



Práctica 1 de la
Asignatura **Programación y Estructuras de
Datos Avanzadas**

Versión 1.0

Escuela Técnica Superior de
Ingeniería Informática

UNED

Curso 2019-2020

1.- ENUNCIADO DE LA PRÁCTICA: Cálculo del elemento mayoritario de un vector

Dado un vector $v[1..n]$ de número naturales, se quiere averiguar si existe un elemento *mayoritario*, es decir que aparezca al menos $n/2+1$ veces en el vector ($n/2$ es división entera).

Por ejemplo, el vector $[4,2,1,3,3,3,2,3]$ no tendría un elemento mayoritario, ya que tiene 8 elementos y ninguno aparece 5 veces o más. En el vector $[3,1,4,5,3,3,3,2,3]$ el 3 sería el elemento mayoritario, ya que el vector tiene 9 elementos y el 3 aparece 5 veces. Es decir, tenemos que para el problema:

$$n=9$$
$$V = [3,1,4,5,3,3,3,2,3]$$

La solución sería 3.

Se pide diseñar un algoritmo, siguiendo el **esquema de divide y vencerás**, que resuelva este problema.

2.- REALIZACIÓN DE LA PRÁCTICA

La entrega de la práctica consta de los siguientes elementos:

- Código fuente adecuadamente documentado y ejecutable.
- Documento **pdf** con la siguiente información:
 - Datos de la asignatura y del alumno:
 - Nombre y código de la asignatura
 - Título de la práctica
 - Nombre y Apellidos
 - NIF
 - Centro Asociado
 - Respuestas a los siguientes apartados:
 1. Describa el esquema algorítmico utilizado y cómo se aplica al problema.
 2. Analice el coste computacional del algoritmo.
 3. Exponga alternativas al esquema utilizado, si las hay, y compare su validez y coste con el de la solución realizada.
 4. Describa los datos de prueba utilizados y los resultados obtenidos con ellos.

Los detalles sobre evaluación pueden encontrarse en la guía de la asignatura.

2.1.- Diseño del algoritmo

La práctica constará de una memoria y de un programa Java original que resuelva el

problema aplicando el esquema de Divide y Vencerás. El diseño debe seguir las directrices que se dan para este problema en el texto base.

2.2.- Argumentos y parámetros

La práctica se invoca usando la siguiente sintaxis:

```
java      mayoritario      [-t][-h]      [fichero_entrada]
[fichero_salida]
```

o

```
java -jar mayoritario.jar  [-t][-h]  [fichero_entrada]
[fichero_salida]
```

Los argumentos son los siguientes:

- **-t:** traza cada paso de manera que se describa la aplicación del algoritmo utilizado.
- **-h:** muestra una ayuda y la sintaxis del comando. Por ejemplo:

```
$ Java mayoritario -h <ENTER>
```

SINTAXIS:

```
mayoritario [-t][-h][fichero_entrada] [fichero_salida]
-t                      Traza las llamadas recursivas
-h                      Muestra esta ayuda
fichero_entrada         Nombre del fichero de entrada
fichero_salida          Nombre del fichero de salida
```

- **fichero_entrada:** es el nombre del fichero del que se leen los datos de entrada; es decir, el número de elementos en el vector de enteros y el propio vector. Si la entrada no es correcta, el programa debe indicarlo. En caso de que no exista el fichero, se utilizará la entrada estándar.
- **fichero_salida:** es el nombre del fichero que se creará para almacenar la salida. Si el fichero ya existe, el comando dará un error. Si falta este argumento, el programa muestra el resultado por pantalla.

2.3- Datos de entrada

El fichero de datos de entrada consta de:

- Una primera línea que indica el valor del parámetro n , número de elementos en el vector.
- Una segunda línea que indica los elementos del vector de enteros separados por espacios en blanco.

De esta forma, para el ejemplo utilizado en la descripción del problema, el fichero de entrada contendría lo siguiente:

```
9
3 1 4 5 3 3 3 2 3
```

2.4- Datos de salida

La salida es una línea con el valor del elemento mayoritario. Si no existe se indicará con un carácter especial, N. En el ejemplo, la salida sería:

3

El programa debe indicar si no existe solución.

2.5.- Implementación del algoritmo

El programa se desarrollará en Java siguiendo un diseño orientado a objetos. Los detalles del entorno recomendado se encuentran en la guía de la asignatura.

Todos estos requisitos son obligatorios para aprobar la práctica.

4.- ENTREGA DE LA PRÁCTICA

1. La asistencia a las sesiones de prácticas es obligatoria SIN excepciones **para todos los alumnos que tengan que realizar las prácticas este curso.**
2. El calendario y procedimiento para asistir a las sesiones de prácticas **está publicado en su Centro Asociado o bien aparece en el foro correspondiente a su centro en el curso virtual.**
3. El plazo de entrega de la documentación y de la práctica **lo establece el Tutor de prácticas de cada Centro Asociado o Campus.**
4. El Tutor califica la práctica, informa al alumno y en su caso la revisa de acuerdo con los horarios y procedimiento que establezca el Centro Asociado.
5. TODOS los alumnos que tengan que realizar la práctica deberán registrarse a través del Curso Virtual (acceso desde Alf) con el Tutor/a con el que hayan asistido a las sesiones presenciales obligatorias a fin de que su práctica pueda ser calificada.
6. Todos los alumnos que no tengan la práctica aprobada en el curso inmediatamente anterior (2018-2019), tendrán que realizar la práctica.
7. Para los alumnos que no tienen aprobada la práctica del curso anterior, la práctica se debe aprobar en la misma o anterior convocatoria para que se pueda calificar la asignatura. En caso contrario la calificación será de suspenso.
8. La práctica se entregará empaquetada tanto en el entorno Alf como al Tutor. La falta de cualquiera de ellas será motivo suficiente para quedar excluida de la convocatoria.

Material que hay que entregar al Tutor

Se confeccionará una memoria, en formato pdf, con índice:

1. Respuesta a las cuestiones planteadas en este enunciado.
2. Un ejemplo de ejecución para distintos tamaños del problema.
3. Un listado del código fuente COMPLETO.

Se enviará al tutor, por los medios que este indique, la siguiente información:

1. Fuentes de cada uno de los módulos.
2. Ejecutable *.jar del programa principal.
3. Ficheros de los casos de prueba.
4. Memoria de la práctica.

Toda esta documentación deberá tener bien legible: **Nombre y apellidos, C. Asociado, teléfono y e-mail.**

El alumno debe asegurarse de que no se da ninguna de las siguientes circunstancias, ya que implican automáticamente una calificación de SUSPENSO:

- **Código:** El código no compila, no está desarrollado en Java, no se corresponde con el pseudocódigo recogido en la documentación, no es original, está copiado de la red, academia, compañero, etc., o no sigue un diseño OO encapsulado o modular.
- **Ejecutable:** El ejecutable no termina, se queda sin memoria con ejemplares pequeños o aborta sin justificación. El ejecutable no lee los ficheros previstos en el formato adecuado. No trata los argumentos o no se ajusta a las especificaciones.
- **Documentación:** No se presenta en papel (o el soporte indicado por el tutor) o está incompleta.
- **Soporte:** No se puede leer, o contiene un virus de cualquier tipo. A este respecto, las prácticas en las que se detecte cualquier tipo de virus estarán suspensas.

TUTORES

1. La práctica se califica mediante un entorno web accesible desde los Cursos Virtuales.
2. Los tutores se podrán dar de alta en la aplicación de entrega de notas de la práctica, sin necesidad de ponerse en contacto con el equipo docente. En el Foro de tutores recibirán las instrucciones pertinentes.
3. Los alumnos se dan de alta en la aplicación de corrección y calificación de prácticas accesible desde Alf de acuerdo con las indicaciones del Tutor. Aquel alumno que no se dé de alta no aparecerá en la lista de presentados a la misma y no constará su nota en el entorno de calificación y revisiones.

La práctica debe realizarse de forma individual y se entregará al TUTOR en las fechas que fije cada Centro Asociado.

El Centro Asociado publicará las notas de las prácticas. El equipo docente **NO INFORMARÁ DE LAS NOTAS DE PRÁCTICAS**, debiendo el Alumno dirigirse al Centro Asociado para conocerlas.