

UNIVERSIDADE DO MINHO



ENGENHARIA WEB

ENGENHARIA DE SISTEMAS DE SOFTWARE

Betting House Web App - Parte 2

Autores:

Manuel Sousa

Tiago Alves

Lázaro Lima

Identificação:

A78869

A78218

ID8120

16 de Junho de 2019

Conteúdo

1	Introdução	2
1.1	Requisitos Não-Funcionais	3
2	Arquitetura do Sistema	4
2.1	Modelo de Dados	5
2.1.1	Utilizador	6
2.1.2	Aposta	6
2.1.3	Evento	6
2.1.4	Resultado	7
2.1.5	Administrador	7
2.2	Vue.js: Presentation Layer	8
2.3	Express.js/Node.js: Data Access Layer	9
3	Conclusões e Trabalho Futuro	10

1 Introdução

O sistema desenvolvido de apostas online baseia-se nos ganhos e nas perdas que são obtidos com as apostas, sendo o mais comum as apostas desportivas. À primeira vista, a adopção de um sistema de apostas possibilita poupar e rentabilizar o dinheiro do apostador que visa a maximização dos lucros das suas apostas desportivas.

Para esta segunda etapa, o objetivo era o de construir tal plataforma, capaz de controlar toda esta lógica de ganhos e perdas. Para construir tal plataforma, foram idealizadas as seguintes tecnologias: *Express.js* para backend; *Vue.js* para frontend; *MySQL* como base de dados relacional e, se possível, *Docker*, para que o sistema fosse expansível. Nas próximas seções serão apresentadas as justificações para a escolha de tais tecnologias.

O foco do desenvolvimento ficou na criação de uma solução responsiva, reativa, acessível, fácil de usar e ainda com segurança. Com a primeira fase desenvolvida, o relatório então apresenta a evolução e correção da primeira parte do trabalho e a criação de uma aplicação do lado do cliente e do lado do servidor.

Requisitos

A aplicação desenvolvida teve em conta utilizadores classificados em três tipos de perfil, são eles:

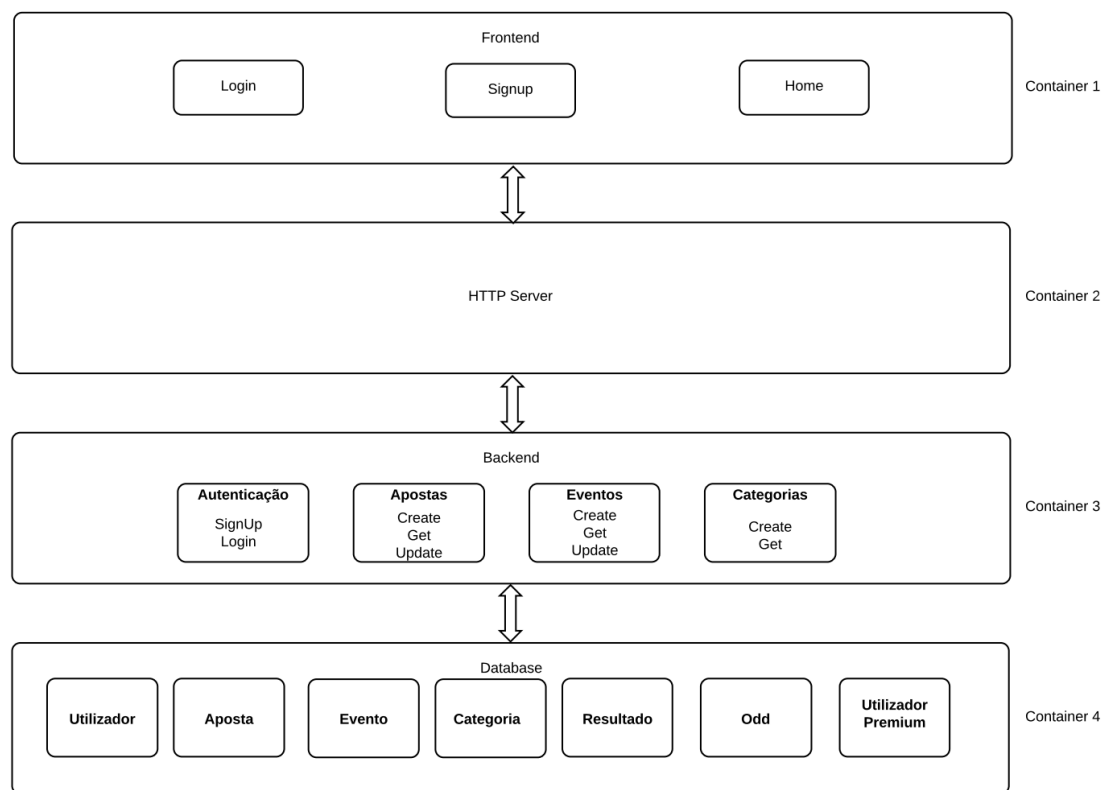
- **Administrador:** perfil responsável pela gestão da informação na aplicação, como os eventos, utilizadores, apostas, etc. Deverá ter ao seu dispor uma interface para que possa cumprir com os seus deveres.
- **Utilizador Premium:** paga uma quantia que lhe dá acesso a informações extra de eventos.
- **Utilizador Regular:** este perfil representa os apostadores regulares.

1.1 Requisitos Não-Funcionais

- **Real-time responsiveness:** a aplicação lida com apostas em eventos, logo apostas em eventos que já terminaram não devem ser permitidas.
- **Escalabilidade:** supostamente a aplicação deve poder aguentar com grandes quantidades de pedidos em certos eventos, como por exemplo, Liga dos Campeões, Super Bowl, etc., e deve responder quase instantaneamente.
- **Interface:** deve seguir pelo menos o nível A do WCAG (Web Content Accessibility Guidelines)

2 Arquitetura do Sistema

Depois da fase de modelação foi decidido, dentro de uma panóplia de ferramentas, adotar aquelas que são enunciadas neste próximo esquema. O objetivo era desenvolver um sistema que viesse ter ao encontro daquilo que foi modelado. Por esta razão e para entender melhor o sistema como um todo foi feito o desenho da arquitetura, o qual pode ser visualizado na Figura 1.



Containers - Docker

Frontend - Vue.js (Progressive JavaScript Framework)

Backend - Node.js/JavaScript + Express (Node.js web application framework)

Database - MySQL

Figura 1: Arquitetura do Sistema

2.1 Modelo de Dados

Tendo em conta os requisitos do sistema Betting House mencionados anteriormente, foi elaborado o seguinte modelo de dados visto na Figura 2:

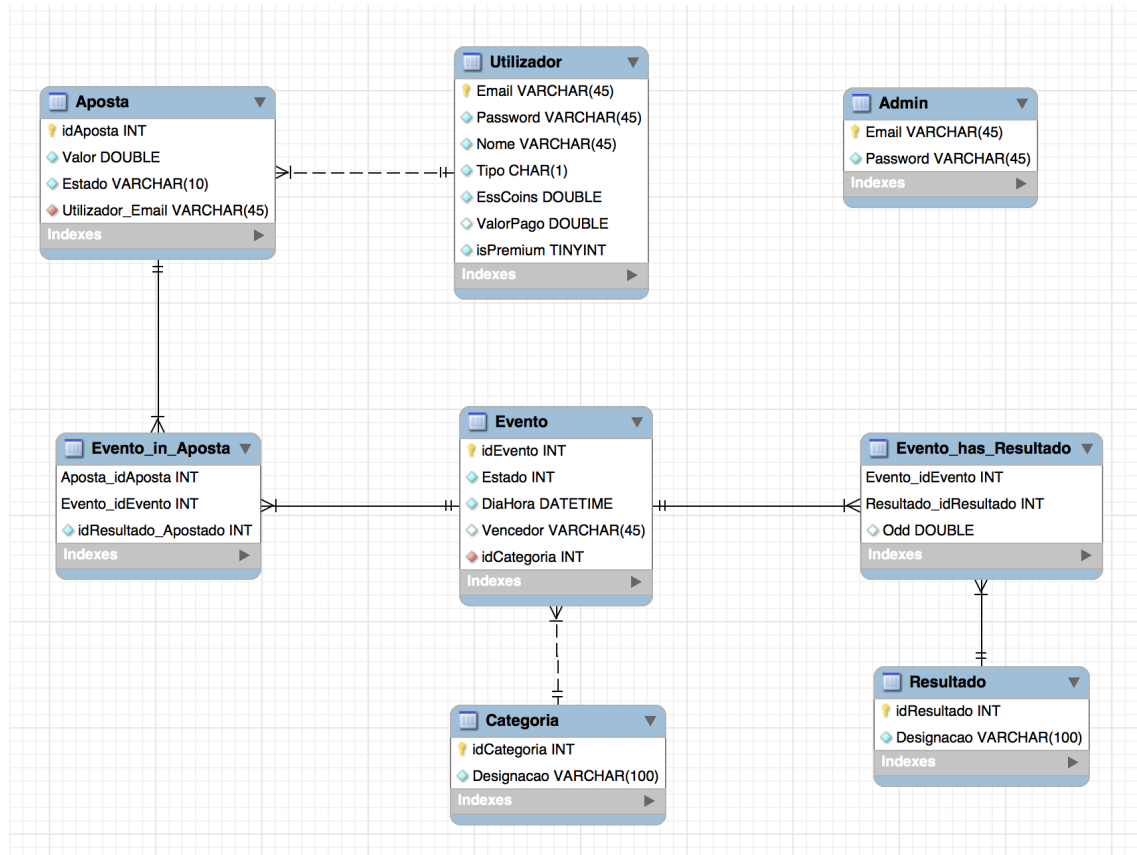


Figura 2: Base de dados do sistema Betting House

O sistema de gestão de base de dados (SGBD) escolhido para construir a solução aqui proposta foi o MySQL [1], o qual é, hoje em dia, um dos motores de base de dados mais populares do mundo.



As características que se destacam no MySQL são a portabilidade, suportando as plataformas atualmente, compatibilidade com diversas linguagens de programação, não exigindo muitos recursos da máquina para funcionar. O MySQL é um software

livre com base na GPL se o sistema também for GPL, possui controlo de transações, suportando *Triggers*, *Cursors*, *Stored Procedures* e *Functions*, para além de uma replicação de fácil configuração.

2.1.1 Utilizador

Os utilizadores que constituem esta aplicação são o *Utilizador Normal* e o *Utilizador Premium*. Cada um deles precisa de se registar no sistema fornecendo o email e password. Além disso, terão também um nome, e as *Ess Coins* associadas a cada um deles.

Estes são os campos comuns de ambos os utilizadores, logo gerou-se uma tabela ***Utilizador***, com o intuito de armazenar esta informação.

2.1.2 Aposta

Uma aposta estará associada a apenas um utilizador, visto esta ser única para com o mesmo. No entanto, este poderá ter várias apostas. Nesta, é guardado o valor que foi apostado pelo utilizador, e controlado o estado em que se encontra. Por consequência da ligação com o utilizador, esta contém como chave estrangeira o seu email.

2.1.3 Evento

Um Evento é aquele pelo qual o utilizador submete uma aposta. A este, estão associadas várias tabelas, como a tabela Aposta, de forma a saber quais são os eventos pertencentes à aposta, a tabela Resultado, de forma a saber quais os participantes do presente Evento, e ainda a tabela Categoria, para associar um tipo de categoria, Futebol, Andebol, Corridas de Cavalos, etc., a um Evento. A este, aquando altura de seu fecho, é associado o nome do Vencedor.

2.1.4 Resultado

Um Resultado é uma entidade que tem a possibilidade de ser o “Vencedor” de um Evento. Nesta tabela são incluídos os nomes de todos os participantes e também um estado de “EMPATE”, o qual é necessário para diversos tipos de categorias.

2.1.5 Administrador

Como responsável pela gestão da plataforma, como adicionar eventos, adicionar categorias, etc., existe o utilizador ***Admin***, que se autentica no sistema com o email e password.



2.2 Vue.js: Presentation Layer

O responsável em ligar as entradas realizadas pelos utilizadores foi o Vue.js [2]. Este é um framework progressivo de código aberto em JavaScript para a construção de interfaces, o qual se foca na renderização declarativa e na composição dos componentes. Há várias bibliotecas fornecidas e mantidas oficialmente que podem controlar, num sistema, o roteamento, a administração de estado e ferramentas de construção.

A seguir temos alguns exemplos da interface criada, o menu de início de sessão, o menu de criação de conta e o menu principal do utilizador, nas Figuras 4, 5 e 3, respetivamente.

Categorias

- Andebol
- Basebol
- Basquetebol
- Futebol
- Hóquei
- Rugby
- Tenis

Eventos

Evento 2 - 01/10/2019 22:30:05

Index	Resultado	Odd	Apostar	Info Extra
1	EMPATE	1.4	Apostar!	Apenas Premium!
2	Desportivo das Aves	3.7	Apostar!	Apenas Premium!
3	Benfica	5.6	Apostar!	Apenas Premium!

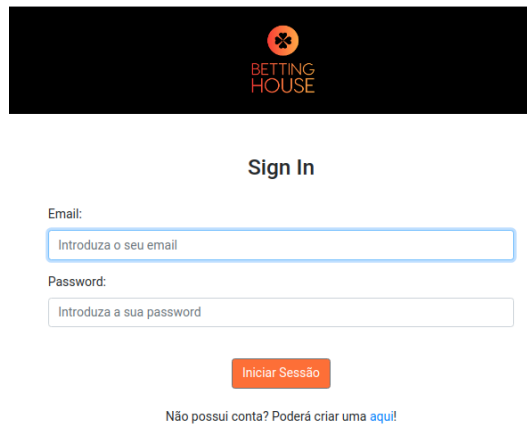
Evento 4 - 01/05/2021 22:30:05

Index	Resultado	Odd	Apostar	Info Extra
1	EMPATE	5	Apostar!	Apenas Premium!
2	Benfica	3.7	Apostar!	Apenas Premium!
3	Porto	2.1	Apostar!	Apenas Premium!

Boletim

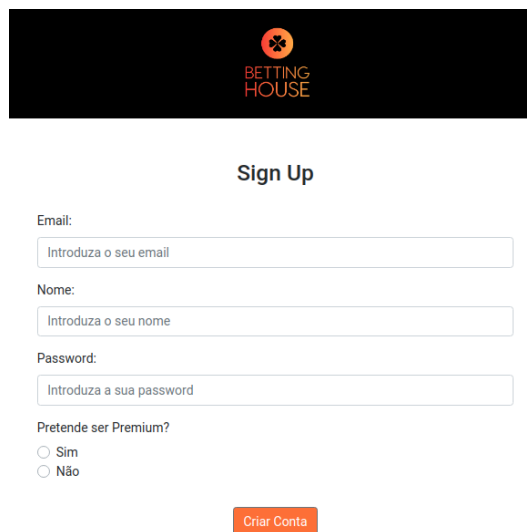
Resultado	Odd	Evento	Remover
Montante a apostar:			
<input type="text" value="0"/>		<button>Submeter!</button>	
Ganhos Possíveis: 0			

Figura 3: Main



The Sign In form features a dark header with the Betting House logo. Below the header, the title "Sign In" is centered. The form includes two input fields: "Email:" with a placeholder "Introduza o seu email" and "Password:" with a placeholder "Introduza a sua password". A red "Iniciar Sessão" button is positioned below the password field. At the bottom, a link "Não possui conta? Poderá criar uma [aqui!](#)" is displayed.

Figura 4: Sign In



The Sign Up form features a dark header with the Betting House logo. Below the header, the title "Sign Up" is centered. The form includes three input fields: "Email:" with a placeholder "Introduza o seu email", "Nome:" with a placeholder "Introduza o seu nome", and "Password:" with a placeholder "Introduza a sua password". Below the password field, there is a section for "Pretende ser Premium?" with two radio buttons labeled "Sim" and "Não". A red "Criar Conta" button is positioned at the bottom.

Figura 5: Sign Up

2.3 Express.js/Node.js: Data Access Layer

As tecnologias utilizadas para o desenvolvimento da aplicação são *Node.js*, desenvolvido para aplicações escaláveis e a camada de acesso aos dados foi construída com o *Express.js* [3], uma *framework* de código-aberto sobe a licença MIT, que é projetada para construir aplicações *Web's* e *API's*.

Através deste servidor, é possível a fácil obtenção dos dados para posteriormente serem processados pelo *frontend*. De notar que o *Express.js* faz uso do padrão MVC, adicionar uma nova funcionalidade (ou uma nova rota) torna-se muito acessível.

3 Conclusões e Trabalho Futuro

A solução proposta pelo grupo para a implementação do sistema *Betting House* foi pensada de tal forma a que um utilizador da aplicação possa realizar apostas em diversos tipos de eventos, ou seja, incluem-se categorias neste sistema, como por exemplo, Futebol, Corridas de Cavalos, Atletismo, etc. O modelo de dados apresentado nos tópicos acima baseia-se exatamente nesta ideia generalizada de apostas.

Foram apresentadas as tecnologias escolhidas, bem como as justificação para a escolha de tal, visto serem tecnologias de código-livre, fácil manutenção de um futuro sistema, ótimo apoio da comunidade, entre outras características.

A aplicação cliente desenvolvida é reativa, responsável, acessível e com boa usabilidade, criando assim um artefato que pode ser aplicado ao mundo das apostas virtuais. No final da segunda etapa do projeto, ainda não foi possível implementar a tecnologia *Docker* como enunciado previamente. Grande parte do nosso tempo foi dedicado às restantes ferramentas, visto que eram estas que nos permitiam construir o sistema pretendido, deixando, evidentemente, o *deployment* para o final. Contudo como trabalho futuro a implementação de tal tecnologia proporcionará uma melhor economia de recursos e uma maior disponibilidade.

Referências

- [1] André Milani. *MySQL-guia do programador*. Novatec Editora, 2007.
- [2] Olga Filipova. *Learning Vue. js 2*. Packt Publishing Ltd, 2016.
- [3] Azat Mardan. *Express. js Guide: The Comprehensive Book on Express. js*. Azat Mardan, 2014.