

Faculdades Integradas de Taquara - Faccat

Av. Oscar Martins Rangel, 4.500

Taquara, RS, CEP 95600-000

Curso de Sistemas de Informação

MINHAGRADE - SISTEMA WEB DE GESTÃO E PLANEJAMENTO DE GRADE CURRICULAR

Rodrigo Krummenauer do Nascimento

Faculdades Integradas de Taquara – Faccat – Taquara – RS – Brasil

rodrigoknascimento@gmail.com

Prof. Dr. Marcelo Cunha de Azambuja

Professor Orientador

Faculdades Integradas de Taquara – Faccat – Taquara – RS – Brasil

marcelo.azambuja@gmail.com

Resumo

Este artigo tem a finalidade de demonstrar as etapas de análise e desenvolvimento de uma aplicação web e móvel denominada MinhaGrade. O objetivo desta ferramenta é facilitar o planejamento e a seleção das disciplinas durante rematrículas pelos alunos, tornar o processo mais sociável através de compartilhamento de informações entre alunos e auxiliar as instituições de ensino a identificar os interesses dos seus alunos. A aplicação surgiu com o intuito de agilizar os processos de rematrícula que acontecem a cada semestre.

Palavras-chave: aplicativo, web, mobilidade, rematrícula, social.

MINHAGRADE - WEB SYSTEM FOR PLANNING AND MANAGEMENT OF CURRICULAR GRADING

Abstract

This article aims to demonstrate the analysis and development steps of a web and mobile application called MinhaGrade. The purpose of this tool is to facilitate the planning and the selection of classes during enrollment by students, to make the process more sociable by sharing information among students and to help educational institutions to identify the interests of their students. The application arose with the purpose of speeding up the re-registration processes that happen every semester.

Key-words: *application, web, mobile, re-enrollment, social*

1 INTRODUÇÃO

Este artigo consiste na criação de uma aplicação *web* e móvel capaz de auxiliar alunos e instituições de ensino a organizar, visualizar e interagir com a grade curricular do curso. Com o objetivo de facilitar o planejamento das disciplinas a serem selecionadas durante rematrículas, prevenindo erros como: planejar a matrícula em disciplinas bloqueadas por pré-requisitos.

Além de auxiliar na organização da grade curricular, esta aplicação tem por objetivo possibilitar o compartilhamento entre amigos do progresso de cada aluno no curso, bem como as escolhas de disciplinas de cada colega no momento da rematrícula, entre outras informações.

Manter o aluno atualizado das mudanças na grade curricular semestral também é um dos objetivos deste projeto, avisando o aluno por e-mail ou *Push Notifications*¹ sobre alterações tais como troca sala ou de professor.

Não existe atualmente um sistema de controle curricular padrão para instituições de ensino. Em geral cada instituição faz uso de um sistema próprio que pode ou não trazer facilidades aos alunos no hora de verificar e gerenciar seus currículos e planejamentos para os semestres seguintes. Na maioria dos casos as instituições provém três sistemas distintos para este fim:

- a) Um sistema para listagem do currículo em andamento com marcação das disciplinas cursadas;
- b) Um sistema para listagem das matérias disponíveis para o semestre;
- c) Um sistema para rematrículas.

A separação entre sistemas de "listagem de andamento de currículo" e "matérias disponíveis para o semestre" gera a necessidade da comparação de informações de ambos os sistemas para definição de quais matérias estão disponíveis para rematrícula.

¹ *Push Notification*: *Push Notification* é uma mensagem que é enviada aos aparelhos móveis de forma direta sem que o aparelho precise buscar pelas mesmas, comumente usadas para informar os usuários quando estes não estão fazendo uso do aparelho ou com a aplicação, que originou o *Push Notification*, fechada (AIRSHIP, 2018)

Normalmente a própria listagem do currículo não faz uma boa implementação de usabilidade, utilizando apenas códigos para listar dependências entre matérias, exigindo que o aluno faça uma busca visual N vezes para cada matéria onde N é igual a quantidade de dependências daquela matéria.

Também é possível citar que, normalmente, os sistemas não possuem funcionalidades úteis aos alunos, tais como:

- a) Interação social permitindo comparação de currículos e planejamento de matérias entre colegas;
- b) Sistema para consulta de interesse em matérias para o semestre seguinte;
- c) Compatibilidade com dispositivos móveis;
- d) Alertas via *Push Notifications* sobre alterações na grade semestral ou curricular.

Os problemas citados anteriormente podem levar os alunos a cometerem equívocos no momento de suas rematrículas, gerando custos e burocracias desnecessárias a curto prazo e atrapalhando os planejamentos tanto da instituição de ensino como dos seus alunos.

O artigo está organizado como segue: a seção 2 apresenta os principais conceitos e tecnologias utilizadas para o desenvolvimento; a seção 3 apresenta a metodologia; A aplicação desenvolvida é apresentada na seção 4; Finalmente, a seção 5 apresenta as conclusões.

2. CONCEITOS E TECNOLOGIAS

2.1 Node.js

Node.js é um ambiente de execução *open-source*, assíncrono e baseado em eventos para a linguagem JavaScript (ABOUT, 2018). É considerado um *framework*² para desenvolvimento de alta performance para aplicações concorrentes que não utiliza o método tradicional de *multithreads*, pois utiliza I/O assíncrono com um modelo de programação dirigida a eventos (TILKOV; VINOSKI, 2010).

² Framework: Framework é uma estrutura conceitual com o objetivo de servir de suporte ou guia para a construção de algo que expanda a estrutura em algo útil (ROUSE, 2015).

2.2 JavaScript

JavaScript é uma linguagem leve, interpretada e baseada em objetos mais conhecida como linguagem de *script* para páginas Web, mas, atualmente, usada em outros ambientes como Node.js. Suporta estilos de programação orientado a objetos, imperativo e funcional (MOZILLA DEVELOPER NETWORK, 2018).

2.3 GraphQL

GraphQL é uma linguagem de busca para Web Services e um ambiente de execução para responder essas buscas com dados existentes. Provê ao cliente o poder de pedir pelo que ele precisa de forma precisa, simplificando a manutenção de APIs³ (MASSE, 2012) ao longo do tempo e permitindo o uso de ferramentas de auxílio (GRAPHQL, 2018).

Toda a comunicação entre o cliente e servidor foi feita via API utilizando GraphQL onde o padrão de resposta utilizado foi JSON⁴, um padrão amplamente conhecido, de fácil leitura e compreensão.

2.4 Meteor

Meteor é um sistema de compilação para aplicações JavaScript que permite compartilhamento de código entre o cliente e servidor, diminuindo assim a repetição de código e facilitando a implementação de testes.

Por padrão, Meteor implementa a comunicação em tempo real via WebSocket nativamente e sem necessidade de configuração, permitindo que o servidor possa enviar dados ao cliente a qualquer momento sem que o cliente precise pedir a informação ao servidor.

Meteor implementa internamente uma tecnologia conhecida como Fibers tornando o código mais limpo e fácil de manter (HOCHHAUS; SCHOEBEL, 2015).

2.5 React

React é uma biblioteca JavaScript *open source* para construção de interfaces, desenvolvida e mantida pelo Facebook. Permite a renderização dinâmica de componentes e

³ API: Application Program Interface. API é um conjunto de métodos, serviços e protocolos que permitem a integração entre sistemas diversos (API, 2018).

⁴ JSON: JavaScript Object Notation: JSON é um formato de troca de dados leve de fácil leitura e escrita por humanos e de fácil codificação e decodificação por máquinas (JSON, 2016)

páginas no navegador web e no servidor, e tem sido adotada por diversos projetos por se mostrar facilmente testável e performática (HUNT, 2016).

2.6 React Native

React Native é um framework JavaScript *open source* para construção de aplicações nativas para iOS e Android desenvolvida e mantida pelo Facebook o qual usa a biblioteca React como linguagem de construção de interfaces gerando componentes nativos ao invés de componentes Web (EISENMAN, 2016).

2.7 Web Services

Web Service é uma interface que descreve uma coleção de operações que são acessíveis através da internet padronizados em JSON ou algum tipo de padrão de informações conhecido, desde que todas as operações sigam o mesmo padrão. Um Web Service deve providenciar todos os meios para a comunicação da aplicação com o servidor, tornando a comunicação padrão e centralizada para todas as aplicações do serviço (GOTTSCHALK; GRAHAM, 2002).

Para o desenvolvimento dos Web Services, foram utilizadas diversas tecnologias em conjunto, proporcionando assim um sistema escalável e mais seguro. Foi utilizado o padrão JSON em seus serviços para que as mesmas pudessem ser utilizadas por todos os softwares que utilizam a plataforma, seja ele um aplicativo ou um servidor, fornecendo ou pedindo informações.

2.8 Infraestrutura

Como infraestrutura, optou-se pela utilização de uma hospedagem PaaS⁵ chamada Heroku para a aplicação de demonstração onde a infraestrutura é contratada por hora de uso e o *deployment*⁶ pode ser feito de forma automática a cada alteração no código fonte enviada à hospedagem do código.

Para instalação direta é recomendado o uso de um servidor de *proxy* reverso⁷ como camada de conexão com a internet, a aplicação foi testada utilizando o servidor *proxy* Nginx,

⁵ PaaS: PaaS ou Plataforma como Serviço (*Platform as a Service*) é um ambiente de desenvolvimento e implantação completo na nuvem onde é possível alugar o serviço de hospedagem de um software sem a necessidade da configuração da infraestrutura (MICROSOFT, 2018).

⁶ *Deployment*: *Deployment* é o ato de enviar um software para o ambiente de produção ou testes.

⁷ *Proxy* reverso: O servidor *proxy* reverso protege servidores HTTP, fornecendo um único ponto de acesso à rede interna (IBM, 2018).

criado em 2002 com o objetivo de servir sites com um enorme volume de acessos. É um software bastante leve e utilizado atualmente em aplicações de grande escala (NEDELUCU, 2010).

Apesar do framework Node.js possuir um servidor HTTP interno, é recomendado que se utilize o servidor Nginx como *proxy* reverso, permitindo a escalabilidade do processo, cache, e um melhor performance quando utilizando SSL (SMITH, 2015).

2.9 iOS

iOS é um sistema operacional para dispositivos móveis desenvolvido pela Apple. Não é permitida a instalação do iOS em hardware de terceiros para garantir mais segurança e performance. A versão estável corrente é 12.1 (APPLE, 2018).

2.10 Android

Android é um sistema operacional para dispositivos móveis desenvolvido pelo Google com o código-fonte e baseado em no sistema operacional Linux. O Android está, atualmente, instalado em centenas de milhões de dispositivos em mais de 190 países ao redor do mundo. É a maior plataforma de dispositivos móveis do mundo e crescendo rapidamente. A versão estável corrente é 9.0 (ANDROID, 2018).

2.11 White Label

White Label é a definição que se dá a customização de aplicações genéricas para aplicação da marca de uma empresa fazendo com que a aplicação pareça ter sido desenvolvida especificamente para a empresa em questão (NIKOGHOSYAN, 2018).

A solução descrita neste artigo tem por objetivo a entrega de um sistema White Label onde as instituições devem fazer suas customizações de marca e lançar os aplicativos como solução própria bem como hospedar o sistema por conta da instituição não visando a oferta de um sistema SAS.

3 METODOLOGIA

A *Extreme Programming (XP)*⁸ foi escolhida como metodologia para o desenvolvimento deste projeto. Por ser uma metodologia ágil, focada em equipes pequenas e médias que desenvolvem aplicações baseadas em requisitos, seu conjunto de valores, princípios e práticas oferecem maior agilidade e qualidade para atingir as metas pré-estabelecidas e assim entregar um produto robusto e confiável.

A metodologia foi aplicada usando as principais práticas propostas pela XP, como Planejamento, Projeto Simples, Entregas Frequentes, Metáfora, Código Padrão e Testes. A escolha adequada do framework de desenvolvimento trouxe ganhos em relação à simplicidade do projeto, visto que o reaproveitamento de código, assim como sua padronização, foram estendidas às novas implementações e funcionalidades.

Com a conclusão das especificações dos requisitos, focou-se, em cada interação, em manter as entregas previstas no planejamento, as quais foram baseadas nos requisitos da interação, sem considerar requisitos futuros permitindo alcançar maior agilidade nas entregas sem ocasionar aumento no escopo.

Cada versão foi definida com o menor tamanho possível, contendo os requisitos de maior valor para a aplicação. A maior frequência de entrega tornou mais rápido o feedback por parte dos usuários.

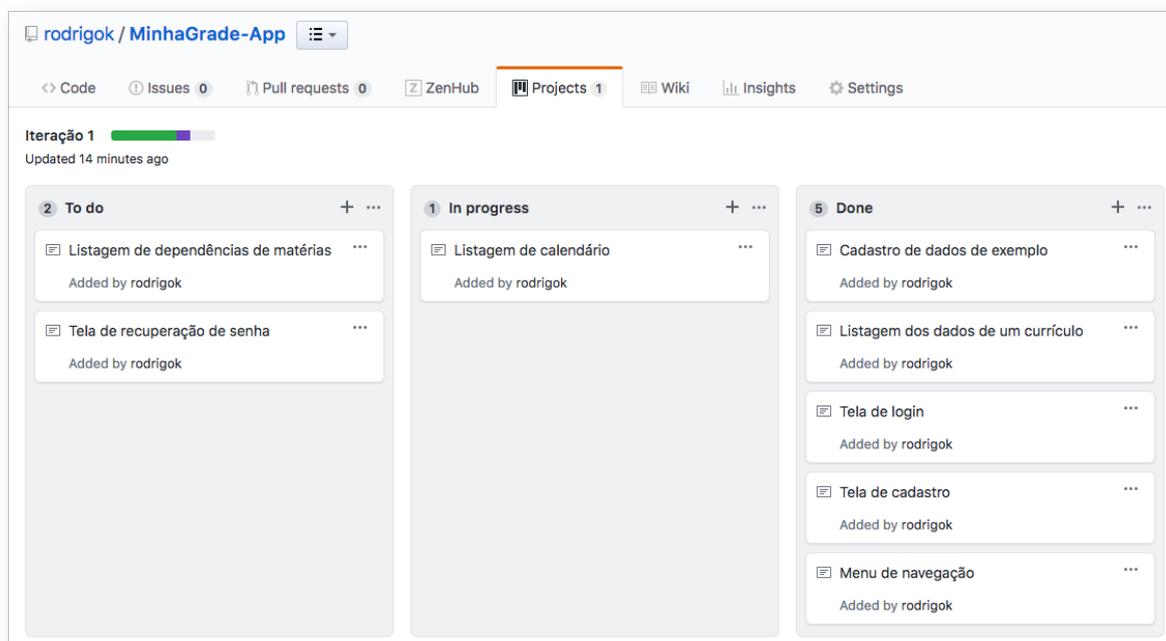
Os testes de cada interação foram feitos de forma manual seguindo um roteiro de uso da aplicação para a garantia das funcionalidades.

Para controle das tarefas de cada interação foi utilizado a funcionalidade *projects* (projetos) de cada um dos repositórios hospedados no GitHub⁹ como demonstrado na Figura 1. As tarefas foram separadas em projetos por interação onde cada projeto possui as colunas *to do* (à fazer), *in progress* (em progresso) e *done* (concluído) para o controle de status de cada tarefa.

⁸ Extreme Programming (XP) é uma metodologia de desenvolvimento de software, nascida nos Estados Unidos ao final da década de 90. Vem fazendo sucesso em diversos países, por ajudar a criar sistemas de melhor qualidade, que são produzidos em menos tempo e de forma mais econômica que o habitual. Tais objetivos são alcançados através de um pequeno conjunto valores, princípios e práticas, que diferem substancialmente da forma tradicional de se desenvolver software. Fonte: DESENVOLVIMENTO ÁGIL (2016)

⁹ GitHub: É uma plataforma web para hospedagem e versionamento de código fonte de softwares, incluindo ferramentas de gerenciamento de projetos (GITHUB, 2018).

Figura 1 – Demonstração do quadro de tarefas



Fonte – Autor

3.1 Análise

A fase de análise de requisitos teve como objetivo abstrair e definir quais seriam os recursos necessários para atender as principais características e funcionalidades do sistema proposto. Nesta etapa, optou-se por uma documentação simples, com foco no detalhamento e descrição do requisito, transformando-o em uma tarefa a ser implementada.

Os resultados obtidos no processo de análise foram transformados em dois tipos de requisitos, no Quadro 1 são listados os requisitos funcionais e no Quadro 2 os requisitos não funcionais.

Tabela 1 – Requisitos funcionais

Requisito funcional	Descrição
Login	Identificação do usuário perante email e senha
Cadastro	Criação de usuário perante informação de nome, email, curso e senha
Listagem de currículo	Listagem das matérias relacionadas ao currículo do aluno com nome, ementa, dependências, semestre, créditos e status
Listagem de calendário	Listagem das matérias disponíveis ao aluno com nome, dependências, professor, sala, semestre, quantidade de interessados, amigos interessados e status

Filtro de calendário	Filtro para a listagem de calendário permitindo listar apenas matérias disponíveis para serem cursadas, sendo um filtro para esconder matérias já cursadas ou em andamento e outro para esconder matérias bloqueadas por requisitos
Marcar interesse	Ação para marcar interesse em um dos itens do calendário
Notificações	Notificação aos alunos de alterações nas matérias marcadas com interesse

Fonte – Autor

Tabela 2 – Requisitos não funcionais

Requisito não funcional	Descrição
Usabilidade	Implementar interface de toda a aplicação com design responsivo, permitindo a adaptação visual em diversos tipos de dispositivos
Segurança	Utilizar camada adicional de segurança através do protocolo HTTPS/SSL
Disponibilidade	Utilizar infraestrutura capaz de manter a aplicação disponível a todos os usuários do sistema com disponibilidade, de pelo menos, 99% do tempo
Banco de dados NoSQL	Armazenar informações em banco de dados NoSQL, flexibilizando o processo de armazenamento de dados com estruturas heterogêneas

Fonte – Autor

Como regra de negócio da aplicação, foram definidos dois perfis de usuários, o aluno e administrador. Desta forma as funcionalidades do sistema são disponibilizadas conforme cada perfil, visando restringir o acesso dos alunos as funcionalidades de administração como gerenciamento de professores, cursos e calendários. Através da Figura 2 é possível visualizar o diagrama de caso de uso, descrevendo as funcionalidades e os atores da aplicação.

Através da Figura 4 é possível visualizar o diagrama de modelo de domínio contemplando conceitos e suas relações. É importante frisar que a aplicação faz uso de um banco de dados não relacional tornando certas definições de relacionamentos e tipos de dados menos tradicionais, como, por exemplo, o tipo de dado **CourseJSON**, que denomina um dado do tipo dicionário onde a chave é definida pelo código do curso e o valor pelo valor do campo como demonstrado na Figura 3.

Figura 2 – Diagrama de caso de uso



Fonte – Autor

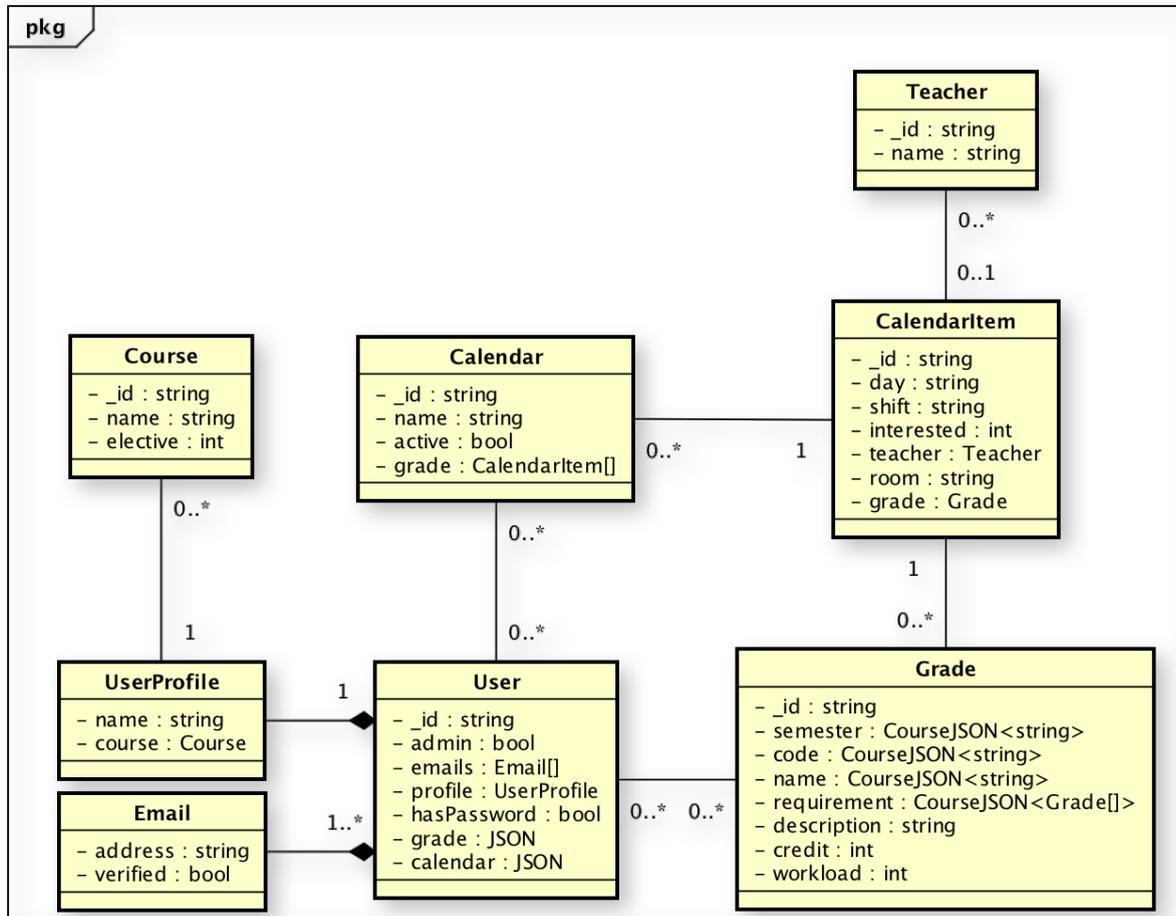
Figura 3 – Demonstração de um registro de grade

```

1  {
2  ..... "_id" : "CBckDJTRtNv7wvPmQ",
3  ..... "semester" : {
4  .....   "SI" : "1",
5  .....   "TSI" : "1"
6  ..... },
7  ..... "code" : {
8  .....   "SI" : "4132",
9  .....   "TSI" : "44174"
10 ..... },
11 ..... "name" : {
12 .....   "SI" : "Algoritmos",
13 .....   "TSI" : "Algoritmos e Programação"
14 ..... },
15 ..... "requirement" : {
16 .....   "SI" : [],
17 .....   "TSI" : []
18 ..... },
19 ..... "credit" : 4,
20 ..... "workload" : 60,
21 ..... "description" : "Conceitos básicos de Algoritmos e Programação, ..."
22 }
    
```

Fonte – Autor

Figura 4 – Diagrama de domínio



Fonte – Autor

4 A APLICAÇÃO DESENVOLVIDA

O trabalho de pesquisa e desenvolvimento em questão originou 4 artefatos, uma aplicação servidor como Web Service na nuvem, uma aplicação web cliente para acesso pelos alunos e administradores, um aplicativo móvel nativo ao sistema operacional iOS e um aplicativo móvel nativo ao sistema operacional Android. Toda a solução foi chamada de MinhaGrade, ela resultou em uma solução simples e prática visualização e gerenciamento de grade curricular e calendário de rematrícula. Esta seção busca apresentar algumas de suas funcionalidades disponíveis.

4.1 Detalhes gerais do sistema

Para acesso ao sistema o aluno ou administrador precisa se autenticar para sua identificação, essa autenticação pode ser feita mediante email e senha ou através de

autorização via Facebook. Para tal o usuário precisa fazer um cadastro no sistema, autenticar via Facebook ou já ter seus dados importados via API para dentro do sistema.

Ser administrador consiste em uma permissão do sistema, sendo assim as formas de acesso são iguais para ambos os tipos de usuário onde a permissão apenas define quais recursos o usuário tem acesso.

Apenas a versão do cliente web possui os recursos de administração, as versões móveis são focadas na experiência do aluno visando facilidade de acesso a informação.

4.2 Interface Web

A interface web foi desenvolvida pensando no aluno e no administrador do sistema, onde o aluno pode visualizar a lista de matérias do seu curso bem como o status de cada matéria e a grade de matérias disponíveis para rematrícula do semestre seguinte. Já o administrador tem acesso aos cadastros internos de professores, cursos e calendários.

4.2.1 Funcionalidades para os alunos

Na tela “Meu Currículo” o aluno pode visualizar a lista de matérias disponíveis para o seu curso em formato de grade como demonstrado na Figura 5 na marcação 1. Cada matéria pode estar nos estados Pendente, Cursando ou Concluído, Figura 5 marcação 2. As dependências são listadas por código e podem estar nas cores Vermelha caso esta dependência ainda esteja pendente, Laranja caso esteja sendo cursada ou Cinza caso já tenha sido concluída. É possível visualizar o nome e semestre da dependência passando o cursor sobre o código da dependência como demonstrado na Figura 6.

É demonstrado também a ementa da disciplina ao se expandir a linha pelo símbolo de + (mais) como pode ser visto na Figura 5 marcação 4.

Na marcação 5 (Figura 5) é demonstrada a barra de progresso onde o aluno pode acompanhar o seu progresso no curso em um cálculo¹⁰ feito pela soma de disciplinas cursadas e em andamento. Na marcação 6 (Figura 5) é descrita a quantidade total de disciplinas, a quantidade de disciplinas sendo cursadas e a quantidade concluída bem como a soma usada na

¹⁰ O total de disciplinas é obtido pela contagem de disciplinas não eletivas do curso mais a quantidade de eletivas indicada no cadastro do curso. A quantidade cursada pelo aluno é obtida pela contagem de disciplinas não eletivas cursadas mais o menor valor entre a quantidade de eletivas indicada no cadastro do curso e a quantidade de eletivas cursadas.

barra de progresso. Também é descrita a quantidade de disciplinas eletivas requeridas e levadas em consideração no cálculo.

Figura 5 – Demonstração das funcionalidades básicas do menu “Meu Currículo” no cliente web

The screenshot shows the 'Meu Currículo' (My Curriculum) page. At the top, there is a navigation bar with 'Meu Currículo' and 'Calendário' tabs, and a user profile for 'Aluno rodrigoknascimento+aluno@gmail.com'. Below the navigation bar, there is a share link: 'Compartilhe seu currículo com amigos: https://faccat.minhagrade.com/shared/eJ2z7DZvFBjwhtFsH'. A progress bar shows '4 de 46 (2 concluídas + 2 cursando) Considerando 1 disciplina eletiva' with an 8% completion rate. The main content is a table of courses with columns for Status, Semestre / Código, Nome, Dependências, and Créditos / Carga Horária. The table lists several courses with their respective statuses (Concluído, Cursando, Pendente) and credit loads (4 / 60). A tooltip is visible over the 'Arquitetura de Computadores' course, showing its dependency on 'Introdução à Computação - Semestre 1'.

Status	Semestre / Código	Nome	Dependências	Créditos / Carga Horária
Concluído	1 / 4132	Algoritmos		4 / 60
Cursando	1 / 4133	Introdução à Computação		4 / 60
Concluído	1 / 4134	Lógica e Matemática Discreta		4 / 60
Cursando	1 / 4305	Matemática Fundamental		4 / 60
Pendente	1 / 4406	Português		4 / 60
Desenvolvimento das habilidades de compreensão, interpretação e produção dos textos oral e escrito.				
Pendente	1 / 4407	Metodologia Científica		4 / 60
Pendente	2 / 4138	Arquitetura de Computadores	4133	4 / 60
Pendente	2 / 4174	Programação I	4132	4 / 60
Pendente	2 / 4306	Matemática Financeira	4305	4 / 60

Fonte – Autor

Figura 6 – Detalhes das dependências de uma matéria

This close-up shows the dependency details for the 'Arquitetura de Computadores' course. It highlights that this course is dependent on 'Introdução à Computação - Semestre 1' (course 4133). The table shows the course is currently 'Pendente' (Pending) with a credit load of 4 / 60.

Pendente	1 / 4407	Metodologia Científica		4 / 60
Pendente	2 / 4138	Arquitetura de Computadores	4133	4 / 60

Fonte – Autor

A funcionalidade de compartilhamento de currículo se dá mediante o envio de uma URL de acesso privada para quem se deseja permitir o acesso, essa URL pode ser obtida no topo da aplicação (Figura 5 marcação 7). O sistema apresentará os dados do aluno para o terceiro limitando-o a apenas visualização dos mesmo bem como identificando o aluno aos quais os dados pertencem no tipo da tela (Figura 7).

Figura 7 – Demonstração do modo de compartilhamento de currículo

Meu Currículo | Calendário | Entrar / Criar Conta

Currículo compartilhado de: rodrigoknascimento+aluno@gmail.com (Sistemas de Informação)

4 de 46 (2 concluídas + 2 cursando) Considerando 1 disciplina eletiva 8%

Status	Semestre / Código	Nome	Dependências	Créditos / Carga Horária
Concluído	1 / 4132	Algoritmos		4 / 60
Cursando	1 / 4133	Introdução à Computação		4 / 60
Concluído	1 / 4134	Lógica e Matemática Discreta		4 / 60
Cursando	1 / 4305	Matemática Fundamental		4 / 60
Pendente	1 / 4406	Português		4 / 60
Pendente	1 / 4407	Metodologia Científica		4 / 60
Pendente	2 / 4138	Arquitetura de Computadores	4133	4 / 60
Pendente	2 / 4174	Programação I	4132	4 / 60
Pendente	2 / 4306	Matemática Financeira	4305	4 / 60
Pendente	2 / 4307	Probabilidade e Estatística	4305	4 / 60

Fonte – Autor

Na tela “Calendário” o aluno pode visualizar as disciplinas disponíveis para o calendário ativo, ou seja, para o calendário do semestre seguinte. Nesta tela as disciplinas são dispostas em forma de grade onde os dias são dispostos em colunas e os turnos são separados em linhas e as disciplinas são acumuladas verticalmente como demonstrado na Figura 8 marcação 1.

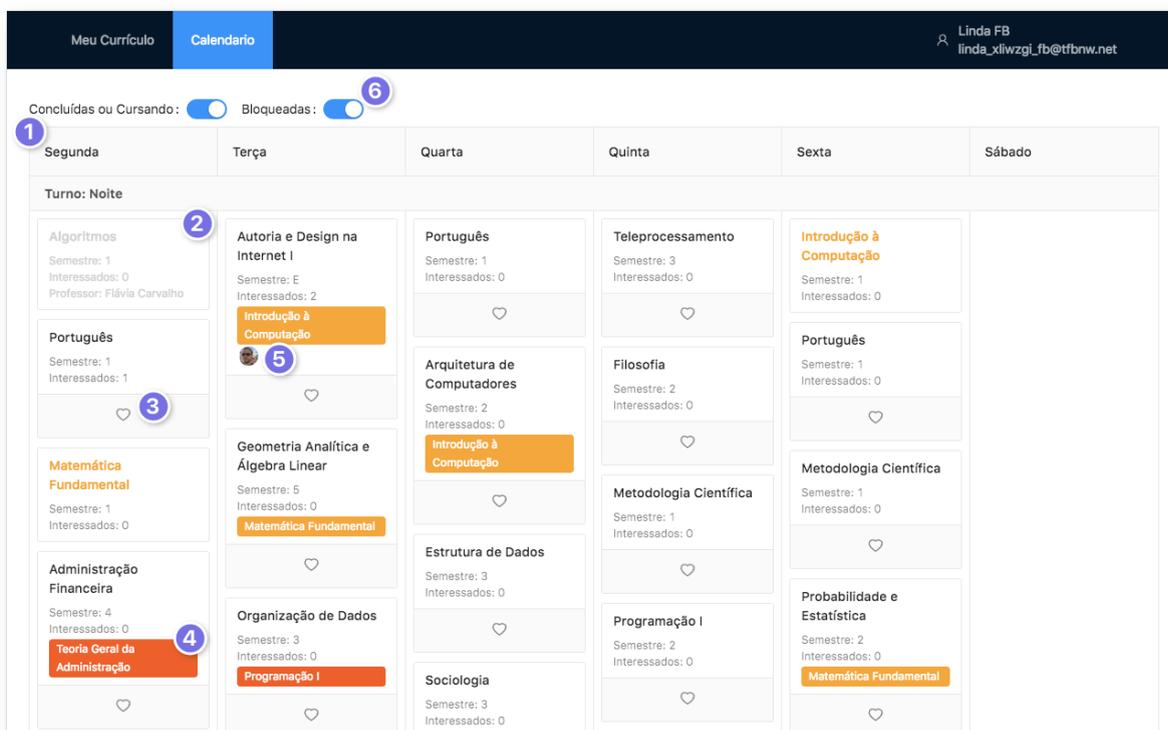
Cada disciplina é demonstrada com o nome referente ao curso ao qual o aluno está matriculado, o mestre referente, a quantidade de alunos interessados nesta, o nome do professor (caso tenha sido informado) e o número da sala (caso tenha sido informado) como é o possível ver na Figura 8 marcação 2. O nome da disciplina pode ser mostrado em 3 cores, Preto quando esta está pendente, Laranja quando está em andamento ou Cinza quando já foi cursada, é importante notar também que as disciplinas já cursadas ou em andamento não possuem o botão com ícone de coração (Figura 8 marcação 3) pois estas não podem mais serem cursadas.

Na Figura 8 marcação 4 é demonstrada a lista de dependências em andamento ou ainda pendentes nas cores Laranja e Vermelha respectivamente. Dependências já concluídas não são mostradas para não poluir a tela com informações desnecessárias.

A relação social, via Facebook, é mostrada abaixo da lista de dependências da disciplina onde o aluno pode verificar quais outros alunos, que sejam amigos no Facebook, marcaram interesse nesta matéria com o intuito de facilitar o planejamento das disciplinas entre alunos e amigos (Figura 8 marcação 5).

É possível também esconder as disciplinas já cursadas ou sendo cursadas para que se tenha a visão apenas das disciplinas que podem ser selecionadas para o semestre seguinte, também é possível esconder as disciplinas com dependências não cursadas, ou seja, bloqueantes (Figura 8 marcação 6). As disciplinas já cursadas ou em andamento são escondidas por padrão nesta tela, já disciplinas bloqueadas são exibidas por padrão para que o aluno possa planejar quais disciplinas precisam ser feitas para liberação de outras.

Figura 8 – Demonstração das funcionalidades básicas do menu “Calendário” no cliente web



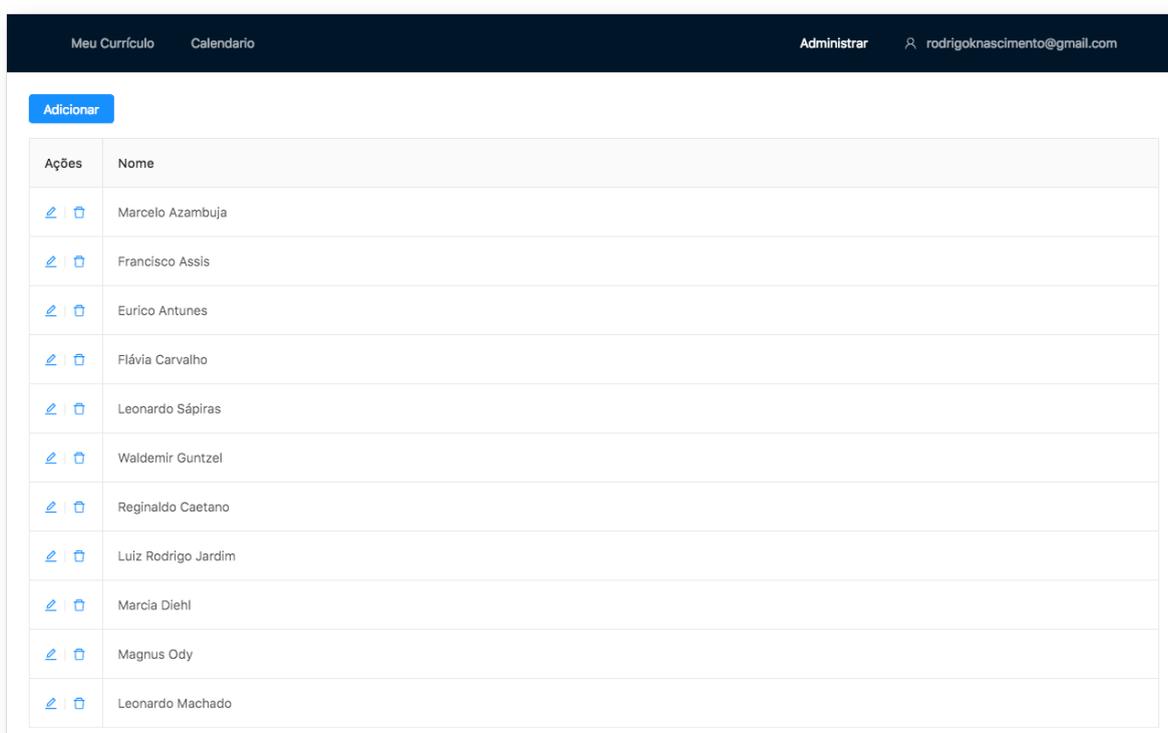
Fonte – Autor

4.2.2 Funcionalidades para os administradores

Para os administradores existem três cadastros disponíveis, Calendários, Professores e Cursos, porém o cadastro de disciplinas deve ser feito via importação de dados da instituição de ensino pois as mesmas podem conter relações diversas entre vários cursos.

O cadastro de Professores é um cadastro simples onde é possível apenas adicionar, editar ou deletar registros os quais possuem apenas o campo nome (Figura 9). Este cadastro serve para popular uma lista de seleção no cadastro de disciplinas em um calendário.

Figura 9 – Demonstração das funcionalidades da tela de edição de Professores no cliente web



Ações	Nome
✎ ✖	Marcelo Azambuja
✎ ✖	Francisco Assis
✎ ✖	Eurico Antunes
✎ ✖	Flávia Carvalho
✎ ✖	Leonardo Sápiras
✎ ✖	Waldemir Guntzel
✎ ✖	Reginaldo Caetano
✎ ✖	Luiz Rodrigo Jardim
✎ ✖	Marcia Diehl
✎ ✖	Magnus Ody
✎ ✖	Leonardo Machado

Fonte – Autor

O cadastro de Cursos permite adicionar, editar ou excluir novos cursos onde é permitida a definição do nome do curso e a quantidade de disciplinas eletivas (Figura 10). O número de disciplinas eletiva é utilizado na tela “Meu Currículo” para calcular o progresso do aluno e demonstrar em uma barra de progresso bem como saber o total de disciplinas a serem cursadas pelo aluno.

Figura 10 – Demonstração das funcionalidades da tela de edição de Cursos no cliente web

Ações	Nome	# Eletivas
 	Jogos Digitais	0
 	<input type="text" value="Sistemas de Informação"/>	<input type="text" value="1"/>
 	Tecnologo para Internet	0

Fonte – Autor

O cadastro de Calendários (Figura 11) permite adicionar e remover registros onde é possível entrar com o nome desejado para o calendário, sendo apenas um identificador para tal, após adicionado é possível selecionar qual calendário é o calendário ativo, ou seja, o calendário que será exibido aos alunos no menu “Calendário”. É possível que não haja calendários ativos, mas é permitido no máximo um calendário ativo por vez.

Diferentemente dos demais cadastros a edição do calendário leva a tela de seleção de disciplinas para compor o calendário do semestre como será visto a seguir.

Figura 11 – Demonstração das funcionalidades da tela de edição de Calendários no cliente web

Ações	Nome	Ativo
 	2019/1	<input checked="" type="checkbox"/>
 	2019/2	<input type="checkbox"/>

Fonte – Autor

Ao editar um calendário é apresentada a tela de disciplinas do calendário (Figura 12) onde é permitido ao administrador adicionar novas disciplinas primeiramente selecionando o turno e dia e depois a matéria a ser adicionada. É possível remover um item previamente adicionado pela coluna Ações, definir um professor para a disciplina pela coluna Professor, definir a sala da disciplina pela coluna Sala e verificar quantos alunos marcaram interesse na disciplina pela coluna Alunos. É importante notar que algumas matérias são únicas mas fazem

parte de mais de um curso e podem possuir nomes diferentes para cada curso, o sistema procura mostrar todos os nomes separados por *pipe* caso sejam diferentes para que o administrador não se engane na hora de montar o calendário.

Figura 12 – Demonstração das funcionalidades da tela de edição de Disciplinas de um Calendário no cliente web

Ações	Turno	Dia	Nome	Alunos	Professor	Sala
Deletar	Noite	Segunda	Algoritmos Algoritmos e Programação	0	Flávia Carvalho	Sala
Deletar	Noite	Segunda	Português	1	Professores	Sala
Deletar	Noite	Segunda	Matemática Fundamental	0	Professores	Sala
Deletar	Noite	Segunda	Administração Financeira	0	Professores	Sala
Deletar	Noite	Segunda	Programação II Laboratório de Programação de Computadores II	0	Professores	Sala
Deletar	Noite	Segunda	Sistemas Operacionais	1	Professores	Sala
Deletar	Noite	Segunda	Engenharia de Software	0	Professores	Sala
Deletar	Noite	Terça	Autoria e Design na Internet I	2	Professores	Sala
Deletar	Noite	Terça	Geometria Analítica e Álgebra Linear	0	Professores	Sala

Fonte – Autor

4.3 Aplicativos iOS e Android

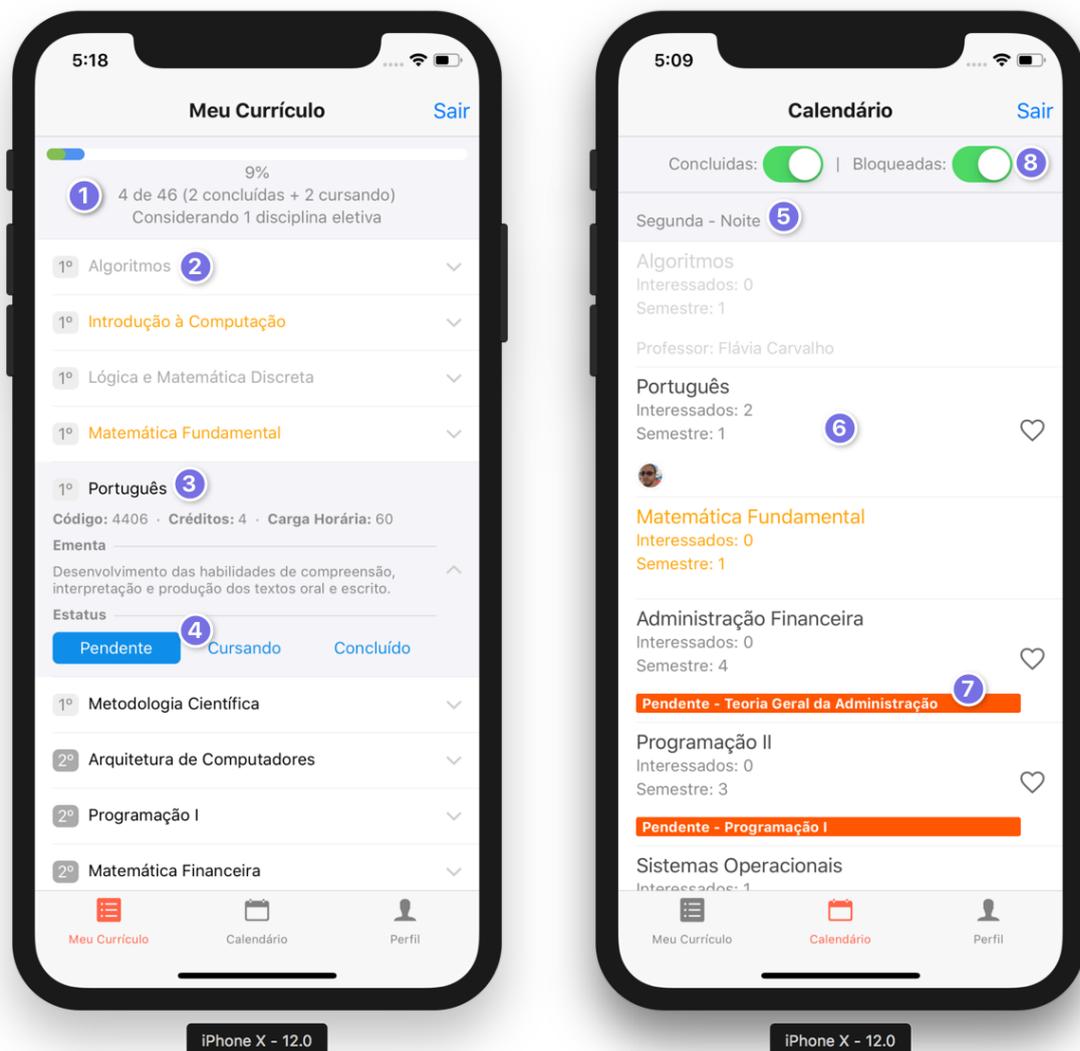
Os aplicativos móveis para iOS e android foram desenvolvidos para facilitar o acesso ao sistema pelos alunos além de trazer a funcionalidade de notificações.

Na Figura 13 são demonstradas as funcionalidades de “Meu Currículo” e “Calendário”, como é possível ver na marcação 1 onde é apresentado o progresso ao aluno bem como a descrição do mesmo logo abaixo. Na marcação 2 é demonstrada a lista de disciplinas em suas cores referentes aos status, também é possível expandir um item da lista para que se tenha acesso aos detalhes da disciplina como demonstrado na marcação 3 bem como aos botões para alterar o status da disciplina em questão (marcação 4).

Ainda na Figura 13 é possível ver a lista de calendário demonstrada pela marcação 5 onde as matérias são agrupadas por dia e turno respectivamente. Cada item (marcação 6)

possui os dados de nome da disciplina, semestre, professor (caso informado) e sala (caso informado). Há também um botão, indicado pelo ícone de coração, para marcação de interesse na disciplina, o qual é representado por um coração vazio quando sem interesse e por um coração cheio quando marcado com interesse. Também é demonstrada a lista de amigos do Facebook interessados na disciplina na marcação 6, bem como a lista de dependências (marcação 7) em suas cores representando o status da disciplina. Por fim são demonstrados os 2 filtros na marcação 8 onde o aluno pode esconder matérias concluídas ou em andamento bem como as bloqueadas por disciplinas ainda pendentes.

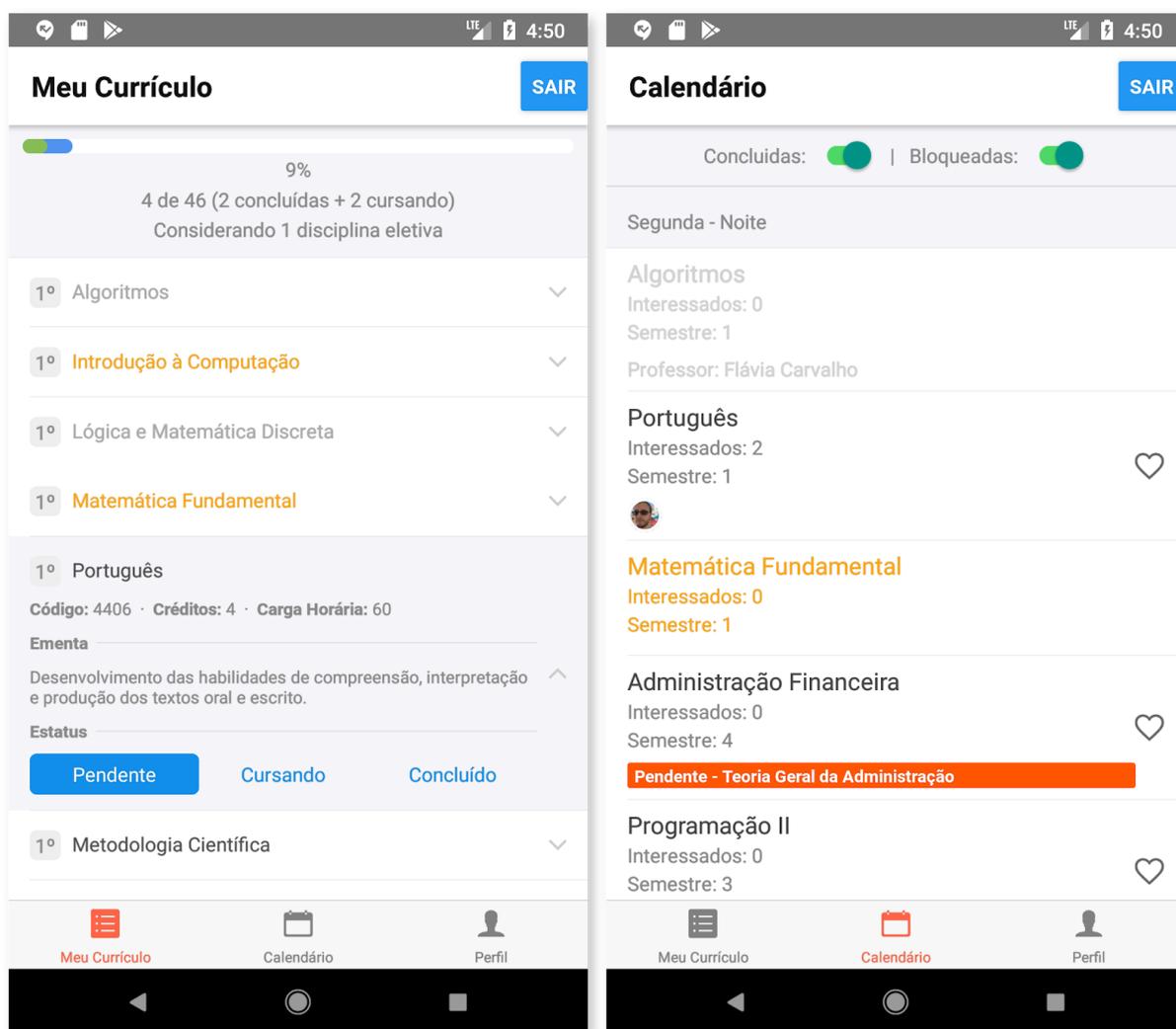
Figura 13 – Demonstração das funcionalidades básicas do aplicativo iOS: Meu Currículo e Calendário.



Fonte – Autor

A aplicação Android (Figura 14) possui os mesmos componentes e comportamentos da aplicação iOS demonstrada na Figura 13.

Figura 14 – Demonstração das funcionalidades básicas do aplicativo Android: Meu Currículo e Calendário.



Fonte – Autor

5 CONCLUSÃO

Este trabalho de pesquisa e desenvolvimento envolveu a criação de um sistema web e aplicativo móvel denominado MinhaGrade, que deverá auxiliar alunos de instituições de ensino no planejamento de suas grades curriculares bem como deve auxiliar as instituições de ensino no compartilhamento de informações e planejamento semestral.

A *Extreme Programming*, metodologia ágil utilizada no desenvolvimento, auxiliou na criação e organização do sistema, pois novos requisitos surgiram ao longo do projeto e o método escolhido contribuiu para manter processo coerente e objetivo.

O resultado do projeto consiste em dois repositórios git¹¹ de código aberto hospedados no GitHub, um contendo o código da aplicação web servidor e cliente localizada em <https://github.com/rodrigok/MinhaGrade> e outro contendo o código da aplicação móvel para iOS e Android localizado em <https://github.com/rodrigok/MinhaGrade-App>.

Ambos os projetos estão prontos para serem utilizados pelas instituições de ensino de forma gratuita sendo hospedados e gerenciados pelas mesmas. A versão final do aplicativo está pronta para ser submetida para a App Store (iOS) e Google Play (Android), pois está completamente funcional, devidamente testada e atinge o objetivo proposto.

A licença escolhida para ambos os projetos foi a MIT a qual permite uso comercial, modificações, distribuição e uso pessoal irrestritos (OPEN SOURCE INITIATIVE, 2018).

Os aplicativos móveis foram desenvolvidos utilizando o framework React Native onde foi possível observar que a ferramenta se encontra estável e atendeu todas as necessidades deste projeto.

A aplicação foi testada por alguns alunos das Faculdades Integradas de Taquara e foi bem aceito, tendo sido utilizado na prática pelo autor por diversas vezes tendo sucesso nos objetivos propostos. A expectativa é que ele seja implementado pelas Faculdades Integradas de Taquara para uso no semestre seguinte bem como por outras instituições de ensino.

REFERÊNCIAS

ABOUT Node.js. Disponível em: <<https://nodejs.org/en/about/>>. Acesso em: 01 nov. 2018.

MOZILLA Developer Network. **Introduction - JavaScript | MDN**. Disponível em: <<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Introduction>>. Acesso em: 2 out. 2018.

TILKOV, S.; VINOSKI, S. **Node.js: Using JavaScript to build high-performance network programs**. IEEE Internet Computing, v. 14, n. 6, p. 8083, 2010. ISSN 10897801.

MASSE, M. **REST API design rulebook**. O'Reilly, 2012. 94 p. ISBN9781449310509. Disponível em: <<https://books.google.com.br/books?hl=en&lr=&id=eABpzyTcJNIC&oi=fnd&pg=PR3&dq=rest+api&ots=vzPw50e9PA&sig=YQ4a4QBtBn0sRmwcgCafguPI4c4#v=onepage&q=rest%20api&f=false>>. Acesso em: 10 jul. 2018.

GRAPHQL: **A query language for your API**. Disponível em: <<https://graphql.org/>>. Acesso em: 01 nov. 2018.

¹¹ Git: *software* para versionamento distribuído de código fonte (GIT, 2018).

APPLE. **App Programming Guide for iOS: Background Execution.** Disponível em: <<https://developer.apple.com/library/content/documentation/iPhone/Conceptual/iPhoneOSProgrammingGuide/BackgroundExecution/BackgroundExecution.html>>. Acesso em: 01 nov. 2018.

ANDROID, **Sistema Operacional Android.** Disponível em: <<https://developer.android.com/about/index.html>>. Acesso em: 2 set. de 2018.

HUNT, Pete et al, **React: Facebook's Functional Turn on Writing JavaScript.** ACM Queue, p. 40, 2016. ISSN 15427749. Disponível em: https://dl.acm.org/ft_gateway.cfm?id=2994373&ftid=1785296&dwn=1

EISENMAN, Bonnie. **Learning React Native.** Building Native Mobile Apps with JavaScript. O'Reilly Media, 2015.

GOTTSCHALK K; GRAHAM, S. K. H. S. J. **Introduction to web services architecture.** IBM Systems Journal, v. 2, n. 41, p. 170–177, 2002.

JSON.ORG. **JSON,** JSON.org. 2016. Disponível em: <<http://www.json.org>>. Acesso em: 9 out. 2018.

API. **API in Dicionário Infopédia de Siglas e Abreviaturas.** Porto: Porto Editora, 2003-2018. Disponível em: <<https://www.infopedia.pt/dicionarios/siglas-abreviaturas/API>>. Acesso em: 12 set. 2018.

NIKOGHOSYAN, Stelina. **What are the 7 white labeling benefits.** 2018. Disponível em: <<https://medium.com/zangi/what-are-the-7-white-labeling-benefits-9fded331921a>>. Acesso em: 09 nov. 2018.

DESENVOLVIMENTO ÁGIL. **Extreme Programming.** 2016. Disponível em: <<http://www.desenvolvimentoagil.com.br/xp/>>. Acesso em: 26 set. 2018.

GITHUB. **Built for developers.** 2018. Disponível em: <<https://github.com/>> Acesso em: 06 out. 2018.

GIT. **About.** 2018. Disponível em: <<https://git-scm.com/about>> Acesso em: 06 out. 2018.

OPEN SOURCE INITIATIVE. **The MIT License.** 2018. Disponível em: <<https://opensource.org/licenses/MIT>> Acesso em: 03 set. 2018.

AIRSHIP, Urban. **Push Notifications Explained.** Disponível em: <<https://www.urbanairship.com/push-notifications-explained>>. Acesso em: 03 set. 2018.

ROUSE, Margaret. **Framework.** 2015. Disponível em: <<https://whatis.techtarget.com/definition/framework>>. Acesso em: 03 set. 2018.

HOCHHAUS, Stephan; SCHOEBEL, Manuel. **Meteor in Action.** [S.l.]: Manning, 2015. 342 p. Disponível em: <https://books.google.com.br/books?id=_s_3rQEACAAJ>. Acesso em: 09 nov. 2018.

SMITH, Floyd. **5 Tips to Increase Node.js Application Performance**. 2015. Disponível em: <<https://www.nginx.com/blog/5-performance-tips-for-node-js-applications/>>. Acesso em: 10 mai. 2018.

NEDELCO, C. **Nginx HTTP Server**. Birmingham, B27 6PA, UK: Packt Publishing, 2010.

MICROSOFT. **O que é PaaS?**. Disponível em: <<https://azure.microsoft.com/pt-br/overview/what-is-paas/>>. Acesso em: 10 mai. 2018.

IBM. **O que É um Servidor Proxy Reverso?**. Disponível em: <https://www.ibm.com/support/knowledgecenter/pt-br/SSKTXQ_9.0.0/admin/config/st_admin_port_rvprxy_overview_c.html>. Acesso em: 11 mai. 2018.