

Programación Avanzada

LABORATORIO 1

Consideraciones Generales:

- La entrega podrá realizarse hasta el **domingo 21 de abril hasta las 23:59 hs.**
- Un integrante de cada grupo deberá subir la entrega de su equipo a través del receptor de entregas en el Campus Virtual del curso: <https://cv.utu.edu.uy/moodle/course/view.php?id=39461>
- El entregable se deberá llamar *lab1GrX.zip*, donde X es el número de grupo indicado. El mismo deberá contener dos directorios;
 - *Parte 1* que contenga el código fuente (.h y .cpp) y el archivo *makefile*.
 - *Parte 2* que contenga en formato *pdf* un documento que contenga el Modelo de Dominio utilizando la plantilla disponible en el Campus Virtual.
- Las entregas que no cumplan estos requerimientos no serán consideradas.
- El hecho de no realizar una entrega implica la insuficiencia del laboratorio completo.

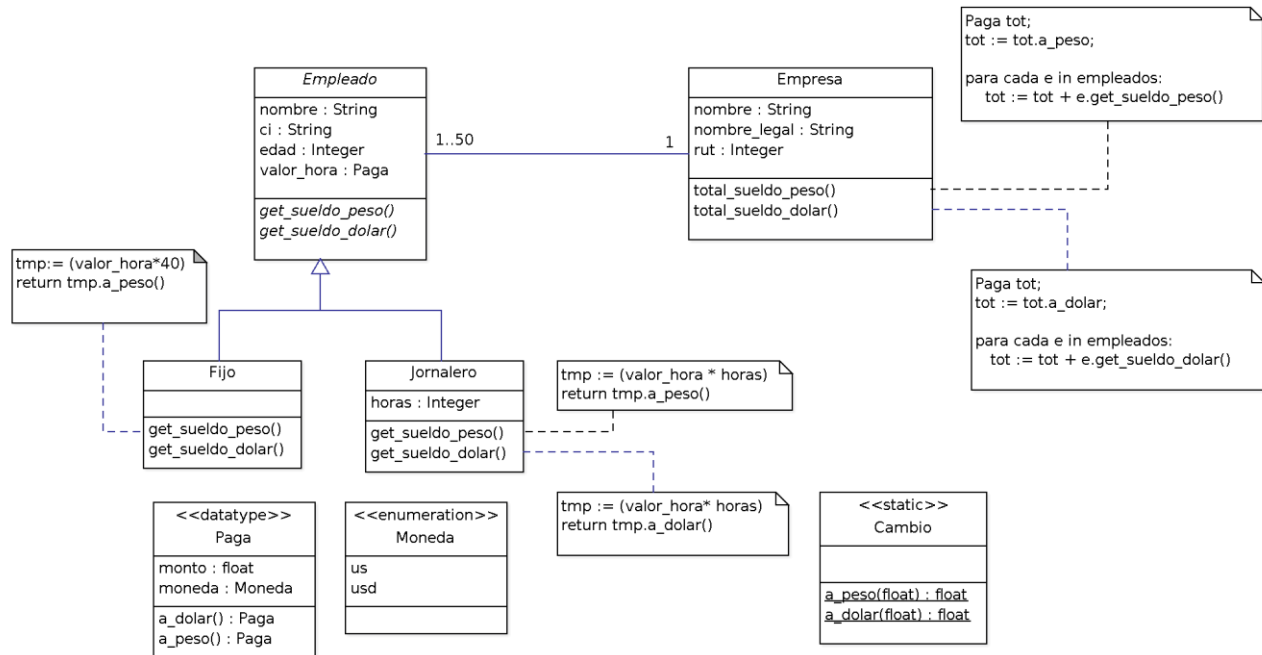
Objetivos:

Con este laboratorio se espera que los estudiantes:

- 1) adquieran competencias en la *implementación* de operaciones básicas, uso básico del lenguaje y entorno de programación y reafirmar *conceptos básicos de orientación a objetos en C++*.
- 2) puedan realizar un *Modelo de Dominio* a partir de una determinada realidad.

Parte 1 - Implementación en C++

Se desea implementar un sistema para el registro de sueldos de los empleados de una empresa. Para esto considerar el siguiente diagrama:



Se pide implementar en C++:

- El enumerado *Moneda*, representado con los valores: "us" y "usd" para peso y dólar respectivamente.
- La clase *Cambio*, con las siguientes operaciones estáticas:
 - `a_pesos(float importe)` : que dado un importe lo convierte a pesos utilizando una cotización fija¹.
 - `a_dolar(float importe)` : que dado un importe lo convierte a dólares utilizando una cotización fija.
- El datatype *Paga*, que representará la paga realizada a los empleados.
 - El constructor por defecto deberá crear una *Paga* con monto 0 y moneda *us*.
 - Deberá contar con dos operaciones que utilizarán la clase *Cambio*:
 - `a_dolar()` : Retorna una *Paga* donde si la moneda es "us" realiza la conversión utilizando *Cambio*.
 - `a_peso()` : Retorna una *Paga* donde si la moneda es "usd" realiza la conversión utilizando *Cambio*.
- Las clases *Empleado* y *Fijo* y *Jornalero* como derivadas de la clase abstracta *Empleado*. Esta última tendrá como *pseudoatributo* un puntero a la clase *Empresa*.
- La clase *Empresa*. Como se muestra en el diagrama, contiene un *pseudoatributo* que será un vector de *Empleado*. Considerarlo de tamaño fijo².
- Realizar un procedimiento `main()` que permita, mediante un menú, dar de alta una empresa, sus respectivos empleados así como calcular el *total por concepto de sueldo* tanto en dólares como en pesos.

¹ Considerar una cotización constante de 39.05 pesos por dólar.

² Definir la constante `MAX_EMPLEADO` con el valor 100.

Parte 2 - Modelado de Dominio

Considere la siguiente realidad:

La creciente cantidad de personas que hacen uso de los servicios de salud brindados por una mutualista y la importancia de la información manejada por dicha institución han sido los impulsores para que las autoridades de la entidad médica encomienden a usted y a su equipo de desarrollo de software un sistema para la gestión de historias médicas, registro de reservas de consultas y almacenamiento de información de usuarios. Existen tres tipos de usuarios del sistema: socios, médicos y administrativos. Los usuarios pueden estar activos o inactivos. Los usuarios administrativos son los únicos que pueden dar de alta (registrar) a otros usuarios, por lo tanto, el sistema desde su puesta en marcha deberá disponer de un usuario administrativo por defecto el cual podrá dar de alta a otros usuarios desde el comienzo. Cuando un usuario se encuentra registrado e ingresa por primera vez al sistema se encontrará en estado inactivo. Para activarse deberá ingresar una contraseña válida (entre 6 y 9 caracteres alfanuméricos A-Z, a-z, 0-9). Para comenzar a utilizar el sistema, los usuarios son registrados especificando su cédula (que los identifica), sexo y fecha de nacimiento. Además, de los usuarios se conoce su nombre y apellido, así como su edad, calculada a partir de la fecha de nacimiento. No puede haber dos usuarios con la misma cédula y ésta solamente puede contener caracteres numéricos. Además, se debe especificar la categoría a la que pertenece el usuario (socio, administrativo, y médico). Tener en cuenta que puede haber usuarios con dos categorías. La única combinación que no está permitida por la institución es la de ser administrativo y médico a la vez. Por razones de auditoría, la mutualista requiere conocer los usuarios dados de alta por los usuarios administrativos. Los socios pueden hacer reservas de consultas comunes, de las cuales interesa conocer la fecha de reserva, fecha y hora de consulta, y el médico tratante. Además, los socios pueden hacer consultas de emergencias, las cuales no tienen reservas. De estas consultas interesa conocer la fecha y hora de consulta, así como el motivo de esta. En el caso de las consultas de emergencia el sistema asignará el médico aleatoriamente.

Los médicos deben llevar registro de la asistencia de los socios a una consulta (de cualquier tipo). En una consulta el médico puede dar múltiples diagnósticos y para cada uno indicar, si corresponde, varios tratamientos. Un diagnóstico está definido por una representación estandarizada por la mutualista del problema de salud y una descripción libre que le permite al médico expresarse como lo desee. La representación de los problemas de salud consiste en un código más una etiqueta. Los códigos son alfanuméricos y están formados por una letra más un número de 2 cifras. Por ejemplo, un paciente que se presenta con problemas de náuseas será diagnosticado mediante "D09 Náusea" más una descripción, si lo desea el médico, como "Le sucede por las noches en los fines de semana". En este caso, "D09" es el código y "Náusea" la etiqueta del problema. En la definición del estándar la letra tiene un significado y es englobar o categorizar los problemas. Así en el ejemplo, la "D" es el código que corresponde a la categoría etiquetada como "Aparato Digestivo". Otro ejemplo de problema de esta categoría sería "D85 Úlcera duodenal". Un ejemplo correspondiente a la categoría "F": "Ojo y Anexos" sería "F71 Conjuntivitis alérgica". A la mutualista le interesa obtener reportes resumidos según la categoría. Los tratamientos tienen una descripción, y pueden ser basados en fármacos o quirúrgicos. Si el tratamiento asignado es basado en fármacos, interesa conocer los nombres de los medicamentos. Si el tratamiento es quirúrgico interesa la fecha y el médico que interviene. El sistema debe notificar a los socios cuando no han concurrido a una consulta a la cual han reservado. En caso de que un usuario no asista a más de una determinada cantidad máxima de reservas, el usuario será considerado como inactivo en el sistema. Por último, se desea mantener un historial para cada paciente, donde se conozcan todas las consultas realizadas junto con su médico tratante, los diagnósticos y los tratamientos asignados si es que éstos existieron.

Se pide:

- Construir el Modelo de Dominio a partir de la visión de la descripción planteada. Presentarlo en un diagrama utilizando UML.
- Expresar cada una de las restricciones del modelo en lenguaje natural.