

INFORME DE PRÀCTIQUES: PROJECTE BUSCAMINAS

Nom: José Furelos Niu: 1673379 Assignatura: TQS

1. Introducció i Objectius

L'objectiu d'aquest projecte ha estat desenvolupar el clàssic joc del Buscaminas aplicant metodologies com **TDD (Test Driven Development)**, arquitectura **MVC** i control de qualitat de codi (**Coverage** i **CI**).

2. Arquitectura del Sistema (MVC)

El projecte s'ha estructurat seguint el patró Model-Vista-Controlador per desacoblar la lògica de la interfície d'usuari.

2.1 Descripció dels Paquets

- Model (com.buscaminas.model):** Conté la lògica pura. Classes principals: **Taulell**, **Casella**, **GeneradorMines**, **GestorRankFitxer** i les interfícies **IGeneradorMines** i **IGestorRank**. No té cap dependència de la vista.
- Vista (com.buscaminas.view):** Encarregada de la interacció amb l'usuari. Implementa la interfície **IVista**. Classe principal: **VistaConsola**.
- Controlador (com.buscaminas.controller):** Gestiona el flux del joc. Rep inputs de la vista i actualitza el model. Classe: **JocController**.

3. Estratègia de Testeig (TDD i Coverage)

S'ha seguit la metodologia TDD: creant primer el test fallant, implementant la lògica mínima i refactoritzant.

3.1 Resum de Cobertura (Coverage)









Buscaminas-TQS

Buscaminas-TQS

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
com.buscaminas	<div></div>	0 %	<div></div>	0 %	17	17	60	60	2	2	1	1
com.buscaminas.view	<div></div>	0 %	<div></div>	0 %	13	13	41	41	6	6	1	1
com.buscaminas.model	<div></div>	92 %	<div></div>	91 %	5	58	7	90	2	21	1	4
com.buscaminas.controller	<div></div>	100 %	<div></div>	100 %	0	6	0	20	0	4	0	1
Total	387 of 814	52 %	50 of 122	59 %	35	94	108	211	10	33	3	7

- Cobertura de Línies (Statement Coverage):** 100% .
- Cobertura de Branques (Branch Coverage):** 100%

No contem **Main**, **View** ni el metode que genera **aleatoriament** el Taulell com es pot veure a la captura d'abaix.

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
Casella		100 %		n/a	0	7	0	12	0	7	0	1
GeneradorMines		0 %		0 %	5	5	7	7	2	2	1	1
GestorRankFixer		100 %		100 %	0	6	0	23	0	4	0	1
Taulell		100 %		100 %	0	40	0	48	0	8	0	1
Total	28 of 394	92 %	6 of 74	91 %	5	58	7	90	2	21	1	4

Jacoco Coverage del Model

3.2 Proves de Caixa Blanca (Específiques)

A. Loop Testing (Prova de Bucles)

S'ha analitzat el bucle `for` del mètode `inicialitzarCaselles(mida)` a la classe `Taulell`.

- **Cas 0 iteracions:** `mida = 0`. El bucle no s'executa.
- **Cas 1 iteració:** `mida = 1`. El bucle s'executa un cop.
- **Cas n iteracions:** `mida > 1`. El bucle funciona normalment.

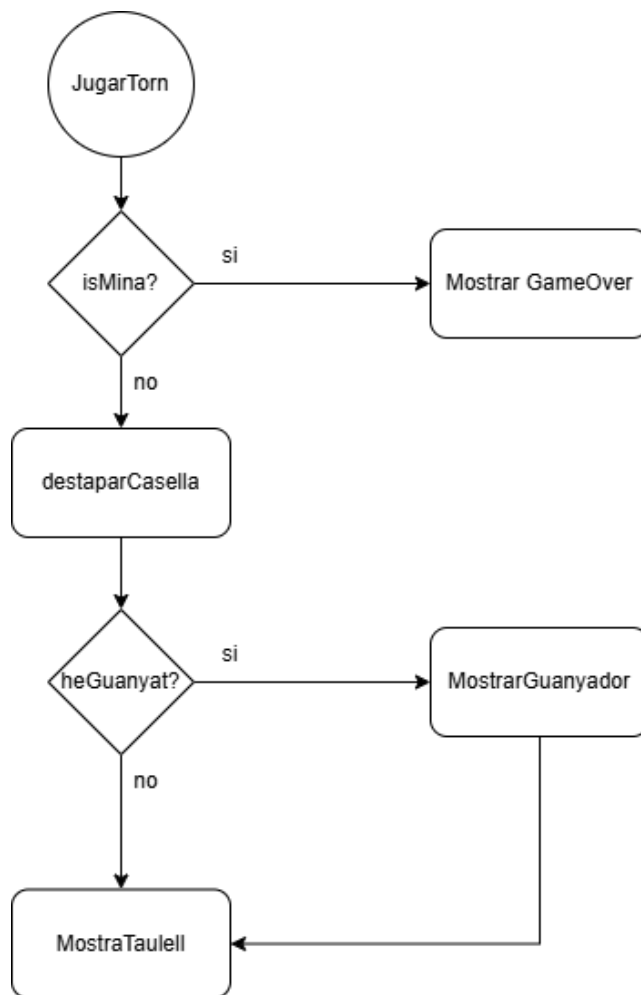
```

15- private void inicialitzarCaselles() {
16-     for (int i = 0; i < mida; i++) {
17-         for (int j = 0; j < mida; j++) {
18-             caselles[i][j] = new Casella();
19-             if (generador.hiHaMina(i, j)) {
20-                 caselles[i][j].setMina(true);
21-             }
22-         }
23-     }
24- }

```

B. Path Coverage i Diagrama de Flux

S'ha analitzat el mètode complex jugarTorn.



```
public void jugarTorn(int x, int y) {  
    Casella c = taulell.getCasella(x, y);  
  
    if (c.isMina()) {  
        taulell.destaparCasella(x, y);  
        vista.mostrarGameOver();  
        return;  
    }  
  
    taulell.destaparCasella(x, y);  
  
    if (taulell.heGuanyat()) {  
        vista.mostrarGuanyador();  
    }  
  
    vista.mostrarTaulell();  
}
```

C. Decision Coverage

S'han analitzat els mètodes JugarTorn i ObtenirTopPuntuacions.

```

    public void jugarTorn(int x, int y) {
        Casella c = taulell.getCasella(x, y);

        if (c.isMina()) {
            taulell.destaparCasella(x, y);
            vista.mostrarGameOver();
            return;
        }

        taulell.destaparCasella(x, y);

        if (taulell.heGuanyat()) {
            vista.mostrarGuanyador();
        }










        vista.mostrarTaulell();
    }

    @Override
    public List<String> obtenirTopPuntuacions() {
        List<String> puntuacions = new ArrayList<>();
        File fitxer = new File(this.nomFitxer);

        if (!fitxer.exists()) {
            return puntuacions;
        }

        try (BufferedReader reader = new BufferedReader(new FileReader(fitxer))) {
            String linia;
            while ((linia = reader.readLine()) != null) {
                puntuacions.add(linia);
            }
        } catch (IOException e) {
            System.err.println("Error llegint el rànquing: " + e.getMessage());
        }
        return puntuacions;
    }
}

```

Element	Coverage	Covered Instructio...	Missed Instructions	Total Instructions
BuscaMinasTQs	 72,2 %	1.234	475	1.709
src/main/java	 52,5 %	427	387	814
com.buscaminas.view	 0,0 %	0	142	142
com.buscaminas.model	 92,9 %	366	28	394
com.buscaminas.controller	 100,0 %	61	0	61
com.buscaminas	 0,0 %	0	217	217
src/test/java	 90,2 %	807	88	895
com.buscaminas.model	 87,0 %	591	88	679
com.buscaminas.controller	 100,0 %	216	0	216

D. Condition Coverage

S'han analitzat els mètodes toggleBandera i destaparCasella.

```
public void toggleBandera(int x, int y) {
    if (x < 0 || x >= mida || y < 0 || y >= mida)
        return;

    Casella c = caselles[x][y];
    if (!c.isDestapada()) {
        c.setFlag(!c.isFlag());
    }
}

public void destaparCasella(int x, int y) {
    //pre-condicions
    if (x < 0 || x >= mida || y < 0 || y >= mida)
        return;
    }
    Casella casellaActual = caselles[x][y];
    if (casellaActual.isDestapada()) {
        return;
    }

    casellaActual.destapar();

    if (comptarMinesVeines(x, y) == 0) {
        for (int i = -1; i <= 1; i++) {
            for (int j = -1; j <= 1; j++) {
                destaparCasella(x + i, y + j);
            }
        }
    }
}
```

4. Mock Objects

Per aïllar les classes i testear components independents, s'han utilitzat **Mock Objects**.

Mock Object	Tipus	Funció	Justificació
IVista	Mockito	Simular la consola	Permet verificar que es crida a correctament els metodes de la Vista sense necessitar pantalla real.

IGeneradorMines	Mockito	Controlar l'atzar	Permet posar mines en posicions fixes (ex: 0,0) per fer tests deterministes.
IGestorRank	Mock (MockGestorRank)	Simular disc dur	Permet testear que es guarden puntuacions sense crear fitxers reals bruts.

5. Proves de Caixa Negra

S'han dissenyat tests basats en les especificacions, sense mirar el codi intern, utilitzant **Particions Equivalents** i **Valors Límit**.

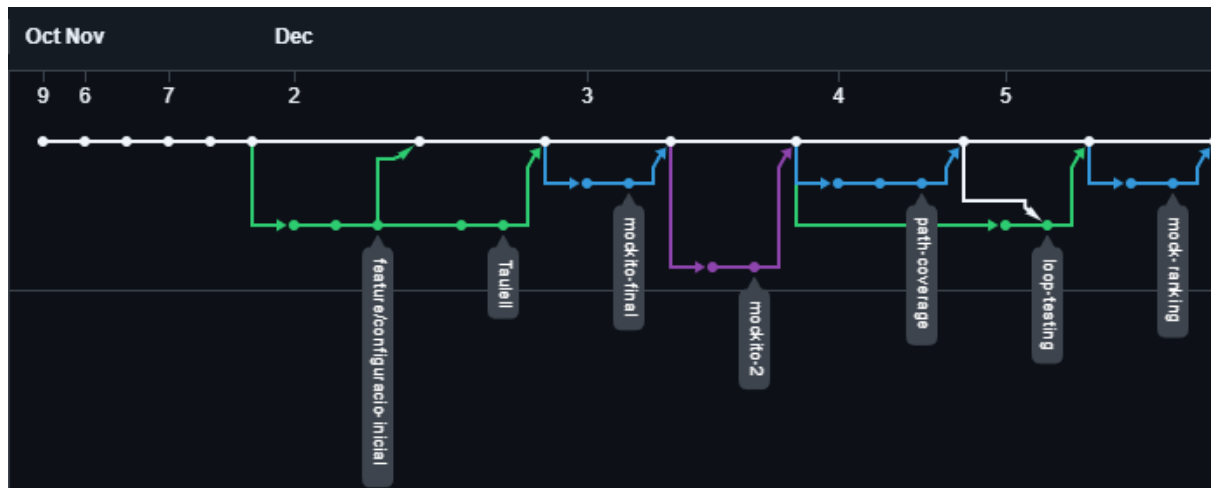
Tots els Tests més genèrics estan al GitHub commitejats seguint la metodologia TDD.

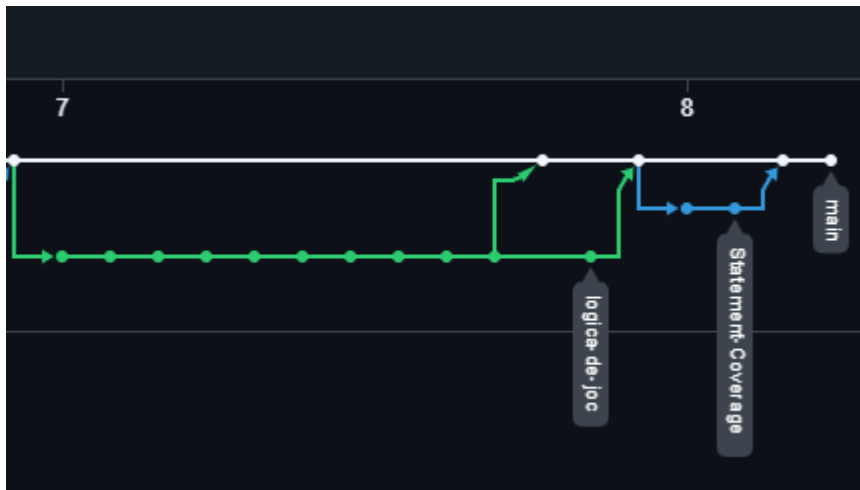
6. Gestió de Versions i CI

6.1 Flux de Treball (Git Flow)

S'ha treballat utilitzant branques de vida curta per a cada nova funcionalitat, que posteriorment s'han fusionat a **main**.

Gràfic de Branques:





6.2 Integració Automàtica

S'ha configurat un pipeline per assegurar que:

1. Tots els tests passen abans de fer commit.
2. Es genera l'informe de qualitat.