

## Proyecto Programado #1

### *Sistema de Gestión de Pulperías*

#### Contenido

1.	Introducción.....	2
2.	Software por desarrollar.....	2
2.1.	Gestión de Inventarios .....	2
2.1.1.	Agregar producto.....	2
2.1.2.	Modificar producto .....	3
2.1.3.	Borrar producto .....	3
2.1.4.	Agregar inventario .....	3
2.1.5.	Reportes.....	3
2.1.5.1.	Reporte productos.....	3
2.1.5.2.	Reporte por categoría.....	4
2.1.5.3.	Reporte por existencia.....	4
2.2.	Facturación .....	4
2.3.	Reportes .....	5
2.3.1.	Reporte de facturas .....	5
2.3.2.	Reporte por producto vendidos.....	5
2.3.3.	Reporte por cliente .....	6
2.3.4.	Estado de utilidades.....	6
2.4.	Salir.....	6
3.	Opcionales .....	7
4.	Aspectos técnicos .....	7
5.	Documentación.....	7
6.	Evaluación.....	8
7.	Forma de trabajo .....	8
8.	Aspectos administrativos.....	8
9.	Rúbrica de evaluación.....	9
10.	Entrega .....	9

## 1. Introducción

Las aplicaciones de **Puntos de Ventas** son de las más utilizadas en todo tipo de negocio donde exista un intercambio monetario a través de productos o servicios. En este caso será una pequeña aplicación donde un pequeño negocio local “**Pulpería**” podrá gestionar sus ventas

## 2. Software por desarrollar

Su trabajo consiste en implementar una aplicación de escritorio para administrar el juego utilizando el lenguaje **Python**. El sistema inicia con un **Menú Principal** donde se puede ingresar a las:

- (I) Inventario,
- (F) Facturar
- (R) Reportes
- (S) Salir.

Si selecciona cualquier opción que no sea **Salir** se desplegará un menú para las diferentes opciones, y dentro de esos menús de podrá devolver al **Menú Principal**.

### 2.1. Gestión de Inventarios

Para acceder a estas funcionalidades el usuario deberá ingresar un **nombre de usuario y clave de acceso (debe estar generada y guardada en disco, el archivo se llamará Acceso.txt)**. El archivo debe de contener **al menos** 3 credenciales de acceso, bajo el siguiente formato; **usuario;clave**, por ejemplo:

```
pperez;1234  
ccastro;admin1234  
mporras;mporras1
```

Una vez autenticado el usuario debe habilitar las siguientes funcionalidades o submenú:

- (A) Agregar producto
- (M) Modificar producto
- (B) Borrar producto
- (I) Agregar inventario
- (R) Reporte de productos
- (S) Salir

#### 2.1.1. Agregar producto

Para agregar un nuevo producto, este solicitar al usuario cada uno de estos valores; **Nombre, categoría y precio Unitario**. Debe existir un archivo llamado **Inventario.txt**, en este se guardará línea por línea cada uno los productos de la siguiente forma:

1; Coca Cola; Bebidas; 1000; 0  
2; Atún Sardimar; Atún; 800; 0

El primer valor será un **código** y este se generará automáticamente, el segundo es el **nombre del producto**, el tercero la **categoría**, el siguiente el **precio unitario** y el último será la **cantidad en inventario**. Este último será modificado en el menú “**Agregar Inventario**” 2.1.4.

Los nombres de los productos no deben repetirse. Una vez generado el producto, deberá imprimir el número de código generado

### 2.1.2.Modificar producto

Con base a un **código** de producto existente, el usuario podrá **modificar** solo el **nombre del producto**, **categorías** y **precio unitario**. Es necesario que previamente el usuario pueda ver el valor anterior para ver qué cambiar. Los nombres de los productos **no deben repetirse**.

### 2.1.3.Borrar producto

Con base a un **código** de producto existente, el usuario podrá **borrar** un producto, antes de ser borrado, debe recibir un mensaje de confirmación, por ejemplo: “Está seguro de que desea borrar el producto 1: Coca Cola?”, tiene 2 opciones a responder, **Si y No**. No se puede borrar un producto que haya sido facturado

### 2.1.4.Agregar inventario

Con base a un código de producto existente, el usuario podrá **agregar** la cantidad inventario de un producto. El número debe ser **mayor a cero y entero**

### 2.1.5.Reportes

Tendrá tres opciones:

#### 2.1.5.1. Reporte de Productos

El usuario podrá imprimir en pantalla **todos los productos** registrados: su **código, nombre y precio unitario y cantidad de inventario**

### 2.1.5.2. Reporte por Categoría

El usuario podrá imprimir **todos los productos** dado una **categoría**, este podrá ser ya sea el **nombre completo** de la categoría o **parte** de este, e imprimirá con el siguiente formato: **su código, nombre y precio unitario y cantidad de inventario**

### 2.1.5.3. Reporte por Existencia

El usuario solo imprimirá los productos cuya existencia de inventario **sea mayor a cero**, y con el siguiente formato: **su código, nombre y precio unitario y cantidad de inventario**

## 2.2. Facturación

Para **facturar**, el sistema debe solicitar al usuario lo siguiente:

- Nombre de cliente

Luego procede a registrar los productos que está adquiriendo, y se debe solicitar:

- El **código** del producto (y luego el sistema imprime el **nombre** de este)
- **Cantidad** por comprar. Debe verificar la **existencia** en el archivo de **Inventario.txt**. Posteriormente debe restarse la cantidad a comprar a la cantidad en existencia del inventario.

Se podrá facturar más de un producto, por cada producto ingresado el sistema imprime las líneas con el siguiente detalle:

Cantidad	Producto	Precio Unitario	Impuesto 13%	Total
2	Coca Cola	1000	260	2260
3	Atún Sardimar	800	312	2712

Al final cuando el cliente solicita **finalizar**, este genera la factura e imprimir el siguiente detalle en pantalla:

#### Factura 1

**Fecha:** 30/03/2023

**Hora:** 1:30 pm

**Cliente:** Pedro Perez

Cantidad	Producto	Precio Unitario	Impuesto	Total
2	Coca Cola	1000	260	2260
3	Atún Sardimar	800	312	2712

Sub total: 4400  
Impuestos 572  
**Total: 4972**

Y registrar la **forma de pago**, estos pueden ser: **Efectivo, Sinpe Movil o Tarjeta de Crédito.**

Debe existir un archivo llamado "**Facturas.txt**" y "**FacturaDetalle.txt**" y el formato interno debe ser de la siguiente forma:

#### **Facturas.txt**

El formato del archivo es, número de factura, nombre del cliente, fecha del sistema, hora del sistema, sub total, total impuesto, total y forma de pago. El número de factura debe ser automático.

1,Pedro Perez,30/3/2023,1:30pm,4400,572,4972, Contado  
2,Carlos Perez,3/4/2023,4:30pm,1000,130,1130, Sinpe movil

#### **FacturasDetalle.txt**

El formato del archivo es, número de factura, código del producto, nombre del producto, cantidad comprada, precio unitario, subtotal, impuesto y total, a continuación, un ejemplo:

1, 1,Coca Cola,2,1000,2000,260,2260  
1, 2, Atún Sardimar, 3, 800, 2400, 312, 2712  
2, 1, Coco Cola, 1,1000,1000,130,1130

## **2.3. Reportes**

Se generarán los siguientes reportes:

### **2.3.1.Reporte de facturas**

Se imprime en pantalla la lista de facturas, con el siguiente formato:

# Factura	Fecha	Cliente	Total Facturado
1	30/03/2023	Pedro Perez	4972
2	3/04/2023	Carlos Perez	1130

**TOTAL 6102**

Y al final la suma **total** de lo facturado

### **2.3.2.Reporte por producto vendidos**

Se solicita el **nombre** del producto o nombre parcial y se imprime en pantalla la lista de facturas, con el siguiente formato:

Producto: Coca Cola

# Factura	Fecha	Precio Unitario	Cantidad	Impuesto	Total Facturado
1	30/03/2023	1000	2	260	2260
2	3/04/2023	1000	1	130	1130

**TOTAL 3390**

Y al final la **suma total** de lo facturado

### 2.3.3.Reporte por cliente

Se solicita el **nombre** del cliente o nombre **parcial** y se imprime en pantalla la lista de facturas, con el siguiente formato:

Cliente: Pedro Perez

# Factura	Fecha	Total Facturado
1	30/03/2023	4972

**TOTAL 4972**

Y al final la **suma total** de lo facturado

### 2.3.4.Estado de utilidades

Imprime los siguientes valores en pantalla:

- Total Facturas: 2
- Cantidad de productos facturados **únicos**: 2
- SubTotal Facturado 5400
- Total Impuesto: 732
- Total Facturado: 6132

## 2.4. Salir

**Finaliza el programa.** Al seleccionar esta opción se debe generar un mecanismo (manejo de archivos) para que los datos sean **persistentes**. La información se debe mantener para los siguientes usos del programa.

### 3. Opcionales

Se darán **10 puntos adicionales** en todos los reportes que se tienen que generar, que además de **imprimirse en pantalla**, este **genere un archivo de salida**, obviamente debe **solicitar** al usuario el **nombre de archivo**. El archivo por generar debe tener un formato “elegante”.

Se darán **5 puntos adicionales** si agrega un reporte de facturas por **rango de fechas** y generar en **pantalla** y también por **archivo aparte**

### 4. Aspectos técnicos

El proyecto deberá estar escrito en el lenguaje de programación **Python**. En caso de requerir librerías adicionales para compilar y ejecutar el programa, deberán **especificarlo** en la documentación (*ejemplo, manejo de fechas*), ya que de lo contrario se descontarán puntos en la evaluación. Al iniciar el programa se carga la información del archivo y al salir se guarda (o una la manera permita que todos los datos registrados sean persistentes).

Y considerar lo siguiente:

- El archivo principal debe llamarse **Proyecto1.py** y la función principal para ejecutar debe ser **menú()**
- Se debe utilizar **iteración** en el desarrollo del proyecto.
- Se deben manejar **mensajes claros** al usuario.
- Realizar **validaciones de captura de campos**. Deben hacer uso de mecanismos de validación eficientes.
- Número de **identificador** es único y secuencial. Auto generado
- Toda **función built-in** de Python **no están permitidas**, a menos que sea necesario según la forma de cómo esté programando el proyecto, pero esto debe ser **consultado al profesor**. Algunas de ellas son: len, with, append(), remove, in (con excepción en el for), replace.
- **Se pueden** usar el Try, Except, Split, Strip, Upper Lower, Capitalize, title, center
- El proyecto debe ser guardado en el repositorio del **GitHub** (<https://classroom.github.com/a/Rq87r1bW> ).

### 5. Documentación

La documentación es un aspecto de gran importancia en el desarrollo de programas, especialmente en tareas relacionadas con el mantenimiento de estos.

Para la **documentación interna**, deberán incluir comentarios descriptivos para cada función, con sus **entradas, salidas, restricciones**. La **documentación externa** deber ser en **PDF** (si lo suben en otro formato, **no será leído**) y deberá incluir:

1. Portada.
2. Manual de usuario: **instrucciones de compilación, ejecución**.

3. Un enlace a un video en **youtube** donde muestre la funcionalidad de su aplicación. Su duración no de ser mayor a los 15 minutos.
4. Descripción del problema.
5. Diseño del programa: decisiones de desarrollo, algoritmos usados.
6. Librerías usadas: creación de archivos, etc.
7. Análisis de resultados: lista detallada de objetivos alcanzados, objetivos no alcanzados, y razones por las cuales no se alcanzaron los objetivos (en caso de haberlos).
8. **Bitácora** en GitHub con los **commit** por usuario incluyendo su descripción al momento de hacerlo.
9. Conclusión (es), su punto de vista sobre el programa desarrollado

## 6. Evaluación

La evaluación se va a centrar en dos elementos: programación y documentación. El proyecto programado tiene un valor de **15%** de la **nota final**, en el rubro de Proyectos.

Desglose de la evaluación del proyecto programado:

1. Documentación interna 5 puntos.
2. Documentación externa 5 puntos.
3. Funcionalidad 80 puntos (ver detalle en Software a Desarrollar)
4. Revisión del proyecto 10 puntos. **(No solo consiste en estar presente, sino saber responder las preguntas dadas por el profesor)**

## 7. Forma de trabajo

El trabajo se debe realizar de forma **individual**.

## 8. Aspectos administrativos

- Crear 2 carpetas llamadas **documentación** y **programa**, en la primera deberá incluir el documento **PDF** solicitado y en la segunda los archivos y/o carpetas necesarias para la implementación de este proyecto programado.
- **NO SUBIR ARCHIVOS EN FORMATO ZIP o SIMILARES**
- Deben modificar el archivo llamado **README.md**, este archivo debe contener la siguiente información:
  - a. Nombre del Estudiante y Número de carné del estudiante
  - b. Estatus de la entrega (debe ser **CONGRUENTE** con la solución entregada):  
[Deplorable | Regular | Buena | MuyBuena | Excelente | Superior]

Con este enlace <https://docs.github.com/es/github/writing-on-github/getting-started-with-writing-and-formatting-on-github/basic-writing-and-formatting-syntax> le ayudará a dar formato al archivo README.md



## 9. Rúbrica de evaluación

FUNCIONALIDAD		80%	
Rubro	Valor	Obtenido	Observaciones
Menú	5		
<b>Opciones Administrativas</b>			
Control de acceso	5		
Gestión de Inventarios	40		
Agregar producto	10		
Modificar producto	8		
Borrar producto	5		
Agregar inventario	7		
Reportes	10		
Facturación	20		
Reportes	30		
<b>Extras</b>			
Reportes en archivo	10		
Reporte por rango de fechas	5		
<b>TOTAL</b>	<b>115</b>	<b>0</b>	

DOCUMENTACIÓN INTERNA 5%

DOCUMENTACIÓN EXTERNA 5%

REVISIÓN DEL PROYECTO 10%

## 10. Entrega

Será el domingo **23 de abril de 2023**. Después de esa fecha con un **máximo de 2 días** de retraso se le restará 10 puntos, después de eso no es aceptado.

**Todo el contenido de cada proyecto debe ser 100% original y en caso de plagio se asignará nota cero y con carta al expediente.**