

**ENGINYERIA DEL PROGRAMARI**

**ç**

**PRÀCTICA 4 - Disseny i Proves Unitàries**

Grup: Pralab1

Roger Castellví Rubinat - 48056998Q

Armand Ciutat Camps - 48056684R

Joel Aumedes Serrano - 48051307Y

Data d’entrega: 17 de Gener de 2021

**Curs 2020-21**

Índex

[1. Introducció 1](#_Toc61693624)

[2. Desenvolupament de la Pràctica 2](#_Toc61693625)

[2.1 El paquet Data 2](#_Toc61693626)

[2.2 El paquet medicalconsultation 2](#_Toc61693627)

[2.3 El cas d’ús *Supervisar tratamiento* 4](#_Toc61693628)

# 1. Introducció

En aquesta pràctica es realitzarà una implementació d’una versió simplificada del cas d’ús *Supervisar tratamiento*, pertanyent a la temàtica que s’ha anat seguint al llarg d’aquest curs a través de les pràctiques: *el Sistema integral personalizado de receta electrónica*.

Concretament, es desenvoluparà el codi i els respectius tests del mateix, estudiant cada cas d’ús possible, esmentant els criteris establerts per a prendre les decisions del disseny.

Per a facilitar la cooperació entre l’equip de treball, s’ha utilitzat un sistema de control de versions Git alhora amb un repositori remot (Github). Per a assolir una millor col·laboració i sincronització entre els membres del grup, s’ha decidit crear una branca per a cada membre per tal de realitzar una càrrega de treball equitativa entre els integrants del grup. El criteri de divisió de branques s’ha fet en base a la realització dels tests de cada classe.

Ha sigut de gran ajut tenir a mà la solució del Diagrama de Classes de la pràctica anterior per a poder definir millor els atributs i referències a les respectives classes.

# 2. Desenvolupament de la Pràctica

## 2.1 El paquet Data

El paquet Data es simplement crear les classes. La realització de tests era trivial. S'ha intentat realitzar una comprovació dels possibles escenaris de cada cas. Per tant, els tests de HealthCardID i ProductID estudien els casos en que sigui nul, amb un format incorrecte i amb un format correcte, creant una instància de cada un per tal de comprovar-ho.En el cas de DigitalSignature només s'estudia el cas que sigui nul, doncs es tracta d'una Array de Bytes i per tant no té un format específic.

## 2.2 El paquet medicalconsultation

Per el que fa als tests de "medicalConsultation" s'ha realitzat el mateix procediment que s'ha mencionat abans amb la majoria de classes, doncs Posology, ProductSpecification i TakingGuideline han de comprovar qué succeeix quan es nul i revisar els casos de format incorrecte/correcte. La classe MedicalPrescription ja realitza un test més ampli i complexe, doncs conté diversos mètodes que s'han d'estudiar. Els casos estudiats són els següents:

**ADD, MODIFY I REMOVE LINE**

* addLine\_NullArgumentTest() & modifyLine\_NullProdIDTest(): cas en que ens passen un id de producte nul.
* addLine\_invalidStringTest(): cas en que ens passen una línia de prescripció invàlida (és a dir, amb elements nuls).
* addLine\_incompleteStringTest(): cas en que ens passen una línia de prescripció incompleta (menys de 6 elements).
* addLine\_exceededStringTest(): cas en que ens passen una línia de prescripció amb massa elements (més de 6 elements).
* addLine\_incorrectFormatTest(): cas en que ens passen una línia de prescripció amb format incorrecte.
* checkAddedLine(): cas en que ens passen una línia vàlida (és a dir, comprovem que el tamany de la llista incrementa un cop fem addLine).
* modifyLine\_ProdID\_NotFoundTest(): cas en que el id introduit no estigui en la llista (i conseqüientment no es pugui modificar la linia).
* checkRemovedLine(): cas en que ens passen una línia vàlida per a eliminar (és a dir, comprovem que el tamany de la llista decrementa un cop fem removeLine)
* removeLine\_ProdID\_NotFoundTest(): cas en que el id introduit no estigui en la llista (i conseqüentment no es pugui eliminar la linia).

==========================================

· A la classe MedicalPrescription s'ha decidit NO contemplar el cas d'afegir un medicament ja existent, es a dir, comprovar que un medicament es trobi repetit, ja que es considera que no té sentit que s'introdueixi el mateix producte més d'un cop, i encara menys un producte amb mateix ProductID però diferent descripció i/o preu.

· Al test checkAddedLine hem tingut problemes al fer assertEquals. No feia la comprovació entre valors tipus float correctament. Per aquest motiu, hem decidit canviar l'asserció i simplement comprovar que l'array de instruccions ha augmentat en 1, d'aquesta manera podem saber si ha afegit la linia correctament.

· De la mateixa manera que s'ha comprovat al test de AddLine, en el test de RemoveLine comprovarem que un cop realitzada l'acció, l'array de instruccions ha decrementat a 0.

· (PRINCIPI SOLID: S) En la classe MedicalPrescription s'ha optat per fer ús de metodes varis per tal de no repetir procediments al llarg del programa, evitant d'aquesta manera els code smells de tipus bloaters (classes o mètodes massa llargs). S'ha intentat mantenir un tamany reduit en cada mètode, buscant simplesa però precisi a més de definir una classe per cada responsabilitat.

· (PRINCIPI SOLID: O) Els canvis han generat codi nou en comptes de la modificació de l'existent, com per exemple... FICAR ALGUN EXEMPLE ESPECÍFIC DE CANVIS

· (PRINCIPI SOLID: I) S'ha fet ús de la interficie HealthNationalService per tal de cambiar les dependencies de clases a dependencies de interficies.

"""A REVISAR"""

· La classe MedicalPrescription és un exemple de classe assignada amb el patró creador, doncs s'encarrega de crear diverses instàncies d'altres classes (com HealthCardID o DigitalSignature)

· DayMoment i FqUnit són exemples de classes assignades amb el patró d'acoblament baix, doncs la dependència amb altres classes és mínima (o en aquest cas inexistent).

## 2.3 El cas d’ús *Supervisar tratamiento*