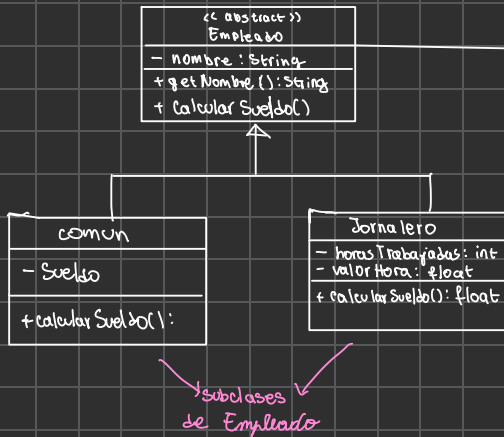


Ejercicio 10:

- a) Identificar **clases**, **atributos**, **operaciones** y **métodos** en la siguiente descripción del diseño de una parte de un sistema de gestión de personal.

Se considera una **empresa** que mantiene información de sus empleados. De cada **empleado** se almacena su **nombre** y se cuenta con una operación para **obtener el mismo**. Los empleados de la empresa pueden ser **comunes**, de los cuales se conoce su **suelo**, o **jornaleros**, de los cuales se conoce la **cantidad de horas que trabajó** y el **valor de la hora**. Todos los meses, al realizarse la liquidación de los sueldos, la empresa **calcula el monto total** por concepto de sueldos, **en base al sueldo fijo para empleados comunes y como la cantidad de horas por el valor de la hora para empleados jornaleros**.



→ Clase que gestiona los empleados

La operación `calcularSueldo()` es abstracta en **Empleado** y se sobrescribe en la subclase **comun** en base al sueldo fijo para los empleados comunes y en la subclase **jornalero** como `horasTrabajadas * valorHora`.

- b) ¿Existe alguna clase abstracta en su solución? ¿En qué situaciones debemos definir a una clase como abstracta?

En este caso la clase **Empleado** es abstracta porque tiene una operación abstracta (no se especifica método para instancias de esa clase).

Se define una clase como abstracta cuando no se quiere instanciar objetos directos de esa clase. En este caso no existen empleados que no sean comunes ni jornaleros.

- c) Dar una descripción (pseudocódigo) de la operación `getTotal()` mediante la cual la empresa calcula el monto total de la liquidación de todos sus empleados. Tener en cuenta que en un futuro pueden incorporarse nuevos tipos de empleados, y que la repercusión en la implementación existente debe ser mínima (en particular sobre las operaciones ya implementadas en clases existentes).

```
getTotal()  
total = 0  
Recorrer lista de empleados  
  Para cada empleado hacer  
    total += sueldo del empleado calculado  
    a partir del método correspondiente  
    al tipo de empleado.  
retornar total
```

- d) Definir el concepto de polimorfismo e identificar su aplicación en el modelo realizado en la parte (b). ¿Qué beneficios le aportó el polimorfismo y cuál otra propiedad le fue de utilidad?

El polimorfismo es la capacidad de asociar diferentes métodos a una misma operación para clases diferentes. Aquí queremos que el método para calcular el sueldo de un empleado dependa de si es común o jornalero, por lo que se sobrescribe el método de la misma operación `calcularSueldo()` para cada subclase de `Empleado`.

En la implementación de la operación `getTotal()` es útil usar la propiedad de `subsumption` que permite sustituir un objeto de la clase derivada por uno de la clase base y de esta manera evitar una estructura condicional según tipo de empleado.