Profesor: Dr. Oldemar Rodríguez Rojas
PF-1319 y PF-1320 Análisis de Datos II
Fecha de Entrega: Domingo 4 de septiembre a las 12 media noche
Instrucciones:

- La solución a cada tarea se debe subir en el aula virtual, no pueden ser enviadas por correo.
- Las tareas son estrictamente individuales.
- Tareas idénticas se les asignará cero puntos.
- Todas las tareas tienen el mismo valor en la nota final del curso.
- Las tareas se pueden entregar tarde, pero cada día de atraso tendrá un rebajo de 20 puntos.

Tarea Número 2

Escriba en Python 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 los gets, sets y un método __str__. Incluya todas estas clases programadas en un mismo módulo, luego desarrolle un programa que importa este módulo y ahí se hacen las pruebas de ejecución de la cada una de las clases programadas.

1. Desarrolle en Python las clases Triángulo y Rectángulo, luego para cada una de ellas programe los métodos `calcular_area` y `calcular_perimetro`:
   - `Clase Triángulo`: Tiene como atributos el lado, la base y altura del triángulo.
   - `Clase Rectángulo`: Tiene como atributos la base y altura del rectángulo.
2. Desarrolle en pyhon la clase `Operación`, que tiene como atributos dos vectores numéricos U y V luego programe los métodos:
   - `Sumar`: Suma ambos vectores y devuelve el resultado.
   - `Restar`: Resta al primer vector el segundo y devuelve el resultado.
   - `Multiplicar`: Calcula el producto punto entre ambos vectores y devuelve el resultado. Es decir, si $U = (u_1, u_2, \ldots, u_n)$ y $V = (v_1, v_2, \ldots, v_n)$ entonces:

$$U \cdot V = u_1v_1 + u_2v_2 + \cdots + u_nv_n = \sum_{i=1}^{n} u_iv_i.$$

   - `Correlacion`: Devuelve la correlación entre los dos vectores.
   - `Covarianza`: Devuelve la covarianza entre los dos vectores.
3. Programe una clase en Python (estilo pythónico) denominada `Jugadores` que tiene como atributo un dataframe de `pandas` que permite almacenar una tabla de datos como la incluida en el archivo `Players1.csv`. Además, programe los siguientes métodos:
   - `actualizar_position`: Recibe el nombre del jugador y el nombre de la nueva posición, dicho método actualiza la posición del jugador y la muestra.

- `resumen_jugador`: Recibe el nombre del jugador y retorna toda su información en formato de texto (`str`).
- `resumen_columna`: Recibe el nombre de una columna y si es numérica retorna su promedio, en caso de ser categórica (`object`) retorna la moda.
- `cantidad_equipos`: Retorna la cantidad de jugadores que existen en el dataframe por equipo, ejemplo Algeria: `10 jugadores`.
- `agregar_jugador`: Recibe el nombre del jugador y los valores de cada columna del dataframe, añade el jugador a la tabla de datos y retorna un mensaje donde se indica que se agregó correctamente. En el siguiente link se muestran algunas maneras de agregar filas a un dataframe: agregar filas.
- `eliminar_jugador`: Recibe el nombre del jugador, lo elimina de la tabla de datos y retorna un mensaje donde se indica que se eliminó correctamente. Debe validar que el jugador exista dentro de la tabla de datos.

Verifique la correctitud de esta clase usando la tabla `Players1.csv` como atributo.

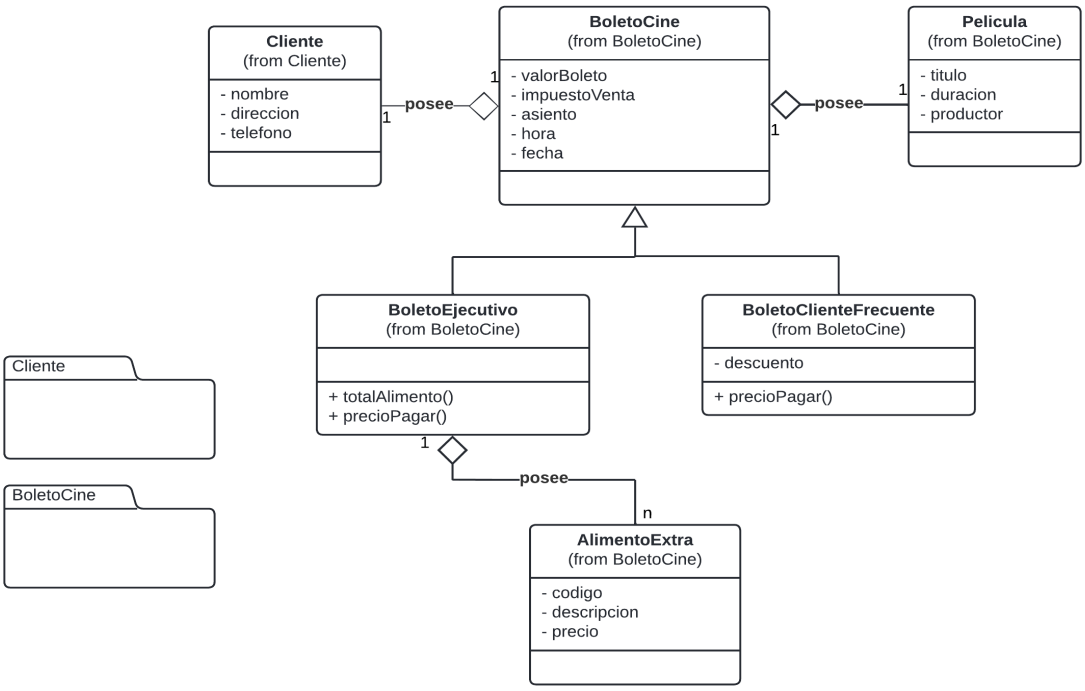
4. Desarrolle una clase denominada `Análisis` que tiene como atributo una matriz numérica tipo `numpy` y defina los siguientes métodos:

   - `as_data_frame`: retorna la matriz convertida en DataFrame de `pandas`.
   - `desviacion_estandar`: retorna un diccionario con la Desviación Estándar de cada columna.
   - `varianza`: retorna un diccionario con la Varianza de cada columna.
   - `moda`: retorna un diccionario con la Moda 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. Una Cadena de Cine desea implementar un sistema para emitir boletos de cine con los siguientes datos:

   - Los Boletos de Cine tienen: `NombreCliente, DirecciónCliente, TeléfonoCliente, ValorBoleto, ImpuestoVenta, PrecioPagar = ValorBoleto + ImpuestoVenta; asiento, hora fecha`, además para cada uno se almacena también la información de la `película; título, duración` y `productor`.
   - Existen también Boletos de Cine para Clientes Frecuentes que tienen además un descuento, de modo tal que `PrecioPagar = ValorBoleto + ImpuestoVenta − descuento * ValorBoleto`.
   - Los Boletos de Cine para Ejecutivos tienen también una lista de Alimentos Extras, donde cada alimento extra tiene `Código, Descripción` y `Precio`, así para los Boletos de Cine para Ejecutivos se tiene que el `TotalAlimentos = suma de cada uno de los precios de la lista de alimentos`, y el precio a pagar del boleto se calcula como sigue: `PrecioPagar = ValorBoleto + ImpuestoVenta + TotalAlimentos`.

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



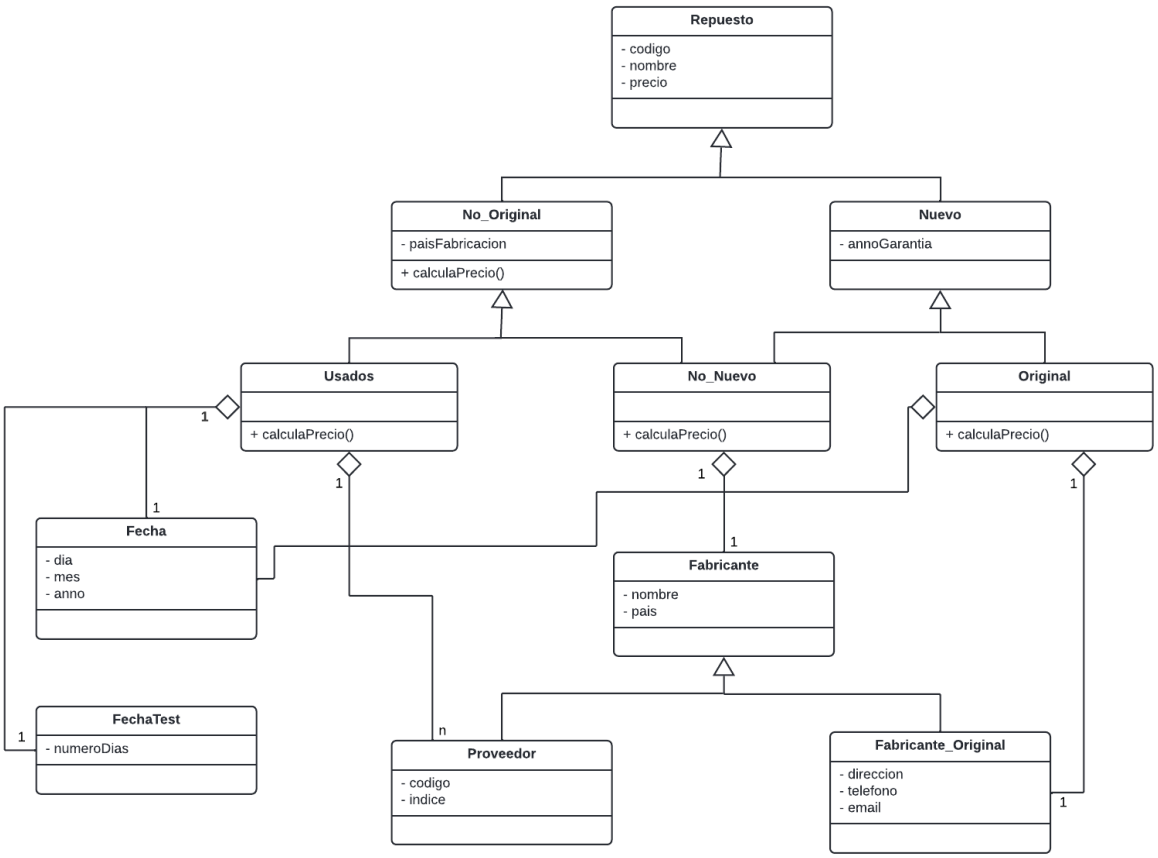
6. Una mecánica desea implementar un sistema para gestionar sus repuestos con los siguientes datos:

   a) Se venden repuestos los cuales tienen al menos un código, un nombre y un precio base. Hay repuestos originales y no originales. Los repuestos originales tienen un plazo de garantía dado en años, una fecha de fabricación (día, mes y año) y su precio de venta se calcula sumando al precio base un 25 % de utilidad. Mientras que los repuestos no originales tienen un precio de venta que se calcula sumando al precio base un 10 % de utilidad.

   b) Los repuestos no originales se dividen en repuestos usados y repuestos nuevos. Los repuestos no originales nuevos tienen también un número de años de garantía, un fabricante y además su precio de venta se calcula sumando al precio un 5 % de utilidad adicional por cada año de garantía.

   c) Los repuestos usados tienen una lista de proveedores, donde para cada proveedor se almacena el código, nombre, el país de origen y un índice que indica el nivel de calidad del repuesto, además los repuestos usados tienen una fecha de facturación, una fecha de test (en la cual se probó el repuesto que tiene además un plazo máximo dado de días que establece para el periodo de garantía).

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

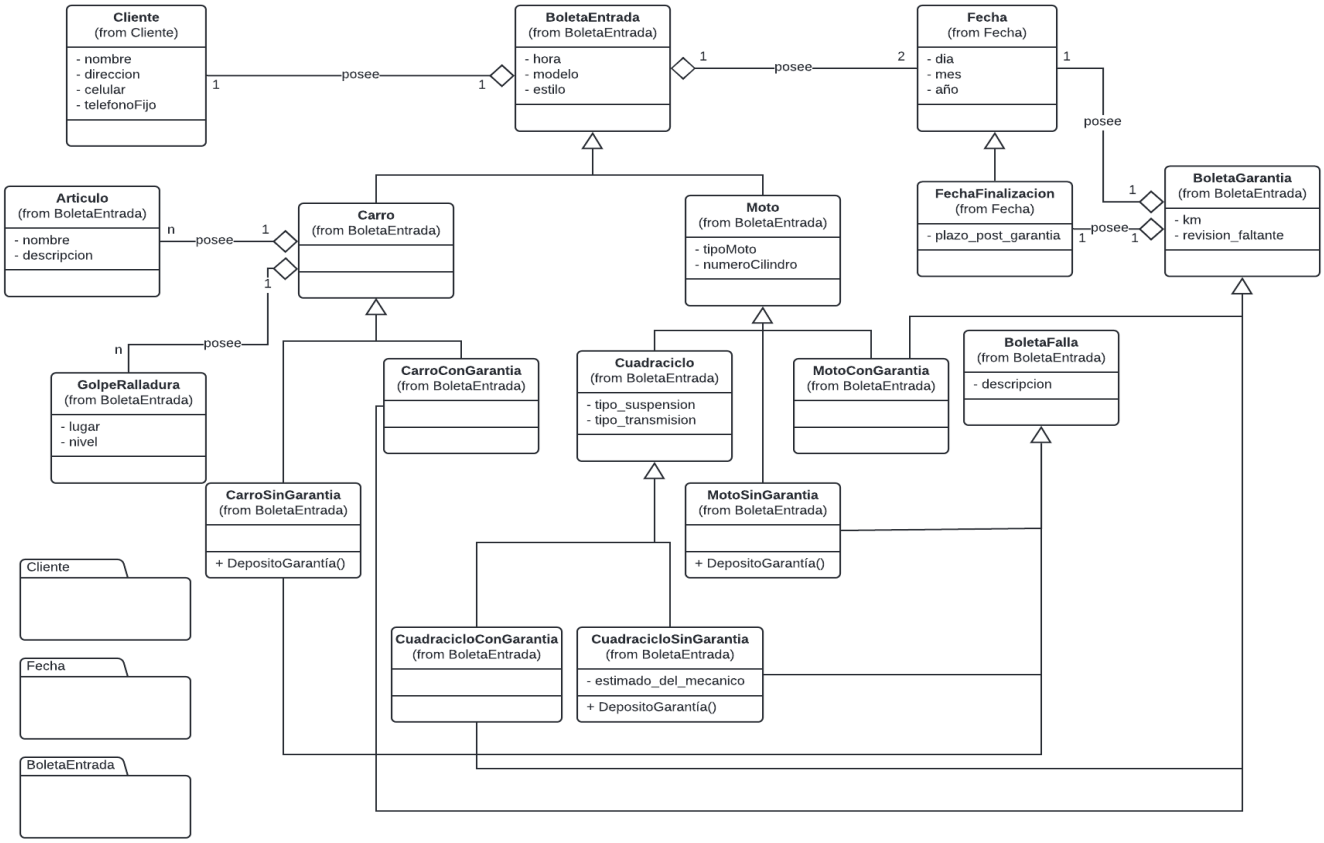


7. **[Optativo 20 puntos]**[1] La compañía HONDA-MOTOR desea implementar un sistema de control de las entradas de vehículos a su taller de servicio mecánico. El objetivo principal del sistema es emitir e imprimir la boleta de control de acceso al taller que es entregada al cliente como garantía de su 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).

[1] La nota máxima posible que se puede obtener en la tarea es 100.

- 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 depósito 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.



Entregables: Debe subir en el Aula Virtual el script de pruebas y un documento autoreproducible con la solución de la tarea.