

# VALIDACIÓ

# Taula de continguts

[Taula de continguts](#)

[Introducció](#)

[Pregunta 1. Els meus errors](#)

[Pregunta 2. El cicle de TDD](#)

[Pregunta 3. Qui fa què](#)

[Conclusions](#)

# Introducció

L'objectiu d'aquest exercici és entendre la necessitat i importància dels test a l'hora de programar.

La primera pregunta t'introdueix el tema del testeig amb una explicació pràctica, el teu propi mètode de testeig.

La segona pregunta ens fa reflexionar sobre el TDD ja que ens fa haver de llegir, entendre i explicar com funciona el testeig en grans projectes.

La tercera i última pregunta ens fa valorar diversos escenaris quotidians de la programació a gran escala i ens fa plantejar-nos quin perfil de tester és més adequat per cada escenari.

## Pregunta 1. Els meus errors

Un dels errors de programació que més mal m'ha fet ha sigut deixar-me de posar un `input.close` a un exercici dels vins.

Vaig veure que no tancava l'arxiu però pensava que l'error no podia ser aquell, que un simple `input.close` no em podia fer fallar la part clau del meu programa. Després de 2 dies encallat vaig demanar-li la ajuda al meu tutor, que em va dir de seguida !Oita!, que no estas tancant el writer. Un cop el vaig tancar va funcionar el codi, em va petar a més llocs però vaig poder avançar molt. Em vaig sentir bastant idiota per un moment, vaig perdre 2 dies enters per culpa d'un estúpid `close`.

A banda d'aquest error estúpid he tingut molts errors al meu codi i seguiré tenint, normalment quan trovo un error faig un `println` especificant els passos que segueix el meu codi per veure en quin moment falla. Sempre que aconseguieixo arreglar el problema em sento bé, realitzat amb i mateix i és una de les coses que més m'agraden de la programació, poder trobar la solució a un problema.

## Pregunta 2. El cicle de TDD

Primera pasa:

Pensar en el funcionament del codi i el que ha de fer per poder generar un test al respecte.

Segona pasa:

Comprovar tots els tests que he fet fins al moment, tots els que se'm vinguin al cap.

Tercera pasa:

Fer un codi que em passi el test, no importa si es bo o dolent si fa la funció que vull que faci.

Quarta pasa:

Arreglar el codi fins que passin tots els tests, fins que no passi tots els tests no el donarem per enllestit.

Cinquena pasa:

Refactoritzar el codi, millorar la qualitat del codi per poder ser millorat i llegit més fàcilment, és el que s'ha de fer en projectes grans per tal de que el nostre codi sigui sòlid.

Sisena pasa:

Repetir.

Crec que encara no puc comparar aquesta forma de codificar amb la meva ja que la dimensió dels treballs no és comparable i en el meu cas els tests estan fets prèviament.

Dit això crec que aquesta és la manera més òptima de programar ja que es basa en bones pràctiques i la qualitat del codi.

## Pregunta 3. Qui fa què

### Proves de sistema

Considero que els que ho testejarien millor serien els usuaris ja que es un test molt superficial, sense adentrar-se en codi, simplement comproven el que es mostra per pantalla o per la sortida del programa.

### Proves d'integració

Crec que els millors testers serien els propis testers, no son proves de codificació diària ja que involucren moltes parts del projecte com per que els programadors puguin estar pendents.

### Proves funcionals

Crec que el millor perfil seria el del tester ja que están millor preparats per poder comprovar i pensar les possibles funcionalitats d'una nau industrial.

### proves unitàries

El millor perfil es el del desenvolupador ja que tenen millor coneixement del codi i son tests més senzills de programar i de resoldre.

# Conclusions

En conclusió el més important del nostre treball és seguir una estructura de codificació per tal de poder fer codi de bona qualitat. És molt important testejar el codi per trobar errors el més aviat possible, quants més errors trobem millor ja que això ens estalviarà problemes al futur.

No hem de tenir vergonya de fer errors ja que els errors són una de les coses que ens defineix com humans, l'important és aprendre dels errors i millorar cada dia.