Examen de Programación – Curso 2018-19
3er Trimestre – Intermedio

1. Escribe un programa **recuentoVotos** que leerá el resultado de una votación desde la base de datos y sacará un listado por pantalla con los resultados en forma de tabla.
   La base de datos contendrá una sola tabla en la que cada registro será un voto.
   Para leer los votos usad la consulta: "SELECT id, voto FROM votacion".
   Se deberá contar cuántos votos tiene cada partido y sacar por pantalla una tabla en este formato:

```
PACMA                120 votos      12%
Partido Pirata        90 votos      10%
...
```

   Adicionalmente, el listado deberá estar ordenado de mayor a menor número de votos.

   La base de datos "votacion1.db" contiene 1000 votos repartidos entre: PSOE:254, PP:244, Ciudadanos:146, Podemos:127, PACMA:118, Vox:111, mientras que "votacion2.db" contiene 436 votos repartidos entre: Vox:118, PACMA:103, PSOE:61, PP:59, Podemos:54, Ciudadanos:41.

2. Escribe una función **transfiereFicheroBD**, que leerá datos desde un fichero de texto en formato CSV y los insertará en una base de datos.
   El fichero contendrá datos de alumnos en el siguiente formato:

```
NOMBRE,FECHA DE NACIMIENTO,CURSO
Juan Fernández García,14/01/1985,2ESO
Elena Marín Hernández,24/06/1984,1ESO
...
```

   Los datos se deberán introducir en la tabla **alumnos** mediante la siguiente consulta: "INSERT INTO alumnos(nombre, fechaNacimiento, idCurso) VALUES ('Juan Fernández García', '1985-01-14 00:00:00.000', 22)" donde el idCurso hay que obtenerlo de la tabla **cursos** mediante la consulta "SELECT id FROM cursos WHERE nombre = '2ESO'".

   La función deberá comprobar que el fichero tiene el formato esperado (la primera línea coincide con la del ejemplo). Si una línea de datos tiene menos de tres campos, esa línea no se introducirá en la BD y se mostrará un mensaje de advertencia por pantalla. Lo mismo sucederá si el curso no existe en la base de datos.

3. Escribe la clase **BuscaTexto** que nos permitirá sacar información de un texto. La clase tendrá los siguientes componentes:
   - Atributos:
     - *String* **texto**, donde almacenaremos el texto en el que vamos a buscar la información
     - *int* **puntero**, donde guardaremos un puntero que nos dirá por dónde vamos buscando en el texto.
   - Constructores:
     - Un constructor vacío, que inicializará el texto a una cadena vacía y el puntero a 0.
     - Un constructor al que le pasamos una cadena que guardará en *texto* y pondrá el puntero a 0.
   - Métodos:
     - **cargaFichero**(*String* fichero), lee el fichero de texto que le pasamos y guarda su contenido en el atributo *texto*.
     - **busca**(*String* cadena), busca la cadena dentro del texto y guarda esa posición en el *puntero*. Si la cadena no se encuentra, guardaremos un 0 en el puntero (si guardamos un -1 puede dar problemas). Siempre buscará desde el principio.
     - **buscaSiguiente**(*String* cadena), busca la cadena dentro del texto a partir de la posición actual del *puntero* y guarda el resultado en el *puntero*.
     - *String* **extraeCadena**(*String* delimitador1, *String* delimitador2), a partir de la posición actual del puntero, busca el *delimitador1*. A partir de ahí irá guardando todos los caracteres que encuentre en una cadena hasta llegar al *delimitador2*. Devolverá la cadena obtenida.
       Ej.: Si el texto es un HTML "<h1>ejemplo</h1>" y decimos extraeCadena("<", ">") nos debería devolver "h1", mientras que si decimos extraeCadena(">", "<") devolvería "ejemplo".
   - Propiedades:
     - **get y set** para el atributo *puntero*. Al modificar el puntero, no se podrá escribir un valor menor que 0 ni mayor que la longitud del texto. Si nos escriben un valor menor que 0 se guardará un 0 y si nos escriben un valor mayor que la longitud del texto, se escribirá la longitud del texto.
     - **get** para el atributo *texto*.

4. Escribe la función **dibujaXestrellitas**(*int* pisos), que nos dibujará un reloj de arena con el número de pisos que le digamos. El número de pisos deberá ser impar y el tamaño mínimo será 3. Como ejemplo os pongo uno de 5 y uno de 7.

```
XXXXX
 XXX
  X
 XXX
XXXXX
```

```
XXXXXXX
 XXXXX
  XXX
   X
  XXX
 XXXXX
XXXXXXX
```