

# Validació de programes

## CODE TESTING

Lorem ipsum dolor sit amet,  
melius commoda splendide  
vix eu, justo graeco sit  
no, mei cu albucius  
euripidis.



22/5/2025

Jeiron Espinal

DAW 1B

# Index

Introducció .....	3
El meus errors.....	4
El cicle de TDD .....	5
Qui fa què .....	6
Conclusió .....	7

# Introducció

En el desenvolupament de programes és molt comú trobar-nos amb errors que al moment no sabem com solucionar. De fet, no només passa en el món de la programació, al llarg de la nostra vida també ens enfrontem a una suma d'errors que semblessin obstacles, però en realitat ens ajuden a créixer i millorar com a persones. D'una manera similar els errors que ocorren en els programes que creem poden semblar un problema, però en realitat són una oportunitat per millorar el codi, mantenir-lo ben estructurat i funcional.

En aquest document s'explica què passa amb els nostres errors a la secció **“Els meus errors”**. A la secció **“El cicle de TDD”** que és una metodologia que ens permetre crear proves abans de crear el codi del nostre programa. Finalment, a la secció **“Qui fa què”** es descriu el perfil de les persones o desenvolupadors encarregats de realitzar les diferents proves, així com la seva responsabilitat dins del procés de desenvolupament.

# El meus errors

Un dels errors que més em va costar detectar va aparèixer en un exercici on havia de desenvolupar un programa capaç de dibuixar taules a partir d'una especificació rebuda per línia de comandes. Em sentia molt frustrat amb aquests errors.

**L'error:** consistia en què la meva implementació d'algunes funcions només donava resposta correcta a algunes especificacions, però fallava amb altres. Les funcions estaven estructurades amb un for dins d'un altre for, per construir la taula fila a fila i columna a columna.

**La detecció:** durant les vacances de Setmana Santa, vaig dedicar temps a revisar aquest exercici. Vaig afegir prints (chivats) en diferents punts del codi per identificar exactament on es produïen els errors i entendre millor què estava passant.

**El procés de trobar-lo:** El meu professor de programació em va ensenyar a fer proves directes a una funció, i vaig utilitzar aquesta tècnica per localitzar l'error. Així, em vaig adonar que en alguns casos les taules dibuixades presentaven errors en algunes columnes o files.

**La correcció:** La meva solució a aquest problema em va costar una mica, però al mateix temps em va ajudar a entendre millor el funcionament de les taules. Concretament, la correcció va consistir a fer el contrari del que feia en les implementacions.

# El cicle de TDD

Les cinc passes que conformen el cicle de Test Driven Development consisteixen, primer de tot, a pensar en les proves per a cada escenari. Després, s'escriuen proves que fallen per adonar-nos dels errors que es poden produir. Per tant, es reescriu el codi d'una manera diferent i més simple per fer que les proves passin. Un cop les proves passen, cal refactoritzar el codi per aconseguir que sigui més llegible i escalable. Finalment, es repeteix aquest procés fins que tots els possibles casos de prova estiguin resolts.

En el meu cas, ara mateix estic molt centrat en la refactorització del meu codi perquè sigui llegible. Tot i això, m'he adonat que fer proves abans d'escriure el codi fa que la programació sigui més interessant i que el desenvolupament del programa sigui més ràpid i fàcil de mantenir.

# Qui fa què

**Proves de sistema:** Aquestes proves poden ser realitzades tant pels testers com pels usuaris. Els testers no tenen cap problema en castigar els sistemes per detectar errors que podrien passar desapercebuts, per això es que són ideals per aquestes proves i els usuaris poden oferir un feedback sobre la usabilitat i el comportament general del sistema.

**Proves d'integració:** aquestes proves les poden fer tant els desenvolupadors com els testers, ja que consisteixen en validar que diferents components funcionin. Aquests dos perfils són els que poden tenir una visió ampla d'aquestes proves.

**Proves funcionals:** Aquestes proves les haurien de fer els desenvolupadors almenys una vegada després d'haver acabat el codi del programa, ja que, tot i això, la major responsabilitat de realitzar-les recau en els testers.

**Proves unitàries:** aquestes proves han de ser creades i executades pels desenvolupadors, preferiblement abans o durant la codificació, seguint la metodologia de desenvolupament basada en les proves (TDD).

# Conclusió

La validació de programes mitjançant proves contribueix a garantir que la usabilitat per als usuaris sigui realment eficient. Per això, hauria de ser un hàbit dels desenvolupadors pensar primer en els possibles errors que poden aparèixer, abans de començar a escriure el codi. Aquesta manera de treballar no només ajuda a evitar problemes, sinó que també millora la qualitat dels programes.

En el meu cas, les proves unitàries són un àmbit que m'ha resultat especialment interessant. De fet, actualment hi estic dedicant molt temps per aprofundir-hi i millorar els meus coneixements sobre aquest concepte, ja que considero que són una eina molt poderosa per al desenvolupament eficient i fiable de programari.