

**UNOESC SÃO MIGUEL DO OESTE  
CIÊNCIA DA COMPUTAÇÃO  
ENGENHARIA DE SOFTWARE II**

Juliel Blatt

**Relatório de desenvolvimento de software: Decora Game**

## INTRODUÇÃO

O aplicativo 'Decora game', foi criado para que de uma maneira simples os usuários possam se divertir e ao mesmo tempo exercitar o cérebro e melhorar a suas capacidades cognitivas, cada vez sendo mais desafiador.

## STORYTELLING

Lucas é um grande entusiasta de jogos de memória e desafios mentais. Além de se divertir, ele também busca maneiras de cuidar da sua saúde cognitiva. Foi assim que ele descobriu o 'Decora Game', um aplicativo que oferece diversão e benefícios para o cérebro.

Ao abrir o aplicativo, Lucas ficou encantado com a interface vibrante e envolvente. Ele sabia que jogos de memória eram excelentes para estimular o cérebro e melhorar suas habilidades cognitivas. O 'Decora Game' prometia exatamente isso.

O jogo era simples, mas desafiador. O objetivo era memorizar e repetir uma sequência de cores. A cada rodada, a sequência ficava mais longa e complexa, exigindo foco, atenção e uma boa memória por parte do jogador.

Lucas logo percebeu que o 'Decora Game' não era apenas uma diversão, mas também uma forma de cuidar da saúde do seu cérebro. Ele sabia que, ao jogar regularmente, estaria exercitando sua memória e melhorando suas habilidades cognitivas.

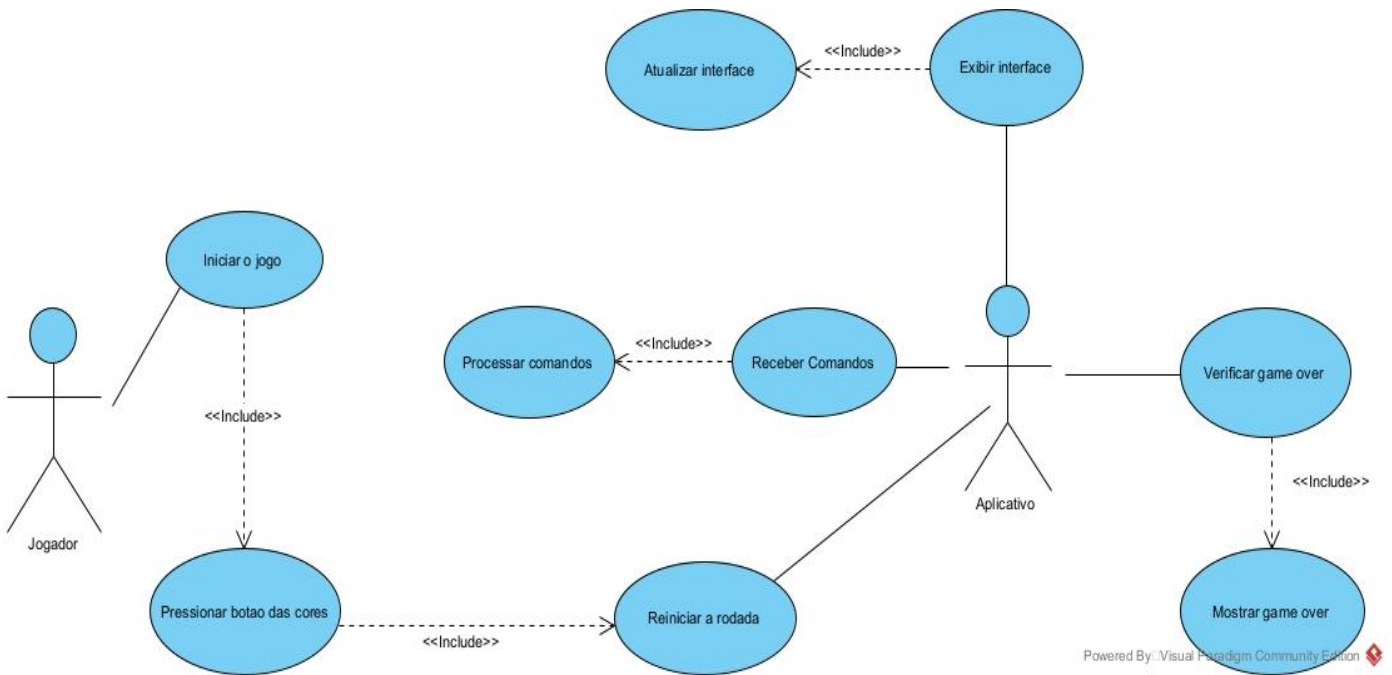
O 'Decora Game' se tornou uma ferramenta de treinamento mental para Lucas. Ele não apenas se divertia jogando, mas também sentia que estava investindo em sua saúde cognitiva a cada partida. O aplicativo se tornou um aliado na busca por uma mente mais afiada e saudável.

Lucas continuou a aproveitar o 'Decora Game', ele sabia que estava fazendo algo bom para si mesmo, enquanto se divertia e estimulava seu cérebro de uma forma prazerosa.

## DIAGRAMAS

A seguir será apresentado os diagramas, que são essenciais para entender o funcionamento do aplicativo:

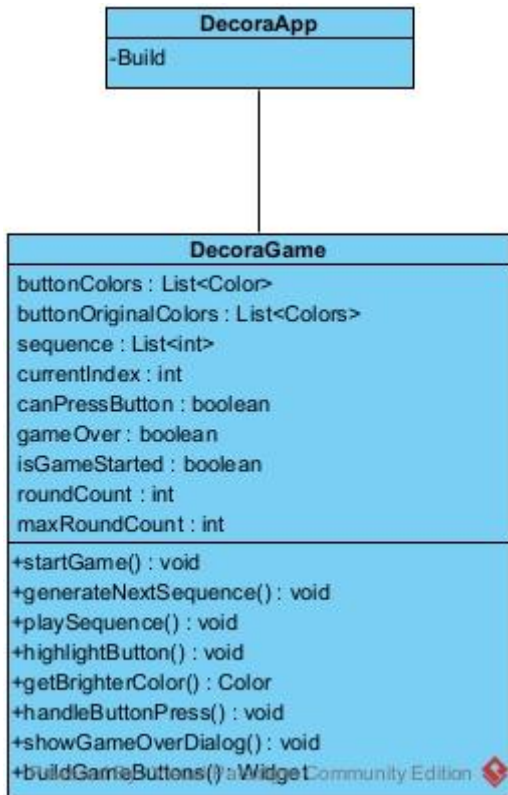
### 1 DIAGRAMA DE CASO DE USO:



Fonte:Autor

Neste diagrama podemos ver as interações do jogador com o game e também todas as ações do aplicativo para verificar, exibir e executar comandos, interfaces e diálogos.

## 2 DIAGRAMA DE CLASSE:



Fonte: Autor

A classe **DecoraApp** é uma classe **StatefulWidget** que representa o aplicativo Decora. Ela possui um método `build` para construir a interface do aplicativo.

A classe **DecoraGame** é uma classe **StatefulWidget** que representa o jogo Decora. Ela herda da classe **State** e possui estados internos que controlam o jogo.

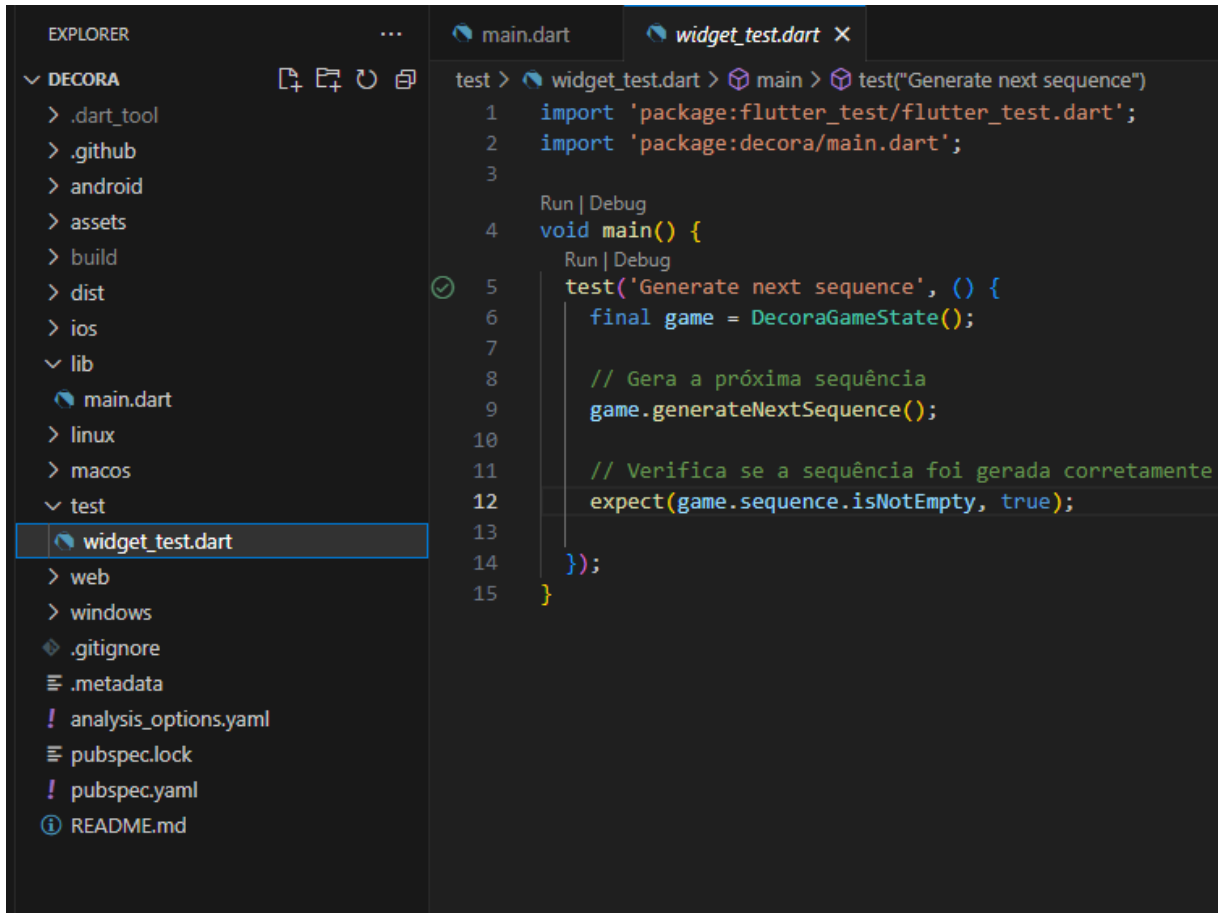
Os atributos `buttonColors`, `buttonOriginalColors`, `sequence`, `currentIndex`, `canPressButton`, `gameOver`, `isGameStarted`, `roundCount` e `maxRoundCount` são os estados do jogo.

Os métodos `startGame`, `generateNextSequence`, `playSequence`, `highlightButton`, `getBrighterColor`, `handleButtonPress`, `showGameOverDialog` e `buildGameButtons` são as ações e funcionalidades do jogo.

O método `buildGameButtons` retorna um **Widget** que representa os botões do jogo.

## TESTES

No aplicativo foi implementado o teste unitário, onde é testado a geração de novas rodadas.

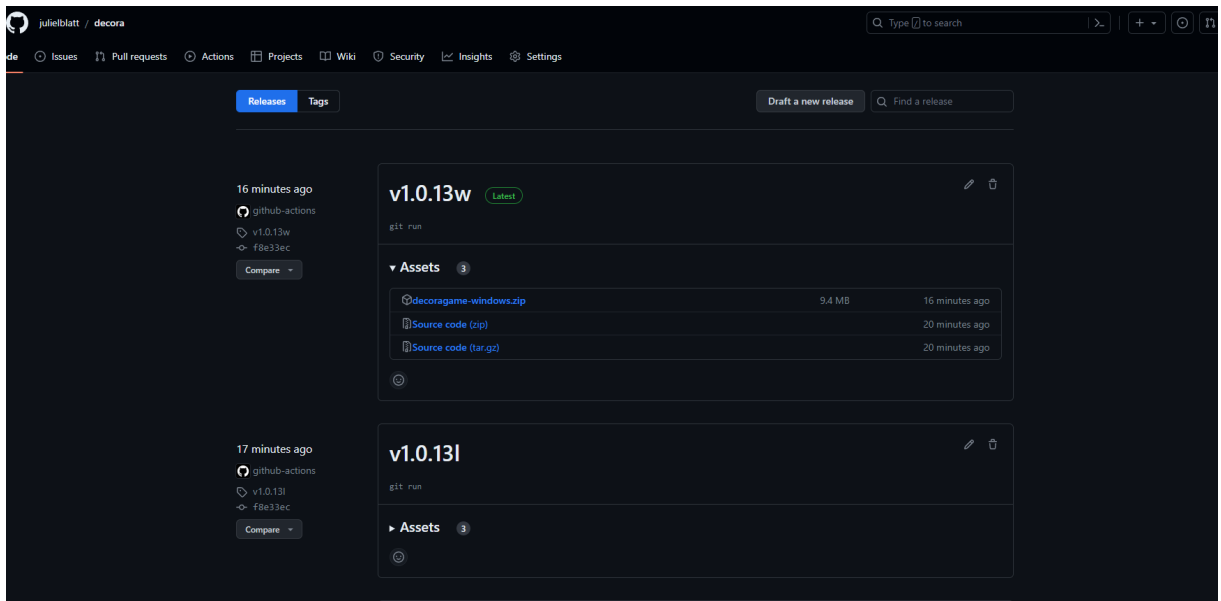


The screenshot shows an IDE interface with a file explorer on the left and a code editor on the right. The file explorer shows a project named 'DECORA' with various folders and files. The file 'widget\_test.dart' is selected. The code editor shows the following Dart code:

```
test > widget_test.dart > main > test("Generate next sequence")
1  import 'package:flutter_test/flutter_test.dart';
2  import 'package:decora/main.dart';
3
4  void main() {
5      test('Generate next sequence', () {
6          final game = DecoraGameState();
7
8          // Gera a próxima sequência
9          game.generateNextSequence();
10
11         // Verifica se a sequência foi gerada corretamente
12         expect(game.sequence.isNotEmpty, true);
13     });
14 }
15
```

Fonte: Autor

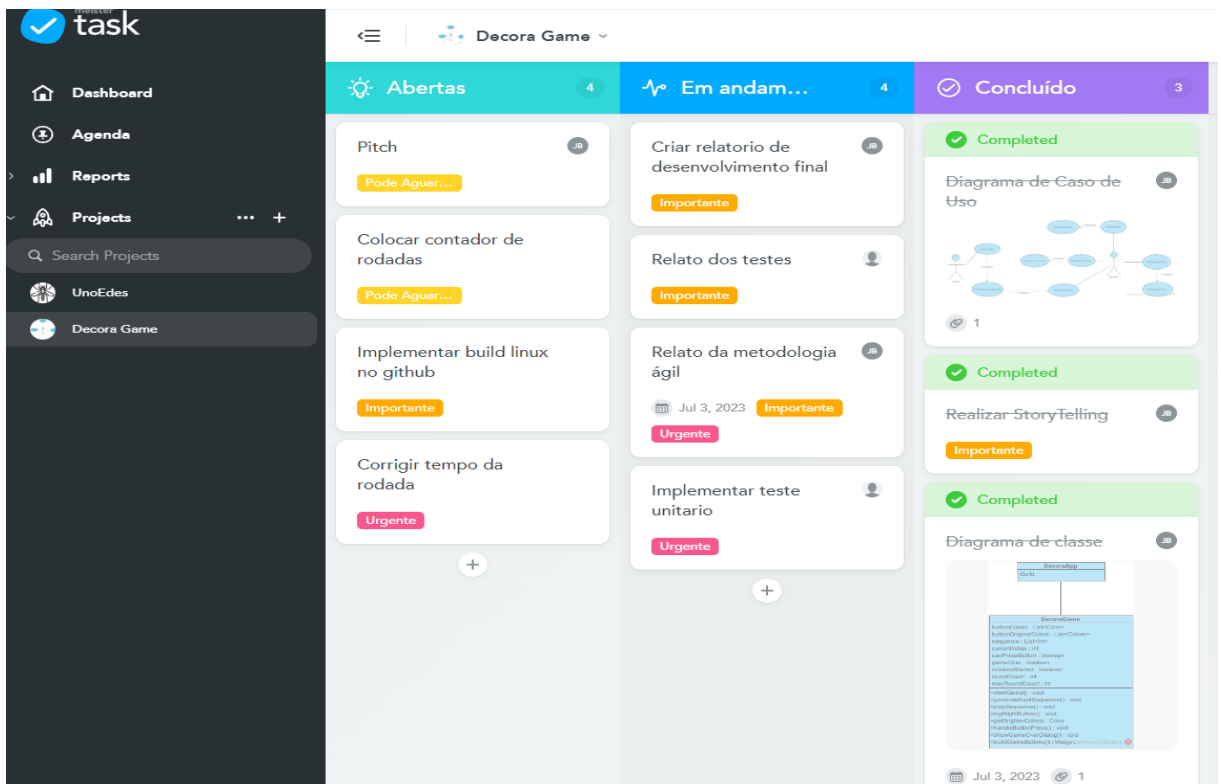
Foram também realizados testes de release com a utilização da ferramenta github, verificando a funcionalidade do aplicativo a cada alteração feita, executando o release feito pelo git actions, ao total foram feitas 26 releases em duas plataformas:



Fonte: Autor

## METODOLOGIA ÁGIL

Para o controle de tarefas realizadas, dada a sua importância e tempo para ser concluída, foi utilizado o site MeisterTask pela sua facilidade de visualização e uso para resolução das atividades apresentadas:



Fonte: Autor

