

**FACULDADE DE TECNOLOGIA DE SÃO BERNARDO DO CAMPO  
“ADIB MOISÉS DIB”**

JANDERSON JUNIO RIBEIRO DE SOUZA  
KAIO SILVA COELHO  
RENAN SANTOS DE OLIVEIRA

**SALVALUNO!: Sistema de Gerenciamento de Notas e Faltas**

São Bernardo do Campo – SP  
Dezembro/2017

**JANDERSON JUNIO RIBEIRO DE SOUZA  
KAIO SILVA COELHO  
RENAN SANTOS DE OLIVEIRA**

**SALVALUNO!: Sistema de Gerenciamento de Notas e Faltas**

Trabalho de conclusão de curso apresentado à Faculdade de Tecnologia de São Bernardo do Campo “Adib Moisés Dib” como requisito parcial para a obtenção do título de tecnólogo em Informática para Negócios.

Orientador: Prof. Esp. Douglas Duarte  
Coorientador: Prof. Me. Nelson Afonso Thomaz

São Bernardo do Campo – SP  
Dezembro/2017

**JANDERSON JUNIO RIBEIRO DE SOUZA  
KAIO SILVA COELHO  
RENAN SANTOS DE OLIVEIRA**

**SALVALUNO!: Sistema de Gerenciamento de Notas e Faltas**

Trabalho de conclusão de curso apresentado à Faculdade de Tecnologia de São Bernardo do Campo “Adib Moisés Dib” como requisito parcial para a obtenção do título de tecnólogo em Informática para Negócios.

Trabalho de Conclusão de Curso apresentado e aprovado em: 29/11/2017.

Banca examinadora:

---

Prof. Esp. Douglas Duarte, FATEC SBC – Orientador

---

Prof. Esp. Roberto Gabriel Labrada, FATEC SBC – Avaliador

---

Prof. Me. Samáris Ramiro Pereira, FATEC SBC – Avaliador

Dedicamos este trabalho a todos os amantes e profissionais de tecnologia. Dedicamos também, em especial, ao nosso amigo Daniel Bernardo por idealizar o nome do nosso projeto.

Agradecemos primeiramente a Deus por nos dar saúde, calma e força para seguir em frente, aos professores da FATEC-SBC pela atenção e conhecimento transmitidos, que contribuíram para a realização do trabalho, aos nossos amigos por nos ajudar durante o projeto e aos nossos familiares pelo apoio e compreensão indispensáveis.

“É impossível progredir sem mudança, e aqueles que não mudam suas mentes não podem mudar nada.”

GEORGE BERNARD SHAW

## RESUMO

No mundo atual, é praticamente impossível imaginar a vida das pessoas sem o uso da tecnologia. Dispositivos eletrônicos, tais como smartphones, tablets, notebooks etc. dispõem de uma enorme variedade de recursos que organizam o dia-a-dia do usuário além de facilitar a comunicação e a manipulação de informações. O SalvAluno!, nome dado a este projeto, é um sistema baseado em um aplicativo mobile que vem com o propósito de “salvar” os estudantes naqueles momentos em que esquecem aquele cálculo para obter a nota necessária em uma matéria, quando eles não sabem, ao certo, quantas faltas teve ou, então, quando não veem algum aviso importante no mural de informações da instituição em que frequentam. Ao cadastrar as informações referentes à instituição, o estudante pode facilmente gerenciar suas notas e faltas de modo off-line, a partir do seu dispositivo móvel. Além disso, é possível resolver problemas de comunicação, imprevistos etc., com o mural eletrônico.

**Palavras-chave:** Tecnologia da Informação. Aplicativo Mobile. Comunicação Online. Inovação. Otimização.

## **ABSTRACT**

In the current world, it is practically impossible to imagine people's lives without the use of technology. Electronic devices such as smartphones, tablets, notebooks etc have a huge variety of features that organize the user's daily life and facilitate the communication and manipulation of information. SalvAluno!, the name given to this project, is a system based on a mobile application that comes with the purpose of “saving” the students in those moments in which they forget that calculation to obtain the necessary grade in a subject, when they do not know, for certain, how many absences they had, or then, when they don't see some important notifications on the information wall of the institution where they attend. By registering information related to the institution, the student can easily manage their grades and absences in offline mode, from their mobile device. Besides, it is possible to solve communication problems, unforeseen events etc., with the electronic wall.

**Keywords:** Information Technology. Mobile Application. Online Communication. Innovation. Enhancement.

## LISTA DE QUADROS

Quadro 1.1 – Elementos da profissão do empreendedor.....	20
--	----

## LISTA DE FIGURAS

Figura 1.1 – Exemplo de modelo canvas .....	22
Figura 1.2 – Arquitetura SQL.....	27
Figura 1.3 – Modelo do QR Code.....	30
Figura 1.4 – Codificação do QR Code.....	31
Figura 1.5 – Tamanho mínimo de impressão do QR Code .....	31
Figura 1.6 – Capabilidade Kanji e Kana .....	32
Figura 1.7 – Sujeira e dano em um código QR .....	32
Figura 1.8 – Padrões de detecção de posição do QR .....	33
Figura 1.9 – Recurso de anexação estruturada do QR .....	34
Figura 2.1 – Cronograma do Projeto .....	38
Figura 3.1 – Modelo de negócios do SalvAluno! .....	45
Figura 3.2 – UML SalvAluno!.....	52
Figura 3.3 – Classe Manter Instituição .....	53
Figura 3.4 – Classes Manter Aluno .....	54
Figura 3.5 – Classes Manter Matéria .....	55
Figura 3.6 – Classes Manter Instrutor .....	56
Figura 3.7 – Classes Manter Aula .....	57
Figura 3.8 – Classes Manter Mural .....	58
Figura 3.9 – Classe Ver Instituição.....	58
Figura 3.10 – Classes Manter Dados Instrutor.....	59
Figura 3.11 – Classes Ver Aluno.....	60
Figura 3.12 – Classes Manter Dados Matéria .....	61
Figura 3.13 – Classes Manter Nota .....	62
Figura 3.14 – Classes Manter Atividade.....	63
Figura 3.15 – Classes Manter Dados Aula.....	64
Figura 3.16 – Classes Manter Faltas.....	65
Figura 3.17 – Classes Ver Mural .....	66
Figura 3.18 – Classes Manter Anotação .....	67
Figura 3.19 – Banco de Dados.....	72
Figura 3.20 – Tela Splash .....	75
Figura 3.21 – Tela Main (Principal) .....	77

Figura 3.22 – Tela Main 2 (Principal) .....	79
Figura 3.23 – Tela ID.....	80
Figura 3.24 – Carteirinha do estudante .....	81
Figura 3.25 – Tela QR Code .....	82
Figura 3.26 – Tela Mural .....	83
Figura 3.27 – Tela Dados .....	84
Figura 3.28 – Tela Matéria .....	85
Figura 3.29 – Tela Atividades.....	86
Figura 3.30 – Tela Recuperação .....	87
Figura 3.31 – Tela Aulas .....	88
Figura 3.32 – Tela Anotações .....	89
Figura 3.33 – Tela Nota.....	90
Figura 3.34 – Tela Faltas .....	91
Figura 3.35 – Tela Definições.....	92
Figura 3.36 – Tela Faculdade .....	93
Figura 3.37 – Tela Instrutores .....	94
Figura 3.38 – Tela Sobre.....	95

## LISTA DE GRÁFICOS

Gráfico 3.1 – Idade dos entrevistados.....	40
Gráfico 3.2 – Localidade dos pesquisados.....	40
Gráfico 3.3 – Escolaridade dos entrevistados .....	41
Gráfico 3.4 – Uso do mural pelos pesquisados .....	41
Gráfico 3.5 – Recordação dos entrevistados para trabalhos escolares .....	42
Gráfico 3.6 – Controle de faltas dos entrevistados.....	43
Gráfico 3.7 – Comunicação entre pesquisados e professores .....	43
Gráfico 3.8 – Opinião do eleitorado sobre a comunicação da instituição .....	44
Gráfico 3.9 – Usabilidade do SalvAluno! .....	44

## LISTA DE TABELAS

Tabela 3.1 – Peso dos Atores .....	68
Tabela 3.2 – Casos de Uso e Pesos .....	68
Tabela 3.3 – Fatores de Complexidade Técnica .....	69
Tabela 3.4 – Fatores de Complexidade Ambiental.....	70
Tabela 3.5 – Gestão do Custo de Software .....	71

## LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
APK	Android Package File
GPS	Global Position System
IDE	Integrated Development Environment
IHC	Interação Humano Computador
JAR	Java Archive
QR	Quick Response Code
RF	Requisitos Funcionais
RNF	Requisitos Não Funcionais
SDK	Software Development Kit
SQL	Structure Query Language
SWOT	Strengths, Weaknesses, Oportunities and Threats
TIC	Tecnologias da Informação e Comunicação
TCC	Trabalho de Conclusão de Curso
UAW	Unadjusted Actor Weight
UCP	Use Case Points
UML	Unified Modeling Language
URL	Uniform Resource Locator
UUCW	Unadjusted Use Case Weight
UUCP	Unadjusted Use Case Points

## SUMÁRIO

<b>INTRODUÇÃO</b> .....	16
<b>1 FUNDAMENTAÇÃO TEÓRICA</b> .....	18
1.1 <b>Definição de sistema</b> .....	18
1.2 <b>As TICs</b> .....	19
1.3 <b>Conceito de empreendedorismo</b> .....	19
1.4 <b>Modelo de negócios</b> .....	20
1.4.1 <b>Análise FOFA</b> .....	22
1.5 <b>Definição de software e ferramentas</b> .....	23
1.5.1 <b>Android</b> .....	23
1.5.2 <b>Eclipse IDE</b> .....	25
1.5.3 <b>Java</b> .....	25
1.5.4 <b>SQLite</b> .....	26
1.5.5 <b>Maven</b> .....	27
1.5.6 <b>API</b> .....	28
1.5.7 <b>API Dropbox V2</b> .....	29
1.5.8 <b>QR Code</b> .....	30
<b>2 METODOLOGIA</b> .....	35
2.1 <b>Cronograma de atividades</b> .....	36
<b>3 DESENVOLVIMENTO</b> .....	39
3.1 <b>Análise e tratamento de dados</b> .....	39
3.2 <b>Modelo de negócios do SalvAluno!</b> .....	45
3.3 <b>Termo de abertura do projeto</b> .....	45
3.4 <b>Requisitos do projeto</b> .....	47
3.4.1 <b>Requisitos funcionais</b> .....	47
3.4.2 <b>Requisitos não funcionais</b> .....	49
3.4.3 <b>Requisitos de domínio</b> .....	49
3.5 <b>UML</b> .....	50
3.5.1 <b>Casos de uso e diagrama de classes</b> .....	50
3.6 <b>Métrica UCP</b> .....	67
3.7 <b>Banco de dados</b> .....	71
3.8 <b>Descrição do método de desenvolvimento</b> .....	73
3.8.1 <b>Diário de Bordo</b> .....	73
3.8.2 <b>Telas</b> .....	74
3.9 <b>Testes</b> .....	96

<b>CONSIDERAÇÕES FINAIS .....</b>	<b>97</b>
<b>REFERÊNCIAS.....</b>	<b>98</b>
<b>GLOSSÁRIO.....</b>	<b>100</b>
<b>APÊNDICE – QUESTIONÁRIO SALVALUNO!.....</b>	<b>101</b>

## INTRODUÇÃO

O avanço da tecnologia vem modificando, cada vez mais, a comunicação e o acesso à informação. Atualmente, as chamadas TICs ou Tecnologias da Informação e Comunicação, permitem que as pessoas se interajam de maneira cada vez mais rápida e acessível.

Isso não seria diferente na vida dos estudantes. Devido à correria do dia a dia, tem momentos em que fica difícil organizar as tarefas ou gerenciar as notas e faltas. Além disso, pode acontecer também deles esquecerem aquela fórmula para calcular a nota necessária em uma atividade, trabalho ou prova ou deixar de ver um aviso importante no mural da instituição em que frequentam. Sabendo-se do grande impacto da tecnologia causado na rotina das pessoas, um aplicativo mobile poderia resolver todos esses impasses. É a partir desta necessidade que o SalvAluno! foi imaginado.

Ele é um sistema que vai permitir, de forma simples e prática, manter o usuário atento a prazos, aulas e informativos da faculdade ou curso que esteja prestando. Além disso, ele visa armazenar registro de faltas e matérias cursadas através de parâmetros pré-configurados.

Esses parâmetros recebidos da instituição, em sua maioria, são ajustáveis para se encaixar com as definições do usuário. É possível alterar de forma geral os dias das aulas e seus horários, professor e também as atividades de uma matéria, seja adicionando novas atividades ou removendo-as, alterar suas datas, os nomes das avaliações, os pesos das notas.

Ao adicionar uma disciplina ao período cursado, os dados da mesma são copiados dos dados gerais do sistema e, desta forma, pode-se também modificar as matérias individualmente após adicionadas, pois, desta forma permite-se que as notas finais sejam calculadas de maneira diferente às demais disciplinas da instituição, caso assim seja necessário.

O aplicativo foi pensado, principalmente, com o intuito de auxiliar o estudante nas suas tarefas diárias. Apesar de já existir um sistema integrado de gestão acadêmica da faculdade, com o SalvAluno!, uma vez cadastradas as informações referentes à instituição que frequenta, o aluno poderá facilmente gerenciar suas notas e faltas off-line, a partir do seu dispositivo móvel. Além disso, será possível resolver problemas de comunicação relacionadas a informações vinculadas à instituição, imprevistos etc., com o mural eletrônico.

O objetivo específico desse projeto é desenvolver um sistema que mescle recursos da informática com a gestão, por meio de um aplicativo que auxilie os estudantes a organizar as tarefas do dia-a-dia da faculdade, controlar suas notas e faltas e estar a par dos eventos que ocorrem na instituição que eles frequentam.

O projeto está estruturado em três capítulos. O primeiro capítulo é constituído da fundamentação teórica e conceitual do projeto, a partir de pesquisas que sustentem as teorias e os conceitos abordados, artigos, livros, documentos eletrônicos e permitam embasar e aprofundar o tema; o segundo descreve a metodologia a ser utilizada e o terceiro apresenta o desenvolvimento do projeto; ao final serão apresentadas as considerações finais. Com isso, espera-se que o aplicativo contribua na organização das tarefas do estudante e, conseqüentemente, melhore seu desempenho nos estudos.

# 1 FUNDAMENTAÇÃO TEÓRICA

Para falar do SalvAluno!, é necessário descrever e embasar os conceitos referentes a sistema, às tecnologias de informação e comunicação, ao empreendedorismo e aos softwares e ferramentas que serão utilizados nele.

## 1.1 Definição de sistema

O conceito de sistema é bastante abrangente e pode ser estendido em várias áreas de conhecimento.

Segundo Chiavenato (1983, p.515), sistema é “um todo organizado e complexo; um conjunto ou combinação de coisas ou partes, formando um todo complexo ou unitário.”

Um sistema pode ser definido também como um conjunto de elementos dinamicamente relacionados entre si, formando uma atividade para atingir um objetivo, operando sobre entradas e fornecendo saídas processadas (CHIAVENATO, 1983).

Para Optner (1971), todo sistema se caracteriza por determinados parâmetros:

- a) Entradas (input): é a força de arranque do sistema, aquilo que fornece o material ou energia para a operação do sistema.
- b) Processo: é o mecanismo de conversão das entradas em saídas ou resultados.
- c) Saídas (output): são os resultados de um processo.
- d) Retroação (feedback): é a função de sistema que visa comparar a saída com um critério ou padrão previamente estabelecido, no sentido de ajustar ou regular o funcionamento do sistema.
- e) Ambiente: é o meio que envolve externamente o sistema.

## 1.2 As TICs

Para Barbosa e Silva (2011, p. 2), as TICs “oferecem maneiras eficientes de processar e trocar informações com diversos objetivos”. Elas permitem criar sistemas computacionais embutidos nos mais diferentes dispositivos eletrônicos, que combinam poder computacional e meios de comunicação (telefonia, rádio, TV, internet etc.)

[...]as TICs estão ocupando espaço nas nossas vidas. Quando as incorporamos no nosso cotidiano, não estamos apenas trocando instrumentos como quem troca de garfo, caneta ou régua. As modificações são mais profundas e significativas, pois modificam também a nossa forma de trabalhar, de prestarmos serviços, de nos relacionarmos com outras pessoas e instituições, de ensinarmos e aprendermos, de participarmos da política, de lidarmos com o dinheiro, de cuidarmos da saúde, e assim por diante[...] (BARBOSA, S. D. J.; SILVA, B. S. da, 2011, p. 5)

## 1.3 Conceito de empreendedorismo

O termo empreendedorismo, ou *entrepreneurship*, em inglês, significa “aprender a fazer coisas novas ou desenvolver maneiras novas e diferentes de fazer essas coisas e essa é a tarefa designada ao empreendedor” (FILION; DOLABELA, 2000, p. 17).

Entre as várias definições de empreendedorismo, encontram-se pelo menos os aspectos os quais serão descritos a seguir:

- 1) iniciativa para criar um novo negócio e paixão pelo que faz;
- 2) utiliza os recursos disponíveis de forma criativa transformando o ambiente social econômico onde vive;
- 3) aceita assumir os riscos e a possibilidade de fracassar.

O Quadro 1.1 identifica os principais elementos que caracterizam a profissão de um empreendedor, baseado nas atividades críticas, nas competências e o aprendizado adquirido.

Quadro 1.1 – Elementos da profissão do empreendedor

<b>ELEMENTOS DA PROFISSÃO DE EMPREENDEDOR</b>			
<b>ATIVIDADES CRÍTICAS</b>	<b>CARACTERÍSTICAS</b>	<b>COMPETÊNCIAS</b>	<b>APRENDIZADO</b>
Identificar oportunidades de negócio	Faro / Intuição	Pragmatismo	Análise setorial
Conceber visões	Imaginação / Independência / Paixão	Concepção / Pensamento sistêmico	Avaliação de recursos
Tomar decisões	Julgamento / Prudência	Visão	Informação / Risco
Realizar visões	Flexibilidade / Constância / Tenacidade	Ação	Feedback
Fazer o equipamento funcionar	Destreza	Polivalência	Técnica
Comprar	Acuidade	Negociação	Diagnostico
Colocar no mercado	Diferenciação / Originalidade	Agenciamento	Marketing / Gestão
Vender	Flexibilidade	Adaptação	Conhecimento do Cliente
Cercar-se	Julgamento	Comunicação	Gestão de RH / Compartilhar
Delegar	Discernimento	Relações / Equipe	Holismo / Gestão operacional

Fonte: Autoria própria, adaptada de FILION; DOLABELA, 2000, p. 227

Para Dornelas (2016), o empreendedor é aquele que identifica uma nova oportunidade, desenvolve um negócio e planeja a geração de capital sobre ela, ciente de assumir riscos calculados.

#### **1.4 Modelo de Negócios**

Segundo o site do Sebrae (2017), qualquer negócio é um sistema. Um modelo é uma descrição de um sistema. Essa descrição pode ser feita de forma linear, com textos e número, por exemplo, ou de forma visual, como um desenho ou um gráfico.

O modelo de negócio é um esquema para o desenvolvimento de uma estratégia organizacional, que será implementada através das estruturas, processos e sistemas da empresa. Conhecido também como modelo canvas, foi criado por Alex Osterwalder e Yves Pigneur.

O modelo de negócios é a possibilidade de visualizar a descrição do negócio, das partes que o compõem, de forma que a ideia sobre o negócio seja compreendida por quem lê da forma como pretendia o dono do modelo. Usar recursos visuais é também uma maneira divertida de trabalhar de forma colaborativa.

Desenhar o modelo de negócios precede a elaboração do plano de negócios. É por meio da análise e reflexão sobre ele que será possível perceber se a ideia original terá validade, se todas as partes se encaixam formando verdadeiramente um sistema.

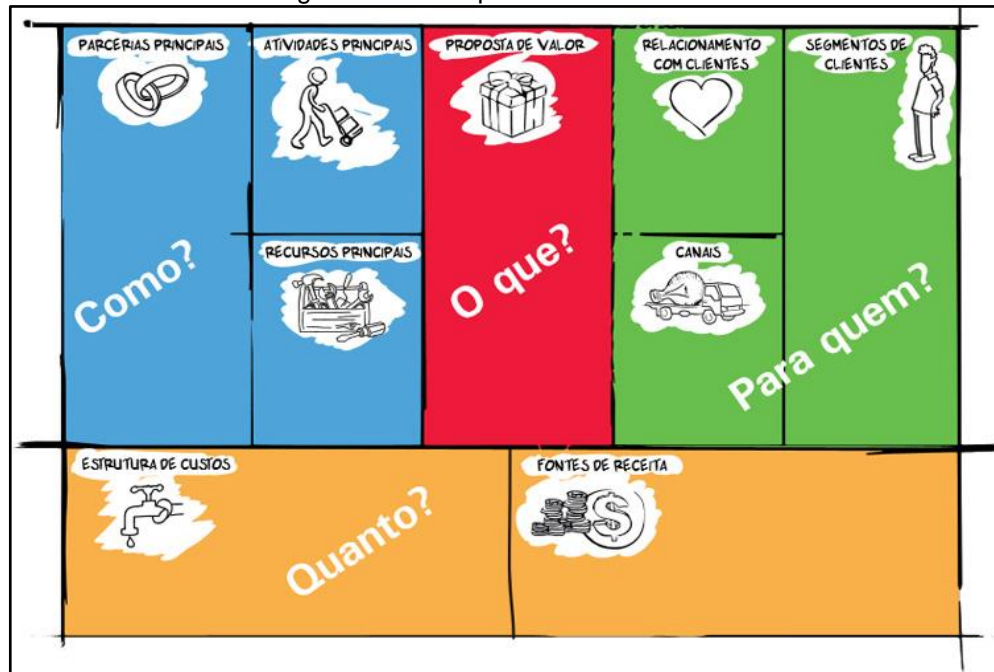
O modelo descreve a lógica de criação do negócio, ou seja, mostra que o raciocínio e a interconexão das partes fazem sentido. O plano de negócios descreve a forma como o negócio será construído, com etapas, prazos, planilhas de custos, receitas etc. Se o modelo de Negócios for alterado, o plano de negócios deverá ser alterado também.

Para criar o modelo de negócios, há um Quadro e ele reúne nove blocos que compõem um negócio, agrupados em quatro questões que precisam ser respondidas:

1. Vou fazer o que? Essa resposta será a sua Proposta de Valor.
2. Para quem vou fazer? Aqui, estão incluídos três blocos: segmento de cliente; canais e relacionamento com clientes.
3. Como vou fazer? Descubra quais são os recursos principais; as atividades e os parceiros principais.
4. Quanto? Avalie quais e como serão obtidas as receitas e qual será a estrutura de custos para viabilizar o negócio.

A Figura 1.1 mostra um exemplo de como desenvolver o modelo canvas e responder as questões propostas.

Figura 1.1: Exemplo de modelo canvas



Fonte: SEBRAE, 2017, p. 19

O propósito é ajudar na organização das ideias, descobrir que cada bloco está relacionado aos demais e permitir que você ajuste o seu modelo quantas vezes for necessário, até conseguir perceber o negócio como um todo.

#### 1.4.1 Análise FOFA

A análise FOFA, sigla para '**Forças, Oportunidades, Fraquezas e Ameaças**', ou *SWOT*, em inglês, é uma técnica que permite analisar e gerenciar os cenários de uma organização. ULRICH (2006, p. 212) contextualiza o SWOT da seguinte maneira:

A análise SWOT é uma das técnicas mais utilizadas em investigação social, quer na elaboração de diagnósticos, quer em análise organizacional ou elaboração de planos. No entanto a sua utilização, nem sempre se traduz em contributos efetivos para os processos, sendo que, muitas vezes, ela surge mais como um ritual ou uma tentativa desprovida de conteúdo real, de sermos mais científicos ou técnicos nos processos de planeamento e/ou avaliação. Na realidade, devido a sua aparente simplicidade, esta técnica emergiu como uma das preferidas por técnicos de todas as áreas.

Oliveira (2015, p. 37) descreve que, a partir do planeamento estratégico, espera-se que uma empresa conheça e melhore os seguintes fatores:

1. Ponto forte é a diferenciação conseguida pela empresa – variável controlável – que lhe proporciona uma vantagem operacional no ambiente empresarial (onde estão os assuntos não controláveis pela empresa).
2. Ponto fraco é a situação inadequada da empresa – variável controlável – que lhe proporciona uma desvantagem operacional no ambiente empresarial.
3. Oportunidade é a força ambiental incontrolável pela empresa, que pode favorecer sua ação estratégica, desde que conhecida e aproveitada, satisfatoriamente, enquanto perdura.
4. Ameaça é a força ambiental incontrolável pela empresa, que cria obstáculos à sua ação estratégica, mas que poderá ou não ser evitada, desde que reconhecida em tempo hábil.

## **1.5 Definição de software e ferramentas**

Segundo Pressman (2011, p. 30), “software de computadores continua a ser a tecnologia única mais importante no cenário mundial”.

Hoje, o software assume um duplo papel. Ele é um produto e, ao mesmo tempo, o veículo para distribuir um produto. Como produto, fornece o potencial computacional representado pelo hardware ou, de forma mais abrangente, por uma rede de computadores que podem ser acessados por hardware local (PRESSMAN, 2011, p. 31).

Mas qual seria o produto mais importante que o software distribui? Para Pressman (2011), a informação seria esse produto mais importante, pois transforma dados pessoais de modo que possam ser mais úteis num determinado contexto; permite gerenciar informações comerciais para aumentar a competitividade; provê um portal para redes mundiais de informação (internet) e os meios para obter informações sob todas as suas formas.

Quanto ao SalvAluno!, o uso de alguns softwares e ferramentas será indispensável, posteriormente, no desenvolvimento do projeto. A partir de agora, serão descritos aqueles que serão utilizados no projeto.

### 1.5.1 Android

A plataforma Android, desenvolvida em Linux, foi lançada pela Google em 2005. Ela permite que o próprio usuário seja capaz de desenvolver software, utilizando diversas funcionalidades, conforme suas reais necessidades ou para resolver problemas, com a ajuda de um ambiente de desenvolvimento completo (LECHETA, 2010).

Além disso, está presente nos dispositivos fabricados pelas maiores empresas de tecnologia existentes na atualidade, por ser uma plataforma completa que contém sistema operacional, *middleware*, aplicações, um kit de desenvolvimento de software (SDK) e interface de programação de aplicativos (API), que auxiliam no desenvolvimento de software, utilizando a linguagem Java (SCHEMBERGER, 2009).

O Android é dividido em quatro camadas: Kernel Linux, bibliotecas, *framework* para aplicações e as próprias aplicações completando-se pela porção *runtime*, necessária para que ocorra a execução dos aplicativos no dispositivo móvel. Na camada de arquitetura, Kernel Linux, é que ocorre o gerenciamento dos processos, threads, arquivos, 18 pastas, memória, drivers.

A base para o desenvolvimento do Android é Java, que contém máquina virtual Dalvik em seu sistema operacional, otimizada para interpretar códigos nos dispositivos móveis, ao compilar o bytecode (.class) para o formato .dex. Após a compilação, o Android gera um arquivo único com a extensão .apk (*Android Package File*), que refere-se à aplicação que deverá ser instalada (SCHEMBERGER, 2009).

Para desenvolver novos projetos no Android, deve-se configurar o ambiente de desenvolvimento e definir a versão mínima de Android, uma vez que cada versão corresponde a um nível de API diferente, com outras funcionalidades e recursos implementados. No caso do SalvAluno, por exemplo, o aplicativo será desenvolvido para trabalhar com Android acima da versão 3.0 (SDK Nível 11 - Honeycomb).

### 1.5.2 Eclipse IDE

O Eclipse é uma IDE, que é um ambiente integrado de desenvolvimento. Inicialmente projeto proprietário da IBM, em 2004 foi doado pela mesma à comunidade, tornando-se um software livre.

Segundo D'Ávila (2005), é a IDE mais utilizada para desenvolver aplicativos na linguagem Java pelas seguintes características:

- a) É um software gratuito, livre e de código aberto;
- b) Oferece ferramentas de ajuda ao desenvolvedor, incluindo geração de código automático, indicação de erros em tempo real e atalhos;
- c) Recebe ajuda de grande comunidade de desenvolvedores, que o tornam melhor e mais estável;
- d) Permite adicionar uma grande variedade de tecnologias;
- e) Possui suporte a C, C++, PHP, XML, UML e HTML, entre outras linguagens.

### 1.5.3 Java

As aplicações Android são desenvolvidas utilizando a linguagem Java. Java é uma linguagem de programação muito popular desenvolvida pela Sun Microsystems (agora propriedade da Oracle).

Desenvolvido muito depois das linguagens C e C ++, Java incorpora muitos dos recursos dessas poderosas linguagens enquanto aborda algumas de suas desvantagens. Mesmo assim, as linguagens de programação são tão poderosas quanto suas bibliotecas. Essas bibliotecas existem para ajudar os desenvolvedores a criar aplicativos. (CONDEY & DARCEY, 2010, tradução nossa)

Ainda segundo os autores, a linguagem possui três principais características:

- 1) É fácil de aprender e entender;

2) Foi projetado para ser independente da plataforma e seguro, usando máquinas virtuais;

3) É orientado a objetos.

Segundo Goodrich e Tamassia (2013), os objetos são considerados os principais atores de um programa Java. São eles que armazenam dados e fornecem métodos para acessar e modificá-los. Todo objeto tem uma classe como instância e uma classe define o tipo e as operações executadas no objeto.

O Android depende muito desses fundamentos Java. O SDK do Android inclui muitas bibliotecas Java padrão (bibliotecas de estrutura de dados, matemática, gráficos, redes, bem como bibliotecas especiais do Android).

#### 1.5.4 SQLite

O SQLite é “um pacote de software de domínio público que fornece um sistema de gerenciamento de banco de dados relacionais, que são usados para armazenar registros definidos pelo usuário em Tabelas grandes” (KRIEBICH, 2010, p. 1, tradução nossa).

O termo “Lite” não se refere às suas capacidades. Em vez disso, o SQLite é leve quando se trata de complexidade de configuração, excesso de cabeçalhos e uso de recursos.

Ainda segundo o autor, o SQLite é definido pelos seguintes recursos:

a) Não possui servidor, pois a biblioteca do SQLite acessa diretamente os arquivos de armazenamento.

b) Por não ter servidor, não precisa ser configurado, o que o torna tão fácil quanto abrir um arquivo.

c) A instância de banco de dados inteira reside em um único arquivo multiplataforma.

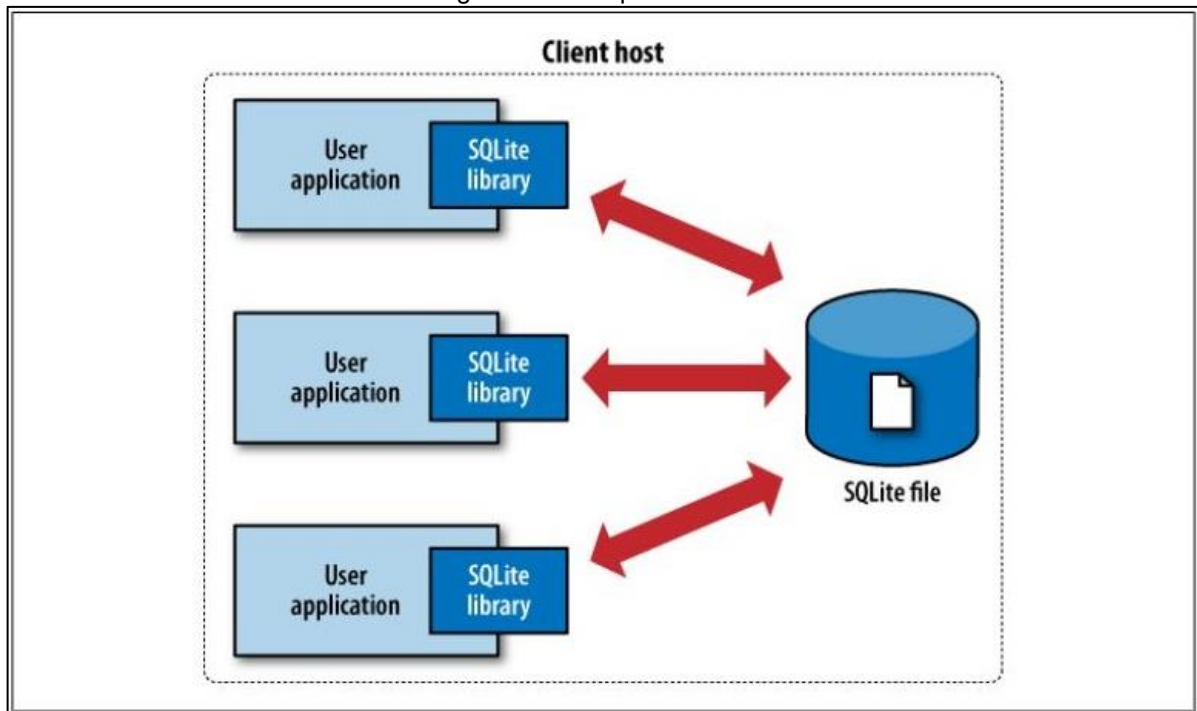
d) Uma única biblioteca contém todo o sistema de banco de dados, que se integra diretamente a um host.

e) A compilação padrão é menor que um megabyte de código e requer apenas alguns megabytes de memória.

f) Suporta a maioria dos recursos de linguagem de consulta encontrados no padrão SQL92 (SQL2)

Além disso, é altamente confiável, fornece um ambiente de banco de dados relacional muito funcional e flexível, consome recursos mínimos e cria mínimo aborrecimento para desenvolvedores e usuários. A Figura 1.2 ilustra como funciona a arquitetura SQL.

Figura 1.2 – Arquitetura SQL



Fonte: KRIEBICH, 2010, p. 3

O SQLite é projetado para ser integrado diretamente em um executável. Isto elimina a necessidade de uma biblioteca externa e simplifica a distribuição e a instalação.

### 1.5.5 Maven

Apache Maven é uma ferramenta de gerenciamento e compreensão de projetos de software, utilizada primariamente em projetos Java e desenvolvida e hospedada pela Apache Software Foundation.

Com base no conceito de um modelo de objeto de projeto (POM), o Maven pode gerenciar a compilação, o relatório e a documentação de um projeto a partir de uma informação central (MAVEN, 2017, tradução nossa).

O Maven baixa bibliotecas Java e seus plug-ins dinamicamente de um repositório online que o programador seleciona, as bibliotecas são baixadas e armazenadas em uma área de cache local. Com a utilização do Maven, o usuário fornece apenas a configuração para a utilização de APIs que estão online e o Maven captura o código do desenvolvedor e aplica na aplicação atual.

Para utilizar o Maven em algum projeto basta o programador definir as coordenadas para as bibliotecas que estão na nuvem, e usufruir de suas aplicações.

### 1.5.6 API

*Application Programming Interface (API)* ou Interfaces de Programação dos Aplicativos trata-se do conjunto das classes disponíveis a serem utilizadas durante o processo de desenvolvimento de software (LECHETA, 2010).

Ainda segundo o autor, o núcleo Android APIs está disponível em todos os telefones Android e para que seja possível a sua utilização, os dispositivos móveis devem possuir certos recursos. Como exemplos, há os serviços de localização através do GPS (Global Position System), as APIs de mídia, a API para gráficos 3D ou ainda as APIs para que seja possível acessar baixo nível de hardware.

Embora se tenha o conhecimento da variedade de funções, a API não deve ser confundida com o conceito de biblioteca, pois determina apenas os recursos disponíveis que podem ser chamados, como se comportam e funcionam.

Através da utilização da API, é possível que o aplicativo trabalhe de forma automática, em qualquer local que esteja, a partir da conexão com a internet e aos serviços da Google e que possa ser utilizado para implementar novos paradigmas de programação que sirvam para resolver problemas antigos ou novos nas mais diversas áreas.

#### 1.5.7 API Dropbox V2

Dropbox é um serviço de armazenamento em nuvem. Criado pela empresa homônima, Dropbox Inc., e oficialmente lançado em 2008, oferece diversos recursos online, inclusive sua utilização em software terceiros através de sua API.

A API do Dropbox é uma extensão da plataforma criada pela Dropbox Inc. que permite que os desenvolvedores trabalhem com arquivos no Dropbox, incluindo funcionalidades avançadas como pesquisa de texto completo, miniaturas e compartilhamento. O explorador da API do Dropbox é a maneira mais fácil de começar a fazer chamadas de API (DEVELOPERS DROPBOX, 2017).

Para sua utilização em linguagem Java basta o usuário ir em sua conta online e gerar um 'token', que é uma chave (senha) eletrônica que será usada no software, baixar o arquivo .jar que a própria Dropbox libera para os usuários, este arquivo contém a interface lógica que é necessária para acessar a conta em nuvem, permitindo o usuário criar seu próprio código.

Segundo o blog Dropbox Developers, em 28 de junho de 2016, a Dropbox informou que sua API estava ultrapassada e que permaneceria ativada por um ano a partir da data do anúncio, para os seus usuários se adaptarem. No entanto, este anúncio não veio para informar a sua desativação, mas sim, a sua migração para a sua segunda versão, a Dropbox API v2 (<https://blogs.dropbox.com/developers/2016/06/api-v1-deprecated/>. - DROPBOX DEVELOPERS BLOG, 2016, tradução nossa).

A API Dropbox v2 é um conjunto de pontos de extremidade HTTP que ajudam um aplicativo a se integrar ao Dropbox. Essa, por sua vez, possui toda as funções da primeira versão e uma melhor estabilidade de sistema, além de trabalhar por Maven, que permite que o usuário apenas utilize os códigos da Dropbox em seu software sem ter de baixar cada atualização que a empresa fizer.

#### 1.5.8 QR Code

QR Code, sigla para “*Quick Response*”, que em português significa ‘resposta rápida’, é um código de barras bidimensional de leitura rápida.

Ele foi criado em 1994 pela empresa japonesa Denso-Wave para identificação de peças automobilísticas. Masahiro Hara foi o responsável pelo seu desenvolvimento (LIAO & LEE, 2010).

Um dos fatores decisivos para a difusão do QR Code foi torná-lo livre para uso de todas as pessoas sem qualquer custo e sem quaisquer problemas, uma vez que a Denso Wave não planejava exercer os seus direitos de patente (SOON, 2008).

Este código tem capacidade de armazenar uma grande quantidade de informação e a sua leitura consegue ser 10 vezes mais rápida do que os códigos tradicionais (SOON, 2008).

A Figura 1.3 mostra uma imagem com a representação de um modelo de QR Code.

Figura 1.3 – Modelo do QR Code



Fonte: [www.qrcode.com](http://www.qrcode.com), 2017

Segundo o QRCode.com (2017, tradução nossa), o código possui várias funções. Uma delas é a alta capacidade de codificar todos os tipos de dados, tais como caracteres numéricos e alfabéticos, Kanji, Kana, Hiragana, símbolos, binário e códigos de controle. A Figura 1.4 mostra um exemplo com vários caracteres, que podem ser codificados em um único código.

Figura 1.4 – Codificação do QR Code



Fonte: [www.qrcode.com](http://www.qrcode.com), 2017

Enquanto códigos de barras convencionais são capazes de armazenar aproximadamente 20 dígitos, no máximo, um QR pode até 7.089 caracteres em um símbolo.

Por transportar informações tanto horizontal quanto verticalmente, o QR Code pode codificar a mesma quantidade de dados em aproximadamente um décimo do espaço de um código de barras tradicional, e isso pode ser visualizado na Figura 1.5.

Figura 1.5 – Tamanho mínimo de impressão do QR Code



Fonte: [www.qrcode.com](http://www.qrcode.com), 2017

Para um tamanho de impressão menor, é possível representar um código de barras em um código Micro QR.

Como uma simbologia desenvolvida no Japão, o QR Code também é capaz de codificar o conjunto de caracteres kanji JIS Nível 1 e Nível 2, representados na Figura 1.6.

Figura 1.6 – Capabilidade Kanji e Kana



Fonte: [www.qrcode.com](http://www.qrcode.com), 2017

No caso do japonês, um caractere Kana ou Kanji de largura total é eficientemente codificado em 13 bits, permitindo que o QR Code mantenha mais de 20% de dados do que outras simbologias 2D.

O QR Code tem capacidade de correção de erros. Os dados podem ser restaurados mesmo que o símbolo esteja parcialmente sujo ou danificado. A Figura 1.7 mostra dois exemplos, um código com sujeira e o outro com dano.

Figura 1.7 – Sujeira e dano em um código QR

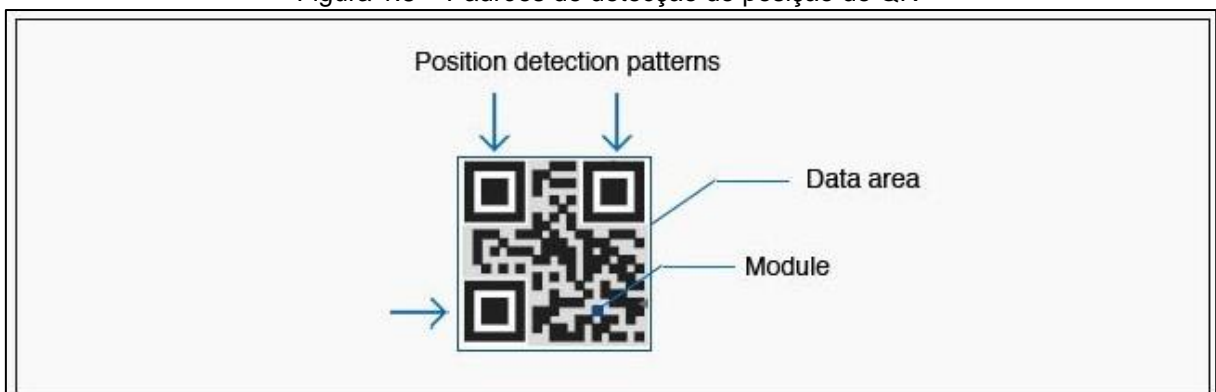


Fonte: [www.qrcode.com](http://www.qrcode.com), 2017

Dependendo da quantidade de sujeira ou dano, pode ser restaurado um máximo de 30% das palavras de código (unidade que constrói a área de dados, equivalente a 8 bits).

O código QRCode pode ler em alta velocidade a 360 graus (omnidirecional), através de padrões de detecção de posição localizados nos três cantos do símbolo, ilustrados na Figura 1.8.

Figura 1.8 – Padrões de detecção de posição do QR



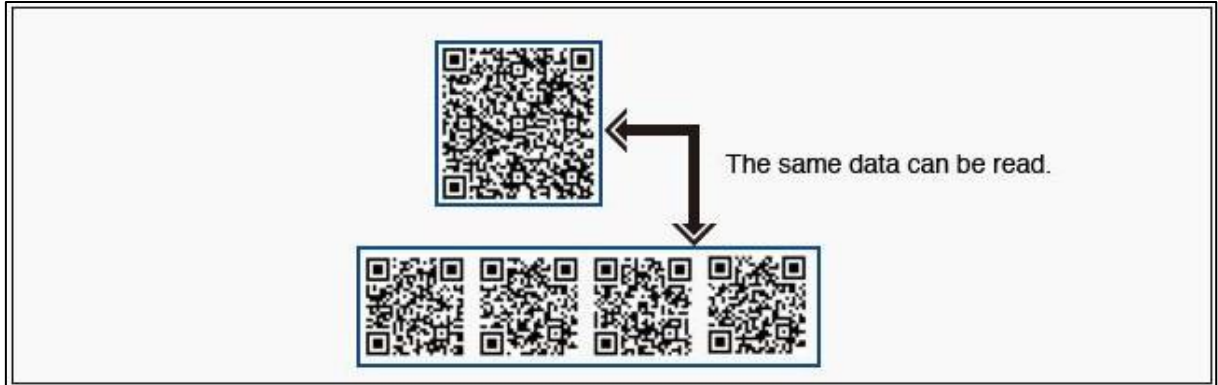
Fonte: [www.qrcode.com](http://www.qrcode.com), 2017

Estes padrões de detecção de posição garantem uma leitura estável de alta velocidade, contornando os efeitos negativos da interferência de fundo.

O QR Code pode ser dividido em várias áreas de dados. Inversamente, as informações armazenadas em vários símbolos de código QR podem ser reconstruídas como um único símbolo de dados. Um símbolo de dados pode ser dividido em até 16

símbolos, permitindo a impressão em uma área estreita. A Figura 1.9 ilustra bem essa função.

Figura 1.9 – Recurso de anexação estruturada do QR



Fonte: [www.qrcode.com](http://www.qrcode.com), 2017

Hoje possui diversas utilizações, geralmente para gerenciamentos de estoque em indústrias, em revistas e propagandas para registrar endereços e URLs, e desde 2003 vem sendo usado, em aplicações mobile para ajudar usuários a obter dados e informações, como links, números, contatos de email, entre outros.

## 2 METODOLOGIA

Segundo Cervo e Bervian (2007), método é a ordem que se deve impor aos diferentes processos necessários para atingir um certo fim ou um resultado desejado. A técnica, por sua vez, é a aplicação do plano metodológico e a forma especial para a sua execução. Comparando, pode-se dizer que a relação existente entre método e técnica é a mesma que existe entre estratégia e tática.

De acordo com Tasca et al. (2010), a análise do contexto, a definição de um problema e das questões direcionadoras dão início ao processo de pesquisa científica, motivando assim os pesquisadores a procurarem informações sobre determinado tema em bases bibliográficas.

Para Severino (2013), a bibliografia tem o objetivo de descrever e classificar os livros e documentos semelhantes, a partir de alguns critérios como autor, gênero literário, tema, data etc.

A metodologia relacionada ao SalvAluno! será de natureza aplicada, com abordagem quantitativa da análise e tratamento de dados. Além disso, terá os estudantes como público alvo.

A pesquisa será de cunho bibliográfico, a partir de artigos, livros e documentos eletrônicos, que sustente as teorias e os conceitos abordados e permita embasar e aprofundar o tema. Além disso, haverá uma pesquisa de campo que será demonstrada a partir de um questionário a ser desenvolvido na segunda parte do trabalho.

Para Gil (2010), questionário é a técnica de investigação composta por um número mais ou menos elevado de questões apresentadas por escrito às pessoas, tendo por objetivo o conhecimento de opiniões, crenças, sentimentos, interesses, expectativas, situações vivenciadas, etc.

Quanto ao aplicativo, ele será desenvolvido para trabalhar com Android acima da versão 3.0, utilizando banco de dados SQLite 3 e sincronização com armazenamento em nuvem com a plataforma Dropbox API v2 via Maven, além de utilizar a API ZXing Core V3.2.1 via Maven, responsável por gerar o QR Code a partir do RA do aluno. Ademais, vai usar as seguintes permissões:

- a) Acesso à internet, utilizado para download dos parâmetros das instituições;
- b) Estado da internet, utilizado para verificar a conexão com a internet;
- c) Ler/modificar conteúdo interno do cartão de memória, utilizado para realizar o backup do banco de dados, para evitar perdas dos dados;
- d) Adicionar/modificar eventos do calendário (datas de atividades ou provas).

Quanto ao sistema, serão utilizados os seguintes métodos:

- a) **Mobile:** Plataforma Eclipse - Android Developer Tools com as ferramentas: Java SE Development Kit 8 Update 91 (jdk-8u111) e Apache Maven para eclipse (M2Eclipse v1.4.1).
- b) **Computador:** Plataforma C++ Builder 2010.

## 2.1 Cronograma de atividades

O cronograma foi estruturado da seguinte maneira:

- 1) **Elaboração do pré-projeto:** O pré-projeto começou a ser elaborado a partir de novembro de 2016. Primeiramente foram definidos os seguintes itens: o professor orientador do projeto e o título do projeto.
- 2) **Escolha do tema:** O tema foi definido posteriormente, além dos objetivos, da justificativa, descrição e os materiais e métodos a serem utilizados do mesmo e, na sequência, a entrega do pré-projeto.

3) **Levantamento bibliográfico:** Para fundamentar e aprofundar o tema, além de desenvolver a metodologia, foi realizada uma pesquisa e análise bibliográficas, a partir de artigos, livros e documentos eletrônicos.

4) **Entrega 1ª parte do projeto e apresentação:** Após a defesa e possível aprovação da primeira parte do trabalho, inicia-se a segunda parte do projeto, que consiste no seu desenvolvimento.

5) **Coleta de dados:** Aqui, haverá a coleta dos dados, que serão obtidos a partir do questionário.

6) **Tratamento e análise dos dados:** A partir da coleta dos dados, serão executados o tratamento e análise deles, para depois demonstrar os resultados obtidos.

7) **Teste e execução do sistema:** Sabendo-se que um sistema é interminável e que sempre está em constante modificação, baseados nas inferências do profº Carlos Cabello, de qualidade de software, os testes executados no sistema permitirão o aprimoramento constante do design e, mais do que isso, adequá-lo às necessidades da instituição, principalmente no que se refere ao controle de faltas e notas, além da divulgação das informações relacionadas a ela.

8) **Revisão da literatura:** Aqui serão apresentadas teorias e outras pesquisas relevantes para o estudo, que consistem da revisão crítica dos textos e explicar como o trabalho dialoga com os estudos revisados.

9) **Entrega da 2ª parte do trabalho;**

10) **Apresentação 2ª parte do projeto.**

A Figura 2.1 mostra os itens definidos no cronograma e as datas (em meses) em que cada item foi ou será desenvolvido.

Figura 2.1 – Cronograma do projeto

Cronograma do Projeto	2016		2017											
	nov	dez	jan	fev	mar	abr	mai	jun	jul	ago	set	out	nov	dez
Elaboração do pré projeto	■													
Escolha do tema	■	■												
Levantamento bibliográfico			■	■	■	■	■							
Entrega 1ª parte do projeto						■								
Apresentação 1ª parte projeto							■	■						
Coleta de dados									■	■				
Tratamento/análise dos dados										■				
Teste/execução do sistema											■	■	■	
Revisão da literatura												■	■	
Entrega 2ª parte projeto													■	
Apresentação 2ª parte projeto													■	■

Fonte: Autoria própria, 2017

### **3 DESENVOLVIMENTO**

Depois de apresentar a fundamentação teórica e conceitual dos dados e da metodologia a ser adotada no projeto, aparece, em seguida, o desenvolvimento. Este capítulo deve apresentar um vínculo entre a teoria fundamentada no primeiro capítulo e os métodos de construção do mesmo.

Após coletar os dados, obtidos a partir do questionário, realizar o tratamento e análise deles e demonstrar os resultados obtidos, será mostrado o modelo de negócios do projeto; em seguida, será discorrido o termo de abertura do projeto.

Na sequência, haverá a abordagem de requisitos de software, casos de uso bem como os diagramas, as modelagens do banco de dados do projeto e a descrição do método de desenvolvimento.

Ademais, também será abordado o UML, linguagem que elabora a estrutura de projetos de software e empregada para visualizar, especificar a documentação de elementos que o compõe.

Sabendo-se que um sistema é infundável e que está sempre em constante aprimoração, serão executados, até a finalização do projeto testes no sistema, a fim de melhorar continuamente o design e adequá-lo às necessidades da instituição.

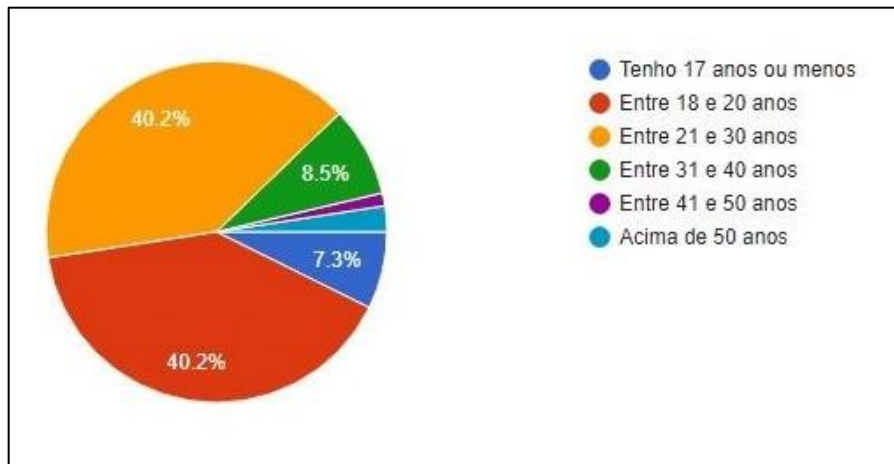
Por fim, ao encerrar esse capítulo, com a revisão da literatura feita, que vai abordar outras teorias e pesquisas importantes para o trabalho, além da revisão crítica deles, serão discorridas as considerações finais.

#### **3.1 Análise e tratamento de dados**

Após realizar a pesquisa de campo, a partir do questionário, houve a coleta dos dados e, com isso, a análise e tratamento deles.

A pesquisa ficou disponível em um formulário, através da ferramenta Google Forms, e permaneceu aberta por 23 dias. No total, foram recolhidas 82 respostas. Primeiramente, foi perguntado a idade dos entrevistados, que pode ser visualizado no Gráfico 3.1

Gráfico 3.1 – Idade dos entrevistados

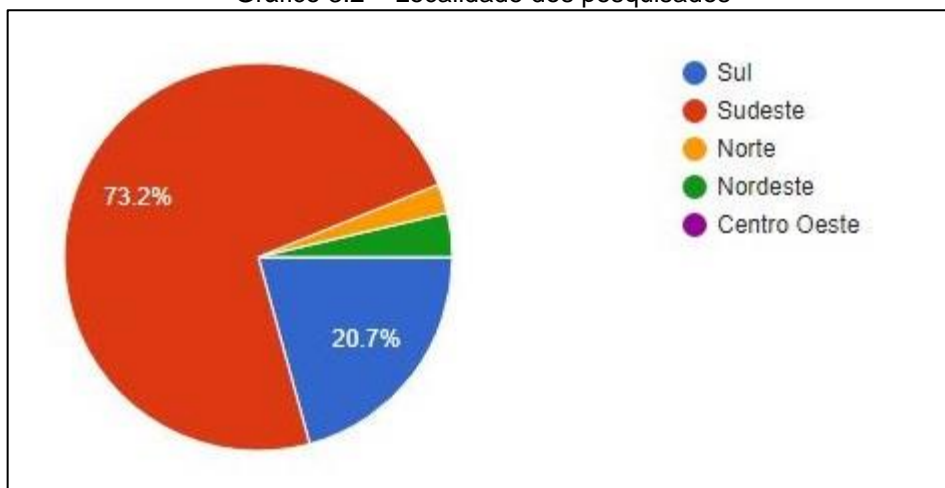


Fonte: Autoria própria, 2017

A partir dessas 82 respostas, concluiu-se que a maioria das pessoas pesquisadas têm acima de 18 anos de idade e apenas 6 pessoas se declararam como menor de idade.

Após isso, foi perguntado a localidade, por região do Brasil, dos entrevistados, no Gráfico 3.2.

Gráfico 3.2 – Localidade dos pesquisados

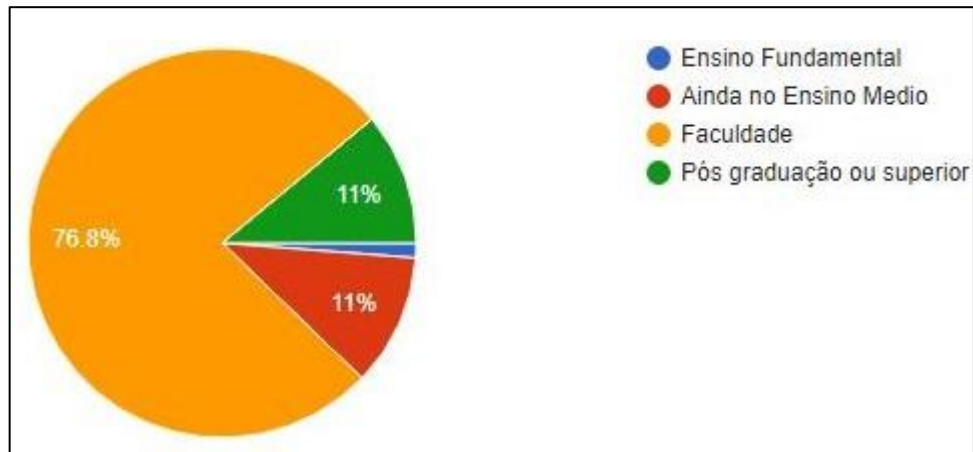


Fonte: Autoria própria, 2017

Dos entrevistados, 73,2% residem na Região Sudeste do Brasil, seguida da Região Sul, com 20,7% do total.

O Gráfico 3.3 mostra a porcentagem da escolaridade dos entrevistados.

Gráfico 3.3 – Escolaridade dos entrevistados

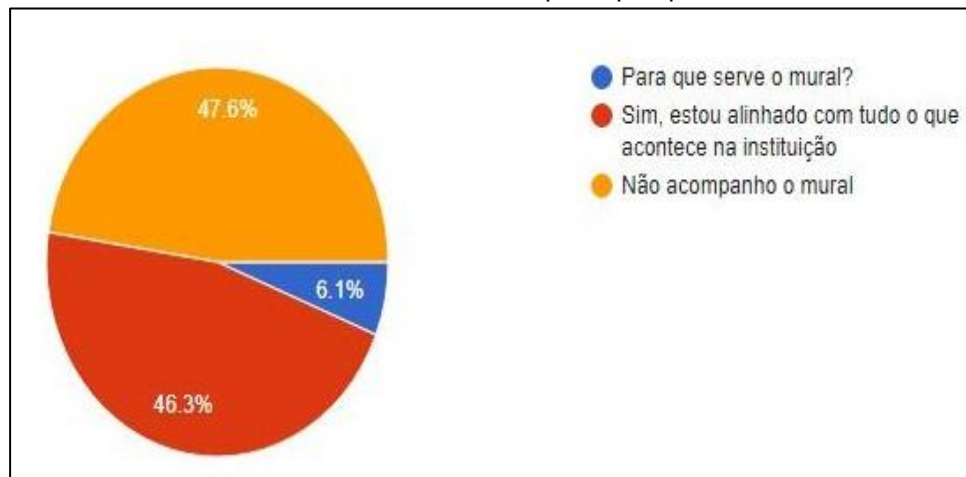


Fonte: Autoria própria, 2017

Quanto à escolaridade, 76,8% dos entrevistados estão cursando o Ensino Superior. Tanto a pós-graduação quanto o Ensino Médio se igualam com 11% do total.

O Gráfico 3.4 apresenta a porcentagem relacionada ao uso do mural por parte dos pesquisados.

Gráfico 3.4 – Uso do mural pelos pesquisados

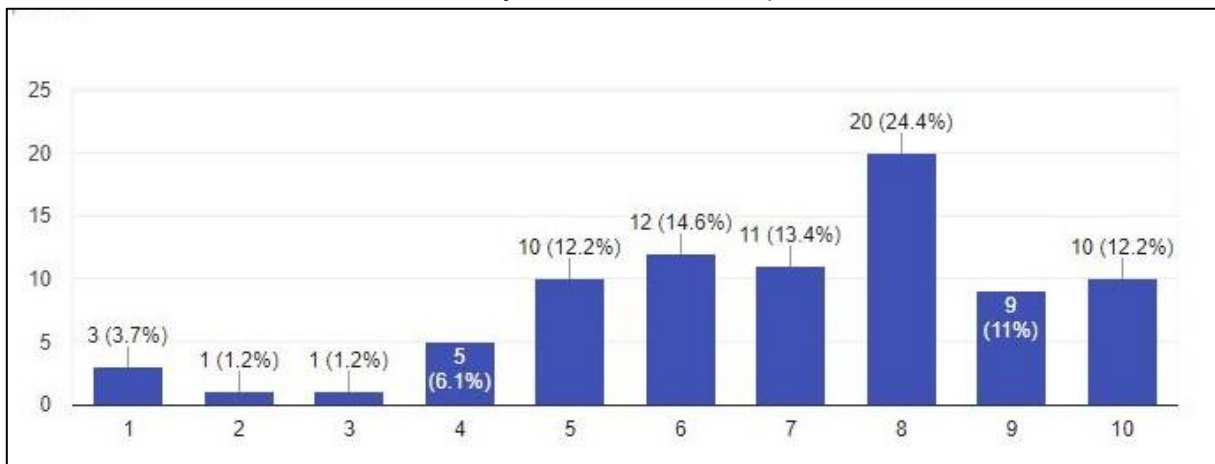


Fonte: Autoria própria, 2017

Apenas 46.3% dos pesquisados acompanham o mural de sua instituição de ensino. O SalvAluno! auxilia a visualização dos dados, de forma intuitiva, para quem não acompanha e de forma mais fácil para quem acompanha o mural.

No Gráfico 3.5 aparece a quantidade e a porcentagem da recordação dos entrevistados para realizar os trabalhos escolares.

Gráfico 3.5 – Recordação dos entrevistados para trabalhos escolares

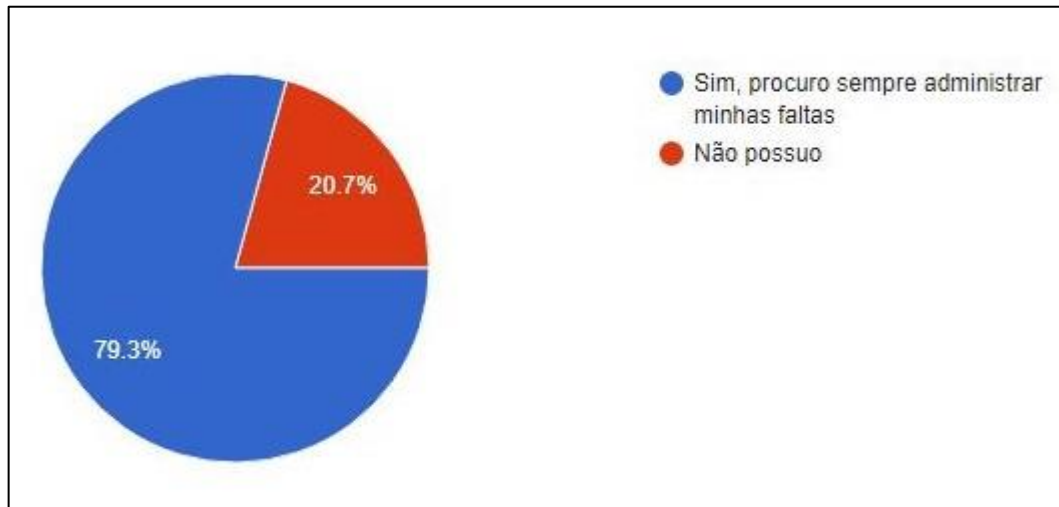


Fonte: Autoria própria, 2017

Quanto à recordação em fazer trabalhos, 20 pessoas deram nota igual ou abaixo de 5. Isso acontece devido aos problemas do dia a dia, e o sistema lembra o usuário para evitar esquecimento de trabalhos.

O Gráfico 3.6 exibe a porcentagem do controle de falta dos entrevistados.

Gráfico 3.6 – Controle de faltas dos entrevistados

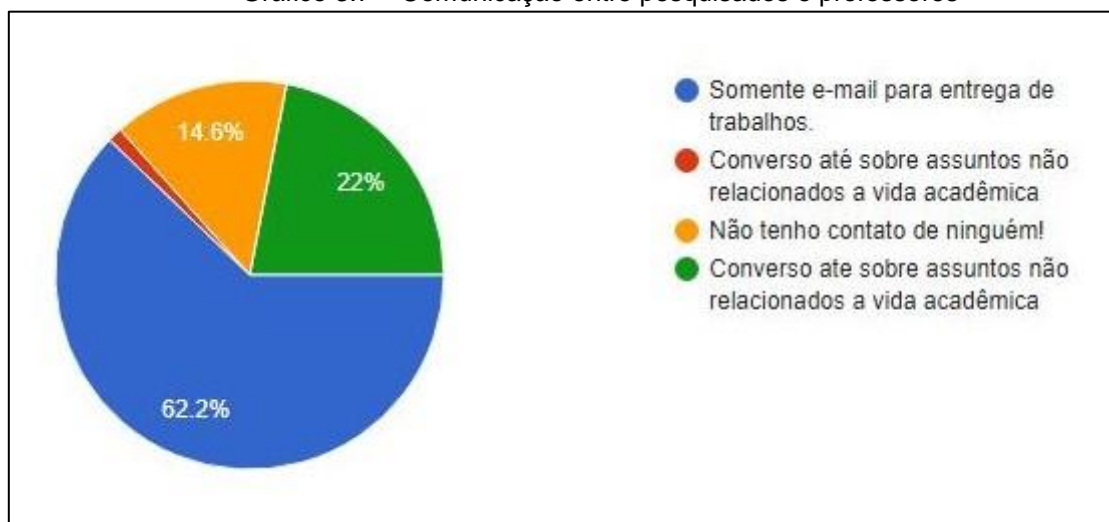


Fonte: Autoria própria, 2017

Dos entrevistados, 20,7% não possuem um controle de faltas e isso pode ocasionar em diversas reprovações por faltas. Esse número diminui consideravelmente com um controle eficiente de faltas.

O Gráfico 3.7 exhibe a porcentagem da comunicação entre pesquisados e professores.

Gráfico 3.7 – Comunicação entre pesquisados e professores

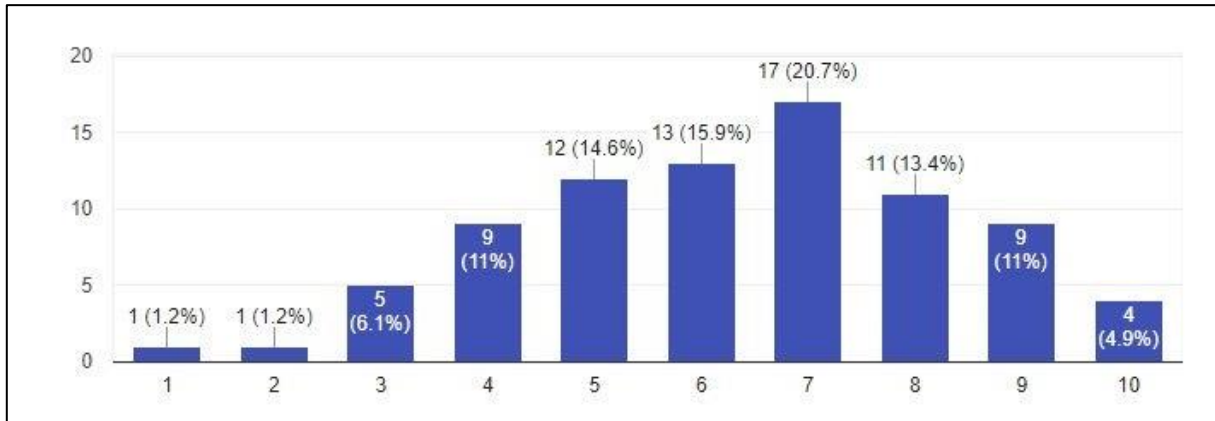


Fonte: Autoria própria, 2017

Dos entrevistados, 12 pessoas não possuem contato fácil com seus professores, por isso que é comum alguém perguntar em grupos o e-mail de algum professor.

No Gráfico 3.8, aparece a quantidade e a porcentagem da opinião do eleitorado sobre a comunicação da instituição.

Gráfico 3.8 – Opinião do eleitorado sobre a comunicação da instituição

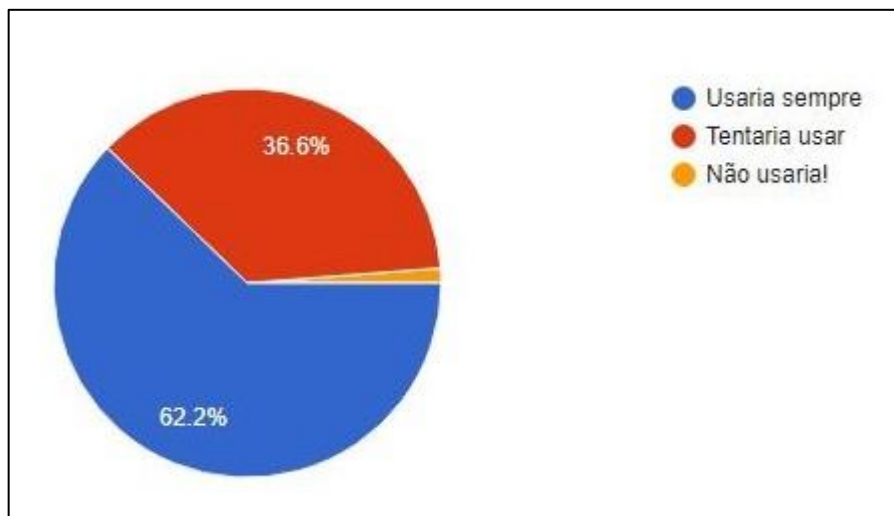


Fonte: Autoria própria, 2017

Em relação à comunicação, 28 pessoas classificaram uma nota menor ou igual a 5 a nota de comunicação da instituição. Caso aconteça algum imprevisto com um professor, dificilmente todos os alunos estarão cientes, causando viagens perdidas.

Por fim, o Gráfico 3.9 mostra, em porcentagem, a usabilidade do sistema SalvAluno! quanto aos pesquisados.

Gráfico 3.9 – Usabilidade do SalvAluno!



Fonte: Autoria própria, 2017

Em relação ao SalvAluno!, apenas 1 pessoa não usaria o sistema por não acreditar que seja uma aplicação funcional. Por outro lado, 52 pessoas usariam o aplicativo e 30 poderiam dar uma chance ao sistema.

No final, a pesquisa de campo trouxe resultados bem positivos em relação à utilização do aplicativo, prometendo ser um sistema que os alunos considerariam indispensável.

### 3.2 Modelo de Negócios do SalvAluno!

Após fundamentar e descrever a respeito do modelo de negócios e suas características no primeiro capítulo, logo abaixo, na Figura 3.1, está representado o modelo de negócios do SalvAluno!

Figura 3.1 – Modelo de negócios do SalvAluno!

<b>Modelo de Negócio SalvAluno!</b>				
Parceiros Chave	Atividades Chave	Proposta de Valor	Relacionamento	Segmentos de Clientes
Instituições de ensino	Controle de notas e faltas	Para a faculdade: Inovação, otimização da comunicação	Empresa Digital	Estudantes que desejam melhorar e organizar sua vida estudantil
Estudantes independentes	Digitalizar Mural	Para o estudante: Comodidade, praticidade em organizar suas tarefas diárias.		Instituições de ensino que desejam otimizar a comunicação com o aluno
Recurso Chave	Canais	Custos	Receitas	
Intelectuais	Parceiros diretos - Web; comunicação online	Desenvolvimento, divulgação, hospedagem.	Pagamento pelo uso	

Fonte: Autoria própria, 2017

### 3.3 Termo de abertura do projeto

Antes de iniciar um projeto, deve-se criar um termo de abertura, que é o documento que especifica o objetivo, o escopo do projeto, bem como outras

características dele, como a justificativa, as premissas e restrições iniciais, o prazo preliminar e a estimativa de custos.

Na sequência, há o termo de abertura do projeto SalvAluno! e suas características principais.

### **Desenvolvimento do projeto**

Um software que auxilie no gerenciamento de notas e faltas.

### **Objetivo do projeto**

Desenvolver um sistema, por meio de um aplicativo que auxilie os estudantes a organizar as tarefas do dia-a-dia da faculdade, controlar suas notas e faltas e estar a par dos eventos que ocorrem na instituição que eles frequentam.

### **Descrição do problema e justificativa**

O problema da correria do dia a dia afeta na organização da vida dos estudantes, cujo impacto é o esquecimento das tarefas ou da fórmula para calcular notas. Uma boa solução seria desenvolver um software para ajudá-los a gerenciar as notas, faltas e ficar por dentro das informações ligadas à instituição de ensino em que frequentam.

### **Premissas iniciais**

- a) No momento da execução do projeto, aqueles que desenvolverem o projeto não devem ser deslocados para outro projeto;
- b) O cliente deve dispor a infraestrutura necessária para desenvolver o projeto.

### **Restrições iniciais**

- a) O custo do projeto não pode ultrapassar o valor previamente combinado;
- b) Se houver descumprimento de datas do cronograma do projeto, os responsáveis serão multados;
- c) Inicialmente, o projeto será desenvolvido em uma equipe de 3 integrantes, podendo aumentar com relação à complexidade das tarefas ligadas ao projeto.

### **Riscos iniciais**

a) Instabilidade do mercado, devido ao cenário econômico atual, o que pode acarretar na onerosidade e interrupção do desenvolvimento do projeto;

b) Barreiras na comunicação, que podem ser acentuadas por informações implícitas ou mal-entendidos causados pela falta de conhecimento do cliente na área da informática;

c) Escolha da equipe, que é completamente importante na execução do projeto. Uma vez despreparada, sem treinamento, habilidades e competência suficientes, a equipe será determinante e poderá comprometer o resultado final do projeto.

### **Prazo preliminar do projeto**

O prazo previsto para concluir o projeto será de um ano, a partir do dia 17/11/2016 até o dia 17/11/2017.

### **Estimativa inicial de custos**

A estimativa inicial de custos prevista será de R\$ 87.480,00 para o desenvolvimento do projeto.

## **3.4 Requisitos do projeto**

Requisitos de software nada mais são do que um conjunto de atividades que o software deve desempenhar, com suas limitações e restrições, além de características não ligadas diretamente às funções desempenhadas pelo software (SOMMERVILLE, 2011).

Os requisitos de sistema, de modo clássico, são classificados como funcionais e não funcionais. A seguir, serão abordados cada uma destas categorias.

### **3.4.1 Requisitos funcionais**

Os requisitos funcionais tratam de funções que o sistema deve fornecer, como o sistema deve se comportar a entradas e a determinadas situações (PRESSMAN, 2011).

Em outras palavras, descrevem a funcionalidade ou serviço que se espera que o sistema forneça. Dependendo do tipo de software do requisito a ser descrito é possível criar subgrupos de requisitos funcionais, normalmente subdivididos como (SOMERVILLE, 2011):

- a) Requisitos funcionais de usuário: normalmente descritos de um modo geral;
- b) Requisitos funcionais de sistema: descrevem a função de sistema detalhadamente, suas entradas, saídas e exceções.

A especificação de requisitos funcionais deve ser completa e consistente, ou seja, todas as funções requeridas pelo usuário devem ser definidas e evidentes. Além disso, não devem ter definições contraditórias.

Quanto ao SalvAluno!, foram definidos os seguintes requisitos funcionais:

- RF1. Manter Instituição
- RF2. Manter Aluno
- RF3. Manter Matéria
- RF4. Manter Instrutor
- RF5. Manter Aula
- RF6. Manter Mural
- RF7. Ver Instituição
- RF8. Manter Dados Instrutor
- RF9. Ver Aluno
- RF10. Manter Dados Matéria
- RF11. Manter Nota
- RF12. Manter Atividades
- RF13. Manter Dados Aula
- RF14. Manter Faltas
- RF15. Ver Mural
- RF16. Manter Anotação

### 3.4.2 Requisitos não funcionais

Os requisitos não funcionais estão relacionados às restrições sobre os serviços ou funções do sistema. Por exemplo, restrição de tempo, restrição do processo de desenvolvimento, padrões, etc. (PRESSMAN, 2011).

Esses requisitos que não dizem respeito às funções específicas fornecidas pelo sistema. Logo, estão diretamente relacionados a propriedades do sistema, como confiabilidade, tempo de resposta e espaço em disco. Podem definir restrições para o sistema, como a capacidade de dispositivos de entrada e saída relacionados e as representações de dados utilizados em um padrão de interface.

Contudo, requisitos não funcionais também dizem respeito ao sistema como um todo e não a características individuais do sistema. Em razão disso, podemos considerá-los mais importantes que requisitos funcionais individuais. Além disso, se houver falhas ao cumprir um requisito não funcional poderá tornar todo o sistema inútil.

Quanto ao SalvAluno, foram definidos os seguintes requisitos não funcionais:

RNF1. Manutenibilidade: capacidade do sistema de ser modificado.

RNF2. Funcionalidade: segurança de acesso e conformidade (o sistema deve estar de acordo com normas ou regulamentações previstas em lei).

RNF3. Eficiência (tempo): desempenho (o sistema deve fornecer tempos de respostas e de processamento quando executa suas funções). Relatório impresso em 5 segundos.

RNF4. Usabilidade: o sistema deve ser compreendido, aprendido, operado e atraente ao usuário, sendo de fácil manuseio.

### 3.4.3 Requisitos de domínio

Os requisitos de domínio originam-se do domínio da aplicação do sistema, refletindo as características deste domínio (SOMMERVILLE, 2011). Podem ser funcionais ou não (PRESSMAN, 2011).

Se estes requisitos não forem atendidos satisfatoriamente, poderá ser impossível fazer a aplicação operar adequadamente.

## 3.5 UML

A UML ou 'Linguagem de Modelagem Unificada' é um tipo de linguagem que especifica, documenta, visualiza e desenvolve sistemas orientados a objetos. Ela compõe um conjunto básico de diagramas e notações que permitem representar as múltiplas perspectivas (estruturais, estáticas e comportamentais e dinâmicas) de um sistema sobre análise e desenvolvimento.

Por meio de seus diagramas é possível representar sistemas de softwares sob diversas perspectivas de visualização. Facilita a comunicação de todas as pessoas envolvidas no processo de desenvolvimento de um sistema - gerentes, coordenadores, analistas, desenvolvedores - por apresentar um vocabulário de fácil entendimento (OMG, 2006).

### 3.5.1 Casos de uso e diagrama de classes

Segundo Cockburn (2005, p. 21), "o caso de uso descreve o comportamento do sistema sob diversas condições conforme o sistema responde a uma requisição de um dos stakeholders, denominado ator primário", ou seja, aquele que inicia uma interação com o sistema para atingir um objetivo.

Um caso de uso pode ser visualizado a partir de uma forma textual ou escrito usando fluxogramas, diagramas ou linguagens de programação (COCKBURN, 2005).

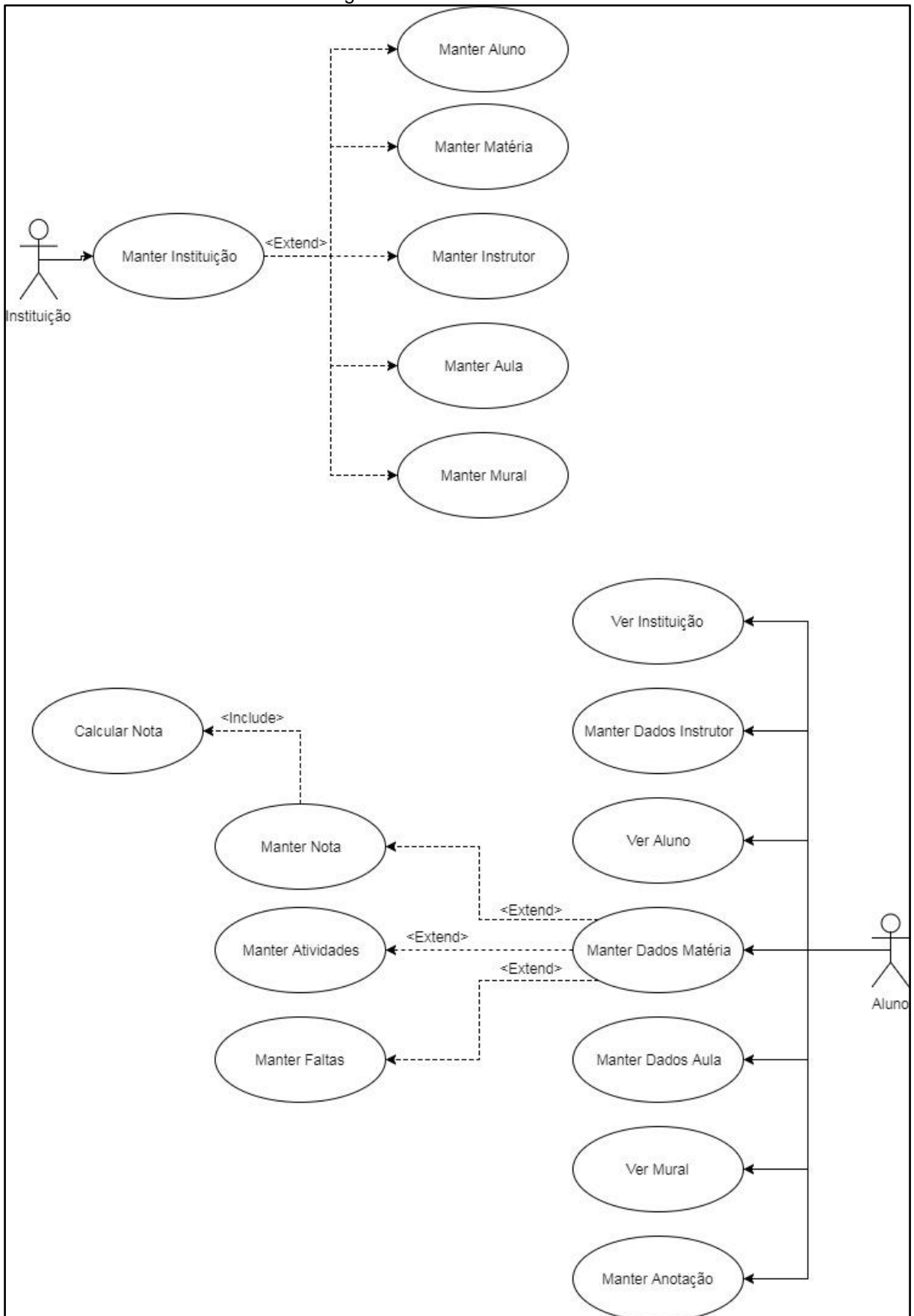
Basicamente, um diagrama de caso de uso descreve as interações entre os usuários e o sistema propriamente dito. Um ator geralmente é uma entidade externa que participa de um caso de uso ao interagir com o sistema.

Nos diagramas de caso de uso há também os relacionamentos entre os casos de uso, que são eles:

- a) <<include>>: Um caso de uso toma lugar dentro de outro caso de uso;
- b) <<extend>>: Um caso de uso é estendido para outro caso de uso;
- c) Generalização: Um caso de uso “herda” características de outro caso de uso.

A seguir, na Figura 3.2 aparece o diagrama de caso de uso do SalvAluno!, com os casos de uso e seus relacionamentos.

Figura 3.2 – UML SalvAluno!



Fonte: Autoria própria, 2017

Segundo Pressman (2011, p. 164), a modelagem baseada em classes representa “os objetos que o sistema irá manipular, as operações (também denominadas métodos ou serviços) que serão aplicadas aos objetos para efetuar a manipulação, os relacionamentos entre objetos e as colaborações que ocorrem entre as classes definidas”.

A seguir, são especificados os casos de uso do SalvAluno! e suas respectivas classes:

### **Manter Instituição**

Sumário: Instituição mantém instituição.

Ator primário: Faculdade.

Fluxo Principal

1. O responsável da instituição obtém os dados da instituição (nome, imagem representativa, endereço, site, login para usuário e senha).

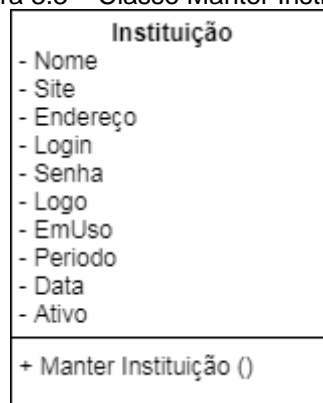
2. O responsável da instituição cadastra os dados da instituição no banco de dados online.

Fluxo Alternativo

1. O responsável da instituição poderá atualizar os dados da instituição no banco de dados online.

A Figura 3.3 apresenta a classe relacionada ao caso de uso Manter Instituição.

Figura 3.3 – Classe Manter Instituição



Fonte: Autoria própria, 2017

## Manter Aluno

Sumário: Instituição mantém aluno.

Ator primário: Faculdade.

Fluxo Principal

1. O responsável da instituição obtém os dados do aluno (nome completo, foto, RG, CPF, RA, curso, senha de usuário).

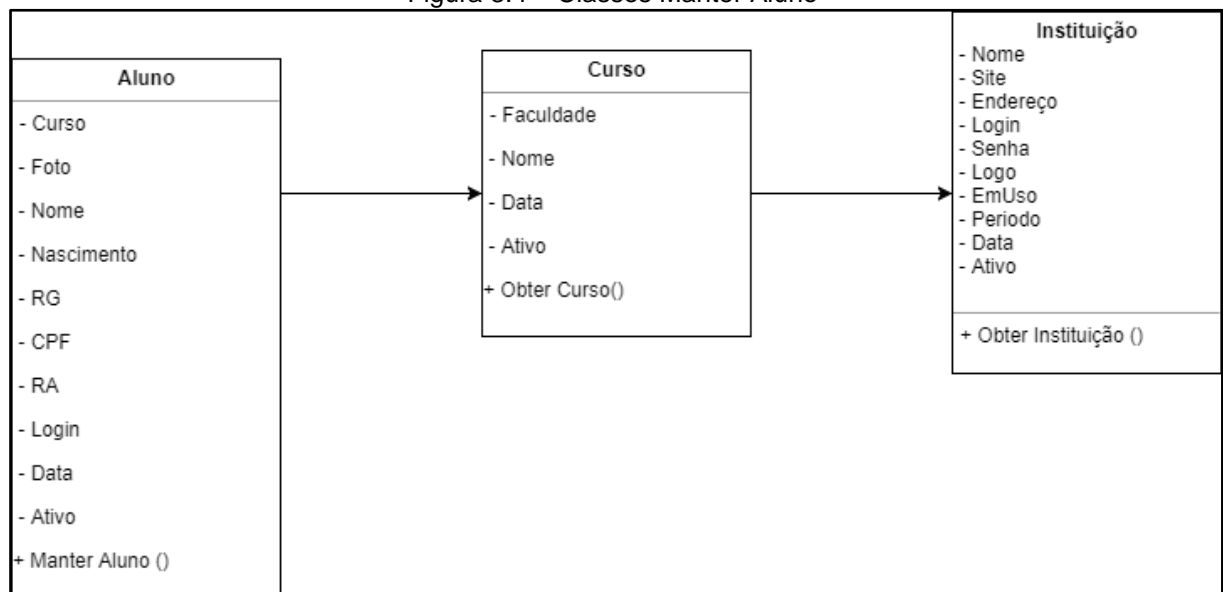
2. O responsável da instituição cadastra os dados do aluno no banco de dados online.

Fluxo Alternativo

1. O responsável da instituição poderá atualizar os dados do aluno no banco de dados online.

A Figura 3.4 apresenta as classes relacionadas ao caso de uso Manter Aluno.

Figura 3.4 – Classes Manter Aluno



Fonte: Autoria própria, 2017

## Manter Matéria

Sumário: Instituição mantém matéria.

Ator primário: Faculdade.

Fluxo Principal

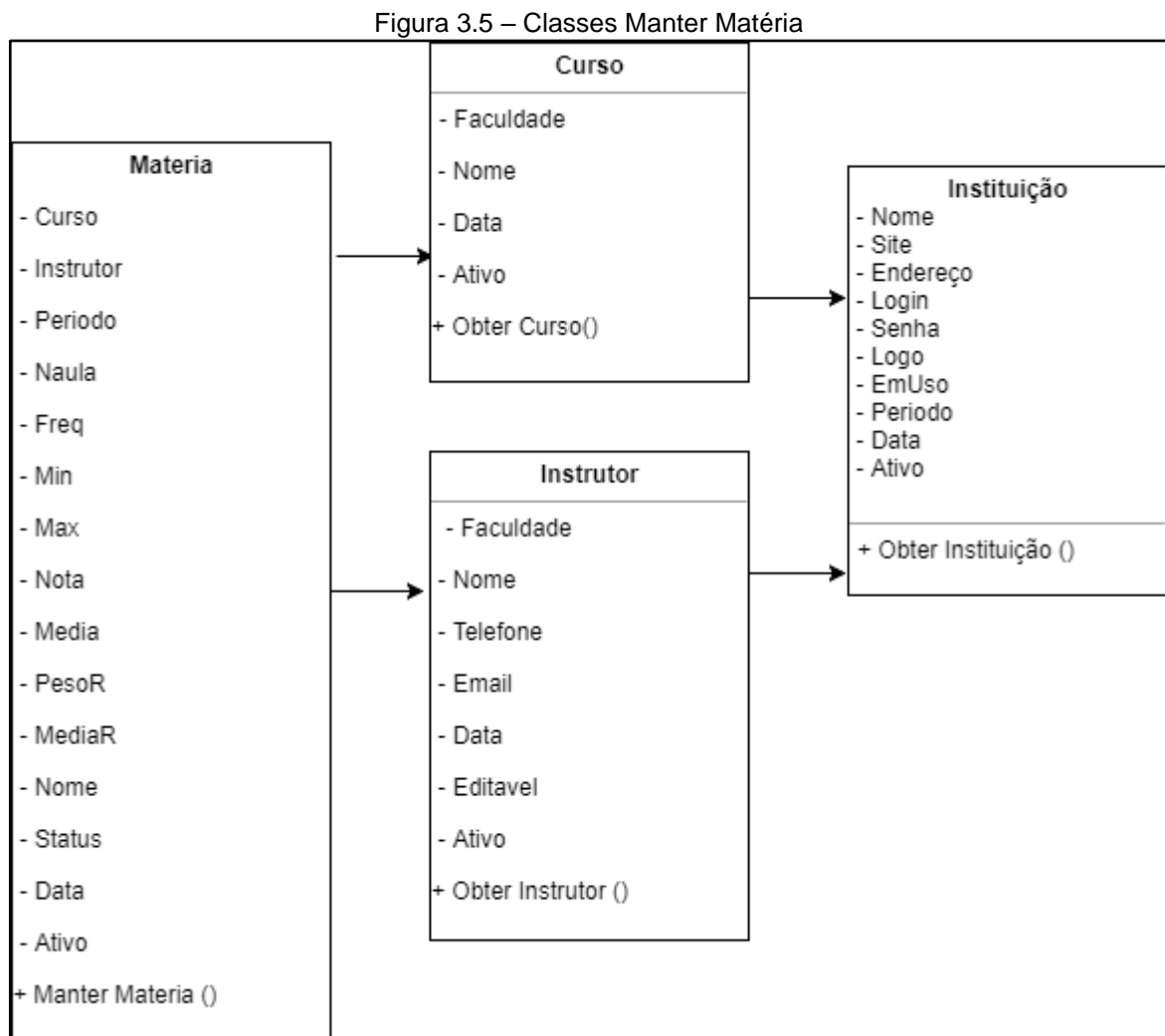
1. O responsável da instituição obtém os dados da matéria (nome da matéria, média mínima, quantidade de aulas, quantidade mínima de aulas frequentadas, peso da média de recuperação, curso e instrutor da matéria).

2. O responsável da instituição cadastra os dados da matéria no banco de dados online.

Fluxo Alternativo

1. O responsável da instituição poderá atualizar os dados da matéria no banco de dados online.

A Figura 3.5 apresenta a classe relacionada ao caso de uso Manter Matéria.



Fonte: Autoria própria, 2017

### Manter Instrutor

Sumário: Instituição mantém instrutor.

Ator primário: Faculdade.

Fluxo Principal

1. O responsável da instituição obtém os dados do Instrutor (nome, telefone de contato e e-mail para contato).

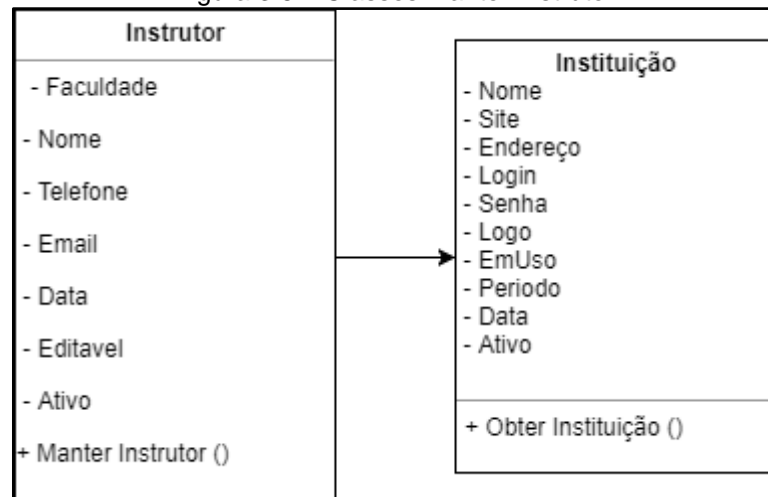
2. O responsável da instituição cadastra os dados do instrutor no banco de dados online.

Fluxo Alternativo

1. O responsável da instituição poderá atualizar os dados da instituição no banco de dados online.

A Figura 3.6 apresenta as classes relacionadas ao caso de uso Manter Instrutor.

Figura 3.6 – Classes Manter Instrutor



Fonte: Autoria própria, 2017

### Manter Aula

Sumário: Instituição mantém aula.

Ator primário: Faculdade.

Fluxo Principal

1. O responsável da instituição obtém os dados da instituição (horário de início, horário de término, dia da semana da aula e matéria da aula).

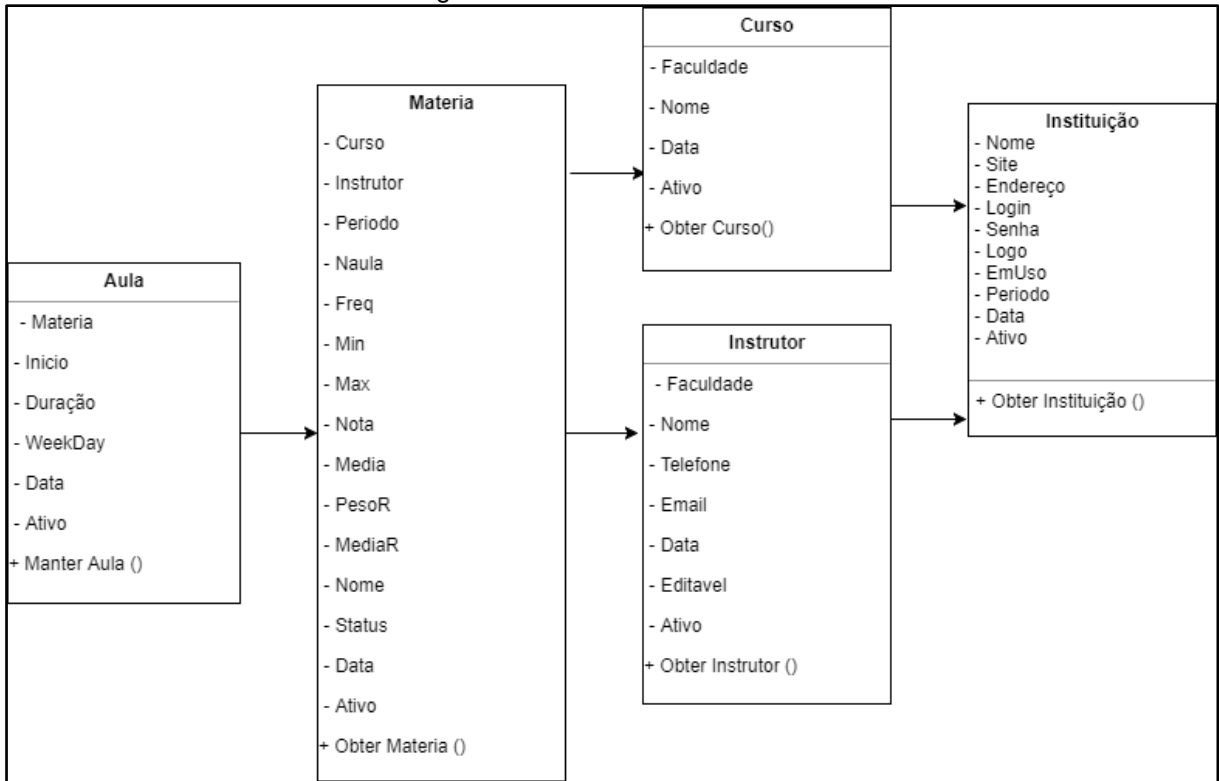
2. O responsável da instituição cadastra os dados da aula no banco de dados online.

Fluxo Alternativo

1. O responsável da instituição poderá atualizar os dados da aula no banco de dados online.

A Figura 3.7 apresenta as classes relacionadas ao caso de uso Manter Aula.

Figura 3.7 - Classes Manter Aula



Fonte: Autoria própria, 2017

### Manter Mural

Sumário: Instituição mantém mural.

Ator primário: Faculdade.

Fluxo Principal

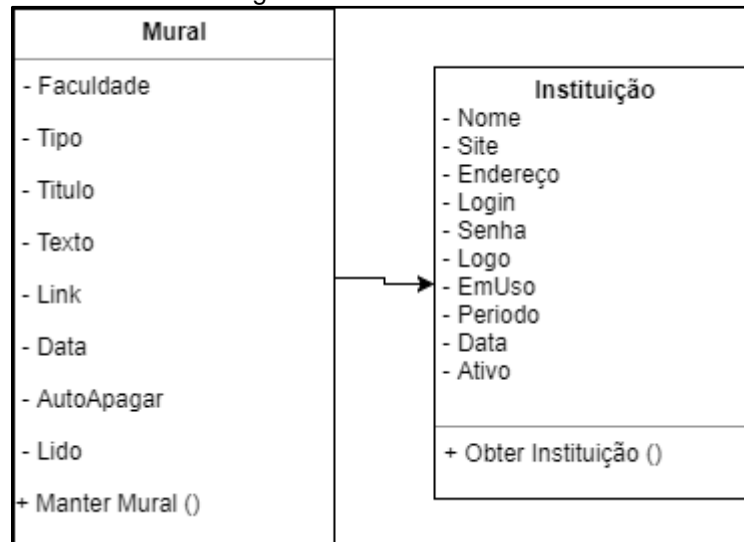
1. O responsável da instituição obtém os dados do mural (título, texto da mensagem, data e tipo da mensagem).
2. O responsável da instituição cadastra os dados da instituição.

Fluxo Alternativo:

1. O responsável da instituição poderá atualizar os dados da instituição.

A Figura 3.8 apresenta as classes relacionadas ao caso de uso Manter Mural.

Figura 3.8 – Manter Mural



Fonte: Autoria própria, 2017

### Ver Instituição

Sumário: Aluno visualiza alguns dados da faculdade (nome, endereço, site) baixados no aparelho.

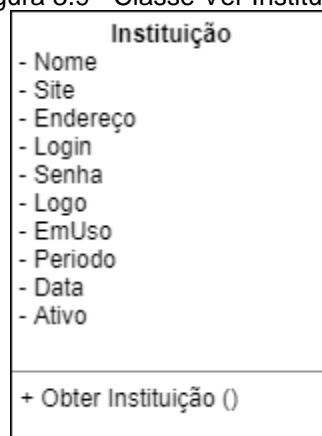
Ator primário: Aluno.

Fluxo Principal

1. O aluno visualiza os dados da instituição em que está cadastrado.

A Figura 3.9 apresenta a classe relacionada ao caso de uso Ver Instituição.

Figura 3.9 - Classe Ver Instituição



Fonte: Autoria própria, 2017

### Manter Dados Instrutor

Sumário: O aluno mantém dados de novos instrutores.

Ator primário: Aluno.

### Fluxo Principal

1. O aluno obtém os dados do novo instrutor (Nome, telefone de contato e e-mail de contato).

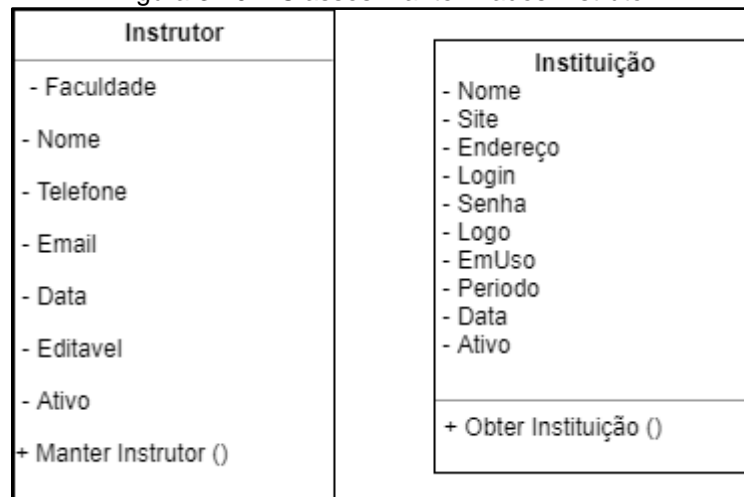
2. O aluno cadastra os dados do novo instrutor, apenas no próprio aparelho.

### Fluxo Alternativo:

1. O responsável da instituição poderá atualizar os dados do novo instrutor cadastrado.

A Figura 3.10 apresenta as classes relacionadas ao caso de uso Manter Dados Instrutor.

Figura 3.10 – Classes Manter Dados Instrutor



Fonte: Autoria própria, 2017

### Ver Aluno

Sumário: Aluno visualiza os dados do aluno em formato de carteirinha estudantil (nome completo, instituição, RA, RG, CPF, curso) baixados no aparelho.

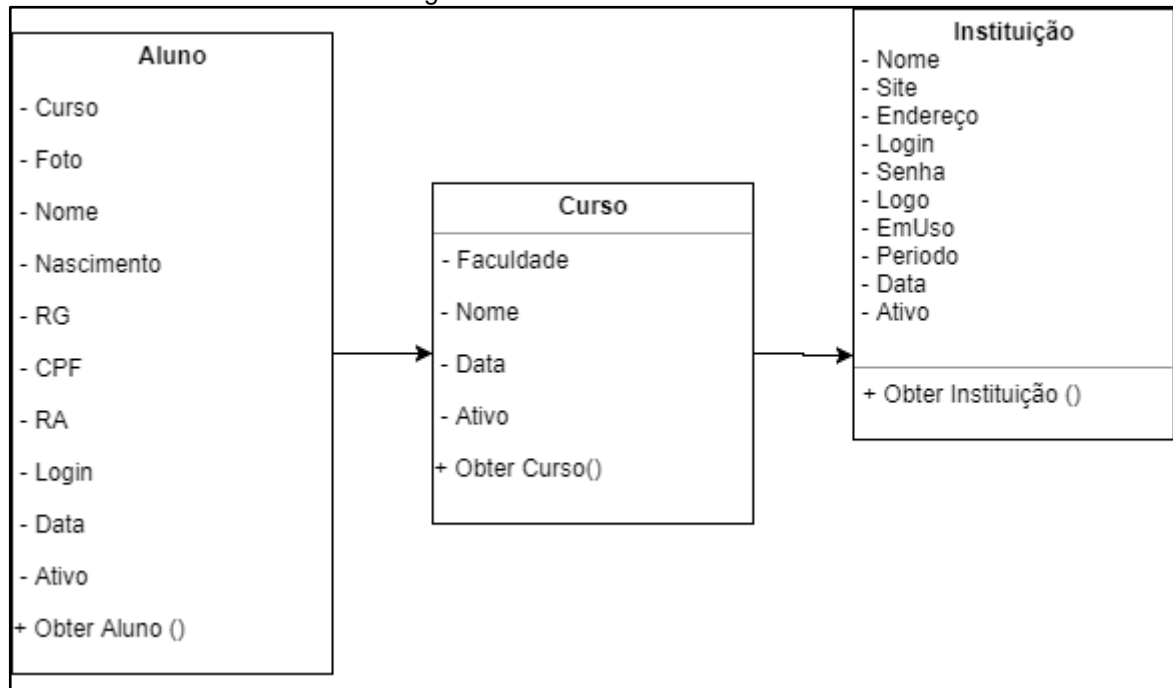
Ator primário: Aluno.

### Fluxo Principal

1. O aluno visualiza os seus dados em formato de carteirinha estudantil.

A Figura 3.11 apresenta as classes relacionadas ao caso de uso Ver Aluno.

Figura 3.11 – Classes Ver Aluno



Fonte: Autoria própria, 2017

### Manter Dados Matéria

Sumário: O aluno mantém dados das matérias baixadas no aparelho.

Ator primário: Aluno.

Fluxo Principal

1. O aluno obtém os dados da nova matéria (nome da matéria, média mínima, quantidade de aulas, quantidade mínima de aulas frequentadas, peso da média de recuperação, curso e instrutor da matéria).

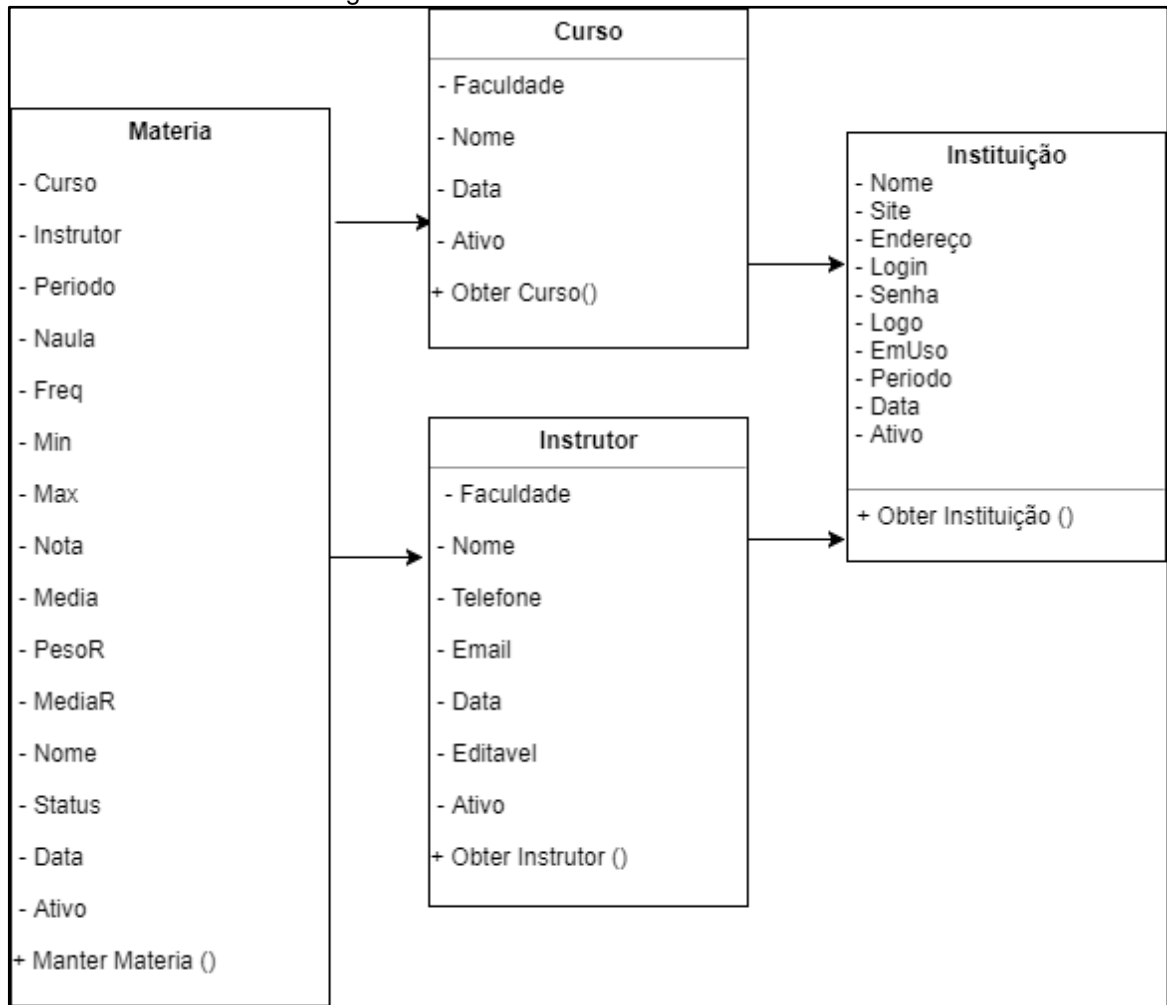
2. O aluno cadastra os dados da nova matéria, apenas no próprio aparelho.

Fluxo Alternativo:

1. O aluno poderá atualizar os dados da nova matéria.

A Figura 3.12 apresenta as classes relacionadas ao caso de uso Manter Dados Matéria.

Figura 3.12 – Classes Manter Dados Matéria



Fonte: Autoria própria, 2017

### Manter Nota

Sumário: O aluno mantém dados das notas no próprio aparelho.

Ator primário: Aluno.

Fluxo Principal

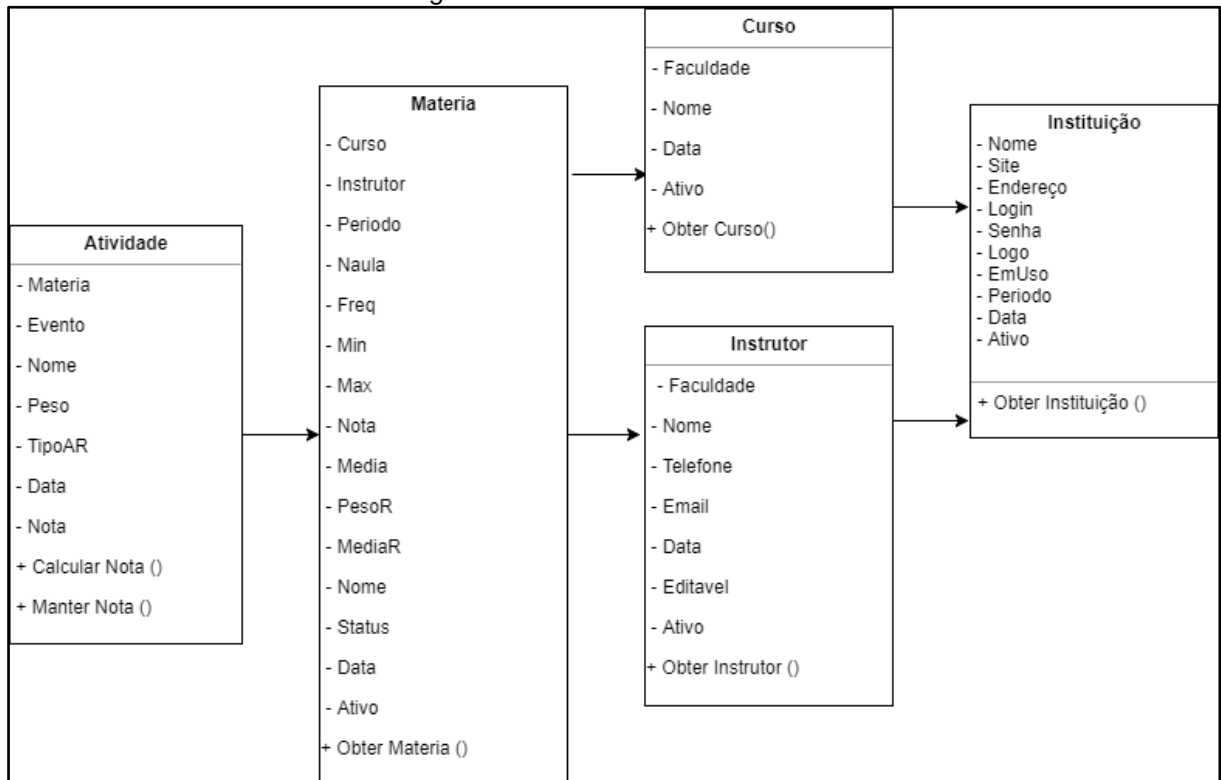
1. O aluno obtém as notas de suas atividades.
2. O aluno cadastra as notas no aplicativo.
3. O aplicativo calcula e retorna as notas de outras atividades ainda não preenchidas, notas essas necessárias para se alcançar a média mínima para ser aprovado, além de calcular a média atual do aluno.

Fluxo Alternativo:

1. O aluno poderá atualizar as notas das atividades.

A Figura 3.13 apresenta as classes relacionadas ao caso de uso Manter Nota.

Figura 3.13 - Classes Manter Nota



Fonte: Autoria própria, 2017

### Manter Atividades

Sumário: O aluno mantém dados das atividades no próprio aparelho.

Ator primário: Aluno.

Fluxo Principal

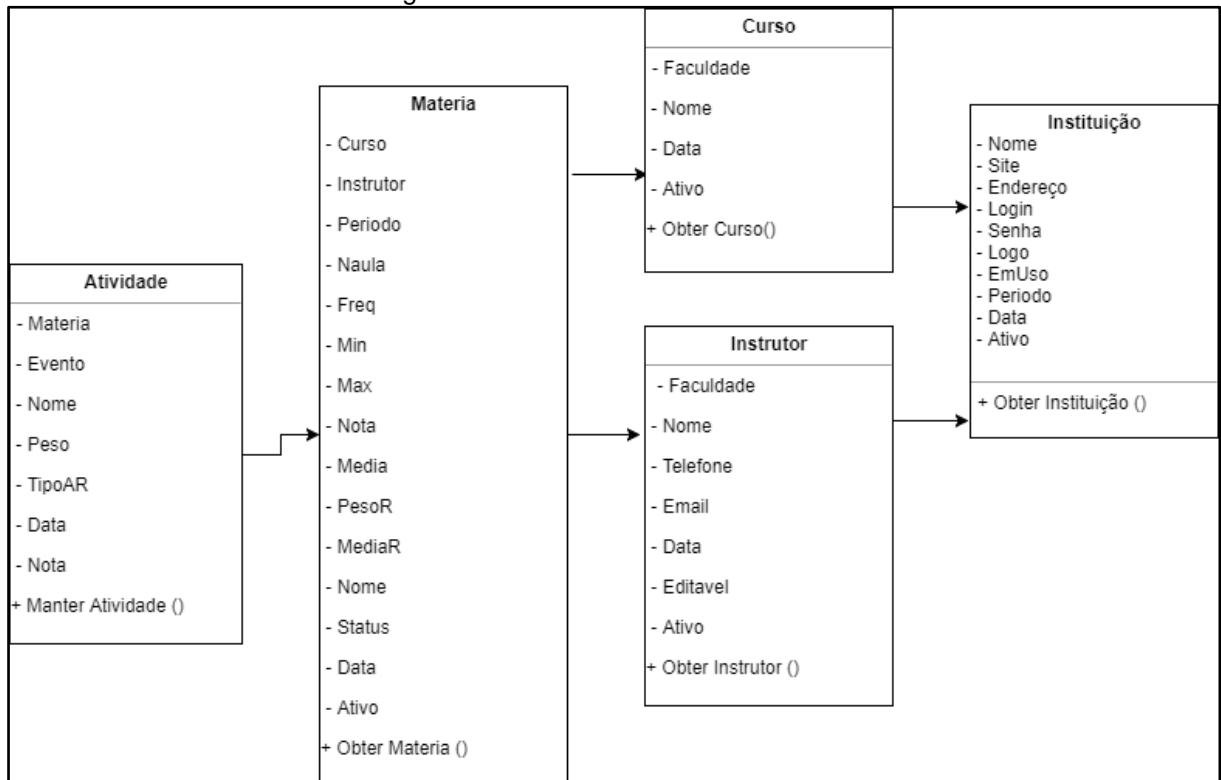
1. O aluno obtém dados da atividade (nome, peso, data).
2. O aluno cadastra os dados da atividade no próprio aparelho.

Fluxo Alternativo:

1. O aluno poderá atualizar os dados da atividade.

A Figura 3.14 apresenta as classes relacionadas ao caso de uso Manter Atividades.

Figura 3.14 - Classes Manter Atividade



Fonte: Autoria própria, 2017

### Manter Dados Aula

Sumário: O aluno mantém dados das aulas no próprio aparelho.

Ator primário: Aluno.

Fluxo Principal

1. O aluno obtém os dados da aula (horário de início, horário de término, dia da semana da aula).

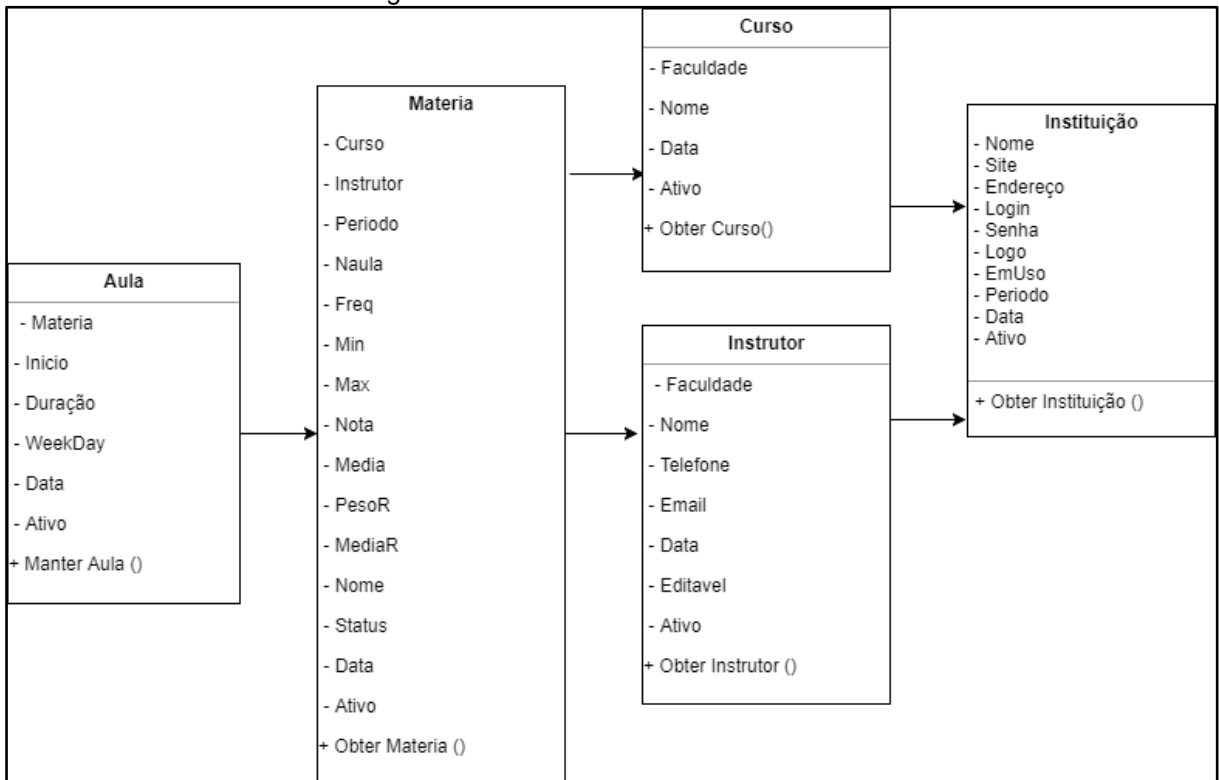
2. O aluno cadastra os dados da aula no próprio aparelho.

Fluxo Alternativo:

1. O aluno poderá atualizar os dados da aula.

A Figura 3.15 apresenta as classes relacionadas ao caso de uso Manter Dados Aula.

Figura 3.15 - Classes Manter Dados Aula



Fonte: Autoria própria, 2017

### Manter Faltas

Sumário: O aluno mantém faltas.

Ator primário: Aluno.

Fluxo Principal

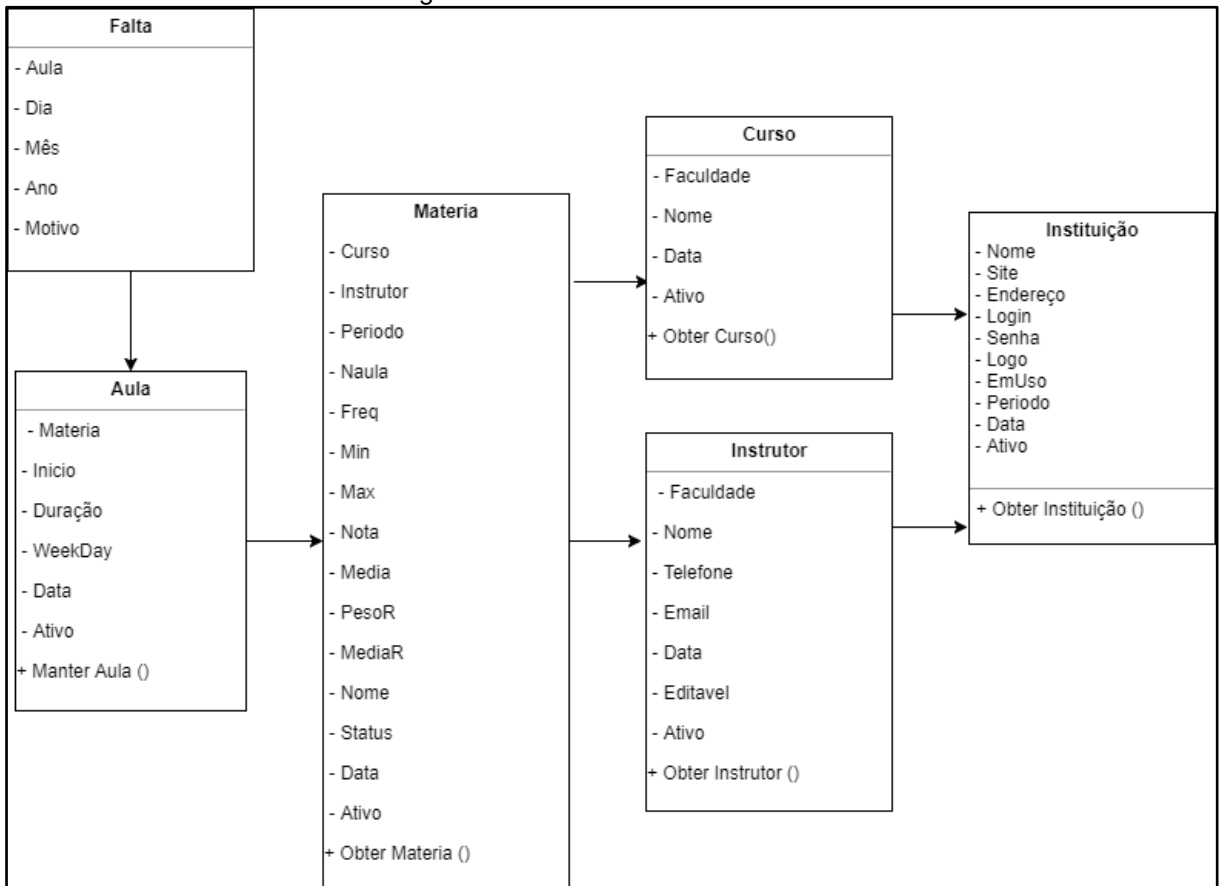
1. O aluno obtém os dados da falta (dia da falta, motivo da falta).
2. O aluno cadastra os dados da falta.

Fluxo Alternativo:

1. O aluno poderá apagar os dados da falta.

A Figura 3.16 apresenta as classes relacionadas ao caso de uso Manter Faltas.

Figura 3.16 - Classes Manter Faltas



Fonte: Autoria própria, 2017

### Ver Mural

Sumário: O aluno visualiza dados do mural no banco de dados online.

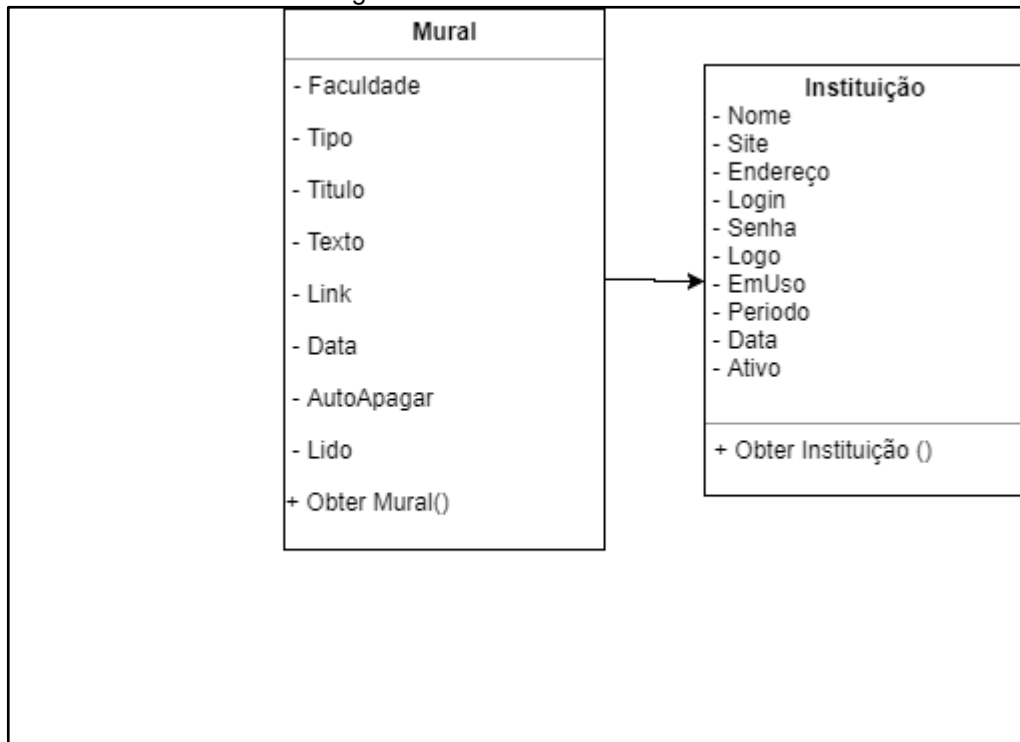
Ator primário: Aluno.

Fluxo Principal

1. O aluno poderá visualizar os dados do mural disponibilizados pela instituição.

A Figura 3.17 apresenta as classes relacionadas ao caso de uso Ver Mural.

Figura 3.17 - Classes Ver Mural



Fonte: Autoria própria, 2017

### Manter Anotação

Sumário: O aluno mantém anotações

Ator primário: Aluno.

Fluxo Principal

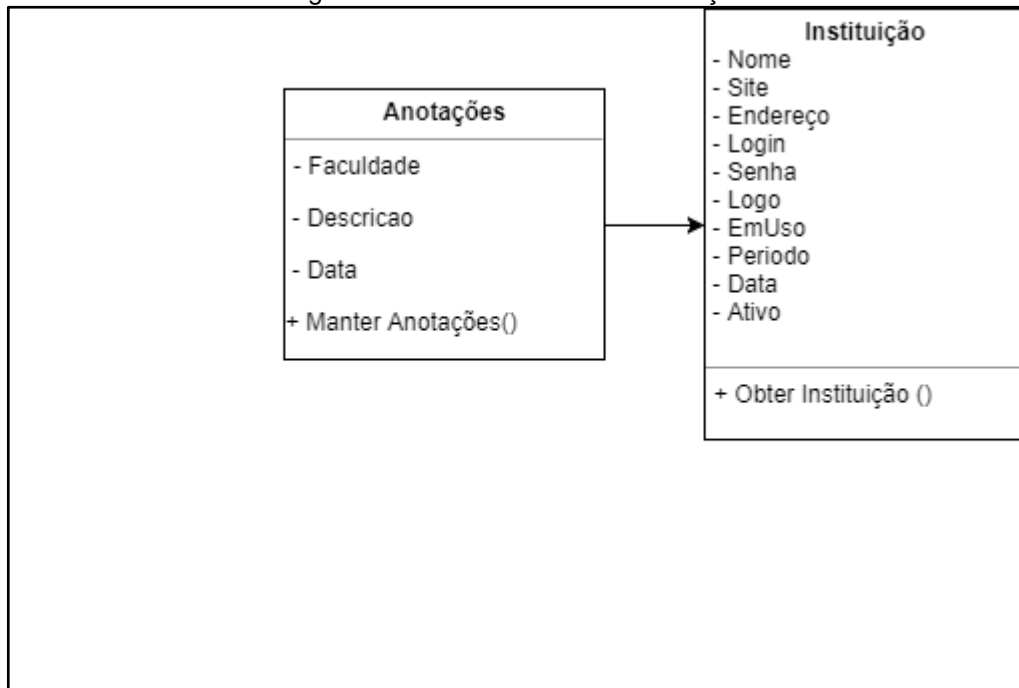
1. O aluno obtém os dados da anotação (Texto da anotação).
2. O aluno cadastra os dados da anotação.

Fluxo Alternativo:

1. O aluno poderá atualizar os dados da anotação.

A Figura 3.18 apresenta as classes relacionadas ao caso de uso Manter Anotação.

Figura 3.18 - Classes Manter Anotação



Fonte: Autoria própria, 2017

### 3.6 Métrica UCP

Para calcular a estimativa de custos, prevista anteriormente no termo de abertura do projeto, foi utilizado a métrica UCP (Use Case Point).

Este método foi criado por Gustav Karner para medir projetos de software orientado a objetos, com base no modelo de caso de uso e realizar a estimativa inicial de custos, com o levantamento de requisitos.

Primeiramente foi definido o peso dos atores do projeto, que no caso são aluno e instituição. Devido à relevância e às funções que possuem nos casos de uso, foram considerados complexos.

É importante ressaltar que para cada ator, há um nível de complexidade. Se for simples, o nível de complexidade é classificado com o valor 1; se for médio, é classificado como 2; e se for complexo, classificado como 3, que é o que acontece no projeto abordado e que pode ser visualizado na Tabela 3.1.

Tabela 3.1 - Peso dos Atores

<b>Peso dos Atores</b>		
<b>Ator</b>	<b>Complexidade</b>	<b>Peso</b>
Instituição	Complexo	3
Aluno	Complexo	3

Fonte: A autoria própria, 2017

Após definir os níveis de complexidade de cada ator, deve-se somar os valores para descobrir o UAW ou, em tradução livre, o peso não ajustado dos atores. É um fator contribui no tamanho do software a ser desenvolvido. No caso estudado, obteve-se o valor 6. Para calculá-lo, deve-se considerar a seguinte conta:

$1 \times \text{Quantidade}(As) + 2 \times \text{Quantidade}(Am) + 3 \times \text{Quantidade}(Ac)$ , nos quais  $As =$  Ator simples,  $Am =$  Ator médio e  $Ac =$  Ator complexo.

Em seguida, são levantados os casos de uso do projeto, a quantidade de classes de cada um para, depois, definir o nível de complexidade e o peso deles. Quando há de 1 a 4 classes, a complexidade é considerada simples; de 5 a 10, complexidade média; e de 11 em diante, do tipo complexa. A Tabela 3.2 mostra os valores.

Tabela 3.2 - Casos de Uso e Pesos

<b>Caso de Uso</b>	<b>Classes</b>	<b>Complexidade</b>	<b>Peso</b>
Manter Instituição	1	Simple	5
Manter Aluno	3	Simple	5
Manter Matéria	4	Simple	5
Manter Instrutor	2	Simple	5
Manter Aula	5	Médio	10
Manter Mural	2	Simple	5
Ver Instituição	1	Simple	5
Manter Dados Instrutor	2	Simple	5
Ver Aluno	3	Simple	5
Manter Dados Matéria	4	Simple	5
Manter Nota	5	Médio	10
Manter Atividades	5	Médio	10
Manter Dados Aula	5	Médio	10
Manter Faltas	6	Médio	10
Ver Mural	2	Simple	5
Manter Anotação	2	Simple	5

Fonte: A autoria própria, 2017

Após definir os níveis de complexidade e os pesos de cada caso de uso, deve-se somar os valores para descobrir o UUCW ou, em tradução livre, o peso não ajustado dos casos de uso. No caso estudado, o valor obtido foi 105.

Para calculá-lo, deve-se considerar a seguinte conta:  $5 \times \text{Quantidade(UCs)} + 10 \times \text{Quantidade(UCm)} + 15 \times \text{Quantidade(UCc)}$ , nos quais UCs = Caso de uso simples, UCm = Caso de uso médio e UCc = Caso de uso complexo.

Ao final disso, deve-se calcular o peso total não ajustado, ou UUCP, que é a soma do UAW com o UUCW. No caso estudado, o valor do UAW, que foi 6 mais o valor do UUCW, que foi 105, o peso total não ajustado ou UUCP foi de 111.

Além dos fatores já supracitados, há também dois fatores importantes para a métrica UCP: TCF e EF. O TCF ou o fator de complexidade técnica consiste nos elementos básicos que influenciam na dificuldade de construção do sistema. A Tabela 3.3 exhibe os tipos de fatores de complexidade considerados no projeto, o valor do fator, peso do índice bem como Tfactor, que é o valor total obtido da multiplicação do fator pelo índice.

Tabela 3.3 - Fatores de Complexidade Técnica

<b>Fator de Complexidade Técnica</b>	<b>Fator</b>	<b>Peso do Índice</b>	<b>FatorxÍndice</b>
Sistema Distribuído	3	2	6
Desempenho para respostas	1	2	2
Usuário Final eficiente	2	1	2
Complexidade de processamento	2	1	2
Reuso de código	1	1	1
Facilidade de instalação	4	0,5	2
Facilidade de uso	1	0,5	0,5
Portabilidade	4	2	8
Facilidade de alterações	3	1	3
Uso concorrente	2	1	2
Recursos de segurança	3	1	3
Acessível por Terceiros	1	1	1
Necessidade de treinamento	1	1	1
		<b>Tfactor</b>	<b>33,5</b>

Fonte: Autoria própria, 2017

O cálculo do TCF é feito da seguinte maneira:

$$TCF=0,6+(0,01 * Tfactor)$$

Sabendo do valor do Tfactor, que foi de 33,5, o valor do TCF foi de 0,935.

O EF ou fator ambiental representa os elementos facilitadores conhecidos que influenciam na produtividade da equipe de desenvolvimento . A Tabela 3.4 exhibe os tipos de fatores de complexidade ambientais considerados no projeto, o valor do fator, o peso do índice bem como Efactor, que é o valor total obtido da multiplicação do fator pelo índice.

Tabela 3.4 - Fatores de Complexidade Ambiental

<b>Fator de Complexidade Ambiental</b>	<b>Fator</b>	<b>Peso do Índice</b>	<b>FatorxÍndice</b>
Seguirá um Processo de Desenvolvimento	2	1,5	3
Experiência na Aplicação	2	0,5	1
Experiência em OO	3	1	3
Capacidade do Analista	2	0,5	1
Motivação	4	1	4
Estabilidade dos Requisitos	2	2	4
Deenvolvimento em tempo parcial	2	-1	-2
Dificuldade com a Ling. Programação	2	-1	-2
		<b>Efactor</b>	12

Fonte: Autoria própria, 2017

O cálculo do EF é feito da seguinte maneira:

$$EF=1,4+(-0,03 * Efactor)$$

Sabendo do valor do Efactor, que foi de 12, o valor do TCF foi de 1,04.

A partir dos cálculos de todos os fatores e os valores obtidos, há o cálculo final do UCP, que consiste da multiplicação dos valores de todos os outros fatores.

$$UCP=UUCP*TCF*EF$$

$$UCP= 111*0,935*1,04$$

O valor final do UCP foi de 108.

O UCP representa o tamanho do projeto e o valor obtido e o valor é importante para converter o UCP em HH (horas/homem), que consiste do esforço aplicado no projeto.

1 UCP equivale de 15 a 30 HH. Para o projeto, foi considerado o valor de 20HH. Com isso, o valor do UCP, que foi de 108 multiplicado pelo valor de 20HH, foi obtido o valor total de 2160 HH para o projeto.

Por fim, foi feita a conversão para horas de trabalho e a gestão do custo do software. Para cada etapa foi considerada uma porcentagem de hora, o valor da hora recurso, as horas de projeto, considerando o índice de produtividade de 20HH e o custo do projeto em cada etapa, como pode ser visualizado na Tabela 3.5.

Tabela 3.5 – Gestão do custo de software

<b>Etapas</b>	<b>Hora Recurso - R\$</b>	<b>Horas do Projeto</b>	<b>Custo do Projeto - R\$</b>
Modelagem de Negócios - 10%	60,00	216	12.960,00
Requisitos - 10%	60,00	216	12.960,00
Análise e Projeto - 20%	40,00	432	17.280,00
Implementação - 30%	35,00	648	22.680,00
Testes - 20%	35,00	432	17.280,00
Implantação -10%	30,00	216	12.960,00
<b>TOTAL</b>		<b>2160</b>	<b>87.480,00</b>

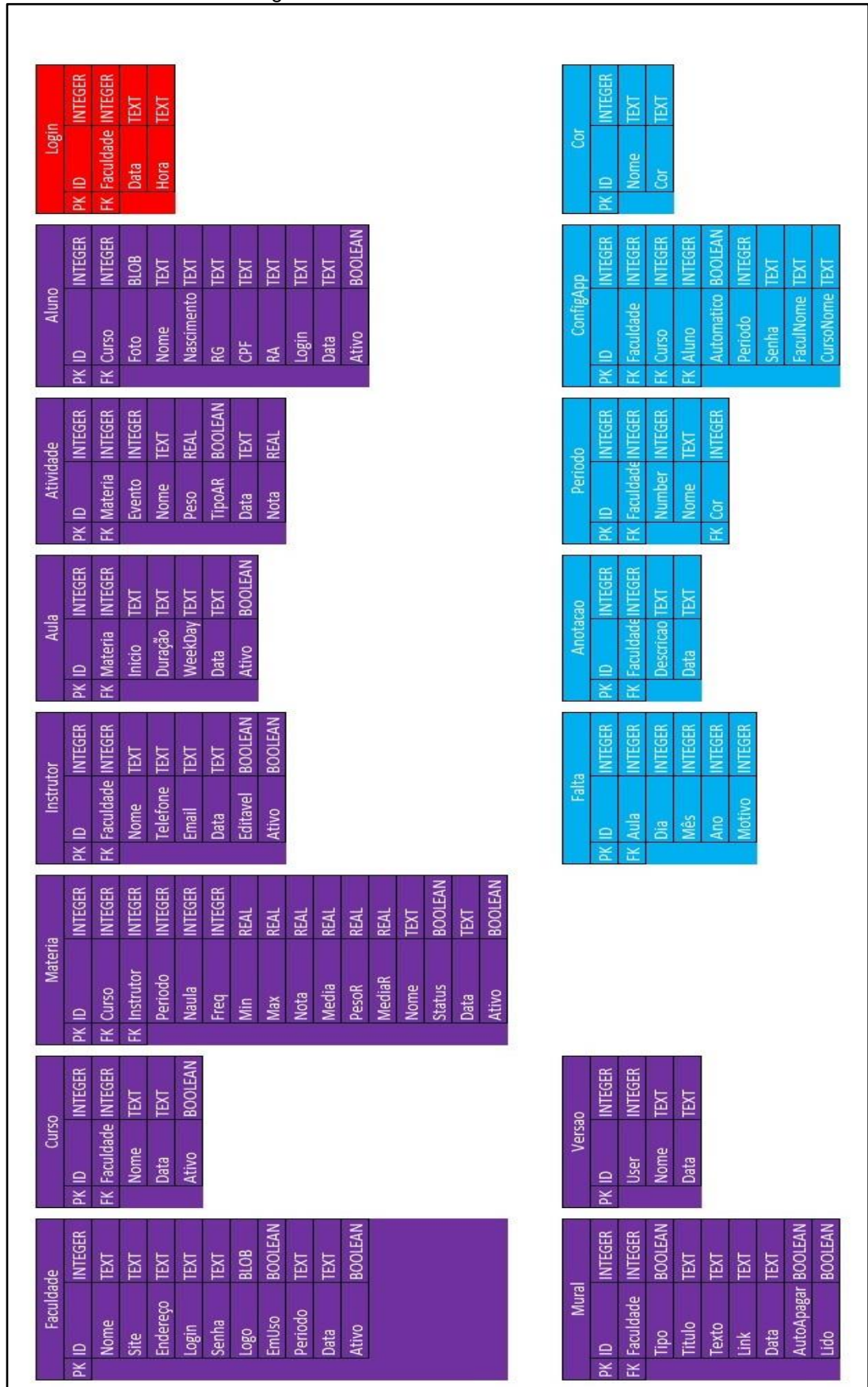
Fonte: Autoria própria, 2017

Com todos os cálculos efetuados, a estimativa do custo inicial do projeto foi no valor de R\$ 87.480,00.

### **3.7 Banco de dados**

Para visualizar as entidades, os atributos bem como as chaves primárias e estrangeiras e os relacionamentos entre as entidades. As entidades em roxo representam aquelas usadas no aplicativo; as entidades em azul, tanto no aplicativo quanto na comunicação online e o login, que está em vermelho. A Figura 3.19 mostra o banco de dados do projeto.

Figura 3.19 - Banco de dados



Fonte: Autoria própria, 2017

### **3.8 Descrição do método de desenvolvimento**

Para descrever os métodos de desenvolvimento do projeto, será apresentado um diário de bordo, com o passo a passo da criação do aplicativo e as imagens das telas que, porventura, vão compor o sistema do projeto.

#### **3.8.1 Diário de Bordo**

##### **Estabelecido o conteúdo do aplicativo**

Tendo como base o funcionamento que o sistema foi planejado e efetuado esboços para o tema e a aparência do aplicativo, assim como também seus princípios de funcionamento e levantados os dados que o aplicativo deverá mostrar e armazenar.

##### **Elaborado plano de ação**

Planejado as etapas do desenvolvimento e suas fases de criação.

##### **Efetuada estruturação do banco de dados**

Levantado e organizados os dados necessários para o armazenamento seguindo os parâmetros estabelecidos, elaborado a ligação entre os dados e esquematizado o layout das Tabelas necessárias.

##### **Estabelecido a quantidade e caminho para as telas**

Levantado através dos esboços e da sequência lógica que as informações devam ser mostradas foi gerado uma ordem para as telas do aplicativo.

##### **Gerado classe de funções auxiliares**

Criado uma classe para funções gerais do aplicativo, funções essas que podem ser acessadas por mais de uma tela, a classe pode ser alterada conforme necessidade do desenvolvimento.

### **Gerado classe de conexão ao banco de dados**

Para trabalhar com o sistema online, e não correr o risco de danificar diretamente o banco de dados, foi criada uma classe de ligação ao banco de dados personalizada, para que pudesse acessar o banco de dados repetidas vezes sem correr o risco de danificá-lo

### **Gerado as telas normais do aplicativo**

A partir do esboço foi realizado a geração das telas do aplicativo, ainda sem funcionamento.

### **Personalizando as telas do aplicativo**

Algumas telas necessitam de características personalizadas, aos quais a API não libera em seu sistema, características que para serem alcançadas devem ser desenvolvidas separadamente.

### **Gerado interligação entre as telas existentes**

Ainda sem gerar o código para o funcionamento do aplicativo, foi criado meios de uma tela ir para uma outra.

#### **3.8.2 Telas**

A seguir serão descritas as telas que compõem, bem como as imagens de cada uma, como exemplo.

#### **Tela “Splash”**

Tela com formulários para inserção do nome da instituição, curso, RA do aluno e sua senha, além do botão de login para logar o usuário no sistema do aplicativo, validando as informações inseridas, como representada na Figura 3.20.

Figura 3.20 - Tela Splash

A tela splash de login possui um fundo azul claro com cantos arredondados no topo. No topo centralizado, há um ícone preto de um capacete acadêmico sobre uma cabeça. Abaixo do ícone, há dois campos de texto brancos com bordas azuis: o primeiro contém o texto 'Faculdade' e o segundo contém 'Curso'. Seguem-se dois campos de entrada de texto brancos com bordas azuis e linhas de base desenhadas, contendo os textos 'Registro Aluno' e 'Password'. Abaixo desses campos, há um botão cinza com o texto 'Logar'. No rodapé, o texto 'Independente' é exibido em uma cor azul mais escura.

Faculdade

Curso

Registro Aluno

Password

Logar

Independente

Fonte: Autoria própria, 2017

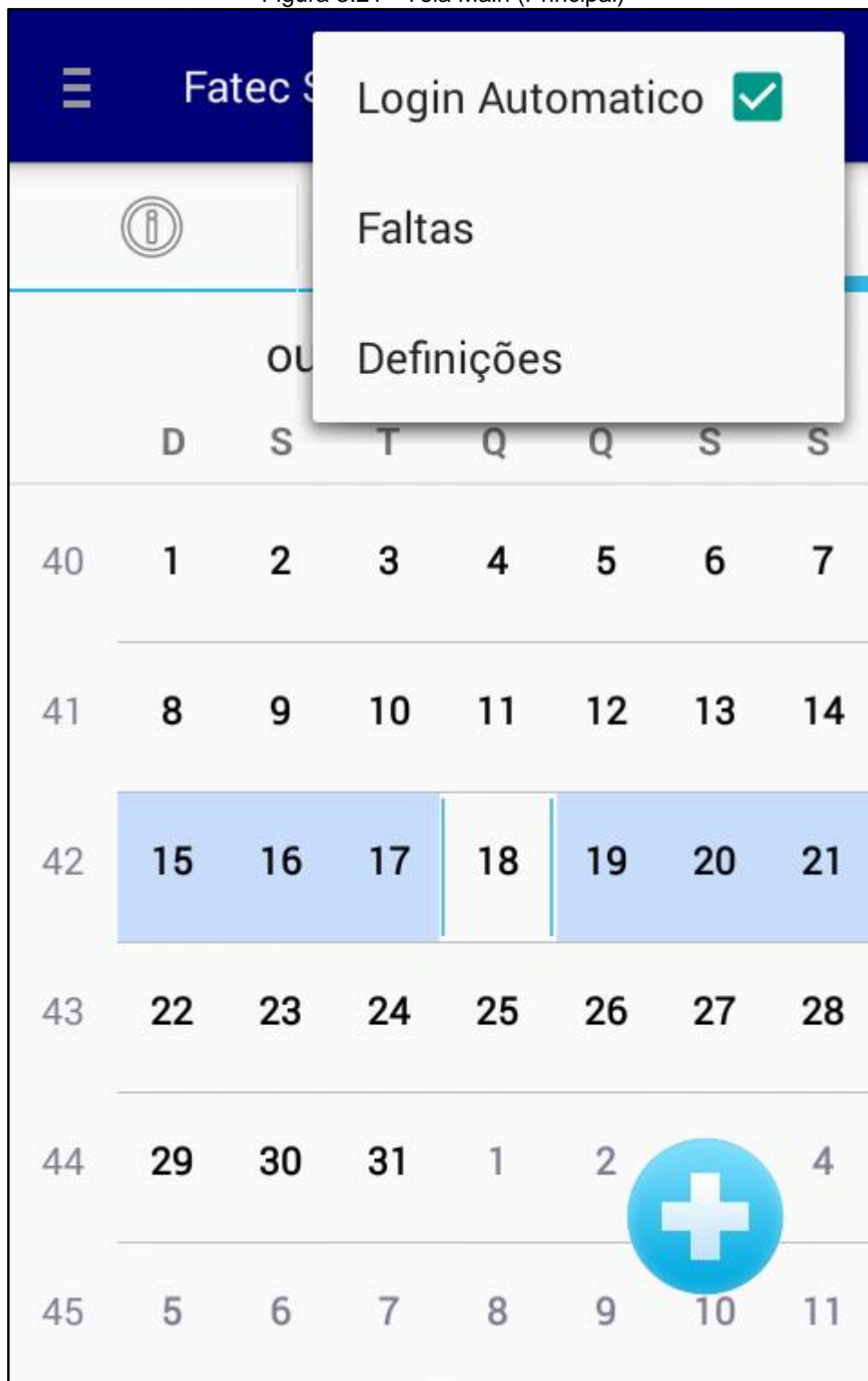
Em caso de o aparelho estar conectado à internet, é verificado o banco de dados online para obter esta informação; caso não esteja conectado, o sistema acessa apenas as informações já baixadas.

No primeiro login do usuário o aplicativo baixa todas as informações necessárias sobre a instituição e do aluno e em todas as demais vezes, baixa apenas os dados para atualizar o Mural.

### **Tela “Main”**

Possuindo no menu opções para acessar a tela de anotações, de faltas e de definições gerais, além da opção de login ao sistema de forma automática, como representado na Figura 3.21.

Figura 3.21 - Tela Main (Principal)



Fonte: Autoria própria, 2017

A tela Main é constituída de sub-telas, sendo:

1) Barra Lateral:

Tela de lista para acessar os períodos, contendo acima uma faixa da faculdade e foto do aluno que acessa a tela de identificação do aluno, além de conter menu próprio para adicionar período.

2) Mural:

Tela de listagem das informações do mural, informando à classe do aviso e seu título. Possui um botão para fácil atualização.

3) Matérias:

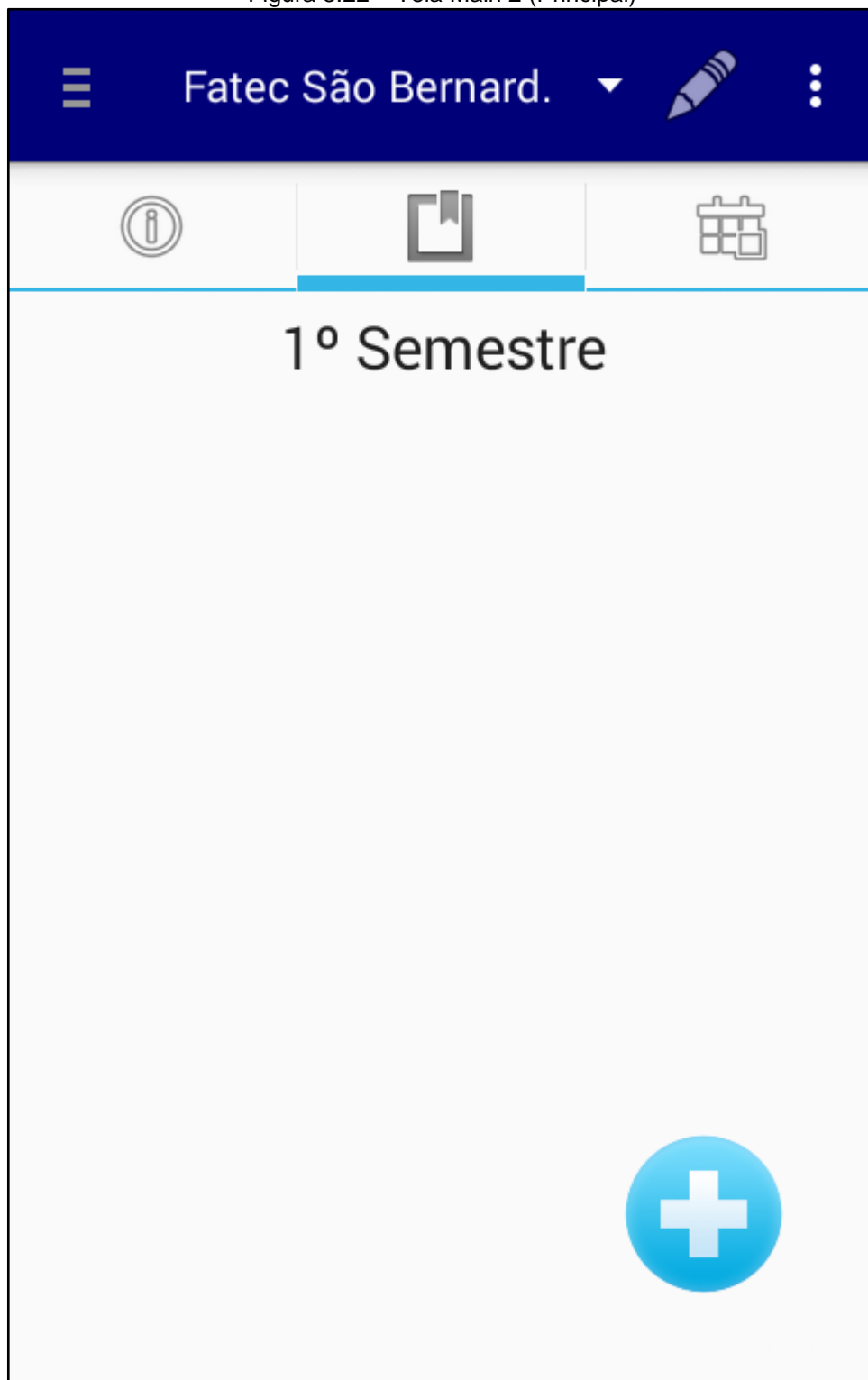
Tela de listagem das informações das matérias no período selecionado. Possui um botão para adicionar matéria.

4) Calendário:

Tela de calendário onde o dia selecionado mostra os eventos. Possui botão para adicionar falta na data atual.

A Figura 3.22 mostra um outro exemplo da tela Main.

Figura 3.22 – Tela Main 2 (Principal)



Fonte: Autoria própria, 2017

### Tela “ID”

Possui foto e dados do aluno e formato de uma carteira de estudante e permite acesso a tela de QR CODE, que pode ser visualizado na Figura 3.23.

Figura 3.23 – Tela ID



Fonte: Autoria própria, 2017

Logo abaixo, na Figura 3.24, aparece o formato da carteirinha virtual do estudante.

Figura 3.24 – Carteirinha do estudante

Fatec São Bernardo do Campo - Adib Moisés Dib

NOME  
Renan Santos de Oliveira

CURSO  
Informática para Negócios

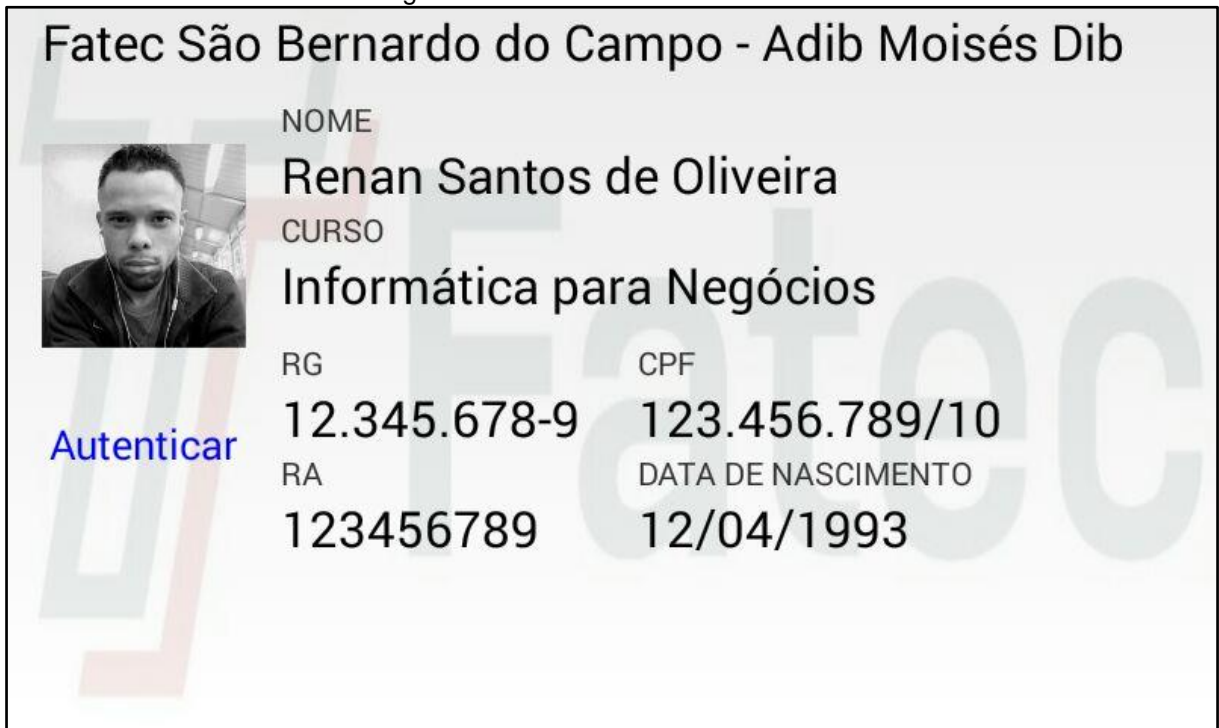
RG  
12.345.678-9

CPF  
123.456.789/10

RA  
123456789

DATA DE NASCIMENTO  
12/04/1993

[Autenticar](#)



Fonte: Autoria própria, 2017

**Tela “QR CODE”**

Tela com um QR Code feito a partir de informações do aluno, que é atualizado a cada minuto. A Figura 3.25 mostra um exemplo de uma tela QR Code.

Figura 3.25 – Tela QR Code

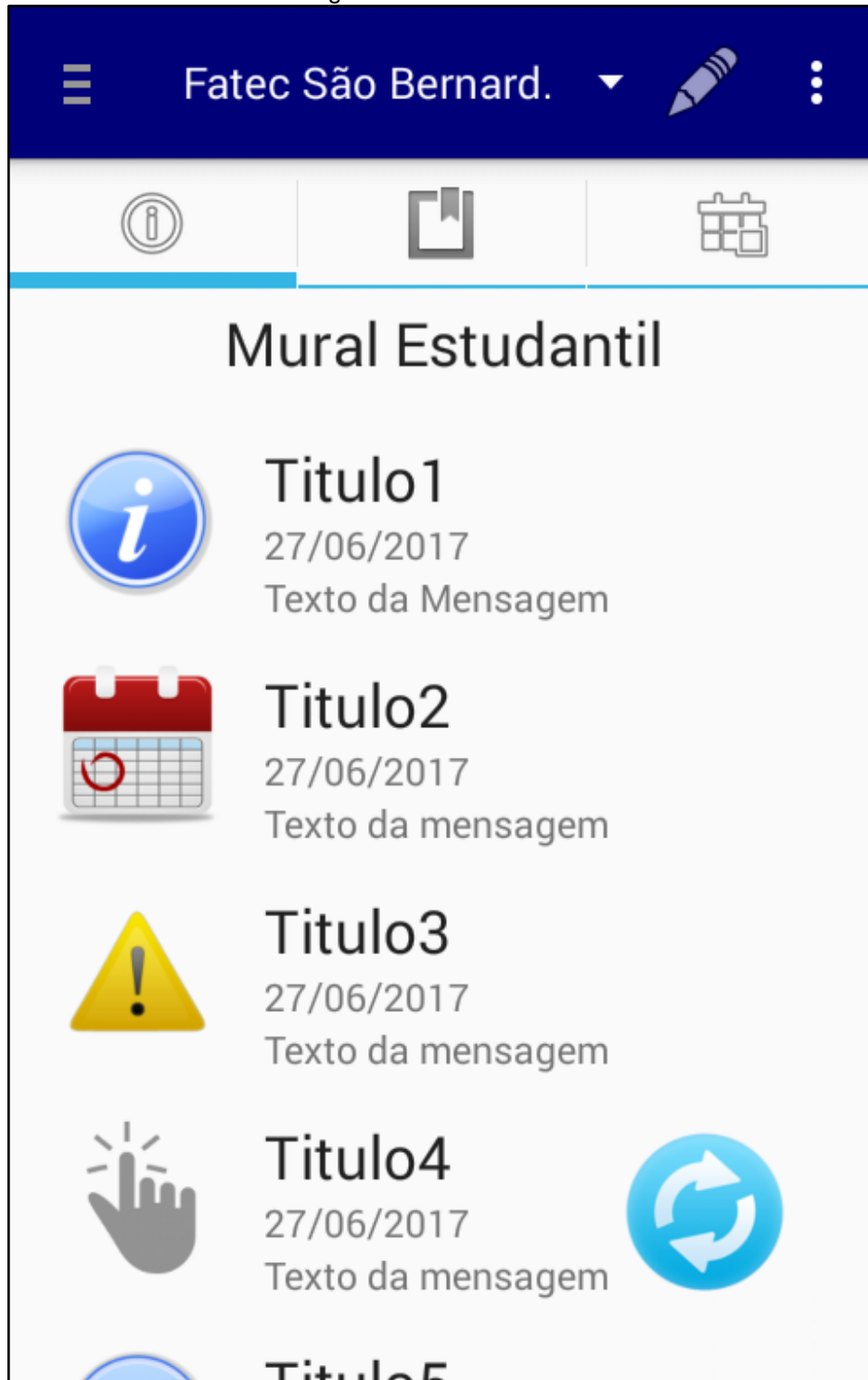


Fonte: Autoria própria, 2017

### **Tela “Mural”**

Mostra todos os dados da mensagem selecionada na lista na sub-tela da tela Main, além de permitir acesso ao link da mensagem, como pode ser visto na Figura 3.26.

Figura 3.26 – Tela Mural



Fonte: Autoria própria, 2017

### Tela “Dados”

Contém os principais dados das matérias, como professor responsável, nome da matéria, notas e médias, e pode ser acessada em três etapas: ao adicionar uma

nova matéria, ao editar os dados da matéria e ao editar dados padrões para o aplicativo. A Figura 3.27 mostra um exemplo da tela Dados.

Figura 3.27 – Tela Dados

← Dados

Materia

Instrutor

Notas:

Min: 0 Máx: 10

ATIVIDADES

Media: 6

RECUPERAÇÃO

Fonte: Autoria própria, 2017

**Tela “Matéria”**

Tela para realizar a inserção das notas em suas respectivas atividades e calcular as notas mínimas para as futuras atividades se baseando na média cadastrada. Possui botão para cálculo rápido, acesso a informações de contato do instrutor, se estas estiverem cadastradas, como visto na Figura 3.28.

Figura 3.28 – Tela Matéria

← Dados

Materia  
Informática

Instrutor  
Cesar

Notas:  
Min: 0 Máx: 10

ATIVIDADES

Media: 6

RECUPERAÇÃO

Fonte: Autoria própria, 2017

### Tela “Atividades”

Lista as atividades registradas, permite adicionar nova atividade, e salva os dados das já registradas no calendário do aparelho, como visto na Figura 3.29.

Figura 3.29 - Tela Atividades



Fonte: Autoria própria, 2017

### Tela “Recuperação”

Lista as recuperações registradas, permite adicionar nova atividade, e salva os dados das já registradas no calendário do aparelho, como visto na Figura 3.30.

Figura 3.30 - Tela Recuperação

← Dados Gerais

Notas:

Min: 0 Máx: 10

ATIVIDADES

Media: 6

RECUPERAÇÃO

Media Recuperação: 6

Peso Recuperação: 50

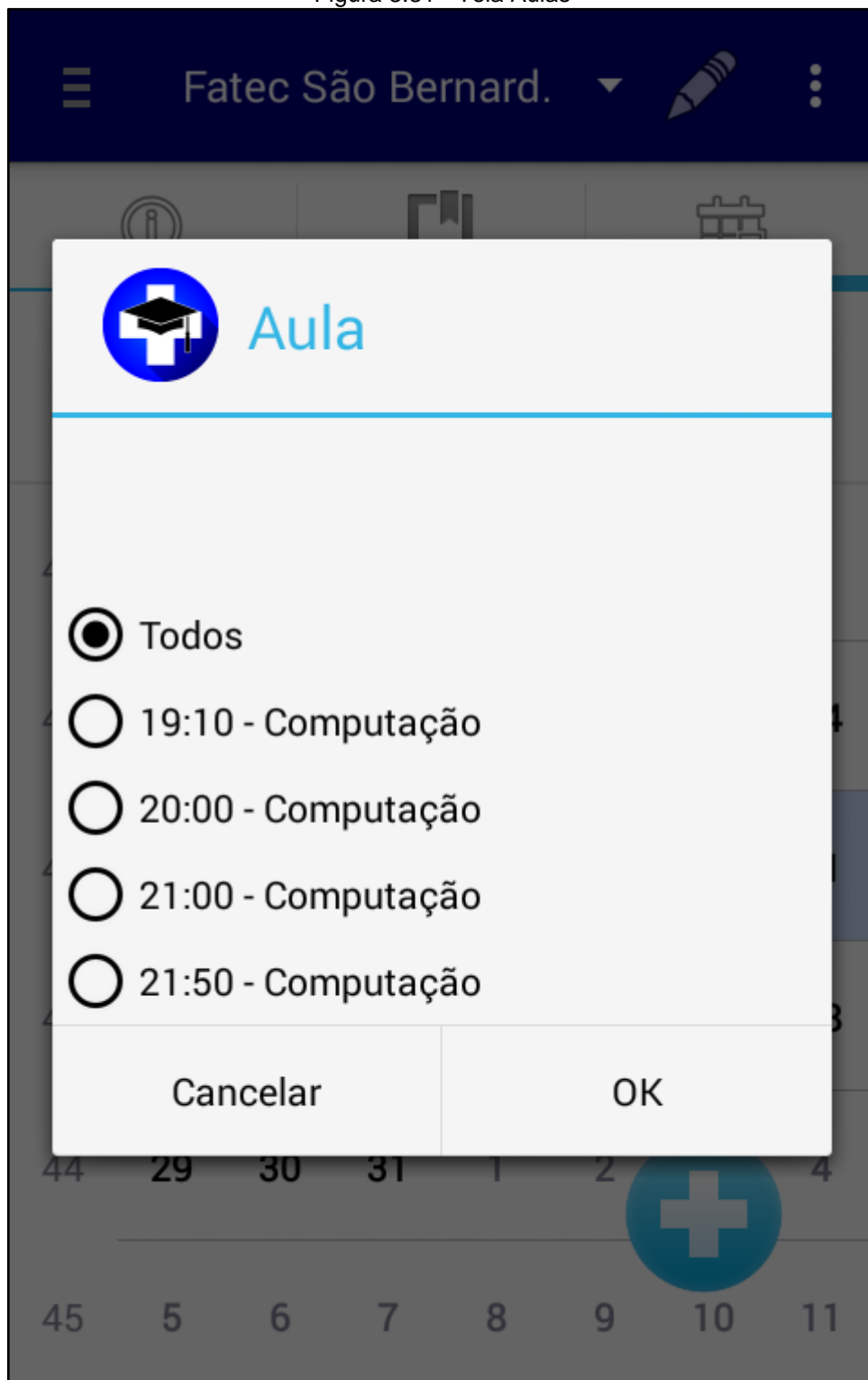
Quant. Aulas:

Fonte: Autoria própria, 2017

**Tela “Aulas”**

Lista as aulas de cada matéria, assim como seu horário de início, de término e data, permite adicionar nova aula, como visualizado na Figura 3.31.

Figura 3.31 - Tela Aulas



Fonte: Autoria própria, 2017

### Tela "Anotações"

Lista as anotações criadas pelo aluno. A Figura 3.32 exibe um exemplo da tela Anotações.

Figura 3.32 - Tela Anotações



Fonte: Autoria própria, 2017

### **Tela “Nota”**

Edita ou cria uma nova nota, como visto na Figura 3.33.

Figura 3.33 - Tela Nota



Fonte: Autoria própria, 2017

### **Tela “Faltas”**

Lista as faltas registradas no aplicativo, além de permitir a adição de novas faltas independente do dia. A Figura 3.34 mostra um exemplo da tela Faltas.

Figura 3.34 - Tela Faltas

Fatec São Bernard.

**Falta Marcada**

Motivo

- Nenhum
- Acidente
- Atrasou
- Doente
- Falta coletiva
- Falta normal
- Não respondeu chamada

Cancelar OK

4 5 6 7 8 9 10 11

Fonte: Autoria própria, 2017

### Tela “Definições”

Permite acesso às telas de configurações do aplicativo, como visto na Figura

3.35.

Figura 3.35 - Tela Definições

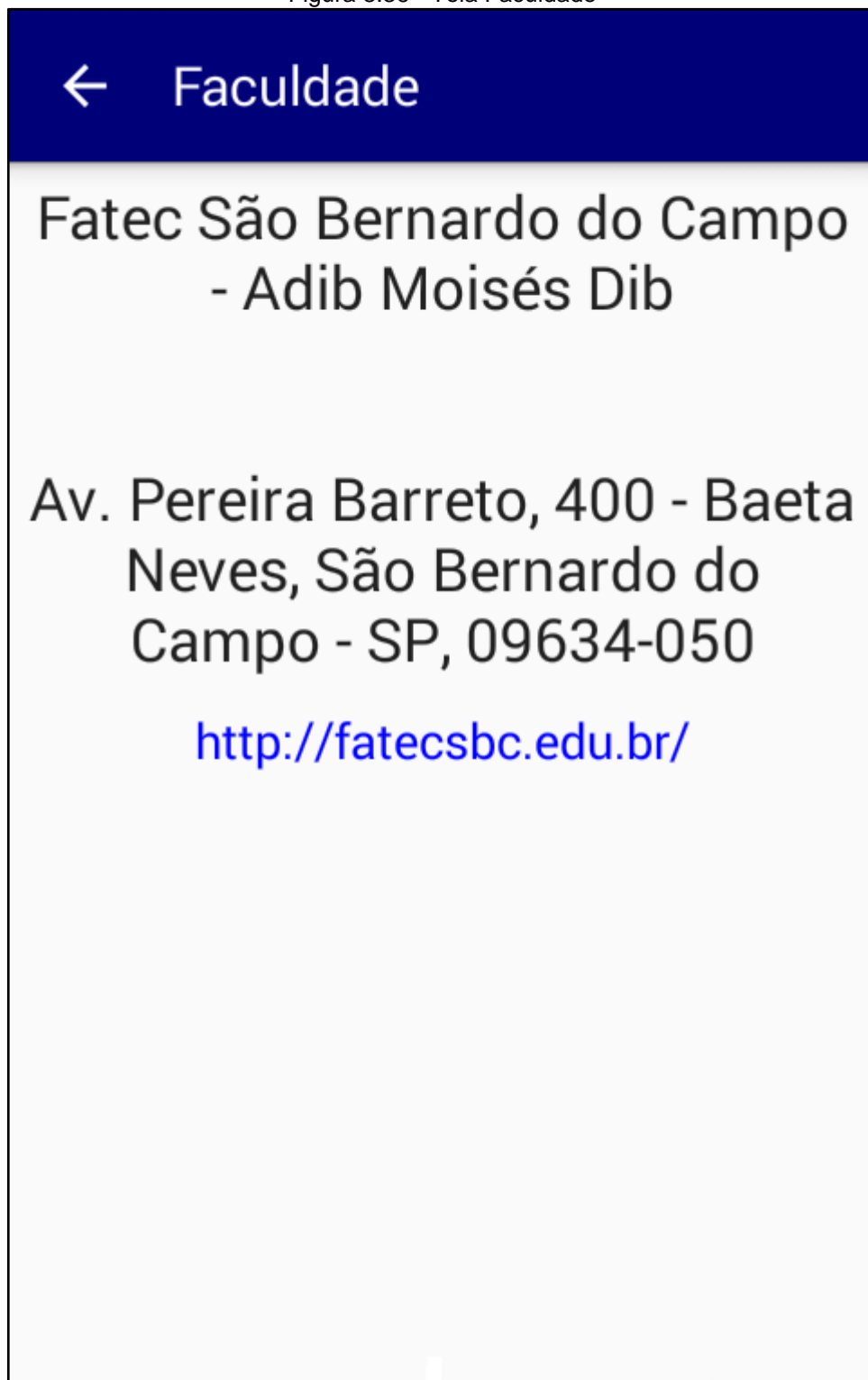


Fonte: Autoria própria, 2017

### **Tela “Faculdade”**

Mostra os principais dados da faculdade, como exibido na Figura 3.36.

Figura 3.36 - Tela Faculdade

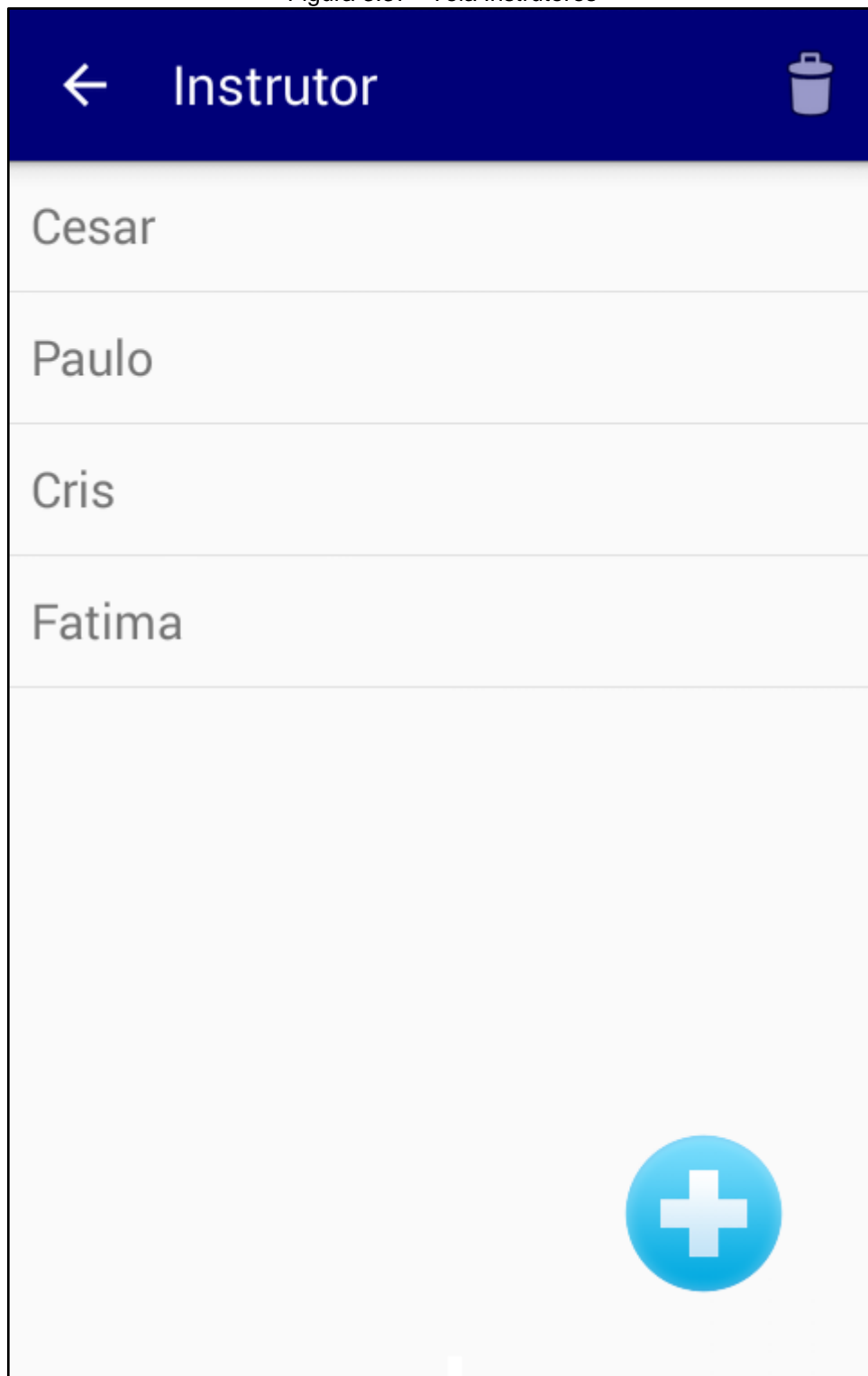


Fonte: Autoria própria, 2017

### **Tela “Instrutores”**

Lista dados de contatos dos instrutores já cadastrados e permite adicionar ou editar novos instrutores. A Figura 3.37 exhibe um exemplo da tela Instrutores.

Figura 3.37 - Tela Instrutores

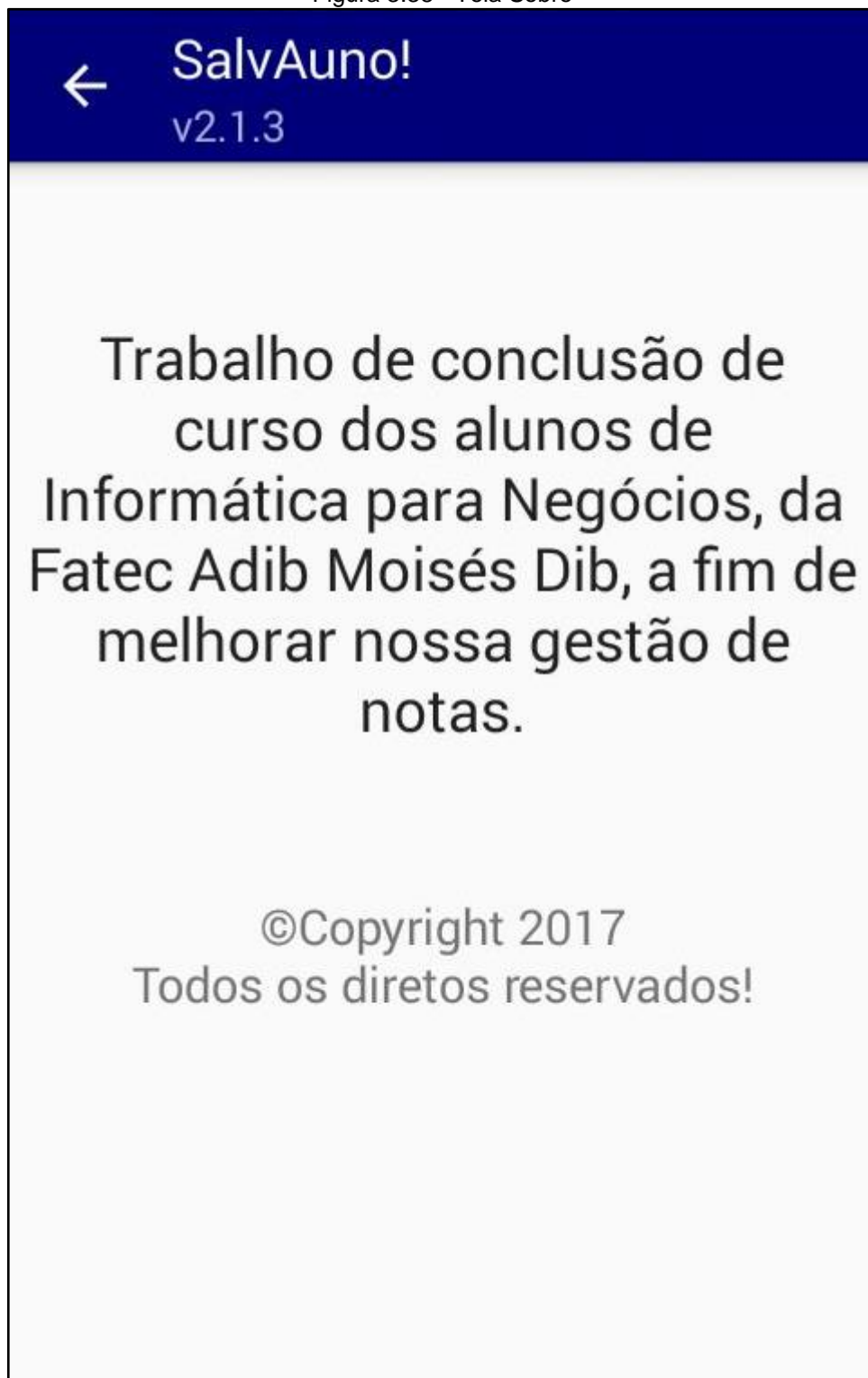


Fonte: Autoria própria, 2017

### Tela “Sobre”

Pequena descrição da existência do aplicativo, como visto na Figura 3.38.

Figura 3.38 - Tela Sobre



Fonte: Autoria própria, 2017

### 3.9 Testes

Quanto aos testes, foram realizados testes para definições de layout, testes de funcionamento e testes para os blocos de código e ambos foram testados individualmente, examinando o funcionamento dos blocos de códigos e da integração entre eles.

Os resultados dos testes decorreram em um bom funcionamento do aplicativo e no levantamento das possibilidades que o sistema pode oferecer em próximas versões. Por possuir alguns componentes mais atuais, o aplicativo só poderá funcionar em sistemas Android acima da versão 3.0.

Os testes executados no desenvolvimento foram efetuados num simulador de Android de API 21 (Android 5.0 Lollipop) e na API 24 (Android 7.0 Nougat), além de aparelhos mobile de versões 4.4, 6.0 e 8.0, apresentando apenas algumas variações de layout e de escrita entre os aparelhos.

## CONSIDERAÇÕES FINAIS

Ao finalizar o projeto, foi possível concluir que a trajetória durante o curso foi essencial para aprimorar a base de conhecimento dos integrantes e a embasar toda a elaboração do *software*, ao unir os conceitos de Tecnologia da Informação e de Administração.

O projeto começou com a ideia de criar um sistema que pudesse ajudar os estudantes a controlar suas notas e faltas na instituição em que frequentam, de maneira simples e prática. Entretanto, com o andamento dele, que sempre prosseguiu em constante melhoria, foi constatado muitas opções de inovação ao funcionamento do sistema.

Agora em sua primeira versão, o *software* apresenta todas as ideias levantadas em seu planejamento, predispondo-o para futuras atualizações e remodelagens de layout e adequando-o, assim, a novos recursos e ferramentas que poderão estar disponíveis na plataforma, além de prepará-lo para trabalhar em outras plataformas disponíveis.

A execução desse projeto de conclusão também permitiu aos integrantes a entender melhor como administrar notas e faltas, com o desenvolvimento de um aplicativo mobile que melhora na comunicação interna e auxilia nos estudos.

Por fim, o SalvAluno! foi pensado, principalmente, com a finalidade de auxiliar os estudantes a otimizar suas tarefas diárias. Com esse aplicativo, espera-se que ele contribua não somente na organização das tarefas do estudante, mas consequentemente, também melhore seu desempenho nos estudos e resolva problemas de comunicação relacionados a informações vinculadas à instituição, imprevistos etc., de forma online, com o mural eletrônico.

## REFERÊNCIAS

- BARBOSA, S. D. J.; SILVA, B. S. da. **Interação Humano-Computador**. Rio de Janeiro: Elsevier, 2011.
- CERVO, A. L.; BERVIAN, P. A. **Metodologia científica**. 6 ed. São Paulo: Prentice Hall, 2007.
- CHIAVENATO, Idalberto. **Introdução à Teoria Geral da Administração**. 3.ed. São Paulo: McGraw Hill, 1983.
- COCKBURN, Alistair. **Escrevendo Casos de Usos Eficazes: Um guia prático para desenvolvedores de software**. Bookman, 2005.
- CONDER, S; DARCEY, L. **Learn Java for Android Development**. MobileTutsPlus, set. 2010. Disponível em: <<https://code.tutsplus.com/tutorials/learn-java-for-android-development-introduction-to-java--mobile-2604>>. Acesso em 31 mar. 2017.
- D'ÁVILA, M. **O Eclipse vai bem, obrigado**. Mdavila, fev. 2005. Disponível em: <[http://mhavila.com.br/topicos/java/eclipse\\_bem.html](http://mhavila.com.br/topicos/java/eclipse_bem.html)>. Acesso em 31 mar 2017.
- DORNELAS, J.C.A. **Empreendedorismo – Transformando Ideias em Negócios**. 6 ed. São Paulo: Atlas, 2016.
- DROPBOX DEVELOPERS BLOG. **API v1 is now deprecated**. Dropbox, jun 2016. Disponível em: <<http://blogs.dropbox.com/developers/2016/06/api-v1-deprecated/>>. Acesso em 20 mar 2017.
- FILION, L.J.; DOLABELLA, F.C.C. **Boa Idéia! E agora?**. São Paulo: Cultura Editores Associados, 2000.
- GIL, A. C. **Como elaborar projetos de pesquisa**. 5. ed. São Paulo: Atlas, 2010.
- GOODRICH, Michael T.; TAMASSIA, Roberto. **Estrutura de Dados e Algoritmos em Java**. 5. ed. Porto Alegre: Bookman, 2013.
- LECHETA, Ricardo R. **Google Android: Aprenda a criar aplicações para dispositivos móveis com o Android SDK**. 2.ed. São Paulo: Novatec Editora, 2010.
- LIAO, K.; LEE, W. (2010). **A novel user authentication scheme based on QR Code**. Journal of networks.
- MAVEN. **Welcome to Apache Maven**. Apache Maven Project, 2017. Disponível em: <<https://maven.apache.org/>>. Acesso em 18 mar 2017.
- OLIVEIRA, D. P. R. de. **Planejamento estratégico: conceitos, metodologia e práticas**. 33. ed. São Paulo: Atlas, 2015.

OPTNER, Stanford L. **Análise de Sistemas Empresariais**. Vol. 3. Rio de Janeiro: Ao Livro Técnico, 1971.

PRESSMAN, Roger S. **Engenharia de software: Uma abordagem profissional**. 7. ed. Porto Alegre: Bookman, 2011.

QRCode.com. **What is a QRCode?**. QRCode.com, 2017. Disponível em: <<http://www.qrcode.com/en/about/>>. Acesso em: 31 mar 2017.

SALIM, César Simões et al. **Construindo planos de negócios: todos os passos necessários para planejar e desenvolver negócios de sucesso**. Rio de Janeiro: Editora Campus Ltda., 2001.

SCHEMBERGER, Elder Elisandro et al. **Plataforma Android**. Disponível em: <[http://www.jornaltech.com.br/wp-content/uploads/2009/09/Artigo\\_Android.pdf](http://www.jornaltech.com.br/wp-content/uploads/2009/09/Artigo_Android.pdf)> Acesso em: 15 mar 2017.

SEBRAE. **Quadro de modelo de negócios: para criar, recriar e inovar**. Sebrae, nov 2017. Disponível em: <<https://www.sebrae.com.br/sites/PortalSebrae/bis/Quadro-de-modelo-de-negocios-para-criar-recriar-e-inovar,a6df0cc7f4217410VgnVCM2000003c74010aRCRD>> Acesso em: 10/11/2017.

SEVERINO, A. J. **Metodologia do Trabalho Científico**. 22 ed. São Paulo: Cortez, 2013.

SOARES, D. **Preparando Ambiente de Desenvolvimento Android para Mac OS X**. Mestre Android, jun. 2016. Disponível em: <<http://www.mestreandroid.com.br/preparando-ambiente-de-desenvolvimento-android-para-mac-os-x/>>. Acesso em: 15 mar 2017.

SOMMERVILLE, Ian. **Engenharia de software**. 9. ed. São Paulo: Addison Wesley, 2011.

SOON, T. J. (2008). **QR Code**. Synthesis jornal.

TASCA, J. E. et al. **An approach for selecting a theoretical framework for the evaluation of training programs**. Journal of European Industrial Training, v. 34, n. 7, p. 631-655, 2010. Disponível em: <<http://www.emeraldinsight.com/doi/abs/10.1108/03090591011070761>>. Acesso em: 15 abr 2017.

ULRICH, Schiefer et al. **MAPA – Manual de Planejamento e Avaliação de Projetos**. 1. ed. Cascais: Principia, 2006.

## GLOSSÁRIO

**Framework** – Biblioteca de classes que suportam uma funcionalidade, mas que baseiam-se em extensão.

**Middleware** – É um programa de computador que faz a mediação entre software e demais aplicações.

**Runtime** – Período em que um programa de computador permanece em execução.

**APÊNDICE – QUESTIONÁRIO SALVALUNO!**

- 1) Qual a sua faixa etária?
- 2) Qual região do Brasil você mora?
- 3) Você estuda atualmente?
- 4) Qual nível escolar você estuda atualmente?
- 5) Você acompanha o sistema de avisos que sua instituição usa (Mural)?
- 6) Dê uma nota para a sua recordação em fazer trabalhos.
- 7) Você possui algum controle de faltas?
- 8) Possui dados de contato de seu instrutor com fácil acesso?
- 9) Defina uma nota a comunicação sobre eventos, faltas de professores, etc.
- 10) O que você diz sobre um aplicativo que controla suas notas e faltas na faculdade?