



Universidade do Minho
Escola de Engenharia

Mestrado em Engenharia Informática
Engenharia Web

Betting House Web App

Interaction model

Equipa de desenvolvimento:

Carla Costa – PG38409
Sandra Sousa – PG38938

Braga, abril de 2019



Índice

Índice de Ilustrações	2
1. Introdução.....	4
2. Requisitos.....	5
2.1. Requisitos Funcionais.....	5
2.2. Requisitos não funcionais	6
3. Arquitetura da solução	7
3.1. Diagrama ER	7
3.2. Modelo IFML.....	8
3.2.1. Home	8
3.2.2. Privada	10
4. Conclusão e trabalho futuro	13
Referências	14

Índice de Ilustrações

Figura 1 - Diagrama ER	7
Figura 2 – Área Geral	8
Figura 3 - View da Área Geral.....	8



Figura 4 - Área de Administrador	9
Figura 5 - View da Área de Administrador.....	9
Figura 6 - Área Geral (Apostas)	10
Figura 7 - View da Área Geral (Apostas)	10
Figura 8 - Área Privada.....	11
Figura 9 - View da Área Privada	11



1. Introdução

O seguinte relatório foi elaborado no contexto da Unidade Curricular de Engenharia Web e pretende demonstrar o modelo IFML (*Interaction Flow Modeling Language*) desenhado para representar a futura aplicação, pois este modelo serve o propósito de expressar o conteúdo, a interação do utilizador e o comportamento do *front-end* de aplicações de software.[1]

Este projeto tem como objetivo entender como estará organizada a aplicação, assim como ver as relações dentro da mesma. Tudo isto para atingir o objetivo principal, que é a criação de uma plataforma web de apostas que permite a utilizadores realizar apostas sobre eventos desportivos.

O documento está organizado por dois capítulos principais e ainda o presente capítulo, e a conclusão e trabalho futuros, fazendo um total de quatro capítulos. No segundo capítulo foram abordados os requisitos necessários para a realização do projeto, tanto funcionais, como não funcionais.

Por fim demonstramos no penúltimo capítulo e justificar o que foi construído, que passa por um diagrama inicial da base de dados (diagrama ER), o modelo IFML e as *views* provenientes.

A metodologia utilizada para a elaboração do modelo e das *views* existiu com auxílio da ferramenta sugerida pelos docentes, *WebRatio*, assim como pesquisa na internet para entender algumas funcionalidades principais.



2. Requisitos

2.1. Requisitos Funcionais

Os requisitos funcionais encontrados e que achamos mais relevantes para a realização deste projeto são:

1. Um apostador deverá registar-se no sistema através de: email, nome e password;
2. Um apostador deverá fazer login com email e password para aceder à aplicação web BetESS;
3. Um administrador poderá ser um apostador;
4. Um apostador poderá ser premium;
5. Um apostador logado deverá ter disponível uma lista de eventos;
6. Um apostador logado poderá depositar uma quantia de ESSCoins;
7. Um apostador logado poderá realizar apostas;
8. Um apostador logado poderá editar informações sobre si (email, nome e password);
9. Um apostador logado poderá consultar o seu histórico de apostas (apostas em aberto, apostas ganhas e apostas perdidas);
10. Um apostador premium logado poderá ter um *cashback* até 10% por cada aposta realizada;
11. Um apostador premium poderá ter melhores odds que os apostadores normais;
12. Um administrador poderá criar eventos para apostar;
13. Um administrador poderá definir as odds dos eventos;
14. Um administrador poderá receber notificação do valor ganho/perdido pelo evento;
15. Um administrador deverá ser notificado do resultado final de uma aposta;
16. Um administrador poderá consultar uma lista de eventos;
17. Um administrador poderá consultar uma lista de apostadores.



2.2. Requisitos não funcionais

Os requisitos não funcionais encontrados e que achamos mais relevantes para a implementação deste projeto são:

1. Um evento a apostar deve apresentar duas equipas, três possíveis resultados e respetivas odds (exemplo: FC Barcelona – SC Braga (1.38, 4.40, 8.00);
2. O apostador ao efetuar uma aposta deverá indicar o resultado pretendido e o valor a apostar;
3. O sistema deve encriptar as passwords na base de dados, de forma a garantir a segurança dos utilizadores do sistema no caso de um ataque à base de dados do sistema.

3. Arquitetura da solução

3.1. Diagrama ER

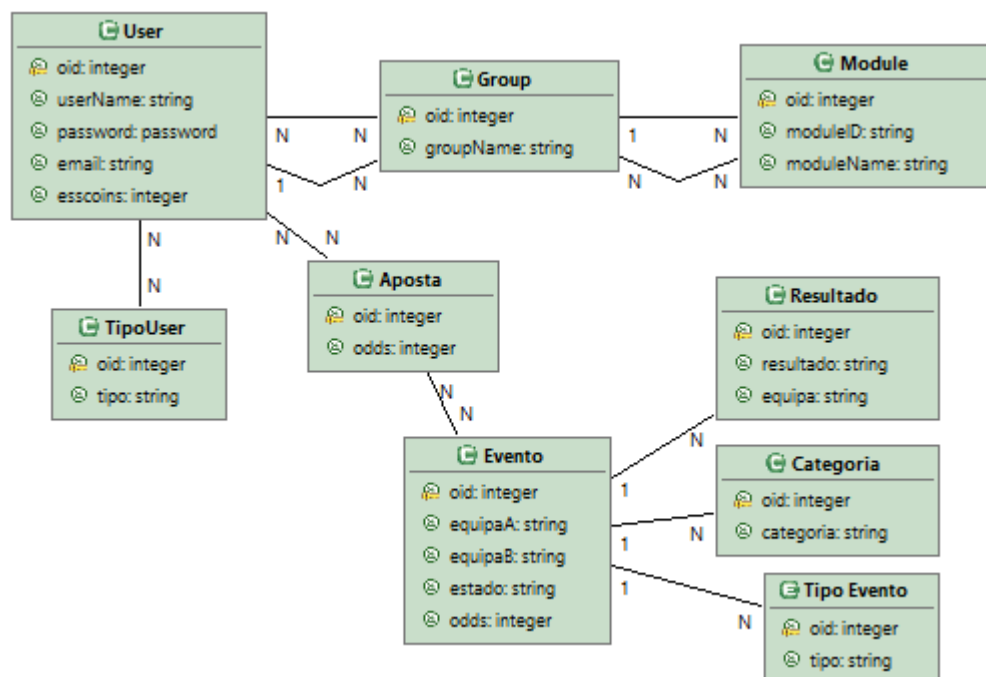


Figura 1 - Diagrama ER

A imagem acima representa como a base de dados da aplicação irá estar organizada, é importante ter em consideração que ao longo do desenvolvimento da mesma, que será a segunda fase do trabalho, esta configuração da base de dados poderá sofrer alterações.

É importante de referir que para este projeto em específico consideramos as entidades *Group* e *Module*, pois são aquelas que nos permitiram realizar as diferentes relações da aplicação no modelo IFML, principalmente a nível do login com a páginas privadas, ou seja, um utilizador ao realizar login só poderá aceder ao que lhe é permitido.

3.2. Modelo IFML

Para o modelo IFML optamos por dividir a sua construção em dois, uma para a homepage que liga nessa mesma à área de administrador e um outro que liga à área privada

3.2.1. Home

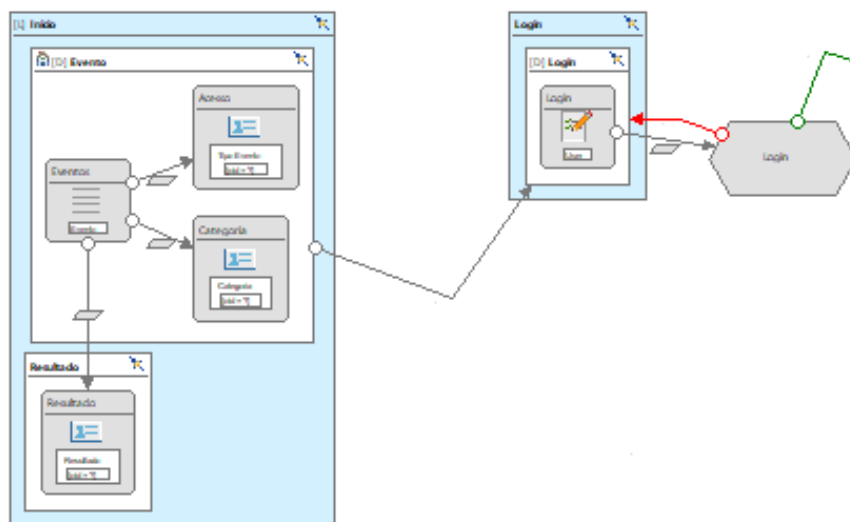


Figura 2 – Área Geral

Inicio						
Inicio > Evento						
Login						
Eventos						
equipaA	equipaB	estado	odds			
> Portugal	França	Aberto		Acesso	Categoria	Resultado
> Barcelona	Madrid	Aberto		Acesso	Categoria	Resultado
> Porto	Benfica	Fechado		Acesso	Categoria	Resultado
> Sporting	Benfica	Aberto		Acesso	Categoria	Resultado
> Brasil	Israel	Aberto		Acesso	Categoria	Resultado

Figura 3 - View da Área Geral

A área geral é aquela em que todos os utilizadores poderão interagir, contém todos os eventos abertos, também é possível ver se o evento é Geral (para todos os

utilizadores registados) ou Premium (para apenas utilizadores premium), ainda é possível ver de que categoria é o evento e por fim, se já o tiver, o resultado do mesmo.

Ainda para todos os utilizadores gerais, estes conseguem aceder à página de login e caso tenham já o registo efetuado entrar na aplicação em mais profundidade.

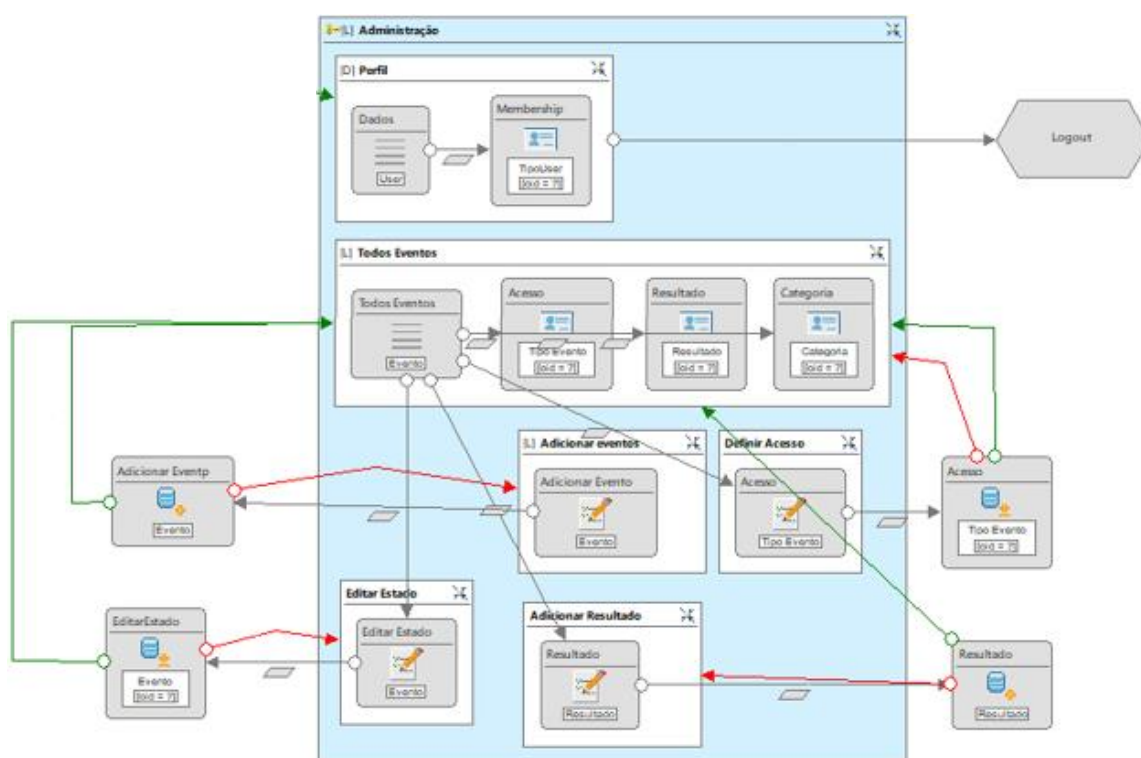


Figura 4 - Área de Administrador

Administração		Início	
Todos Eventos			
Adicionar eventos			
Logout			
Dados			
userName	password	email	
admin	admin	admin@email.com	
		Membership	

Figura 5 - View da Área de Administrador

Um utilizador que seja administrador ao entrar na sua área poderá efetuar ações como adicionar eventos, editar o estado dos eventos (se estão abertos ou fechados) e ainda adicionar os resultados referentes ao evento.

Outras ações como visualizar o perfil e todos os eventos até à data também estão permitidos ao utilizador.

3.2.2. Privada

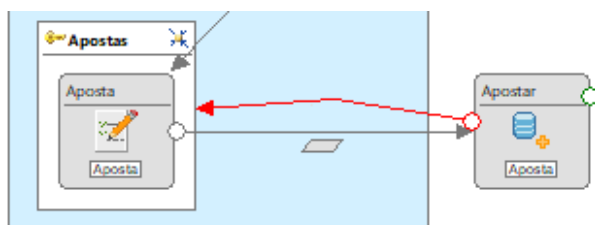


Figura 6 - Área Geral (Apostas)

Início

Privada

Início > Evento

Eventos

equipaA	equipaB	estado	odds		Acesso	Categoria	Resultado	Apostar
> Portugal	França	Aberto			Acesso	Categoria	Resultado	Apostar
> Barcelona	Madrid	Aberto			Acesso	Categoria	Resultado	Apostar
> Porto	Benfica	Fechado			Acesso	Categoria	Resultado	Apostar
> Sporting	Benfica	Aberto			Acesso	Categoria	Resultado	Apostar
> Brasil	Israel	Aberto			Acesso	Categoria	Resultado	Apostar

Figura 7 - View da Área Geral (Apostas)

Para utilizadores privados, que sejam apostadores estes terão na Área Geral a opção Apostar, que apenas será permitida de aceder quando o utilizador estiver logado.

Também dependendo do tipo de *membership* este poderão apostar ou não em certos eventos, isto é, se um utilizador for do tipo Normal então apenas poderão apostar em eventos com estado Geral. Já um utilizador do tipo Premium poderá apostar em todos os eventos, quer este sejam tipo Geral ou Premium.

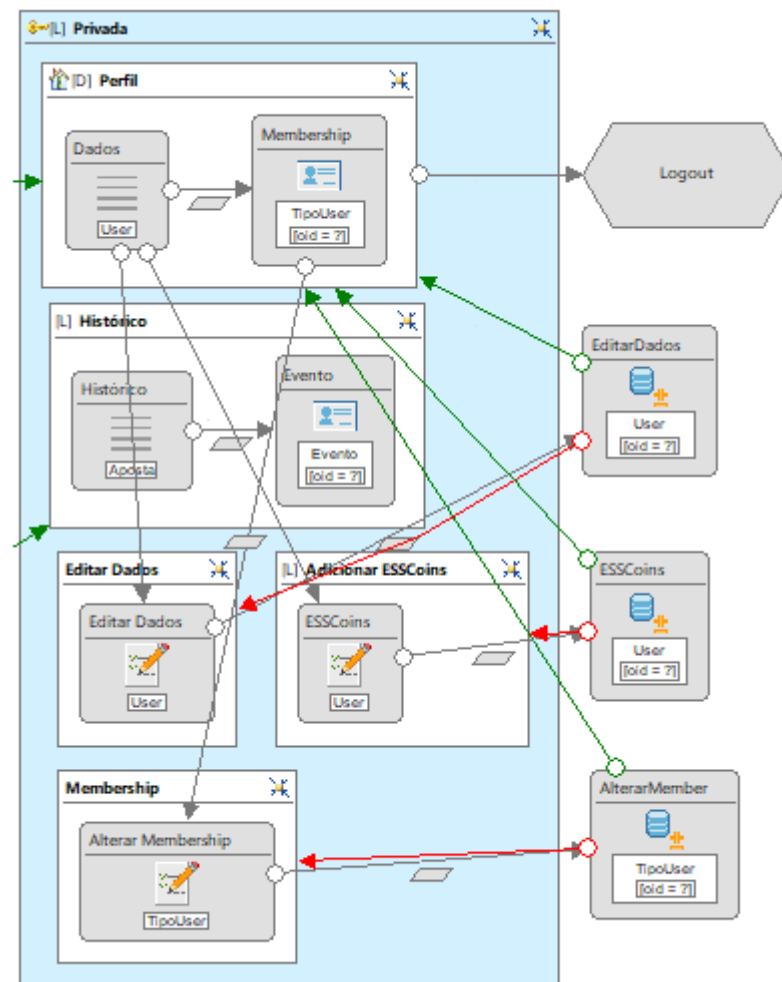


Figura 8 - Área Privada



Figura 9 - View da Área Privada

Por fim, temos a Área Privada, que é acedida por utilizadores apostadores que se tenham registado e logado. Nesta área o utilizador poderá ver e editar os seus dados de perfil, tanto dados de login, como adicionar ESSCoins, que é a moeda de aposta, e



alterar a *membership*. Também terá a opção de ver o seu histórico de apostas e ver o resultado das mesmas.



4. Conclusão e trabalho futuro

Neste trabalho abordamos o modelo IFML, elaboramos o mesmo seguindo os requisitos identificados no segundo capítulo. Concluindo que a construção do modelo ajuda a entender melhor a futura aplicação e a relações entre eles.

Concluimos que atingimos o objetivo principal, que foi contruir um modelo IFML que represente a aplicação e a relações dentro desta. Encontramos alguns entraves ao nível da utilização da aplicação WebRatio, nomeadamente para criar as exceções em alguns parâmetros, como por exemplos conseguir que apareçam apenas os eventos abertos na página principal.

Apesar dos entraves identificados conseguimos obter uma modelo capaz de representar bem a nossa futura aplicação, onde acreditamos que será mais fácil de criar a exceção referenciada, entre outras situações como a opção de registo que não nos foi possível elaborar por falta de alguns recursos.

Como trabalho futuro pretendemos desenvolver a aplicação web em Node.js e AngularJS, utilizando uma base de dados em MySQL.

A *framework* node.js vai ser utilizada no *backend*, visto que é uma ferramenta gratuita e de código aberto usada para criar aplicações do lado do servidor. Uma das maiores vantagens é que as aplicações escritas em Node.js podem ser executadas em Linux, Microsoft e iOS. Para além disso, esta *framework* oferece uma biblioteca com vários módulos que simplificam o processo de desenvolvimento web. [2]

A *framework* AngularJS vai ser utilizada no *frontend*, visto que é uma ferramenta de aplicações web dinâmicas e permite que seja usado HTML para a criação da interface. Uma vantagem desta *framework* é que esta suporta a estrutura de programação MVC (*Model View Controller*). [3] [4]

A base de dados será em MySQL, visto que é *opensource*, confiável e fácil de gerir.



Referências

- [1] IFML, “IFML: The Interaction Flow Modeling Language | The OMG standard for front-end design,” 2018. [Online]. Available: <https://www.ifml.org/>. [Accessed: 14-Apr-2019].