Ejemplo de una clase entera escrita "estilo en papel"

A raíz de una consulta, armamos este pequeño documento para mostrar un ejemplo concreto del formato para describir una clase que indicamos en los slides de teoría.

Supongamos que para un planificador de proyectos para una organización tenemos que crear la clase Tarea, que tiene responsable (que es una Persona), fechas inicio y fin (que son dos Date), índice de complejidad e índice de criticidad (que son números). Nos dicen que las personas entienden el mensaje antiguedad, que nos devuelve la cantidad de años que lleva cada una en la organización.

Queremos que las tareas entiendan los siguientes mensajes

•duracion "en días"

•estaActivaElDia: unDia "true si unDia está entre las fechas inicial y final"

•indiceDeRiesgo
" un número que se calcula como duración / 20 + complejidad + criticidad"

•puedeSerConflictiva
"es cierto si la antigüedad del responsable es menor a 12 años o el índice de riesgo es mayor a 15"

También nos dicen que una tarea sin responsable no tiene sentido, entonces tenemos que hacer un constructor al que le pasamos el responsable.

El código es así

Tarea

variablesDeInstancia: responsable, fechaInicio, fechaFin, indiceComplejidad, indiceCriticidad

Protocolo de clase

```
>> newConResponsable: unaPersona
        | nuevaTarea |
        nuevaTarea := self new.
        nuevaTarea responsable: unaPersona.
        ^nuevaTarea
```

Protocolo de instancia

```
>> responsable
        ^responsable

>> responsable: unaPersona
        responsable := unaPersona
```

"... los otros setters y getters que decidamos que van ..."

```
>> duracion
	^(self fechaFin subtractDate: self fechaInicio) + 1

>> estaActivaElDia: unDia
	^unDia between: self fechaInicio and: self fechaFin

>> indiceDeRiesgo
	^(self duracion / 20) + self indiceComplejidad + self indiceCriticidad

>> puedeSerConflictiva
	^(self responsable antigüedad <  12) | (self indiceDeRiesgo > 15)
```

Ahora nos dicen que se agregan

•las tareas remotas, que son como las tareas, pero: se sabe en qué lugar se hace, y si se hace en un lugar que está a más de 3000 metros de altura puede ser conflictiva, además de los criterios de las tareas en general; y

•las tareas colaborativas que se hacen en conjunto con otras organizaciones, de cada una se sabe con cuántas otras organizaciones hay que colaborar, y al índice de riesgo hay que sumarle 3 si se colabora con más de 5 organizaciones.

Tenemos que agregar estas clases

TareaRemota

superclase: Tarea

variablesDeInstancia: lugar

Protocolo de instancia

```
>> lugar
        ^lugar

>> lugar: unLugar
        lugar := unLugar

>> puedeSerConflictiva
        ^(lugar altura > 3000) | super puedeSerConflictiva
```

TareaColaborativa

superclase: Tarea

variablesDeInstancia: cantidadDeColaboradores

Protocolo de instancia

```
>> cantidadDeColaboradores
        ^cantidadDeColaboradores

>> cantidadDeColaboradores: unNumero
        cantidadDeColaboradores := unNumero

>> indiceDeRiesgo
        | delta |
        delta := 0.
        (self cantidadDeColaboradores > 5) ifTrue: [delta := 3].
        ^super indiceDeRiesgo + delta
```