

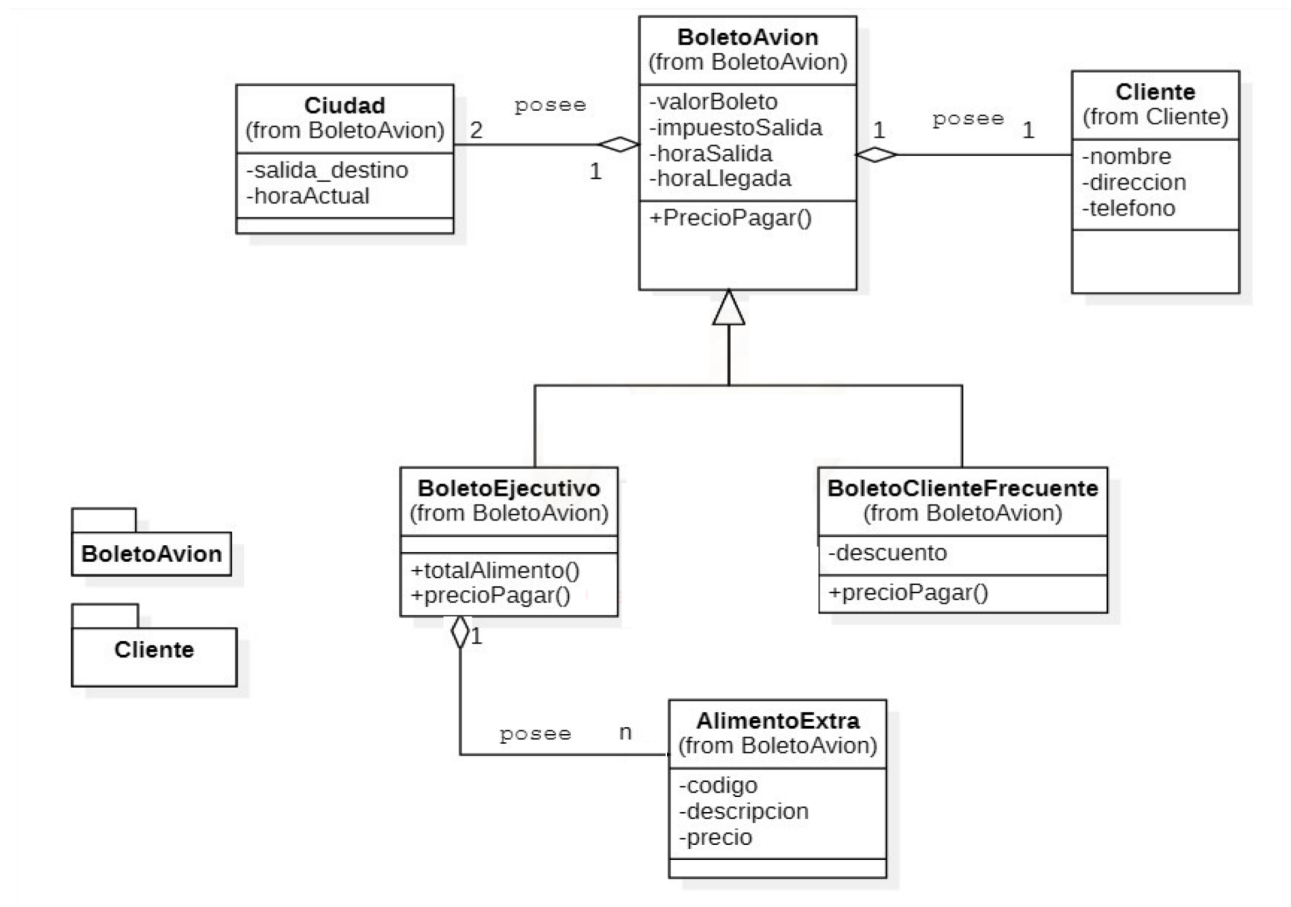
TAREA NÚMERO 10

- Las tareas son estrictamente de carácter individual, tareas idénticas se les asignará cero puntos.
- Todas las tareas tienen el mismo valor en la nota final del curso, es decir, el promedio de las notas obtenidas en la tareas será la nota final del curso.
- Todas las preguntas tienen el mismo puntaje.
- Incluir al menos 3 casos de prueba en las funciones programadas (cuando corresponda).

Escriba en Python con los siguientes ejercicios con al menos una prueba de ejecución de cada clase. Todas las clases deben seguir el estilo correcto de python (estilo pythónico) y poseer un método `__str__`.

1. [20 puntos] Una Aerolínea desea implementar un sistema para emitir boletos de avión con los siguientes datos:
 - Los Boletos de Avión tienen: NombreCliente, DirecciónCliente, TeléfonoCliente, ValorBoleto, ImpuestoSalida, PrecioPagar = ValorBoleto + ImpuestoSalida; HoraSalida, HoraLlegada, CiudadSalida, CiudadDestino, además para cada una de estas ciudades se almacena la HoraActual.
 - Existen también Boletos de Avión para Clientes Frecuentes que tienen además un PorcentajeDescuento, de modo tal que $\text{PrecioPagar} = \text{ValorBoleto} + \text{ImpuestoSalida} - \text{PorcentajeDescuento} * \text{ValorBoleto}$.
 - Los Boletos de Avión para Ejecutivos tienen también una lista de Alimentos Extras, donde cada alimento extra tiene Código, Descripción y Precio, así para los Boletos de Avión para Ejecutivos se tiene que el $\text{TotalAlimentos} = \text{suma de cada uno de los precios de la lista de alimentos}$, y el precio a pagar del boleto se calcula como sigue: $\text{PrecioPagar} = \text{ValorBoleto} + \text{ImpuestoSalida} + \text{TotalAlimentos}$.

El diseño UML del problema anterior se muestra en la figura siguiente. Programe en Python (estilo pythónico) esta jerarquía de clases:



2. [20 puntos] Programe una clase en Python (estilo pythónico) denominada **Fracción** que tiene dos atributos (números reales), el **numerador** y el **denominador**. Además, usted debe programar métodos que reciben dos números racionales (fracciones) y retornan la suma, la resta, la multiplicación y la división entre estas dos fracciones (el resultado deberá ser una fracción de la forma a/b). El método `__str__` debe imprimir la fracción de la forma: **numerador/denominador**. Es decir, use las siguientes definiciones:

- Suma $\frac{a}{b} + \frac{c}{d} = \frac{ad + cb}{bd}$.
- Resta $\frac{a}{b} - \frac{c}{d} = \frac{ad - cb}{bd}$.
- Multiplicación $\frac{a}{b} \cdot \frac{c}{d} = \frac{ac}{bd}$.
- División $\frac{a}{b} \div \frac{c}{d} = \frac{a d}{b c}$.

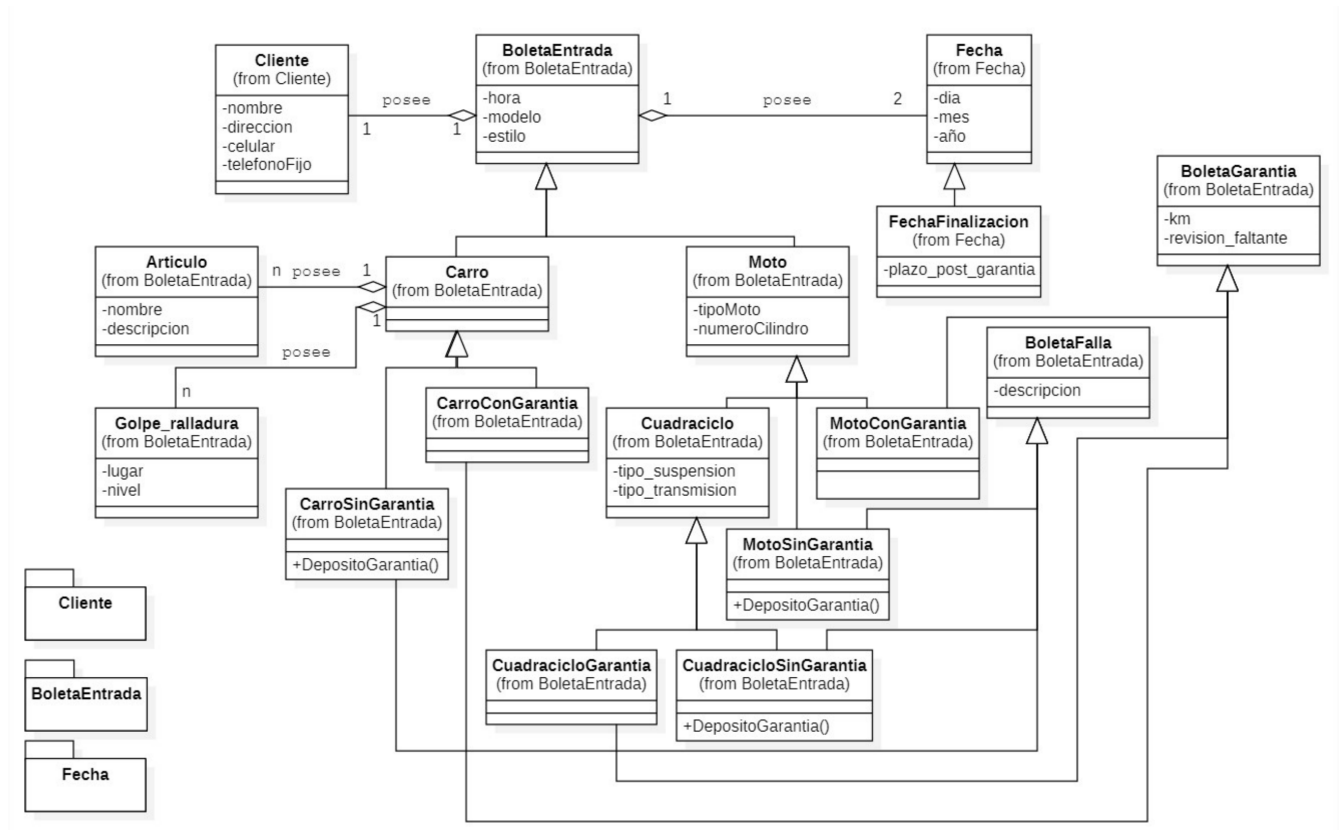
Por ejemplo, el resultado de restar $\frac{7}{5} - \frac{3}{4} = \frac{13}{20}$ y deberá imprimirse usando el método `__str__` de la forma 13/20.

3. [20 puntos] Agregue a la clase `class mi_DF()` vista en clase los siguientes métodos:

- Retorna la cantidad de valores de este DataFrame que son divisibles entre 3 (Pruebe este método leyendo un archivo de datos, esto en un Script de pruebas).

- Recibe dos números de columna y que retorna en un diccionario con el nombre de las variables correspondientes a las columnas, la covarianza y la correlación entre esas dos variables (Pruebe este método leyendo un archivo de datos, esto en un Script de pruebas).
 - Reescriba la clase `class mi_DF()` pero heredando de la clase `DataFrame` de la biblioteca `pandas` en lugar de tener una relación componente-compuesto. Incluya las funciones creadas en los dos puntos anteriores.
4. [20 puntos] Desarrolle una clase denominada **Análisis** que tiene como atributo una matriz numérica tipo `numpy` y defina los siguientes métodos:
- `as_data_frame`: retorna la matriz convertida en `DataFrame` de `pandas`.
 - `desviacion_estandar`: retorna un diccionario con la Desviación Estándar de cada columna.
 - `media`: retorna un diccionario con la Media de cada columna.
 - `mediana`: retorna un diccionario con la Mediana de cada columna.
 - `maximo`: retorna el máximo de toda la matriz.
 - `buscar`: recibe un número y busca el valor en la matriz, retorna los índices del primer valor encontrado o `None` en caso de no encontrar el valor.
5. [20 puntos] La compañía **HONDA-MOTOR** desea implementar un sistema de control de las entradas de vehículos a su taller de servicio mecánico. El objetivo principal del sistema es emitir e imprimir la boleta de control de acceso al taller que es entregada al cliente como garantía de su vehículo. Para esto se tiene la siguiente información:
- Una Boleta de entrada tendrá al menos los siguientes datos: Hora y Fecha de Entrada y Fecha estimada de Salida del vehículo, modelo (ej. 2001), estilo (ej. Civic), los datos del cliente, a saber Nombre, Dirección, Número de Teléfono celular y fijo. Una razón de entrada que puede ser revisión de garantía o falla mecánica, en caso de revisión de garantía se incluye cuál es (1000 km, 5000 km, 10000 km, 20000 km, . . . , 100000 km) y si ha faltado o no a una revisión anterior. En caso de entrada por falla mecánica se incluye una descripción de la misma.
 - El taller **HONDA-MOTOR** recibe automóviles, 4×4 y motocicletas (de 2 ruedas y cuadraciclos) para el caso de automóviles y 4×4 se tiene además una lista de artículos que se encuentran dentro del vehículo, de modo que para cada artículo se tiene un nombre y una descripción. Para automóviles y 4×4 también se tiene una descripción del estado de la carrocería con una lista de golpes o ralladuras en las que se indica el lugar donde se encuentra y el nivel (leve, media o fuerte).
 - Para motocicletas se tiene un descriptor del tipo de motor que indica si es 4 tiempos o 2 tiempos y el número de cilindros. Para los cuadraciclos se incluye además el tipo de suspensión (independiente o no) y el tipo de transmisión (cadena, faja, barra, etcétera).
 - Al taller ingresan vehículos con y sin garantía. Una garantía incluye fecha de emisión, fecha de finalización (que tiene día-mes-año y un plazo en meses de post-garantía). Si el vehículo no tiene garantía el cliente tendrá que pagar un de-pósito de garantía que se calcula como sigue: Si es un automovil o 4×4 la fórmula es `depósito_garantía = modelo*100`, si es moto `depósito_garantía = modelo*10` y si es cuadraciclo entonces `depósito_garantía = modelo*150 + estimado_del_mecánico`.

El diseño UML del problema anterior se muestra en la siguiente figura. Programe en Python esta jerarquía de clases:



Entregables: Genere desde Jupyter un documento autoreproducible con la solución de la tarea y súbalo en el Aula Virtual, incluya casos de prueba en todos los ejercicios.



oldemar rodíguez
CONSULTOR en MINERÍA DE DATOS