

## UNIDAD TEMÁTICA 2: Introducción a JAVA, PARTE 2

### TRABAJO DE APLICACIÓN 1

#### Ejercicio # 1

**Referencias:** tutoriales y material “*Manejo básico de lectura / escritura de archivos*” publicados.

Se solicita **extender** las funcionalidades del Contador de Palabras desarrollado en la Unidad anterior. Las nuevas funcionalidades son:

- Retornar un **vector** de Strings con las líneas de texto contenidas en un archivo dado, e indicar la cantidad de líneas leídas.
- Dado un vector de Strings que contiene las líneas de texto, retornar la cantidad de palabras contenidas.

#### Firmas que deben tener los métodos solicitados

```
class ContadorPalabras {  
  
...  
  
    public String[] obtenerLineas (String archivo);  
    public int cantPalabras (String[] lineasArchivo);  
  
}
```

#### A realizar (finaliza 20:30):

1. Describir en lenguaje natural las nuevas funcionalidades.
2. Seleccionar, dentro del Equipo, la mejor implementación de UT1\_TA3 Ejercicio #2
3. Crear un nuevo paquete “TA1” dentro del paquete “UT2”
4. Implementar las funcionalidades solicitadas, modificando la clase “**ContadorPalabras**” anteriormente desarrollada. Poner la nueva clase (y toda otra que se desarrolle) en el paquete “TA1” de UT2. Revisar la documentación de la API de JAVA para las clases y métodos utilizados!!!!.
5. Desarrollar una clase “**Principal**” capaz de probar las nuevas funcionalidades a través del método “**main**”. En este método, se ha de instanciar un **ContadorPalabras** y emitir por consola la cantidad de palabras de un archivo cuyo nombre se pasa por parámetro de línea de comando (revisar la documentación del método **main**). Se proveerá un archivo de texto de ejemplo en la webasignatura.
6. Remitir todo el código generado en un archivo zip (solamente los archivos fuente, “\*.java”) hasta la **hora 20:30 en la tarea UT2\_TA1**

## Ejercicio # 2 – PRUEBAS Y JUNIT:

Se desea implementar los casos de prueba necesarios para la clase **ContadorPalabras** que se ha desarrollado, y a la vez documentar el código recientemente construido, utilizando JAVADOC.

### Procedimiento a realizar por cada Equipo:

#### **PARTE 1: DESCRIPCIÓN EN LENGUAJE NATURAL (hasta la hora 20:50):**

1. El Equipo se dividirá en 2 subequipos.
  - a. El subequipo **A** desarrolla en **lenguaje natural** la especificación para el **caso de prueba** correspondiente a la funcionalidad de contar palabras en una string (debe luego subirse a la tarea correspondiente)
  - b. El subequipo **B** desarrolla en **lenguaje natural** la especificación para el **caso de prueba** correspondiente a la funcionalidad de leer el archivo de texto (que retorna el vector de strings con las correspondientes líneas) (debe luego subirse a la tarea correspondiente)
2. Los subequipos se intercambian y revisan las especificaciones.

#### **PARTE 2: PROGRAMACIÓN Y PRUEBAS (entrega hasta la hora 21:05 en la tarea UT2\_TA1)**

3. Documentación del código y creación de los casos de prueba:
  - a. SUBEQUIPO A:
    - i. genera en JAVA-JUNIT el caso de prueba para la funcionalidad de leer el archivo de texto, basándose en la especificación realizada por el subequipo B y agrega JAVADOC al mismo.
  - b. SUBEQUIPO B:
    - i. genera en JAVA-JUNIT el caso de prueba para la funcionalidad de contar palabras de una string, basándose en la especificación realizada por el subequipo A.
4. Ejecutar los casos de prueba y reparar errores.
5. Los subequipos se intercambian y revisan los casos de prueba creados por el otro subequipo

**ENTREGA:** TODO EL CODIGO Y CASOS DE PRUEBA en la **tarea UT2\_TA1**, hasta la hora 21:10

### Ejercicio # 3

Dada la palabra indicada como “palabra1”, reemplazar todas las ocurrencias de la misma en el archivo fuente por la “palabra2”, y escribir un archivo de salida llamado “sinonimos.txt”. Contar la cantidad de ocurrencias de “palabra2” en el archivo destino. Actualizar la entrega del zip en la webasignatura.