



Instituto de Formación Técnica Superior N°18

<http://www.ifts18.edu.ar>

Mansilla 3643 (C1425BBW), Ciudad Autónoma de Buenos Aires

Técnico Superior en Análisis de Sistemas

---

# Examen Parcial

---

*Asignatura:*  
Estructura de Datos

*Autor:*  
Prof. Leandro E. Colombo Viña

*última modificación:*  
25 de octubre de 2017

# Índice general

<b>1 Examen Parcial</b>	<b>1</b>
1.1 Introducción . . . . .	1
1.2 Consultas . . . . .	1
1.2.1 Archivo de entrada . . . . .	1
1.2.2 Consultas . . . . .	2
1.3 Criterios de aprobación . . . . .	2
1.3.1 Informe . . . . .	3
1.3.2 Programa . . . . .	3
1.4 Entrega . . . . .	7
1.4.1 Clase de Consulta . . . . .	7
<b>2 Recursos y sugerencias</b>	<b>9</b>
2.1 Código fuente . . . . .	9
2.1.1 Sistema de Control de Versiones . . . . .	9
2.1.2 Opcional - Documentación . . . . .	9

## **Resumen**

El presente documento contiene las consignas y consideraciones para la realización del examen parcial de la asignatura Estructura de Datos que se dicta durante el 2° cuatrimestre del primer año de la carrera Técnico Superior en Análisis de Sistemas en el Instituto de Formación Técnica Superior N°18. Esta consigna es válida para la cursada correspondiente al año 2017.

# Capítulo 1

## Examen Parcial

### 1.1 Introducción

El departamento de contabilidad de una importante droguería requiere de una actualización al programa que utilizan. Necesitan que este les permita hacer consultas sobre su nueva base de datos, en la que almacenan información de las compras de los medicamentos que realizaron sus clientes en el período de un mes. Adicionalmente, el departamento de sistemas solicita que el nuevo programa sea una aplicación web utilizando Python 3 y Flask como framework de desarrollo.

Considerando que la información está almacenada en archivos CSV, su influenciable gerencia ha aceptado que parte de la aplicación sea desarrollada por un grupo de estudiantes terciarios que se encuentran estudiando el uso de archivos.

### 1.2 Consultas

#### 1.2.1 Archivo de entrada

El programa deberá permitir procesar información referida a las ventas de la droguería desde un archivo CSV. En el fichero, cada línea representa una venta y tiene los campos:

```
CLIENTE, CODIGO, PRODUCTO, CANTIDAD, PRECIO
```

en donde CANTIDAD es la cantidad de unidades vendidas en la compra y PRECIO es el precio unitario del producto.

Sin embargo, aún no se sabe si el orden de los campos será ese, por lo que el programa deberá identificarlos de acuerdo a la cabecera del archivo.

Un ejemplo válido, entonces, podría ser:

```
CODIGO,PRODUCTO,CLIENTE,CANTIDAD,PRECIO
00001,ALIVIOL 3000,FARMACIA NUEVA CHINGOLO,1.00,24.500
00042,CLONAPLEX 400 X12,LONGCHAMPS MEDICAL CENTER,1.00,34.132
020102,PARAMITOL 2MG FRASX20,CLIENTE PARTICULAR 00,4.00,12.00
31100,RESACAL X2,CENTRO DE CUIDADOS VILLA URQUIZA,2.00,15.00
```

Es necesario que se contemplen, también, los siguientes casos:

- Que el archivo de datos de entrada sea inexistente.
- Que algún registro tenga una cantidad inválida de campos.
- Que algún campo de CODIGO no sea vacío.
- Que algún campo de CANTIDAD no contenga un número entero.
- Que algún campo de PRECIO no contenga un valor decimal.

en donde se espera que el programa imprima un mensaje informando el error correspondiente.

### 1.2.2 Consultas

El programa deberá poder realizar las siguientes consultas:

- Listar todos los productos que compró un cliente.
- Listar todos los clientes que compraron un determinado producto.
- Listar los  $n$  productos más vendidos.
- Listar los  $n$  clientes que más plata gastaron.

Además, para facilitarle el trabajo a los usuarios de la aplicación, la búsqueda por producto o por cliente debe poder permitir el ingreso del nombre parcial con un mínimo de 3 caracteres; de esta manera no es necesario que los empleados del departamento de contabilidad recuerden los nombres completos de los medicamentos ni de los clientes.

Por ejemplo, si se ingresa como nombre de cliente:

FARMA

se le mostrará un menú para que seleccione entre todos los clientes cuyo nombre contenga la cadena ingresada, por ejemplo:

- FARMACIAS MUNRO.
- FARMACIA DEL SUR.
- FARMACIA NUEVA CHINGOLO.
- FARMACENTER GLEW.

## 1.3 Criterios de aprobación

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

### 1.3.1 Informe

El informe deberá consistir en una descripción del diseño del programa escrito en el archivo `README.md` en la raíz del repositorio.

Algunas preguntas que deberían responderse:

- A grandes rasgos, ¿cómo será el flujo del programa?
- ¿Qué estructura se utilizará para representar la información del archivo?
- ¿Cómo se usa el programa?
- ¿Qué clases se diseñaron? ¿Por qué?

### 1.3.2 Programa

Además de satisfacer la especificación de la consigna, el programa debe cumplir con los siguientes requerimientos:

- No debe fallar ninguna prueba.
- Debe estar adecuadamente estructurado y modularizado, utilizando funciones definidas en forma lo más genérica posible.
- El código debe ser claro y legible.
- Todas las funciones deben estar adecuadamente documentadas, y donde sea necesario el código debe estar acompañado de comentarios.

### Flujo

A continuación se puede observar un flujo posible del uso del sistema. Ingresando a la aplicación se podrá ver al menos los alumnos que realizaron el desarrollo y se puede agregar un breve párrafo de introducción al uso del sistema.



Figura 1.1: Pantalla principal

Se deberá poder acceder al sistema con nombre de usuario y contraseña. Las credenciales de los usuarios estarán almacenadas en un archivo CSV en el servidor.



Figura 1.2: Pantalla de login

Una vez logueado correctamente, el sistema redireccionará al usuario a una pantalla donde se podrán ver las últimas ventas realizadas y se tendrá el acceso a las posibles consultas del sistema.

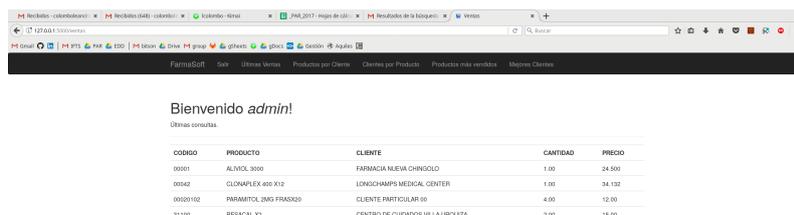


Figura 1.3: Listado de últimas ventas

Las pantallas de consultas pueden ser similares a las siguientes:

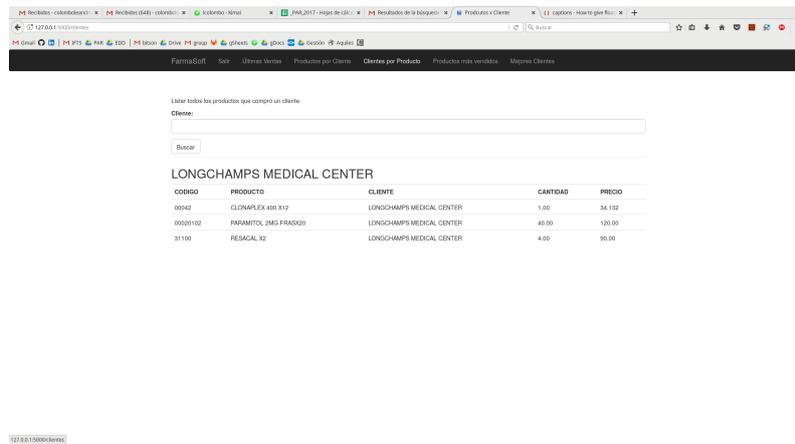


Figura 1.4: Listado de productos que compró un cliente

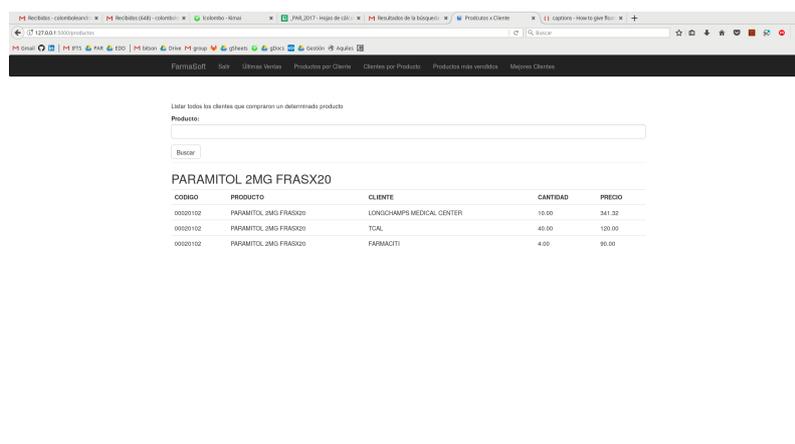


Figura 1.5: Listado de clientes que compararon un producto

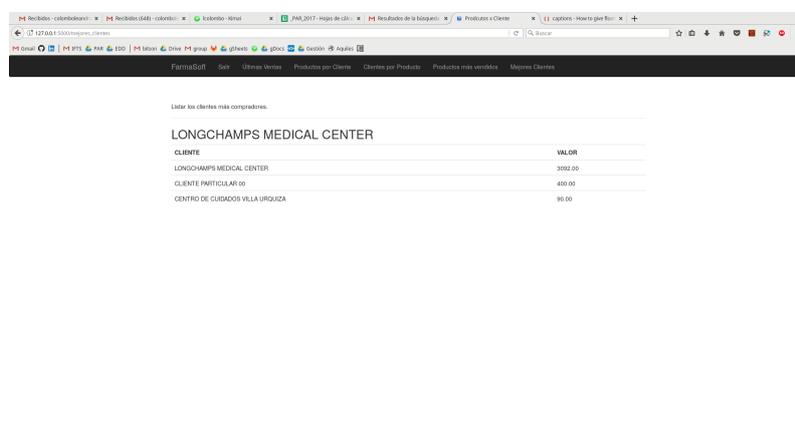


Figura 1.6: Listado de clientes más compradores

Listar los productos más vendidos		
PARAMITOL 2MG FRASX20		
CODIGO	PRODUCTO	CANTIDAD
0020102	PARAMITOL 2MG FRASX20	100.00
00001	ALIVOL 3000	60.00
31100	RESICAL X2	4.00

Figura 1.7: Listado de productos más vendidos

Se deben contemplar que pueden existir fallas en la aplicación y el usuario deberá ver pantallas según corresponda:



Figura 1.8: Pantallas de error: Error interno (500)

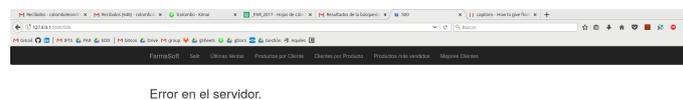


Figura 1.9: Pantallas de error: No encontrado (404)

Si el usuario intentara acceder a algún recurso de estos sin estar logueado en el sistema, el sistema deberá redireccionarlo a la página de ingreso para que pueda loguearse con nombre

de usuario y contraseña.

## 1.4 Entrega

La entrega del trabajo consiste en:

- Código fuente: el alumno deberá entregar el código indicando la URL del repositorio público.
- **OPCIONAL:** adicionalmente se puede utilizar algún sistema de prueba como pythonanyware o heroku para tener una demo del sitio.

La entrega se realizará a través de la lista de correos de la materia:

`estructura@listas.bitson.com.ar`

con el asunto:

**Parcial EDD - <apellido1> [& <apellido2>]**

Este examen se desarrolla en forma grupal de a 2 personas o de manera individual. El plazo de entrega vence el 08 de noviembre de 2017 hasta las 23:59hs (horario local Argentina).

### 1.4.1 Clase de Consulta

Para cualquier duda al respecto sobre el examen se puede utilizar la lista de correo electrónico de la materia.

`estructura@listas.bitson.com.ar`

Dispondremos además del tiempo de las últimas 3 clases durante la cursada que figuran en el calendario de la materia en el sitio oficial de la misma.



## Capítulo 2

# Recursos y sugerencias

### 2.1 Código fuente

Se puede obtener una versión del código fuente que brindó la droguería desde el repositorio en GitHub:

```
https://github.com/ifts18/farmasoft
```

El mismo está escrito en la versión 2.7 de Python.

#### 2.1.1 Sistema de Control de Versiones

Se recomienda utilizar un sistema de control de versiones como Git, pueden clonar directamente el repositorio original y trabajar sobre el mismo.

Es conveniente sacar una cuenta en GitHub para compartir el código más fácilmente.

Si ya tienen instalado y configurado Git, pueden clonar el repositorio con:

```
$ git clone https://github.com/ifts18/farmasoft
```

#### 2.1.2 Opcional - Documentación

Para la documentación se puede usar cualquier procesador de texto que tenga la capacidad de exportar el documento a formato PDF. Una interesante opción para realizar documentos centrados en el contenido es la utilización de [L<sup>A</sup>T<sub>E</sub>X](#).

Existen diferentes proyectos de documentación que sirven para facilitar este trabajo. Uno de los más populares entre la comunidad de Python que se puede utilizar es [sphinx](#).

Si prefieren pueden utilizar GitHub pages para realizar la documentación del proyecto. Pueden complementar la realización del mismo con un sistema de generación de páginas estáticas como [Nikola](#).