PROGRAMACIÓN AVANZADA

Guía de Trabajo Práctico N° 2

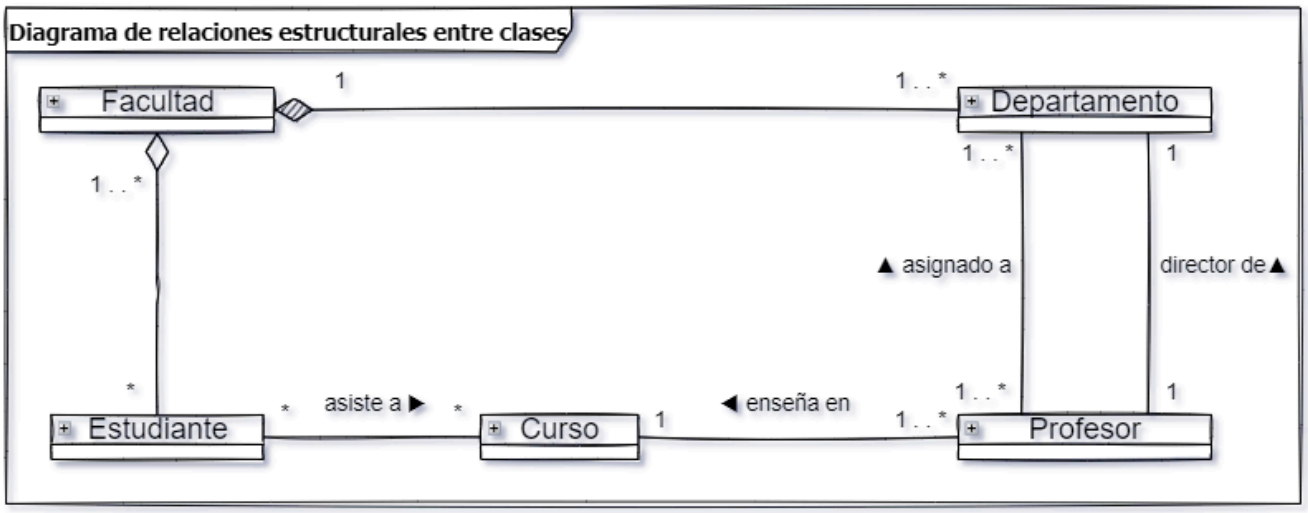
Diseño y programación orientada a objetos

Objetivos:

- Aplicar conceptos de diseño y programación orientada a objetos en la resolución de problemas y proyectos simples.

1 *Sistema de Información Universitaria*

La siguiente figura muestra un posible diseño de relaciones estructurales entre clases para un sistema de información universitaria de cursos, estudiantes y docentes de una universidad.



Del diagrama se pueden deducir las siguientes relaciones:

- La facultad puede tener cero o más estudiantes y cada estudiante puede ser miembro de una o más facultades.
- La facultad tiene al menos uno o más departamentos y cada departamento pertenece a una única facultad.
- Los estudiantes asisten a cursos. Cada estudiante puede asistir a cualquier número de cursos y cada curso puede tener cualquier número de estudiantes.

- Cada profesor puede pertenecer a uno o más departamentos y cada departamento puede tener uno o más profesores.
- Para cada departamento hay un único profesor que es el director del departamento y un profesor puede ser director de un departamento a lo sumo.

Teniendo en cuenta las relaciones detalladas previamente se solicita:

a) Extender el diagrama mostrado para que represente las clases, definiendo los atributos y métodos necesarios para atender a las responsabilidades elementales presentes en el diagrama. Por ejemplo, una responsabilidad que emerge consiste en que algunas clases tengan la posibilidad de añadir objetos de otra clase como sus componentes durante la inicialización, o como agregación posteriormente a la inicialización, también, podría ser interesante si la institución pudiera listar sus estudiantes, entre otras posibilidades.
b) Identificar y plantear en el diagrama posibles jerarquías (de contención y herencia) entre clases justificando la propuesta.
c) A partir del diagrama resultante del punto b), escribir el código de las clases correspondientes.
d) Una vez resueltos los ítems anteriores, construir un programa con una interfaz de consola que despliegue el siguiente menú de opciones:

```
########################################
#  Sistema de Información Universitaria  #
########################################
Elige una opción
1 - Inscribir alumno
2 - Contratar profesor
3 - Crear departamento nuevo
4 - Crear curso nuevo
5 - Inscribir estudiante a un curso
6 - Salir
```

- Las opciones 1 y 2 deben permitir ingresar desde la interfaz de consola los datos personales del inscripto/contratado.
- En la opción 3 se debe seleccionar el director del departamento nuevo entre los profesores presentes en el sistema. Al finalizar, listar todos los departamentos existentes.
- En la opción 4, al igual que en la opción anterior, se debe seleccionar el titular del curso nuevo entre los profesores presentes en el sistema. Listar los cursos existentes en el departamento donde se creó el curso inmediatamente después de la creación.

- En la opción 5, seleccionar el estudiante a inscribir entre los alumnos presentes en el sistema.

Para agilizar la corrección de este problema, al inicio del programa, cargar de forma automática 4 alumnos y 4 profesores leyendo su información personal desde un archivo txt.

2 *Proyecto Cinta transportadora*

Diseñar, desarrollar y testear un programa que controle el funcionamiento de una cinta transportadora y la carga de alimentos en cajones mediante una interfaz web. La cinta transporta frutas y verduras y posee sensores que detectan e informan el tipo de alimento y su peso. Además, el programa debe controlar la carga de cajones con una cantidad de elementos "N" fijada por el usuario. Estos cajones son almacenados y transportados para su distribución final en comercios minoristas.

Un factor importante para garantizar la conservación de un alimento es conocer su **actividad de agua o actividad acuosa** (**aw,** factor entre 0 y 1). La **aw** es una medida de la proporción de agua disponible en el alimento, sustancia fundamental en reacciones biológicas o químicas (por ejemplo, proliferación de bacterias, levaduras, etc.) y, por lo tanto, depende del contenido de agua del alimento, que a su vez se relaciona con la masa del mismo.

El transporte de los cajones se realiza en camiones refrigerados para la correcta conservación de los alimentos. A fin de garantizar la óptima calidad y seguridad de los alimentos en cada cajón, se mide la **aw** de los alimentos para establecer el tiempo de vida útil y aplicar métodos para prolongarla si es preciso (por ejemplo, mediante agregado de solutos, alteración de ph, envasado al vacío, etc.). Un alimento con **aw > 0.90** se considera susceptible a la proliferación de microorganismos.

En la primera versión del programa, se trabajará con 2 frutas (kiwi y manzana) y 2 verduras (papa y zanahoria), el programa debe calcular y reportar:

- El peso total del cajón
- La actividad acuosa promedio de cada alimento (**aw_prom_*:** promedio de las **aw** de cada alimento, por ejemplo **aw_prom_kiwi**)
- La actividad acuosa promedio por tipo de alimento (**aw_prom_frutas** y **aw_prom_verduras**)
- La actividad acuosa promedio total del conjunto de alimentos (**aw_total**)

En versiones futuras del programa se incorporarán otras frutas y verduras, esto debe contemplarse en el diseño, de forma tal que las nuevas incorporaciones no produzcan grandes cambios en el proyecto.

La actividad acuosa se calcula de forma distinta para cada alimento, usando las siguientes fórmulas:

$$aw_{manzana} = 0.97 \cdot \frac{(Cm)^2}{(1+(Cm)^2)}, \quad C = 15\,kg^{-1}$$

$$aw_{kiwi} = 0.96 \cdot \frac{1-e^{-Cm}}{1+e^{-Cm}}, \quad C = 18\,kg^{-1}$$

$$aw_{papa} = 0.66 \cdot \arctan(Cm), \quad C = 18\,kg^{-1}$$

$$aw_{zanahoria} = 0.96 \cdot (1 - e^{-Cm}), \quad C = 10\,kg^{-1}$$

Los resultados de estas fórmulas están limitadas a valores entre 0 y 1 para valores de la masa m menores a 600 g. El programa debe avisar al operario mediante un mensaje de advertencia si alguno de los promedios calculados supera el valor de 0.90.

Un ejemplo de interfaz se muestra en la figura 1. Al pulsar el botón de inicio empieza el transporte de los alimentos y la detección en la cinta transportadora. La detección del alimento se va a simular usando una clase provista por la cátedra en el siguiente enlace. Esta función retorna la información del alimento detectado en un diccionario con claves: “alimento” y “peso”.

La detección del alimento en la cinta transportadora puede fallar en algunas ocasiones, devolviendo información indefinida (“undefined”). En este caso, el alimento debe ser desviado del recorrido sin cargarlo al cajón, sin embargo, debe mantenerse el número “N” de alimentos por cajón fijado inicialmente.

Para medir la **aw_prom** de los alimentos, considerar al cajón como un contenedor iterable de forma que el mismo pueda ser recorrido usando un ciclo **for** de la siguiente manera:

```
for alimento in cajon:
      print(alimento)
```

Figura 1: Vista de Interfaz web del programa.

Entregas:

a) Diagrama UML mostrando las relaciones entre las clases identificadas en el diseño.

b) Implementación del programa con interfaz web que cumpla con el diseño del diagrama UML y los requerimientos indicados en el enunciado.

c) Implementación de un conjunto de pruebas unitarias que tenga una cobertura del código base mayor al 50%.