



UNIVERSIDADE FEDERAL DO CEARÁ  
CAMPUS QUIXADÁ  
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

**RONALDO NERY DANTAS FILHO**

**ROUTINE: FERRAMENTA DE AUXÍLIO AO CONTROLE DE  
PRESENÇA EM SALA DE AULA BASEADA EM DISPOSITIVOS  
MÓVEIS**

**QUIXADÁ  
2013**

**RONALDO NERY DANTAS FILHO**

**ROUTINE: FERRAMENTA DE AUXÍLIO AO CONTROLE DE  
PRESENÇA EM SALA DE AULA BASEADA EM DISPOSITIVOS  
MÓVEIS**

Trabalho de Conclusão de Curso submetido à Coordenação do Curso Bacharelado em Sistemas de Informação da Universidade Federal do Ceará como requisito parcial para obtenção do grau de Bacharel.

Área de concentração: Computação

Orientador Prof. Ricardo Reis Pereira

**QUIXADÁ  
2013**

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Biblioteca do Campus de Quixadá

---

D21r Dantas Filho, Ronaldo Nery  
Routine: ferramenta de auxílio ao controle de presença em sala de aula baseada em dispositivos  
móveis / Ronaldo Nery Dantas Filho. – 2013.  
54 f. : il. color., enc. ; 30 cm.

Monografia (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de  
Sistemas de Informação, Quixadá, 2013.  
Orientação: Prof. Me. Ricardo Reis Pereira  
Área de concentração: Computação

1. Software- desenvolvimento 2. Computação móvel 3. Engenharia de Software I. Título.

**RONALDO NERY DANTAS FILHO**

**ROUTINE: FERRAMENTA DE AUXÍLIO AO CONTROLE DE PRESENÇA EM  
SALA DE AULA BASEADA EM DISPOSITIVOS MÓVEIS**

Trabalho de Conclusão de Curso submetido à Coordenação do Curso Bacharelado em Sistemas de Informação da Universidade Federal do Ceará como requisito parcial para obtenção do grau de Bacharel.

Área de concentração: computação

Aprovado em: \_\_\_\_\_ / dezembro / 2013.

**BANCA EXAMINADORA**

---

Prof. MSc. Ricardo Reis Pereira (Orientador)  
Universidade Federal do Ceará-UFC

---

Prof. Dr. David Sena Oliveira  
Universidade Federal do Ceará-UFC

---

Prof. MSc. Regis Pires Magalhães  
Universidade Federal do Ceará-UFC

Dedico esse trabalho à minha família...

## AGRADECIMENTOS

Á Deus por sempre ter me guiado e me dado força durante todos os momentos da minha vida.

À meus pais Ronaldo e Adriana, que me ensinaram e ainda me guiam na forma de pensar e de agir com as batalhas do dia-a-dia. Por terem me ensinado a agir como uma boa pessoa e a saber diferenciar as coisas da vida. Por terem me apoiado em todos os momentos de todas as maneiras possíveis. Por me darem orgulho a ponto de me servir como exemplos de vida e por serem minha meta como pessoa.

Á minha irmã Luana e meu tio Antonio Filho, que sempre souberam muito bem me alegrar nos momentos em que preocupações me dominavam.

Á minha família tanto paterna quanto materna, que sempre foram sagradas para mim e sempre me incentivaram e me trouxeram momentos de felicidade.

Ao meu orientador, Prof. Msc. Ricardo Reis Pereira, pela relação não só de professor e orientador, mas de um grande amigo. Por ter me sugerido este trabalho e ter tido tempo e paciência suficiente para o auxílio no desenvolvimento. Por ter me proporcionado uma considerável sabedoria através de seu conhecimento no decorrer do curso e por ter me dado incentivo e oportunidade para estudar uma tecnologia que me levou ao meu principal objetivo no momento: mercado de trabalho.

À professora Ticiano Linhares, pelo auxílio oferecido no início desse projeto quanto a questão do banco de dados. Foi de grande valia tanto para mim quanto para meu orientador.

Ao professor David Sena, pelas ótimas orientações que foram dadas justamente em um momento cheio de dúvidas em que eu me encontrava.

A minha namorada Priscila Brito, que está comigo desde o primeiro semestre da faculdade e soube me acompanhar e me motivar bastante em momentos difíceis na construção desse trabalho.

A meu amigo Elivando Amorim, que dividiu o primeiro apartamento em que morei em Quixadá. Foi uma grande pessoa por ter fornecido esse apoio e merece todo meu agradecimento.

A meu amigo Siqueira, que me apresentou o Django e o Python que foram cruciais para o desenvolvimento deste trabalho. Por ter me guiado em vários tropeços durante o aprendizado destas tecnologias.

Aos grandes amigos-irmãos Bruno, Higo, Siqueira e Italo que dividiram comigo esses anos de faculdade um apartamento pequeno demais para tanta alegria.

Aos grandes amigos-irmãos de turma Mardson, Paulo Ivo e Alessandro que compartilharam 8 semestres de estudo acompanhado de muitos momentos de felicidade.

A todas as pessoas que de alguma forma contribuíram para a elaboração deste trabalho.

"Leve com você só o que foi bom,  
ódio e rancor não dão em nada."  
(Natiruts)

## RESUMO

Este trabalho contém todos os procedimentos realizados para o desenvolvimento do Routine. O Routine é uma aplicação para dispositivo móvel que visa auxiliar o corpo docente em sala de aula no controle da atividade rotineira de controle de frequência dos alunos. Para isso foi-se necessário descrever a atividade de forma detalhada, levantar os requisitos, construir tanto o servidor web como o aplicativo móvel, testar de acordo com o que foi documentado e implantar a aplicação em sala de aula. Os conceitos da engenharia de software juntamente com os benefícios dos dispositivos móveis serviram de base teórica para esse trabalho. Tecnologias como Python, Django, C++, Qt e QtQuick foram responsáveis pelo desenvolvimento da aplicação. O professor é principal beneficiado uma vez que o aplicativo busca auxiliar uma atividade pertencente a ele. Foco em integração de outras funcionalidades relacionadas às atividades rotineiras e uma maior preocupação em consistência de dados são soluções para trabalhos futuros advindos deste.

Palavras chave: Software. Dispositivos móveis. Controle de frequência.

## **ABSTRACT**

This work contains all the procedures performed for the development of the Routine. The Routine is an application for mobile which aims to assist the faculty in the classroom in the control of routine activity frequency control students. To this was it necessary to describe the activity in detail, raising the requirements, build both the web server and the mobile application testing in accordance with what has been documented and deploy the application in the classroom. The concepts of software engineering along with the benefits of mobile devices served as the theoretical basis for this work. Technologies like Python, Django, C++ , Qt and QtQuick were responsible for developing the application. The teacher is the main beneficiary once the application seeks to assist an activity belonging to him. Focus on integration of other features related to daily activities and a greater focus on data consistency are solutions to this coming future work.

Keywords: Software. Mobile device. Frequency control.

## LISTA DE ILUSTRAÇÕES

Figura 1 - Etapas realizadas no desenvolvimento .....	22
Figura 2 - Relação entre Docente, Disciplina e Oferta.....	24
Figura 3 - Relação entre Oferta, Matrícula e Aluno .....	25
Figura 4 - Relação entre Aluno, Aula e Presença.....	26
Figura 5 - Tela de Login - Web .....	28
Figura 6 - Tela de Entidades.....	29
Figura 7 - Tela de Cadastro de Disciplina .....	29
Figura 8- Tela de Disciplinas c/ Edição e Remoção.....	29
Figura 9 - Cadastro de uma Oferta .....	30
Figura 10 - Tela de Login - Móvel .....	31
Figura 11 - Tela de Ofertas.....	31
Figura 12 - Telas de Aulas e Alunos de Cálculo I.....	32
Figura 13 - Telas de Adicionar Aula c/ Dialog .....	33
Figura 14 - Telas de Nova Aula c/ Alunos Presentes e Faltantes.....	34
Figura 15 - Tela de Aulas Atualizada.....	34

## SUMÁRIO

1 INTRODUÇÃO.....	15
2 REVISÃO BIBLIOGRÁFICA.....	17
2.1 Engenharia de Software.....	17
2.2 Dispositivos Móveis.....	20
3 RELATO GERAL DO DESENVOLVIMENTO.....	22
3.1 Introdução.....	22
3.2 Etapas Realizadas.....	22
4 DESENVOLVIMENTO.....	24
4.1 Projeto da Aplicação.....	24
4.2 Construção da Aplicação.....	26
4.2.1 Ambiente de Utilização.....	26
4.2.2 Tecnologias Utilizadas.....	27
4.2.3 Arquitetura da Aplicação.....	28
a) Modelo de dados do Servidor.....	28
b) Cliente na Web.....	28
c) Cliente no Dispositivo Móvel.....	30
d) Relação Cliente-Servidor.....	35
e) Serviços.....	36
5 VALIDAÇÃO.....	38
6 CONCLUSÃO.....	40
6.1 Objetivos Alcançados.....	40
6.2 Projetos Futuros.....	41
7 RELATO DE EXPERIÊNCIA.....	42
REFERÊNCIAS.....	44
APÊNDICES.....	45
APÊNDICE A – Documento de Modelagem do Banco de Dados.....	45
APÊNDICE B – Diagrama de Casos de Uso.....	46
APÊNDICE C – Documento de Requisitos do Routine.....	46

## 1 INTRODUÇÃO

O crescente e notável avanço da tecnologia da informação tem sido aproveitado em vários projetos de diferentes contextos. A internet tem propiciado esse avanço gerando produção constante de novos sistemas e proporcionando maior eficiência nas atividades do dia-a-dia. Em sala de aula, por exemplo, o controle de presença saiu do tradicional "papel e caneta" e passou a ser realizado com o auxílio de softwares. Isso proporcionou melhor gerenciamento dos dados permitindo as instituições de ensino tomadas de decisão mais precisas.

As atividades de sala de aula passaram a ser realizadas de maneira informatizada. Os professores passaram a utilizar e depender de dispositivos que auxiliavam essa automatização. Conforme pesquisa realizada pelo IAB Brasil (2012), o computador desktop lidera como o tipo de canal mais utilizado. Entretanto, o computador desktop não permite ao professor mobilidade suficiente. O uso de computadores móveis como notebooks e *tablets* muda essa realidade.

A mobilidade e praticidade de uso tem atraído um grande número de usuários aos dispositivos móveis. Dados do IBOPE Media (2013) destacam que 134 milhões de pessoas têm acesso a telefone móvel no Brasil, sendo 54 milhões com acesso à internet pelo celular.

Diversos trabalhos têm sido publicados sobre a vantagem da utilização dos dispositivos móveis. No trabalho de Galvin (2004), é apresentada uma aplicação para dispositivos móveis que visa auxiliar no relacionamento entre clientes e organizações pertencentes ao ramo de vendas. Sibaldo et al. (2007), desenvolveram uma aplicação para dispositivos móveis que consiste de um sistema de recomendação para encontrar potenciais usuários que ajudem no processo de aprendizagem de alunos.

O controle de frequência de alunos em sala de aula é uma prática comum em escolas e universidades. Permite acompanhar a assiduidade de cada estudante além de apresentar um critério de avaliação. Este trabalho trata de uma aplicação para dispositivos móveis (que denominamos Routine). Seu foco é permitir melhor acessibilidade, usabilidade e mobilidade no controle de listas de frequência dos alunos.

O professor é diretamente beneficiado, pois tem o poder de controlar mais dinamicamente a evasão de alunos e conseqüentemente estabelecer relações entre estes resultados e o rendimento nas avaliações.

O Routine junta benefícios providos por sistemas web, com significantes vantagens dos dispositivos móveis. Ele busca resolver problemas relacionados com infraestrutura (peso, conectividade) ou com limitações do corpo docente (questão de deficiências).

A aplicação torna possível aproveitar melhor o tempo da aula na transmissão de informação, além de prover ao professor à portabilidade de informações referentes à assiduidade de seus alunos.

O objetivo geral deste trabalho é desenvolver uma aplicação para dispositivo móvel, que auxilie o professor em sala de aula na atividade de controle de frequência de alunos. Para isso, seis objetivos específicos são necessários:

- a) Elaborar a arquitetura da aplicação.
- b) Levantar os requisitos do sistema.
- c) Desenvolver a aplicação web (serviços e sistema).
- d) Desenvolver o aplicativo android.
- e) Executar testes na aplicação para verificar a conformidade com os requisitos.
- f) Implantação da aplicação em uma aula real.

As próximas seções estão divididas em seis etapas. Na *revisão bibliográfica* são apresentados conceitos relevantes sobre engenharia de software e dispositivos móveis. No *relato geral sobre o desenvolvimento*, é descrita a forma geral de todas as fases realizadas durante a construção da aplicação. No *desenvolvimento* são detalhadas as fases da etapa anterior. A *validação* apresenta uma contextualização da aplicação. Seções de *considerações finais e referências* são apresentadas no final do trabalho.

## 2 REVISÃO BIBLIOGRÁFICA

### 2.1 Engenharia de Software

Uma proposição realizada por Fritz Bauer no livro de Naur e Randall (1969) define engenharia de software como: a criação e a utilização de sólidos princípios de engenharia a fim de obter softwares econômicos que sejam confiáveis e que trabalhem eficientemente em máquinas reais. Com o passar do tempo, vários novos conceitos foram sendo abordados, porém sem nunca sair do foco principal: uso dos aspectos de engenharia para construir sistemas eficientes para a realidade.

Já Sommerville (2011) conceitua engenharia de software como uma disciplina de engenharia cujo foco está em todos os aspectos da produção de software, desde os estágios iniciais da especificação do sistema até sua manutenção, quando o sistema já está sendo usado. Para obter alta qualidade em sistemas, engenheiros de software adotam abordagens sistemáticas pela eficiência que isso causa. Porém existe um hábito pertencente a engenharia que sempre é praticado: a atitude de selecionar o método mais adequado para um conjunto de circunstâncias. Ou seja, ter o costume de agir com criatividade e com uma formalidade razoável pode ser bem mais eficiente em alguns casos.

Quando se trata de aspectos da produção de software ou de abordagem sistemática, temos os processos de software. Sommerville (2011) define processos de software como um conjunto de atividades relacionadas que levam à produção de um produto de software. Existem vários processos de software distintos, porém quatro atividades fundamentais para a engenharia de software devem estar inclusas no processo: especificação de software, projeto e implementação de software, validação de software e evolução do software.

Na atividade de especificação de software ou também chamada de engenharia de requisitos, Pádua (2009) diz que é empregado um conjunto de técnicas para levantar, detalhar, documentar e validar os requisitos de um produto. Já Pressman (2010) define engenharia de requisitos como um auxílio para os engenheiros de software na melhor compreensão do problema do cliente que eles vão trabalhar para resolver. Esta é uma etapa extremamente importante visto que projetar e construir um programa de computador elegante que resolva o problema errado não serve às necessidades de ninguém.

A etapa de projeto e implementação de software, como o nome já diz, envolve duas partes que estão presentes no processo de desenvolvimento do software: análise e projeto; e implementação de software. A etapa de projeto antecede a de implementação, por esse motivo

ela caracteriza-se principalmente por fornecer uma descrição da estrutura do software a ser implementado. Sommerville (2011) define essa etapa como o processo de conversão de uma especificação do sistema em um sistema executável.

Conforme o sistema vai sendo projetado e construído, é necessário que ele seja verificado e validado. A validação de software é a fase onde o software é analisado se está seguindo e se adequando as especificações documentadas no mesmo momento em que satisfaz as especificações do cliente. Teste é o processo principal da técnica de validação. Como dito por Pádua (2009), o processo de testes consiste em operar um sistema ou componente sob condições especificadas, observando ou registrando os resultados e avaliando algum aspecto desse sistema ou componente. Um teste bem feito pode influenciar de forma positiva na qualidade do software. Com isso, alguns atributos qualitativos podem ser conquistados e isso é importante para qualquer tipo de software. Atributos esses que são: funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade.

Com um software validado, ele já pode entrar em estado de manutenção. A fase de evolução ou manutenção do software é onde todas as necessidades existentes no projeto são mantidas e gerenciadas. Isso permite o controle da flexibilidade dos sistemas de software de forma eficiente. Segundo Sommerville (2011), essa flexibilidade é uma das principais razões pelas quais a prática de manutenção vêm sendo, cada vez mais, incorporados em sistemas grandes e complexos. Mudanças no software podem ser feitas a qualquer momento durante ou após o desenvolvimento do sistema e custos de manutenção são frequentemente mais altos que os custos iniciais de desenvolvimento. Logo, é mais realista ver a engenharia de software como um processo evolutivo, onde o software é alterado em todo seu ciclo de vida de acordo com as necessidades do cliente e das alterações dos requisitos.

O desenvolvimento do Routine passou por cada fase citada anteriormente, visando obter todos os benefícios provenientes do bom uso de suas técnicas e metodologias de aplicação. O modelo utilizado na construção da aplicação seguiu as características do modelo iterativo e incremental, visto que a realidade do ambiente de desenvolvimento encaixava-se mais com essa metodologia. Uma justificativa dessa escolha é o fato de que esse modelo permite a participação e comunicação entre o desenvolvedor e o usuário. Outro benefício é que os riscos podem ser administrados de maneira melhor por pedaços do sistema a serem desenvolvidos em pequenos espaços de tempo. Com o modelo iterativo, alterações nos requisitos podem ser rapidamente incorporada no processo de desenvolvimento.

Os conceitos da engenharia de software foram essenciais para esse projeto. Pois ajudaram a evitar erros e aceleraram o desenvolvimento. A forma com a qual Sommerville (2011) conceituou engenharia de software foi de grande contribuição pois segundo o mesmo, o foco deve estar os estágios iniciais da especificação do sistema até as etapas finais da construção.

## 2.2 Dispositivos Móveis

Uma enorme quantidade de dados e de informação tem sido transmitida constantemente desde a popularização de equipamentos com processamento de informações. Como Lee (2005) comenta, esses equipamentos começaram em caixa eletrônicos de bancos, entraram no acesso ao povo através de computadores pessoais, notebooks e, posteriormente, foi a vez dos dispositivos móveis.

Existe uma grande atenção voltada para os dispositivos móveis. Existem empresas desenvolvendo com o hardware e, paralelamente e em maior quantidade, pessoas focadas no desenvolvimento de software para esse tipo de equipamento. Quando se trata em hardware, temos que esses dispositivos estão cada vez menores e com um maior poder de processamento e funcionalidades vistas até então somente em computadores.

No foco em software para equipamentos móveis, vale considerar o quanto eles ajudam na questão da mobilidade. Lee (2005) define esta mobilidade como sendo o uso pelas pessoas de dispositivos móveis portáteis funcionalmente poderosos que ofereçam a capacidade de realizar facilmente um conjunto de funcionalidades, sendo também capazes de conectar-se, obter dados e fornecê-los a outros usuários, aplicações e sistemas.

No contexto do desenvolvimento de software para dispositivos móveis é possível notar inúmeros benefícios que a utilização desses dispositivos proporcione em vários ambientes. Com vários sistemas Web já automatizando e ajudando a vida das pessoas, considere funcionalidades específicas desses sistemas em uma aplicação instalada em um smartphone pronto para ser utilizado. Isso aproveitaria os benefícios já provenientes do sistema Web e acrescentaria outras consideráveis vantagens particulares dos dispositivos móveis, como: mobilidade, usabilidade, economia de tempo, independência temporária de conexão, além de facilitar a organização pessoal. Existem diversos exemplos de dispositivos móveis, como assistentes pessoais digitais, celulares e tablets. Sendo esses dois últimos os mais usados nos últimos tempos.

Vários profissionais podem possuir um ganho significativo na produtividade ao utilizar esses dispositivos, tais profissionais são citados por Lee (2005):

- Profissionais que necessitam ficar sempre em trânsito remoto para a execução de suas atividades;
- Agentes de Vendas;
- Agentes de Serviços;

- Consultores;
- Profissionais em trânsito local;
- Alguns profissionais de escritório;
- Funcionários que trabalham em estoque;
- Profissionais de saúde no geral;

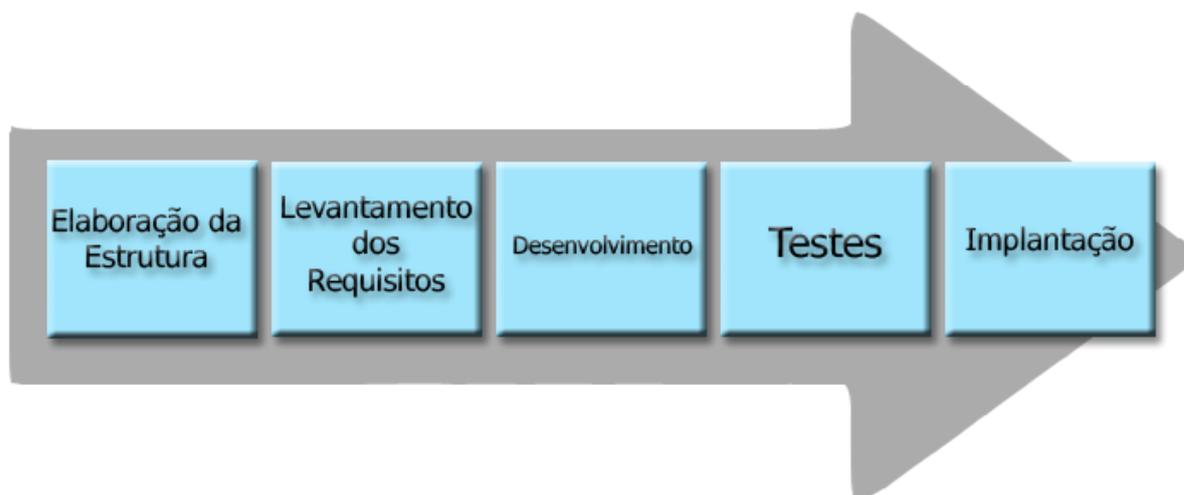
Além de profissionais da Educação, onde a utilização do dispositivo móvel pode influenciar no aprendizado dos alunos.

Ao considerar os benefícios provenientes da utilização de aplicações móveis em diferentes âmbitos, pode-se tornar possível aproveitá-los também nas universidades e nas escolas. Na escola, professores e alunos necessitam de uma organização de atividades. Quando se trata de professores, a necessidade chega a ser ainda mais específica por suas atividades dentro da instituição serem mais fixas e fazerem parte da rotina. Com isso, o Routine aproveita todos esses benefícios da utilização dos dispositivos móveis e os aplicará no âmbito acadêmico, principalmente quando se trata da usabilidade. A interface do Routine foi projetada com o objetivo de tornar a atividade a ser executada mais usual e prática. Outro significativo ganho é que os componentes na aplicação possuem características específicas, que resolvem problemas ligados à limitações do usuário. Limitações essas que estão diretamente ligadas a deficiência física (visual, principalmente). Logo, o aplicativo possui características de interface que buscam resolver isso, como: cores agradáveis, fontes e botões bem visíveis, dentre outros. Tamanho, maneira como são executados, dados mais acessíveis, informações simples e bem visíveis são algumas das principais características.

Os trabalhos de Galvin (2004) e Sibaldo et al. (2007) contribuíram bastante pois ambos provaram que o uso dos dispositivos móveis em diferentes ambientes pode proporcionar consideráveis valores.

### 3 RELATO GERAL DO DESENVOLVIMENTO

Figura 1 - Etapas realizadas no desenvolvimento



Fonte: autor.

#### 3.1 Introdução

Esta seção destaca as etapas (Figura 1) que foram cumpridas para a realização do trabalho. A primeira etapa foi a atividade de projeto, onde as entidades envolvidas na atividade foram descritas e as funcionalidades foram projetadas junto da estrutura do aplicativo. Na segunda etapa e terceira etapa foram implementados o servidor web e a aplicação móvel de acordo com o projeto definido. Na quarta etapa, as funcionalidades documentadas foram testadas e alteradas quando necessário. Na última etapa, a aplicação foi implantada em sala de aula.

#### 3.2 Etapas Realizadas

A descrição da atividade de frequência em sala de aula foi realizada com um docente. Isto permitiu melhor compreensão da estrutura do aplicativo e, conseqüentemente, do seu desenvolvimento. Com a descrição da atividade, tornou-se possível o levantamento dos requisitos funcionais. Nesse momento, casos de uso foram feitos e o diagrama pode ser visto no apêndice B. O desenvolvimento da aplicação pôde ser realizado a partir da versão pronta e bem definida da documentação dos requisitos que está no apêndice C.

Na **etapa de projeto** as funcionalidades descritas no documento de requisitos foram analisadas individualmente. A partir dessa análise o “esqueleto” da aplicação foi construído. Nessa fase foi projetado como o aplicativo exibirá suas funções e de que forma elas seriam

desenvolvidas. Essa etapa teve como enfoque tornar o projeto mais simples para o desenvolvedor e incentivar na eficiência do produto. Um software prático tanto para o desenvolvedor quanto para o usuário final foi consequência desse procedimento. O detalhamento desta atividade junto da descrição do controle de frequência encontram-se na seção 4.1.

Após a criação da primeira versão da documentação de análise e projeto, **a construção do sistema** foi iniciada. O processo de desenvolvimento do sistema foi dividido em duas partes: construção do lado web e do aplicativo móvel. Cada parte foi desenvolvida com tecnologias distintas, mas com capacidade de interação entre as mesmas. O detalhamento desta atividade encontra-se na seção 4.2.

A parte web do sistema provê o controle das informações do sistema (cadastro de alunos, disciplinas e etc.). A linguagem de programação Python juntamente com o framework web Django foram utilizados para o desenvolvimento do lado web da aplicação. A justificativa da escolha foi simplicidade e rapidez de desenvolvimento que as tecnologias proporcionam.

O lado móvel da aplicação foi implementado em linguagem C++ utilizando o framework Qt para dispositivos com o sistema operacional Android. O desenvolvimento da aplicação teve como principal preocupação a usabilidade, praticidade e agilidade.

Na fase de testes, os componentes da etapa de desenvolvimento foram testados e verificados. Com isso, as principais funcionalidades foram testadas e alteradas conforme necessário. Isso permitiu conferir se os requisitos destacados na etapa inicial estavam sendo atendidos. O detalhamento da arquitetura da aplicação após os testes encontra-se na seção 4.2.3.

Após a aplicação ser documentada, projetada, desenvolvida e testada, **ela foi implantada** em um dispositivo para validação. Onde demonstrou boa iteratividade com a aplicação do lado web. O detalhamento desta atividade encontra-se na seção 5.

## 4 DESENVOLVIMENTO

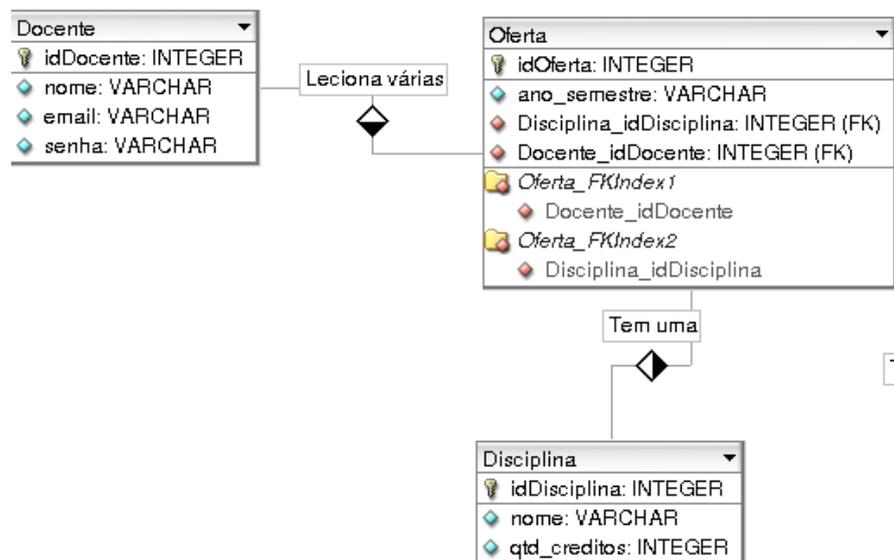
Nesta seção, as etapas citadas no relato geral do desenvolvimento são descritas de forma detalhada. As subseções detalham: as informações da atividade, a forma com a qual os requisitos do Routine foi definido, a estrutura da construção da aplicação em si, os testes utilizados e por fim a aplicação do sistema em sala de aula.

### 4.1 Projeto da Aplicação

O controle de frequência consiste no registro das presenças individuais de alunos no contexto de uma disciplina, seja numa escola ou universidade. A implementação do controle de frequência pelo Routine envolve as seguintes entidades:

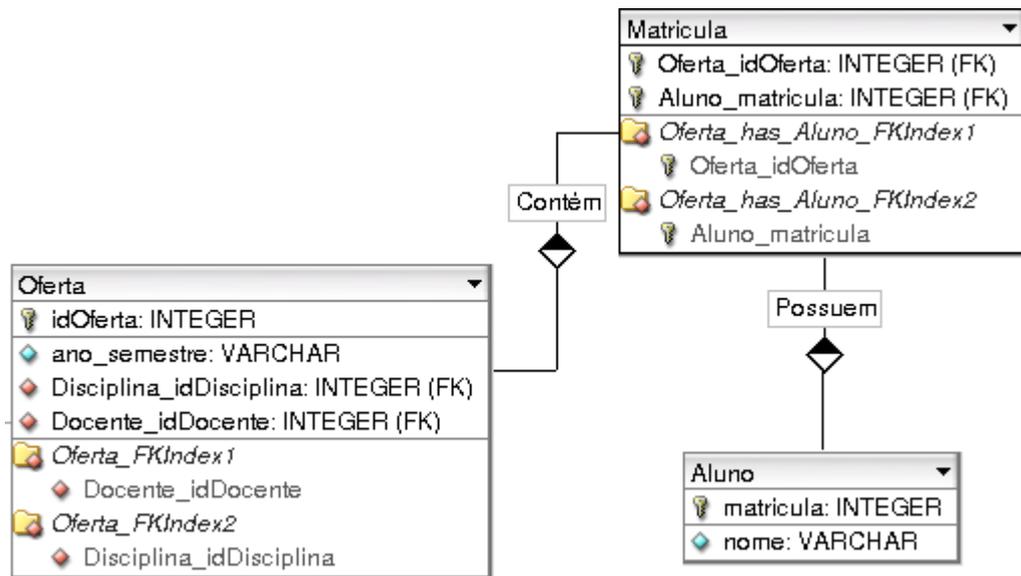
- A primeira é a **disciplina**, que corresponde a um curso de conteúdo programático definido a ser ministrado.
- A segunda entidade é o **docente**, responsável por ministrar as disciplinas e efetuar o controle de frequência. Um docente leciona uma ou mais **disciplinas** durante um período. Uma disciplina pode ser ofertada semestralmente ou anualmente e por docentes diferentes.
- Uma **oferta** corresponde a associação entre um docente, uma disciplina e o período no qual ela ocorre. Várias ofertas de uma mesma disciplina podem estar associadas ao mesmo docente. A relação entre docente, disciplina e oferta pode ser vista na Figura 2.

Figura 2 - Relação entre Docente, Disciplina e Oferta



- Um **aluno** representa um indivíduo que pode inscrever-se em uma ou mais ofertas através de uma matrícula.
- A **matrícula** trata-se da associação entre um aluno e uma oferta existente. Estão associadas a uma mesma oferta um grupo de alunos (turma) e o conjunto das aulas ministradas pelo professor envolvido. A relação entre oferta, aluno e matrícula pode ser vista na Figura 3

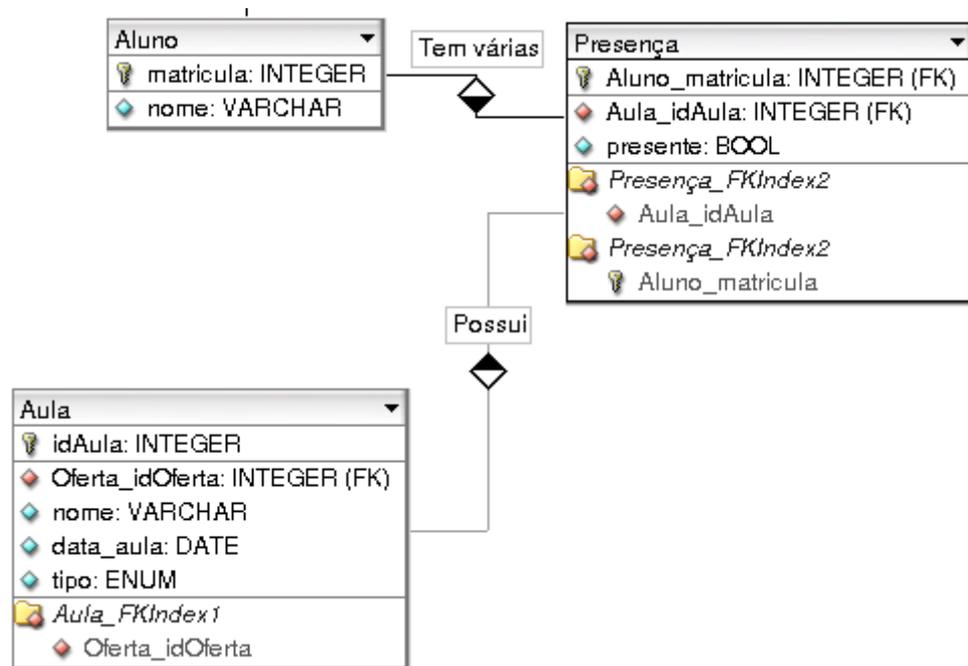
Figura 3 - Relação entre Oferta, Matrícula e Aluno



Fonte: autor

- A entidade **aula**, que representa a unidade de conteúdo de uma disciplina mantém informações sobre a data, hora, tipo (normal, reposição ou feriado) e uma descrição (assunto da aula) do conteúdo ministrado.
- As aulas devem ser cadastradas pelo docente juntamente com as **presenças** dos seus alunos. Cada presença refere-se à uma aula, ou seja, em casos de aulas seguidas de uma mesma disciplina no turno (horário ABCD) são contabilizadas apenas duas presenças. A relação entre aluno, aula e presenças pode ser vista na Figura 4.

Figura 4 - Relação entre Aluno, Aula e Presença



Fonte: autor

Após a definição de entidades, o banco de dados foi modelado e construído. A modelagem completa desses dados segue como apêndice A. A interface gráfica foi definida baseada em layouts utilizados em diários escolares e em padrões webs usuais. Serviços web foram implementados para fornecer os dados requisitados pela aplicação. Tanto o lado servidor quanto o lado cliente foram implementados em tecnologias que trouxeram vantagens significativas tanto para o usuário final quanto para o desenvolvedor.

## 4.2 Construção da Aplicação

Esta seção está dividida nas subseções seguintes: ambiente de utilização, onde é descrito o necessário para a execução da aplicação; tecnologias utilizadas, onde são destacadas as tecnologias empregadas e a motivação para utilizá-las; e a arquitetura da aplicação, onde está descrito a estrutura do aplicativo (modelagem dos dados, interface do cliente e do servidor e descrição dos serviços fornecidos pelo lado web).

### 4.2.1 Ambiente de Utilização

Um ambiente mínimo para a execução estável do Routine inclui: sala de aula, professor, grupo de alunos (um ou mais), contexto de disciplina, rede (internet ou intranet) e um dispositivo móvel que conecte-se a rede.

A **sala de aula** é representada por uma infraestrutura mínima que permita a execução de aulas por um professor perante um grupo de alunos. Note que isto tanto pode corresponder a um espaço clássico (escolas, universidades, etc.) quanto a ambientes virtuais (videoconferência ou *pod-casts*).

O **professor** é o usuário da aplicação utilizando tanto o lado cliente quanto o lado web. O lado cliente é utilizado em sala de aula no dispositivo móvel para exclusivo controle de frequência. No lado web, o professor pode gerenciar inscrições de alunos e ofertas de disciplinas.

O **aluno** é um indivíduo que frequenta a sala de aula para assistir aulas ministradas pelo professor. A ausência do aluno implica em registro de faltas na aplicação e faltas podem implicar em reprovação, conforme os critérios estabelecidos pela instituição de ensino a qual o professor presta serviço.

O **contexto de disciplina** é representado pela relação que existe entre a instituição de ensino e o aluno, tendo o professor como executor. Nesta relação, o aluno possui obrigações com a instituição no que diz respeito a presença e rendimento escolar.

A **rede** é um meio de comunicação entre a aplicação web e a aplicação no dispositivo móvel. Pode corresponder a internet ou a uma intranet onde o lado web está instalado no servidor.

Os **dispositivos móveis** podem ser *smartphones* ou *tablets* com o sistema operacional Android e acesso a rede sem fio.

#### 4.2.2 Tecnologias Utilizadas

No lado web da aplicação, a linguagem de programação Python juntamente com o framework Django foram utilizados no desenvolvimento devido a simplicidade de uso e o rápido desenvolvimento que propiciam. Django ainda disponibiliza para o usuário uma área administrativa para fins de leitura e entrada de dados. Tais dados são acessados via JSON.

No lado cliente (Android), foi utilizado o framework Qt (C++) por possuir diversos módulos especializados incluindo rede, banco de dados, *multithreading*, interface gráfica, dentre outros. Dois módulos do framework Qt foram fundamentais na realização deste trabalho: QNetwork e QtQuick. QNetwork permitiu que a comunicação entre os dois lados da aplicação fosse possível. O módulo QtQuick permitiu a construção de uma interface iterativa baseada em mudança de estados graduais que favorecem a usabilidade.

### 4.2.3 Arquitetura da Aplicação

Esta seção trata da representação de dados em ambos os lados da aplicação e da forma como estes dados são enviados e recebidos pela rede.

#### a) Modelo de dados do Servidor

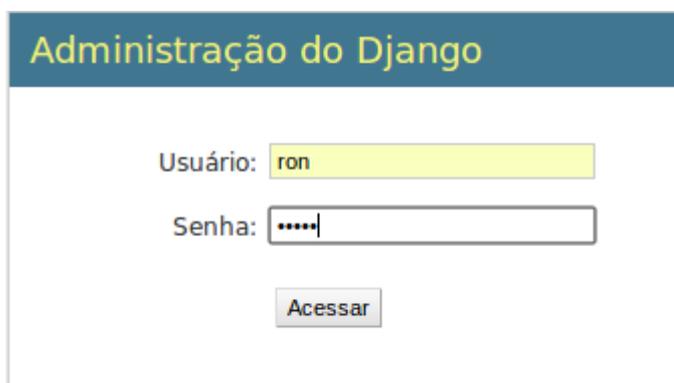
Os dados existentes no servidor estão modelados em sete entidades e um conjunto de relações entre elas. As entidades são: docente, disciplina, oferta, aula, matrícula, aluno e presença. Docente compõe-se do nome do professor junto do e-mail e senha (informações de acesso ao sistema). A entidade Disciplina contém um nome e quantidade de créditos. Existe uma relação muitos-para-muitos (*many-to-many*) entre Docente e Disciplina representada pela entidade Oferta que mantém referência entre um docente, uma disciplina e o ano/semestre de execução. Também existe uma relação muitos-para-muitos entre Oferta e Aluno representada pela entidade Matrícula que mantém referência entre uma oferta e um aluno. A entidade Aula compõe-se de uma referência a uma oferta, um assunto tratado, a data de realização e o tipo, que pode ser normal (segue calendário acadêmico), ou reposição (aula marcada fora do dia considerado normal), ou do tipo feriado.

#### b) Cliente na Web

Todas as entidades são editáveis no lado web da aplicação. Vale ressaltar que é mais conveniente utilizá-la na entrada de dados não dinâmicos (fornecidos uma vez por semestre como disciplinas, ofertas e alunos)

O cadastro via web de qualquer dado pode ser feito a partir da permissão de acesso ao admin (é necessário que o usuário faça o login). A Figura 5 contém a tela de login.

Figura 5 - Tela de Login - Web



Administração do Django

Usuário: ron

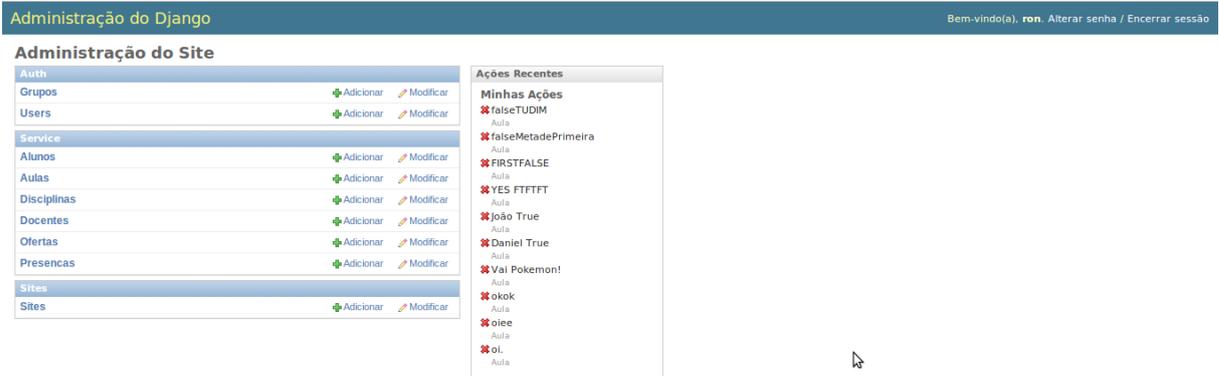
Senha: .....

Acessar

Fonte: autor.

Com a validação do usuário concluída, o cliente possui acesso às entidades existentes e a opção de cadastro das mesmas pode ser escolhida através do link *Adicionar* referente à classe em que deseja-se cadastrar. Podemos ver isso na Figura 6.

Figura 6 - Tela de Entidades



Fonte: autor.

Ao clicar em *Adicionar*, uma tela é exibida contendo um formulário com os campos a serem preenchidos referente à entidade. A Figura 7 contém a tela referente ao cadastro de uma nova disciplina.

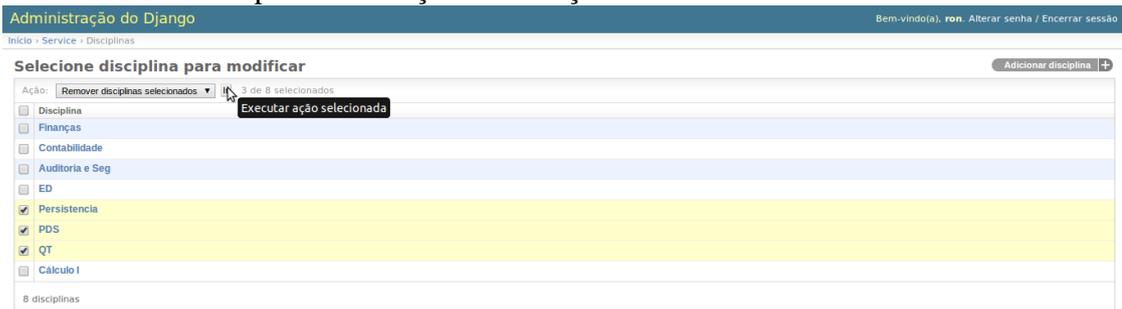
Figura 7 - Tela de Cadastro de Disciplina



Fonte: autor

Ao preencher o campo Nome e Qtd créditos, já pode-se confirmar o cadastro da disciplina desejada. Com o cadastro da disciplina confirmado, pode-se observar que a mesma já encontra-se na lista de disciplinas existentes. Neste momento, torna-se possível a edição ou exclusão de algum item existente ou de vários itens. Isso pode ser visto na Figura 8.

Figura 8- Tela de Disciplinas c/ Edição e Remoção



Fonte: autor.

Ao clicar na disciplina, a mesma tela de cadastro de disciplinas é exibida, porém com os valores referentes já preenchidos. Ou seja, prontos para serem alterados. Já para remover o item, basta que ele seja selecionado via *CheckBox* no componente e que a opção de Remover Item(s) Selecionado(s) seja clicada e confirmada.

Para exemplificar o uso desta disciplina cadastrada na figura, a Figura 9 refere-se ao cadastro de uma nova Oferta.

Figura 9 - Cadastro de uma Oferta



A imagem mostra a interface de administração Django para o cadastro de uma oferta. O formulário "Adicionar oferta" contém os seguintes campos:

- Docente:** Ricardo Reis (com ícone de adicionar)
- Disciplina:** Cálculo I (com ícone de adicionar)
- Ano semestre:** 2013.2

Na parte inferior do formulário, há três botões de ação: "Salvar e adicionar outro(a)", "Salvar e continuar editando" e "Cancelar".

Fonte: autor.

Pode-se observar que ao preencher o campo Disciplina, é possível ver a disciplina "Cálculo I" disponível. O campo Docente também exibe opções que já foram cadastradas anteriormente.

O cliente na web tem acesso à quatorze (14) telas contando com a tela de login. Com 3 cliques é possível adicionar ou editar algum dado em alguma entidade. Para remover algum valor, esse número aumenta para 4 devido ao clique de confirmação. A próxima seção contém as capacidades existentes no cliente do dispositivo móvel e como elas são exibidas e acessadas. Os requisitos funcionais relacionados ao lado web estão disponíveis no apêndice C, seção 3.1.

### c) Cliente no Dispositivo Móvel

Com os dados definidos no servidor, o cliente torna-se capaz de proceder em vários momentos de acordo com sua preferência. Suas capacidades dividem-se em duas situações: entrada de dados e leitura de dados.

Quando se trata de leitura de dados, o cliente possui acesso à todas as informações pertencentes no servidor referente a ele. Já na entrada de dados, o cliente só pode cadastrar aulas e presenças. Esse cadastro pode ser feito a partir da permissão de acesso ao servidor. Ou seja, é necessário que o usuário faça o login e seja validado para ter permissão tanto de leitura como de escrita no banco de dados. A Figura 10 contém a tela de login.

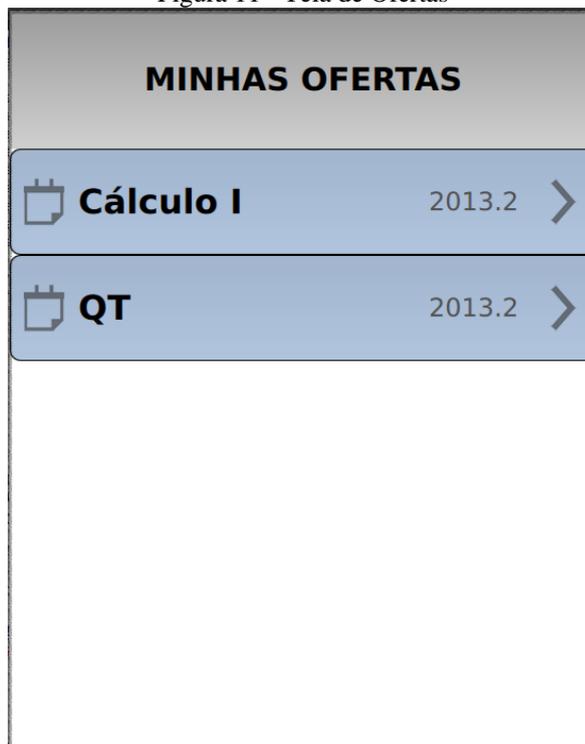
Figura 10 - Tela de Login - Móvel



Fonte: autor.

Com a validação do usuário concluída, o cliente (docente) possui acesso às ofertas em que ele está cadastrado. Podemos ver isso na Figura 11.

Figura 11 - Tela de Ofertas



Fonte: autor.

Ao clicar numa oferta, uma tela é exibida com uma aba já aberta chamada "Aulas" contendo uma lista das aulas já cadastradas. Além de uma aba "Alunos", que quando clicada torna possível a visualização de todos os alunos matriculados nessa oferta juntamente da porcentagem de presença de cada na disciplina. Um botão "+" referente a adição de uma nova aula também está presente ao lado da aba "Aulas". Nesse momento, o usuário pode seguir três caminhos: visualizar as aulas cadastradas que já são exibidas por padrão, visualizar os alunos matriculados e adicionar uma nova aula. Na exibição de aulas, cada item da lista possui o nome, o tipo e a data em que a mesma foi realizada. A Figura 12 contém a tela referente a oferta "Cálculo 1" com as exibições das abas "Aulas" e "Alunos" e do botão "+".

Figura 12 - Telas de Aulas e Alunos de Cálculo I



Fonte: autor.

Caso o usuário deseje criar uma nova aula, após o clique no botão "+", uma nova tela é exibida. Essa tela contém três seções referentes à aula: a primeira é um título referente ao nome da nova aula; a segunda seção é o tipo, onde o usuário tem a opção de escolher entre "NORMAL", "REPOSIÇÃO" e "FERIADO"; a terceira e última seção é a data que por padrão já exibe a atual, porém é editável. Após preencher os valores e clicar no botão "Adicionar", uma mensagem de confirmação aparece. O usuário pode cancelar e voltar para a

criação ou apenas confirmar a criação clicando em "Sim". A Figura 13 contém a tela de adição de aula vazia, preenchida e a tela da mensagem de confirmação.

Figura 13 - Telas de Adicionar Aula c/ Dialog



Fonte: autor.

Com isso, uma quarta seção surge na tela contendo uma lista com os alunos matriculados onde cada item exibe seu nome, sua matrícula e um valor de presença ou ausência (falta). Por padrão todos os alunos já estão presentes. Para registrar algum aluno ausente basta um clique em cima do item referente ao aluno que as informações presentes no item modificarão e passarão a exibir "Faltou" ao invés de "Presente". A Figura 14 contém as telas referentes à tela com a exibição dos alunos e seus estados de presença.

Figura 14 - Telas de Nova Aula c/ Alunos Presentes e Faltantes



Fonte: autor.

Ao clicar em "Confirmar", as presenças referentes à aula são realizadas e a tela de "Aulas" volta a ser exibida já contendo a nova aula na lista. Isso pode ser conferido na Figura 15.

Figura 15 - Tela de Aulas Atualizada



Fonte: autor.

O cliente no dispositivo móvel tem acesso à oito (8) telas contando com a tela de login. Com 4 cliques é possível criar uma nova aula e estar pronto para controlar a frequência. Os requisitos funcionais relacionados ao lado do dispositivo móvel estão disponíveis no apêndice C, seção 3.2. A próxima seção explica de forma detalhada como o cliente consome e registra dados no servidor.

#### **d) Relação Cliente-Servidor**

Para que as capacidades do cliente sejam possíveis, é necessário que requisições sejam executadas para o servidor e o mesmo envie uma resposta contendo o conteúdo requerido. Isso ocorre em toda atitude do cliente em que ele encontra-se em uma situação onde exibe alguma informação referente ao usuário. Ou seja, os clientes rodam o aplicativo no seu dispositivo móvel e a partir do login requisições HTTP são feitas para um servidor que as analisa e prepara as respostas para o usuário.

Dois métodos de requisições são feitas a partir do cliente: o método GET e o método POST. Requisições GET são chamadas em toda situação onde o usuário deseja ter acesso de leitura para algum dado. Todas as listas (ofertas, aulas e alunos) juntamente com suas informações são dados respondidos a partir de requisições GET. Requisições POST são efetuadas em todo momento onde o cliente deseja cadastrar um novo dado. O cadastro de aulas e de presenças realizado através do cliente no dispositivo móvel ou qualquer cadastro ou alteração realizado através do cliente na web são requisições POST.

O primeiro acesso do cliente na aplicação exige uma conexão com a internet (mais talhes na seção 5). Ele deve possuir um cadastro já registrado pelo administrador. Com o cadastro feito, a partir do dispositivo móvel ele já pode efetuar o login. Esse primeiro login busca no servidor todos os dados relacionados (ofertas, aulas e alunos) com a conta. Esses dados ficam armazenados de forma remota. Com esse acesso, torna-se possível que acessos posteriores sejam realizados sem a necessidade de uma conexão com internet. Ou seja, todos os dados referentes ao usuário ficarão armazenados de forma local depois do primeiro acesso. Esses dados são: valores referentes ao login, ofertas, aulas dessas ofertas, alunos dessas ofertas e às presenças dos alunos nessas aulas. Ou seja, estando offline a partir do segundo acesso ao aplicativo é possível ver todas as informações que já foram consultadas remotamente e também cadastrar novas aulas e presenças. Porém, para que essas novas requisições estejam armazenadas no servidor é necessário um novo acesso com internet. Em

outras palavras, é obrigatório estar online para que os dados cadastrados no estado offline sejam sincronizados com os valores existentes no servidor.

Todos os dados que o servidor dispõe são acessados pelo cliente através de serviços. Cada ação do usuário no dispositivo ou na web é requisitada para o servidor e o mesmo a trata de acordo com um serviço específico. A próxima seção descreve cada serviço existente no servidor.

### **e) Serviços**

Como dito na seção anterior, o servidor possui alguns serviços que são consumidos pelo cliente. No total são sete (7) serviços: informações do docente, ofertas de um docente, aulas de uma oferta, adição de aula, alunos de uma oferta, presenças de uma aula e adição de presenças. Sendo quatro (4) tratadas a partir de uma requisição GET e três (3) de requisições POST.

Quando o cliente executa uma requisição POST, três serviços podem ser solicitados. O primeiro é o serviço de informações do docente. Tal requisição deve conter o login do usuário como dado enviado. Com esse valor o serviço busca as demais informações referentes à esse usuário e responde em formato JSON para o usuário para fins de validação ou para aproveitamento de valores, como o nome e e-mail do docente. O segundo trata-se do serviço de adição de aula que deve ser requisitado contendo o id da oferta e as informações da aula (nome, tipo e data) como dados enviados. O resultado dessa chamada é a adição da aula na lista de aulas da oferta em específico. Por fim, o terceiro e último serviço POST é o de adição de presença que é solicitado passando o id da aula, o id do aluno e o campo presença como dados enviados. O resultado dessa chamada é a adição da presença na lista de presenças da aula em específico.

Já quando o cliente executa uma requisição GET, quatro serviços podem ser solicitados:

- O primeiro serviço é o de ofertas de um docente que é solicitado passando o id do docente como parâmetro. O resultado dessa chamada é um JSON contendo todas as ofertas do docente e seus respectivos campos: nome e id do docente; nome e id da disciplina; e anoSemestre.
- O segundo serviço trata-se de aulas de uma oferta que é solicitado passando o id da oferta como parâmetro. O resultado dessa chamada é um

JSON contendo todas as aulas cadastradas nessa oferta e seus respectivos campos: id da aula; id da oferta; nome da aula; tipo da aula; e data da aula.

- O terceiro serviço é o de alunos de uma oferta que é solicitado passando o id da oferta como parâmetro. O resultado dessa chamada é um JSON contendo todas os alunos cadastrados nessa oferta e seus respectivos campos: id do aluno; id da oferta; nome da aluno; e matrícula do aluno.
- O quarto e último serviço é o de presenças de uma aula que é solicitado passando o id da aula como parâmetro. O resultado dessa chamada é um JSON contendo todas as presenças cadastradas nessa aula e seus respectivos campos: id da presença; id da aula; id do aluno; e um campo presença que pode ser "true" se o aluno estiver presente ou "false" caso contrário.

## 5 VALIDAÇÃO

O sistema teve todos os seus requisitos funcionais atendidos (essencial, importante e desejável). Tais requisitos estão divididos entre lado web e lado android. No lado web são quinze (15) requisitos no total. Destes quinze, são seis (6) com prioridades essenciais e nove (9) importantes. No lado android são oito(8) no total, sendo seis essenciais e dois (2) importantes.

Na etapa final de implantação da aplicação em sala de aula. Critérios apresentados na seção de ambiente de utilização foram seguidos e pôde-se executar o aplicativo em um dispositivo móvel. A primeira execução foi realizada nas seguintes condições:

- Professor em pessoa para ser o usuário.
- Cadastros de disciplinas e alunos (fictícios) e o cadastro de login desse professor no sistema.
- Smartphone *Android* com versão 2.3.6, conexão wireless e 100mb disponível de memória interna.
- Computador desktop com sistema operacional *Ubuntu 12.04 LTS*, rede local criada e compartilhada. Navegador *Google Chrome v31*.
- Sala de aula fictícia. Estrutura acadêmica, porém sem alunos e sem aula.

O primeiro acesso do cliente na aplicação exige uma conexão com a internet (*online*). A partir do segundo acesso, a aplicação passa a funcionar *offline*, porém os dados cadastrados nesse momento só serão sincronizados após o próximo acesso *online*. No primeiro acesso, os dados referentes às ofertas de disciplina do professor são todos visualizados no dispositivo móvel. Caso o usuário cadastre aulas e presenças nestas ofertas ainda estando *online*, esses novos dados são enviados diretamente para o servidor e armazenados. Quando *offline*, cada requisição feita a partir do dispositivo é salva em um arquivo e quando a aplicação recupera sua conexão, cada requisição do arquivo é feita e assim atualizando os dados no serviço. Dessa forma, o usuário não consegue ter acesso aos novos dados cadastrados via web estando com o dispositivo sem conexão realizada. O aplicativo rodou com sucesso e a chamada foi realizada. Como existia uma conexão com acesso a internet, os dados tanto foram lidos diretamente do servidor como foram escritos nele.

Tanto no sistema web (lado web) quanto na aplicação android (lado móvel) foi utilizado como banco de dados o SQLite. Não existe um banco local no dispositivo, a base de dados fica no servidor. Essa preocupação não pertencia ao escopo deste trabalho e serve como trabalho futuro, como pode ser lido na seção 6.2.

Por fim, duas observações ligadas ao dispositivo móvel foram destacadas após a execução da aplicação:

- A aplicação móvel funciona em dispositivos com no mínimo a versão 2.2.
- Versões até a 2.3.6 demoram para executar bem mais quando comparado às versões mais modernas (3.0 até 4.3).

## 6 CONCLUSÃO

### 6.1 Objetivos Alcançados

Esse trabalho teve como objetivo geral o desenvolvimento de uma aplicação para dispositivo móvel que auxiliasse o professor em sala de aula no controle da atividade de controle de frequência dos alunos. Para isso, cinco objetivos específicos foram traçados: elaborar uma estrutura do aplicativo; levantar os requisitos do sistema; desenvolver o servidor Web e o aplicativo móvel; executar testes na aplicação para verificar a conformidade com os requisitos; e implantação da aplicação em uma aula real.

A elaboração da estrutura do aplicativo foi concluída. As principais características relacionadas foram detalhadas e apresentadas. Funcionalidades advindas desse detalhamento foram discutidas e analisadas.

A versão inicial dos requisitos foi definida com as funcionalidades levantadas possibilitando iniciar a modelagem dos dados. O documento de requisitos do Routine pode ser visto no final deste trabalho como apêndice C.

Após o levantamento dos requisitos iniciou-se a implementação do servidor web. Após a finalização do lado web, foi construído o lado cliente executando em dispositivo móvel. Vale ressaltar que esta ordem de desenvolvimento permitiu estruturação mais sólida do cliente haja vista os serviços webs já estarem disponíveis. Durante o desenvolvimento foi observado e corrigido que dados advindos de serviços poderiam ir para o cliente de maneira mais simplificada ao invés da forma definida inicialmente. Os requisitos básicos definidos foram suficientes para que a primeira fase fosse concluída.

Na construção do servidor Web, primeiramente foi documentado como os dados seriam organizados, relacionados e modelados. A primeira versão dessa modelagem foi feita com o apoio de um docente especializado na área de banco de dados. Além do servidor, foi acionado o site administrativo advindo do framework Django permitindo entrada de dados inicialmente para testes.

Inicialmente, no que diz respeito a interface do cliente, a intenção era efetuar a construção utilizando widgets tradicionais no Qt. Entretanto, devido a natureza estática dos widgets, foi aplicado o módulo QtQuick que permitia estruturação de interfaces com fluidez entre estados.

Na etapa de testes foi possível aprimorar a usabilidade da aplicação reajustando os componentes visuais. Algumas mudanças também foram feitas nos requisitos nesta etapa.

A importância da participação do usuário no decorrer do desenvolvimento foi um ponto positivo referente a realização deste trabalho que vale ser ressaltado.

Um ponto negativo da aplicação é a necessidade da presença de rede para funcionamento. Sugerimos a persistência dos dados no dispositivo e seu respectivo sincronismo com os dados no servidor como trabalho futuro.

## **6.2 Projetos Futuros**

Trabalhos futuros podem focar na consistência de dados. Criar um banco de dados local com todas as entidades sincronizadas com os dados remotos vai proporcionar um aplicativo persistente e uma maior segurança para o usuário quanto aos dados armazenados. Outras funcionalidades como podem ser bastante significantes para o âmbito acadêmico, como:

- Submissão de trabalhos.
- Agendamento de lembretes.
- Lado aluno do aplicativo.
- Controle de frequência com duas presenças por aula.

Esse trabalho não teve enfoque em questões de segurança, nem de persistência no banco de dados por não pertencer ao seu escopo.

## 7 RELATO DE EXPERIÊNCIA

O desenvolvimento deste trabalho exigiu a utilização de conhecimentos adquiridos no decorrer do curso de Sistemas de Informação. Foi necessária a realização de pesquisas, um planejamento adequado para desenvolver um software e a avaliação de diferentes tecnologias para concluir esse projeto. Conceitos de engenharia de software, banco de dados e utilização de ferramentas diversas estiveram presentes durante a construção.

Para desenvolver o Routine foi necessário: ter uma visão geral das atividades e ferramentas que auxiliam o processo de desenvolvimento de software. Além da identificação e comparação dos modelos desse processo, do seu ciclo de vida e das metodologias de análise/projeto. As disciplinas de *Análise e Projeto de Sistemas (APS)* e de *Engenharia de Software* foram proveitosas nesse momento.

Em relação ao banco de dados, a construção da aplicação exigiu o conhecimento de modelagem conceitual de dados, modelo relacional e gerenciamento de dados remotos. A decisão de documentar a modelagem de dados antes da implementação do banco foi resultado de estudos que destacaram essa atitude como uma boa prática. Também foi preciso entender o real funcionamento por trás das linhas de código do framework Django para tornar possível uma melhor forma de utilizá-lo. Os conceitos abordados na disciplina *Fundamentos de Banco de Dados*, *Desenvolvimento de Software para Web* e *Desenvolvimento de Software para Persistência* auxiliaram nessa fase.

Como o projeto incluiu a atividade de programação, era crucial o conhecimento em tecnologias que fornecessem o suporte para a implementação. A habilidade de programar pôde ser aplicada e aprimorada através de algumas disciplinas do curso, como: *Fundamentos de Programação*, *Programação Orientada a Objetos* e *Desenvolvimento em Qt*. Ainda era preciso avaliar e comparar tais tecnologias para tomar uma decisão de qual seria a melhor em cada contexto aplicado. Decisões essas eram tomadas através de conceitos abordados na disciplina de *Linguagens de Programação*, onde as principais características de diferentes linguagens foram abordadas e comparadas.

De maneira geral, com base no perfil do curso, esse trabalho conseguiu exigir de competências que proporcionaram:

- O desenvolvimento de um sistema de informação.

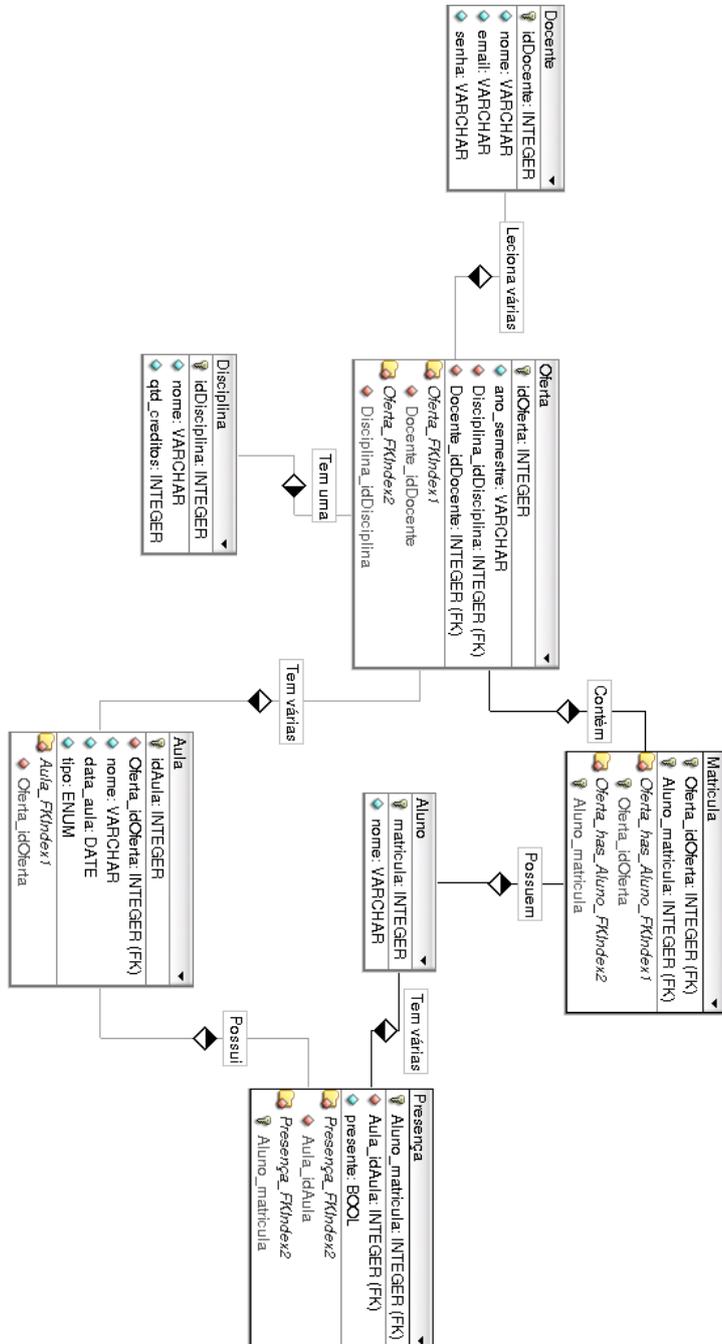
- A modelagem de soluções de melhoria com o uso dos conceitos, métodos, técnicas e ferramentas apresentadas no curso. Com isso, a montagem de um projeto específico para a viabilização das soluções apontadas.
- Validação e transmissão da solução de melhoria de uma situação em uma forma efetiva e contextualizada ao estado original.

## REFERÊNCIAS

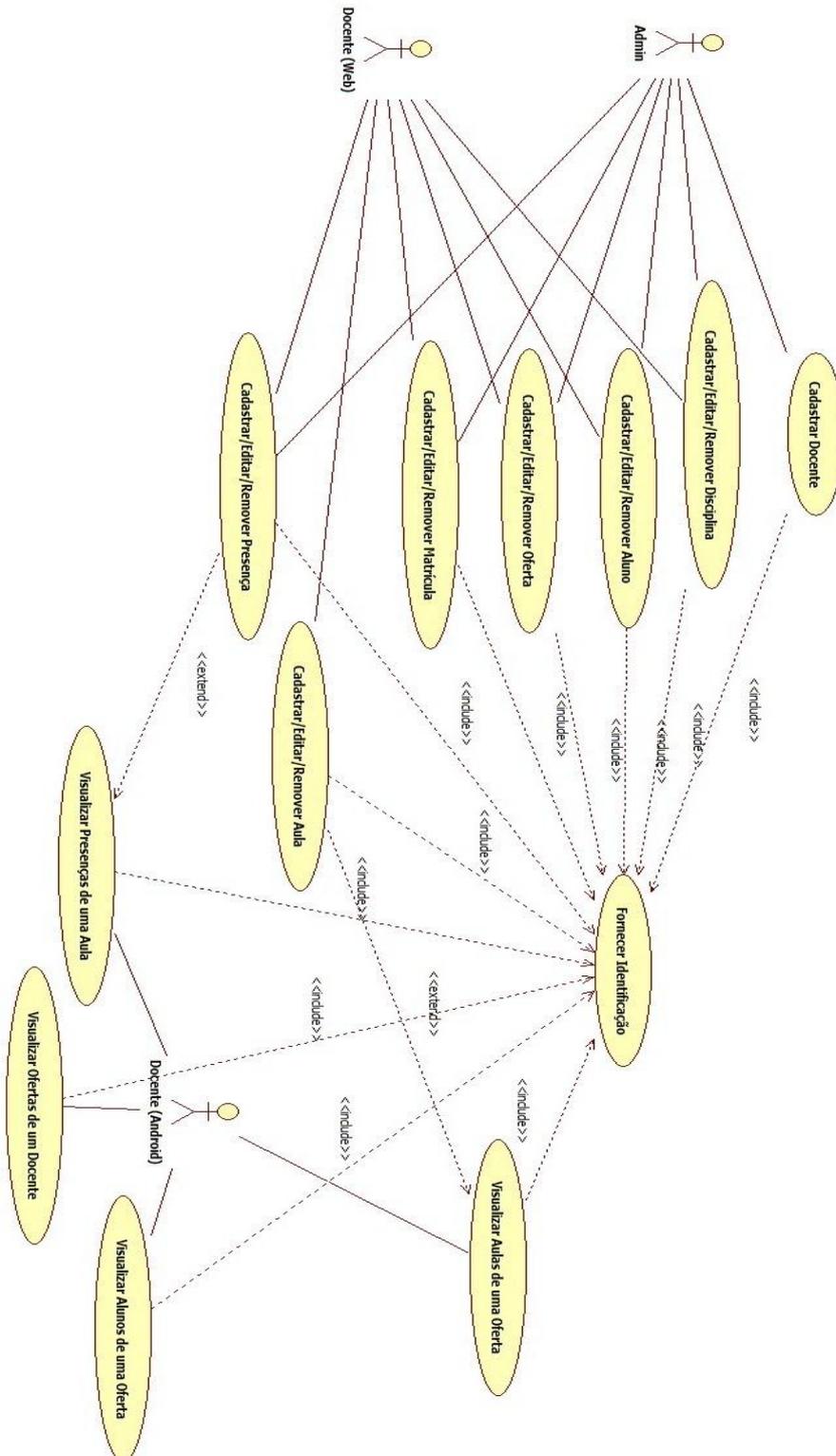
- CONVERGÊNCIA DIGITAL (Brasil) (Ed.). **Para 56% dos brasileiros, acesso à Internet é por smartphones e tablets.** Disponível em: <<http://convergenciadigital.uol.com.br/cgi/cgilua.exe/sys/start.htm?inford=30382&sid=8#.UctqRfkQass>>. Acesso em: 11 dez. 2013.
- FILHO, Wilson de P. P. **Engenharia de Software: fundamentos, métodos e padrões.** 3.ed. Rio de Janeiro: LTC, 2009.
- GALVIN, Deleon. **Protótipo de Sistema CRM para Dispositivos Móveis Utilizando a Tecnologia .NET.** 2004. 90 f. Dissertação (Bacharelado) - Curso de Ciências da Computação, Departamento de Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau, 2004.
- LEE, V.; SCHNEIDER, H.; SCHELL, R. **Aplicações Móveis – Arquitetura, Projeto e Desenvolvimento.** São Paulo: Pearson Makron Books, 2005.
- LIMA, Luciano. **52 milhões de pessoas têm acesso à web pelo celular, aponta IBOPE Media.** Disponível em: <<http://iabbrasil.net/portal/52-milhoes-de-pessoas-tem-acesso-a-web-pelo-celular-aponta-ibope-media/>>. Acesso em: 11 dez. 2013
- NAUR, P. e RANDALL, B., (eds.), **Software Engineering: A Report on a Conference Sponsored by the NATO Science Committee.** NATO, 1969.
- PRESSMAN, Roger S. **Engenharia de Software.** 6.ed. Porto Alegre: AMGH, 2010.
- SIBALDO, Maria Aparecida A. et al. Mobile GraW: uma aplicação para dispositivos móveis baseada em comunidades virtuais de aprendizagem com suporte a recomendação. In: SBIE, 18., 2007, São Paulo. **XVIII Simpósio Brasileiro de Informática na Educação.** São Paulo: Sbie, 2007. p. 214 - 217.
- SOMMERVILLE, Ian. **Engenharia de Software.** 9.ed. São Paulo: Pearson Prentice Hall, 2011.

## APÊNDICES

### APÊNDICE A – Documento de Modelagem do Banco de Dados



## APÊNDICE B – Diagrama de Casos de Uso



## APÊNDICE C – Documento de Requisitos do Routine

## **Documento de Requisitos - Routine**

**Versão 3.0**

## Histórico de Alterações

<b>Data</b>	<b>Versão</b>	<b>Descrição</b>	<b>Autor</b>
05/08/2013	1.0	Criação do documento	Ronaldo Dantas
20/08/2013	2.0	Escopo reduzido e funcionalidades retiradas	Ronaldo Dantas
28/10/2013	3.0	Divisão de requisitos definidos por: lado web e lado android.	Ronaldo Dantas

# Sumário

<b>1. INTRODUÇÃO .....</b>	<b>5</b>
1.1 VISÃO GERAL DO DOCUMENTO .....	5
1.2 CONVENÇÕES, TERMOS E ABREVIACÕES .....	5
1.2.1 Identificação dos requisitos .....	5
As abreviações usadas no documento são: .....	5
• RF00X – Requisito Funcional; .....	5
• RNF00X – Requisito Não Funcional; .....	5
<b>2. DESCRIÇÃO GERAL DA APLICAÇÃO .....</b>	<b>6</b>
<b>3. REQUISITOS FUNCIONAIS (CASOS DE USO).....</b>	<b>6</b>
3.1 LADO WEB .....	6
3.1.1 Login.....	6
[RF001] Fazer login.....	6
3.1.2 Disciplinas .....	6
[RF002] Criar disciplina.....	6
[RF003] Editar disciplina .....	7
[RF004] Apagar oferta.....	7
3.1.3 Ofertas .....	7
[RF005] Criar oferta.....	7
[RF006] Editar oferta .....	7
[RF007] Apagar oferta.....	7
3.1.4 Alunos .....	8
[RF008] Adicionar aluno .....	8
[RF009] Editar aluno.....	8
[RF010] Apagar aluno .....	8
3.1.5 Matrículas.....	8
[RF011] Adicionar matrícula.....	8
[RF012] Editar matrícula .....	9
[RF013] Apagar matrícula.....	9
3.1.6 Aulas .....	9
[RF011] Adicionar aula.....	9
[RF012] Editar aula.....	9
[RF013] Apagar aula.....	10
3.1.7 Presenças.....	10
[RF014] Adicionar presença.....	10
[RF015] Editar presença.....	10
3.2 LADO ANDROID.....	10
3.2.1 Login.....	10
[RF001] Fazer login.....	10
3.2.2 Ofertas .....	11
[RF002] Visualizar oferta(s) e suas informações.....	11
3.2.3 Alunos .....	11
[RF003] Visualizar alunos e suas situações de presença .....	11
3.2.4 Aulas .....	11
[RF004] Visualizar aulas .....	11
[RF005] Adicionar aula .....	11
[RF006] Editar aula.....	12
3.2.5 Presenças.....	12
[RF007] Adicionar presença.....	12
[RF008] Editar presença.....	12
<b>4. REQUISITOS NÃO FUNCIONAIS .....</b>	<b>12</b>
[RNF001] Usabilidade.....	13
[RNF002] Desempenho.....	13



## 1. Introdução

Este documento especifica oficialmente os requisitos funcionais e não funcionais da aplicação Routine. Serve como base para a equipe de desenvolvimento, fornecendo informações necessárias para a implementação das funcionalidades ao longo de todo o ciclo de vida do produto.

### 1.1 Visão geral do documento

Além desta seção introdutória, as seções seguintes estão organizadas como descrito abaixo.

- **Seção 2 – Descrição geral da aplicação:** apresenta uma visão geral da aplicação, caracterizando qual é o seu escopo e descrevendo seus usuários;
- **Seção 3 – Requisitos funcionais (casos de uso):** especifica todos os casos de uso da aplicação, descrevendo os fluxos de eventos, prioridades, atores, entradas e saídas de cada caso de uso a ser implementado;
- **Seção 4 – Requisitos não-funcionais:** especifica todos os requisitos não funcionais da aplicação, divididos em requisitos de usabilidade e de confiabilidade.

### 1.2 Convenções, termos e abreviações

A correta interpretação deste documento exige o conhecimento de algumas convenções e termos específicos, que são descritos a seguir.

#### 1.2.1 Identificação dos requisitos

As abreviações usadas no documento são:

- RF00X – Requisito Funcional;
- RNF00X – Requisito Não Funcional;

#### 1.2.2 Prioridades dos requisitos

Para estabelecer a prioridade dos requisitos, nas seções 4 e 5, foram adotadas as denominações “essencial”, “importante” e “desejável”.

- **Essencial** é o requisito sem o qual a aplicação não entra em funcionamento. Requisitos essenciais são requisitos imprescindíveis, que têm que ser implementados impreterivelmente.
- **Importante** é o requisito sem o qual a aplicação entra em funcionamento, mas de forma não satisfatória. Requisitos importantes devem ser implementados, mas, se não forem, a aplicação poderá ser implantado e usado mesmo assim.
- **Desejável** é o requisito que não compromete as funcionalidades básicas da aplicação, isto é, a aplicação pode funcionar de forma satisfatória sem ele. Requisitos desejáveis podem ser deixados para versões posteriores da aplicação,

caso não haja tempo hábil para implementá-los na versão que está sendo especificada.

## 2. Descrição geral da aplicação

O Routine tem por objetivo auxiliar no gerenciamento da atividade de controle de frequência. O professor pode realizar as "chamadas" via dispositivo móvel com o sistema operacional Android acessando os dados de maneira remota. A leitura e a escrita dos dados podem ser feitas a partir do lado web ou do lado android. A partir do lado web, uma interface administrativa auxilia no gerenciamento dos dados de todas as entidades. Já no lado android, apenas dados referentes a aulas e presenças podem ser submetidos ao servidor enquanto os outros apenas podem ser lidos. A próxima seção mostrará todas as funcionalidades da aplicação.

## 3. Requisitos funcionais (casos de uso)

### 3.1 Lado Web

#### 3.1.1 Login

#### [RF001] Fazer login

**Descrição do caso de uso:** Este caso de uso permite que o usuário faça login diretamente no site administrativo fornecendo login e senha.

**Prioridade:**  Essencial  Importante  Desejável

**Pré-condições:** já existir uma conta no servidor com as informações digitadas e possuir permissão concedida pelo administrador.

**Pós-condição:** o usuário realiza o login e passa para a tela de exibição de entidades.

#### 3.1.2 Disciplinas

#### [RF002] Criar disciplina

**Descrição do caso de uso:** Este caso de uso permite que o usuário crie uma oferta.

**Prioridade:**  Essencial  Importante  Desejável

**Pré-condições:** não tem.

**Pós-condição:** o usuário cria a oferta desejada.

<b>[RF003] Editar disciplina</b>
----------------------------------

**Descrição do caso de uso:** Este caso de uso permite ao usuário no lado web abrir uma disciplina para alteração.

**Prioridade:**       Essencial                       Importante                       Desejável

**Pré-condições:** a disciplina deve existir.

**Pós-condição:** o usuário consegue abrir a disciplina desejada e editá-la.

<b>[RF004] Apagar oferta</b>
------------------------------

**Descrição do caso de uso:** Este caso de uso permite que o usuário apague uma disciplina existente.

**Prioridade:**       Essencial                       Importante                       Desejável

**Pré-condições:** a disciplina deve existir.

**Pós-condição:** o usuário consegue apagar a disciplina desejada.

### 3.1.3 Ofertas

<b>[RF005] Criar oferta</b>
-----------------------------

**Descrição do caso de uso:** Este caso de uso permite que o usuário crie uma oferta.

**Prioridade:**       Essencial                       Importante                       Desejável

**Pré-condições:** já existir um docente validado e uma disciplina.

**Pós-condição:** o usuário cria a oferta desejada.

<b>[RF006] Editar oferta</b>
------------------------------

**Descrição do caso de uso:** Este caso de uso permite ao usuário no lado web abrir uma oferta para alteração.

**Prioridade:**       Essencial                       Importante                       Desejável

**Pré-condições:** a oferta deve existir.

**Pós-condição:** o usuário consegue abrir a oferta desejada e editá-la.

<b>[RF007] Apagar oferta</b>
------------------------------

**Descrição do caso de uso:** Este caso de uso permite que o usuário apague uma oferta existente.

**Prioridade:**       Essencial                       Importante                       Desejável

**Pré-condições:** a oferta deve existir.

**Pós-condição:** o usuário consegue apagar a oferta desejada.

### 3.1.4 Alunos

#### [RF008] Adicionar aluno

**Descrição do caso de uso:** Este caso de uso permite que o usuário adicione um aluno à uma oferta já existente.

**Prioridade:**  Essencial  Importante  Desejável

**Pré-condições:** existir a oferta desejada.

**Pós-condição:** o usuário cria o aluno e o vincula à oferta desejada.

#### [RF009] Editar aluno

**Descrição do caso de uso:** Este caso de uso permite ao usuário no lado web abrir dados de um aluno para alteração.

**Prioridade:**  Essencial  Importante  Desejável

**Pré-condições:** o aluno deve existir.

**Pós-condição:** o usuário consegue abrir o aluno desejado e editá-lo.

#### [RF010] Apagar aluno

**Descrição do caso de uso:** Este caso de uso permite que o usuário apague um aluno existente.

**Prioridade:**  Essencial  Importante  Desejável

**Pré-condições:** o aluno deve existir.

**Pós-condição:** o usuário consegue apagar o aluno desejado.

### 3.1.5 Matrículas

#### [RF011] Adicionar matrícula

**Descrição do caso de uso:** Este caso de uso permite que o usuário adicione uma matrícula referente a um aluno à uma oferta já existente.

**Prioridade:**  Essencial  Importante  Desejável

**Pré-condições:** existir o aluno e a oferta desejada.

**Pós-condição:** o usuário cria a matrícula.

<b>[RF012] Editar matrícula</b>
---------------------------------

**Descrição do caso de uso:** Este caso de uso permite ao usuário no lado web abrir dados de uma matrícula para alteração.

**Prioridade:**       Essencial                       Importante                       Desejável

**Pré-condições:** a matrícula deve existir.

**Pós-condição:** o usuário consegue abrir a matrícula desejada e editá-la.

<b>[RF013] Apagar matrícula</b>
---------------------------------

**Descrição do caso de uso:** Este caso de uso permite que o usuário apague uma matrícula existente.

**Prioridade:**       Essencial                       Importante                       Desejável

**Pré-condições:** a matrícula deve existir.

**Pós-condição:** o usuário consegue apagar a matrícula desejada.

### 3.1.6 Aulas

<b>[RF011] Adicionar aula</b>
-------------------------------

**Descrição do caso de uso:** Este caso de uso permite que o usuário adicione uma aula em uma oferta.

**Prioridade:**       Essencial                       Importante                       Desejável

**Pré-condições:** a oferta que terá a aula deve existir.

**Pós-condição:** o usuário consegue adicionar a nova aula na oferta desejada.

<b>[RF012] Editar aula</b>
----------------------------

**Descrição do caso de uso:** Este caso de uso permite que ao usuário edite uma aula.

**Prioridade:**       Essencial                       Importante                       Desejável

**Pré-condições:** a aula deve estar salva.

**Pós-condição:** o usuário consegue editar a aula desejada.

<b>[RF013] Apagar aula</b>
----------------------------

**Descrição do caso de uso:** Este caso de uso permite que ao usuário apague uma aula.

**Prioridade:**       Essencial                       Importante                       Desejável

**Pré-condições:** a aula deve está salva.

**Pós-condição:** o usuário consegue apagar a aula desejada.

### 3.1.7 Presenças

<b>[RF014] Adicionar presença</b>
-----------------------------------

**Descrição do caso de uso:** Este caso de uso permite que o usuário adicione uma presença em uma aula de uma oferta referente a um aluno.

**Prioridade:**       Essencial                       Importante                       Desejável

**Pré-condições:** a aula deve existir.

**Pós-condição:** o usuário consegue adicionar a presença do aluno na aula desejada.

<b>[RF015] Editar presença</b>
--------------------------------

**Descrição do caso de uso:** Este caso de uso permite que ao usuário edite uma presença de um aluno.

**Prioridade:**       Essencial                       Importante                       Desejável

**Pré-condições:** a presença deve está salva.

**Pós-condição:** o usuário consegue editar a presença desejada.

## 3.2 Lado Android

### 3.2.1 Login

<b>[RF001] Fazer login</b>
----------------------------

**Descrição do caso de uso:** Este caso de uso permite que o usuário faça login através do seu dispositivo móvel fornecendo e-mail e senha.

**Prioridade:**       Essencial                       Importante                       Desejável

**Pré-condições:** já existir uma conta no servidor com as informações digitadas.

**Pós-condição:** o usuário realiza o login e passa para a tela de exibição de ofertas.

### 3.2.2 Ofertas

#### [RF002] Visualizar oferta(s) e suas informações

**Descrição do caso de uso:** Este caso de uso permite que o usuário visualize a(s) oferta(s) referente a seu login juntamente de dados como número de alunos, número de aulas já realizadas e quantidade de créditos.

**Prioridade:**  Essencial  Importante  Desejável

**Pré-condições:** já existir um docente validado e uma oferta vinculada a ele.

**Pós-condição:** o usuário visualiza a(s) oferta(s) e suas informações na tela.

### 3.2.3 Alunos

#### [RF003] Visualizar alunos e suas situações de presença

**Descrição do caso de uso:** Este caso de uso permite que o usuário visualize os alunos matriculados em uma oferta já existente juntamente de uma porcentagem de presença.

**Prioridade:**  Essencial  Importante  Desejável

**Pré-condições:** existir a oferta desejada e os alunos matriculados.

**Pós-condição:** o usuário visualiza os alunos e suas porcentagens de presença na tela.

### 3.2.4 Aulas

#### [RF004] Visualizar aulas

**Descrição do caso de uso:** Este caso de uso permite que o usuário visualize as aulas já realizadas na oferta relacionada.

**Prioridade:**  Essencial  Importante  Desejável

**Pré-condições:** a oferta que possui as aulas deve existir.

**Pós-condição:** o usuário consegue visualizar as aulas na tela.

#### [RF005] Adicionar aula

**Descrição do caso de uso:** Este caso de uso permite que o usuário adicione uma aula com suas informações em uma oferta.

**Prioridade:**      Essencial                     Importante                     Desejável

**Pré-condições:** a oferta que terá a aula deve existir.

**Pós-condição:** o usuário consegue adicionar a nova aula na oferta desejada.

#### [RF006] Editar aula

**Descrição do caso de uso:** Este caso de uso permite que ao usuário edite uma aula.

**Prioridade:**      Essencial                     Importante                     Desejável

**Pré-condições:** a aula deve está salva.

**Pós-condição:** o usuário consegue editar a aula desejada.

### 3.2.5 Presenças

#### [RF007] Adicionar presença

**Descrição do caso de uso:** Este caso de uso permite que o usuário adicione uma presença em uma aula de uma oferta referente a um aluno.

**Prioridade:**      Essencial                     Importante                     Desejável

**Pré-condições:** a aula deve existir.

**Pós-condição:** o usuário consegue adicionar a presença do aluno na aula desejada.

#### [RF008] Editar presença

**Descrição do caso de uso:** Este caso de uso permite que ao usuário edite uma presença de um aluno.

**Prioridade:**      Essencial                     Importante                     Desejável

**Pré-condições:** a presença deve está salva.

**Pós-condição:** o usuário consegue editar a presença desejada.

## 4. Requisitos não funcionais

**[RNF001] Usabilidade**

A interface com o usuário é de vital importância para o sucesso da aplicação. Principalmente por ser uma aplicação móvel, onde o espaço é limitado para informações. Desta forma, uma interface amigável ao usuário é um requisito essencial.

**Prioridade:**      Essencial                     Importante                     Desejável

**[RNF002] Desempenho**

Embora não seja um requisito essencial para a aplicação, deve ser considerada por corresponder a um fator de qualidade de software.

**Prioridade:**      Essencial                     Importante                     Desejável