Consideramos el poema de Jaime Gil de Biedma (1929-1990) al final de la práctica. Lo primero que hacemos es copiarlo a un fichero.txt (digamos 'amistad.txt'). Usaremos el siguiente código para insertar el texto línea a línea en MongoDB:

```
use poemas
var file = cat('/.../amistad.txt') // cambiar … por el camino al fichero!
var versos = file.split('\n')
l = versos.length
db.amistad.drop()
for(i=0; i<l; i++) db.amistad.insert({verso:versos[i], longitud:versos[i].length,pos:i})
```

Ahora podemos probar a ver si el texto se ha insertado correctamente:

```
db.amistad.find({},{_id:0})
```

1) Define una consulta con una expresión regular ($regex) para encontrar el número de versos que contienen la secuencia 'en'.
Copia en la solución
consulta: la consulta
número: número que se obtiene
Solución:
consulta: db.amistad.find({verso:{$regex:'en'}}).count()
Número: 20

2) Crea un índice de tipo textual sobre la clave verso, y asignarle el nombre “iverso”.
Copia en la solución:
instrucción: instrucción que crea el índice.

Solución:
instrucción:db.amistad.createIndex({verso:"text"}, "iverso")

3) Utilizar el índice creado para encontrar cuántas veces se repite la palabra “en”.
Copia en la solución:
consulta: la consulta
número: número que se obtiene

Solución:
consulta: db.amistad.find({$text:{$search:"en"}}).count()
número: 5

4) Excribe una expresión que permita calcular veces aparece la subcadena “el” sin ser una palabra.
Copia en la solución
expresión: la expresión
número: el número que se obtiene
Solución:
expresión: db.amistad.find({verso:{$regex:"el"}}).count()-db.amistad.find({$text:
{$search:"el"}}).count()
número: 2

5) Crea

- Un índice sobre la clave longitud, descendente, con nombre “longdown”

- Otra sobre la clave pos, ascendente y único, con nombre “posup”
  Copia en la solución:

  Instrucciones: Las instrucciones que han permitido crear los dos índices

Solución:
instrucciones:
db.amistad.createIndex({longitud:-1},"longdown")
db.amistad.createIndex({pos:1},"posup",{unique:1})

6) Escribe una instrucción que muestre las longitudes y posiciones (solo las longitudes y posiciones), de la colección *amistad*, en las que la longitud es mayor que 30, ordenadas en orden ascendente en longitud y descendente en pos.
Copia en la solución:
consulta: la consulta

Solución:
consulta: db.amistad.find({longitud:{$gt:30}},{longitud:1,pos:1,_id:0}).sort({longitud:1,pos:-1})

7) ¿Qué índices se han usado y cuántos documentos y claves ha hecho falta consultar para ejecutar la consulta anterior?
Copia en la solución:
indices: nombres de los índices separados por comas (o “ninguno”),
documentos: el total de documentos accedidos
claves: total de claves de índice a las que se han accedido,
Instrucciones: instrucciones que te han permito averiguar esta información (una por línea)
Solución:
Índices: longdown
documentos: 23
clave:23
Instrucciones:
explica = db.amistad.explain("executionStats")
explica.find({longitud:{$gt:30}},{longitud:1,pos:1,_id:0}).sort({longitud:1,pos:-1})

8) ¿Qué indice crearías para mejorar la eficiencia de esta consulta?
Copia
indice: la instrucción para crear el índice
comprobar: la instrucción que te ha permitido comprobar la mejora
explicación: una breve explicación que indique por qué se ha mejorado
Solución:
indice: db.amistad.createIndex({longitud:1,pos:-1},"longupposdown")
comprobar: explica.find({longitud:{$gt:30}},{longitud:1,pos:1,_id:0}).sort({longitud:1,pos:-1})
explicación:el plan anterior aparece rechazado, y el nuevo ocupa su lugar

9) Queremos ver el plan de ejecución que sigue MongoDB al hacer una consulta para ver si existe algún documento con “verso” igual a “tururu” dentro de la colección “amistad”.
Copia en la solución:
stage: Escribe el valor que se encuentra en la clave “stage” del plan usado, y explica qué significa ese valor, en particular su relación con el uso de índices.
Solución:
stage: “COLLSCAN”, indica que se ha hecho un recorrido completo de la tabla, sin usar índices
Aunque habíamos creado un índice para la columna verso, se trataba de un índice de tipo texto, que descompone cada frase en palabras, no de un índice para el valor de “verso” al completo.

10) Escribe en el shell una consulta que muestre aquellos versos para los que se verifique que su posición (pos) es mayor que su longitud.
Copia en la solución:
consulta: la consulta
índices: nombre de los índices utilizados para resolverlos
rechazados:la lista de planes rechazados ([] para ninguno)
No hace falta decir cómo se ha averiguado el nombre de los índices ni la lista de planes rechazados.
Solución:
consulta: db.amistad.find({ $where : "this.pos > this.longitud" })
índices: ninguno
rechazados:[]
Nota: El código JavaScript no es asequible al plan de ejecución.

Pasan lentos los días
y muchas veces estuvimos solos.
Pero luego hay momentos felices
para dejarse ser en amistad.

Mirad:
somos nosotros.

Un destino condujo diestramente
las horas, y brotó la compañía.
Llegaban noches. Al amor de ellas
nosotros encendíamos palabras,
las palabras que luego abandonamos
para subir a más:
empezamos a ser los compañeros
que se conocen
por encima de la voz o de la seña.
Ahora sí. Pueden alzarse
las gentiles palabras
-ésas que ya no dicen cosas-,
flotar ligeramente sobre el aire;
porque estamos nosotros enzarzados
en mundo, sarmentosos
de historia acumulada,
y está la compañía que formamos plena,
frondosa de presencias.
Detrás de cada uno
vela su casa, el campo, la distancia.

Pero callad.
Quiero deciros algo.
Sólo quiero deciros que estamos todos juntos.
A veces, al hablar, alguno olvida
su brazo sobre el mío,
y yo aunque esté callado doy las gracias,
porque hay paz en los cuerpos y en nosotros.
Quiero deciros cómo trajimos
nuestras vidas aquí, para contarlas.
Largamente, los unos con los otros
en el rincón hablamos, tantos meses!
que nos sabemos bien, y en el recuerdo
el júbilo es igual a la tristeza.
Para nosotros el dolor es tierno.

Ay el tiempo! Ya todo se comprende.