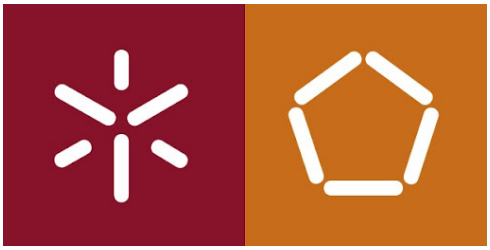

Universidade do Minho

Mestrado em Engenharia Informática

RasBet

Requisitos e Arquiteturas de Software

José Santos (a84288)
Leonardo Marreiros (pg47398)
Pedro Fernandes (pg47559)

21 de janeiro de 2022

Conteúdo

Lista de Figuras

1 Novos Requisitos

1.1 Modulação

A empresa que desenvolve o produto RASBet, com o objetivo de se diferenciar da concorrência, pretende permitir que os apostadores possam ter uma carteira com diferentes moedas (carteira multi-moeda).

Para implementar este novo requisito, começamos por modelar os componentes que iriam sofrer mudanças. Essas mudanças encontram-se nas figuras em seguida indicadas a vermelho. Foram também modulados os diagramas de sequência dos métodos que seriam alterados mas uma vez que as mudanças não foram muitas é redundante estar a apresentar estes diagramas.

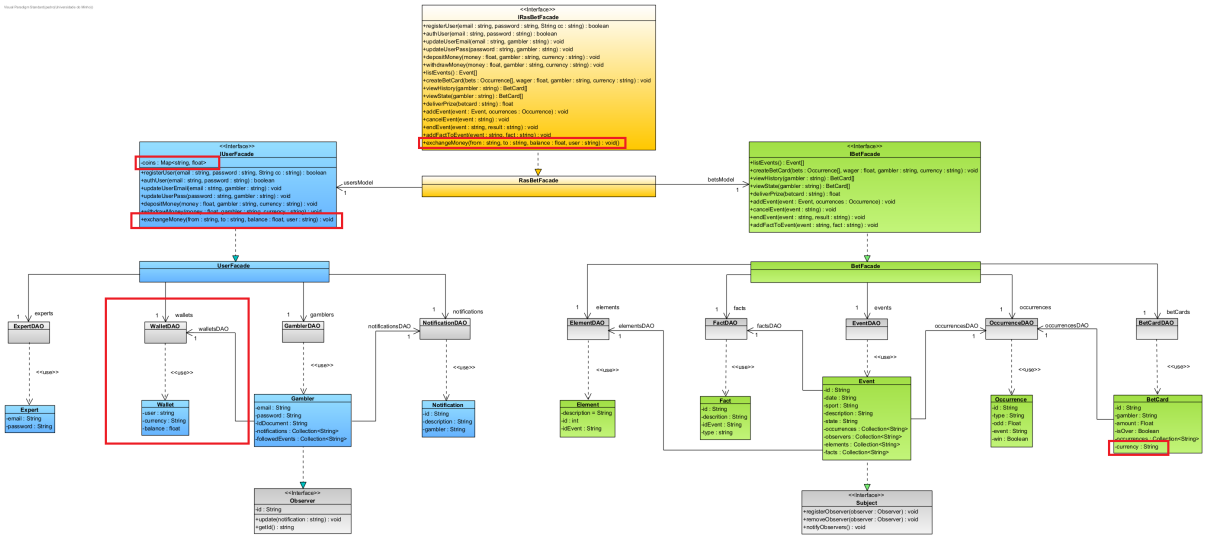

Figura 1: Diagrama de classes

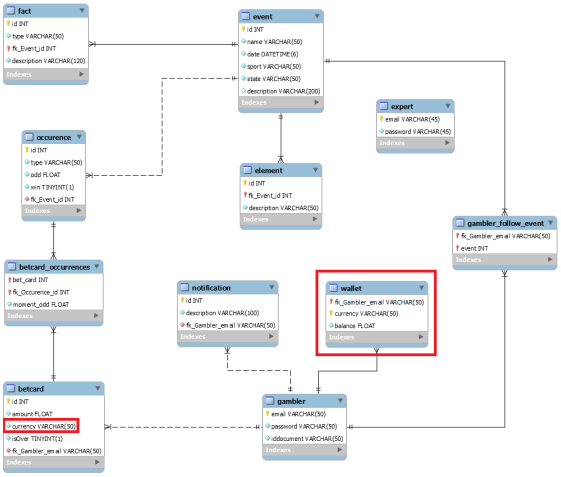

Figura 2: Modelo lógico

1.2 Implementação

Após realizada a modelação, foram rapidamente realizadas as seguintes alterações e adições de funcionalidades:

- **Estrutura de dados no userFacade:** De modo a obter uma relação entre cada moeda e o seu valor, foi criada uma estrutura em memória que indica o valor de cada moeda face a uma outra de modo a ser possível realizar trocas entre moedas.
- **Tipo de moeda no boletim de apostas:** Para permitir ao sistema guardar a informação de qual tipo de moeda o utilizador apostou, foi necessário criar um atributo em cada boletim para armazenar essa informação.
- **Nova tabela dedicada a carteiras na base de dados:** Anteriormente, a carteira era um simples atributo dentro de cada utilizador que, através do uso de um *float*, permitia-nos saber a quantidade de dinheiro do cada um. Tendo agora cada utilizador mais que uma carteira, uma para cada tipo de moeda, foi necessário criar uma nova tabela na base de dados contendo a chave do utilizador que a possui, a quantidade de dinheiro e também o tipo de moeda que ela tem.
- **Funcionalidades que envolvem dinheiro pedem tipo de moeda:** As funcionalidades que lidavam com transação de dinheiro tiveram agora que simplesmente receber um atributo adicional, o tipo de moeda, para poderem executar as suas tarefas na carteira certa.
- **Adicionar funcionalidade para converter moedas:** Este novo requisito naturalmente exigiu a adição de uma nova funcionalidade, a possibilidade de conversão entre moedas.

1.3 Conclusão

Para concluir a adição deste novo requisito foi de muita fácil implementação, uma vez que apenas foi necessário adicionar o conceito de múltiplas carteiras e a funcionalidade de converter moedas. Esta adição não gerou nenhuma mudança em relação ao estruturado até então, simplesmente um atributo carteira no utilizador, passou a ser um conjunto de carteiras. Podemos assim perceber a utilidade do trabalho feito nas fases anterior: ao ter uma arquitectura e modelação já definidas, a implementação foi rápida, eficaz e modularidade pelo que a adição de uma nova funcionalidade revelou-se uma tarefa bastante fácil.