

Universidade do Minho - Escola de Engenharia

Relatório do trabalho prático de Engenharia Web

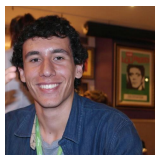
BetESS - PARTE I

Autores :

Diana Costa (A78985)



Marco Silva(A79607)



Resumo

O presente relatório é uma continuação do projeto "BetESS", pertencente à unidade curricular de Arquiteturas de *Software*, e tem por objetivo definir a interação entre os utilizadores e uma aplicação, utilizando linguagem IFML e a ferramenta *WebRatio*.

Conteúdo

1	Introdução	3
2	Problema	3
3	Base de Dados	4
3.1	Primeira versão	4
3.2	Versão final	5
4	Arquitetura do sistema	6
4.1	Site View - Main	6
4.1.1	Área About Us	7
4.1.2	Área Sign in/up	8
4.2	Site View - Admin	10
4.2.1	Área Sports	11
4.2.2	Área Leagues	13
4.2.3	Área Bets	15
4.2.4	Área Events	16
4.2.5	Área Teams	18
4.2.6	Área Logout	20
4.3	Site View - Users	21
4.3.1	User	21
4.3.1.1	Área Credits	22
4.3.1.2	Área Profile	24
4.3.1.3	Área Bet	26
4.3.1.4	Área My Bets	27
4.3.1.5	Área Notifications	28
4.3.1.6	Área Exit	29
4.3.2	User Premium	30
5	Conclusões e Sugestões	32

1 Introdução

Este projeto foi elaborado no âmbito da Unidade Curricular de Engenharia Web, do 4^a ano do Mestrado Integrado em Engenharia Informática, com vista à definição da interação entre *users* e uma aplicação, utilizando a ferramenta *WebRatio* e a linguagem IFML. A aplicação a simular segue a mesma linha de raciocínio do projeto "BetESS", já realizado pelo grupo, que consistia na implementação de uma plataforma de apostas. As funcionalidades a desenvolver mantêm-se, sendo que apenas seria necessário adicionar utilizadores *premium*, que teriam acesso a eventos exclusivos, desde que pagassem a quantia requerida para tal. Assim, esta aplicação deverá suportar o registo de apostadores, e será gerida por um administrador, que deverá criar eventos, associando as equipas existentes com as *odds* respetivas. Os utilizadores registados podem aceder a uma lista de eventos disponíveis e realizar uma ou mais apostas, sendo que apenas possível apostar num resultado (apostas sobre múltiplos resultados não são permitidos). Aquando da realização da aposta, é necessário que o jogador indique a quantia a apostar e o resultado pretendido, para que o sistema possa manter uma lista das apostas realizadas por cada apostador. Esta aposta, que possui um estado conforme a sua condição, quando passar de aberta para fechada, notifica os apostadores do resultado da mesma, e se for o caso, do valor ganho em *ESScoins* conforme a *odd*.

Assim, neste relatório, pode-se constatar o trabalho desenvolvido pelo grupo referente a duas fases: uma primeira, onde se construiu o *domain model*, de forma a servir de estrutura de base de dados para a aplicação, e uma outra, onde se modelaram as diferentes *site views*, necessárias ao *design* da interface e a todas as operações de *backend* envolvidas. Todas estas fases implicaram a tomada de decisões pelo grupo, que são explicadas ao longo do relatório.

Desta forma, o relatório começa por introduzir o problema e declarar e explicar a base de dados, apresentando a sua evolução. De seguida, apresenta-se toda a arquitetura das *site views*. Nesta secção são discutidas as várias *site views*, juntamente com as decisões do grupo, tanto ao nível de design, como ao nível da camada de negócio. Depois, o grupo achou por bem dividir a secção da *site view* do *user* em duas, para que fosse discutido todo o trabalho e pensamento que levaram ao desenvolvimento do *user premium*. O relatório termina com as conclusões, em que o grupo faz uma análise do trabalho desenvolvido ao longo do semestre, tendo em conta as dificuldades obtidas.

2 Problema

Pretende-se simular um sistema que possibilite o registo de utilizadores, que indicam uma quantia monetária (*ESScoins*) disponível para aposta. Ao verificar uma lista de eventos, o apostador escolhe um evento "Aberto" com a respetiva *odd* (no formato equipa1 - equipa2(<*odds* respetivas>)), e decide apostar uma quantia, sendo que o sistema deverá manter uma lista de apostas por utilizador. Quando as apostas realizadas por um *user* passarem ao estado "Fechada" (assume-se que o resultado do evento é conhecido), o utilizador deverá ser notificado de tal ocorrência, juntamente com o seu ganho com a mesma. Para determinar o valor ganho numa aposta deverá ser definido para cada evento as *odds* para os possíveis resultados e sobre essas *odds* será calculado o valor ganho numa aposta. Exemplificando, no caso de vitória do SC Braga na aposta "FC Barcelona – SC Braga (1.38 , 4.40, 8.00):

1. os apostadores que tenham indicado a vitória do SC Braga recebem 8.00 *ESScoins* por cada *ESScoin* apostada;
2. os apostadores que tenham indicado a vitória do FC Barcelona ou o empate recebem 0.

É de realçar que **não existe o estado "A decorrer"** como em muitos sistemas reais, sendo que este, no sistema do grupo, é apenas ilusório. Não serão permitidas efetuar operações como atualizar *odds* ou *cashout* enquanto um evento decorre.

3 Base de Dados

Para ser possível a persistência dos dados, o grupo viu-se na obrigação da criação de uma base de dados. Esta tarefa é facilitada pelo próprio *WebRatio*, que permite a criação de um *domain model* que automatiza este processo. Ainda assim, é importante notar que **as cardinalidades nesta ferramentas são opostas às que seriam observáveis num modelo lógico de dados comum.**

Posto isto, o grupo vai detalhar o raciocínio utilizado para a elaboração do domain model, assim como a sua evolução, que implicou o aparecimento de dois *models* diferentes.

3.1 Primeira versão

Numa primeira fase, o grupo aproveitou a estrutura de dados proveniente do projeto da unidade curricular de arquiteturas de *software*, que se apresenta abaixo. São visíveis as entidades:

- **Desporto**, que envolve várias ligas e equipas;
- **Equipa**, já que um evento envolve sempre duas equipas;
- **Liga**, que possui várias equipas;
- **Evento**, envolvendo equipas, que pertencem a determinada liga e a determinado desporto;
- **Aposta**, que é efetuada sobre um evento;
- **Notificação**, que está relacionada com um utilizador e com uma aposta;
- **Utilizador**, onde são armazenados os dados de um *user*;
- **Group**, entidade típica do *WebRatio* para definir grupos de utilizadores;
- **Module**, que serve para atribuir *site views* aos diferentes tipos de utilizadores.

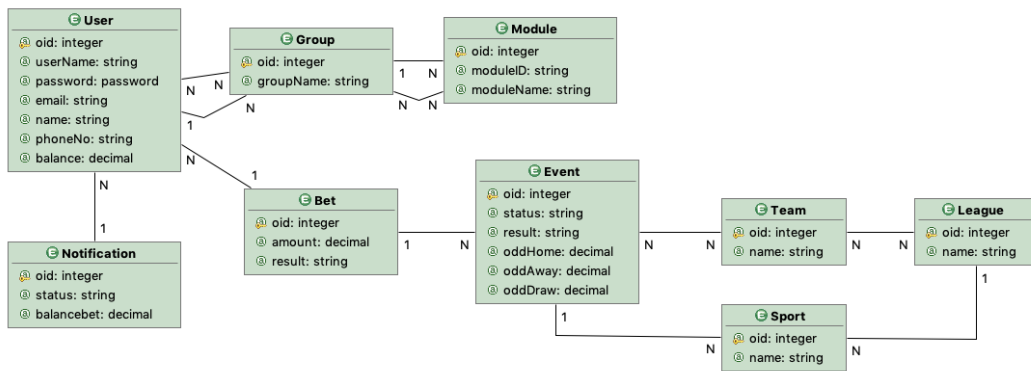


Figura 1: Versão inicial do *domain model*.

No entanto, e como todas as versões iniciais, este modelo de dados foi modificado para melhor atender aos requisitos do projeto e para conceber rapidez às *queries*.

3.2 Versão final

Do modelo de dados anterior para o presente, as entidades mantiveram-se, e apenas os relacionamentos e alguns atributos sofreram alterações para melhor satisfazer as necessidades do projeto. Desta forma, apenas serão explicadas e justificadas as modificações que levaram ao aparecimento de um novo modelo.

Começando pelas mudanças nos **atributos**, adicionou-se o campo "Premium" ao User, de forma a poder saber se ele é *premium* ou não, o que lhe dá acesso a eventos exclusivos. Na mesma linha de pensamento, acrescentou-se o atributo "Premium" ao evento, para assim ser possível distinguir os eventos normais dos exclusivos. Por fim, adicionou-se também o campo "paid" à bet, como forma de indicar o status da aposta (mais rápida leitura para verificar se corresponde a um evento a decorrer ou terminado).

Quanto aos **relacionamentos**, uniu-se a notificação ao evento, uma vez que, de acordo com o modelo anterior, não seria possível atribuir uma notificação de um evento unicamente a um utilizador. Para além disso, alterou-se os relacionamentos envolventes do evento, de forma a facilitar as queries e a simplificar o modelo. O relacionamento entre o evento e a equipa foi duplicado e modificado na sua cardinalidade, para obrigar o evento a ter duas equipas, e para conseguir atribuir ordem inequívoca entre a equipa da casa e a equipa de fora.

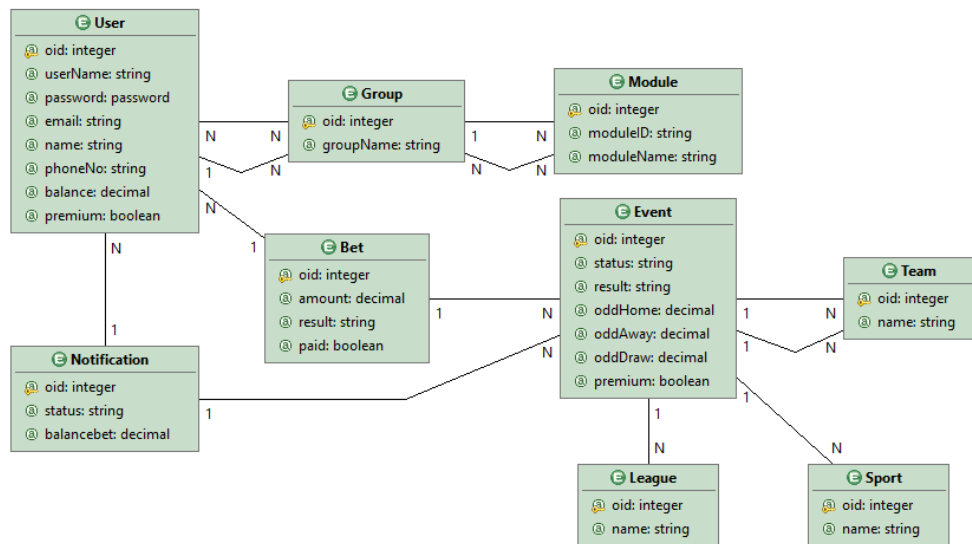


Figura 2: Versão final do *domain model*.

4 Arquitetura do sistema

Uma vez definido o modelo de dados, o grupo passou para o desenvolvimento das *site views*, o que consumiu grande parte do tempo dispendido no projeto. As *site views* criadas resumem-se a apenas três:

- **Main:** onde se engloba a lógica de registo e login;
- **Users:** view onde estão presentes todas as funcionalidades dos utilizadores (normais ou *premium*);
- **Admin:** onde estão todas as tarefas relacionadas com o administrador da plataforma.

Cada *site view* é complexa e precisa de ser detalhada, sendo que as secções abaixo tratam de o fazer.

4.1 Site View - Main

Como já referido, esta *site view* é responsável por abranger as tarefas de login ou registo de um utilizador. O grupo optou por dividir em duas áreas, uma para receber os utilizadores na plataforma, e outra para possibilitar a entrada no sistema, como se demonstra abaixo.

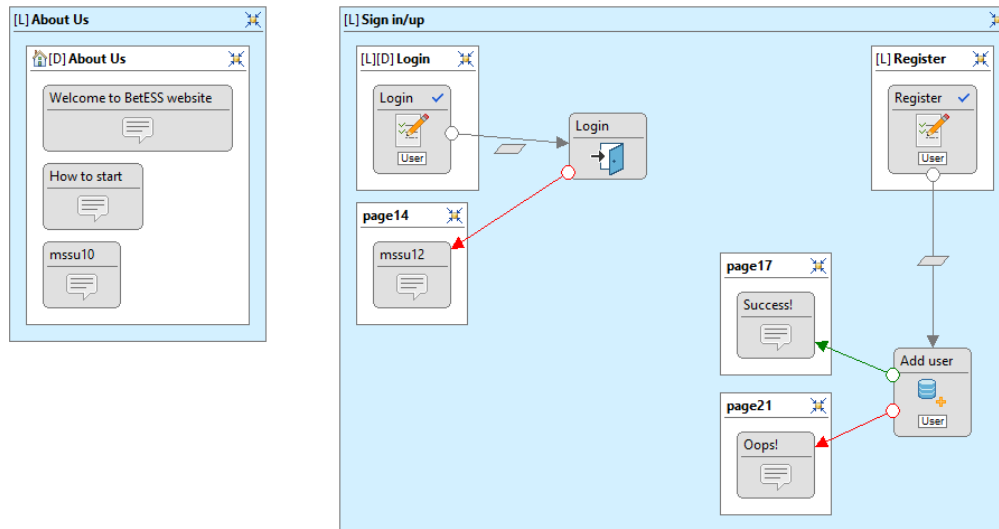


Figura 3: Arquitetura da *site view* "Main" no WebRatio

Posto isto, nas próximas secções explica-se cada área mencionada, explicando a lógica seguida.

4.1.1 Área About Us

Esta área serve apenas o propósito de receber os utilizadores ao website de apostas. Aqui é mostrada uma mensagem de boas-vindas, assim como uma breve explicação de como o utilizador pode apostar.

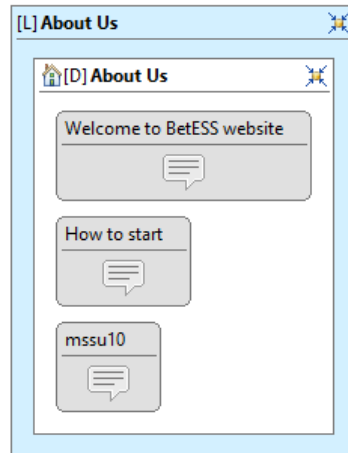


Figura 4: Lógica da área "Main" no *WebRatio*

Assim, só existe uma página principal, onde são mostradas as mensagens personalizadas (View Component - "Message"). A mensagem inferior "Happy betting" foi propositadamente deslocada através do Layout do *WebRatio*, de forma a obter uma melhor aparência.

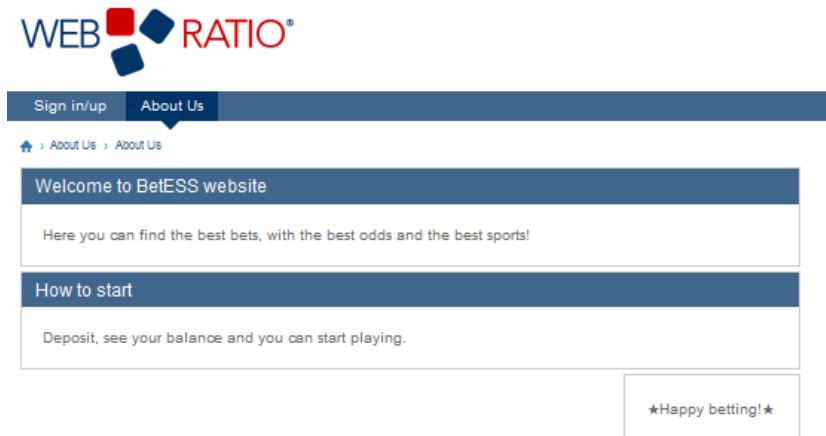


Figura 5: Área "About Us" - Página "About Us"

4.1.2 Área Sign in/up

Esta área está dividida em duas páginas principais: a de "Login" e a de "Register".

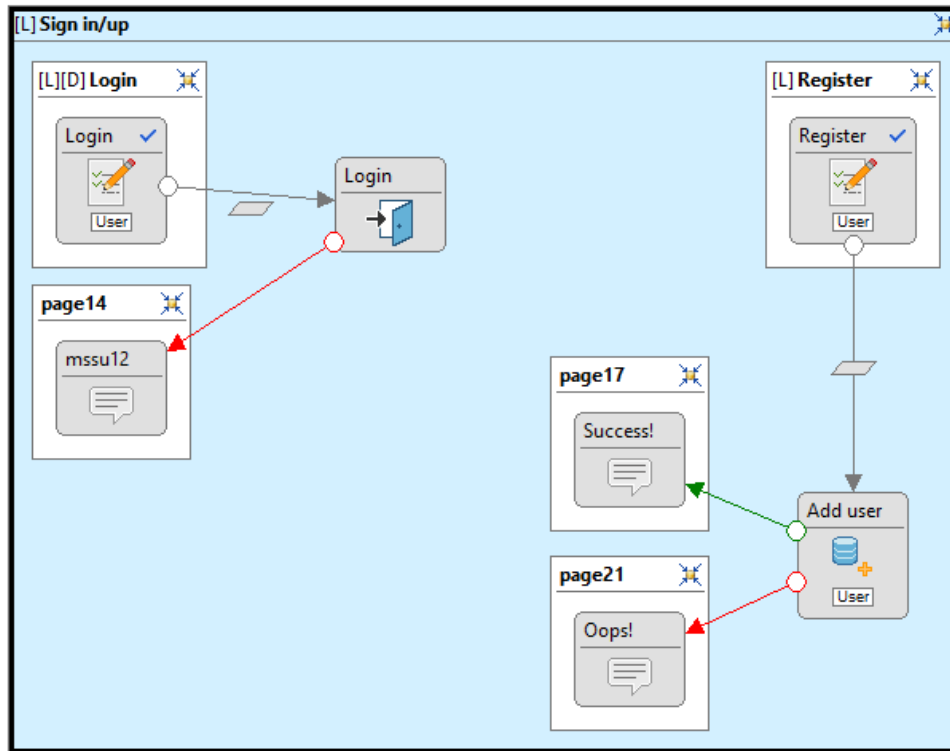


Figura 6: Lógica da área "Sign in/up" no *WebRatio*

Para o *login*, o grupo optou por utilizar o View Component "Form", para que o utilizador pudesse inserir o *username* e *password* necessárias a esta operação. Em caso de sucesso, o *user* é redirecionado para a sua *site view* correspondente, através do Session Component "Login". Caso contrário, é-lhe mostrada uma mensagem de erro.

A captura de tela mostra a interface de usuário da página "Login". No topo, há o logotipo "WEB RATIO" e uma barra de navegação com os links "Sign in/up" e "About Us". Abaixo da barra de navegação, há um link "Sign in/up" e um link "Login". O formulário de login é composto por dois campos de entrada: "Username" e "Password", ambos com botões de seta para alternar entre mostrar e ocultar o conteúdo. Abaixo dos campos, há um botão "Login".

Figura 7: Área "Sign in/up" - Página "Login"

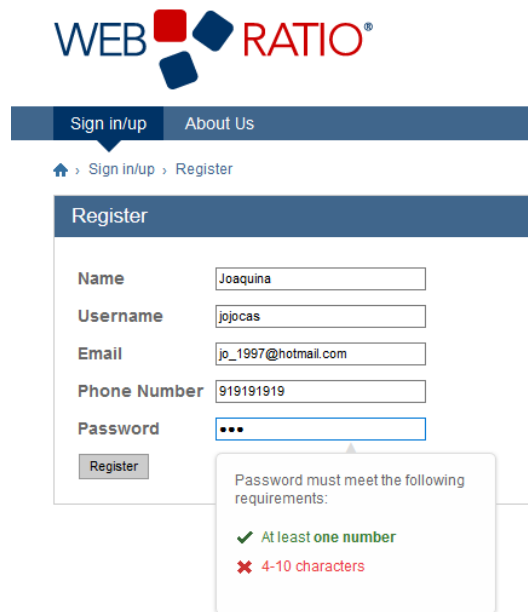
Já para a página de registo, para além do formulário de recolha de dados, foi necessário adicionar uma Operation "Create", de forma a enviar os dados introduzidos para a base de dados, criando assim um novo utilizador. Caso a operação corra bem ou mal, é mostrada uma mensagem conforme ao utilizador, que, posteriormente, terá de efetuar o login para entrar na plataforma.



The screenshot shows the WebRatio logo at the top. Below it is a navigation bar with 'Sign in/up' and 'About Us'. A breadcrumb trail indicates the path: Home > Sign in/up > Register. The main heading is 'Register'. The form contains five input fields: 'Name', 'Username', 'Email', 'Phone Number', and 'New Password'. A 'Register' button is located at the bottom left of the form.

Figura 8: Área "Sign in/up" Página "Register"

É de notar as regras que a *password* impõe, definidas pelo grupo, para simular um ambiente real mais seguro.



This screenshot shows the same Register form as Figure 8, but with sample data entered: Name (Joaquina), Username (jpocas), Email (jo_1997@hotmail.com), and Phone Number (919191919). The Password field is masked with dots. A tooltip is displayed over the Password field, stating 'Password must meet the following requirements:'. It lists two requirements: a green checkmark for 'At least one number' and a red X for '4-10 characters'.

Figura 9: Área "Sign in/up" Página "Register- Força da password

4.2 Site View - Admin

Nas próximas secções será explicada toda a lógica da *site view* do administrador da plataforma. Começa-se, nesta secção, por mostrar e explicar as decisões do grupo num nível mais elevado e menos complexo, continuando, nas secções seguintes, a aumentar o detalhe, explicando cada área.

Assim, apresenta-se abaixo uma arquitetura geral desta *site view*, de forma a mostrar a primeira linha de raciocínio do grupo.

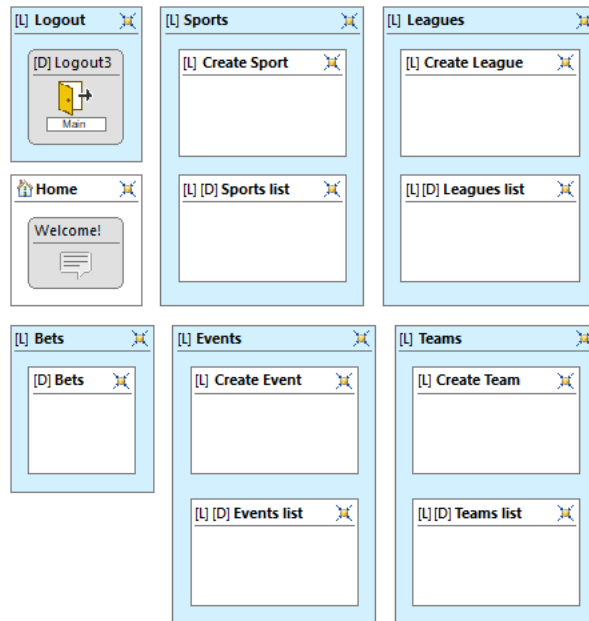


Figura 10: Arquitetura geral da *site view* do *admin* no *WebRatio*

Tal arquitetura equivale às seguintes páginas no *browser*. Cada página marcada com "L" (*Landmark*) é um "menu", como o "Events list" e "Create Event" mostrados abaixo. Se nenhuma das opções for clicada, e apenas se seleccionar "Events", vai para a página por *default*, marcada com um "D" em cada área.

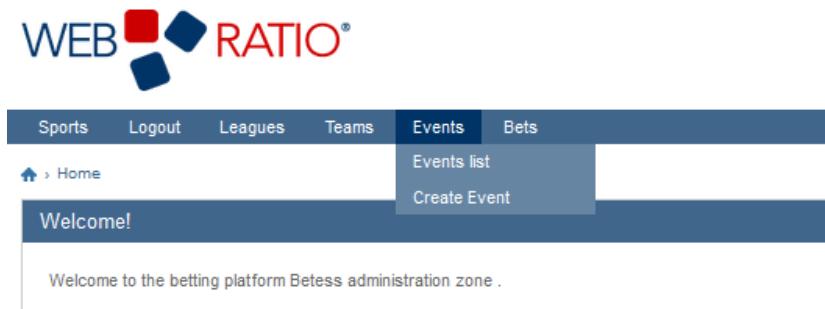


Figura 11: Imagem do *browser* correspondente à arquitetura no *WebRatio*

4.2.1 Área Sports

Esta área é reservada para que o administrador possa gerir os desportos da plataforma, ou seja, ver, adicionar ou eliminar os mesmos.

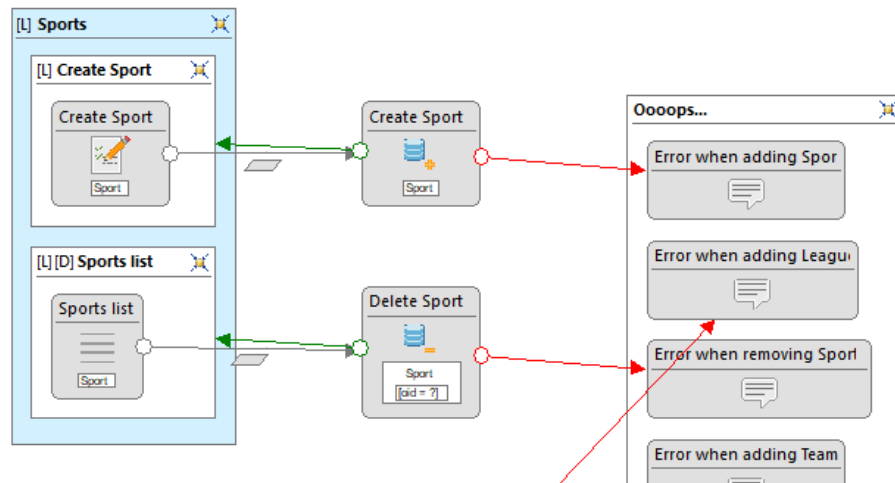


Figura 12: Lógica da área "Sports" no *WebRatio*

Desta maneira, observam-se duas páginas: "Create Sport" e "Sports' list". A primeira serve unicamente para que se possam adicionar desportos, inserindo o nome, como *Futebol* ou *Ténis*. Assim, através de um View Component "Form" é recebido o nome do desporto, para que o mesmo seja adicionado à base de dados, através da Operation "Create". Em caso de sucesso, o admin permanece na página e, em caso contrário, é-lhe mostrada uma mensagem de erro personalizada.

A captura de tela mostra a interface de usuário da página "Create Sport" no WebRatio. No topo, há um menu de navegação com os itens "Sports", "Logout", "Leagues", "Teams", "Events" e "Bets". Abaixo do menu, há um breadcrumb "Sports > Create Sport". O título da página é "Create Sport". Abaixo do título, há um formulário com um campo de texto rotulado "name" e um botão "Create".

Figura 13: Área "Sports": Página "Create Sport"

A segunda página disponibiliza ao admin uma lista de desportos existentes (**ordenados por nome**), com um botão que possibilita a eliminação de algum deles. Tal é conseguido através da View Component "Simple List", e pela Operation "Delete", que remove o desporto da base de dados. Em caso de sucesso, o admin permanece na página e, em caso contrário, é-lhe mostrada uma mensagem de erro personalizada.

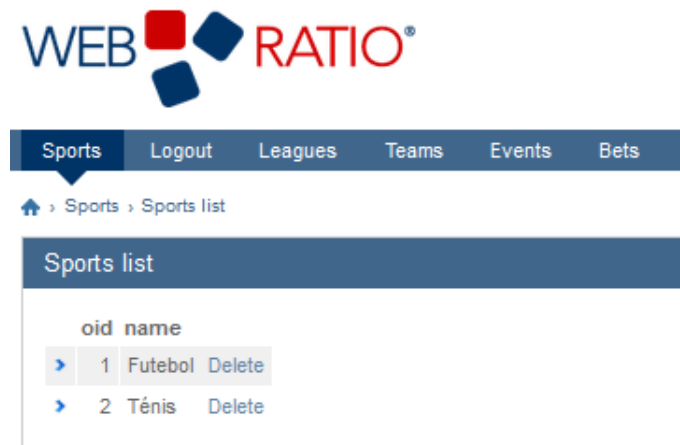


Figura 14: Área "Sports": Página "Sports' list"

4.2.2 Área Leagues

A presente área destina-se à visualização e manipulação de ligas desportivas, i.e., o administrador pode ver, adicionar ou eliminar ligas do sistema.

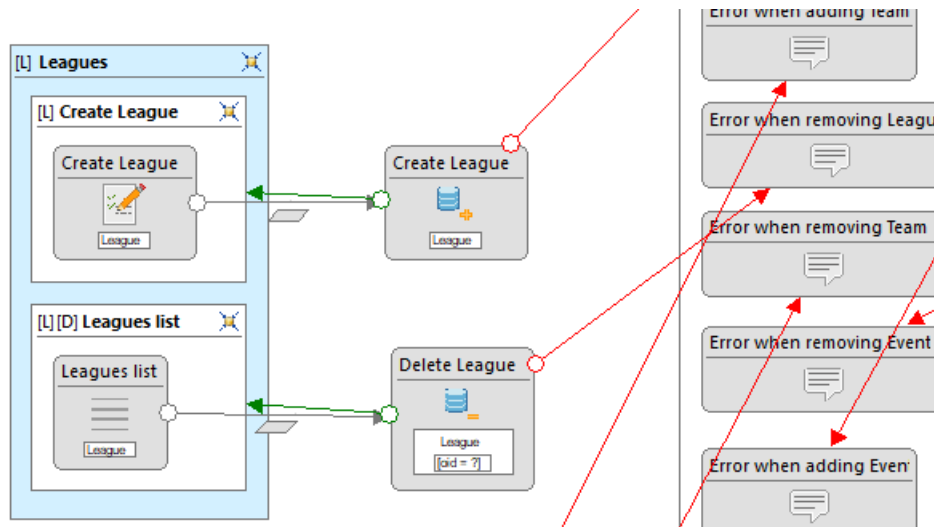


Figura 15: Lógica da área "Leagues"no *WebRatio*

Assim, existem duas páginas: "Create League" e "Leagues' list". A primeira serve como apoio à criação de ligas e, tal como na secção anterior, serve-se de um "Form" e "Create" para tal. Em caso de sucesso, o admin permanece na página e, em caso contrário, é-lhe mostrada uma mensagem de erro personalizada.

A captura de tela mostra a interface do usuário para a criação de uma nova liga. No topo, há o logótipo 'WEB RATIO' e uma barra de navegação com links para 'Sports', 'Logout', 'Leagues' (destacado), 'Teams', 'Events' e 'Bets'. Abaixo da barra, há um breadcrumb 'Leagues > Create League'. O formulário principal tem o título 'Create League' e contém um campo de texto rotulado 'Name' e um botão 'Create'.

Figura 16: Área "Leagues": Página "Create League"

Por fim, na página "Leagues' list" são visíveis todas as ligas desportivas disponíveis na base de dados, **ordenadas por nome**. A lógica do *WebRatio* é igual à secção anterior, pelo que se poupam mais explicações.



name	oid
> Campeonato de Portugal	3 Delete
> Primeira Liga	1 Delete
> Segunda Liga	2 Delete

Figura 17: Área "Leagues"– Página "Leagues' list"

4.2.3 Área Bets

A área em questão permite ao administrador controlar as apostas efetuadas pelos vários jogadores.

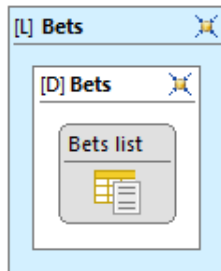


Figura 18: Lógica da área "Bets" no *WebRatio*

Deste modo, esta área é composta por uma única página, que disponibiliza ao admin uma lista personalizada com dados de apostas e utilizadores. Tal é conseguido através do Utility Component "Query".

A imagem mostra a interface de usuário da área 'Bets' no WebRatio. No topo, há o logotipo 'WEB RATIO' com ícones de cubos. Abaixo dele, uma barra de navegação contém links para 'Sports', 'Logout', 'Leagues', 'Teams', 'Events' e 'Bets' (destacado). Abaixo da barra, há uma barra de breadcrumbs: 'Bets > Bets'. O conteúdo principal é uma tabela intitulada 'Bets list'.

Bet	Event	Home Team	Away Team	Betted Amount	Expected Result	Paid	Player's username
> 1	Event nº1	SL Benfica	FC Porto	2	winHome	no	user

Figura 19: Área "Bets": Página "Bets"

4.2.4 Área Events

Esta é a área mais complexa e onde se foca a maior funcionalidade do administrador - fechar eventos. O fecho de eventos implica, numa visão de alto nível, o lançamento de notificações a todos os utilizadores que apostaram naquele evento, juntamente com a atualização dos saldos respetivos.

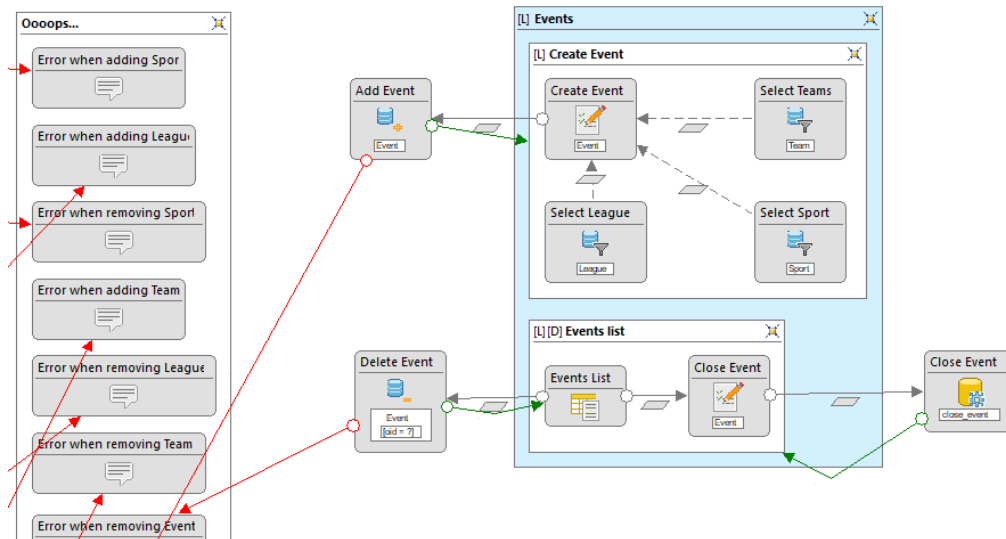



Figura 20: Lógica da área "Events" no *WebRatio*

Desta forma, existem duas páginas dentro da área: "Events' list" e "Create Event". A primeira fornece ao administrador da plataforma uma lista (ordenada) com os eventos abertos, ou seja, que ainda não terminaram. Desta lista, este pode escolher eliminar o evento ou fechá-lo. Isto é possível a partir de uma Utility Component "Query", que mostra a lista personalizada dos eventos. O botão de *delete* serve-se da Operation "Delete", e o botão de *close event* utiliza a Operation "Stored Procedure" para efetuar todas as alterações necessárias, entre elas:

- Percorrer todas as apostas;
- Somar a (quantia apostada * Odd) ao saldo de cada utilizador, caso este tenha acertado no resultado do evento;
- Lançamento de notificação ao *user*;
- Alteração do estado da aposta, que passa a estar "paga".



[Sports](#)
[Logout](#)
[Leagues](#)
[Teams](#)
[Events](#)
[Bets](#)

[Home](#) > [Events](#) > [Events list](#)

Close Event

Result


Event id

Events List

	Event Id	Home Team	Away Team	Home odd	Away odd	Draw odd	League	Sport	Status	Result	Premium	
>	1	SL Benfica	FC Porto	2	2	2	Liga Portuguesa	Futebol	Open	NA	no	Delete event Close event
>	2	FC Porto	SportingCP	5.6	1.3	5.5	Liga Portuguesa	Futebol	Open	NA	no	Delete event Close event
>	3	Rio Ave	Vitoria SC	3.4	2.4	8	Liga Portuguesa	Futebol	Open	NA	no	Delete event Close event
>	4	SCBraga	SportingCP	2.3	2	3.9	Liga Portuguesa	Futebol	Open	NA	no	Delete event Close event

Figura 21: Área "Events"– Página "Events' list"

Já a página "Create Event", com a ajuda de um formulário, permite que seja adicionada à base de dados um novo evento, através da Operation "Create". Os "Selectors" visíveis servem para o auto preenchimento dos campos de multi seleção, se que permitem escolher, por exemplo, uma das equipas já existentes.



[Sports](#)
[Logout](#)
[Leagues](#)
[Teams](#)
[Events](#)
[Bets](#)

[Home](#) > [Events](#) > [Create Event](#)

Create Event

Event League

Event Sport

Home odd

Away odd

Draw odd

Home Team

Away Team

Premium ☐ yes ☒ no

Figura 22: Área "Events"– Página "Create Event"

4.2.5 Área Teams

Esta área segue a lógica da área Sports ou Leagues, mas dedica-se à criação, eliminação e visualização de equipas.

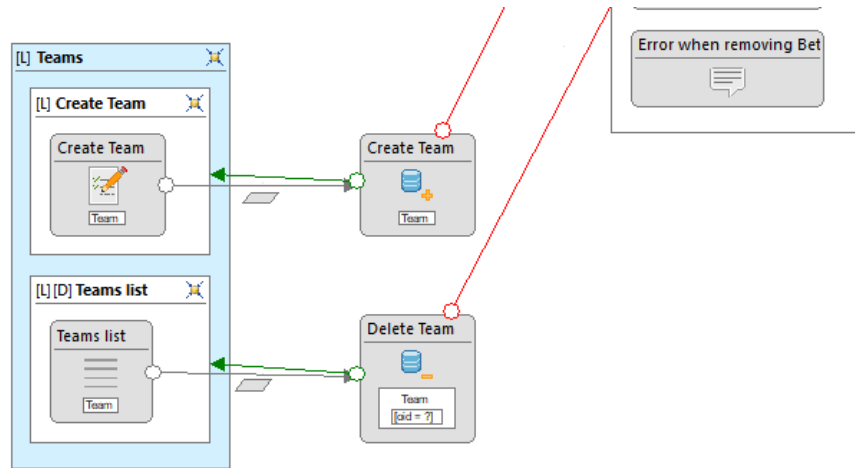


Figura 23: Lógica da área "Teams" no *WebRatio*

Assim, a primeira página "Create Team" serve-se de um formulário para que o administrador possa introduzir o nome da nova equipa. Esta é adicionada à BD, através do "Create", operação que, em caso de insucesso, mostra ao admin uma mensagem de erro personalizada. Em caso de sucesso, este permanece na mesma página.

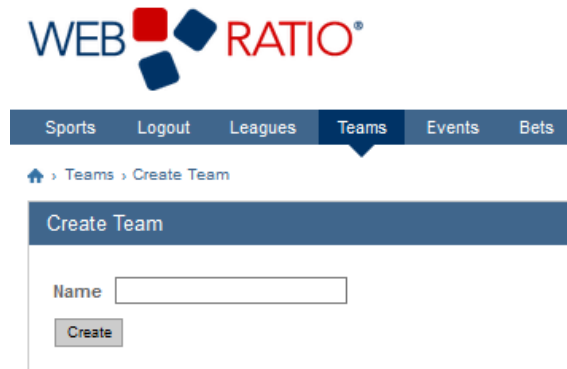


Figura 24: Área "Teams": Página "Create Team"

Por fim, a página "Teams' list" mostra todas as equipas do sistema (**ordenadas por nome**), e a opção de eliminar as mesmas. Isto é conseguido através da View Component "Simple List" e através de um botão, que efetua uma operação de "Delete". Em caso de sucesso, o admin permanece na página e, em caso contrário, é-lhe mostrada uma mensagem de erro personalizada.



Figura 25: Área "Teams" Página "Teams' list"

4.2.6 Área Logout

Esta área apenas possui o Session Component "Logout", e permite ao administrador sair da sua conta, sendo redirecionado para a *site view* "Main"(onde são efetuados os *logins* e registros).

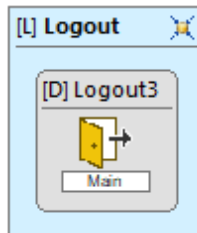


Figura 26: Lógica da área "Exit"no *WebRatio*

No *browser*, basta clicar na seguinte opção:

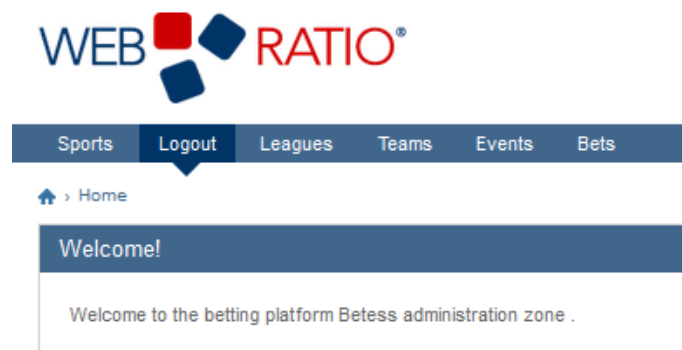


Figura 27: Área "Logout": Página "Logout"

4.3 Site View - Users

Nas próximas secções será explicada toda a lógica da site view dos *users*. Começa-se, na primeira secção, por mostrar e explicar as decisões do grupo ao longo do processo de desenvolvimento e, na "Users Premium" por explicar a evolução da *site view* "Users" para englobar ambos os tipos de utilizadores, uma vez que ambos são semelhantes. **Na verdade, o que separa *users* de *users premium* é o acesso a eventos exclusivos, obtidos através de uma quantia paga para melhoria de estatuto.**

4.3.1 User

Apresenta-se, de seguida, uma arquitetura geral da *site view* em questão, de forma a mostrar a primeira linha de raciocínio do grupo. **Esta arquitetura ainda não tem em conta o user premium, sendo que a evolução será explicada na secção seguinte, "User Premium".**

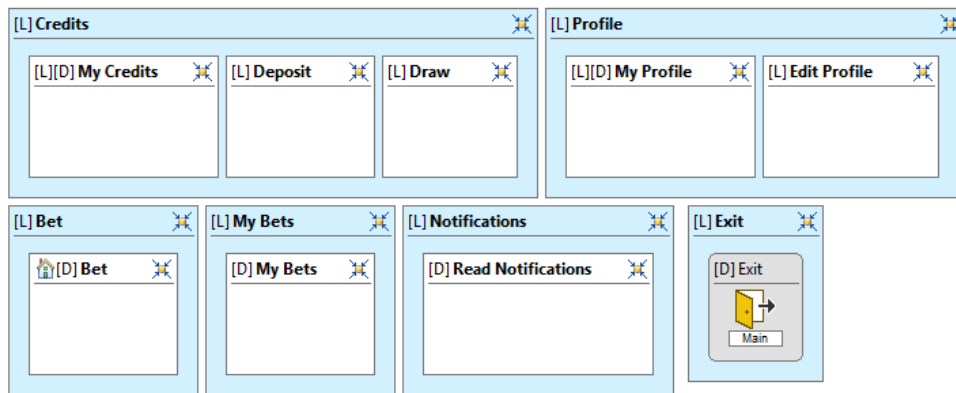


Figura 28: Arquitetura geral da *site view* do user no *WebRatio* (sem considerar user premium)

Assim, dividiu-se toda a lógica do programa em diferentes páginas, que corresponderiam às seguintes no *browser*.

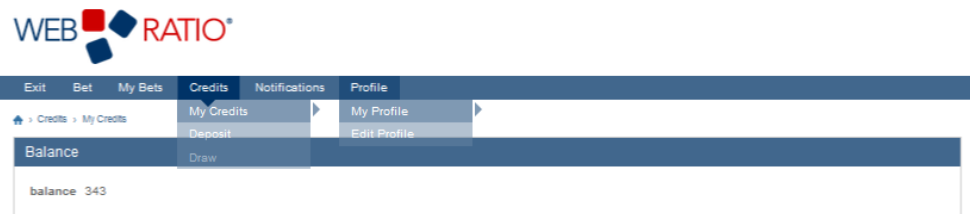


Figura 29: Imagem do *browser* correspondente à arquitetura no *WebRatio* (sem considerar user premium)

Nas próximas secções será detalhada cada área criada, juntamente com uma explicação de toda a lógica interna das mesmas.

4.3.1.1 Área Credits Esta área é reservada para que o utilizador possa gerir as suas despesas, ou seja, verificar os créditos que possui, assim como carregar ou levantar dinheiro. **É importante notar que se assume que o dinheiro é extraído diretamente da conta bancária do utilizador, e levantado para a mesma, sendo que a complexidade extra não é tida em conta.**

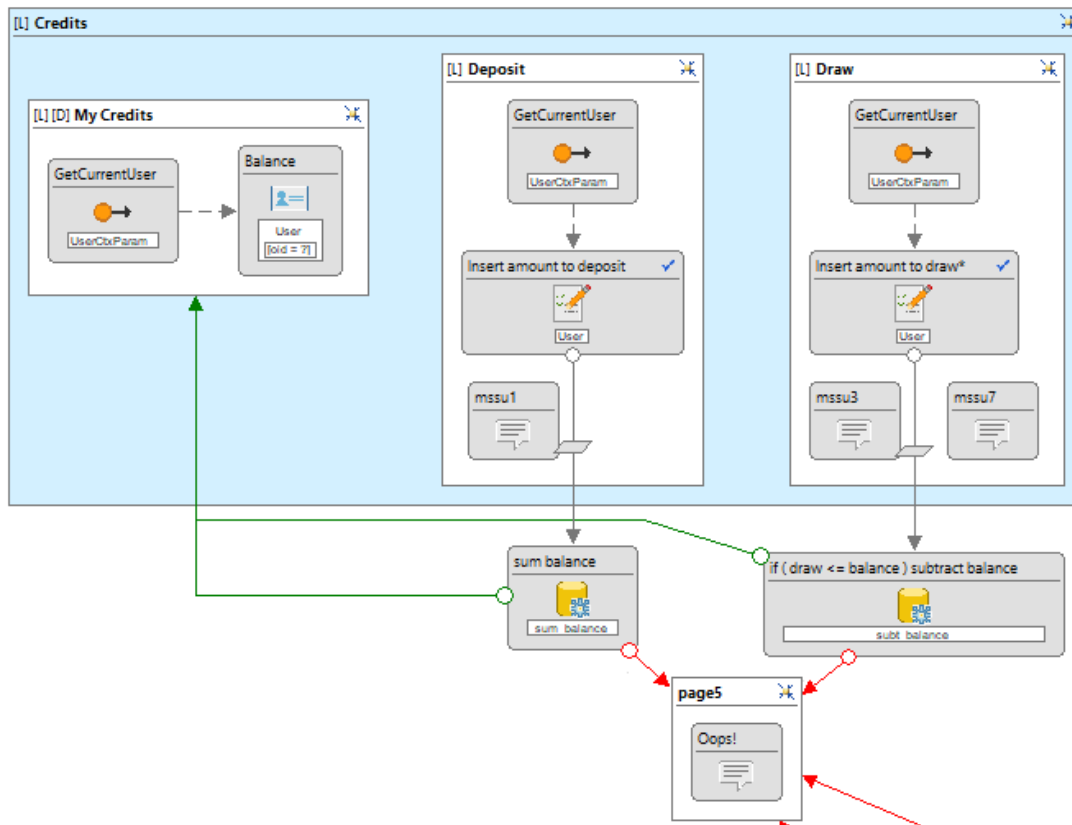


Figura 30: Lógica da área "Credits" no WebRatio

Desta maneira, observam-se três páginas: "My Credits", "Deposit" e "Draw". A página "My Credits" serve para o utilizador verificar os créditos que possui. Esta não é a única que serve para tal, uma vez que, na área de perfil, o utilizador poderá consultar todos os seus dados, inclusive o seu saldo. Ainda assim, através da View Component "Details" e do Session Component "Get", rapidamente se definiu esta página simples.

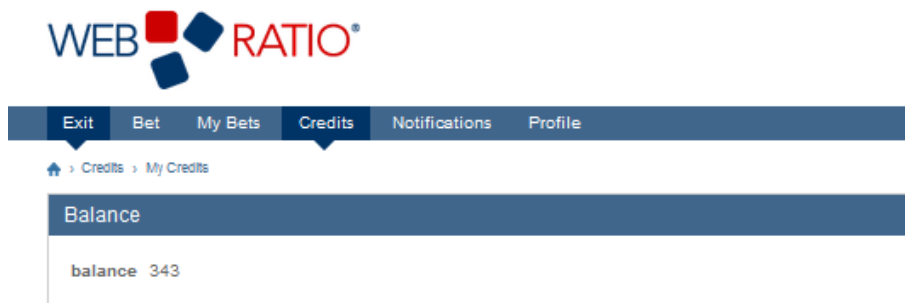


Figura 31: Área "Credits" - Página "My Credits"

A segunda página tem o intuito de levar o utilizador a carregar créditos, para assim poder jogar. Desta forma, foi necessário pedir ao utilizador a quantia que ele deseja carregar, avisando-o que tal será retirada da sua conta bancária para a plataforma, através de um "Form". Com isto, e com o id do *user* logado, utilizou-se um "Stored Procedure" com o intuito de somar a quantia a depositar ao seu saldo precedente. Em caso de sucesso, o utilizador é redirecionado à página "Credits", e em caso de insucesso, é-lhe mostrada uma mensagem de erro.

(É de notar que o Session Component "Get" poderia estar diretamente ligado ao stored procedure.)

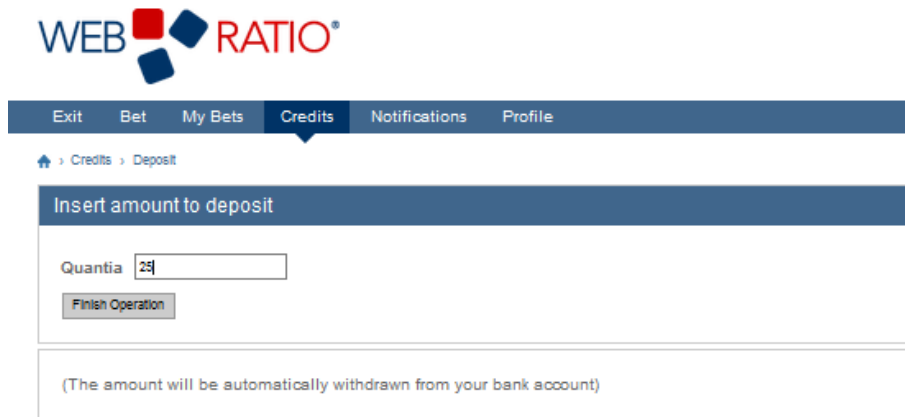


Figura 32: Área "Credits"– Página "Deposit"

Por fim, a página "Draw" segue a mesma lógica que a "Deposit", com a exceção do "Stored Procedure". Neste, é adicionalmente verificado se o utilizador possui os créditos que deseja extrair da plataforma para a conta bancária, e só em caso positivo é que tal acontece.

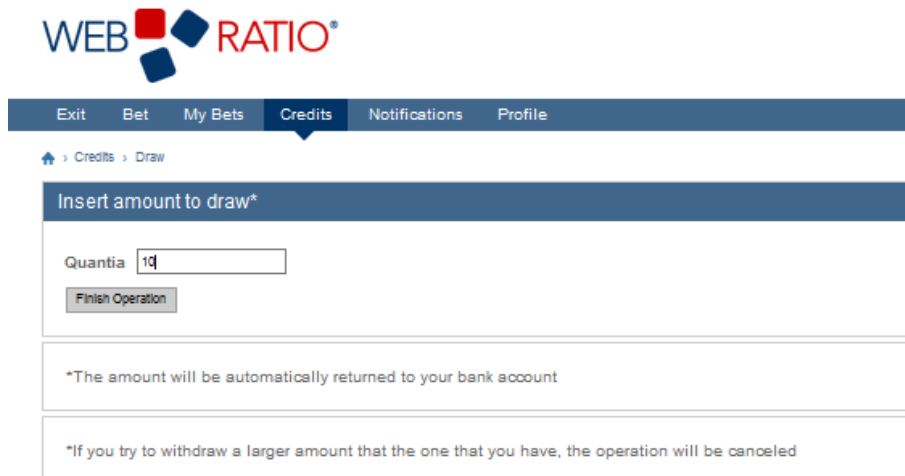


Figura 33: Área "Credits"– Página "Draw"

4.3.1.2 Área Profile Esta área é dedicada à visualização e edição do perfil do utilizador, típico de qualquer *website*.

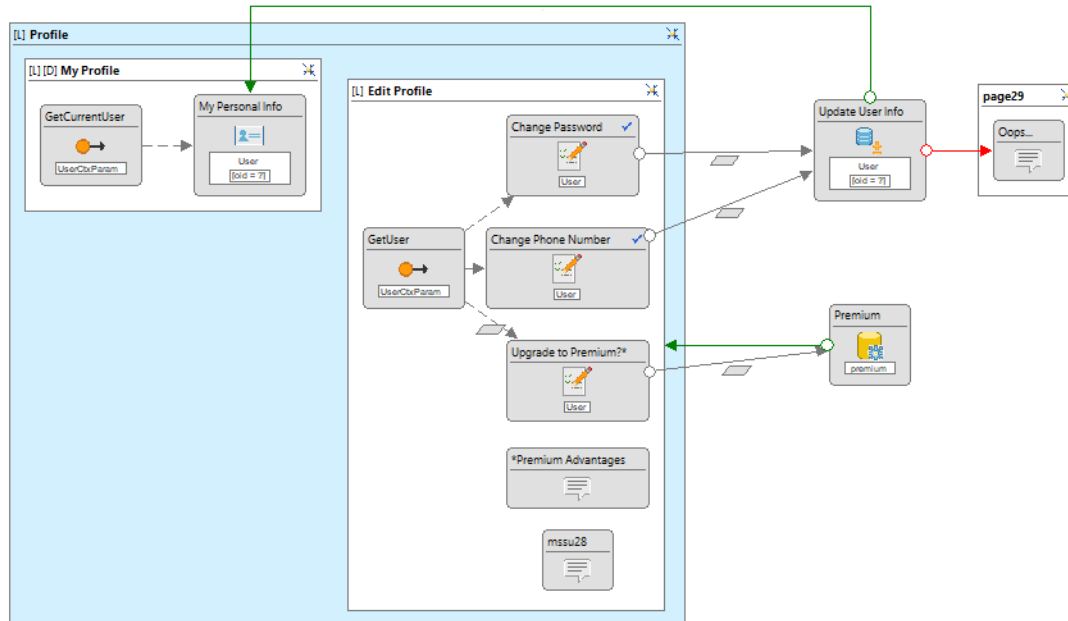


Figura 34: Lógica da área "Profile" no *WebRatio*

Logicamente, está dividida em duas páginas: "My Profile", que fornece ao *user* os seus dados atuais e "Edit Profile", que permite ao mesmo alterar algumas das suas informações. Assim, a primeira página mencionada elaborou-se com um simples "Details" e "Get", que capta o identificador do *user* logado. Este id permite que se extraia e mostre toda a informação do utilizador.



Figura 35: Área "Profile": Página "My Profile"

Já a página "Edit Profile" apenas concede ao *user* a possibilidade de alterar o seu número de telefone ou a password, uma vez que dados como o username, por exemplo, não devem ser modificados. Para além disso, é também aqui que o mesmo pode efetuar o *upgrade* para um utilizador *premium*. Posto isto, bastou utilizar a Operation "Update", que se serve do id do utilizador e dos dados introduzidos referentes à mudança para atualizar a base de dados. Em caso de sucesso, o *user* é redirecionado para a página "My Profile" e, em caso contrário, depara-se com uma mensagem de erro. No caso da secção "Upgrade to Premium?" bastou utilizar um Form sem campos (que permite simular um botão), juntamente com uma mensagem de aviso ao utilizador para que ele tenha noção do custo envolvente da operação.

Figura 36: Área "Profile": Página "Edit Profile"

Chama-se à atenção para a *password strength* necessária para modificar o campo "New Password".

Figura 37: Área "Profile": Página "Edit Profile- Força da password"

4.3.1.3 Área Bet A área em questão é uma das mais importantes, uma vez que reúne o foco da plataforma de apostas.

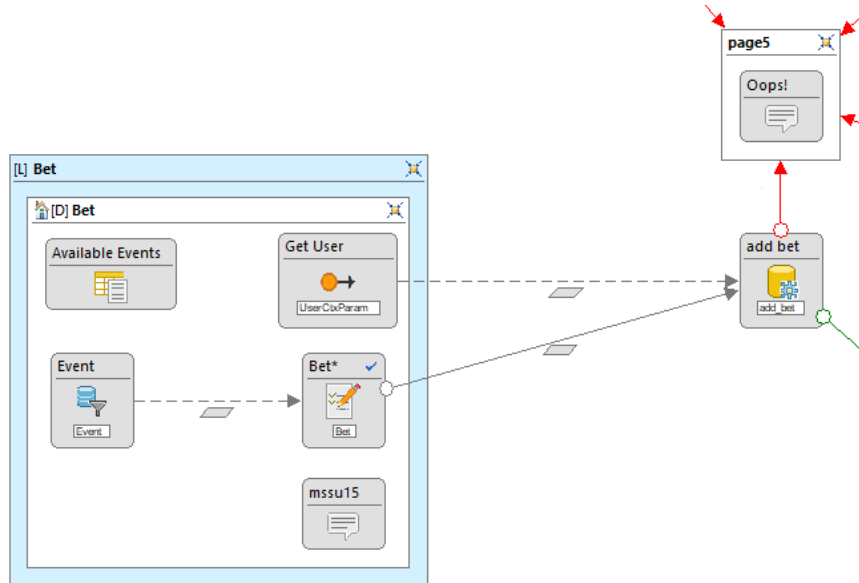


Figura 38: Lógica da área "Bet" no WebRatio

Assim, esta área é composta por uma única página, que disponibiliza ao *user* uma **lista de eventos disponíveis (apenas, uma vez que apostar num evento passado não seria possível)**, através do Utility Component "Query", juntamente com um formulário que lhe questiona acerca da quantia a apostar, resultado pretendido (só há três possíveis, *win home*, *win away* e *draw*), e o número do evento que deseja. No fim, através da Operation "Stored Procedure", e com auxílio do Session Component "Get", adiciona-se à base de dados a aposta, **apenas** se o utilizador tiver saldo suficiente para apostar. Em caso de insucesso, é-lhe mostrada uma mensagem de erro.

Evento	Home Team	Away Team	Odd Home	Odd Away	Odd Draw	League	Sport
Event nº1	FCPorto	SportingFC	3.75	10	4	Primeira Liga	Futebol
Event nº2	SportingFC	SCBraga	8	5.7	2.3	Primeira Liga	Futebol
Event nº3	VitoriaSC	Rio Ave	4.9	6	2.9	Primeira Liga	Futebol
Event nº4	FCPorto	SLBenfica	2	1.4	5	Primeira Liga	Futebol

Figura 39: Área "Bet": Página "Bet"

4.3.1.4 Área My Bets A presente área serve o propósito de o utilizador poder controlar as suas apostas (enquanto o evento não é fechado, i.e., não acaba).

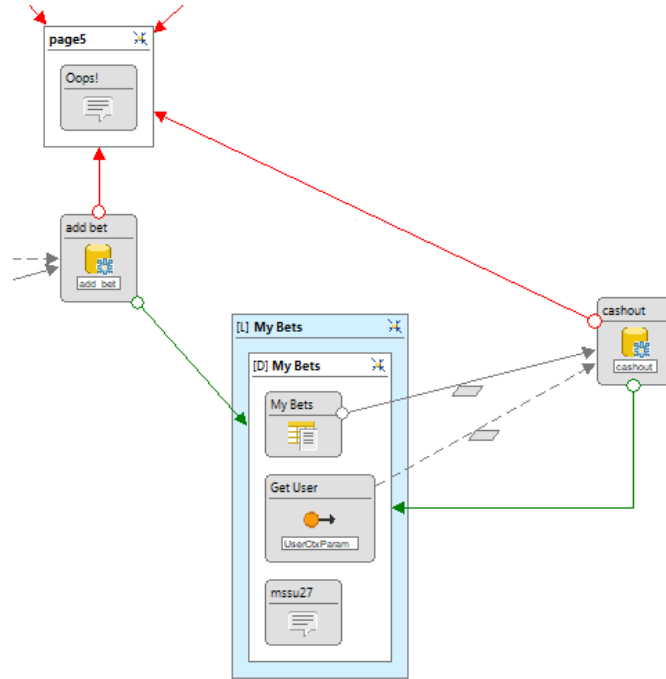


Figura 40: Lógica da área "My Bets" no *WebRatio*

Desta forma, esta área possui apenas uma página, que mostra ao *user* as suas apostas, através do Utility Component "Query". É permitido que este efetue o *cashout* - **apenas de apostas cujos eventos ainda estejam abertos** -, sendo que o mesmo é avisado que, se desistir da aposta, apenas receberá metade da quantia apostada, através de uma mensagem. Se o utilizador quiser prosseguir com o *cashout*, é utilizado um "Stored Procedure" que, para além de alterar o estado da aposta (retirando-a da sua lista de apostas), lhe atualiza o saldo. Em caso de insucesso, é-lhe mostrada uma mensagem de erro.

WEB RATIO®

Exit Bet **My Bets** Credits Notifications Profile

My Bets > My Bets

My Bets

Bet	Event	Home Team	Away Team	Betted Amount	Expected Result	
> 1	Event n°1	FCPorto	SportingFC	50	winAway	Cashout
> 2	Event n°2	SportingFC	SCBraga	30	winHome	Cashout
> 3	Event n°3	VitoriaSC	Rio Ave	15	draw	Cashout

*(If you try to cashout, you'll only receive half of your betted amount)

Figura 41: Área "My Bets" - Página "My Bets"

4.3.1.5 Área Notifications Esta área é dedicada às notificações do utilizador, que as recebe sempre que um evento termina. São-lhe fornecidos dados como o novo saldo, o resultado do evento (que equipa ganhou), entre outros.

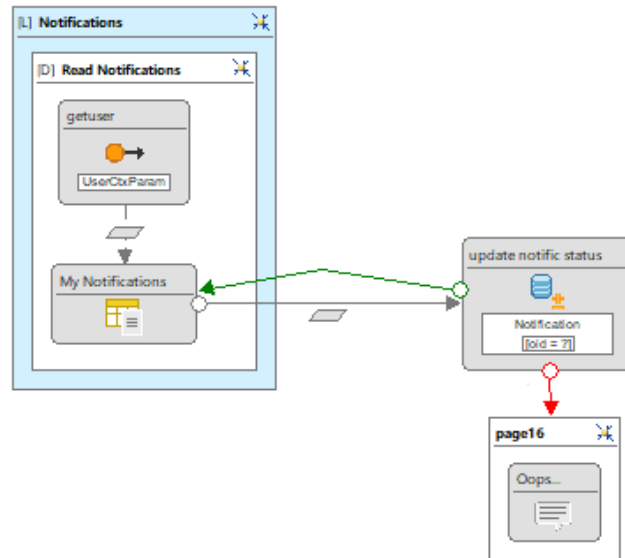


Figura 42: Lógica da área "Notifications" no *WebRatio*

A área mencionada tem apenas uma página com o Utility Component "Query", que mostra ao *user* (obtido através do Session Component "Get") uma lista personalizada com notificações, que ele pode marcar como lidas. Se ele escolher marcar as notificações, a base de dados é atualizada no respetivo campo, através da Operation "Update". Em caso de erro, é mostrada uma mensagem ao utilizador e, em caso de sucesso, mantém-se na mesma página, que atualiza. **Estas notificações pertencem a eventos passados, i.e., que já foram fechados e para os quais já se conhece o resultado.**

WEB  RATIO®

[Exit](#) [Bet](#) [My Bets](#) [Credits](#) [Notifications](#) [Profile](#)

[Home](#) > [Notifications](#) > [Read Notifications](#)

My Notifications

	Bet	Home Team	Away Team	Event	Result	Balance	Status	
➤ 1	FCPorto	SportingFC	1	win home	300	seen	Mark as 'seen'	
➤ 3	VitoriaSC	Rio Ave	3	draw	250	not seen	Mark as 'seen'	
➤ 2	SportingFC	SCBraga	2	draw	400	not seen	Mark as 'seen'	

Figura 43: Área "Notifications": Página "Read Notifications"

4.3.1.6 Área Exit Esta área apenas possui o Session Component "Logout", e permite ao utilizador sair da sua conta, sendo redirecionado para a *site view* "Main"(onde são efetuados os *logins* e registos).

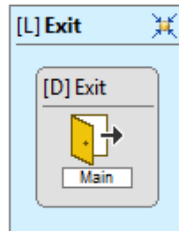


Figura 44: Lógica da área "Exit"no *WebRatio*

No *browser*, basta clicar na seguinte opção:

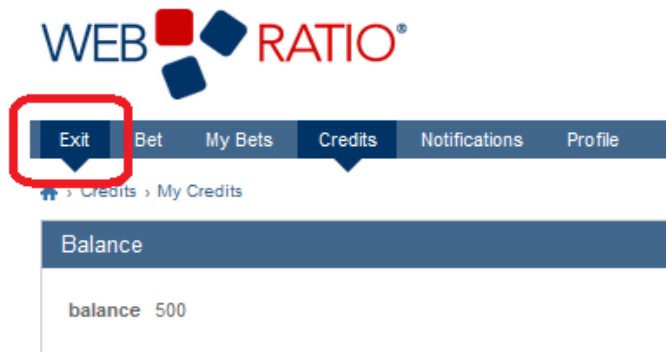


Figura 45: Área "Exit": Página "Exit"

4.3.2 User Premium

Um utilizador *premium* tem como **única** vantagem o acesso a eventos exclusivos e, consequentemente, mais oportunidades para apostar. Qualquer utilizador se pode tornar premium, desde que pague a modesta quantia de cinquenta ESScoins. Este utilizador, por ter quase tudo em comum com um *user* normal, utilizará a mesma *site view* do user, apenas sendo-lhe atribuída uma área reservada para visualização de eventos *premium*.

Com isto, mostra-se, finalmente, a arquitetura completa da *site view* "Users", que conta com mais uma área já mencionada.

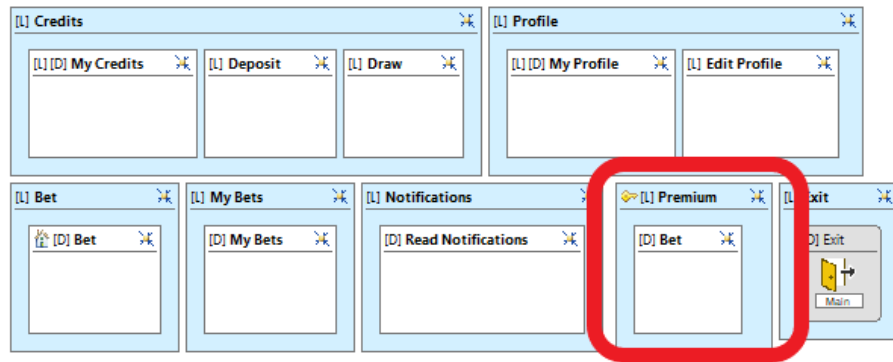


Figura 46: Arquitetura geral da *site view* do user no *WebRatio*, considerando o user *premium*

Para conceder acesso a esta área, o grupo criou um novo *group* de utilizadores, os *premium*. Estes utilizadores, para além de verem todas as páginas que os outros vêem, conseguem ver a área em questão, através de um novo *module* que a referencia. Assim, esta página é exatamente igual à anterior, mas o Utility Component "Query" mostra uma lista de eventos personalizados reservados. Tal como já explicado na secção anterior, o utilizador pode escolher um dos eventos para apostar (através de um "Form"), e a aposta será registada na base de dados, com o auxílio da Operation "Stored Procedure", que adiciona a aposta apenas se o *user* tiver saldo para tal. Em caso de sucesso, este é redirecionado para a área "My Bets" e, em caso contrário, é-lhe mostrada uma mensagem de erro personalizada.

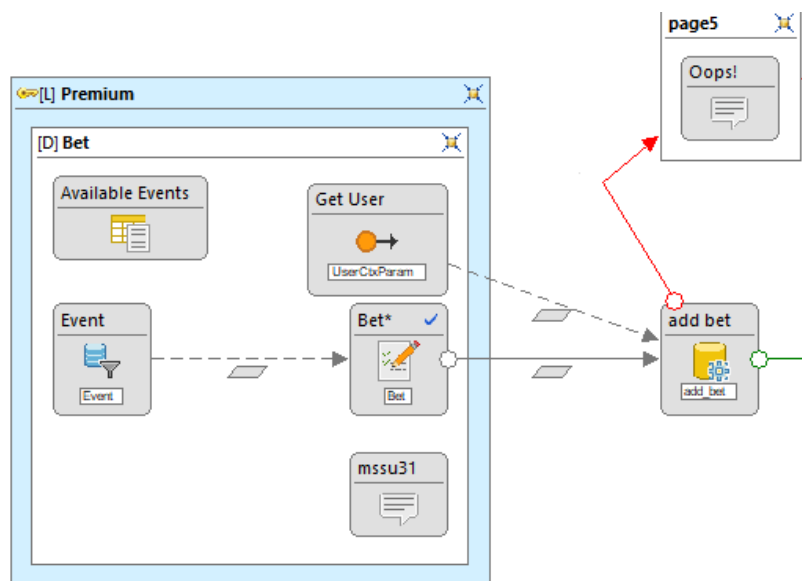


Figura 47: Lógica da área "Bet" no *WebRatio*

5 Conclusões e Sugestões

A resolução deste trabalho prático foi bastante importante e enriquecedora, pois permitiu aos membros do grupo perceber e interiorizar melhor os conceitos abordados nas aulas da Unidade Curricular de Engenharia Web, para além de que serve de uma boa introdução para a segunda fase do projeto.

Deste modo, foram aprofundados conhecimentos relativos à linguagem IFML e à ferramenta *WebRatio*. O grupo alcançou uma maior consciência acerca da importância da abstração e modulação de um problema, por forma a criar soluções simples e melhorar as interações com o utilizador. Tarefas como ligações manuais a uma base de dados, composição de páginas ou integração de serviços foram simplificadas pela agilidade e rapidez que a ferramenta fornece.

Assim, no que diz respeito à elaboração do *domain model*, o grupo conseguiu, com bastante facilidade, a sua elaboração, dado o seu background em UC's como Bases de Dados. Ainda assim, a alteração e evolução deste modelo foi inevitável, uma vez que, com o avançar do projeto, a equipa apercebeu-se que para a implementação de determinadas funcionalidades, uma alteração no modelo de dados seria a estratégia a adotar.

No que toca à parte do trabalho onde o grupo despendeu mais tempo - as *site views* -, é relevante mencionar a acentuada curva de aprendizagem, uma vez que o grupo teve que aprender uma nova linguagem de modelação, a *Interaction Flow Modeling Language*, sendo esta utilizada na modelação da interação do *user* com o sistema. A pouca documentação disponível, ou a simplicidade dos exemplos disponíveis, dificultaram também o trabalho do grupo, que se deparou com vários impasses. Após ultrapassados os primeiros obstáculos, o restante desenvolvimento decorreu sem problemas.

É ainda importante mencionar a quantidade de cenários de inconsistência da ferramenta *WebRatio* no que toca aos resultados apresentados, o que atrasou bastante o trabalho da equipa inicialmente, para além das dificuldades de *merge* do projeto.

Em suma, é feita uma apreciação positiva relativamente ao trabalho realizado, visto que a implementação de todas as funcionalidades propostas foram conseguidas com sucesso. O grupo conseguiu tirar partido dos conhecimentos adquiridos neste projeto, sentido-se capaz de, num contexto futuro, aplicar os conceitos subjacentes de forma eficaz. É evidente que, num outro contexto (como um projeto de grandes dimensões), seria benéfico que fossem implementadas um maior conjunto de funcionalidades adicionais para uma melhor concretização do sistema.