

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SÃO PAULO  
CÂMPUS CAMPINAS**

**FELIPE MATHEUS LOPES**

**SISTEMA ONLINE PARA AVALIAÇÃO DE DESEMPENHO DOS ALUNOS**

**CAMPINAS**

**2017**

FELIPE MATHUES LOPES

**SISTEMA ONLINE PARA AVALIAÇÃO DE DESEMPENHO DOS ALUNOS**

Trabalho de Conclusão de Curso apresentado como exigência parcial para obtenção do diploma do Curso de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia Câmpus Campinas.

Orientador: Prof. Me. Fernando de Almeida Freitas.

CAMPINAS

2017

L864s Lopes, Felipe Matheus  
Sistema online para avaliação de desempenho dos alunos. / Felipe Matheus  
Lopes. – Campinas, 2017.  
40f. : il.

Orientador: Fernando de Almeida Freitas.

Monografia (Graduação) – Instituto Federal de São Paulo – Câmpus  
Campinas. Curso de Tecnologia em Análise e Desenvolvimento de Sistemas,  
2017.

1. Gerador de prova. 2. Sistema para educação. 3. Análise de desempenho.  
4. Sistema web. I. Instituto Federal de São Paulo - Câmpus Campinas. Curso de  
Tecnologia em Análise e Desenvolvimento de Sistemas. II. Título.

Felipe Matheus Lopes

SISTEMA ONLINE PARA AVALIAÇÃO DE DESEMPENHO DOS ALUNOS

Trabalho de Conclusão de Curso apresentado como exigência parcial para obtenção do diploma do Curso de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo Câmpus Campinas.

Aprovado pela banca examinadora em: 28 de novembro de 2017.

BANCA EXAMINADORA



---

Prof. Me. Fernando de Almeida Freitas  
IFSP Câmpus Campinas

Prof.<sup>a</sup> Dr.<sup>a</sup> Cecília Pereira de Andrade  
IFSP Câmpus Campinas

Prof. Dr. Ricardo Barz Sovat  
IFSP Câmpus Campinas

*Dedico este trabalho aos meus familiares,  
colegas de classe, professores e servidores do Instituto  
que colaboraram em minha jornada formativa.*

## **AGRADECIMENTOS**

Agradeço primeiramente a Deus, pelo dom da vida e pela oportunidade de concluir mais uma etapa de minha experiência acadêmica. Agradeço a todos os professores e servidores do IFSP campus Campinas, que contribuíram direta e indiretamente para a conclusão desse trabalho. Agradeço também à minha família, que deu todo o apoio necessário para que eu chegasse até aqui. Agradeço em especial ao meu orientador que me auxiliou a solucionar as dificuldades encontradas no caminho.

*"É fazendo que se aprende a fazer aquilo  
que se deve aprender a fazer".*

*Aristóteles*

## RESUMO

A maior parte da elaboração, aplicação e correção de provas para avaliar o desempenho dos alunos do Instituto Federal de Educação, Ciência e Tecnologia São Paulo campus Campinas são feitas de forma manual, exigindo um esforço do professor para preparar a avaliação de acordo com o assunto e dificuldade pretendidos. Para ajudar na melhoria deste processo, este projeto tem como objetivo o desenvolvimento de um sistema *online* para avaliação de estudantes baseado em uma plataforma web para uso interno. Além disso, com as informações estruturadas sobre as avaliações é possível analisar mais facilmente o desempenho dos alunos e identificar possíveis assuntos que determinado aluno ou grupo de indivíduos encontraram maior dificuldade, ajudando assim, o professor a aumentar sua produtividade na análise de desempenho dos estudantes.

**Palavras-chave:** Gerador de Prova. Sistema para Educação. Análise de Desempenho. Sistema Web.



## **ABSTRACT**

Most of the preparation, application and correction of the exams to measure the students performance at high school of Instituto Federal de Educação, Ciência e Tecnologia São Paulo campus Campinas are made manually, requiring an effort of the professor to make the exam according to the subject and level wanted. To help in the improvement of this process, this project has as objective the development of an online system to the evaluation of the students based in a web platform to user internally. Moreover, with the structured information about the exams it is possible to easily analyse the students performance and identify possibly subjects that a specific student or group of individuals struggled to solve it, helping the professor to improve the productivity in the analysis of the students performance.

**Keywords:** Exam Generator. Education System. Performance Analysis. Web System.

## LISTA DE FIGURAS

Figura 1 – Código “Olá Mundo” em um componente React.....	20
Figura 2 – Código “Olá Mundo” em node.js com o <i>framework</i> express.....	21
Figura 3 – Representação da Arquitetura do Sistema.....	22
Figura 4 – Diagrama de Caso de Uso.....	24
Figura 5 – Diagrama de Classe.....	24
Figura 6 – Diagrama da Arquitetura do Sistema.....	25
Figura 7 – Modelo de entidade-relacionamento.....	27
Figura 8 – Página de Login.....	30
Figura 9 – Página de Login do Aluno.....	31
Figura 10 – Página Inicial do Aluno.....	32
Figura 11 – Página de Avaliação.....	33
Figura 12 – Página de Login do Professor.....	34
Figura 13 – Página Inicial do Professor.....	35
Figura 14 – Página de Desempenho.....	36
Figura 15 – Página de Criar Avaliação.....	37
Figura 16 – Página de Cadastrar Classe.....	38

## LISTA DE QUADROS

Quadro 1 – Tecnologias Utilizadas para Implementar o Sistema.....	26
---	----

## LISTA DE SIGLAS

API	Appplication Program Interface
BD	Banco de Dados
DOM	Document Object Model
HTTP	Hypertext Transfer Protocol
HTML	Hypertext Markup Language
IFSP	Instituto Federal de Educação, Ciência e Tecnologia de São Paulo
REST	Representational State Transfer

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>12</b>
<b>2</b>	<b>OBJETIVOS DO TRABALHO</b>	<b>14</b>
2.1	Objetivo Geral	14
2.2	Objetivo Específico	14
<b>3</b>	<b>JUSTIFICATIVA</b>	<b>15</b>
<b>4</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>16</b>
4.1	Avaliação	16
4.2	Desenvolvimento do Sistema	17
4.2.1	<i>Frontend</i>	18
4.2.2	<i>Backend</i>	20
4.2.3	<i>Arquitetura do Sistema</i>	21
<b>5</b>	<b>METODOLOGIA</b>	<b>22</b>
5.1	Levantamento de Requisitos	22
5.2	Definição da Arquitetura do Sistema	24
5.3	Seleção das Tecnologias	25
5.4	Modelagem e Criação do Banco de Dados	26
5.5	Implementação do Sistema	27
5.5.1	<i>Implementação do Frontend</i>	27
5.5.2	<i>Implementação do Backend</i>	27
5.5.3	<i>Testes</i>	28
<b>6</b>	<b>RESULTADOS</b>	<b>29</b>
6.1	Fluxo do Aluno	30
6.1.1	<i>Login</i>	30
6.1.2	<i>Página Inicial</i>	31
6.1.3	<i>Avaliação</i>	32
6.2	Fluxo do Professor	33
6.2.1	<i>Login</i>	33
6.2.2	<i>Página Inicial</i>	34
6.2.3	<i>Detalhar Avaliação</i>	35
6.2.4	<i>Criar Avaliação</i>	36

<i>6.2.5 Página onde se cadastram Classes</i> .....	37
<b>7 CONCLUSÃO</b> .....	38
<b>REFERÊNCIAS</b> .....	39

## 1 INTRODUÇÃO

A educação é importante na sociedade moderna, pois ela capacita os indivíduos a uma melhor integração com a comunidade e os qualifica para o mercado de trabalho. Atualmente não se pode pensar nossa sociedade sem a relacionar com tecnologia. Segundo Moran (2013) estas tecnologias estão afetando e gerando mudanças na educação. Ainda segundo o autor, a educação está englobando práticas comuns na educação a distância.

Dentre os processos que poderiam beneficiar-se do uso da tecnologia, podemos mencionar a elaboração de testes para estudantes. De acordo com Gatti (2003):

A avaliação tem por finalidade acompanhar os processos de aprendizagem escolar, compreender como eles estão se concretizando, oferecer informações relevantes para o próprio desenvolvimento do ensino na sala de aula em seu dia-a-dia, para o planejamento e replanejamento contínuo da atividade de professores e alunos.

Gatti (2003) ainda afirma que as provas são consideradas uma ferramenta para avaliar o ensino e que de acordo com dados obtidos dos docentes, não existe um único e mais eficiente método para avaliar os alunos, e recomenda que existam avaliações com uma determinada frequência, evitando assim a pressão sobre os alunos, uma vez que eles seriam avaliados em mais períodos e com um acúmulo menor de conteúdo, além de facilitar o acompanhamento de seu rendimento.

Atualmente, grande parte do processo de aplicação e correção das avaliações dos alunos no IFSP campus Campinas é realizada manualmente. Além do esforço para elaborar um teste de desempenho, o docente precisa ainda dedicar parte de seu tempo para fazer a correção deste e então efetuar a análise a partir dos resultados obtidos. Este procedimento acaba sendo repetitivo e muitas vezes acumulado em decorrência da rotina do professor, dificultando a aplicação de exames periódicos, como recomendado por Gatti (2003).

Entendendo os fatos apontados, este projeto propõe o desenvolvimento de um sistema *online* para a elaboração e aplicação das avaliações, diminuindo assim a complexidade e o tempo do professor para elaboração e correção das provas. Com esta proposta será possível analisar mais facilmente o desempenho dos alunos e identificar possíveis assuntos em que determinado aluno ou grupo de indivíduos encontraram maior dificuldade através de relatórios de desempenho do estudante.

Com o sistema espera-se uma maior eficiência na tarefa de avaliar e, conseqüentemente, uma otimização no tempo do professor e recursos da instituição, uma vez

que não será necessário imprimir as avaliações quando existe a opção de realizar a mesma através de um sistema web.

Com o ganho de tempo e facilidade proporcionados pelo software, o docente teria mais tempo livre para participar de outras atividades do curso, como pesquisas e projetos de extensão que a instituição fomenta, podendo assim colaborar ainda mais com a qualidade do ensino.

Ademais, dentre as vantagens da utilização de um sistema para gerar provas, pode-se citar:

- Banco de dados de questões;
- Geração de provas personalizadas;
- Histórico de avaliações;
- Relatório sobre o desempenho dos alunos;
- Agilidade para criar avaliação;
- Acompanhamento do desempenho dos alunos na avaliação em tempo real.

As próximas seções consistem em mais seis capítulos, que se encontram organizados da seguinte forma:

- O Capítulo 2 apresenta o objetivo geral do sistema, bem como os específicos;
- O Capítulo 3 apresenta a importância do projeto para a comunidade e sua justificativa para desenvolvimento;
- O Capítulo 4 apresenta as fundamentações teóricas que sustentam a justificativa, bem como as decisões para o desenvolvimento do sistema;
- O Capítulo 5 apresenta as técnicas de implementação do sistema e as razões da escolha das tecnologias utilizadas;
- O Capítulo 6 apresenta o sistema implementado e suas principais funcionalidades;
- Por fim, o Capítulo 7 apresenta as conclusões e resultados que a execução do projeto proporcionou.

## **2 OBJETIVOS DO TRABALHO**

Para a elaboração do trabalho foram definidos objetivos afim de guiar o autor do projeto para a conclusão do mesmo. Abaixo encontram-se o objetivo geral e o objetivo específico.

### **2.1 Objetivo Geral**

O principal objetivo desse projeto é o desenvolvimento de um sistema *online* para avaliação de alunos, permitindo que os educadores utilizem a tecnologia como instrumento para avaliação.

### **2.2 Objetivo Específico**

Para atingir o objetivo geral, o projeto se propõe a realizar os seguintes objetivos específicos:

- Desenvolver uma aplicação web com as recentes tecnologias do mercado;
- Construção de uma plataforma para criação das avaliações e geração de relatórios;
- Criação de uma interface simples e fácil de utilizar.



### 3 JUSTIFICATIVA

O Instituto Federal de Educação, Ciência e Tecnologia de São Paulo campus Campinas iniciou suas atividades com a primeira turma de Análise e Desenvolvimento de Sistemas no segundo semestre de 2013. Naquela época a comunidade da instituição era menor e os professores lecionavam aula somente para estes alunos. Atualmente o campus atende um número maior de alunos. Além disso, foram adicionados novos cursos, como o Ensino Médio e o técnico em Eletrônica.

Com o crescimento do número dos alunos e o aumento das disciplinas, muitos professores passaram a atender mais de uma turma e mais de um curso, exigindo assim um maior alocamento de recursos para gerenciar as aulas e desempenho desses indivíduos. Como a análise de aprendizagem é uma importante atividade e visando facilitar e agilizar a aplicação de provas aos estudantes, este projeto tem como objetivo o desenvolvimento de um sistema *online* para avaliação de alunos.

Com o sistema proposto para este trabalho, espera-se aumentar a eficiência dos recursos da instituição e agilizar um processo trabalhoso e repetitivo, instrumentalizando o professor no processo avaliativo, além de oferecer uma alternativa para realizar tal tarefa.

Apesar de alguns docentes já utilizarem o Moodle<sup>1</sup> – um software para apoio à educação – em suas disciplinas, não são todos os professores que aderiram à plataforma. Além disso, é preciso uma configuração prévia para iniciar sua utilização e adicionar os alunos ao curso *online*, algo que desmotiva muitos professores a iniciar sua utilização.

A proposta de um sistema de avaliação *online* vem justamente para solucionar este problema. Idealizado para ser um sistema simples, ele foca somente na atividade de avaliar os alunos e gerar estatísticas sobre a mesma para que o professor possa extrair informações importantes sobre como os alunos desempenharam. Por não precisar de uma configuração prévia (somente o cadastro do professor e da disciplina deste professor) e nem o gerenciamento dos alunos no sistema, o esforço para a iniciação da utilização do software é pequeno.

Ademais, o sistema tem o código aberto e foi desenvolvido com o objetivo de ser gratuito e um banco de dados das mais variadas questões, disponível para qualquer docente que queira utilizar a ferramenta.

---

<sup>1</sup> Disponível em <<https://moodle.org>> Acesso em 20 de Novembro.

## 4 FUNDAMENTAÇÃO TEÓRICA

Como apontado na introdução, esta seção tem por objetivo elencar os principais referenciais teóricos que embasaram este trabalho. Para tal, foi separada em duas seções, sendo a última dividida em três para uma melhor explicação dos conceitos. Abaixo encontram-se as seções com suas respectivas explicações.

### 4.1 Avaliação

Segundo Frederiksen e Collins (1989) “existem grandes apostas no desempenho dos alunos em avaliações e, conseqüentemente, uma grande pressão sobre os administradores da escola, os professores e os alunos para melhorarem seus resultados nestes testes”. Por esta razão a avaliação é um processo importante em uma instituição de ensino.

Já Luckesi (2011) aponta que o exame escolar como conhecemos hoje é uma ferramenta antiga, desenvolvida entre os séculos XVI e XVII, que durante todo esse período não houve uma profunda transformação no modo como foi idealizada. Ainda segundo o autor, o ato de avaliar é uma prática comum mas muitas vezes essas avaliações falham em identificar se os indivíduos realmente aprenderam determinado assunto.

Luckesi (2011) afirma que os resultados do rendimento dos estudantes decorrentes do ensino do professor podem depender de muitos motivos, incluindo a ação pedagógica. Dentre elas podemos mencionar a falta de atenção às dificuldades dos alunos e ferramentas impróprias para levantar informações sobre desempenho. Conhecendo os fatos apontados, esse projeto propõe a criação de um sistema web para fornecer uma ferramenta que possa auxiliar o professor no processo de avaliar o aprendizado.

Além disso, “o desenvolvimento de sistemas *online* e baseados na web está se tornando cada vez mais crítico para a estratégia de negócios de muitas organizações” (LOWE, 2003). Seja para automatizar uma tarefa ou cortar custos, atualmente os sistemas mostram-se como uma importante ferramenta. Lowe (2003) aponta que o uso do ambiente web apresenta características únicas. Dentre os diferenciais, o autor cita:

- Pouco tempo para entrega inicial;
- Precisas evoluções e manutenções;
- Aumento na ênfase da interface do usuário;
- Mudança rápida de tecnologias.

Ademais, a utilização da web possibilita que o usuário tenha liberdade para utilizar o sistema de qualquer plataforma e localização, uma vez que para acessar o sistema é necessário somente ter acesso à internet. Sabendo da importância da avaliação dos alunos, este projeto busca o aperfeiçoamento para um procedimento realizado na instituição a fim de otimizar o tempo dos professores e instrumentalizar o mesmo para uma melhor análise de resultados.

## 4.2 Desenvolvimento do Sistema

Dentre as práticas recomendadas para o ciclo de vida do software, a documentação tem sido relevante Souza, Anquetil e Oliveira (2005). Segundo uma pesquisa realizada com programadores pelos autores, as documentações essenciais para um software, dentre os recursos disponíveis, são:

- Código fonte e comentários;
- Modelo de dados;
- Informação sobre os requisitos.

Dentro deste contexto, é necessário efetuar um levantamento de requisitos para identificar as funcionalidades que são fundamentais para que o projeto funcione de acordo com as necessidades dos usuários para então ter os dados para a criação das documentações recomendadas e iniciar o desenvolvimento do sistema.

Durante a implementação de um sistema, o uso de *frameworks* é uma prática comum pois, como apontado por Vuksanovic e Sudarevic (2011), “o *framework* é uma camada de abstração fornecendo uma biblioteca com soluções para os problemas mais comuns de programação, com o objetivo de eliminar operações repetitivas”. Além disso, Vuksanovic e Sudarevic (2011) afirmam que o uso deste oferece muitas vantagens, dentre as quais podemos destacar:

- Implementação do software mais rapidamente;
- Segurança melhorada;
- Disponibilidade de bibliotecas úteis e praconizadas.

Mesmo com o uso de um *framework*, a etapa de testes do software é necessária para garantir a qualidade do mesmo. Segundo Chappell (2017) o teste do software geralmente procura identificar erros nas funcionalidades do sistema, enquanto na verdade o conceito de qualidade muda de perspectiva de acordo com as pessoas envolvidas no projeto. Enquanto o usuário espera que o sistema funcione com qualidade, o programador espera que o código tenha

qualidade. Dentre as possibilidades, o autor afirma que a qualidade de software engloba basicamente três aspectos:

- Qualidade Funcional (necessidades dos usuários);
- Qualidade Estrutural (estrutura do código);
- Qualidade de Processo (prazo e orçamento).

Outra etapa importante é a seleção das tecnologias a serem utilizadas na implementação do sistema. Como citado na Seção 4.1, o ambiente web tem uma mudança rápida nas tecnologias, então é preciso avaliar o quão ativa é a comunidade e o desenvolvimento destas, garantindo assim que os problemas de segurança sejam corrigidos e que inovações sejam implementadas.

Para melhor separação de conceitos, o desenvolvimento da aplicação foi dividido basicamente em duas partes: *frontend* e *backend*. O *frontend* é a parte responsável pela interface e interação das informações com o usuário. Já o *backend* é a parte responsável por processar estas informações e persistir no banco de dados (BD). Abaixo encontra-se a explanação de cada segmento.

#### **4.2.1 Frontend**

O *frontend* é basicamente a interface do usuário, onde ele irá interagir com os dados e fará as ações necessárias através desta. Uma das atividades na implementação do *frontend* é a definição de quais elementos serão exibidos para o usuário e suas respectivas posições. Este processo também é conhecido como “definição do leiaute”. Durante esta etapa, procurou-se seguir as Heurísticas de Nielsen, que são os princípios para evitar erros de usabilidade. Essas heurísticas são os dez princípios gerais de design da interface do usuário definido por Nielsen (1994):

1. Visibilidade do estado do sistema;
2. Correspondência entre o sistema e o mundo real;
3. Liberdade e controle para o usuário;
4. Consistência e padrões;
5. Prevenções de erros;
6. Reconhecimento em vez de relembrar;
7. Flexibilidade e eficiência de uso;
8. Estética e design minimalista;
9. Ajudar usuários a reconhecerem, diagnosticarem e recuperarem-se de erros;

## 10. Ajuda e documentação.

Em relação à tecnologia, foi utilizado a biblioteca React<sup>2</sup>. De acordo com o *website* oficial do React, ele é uma biblioteca JavaScript para o desenvolvimento de interfaces de usuário. É um projeto de código aberto mantido e desenvolvido pelo Facebook<sup>3</sup>, mas com grande participação da comunidade e de outras grandes empresas. Além do Facebook e seus produtos como o Instagram<sup>4</sup>, essa biblioteca é utilizada por famosas empresas, como Airbnb<sup>5</sup> e Uber<sup>6</sup>, que inclusive disponibilizam componentes de código aberto desenvolvido para a utilização em aplicações baseadas no React.

O React destaca-se por usar componentes. Estes componentes foram criados com o princípio que funcionem independentemente, tendo assim uma baixa ou nenhuma dependência de outros elementos da aplicação (HUNT, 2017a). Com isso, é possível reutilizar estes componentes permitindo que o tempo de desenvolvimento e a manutenção do código sejam mais eficientes. Outra característica que vale ressaltar é a inserção do código de marcação HTML direto no JavaScript (HUNT, 2017b), eliminando assim a necessidade de criar um outro arquivo somente para armazenar este código de marcação, como vemos em *frameworks frontend* (Angular<sup>7</sup>, Ember.js<sup>8</sup> e Vue.js<sup>9</sup>). Na Figura 1, encontra-se o código de um simples componente em React.

Figura 1 – Código “Olá Mundo” em um componente React

```
1  import React from 'react';
2
3  export default class Exemplo extends React.Component {
4    render() {
5      return (
6        <div>
7          <p>
8            Olá mundo!
9          </p>
10         </div>
11       );
12     }
13 }
```

Fonte: elaborado pelo autor

<sup>2</sup> Disponível em <<https://reactjs.org>> Acesso em 20 de Novembro.

<sup>3</sup> Disponível em <<https://www.facebook.com>> Acesso em 20 de Novembro.

<sup>4</sup> Disponível em <<https://www.instagram.com>> Acesso em 20 de Novembro.

<sup>5</sup> Disponível em <<https://www.airbnb.com>> Acesso em 20 de Novembro.

<sup>6</sup> Disponível em <<https://www.uber.com>> Acesso em 20 de Novembro.

<sup>7</sup> Disponível em <<https://angular.io>> Acesso em 20 de Novembro.

<sup>8</sup> Disponível em <<https://www.emberjs.com>> Acesso em 20 de Novembro.

<sup>9</sup> Disponível em <<https://vuejs.org>> Acesso em 20 de Novembro.

### 4.2.2 Backend

Para a implementação do *backend* foi utilizado o Node.js<sup>10</sup>, um ambiente JavaScript orientado a eventos não bloqueantes. Segundo Tilkov e Vinoski (2010), o processamento de requisições deste difere das modernas tecnologias *multithreads*. Geralmente os servidores alocam um espaço em memória para cada requisição em uma fila e somente atendem à próxima requisição quando a anterior está concluída, mesmo que esta tenha um evento bloqueante (como acesso à disco). Já o Node.js trabalha com uma rotina para receber todas as requisições e responder o mais rápido possível. Para tal tarefa, esta ferramenta trabalha com dois principais elementos: uma rotina para receber as requisições e uma fila para tratar as ações bloqueantes. Então, quando uma requisição é solicitada ao servidor, o Node.js vai processá-la e, caso esta requisição exija uma ação bloqueante, ela será incluída na fila para ser executada assim que possível. Quando esta ação bloqueante é finalizada, é então enviada a resposta ao cliente, permitindo que durante este tempo o servidor consiga atender outras requisições, potencializando assim a eficiência do servidor. Na Figura 2, encontra-se o código de um simples servidor em Node.js utilizando o *framework* Express<sup>11</sup>.

Figura 2 – Código “Olá Mundo” em Node.js com o *framework* Express

```

1  var express = require('express')
2  var app = express()
3
4  app.get('/', function (req, res) {
5    |   res.send('hello world')
6  |   })
7
8  app.listen(3000)

```

Fonte: elaborado pelo Autor

Definida a tecnologia utilizada no servidor para atender às requisições, é necessário o uso de um banco de dados para gerenciar os dados do sistema. O banco de dados selecionado foi o MySQL<sup>12</sup>, devido à sua grande popularidade e da experiência do autor do projeto com ele. “Usamos o servidor de banco de dados para fornecer armazenamento confiável e persistente de dados” (MIKOWSKI; POWELL, 2013). Segundo Codd (1970), existem muitas formas de

<sup>10</sup> Disponível em <<https://nodejs.org>> Acesso em 20 de Novembro.

<sup>11</sup> Disponível em <<https://expressjs.com>> Acesso em 20 de Novembro.

<sup>12</sup> Disponível em <<https://www.mysql.com>> Acesso em 20 de Novembro.

criar-se um modelo relacional para um banco de dados, sendo o recomendado seguir as normas padrões (NP). Caso as relações no banco de dados tenham redundância, isto talvez cause anomalias nos dados – como atualizar o dado em uma tabela e não atualizar onde o dado é repetido. Um processo que trata essas indesejadas anomalias é a normalização.

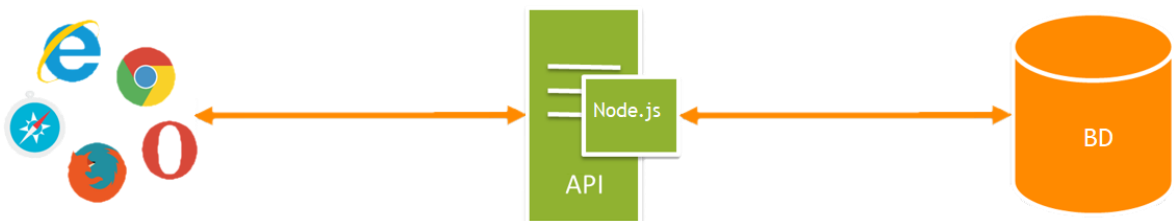
#### 4.2.3 Arquitetura do Sistema

Apresentado o conceito das partes do sistema, é preciso entender como elas se relacionam. Segundo Fielding e Taylor (2002):

Uma arquitetura determina como elementos do sistema são identificados e alocados, como o elemento interage para formar um sistema, a quantidade de granularidade da comunicação necessária para interação, e os protocolos de interface para comunicação.

Parte desta arquitetura do sistema é conhecida como *Application Programming Interface* (API) e foi definida de acordo com os métodos já consolidados para o desenvolvimento web. Ela consiste em uma interface para aplicação do *frontend* interagir com o *backend* e então acessar ou persistir as informações no banco de dados. Esse modelo de desenvolvimento visa permitir que aplicações possam acessar a mesma informação no mesmo servidor em um determinado formato, permitindo assim a reutilização da API para outras aplicações, caso necessário. A Figura 3 ilustra o tráfego de informações da aplicação.

Figura 3 – Representação da Arquitetura do Sistema



Fonte: elaborado pelo autor

Ademais, o modelo da arquitetura utilizada no sistema foi o *Representational State Transfer* (REST). Além de vantagens técnicas, ela é usada largamente em grandes empresas como Facebook, Twitter<sup>13</sup> e Instagram. De acordo com Fielding; Taylor (2002), “REST é um conjunto coordenado de restrições arquitetônicas que tentam minimizar a latência da rede de comunicação enquanto que, ao mesmo tempo, maximiza a independência e escalabilidade das implementações dos componentes”.

<sup>13</sup> Disponível em <<https://twitter.com>> Acesso em 20 de Novembro.

## 5 METODOLOGIA

Para cumprir o objetivo de desenvolver um sistema *online* para avaliação de alunos, a execução do projeto foi dividido em cinco seções, sendo a última subdividida em três para melhor separação das responsabilidades. Essa divisão está em ordem de implementação e ajuda a entender a importância de cada etapa e as decisões do autor do trabalho. Abaixo encontram-se as seções com suas respectivas explicações.

### 5.1 Levantamento de Requisitos

Para o desenvolvimento do projeto foram definidas primeiramente as funções que o sistema teria que desempenhar. Esta etapa foi fundamental para compreender as necessidades que o sistema em questão precisaria atender e prosseguir para as próximas etapas com mais segurança em relação aos seus requisitos mínimos. Os requisitos foram levantados para desenvolver um sistema que gerasse avaliações automaticamente utilizando as questões das versões anteriores do Exame Nacional do Ensino Médio. Como não foram encontradas as questões de uma forma estruturada, optou-se por desenvolver um sistema com a capacidade de reproduzir as provas do exame, necessitando cadastrar as questões manualmente.

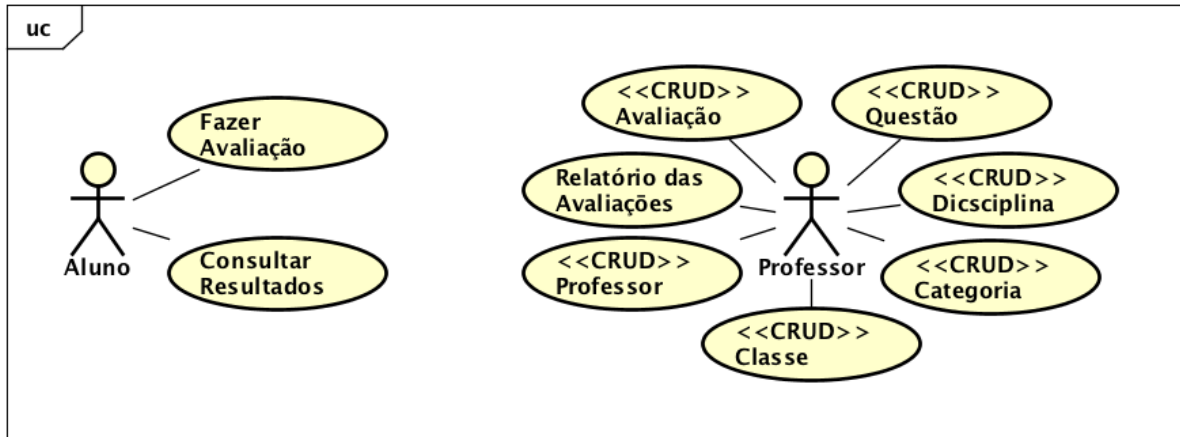
Nesta etapa optou-se também por algumas facilidades no sistema, afim de reduzir a necessidade de configuração deste e agilizar seu uso. Podemos citar:

- Criar o BD com dados iniciais, como professor e disciplina;
- O professor será o próprio administrador do sistema;
- O aluno não precisa de um pré-cadastro para utilizar o sistema.

Nas Figuras 4 e 5 encontram-se os diagramas de caso de uso e de classe elaborados no início do projeto.



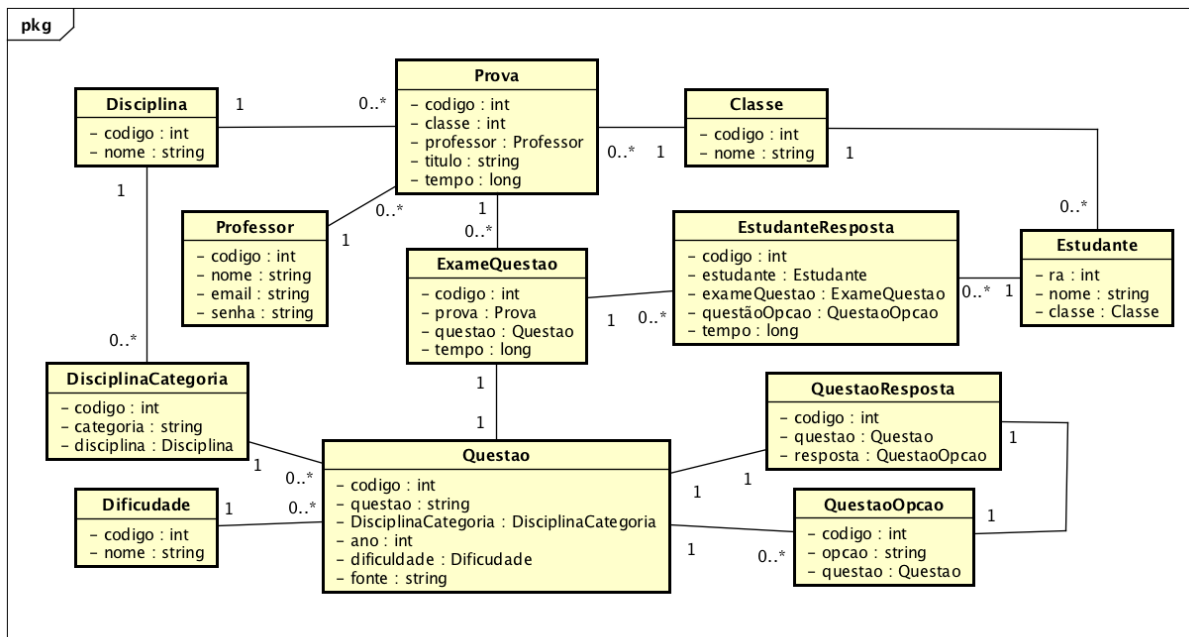
Figura 4 – Diagrama de Caso de Uso



powered by Astah

Fonte: elaborado pelo autor

Figura 5 – Diagrama de Classe



powered by Astah

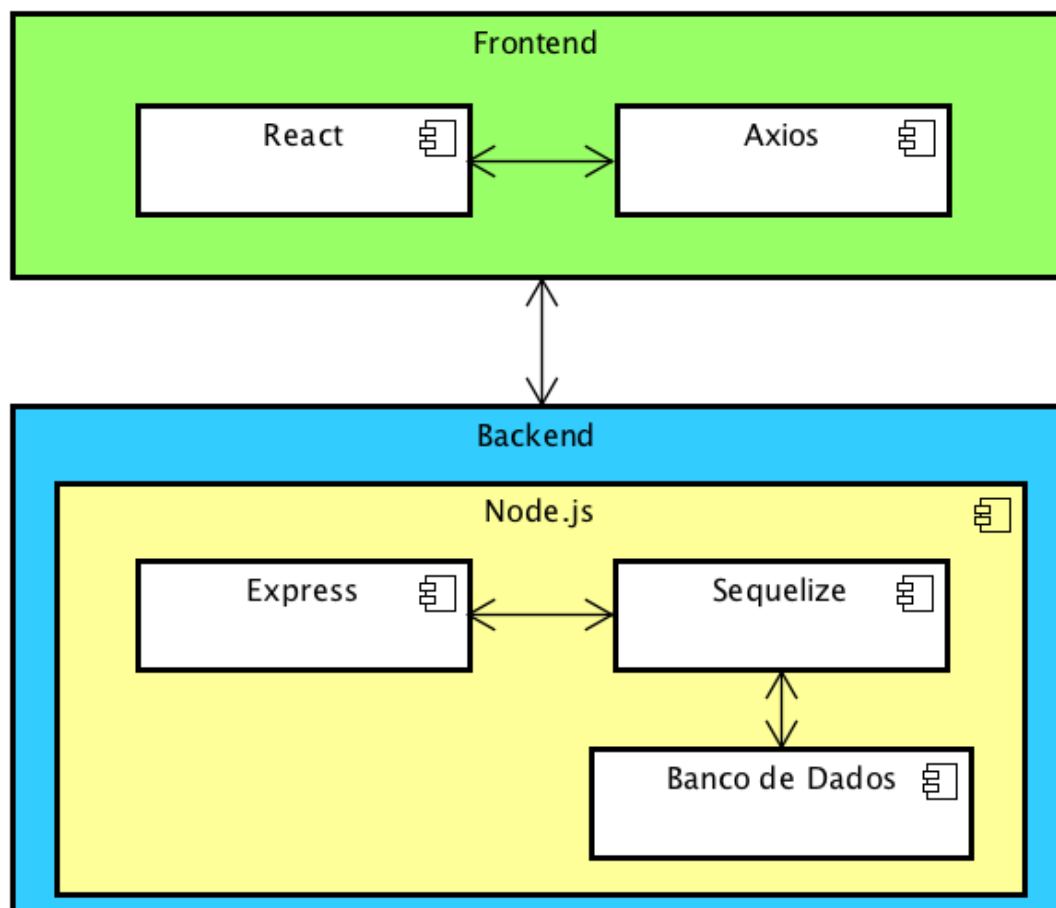
Fonte: elaborado pelo autor

## 5.2 Definição da Arquitetura do Sistema

Após a definição de escopo do projeto e suas funcionalidades, iniciou-se o planejamento da arquitetura do sistema, que foi definida de acordo com os métodos já consolidados para o desenvolvimento web. A arquitetura utilizada baseou-se em uma aplicação web com requisições assíncronas, onde não é preciso atualizar a página para fazer uma solicitação ao servidor. Como pode observar-se na Figura 6, essas requisições são enviadas ao servidor (*backend*), processadas e então enviada a resposta para a aplicação (*frontend*) utilizando o mesmo fluxo da comunicação, mas no sentido inverso. Repare que a biblioteca Sequelize só é acessada quando existe a necessidade de acessar o BD.

Com a arquitetura definida, foi então realizada a escolha das tecnologias usadas para o desenvolvimento do sistema. Na Figura 6 é possível visualizar a arquitetura do sistema e a utilização das respectivas tecnologias.

Figura 6 – Diagrama da Arquitetura do Sistema



### 5.3 Seleção das Tecnologias

A seleção das tecnologias baseou-se no que as grandes empresas usam, com os parâmetros definidos na Seção 4.2 e com a experiência do autor com tecnologias deste projeto. No Quadro 2 encontram-se as tecnologias selecionadas para utilizar no sistema, selecionadas de acordo com os requisitos apontados. Nas próximas seções encontra-se a explicação dessas bibliotecas e suas características.

Quadro 1 – Tecnologias Utilizadas para Implementar o Sistema

<b>Tecnologia</b>	<b>Descrição</b>
React	Biblioteca para desenvolver interface de usuário
Bootstrap <sup>14</sup>	Biblioteca para estilização da interface de usuário
Axios <sup>15</sup>	Biblioteca para requisições HTTP na interface do usuário
Node.js	Ambiente JavaScript para atender às requisições HTTP
Express	<i>Framework</i> para o Node.js
Sequelize	<i>Framework</i> para a comunicação entre o servidor e o BD
MySQL	Banco de Dados

Fonte: elaborado pelo autor

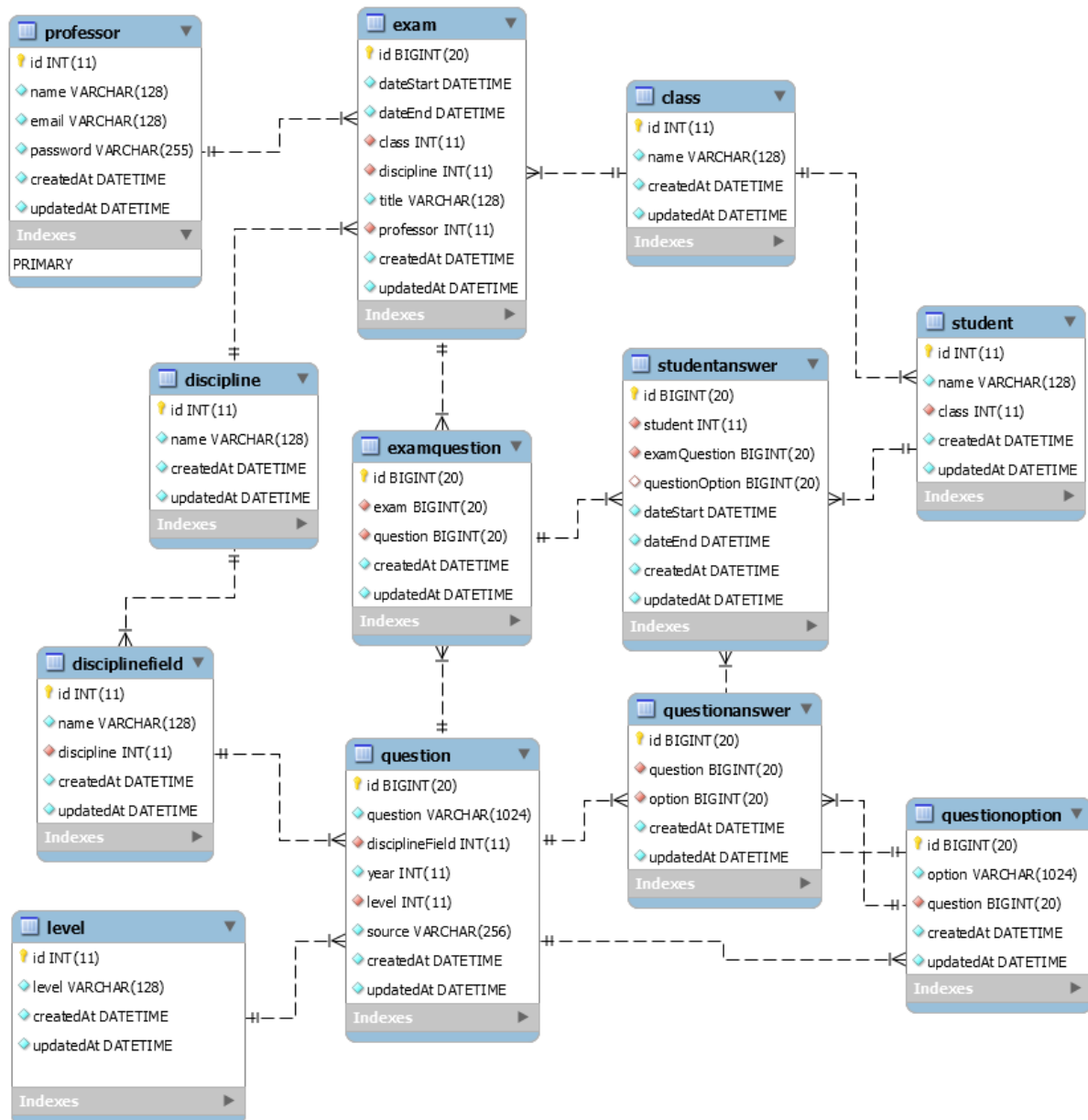
<sup>14</sup> Disponível em <<http://getbootstrap.com>> Acesso em 20 de Novembro.

<sup>15</sup> Disponível em <<https://github.com/axios/axios>> Acesso em 20 de Novembro.

## 5.4 Modelagem e Criação do Banco de Dados

Para o banco de dados do sistema utilizou-se o MySQL. Essa escolha deve-se aos fatos apontados na Seção 4.2.2 e à disponibilização da ferramenta gratuitamente. As tabelas do banco de dados foram implementadas de acordo com Modelo de entidade-relacionamento (Figura 7).

Figura 7 – Modelo de entidade-relacionamento



Fonte: elaborado pelo autor

## 5.5 Implementação do Sistema

Finalizado o planejamento da arquitetura, do banco de dados e a seleção das tecnologias foi feito então o desenvolvimento da aplicação em si, começando primeiramente pelo *frontend*, definindo e desenvolvendo o leiaute. Após a interface do usuário atingir um nível de maturidade, foi dado início à implementação do *backend* e à finalização do sistema em si. Como o *frontend* não estava finalizado ainda, buscou-se no primeiro momento garantir que os requisitos já implementadas na interface do usuário também funcionassem no *backend*, acessando e gerenciando as informações no banco de dados. Após essa etapa, a implementação do sistema foi prosseguida com o desenvolvimento das duas partes do sistema em paralelo, garantindo que todas as funcionalidades estivessem funcionando como o esperado. Abaixo encontra-se a explicação e as metodologias utilizadas no *frontend* e no *backend*.

### 5.5.1 Implementação do Frontend

Para o desenvolvimento do *frontend*, foi utilizada a biblioteca React para implementar as interfaces do usuário, usando esta tecnologia para gerar o HTML e responder às ações de interação do usuário.

Como observado na Figura 6, foi utilizado a biblioteca Axios para fazer a comunicação com o *backend*. Esta escolha deve-se ao fato que este componente torna mais simples o envio de requisições HTTP ao servidor, sendo possível fazer este tipo de solicitação com apenas uma linha de código sem precisar recarregar a página.

Para facilitar a execução do projeto e garantir uma melhor experiência ao usuário, foi utilizado a biblioteca Bootstrap para a estilização da interface. Esse componente de código aberto foi criado por dois funcionários do Twitter<sup>16</sup> e possui elementos pré estilizados como botões, tabelas e listas.

### 5.5.2 Implementação do Backend

Na questão do *backend*, foi utilizado o ambiente JavaScript Node.js. As principais razões para a escolha dessa tecnologia se deve ao fato de ser um servidor baseado em JavaScript – a mesma linguagem usada no *frontend* - e de sua fácil configuração para iniciar um servidor,

---

<sup>16</sup> Disponível em <https://twitter.com>> Acesso em 30 de Novembro.

além de ser multiplataforma. Como apontado na Seção 4.2.2, este interpretador javascript apresenta um maior desempenho que as tecnologias tradicionais devido à sua implementação não bloqueante.

Para agilizar o desenvolvimento, foram utilizados dois *frameworks*: o Express para processar as requisições e rotas da API e o Sequelize<sup>17</sup> para as operações no BD, ambos os mais populares da categoria para a plataforma. O Express é um framework de código aberto projetado para construir aplicações web e APIs, que fornece uma camada de recursos fundamentais da aplicação web. Já o Sequelize é um *framework* que mapeia as tabelas do BD com classes, permitindo realizar operações através de métodos.

Para iniciar o desenvolvimento do servidor, foi utilizado um ferramenta geradora para criar a estrutura e os arquivos básicos do servidor. A ferramenta é fornecida pelo framework Express, e seu nome é express-generator<sup>18</sup>. Para instalar, basta utilizar o gerenciador de dependências do Node.js, o npm. Executando o comando “npm install express-generator -g”, a ferramenta será instalada globalmente e permitirá o uso da mesma em qualquer lugar do sistema. Para criar a estrutura básica do servidor, basta executar o comando “express nome”, onde nome seria o nome da pasta que deseja-se criar o servidor. Ao fim desta etapa, os arquivos necessários para iniciar o servidor já foram criados, precisando somente instalar as dependências do projeto, como o próprio *framework* Express. Para isto, basta executar o comando “npm install” que o gerenciador de dependências do Node.js instalará todos os pacotes necessários para que o servidor funcione corretamente. Feito isto, já é possível a executar o servidor utilizando o comando “node app.js”.

### 5.5.3 Testes

Os testes foram efetuados em conjunto com o desenvolvimento do sistema. Procurou-se efetuar os testes a cada adição de uma nova funcionalidade, garantindo assim que o comportamento desta fosse como o esperado. Ao final do projeto, foi executado um teste geral das funcionalidades do sistema, assegurando que todas as funcionalidades propostas no levantamento de requisitos não apresentassem erros.

---

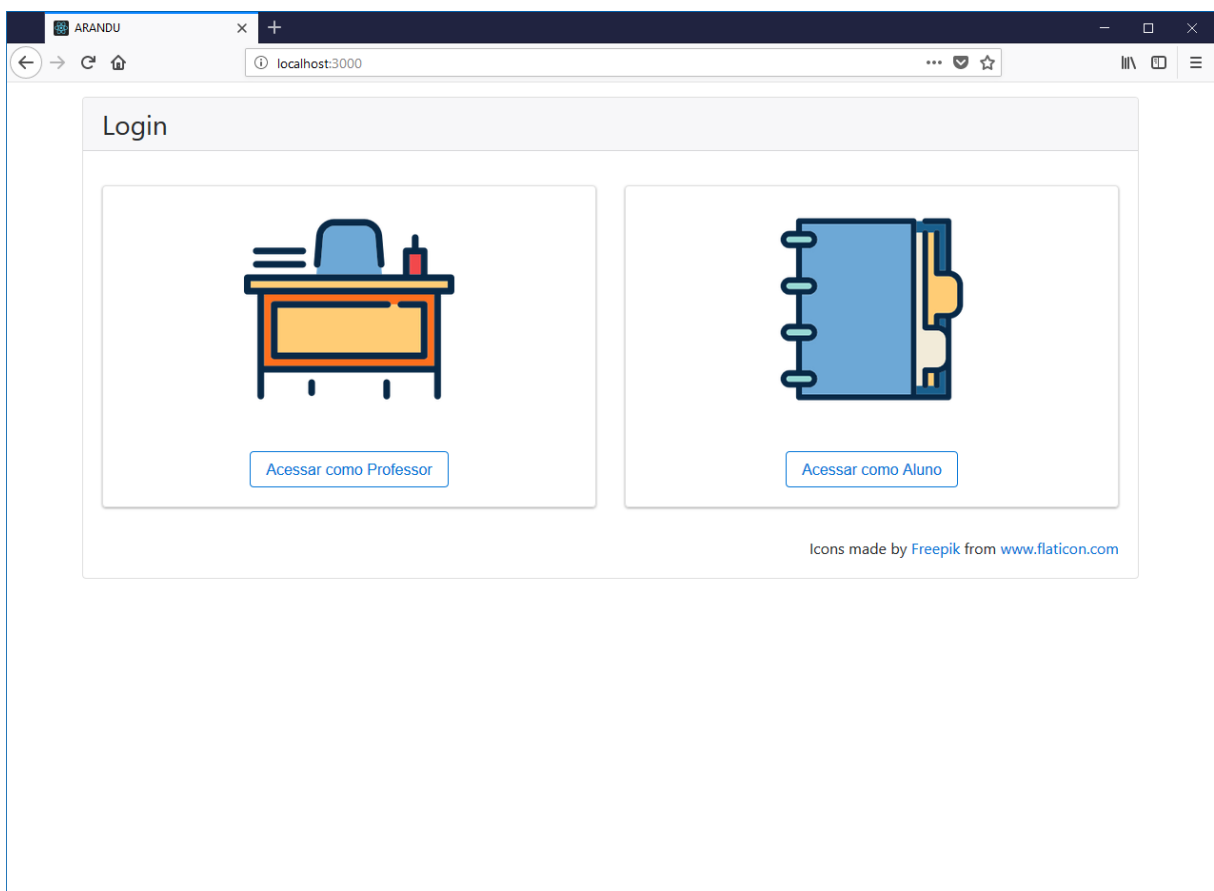
<sup>17</sup> Disponível em <<https://github.com/sequelize/sequelize>> Acesso em 20 de Novembro.

<sup>18</sup> Disponível em <<https://expressjs.com/en/starter/generator.html>> Acesso em em 29 de Novembro.

## 6 RESULTADOS

Entendido os problemas, as tecnologias disponíveis e as motivações do autor, é possível então apresentar o projeto proposto. Da Figura 8 à Figura 16 serão exibidos algumas telas do sistema de acordo com as funcionalidades identificadas no levantamento de requisitos. Para identificar o usuário, foi criado uma tela de login. Ao acessar o sistema pela primeira vez, o professor e o aluno acessarão a mesma página, exibida na Figura 8.

Figura 8 – Página de Login



Fonte: elaborado pelo autor

Como pode-se observar na Figura 6, o acesso do aluno e do professor estão separados, pois tanto os dados para entrar no sistema quanto as funcionalidades são distintas. Por este motivo, dividiu-se esta seção para os dois tipos de usuários.

## 6.1 Fluxo do Aluno

O aluno poderá realizar basicamente três operações no sistema:

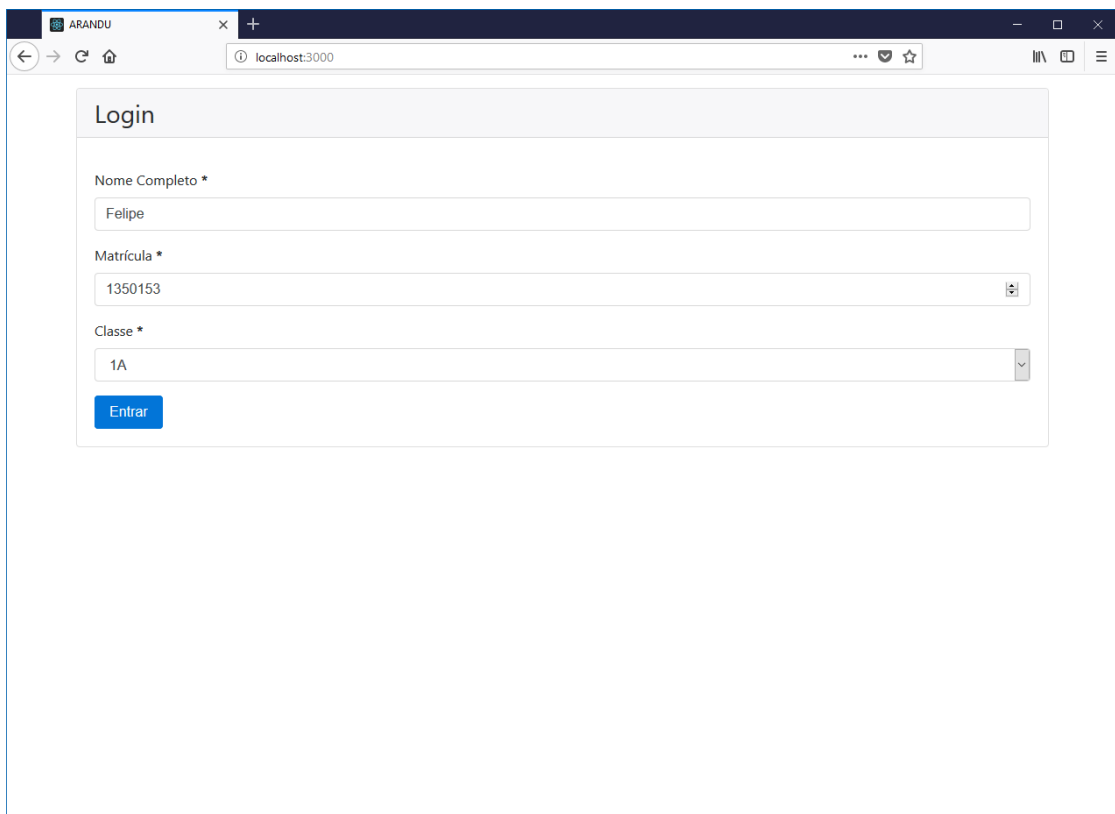
- Autenticar-se;
- Consultar exames anteriores;
- Realizar exames disponíveis.

Essas ações encontram-se descritas e detalhadas nas subseções seguintes.

### 6.1.1 Login

Esta etapa inicial é onde o aluno é identificado no sistema. Por questões de facilidade de gerenciamento e cadastro, qualquer aluno pode acessar o sistema apenas com suas informações básicas. Assim um professor não precisa cadastrar nenhum novo aluno, pois qualquer estudante que fizer a autenticação no sistema terá seus dados armazenados ou atualizados. Ao realizar a autenticação no sistema, o aluno é então redirecionado para a tela inicial do sistema. A página de *login* do aluno pode ser observada na Figura 9.

Figura 9 – Página de Login do Aluno



A imagem mostra uma captura de tela de um navegador web. No topo, a barra de endereço indica o endereço 'localhost:3000'. O conteúdo principal da página é um formulário de login com o título 'Login' em um cabeçalho cinza. O formulário contém três campos de entrada obrigatórios, marcados com um asterisco (\*): 'Nome Completo \*' com o valor 'Felipe', 'Matrícula \*' com o valor '1350153', e 'Classe \*' com o valor '1A'. Abaixo dos campos, há um botão azul com o texto 'Entrar'.

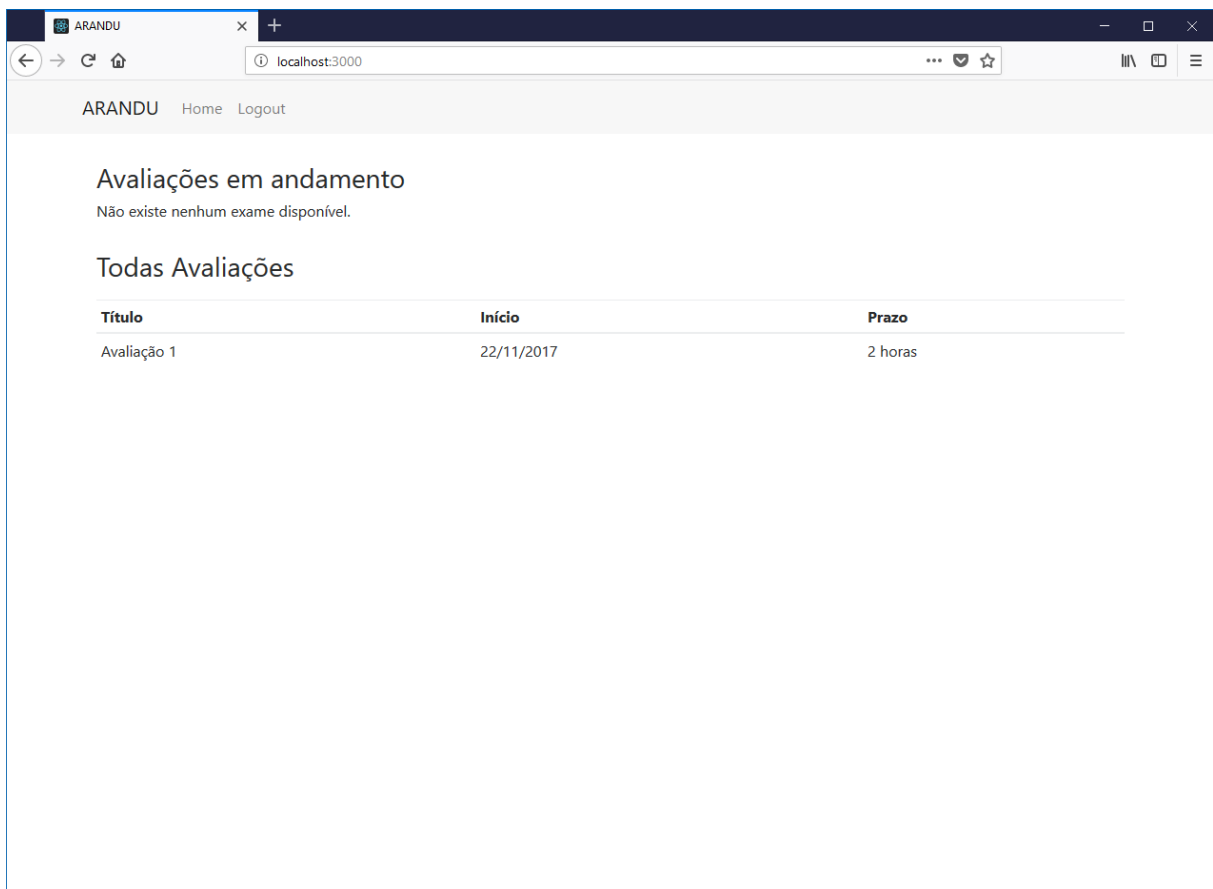
Fonte: elaborado pelo autor



### 6.1.2 Página Inicial

Após a autenticação, é exibida a página inicial do sistema com todas as avaliações disponíveis para serem efetuadas e as já realizadas pelo aluno, conforme observado na Figura 8. Ao clicar no título de alguma avaliação disponível, o aluno é então redirecionado para a tela onde será realizada a avaliação.

Figura 10 – Página Inicial do Aluno

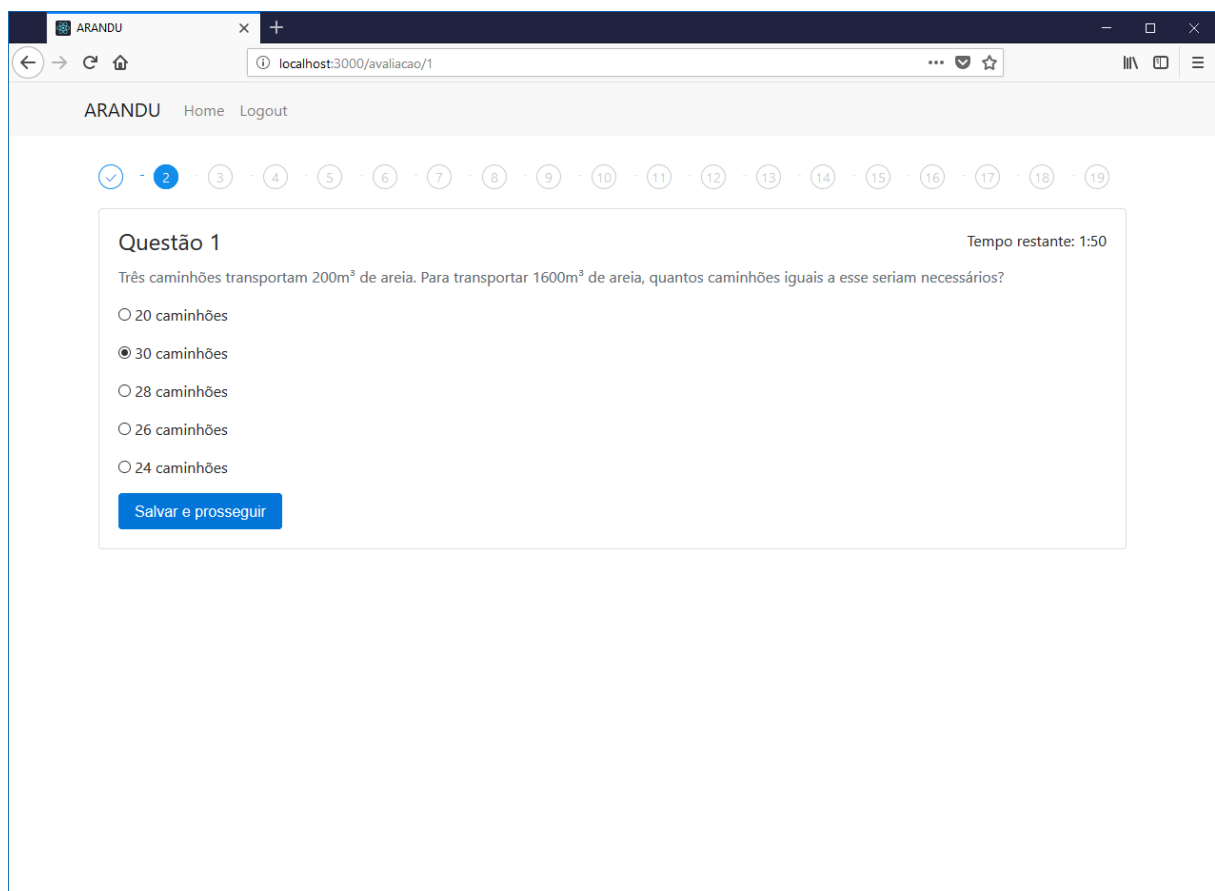


Fonte: elaborado pelo autor

### 6.1.3 Avaliação

Chegando nesta página, o aluno é submetido às questões cadastradas previamente pelo professor. Para manter seu foco, uma questão é exibida por vez e só é possível prosseguir para a próxima questão após finalizar a atual. No canto superior direito é exibido o tempo restante para finalizar o exame, o qual é cadastrado pelo professor quando ele está criando a avaliação. Na Figura 11 pode-se observar esta página.

Figura 11 – Página de Avaliação



Fonte: elaborado pelo autor

## 6.2 Fluxo do Professor

O professor realizará uma quantidade maior de operações do que o estudante. Dentre elas, pode-se destacar:

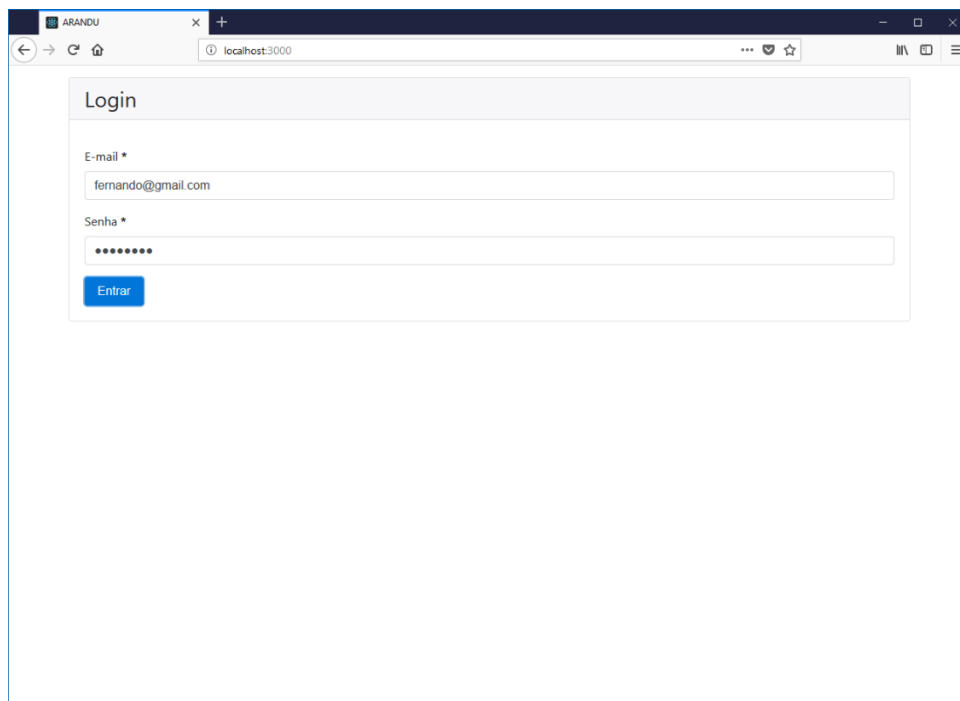
- Autenticar-se;
- Consultar exames anteriores;
- Criar novos exames;
- Detalhar o desempenho dos aluno nas avaliações;
- Cadastrar uma nova classe;

Essas ações encontram-se descritas e detalhadas nas próximas subseções.

### 6.2.1 Login

Esta etapa inicial é onde o professor será identificado no sistema. O sistema já tem um cadastro prévio de alguns professores, que precisam desta informações para acessar o sistema. O docente precisa estar cadastrado no sistema com seu e-mail e senha, conforme Figura 12. Caso ele não esteja cadastrado, outro professor poderá realizar o cadastro. Ao realizar a autenticação no sistema, o professor é então redirecionado para a tela inicial do sistema.

Figura 12 – Página de Login do Professor



A imagem mostra uma janela de navegador com o endereço 'localhost:3000'. O formulário de login contém os seguintes elementos:

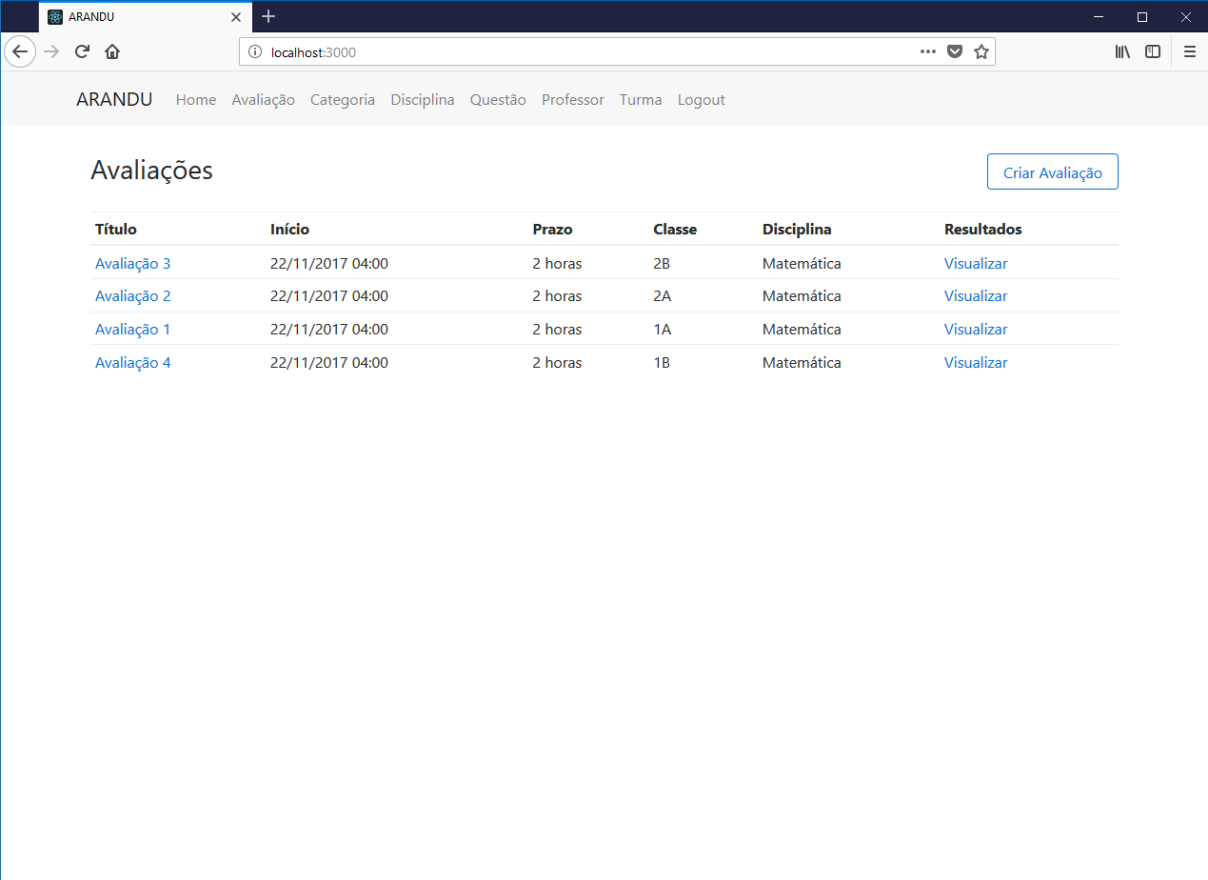
- Um campo de entrada para 'E-mail \*' com o valor 'fernando@gmail.com'.
- Um campo de entrada para 'Senha \*' com caracteres ocultos por pontos.
- Um botão azul com o texto 'Entrar'.

Fonte: elaborado pelo autor

### 6.2.2 Página Inicial

Após a autenticação, é exibida a página inicial do sistema com todas as avaliações já cadastradas pelo professor, como pode ser observado na Figura 13. Ao clicar no título de alguma avaliação, é possível visualizar o exame e suas respectivas questões. Ao clicar na última coluna, o professor é então redirecionado para a tela de detalhar os resultados da avaliação.

Figura 13 – Página Inicial do Professor



Título	Início	Prazo	Classe	Disciplina	Resultados
<a href="#">Avaliação 3</a>	22/11/2017 04:00	2 horas	2B	Matemática	<a href="#">Visualizar</a>
<a href="#">Avaliação 2</a>	22/11/2017 04:00	2 horas	2A	Matemática	<a href="#">Visualizar</a>
<a href="#">Avaliação 1</a>	22/11/2017 04:00	2 horas	1A	Matemática	<a href="#">Visualizar</a>
<a href="#">Avaliação 4</a>	22/11/2017 04:00	2 horas	1B	Matemática	<a href="#">Visualizar</a>

Fonte: elaborado pelo autor

### 6.2.3 Detalhar Avaliação

Ao detalhar a avaliação, o professor já consegue um relatório com a quantidade de alunos que acertaram determinada questão, podendo identificar as dificuldades dos alunos por área dentro da disciplina, conforme Figura 14.

Figura 14 – Página de Desempenho

Questão	Categoria	Dificuldade	Qtd. Respostas	Qtd. Acertos	Tempo Médio
1	1	1	0	0	-
2	1	1	0	0	-
3	1	1	0	0	-
4	1	1	0	0	-
5	2	1	0	0	-
6	2	1	0	0	-
7	2	1	0	0	-
8	2	1	0	0	-
9	3	1	0	0	-
10	3	1	0	0	-
11	3	1	0	0	-
12	3	1	0	0	-
13	3	1	0	0	-
14	3	1	0	0	-
15	3	1	0	0	-
16	3	1	0	0	-
17	3	1	0	0	-
18	3	1	0	0	-

Fonte: elaborado pelo autor

### 6.2.4 Criar Avaliação

Nesta página é possível criar uma avaliação para determinada classe de estudantes. O cadastro foi dividido por etapas para facilitar a criação de um novo exame. Na primeira etapa são mostradas as informações básicas, como o título, a disciplina, a classe e a data de início e fim. Nesta etapa o professor escolhe também o número de seções, que consiste nas questões agrupadas por área, dificuldade, fonte e ano. Então o professor pode criar uma prova com múltiplas seções, mesclando todos esses parâmetros, como observado na Figura 15.

Figura 15 – Página de Criar Avaliação

ARANDU Home Avaliação Categoria Disciplina Questão Professor Classe Logout

## Criar Avaliação

1 Início 2 Seção 1 3 Seção 2 4 Seção 3

Número de questões insuficiente. Existem 4 questões cadastradas com os parâmetros selecionados.

Número de Questões da Seção \*

5

Categoria \*

Matemática Regra de Três

Dificuldade

SELECIONE

Fonte

SELECIONE

Ano das Questões

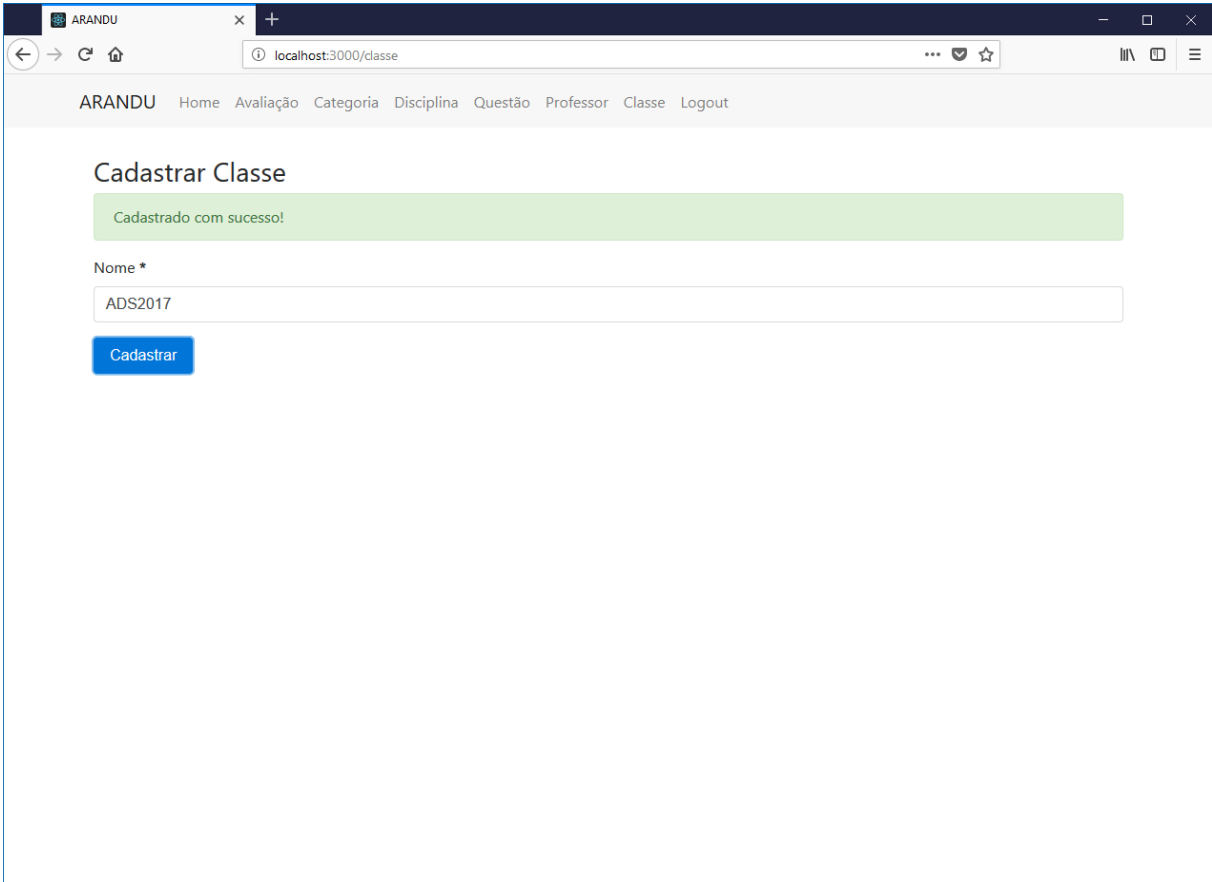
Próximo

Fonte: elaborado pelo autor

### 6.2.5 Página onde se cadastram Classes

Dentre todos os cadastros disponíveis, este é um dos expostos neste documento. Já que os alunos não precisam de cadastro para acessar o sistema, precisa-se pelo menos de um cadastro prévio da classe do indivíduo. Permitindo que o aluno digite este dado, não há garantias que exista uma padronização de como este será inserido, gerando a mesma informação com diferentes valores. Na Figura 16 encontra-se a página de cadastrar classe.

Figura 16 – Página de Cadastrar Classe



The screenshot displays a web browser window with the URL `localhost:3000/classe`. The page title is "Cadastrar Classe". A green success message "Cadastrado com sucesso!" is visible. Below it, there is a form with a label "Nome \*" and a text input field containing the value "ADS2017". A blue button labeled "Cadastrar" is positioned below the input field. The browser's navigation bar shows the site name "ARANDU" and a menu with items: Home, Avaliação, Categoria, Disciplina, Questão, Professor, Classe, and Logout.

Fonte: elaborado pelo autor

## 7 CONCLUSÃO

Diante do objetivo proposto do projeto, ele foi atingido com sucesso. Quanto ao uso do sistema no processo de avaliação de alunos, mostra-se bem promissor para ajudar os professores na aplicação e análise de exames.

Com o desenvolvimento do sistema, adicionou-se um método auxiliar para que os professores possam avaliar os alunos com a ajuda da tecnologia, visando assim um melhor aproveitamento do tempo dos professores, uma vez que o sistema busca agilizar as tarefas realizadas pelos docentes, muitas vezes repetitivas.

É em busca da otimização do tempo dos professores que o projeto foi proposto. Caso utilizado na prática, ele proporcionará uma grande agilidade na avaliação de alunos que, futuramente, poderia ser ampliado para um sistema de uso por todas as universidades que tenham interesse em otimizar seus processos, neste caso, especificamente no que tange à avaliação de alunos.

Durante a elaboração do projeto, surgiram algumas ideias e sugestões que visam aprimorar o sistema e garantir uma solução mais completa. Ao apresentar o projeto para professor Fernando, indentificaram-se algumas funcionalidades que poderiam ser incorporadas ao projeto em um trabalho futuro. Dentre elas, pode-se citar:

- Identificar quando o aluno abandonou o sistema durante a prova;
- Permitir que o aluno classifique a dificuldade das questões da avaliação;
- Possibilidade de efetuar a avaliação *offline*, precisando estar *online* somente para enviar as respostas.
- Exportar as questões e provas para o Moodle, *software* utilizado na instituição como apoio à aprendizagem;
- Gerar estatísticas da avaliação;
- Efetuar testes do sistema em casos reais e com pedagogos.

Com a conclusão deste projeto, é correto afirmar que foi proporcionado um aprofundamento técnico em desenvolvimento web, exigindo a habilidade de aprender novas metodologias e novas tecnologias.



## REFERÊNCIAS

CHAPPELL, D. **The Three Aspects of Software Quality**: functional, structural, and process. San Francisco, California. Disponível em <[http://www.davidchappell.com/writing/white\\_papers/The\\_Three\\_Aspects\\_of\\_Software\\_Quality\\_v1.0-Chappell.pdf](http://www.davidchappell.com/writing/white_papers/The_Three_Aspects_of_Software_Quality_v1.0-Chappell.pdf)> Acesso em: 12 novembro 2017.

CODD, E. Relational Model of Data Large Shared Data Banks. **Communications of the ACM**, Estados Unidos, v. 13, p. 377-387, 1970.

FIELDING, R. T.; TAYLOR, R. Principled Design of the Modern Web Architecture. **ACM Transactions on Internet Technology (TOIT)**, Nova York. Volume 2, 2002.

FREDERIKSEN, J.; COLLINS, A. A Systems Approach to Educational Testing. **Educational Researcher**, v. 18, p. 27-32, 1989.

GATTI, B. **O Professor e a Avaliação em Sala de Aula**. São Paulo, 2003.

HUNT, P. **Why did we build React?** Disponível em: <<https://reactjs.org/blog/2013/06/05/why-react.html>> Acesso em: 28 setembro 2017a.

HUNT, P. **React: rethinking best practices**. Disponível em <<https://2013.jsconf.eu/speakers/pete-hunt-react-rethinking-best-practices.html>> Acesso em: 01 novembro 2017b.

LOWE, D. Web system requirements: an overview. **Requirements Engineering**, v. 8, p. 102-113, 2003.

LUCKESI, C. **Avaliação da aprendizagem escolar**: estudos e proposições. 22.ed. São Paulo: Cortez Editora, 2011.

MIKOWSKI M.; POWELL J. **Single Page Web Applications**: JavaScript end-to-end. Edição 1. Manning Publications: 30 de Setembro de 2013. 407 páginas.

MORAN, J. **A Educação que desejamos**: novos desafios e como chegar lá. 5.ed. Campinas: Papirus, 2013, p. 89-90.

NIELSEN, J. **10 Usability Heuristics for User Interface Design**. San Francisco, California. Disponível em <<https://www.nngroup.com/articles/ten-usability-heuristics/>> Acesso em: 10 outubro 2017.

SOUZA S.; ANQUETIL N.; OLIVEIRA K. **A study of the documentation essential to software maintenance**. Coventry, United Kingdom: Proceedings of the 23rd annual international conference on Design of communication: documenting & designing for pervasive information. p. 68-75, 2005.

TILKOV, S.; VINOSKI, S. Node.js: Using JavaScript to Build High-Performance Network Programs. **IEEE Internet Computing**, v. 14, p. 80-83, 2010.

VUKSANOVIC I. P., SUDAREVIC B. **Use of Web Application Frameworks in the Development of Small Applications**. Opatija, Croatia: 34th International Convention MIPRO, 2011.