

Parcial - Primera Fecha

Orientación a Objetos II

1. Ejercicio 1 - Dependencias y Frameworks

Sean las clases que se muestran en la figura 1.

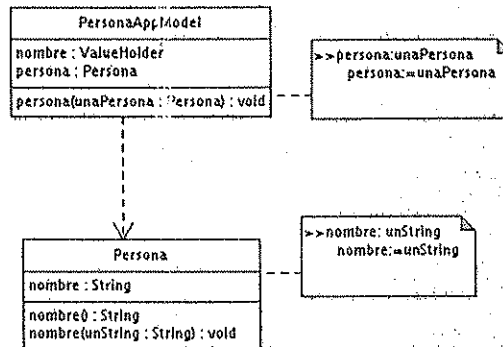


Figura 1: Diagrama de clases de Persona y PersonaAppModel

Tareas:

1. Implemente el código necesario para lograr que una instancia `PersonaAppModel` actualice su `ValueHolder` con el nombre cada vez que éste cambie. Su código debe considerar que:
 - a) La instancia `PersonaAppModel` puede recibir el mensaje: `#persona:` y cambiar la `persona` sobre la cual trabaja.
 - b) Inicialmente la instancia de `PersonaAppModel` no tiene ninguna persona asociada.
 - c) El contenido del `value holder` se debe actualizar cuando cambie el nombre de la persona asociada.
2. Explique mediante un diagrama de secuencia cómo el código que ud. ha escrito permite que múltiples ventanas puedan trabajar sobre la misma instancia de persona y ser actualizadas de forma consistente. Para el diagrama considere el caso de una instancia de `Persona` y dos instancias de `PersonaAppModel`.

2. Ejercicio 2 - Refactoring

En una aplicación web donde se pueden loggear los usuarios, ud. encuentra el siguiente código que muestra cómo se realiza la operación de `login`.

```
WebApplication>>loginUser: aUser with: password
user isNil ifTrue:[ ^false.].
(loginValidator isValid: aUser with: password)
    ifTrue:[^true].
aUser failedAttempts: aUser failedAttempts +1.
aUser failedAttempts >=3
    ifTrue: [aUser lockOut: true.].
^false.
```

Tareas:

1. Indique qué *bad smell* detecta en el código (si no recuerda el nombre explique brevemente el problema encontrado). Nombre el/los *refactorings* con el que se podrá solucionar

Refactor