacion Estandar");  
            desvEstandar = scanner.nextDouble();  
  
            System.out.println("Ingrese x");  
            x = scanner.nextDouble();  
  
            FdensidadNormal calc1 = new FdensidadNormal(media, desvEstandar);  
            System.out.println("El valor evaluado es: " + df.format(calc1.calcularaValorFDN(x)));  
            System.out.println("El valor de probabilidad es: " + calc1.calcularProbabilidadFDN( limitInf: 2, limitSup: 11, delta: 0.001));  
        }catch(Exception e){  
            System.out.println("Tienen que ser datos numericos");  
            System.out.println(e.toString());  
        }  
    }  
}
```

Clase FdensidadNormal

```
public class FdensidadNormal {
    private double media;
    private double desviacionEstandar;

    FdensidadNormal(double media, double desviacionEstandar) throws Exception {
        this.media = media;

        if(desviacionEstandar == 0)
            throw new ArithmeticException("La desviacion estandar no puede ser 0");

        this.desviacionEstandar = desviacionEstandar;
    }

    /**
     * @param x
     * @return (double) el calculo de la funcion de densidad normal
     */
    double calculaValorFDN(double x){

        double e1 = -0.5 * (Math.pow(x - this.media, 2) / Math.pow(desviacionEstandar, 2));
        return (1 / (this.desviacionEstandar * Math.sqrt(2 * Math.PI))) * Math.exp(e1);
    }
}
```

```
/**
 * @param x
 * @return (double) el calculo de la funcion de densidad normal
 */
double calculaValorFDN(double x){

    double e1 = -0.5 * (Math.pow(x - this.media, 2) / Math.pow(desviacionEstandar, 2));
    return (1 / (this.desviacionEstandar * Math.sqrt(2 * Math.PI))) * Math.exp(e1);
}

/**
 * @param limitInf
 * @param limitSup
 * @param delta
 * @return (double) el calculo la Probabilidad de FDN
 */
double calcularProbabilidadFDN(double limitInf, double limitSup, double delta){

    double res = 0;
    for (double i = limitInf; i <= limitSup; i += delta)
        res += calculaValorFDN(i) * delta;

    return res;
}
}
```

Salida

```
Ingrese la media
a
Tienen que ser datos numericos
java.util.InputMismatchException

Process finished with exit code 0
|
```

```
Ingrese la media
1
Ingrese la desviacion Estandar
0
Ingrese x
12
Tienen que ser datos numericos
java.lang.ArithmeticException: La desviacion estandar no puede ser 0

Process finished with exit code 0
```