

# **Validació d'aplicacions i programes**

---

Pau Pedrós Ribas

2023-2024

DAM 1A

M5 UF2

30/04/2024

---

<b>Introducció</b>	<b>3</b>
<b>Els meus errors</b>	<b>4</b>
<b>El cicle TDD</b>	<b>5</b>
<b>Qui fa què</b>	<b>6</b>

---

## Introducció

---

En aquest exercici farem un breu resum dels continguts explicats a la classe de teoria del dia 30 d'abril, on es va donar el temari de la part de validació d'aplicacions. Aquest tema engloba diferents parts: errors, proves, classes de proves i proves unitàries. En aquest exercici farem 3 preguntes, comentarem una mica els errors, el cicle TDD i els rols dintre les classes de proves.

## Els meus errors

---

Enunciat:

Aquesta pregunta fa referència als continguts [Els errors](#).

Si has arribat a aquestes alçades del curs, probablement ja fa un cert temps que desenvolupes i segur que en tot aquest temps has comés errors. Probablement molts. No cal que t'avergonyeïxis, ens passa a tothom!

En aquesta pregunta, et demano que pensis en un error de programació concret que t'hagi passat. Intenta trobar un que et costés prou de trobar o/i de corregir. Descriu en detall:

- l'error: en què consistia
- la detecció: com te'n vas adonar
- el procés de trobar-lo. ex. com vas realitzar les proves
- la correcció: en què va consistir la correcció
- ah! intenta incloure com et vas sentir a cadascuna de les quatre fases anteriors! (encara que tothom pot imaginar-se la darrera si va costar-te molt, oi?)

L'error:

L'error consistia en que jo tenia una hora formatada en hores, minuts i segons, li havia de sumar uns certs segons a aquesta hora. Jo el que intentava era passar els segons a incrementar a una hora a la qual podia sumar les dues. Vaig provar moltes formules per incrementar els segons a la hora, em posava molt nervios perquè no em sortia.

La detecció:

El vaig detectar al fer les proves en un entorn controlat, la suma no retornava el valor esperat a sumar dues hores.

El procés de trobar-lo:

Vaig definir un entorn controlat on comprovava el resultat de la suma entre hores. Vaig estar fent proves amb diferents hores i segons a incrementar, la suma no donava la hora resultant correcte

Correcció:

La correcció que se'm va ocórrer va ser passar la hora original a segons i sumar els segons i

passar el resultat de la suma a una hora ben formatada amb hores, minuts i segons.

## El cicle TDD

---

Enunciat:

Aquesta pregunta fa referència als continguts [Les proves](#).

Describeix amb les teves paraules les cinc passes que conformen el cicle de *Test Driven Development* segons aquest article de la [Wikipedia sobre el TDD](#).

Compara aquest cicle amb la teva manera de programar actual. Quina et sembla més adequada segons criteris com velocitat de desenvolupament, qualitat de codi generat, mantenibilitat, etc.

Descripció de les cinc passes:

### 1. List the test scenarios for the new feature

Aquest pas és bàsicament llistar tots els escenaris que poden ocórrer amb la nova funcionalitat.

### 2. Add one test from the list

En aquest pas consta d'afegir al joc de proves un test de la llista.

### 3. Run all tests. The new test should *fail* – for *expected* reasons

Executar el joc de proves, llavors el test nou afegit hauria de fallar, ja que no hem implementat encara la funcionalitat

### 4. Write the simplest code that passes the new test

Escriure el mínim codi possible per a que el nou test passi.

### 5. All tests should now pass

Els test haurien de passar correctament!

### 6. Refactor as needed while ensuring all tests continue to pass

Ara en aquest pas el que s'ha de fer es recodificar lo necessari per a que la funcionalitat

funcioni com s'espera, tot comptant que els tests continuen donant OK.

## Qui fa què

---

Enunciat:

Aquesta pregunta fa referència als continguts [Classes de proves](#).

Per cadascun dels nivells de prova, indica quin és el perfil dels *testejadors* més adequats per realitzar-les. Justifica-ho.

Proposa qui hauria de dissenyar aquestes proves i quan ho hauria de fer (a quina fase del desenvolupament).

Nivells de prova:

De sistema:

El perfil més adequat per aquestes proves són els usuaris, ja que són finalment qui faran servir el sistema i poden provar totes les funcionalitats innocentment, no com un programador que ja sap com va i estar influenciat per que el codi és "seu". També les poden fer testejadors adequats i amb formació del camp del projecte. Aquestes proves les han de fer tant els usuaris com els testers. Es farien a la fase d'implementació ja que es prova tot el sistema en conjunt.

D'integració:

Aquestes proves les ha de realitzar els testejadors ja que requereix tenir el coneixement de com funcionen varies parts del sistema quan s'ajunten. Es fan a la fase d'integració quan es desenvolupa una nova funcionalitat per provar-la en conjunt amb altres parts.

Funcionals:

Són les que proven els casos d'ús específics. Aquestes proves es fan a la fase de proves, i aquestes són fetes per els testers i els desenvolupadors, ja que es necessiten nocions de com funciona el codi per a poder testear-lo.

Unitaries:

Aquestes proves testejen parts específiques del codi, aquestes proves es fan a la fase de

desenvolupament, llavors es fan pels desenvolupadors.