

# Algoritmos y Programación I (75.40)

## Trabajo práctico n.º 2

Segundo cuatrimestre 2016

### Consigna

Implementar una herramienta que permita guardar notas. Debe contar con un menú de opciones interactivo que permitirá:

- Agregar una nota, opcionalmente etiquetándola con una palabra clave
- Ver el listado completo de notas, agrupadas por etiqueta
- Buscar notas que contengan un texto determinado
- Eliminar una nota
- Generar una vista en formato HTML
- Salir

La próxima vez que inicie el programa, se espera que “recuerde” todas las notas que se hayan guardado.

### Modo de uso

A continuación se describe en detalle el comportamiento esperado para cada una de las opciones del menú interactivo.

En todos los casos se espera que el programa valide los datos ingresados por el usuario, mostrando un mensaje de error apropiado en caso de ser necesario.

---

**Agregar una nota:** La herramienta debe pedir el texto de la nota. El usuario debe escribir el texto deseado y presionar la tecla <Enter> para terminar.

Luego se le pide al usuario que ingrese una etiqueta para la nota. La etiqueta puede ser una palabra (sin espacios):

Ingrese el texto de la nota: arroz blanco  
Etiqueta: super

El usuario también puede ingresar un texto vacío para indicar que no desea etiquetar la nota:

Ingrese el texto de la nota: Ver Game of Thrones!  
Etiqueta:

---

**Listar notas:** Debe mostrar todas las notas, agrupadas por etiqueta.

[super]  
- arroz blanco  
- leche  
- huevos  
- shampoo y acondicionador

[Sin etiqueta]  
- Ver Game of Thrones!  
- Contratar dos investigadores privados. Hacer que se sigan el uno al otro.

---

**Buscar:** Debe pedir al usuario un texto y luego mostrar todas las notas que contengan ese texto (ignorando mayúsculas y minúsculas), agrupadas por etiqueta.

Ingrese el texto a buscar: con

[super]

- shampoo y acondicionador

[Sin etiqueta]

- Contratar dos investigadores privados. Hacer que se sigan el uno al otro.

---

**Eliminar:** Debe pedir al usuario una etiqueta (que puede ser vacía), luego listar todas las notas con esa etiqueta marcadas con un número. El usuario ingresa el número de nota que desea eliminar (o un texto vacío para cancelar). Luego se le pide confirmación al usuario.

Ingrese una etiqueta: super

[super]

1. arroz blanco

2. leche

3. huevos

4. shampoo y acondicionador

Cuál desea eliminar? 3

3. huevos

Eliminar [s/n]? s

---

**Reporte HTML:** Debe generar un archivo `notas.html` (sobreescribiéndolo si ya existía) que contenga el listado completo de notas en formato HTML<sup>1</sup>. Al cargar este archivo utilizando cualquier navegador web debería verse el listado de notas con encabezados, texto en negrita, etc.

A continuación se muestra un ejemplo del contenido de `notas.html`, y en la Figura 1 el resultado de cargar el archivo en un navegador web.

---

<sup>1</sup> <https://es.wikipedia.org/wiki/HTML>

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Listado de notas</title>
  </head>
  <body>
    <h1>Listado de notas</h1>
    <h2>super</h2>
    <ul>
      <li>arroz blanco</li>
      <li>leche</li>
      <li>huevos</li>
      <li>shampoo y acondicionador</li>
    </ul>
    <h2>Sin etiqueta</h2>
    <ul>
      <li>Ver Game of Thrones!</li>
      <li>Contratar dos investigadores privados.
        Hacer que se sigan el uno al otro.</li>
    </ul>
  </body>
</html>
```

---

**Salir:** Debe finalizar la ejecución del programa.



Figura 1: El archivo `notas.html` cargado en un navegador web

## Persistencia

Se espera que la herramienta “recuerde” todas las notas guardadas, incluso luego de finalizar la ejecución y ser iniciada de nuevo. Para esto, la lista de notas debe ser guardada en un archivo `notas.csv`.

La información se guardará en formato CSV, con una nota por línea, separados por el caracter `|`. La primera columna será la etiqueta y la segunda será el texto de la nota:

```
super|arroz blanco
super|leche
super|huevos
super|shampoo y acondicionador
|Ver Game of Thrones!
|Contratar dos investigadores privados. Hacer que se sigan el uno al otro.
```

Si el archivo `notas.csv` no existe, la herramienta debe crearlo.

En caso de que ocurra algún error al cargar o guardar el archivo, se debe informar al usuario y finalizar la ejecución.

## Criterios de aprobación

A continuación se describen los criterios y lineamientos que deben respetarse en el desarrollo del trabajo.

### Informe

El informe debe consistir en una descripción del **diseño** del programa.

Debe recordarse que la etapa de diseño es *anterior a la implementación*, por lo tanto debe describirse, utilizando texto y/o diagramas, cómo se va a estructurar el código para cumplir con las especificaciones de la consigna.

Algunas preguntas que deberían responderse:

- A grandes rasgos, ¿cómo será el flujo del programa? ¿Cuáles archivos se leen o escriben, y en qué momentos?
- ¿Qué estructura(s) de datos se usan para representar la base de datos de notas?

### Código

Además de satisfacer las especificaciones de la consigna, el código entregado debe cumplir los siguientes requerimientos:

- El código debe ser claro y legible.
- El código debe estructurarse en funciones y, cuando corresponda, módulos. Las funciones deben definirse de la manera más genérica posible.
- Todas las funciones deben estar adecuadamente documentadas, y donde sea necesario el código debe estar acompañado de comentarios.

## Entrega

La entrega del trabajo consiste en:

- El informe y código fuente impresos. Para el código fuente utilizar una tipografía `monoespacio`.
- El informe en formato *PDF*.
- Una versión digital de todos archivos `.py` de código, separados del informe. En el caso de ser más de un archivo, comprimidos en un `.zip`.

El informe impreso debe entregarse en clase. Los dos últimos (PDF y código fuente) deben enviarse a la dirección electrónica `tps.7540rw@gmail.com` con el asunto "TP2 - PADRÓN1 - PADRÓN2".

Este trabajo práctico se desarrolla en forma **grupal**. El plazo de entrega vence el **lunes 17 de octubre de 2016**.