

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIA DA COMPUTAÇÃO – BACHARELADO

**FURB MOBILE: UMA APLICAÇÃO PARA VISUALIZAÇÃO E
ACOMPANHAMENTO DA MATRIZ CURRICULAR**

FILIFE FRANZ BARRETO

BLUMENAU
2016

FILIFE FRANZ BARRETO

**FURB MOBILE: UMA APLICAÇÃO PARA VISUALIZAÇÃO E
ACOMPANHAMENTO DA MATRIZ CURRICULAR**

Trabalho de Conclusão de Curso apresentado ao curso de graduação em Ciência da Computação do Centro de Ciências Exatas e Naturais da Universidade Regional de Blumenau como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação.

Prof. Aurélio Faustino Hoppe - Orientador

**BLUMENAU
2016**

FURB MOBILE: UMA APLICAÇÃO PARA VISUALIZAÇÃO E ACOMPANHAMENTO DA MATRIZ CURRICULAR

Por

FILIPPE FRANZ BARRETO

Trabalho de Conclusão de Curso aprovado para obtenção dos créditos na disciplina de Trabalho de Conclusão de Curso II pela banca examinadora formada por:

Presidente: _____
Prof. Aurélio Faustino Hoppe, Mestre – Orientador, FURB

Membro: _____
Prof. Dalton Solano dos Reis, Mestre – FURB

Membro: _____
Prof. Alexander Roberto Valdameri, Mestre – FURB

Blumenau, 7 de julho de 2016

Dedico este trabalho a todos que de alguma forma ajudaram para o mesmo ser concluído.

AGRADECIMENTOS

A minha família, pelo apoio e incentivo que me deram em todos os momentos para continuar seguindo meus objetivos, especialmente aos meus pais, Lievaldo dos Santos Barreto e Elfrida Franz.

Aos meus amigos e colegas de trabalho, que me apoiaram e que de alguma forma contribuíram para a construção desse trabalho.

Ao meu orientador, Aurélio Hoppe, por todo apoio e conhecimento passados durante todo o processo de desenvolvimento desse trabalho.

Os que se encantam com a prática sem a ciência são como os timoneiros que entram no navio sem timão nem bússola, nunca tendo certeza do seu destino.

Leonardo da Vinci

RESUMO

Com a grande quantidade de dispositivos móveis sendo utilizados pela população e a popularização na utilização de aplicativos, foi notada a carência de aplicações para facilitar a vida do acadêmico e outras pessoas interessadas em ingressar em uma universidade. Diante disso, este trabalho apresenta o desenvolvimento de um aplicativo em Android que possibilita a visualização informações e o acompanhamento da grade curricular dos cursos ofertados pela FURB. No desenvolvimento do trabalho foi utilizado o framework Ionic, que possibilitou o uso de HTML5, AngularJS e CSS sem a necessidade de utilização de linguagens específicas do sistema operacional. Como resultado, foi implementado uma aplicação, que permite ao usuário acompanhar a grade curricular do seu curso, visualizando informações como pré-requisito, disciplinas aprovadas e horários de aula. Já os usuários visitantes, podem ver os cursos disponibilizados pela FURB e informações a respeito da grade curricular, valores das mensalidades, formas de contato e como ingressar no curso. Os resultados obtidos a partir de experimentos e testes de usabilidade demonstram que o aplicativo desenvolvido, apesar das limitações, é capaz de facilitar o acompanhamento e a interpretação das informações disponibilizadas pelos cursos da FURB.

Palavras-chave: FURB-Mobile. Grade curricular. Android. Ionic.

ABSTRACT

With the large amount of mobile devices being used by the population and popularizing the use of applications, the shortage of applications was noted to facilitate the life of the academic and other people interested in joining a university. Considering this, this work presents the development of an application for Android that allows viewing information and monitoring of the curriculum of the courses offered by FURB. In developing this work we used the Ionic framework, which enabled the use of HTML5, AngularJS and CSS without the need to use operating system specific languages. As a result, an application was implemented that allows the user to follow the curriculum of their course, viewing information as a prerequisite, approved courses and class schedules. Already guest users can see the courses offered by FURB and information about the curriculum, the monthly values, contact forms and joining the course. The results from experiments and usability tests show that the application developed, despite the limitations, is able to facilitate the monitoring and interpretation of the information provided by FURB courses.

Key-words: FURB-Mobile. Curriculum grating. Android. Ionic.

LISTA DE FIGURAS

Figura 1 - Matriz Curricular do curso de Ciência da Computação.....	15
Figura 2 – Arquitetura Ionic	16
Figura 3 – Tela principal do aplicativo Tô Aqui!.....	18
Figura 4 - Diagrama de casos de uso	23
Figura 5 – Arquitetura da aplicação	24
Figura 6 – MER do banco de dados	25
Figura 7 – Fluxo de comunicação entre os arquivos	26
Figura 8 – Listagem com os serviços disponíveis	28
Figura 9 – Tela de <i>login</i>	32
Figura 10 – Tela de visualização da grade curricular.....	33
Figura 11 – Tela com disciplinas com pré-requisitos.....	34
Figura 12 – Tela de horários.....	34
Figura 13 – Tela de cursos e tela de informações.....	35
Figura 14 – Tela de como ingressar e contato	35
Figura 15 – Tela da grade curricular e mensalidade do curso.....	36

LISTA DE QUADROS

Quadro 1 - Análise e comparação dos trabalhos correlatos	21
Quadro 2 – Classe de retorno da entidade <code>GradeAcademico</code>	29
Quadro 3 – Retorno JSON do serviço de usuários	29
Quadro 4 – Criação do contexto do banco de dados	30
Quadro 5 – Classe <code>GradeAcademico</code> mapeada	30
Quadro 6 – Configuração de rotas da aplicação mobile na classe <code>app.js</code>	31
Quadro 7 – Factory utilizada no sistema	31
Quadro 8 – Utilização da <code>factory</code> na <code>controller cursosCtrl</code>	32
Quadro 9 – Respostas sobre o perfil do usuário	37
Quadro 10 – Respostas sobre a interface do aplicativo	38
Quadro 11 – Respostas sobre a avaliação geral do aplicativo	38
Quadro 12 - Respostas referente a usabilidade do aplicativo	38
Quadro 13 – Comparação em relação às características dos trabalhos relacionados	39
Quadro 14 - Caso de uso UC01 - Login	43
Quadro 15 - Caso de uso UC02 - Visualizar grade curricular	43
Quadro 16 – Caso de uso UC03 - Visualizar pré-requisitos	44
Quadro 17 – Caso de uso UC04 - Visualizar matérias concluídas	44
Quadro 18 – Caso de uso UC05 - Visualizar horários	44
Quadro 19 – Caso de uso UC06 - Visualizar cursos	45
Quadro 20 – Caso de uso UC07 - Visualizar como ingressar	45
Quadro 21 – Caso de uso UC08 - Visualizar mensalidade	45
Quadro 22 – Caso de uso UC09 - Visualizar contato	46
Quadro 23 – Caso de uso - UC10 - Visualizar grade curricular	46

LISTA DE ABREVIATURAS E SIGLAS

API - *Application Programming Interface*

APK - *Android Package*

CSS - *Cascading Style Sheets*

DOM - *Document Object Model*

FURB - *Fundação Universidade Regional de Blumenau*

GPS - *Global Positioning System*

HTML - *HyperText Markup Language*

HTTP - *HyperText Transfer Protocol*

IDE - *Integrated Development Enviornment*

IIS - *Internet Information Services*

MER - *Modelo Entidade Relacionamento*

MVC - *Model View Controller*

JSON - *JavaScript Object Notation*

ORM - *Object-Relational Mapping*

SDK - *Software Development Kit*

UC - *Use Case*

UML - *Unified Modeling Language*

SUMÁRIO

1 INTRODUÇÃO.....	12
1.1 OBJETIVOS.....	13
1.2 ESTRUTURA.....	13
2 FUNDAMENTAÇÃO TEÓRICA	14
2.1 FURB-MOBILE.....	14
2.2 MATRIZ CURRICULAR.....	14
2.3 <i>FRAMEWORK IONIC</i>	16
2.4 TRABALHOS CORRELATOS.....	17
2.4.1 Tô Aqui!: Aplicativo para georreferenciamento em ambientes restritos	17
2.4.2 Sistema móvel para compartilhamento de geolocalização usando mapas e notificações da Google	19
2.4.3 iAVA – Aplicação web para disponibilizar recursos do AVA no dispositivo móvel iPhone.....	20
2.4.4 Análise e Comparação dos Trabalhos Correlatos	21
3 DESENVOLVIMENTO.....	22
3.1 REQUISITOS.....	22
3.2 ESPECIFICAÇÃO	22
3.2.1 Diagrama de casos de uso	22
3.2.2 Diagrama de arquitetura da aplicação	23
3.3 IMPLEMENTAÇÃO	27
3.3.1 Técnicas e ferramentas utilizadas.....	27
3.3.2 Etapas da implementação.....	28
3.4 RESULTADOS E DISCUSSÕES.....	36
3.4.1 Experimento do aplicativo	36
3.4.2 Comparação com trabalhos correlatos e discussões.....	38
4 CONCLUSÕES.....	40
4.1 EXTENSÕES	40
APÊNDICE A – DETALHAMENTO DOS CASOS DE USO	43

1 INTRODUÇÃO

Com o passar do tempo os telefones celulares foram evoluindo juntamente com a criação de *tablets*. Eles estão ganhando cada vez mais recursos e se tornando um item quase indispensável na vida das pessoas (MONTEIRO, 2012, p. 1). A partir desta popularização, nos últimos anos foram desenvolvidas ou adaptadas inúmeras aplicações com o objetivo de oferecer maior conforto aos usuários destas tecnologias, tais como, aplicações para ouvir música, gravar vídeos, acessar a internet, utilizar GPS e entre outras tantas aplicações de diversos tipos.

Atualmente, a Universidade Regional de Blumenau (FURB) está desenvolvendo o projeto FURB-Mobile. Ele vai de encontro com o lema da Universidade, que acredita na importância de extrapolar os limites de salas e laboratórios, alcançando as pessoas onde quer que elas estejam levando-as aonde elas desejam estar. Um dos objetivos do projeto FURB-Mobile é integrar o público e a comunidade com a universidade, acolhendo aqueles que nela necessitam estar e fazer uso, diminuindo o tempo ocioso, propiciando a autonomia e informação (FURB MOBILE, 2015).

Hoje, a FURB disponibiliza em seu site informações onde os acadêmicos podem consultar as disciplinas que estão ou já foram cursadas através do sistema de matrículas. Porém, não é possível visualizar todas as disciplinas do currículo em que o aluno se matriculou ao entrar na universidade. Outro problema é que nem todos os cursos de graduação oferecidos pela FURB possuem uma matriz curricular de forma gráfica, ou seja, o acadêmico em muitos casos não possui uma visão clara da matriz curricular do seu curso, como por exemplo, quais são os pré-requisitos para cursar outras disciplinas ou até mesmo, quais disciplinas já foram feitas ou que precisam ser cursadas para terminar a graduação.

Diante do exposto, este trabalho apresenta o desenvolvimento de uma aplicação em Android, que será um módulo do projeto FURB-Mobile, que tem como intenção apresentar a matriz curricular de cada curso da FURB. Assim como, permitir que o acadêmico possa visualizar o andamento do seu curso, *status*, pré-requisitos das disciplinas e os seus respectivos horários de aula. Também, possui um modo para visitantes, onde é possível visualizar todos os cursos ofertados pela FURB e suas disciplinas, informações de como ingressar, contato e mensalidade dos cursos.

1.1 OBJETIVOS

Este trabalho tem como objetivo desenvolver uma aplicação Android para que alunos ou visitantes da FURB possam acompanhar ou visualizar informações dos cursos através de *smartphones* ou *tablets*.

Os objetivos específicos deste trabalho são:

- a) disponibilizar um módulo para que alunos da FURB possam acompanhar o andamento do curso;
- b) disponibilizar um módulo para que alunos da FURB possam acompanhar os horários de aula das disciplinas ao qual está matriculado;
- c) disponibilizar um módulo para usuários que não estão matriculados na FURB possam visualizar informações dos curso oferecidos pela instituição.

1.2 ESTRUTURA

Este trabalho está dividido em quatro capítulos. No primeiro, consta uma introdução sobre o assunto. O segundo capítulo apresenta a fundamentação teórica necessária para o desenvolvimento do aplicativo, utilizando conceitos de matriz curricular voltada para o projeto FURB-Mobile utilizando o *framework* Ionic. O terceiro capítulo traz informações sobre o desenvolvimento do trabalho, requisitos, especificação, implementação, operacionalidade e resultados. No quarto capítulo são relatadas as conclusões do trabalho e suas possíveis extensões.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo está organizado em três seções. A seção 2.1 apresenta o projeto FURB-Mobile. A seção 2.2 tem por objetivo esclarecer os conceitos relacionados a uma matriz curricular. Na seção 2.3 se apresenta o *Framework* Ionic para geração de aplicativos móveis. Ao final, a seção 2.4 apresenta os trabalhos correlatos.

2.1 FURB-MOBILE

O projeto FURB-Mobile foi criado com a intenção de ir de encontro com o lema da universidade. A FURB tem como uma das suas principais intenções, sair das salas e laboratórios e assim alcançar as pessoas em qualquer lugar que estiverem e leva-las onde elas desejam estar (FURB MOBILE, 2015).

O principal objetivo do projeto é desenvolver um aplicativo para dispositivos móveis, primeiramente em menor escala e ampliar gradativamente o projeto com outras funcionalidades, tendo como objetivo melhorar a interação dos acadêmicos e a comunidade que utilizam as dependências da FURB, sendo uma dessas melhorias, a locomoção dos usuários dentro da universidade (FURB MOBILE, 2015).

O projeto possui como objetivos específicos, apontar no aplicativo eventos dentro da universidade e sua localização no momento, ler e reconhecer imagens QR code colocadas em lugares específicos para informar a sua localização momentânea, além de integrar a população com universidade, acolhendo aqueles que tenham que fazer das instalações da FURB, de forma mais ágil proporcionando autonomia e informação (FURB MOBILE, 2015).

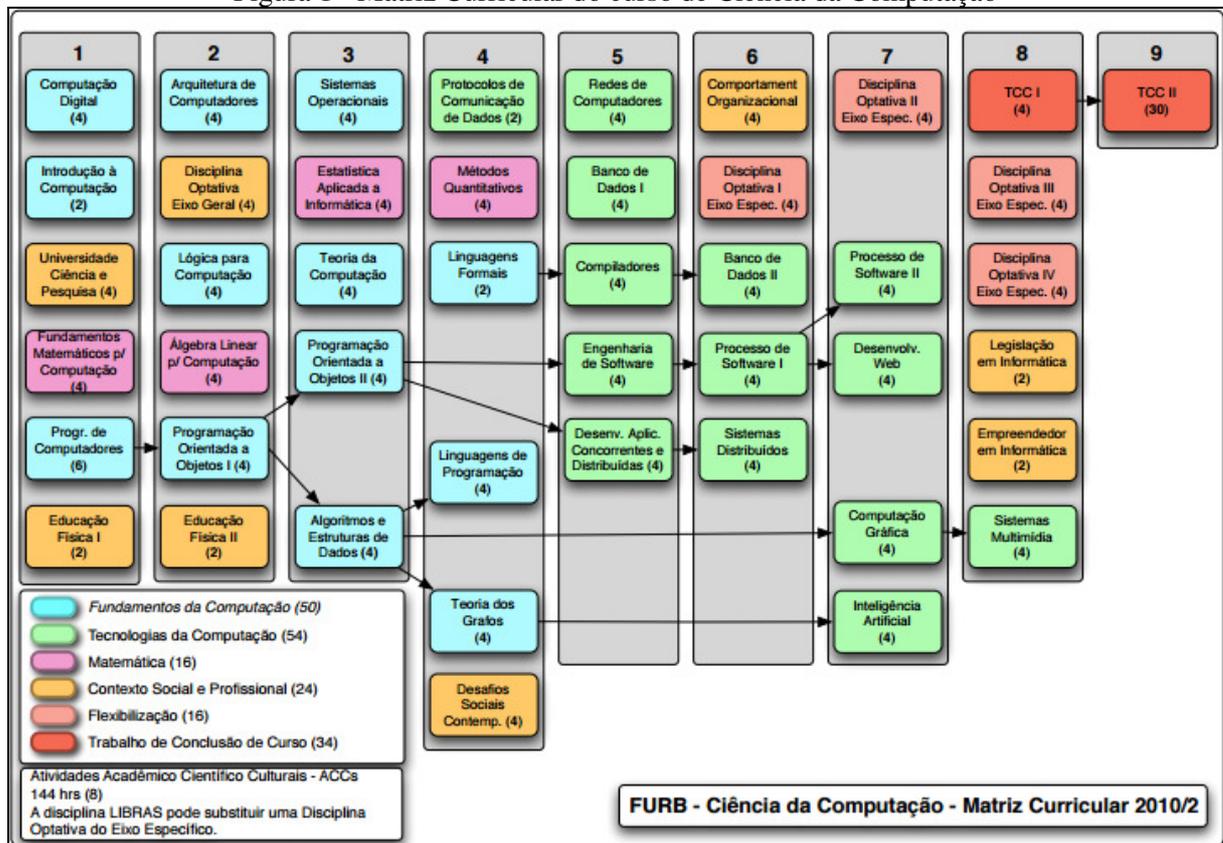
2.2 MATRIZ CURRICULAR

Uma matriz curricular nada mais é que uma lista de todos os componentes curriculares (disciplinas) que o aluno precisa cumprir para a conclusão do seu curso. Cada disciplina tem um número “x” de horas que precisam ser cumpridas. A soma de todas estas horas é o total necessário para a conclusão do curso (FADERGS, 2014).

Geralmente no começo do ano letivo os professores se perguntam o que ensinar e de que forma ensinar e o que realmente seus alunos precisam ter conhecimento. Um dos métodos para se organizar essas questões com maior eficácia é a construção de matriz curricular. Um plano de ensino geralmente é feita de forma coletiva, e que deve conter uma fundamentação teórica, os objetivos de aprendizagem, os conteúdos que serão trabalhados no período, orientações didáticas e referências bibliográficas preferencialmente seguidas de sugestões de atividades e leituras complementares (CAIADO, 2015).

Segundo Hoppe (2015), na FURB um curso de graduação é composto por diversos componentes curriculares (disciplinas). Elas são separadas em fases, onde determinadas disciplinas são pré-requisitos para disciplinas de semestres futuros. Ou seja, se o acadêmico reprovar em uma disciplina que é pré-requisito para outra, ele poderá cursá-la na fase seguinte ou em fases subsequentes. Porém, em ambos os casos, poderá ocorrer conflito de horários, ao qual o aluno terá dificuldades para encaixar as disciplinas pendentes com as do semestre corrente. Entretanto, na maioria das vezes, o aluno não tem conhecimento do pré-requisito ou da importância dele. Diante desse contexto, alguns cursos da FURB, visando melhorar o entendimento da organização pedagógica, disponibilizam sua matriz curricular de forma gráfica, onde os itens descritos acima estão representados de maneira mais fácil para melhor compreensão dos alunos, conforme mostra a Figura 1.

Figura 1 - Matriz Curricular do curso de Ciência da Computação



Fonte: Departamento de Sistemas e Computação (2010).

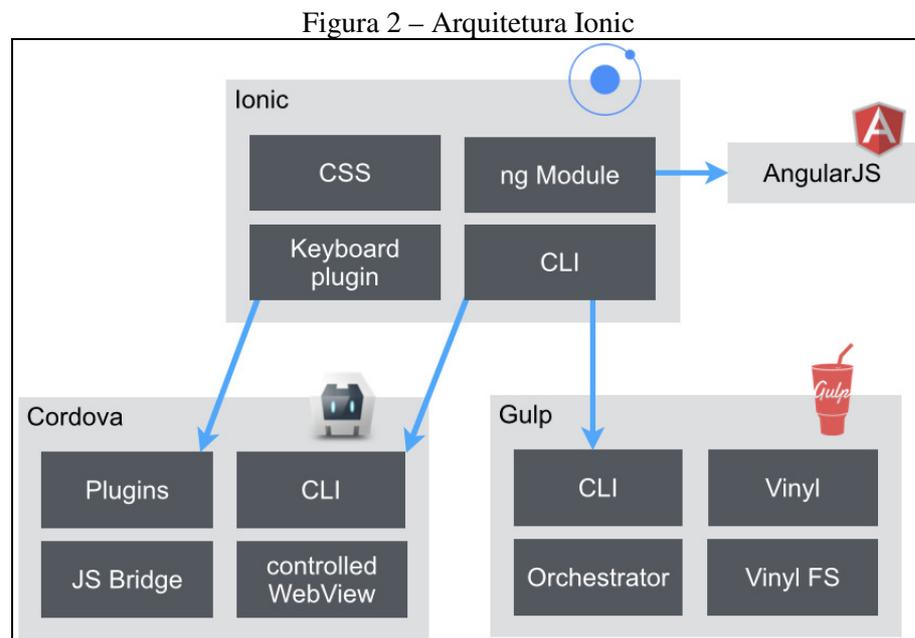
A Figura 1 mostra a matriz curricular do curso de ciência da computação. A partir dela, pode-se observar que as caixinhas coloridas representam as componentes que compõem a matriz curricular. Cada componente curricular pode conter flechas apontando para outras disciplinas de semestres a frente. Essas setas servem para indicar que a referida disciplina possui pré-requisito. As caixas também estão separadas por colunas, e cada coluna representa

um semestre, sendo indicado com uma numeração no topo de cada coluna. Por fim, as cores servem para diferenciar as áreas temáticas existentes na matriz curricular do curso.

2.3 FRAMEWORK IONIC

O surgimento do Ionic teve como ponto de partida ser um *framework* de componentes visuais com foco em aplicações mobile sob a ferramenta AngularJS. Logo, pode-se dizer que ele segue o modelo de *Single Page Application* do AngularJS. Os componentes do Ionic foram construídos com foco em um visual independente de plataforma e sua usabilidade aprimorada para aplicativos móveis (LOPES, 2016, p. 129).

Atualmente, ao se criar um projeto com Ionic, todos os recursos da plataforma já veem inclusos no projeto. Um projeto Ionic tem como base o *framework* Cordova, tendo como diferencial a disponibilização de vários recursos, sendo que o Ionic *framework* evoluiu para algo conhecido como Ionic Platform, que agrega serviços além do *framework* padrão (LOPES, 2016, p. 130). Na Figura 2 pode-se ver como é composta a arquitetura do *framework* Ionic.



Fonte: Ripkens (2014).

O núcleo do Ionic é formado por quatro partes, a primeira delas é a parte do CSS que possui um *layout* padrão otimizado para dispositivos móveis. A segunda é o módulo AngularJS que é responsável por definir as diretivas e elementos personalizados do *framework*. A terceira é uma poderosa ferramenta de linha de comando que permite agir como um proxy para o Cordova e o Gulp. Na quarta, o *plugin* Keyboard, que mesmo não sendo obrigatório, fornece informações sobre o estado atual do aplicativo (RIPKENS, 2014).

O *framework* AngularJS, tem como função lidar com as deficiências do desenvolvimento em HTML, abstraindo o HTML, CSS e JavaScript, fornecendo uma maneira imperativa para manipular o *Document Object Model* (DOM). Já o jQuery tem como função, auxiliar na construção da interface gráfica para o usuário.

O jQuery, desde sua criação foi disponibilizada como software livre e aberto. Ele simplifica a manipulação de documentos HTML, eventos, animações e interações com AJAX para desenvolvimento rápido de aplicações *web* (BALBUINO, 2012, p. 33). Tendo a simplicidade como palavra-chave, o jQuery resume e norteia o desenvolvimento de uma aplicação. Onde, linhas a mais de código são necessárias para conseguir um simples efeito utilizando JavaScript. No jQuery, são necessárias poucas linhas, simplificando a codificação, manutenção e entendimento do código (SILVA, 2014, p. 26).

O AngularJS é um *framework* JavaScript que tem como objetivo simplificar o desenvolvimento de aplicações *web* robustas, viabilizando a implementação do conceituado modelo *Model View Controller* (MVC) (PEREIRA, 2014, p. 13). Este *framework* permite aplicar práticas de engenharia de software testadas e aprovadas, tradicionalmente utilizadas no lado servidor e na programação do lado cliente para acelerar o desenvolvimento de *frontends*. Ele também oferece uma estrutura consistente e escalável, que facilita desenvolver aplicações complexas (SESHADRI; GREEN, 2014, p. 20).

2.4 TRABALHOS CORRELATOS

Nesta seção serão descritos três trabalhos acadêmicos. A seção 2.4.1 detalha “Tô Aqui!: Aplicativo para georreferenciamento em ambientes restritos” de Rocha (2015). Na seção 2.4.2 é descrito mais um trabalho de conclusão de curso que aborda o tema de “Sistema móvel para compartilhamento de geolocalização usando mapas e notificações da Google” de Kestring (2014) e, na seção 2.4.3 é descrito o trabalho sobre “iAVA – Aplicação web para disponibilizar recursos do AVA no dispositivo móvel iphone” de Sampara (2010). Por fim, na seção 2.4.4 é feita uma análise comparativa entre as características dos trabalhos correlatos.

2.4.1 Tô Aqui!: Aplicativo para georreferenciamento em ambientes restritos

Rocha (2015) desenvolveu um projeto de georreferenciamento para dispositivos móveis que tem o intuito de facilitar a localização de pessoas em ambientes restritos. Para alcançar seus objetivos, Rocha (2015) utilizou as tecnologias Beacon e *Assisted Global Positioning System* (AGPS) para aprimorar a precisão de localização juntamente com uma

aplicação desenvolvida em Phonegap. A aplicação de Rocha (2015) tinha como intuito facilitar a localização de pessoas dentro de ambientes restritos, funcionando de modo que o usuário veja a sua localização através de um ponto no mapa que representa a universidade. A Figura 3 mostra a tela principal do aplicativo Tô Aqui!, onde é possível verificar sua localidade dentro do espaço da FURB.

Figura 3 – Tela principal do aplicativo Tô Aqui!



Fonte: Rocha (2015).

O aplicativo de Rocha (2015) permite que o usuário possa visualizar no mapa eventos e ambientes próximos a sua posição, informações sobre sua posição atual e eventos, além de permitir a escolha dos tipos de eventos a serem visualizados no mapa. O software foi desenvolvido com o *framework* Cordova, para os sistemas Android e iOS, através do Notepad++, pois o mesmo não precisa de Integrated Development Environment (IDE) para ser utilizado. Foi utilizado um plugin de terceiro que comporta uma comunicação com iBeacon e aplicações multiplataforma. Além de utilizar plugins com a finalidade de aprimorar a experiência do usuário como: Phonegap Google Maps Plugin para integração com o Google Maps, Cordova Local Notification Plugin para notificações e TTS Plugin para síntese de voz visando melhorar a acessibilidade do aplicativo. Na questão da interface foram utilizadas as bibliotecas JQuery Mobile e Knockout para facilitar o desenvolvimento.

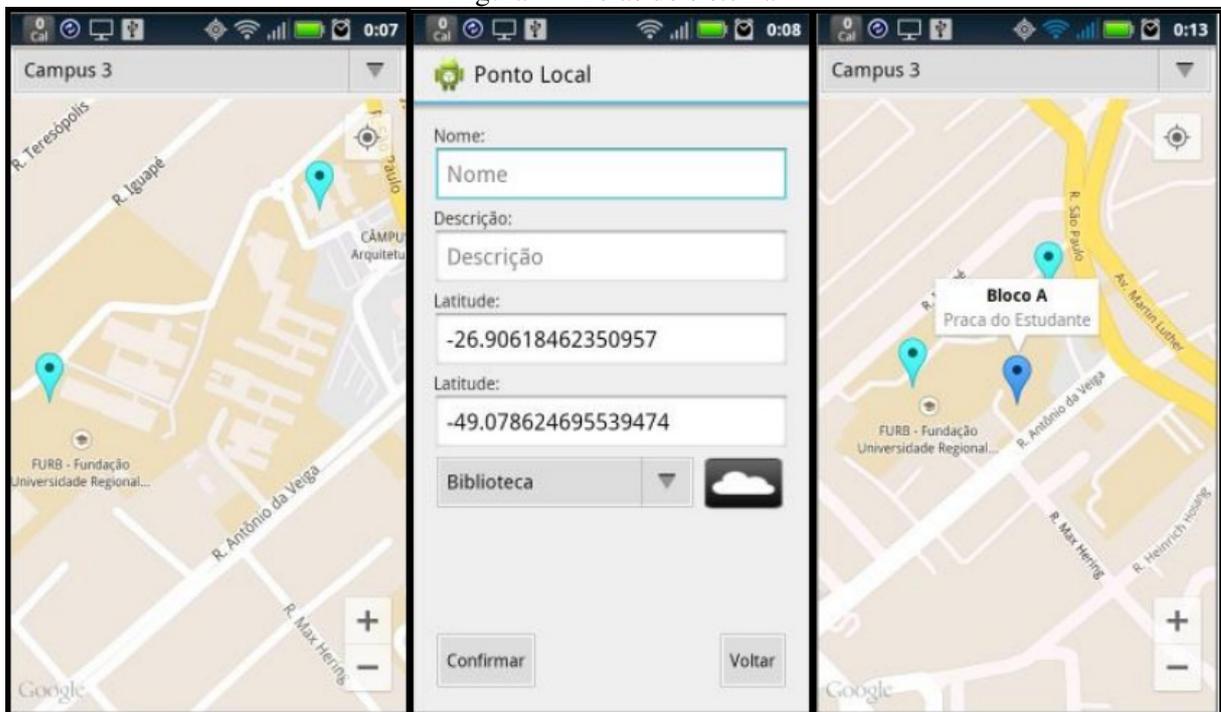
Como resultado, a aplicação teve um bom desempenho, pois conseguiu ser de fácil utilização para os usuários. Além disso, cumpriu seus objetivos que eram a localização em ambientes internos, que funcionou mesmo sem sinal de Global Positioning System (GPS) por conta do uso dos Beacons, a disponibilização multiplataforma do aplicativo fez aumentar o

alcance de usuários e a facilidade de manutenção, além de conseguir melhorar a percepção do usuário referente à posição dele em um ambiente.

2.4.2 Sistema móvel para compartilhamento de geolocalização usando mapas e notificações da Google

O trabalho de conclusão de curso de Kestring (2014) teve como principal objetivo desenvolver uma aplicação que possibilite identificar a posição do usuário utilizando mapas e pontos de referência só utilizando informações do GPS, como pode ser visto na Figura 4. Nas telas é possível notar a tela Mapa (esquerda), tela de cadastro do local do ponto (centro) e o ponto na tela do mapa (direita).

Figura 4 – Telas do sistema



Fonte: Kestring (2014).

Para atingir o seu objetivo, Kestring (2014) utilizou em seu trabalho a biblioteca `google_play_services.jar` para acessar o Google Maps dentro do aplicativo, e também o Google Cloud Messaging, necessária para enviar e receber notificações. A aplicação foi desenvolvida em camadas, a camada Data Access Object (DAO) que é responsável pela centralização de acesso ao banco de dados. A camada webservice que foi implementada com o objetivo de receber requisições HyperText Transfer Protocol (HTTP), acessar o banco de dados através da camada DAO e retornar as informações para quem fez a requisição. Outra parte é a questão de envio e recebimento de notificações tanto do servidor quanto no Android.

Por último, a parte principal do sistema que possui a visualização de um mapa em que aparece a posição atual do usuário, os seus pontos de interesse e seus amigos.

Ao final do trabalho, Kestring (2014) conclui que os resultados foram satisfatórios, mas não foram os ideais, por motivos de falta de familiarização das ferramentas utilizadas, pouco material de consulta disponível e demora ao efetuar grande quantidade de requisições ao web service.

2.4.3 iAVA – Aplicação web para disponibilizar recursos do AVA no dispositivo móvel iPhone

O trabalho de Sampara (2010) teve como objetivo desenvolver uma aplicação que contém os recursos do Ambiente Virtual de Aprendizagem sendo visualizada em um dispositivo móvel iPhone, onde é possível ter acesso a uma matéria e suas respectivas pastas, como pode ser visto na Figura 5.

Figura 5 - Tela de área de estudo do aplicativo



Fonte: Sampara (2010).

Para alcançar seus objetivos, Sampara (2010) utilizou a ferramenta de desenvolvimento Xcode juntamente com Interface Builder que é um editor gráfico, cujo objetivo é facilitar a criação, edição e o design de componentes de interfaces para aplicações e o Dashcode que é uma ferramenta que permite criar aplicativos do tipo widgets e Web, utilizando arquivos de código do tipo HTML, CSS e JavaScript. Nesse projeto utilizou-se três

modelos de padrões Design Patterns, o MVC, Comand e Sigleton. A aplicação conta com 2 servidores um que controla a interface e o outro os dados do aplicativo. O servidor de interface se comunica com o outro servidor através de chamadas.

Como resultado dessa aplicação, pode-se afirmar que ele atingiu os objetivos propostos apesar das dificuldades encontradas no acesso do Webservice do portal AVA, falta de suporte para o componente `HTML input type="file"` no iPhone, para que o usuário possa selecionar um arquivo localmente, entre outras dificuldades. Contornados tais imprevistos, concluiu-se que o desenvolvimento de uma interface para o iPhone é muito diferente de uma para desktop.

2.4.4 Análise e Comparação dos Trabalhos Correlatos

A partir das informações obtidas, foi montado o Quadro 1 com as principais características dos trabalhos descritos acima.

Quadro 1 - Análise e comparação dos trabalhos correlatos

Característica / Trabalho	Rocha (2015)	Kestring (2014)	Sampara (2010)
Aplicação Mobile	Sim	Sim	Sim
Utiliza Web Service	Não	Sim	Sim
Desenvolvimento voltado para multiplataforma	Sim	Não	Não
Facilita a integração de acadêmicos e comunidade com a FURB	Sim	Sim	Não
Possui acesso a informações dos alunos	Não	Não	Sim
Facilita o acompanhamento das disciplinas	Não	Não	Sim
Plataforma	Android e iOS	Android	iOS

A partir do Quadro 1, pode-se verificar que todos os trabalhos procuram facilitar e dar mobilidade ao usuário. Os trabalhos possuem objetivos diferentes, por isso, fica difícil afirmar qual é o mais completo. O trabalho de Rocha (2015) trabalha com a questão do aumento da precisão da localidade de um usuário dentro do espaço da universidade. Já Kestring (2014), além de fornecer a localização do usuário na FURB, tem como característica socializar informações com amigos. E, por último, Sampara (2010) aborda a questão de acesso ao AVA através do dispositivo iPhone. Porém, nenhum permite o acompanhamento ou a visualização de informações das disciplinas que compõem a matriz curricular de um curso de graduação.

3 DESENVOLVIMENTO

Neste capítulo são apresentadas as etapas do desenvolvimento do aplicativo. Na seção 3.1 são enumerados os principais requisitos do projeto desenvolvido. A seção 3.2 apresenta a arquitetura e especificação do aplicativo. A seção 3.3 detalha a implementação do aplicativo. Por fim, a seção 3.4 apresenta os experimentos realizados e resultados obtidos.

3.1 REQUISITOS

O aplicativo deverá atender os seguintes requisitos:

- a) permitir a visualização da matriz curricular de cada curso (Requisito Funcional - RF);
- b) permitir que o usuário possa visualizar as dependências das disciplinas da sua matriz curricular (RF);
- c) permitir que o usuário possa visualizar os horários das disciplinas ao qual ele está matriculado (RF);
- d) permitir usuários visitantes possam visualizar informações sobre os cursos da FURB (RF);
- e) executar na plataforma Android (Requisito Não-Funcional - RNF);
- f) ser desenvolvido na plataforma Cordova/Ionic com auxílio do *framework* AngularJS (RNF);
- g) possuir uma interface amigável (RNF).

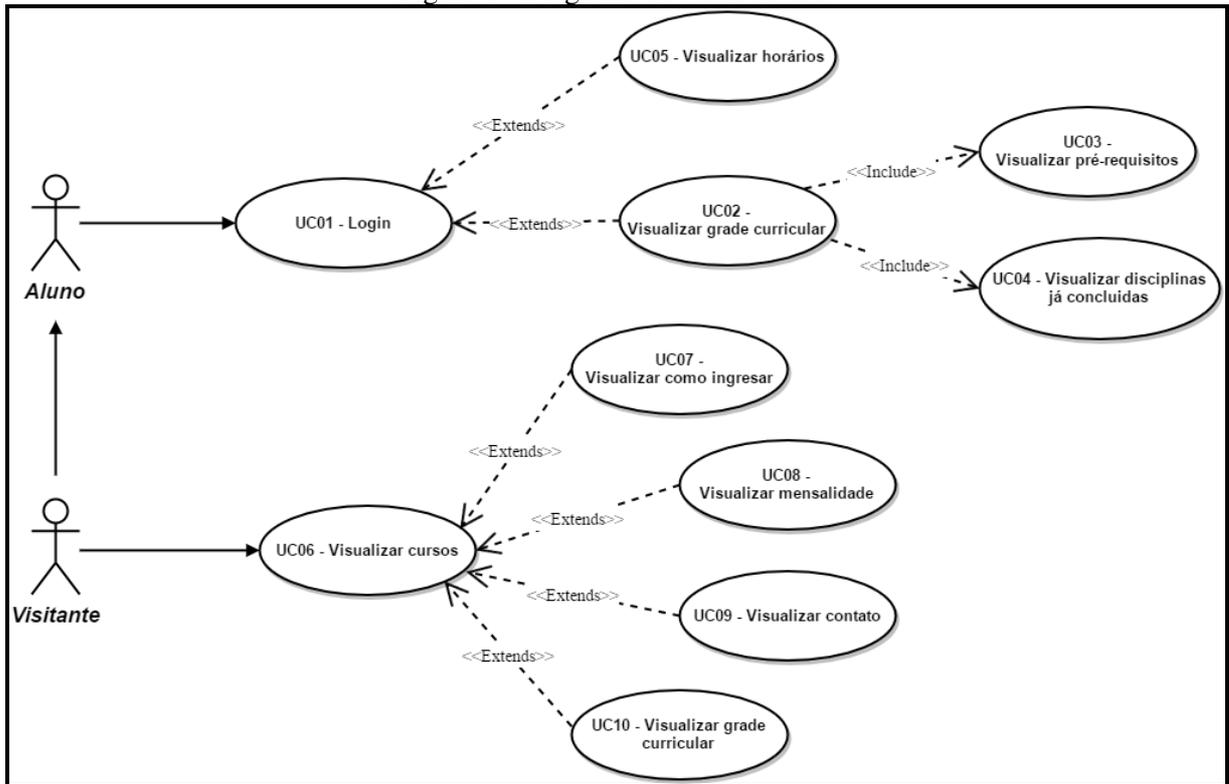
3.2 ESPECIFICAÇÃO

A especificação da aplicação foi representada em diagramas da *Unified Modeling Language* (UML), utilizando o site *Flowchart Maker & Online Diagram Software*. Foram elaborados os diagramas de casos de uso e o diagrama de arquitetura da aplicação, sendo descritos nas próximas seções.

3.2.1 Diagrama de casos de uso

A Figura 4 exhibe o diagrama de casos de uso com as ações disponibilizadas pelo aplicativo para o ator *Aluno* e *Visitante*.

Figura 4 - Diagrama de casos de uso

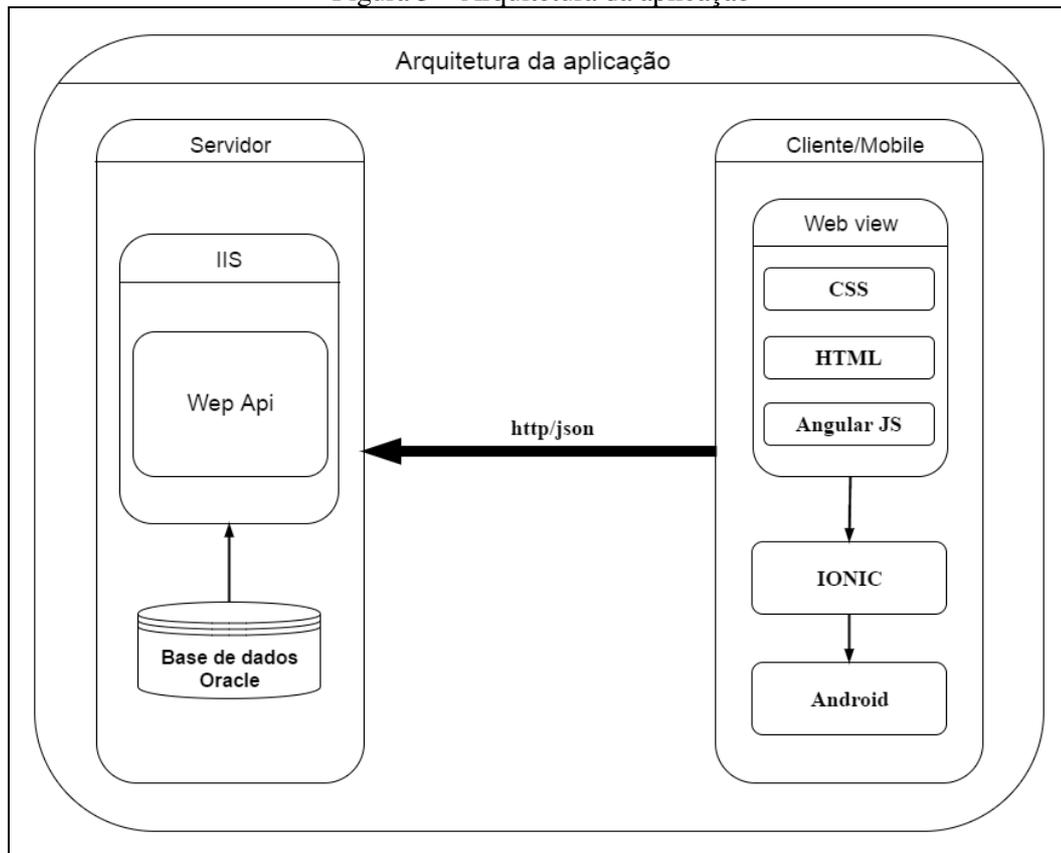


O caso de uso UC01 - Login permite ao aluno entrar na aplicação. No caso de uso UC02 - Visualizar grade curricular o aluno pode ver a grade curricular do seu curso. No caso de uso UC03 - Visualizar pré-requisitos o aluno pode ver quais disciplinas são dependentes da disciplina selecionada. No caso de uso UC04 - Visualizar matérias concluídas o aluno pode verificar quais disciplinas já foram feitas observando sua cor. No caso de uso UC05 - Visualizar horários o aluno consegue ver os horários das disciplinas ao qual está matriculado. No caso de uso UC06 - Visualizar cursos o usuário pode ver uma lista com todos os cursos oferecidos pela FURB. O caso de uso UC07 - Visualizar como ingressar permite ao usuário consultar as formas de ingresso na universidade. No caso de uso UC08 - Visualizar mensalidade o usuário pode ver o custo da mensalidade do curso selecionado. No caso de uso UC09 - Visualizar contato o usuário pode ver os contatos de um determinado curso. No caso de uso UC10 - Visualizar grade curricular o usuário consegue ver todas as disciplinas divididas por semestre do curso. O detalhamento dos casos de uso do aplicativo pode ser consultado no Apêndice A.

3.2.2 Diagrama de arquitetura da aplicação

Esta seção apresenta a arquitetura da aplicação. A aplicação foi dividida em duas partes: servidor e cliente, conforme mostra a Figura 5.

Figura 5 – Arquitetura da aplicação



No servidor está instalado o *Internet Information Services* (IIS), que é um servidor web voltado para sistemas operacionais da Microsoft, sendo capaz de integrar diversas tecnologias. Uma de suas funções é gerenciar o ASP.NET para dois tipos de aplicações, páginas *web* tradicional e *web service*, que é o caso dessa aplicação.

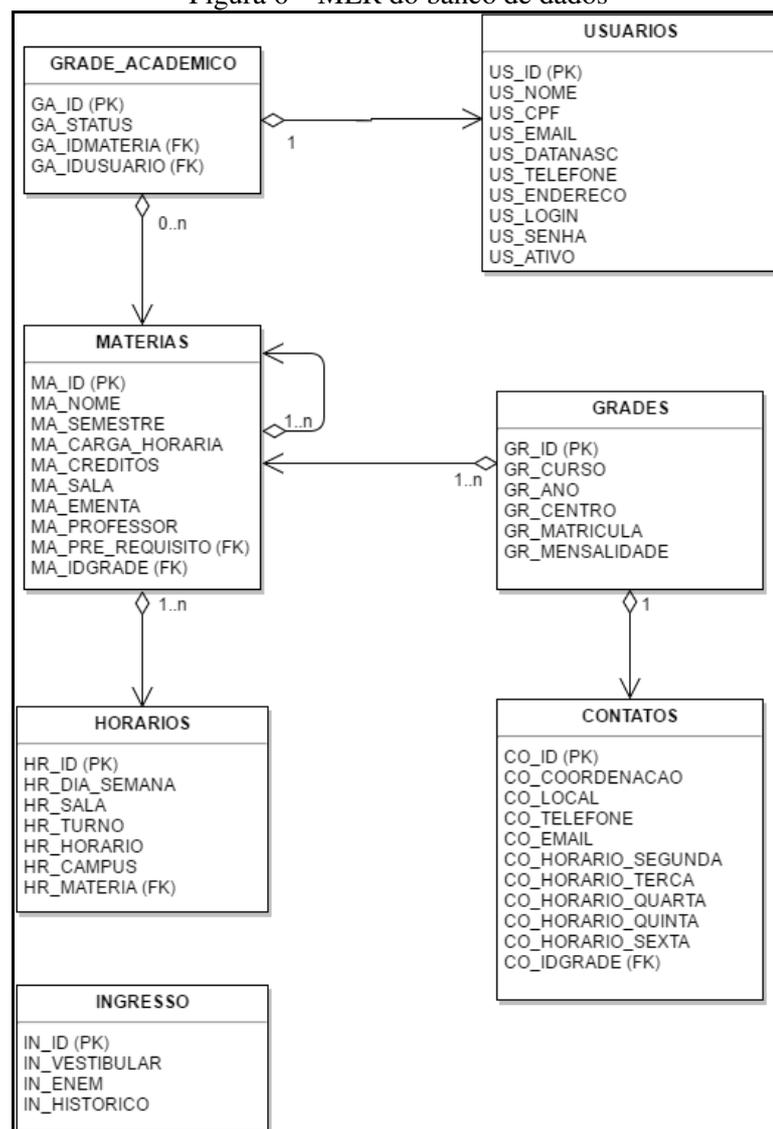
O servidor também utiliza a Web API, que serve para recuperar somente os dados solicitados de um banco de dados a partir de um site. Na aplicação, ela é utilizada para recuperar os dados da grade curricular do aluno e informações dos cursos ofertados pela FURB. A Web API também disponibiliza os serviços que serão utilizados pela aplicação. Como base de dados foi escolhido o Oracle, pois já é utilizado pela universidade. A aplicação se conecta ao banco de dados Oracle utilizando *Entity Framework Code First*. Ele é responsável pela manipulação dos dados. A comunicação entre o servidor e a aplicação ocorre através uma requisição HTTP para o servidor, e o mesmo retorna um JSON com as informações para o aplicativo.

Foram construídas classes que servem de base para montar a estrutura dos dados. Essa estrutura foi planejada pensando no melhor relacionamento possível entre as entidades, diminuindo quantidade ligações entre várias tabelas diferentes. Essa estrutura de dados,

resgatada do banco de dados, é transformada em JSON e, posteriormente utilizadas/enviadas para a aplicação móvel.

Uma das classes que a aplicação possui é a classe de usuário. Ela tem como função validar se o aluno possui vínculo com a FURB. Além disso, ela também é responsável por tratar as informações da grade do acadêmico. A grade do acadêmico foi criada com o intuito de centralizar todas as informações necessárias de um aluno tais como, grade curricular e suas especificidades. A classe matéria é utilizada para formar a matriz curricular dos diversos cursos disponíveis pela instituição. E, por último, o objeto horário é utilizado para identificar os horários e local onde as disciplinas irão acontecer. Optou-se por não fazer diagrama de classe, pois as classes utilizadas apenas possuem atributos da estrutura das tabelas do banco de dados. Para essa finalidade se criou um Modelo Entidade Relacionamento (MER) do banco de dados, como pode ser visto na Figura 6.

Figura 6 – MER do banco de dados

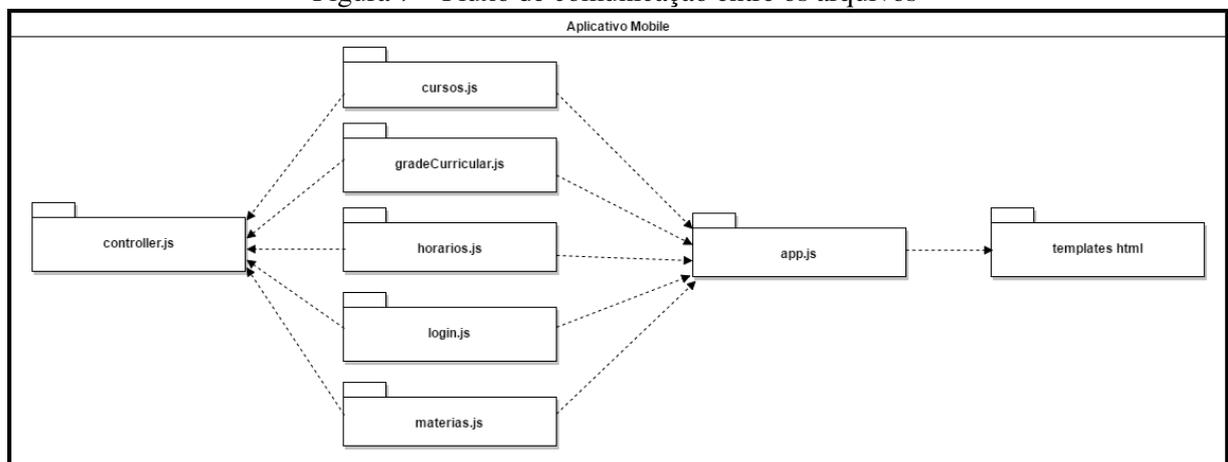


Na figura a cima, pode-se verificar a estrutura das sete tabelas que formam o conjunto de dados da aplicação. A tabela `GRADE_ACADEMICO` possui ligação com a tabela `USUARIOS` sendo possível apenas ter um usuário por grade do acadêmico, além de vínculo com a tabela `MATERIAS` onde é possível ter uma ou mais matérias na grade do acadêmico. A tabela `MATERIAS` possui um vínculo com ela mesma para identificar matérias de pré-requisito, além de também pertencer a uma grade que está armazenada na tabela `GRADES`, e cada matéria pode conter um ou mais horários contidos na tabela `HORARIOS`. Na tabela `GRADES` possui-se um vínculo com a tabela `CONTATOS` onde haverá um contato para cada grade. E a tabela ingresso apenas conterá informações de âmbito geral de como ingressar na universidade.

A parte cliente da aplicação foi desenvolvida utilizando o *framework* Ionic. Na construção do aplicativo foram utilizados HTML, CSS, AngularJS e JavaScript. O HTML é utilizado para construção das telas, tendo auxílio de *tags* específicas fornecidas pelo Ionic, além do CSS para melhorar o visual das telas. O AngularJs e o JavaScript, são responsáveis por toda a manipulação do DOM, tanto nas telas como nas *controllers*.

Para melhor organização do código, optou-se por separar as *controllers* do aplicativo em diferentes arquivos, de acordo com sua especificidade. A Figura 7 exibe o fluxo de comunicação entre os arquivos que compõem o aplicativo.

Figura 7 – Fluxo de comunicação entre os arquivos



A partir do diagrama da Figura 7, pode-se visualizar a estrutura básica de como é o funcionamento e a interatividade entre os arquivos no aplicativo mobile. O arquivo `controller.js` é responsável por toda a parte de conexão entre a aplicação e o servidor. Ele, também realiza a leitura dos arquivos que estão salvos na própria aplicação. Os arquivos `cursos.js`, `gradeCurricular.js`, `horario.js`, `login.js` e `matérias.js` são responsáveis por manipular e validar os dados recebidos e deixar prontos para a utilização e requisição de uma *view* da aplicação. O arquivo `cursos.js` é responsável por manipular todos os dados

referentes aos diversos cursos oferecidos pela FURB, sendo que esta funcionalidade estará disponível apenas para usuários não matriculados pela FURB. O arquivo `gradeCurricular.js` é responsável por reger as informações referentes de uma grade curricular de um aluno matriculado na universidade. O arquivo `horários.js` é incumbido de manipular os dados para a apresentação dos horários das disciplinas que o universitário está matriculado no semestre. O arquivo `login.js` é responsável por autenticar a entrada dos usuários no sistema. O arquivo `materias.js` é encarregado por manipular os dados das disciplinas para visualização a partir de um determinado curso para usuários não vinculados a FURB. Já o arquivo `app.js` incumbido de construir as rotas da aplicação, e assim encaminhar um determinado arquivo para sua respectiva *view*. E, o arquivo de `templates.html` contém os arquivos que possuem as respectivas telas de cada arquivo.

Após concluir a implementação, o Ionic é utilizado para fazer a construção do aplicativo para código nativo de algum sistema operacional mobile, que no caso dessa aplicação será gerada uma Android Package (APK) para o sistema Android.

3.3 IMPLEMENTAÇÃO

Nesta seção são mostradas as técnicas e ferramentas utilizadas e a operacionalidade da implementação. A seção 3.3.1 apresenta o detalhamento das ferramentas e as técnicas utilizadas. A seção 3.3.2 demonstra o processo operacional da implementação.

3.3.1 Técnicas e ferramentas utilizadas

A aplicação foi desenvolvida em duas partes diferentes, uma delas sendo a parte de servidor e a outra a parte a cliente (aplicação mobile). Na parte servidor, foi utilizado a *Integrated Development Environment* (IDE) Visual Studio 2013 e a linguagem de programação utilizada foi o C#. Na parte de comunicação é utilizado o *framework* de comunicação Asp.Net Web *Application Programming Interface* (API) para a construção dos serviços. Para o armazenamento de dados foi utilizada o Oracle 11g e para fazer o mapeamento das classes e conexão com o banco de dados é utilizado o *Object-Relational Mapping* (ORM) *Entity Framework* juntamente com *Code First*.

Na aplicação móvel foi utilizada a IDE *Brackets* para o desenvolvimento da aplicação. Para a codificação foram utilizados recursos como *HyperText Markup Language* (HTML), *Cascading Style Sheets* (CSS) e Javascript e para a construção do aplicativo foi utilizado o *framework* para aplicações híbridas Ionic que por sua vez utiliza o *framework* Javascript AngularJS e é executado pelo NodeJS. A versão do *Software Development Kit* (SDK) do

Android mínima escolhida foi a 16, que é a versão 4.1 do Android (Jelly Bean), ou seja, a aplicação funciona para qualquer celular com Android na versão Jelly Bean ou superior.

3.3.2 Etapas da implementação

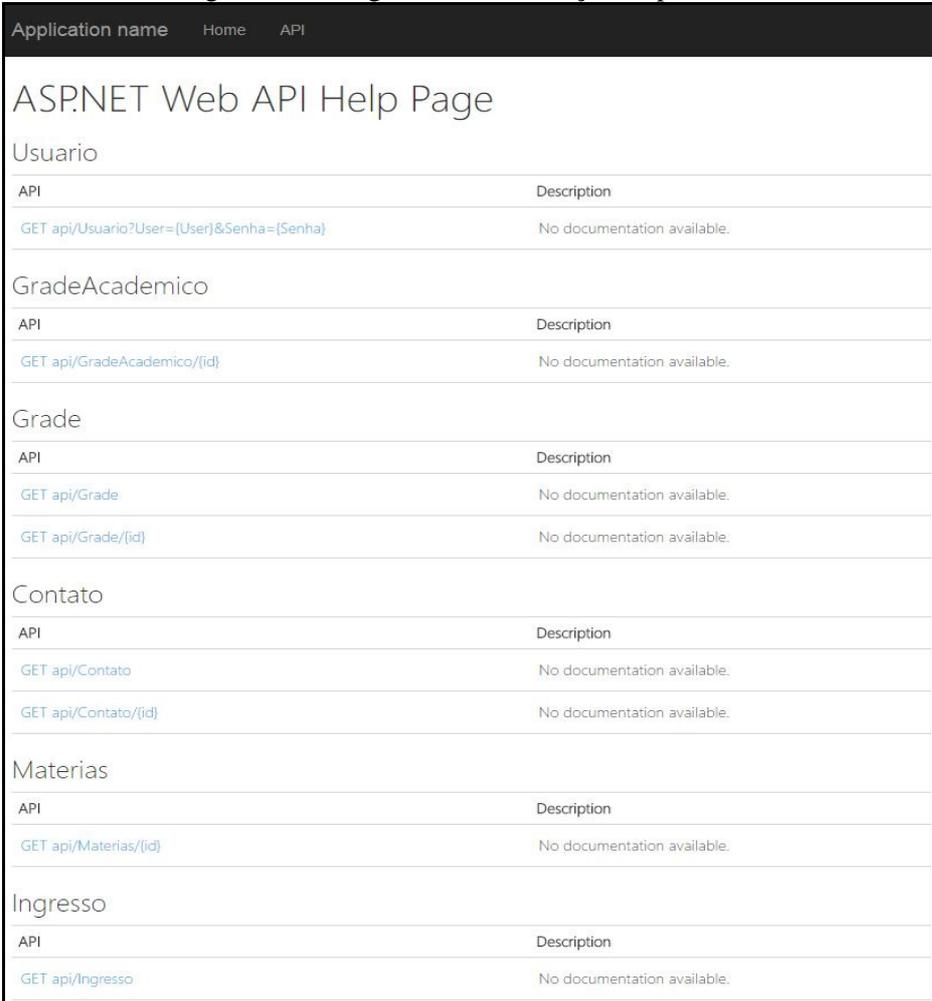
O funcionamento da aplicação será explicado em 2 etapas. A primeira detalha a API utilizada para a construção do *webservice* (seção 3.3.2.1) e, na sequência (seção 3.3.2.2) a aplicação mobile.

3.3.2.1 Web API

No desenvolvimento da aplicação, mais especificamente na parte servidor, é utilizada a Web API. Ela é utilizada para recuperar os valores requisitados a partir de uma *Uniform Resource Locator* (URL).

No projeto, essa interface é utilizada para retornar os dados requisitados pela aplicação mobile em formato JSON. A Figura 8 apresenta uma página *web* que contém uma listagem com todos os serviços disponíveis para a aplicação.

Figura 8 – Listagem com os serviços disponíveis



Application name Home API	
ASP.NET Web API Help Page	
Usuario	
API	Description
GET api/Usuario?User={User}&Senha={Senha}	No documentation available.
GradeAcademico	
API	Description
GET api/GradeAcademico/{id}	No documentation available.
Grade	
API	Description
GET api/Grade	No documentation available.
GET api/Grade/{id}	No documentation available.
Contato	
API	Description
GET api/Contato	No documentation available.
GET api/Contato/{id}	No documentation available.
Materias	
API	Description
GET api/Materias/{id}	No documentation available.
Ingresso	
API	Description
GET api/Ingresso	No documentation available.

Para fazer as requisições, são utilizadas *controllers* que herdam da *controller* da interface Web API. Cada serviço possui uma URL que aponta para a *controller* específica, juntamente com o método a ser utilizado e os parâmetros necessários para esse método. No Quadro 2 pode-se ver como é feita a busca dos dados a partir de uma URL, como por exemplo, `http://ip:porta/api/GradeAcademico/idAluno`.

Quadro 2 – Classe de retorno da entidade `GradeAcademico`

```

1 public class GradeAcademicoController : ApiController
2 {
3     public List<GradeAcademico> Get(int id)
4     {
5         var gradeAcademicoCompleta = new List<GradeAcademico>();
6
7         using (var db = new FurbBaseContext())
8         {
9             gradeAcademicoCompleta = db.GradeAcademico
10                .Include("Materia").Include("Materia.Horarios")
11                .Where(x => x.UsuarioId == id).ToList();
12        }
13        return gradeAcademicoCompleta;
14    }
15 }

```

O código do Quadro 2, retorna uma lista de objeto `GradeAcademico` através do método `Get`, que espera de parâmetro um `id` do tipo `int`. Depois disso, se instancia o contexto da base de dados e faz a busca da grade do acadêmico incluindo as dependências da classe `Materia` e da classe `Horario`, além de fazer um filtro utilizado uma expressão lambda comparando o `id` do usuário com o parâmetro enviado. O Quadro 3 mostra o retorno de uma chamada para o serviço de usuários, que é retornada em formato JSON.

Quadro 3 – Retorno JSON do serviço de usuários

```

1 {"ID":4,"Nome":"Filipe Franz
2   Barreto","CPF":"78563214965","Email":"fbarreto@furb.br",
3   "DataNascimento":"1992-12-21T00:00:00","Telefone":"96328654",
4   "Endereco":"Rua do teste nº zero","Login":"fbarreto",
5   "Senha":"123456","Ativo":"S"}

```

Na parte do servidor, também é utilizado o ORM *Entity Framework Code First*. Este *framework* é utilizado com o intuito de facilitar a manipulação dos dados, sendo um software orientado a objeto e utilizando banco de dados relacionais, assim o *framework* fica responsável por buscar e persistir os dados. No Quadro 4 pode-se verificar a construção do contexto do banco de dados da aplicação.

Quadro 4 – Criação do contexto do banco de dados

```

1 public class FurbBaseContext : DbContext
2 {
3     public DbSet<Usuario> Usuario { get; set; }
4     public DbSet<GradeAcademico> GradeAcademico { get; set; }
5     public DbSet<Materia> Materia { get; set; }
6     public DbSet<Grade> Grade { get; set; }
7     public DbSet<Horario> Horario { get; set; }
8     public DbSet<Ingresso> Ingresso { get; set; }
9     public DbSet<Contato> Contato { get; set; }
10 }

```

O contexto do banco de dados é utilizado para mapear as classes para o banco, assim como, para vincula-las a um objeto. Para isso, foi utilizada classe `DbSet`. O *Code First* é o responsável por controlar todos os aspectos do mapeamento das classes com o banco de dados, entre eles o nome da tabela no banco e seus respectivos campos, como pode ser visto no Quadro 5.

Quadro 5 – Classe `GradeAcademico` mapeada

```

1 [Table("GRADE_ACADEMICO", Schema = "SYSTEM")]
2 public class GradeAcademico
3 {
4     [Key]
5     [Column("GA_ID")]
6     public int ID { get; set; }
7
8     [Column("GA_STATUS")]
9     public string Status { get; set; }
10
11     [Column("GA_IDMATERIA")]
12     public int? MateriaId { get; set; }
13     [ForeignKey("MateriaId")]
14     public Materia Materia { get; set; }
15
16
17     [Column("GA_IDUSUARIO")]
18     public int UsuarioId { get; set; }
19     [ForeignKey("UsuarioId")]
20     public Usuario Usuario { get; set; }
21 }

```

O código acima mostra que o mapeamento é feito através de *DataAnnotations*, que são os atributos localizados em cima da propriedade ou da própria classe. Olhando para a primeira linha, pode-se ver que o atributo `Table` está recebendo como parâmetro o nome da tabela no banco de dados e também o nome do *schema* utilizado. Na linha 4 pode-se ver o atributo `Key`, que serve para indicar que é a chave primária da tabela. O atributo `Column` existente na linha 5, serve para indicar qual é a respectiva coluna na tabela. Na linha 11, pode-se visualizar como é feita a relação com outra tabela, primeiro deve se ter o `id` da tabela correspondente e sua coluna. Esse `id` servirá de parâmetro para o atributo `ForeignKey` que terá uma propriedade com o objeto respectivo a sua tabela vinculada.

Como a aplicação mobile é híbrida, se trata de aglomerado de páginas HTML *web*, por sua vez elas possuem rotas ou URLs para direcionar para algum lugar. No caso de aplicações híbridas que utilizam AngularJS, esses lugares são as *controllers*. Mas para que isso aconteça, a aplicação utiliza `$stateProvider` e `$urlRouterProvider` na sua classe de configuração chamada de `app.js`. É nela que são feitos os vínculos das *views* ou telas para suas respectivas *controllers*, conforme pode ser visto no trecho do Quadro 6.

Quadro 6 – Configuração de rotas da aplicação mobile na classe `app.js`

1	<code>.config(function (\$stateProvider, \$urlRouterProvider) {</code>
2	<code> \$stateProvider</code>
3	<code> .state('app', {</code>
4	<code> url: '/app',</code>
5	<code> abstract: true,</code>
6	<code> templateUrl: 'templates/menu.html'</code>
7	<code> })</code>
8	
9	<code> .state('login', {</code>
10	<code> url: '/login',</code>
11	<code> templateUrl: 'templates/login.html',</code>
12	<code> controller: 'LoginCtrl'</code>
13	<code> })</code>

Em AngularJS, existem estruturas para centralizar funções que serão usadas em diversas partes da aplicação, uma dessas estruturas é o `factory`. O `factory` pode retornar qualquer tipo de dado ao ser chamado em uma `controller`. No Quadro 7 mostra a utilização desta estrutura.

Quadro 7 – Factory utilizada no sistema

1	<code>.factory('dataService', ['\$http', function (\$http) {</code>
2	
3	<code> function getGradeJson(id)</code>
4	<code> return \$http.get('http://ip:porta/api/GradeAcademico/'+id)</code>
5	<code> .then(function (response) {</code>
6	<code> return response.data;</code>
7	<code> });</code>
8	<code> }; return {</code>
9	<code> getGradeJson: getGradeJson</code>
10	<code> };</code>
11	<code> });</code>

No Quadro 7, pode-se ver que dentro de uma `factory` são criadas as funções que irão buscar os dados no servidor. Através do comando `$http.get` que recebe a URL de um serviço específico e, em caso de sucesso, essa função retorna os dados que serão retornados para `controller` que chamou a função. O Quadro 8 mostra a utilização do `factory`.

Quadro 8 – Utilização da *factory* na *controller* *coursesCtrl*

```

1  .controller('coursesCtrl', ['$scope', 'dataService', function ($scope,
2  dataService) {
3
4      $scope.listaCursos = {};
5      dataService.getCursosJson().then(function (dadosCursos) {
6          $scope.listaCursos = dadosCursos;
7      });
8
9  });

```

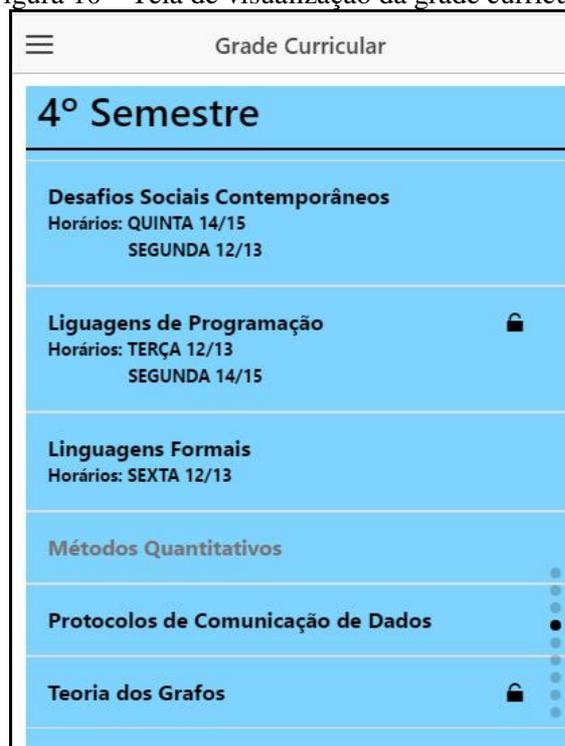
3.3.2.2 Aplicação mobile

A aplicação possui duas formas de visualização: usuários alunos e visitantes. Os usuários alunos podem entrar na aplicação através do mesmo usuário e senha do Ambiente Virtual de Aprendizagem (AVA), onde poderão visualizar informações específicas sobre sua grade curricular, como por exemplo pré-requisitos de uma matéria, horários das aulas de suas disciplinas matriculadas no semestre entre outras. A Figura 9 mostra a tela de *login* da aplicação para alunos da FURB. Já o modo visitante, permite navegar e ver informações relacionadas aos cursos oferecidos pela FURB. Este modo será explicado mais para frente.

Figura 9 – Tela de *login*

Para que o aluno possa visualizar a sua matriz curricular depois de efetuar o *login*, foi criada uma tela onde é visto algumas informações sobre andamento do curso ao qual o aluno está matriculado. Nela, são visualizadas todas as disciplinas do curso divididas por semestre, sendo que é possível visualizar se a disciplina possui pré-requisitos em aberto. Assim como, também mostra os horários das disciplinas ao qual o acadêmico está cursando no momento. E, para ajudar o acadêmico, a aplicação sinaliza se ele já foi aprovado ou não na referida disciplina, conforme mostra a Figura 10.

Figura 10 – Tela de visualização da grade curricular



Olhando a tela apresentada na Figura 10, pode-se observar uma barra lateral que indica a quantidade de semestres do curso. O aluno pode ir para as outras telas fazendo movimento horizontal com o dedo sobre a tela para visualizar os outros semestres, sendo que para cada semestre o tom da cor azul muda, quando mais para o fim do curso mais escuro fica. No alto da tela, em baixo do nome Grade Curricular, pode-se observar a indicação do semestre e suas respectivas disciplinas e os horários, quando elas estão sendo cursadas no semestre corrente. Ao lado do nome da disciplina pode haver um cadeado, que indica que a disciplina possui pré-requisitos e se o mesmo estiver aberto significa que o aluno já foi aprovado nas disciplinas que são pré-requisitos necessárias para cursá-la. Caso o cadeado esteja fechado, isso significa que algum pré-requisito ainda não foi feito. A indicação de que o aluno já foi aprovado na disciplina é representada pelo tom da cor da descrição da disciplina, se o aluno já obteve aprovação, a cor da letra será cinza, do contrário será preta.

Ao selecionar uma disciplina, é possível visualizar as disciplinas que são pré-requisitos para cursá-las, assim como, de quais disciplinas ela é pré-requisito. A Figura 11 exemplifica a seleção da disciplina de teoria dos grafos.

Figura 11 – Tela com disciplinas com pré-requisitos



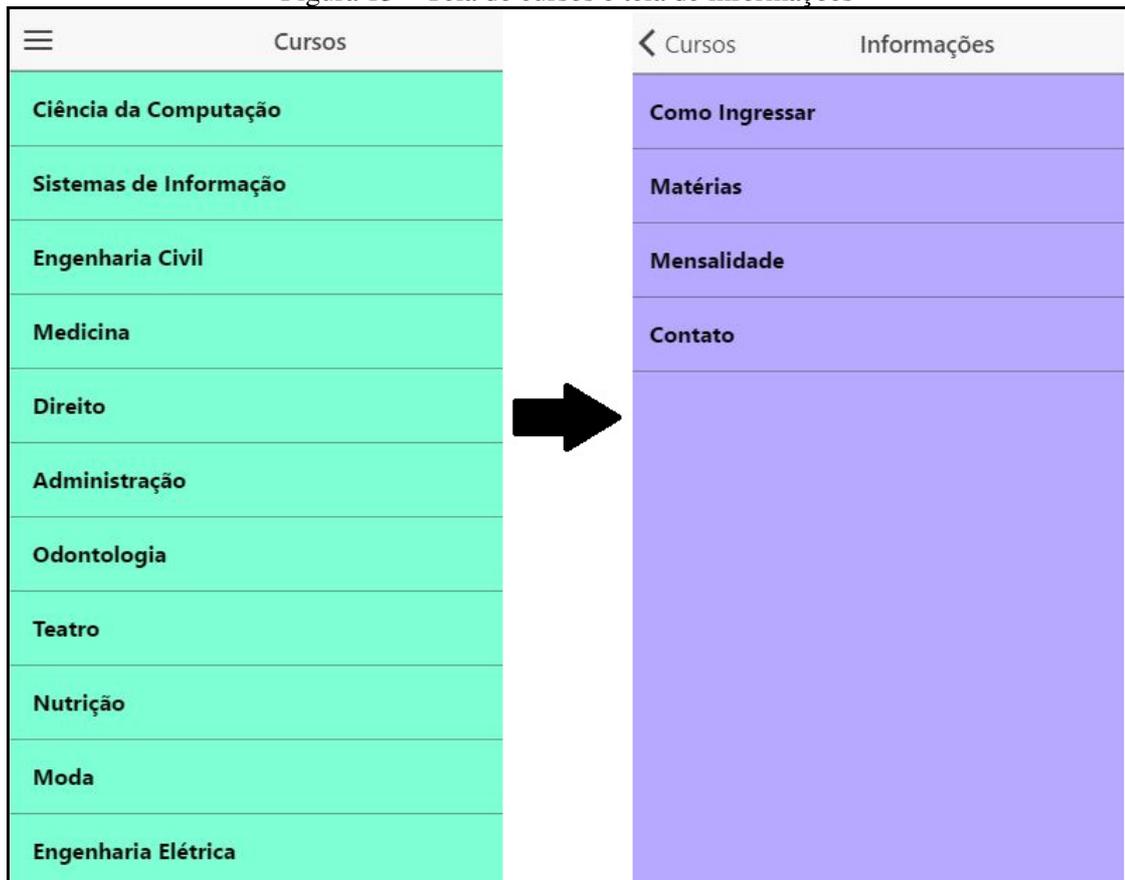
A tela para a visualização de horários é composta pelos dias da semana, de segunda à sábado. Para cada dia da semana, são mostradas as disciplinas que o aluno tem naquele dia. Para cada disciplina é apresentada o nome da dela, o horário, sala e campus da universidade onde será ministrada, conforme mostra a Figura 12.

Figura 12 – Tela de horários



No modo visitante, ao entrar no aplicativo será mostrado uma listagem de todos os cursos ofertados pela FURB, onde o usuário poderá visualizar a grade curricular do curso, mensalidade, formas de contato e como ingressar no curso, conforme exemplifica a tela da Figura 13.

Figura 13 – Tela de cursos e tela de informações



Ao selecionar um curso, o usuário pode-se escolher qual informação deseja visualizar. Uma dessas opções é o como ingressar. Nela, é possível ter acesso às informações sobre os principais modos de ingresso aos cursos da FURB, entre eles estão o vestibular, ENEM e histórico escolar. Ao clicar sobre eles, o usuário será redirecionado para as páginas específicas do portal da FURB, conforme pode ser visto na Figura 14a.

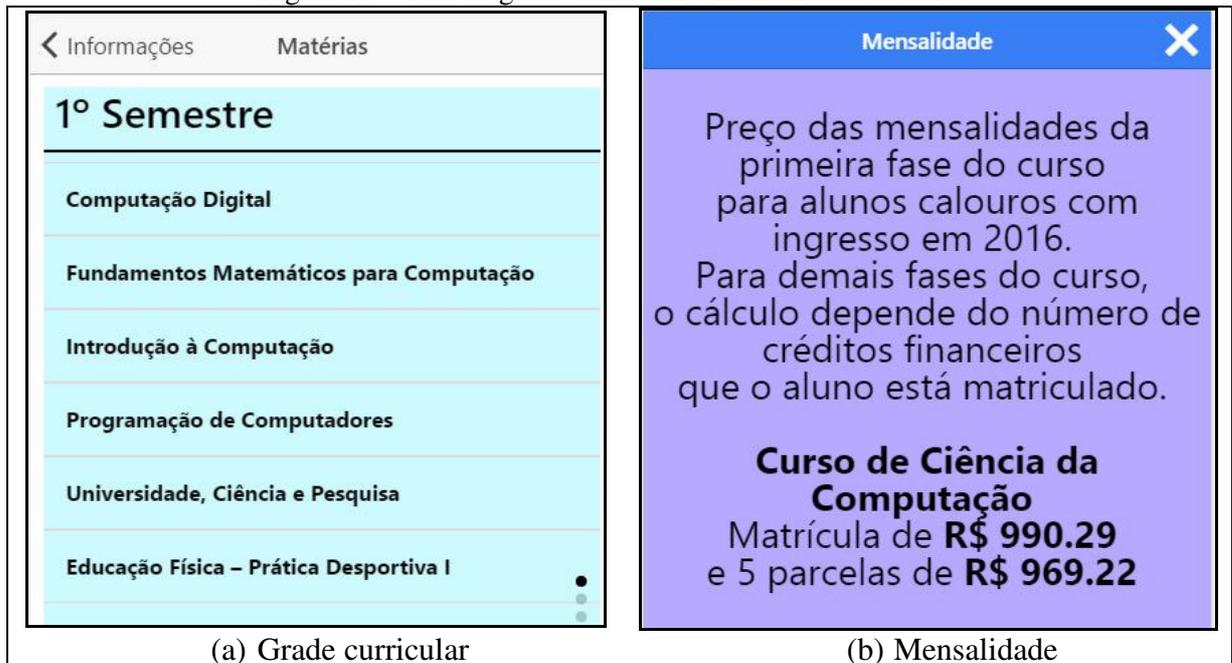
Figura 14 – Tela de como ingressar e contato



Existe também a opção de visualizar as formas de contato específico do respectivo curso. Nesta tela (Figura 14b), pode-se ver o nome do coordenador no curso, local de atendimento, telefone e e-mail para contatos, além do horário de atendimento.

O usuário visitante também pode visualizar a grade curricular do curso. Porém, neste caso, sem as informações de horário e local, conforme mostra a Figura 15a.

Figura 15 – Tela da grade curricular e mensalidade do curso



Outra opção disponível a partir da seleção do curso é a visualização do valor da mensalidade (Figura 15b). Nela, é informado qual o valor da matrícula do curso e o valor das demais parcelas.

3.4 RESULTADOS E DISCUSSÕES

Nesta seção são apresentados os experimentos feitos com o aplicativo. Na seção 3.4.1, detalha-se o cenário, a avaliação do aplicativo, o perfil dos usuários, a aplicação dos testes, a análise e interpretação dos dados coletados. Por fim, a seção 3.4.2 faz a comparação e discussão entre o aplicativo desenvolvido e os trabalhos correlatos.

3.4.1 Experimento do aplicativo

O experimento foi realizado com 10 usuários para avaliar a eficiência e a facilidade de utilização das funcionalidades implementadas no aplicativo.

3.4.1.1 Metodologia

O experimento aconteceu durante o mês de junho por meio de testes em dupla com os usuários. Para realizar os testes foi disponibilizado um dispositivo moto G2, com sistema operacional Android 5.0.2. Neste dispositivo foi disponibilizada uma versão do aplicativo que continha arquivos salvos na própria aplicação. Para o recolhimento de dados foi aplicado um questionário de forma verbal com o usuário.

3.4.1.2 Cenário do teste

O cenário utilizado no teste consiste na disponibilização do dispositivo móvel ao usuário e uma explicação sobre o que a aplicação deve fazer. Cada usuário utilizou o aplicativo por 6 a 12 minutos, para entender e utilizar todas as funções.

3.4.1.3 Aplicação do teste

Para iniciar a avaliação, cada voluntário foi orientado sobre o objetivo do teste e sobre o funcionamento do aplicativo. As perguntas foram feitas em um âmbito geral sobre a aplicação como um todo, deixando livre para a possibilidade de realizar sugestões ou reclamações. As perguntas procuraram obter as impressões do usuário sobre o aplicativo, sobre sua usabilidade e interface gráfica. Os resultados deste experimento são apresentados na próxima seção.

3.4.1.4 Análise e interpretação dos dados coletados

Primeiramente, analisou-se dos dados coletados através do questionário de perfil de usuário. No Quadro 9 são exibidos os perfis dos dez usuários envolvidos no teste de usabilidade.

Quadro 9 – Respostas sobre o perfil do usuário

Sexo	90% masculino e 10% feminino
Idade	30% entre 18 e 25 anos 60% entre 25 e 35 anos 10% mais de 35 anos
Nível de escolaridade	30% ensino superior incompleto 70% ensino superior completo
Possui algum dispositivo móvel	100% possui <i>smartphone/tablet</i>

Através dos resultados tabulados, foi possível observar que todos os voluntários têm mais de 18 anos e sua maioria está na faixa entre 25 e 35 anos e são do sexo masculino e também sua grande maioria possuem superior completo além de todos os voluntários possuírem um *smartphone*. Após análise do perfil dos usuários voluntários, prosseguiu-se

para a avaliação dos resultados obtidos com o questionário de avaliação, como pode ser visto no Quadro 10.

Quadro 10 – Respostas sobre a interface do aplicativo

Perguntas/Respostas	muito boa	boa	médio	ruim
O que você achou da interface do aplicativo?	20%	50%	30%	0%

A metade dos usuários que testaram a aplicação acharam a interface boa, sugerindo que poderia ser melhorado a aparência da tela, deixando-a mais amigável. O Quadro 11 apresenta a avaliação geral do aplicativo.

Quadro 11 – Respostas sobre a avaliação geral do aplicativo

Perguntas/Respostas	muito bom	bom	médio	ruim
O que você achou do aplicativo de um modo geral?	60%	40%	0%	0%

A partir desta questão, pode-se perceber que a grande maioria dos usuários acharam a aplicação boa e entenderam o que aplicativo se propõem a fazer. Posteriormente, foi perguntado se eles usariam o aplicativo, conforme pode ser visto no Quadro 12.

Quadro 12 - Respostas referente a usabilidade do aplicativo

Perguntas/Respostas	sim	talvez	não
Você usaria o aplicativo?	70%	30%	0%

Considerando os resultados das questões acima, pode-se constatar que a aplicação teve boa aceitação. Porém, foram sugeridas algumas melhorias como na interface, mais especificamente no próprio CSS das telas, além de outras funcionalidades, como por exemplo, conter uma versão do AVA só que mobile. Além disso, se questionou se uma pessoa que não é aluno da FURB utilizaria o aplicativo para ver informações sobre o curso. Para alguns usuários, não ficou nítido à primeira vista, que as disciplinas em cinza são disciplinas que o aluno já foi aprovado. Destacaram positivamente a possibilidade de o aluno ver os horários de aula juntamente com sua sala, pois assim seria dispensáveis outros meios de reter essa informação que normalmente são através de papel ou da própria memória, sendo que os horários e salas mudam todo o semestre. De forma geral, pode-se entender que o aplicativo funciona de forma bem intuitiva e de forma simples para a maioria das pessoas que testaram.

3.4.2 Comparação com trabalhos correlatos e discussões

Após a avaliação dos resultados extraídos com os testes e avaliações, o Quadro 13 apresenta um comparativo entre o aplicativo desenvolvido e os trabalhos relacionados.

Quadro 13 – Comparação em relação às características dos trabalhos relacionados

Característica / Trabalho	Rocha (2015)	Kestring (2014)	Sampara (2010)	Trabalho proposto
Aplicação Mobile	Sim	Sim	Sim	Sim
Utiliza Web Service	Não	Sim	Sim	Sim
Desenvolvimento voltado para multiplataforma	Sim	Não	Não	Sim
Facilita a integração de acadêmicos e comunidade com a FURB	Sim	Sim	Não	Sim
Possui acesso a informações dos alunos	Não	Não	Sim	Sim
Facilita o acompanhamento das disciplinas	Não	Não	Sim	Sim
Plataforma	Android e iOS	Android	iOS	Android

Comparando as características do aplicativo desenvolvido com dos trabalhos correlatos, é possível verificar que um dos pontos positivos é a possibilidade de gerar o aplicativo tanto para Android, iOS e Windows Phone. Isso é possível por causa da utilização do *framework* Ionic, que utiliza o *framework* Cordova para rodar as aplicações geradas nos mais diversos tipos de dispositivos móveis. O aplicativo desenvolvido permite um isolamento no que se refere ao servidor, pois utiliza uma estrutura própria para montar e enviar os dados para envio para o aplicativo, não necessitando compartilhar a mesma forma do que é utilizado no AVA (SAMPARA, 2010).

A integração com acadêmico e a sociedade ocorre de uma forma um pouco diferente, pois os aplicativos de Rocha (2015) e Kestring (2014) são voltados para a parte de localizações dentro do espaço da FURB, já o aplicativo desenvolvido atua na área de informações acadêmicas em geral visando integrar tanto os alunos e os visitantes com informações sobre a grade curricular ou sobre os cursos oferecidos pela universidade.

4 CONCLUSÕES

Este trabalho propôs o desenvolvimento de um aplicativo Android que permitisse realizar a visualização de diversas informações relacionada a matriz curricular de um aluno da FURB e, no caso de não alunos, visualizar informações referentes aos cursos oferecidos pela universidade.

Para o desenvolvimento do aplicativo foram utilizados HTML, CSS, Javascript, AngularJS e o *framework* Ionic na construção com auxílio da IDE Brackets. Para a parte de servidor foi utilizada a linguagem de programação C# utilizando o *framework* de serviços Web API, através da IDE do Visual Studio.

Pode-se considerar que os resultados obtidos pelo aplicativo foram satisfatórios, pois conseguiu contemplar em sua totalidade aquilo que se havia proposto, onde a aplicação conseguiu montar uma grade curricular para a visualização das disciplinas divididas por seus respectivos semestres e mostrando seus pré-requisitos. Já para pessoas sem vínculo acadêmico com a FURB, possibilitou a visualização dos cursos e outras informações adicionais.

Referente a implementação, pode-se destacar a falta de conhecimento e de experiência com desenvolvimento de aplicativos móveis tendo como premissa um conceito de aplicação totalmente novo e com a utilização do AngularJS e do Ionic, onde foram utilizados muitos recursos específicos desses *frameworks* para agilizar a construção do aplicativo, além de deixar em aberto a possibilidade de gerar para outros sistemas operacionais moveis. Um ponto negativo que se pode levar em consideração foi a parte visual da aplicação, onde o CSS poderia ser mais trabalhado com o intuito de trazer uma interface mais amigável para os usuários.

Por fim, a integração entre as várias tecnologias utilizadas se mostrou de grande valia devido às possibilidades e facilidades que as ferramentas possuem. Apesar de algumas limitações na implementação principalmente devido a dificuldades de encontrar materiais de estudo ou de exemplo principalmente para o aplicativo mobile.

4.1 EXTENSÕES

Algumas das extensões possíveis para este trabalho são:

- a) implementar um módulo para professores acompanharem seus horários de aula;
- b) desenvolver módulo para acompanhamento de notas das disciplinas que estão sendo cursadas;
- c) implementar um módulo para montar a melhor combinação de disciplinas a serem cursadas, por semestre, para que o aluno possa se formar o mais rápido possível;

- d) desenvolver um módulo para que seja possível fazer matrícula pelo aplicativo;
- e) criar uma base de dados local no aplicativo para reduzir o número de chamadas ao *webservice*, assim sendo possível o aplicativo trabalhar *offline*;
- f) realizar um sincronismo dos dados contidos na base de dados local do dispositivo com a base de dados no *webservice*;
- g) efetuar melhorias na interface gráfica, tornando-a mais amigável.

REFERÊNCIAS

- BALBUINO, Plínio. **Dominando JavaScript com jQuery**. São Paulo: Casa do Código, 2012.
- CAIADO, Elen C. C. **Elaborando a matriz curricular de forma coletiva** [S.I.], 2015. Disponível em: <<http://educador.brasilecola.com/orientacao-escolar/elaborando-matriz-curricular-forma-coletiva.htm>> Acesso em: 20 set. 2015.
- DEPARTAMENTO DE SISTEMAS E COMPUTAÇÃO. [S.I.], 2010 Disponível em: <http://www.inf.furb.br/bcc/curriculo/2010_2/curriculo_bcc_2010_2.pdf> Acesso em: 20 set. 2015
- FADERGS [S.I.], 2014. Disponível em: <<http://www.fadergs.edu.br/fadergs/user/file/Perguntas-Frequentes-Mudanca-de-Matriz-Curricular.pdf>> Acesso em: 06 nov. 2015.
- FURB MOBILE [S.I.], 2016. Disponível em: <http://gcg.inf.furb.br/?page_id=3117> Acesso em: 13 jul. 2016.
- HOPPE, Aurélio F. Matriz Curricular. Blumenau, FURB, 13 nov. 2015. Entrevista a Filipe Franz Barreto.
- KESTRING, Bruno A. **Sistema móvel na plataforma Android para compartilhamento de geolocalização usando mapas e notificações da Google**. 2014. 73f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.
- LOPES, Sérgio. **Aplicações mobile híbridas com Cordova e Phonegap**. São Paulo: Novatec, 2016.
- MONTEIRO, João B. **Google Android: Crie aplicações para celulares e tablets**. São Paulo: Casa do Código, 2012.
- PEREIRA, Michael H. R. **AngularJS: Uma abordagem prática e objetiva**. São Paulo: Novatec, 2014.
- RIPKENS, Ben. **Ionic: AngularJS based framework on the rise** [S.I.], 2014. Disponível em: <<https://blog.codecentric.de/en/2014/11/ionic-angularjs-framework-on-the-rise/>> Acesso em: 23 jun. 2016.
- ROCHA, Diego T. **Tô Aqui: aplicativo para georreferenciamento em ambientes restritos**. 2015. 69f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.
- SAMPARA, Fernanda. **Iava Aplicação web para disponibilizar recursos do AVA no dispositivo móvel Iphone**. 2010. 93f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.
- SESHADRI, Shyam; GREEN, Brad. **Desenvolvendo com AngularJS: Aumento de produtividade com aplicações web estruturadas**. Tradução Lúcia A. Kinoshita. São Paulo: Novatec, 2014.
- SILVA, Maurício S. **jQuery: A biblioteca do programador JavaScript**. 3. ed. São Paulo: Novatec, 2014.

APÊNDICE A – Detalhamento dos casos de uso

A seguir é apresentado o detalhamento dos casos de uso, com uma descrição, pré e pós condições e cenário.

O caso de uso UC01 - Login descreve a interação de aluno e visitante para entrar na aplicação. Detalhes sobre o caso de uso estão descritos no Quadro 14.

Quadro 14 - Caso de uso UC01 - Login

Número	01
Caso de uso	Entrar no sistema
Descrição	Este caso de uso permite fazer a autenticação do aluno ou do visitante no aplicativo
Ator	Aluno e Visitante
Cenário principal	<ol style="list-style-type: none"> 1. O aluno abre a aplicação 2. O aluno insere seu usuário e senha 3. O aluno aperta o botão de entrar 4. O aluno entra no sistema
Fluxo alternativo	<ol style="list-style-type: none"> 1. O visitante abre a aplicação 2. O aluno aperta o botão de entrar como visitante 3. O visitante entra no sistema

O caso de uso UC02 - Visualizar grade curricular descreve a interação entre de aluno e visitante para entrar na aplicação. Detalhes sobre o caso de uso estão descritos no Quadro 15.

Quadro 15 - Caso de uso UC02 - Visualizar grade curricular

Número	02
Caso de uso	Visualizar grade curricular
Descrição	Este caso de uso permite o aluno visualizar sua grade curricular
Ator	Aluno
Cenário principal	<ol style="list-style-type: none"> 1. Será executado UC01 - Login 2. O aluno abre o menu lateral 3. O aluno escolhe a opção grade curricular 4. O aluno visualiza a grade curricular

O caso de uso UC03 - Visualizar pré-requisitos descreve a interação do aluno para visualizar pré-requisitos de uma matéria. Detalhes sobre o caso de uso estão descritos no Quadro 16.

Quadro 16 – Caso de uso UC03 – Visualizar pré-requisitos

Número	03
Caso de uso	Visualizar pré-requisitos
Descrição	Este caso de uso permite o aluno visualizar pré-requisitos de uma matéria
Ator	Aluno
Cenário principal	<ol style="list-style-type: none"> 1. Será executado UC01 - Login 2. Será executado UC02 - Visualizar grade curricular 3. O aluno seleciona uma matéria 4. O aluno visualiza as disciplinas que são pré-requisito da disciplina selecionada

O caso de uso UC04 – Visualizar matérias concluídas descreve a interação do aluno para visualizar disciplinas já concluídas. Detalhes sobre o caso de uso estão descritos no Quadro 17.

Quadro 17 – Caso de uso UC04 – Visualizar matérias concluídas

Número	04
Caso de uso	Visualizar disciplinas concluídas
Descrição	Este caso de uso permite o aluno visualizar disciplinas já concluídas
Ator	Aluno
Cenário principal	<ol style="list-style-type: none"> 1. Será executado UC01 - Login 2. Será executado UC02 - Visualizar grade curricular 3. O aluno verifica as disciplinas já concluídas, estão caracterizadas com a cor da fonte em cinza

O caso de uso UC05 – Visualizar horários descreve a interação do aluno para visualizar seus horários de aula. Detalhes sobre o caso de uso estão descritos no Quadro 18.

Quadro 18 – Caso de uso UC05 – Visualizar horários

Número	05
Caso de uso	Visualizar horários
Descrição	Este caso de uso permite o aluno visualizar seus horários
Ator	Aluno
Cenário principal	<ol style="list-style-type: none"> 1. Será executado UC01 - Login 2. O aluno visualiza seus horários
Fluxo alternativo	<ol style="list-style-type: none"> 1. Caso o aluno esteja visualizando a grade curricular, o aluno deve abrir o menu lateral 2. O aluno escolhe a opção meus horários 3. O aluno visualiza seus horários

O caso de uso UC06 – Visualizar cursos descreve a interação do visitante para visualizar os cursos da FURB. Detalhes sobre o caso de uso estão descritos no Quadro 19.

Quadro 19 – Caso de uso UC06 – Visualizar cursos

Número	06
Caso de uso	Visualizar cursos
Descrição	Este caso de uso permite o visitante visualizar os cursos ofertados pela FURB
Ator	Visitante
Cenário principal	1. Será executado UC01 - Login 2. O visitante visualiza os cursos

O caso de uso UC07 – Visualizar como ingressar descreve a interação do visitante para visualizar como ingressar em um determinado curso da FURB. Detalhes sobre o caso de uso estão descritos no Quadro 20.

Quadro 20 – Caso de uso UC07 – Visualizar como ingressar

Número	07
Caso de uso	Visualizar como ingressar
Descrição	Este caso de uso permite o visitante visualizar como ingressar em um curso
Ator	Visitante
Cenário principal	1. Será executado UC06 – Visualizar cursos 2. O visitante seleciona a opção como ingressar 3. O visitante visualiza as opções de como ingressar

O caso de uso UC08 – Visualizar mensalidade descreve a interação do visitante para visualizar valores da mensalidade de um determinado curso da FURB. Detalhes sobre o caso de uso estão descritos no Quadro 21.

Quadro 21 – Caso de uso UC08 – Visualizar mensalidade

Número	08
Caso de uso	Visualizar mensalidade
Descrição	Este caso de uso permite o visitante visualizar os valores da mensalidade de um curso
Ator	Visitante
Cenário principal	1. Será executado UC06 – Visualizar cursos 2. O visitante seleciona a opção mensalidade 3. O visitante visualiza os valores da mensalidade

O caso de uso UC09 – Visualizar contato descreve a interação do visitante para visualizar formas de contato a um determinado curso da FURB. Detalhes sobre o caso de uso estão descritos no Quadro 22.

Quadro 22 – Caso de uso UC09 - Visualizar contato

Número	09
Caso de uso	Visualizar contatos
Descrição	Este caso de uso permite o visitante visualizar os contatos de um curso
Ator	Visitante
Cenário principal	<ol style="list-style-type: none"> 1. Será executado UC06 – Visualizar cursos 2. O visitante seleciona a opção contato 3. O visitante visualiza as formas de contato do curso

O caso de uso UC10 - Visualizar grade curricular descreve a interação do visitante para visualizar a grade curricular de um curso da FURB. Detalhes sobre o caso de uso estão descritos no Quadro 23.

Quadro 23 – Caso de uso - UC10 - Visualizar grade curricular

Número	10
Caso de uso	Visualizar grade curricular
Descrição	Este caso de uso permite o visitante visualizar a grade curricular do curso
Ator	Visitante
Cenário principal	<ol style="list-style-type: none"> 1. Será executado UC06 – Visualizar cursos 2. O visitante seleciona a opção grade curricular 3. O visitante visualiza a grade curricular do curso