AGÊNCIA EXPERIMENTAL DE ENGENHARIA DE SOFTWARE

PROJETO SIMULADOS

Documento de Arquitetura e Projeto Detalhado

Professor Responsável:
Eduardo Arruda

Arquiteto Contratado:
Cassio Trindade

Arquitetos Responsáveis:
Alan de Oliveira Quadros
Homero Oliveira Santos
Israel Deorce Vieira Jr.
Ricardo Borges Da Silva

7 de Julho de 2020

Conteúdo

1 Introdução

O objetivo deste documento é fornecer uma visão geral do planejamento da arquitetura e do projeto detalhado no desenvolvimento do projeto Simulados, realizado durante os semestre 2018/02 na Agência Experimental de Engenharia de Software (Ages) do curso de Engenharia de Software (ES-360) da Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS). Este documento abrange o propósito, escopo, definição, acrônimos, abreviações, referências e a visão geral da Arquitetura de Software e do Projeto Detalhado utilizados no projeto.

2 Planejamento Estrutural da ferramenta CASE Astah

Em nosso projeto, utilizaremos a ferramenta CASE Astah para a representação de diagramas.

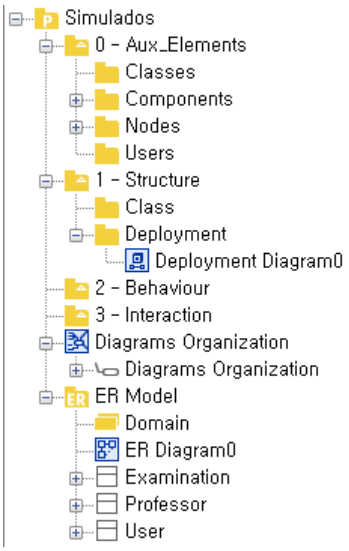

Figura 1: Overview da estrutura dos diagramas no Astah

http://astah.net/manual

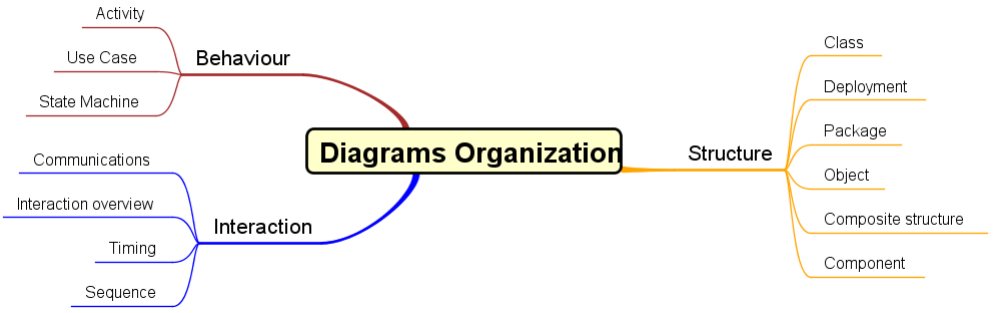

Figura 2: Mapa mental de como organizar diagramas UML

3 Visão de Deployment

O projeto foi dividido em camadas que executam em dispositivos diferentes, e podem ser visualizadas na figura a seguir, juntamente das principais tecnologias envolvidas.

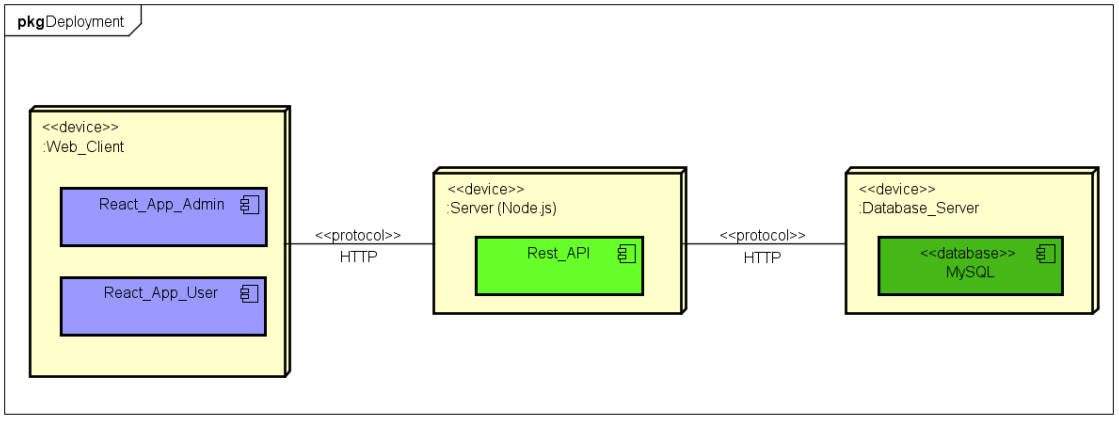

Figura 3: Diagrama de Deployment

- O dispostivo **WebClient** faz parte da camada de Front-End, e possui um componente ReactApp que é uma abstração de uma aplicação desenvolvida na linguagem React.JS para comunicação com o cliente.
- O dispositivo Server faz parte da camada de API Back-end, e possui um componente **RestAPI** que é uma abstração de uma aplicação desenvolvida na linguagem Node.JS para comunicar a camada do cliente com o servidor de banco de dados através de protocolo HTTP.

- O dispositivo **DatabaseServer** faz parte da camada de Persistência de dados, o banco de dados. Este dispositivo possui um componente que é uma abstração de um banco de dados relacional MySQL, que é responsável por armazenar e gerenciar todos os dados do sistema.

4 Arquitetura MRC

Model - Route - Controller

5 Visão de Componentes

5.1 REST-API

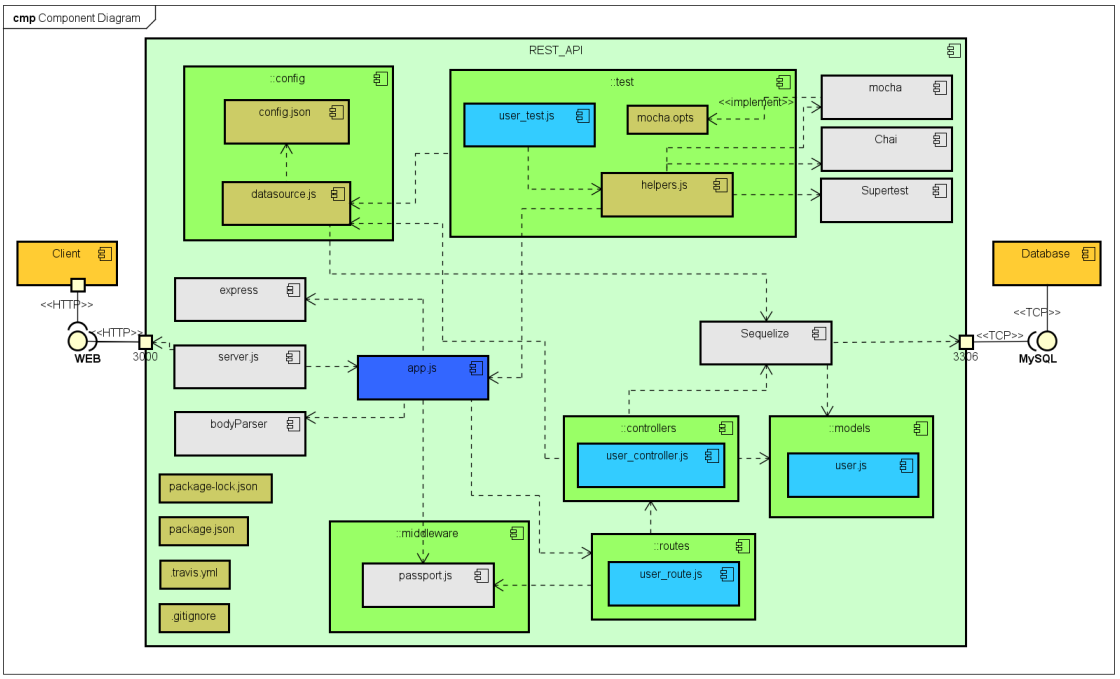

Figura 4: Diagrama de Componentes

A listagem abaixo explica o funcionamento de cada componente separado pelas cores e suas funcionalidades.

- Laranja: Componentes externos à *API*.

  - *Client*: comunica-se diretamente com a API via interface WEB utilizando protocolo $HTTP$.
  - *Database*: comunica-se diretamente com a API via interface $MySQL$ e utiliza protocolo TCP local.

- Azul Escuro: Componentes de controle geral
  - *app.js*: É o único componente da categoria. É responsável por inicializar a aplicação, carregar os demais componentes e gerir a comunicação entre eles.

- Azul claro:
  - Os componentes em azul claro representam as tabela individuais presentes no banco de dados e as ações da API sobre elas. O componente $user.js$ foi utilizado como exemplo, porém muitos outros como questões, provas e respostas deverão compor este grupo de componentes.

    Para cada tipo de componente é necessário a sua representação de modelo, controller, routes, e testes. Verifique a sessão Arquitetura MRC e a sessão 5 Modelo ER.

- Cinza:
  - Componentes $middlewares$ ou $frameworks$ que possuem um propósito específico. Não há necessidade de entender a fundo como estes componentes operam, sendo esta abstração o real valor de se utilizá-los.
    * *server.js*: Responsável pela inicialização do servidor da API, e funciona como um canal simples de entrada e saída de informações com o *Cliente*.
    * *express*: $Framework$ popular de servidor para $Node.js$. O $Express$ evita a repetição de código e possui funções prontas que facilitam ações como: análise de carga útil, *cookies*, guardamento de sessões na memória ou Redis, escolha de rotas corretas.
    * $bodyParser$:

- Marrom:
  - Componentes de configurações específicas. Definem ambiente, e variaveis a serem utilizadas pelos demais componentes.

6 Modelo de Entidade e Relacionamento

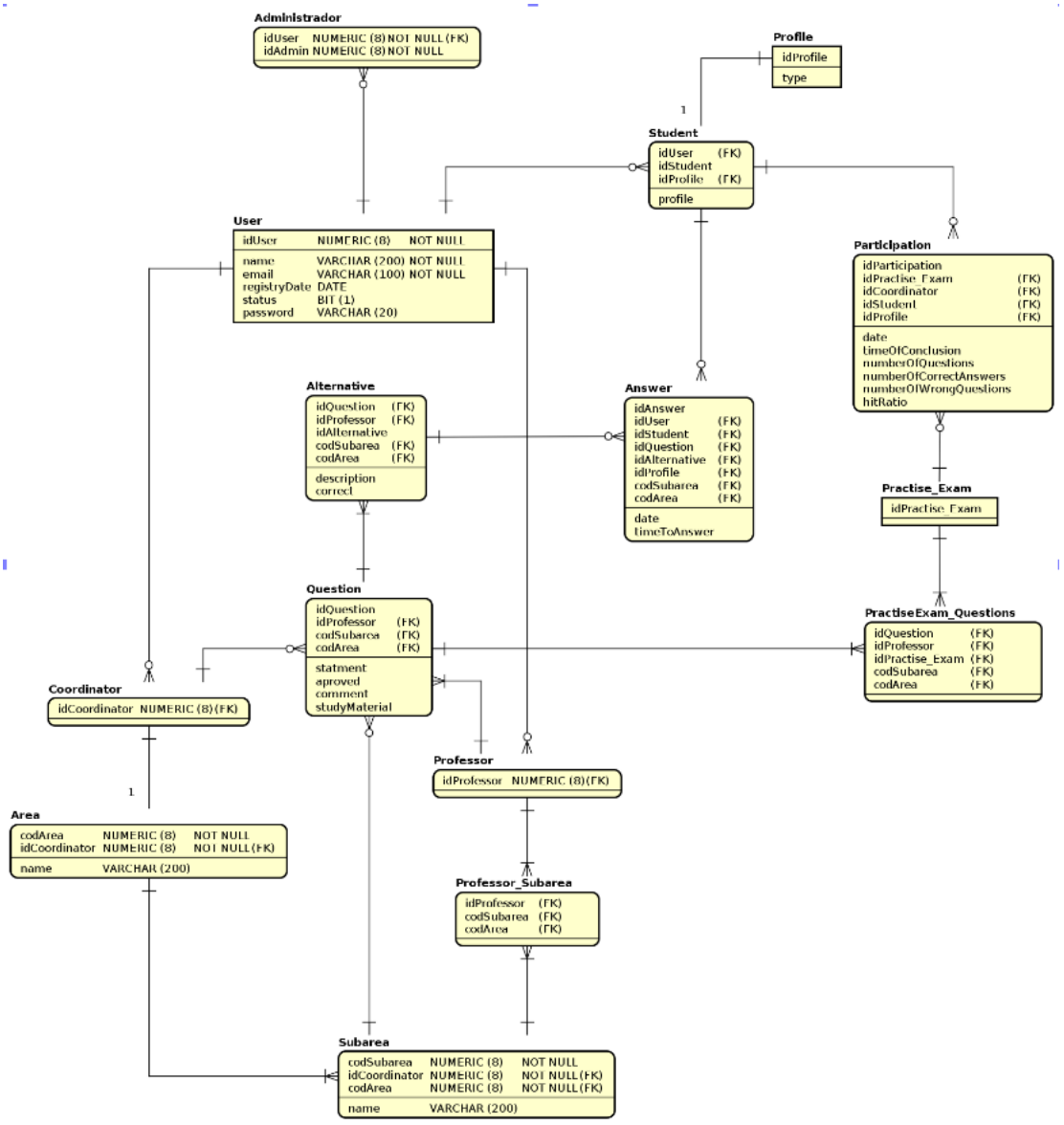

Figura 5: Modelo de Entidade e Relacionamento

Referências