**Relatório Final**

**Jogo Mussum** &​ **Subida**​

**Projeto Detalhado de Software**

Professora Paulyne Matthews Jucá

Eduardo Alves, Guilherme Machado e João Pedro.

O Jogo “​**Mussum**​&**Subida”**​ é um ​*game* de percurso, em geral de 100 casas, atravessado por cobras e escadas. Caso o jogador caia na cabeça da cobra, ele descerá para a casa correspondente ao rabo da cobra e, caso caia na casa onde tem a parte inferior de uma escada, ele avançará até a parte inicial da escada, ou seja, pulará algumas casas. No caso deste trabalho, também foram acrescentadas às CASAS ESPECIAIS (Azar e Sorte), onde correspondem a casas que tem efeitos especiais, que podem fazer com que o jogador avance ou recue algumas casas.

O projeto está dividido nos seguintes pacotes:

1. **CONTROLADORES**​: Foram criados controladores de casas, cartas, dados, jogadores e jogo.
2. **DAOs**​: Foram criados três, um de jogador, um de casa e um de cartas.
3. **FÁBRICAS**:​ Foi criada uma fábrica de casas e uma de cartas.
4. **MODELS**​: Que está dividido nas subpastas: model.cartas, model.casa, model.dado e model.jogador.
5. **RECURSOS**:​ Contendo as imagens do tabuleiro, os pinos dos jogadores e o dado.
6. ***VIEWS***​: A “Main” para executar o jogo. Uma classe “PovoarCartasTabuleiro”, para adicionar as cartas. E a “PovoarCasasTabuleiro”, para mapear as casas do tabuleiro. A “View” para desenhar o jogo.
7. **INTERFACES**​: Foi criada uma interface com o comportamento do jogo, uma com os efeitos de ação, duas interfaces de fábrica, sendo uma para Fábrica de Cartas e a outra para Fábrica de Casas e uma do iterator de Jogador e outro iterator de Cartas.

● Os padrões utilizados para o desenvolvimento do jogo foram:

1. ***SINGLETON***​*:* O ​**Singleton** foi utilizado nos controladores e nos DAOs. O motivo do uso desse padrão foi pela necessidade de criar apenas uma instância de cada classe, pois seu uso garante que diversas chamadas de uma mesma classe não interfiram no desempenho. Também permite um acesso global a cada objeto.

1. **FÁBRICA**​: O padrão ​**Fábrica** ​foi utilizado para delegar a criação de instâncias em uma única classe. Assim, remove o forte acoplamento. Dentro deste trabalho o uso do padrão Fábrica foi importante pois ele tira a responsabilidade de diversas classes criarem o mesmo tipo de objeto, diminui a duplicação de código de objetos e junta essas responsabilidades em uma única classe.

1. ***ITERATOR***​: O padrão **Iterator**​ foi usado pois permite o acesso sequencial aos elementos de um conjunto sem expor sua implementação. Neste trabalho o uso do iterator foi para encontrar as cartas e para saber o jogador da vez.