

# UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS

Universidad del Perú, Decana de América

Facultad de Ingeniería de Sistemas e Informática



## EJERCICIOS CON TABLAS DE SÍMBOLOS

### GRUPO DE TRABAJO 3

**ASIGNATURA:** Estructura de Datos II

#### INTEGRANTES:

- |                                    |          |
|------------------------------------|----------|
| • ALEJO CARNICA, Bryan Martin      | 17200256 |
| • CHÁVEZ SILUPÚ, Erick Alberto     | 17200267 |
| • CUELLO APAZA, Alexander Gabriel  | 17200269 |
| • CORDOVA SANDOVAL, Rafael Anthony | 17200268 |

**DOCENTE:** Javier Antonio Prudencio Vidal.

**ESCUELA ACADÉMICO-PROFESIONAL:** Ingeniería de Software.

LIMA - PERÚ  
2022

## Ciente de prueba de una Tabla de símbolos para seguimiento

Crea la tabla de símbolos asociando el valor *i* con la *i*-ésima cadena de la entrada estándar.

```
public static void main(String[] args)
{
    ST<String, Integer> st = new ST<String, Integer>();
    for (int i = 0; !StdIn.isEmpty(); i++)
    {
        String key = StdIn.readString();
        st.put(key, i);
    }
    for (String s : st.keys())
        StdOut.println(s + " " + st.get(s));
}
```

```
keys  S E A R C H E X A M P L E
values 0 1 2 3 4 5 6 7 8 9 10 11 12
```

output

```
A 8
C 4
E 12
H 5
L 11
M 9
P 10
R 3
S 0
X 7
```

Ejercicio: Crear un programa similar que permita el ingreso por teclado de los caracteres, o que recibe el nombre del archivo por teclado.

java ST < tinyST.txt

16

## CLASE MAIN PARA LEER CARACTERES POR CONSOLA

```
public class TSimbolos16 {

    static Scanner entrada = new Scanner(System.in);

    public static void main(String[] args) {
        Map<String,Integer>obj = new TreeMap();
        int opc,contador=0;
        char caracter;
        System.out.println("===MENU DE OPCIONES===");
        System.out.println("1)INGRESAR CHARACTER");
        System.out.println("2)MOSTRAR LISTA");
        System.out.println("3)SALIR");
        do{
            System.out.print("\nINGRESE SU OPCION :");
            opc = entrada.nextInt();
            switch(opc){
                case 1:
                    System.out.print("Ingrese el caracter :");
                    caracter = entrada.next().charAt(0);
                    obj.put(Character.toString(caracter),contador);  contador++; break;
                case 2:
                    System.out.println("\033[32mCLAVE\tVALOR\033[37m");
                    for( HashMap.Entry<String,Integer> i: obj.entrySet()){
                        System.out.println(i.getKey()+"\t"+i.getValue());
                    } break;
                case 3: System.out.println("FIN PROGRAMA"); break;
                default: System.out.println("\033[31mOPCION ERRONEA\033[37m"); break;
            } }while(opc!=3);
    }
```

## EJECUCION DEL CODIGO .JAVA

```
Output - ED2 (run) x
run:
===MENU DE OPCIONES===
1) INGRESAR CARACTER
2) MOSTRAR LISTA
3) SALIR

INGRESE SU OPCION :1
Ingrese el caracter :S

INGRESE SU OPCION :1
Ingrese el caracter :E

INGRESE SU OPCION :1
Ingrese el caracter :A

INGRESE SU OPCION :1
Ingrese el caracter :R

INGRESE SU OPCION :1
Ingrese el caracter :C

INGRESE SU OPCION :1
Ingrese el caracter :H

INGRESE SU OPCION :1
Ingrese el caracter :E

INGRESE SU OPCION :1
Ingrese el caracter :X

Output - ED2 (run) x
Ingrese el caracter :X

INGRESE SU OPCION :1
Ingrese el caracter :A

INGRESE SU OPCION :1
Ingrese el caracter :M

INGRESE SU OPCION :1
Ingrese el caracter :P

INGRESE SU OPCION :1
Ingrese el caracter :L

INGRESE SU OPCION :1
Ingrese el caracter :E

INGRESE SU OPCION :2
CLAVE  VALOR
A      8
C      4
E     12
H      5
L     11
M      9
P     10
R      3
S      0
X      7
```

## LÁMINA 18:

### Implementación del contador de frecuencias

```
public class FrequencyCounter
{
    public static void main(String[] args)
    {
        int minlen = Integer.parseInt(args[0]);
        ST<String, Integer> st = new ST<String, Integer>();
        while (!StdIn.isEmpty())
        {
            String word = StdIn.readString();
            if (word.length() < minlen) continue;
            if (!st.contains(word)) st.put(word, 1);
            else
                st.put(word, st.get(word) + 1);
        }
        String max = "";
        st.put(max, 0);
        for (String word : st.keys())
            if (st.get(word) > st.get(max))
                max = word;
        StdOut.println(max + " " + st.get(max));
    }
}
```

Ejercicio: Crear un programa similar que acepte el nombre del archivo por teclado.

Crear ST

Leer cadena y frecuencia de actualización

Imprimir la cadena con max freq

18

```
import java.util.*;
import java.io.*;
```

```
public class FrequencyCounter {
    public static void main(String[] args) throws FileNotFoundException {
        Scanner console = new Scanner(System.in);
        System.out.println("Nombre de archivo: ");
        String archivo = console.nextLine();
        Scanner input = new Scanner(new File(archivo));
        System.out.println("Longitud minima de la palabra: ");
        int MinLen = console.nextInt();
        Map<String, Integer> ContarFrecuencia = new HashMap<>();
        while(input.hasNext()){
            String palabra = input.next().toLowerCase();
            if(MinLen <= palabra.length()){
                if(!ContarFrecuencia.containsKey(palabra)){
                    ContarFrecuencia.put(palabra, 1);
                } else{
                    int conteo = ContarFrecuencia.get(palabra);
                    ContarFrecuencia.put(palabra, conteo + 1);
                }
            }
        }
        String max = "";
        ContarFrecuencia.put(max, 0);
        for(String palabra : ContarFrecuencia.keySet())
            if(ContarFrecuencia.get(palabra) > ContarFrecuencia.get(max))
                max = palabra;
        System.out.println("La palabra es: "+max+" "+ContarFrecuencia.get(max));
    }
}
```

**EJECUCIÓN DEL CÓDIGO**  
**ARCHIVO tinyTale.txt**

run:

Nombre de archivo:

tinyTale.txt

Longitud minima de la palabra:

1

La palabra es: it 10

**BUILD SUCCESSFUL (total time: 5 seconds)**

**ARCHIVO tale.txt**

run:

Nombre de archivo:

tale.txt

Longitud minima de la palabra:

8

La palabra es: business 122

**BUILD SUCCESSFUL (total time: 4 seconds)**

**ARCHIVO leipzig1M.txt**

run:

Nombre de archivo:

leipzig1M.txt

Longitud minima de la palabra:

10

La palabra es: government 25702

**BUILD SUCCESSFUL (total time: 28 seconds)**