

Tecnólogo Informático – San José

Programación Avanzada

Parcial Final 2020

- *Completar TODAS las hojas con el nombre y el número de cédula.*
- *Numerarlas y escribir el total en la primer hoja.*
- *Sólo se contestan dudas acerca de la letra de los ejercicios.*
- *El parcial dura 2 horas y 30 minutos.*

Ejercicio 1 (35 puntos)

Las empresas de software típicamente se dividen en dos tipos: empresas de producto y empresas de servicio. Las primeras cuentan con un producto de software previamente desarrollado que se busca vender tal cual está a la mayor cantidad de clientes posibles (ej: una "app") mientras que las segundas se dedican al desarrollo de software a medida para clientes medianos y grandes (ej: una software factory).

Es muy común que las empresas de servicios les vendan a otras empresas de servicios (ej: una tercerización) así como a empresas de producto (ej: ayudándolas a desarrollar su producto). Estas ventas se representan como contratos, los cuales establecen la fecha del contrato, la empresa proveedora (siempre de servicios) y la empresa cliente, así como fechas de inicio y fin del contrato y un código identificador del contrato y un nombre descriptivo.

Los desarrolladores (identificados por cédula y con un nombre) por su parte trabajan en proyectos que se enmarcan en dichos contratos, debiendo saberse la fecha en la que inició su trabajo en ese proyecto y la fecha en la que lo finalizó, de forma de poder luego ser asignado a otro proyecto.

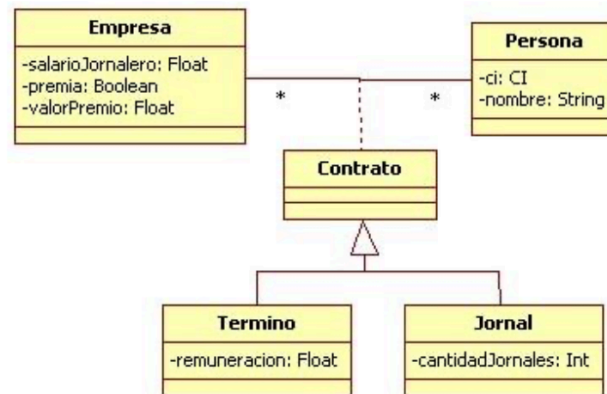
Considere el siguiente Caso de Uso:

Caso de Uso:	Alta Proyecto
Actores:	Usuario
Descripción:	<p>El CU comienza cuando el usuario solicita un listado de todos los contratos activos del Sistema (aquellos cuya fecha de finalización son posteriores a la fecha actual). Luego el usuario selecciona uno de esos contratos al cual le quiere asignar el nuevo proyecto.</p> <p>Luego el usuario solicita una lista de todos los desarrolladores y, para cada uno que desee agregar al proyecto, se consulta su disponibilidad (es decir si el desarrollador se encuentra sin asignación en las fechas del contrato). Solo se permitirá agregar desarrolladores que estén disponibles.</p> <p>El CU finaliza cuando el usuario confirma el alta del nuevo proyecto, con los desarrolladores (disponibles) elegidos, y eligiendo un nombre para dicho proyecto.</p>

- Se pide:** realice un Modelo de Dominio de esta realidad con restricciones en lenguaje natural.
- Se pide:** realice el Diagrama de Secuencia del Sistema (DSS) correspondiente incluyendo el uso de datatypes y memoria del Sistema en caso de ser necesario.

Ejercicio 2 (30 puntos)

Se cuenta con el siguiente modelo de dominio que representa los contratos de personal de distintas empresas.



Se quiere diseñar la operación que realiza la liquidación de las remuneraciones de todas las personas. Esta operación no recibe parámetros y devuelve un conjunto de valores. Cada uno de estos valores tiene tres atributos: cédula de la persona, nombre de la persona y el importe de su remuneración.

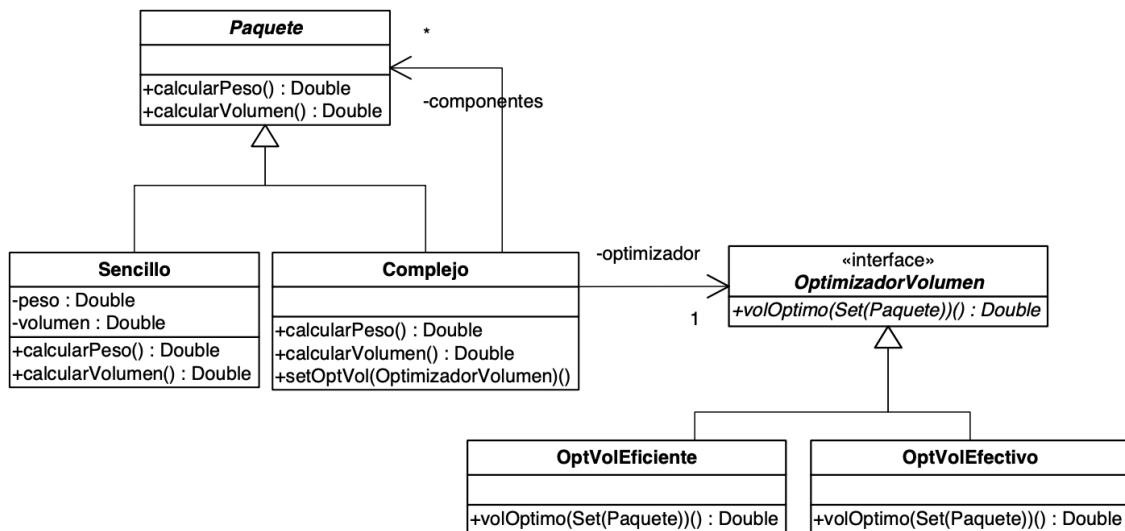
La base del cálculo de la liquidación se ha delegado en la clase **Contrato**, que es el experto de la información. Calcular el importe de remuneración de un contrato consiste en la suma entre el premio que otorga la empresa (si es que corresponde) y el cálculo en sí de la remuneración que depende del tipo de contrato. Si es a término la remuneración es la fijada en la definición del contrato. Si es jornalero la remuneración se calcula multiplicando el valor del salario de jornalero de la empresa y la cantidad de jornales acordada en la definición del contrato.

Se pide:

- Realice el Diagrama de Comunicación completo de la operación *liquidación*. Indicar claramente los parámetros y el tipo del resultado de todas las operaciones involucradas en su solución. No es necesario indicar las visibilidades.
- Realice el Diagrama de Clases de Diseño (DCD) correspondiente.

Ejercicio 3 (35 puntos)

Una empresa de servicios de transporte interurbano de objetos frágiles, ha encomendado el desarrollo de un sistema de apoyo a la gestión en el armado de los paquetes chicos y medianos. Los objetos son empaquetados en cajas, ya sea de forma individual o agrupados. Los paquetes así conformados, pueden a su vez ser parte de otros paquetes más grandes. La figura muestra el diseño de la estructura determinada para representar la organización de los paquetes.



Para organizar la carga en los vehículos, es de interés poder estimar el peso y volumen de los paquetes. El cálculo del peso es relativamente sencillo, mientras que el del volumen implica la ejecución de un algoritmo de optimización de ubicación de volúmenes, que tiene un alto costo computacional. Por este motivo, para la estimación del volumen de un paquete que está formado por otros, se utiliza un optimizador de volúmenes. Existen varias versiones de dicho optimizador: algunas estiman de mejor forma el volumen con un alto costo computacional (efectivas), mientras que otras hacen una estimación menos ajustada pero ejecutan de manera más rápida (eficientes).

A su vez, se deben tener en cuenta las siguientes consideraciones:

- El peso y el volumen de un paquete **Sencillo** se calculan directamente a partir de sus atributos.
- El peso de un paquete **Complejo** es la suma de los pesos de sus componentes.
- El volumen de un paquete **Complejo** se calcula invocando a la operación `valOptimo` de la interfaz **OptimizadorVolumen**, la cual recibe una colección con los componentes del paquete en cuestión.
- La construcción de un paquete **Complejo** se realiza a partir de sus componentes y de una instancia de una clase que implemente la interfaz **OptimizadorVolumen**.
- La destrucción de un paquete **Complejo** implica la destrucción de todos sus componentes.

Se pide:

Implementar completamente en **C++** las clases **Paquete**, **Sencillo**, **Complejo** y la interfaz **OptimizadorVolumen**.