

TRABAJO PRÁCTICO

Sistema de Información de Venta de Locales

Grupos de Trabajo

Deberán formar grupos de 3 (tres) personas. Queda a cargo de los alumnos el armado de estos y deberán notificar al docente de su comisión vía mail sobre la forma como quedaron conformados los grupos en la semana del 15 de mayo.

Cada grupo recibirá de su docente vía mail la confirmación del grupo.

Fecha de Entrega del Temario: 15 de mayo 2023

Fecha límite de Entrega: la semana del 12 de Junio 2023 según comisión a las 23:59 horas del día de la clase.

Sistema a Implementar

El sistema que deberá implementar consistirá en un programa escrito en Python que procese datos de locales en venta durante el año 2020 en la Ciudad Autónoma de Buenos Aires.

Los datos son reales, y están en el archivo “**locales-en-venta-2020.csv**” que puede descargarse del campus.

El programa leerá y procesará los datos encontrados en el archivo mencionado anteriormente para obtener los resultados pedidos que se deben almacenar en distintos archivos.

Descripción Funcional del Programa

El programa deberá realizarse en Python. y debe ser amigable. Se entiende por amigable un programa que:

- Ofrezca un menú de opciones a través del cual los usuarios pueden ejecutar las diferentes tareas.
- Que informe al usuario los datos que debe ingresar y valide que la información sea correcta según lo pedido.
- No falle en ejecución.

El programa debe estar estructurado de tal forma que haga uso de funciones para la implementación de la solución del proyecto:

TRABAJO PRÁCTICO

Sistema de Información venta Locales

- Dividir el proyecto en tareas, implementadas en una o más funciones.
- Reusabilidad de las funciones más comunes

El programa debe mostrar al usuario un menú que debe tener las siguientes opciones:

1. Cargar los datos del archivo locales-en-venta-2020.csv
2. Generar estadísticas. Debe desplegar un submenú únicamente al escoger la opción mostrando las diferentes opciones de resultados a generar.
3. Agregar registros de nuevos locales.
4. Actualizar los datos en el archivo locales-en-venta-2020.csv
5. Salir

El programa abrirá el archivo "locales-en-venta-2020.csv" y luego procesará cada línea para obtener los resultados que se piden más adelante.

I.- Carga de datos

El archivo "locales-en-venta-2020.csv" es un archivo del tipo "Comma Separated Values", que puede abrirse en una planilla de cálculo (por ejemplo Libre Office Calc) o bien en un editor de texto (por ejemplo notepad++).

En la primera línea se encuentran los títulos de los campos, separados por "," (coma):

CALLE,SUPERFICIE,PRECIOUSD,PRECIOPESOS,USDM2,ANTIG,EN_GALERIA,COTIZ,TRIMESTRE,BARRIO,COM UNA

Luego, a partir de la segunda línea se encuentran los datos de los locales en venta, también separados por "," (coma):

"ACUÑA DE FIGUEROA, FRANCISCO 1000",26,87000,5481000,3346,5,SI,63,PRIMER,ALMAGRO,5
"ACUÑA DE FIGUEROA, FRANCISCO 400",34,92000,5796000,2706,45,NO,63,PRIMER,ALMAGRO,5
"ACUÑA DE FIGUEROA, FRANCISCO 900",60,110000,6930000,1833,0,NO,63,PRIMER,ALMAGRO,5
Etc.

Los campos tienen las siguientes características.

1. **CALLE:** Calle y número de ubicación del local. Puede tener espacios y comas, y por eso puede estar entre comillas dobles. No está en blanco nunca.
2. **SUPERFICIE:** Metros cuadrados de superficie del local. No está en blanco nunca.
3. **PRECIOUSD:** Valor en dólares del local. No está en blanco nunca.
4. **PRECIOPESOS:** Valor en pesos del local. No está en blanco nunca.
5. **USDM2:** Valor en dólares del metro cuadrado. No está en blanco nunca.
6. **ANTIG:** Antigüedad del local, en años. No está en blanco nunca.
7. **EN_GALERIA:** Indica si el local está dentro de una galería o no. Puede estar en blanco.
8. **COTIZ:** Cotización del dólar en el momento en que se calculó el precio. Ej: si dice 80 el dólar estaba a -\$80. No está en blanco nunca.

TRABAJO PRÁCTICO

Sistema de Información venta Locales

9. **TRIMESTRE:** En qué trimestre del año se hizo la cotización. Puede valer únicamente **PRIMER**, **SEGUNDO**, **TERCER** o **CUARTO**. No está en blanco nunca.
10. **BARRIO:** Barrio de Ubicación. No está en blanco nunca.
11. **COMUNA:** Comuna de Ubicación. No está en blanco nunca.

Para el procesamiento, los datos del archivo pueden necesitar conversión de tipo o de contenido, los cuales se especifican a continuación:

1. **CALLE:** Se considera tipo String. Al guardar como dato separado, sacarle las comillas dobles si las tuviera.
2. **SUPERFICIE:** Se considera tipo int.
3. **PRECIOUSD:** Se considera tipo int.
4. **PRECIOPEOS:** Se considera tipo int.
5. **USDM2:** Se considera tipo int.
6. **ANTIG:** Se considera tipo int. Si es a estrenar, vale 0.
7. **EN_GALERIA:** Se considera tipo String. Puede valer únicamente **SI** o **NO**, o bien estar vacío (no se tiene el dato)
8. **COTIZ:** Se considera tipo int.
9. **TRIMESTRE:** Se considera tipo String Puede valer únicamente **PRIMER**, **SEGUNDO**, **TERCER** o **CUARTO**.
10. **BARRIO:** Se considera tipo String
11. **COMUNA:** Se considera tipo int. Son números del 1 al 15.

Los datos deben cargarse sólo una vez por ejecución. Si ya fueron, se le debe informar al usuario y no deben volver a permitir leer el archivo.

2- Estadística a Obtener

Opciones de estadística, esto debe ser un submenu:

1.- Debe generar el archivo **semestre1.csv** que contenga los datos de locales en venta de los trimestres **PRIMER** y **SEGUNDO**. Y deberán guardarse ordenados por **COMUNA** y **BARRIO** (orden de comuna de menor a mayor, y a igual comuna, orden de barrio de A a la Z).

Los datos a guardar son: **CALLE**, **SUPERFICIE**, **PRECIOUSD**, **USDM2**, **ANTIG**, **BARRIO**, **COMUNA**

Ejemplo:

```
CALLE,SUPERFICIE,PRECIOUSD,USDM2,ANTIG,BARRIO,COMUNA
"Mexico 1700",224,276000,1232,0,MONTSERRAT,1
"Rivadavia 548",150,480000,3200,0,MONTSERRAT,1
"Lavalle 700",2400,4000000,1667,0,SAN NICOLAS,1
"Florida al 400",45,87000,1933,0,SAN NICOLAS,1
"Florida al 500",31,275000,8871,0,SAN NICOLAS,1
...
```

Una vez finalizada la tarea debe visualizar por pantalla el mensaje "Archivo semestre1.csv creado con éxito".

TRABAJO PRÁCTICO

Sistema de Información venta Locales

2.- Debe generar el archivo **galeriaEstrenar.csv** que contenga los datos de locales en venta en galerías, y con antigüedad 0. Deberán guardarse ordenados por DIRECCION (de A a la Z).

Los datos a guardar son: **CALLE, SUPERFICIE, PRECIOUSD, BARRIO, COMUNA**

Ejemplo:

```
CALLE, SUPERFICIE, PRECIOUSD, BARRIO, COMUNA
"Boyaca", 113, 500000, FLORES NORTE, 7
"GALLO 400", 538, 1100000, ALMAGRO, 5
...
```

Una vez finalizada la tarea debe visualizar por pantalla el mensaje "Archivo **galeriaEstrenar.csv** creado con éxito".

3.- Debe calcular los datos de locales en venta con mayor precio en cada barrio y mostrar por pantalla ordenados por precio de mayor a menor. De haber más de uno con el mismo precio, se deben mostrar todos ellos.

Mostrar por páginas, donde cada página no debe tener más de 20 líneas. El usuario presionará ENTER para avanzar entre páginas

Los datos a mostrar son: **CALLE, SUPERFICIE, PRECIOUSD, BARRIO, COMUNA**

Ejemplo:

```
CALLE, SUPERFICIE, PRECIOUSD, BARRIO, COMUNA
"Lavalle 700", 2400, 4000000, SAN NICOLAS, 1
"Rivadavia 548", 150, 480000, MONTSERRAT, 1
...
```

4.- Generar el archivo **comunaX.txt** con el promedio de superficies de locales con más de 10 años de antigüedad en una comuna seleccionada por el usuario, discriminado por Barrio, en orden de A a Z. Este archivo no es un archivo separado por comas. Deberá mostrarse en este formato:

```
Barrio: xxx
Promedio: xxx.xx
---
Barrio: xxx
Promedio: xxx.xx
---
```

Se le tiene que mostrar al usuario un mensaje para que ingrese la comuna, que debe ser validada entre 1 y 15.

Ejemplo:

Si el usuario ingresa comuna 9

```
Barrio: Liniers
Promedio: 120.25
---
```

```
Barrio: Mataderos
Promedio: 100
---
```

```
Barrio: Parque Avellaneda
Promedio: 121
```

Una vez finalizada la tarea debe visualizar por pantalla el mensaje "Archivo creado con éxito".

5.- Volver al menú principal.

3- Agregar registros de nuevos locales.

Debe solicitarle al usuario que ingrese todos los datos necesarios, validando y cumpliendo con los siguientes requerimientos:

1. **CALLE:** Calle y número de ubicación del local. Puede tener espacios y comas, y por eso puede estar entre comillas dobles. No está en blanco nunca.
2. **SUPERFICIE:** Metros cuadrados de superficie del local. No está en blanco nunca. Debe ser positivo.
3. **PRECIOUSD:** Valor en dólares del local. No está en blanco nunca.
4. **PRECIOPESES:** Valor en pesos del local. No está en blanco nunca.
5. **USDM2:** Valor en dólares del metro cuadrado. Debe calcularse a partir del ingreso del precio en USD dividido la superficie del local.
6. **ANTIG:** Antigüedad del local, en años. No está en blanco nunca.
7. **EN_GALERIA:** Indica si el local está dentro de una galería o no. Puede estar en blanco.
8. **COTIZ:** Cotización del dólar en el momento en que se calculó el precio. Debe calcularse a partir del ingreso del precio en Pesos dividido el precio en USD.
9. **TRIMESTRE:** Se considera tipo String Puede valer únicamente **PRIMER, SEGUNDO, TERCER o CUARTO.**
10. **BARRIO:** Se considera tipo String
11. **COMUNA:** Se considera tipo int. Son números del 1 al 15.

Los datos ingresados no deben guardarse en el archivo todavía. Si el usuario calcula las estadísticas nuevamente, estos datos deben ser considerados.

Al finalizar el ingreso se debe preguntar si quiere ingresar uno nuevo LOCAL. El usuario debe contestar NO, entonces volverá al menú principal.

4- Actualizar los datos en el archivo

Debe actualizar el archivo “locales-en-venta-2020.csv” manteniendo el formato inicial.

5- Salir

De no haber actualizado los datos, debe indicar que perderá los cambios si no los guarda, y preguntarse si quiere continuar o volver al menú principal.

Consideraciones

1. Tener en cuenta que al guardar la CALLE, se pone entre comillas dobles.
2. RESPETAR LOS NOMBRES DE LOS ARCHIVOS DE SALIDA
3. Deben hacer uso de los temas vistos: Validaciones, Funciones, Listas, Listas de listas, archivos, excepciones para el manejo de archivos, etc.
4. No se deben usar módulos o tipos de datos que no hayan sido vistos en clase. Pueden desarrollar sus propios módulos que resuelvan la problemática.

Consideraciones de Entrega

- El sistema de venta debe ser escrito en lenguaje Python, en un archivo que se llame `sistema_ventasTPGrupoXComisionY.py`, donde X es el número de grupo asignado e Y es la letra de la comisión (de la A a la D). Ejemplo: `TPGrupo10ComisionA.py`
- Si decide separar las funciones en otro(s) archivo(s) (módulo(s)), debe tener en cuenta que en esto(s) archivo(s) solo deben estar las definiciones de las funciones.
- Los programas no deben fallar por ningún motivo, incluso la introducción de cualquier tipo de dato incorrecto por parte del usuario.
- El TP debe entregarse a través de una actividad en el campus, en una carpeta comprimida en un archivo .zip, llamada “TPGrupoXComisionY”. La misma debe contener:
 - El archivo “sistema_ventasTPGrupoXComisionY.py”.
 - El archivo “ `semestre1.csv`” generado.
 - El archivo “`galeriaEstrenar.csv`” generado.
 - El archivo “ `comunaX.txt`” generado.
 - `funcionesGrupoXComisionY.py`. Creado por usted con las funciones definidas por usted.

Criterios de evaluación y aprobación.

Debe hacer las validaciones correspondientes para evitar que el programa salga en forma abrupta.

Además, se valorará que cumpla con las características de eficiencia y estilo. Un programa es eficiente cuando usa los recursos en forma adecuada.

TRABAJO PRÁCTICO

Sistema de Información venta Locales

Un programa tiene estilo cuando está escrito de manera clara y simple. Esto incluye respetar las partes del programa, usar variables, funciones con nombres significativos, no repetir código.

El código tiene que ser claro y simple, con una correcta modularización en funciones. Las funciones no pueden ser muy extensas. Además, respetar el orden de las distintas partes del programa: imports, definición de funciones, programa principal.

No se deben usar variables globales. Todos los datos que serán usados en la función deben ser pasados como parámetros y retornados por medio del *return*.

Todo lo visto en clase es suficiente para realizar el trabajo. No se pueden usar módulos no autorizados.

Si el programa al ejecutarse por primera vez muestra un error en alguna línea del programa (salvo la referencia a la ruta de los archivos) se considerará **desaprobado**.

Si en el programa no se desarrolla ninguna función y/o ninguna estructura vista en clase se considerará **desaprobado**.

Errores graves de estilo o eficiencia **descontarán puntos**.

Preguntas Frecuentes

1. Es importante que haga el TP?

Si. Debe tener presente que este programa será usado para realizar el segundo parcial de la materia, por lo cual es importante que funcione perfectamente y lo más importante es poniendo en práctica en un caso de la vida real todos los conocimientos adquiridos durante el cuatrimestre.

2. Puedo usar las funciones split, strip, sum, MAX, MIN, sort, sorted,... ?

No. No pueden usar funciones que no se hayan autorizado durante la cátedra, si las quieres usar las debe programar. En caso de duda debes preguntar a los profesores.

3. Puedo cambiar el nombre de los archivos que me piden generar.?

No. Debes respetar los formatos dados y pedidos durante el proyecto. Todo proyecto trabajado en equipo debe tener reglas que permitan incrementar la eficiencia del trabajo en equipo. Adicionalmente el no respetar la consigna dada baja puntos.

4. El proyecto se recupera?

No. El proyecto no se recupera.

4. Si saca una nota menor a 8 en el TP puedo promocionar la materia?

No. La nota mínima del TP para promocionar es 8.

5. Si no hago uso de funciones o hago sólo una se me bajan puntos ?

Si. Es importante el concepto de modularidad y usted debe tener presente que cada tarea deberá realizarse en una función distinta, lo cual permite hacer reusabilidad

6. Si subo al campus la entrega más de una vez, se considera la última subida?

No. Es necesario verificar la entrega subida al campus, respetando el límite de fecha y hora establecidos para la entrega. En caso de subir más de una versión, se considerará la última subida dentro del tiempo estipulado.