

## Parcial Programación 1 – Python – TEMA A

Se desea desarrollar un programa en Python que permita generar facturas de un negocio

Crear proyecto ParcialXXXXX donde XXXXX es su apellido

**Crear las siguientes clases Python en archivos distintos**

**Articulo.py**

**Factura.py**

**DetalleFactura.py**

**Clase Articulo**

**Atributos:**

Código (numero), denominación (str), rubro(str), marca(str), precioVenta(numero)

**Clase Factura**

**Atributos:**

fecha (día de hoy), numero (autogenerado iniciando en 1), total (será calculado por la suma de los detalles), listaDetalles (lista de objetos DetalleFactura)

Para la fecha

```
from datetime import date

#Día actual
fechaHoy = date.today()
print(fechaHoy)
```

**Clase DetalleFactura**

**Atributos:**

cantidad (numero entero), articulo (objeto Articulo asociado), subtotal (numero, será calculado precio \* cantidad)

**Funcionalidades:**

Crear una clase **GenerarFacturasMain.py** que deberá tener 4 opciones:

- a- Ver Facturas Cargadas
- b- Cargar 1 o más Facturas
- c- Generar Archivo Factura por numero
- d- Salir

El programa deberá poder:

1. **Al ejecutar el programa lo primero que deberá realizar es leer** el archivo de texto llamado articulos\_venta.txt, donde cada línea tiene el siguiente formato:

Código;Denominación;Rubro;Marca;Precio Venta

Ejemplo:

1;ACEITE DE COCO AEROSOL CHIA GRAAL;ACEITES COMESTIBLE;CHIA GRAAL;12465

10;PIMIENTA NEGRA GRANO SIN TACC-DICOMERE;SIN TACC;DICOMERE;1835

100;PAN DULCE INTEGRAL - HERBOTECA;PANIFICADOS;HERBOTECA;5294

.....

**Al leer el archivo deberá Generar un diccionario “articulosDataBase”** donde la clave es el código del alumno y el valor será el objeto Artículo correspondiente.

**Opciones del menú:**

**a- Ver Facturas Cargadas:**

- Si se selecciona esta opción se deberá mostrar una lista de las facturas cargadas hasta el momento mostrando solo fecha, numero de factura y total

**b- Cargar 1 o más Facturas**

- Para esta opción del menú se deberá crear una Factura asignar sus atributos y cargar su detalle correspondiente. Para la carga del detalle se deberá solicitar el código del artículo a cargar y obtenerlo del diccionario articulosDataBase, en caso de que el código no exista, indicarlo por pantalla y solicitarlo nuevamente. Se deberán cargar tantos detalles como se deseen hasta que el usuario indique que no quiere continuar. Validar que el atributo cantidad sea mayor a cero. Validar que se ingrese al menos 1 DetalleFactura.
- Finalizada la carga de la Factura guardarla en una variable listaFacturas[] y preguntarle al usuario si quiere cargar una nueva factura.
- Si indica que SI deberá repetir el proceso para una nueva factura

**c- Generar Archivo Factura por numero**

- Solicitar al usuario que ingrese un numero de factura; buscar la factura por su número en la listaFacturas[], si el número de factura no existe emitir el mensaje “La Factura con número XXXX no existe”.
- Si la factura existe mostrarla por pantalla con el siguiente formato

**Factura N° XXXXX**

**Fecha: XXXXX**

----- **Articulos Vendidos** -----

<b>Codigo</b>	<b>Denominación</b>	<b>Rubro</b>	<b>Marca</b>	<b>Cantidad</b>	<b>SubTotal</b>
XXXXXX	XXXXXX	XXXXXX	XXXXXX	XXXXXX	XXXXXX

**TOTAL: XXXXXX**

- Finalmente después de mostrar la factura por pantalla preguntar al usuario.

¿Desea Generar el archivo de la Factura? Ingrese S para generar u otro carácter sino lo desea.

Si el usuario ingresa S genere un archivo txt llamado factura\_nro\_XXXXX.txt con un formato lo similar posible al mostrado por pantalla.

**d- Salir** -> Finaliza el programa