



Programação para a Web II - 2019/2 Trabalho Prático

Introdução

O trabalho prático da disciplina de Programação Web II será a implementação de uma aplicação para partidas de xadrez. A aplicação deverá ser desenvolvida em node.js, express, socket.io, handlebars e sequelize.



Figura 1: As peças de um jogo de xadrez

Sobre o jogo de Xadrez

O xadrez é um jogo de tabuleiro, de caráter competitivo, disputado entre dois participantes. Cada um é representado por peças de cores opostas, geralmente são utilizadas pretas e brancas. O objetivo do jogo é conquistar o “rei” de seu adversário.

Para jogar é necessário um tabuleiro composto por oito colunas e oito linhas, o que resulta em 64 casas possíveis para a mobilidade das peças. As peças são compostas de oito peões, duas torres, dois cavalos, dois bispos, uma rainha e um rei.

Cada peça tem sua particularidade no modo de movimentar-se sobre o tabuleiro. Ao peão, são apenas permitidos movimentos frontais, de modo que o primeiro movimento de cada peão pode abranger duas casas, os outros movimentos se restringem a uma casa à frente. Embora se movimente para frente, o ataque do peão sempre ocorre na diagonal. A torre pode correr, sem restrições de número de casas, para frente/trás/direita/esquerda. O cavalo realiza movimentos em “L” (duas casas em um sentido e uma casa em sentido perpendicular àquele), para qualquer direção. O movimento do bispo ocorre, assim como no caso das torres, sem limitação de casas, porém apenas no sentido diagonal. A rainha tem livre movimentação no jogo. O rei pode apenas ser movimentado de casa em casa, ainda que em qualquer direção do tabuleiro.

A movimentação das peças, por parte dos jogadores, é feita a partir de estratégia bastante pensada. É por isso que se costuma usar o xadrez como analogia para quaisquer outras ações de estratégias, como as ações políticas, por exemplo.

Aplicação de exemplo

O endereço <https://xadrezapp.herokuapp.com> contém uma aplicação completa que atende todas as especificações deste trabalho. Essa aplicação deverá ser usada como base para o desenvolvimento deste trabalho prático.

As bibliotecas chessboard.js e chess.js

A aplicação deverá ser desenvolvida através das bibliotecas **chessboardjs** e **chess.js**. A chessboard.js é útil para criar o tabuleiro de xadrez com as peças brancas e pretas na página do usuário, e define uma série de métodos e callbacks que serão muito úteis no desenvolvimento da aplicação.

Embora a biblioteca chessboard.js defina métodos para movimentação das peças entre as casas do tabuleiro, essa biblioteca não é capaz de avaliar se o movimento requisitado pelo usuário é válido ou não. A chessboard.js também não é capaz de dizer se um competidor está em check ou checkmate. Para isso será usado a biblioteca chess.js, que é uma biblioteca para validação de movimentos, e detecção de checks e checkmates. Abaixo, segue alguns links úteis sobre tais bibliotecas:

Biblioteca chessboardjs:

- Repositório NPM: <https://www.npmjs.com/package/@chrisoakman/chessboardjs>
- Site oficial: <https://chessboardjs.com/>

Biblioteca chess.js:

- Repositório NPM: <https://www.npmjs.com/package/chess.js>
- Github: <https://github.com/jhlywa/chess.js>

Integração das bibliotecas chessboard e chess.js:

- Documentação: <https://chessboardjs.com/examples#5000>

Para ajudá-los na implementação, foi disponibilizado um protótipo de integração do node/express com as bibliotecas chessboard e chess.js, que está disponível em <https://github.com/dbfernandes/SkelChess>.

Regras do Trabalho Prático

A implementação da aplicação de Xadrez deverá seguir TODAS as regras abaixo:

Regra 01: A aplicação deverá ter uma rota **/sobre**, que deverá conter o conteúdo da seção **Sobre o jogo de Xadrez** deste documento. Essa página também deverá conter uma imagem das peças de xadrez, tal como mostrado em <https://xadrezapp.herokuapp.com/sobre>.

Regra 02: Os usuários poderão se cadastrar na aplicação através da rota **/signup**. Os campos do formulário de cadastro são: nome completo (precisa ter entre e 100 caracteres), endereço de e-mail (precisa ser um email válido), curso na UFAM, senha de acesso e confirmação de senha. A senha precisa ter 6 caracteres ou mais, e é importante verificar se a senha digitada pelo usuário é igual à senha digitada no campo de confirmação.

Regra 03: O banco de dados deverá obedecer o esquema da Figura 2. Cada uma das tabelas deverá conter um Modelo e ao menos uma Migração. A tabela area deverá ser alimentada através de um **seeder** do Sequelize (vide slides da disciplina).

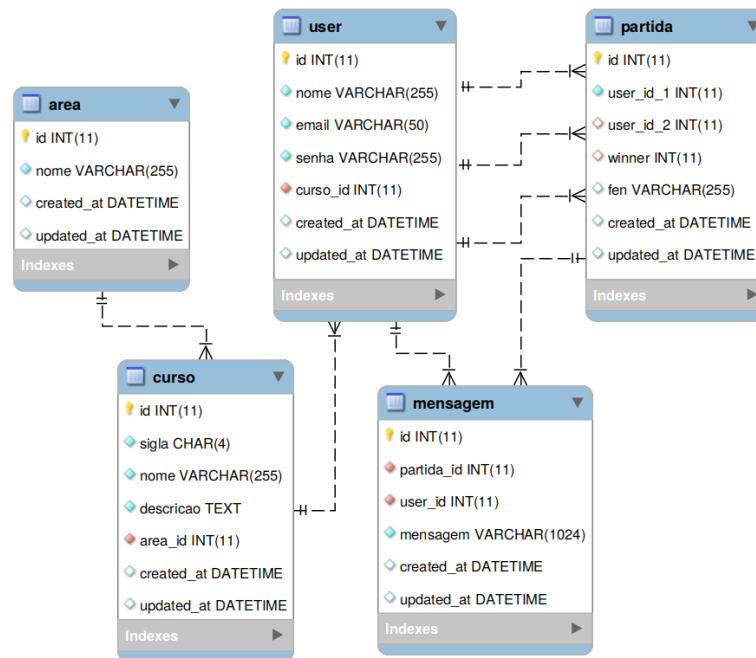


Figura 2: Esquema do banco de dados.

Regra 04: As senhas deverão ser armazenadas no banco de dados de forma criptografada, através do módulo **bcrypt** (vide slides da disciplina).

Regra 05: Quando o usuário não estiver logado na aplicação, o menu superior deverá conter apenas as opções: **Sobre** (rota /sobre), **Login** (rota /login) e **Sign Up** (rota /signup). Ao acessar a tela de login (que deverá conter apenas os campos **email** e **senha**) e informar as credenciais corretamente, o menu superior deverá passar a conter as seguintes opções: **Nova Partida** (rota /partida), **Ranking** (rota /ranking), **Curso** (rota /curso), **Sobre** (rota /sobre) e **Logout** (rota /logout).

Regra 06: A opção **Nova Partida** (rota /partida) irá iniciar uma nova partida de xadrez. No entanto, o jogador que iniciou a partida deverá aguardar a chegada de algum oponente (vide Figura 4). As peças brancas serão do jogador que iniciou a partida, enquanto as peças pretas serão do oponente. A Figura 3 mostra a página que é carregada quando a usuária Mariana Mozart clica em **Nova Partida**.

Regra 07: A página principal de todos os usuários irá apresentar uma listagem das partidas aguardando oponentes. A Figura 4 mostra a página principal de um usuário fictício chamado Eduardo Berlot. Quando o Eduardo aceita o desafio da Mariana (clcando na partida mostrada na página principal), ele será direcionado para a página contendo o tabuleiro e poderá iniciar o jogo com a Mariana (vide Figura 5). Note que, como o Eduardo Berlot ficou com as peças pretas, então a parte de baixo de seu tabuleiro deverá conter as peças pretas. Por outro lado, como a Mariana ficou com as peças brancas, a parte de baixo de seu tabuleiro deverá conter as peças brancas.

Regra 08: A implementação do chat apresentado nas Figuras 3 e 5 não é obrigatória. No entanto, o desenvolvimento do chat irá render alguns pontos extras para o aluno. A ideia do chat é a troca de mensagens entre os dois competidores da partida.



Figura 3: Usuária Mariana Mozart aguardando oponente.



Figura 4: Página principal do usuário Eduardo Berlot, que mostra uma listagem de partidas esperando oponentes. Para o Eduardo aceitar a partida iniciada pela Mariana, basta ele clicar no tabuleiro mostrado na figura.



Figura 5: Partida entre Mariana Mozart e Eduardo Berlog, vista a partir do usuário Eduardo. Note que, ao contrário do tabuleiro visto pela Mariana, a parte de baixo do tabuleiro contém as peças pretas.

Regra 09: A opção **Ranking** do menu superior (rota /ranking) deverá mostrar uma página contendo uma listagem dos usuários com seus respectivos números de vitórias. O ranking deve mostrar apenas usuários que já venceram ao menos uma partida, e é ordenado de forma decrescente de acordo com o número de vitórias de cada um.

Regra 10: Todos os acessos à aplicação deverão gerar logs através do middleware **Morgan** com a opção **short**.

Regra 11: Os pacotes **@fortawesome/fontawesome-free**, **jquery**, **popper.js** e **bootstrap** deverão ser instalados na aplicação. Esses pacotes serão usados pelo template da aplicação (vide regra 08). Para maiores informações sobre esses pacotes, favor consultar os slides da disciplina.

Regra 12: O site **Bootswatch** – <https://bootswatch.com/> – possui um conjunto de templates baseados no Bootstrap. Vá até esse site, escolha um dos temas disponíveis e instale ele em sua aplicação.

Regra 13: Deverá ser implementado um **CRUD** para o modelo Curso, nos moldes do que fora implementado pelo professor durante as aulas. As páginas do CRUD deverão usar ícones do pacote **@fortawesome/fontawesome-free**. Todos os formulários deverão usar o **CSRF** - Cross-Site Request Forgery.

Cursos da UFAM

[+ Adicionar](#)










Sigla	Nome	Opções
IE08	Ciência da Computação	  
IE15	Engenharia de Software	  
FT05	Engenharia da Computação	  

Figura 6: Página principal (/curso/index) do CRUD de curso. Note que a página usa ícones do pacote @fortawesome/fontawesome-free

Observações

O trabalho é individual, e não é permitido usar código desenvolvido por outros alunos da disciplina.

Data e Forma de Entrega

A entrega do trabalho deverá ser feita até o dia **06 de dezembro (sexta-feira)**. Para entregar o código, basta criar um repositório privado chamado AppXadrez em seu Github com o código de sua aplicação. Após isso, é necessário me adicionar como colaborador do seu repositório, para que eu tenha acesso ao mesmo.

Também será preciso informar qual o link do seu repositório AppXadrez no Github. Para isso, foi adicionado o campo **Endereço do Repositório AppXadrez no Github** no **Sign Up** da aplicação disponível em <https://xadrezapp.herokuapp.com>. Desta forma, é importante que você se cadastre na aplicação para que o professor saiba qual o seu repositório.

Sua aplicação deverá rodar na **porta 4567**.

Para instalar e corrigir sua aplicação, o professor irá seguir os seguintes passos:

1. Executar o comando `npm install` no diretório da aplicação
2. Rodar as migrações e os seeders
3. Executar o comando `npm start`
4. Abrir sua aplicação através do endereço <http://localhost:4567>
5. Criar dois usuários fictícios, e tentar jogar uma partida entre ambos
6. Olhar o código e navegar pela aplicação para verificar se cada uma das regras deste trabalho prático foi devidamente seguida