

202111718

# Manual Técnico

Walter Javier Santizo  
Mazariegos

walterjav19@gmail.com

---

## **Introducción**

Para las organizaciones, registrar los eventos económicos, financieros y administrativos para luego analizarlos es de gran importancia. Debido a esto, disponer de elementos gráficos que puedan presentar al usuario de manera amigable es especialmente útil. Estos datos antes de ser mostrados deben ser estructurados de tal forma que favorezcan su análisis. Por lo tanto se desarrolla una aplicación que permita leer archivos csv que contengan información importante para graficar en una gráfica de barras

## Métodos Usados

```
public String obtenerTexto(String ruta) { //path o ruta del textfield2
    String texto = "";

    try {
        BufferedReader br = new BufferedReader(new FileReader(ruta));
        String linea;
        while((linea = br.readLine()) != null) { //leer
            //área de lectura línea por línea, estructura: categoria,valor [Salto de línea] pollo,10
            String columnas[] = linea.split(",");
            texto += columnas[0] + "," + columnas[1] + "\n";
        }
        br.close(); //cerrar el flujo y liberar recursos
    } catch(IOException e) {
        //en caso de error
    }

    return texto;
}
```

Activar

Obtener la ruta del archivo con el cual se leerá y colocaran los datos necesarios para la grafica de barras

```
Cronometro(JLabel cronometro) {
    this.eti=cronometro;
}

public void run(){
    try{
        int x=0;

        while(Orden.iniciohilo){
            Thread.sleep(1);
            ejecutarHiloCronometro(x);
            x++;
        }
    } catch(Exception e){
        System.out.println("error");
    }
}
```

El hilo que llevara a cabo la acción de aumentar la variable x con el fin de imitar un cronometro

```

private void ejecutarHiloCronometro(int x) {
    Orden.milisegundo++;

    if(Orden.milisegundo>399){
        Orden.milisegundo=0;
        Orden.segundo++;
    }
    if(Orden.segundo>59){
        Orden.segundo=0;
        Orden.minuto++;
    }
    if(Orden.minuto>59){
        Orden.minuto=0;
        Orden.hora++;
    }

    String txtsegundo="",txtmin="",txthora="",txtmilisegundo="";

    if(Orden.milisegundo<10){
        txtmilisegundo="0"+Orden.milisegundo;
    }else{
        txtmilisegundo="" + Orden.milisegundo;
    }
    if(Orden.segundo<10){
        txtsegundo="0"+Orden.segundo;
    }
}

```

aquí reside toda la parte visual del cronómetro que almacena el contador x++ y que se colocará el texto en el label indicado el orden minutos:segundos:milisegundos

```

private String categoria;
private int valor;

public Dato (String categoria, String valor) {
    this.categoria = categoria;
    this.valor = Integer.parseInt(valor);
}

/**
 * @return the categoria
 */
public String getCategoria() {
    return categoria;
}

/**
 * @param categoria the categoria to set
 */
public void setCategoria(String categoria) {
    this.categoria = categoria;
}

```

El constructor necesario para que los datos usados en la grafica se almacenen en el atributo del constructor

```

public int getValor() {
    return valor;
}

/**
 * @param valor the valor to set
 */
public void setValor(int valor) {
    this.valor = valor;
}

```

Métodos setters y getters para buscar y retornar los valores de la grafica

```

public void burbujaAscendente(int[] arreglo) {
    int limite = arreglo.length;
    int auxiliar;

    for(int i = 0; i < limite; i++) {
        for(int j = 0; j < limite - 1; j++) {
            if(arreglo[j] > arreglo[j + 1]) {
                auxiliar = arreglo[j];
                arreglo[j] = arreglo[j + 1];
                arreglo[j + 1] = auxiliar;
            }
        }
    }
}

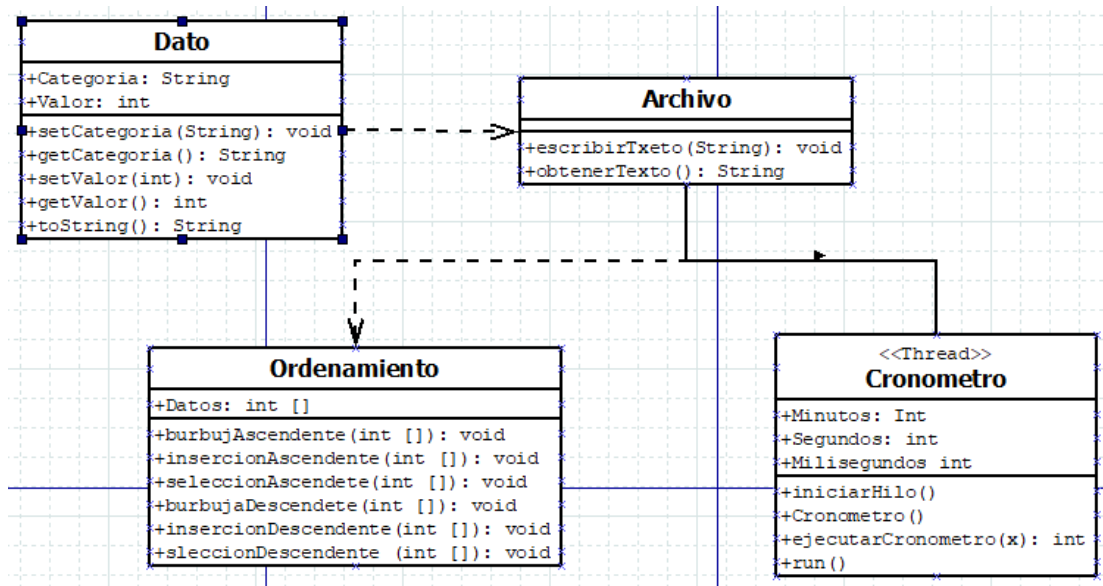
public void seleccionAscendente(int[] arreglo) {
    int limite = arreglo.length;
    int auxiliar, indiceMinimo;

    for(int i = 0; i < limite; i++) {
        indiceMinimo = i;
        for(int j = i + 1; j < limite; j++) {
            if(arreglo[j] < arreglo[indiceMinimo]) {
                indiceMinimo = j;
            }
        }
    }
}

```

Todos los métodos que reciben el arreglo de números los cuales van a ser re ordenados para usarlo en el nuevo ordenamiento

## Diagrama de Clases



## Diagrama de Casos de Uso

