

Tecnología Digital 1: Introducción a la Programación

Trabajo Práctico 2

Escuela de Negocios, UTDT - Primer semestre de 2023

El objetivo de este Trabajo Práctico es escribir programas para procesar y efectuar consultas sobre datos públicos de departamentos en venta en la ciudad de Buenos Aires, como parte de un (imaginario) sistema informático usado para estudios de mercado.

El archivo adjunto [departamentos-en-venta-2020-curado.csv](#) fue descargado el 12/5/2023 de la página web de Buenos Aires Data (<https://data.buenosaires.gob.ar/dataset/departamentos-venta>). Contiene información de más de 150 mil departamentos publicados durante el año 2020, incluyendo la dirección, el precio de publicación, la cantidad de ambientes y otros.¹

El objetivo de este Trabajo Práctico es escribir programas para modelar, procesar y efectuar algunas consultas sobre estos datos. En concreto, se pide:

1. Implementar una [clase Departamento](#) que modele un departamento individual en este problema y las consultas que se le pueden hacer. Debe tener al menos los siguientes atributos y métodos:
 - `Departamento(...)`: construye un nuevo objeto de la clase `Departamento`, con los parámetros que se consideren necesarios.
 - `d.direccion`: la dirección del departamento. Ejemplo: "Pje Cnel J F Bogado 4561".
 - `d.metros_cuadrados`: la superficie del departamento en metros cuadrados, de tipo `float`.
 - `d.precio_en_dolares` y `d.precio_en_pesos`, ambos de tipo `int`.
 - `d.m2_en_dolares` y `d.m2_en_pesos`, ambos de tipo `int`.
 - `d.ambientes`: la cantidad de ambientes del departamento.
 - `d.barrio`, de tipo `str`.
 - `str(d)`: devuelve una representación como string del departamento `d` (esto se logra definiendo el método `__repr__`), que debe incluir entre símbolos `<` y `>`: su superficie en metros cuadrados (redondeada al entero más cercano), el string `M2`, una arroba y su barrio. Ejemplos: `'<64M2@SAAVEDRA>'`, `'<30M2@VILLA CRESPO>'`.
2. Implementar una [clase DataSetInmobiliario](#) que encapsule el concepto de una colección de datos de (muchos) departamentos y las consultas que se puede hacer sobre ella. Debe tener al menos los siguientes métodos:
 - `DataSetInmobiliario(archivo_csv)`: construye un dataset a partir del archivo CSV pasado como argumento. Por ejemplo, `DataSetInmobiliario('mis-datos.csv')` construye un objeto de la clase `DataSetInmobiliario` conteniendo los datos de los departamentos de un archivo llamado `mis-datos.csv`.
 - `d.tamano()`: devuelve la cantidad de departamentos en el dataset `d`. Debe tener complejidad temporal $O(1)$ en el peor caso.
 - `d.barrios()`: devuelve el conjunto de todos los barrios existentes en el dataset `d`. Debe tener complejidad temporal $O(N * B)$ en el peor caso, donde N es la cantidad de departamentos y B la cantidad de barrios en `d`.

¹Del listado original se eliminaron 38 departamentos sin barrio especificado o con una cantidad inválida de ambientes. Además, se corrigió el error de codificación "NU?æEZ" por "NUÑEZ" en los barrios.

- `d.deptos_del_barrio(barrio)`: devuelve una lista con los departamentos del dataset `d` que están en el barrio indicado, ordenados por precio (en dólares) de menor a mayor. Esta operación debe tener complejidad temporal $O(N)$ en el peor caso, donde N es la cantidad de departamentos en `d`.
- `d.oferta_por_barrio(ambientes)`: devuelve un diccionario que indica la cantidad de departamentos con los ambientes indicados en venta en cada barrio. Por ejemplo, para el dataset completo de `departamentos-en-venta-2020-curado.csv`, retorna un diccionario que (entre otra información) contiene:²

```
{...
  'BOCA': 256,
  'PUERTO MADERO': 521,
  'VILLA LUGANO': 335,
  ...}
```

La complejidad temporal de `oferta_por_barrio` debe ser $O(N * B)$ en el peor caso, donde N es la cantidad de departamentos y B la cantidad de barrios en `d`.

- `d.filtrar(prec_min, prec_max, conj_barrios)`: modifica el dataset `d`, de manera que queden solamente los departamentos con precio en dólares dentro del rango dado (inclusive en ambos extremos) y que estén en un barrio perteneciente al conjunto de barrios provisto. La complejidad temporal de `filtrar` debe ser $O(N * B)$ en el peor caso, donde N es la cantidad de departamentos y B la cantidad de barrios en `d`.
- `d.exportar(archivo_csv)`: genera un archivo con nombre `archivo_csv` con las mismas columnas que el archivo `departamentos-en-venta-2020-curado.csv`, que contenga los departamentos en el dataset `d` (en cualquier orden). Por ejemplo, luego de ejecutar el siguiente código para algún archivo `mis-datos.csv`:

```
1 d1:DataSetInmobiliario = DataSetInmobiliario('mis-datos.csv')
2 d1.exportar('copia.csv')
3 d2:DataSetInmobiliario = DataSetInmobiliario('copia.csv')
```

los datasets `d1` y `d2` deben tener exactamente los mismos departamentos.

Modo de entrega:

Se deben entregar los siguientes archivos, respetando los nombres con exactitud:

- **departamento.py**, con la definición de la clase `Departamento`.
- **departamento-test.py**, con los tests de unidad de la clase `Departamento`.
- **dataset.py**, con la definición de la clase `DataSetInmobiliario`.
- **dataset-test.py**, con los tests de unidad de la clase `DataSetInmobiliario`.
- **informe.pdf**, un documento en el cual se explique por qué cada método cumple el orden requerido. Incluir también cualquier aclaración adicional que se considere necesaria sobre cualquier parte del trabajo. Se espera que este documento sea conciso, de tres o cuatro páginas, y en formato PDF.
- Los archivos CSV necesarios para testear `DataSetInmobiliario`.

La carpeta adjunta `templates` contiene archivos de ejemplo para usar como referencia.

²Al imprimir el diccionario, los pares `clave:valor` podrían aparecer en otro orden.

Observaciones:

- El trabajo se debe realizar en grupos de **tres personas**. La entrega consistirá en un trabajo original realizado íntegra y exclusivamente por las personas que integran el grupo. Se espera que las tres personas participen en toda la resolución. Se desaconseja fuertemente dividir y repartirse el trabajo.
- La fecha límite de entrega es el **jueves 15 de junio a las 23:55**. Los TPs entregados fuera de término serán aceptados hasta 48h más tarde, pero la demora incidirá negativamente en la nota.
- Los archivos deben subirse a la *Entrega del TP2* en la página de la materia en el campus virtual.
- El código entregado debe ejecutarse correctamente en Python3, y sólo pueden usarse elementos de Python vistos en clase. En particular, está permitido usar el método `ls.sort()` de listas. Ante la duda, consultar.
- Sólo está permitido importar las bibliotecas `typing` (para sugerencias de tipos), `unittest` (para el testing de unidad) y `csv` (para la lectura de archivos CSV; sugerimos usar la clase `csv.DictReader`).
- Es obligatorio especificar el tipo de todas las variables y funciones mediante sugerencias de tipos (*type hints*). Las funciones deben especificarse con el formato de *docstrings* visto en clase.
- Los archivos `departamento.py` y `dataset.py` no deben tener código principal; solamente definiciones de clases. El objetivo es que dichos archivos sean importados y usados por otros programas, como parte de un proyecto más grande.
- Durante el desarrollo y para el testing de unidad, se recomienda **evitar trabajar sobre el archivo CSV completo**. Por su tamaño, su lectura demora tiempo en cada ejecución y resulta difícil saber si los resultados son correctos. En cambio, conviene crear archivos CSV pequeños, conteniendo pocos registros tomados del archivo original, para tener mayor control de las ejecuciones.
- Puede suponerse que los objetos y sus métodos siempre se usarán de manera correcta. Es decir, si hay errores en la cantidad, tipo o valor de los argumentos, no se espera ningún comportamiento particular (la ejecución podría colgarse o terminar en un error, por ejemplo).
- **Sobre la evaluación del TP:** se tendrá en cuenta no solamente que el código sea correcto, sino también que sea claro, ordenado y modular, que esté comentado adecuadamente, que cumpla los órdenes de complejidad requeridos (con las justificaciones correspondientes), que el testing de unidad tenga un cubrimiento adecuado, que se respete con cuidado todo lo pedido en el enunciado, y que los archivos entregados sigan las indicaciones dadas.