

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

LETICIA UEDA
VANESSA M. SILVA

**MAAS - SISTEMA PARA INTEGRAÇÃO DE COMPROMISSOS ACADÊMICOS
EMPREGANDO TECNOLOGIA MÓVEL**

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA

2013

LETICIA UEDA
VANESSA M. SILVA

**MAAS - SISTEMA PARA INTEGRAÇÃO DE COMPROMISSOS ACADÊMICOS
EMPREGANDO TECNOLOGIA MÓVEL**

Trabalho apresentado à disciplina de Trabalho de Conclusão de Curso 2 do Curso Superior de Bacharelado em Sistemas de Informação do Departamento Acadêmico de Informática – DAINF - da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Orientador: Prof. Dr. Paulo Cezar Stadzisz

CURITIBA

2013

AGRADECIMENTOS

Agradecemos o nosso orientador Paulo Cezar Stadzisz, por ter nos guiado e motivado no desenvolvimento deste Trabalho de Conclusão de Curso.

Às professoras da disciplina de Trabalho de Conclusão de Curso Mariângela de Oliveira Gomes Setti e Marília Abrahão Amaral, pelo grande auxílio dado ao longo do ano.

Ao Cion Ayres do Nascimento, representante do RLE, por ter fornecido um servidor para hospedar a aplicação Web deste projeto.

Por último, mas não menos importante, agradecemos a Deus.

DEDICATÓRIA

(Vanessa)

Dedico este trabalho ao meu marido Marcos Tupan que me concedeu a oportunidade de realizar um sonho, que o tornou possível, acreditou em meu potencial e com seu amor apoio e carinho me deu forças para seguir adiante.

(Letícia)

Dedico este trabalho aos meus pais, que sempre me deram bons conselhos e oportunidades durante minha vida, e ao meu namorado Gabriel Centenaro, que está sempre ao meu lado, me motivando e apoiando.

RESUMO

UEDA, Letícia; SILVA, Vanessa M.. MAAS - SISTEMA PARA INTEGRAÇÃO DE COMPROMISSOS ACADÊMICOS EMPREGANDO TECNOLOGIA MÓVEL. 100f. Trabalho de Conclusão de Curso – Curso Superior de Bacharelado em Sistemas de Informação, Universidade Tecnológica Federal do Paraná. Curitiba, 2013.

Este trabalho de conclusão de curso aborda uma pesquisa tecnológica que apoiou o desenvolvimento de um software para gestão acadêmica. O software desenvolvido representa uma agenda acadêmica implementada para Windows Phone. Este artefato pode ser constantemente atualizado com informações de compromissos acadêmicos, que são requisitados por várias fontes em um servidor web. Esse software é apresentado como uma possível solução para algumas dificuldades vivenciadas no ambiente acadêmico, tais como a dificuldade de comunicação eficiente entre professores e alunos e a dificuldade de gerenciamento das atividades acadêmicas pelos alunos. O software envolve a utilização de tecnologias estudadas pela equipe durante o curso de Bacharelado em Sistemas de Informação como *smartphone*, arquitetura cliente-servidor, conceitos de design de interação e banco de dados.

Palavras-chave: Agenda acadêmica. Windows Phone.

ABSTRACT

UEDA, Letícia; SILVA, Vanessa M.. MAAS – MOBILE ACADEMIC APPOINTMENT SYSTEM. 100f. Trabalho de Conclusão de Curso – Curso Superior de Bacharelado em Sistemas de Informação, Universidade Tecnológica Federal do Paraná. Curitiba, 2013.

This monograph discusses the development of a software for academic activities' management. The software includes an agenda implemented for Windows Phone. The artifact can be constantly updated with information about academic appointments, which are included from different sources in a web server. The software represents a possible solution for some difficulties experienced in the academic environment, such as the difficulties to have efficient communication among professors and students and to manage the students' academic activities. The software involves the use of technologies studied by the development team during the Information Systems university course, such as smartphones, client-server architecture, interaction design concepts and data bases.

Key-words: Academic Calendar. Windows Phone.

LISTA DE FIGURAS

Figura 1- Aviso de ausência de professor no quadro negro.	16
Figura 2 - Avisos de Alteração de Local de Prova.	17
Figura 3 - Estilo Metro em Windows Phone.	24
Figura 4 - Tela My Class Schedule.	28
Figura 5 - Tecnologias Utilizadas.....	29
Figura 6 – Tela inicial do Visual Studio 2012.....	32
Figura 7 – Estrutura do Objeto JSON.....	34
Figura 8 - Estrutura do Array JSON.....	34
Figura 9 - Arquitetura MVC.....	41
Figura 10 - Arquitetura do MAAS	41
Figura 11 – Diagrama de Caso de Uso Efetuar <i>Login</i>	51
Figura 12 – Diagrama de Caso de Uso Conectar ao servidor.....	51
Figura 13 - Diagrama de Caso de Uso Apresentar Calendário.....	52
Figura 14 - Diagrama de Caso de Uso Efetuar <i>Login</i> Servidor.....	53
Figura 15 - Diagrama de Caso de Uso Expandir Compromissos e Eventos do Dia.....	53
Figura 16 - Diagrama de Caso de Uso Efetuar <i>Logout</i>	54
Figura 17 - Modelo do Banco de Dados Servidor Web.	55
Figura 18 - Diagrama de classes do servidor.....	56
Figura 19 - Modelo do banco de dados da aplicação móvel	57
Figura 20 - Diagrama de classes da aplicação móvel.....	58
Figura 21 - <i>Mock-up</i> Efetuar <i>Login</i> Dispositivo Móvel	60
Figura 22 - <i>Mock-up</i> Apresentar Agenda Dispositivo Móvel	61
Figura 23 - <i>Mock-up</i> Visualizar Agenda Modo Dia.....	61
Figura 24 - <i>Mock-up</i> Apresentar Compromisso/Evento.....	62
Figura 25 - <i>Mock-up</i> Tela <i>Login</i> Servidor.....	63
Figura 26 - <i>Mock-up</i> Tela Agenda do Servidor.....	63
Figura 27 - Tela de <i>Login</i> do MAAS	65
Figura 28 - Tela Principal do MAAS	66
Figura 29 - Tela Principal do MAAS na Visão Detalhes de um Evento.	66
Figura 30 - Tela Efetuar <i>Login</i> do Aplicativo Móvel.....	67
Figura 31 - Tela principal da aplicação móvel	68

Figura 32 - Tela de compromissos e eventos do dia da aplicação móvel.....	69
Figura 33 - Tela Visualizar Detalhes do Aplicativo Móvel.....	70
Figura 34 - Diagrama de atividades do fluxo principal do servidor web.	73
Figura 35 - Diagrama de atividades da aplicação móvel.....	74

LISTA DE TABELAS

Tabela 1 - Mercado Brasileiro de <i>smartphones</i> em porcentagem anual de vendas.....	26
Tabela 2 – Recursos de Hardware necessários	38
Tabela 3 - Recursos de Software necessários	39
Tabela 4 - Nível de dificuldade apontado por professores e alunos	79

LISTA DE QUADROS

Quadro 1 - Requisito Efetuar <i>Login</i>	43
Quadro 2 - Requisito Efetuar Logout.	43
Quadro 3 - Requisito Editar Agenda da Fonte.	43
Quadro 4 - Requisito Expandir Compromissos e Eventos.	43
Quadro 5 – Requisito Criar Compromissos.....	44
Quadro 6 - Requisito Editar Compromisso.	44
Quadro 7 - Requisito Criar Evento.	44
Quadro 8 - Requisito Editar Evento.	44
Quadro 9 - Autenticar Dispositivo.....	44
Quadro 10- Requisito Efetuar Conexão.....	45
Quadro 11 - Requisito Fornecer Atualização.	45
Quadro 12 - Requisito Acesso Área Restrita.....	45
Quadro 13 - Requisito Origem dos Dados.....	46
Quadro 14 - Requisito Dados Cadastráveis.	46
Quadro 15 - Interface com o Usuário	46
Quadro 16 - Requisito Ambiente Operacional.	46
Quadro 17 - Requisito Atualização Banco de Dados.	46
Quadro 18 - Requisito Idioma.	47
Quadro 19 - Requisito Portabilidade.	47
Quadro 20 - Requisito Conectar ao Servidor.....	47
Quadro 21 - Requisito Apresentar Agenda.....	48
Quadro 22 - Requisito Expandir Compromissos do Dia.	48
Quadro 23 - Requisito Visualizar Compromisso.....	48
Quadro 24 - Requisito Visualizar Evento.....	48
Quadro 25 - Requisito Estilo.	49
Quadro 26 - Requisito Interface com o Usuário Dispositivo Móvel.....	49
Quadro 27 - <i>Script</i> executado pelo aluno.	76
Quadro 28 - <i>Script</i> executado pelo professor	76
Quadro 29 - Cenários de execução dos testes.....	77
Quadro 30 - Perfis dos usuários.....	77
Quadro 31 - Plano de Testes Servidor.	81

Quadro 32 - Casos de Testes do Servidor.....	81
Quadro 33 - Procedimentos de Teste Servidor.....	82
Quadro 34 - Relatório de Testes Incidentes Servidor.....	83
Quadro 35 - Relatório de Resumo de Testes Servidor.	83
Quadro 36 - Plano de Testes Dispositivo Móvel.....	84
Quadro 37 - Casos de Testes do Dispositivo Móvel.	85
Quadro 38 - Procedimentos de Testes do Dispositivo Móvel.	85
Quadro 39 - Relatório de Resumo de Testes do Dispositivo Móvel.	86

LISTA DE ABREVIATURAS E SIGLAS

MAAS	Mobile Academic Appointment System.
ASP.NET	Active Server Pages.
RLE	Rede Local de Ensino.
SDK	Software Development Kit.
UML	Unified Modeling Language.
WP7	Windows Phone 7.
C#	C Sharp (Linguagem de Programação).
WBS	Work Breakdown Structure.
IIS	Internet Information Services.
HTTP	Hypertext Transfer Protocol.
UTFPR	Universidade Tecnológica Federal do Paraná.
XAML	Extensible Application Markup Language.
XML	Extensible Markup Language.

SUMÁRIO

1 INTRODUÇÃO.....	15
1.1 OBJETIVOS	18
1.2 JUSTIFICATIVA.....	18
1.3 MOTIVAÇÃO	19
1.4 APRESENTAÇÃO DO DOCUMENTO.....	20
2 REFERENCIAL BIBLIOGRÁFICO E ESTADO DA ARTE	21
2.1 SITUAÇÃO ACADÊMICA	21
2.2 TECNOLOGIA NO AMBIENTE ACADÊMICO	22
2.3 SMARTPHONE.....	23
2.3.1 Windows Phone	23
2.3.2 Mercado	25
2.3.3 Aplicativo Calendário/Agenda	27
2.4 TECNOLOGIAS UTILIZADAS	28
2.4.1 ASP.NET	29
2.4.2 Windows Server.....	30
2.4.3 Banco de Dados	31
2.4.4 Visual Studio	32
2.4.5 Windows Phone SDK.....	33
2.4.6 Web Service.....	33
2.5 DESIGN DE INTERAÇÃO.....	34
3 METODOLOGIA	36
3.1 TECNOLOGIAS.....	37
4 RECURSOS DE HARDWARE E SOFTWARE	38
4.1 RECURSOS DE HARDWARE.....	38
4.2 RECURSOS DE SOFTWARE	38
5 ESPECIFICAÇÃO DO SOFTWARE	40
5.1 ARQUITETURA DO SISTEMA	40
5.2 APLICAÇÃO SERVIDOR.....	42
5.2.1 Requisitos Funcionais	42
5.2.2 Requisitos não Funcionais	45
5.3 APLICAÇÃO CLIENTE	47
5.3.1 Requisitos Funcionais	47
5.3.2 Requisitos não Funcionais	48
6 DESENVOLVIMENTO	50
6.1 DIAGRAMAS DE CASOS DE USO.....	50
6.1.1 Com relação à aplicação do dispositivo móvel.....	50
6.1.2 Com relação à aplicação do servidor web	52

6.2	MODELO ESTRUTURAL.....	54
6.3	PROTOTIPAÇÃO	59
6.3.1	Com Relação à Aplicação do Dispositivo Móvel.....	59
6.3.2	Com Relação à Aplicação do Servidor Web	62
6.4	SOFTWARE MAAS.....	64
6.4.1	Visão Geral das Interfaces	64
6.4.2	Descrição das Funcionalidades.....	70
6.4.3	Regras de Navegação.....	72
6.5	ENSAIOS DE INTERAÇÃO	74
6.5.1	Resultados do Ensaio de Interação	78
6.6	TESTES DE SISTEMA	79
6.6.1	Com Relação à Aplicação Servidor.....	80
6.6.2	Com Relação à Aplicação Cliente	84
7	CONCLUSÃO.....	87
7.1	TRABALHOS FUTUROS	89
	REFERENCIAS	90
	APÊNDICES	94

1 INTRODUÇÃO

No ambiente acadêmico os alunos se deparam com diversos compromissos e eventos importantes relacionados à faculdade e as suas atividades. Por exemplo, cada disciplina possui o seu planejamento, contendo provas, exercícios, entregas de trabalhos, apresentações, entre outros, cada um em uma data específica. Dependendo do número de disciplinas que o aluno cursa, gerenciar todas essas atividades se torna algo difícil sem o auxílio de uma agenda, podendo levar ao esquecimento de algum compromisso. Além disso, na necessidade do professor em atualizar, de última hora, algumas informações sobre as atividades marcadas, não existe um meio de comunicação eficiente entre professor e aluno para que a nova informação seja entregue de maneira rápida e prática.

O Trabalho de Conclusão de Curso apresentado neste documento consiste no desenvolvimento de um protótipo de software com a função de apoio para gerenciamento de compromissos acadêmicos servindo a alunos, professores e universidade. Este software foi denominado MAAS – MOBILE ACADEMIC APPOINTMENT SYSTEM – e utilizou informações baseadas nas existentes no sistema acadêmico. Futuramente, para melhor eficiência do software, ele poderia ser integrado ao sistema acadêmico das Universidades, eliminando a necessidade de cadastrar dados acadêmicos de diferente maneira.

As dificuldades de comunicação entre os atores da universidade (alunos, professores, coordenadores, entre outros) e o esforço necessário para gestão do tempo e compromissos pelos alunos são problemas reais no âmbito acadêmico nas Universidades. Segundo Puentes e Aquino, (2008), há a necessidade de novos instrumentos de auxílio ao gerenciamento do tempo acadêmico em razão destas dificuldades.

Falhas na dinâmica de comunicação entre alunos e professores acarreta uma ineficiência acadêmica considerável. Um exemplo disso é a mudança de última hora de horário e/ou local de aula. Isso atrapalha o andamento das atividades planejadas pelos professores e tempo de aula é perdido, pois estes têm que se preocupar em avisar todos os alunos, recorrendo, por exemplo, a recados colados em portas de sala de aula. Nesta situação, o professor ainda teria que aguardar a chegada dos alunos para se certificar de que a maioria dos interessados teve acesso à nova informação de última hora.

A Figura 1 mostra um exemplo corriqueiro do aviso de uma ocorrência (neste caso, ausência de um professor) na Universidade Tecnológica Federal do Paraná (UTFPR), destacado na própria sala de aula na qual haveria o compromisso. Deve-se notar que além

desta informação ser tardia, não há segurança de que a fonte seja legítima. A Figura 2, também capturada na UTFPR, por sua vez, mostra novamente um aviso improvisado, na forma de um bilhete que informa a mudança no local de aplicação de uma prova. Este caso provocaria uma considerável perda de tempo de prova aos alunos que precisariam se dirigir à sala original para chegar à informação de aplicação da prova em outro local.

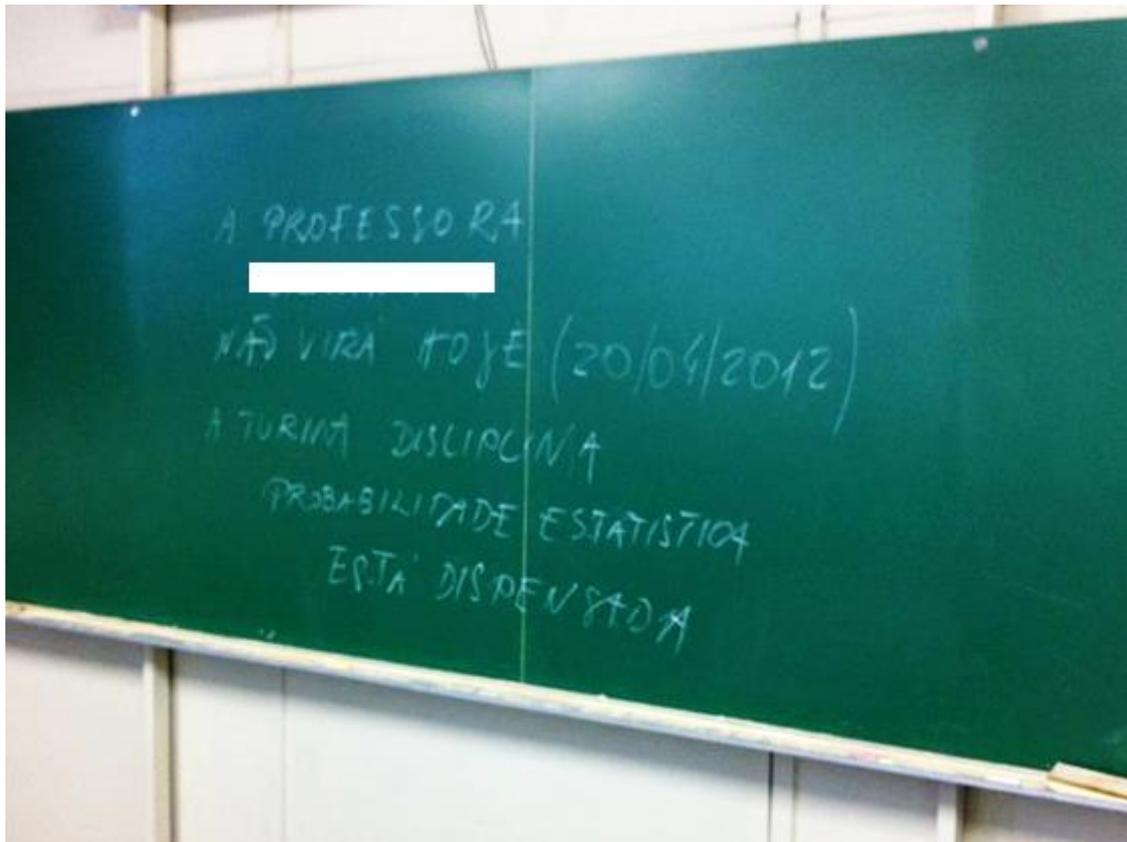


Figura 1- Aviso de ausência de professor no quadro negro.

Com a utilização do software proposto neste trabalho, a divulgação de eventos e lembretes como o possível atraso de um professor, ou a sua possível ausência poderão ser informados aos alunos em tempo real e de forma ágil. Por ser um software desenvolvido para tecnologia móvel (*smartphones*) os alunos seriam notificados assim que uma atualização ocorrer na programação de sua agenda.



Figura 2 - Avisos de Alteração de Local de Prova.

A escolha de trabalhar com tecnologia para dispositivos móveis foi justamente pela mobilidade oferecida, mas além dessa característica existe também a preocupação com o baixo custo para o usuário. Estudos realizados no Brasil relatam que atualmente os *smartphones* podem ser adquiridos a preço de um celular convencional e o crescente uso dessas tecnologias a baixo custo foi comprovado em pesquisas recentes. Segundo PERES, TRIBUG, GERALDO (2010), *smartphones* são "telefones inteligentes" que oferecem capacidades avançadas, geralmente encontradas em computadores maiores. A justificativa ao uso de *smartphones* por jovens universitários ou recém-formados foi atribuída à qualidade de serviços e comodidade proporcionada pelo seu uso.

À medida que a tecnologia móvel foi se desenvolvendo, observou-se o seu valor social, ou a interação social que vem ocorrendo por meio de comunicação móvel. Coelho e Hagenauer (2010) afirmam que em função desse desenvolvimento tecnológico e de sua poderosa capacidade de atender a várias necessidades cotidianas – como prover um volume inimaginável de informações em tempo real – alguns esforços educacionais têm sido feitos no sentido de incorporar essas vantagens que a tecnologia oferece.

Com base na análise das pesquisas realizadas identificou-se a oportunidade da utilização deste meio para apoiar a comunicação na universidade e a gestão de compromissos. Juntamente a isso, alguns conceitos de design de interação, como a computação ubíqua, foram

utilizados para trazer um aplicativo que lembre uma agenda física, mas tenha todas as novidades e recursos oferecidos pela tecnologia móvel.

1.1 OBJETIVOS

O objetivo principal deste trabalho foi criar um software que envolvesse tecnologia de dispositivos móveis para modernizar a comunicação entre membros de uma Universidade sobre eventos e compromissos acadêmicos. Os objetivos específicos do trabalho envolveram:

- Criar um sistema para auxiliar a divulgação de compromissos e eventos acadêmicos, bem como à organização dos compromissos acadêmicos entre alunos e professores que possuem um smartphone;
- Estudar as tecnologias envolvidas para o desenvolvimento do MAAS;
- Criar um sistema de arquitetura cliente – servidor que possa ser expandido para múltiplas plataformas móveis;
- Criar um sistema integrador dos compromissos acadêmicos para uso tanto por alunos quanto por professores;
- Integrar horários de aulas, provenientes do sistema acadêmico ao aplicativo;
- Desenvolver o protótipo proposto seguindo padrões de *Design* de Interação;
- Validar o conceito desenvolvido junto ao conjunto de usuários representantes do público alvo.

1.2 JUSTIFICATIVA

A escolha do tema deste trabalho de conclusão de curso originou-se na observação de dificuldades vivenciadas no ambiente universitário a respeito do controle de compromissos dos estudantes e os impactos negativos que isso gera.

Com o surgimento de diversas tecnologias em torno da internet, que trazem benefícios como comunicação via e-mail e Moodle, os problemas de comunicação entre professor e aluno puderam ser reduzidos. Hoje em dia há, inclusive, a possibilidade do professor interagir com os alunos fora do horário de aula sobre assuntos da disciplina que ministra. Com o advento e grande repercussão e aceitação dos *smartphones*, surgiu a oportunidade de melhorar ainda mais esta comunicação.

Neste trabalho de conclusão de curso propôs-se um software que consiste em uma possível solução para comunicação entre professores e Universidade com os alunos empregando tecnologia móvel, internet e servidores. A principal justificativa desta proposta é o fato de que há uma necessidade real em melhorar a comunicação para fins de divulgação de compromissos e eventos dentro do meio acadêmico. Esta constatação vem da própria experiência dos autores ao longo de sua graduação, do testemunho de seus colegas e professores e da observação de muitas ocorrências em termos de ineficiência ou de falhas de comunicação. Além disso, os sistemas existentes, como e-mail, Moodle¹, mensagens SMS e Messenger e redes sociais, contribuem, mas não atendem as especificidades desta necessidade.

Ainda como justificativa, o software proposto aplica o conceito de computação ubíqua por meio do emprego de plataformas móveis (*smartphones*). Desta forma, a comunicação é agilizada fornecendo acesso à informação em tempo real e com alta disponibilidade. Esta efetividade da comunicação sobre compromissos e eventos é essencial no meio acadêmico, dada a sua multiplicidade de ocorrências.

Finalmente, o desenvolvimento de um software para *smartphone* ao invés de um portal de internet (web site) se justifica pela integração de uma memória local (i.e. no *smartphone*) que habilita seu uso *off-line* permitindo a realização de consultas locais.

Desta maneira, o software desenvolvido neste trabalho possibilita essa melhora na interação entre professor e aluno, garantindo eficiência na distribuição, entrega imediata da informação aos interessados, poupando possíveis perdas de tempo e esforço, caso a informação seja entregue tardiamente.

1.3 MOTIVAÇÃO

Além dos motivos relacionados à melhoria da comunicação no ambiente acadêmico, buscou-se trabalhar em integrar um grande número de conteúdos da área de Tecnologia de Informação para aprofundar os conhecimentos.

O Trabalho de Conclusão de Curso é um instrumento de avaliação final de um curso superior². Sendo assim, a equipe buscou compor um projeto que pudesse convergir boa parte

¹ Moodle – DAINF. Disponível em: <http://moodle.dainf.ct.utfpr.edu.br/>

² Significados. Significado de TCC, 2012, Disponível em: <http://www.significados.com.br/tcc/>. Acesso em: 16 abr. 2013.

das disciplinas cursadas para demonstrar o entendimento dos conteúdos vistos durante o período de graduação.

O desenvolvimento do software MAAS utilizou noções de disciplinas como: Banco de dados, fundamentos de programação, sistemas distribuídos, algoritmos, engenharia de software, análise de sistemas, programação para dispositivos móveis e *design* de interação. Isso possibilitou a revisão e o aprofundamento dos conhecimentos nessas áreas, bem como uma demonstração adequada de compreensão teórica e prática.

O trabalho permitiu, não só aplicar, mas também integrar os conhecimentos adquiridos ao longo do curso, o que tornou o desenvolvimento do trabalho algo gratificante para a equipe.

1.4 APRESENTAÇÃO DO DOCUMENTO

Este documento de Trabalho de Conclusão de Curso está organizado em 7 capítulos incluindo esta introdução, sendo:

- **Capítulo 2 – Referencial Bibliográfico e Estado da Arte:** apresenta o referencial teórico do trabalho incluindo conceitos e análises necessárias para embasamento e justificativa. Conteúdos como contexto acadêmico, *smartphones*, plataformas alvo e tecnologias que aplicam-se ao tipo de desenvolvimento de software pretendido, estão citados neste capítulo.
- **Capítulo 3 - Metodologia:** no tópico de metodologia é apresentada a estratégia de gerenciamento do desenvolvimento do trabalho.
- **Capítulo 4 – Recursos de Hardware e Software:** aqui são citadas todas as ferramentas necessárias no desenvolvimento deste trabalho, e como elas foram adquiridas.
- **Capítulo 5 – Especificação do Software:** apresenta a especificação detalhada do software desenvolvido, suas principais funcionalidades e explicações de seu funcionamento.
- **Capítulo 6 - Desenvolvimento:** este capítulo apresenta o desenvolvimento do software desde a criação dos diagramas até a realização de testes com usuários.
- **Capítulo 7 – Conclusão:** apresenta as conclusões geradas a partir da realização do trabalho e os trabalhos futuros.

2 REFERENCIAL BIBLIOGRÁFICO E ESTADO DA ARTE

Neste capítulo são descritos os resultados da fundamentação teórica realizada para o desenvolvimento deste trabalho. São definidos também alguns conceitos e outras informações, como a problemática envolvida e a solução sugerida, necessárias para a compreensão do projeto como um todo.

2.1 SITUAÇÃO ACADÊMICA

Os alunos universitários estão sempre abarrotados de entregas, trabalhos, provas e diversos outros compromissos relacionados à vida acadêmica, além de precisarem memorizar muitas datas importantes, aulas, salas de aula e horários.

Além de já terem várias datas de compromissos para memorizar, essas datas ainda estão sujeitas a mudanças, que às vezes são realizadas em momentos próximos ao compromisso. Se o aluno não estiver atento a essas datas e suas mudanças, ele pode deixar de desenvolver suas obrigações acadêmicas no prazo definido. Fica difícil manter a organização e lembrar de tantos fatos sem o auxílio de uma ferramenta que gerencie toda essa informação.

Existe também a dificuldade na comunicação do professor com o aluno fora da sala de aula, no que diz respeito a assuntos de urgência, como notificar de última hora que o professor não poderá dar aula em um determinado dia ou avisar os alunos para que imprimam tal material para a aula do dia, para torná-la mais eficiente, entre outros. O mesmo ocorre com a necessidade das Universidades em divulgar seus eventos importantes para parte de seus alunos, visando garantir máxima participação. Com esse problema de comunicação, o aluno pode perder algumas boas oportunidades de aumentar o seu conhecimento e experiência.

Nessa situação, o aluno precisa ter capacidade para gerenciar todas essas informações, bem como o tempo que necessitará para a realização de suas tarefas. A falta de organização sobre eventos e compromissos pode influenciar o resultado acadêmico, reduzindo o desempenho dos estudantes (LEITE, TAMAYO e GÜNTHER, 2003).

Portanto, obter um calendário com dados de todos os compromissos e eventos acadêmicos marcados, o horário e local das aulas todos os dias, e receber atualizações em tempo real, tendem a auxiliar o aluno a se organizar e a ter disponível a qualquer hora esse tipo de informação, agilizando suas atividades.

2.2 TECNOLOGIA NO AMBIENTE ACADÊMICO

A tecnologia de informação tem sido amplamente utilizada dentro do ambiente acadêmico como um meio para apoiar soluções que minimizem as dificuldades enfrentadas por alunos e professores. O artigo de Oishi (2007) apresenta uma pesquisa sobre a utilização de ferramentas colaborativas *online* como um recurso para os professores organizarem suas dezenas de atividades e facilitarem o meio de comunicação com os alunos. Essas ferramentas permitem que várias pessoas trabalhem concomitantemente sem a necessidade de estarem no mesmo local ao mesmo tempo. Esses professores usam aplicativos de propriedade do Google, tais como Google Calendar e Google Docs & Spreadsheets. Ao obter a conta de todos os alunos, os professores poderiam marcar reuniões em seu calendário e adicioná-los para serem notificados. Os alunos também teriam possibilidade de marcar reuniões com professores, além de poderem manipular documentos colaborativamente.

Ao observar as ferramentas citadas anteriormente surgiu o interesse de desenvolver algo mais apropriado às funcionalidades discutidas, com características como criatividade e interatividade, de modo específico, gerando desta maneira um aplicativo que valorize a interação social do indivíduo com seu grupo e, conseqüentemente, com a ferramenta. Para tornar mais eficiente a interação social faz-se necessário embasamento em literaturas de computação ubíqua.

O termo computação ubíqua, neste caso, está inserido no contexto de tecnologias de informação ubíquas. “O termo ubíquo tem como uma de suas origens a era da tecnologia calma (*calm technology*), quando a computação passa a ser subjacente às nossas vidas. Os computadores passam a ser tão naturais, tão sob medida e tão embutidos em todos os locais e nos mais diferentes objetos; eles tendem a se tornar praticamente invisíveis, isto é, nós os utilizaremos quase sem pensar, tal qual utilizamos a energia elétrica atualmente.” (SACCOL, REINHARD, 2007).

A computação ubíqua será aplicada no módulo móvel da aplicação, ou seja, a interação social que opera no dispositivo *smartphone*. Segundo Pellanda (2009), a evolução da Internet móvel tem demonstrado desdobramentos e possibilidades inéditas. Os “olhos da sociedade” presentes nas câmeras conectadas estão cada vez mais eficientes, bem como a informação sobre o redor do cidadão. Talvez um dos principais apontamentos neste momento é o processo de transformação cultural que está intrínseco nessa plataforma tecnológica.

2.3 SMARTPHONE

O termo *smartphone* se refere a uma nova classe de telefones celulares que podem facilitar o acesso à informação e que possuem um processamento com poder computacional significativo (ALMEIDA, 2012). Diferentemente dos celulares comuns, como o próprio nome do dispositivo já explica (“telefone inteligente”) as funcionalidades de um *smartphone* podem ser estendidas por meio da instalação de aplicativos, criados para executar nesse tipo de dispositivo, além de possuir uma interface mais elaborada e um sistema operacional robusto.

De acordo com Tiecher (2011), o *smartphone* possui funcionalidades avançadas que são executadas em seu sistema operacional. Hoje em dia existem diversos sistemas operacionais de celulares, tais como Android, iOS e Symbian, que são abertos para que desenvolvedores criem aplicações para executar nesses telefones. Os “telefones inteligentes” possuem, basicamente, características mínimas de hardware e software, capacidade de conexão com a internet e capacidade de sincronização de dados com um computador pessoal.

2.3.1 Windows Phone

Windows Phone é o sistema operacional móvel da Microsoft que foi lançado em 2010 com o objetivo de substituir a antiga plataforma Windows Mobile. A nova tecnologia não é compatível com a antiga, porém o kernel do Windows Phone é o mesmo que o utilizado no Mobile, isto é, o Windows CE Kernel (*Windows Embedded Compact*) ainda está presente nesse novo sistema operacional para *smartphones*. O kernel (núcleo do sistema) foi lançado no mercado no ano de 1994 e desde então vem sendo aprimorado³. Uma das principais vantagens do uso do Windows Embedded Compact é o fato de se tratar de um genuíno sistema operacional em tempo-real (RTOS - *Real-Time Operating System* é um sistema operacional que garante uma certa capacidade dentro de uma restrição de tempo especificado). Este sistema é amplamente usado em grande número de mercados incluindo desde automação industrial e comercial até medicina. Como consequência, a Windows Phone herda uma grande robustez e riqueza de recursos.

A interface inovadora apresentada pelo Windows Phone é uma das suas características mais impactantes. Seguindo o estilo denominado Metro, difere muito das

³ MARTIN, Fernando. A História da plataforma Mobile. Disponível em: <<http://channel9.msdn.com/posts/A-Histria-da-plataforma-Mobile>>

tecnologias conhecidas iOS e Android. Para o desenvolvimento do protótipo proposto, optou-se por trabalhar com esta plataforma, por motivos de arbítrio em aprofundar o conhecimento nesta tecnologia.

A plataforma Windows Phone possui interface intuitiva e de fácil navegação (AAWP, 2012), o desenvolvimento de aplicativos para esta tecnologia pode ser realizado por meio de duas linguagens de programação, Visual Basic e C#. Optou-se pela linguagem C#, que já foi explorada pela equipe durante o curso de BSI.

O Metro é uma linguagem de *design* criada pela Microsoft baseada em tipografia, cujo objetivo é dar ordem estrutural e forma à comunicação. Esse *design*, como é ilustrado na Figura 3 **Erro! Fonte de referência não encontrada.**, é totalmente original e moderno e foi utilizado não apenas para dispositivos móveis, mas também em outros produtos e serviços, gerando uma interface única e distinta (AAWP, 2012).



Figura 3 - Estilo Metro em Windows Phone.
Fonte: All About Windows Phone (AAWP), 2012

A linguagem de *design* foi escolhida, pois torna a experiência do usuário agradável e fácil, obedecendo aos princípios de usabilidade. Segundo Baranauskas e Oliveira (2008) a linguagem de *design* consiste de elementos de *design* e princípios de composição. Como a linguagem natural, a linguagem de *design* é usada para geração (criação) e interpretação (leitura), ou seja, é usada para se criar objetos que "expressam o que são, o que fazem, como devem ser usados e como contribuem para a experiência.

Assim como o conhecimento das pessoas sobre a linguagem natural é a base para a sua interpretação do que as outras pessoas dizem, uma linguagem de *design* pode ser a base para como as pessoas entendam e interajam com um produto - em nosso caso, o software.

A linguagem Metro também segue os princípios de *design* incluindo: (Microsoft, 2012):

- Tipografia: referente à estrutura e posicionamento dos elementos de uma aplicação.
- Movimento: é o que traz vida à interface. Refere-se à transição nas interfaces do aplicativo. Quando esse princípio é bem atendido, melhora o desempenho percebido de toda a interface pelo usuário.
- Conteúdo sem decoração: é um dos princípios mais originais do Metro. Remover todo o conteúdo extra, desnecessário na interface do usuário. Desta maneira, o conteúdo torna-se o foco principal. Isto é algo muito importante quando se trata de um dispositivo com a tela pequena.
- Honestidade: esse princípio leva em consideração a necessidade de atender as limitações de um dispositivo móvel. Buscando simplicidade na interação, formulários, entre outros.

Outro diferencial da linguagem de *design* está em componentes de interface desenvolvidos exclusivamente para mobile, como cores e temas fortes, navegação de interface e componentes de nível de aplicativo.

Para o desenvolvimento desse projeto, essa linguagem de *design*, além de tornar a experiência do usuário agradável e fácil, obedecendo aos princípios de usabilidade, proporcionou ganhos em questão de tempo. Neste caso o *layout* das telas segue um padrão baseado em Metro, desta forma não foi gasto tempo para desenvolvimento de uma linguagem de *design* da aplicação.

2.3.2 Mercado

O mercado de *smartphones* no Brasil, ainda se encontra na fase inicial, mas vem crescendo aceleradamente a cada ano. Segundo o site de notícias de TI, *Computerworld*, o

número de vendas de *smartphones* no período de 2012 foi 78% maior que o ano anterior, o que significa 16 milhões desses aparelhos vendidos⁴.

A queda do preço desse produto é um fator de extrema relevância na garantia do seu maior acesso pelas classes médias e baixas, que compõem mais da metade da população brasileira (TIECHER, 2011). Recentemente o governo brasileiro decretou que as contribuições de PIS (Programa de Integração Social) e Cofins (Contribuição para Financiamento da Seguridade Social) serão desonerados para *smartphones* produzidos no Brasil. Esta medida implica uma redução em 9,25% o preço do produto⁵. O governo havia estimado 65 milhões de *smartphones* no Brasil no ano de 2014. Com a medida, essa previsão sobe para 130 milhões⁵.

O Mercado brasileiro para o Windows Phone aumentou em 10% do ano de 2011 para 2012, ficando em terceiro colocado no mercado brasileiro em relação à outros sistemas operacionais⁶.

Com o crescente mercado dessa tecnologia, evidencia-se uma oportunidade em desenvolver aplicativos para esse sistema operacional. A tabela 1 apresenta a porcentagem de *smartphones* vendidos com tecnologia Windows Phone no período de 2011 e 2012.

Tabela 1 - Mercado Brasileiro de *smartphones* em porcentagem anual de vendas

<i>Marca</i>	<i>2011</i>	<i>2012</i>	<i>Diferença</i>
Symbian	71.7%	22.0%	-49.7
Research In Motion	4.0%	4.5%	0.5
iOS	3.2%	7.5%	4.3
Windows Phone	4.6%	14.9%	10.3
Android	14.0%	46.8%	32.8
Bada	0.4%	3.3%	2.9
Outros	2.2%	1.1%	-1.1

Fonte: VNext (2012)⁵.

É importante ressaltar que a escolha por trabalhar com Windows Phone foi por motivos diferentes no que diz respeito à quantidade de usuários de *smartphones* existente nos dias de hoje. Apesar de não ser a plataforma que possui mais usuários no Brasil, o software proposto neste trabalho não teria sentido se fosse destinado à apenas um sistema operacional

⁴ Notícia veiculada no Jornal Folha de São Paulo na data de 13/09/2012.

⁵ Notícia veiculada no Jornal Gazeta do Povo na seção Tecnologia na data de 09/04/2013.

⁶ VNEXT. Disponível em: <http://vnext.com.br/14206/windows-phone-cresce-10-em-um-ano-e-ja-e-o-terceiro-colocado-em-marketshare-no-brasil>.

específico de smartphone, visto que em qualquer ambiente existem usuários com *smartphones* de diferentes plataformas.

2.3.3 Aplicativo Calendário/Agenda

Para melhor entendimento de um aplicativo que represente uma agenda, foram analisados alguns softwares de calendário existentes. Atualmente, qualquer dispositivo móvel possui algum tipo de calendário, no qual se podem adicionar eventos para serem notificados algum tempo antes de sua ocorrência. Com o aparecimento dos *smartphones*, as aplicações de calendário têm se tornado cada vez mais sofisticadas, com melhora na interface, número de ações que podem ser executadas e, até mesmo, com a possibilidade de compartilhar eventos com outros usuários.

Na maioria dos *smartphones* a aplicação de calendário possui possibilidade de visualização por dia, semana e mês, com uma interface simples. O usuário pode criar e editar eventos definindo seu título, horário e localização. Em uma breve análise de alguns *smartphones*, tal como Samsung Galaxy 5 e Nokia X2-01, com respectivos sistemas operacionais Android 2.2 e Symbian S40, notou-se que em ambos a interface dos calendários são simples, podendo o usuário visualizar os eventos por dia, mês ou por lista de eventos. As principais funções são a criação e edição de eventos com possibilidade de definir o tempo de antecedência para ser notificado. Esses eventos podem ser repetidos diariamente, semanalmente ou mensalmente. Nos dispositivos que possuem conexão com internet, há ainda a possibilidade de adicionar participantes aos eventos, por e-mail.

Existem alguns calendários que permitem compartilhamento de seu conteúdo com demais usuários, como o Outlook e o Google Agenda. O sistema de e-mail Outlook possui um calendário para cada usuário que pode ser parcialmente visto por outros. Isso auxilia no agendamento de reuniões sem a ocorrência de conflitos com outros compromissos. Essa ferramenta é bastante utilizada no ambiente corporativo, pois traz diversos benefícios relacionados à colaboração e à disseminação de informações importantes. Semelhante a esse sistema, a agenda do Google também oferece a opção de visualizar a agenda de outros usuários, contanto que haja permissão para tal acesso.

Para dispositivos móveis algumas soluções encontradas no mercado foram os aplicativos para a plataforma Android no Google Play, tais como: My Class Schedule ilustrado na Figura 4 e School Assistant, ambos são aplicativos que permitem ao usuário

cadastrar os horários de aulas, disciplinas, professores e a visão do calendário é por dia e semana.

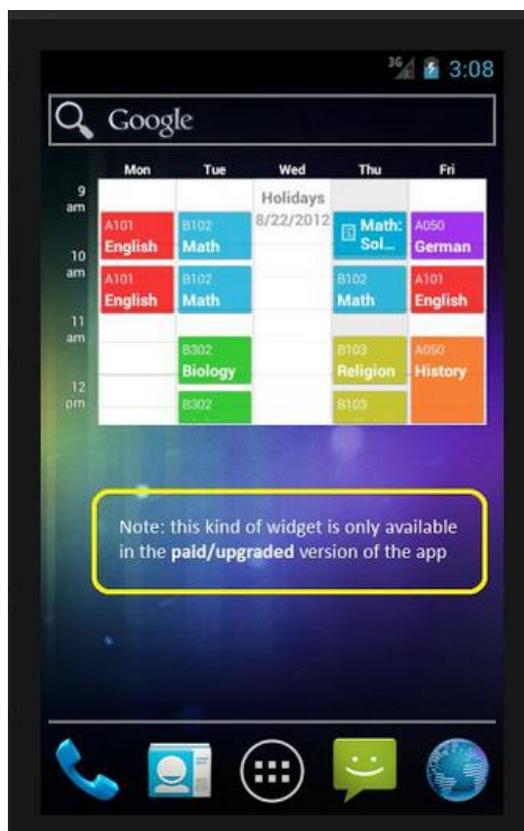


Figura 4 - Tela My Class Schedule.
Fonte: Google Play (GOOGLE, 2012)

Analisando os sistemas existentes pode-se observar que eles não atendem adequadamente as necessidades das Universidades. O compartilhamento de agendas não é uma tarefa fácil, principalmente quando se trata de um grande número de pessoas. Além disso, pode-se misturar questões pessoais com profissionais e acadêmicas, o que seria inconveniente. Estes sistemas também não são centralizados e não podem ser integrados a outros sistemas como bases de dados das Universidades. Em geral, eles são mais indicados para uso individual, com alguns poucos compartilhamentos, do que como um sistema de comunicação de compromissos amplo para toda uma instituição.

2.4 TECNOLOGIAS UTILIZADAS

Após definição do projeto, pôde-se buscar referências sobre as tecnologias que foram utilizadas no desenvolvimento do software MAAS.



Figura 5 - Tecnologias Utilizadas.

2.4.1 ASP.NET

O desenvolvimento de aplicações para a Internet (aplicações Web) pode ser feito utilizando várias alternativas de tecnologias disponíveis no mercado como Java, Ruby e ASP. Cada uma destas tecnologias oferece recursos para a construção de aplicações que podem ser acessadas remotamente pela Internet empregando *Browsers*. A tecnologia escolhida neste trabalho foi ASP.NET, pois é bem difundida e possui vários sites e fóruns de suporte ao desenvolvedor, por exemplo, StakOverflow, MSDN entre outros⁷. ASP.NET é a plataforma da Microsoft para o desenvolvimento de aplicações Web e é o sucessor da tecnologia ASP (Active Server Pages). É um componente do IIS (Internet Information Services) que permite por meio de uma linguagem de programação integrada na .NET Framework, criar páginas dinâmicas. Não é nem uma linguagem de programação, nem um servidor web como IIS ou Apache⁸.

Para a melhor compreensão do que é ASP.NET, precisa-se complementar com outro conceito, o de *framework*: "*framework* é um conjunto de classes que colaboram para realizar uma responsabilidade para um domínio de um subsistema da aplicação." (FAYAD,1997).

O ASP.NET é baseado no Framework .NET, herdando todas as suas características. Por isso, como qualquer aplicação .NET, as aplicações para essa plataforma podem ser

⁷ STACKOVERFLOW, Fórum para Programadores. Disponível em: <http://stackoverflow.com/>.

⁸ WIKIPEDIA, Asp.Net. Disponível em: <https://pt.wikipedia.org/wiki/ASP.NET>.

escritas em várias linguagens, como C# e Visual Basic. O .NET Framework é uma plataforma de desenvolvimento popular para a criação de aplicativos para Windows⁹.

Embora se possa desenvolver aplicações ASP.NET utilizando somente um editor de texto e o compilador .NET, o ambiente de desenvolvimento mais comum das aplicações ASP.NET é o Visual Studio .NET já que possui algumas características que facilitam o trabalho do programador, como os componentes visuais para criação de formulários de páginas web¹⁰.

2.4.2 Windows Server

No desenvolvimento de aplicações seguindo o modelo cliente-servidor, há a necessidade de hospedar uma parte do sistema em um computador servidor. Esta parte do sistema (subsistema servidor) irá fornecer acesso a serviços via uma interface gráfica com usuários e via protocolo HTTP (Hypertext Transfer Protocol) com aplicações clientes.

O computador servidor deve, por consequência, ser equipado com um sistema operacional apropriado, como é o caso do Windows Server.

Windows Server faz parte de uma série de sistemas operacionais desenvolvidos para servidores pela Microsoft Corporation. Os servidores Windows são versões mais poderosas do que o sistema operacional para desktop e são projetados para lidar de forma mais eficiente no mundo corporativo, de redes, Internet / intranet hospedagem, bases de dados, corporativo em escala de mensagens e funções similares¹¹.

O nome Windows Server fez sua estreia com o lançamento do Windows Server 2003 e continua com a versão atual, o Windows Server 2008 R2, que partilha a sua base de código com o Windows 7. Windows Server 2008 R2, que foi instalado no servidor do MAAS, estreou em outubro de 2009 e, atualmente, tem o seu primeiro *service pack* (SP1).

⁹ MICROSOFT, Developer Network . Disponível em: <http://msdn.microsoft.com/pt-br/vstudio/aa496123>.

¹⁰ MICROSOFT, Visual Studio 2012. Disponível em: <http://www.microsoft.com/visualstudio/ptb/products/visual-studio-ultimate-2012>.

¹¹ WEBOPEDIA ,Windows Server, Disponível em: http://www.webopedia.com/TERM/W/Windows_Server.htm

2.4.3 Banco de Dados

Segundo Guimarães (2008), um banco de dados ou base de dados é uma coleção de dados ou informações relacionadas entre si. Elas representam aspectos do mundo real com significado próprio e que se deseja armazenar para uso futuro.

Ao se projetar um banco de dados deve-se ter em mente um conjunto de aplicações que se deseja fazer sobre os dados, pois elas determinam o uso principal que se deseja fazer do banco de dados (GUIMARÃES, 2008). Bancos de dados podem variar seu tamanho e complexidade conforme o que se deseja armazenar e para que será utilizado.

Para a construção, manutenção e segurança de uma base de dados utilizam-se Sistemas Gerenciadores de Bancos de Dados (SGBD). Estes são sistemas que reúnem e mantêm organizadas coleções de informações que se relacionam, ou seja, gerencia os bancos de dados que o usuário criou. Para este trabalho utilizou-se dois sistemas gerenciadores de banco de dados (SGBD), o SQL Server e SQL Server Compact Edition, estão descritos a seguir.

2.4.3.1 SQL Server

É um SGBD criado pela Microsoft em parceria com a Sybase. Foi lançado em 1988 e, desde então, vem sendo aprimorado para atender as requisições do mercado empresarial. Possui ferramentas de administração com interface gráfica excelentes, que possibilitam um gerenciamento mais fácil e produtivo, além de ser mais fácil de administrar e de programar do que outros bancos de dados como o Oracle (PRADO, 2012).

Para este trabalho, optou-se pelo uso do SQL Server 2008, penúltima versão lançada para armazenamento de dados no servidor da solução proposta. A escolha por este SGBD foi motivada pela sua alta qualidade e facilidade de uso, além de sua fácil disponibilidade na UTFPR via programa DreamSpark.

2.4.3.2 SQL Server Compact Edition

O SQL Server CE foi criado pela Microsoft para aplicações que são executadas tanto para *smartphones* quanto para desktop. Trata-se de um SGBD compacto desenvolvido para

atender as limitações de recursos dos dispositivos móveis. No Windows Phone esse é o local de armazenamento usado pela aplicação móvel do protótipo, juntamente com o framework LINQ (*Language Integrated Query*), que possibilita realizar ações no banco de dados utilizando a linguagem de programação da aplicação.

2.4.4 Visual Studio

O Visual Studio foi o ambiente de desenvolvimento de software utilizado na implementação do sistema proposto. Esta ferramenta foi desenvolvida pela Microsoft, dedicado às linguagens Visual Basic, C, C++, C# e J#, sendo sua última versão a 2012. Sua interface foi desenhada em WPF (*Windows Presentation Foundation*) que facilita e muito os recursos da IDE como customização total da janela de Start do programa e o gerenciamento de qualquer janela existente. A Figura 6 ilustra a última versão da ferramenta.

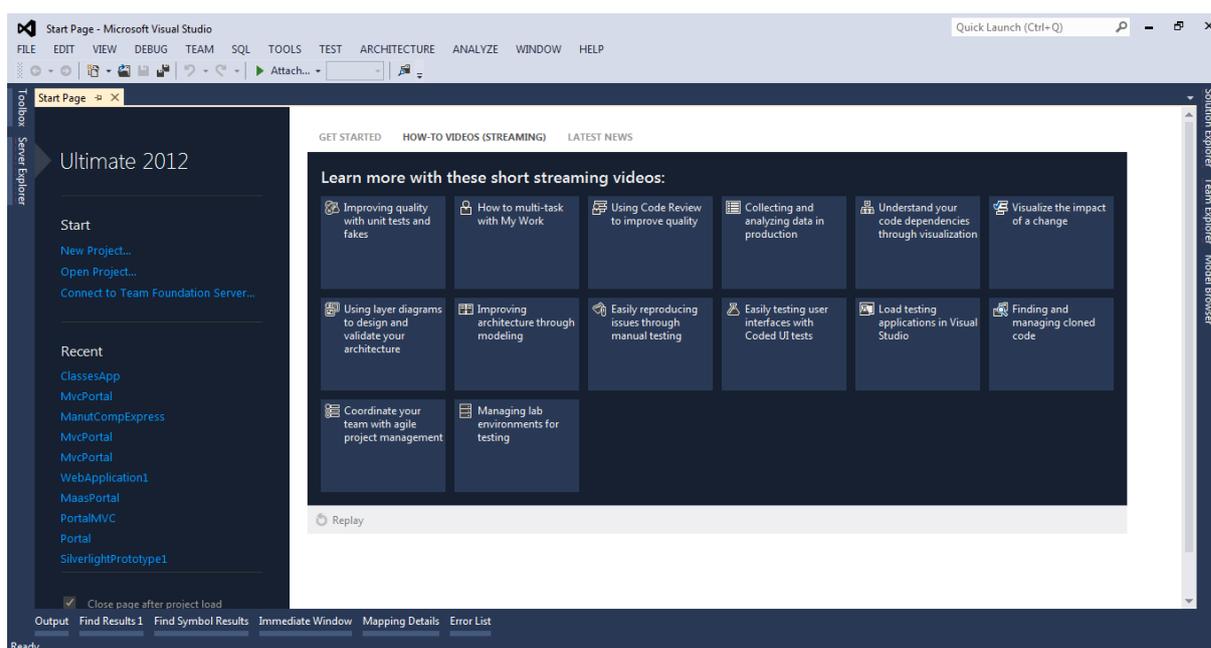


Figura 6 – Tela inicial do Visual Studio 2012.

O Visual Studio foi utilizado para o desenvolvimento de todo o sistema proposto, tanto do subsistema embarcado para smartphone quanto do subsistema servidor. A escolha desta ferramenta foi motivada, basicamente, pela sua compatibilidade com a plataforma Windows, pela compatibilidade com a linguagem C# e pela produtividade que esta ferramenta oferece aos desenvolvedores.

2.4.5 Windows Phone SDK

O Windows Phone SDK 7.1 é um kit de desenvolvimento que provê todas as ferramentas necessárias para o desenvolvimento de uma aplicação para a plataforma Windows Phone¹². A versão 7.1 foi a utilizada no desenvolvimento do protótipo, o kit possui um emulador de smartphone que simula um aparelho executando a aplicação criada pelo desenvolvedor, o que facilita a implementação, pois o desenvolvedor não precisa testar a aplicação diretamente no aparelho, eliminando a necessidade de possuir um aparelho com o sistema operacional Windows Phone.

2.4.6 Web Service

O Web Service é um sistema que suporta a interação entre aplicações distintas sobre uma rede¹³. A comunicação ocorre por meio da linguagem XML. Essa é a tecnologia utilizada para realizar a comunicação entre a aplicação do smartphone com a aplicação servidor.

2.4.6.1 JSON

JSON (JavaScript Object Notation) é um formato de troca de dados leve e de fácil compreensão tanto humana, isto é, próximo a linguagem natural, quanto de máquina (JSON, 2013). O JSON tem uma estrutura simples, ou seja, não utiliza grande processamento para sua interpretação. É baseado em um subconjunto da linguagem de programação JavaScript e possui um formato de texto que é completamente independente do idioma, mas usa convenções que são familiares aos programadores de diversas linguagens de programação (JSON, 2013). Essas propriedades fazem JSON uma linguagem de troca de dados ideal para o sistema proposto, que pode conter um volume considerável de troca de dados. JSON é construído em duas estruturas, conforme ilustrado na Figura 7 e Figura 8.

O MAAS utiliza algumas das estruturas de dados do JSON para realizar a troca de dados: *object* (objeto) e *array* (vetor) que assumem as formas:

¹² TEZINI, Gilberto. Desenvolvimento Windows Phone – Primeiros Passos. 13 Jul. 2012. Disponível em: <http://windowsphonebrasil.com.br/desenvolvimento-windows-phone-primeiros-passos/>.

¹³ W3SCHOOLS . Web Services. Disponível em: < http://www.w3schools.com/webservices/ws_intro.asp>.

- **Objeto:** este é representado em uma coleção de pares, sendo nome e valor o par que representa um objeto como pode ser observado na Figura 7, em várias linguagens de programação, esta estrutura é percebida como um objeto, *struct*, dicionário.

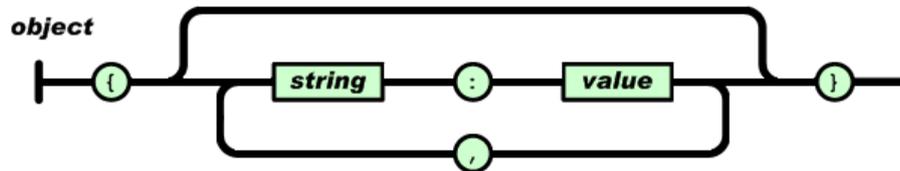


Figura 7 – Estrutura do Objeto JSON
Fonte: JSON (2013).

- **Vetor:** uma lista ordenada de valores. Na maioria das linguagens de programação, esta lista é percebida como uma matriz, vetor, lista sequencial. Um vetor é uma lista de objetos que assume a forma representada pela Figura 8.

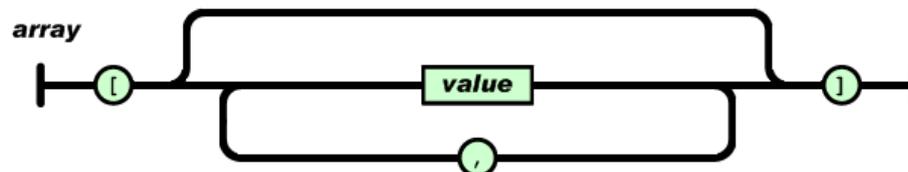


Figura 8 - Estrutura do Array JSON
Fonte: JSON (2013).

Entende-se que objeto é um conjunto desordenado de pares nome e valor (nome:valor), um objeto começa com {(chave esquerda) e termina com } (chave direita).

Cada nome é seguido por “:” (dois pontos) e os pares nome e valor são separados por “,” (vírgula), sendo um vetor uma lista de objetos.

2.5 DESIGN DE INTERAÇÃO

Design de Interação é o ramo do *design* da experiência do usuário que esclarece o relacionamento entre as pessoas e os artefatos interativos que elas usam (PREECE, 2008). Seu foco está na definição de diálogos de uma forma geral que ocorrem entre as pessoas e seus artefatos interativos dos mais variados tipos, desde computadores e dispositivos móveis até utensílios comuns. Dentro do estudo de *design* de interação existem vários conceitos e teorias para aprimorar a experiência do usuário (PREECE, 2008). Dentre eles destacamos dois, os

quais são “usabilidade” e “ensaios de interação” que foram usados para validar a interface do software proposto.

Usabilidade é geralmente considerada o fator que assegura que os produtos são fáceis de usar, eficientes e agradáveis (pela perspectiva do usuário), o que implica em otimizar as interações estabelecidas pelas pessoas com produtos interativos, de modo a permitir que realizem suas atividades no trabalho, na escola e em casa (PREECE, 2008). A usabilidade possui princípios para nortear o desenvolvimento da interação com o usuário, os mais conhecidos são visibilidade, *feedback*, restrições, mapeamento, consistência e *affordance*. (PREECE, 2008)

Ensaio de interação consiste em uma simulação de uso do sistema da qual participam pessoas representativas de sua população-alvo, tentando fazer tarefas típicas de suas atividades, com uma versão do sistema pretendido. Sua preparação requer um trabalho detalhado de reconhecimento do usuário-alvo e de sua tarefa típica para a composição dos cenários e *scripts* que serão aplicados durante a realização dos testes. (LABIUTIL, 2013).

Os testes citados à cima destinam-se a validar ou não as hipóteses de mau funcionamento de determinadas funções, quando utilizadas em uma situação real de trabalho (MOÇO, 1996). Eles devem ser aplicados em uma avaliação de software, para que o analista observe o usuário interagindo com o software. Os testes desempenham o mesmo papel que a análise da atividade tem em uma análise ergonômica, isto é, com a aplicação dos testes, observa-se e detecta-se como o usuário elabora e executa sua tarefa com o auxílio do aplicativo (MOÇO, 1996).

3 METODOLOGIA

Este Trabalho de Conclusão de Curso teve como objetivo o desenvolvimento de um sistema de informação que utilizasse tecnologia Web e tecnologia para dispositivos móveis, mais especificamente para *smartphones*, com a função de melhorar a comunicação existente no ambiente acadêmico.

Para alcançar este objetivo foram definidos alguns pacotes de trabalhos e atividades que estão listados e descritos a seguir:

- **Análise dos produtos semelhantes ao proposto:** foi necessário um aprofundamento no estudo nas tecnologias existentes que são semelhantes à proposta, para que funções já existentes não fossem reinventadas e para que se pudesse comparar o protótipo com as soluções existentes;
- **Estudo do público alvo:** esta foi uma das partes mais relevantes do projeto, pois deve haver aceitação da solução oferecida pelos usuários do sistema. Os maiores interessados no produto seriam, inicialmente, alunos e professores, mas é relevante considerar perfis diferenciados.
- **Planejamento do projeto:** nesta etapa foram realizadas a definição da estrutura de projeto, alcance do trabalho a ser realizado, detalhamento dos recursos, custos, cronograma, escopo de projeto e levantamento de riscos.
- **Análise do sistema:** esta fase consistiu o levantamento de requisitos, modelagem e *design* do sistema utilizando UML, modelagem dos dados e o detalhamento da arquitetura do sistema. Nesta fase também foram estudados os princípios de *design* de interação e computação ubíqua aplicados ao software. As atividades de trabalho que competem à área de *design* de interação dizem respeito à identificação das necessidades em estabelecer requisitos referentes ao processo de interação e à interface. Isso envolve o desenvolvimento de *designs* alternativos para tornar a interface mais atrativa ao usuário e mais fácil de utilizar, buscando inspiração na linguagem de *design* Metro.
- **Implementação:** esta fase envolve a construção, propriamente dita, do sistema de informação proposto. O software foi desenvolvido em linguagem C#, utilizando .NET Framework 4.5, em sistema operacional para dispositivos

móveis Windows Phone 7. Os dados serão gerenciados pelo sistema gerenciador de banco de dados Microsoft SQL Server.

- **Garantia de qualidade:** para garantir a qualidade do software, foi proposta a realização de testes de sistema e de aceitação do usuário. A qualidade da interface e interação com o usuário será medida por meio de ensaios de interação, que serão aplicados a uma pequena amostra do público alvo.

3.1 TECNOLOGIAS

Para o desenvolvimento do sistema proposto foi definido um conjunto de tecnologias que oferecessem os meios necessários para implementar as funcionalidades desejadas. Como a arquitetura do sistema adota um modelo cliente-servidor, foram definidas tecnologias para os dois subsistemas. Nos dois subsistemas optou-se por tecnologias da Microsoft, embora tecnologias de outras fontes pudessem ser empregadas também. A escolha pelas tecnologias Microsoft teve como principais motivações (i) a disponibilidade de todas as ferramentas necessárias pelo programa DreamSpark do Departamento Acadêmico de Informática (DAINF) da UTFPR, (ii) a familiaridade dos autores no uso destas ferramentas e (iii) a percepção dos autores de maior produtividade empregando estas tecnologias. As principais tecnologias adotadas estão detalhadas na seção 4.1 e 4.2.

4 RECURSOS DE HARDWARE E SOFTWARE

Esta seção descreve os recursos de hardware e software utilizados na realização deste projeto, e a maneira como foram adquiridos.

4.1 RECURSOS DE HARDWARE

Para a realização do projeto foram utilizados dois *notebooks* para suportar softwares de desenvolvimento, um servidor com suporte ASP.NET para hospedar aplicação web e banco de dados e um dispositivo móvel para realizar testes do sistema finalizado. A Tabela 2 detalha a configuração de cada um desses recursos.

Tabela 2 – Recursos de Hardware necessários

<i>Hardware</i>	<i>Processador</i>	<i>Memória</i>	<i>Licença de S.O.</i>
Notebook 1	Core i3	4GB	Windows 7 Ultimate
Notebook 2	Core i5	8GB	Windows 7 Ultimate
Servidor	Core 2 DUO	4GB	Windows Server 2008 R2
Dispositivo móvel	Qualcomm	512MB	Windows Phone 7.5

O servidor foi fornecido pelo Cion Ayres do Nascimento, responsável pelo setor RLE (Rede Local de Ensino) do DAINF-CT/UTFPR. O *smartphone* com Sistema Operacional Windows Phone foi fornecido para teste pelo orientador do trabalho Professor Doutor Paulo Cezar Stadzisz.

4.2 RECURSOS DE SOFTWARE

Os softwares utilizados para o desenvolvimento do MAAS encontram-se citados e descritos na Tabela 3.

Tabela 3 - Recursos de Software necessários

<i>Licença de SW</i>	<i>Versão</i>	<i>Utilização</i>
Windows Phone SDK	7.1	Desenvolvimento de aplicativo WP
MS Visual Studio	2010 Express for WP	Desenvolvimento de aplicativo WP
SQL Server	2008 R2 <i>Developer</i>	Armazenar dados
Astah	<i>Community</i>	Diagramação do sistema
Windows Server	2008 R2	Para suportar o servidor do MAAS

Pela opção de trabalhar com o Sistema Operacional Mobile Windows Phone 7, o software MAAS foi implementado no ambiente de desenvolvimento Visual Studio 2010, que inclui o kit de desenvolvimento Windows Phone SDK 7.1, e Visual Studio 2012 utilizado no desenvolvimento do servidor web.

A linguagem de programação escolhida foi C# por ser uma linguagem robusta com grande suporte de recursos e por já ter sido explorada pela equipe durante o curso de BSI.

Para o servidor foi utilizado o sistema operacional Windows Server 2008 R2 e que hospeda a aplicação web e mantém um banco de dados SQL Server 2008 Express.

Todos os produtos da Microsoft listados na Tabela 3, foram disponibilizados pelo Programa denominado DreamSpark¹⁴, proporcionado pela empresa. Esse programa visa auxiliar a educação técnica, fornecendo acesso gratuito a softwares da Microsoft para fins de aprendizado, ensino e pesquisa.

¹⁴ Microsoft DreamSpark. Disponível em: <<https://www.dreamspark.com>>

5 ESPECIFICAÇÃO DO SOFTWARE

O MAAS é composto por dois softwares, o servidor web e a aplicação móvel, ajustados em uma arquitetura cliente-servidor. O servidor comporta um portal web que faz a interface com o professor. Nesse portal o professor efetua as ações de cadastro, edição, cancelamento e visualização de compromissos e atividades. É no servidor que são armazenados os compromissos relacionados às disciplinas e eventos oferecidos pela Universidade. A aplicação móvel é a parte do software que representa a agenda acadêmica de cada aluno. A partir de seu número de matrícula, seus compromissos e eventos acadêmicos são mostrados em seu *smartphone*. Esses subsistemas fazem conexão via uma tecnologia denominada Webservices. A aplicação no *smartphone* faz requisições ao servidor web das informações relevantes para o seu usuário cadastrado periodicamente. Desta maneira, as novas informações são mandadas ao *smartphone* para que a agenda do aluno seja atualizada.

A próxima subseção apresenta a arquitetura geral do sistema e as demais subseções apresentam os requisitos levantados para cada um dos subsistemas do MAAS

5.1 ARQUITETURA DO SISTEMA

A arquitetura utilizada no sistema é baseada no padrão MVC (*Model View Controller*). Este fornece uma maneira de dividir o projeto em três camadas interligadas, de maneira que a camada de Visão que é a interface com o usuário se comunica com o a de Controle que, por sua vez, faz a interface com a camada de Modelo, que delimita toda a camada de acesso a dados. A Figura 9 é uma representação desse modelo e é uma implementação do modelo MVC no projeto MAAS.

A arquitetura MVC não é nova e foi originalmente desenvolvida para mapear as tarefas tradicionais de entrada, processamento e saída para o modelo de interação com o usuário (MACORATTI, 2013). A utilização do padrão MVC torna mais fácil mapear esses conceitos no domínio de aplicações Web multicamadas.

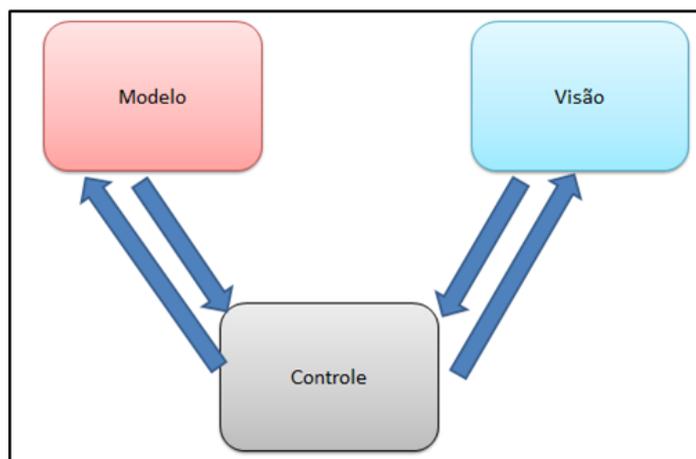


Figura 9 - Arquitetura MVC

A estrutura do software foi desenvolvida no modelo MVC para proporcionar melhor estruturação e maior portabilidade para outras plataformas. No caso da aplicação móvel é de extrema importância ter essa facilidade de portabilidade para poder oferecer esse serviço para o maior número de interessados, visto que existem diversas outras plataformas de *smartphones* que são amplamente utilizadas.

O MAAS foi projetado em dois módulos envolvendo o aplicativo móvel e o servidor web que engloba um aplicativo web e um serviço web para acessar os dados e retornar ao dispositivo móvel as atualizações. O acesso ao aplicativo web é feito via *browser* pelos usuários cadastrados como professores e responsáveis pela divulgação de eventos da Universidade. A Figura 10 apresenta uma visão sistêmica do software desenvolvido.

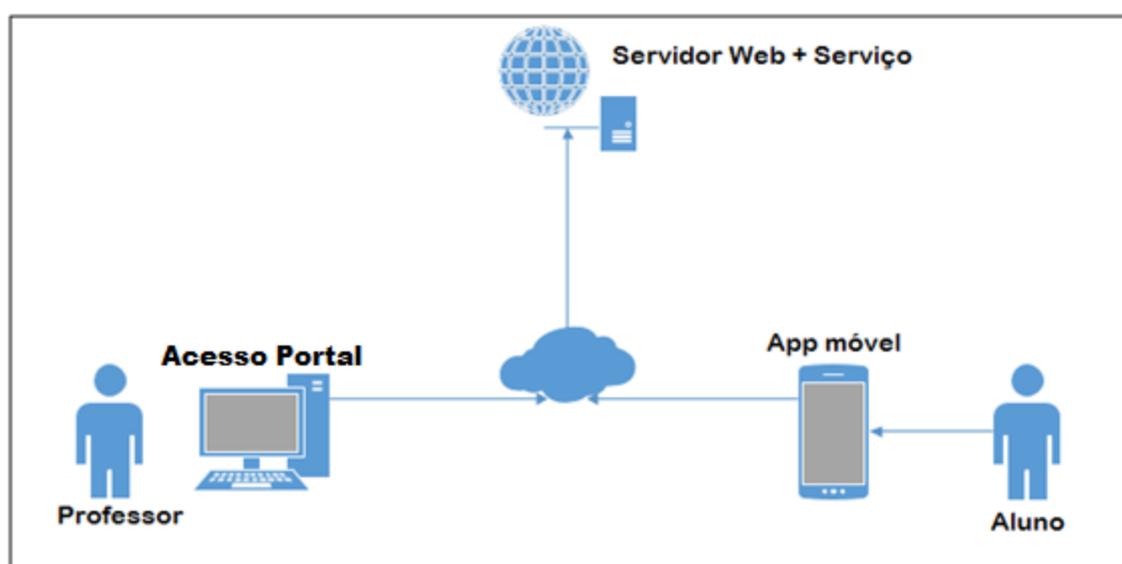


Figura 10 - Arquitetura do MAAS

O serviço web emprega o framework WCF (Windows Communication Foundation), reconhecida por ser uma ferramenta muito utilizada para implementar uma arquitetura orientada a serviços (SOA). Ele foi projetado utilizando princípios de arquitetura orientada a serviços para apoiar a computação distribuída, em que os serviços têm consumidores remotos. Os clientes podem consumir vários serviços, e estes os serviços, por sua vez, podem ser consumidos por vários clientes¹⁵.

O servidor web recebe do aplicativo móvel uma requisição de atualização via protocolo SOAP (Simple Object Access Protocol). Este é um protocolo para troca de informações estruturadas em uma plataforma descentralizada e distribuída usado para acessar o webservice do servidor MAAS. Ele se baseia na Linguagem de Marcação Extensível (XML) para seu formato de mensagem, e normalmente baseia-se em outros protocolos da camada de aplicação, mais notavelmente em Chamada de Procedimento Remoto (RPC) e Protocolo de Transferência de Hipertexto (HTTP), para negociação e transmissão de mensagens¹⁶. No caso do MAAS, o dispositivo móvel representa um dos clientes. O outro tipo de cliente é o navegador browser que é usado para acessar o servidor, este por sua vez encontra-se hospedado em um dos servidores da Rede Local de Ensino (RLE), nas dependências da UTFPR.

5.2 APLICAÇÃO SERVIDOR

As seções seguintes apresentam detalhes da especificação que dizem respeito da parte do servidor web do software que interage com o professor e membros da Universidade. As subseções à seguir iniciam a descrição dos requisitos funcionais.

5.2.1 Requisitos Funcionais

Requisitos são exigências ou restrições estabelecidas por clientes e usuários que definem as diversas funções e propriedades do sistema. Tradicionalmente, os requisitos de software são separados em requisitos funcionais e não funcionais (SCHEUNEMANN, 2013).

¹⁵ WIKIPEDIA. *Windows Communication Foundation*. 2013. Disponível em: <http://en.wikipedia.org/wiki/Windows_Communication_Foundation>.

¹⁶ W3SCHOOLS, SOAP. 2013. Disponível em: <http://www.w3schools.com/soap/soap_intro.asp>.

Nas seções seguintes estão especificados os requisitos funcionais e não funcionais referentes à aplicação servidor do software MAAS.

Os requisitos funcionais envolvem a descrição das diversas funções que clientes e usuários querem ou precisam que o software ofereça. Eles definem as funcionalidades desejadas do *software*. O termo “função” é usado no sentido genérico de operação que pode ser realizada pelo sistema, sejam por meio de comandos dos usuários, seja pela ocorrência de eventos internos ou externos ao sistema. (SCHEUNEMANN, 2013). Nesta seção são apresentados os requisitos funcionais do servidor web.

Quadro 1 - Requisito Efetuar *Login*.

REQ001 – Efetuar <i>Login</i>			
SOLICITANTE:	Letícia Ueda Vanessa Maria da Silva	REQ. ORIGEM:	
TIPO DO REQUISITO:	Funcional	IMPACTO NA ARQUITETURA:	Alto
DESCRIÇÃO:	O software do servidor web deverá permitir identificar o utilizador do sistema. Os utilizadores do sistema serão os professores da instituição.		

Fonte: Autoria Própria.

Quadro 2 - Requisito Efetuar *Logout*.

REQ001.1 – Efetuar <i>Logout</i>			
SOLICITANTE:	Letícia Ueda Vanessa Maria da Silva	REQ. ORIGEM:	REQ001
TIPO DO REQUISITO:	Funcional	IMPACTO NA ARQUITETURA:	Alto
DESCRIÇÃO:	O software do servidor web deverá permitir ao usuário deslogar de sua conta do servidor ao finalizar as ações desejadas.		

Fonte: Autoria Própria.

Quadro 3 - Requisito Editar *Agenda da Fonte*.

REQ002 – Editar <i>Agenda da Fonte</i>			
SOLICITANTE:	Letícia Ueda Vanessa Maria da Silva	REQ. ORIGEM:	
TIPO DO REQUISITO:	Funcional	IMPACTO NA ARQUITETURA:	Alto
DESCRIÇÃO:	O software do servidor web deverá permitir que o usuário realize múltiplas atividades em sua agenda, com relação à compromissos e eventos, tais quais sua visualização, adição, e edição.		

Fonte: Autoria Própria.

Quadro 4 - Requisito Expandir *Compromissos e Eventos*.

REQ002.1 – Expandir <i>Compromissos e Eventos</i>			
SOLICITANTE:	Letícia Ueda Vanessa Maria da Silva	REQ. ORIGEM:	REQ002
TIPO DO REQUISITO:	Funcional	IMPACTO NA ARQUITETURA:	Alto
DESCRIÇÃO:	O software do servidor web deverá permitir que o usuário visualize de maneira expandida os compromissos e eventos marcados para um dia específico ao clicar num evento ou compromisso específicos.		

Fonte: Autoria Própria.

Quadro 5 – Requisito Criar Compromissos.

REQ002.2 – Criar Compromisso			
SOLICITANTE:	Letícia Ueda Vanessa Maria da Silva	REQ. ORIGEM:	REQ002
TIPO DO REQUISITO:	Funcional	IMPACTO NA ARQUITETURA:	Alto
DESCRIÇÃO:	O software do servidor web deverá permitir a criação de novos compromissos acadêmicos nos dias especificados.		

Fonte: Autoria Própria.**Quadro 6 - Requisito Editar Compromisso.**

REQ002.3 – Editar Compromisso			
SOLICITANTE:	Letícia Ueda Vanessa Maria da Silva	REQ. ORIGEM:	REQ002
TIPO DO REQUISITO:	Funcional	IMPACTO NA ARQUITETURA:	Alto
DESCRIÇÃO:	O software do servidor web deverá permitir editar os compromissos criados pelo utilizador do sistema.		

Fonte: Autoria Própria.**Quadro 7 - Requisito Criar Evento.**

REQ002.4 – Criar Evento			
SOLICITANTE:	Letícia Ueda Vanessa Maria da Silva	REQ. ORIGEM:	REQ002
TIPO DO REQUISITO:	Funcional	IMPACTO NA ARQUITETURA:	Alto
DESCRIÇÃO:	O software do servidor web deverá permitir a criação de novos eventos acadêmicos. Estes serão notificados aos alunos de interesse, como uma pergunta.		

Fonte: Autoria Própria.**Quadro 8 - Requisito Editar Evento.**

REQ002.5 – Editar Evento			
SOLICITANTE:	Letícia Ueda Vanessa Maria da Silva	REQ. ORIGEM:	REQ002
TIPO DO REQUISITO:	Funcional	IMPACTO NA ARQUITETURA:	Alto
DESCRIÇÃO:	O software do servidor web deverá permitir editar os eventos criados pelo utilizador do sistema.		

Fonte: Autoria Própria.**Quadro 9 - Autenticar Dispositivo.**

REQ003 – Autenticar Dispositivo			
SOLICITANTE:	Letícia Ueda Vanessa Maria da Silva	REQ. ORIGEM:	
TIPO DO REQUISITO:	Funcional	IMPACTO NA ARQUITETURA:	Alto
DESCRIÇÃO:	O software do servidor web deverá reconhecer os novos dispositivos móveis para que possa haver comunicação entre eles.		

Fonte: Autoria Própria.

Quadro 10- Requisito Efetuar Conexão.

REQ004 – Efetuar Conexão			
SOLICITANTE:	Letícia Ueda Vanessa Maria da Silva	REQ. ORIGEM:	
TIPO DO REQUISITO:	Funcional	IMPACTO NA ARQUITETURA:	Alto
DESCRIÇÃO:	O software do servidor web deverá efetuar conexões frequentes com os dispositivos móveis autenticados para notificar novos dados e dados atualizados.		

Fonte: Autoria Própria.**Quadro 11 - Requisito Fornecer Atualização.**

REQ005 – Fornecer Atualização			
SOLICITANTE:	Letícia Ueda Vanessa Maria da Silva	REQ. ORIGEM:	
TIPO DO REQUISITO:	Funcional	IMPACTO NA ARQUITETURA:	Alto
DESCRIÇÃO:	<p>Fornecer atualização para cada cliente individualmente.</p> <p>O cliente requisita ao <i>WebService</i> uma atualização.</p> <p>O Servidor recebe a ID do dispositivo (ou usuário) e pesquisa as fontes para ver se há atualizações. Então envia ao cliente.</p> <p>O cliente solicita automaticamente atualização a cada 15min ou manualmente via botão de Atualização ou na conexão.</p>		

Fonte: Autoria Própria.

5.2.2 Requisitos não Funcionais

Estes definem propriedades e restrições do sistema com relação ao sistema, ou parte dele, neste caso estão descritos os requisitos não funcionais do dispositivo do software servidor.

Quadro 12 - Requisito Acesso Área Restrita.

REQ006 – Acesso Área Restrita			
SOLICITANTE:	Letícia Ueda Vanessa Maria da Silva	REQ. ORIGEM:	
TIPO DO REQUISITO:	Não Funcional	IMPACTO NA ARQUITETURA:	Alto
DESCRIÇÃO:	O software do servidor web deverá permitir somente que professores acessem a área de Agenda do servidor . E cada professor só terá acesso aos grupos de qual participa.		

Fonte: Autoria Própria.

Quadro 13 - Requisito Origem dos Dados.

REQ007 – Origem dos dados			
SOLICITANTE:	Letícia Ueda Vanessa Maria da Silva	REQ. ORIGEM:	
TIPO DO REQUISITO:	Não Funcional	IMPACTO NA ARQUITETURA:	Alto
DESCRIÇÃO:	O software do servidor web deverá conter uma base de dados já populada com os dados provindos de sistema externo, ou seja, os seguintes dados serão obtidos externamente ao MAAS: Departamentos; Cursos; Turmas; Alunos; Professores; Disciplinas; Horários;		

Fonte: Autoria Própria.

Quadro 14 - Requisito Dados Cadastráveis.

REQ008 – Dados Cadastráveis			
SOLICITANTE:	Letícia Ueda Vanessa Maria da Silva	REQ. ORIGEM:	
TIPO DO REQUISITO:	Não Funcional	IMPACTO NA ARQUITETURA:	Alto
DESCRIÇÃO:	O software do servidor web deverá permitir apenas os cadastros de Compromissos e Eventos.		

Fonte: Autoria Própria.

Quadro 15 - Interface com o Usuário

REQ009 – Interface com o Usuário			
SOLICITANTE:	Letícia Ueda Vanessa Maria da Silva	REQ. ORIGEM:	
TIPO DO REQUISITO:	Não Funcional	IMPACTO NA ARQUITETURA:	Alto
DESCRIÇÃO:	O software do servidor web deverá ser de acesso simples ao usuário, com poucos cliques na tela o sistema deverá permitir que o usuário consulte seus dados, inclua novos compromissos e/ou eventos, os altere e cancele caso seja o seu desejo.		

Fonte: Autoria Própria.

Quadro 16 - Requisito Ambiente Operacional.

REQ010 – Ambiente Operacional			
SOLICITANTE:	Letícia Ueda Vanessa Maria da Silva	REQ. ORIGEM:	
TIPO DO REQUISITO:	Não Funcional	IMPACTO NA ARQUITETURA:	Alto
DESCRIÇÃO:	O software do servidor web deverá operar em sistema Windows.		

Fonte: Autoria Própria.

Quadro 17 - Requisito Atualização Banco de Dados.

REQ011 – Atualização Banco de Dados			
SOLICITANTE:	Letícia Ueda Vanessa Maria da Silva	REQ. ORIGEM:	
TIPO DO REQUISITO:	Não Funcional	IMPACTO NA ARQUITETURA:	Alto
DESCRIÇÃO:	O banco de dados do servidor deve ser atualizado em tempo real. E os dados externos a cada novo semestre, antes do início deste.		

Fonte: Autoria Própria.

Quadro 18 - Requisito Idioma.

REQ012 – Idioma			
SOLICITANTE:	Letícia Ueda Vanessa Maria da Silva	REQ. ORIGEM:	
TIPO DO REQUISITO:	Não Funcional	IMPACTO NA ARQUITETURA:	Alto
DESCRIÇÃO:	O sistema MAAS deverá ser construído com interface no idioma português do Brasil. Exceto o nome do Sistema que poderá ser escrito em inglês ou português.		

Fonte: Autoria Própria.

Quadro 19 - Requisito Portabilidade.

REQ013 – Portabilidade			
SOLICITANTE:	Letícia Ueda Vanessa Maria da Silva	REQ. ORIGEM:	
TIPO DO REQUISITO:	Não Funcional	IMPACTO NA ARQUITETURA:	Alto
DESCRIÇÃO:	O sistema MAAS deverá ser construído na arquitetura MVC (Model, View, Controller), para que haja camadas de software que separem modelo, interface e acesso a dados, para tornar o sistema facilmente portátil.		

Fonte: Autoria Própria.

5.3 APLICAÇÃO CLIENTE

A seguir serão descritos detalhes da especificação que dizem respeito à aplicação desenvolvida para a aplicação cliente do MAAS, criada para a plataforma Windows Phone.

5.3.1 Requisitos Funcionais

O módulo de software do dispositivo móvel possui seus próprios requisitos, por se tratar de ambiente e funções diferenciadas do servidor web, conforme descrito a seguir.

Quadro 20 - Requisito Conectar ao Servidor.

REQ001 – Conectar ao Servidor			
SOLICITANTE:	Letícia Ueda Vanessa Maria da Silva	REQ. ORIGEM:	
TIPO DO REQUISITO:	Funcional	IMPACTO NA ARQUITETURA:	Alto
DESCRIÇÃO:	O software do dispositivo móvel deverá permitir identificar o utilizador do sistema apenas na primeira vez após sua instalação. Nas utilizações posteriores do aplicativo a identificação ocorrerá automaticamente.		

Fonte: Autoria Própria.

Quadro 21 - Requisito Apresentar Agenda.

REQ002 – Apresentar Agenda			
SOLICITANTE:	Letícia Ueda Vanessa Maria da Silva	REQ. ORIGEM:	
TIPO DO REQUISITO:	Funcional	IMPACTO NA ARQUITETURA:	Alto
DESCRIÇÃO:	O software do dispositivo móvel deverá apresentar a agenda do usuário. Apresentar a agenda em modo mês; Apresenta um indício de que há compromisso em determinado dia;		

Fonte: Autoria Própria.

Quadro 22 - Requisito Expandir Compromissos do Dia.

REQ003 – Expandir Compromissos do Dia			
SOLICITANTE:	Letícia Ueda Vanessa Maria da Silva	REQ. ORIGEM:	
TIPO DO REQUISITO:	Funcional	IMPACTO NA ARQUITETURA:	Alto
DESCRIÇÃO:	O software do dispositivo móvel permitirá que o usuário visualize uma lista de todos os compromissos e eventos marcados para um dia selecionado.		

Fonte: Autoria Própria.

Quadro 23 - Requisito Visualizar Compromisso.

REQ004 – Visualizar Compromisso			
SOLICITANTE:	Letícia Ueda Vanessa Maria da Silva	REQ. ORIGEM:	
TIPO DO REQUISITO:	Funcional	IMPACTO NA ARQUITETURA:	Alto
DESCRIÇÃO:	O software do dispositivo móvel permitirá ao usuário a visualização de um compromisso com mais detalhes ao clicar no compromisso apresentado no dia selecionado.		

Fonte: Autoria Própria.

Quadro 24 - Requisito Visualizar Evento.

REQ005 – Visualizar Evento			
SOLICITANTE:	Letícia Ueda Vanessa Maria da Silva	REQ. ORIGEM:	
TIPO DO REQUISITO:	Funcional	IMPACTO NA ARQUITETURA:	Alto
DESCRIÇÃO:	O software do dispositivo móvel deverá permitir ao usuário a visualização de um evento com mais detalhes ao clicar no evento apresentado no dia selecionado.		

Fonte: Autoria Própria.

5.3.2 Requisitos não Funcionais

Os requisitos não funcionais definem propriedades e restrições com relação ao sistema, ou parte dele. Nesta seção estão descritos os requisitos não funcionais do dispositivo do software móvel.

Quadro 25 - Requisito Estilo.

REQ007 – Estilo			
SOLICITANTE:	Letícia Ueda Vanessa Maria da Silva	REQ. ORIGEM:	
TIPO DO REQUISITO:	Não Funcional	IMPACTO NA ARQUITETURA:	Alto
DESCRIÇÃO:	O software do dispositivo móvel deverá ter o <i>design</i> baseado no estilo metro, seguindo o <i>designer</i> do sistema operacional do smartphone, para fácil associação do usuário.		

Fonte: Autoria Própria.

Quadro 26 - Requisito Interface com o Usuário Dispositivo Móvel.

REQ008 – Interface com o Usuário			
PRIORIDADE:	Alta	ESTABILIDADE	Alto
SOLICITANTE:	Letícia Ueda Vanessa Maria da Silva	REQ. ORIGEM:	
TIPO DO REQUISITO:	Não Funcional	IMPACTO NA ARQUITETURA:	Alto
DESCRIÇÃO:	O software do dispositivo móvel deverá ser de acesso simples ao usuário, realizando de 3 a 4 toques na tela do sistema para passar as informações ao usuário.		

6 DESENVOLVIMENTO

A partir da metodologia e especificação do software, iniciou-se o desenvolvimento do protótipo. Neste capítulo são mostradas as etapas do trabalho como as modelagens de casos de uso, dados, e classes, os primeiros protótipos da interface do software, os ensaios de interação e testes de software.

6.1 DIAGRAMAS DE CASOS DE USO

Os casos de uso constituem uma técnica baseada em cenários para descrever os requisitos, sendo cada cenário um encadeamento isolado ao longo do caso de uso (SOMMERVILLE, 2007). Neste documento são descritas a sequência de atividades necessárias para a realização da funcionalidade, bem como a interação necessária.

Para demonstrar e descrever as funcionalidades e definir o escopo deste projeto foi desenvolvido os casos de uso referentes ao software, apresentados nas seções a seguir.

6.1.1 Com relação à aplicação do dispositivo móvel

A partir do levantamento dos requisitos realizado para a aplicação móvel, foram criados os diagramas de casos de uso mencionados nas próximas subseções, juntamente com sua descrição, pré-condições e pós-condições.

6.1.1.1 Efetuar *Login*

Caso de uso executado logo após a instalação do aplicativo, Figura 11. O Aluno, ator principal, deve informar seu número de matrícula e senha para autenticação. Confirmado o *login*, a aplicação irá fazer conexão com o servidor web para que os dados de interesse sejam trazidos para a base de dados do *smartphone*.



Figura 11 – Diagrama de Caso de Uso Efetuar *Login*

- Pré-condições: o aluno deve possuir um número de matrícula e ter instalado o aplicativo em seu celular *smartphone*.
- Pós-condições: os compromissos e eventos referentes ao aluno serão sincronizados ao calendário instalado.

6.1.1.2 Conectar ao Servidor

Caso de uso executado a todo o momento em que o *smartphone* detecta uma rede de internet. Há a necessidade de iniciar uma conexão com o servidor para atualizar os dados da agenda. A autenticação do aluno ocorre automaticamente depois da efetuação do *Login* apresentado na Figura 12.



Figura 12 – Diagrama de Caso de Uso Conectar ao servidor.

- Pré-condições: o aluno deve possuir um número de matrícula e ter efetuado o primeiro *login* na aplicação.
- Pós-condições: após a conexão estabelecida, as novas informações interessantes para o aluno serão atualizadas em sua agenda.

6.1.1.3 Apresentar Calendário

Este caso de uso pode ser ativado a qualquer momento pelo ator principal, o aluno, quando houver necessidade de checar a agenda. A partir da apresentação da agenda há as opções de expandir o dia desejado para que o ator possa visualizar quais são os compromissos

e eventos agendados, e, a partir daí, também há possibilidade de expandir visualmente o compromisso ou evento para maiores detalhes, conforme Figura 13.

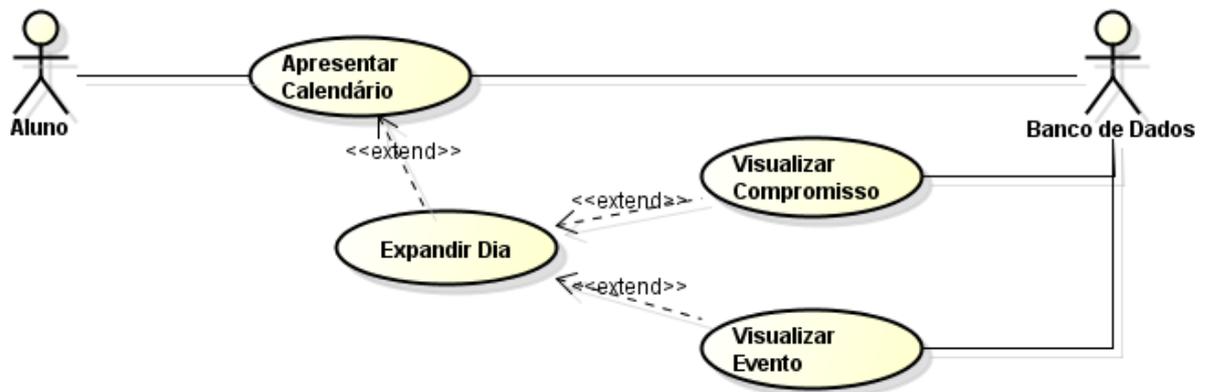


Figura 13 - Diagrama de Caso de Uso Apresentar Calendário.

- Pré-condições: o aluno deve ter passado seu numero de matrícula à aplicação e deve estar conectado à internet.
- Pós-condições: o usuário poderá visualizar o calendário com os devidos compromissos e eventos.

6.1.2 Com relação à aplicação do servidor web

A partir do levantamento dos requisitos realizado para a aplicação móvel, foram criados os diagramas de casos de uso mencionados nas próximas subseções, juntamente com sua descrição, pré-condições e pós-condições.

6.1.2.1 Efetuar *Login*

Esse diagrama de caso de uso é acionado toda vez que o ator (i.e., professor ou responsável da Universidade) acessar o servidor do sistema, como apresentado na Figura 14. Ele precisará indicar seus dados de *login* para poder acessar o servidor web.



Figura 14 - Diagrama de Caso de Uso Efetuar *Login* Servidor.

- Pré-condições: o ator deve possuir um *login* e uma senha cadastrados no sistema.
- Pós-condições: o ator acessará a aplicação e poderá efetuar diversas ações.

6.1.2.2 Editar Agenda

Diagrama de Caso de uso executado quando o professor desejar realizar alguma operação sobre a sua agenda Figura 15. O ator pode visualizar a sua agenda em forma de calendário. Para expandir os eventos e compromissos marcados para um dia específico, basta que o professor clique em cima do dia desejado no calendário. A partir daí ele pode expandir os dados do evento/compromisso e assim modifica-lo. Há também a possibilidade de criar um novo compromisso/evento para adicionar no calendário de um grupo de interesse.

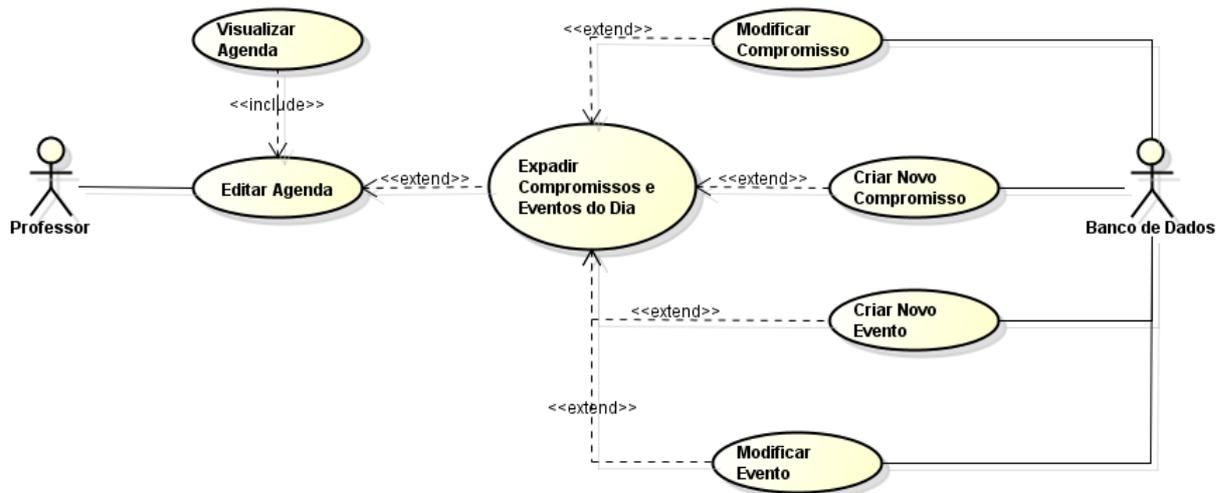


Figura 15 - Diagrama de Caso de Uso Expandir Compromissos e Eventos do Dia.

- Pré-condições: o ator deve estar logado no sistema, e deve selecionar uma turma para receber a notificação do compromisso.
- Pós-condições: a agenda será atualizada, e uma notificação de novo compromisso será enviada à turma selecionada pelo usuário.

6.1.2.3 Efetuar *Logout*

Caso de uso executado quando o professor terminou as atividades no servidor desejar sair de sua conta é exibido na Figura 16.

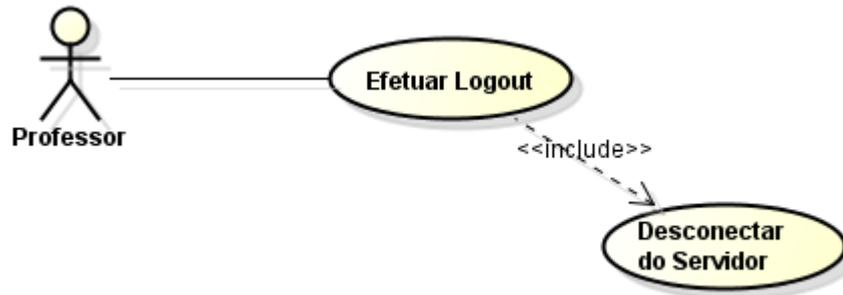


Figura 16 - Diagrama de Caso de Uso Efetuar *Logout*.

- Pré-condições: o ator deve estar logado no sistema.
- Pós-condições: o usuário não terá mais acesso à sua agenda acadêmica pelo servidor web.

6.2 MODELO ESTRUTURAL

Para representar o modelo estrutural do MAAS foi desenvolvido o Diagrama de Entidade e Relacionamento (DER) apresentado na Figura 17. O software utiliza uma base de dados já preenchida com os dados acadêmicos dos alunos da Universidade, como disciplinas matriculadas, horários das disciplinas, etc., estes são baseados nos dados presente no servidor do sistema acadêmico. Isso descarta a necessidade de cadastrar todas essas informações, tornando o aplicativo mais prático que as soluções encontradas no mercado até então.

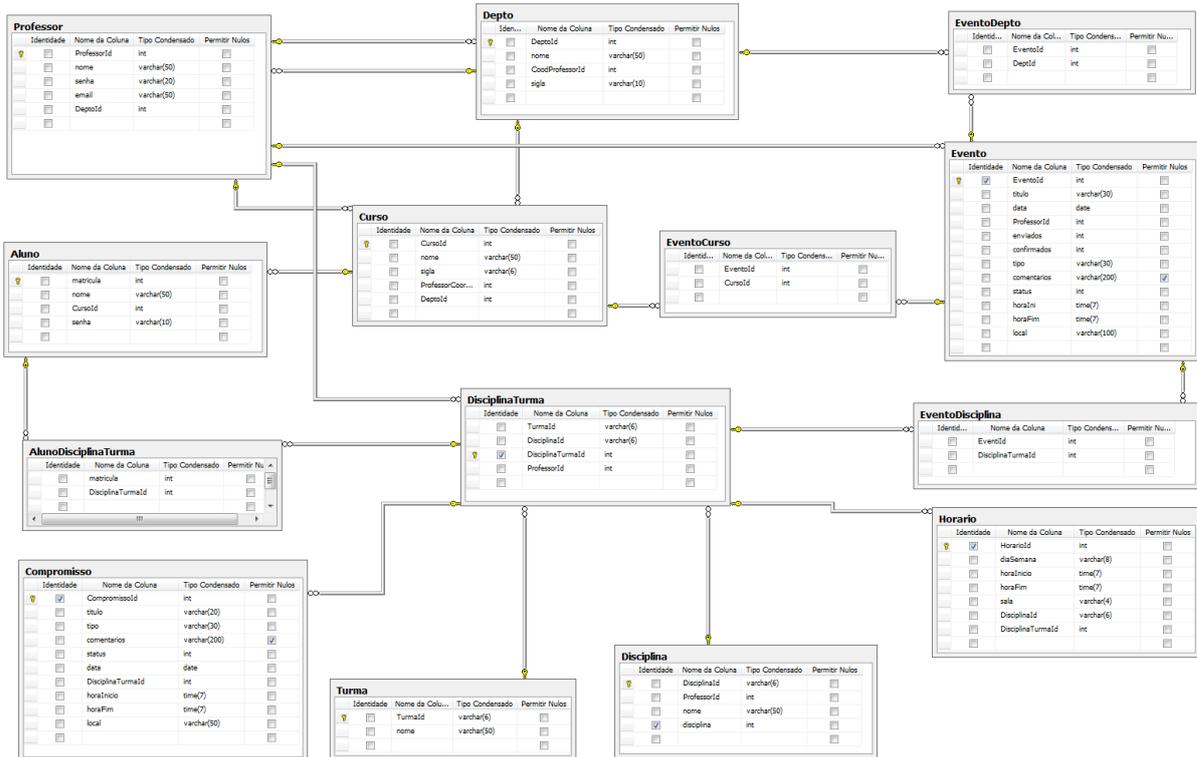


Figura 17 - Modelo do Banco de Dados Servidor Web.

Ao total foram criadas 14 tabelas incluindo as intermediárias (nxn). O modelo foi criado baseando-se nos relacionamentos existentes entre as entidades reais na UTFPR. Os vínculos de evento e compromisso foram criados de modo que os compromissos pertencem somente a entidade DisciplinaTurma (disciplina de uma determinada turma), os eventos pertençam as entidades Departamento, Curso e DisciplinaTurma. Como a entidade Professor é quem cria as instâncias de eventos e compromissos, ela tem relacionamentos com as entidades a qual os dados de evento e compromisso pertencem.

A partir do modelo de banco de dados foram criados os diagramas de classes que refletem o modelo de dados. Nele são representadas as classes existentes na aplicação do servidor web, bem como os relacionamentos existentes entre as entidades do pacote Model. O diagrama de classes inicial da parte referente ao servidor web está representado na Figura 18.

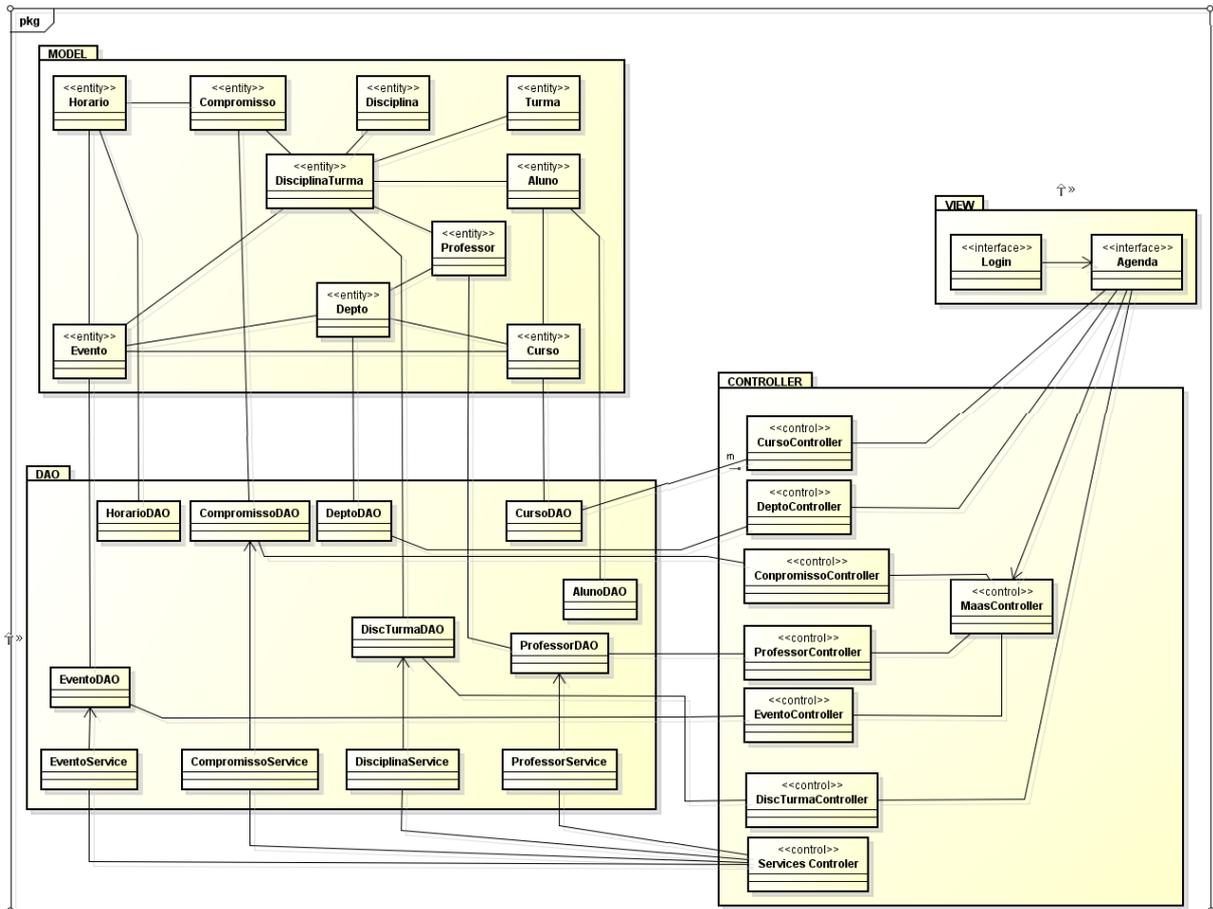


Figura 18 - Diagrama de classes do servidor.

O diagrama de entidade e relacionamento criado para a aplicação móvel contém menos classes que a do servidor web. Por necessitar de menos dados, seu banco de dados é composto por 4 tabelas: uma referente aos compromissos do aluno, outra para os eventos, para as disciplinas que o aluno cursa e os professores que ministram essas disciplinas, como é apresentado na Figura 19. A justificativa de manter um banco de dados no *smartphone* é de possibilitar que o usuário acesse os dados de sua agenda em modo *off-line*. Aumentando assim a sua utilidade.

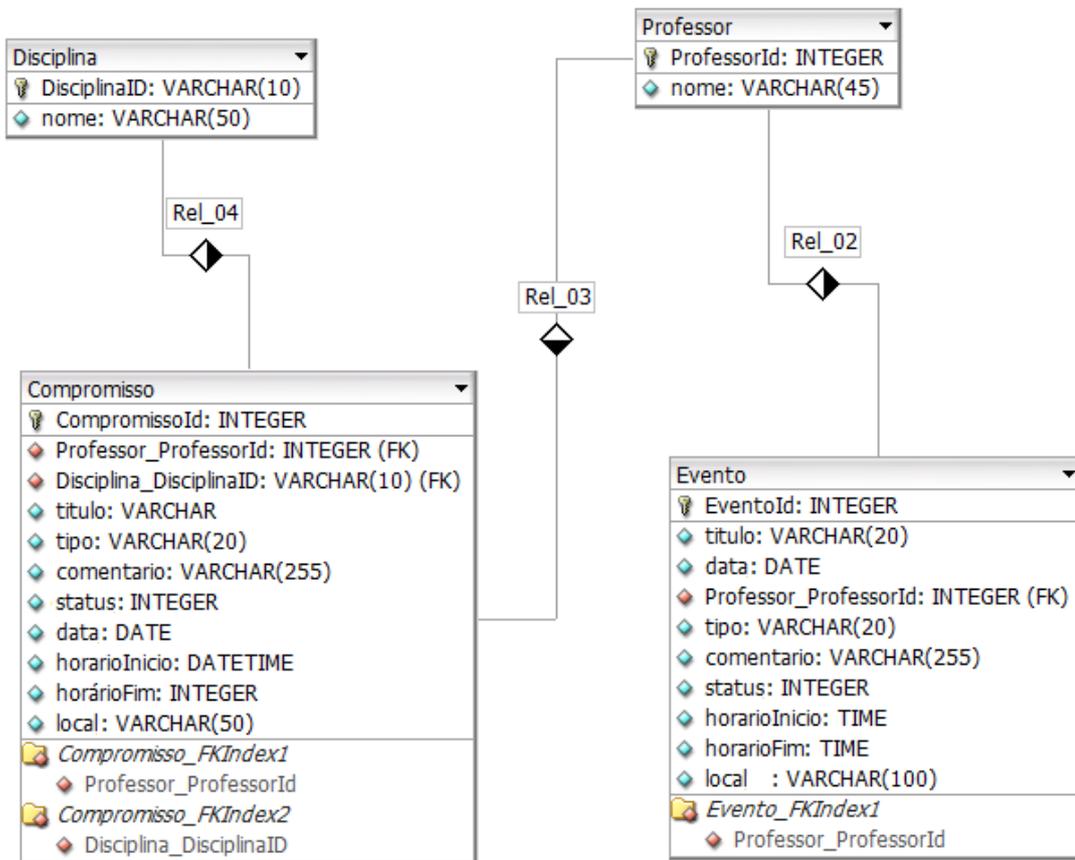


Figura 19 - Modelo do banco de dados da aplicação móvel

O banco de dados da aplicação móvel será populado por meio do envio de dados presentes no servidor. Requisições são realizadas periodicamente para buscar novas atualizações de compromissos e eventos na agenda do usuário.

A partir da criação do modelo de dados da aplicação móvel, foi desenvolvido o diagrama de classes, Figura 20, representando as classes e os relacionamentos existentes entre nas classes da aplicação móvel.

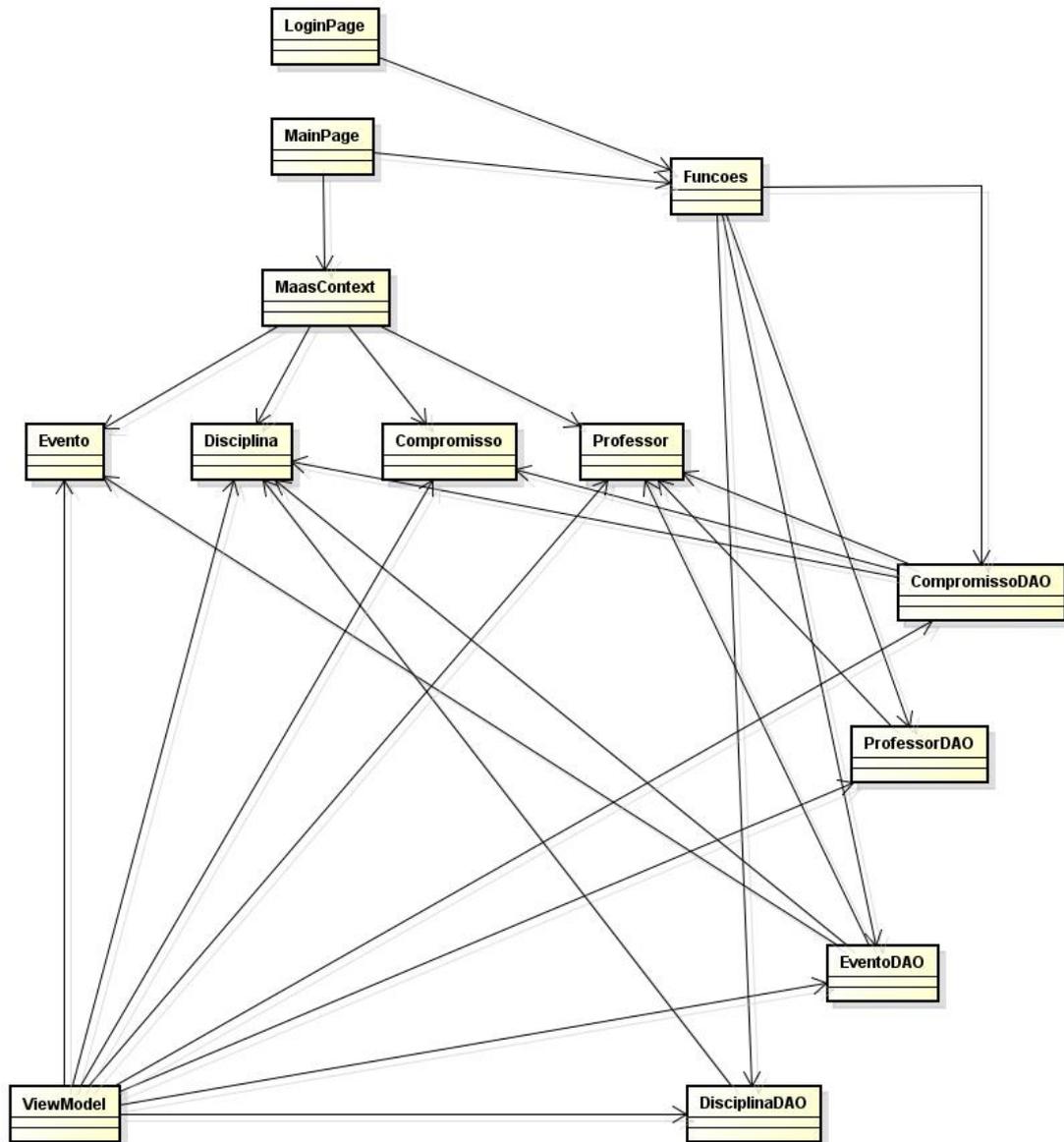


Figura 20 - Diagrama de classes da aplicação móvel

Obtidos esses dados o sistema poderá filtrar os eventos, compromissos e aulas de interesse para o aluno.

No sistema web, o usuário deverá fazer o *login* para entrar em sua conta, e poderá efetuar operações referentes à eventos, compromissos e aulas. As operações de adição, modificação e exclusão de qualquer item, resultará na atualização do calendário do estudante e no disparo de uma notificação, caso a aplicação esteja em execução no *smartphone*.

6.3 PROTOTIPAÇÃO

Desenvolvidos os diagramas, alguns protótipos das telas do sistemas foram elaborados, com o intuito de pré-visualizar a melhor maneira de criar as interfaces do software. Protótipos são úteis para o entendimento dos *stakeholders* eles facilitam a comunicação entre os membros da equipe e funcionam como uma maneira de testar ideias. Um “*protótipo é uma representação limitada de um design que permite interagir com ele e explorar sua conveniência*” (PREECE, 2008).

Na elaboração do MAAS esses protótipos puderam idealizar o fluxo de navegação, dar suporte ao usuário, medir aproximadamente o tempo de desenvolvimento e grau de dificuldade da implementação, mas principalmente tornou possível a aquisição do *feedback* dos usuários sobre projetos e ideias de *design* no início do processo de desenvolvimento. Para esta fase do projeto foram escolhidos *mock-ups*, que são "protótipos iniciais" feitos de materiais de baixo custo. O usuário, auxiliado pelo *designer*, pode testar o *mock-up* (imaginando que ele funciona) e, assim, fornecer um *feedback* valioso sobre a funcionalidade, usabilidade, compreensão da ideia de projeto básico (INTERACTION DESIGN FOUNDATION, 2010).

6.3.1 Com Relação à Aplicação do Dispositivo Móvel

Os *mock-ups* do dispositivo móvel foram desenvolvidos em Silverlight na ferramenta Blend versão 4 da Microsoft. Esta ferramenta é destinada para a criação de interfaces, gerando automaticamente o código XAML utilizado na implementação de aplicações Windows Phone. Os *mock-ups* desenvolvidos para o dispositivo móvel foram reutilizados na implementação, sendo importados na fase de implementação.



Agenda Acadêmica

Efetuar Login

Matrícula:

Senha:

Entrar

Figura 21 - Mock-up Efetuar Login Dispositivo Móvel

O primeiro *mock-up* representa tela de *Login* do aplicativo móvel, Figura 21. Nesta tela o usuário precisaria informar seu número de matrícula e senha gerada para o uso da aplicação. Este corresponde ao requisito REQ001 Conectar ao Servidor.

O *mock-up* representado na Figura 22 representa a tela principal da Agenda Acadêmica, na qual o usuário pode escolher um dia para visualizar seus compromissos e eventos marcados para esta data. Este corresponde ao requisito REQ002 – Apresentar Agenda.



Figura 22 - Mock-up Apresentar Agenda Dispositivo Móvel

Após clicar em um dia, a tela Agenda é expandida, apresentando todos os eventos e compromissos do dia, conforme a Figura 23. Esta tela correspondendo ao requisito REQ003 – Expandir Compromissos do Dia.



Figura 23 - Mock-up Visualizar Agenda Modo Dia

Para visualizar os detalhes de um compromisso ou evento que foi listado na tela de Dia Expandido, o usuário deve clicar sobre o item da lista. Assim, uma nova tela é projetada

para que exiba o evento ou compromisso escolhido pelo usuário, conforme a Figura 24, o qual possui os detalhes da atividade apresentada e possíveis observações dos professores, este corresponde ao requisito REQ004 – Visualizar Compromisso e REQ005 – Visualizar Evento.

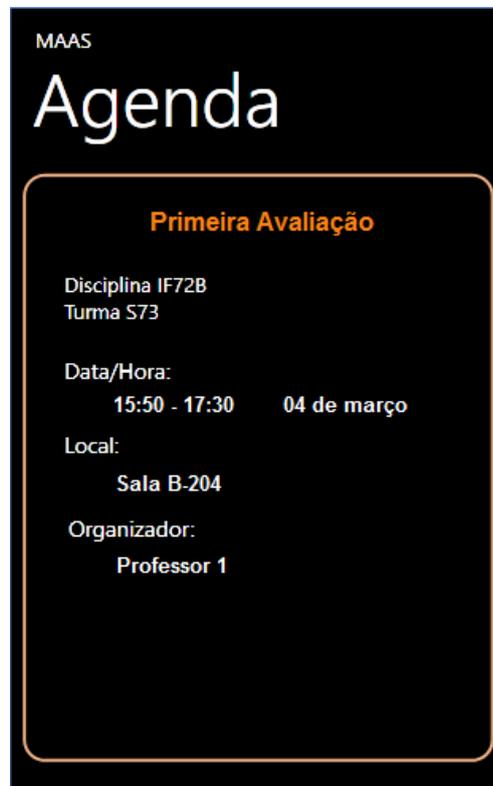


Figura 24 - *Mock-up* Apresentar Compromisso/Evento

Assim como o aplicativo mobile, o servidor também teve as interfaces com o usuário projetadas. É o que será apresentado na seção adiante.

6.3.2 Com Relação à Aplicação do Servidor Web

Os *mock-ups* do servidor web foram desenvolvidos na ferramenta Balsamiq Mock-ups (BALSAMIQ, 2013), diferente da aplicação do dispositivo móvel. O *mock-up* da Figura 25 representa tela de Login do servidor web, este corresponde ao requisito REQ001 – Efetuar *Login*.

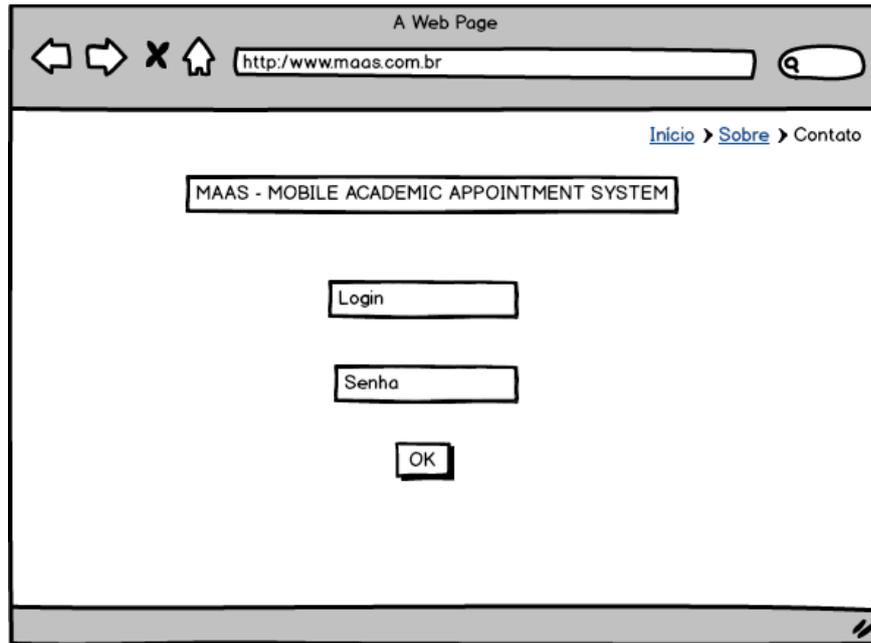


Figura 25 - Mock-up Tela Login Servidor

O protótipo da tela Agenda, Figura 26, apresenta a tela responsável pela administração de compromissos e eventos que é mantida por cada professor (fonte), na qual ele pode visualizar a sua agenda, a lista de atividades do dia (compromissos, eventos, aulas, etc) pode criar ou editar as atividades pelas quais é responsável. Esta tela corresponde aos requisitos REQ002 – Editar Agenda da Fonte, REQ002.1 – Expandir Eventos e Compromissos, REQ002.2 – Criar Compromisso, REQ002.3 – Editar Compromisso, REQ002.4 – Criar Evento, REQ002.5 – Editar Evento e REQ009 – Interface com o Usuário.

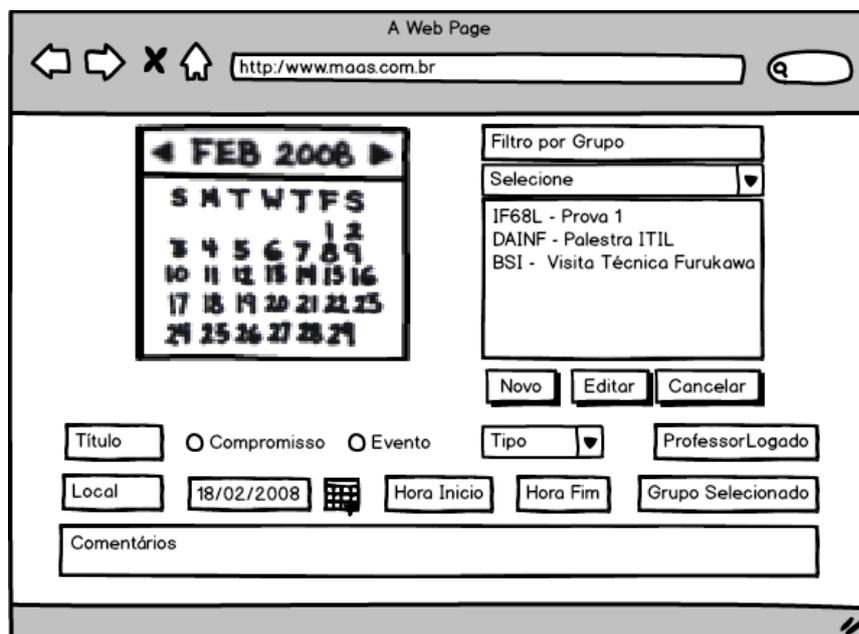


Figura 26 - Mock-up Tela Agenda do Servidor

O desenvolvimento de protótipos foi muito importante para tomada de decisões, percepção das deficiências a serem tratadas e definição de tarefas por funcionalidades que a tela utiliza. Para o desenvolvedor a interface diz muito, ela elucida tarefas que estão nas camadas mais inferiores da implementação, por exemplo, para mostrar na interface um compromisso expandido, o desenvolvedor terá de acessar a camada de controle da aplicação passando somente parâmetros necessários. E, por sua vez, a camada de modelo realiza o acesso aos dados na base de dados e os retornar para visualização.

6.4 SOFTWARE MAAS

A partir da definição a respeito das funcionalidades do software MAAS e das ferramentas e tecnologias que seriam utilizadas no seu desenvolvimento, a implementação deste software tornou-se algo bastante objetivo, sem distrações desnecessárias. As seções a seguir apresenta o que, de fato, foi desenvolvido do protótipo.

6.4.1 Visão Geral das Interfaces

As interfaces do MAAS são utilizadas em ambientes diferentes: servidor web e dispositivo móvel. Estas interfaces são diferenciadas, pois os papéis dos usuários que as utilizarão têm perfis distintos. Os estudantes utilizarão a interface do dispositivo móvel que será atualizada fazendo requisições ao servidor web. Os professores e administrativos da Universidade utilizarão a interface do servidor web para alimentar o sistema de agenda acadêmica para que o aluno seja o usuário destas informações. Desta maneira, o aluno não precisaria inserir os dados de suas atividades acadêmicas, disciplinas e professores como ocorre em outras aplicações destinadas à organização de compromissos acadêmicos. Isso diferencia o software desenvolvido neste trabalho aos concorrentes de mercado. A atualização ocorre automaticamente, basta que o aluno conecte-se a internet para receber as novas informações.

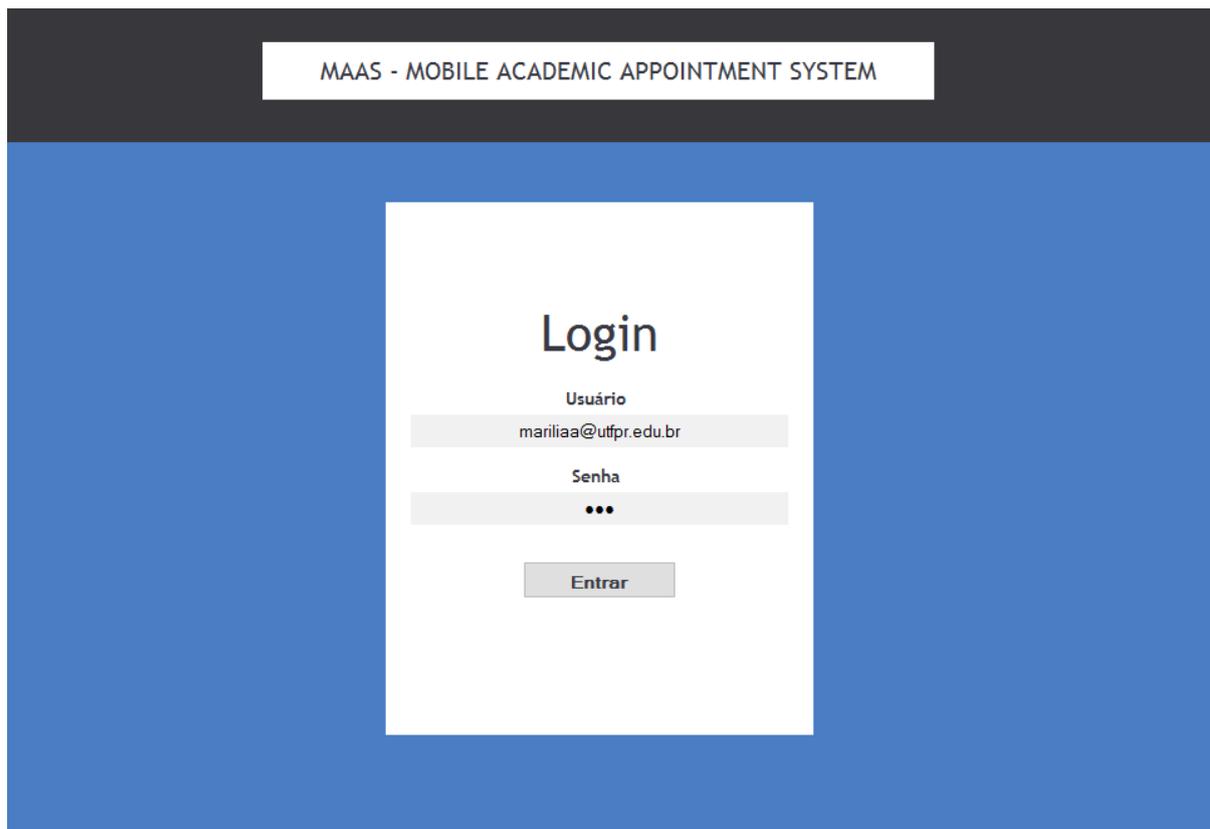


Figura 27 - Tela de Login do MAAS

A tela inicial do servidor web é a tela de *Login* do professor ou administrativo conforme ilustrado na Figura 27. Após a ocorrência de autenticação bem sucedida, o usuário tem acesso à tela principal Agenda, ilustrada na Figura 28. Nesta tela o professor deve administrar os compromissos e eventos pelos quais é responsável, de acordo com os grupos em que está vinculado. Por exemplo, um professor pertence a um departamento e ministra aulas em disciplinas de determinados cursos da instituição. O departamento, as disciplinas e os cursos são exemplos de grupos.

Esta tela possui todos os recursos necessários para a criação e manutenção dos eventos e compromissos que serão cadastrados no banco de dados do servidor e ficarão aguardando a requisição do dispositivo móvel para envio de informações.

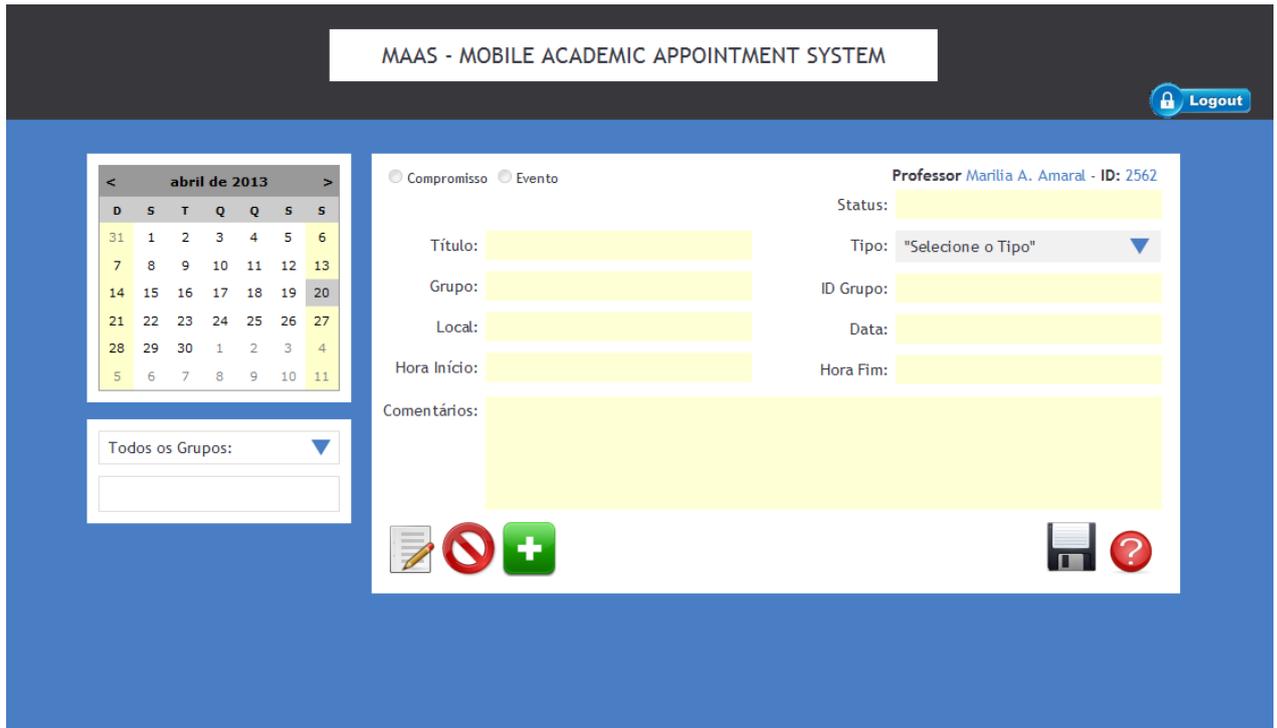


Figura 28 - Tela Principal do MAAS

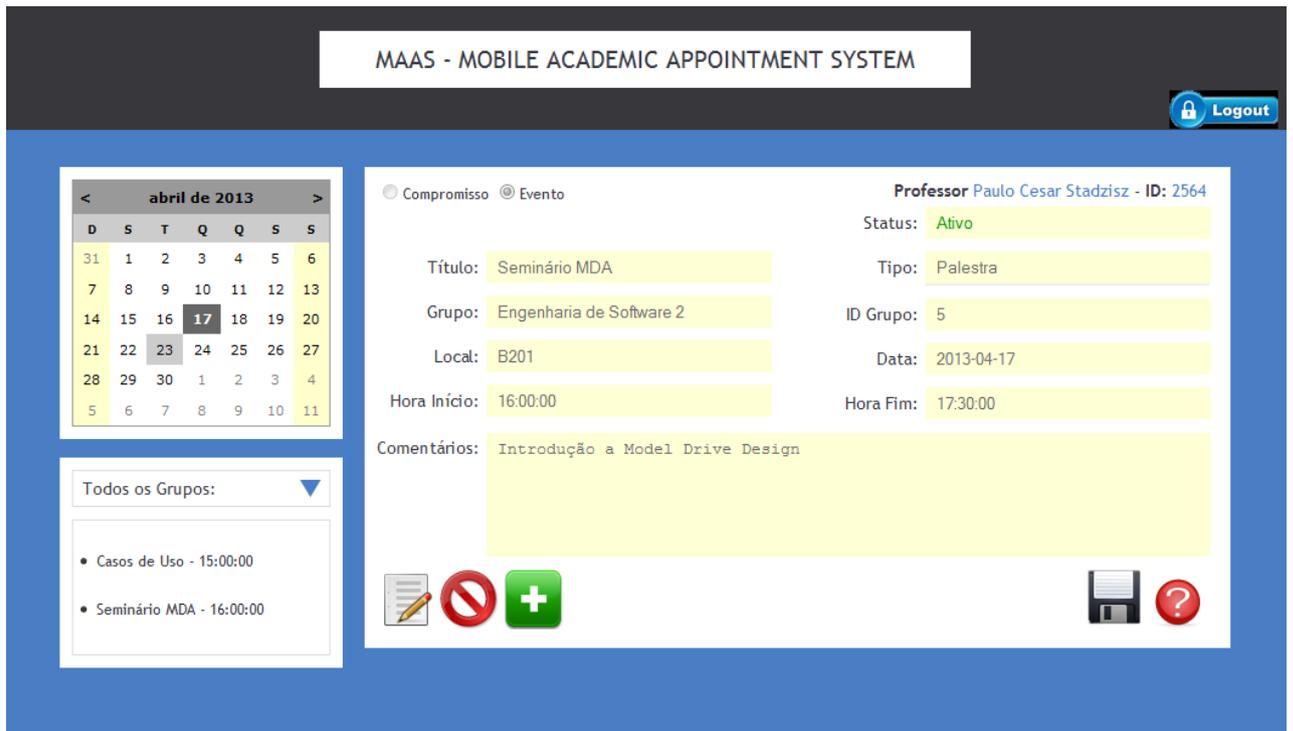


Figura 29 - Tela Principal do MAAS na Visão Detalhes de um Evento.

A tela principal tem muitas funções, pois todas as ações do usuário com exceção de *login* serão realizadas nesta tela a Figura 29 ilustra a visualização dos detalhes de um evento cadastrado pela fonte, nesta mesma tela é possível a realização da inclusão de novos compromissos e eventos, como também a edição deles.

Na aplicação móvel, após a instalação do aplicativo no smartphone, a tela inicial do software é a de *login* ilustrada na Figura 30. Nela devem ser informados o número de matrícula e uma senha gerada para o uso do protótipo. A tela de *login* será apresentada apenas na primeira vez que a autenticação for efetuada, nas demais vezes a aplicação fará a autenticação automaticamente.



Figura 30 - Tela Efetuar *Login* do Aplicativo Móvel.

Após o *login* do usuário, sua agenda é carregada com os compromissos e eventos dos grupos em que se encontra inserido. Mostra-se o mês relativo à data em que o aplicativo está sendo utilizado, como no exemplo da Figura 31 no mês de Abril. Os dias que contém compromissos ou eventos marcados serão destacados com a cor azul escura, para facilitar a visualização do aluno. Nesta tela há a opção de navegar entre os meses, por meio das flechas localizadas no lado direito e esquerdo do mês.

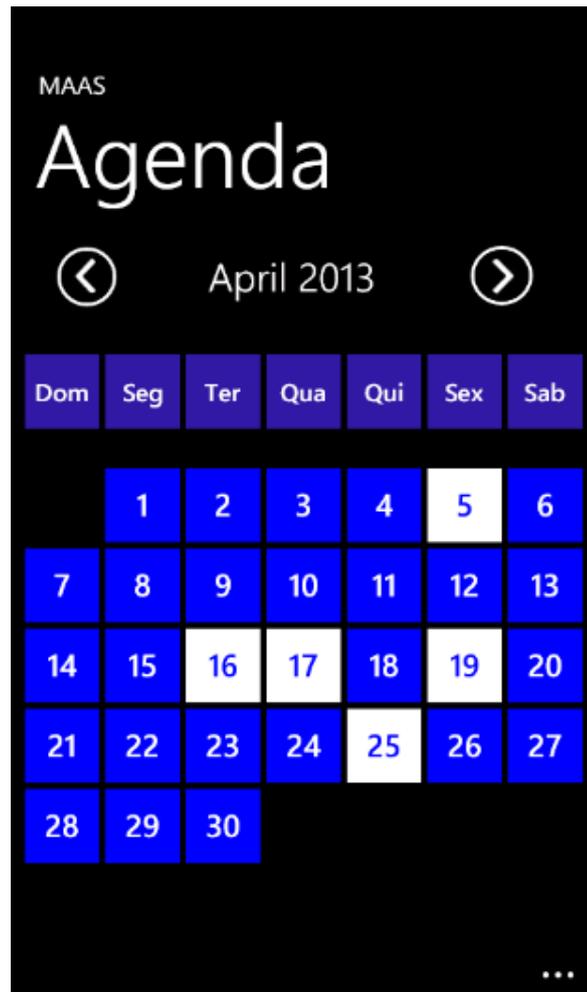


Figura 31 - Tela principal da aplicação móvel

Ao clicar em algum dia, a tela representada na Figura 32 será exibida com os devidos compromissos e eventos associados à data escolhida. Eles são listados de maneira simplificada com o objetivo de informar seu título e horário.



Figura 32 - Tela de compromissos e eventos do dia da aplicação móvel

Para observar maiores detalhes de algum compromisso ou evento, basta que o usuário clique em cima de um item para uma nova tela aparecer, conforme a Figura 33. Nela serão informados todos os detalhes do compromisso/evento, como comentário deixado pelo professor, local e disciplina associada.

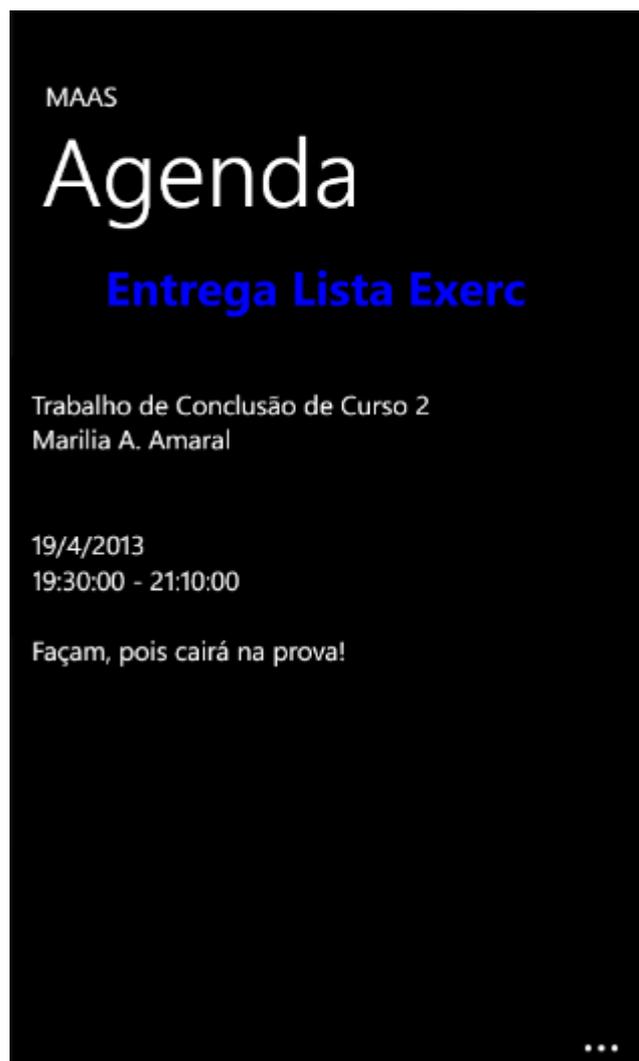


Figura 33 - Tela Visualizar Detalhes do Aplicativo Móvel.

As interfaces apresentadas nesta seção possuem funcionalidades que o usuário só terá a percepção assim que iniciar o uso do aplicativo, sendo assim as funcionalidades do software estão descritas na seção seguinte.

6.4.2 Descrição das Funcionalidades

As funcionalidades do MAAS permitem que o aplicativo atenda aos requisitos propostos para o software disponibilizando uma solução para agenda acadêmica.

O servidor web possui o conjunto de funcionalidades descritas a seguir:

- **Efetuar *Login***: esta função é responsável por autenticar o usuário que deseja acessar o servidor MAAS. Se o usuário for autenticado, o aplicativo liberará o uso do software servidor.
- **Inserir Compromisso**: esta função é responsável por capturar os dados relacionados a um novo compromisso inseridos na interface e registrá-los no banco de dados do servidor. Para tal, o usuário deve escolher um grupo e então clicar no botão novo.
- **Inserir Evento**: esta função é responsável por capturar os dados relacionados a um novo evento inseridos na interface e registrá-los no banco de dados do servidor. Para tal, o usuário deve escolher um grupo e então clicar no botão novo.
- **Visualizar Compromisso/Evento**: esta função permite que o usuário visualize os compromissos e eventos cadastrados no banco de dados do servidor em forma de lista diária. Para visualizá-los o usuário deve escolher um dia no calendário existente na tela e para ver os detalhes de um compromisso ou evento deve-se clicar no item desejado que será exibido na tela do servidor web com os campos desabilitados.
- **Editar Compromisso/Evento**: esta função é responsável por editar as alterações realizadas em um compromisso ou evento escolhido pelo usuário, sendo que este já existe na base de dados do servidor. Para tal, o usuário deve escolher um item na lista e então clicar no botão editar.
- **Atualizar Professores**: recebe uma requisição do dispositivo móvel contendo o número de matrícula de um aluno e responde a lista dos professores desse aluno no semestre corrente.
- **Atualizar Disciplinas**: recebe uma requisição do dispositivo móvel contendo o número de matrícula de um aluno e responde a lista das disciplinas que o aluno está matriculado no semestre corrente.
- **Atualizar Compromissos**: recebe uma requisição do dispositivo móvel contendo o número de matrícula de um aluno e responde a lista dos compromissos das disciplinas que o aluno está cadastrado no semestre corrente.
- **Atualizar Eventos**: recebe uma requisição do dispositivo móvel contendo o número de matrícula de um aluno e responde a lista dos eventos

correspondentes às disciplinas, curso e departamento ao qual o aluno está cadastrado no semestre corrente.

O dispositivo móvel (aplicação cliente) tem funções distintas do servidor. Os dados necessários para este módulo da aplicação são em quantidade menor e são personalizados, ou seja, o cliente recebe somente dados relacionados ao aluno usuário da aplicação em seu *smartphone* em particular.

- **Efetuar *Login***: esta função é responsável por autenticar o usuário que deseja acessar o aplicativo MAAS, se o usuário for autêntico o software liberará a entrada do usuário.
- **Visualizar *Agenda Mensal***: esta função permite que o usuário visualize a agenda mensal do MAAS. Os dias em que existem compromissos e/ou eventos cadastrados no banco de dados do dispositivo móvel estão destacados em cor mais forte que os demais dias.
- **Expandir compromissos e eventos do dia**: esta função permite que o usuário visualize os compromissos e eventos cadastrados no banco de dados do dispositivo móvel em forma de lista, exibindo o nome da atividade, nome da disciplina e horário de início e fim. Para visualizá-los o usuário deve escolher um dia no calendário da agenda.
- **Visualizar Detalhes de um Compromisso/Evento**: esta função permite que o usuário visualize detalhes dos compromissos e/ou eventos cadastrados no banco de dados do dispositivo móvel. Para visualizá-los o usuário deve escolher um item da lista e tocar sobre ele que será expandido e exibido na *smartphone* com os campos desabilitados.

6.4.3 Regras de Navegação

A navegação deve ser algo simples e funcional ao usuário para que este tenha acesso às informações do MAAS sem que precise utilizar muitos cliques. Este critério é de grande importância, pois se o usuário não conseguir navegar de forma simples pela aplicação aos poucos deixará de usá-la (SILVINO, ABRAHÃO, 2003). Para representar este critério utilizaram-se diagramas de atividade, cujo objetivo é mostrar o fluxo de atividades em um

único processo. As atividades são conectadas através de arcos (transições), que mostram as dependências entre elas (SAUVÉ, 2013). A Figura 34 descreve o fluxo principal de atividades do servidor web desde o *login* até o *logout* do usuário.

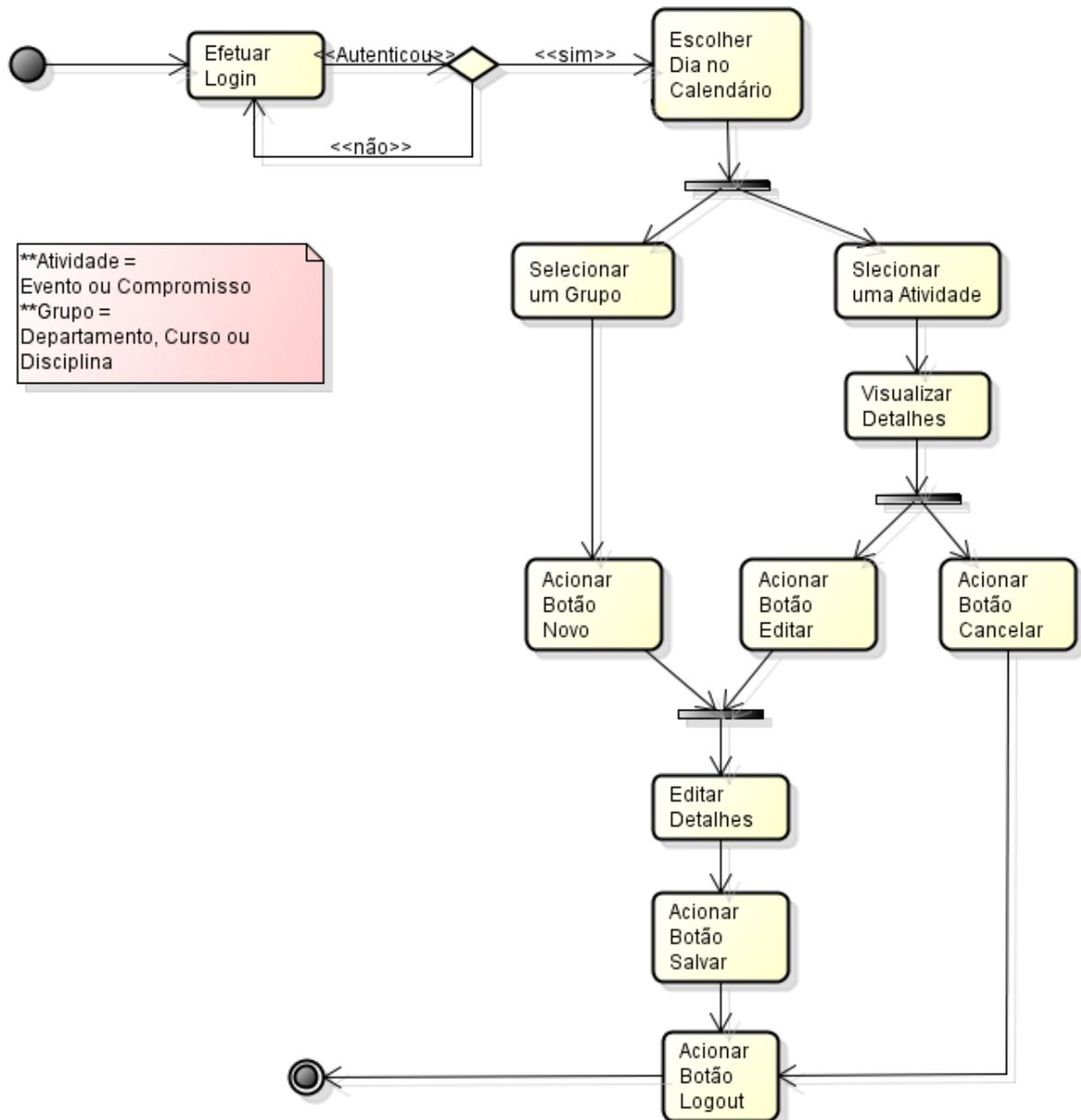


Figura 34 - Diagrama de atividades do fluxo principal do servidor web.

A aplicação cliente é simples, pois somente é de visualização e para navegação. A cada toque na tela o usuário visualiza uma função diferente, que o leva até os detalhes de uma atividade de interesse, sem a necessidade de que conclua o fluxo completo para consultar seus compromissos diários conforme a Figura 35. A navegação de saída do aplicativo é mais

intuitiva, pois se utiliza as funções do hardware do smartphone (i.e., botão de navegação para encerramento da aplicação).

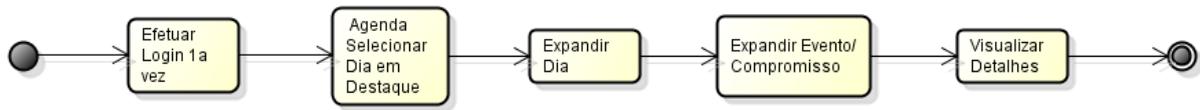


Figura 35 - Diagrama de atividades da aplicação móvel.

Acredita-se que o critério de navegação do MAAS seja de fácil entendimento e aprendido. A simplicidade impede que sejam percorridos diferentes caminhos de navegação no servidor web e também no aplicativo móvel, porém estas afirmações são baseadas apenas em ensaios de interação descritos na próxima seção.

6.5 ENSAIOS DE INTERAÇÃO

Para tornar o software utilizável, ele foi avaliado não só em termos de funcionalidade, mas também sob a visão de usabilidade. O objetivo foi demonstrar que o MAAS pode ser eficaz, eficiente, seguro, de boa utilidade, fácil de aprender e fácil de lembrar como se usa. Foram criados ensaios de interação para validar estas características, seguindo o processo sugerido pelo Laboratório de Utilizabilidade da Informática (LabiUtil) da Universidade Federal de Santa Catarina.

Análise Preliminar é a etapa inicial do processo na qual analistas tomam conhecimento dos fatos a cerca do software e de seu contexto de desenvolvimento e realizam um pré-diagnóstico dos problemas ergonômicos de sua interface com o usuário (LABIUTIL, 2013). Nessa etapa está contido o passo de reconhecimento do software, o qual aponta questões que abrangem a equipe de projeto do software, incluindo:

- **População alvo:** para que tipo de usuário foi destinado o software?;
Professores e alunos da UTFPR.
- **Tipo de tarefa que o software visa atender:** Que tipo de tarefa o usuário poderá desenvolver com este aplicativo?;
Marcação de provas, APS, trabalhos, mudanças de horário, palestra, avisos em geral, na agenda acadêmica do aluno e do professor.
- **Funções principais do produto:** Quais as funcionalidades que, na opinião dos projetistas, têm maior impacto na tarefa e na organização do trabalho?;

Cadastro de eventos e compromissos, atualização da agenda do aluno.

- **Equipe de projetistas:** Quantas pessoas foram envolvidas no projeto, houve ergonomistas?;

Duas pessoas, não ergonomistas.

- **Dados sobre o sistema:** Qual o ambiente de programação em que foi desenvolvido o software?;

Ambiente Visual Studio 2010 Express e visual Studio 2012, Linguagem C#.

O tipo de avaliação escolhida foi o *Check List*, embasado em (LABIUTIL, 2013). Este método foi aplicado, pois apesar de ter sua origem voltada à avaliação de Menus, ele atende aos requisitos do ensaio de interação proposto, já que o artefato foi desenvolvido com opções de funcionalidades que remetem a um menu. Esta escolha foi embasada nas vantagens de baixo custo e retorno rápido. Este método foi aplicado para avaliação do sistema desenvolvido pela equipe desenvolvedora e obteve-se os seguintes resultados:

- **Fontes:** manteve-se um padrão de estilo e tamanho da fonte, para todos os detalhes;
- **Cores:** fundo do painel branco e fonte em cor contrastante (preto);
- **Fundos:** as cores de fundo da tela são primárias remetem ao estilo Metro;
- **Ícones (botões):** todos os ícones seguem um padrão de *design*, estando contextualizados e alinhados;
- **Textos:** uso de caixa alta e baixa, alinhamento geralmente à esquerda;
- **Agrupamento por Formato:** os ícones e menus que tratam de assuntos em comum mantem seus padrões;
- **Densidade Informacional:** as telas possuem poucos textos e são divididas em áreas bem definidas para melhor visualização e compreensão do usuário, característica bastante presente na linguagem de design Metro, a qual as interfaces deste projeto foram baseadas. A única tela que contem maior quantidade de campos é a tela Agenda, porém esta está reservada apenas para esta finalidade;
- **Legibilidade:** estilo, cor e tamanho da fonte são favoráveis a leitura sem dificuldades;
- **Ações Explícitas:** por meio de botões e menus o usuário é capaz de controlar a navegação pelo ambiente desenvolvido;

- **Feedback:** as telas da aplicação contem *feedback* ao usuário nos ícones, em forma de mensagens quando o usuário passa o *mouse* sobre o ícone (*tooltip*) e no botão ajuda está o fluxo de interação da tela;

Desta forma obteve-se o pré-diagnóstico das interfaces, constatando que foram projetadas para oferecer a melhor interação possível entre usuário e aplicação. As funções do servidor web foram concentradas em uma única tela para remeter melhor à agenda de papel na qual o usuário pode fazer suas anotações em uma única folha, sem que precise a cada passo adentrar uma nova tela. O *design* foi baseado no estilo Metro, assim como o software mobile.

A definição dos *Scripts*, dos Cenários e da Amostra de usuários é a segunda etapa do processo na qual são elaborados os *scripts* para cada perfil de usuário, quais são os cenários que os usuários poderão vivenciar e qual será a amostra de usuários, qual será o seu tamanho da amostra por perfil.

Scripts envolvem o conjunto de tarefas que uma amostra de usuários representativos da população alvo do sistema deverá realizar durante os ensaios, conforme apresentado nos quadros 27 e 28.

Quadro 27 - Script executado pelo aluno.

SCRIPT ALUNO	
Passo	Descrição
1	Acessa o dispositivo no emulador, insere seu <i>Login</i> e senha e acessa o MAAS.
2	Clica em um dia da Agenda e visualiza os compromissos do dia;
3	Escolhe um compromisso para expandir;
4	Visualiza o Compromisso/ Evento;
5	Clica em voltar, consecutivamente até sair do aplicativo.

Fonte: Autoria Própria.

Quadro 28 - Script executado pelo professor

SCRIPT PROFESSOR	
Passo	Descrição
1	Acessa o servidor MAAS em sua sala de trabalho (no departamento), insere seu <i>Login</i> e senha e acessa o MAAS.
2	Clica em um dia da Agenda e visualiza as atividades do dia;
3	Escolhe uma atividade para expandir;
4	Visualiza a atividade;
5	Clica em editar;
6	Sobrescreve os campos que deseja alterar;
7	Clica em salvar;
8	Visualiza mensagem de confirmação;
9	Desloga da aplicação.

Fonte: Autoria Própria.

Cenários se referem às condições ambientais e organizacionais que serão trazidas para os testes representados no quadro 30.

Quadro 29 - Cenários de execução dos testes.

CENÁRIOS	
Cenário 1	Ambiente com conexão a internet sem fio.
Cenário 2	Sala de aula, com conexão cabeada.
Cenário 3	Sala dos professores munida de computador com acesso a internet.
Cenário 4	Ambiente sem acesso a internet.
Cenário 5	Professor sem acesso a computadores, mas tem um dispositivo com acesso à internet, acessa o servidor MAAS.
Cenário 6	Professor munido de dispositivo e sem acesso a internet, não tem acesso ao MAAS.

Fonte: Autoria Própria.

Durante a etapa de Definição acontece o reconhecimento do perfil do usuário. A primeira atividade de reconhecimento do usuário consiste em contatar pessoas do público-alvo em seus locais de trabalho e verificar se as pessoas contatadas possuem efetivamente o perfil imaginado pelos projetistas (LABIUTIL, 2013). Nessa etapa é possível pré-selecionar um grupo de usuários que poderão vir a participar dos ensaios. Os perfis dos usuários estão apresentados no quadro 4. Para o grupo deve ser explicada qual a finalidade da análise, quais os procedimentos que a equipe adotará e deixá-los livres para participar ou não da atividade proposta.

Quadro 30 - Perfis dos usuários.

PERFIL	NOME
Aluno	A1
Aluno	A2
Aluno	A3
Aluno	A4
Aluno	A5
Aluno	A6
Professor	P1
Professor	P2
Professor	P3

Fonte: Autoria Própria.

A finalidade do teste é descobrir se o usuário consegue usar de maneira instintiva e quase sem ajuda, apenas pelo conhecimento de agenda que o mesmo já possui e o *feedback* do software (*i.g.* função ajuda). Os procedimentos para a realização do teste são a preparação do ambiente de simulação do aplicativo móvel, instrução dos usuários, observar execução da tarefa e coleta dos resultados.

Coleta de informações sobre o usuário e sua tarefa pode ser necessária para realização de uma etapa mais detalhada, para obtenção de dados. Nela, o analista deve elaborar questionários destinados a buscar os dados de uma grande amostra de usuários. Além de ser enviado aos usuários, o questionário pode, também, servir de roteiro para entrevistas presenciais ou a distância.

Por meio de questionários pode-se coletar dados a respeito (LABIUTIL, 2013):

- **Dos recursos disponíveis, tanto técnicos quanto físicos, para a realização da tarefa:** também é importante saber qual o tipo de suporte que a instituição oferece aos usuários quanto a treinamento e apoio técnico.
- **Do contexto da tarefa:** durante as entrevistas e observações, os analistas tomam conhecimento do vocabulário utilizado pelos usuários, das diversas atividades que eles desenvolvem. Uma amostra do resultado final do trabalho dos usuários pode ser bastante útil para a montagem dos cenários;
- **Do nível dos usuários:** dados como formação geral e específica em informática e no aplicativo em análise, tempo de instituição, tempo na atividade desenvolvida e o conhecimento de outros aplicativos permitem diferenciar os usuários novatos e os experientes.
- **Da utilização do sistema:** em especial os questionários visam obter uma visão geral sobre a utilização de um sistema pronto ou em desenvolvimento. As questões devem estar direcionadas para as funcionalidades, buscando conhecer aquelas que o usuário considera de maior impacto positivo e negativo sobre seu trabalho. Deve-se também buscar conhecer as frequências de utilização de cada funcionalidade.

O instrumento de coleta de dados que cobre tais dados está presente no apêndice A. A terceira a última etapa do processo de ensaio de interação é a realização do ensaio que está apresentado na seção seguinte.

6.5.1 Resultados do Ensaio de Interação

Após aplicação do ensaio e do questionário a uma pequena amostra de professores e alunos (dois professores e cinco alunos), pôde-se observar como esses atores interagem com o software MAAS, sem possuírem nenhum conhecimento prévio de seu uso, bem como as suas visões de utilidade desse sistema caso tivessem o aplicativo disponível para uso.

Nas duas partes do ensaio, realizadas com professores e alunos, tiveram questões relacionadas ao nível de dificuldade de uso do software (numa escala de 1 à 5, sendo 1 o nível mais fácil) e à utilidade do software na organização de compromissos e eventos acadêmicos. No geral as respostas foram positivas nos dois lados, lembrando que todos os participantes do ensaio possuem conhecimentos prévios de informática e costumam usar frequentemente a internet para acessar e-mails e outros recursos.

Com relação ao nível de dificuldade do uso do software a Tabela 4 apresenta o nível apontado por cada um dos perfis analisados. Pode-se notar que não houve grandes problemas ao utilizar o software, tanto da parte do servidor, quanto da parte cliente.

Tabela 4 - Nível de dificuldade apontado por professores e alunos

Perfil	Nível de dificuldade
P1	2
P2	2
A1	1
A2	1
A3	1
A4	1

Com relação à utilização que o software MASS pode trazer para o cotidiano dos perfis, obteve-se 100% de aceitação, com comentários positivos afirmando que a agenda acadêmica facilitaria a organização de compromissos e eventos acadêmicos e que eles utilizariam o software se, futuramente, ele fosse adotado na Universidade.

Houve também contribuições para funcionalidades futuras que os usuários gostariam que o software englobasse, tais como reativar uma atividade (compromisso ou evento) que foi cancelada pelo professor e possibilitar o aluno de adicionar compromissos e eventos na aplicação móvel.

Como resultado geral dos ensaios de interação conclui-se que eles são muito úteis para o desenvolvimento de uma aplicação, pois possibilitou a descobrimento de falhas na usabilidade projetada permitindo aprimorar a interface e consertar as falhas detectadas.

6.6 TESTES DE SISTEMA

O teste do software é a investigação do software a fim de fornecer informações sobre sua qualidade em relação ao contexto em que ele deve operar (BORGES, 2013). O teste é um

processo que revela os erros de codificação, requisitos entre outros erros que podem aparecer no processo de engenharia de software (BORGES, 2013). A técnica utilizada para testar o MAAS foi o teste funcional, mais conhecido como teste de caixa preta (*black box*), cujo conteúdo é desconhecido e da qual só é possível visualizar as interfaces, ou seja, os dados de entrada fornecidos pelo usuário na interface e as respostas produzidas como saída também na interface. Essa técnica de teste funcional verifica as funções do sistema sem se preocupar com os detalhes de implementação. A técnica da caixa-preta é útil para todos os níveis de teste, desde os de componentes até os de sistema e aceitação, pois é fundamental que o software atenda suas especificações (ANDRADE, VIANA 2012).

Teste de sistema é uma fase de teste de funcional. O objetivo é executar o sistema sob o ponto de vista de seu usuário final, testando as funcionalidades em busca de falhas em relação aos objetivos desejados contidos na especificação do software (BORGES, 2013).

Os testes de sistema do MAAS foram escritos inspirados em uma parte da norma IEEE *Standard for Software and System Test Documentation* 829-2008 (Padrão 829 para Documentação de Teste de Software e Sistemas). Esta norma tem como um de seus objetivos definir o conteúdo e uso da documentação relacionada a testes. Este padrão que especifica a forma de um conjunto de documentos para uso em oito estágios definidos de teste de software, cada estágio potencialmente produzindo seu próprio tipo de documento distinto (IEEE, 2008). A norma especifica o formato desses documentos, dos quais a equipe optou por utilizar apenas cinco deles: Plano de Testes, Casos de Testes, Procedimentos de Testes, Relatório de Testes Incidentes e Relatório de Resumo de Testes (*Test Plan, Test Cases, Test Procedure, Interim Test Report, Test Log*) (IEEE, 2008, p15).

6.6.1 Com Relação à Aplicação Servidor

Para o desenvolvimento de testes do software foi utilizada uma documentação de teste baseado na norma IEEE 829, exemplificada no artigo de Blanco (2012). Os quadros a seguir fazem referência a uma sequência de testes realizados para a aplicação do servidor web.

Quadro 31 - Plano de Testes Servidor.No quadro Casos de Teste, Quadro 32, são apresentadas algumas funcionalidades que serão colocadas em ação para o teste, bem como o roteiro a ser seguido e se houve sucesso ou fracasso em sua realização. Nos procedimentos de

teste, Quadro 33 são apresentadas as pré-condições para realizar determinada ação, e as pós-condições resultantes da ação realizada quando obtém-se sucesso.

Quadro 31 - Plano de Testes Servidor.

PLANO DE TESTES	
Nome do Projeto	MAAS – Mobile Academic Appointment System – Modulo Servidor Web
Pessoas Envolvidas	Letícia Ueda - Desenvolvedor 1/ Vanessa Silva - Desenvolvedor 2
Funcionalidades	<i>Login</i> do usuário, cadastro de compromissos e eventos, edição de compromissos e eventos, visualização de compromissos e eventos e cancelamento de compromissos e eventos.
Equipamentos / Softwares	O sistema deve funcionar em um servidor web com acesso via browser em desktop e dispositivos móveis com acesso a web
Cronograma	Data de Início e Fim do Projeto: 02/05/2012 – 22/04/2013 Data de Início e Fim do Teste: 18/04/2013 a 22/04/2013
Local dos Testes	O software será testado nas máquinas dos usuários envolvidos em locais aleatórios
Crítérios para considerar o teste finalizado	Será considerado finalizado ao preencher a coluna do resultado do teste do documento de casos de teste como "Executado com Sucesso"
Observações	Foram criados os casos de testes, os relatórios de incidentes e o relatório resumo de teste

Fonte: Autoria Própria.

Quadro 32 - Casos de Testes do Servidor.

CASOS DE TESTES					
Id	Descrição	Roteiro	Resultado Esperado	Resultado do Desenvolvedor	Resultado do teste
01	Efetuar <i>login</i>	a)Acessar o aplicativo web; b)Inserir email; c)Inserir senha d)Clicar em entrar.	Exibir a tela agenda	Desenvolvedor 2: Executado com Sucesso	Desenvolvedor 1: Executado com Sucesso
02	Cadastrar Compromisso/ Cadastrar Evento	a)Selecionar dia; b)Selecionar grupo; c)Clicar em NOVO; d)Escolher radio button; d)Editar campos; e)Clicar em SALVAR.	Mensagem de Confirmação, dados cadastrados no BD.	Desenvolvedor 2: Não cadastra evento nem compromisso	Desenvolvedor 1: Executado com Sucesso
03	Editar Compromisso/ Editar Evento	a)Selecionar dia; b)Selecionar atividade; c)Clicar em Editar; d)Editar campos; e)Clicar em SALVAR.	Mensagem de Confirmação, dados cadastrados no BD.	Desenvolvedor 2: Não desabilita os campos após inclusão	Desenvolvedor 1: Executado com Sucesso
04	Visualizar Compromisso/Evento	a)Selecionar dia; b)Visualizar todas as atividades dos dia; c)Filtrar grupo; d)Visualizar todas as atividades do grupo para o dia selecionado;	Exibe a área de detalhes com os detalhes da atividade selecionada.	Desenvolvedor 2: Executado com Sucesso	Desenvolvedor 1: Executado com Sucesso
05	Cancelar Compromisso/ Evento	a)Selecionar dia; b)Selecionar atividade; c)Clicar em CANCELAR; d) Clicar em OK;	Mensagem de Confirmação, dados alterados no BD.	Desenvolvedor 2: Executado com Sucesso	Desenvolvedor 1: Executado com Sucesso

Fonte: Autoria Própria.

A partir deste guia, os casos de teste deverão ser executados. No caso apresentado para a aplicação do servidor web, algumas funcionalidades não chegaram ao resultado ideal na primeira bateria de testes, e tiveram de ser reavaliados para passarem pela fase de teste novamente.

Os procedimentos de testes, segundo IEEE 829-2008, contem instruções detalhadas para a configuração, execução e avaliação dos resultados de um dado caso de teste, esta é a documentação que especifica uma sequência de ações para a execução de um teste. Foram detalhadas em procedimentos de teste todas as funcionalidades do software que podem ser observadas no Apêndice B, na subseção corrente foi apenas apresentado o teste T002 – Cadastrar Compromisso que representa uma função crítica do software conforme apresentado no Quadro 33.

Quadro 33 - Procedimentos de Teste Servidor.

PROCEDIMENTOS DE TESTES			
T002 – Cadastrar Compromisso			
Pré-condições	Passo	Ação	Resultado Esperado
1. O usuário deve ter acesso à URL em que o sistema está hospedado. 2. O sistema deve estar disponível. 3. O usuário deve possuir <i>login</i> e senha para acesso ao sistema, o mesmo deve estar logado.	1	O usuário deve clicar em um dia do calendário	O MAAS exibe uma lista com todos os eventos e compromissos do dia.
	2	O usuário deve selecionar um grupo (Departamento, Curso ou Disciplina).	O MAAS filtra a lista exibindo apenas os eventos e compromissos do grupo selecionado.
	3	O usuário deve clicar no botão “ <i>Novo</i> ”.	O MAAS preenche o campo “ <i>Status</i> ” da área de detalhes com o status “ Ativo ”.
	4	O usuário deve selecionar o radio button “ <i>Compromisso</i> ”	O MAAS habilita os campos da área de detalhes com exceção dos campos “ <i>Status</i> ”, “ <i>Grupo</i> ” e “ <i>Id Grupo</i> ”.
	5	Preencher todos os campos e clicar no botão “ <i>Salvar</i> ”.	O MAAS exibe a mensagem de confirmação “Compromisso Cadastrado com Sucesso” e desabilita todos os campos da área de detalhes.
	6	Clicar no botão “ <i>Ok</i> ” da mensagem	O MAAS exibe o novo compromisso na lista de atividades. Importante: Todos os dados salvos devem constar na base de dados.

Fonte: Autoria Própria.

O formato e agrupamento dos relatórios de teste são opcionais, eles definidos pelo utilizador e variará de acordo com o assunto do teste (IEEE, 2008).

Quadro 34 - Relatório de Testes Incidentes Servidor.

RELATÓRIO DE TESTES INCIDENTES					
Id	Status	Responsável pela correção	Prioridade de correção	Descrição do erro	Data e Nome de quem Corrigiu
02	Pronto para Reteste	Desenvolvedor 2	Alta	Não cadastra compromisso.	21/04/2013 Desenvolvedor 2
03	Pronto para Reteste	Desenvolvedor 2	Alta	Não cadastra evento.	21/04/2013 Desenvolvedor 2
08	Pronto para Reteste	Desenvolvedor 2	Baixa	Não cadastra evento.	21/04/2013 Desenvolvedor 2

Fonte: Autoria Própria.

O Quadro 34 apresenta todos os procedimentos que apresentaram erro. Por fim, o Quadro 35 apresenta um relatório quantitativo do plano de teste. Este plano de testes cobre apenas os testes de sistema, não realizando testes unitários e estruturais, as tabelas a seguir apresentam os documentos de testes com relação ao servidor web.

Este relatório deve fornecer informações gerais que se aplicam a todos os testes, tais como os itens que estão em teste, as alterações dos documentos de teste, data de início e fim e nome das pessoas que executam o teste.

Quadro 35 - Relatório de Resumo de Testes Servidor.

RELATÓRIO DE RESUMO DE TESTES	
Nome Projeto	MAAS – Mobile Academic Appointment System
Data de início dos testes	18/04/2013
Data de fim dos testes	22/04/2013
Descrição dos testes	Com a criação dos casos de teste houve a detecção e a correção dos erros. O relatório de incidência de teste demonstrou de forma clara a correção do programador e o momento certo para o retorno para equipe de teste, na qual foi realizado o teste novamente.
Pessoas envolvidas	Desenvolvedor 1, Desenvolvedor 2.
NÚMEROS DOS TESTES	
Casos de testes criados antes do teste	6
Casos de testes criados durante o teste	2
Casos de testes executados	8
Casos de teste com erro	3
Casos de testes enviados para correção	3
PERCENTUAL	
Casos de testes executados	100%
Casos de testes executados com sucesso	62,5%
Casos de testes com incidência de erro	37,5%
Casos de testes corrigidos pelo desenvolvedor	100%

Fonte: Autoria Própria.

6.6.2 Com Relação à Aplicação Cliente

Para o desenvolvimento de testes do software foi utilizada uma documentação de teste baseado na norma IEEE 829, exemplificada no artigo de Blanco (2012). As tabelas abaixo fazem referência a uma sequência de testes realizados para a aplicação do dispositivo móvel. O Quadro 36, descreve as funcionalidades que devem ser cobertas pelos casos de teste.

Quadro 36 - Plano de Testes Dispositivo Móvel.

PLANO DE TESTES	
Nome do Projeto	MAAS – Mobile Academic Appointment System – Modulo Aplicação móvel
Pessoas Envolvidas	Letícia Ueda - Desenvolvedor 1/ Vanessa Silva - Desenvolvedor 2
Funcionalidades	Efetuar, visualizar agenda por mês, expandir compromissos e/ou eventos do dia, expandir detalhes de um evento/compromisso.
Equipamentos / Softwares	O sistema deve funcionar em um servidor web com acesso via browser em desktop e dispositivos móveis com acesso a web
Cronograma	Data de Início e Fim do Projeto: 02/05/2012 – 22/04/2013 Data de Início e Fim do Teste: 18/04/2013 a 22/04/2013
Local dos Testes	A aplicação móvel será testada no emulador do <i>Visual Studio</i> 2010 por pessoas envolvidas em locais aleatórios
Critérios para considerar o teste finalizado	Será considerado finalizado ao preencher a coluna do resultado do teste do documento de casos de teste como "Executado com Sucesso"
Observações	Foram criados os casos de testes, os relatórios de incidentes e o relatório resumo de teste

Fonte: Autoria Própria.

Os casos de teste observados no Quadro 37 apresentam de forma geral os casos de testes com seus roteiros de execução, resultado esperado e resultados dos testes.

O Quadro 38 apresenta as pré-condições para realizar determinada ação e os passos que conduzem as pós-condições resultantes da ação realizada quando se obtém sucesso.

Quadro 37 - Casos de Testes do Dispositivo Móvel.

CASOS DE TESTES					
Id	Descrição	Roteiro	Resultado Esperado	Resultado do Desenvolvedor	Resultado do teste
01	Efetuar <i>login</i>	a) Acessar o aplicativo móvel; b) Inserir número de matrícula; c) Inserir senha da aplicação d) Clicar em entrar.	Exibir a tela que representa a agenda do aluno;	Desenvolvedor 1: executado com Sucesso	Desenvolvedor 2: Executado com Sucesso
02	Vizualizar agenda	a) Navegar pelos meses clicando nos botões em forma de flecha;	Exibir agenda redesenhada com o mês seguinte ou anterior;	Desenvolvedor 1: executado com sucesso	Desenvolvedor 2: Executado com Sucesso
03	Expandir atividades do dia	a) Selecionar dia destacado;	Exibir na tela lista de compromissos e eventos de maneira simples, marcados para o dia selecionado;	Desenvolvedor 1: executado com sucesso	Desenvolvedor 2: Executado com Sucesso
04	Visualizar detalhes d Compromisso/Evento	a) Selecionar compromisso ou evento listado;	Exibir na tela todos os detalhes do compromisso/evento	Desenvolvedor 1: executado com sucesso;	Desenvolvedor 2: Executado com Sucesso

Fonte: Autoria Própria.

Foram detalhadas em procedimentos de teste todas as funcionalidades do software que podem ser observadas no Apêndice B, na subseção corrente foi apenas apresentado o teste T001 – Efetuar *Login* que representa uma das funções do software conforme apresentado no Quadro 38.

Quadro 38 - Procedimentos de Testes do Dispositivo Móvel.

PROCEDIMENTOS DE TESTES			
T001 – Efetuar <i>Login</i>			
Pré-condições	Passo	Ação	Resultado Esperado
1. O usuário deve ter instalado em seu <i>smartphone</i> o aplicativo. 3. O usuário deve possuir número de matrícula e senha para acesso ao sistema.	1	O usuário executa a aplicação em seu <i>smartphone</i> .	O MAAS exibe sua tela de login, se esta for a primeira vez que é utilizado.
	2	O usuário deve preencher os campos “ <i>Matrícula</i> ” e “ <i>Senha</i> ” e então clicar no botão “ <i>Entrar</i> ”.	O MAAS autentica o usuário, verificando na base de dados do servidor se o número de matrícula e senha existem, e então exibe a tela Agenda;

Fonte: Autoria Própria.

Quadro 39 - Relatório de Resumo de Testes do Dispositivo Móvel.

RELATÓRIO DE RESUMO DE TESTES	
Nome Projeto	MAAS – Mobile Academic Appointment System
Data de início dos testes	18/04/2013
Data de fim dos testes	22/04/2013
Descrição dos testes	Com a criação dos casos de teste pôde-se verificar que a navegação e as funcionalidades da aplicação móvel estão operando adequadamente.
Pessoas envolvidas	Desenvolvedor 1, Desenvolvedor 2.
NÚMEROS DOS TESTES	
Casos de testes criados antes do teste	4
Casos de testes criados durante o teste	0
Casos de testes executados	4
Casos de teste com erro	0
Casos de testes enviados para correção	0
PERCENTUAL	
Casos de testes executados	100%
Casos de testes executados com sucesso	100%
Casos de testes com incidência de erro	0%
Casos de testes corrigidos pelo desenvolvedor	0%

Fonte: Autoria Própria.

A compreensão e realização dos testes reforçou o valor que o teste de software agrega ao artefato. Este faz-se válido, pois menor que seja a quantidade de testes estes tem capacidade de indicar falhas e riscos mesmo na elaboração dos testes pode-se detectar problemas na própria documentação, reduzindo custos e retrabalho , tornando indispensável esta fase do desenvolvimento de software.

7 CONCLUSÃO

Neste trabalho de conclusão de curso foram apresentados alguns problemas reais que são enfrentados por alunos e professores no ambiente acadêmico da UTFPR. Um deles diz respeito à ineficiência na comunicação entre alunos e professores fora da sala de aula, o que pode trazer prejuízos tanto para os alunos, quanto para os professores. Outro problema remete a organização e armazenamento de inúmeras datas importantes de compromissos e eventos acadêmicos dos alunos. Muitos deles cursam diversas disciplinas, e assim, possuem várias atividades acadêmicas para lembrar e organizar. Isso acaba sendo difícil quando não se tem uma ferramenta de auxílio.

O protótipo MAAS desenvolvido neste trabalho foi proposto como uma potencial solução para amenizar as dificuldades citadas à cima. Ele serviria como um meio de comunicação eficiente e rápido entre professores e Universidade com os alunos, no que diz respeito aos compromissos e eventos acadêmicos, e também como um meio dos alunos terem armazenado em seus smartphones as atividades acadêmicas importantes, possibilitando sua visualização.

Abrangendo essa situação, o protótipo MAAS foi criado. Suas características e funcionalidades atendem os requisitos para amenizar os problemas acadêmicos tratados neste trabalho, tornando-o um Sistema de Informação eficiente.

A partir da criação da proposta do software MAAS, a equipe passou a buscar um conhecimento mais aprofundado sobre diversos assuntos referentes às dificuldades acadêmicas abordadas, soluções existentes para essas dificuldades, a influência dos smartphones nos dias atuais e também sobre algumas ferramentas e tecnologias utilizadas na implementação do software. Essa análise também possibilitou justificar as escolhas realizadas no desenvolver deste trabalho. Após análise, a metodologia foi criada para organizar o andamento do trabalho num período de tempo estimado, com o objetivo de reduzir riscos de atrasos. Finalmente pôde-se começar a trabalhar na especificação detalhada do software e enfim partir para sua implementação. A equipe ainda identificou a necessidade de realizar ensaios de interação com uma pequena amostra do público alvo (alunos e professores) para obter alguns resultados referentes à usabilidade do software, além da realização de testes de software para obter uma medida de qualidade.

Como produto desta pesquisa tecnológica obteve-se um protótipo de agenda acadêmica que, a partir de requisições efetuadas ao servidor web, atualiza dados acadêmicos individuais dos alunos de uma Universidade, praticamente em tempo real. A utilização da tecnologia móvel foi um grande diferencial para atingir todos os objetivos deste trabalho. Com a aplicação da solução MAAS nas Universidades os alunos poderiam ter em mãos uma agenda eletrônica com as datas de todas as provas, apresentações, entregas de trabalhos e palestras, que são cadastradas pelos professores e membros da Universidade, recebendo atualizações em tempo real de qualquer mudança que possa acontecer em sua rotina. Outras vantagens da utilização deste software seria a confiabilidade da fonte das informações da agenda e também a distribuição da informação, o que não se pode ter quando, por exemplo, recados são deixados informalmente em quadros negros. Além disso, todos os alunos matriculados na instituição e que fazem parte do grupo de interesse do compromisso ou evento vão receber em seus smartphones uma atualização assim que se conectarem à internet, o que se torna mais eficiente que o uso do e-mail pode gerar falha na comunicação caso seja cadastrado incorretamente pelo professor. Ainda foi possível concluir que a utilidade deste software possui algumas limitações. É necessário que todos os professores e responsáveis pela instituição utilizem o software para cadastrar as atividades acadêmicas, pois uma agenda acadêmica que não possua todos os compromissos e eventos acadêmicos do aluno deixa de ser confiável inteiramente confiável. Ele precisará de outro meio para gerenciar as atividades dos professores que não fizerem uso do MAAS o que compromete a eficiência do sistema. Também existe uma questão cultural no que diz respeito à utilização de um novo sistema. Os professores que já possuem um meio de divulgar suas atividades, pode optar por não utilizar o novo software.

Este protótipo proporcionou o despertar de percepções que envolvem os sistemas informacionais, como a importância da área de humanidades e design, por exemplo, para elaborar um software que seja utilizável deve-se pensar no usuário e como usuário, papel de extrema dificuldade para o profissional da computação, porém o bacharel em sistemas de informação não pode limitar-se ao mundo da computação, mas sim analisar e utilizar a gama de disciplinas que norteiam os sistemas informacionais, para prover bom uso, eficiência e bom senso no trabalho, seja este artefato, serviço ou pesquisa.

Enxergar essa solução para um problema comum na Universidade não foi algo trivial. Além da vontade em atacar um problema vivenciado pela equipe, houve interesse em utilizar um grande número de conteúdos estudados no decorrer do período de graduação. Isso pôde demonstrar o conhecimento nas áreas compreendidas no desenvolvimento do software e

também possibilitou que a equipe complementasse o aprendizado por meio das dificuldades enfrentadas ao longo do desenvolvimento, tornando a realização deste trabalho de conclusão de curso algo importante e gratificante para a equipe.

7.1 TRABALHOS FUTUROS

O software apresentado neste Trabalho de Conclusão de Curso é apenas um protótipo do que se observa como um software ideal, e diversas melhorias podem ser realizadas para aumentar sua qualidade e eficiência.

- A primeira delas seria poder integrá-lo com o sistema acadêmico da Universidade à que serve. Eliminando a necessidade de criar um meio para obter os dados acadêmicos necessários para o bom funcionamento da ferramenta.
- Para atingir a máxima acessibilidade do software, haveria necessidade de desenvolver clientes para outras plataformas móveis como Android, iOS, entre outros.
- Desenvolver uma aplicação móvel para a utilização dos professores.
- Trabalhar as interfaces da aplicação web e móvel para serem fiéis aos princípios da linguagem de *design* Metro e atenderem os princípios de *design* de interação.
- Adicionar funcionalidades como a aceitação de eventos da Universidade para geração de relatórios e estatísticas sobre o número de participantes
- Apresentar notificação de atualizações de agenda aos alunos, mesmo não estando com a aplicação aberta no smartphone.
- Incluir módulo para usuários de outros setores da universidade.

REFERENCIAS

All About Windows Phone. **Review: Group me.** 2012. Disponível em: < http://allaboutwindowsphone.com/reviews/item/14487_Group_Me.php> Acesso em: 25 abr. 2012.

ANDRADE, Ana P.; VIANA, Phil. **Criação e Geração de Planos de Teste de Software.** Dez. 2013. Disponível em: <http://www.ibm.com/developerworks/br/local/rational/criacao_geracao_planos_testes_softw_are/>. Acesso em: 19 Abr. 2013.

BALSAMIQ. **Mockups Web Demo.** 2013. Disponível em: <<http://builds.balsamiq.com/b/mockups-web-demo/>>. Acesso em: 20 jan. 2013.

BERNERS-LEE, T. **Web Services: Program Integration across Application and Organization boundaries.** 2009. Disponível em: <<http://www.w3.org/DesignIssues/WebServices.html>>. Acesso em: 16 abr. 2013.

BORGES, Gerson. **Engenharia de Software: metodologias de Desenvolvimento de Sistemas.** 2013, p. 18-25. Disponível em: <http://www.professorgersonborges.com.br/site/pdf/apostila/Engenharia/Intru_Engenharia_Software.pdf#page=92>. Acesso em: 18 abr. 2013

BURGOS, Pedro. **Nokia e Android crescem no mercado brasileiro de smartphones.** Gizmodo, 2010. Disponível em: < <http://www.gizmodo.com.br/conteudo/nokia-e-android-crescem-no-mercado-brasileiro-de-smartphones/>>. Acesso em: 05 abr. 2012.

COELHO, Claudio Ulisses; HAGUENAUER, Cristina. “**As Tecnologias da Informação e da Comunicação e sua Influência na Mudança de Perfil e da Postura do Professor**”; Revista Educação Online, Laboratório de Pesquisa em Tecnologias da Informação e Comunicação - LATEC/UFRJ, Janeiro/Abril 2007, vol.1 nº1 ISSN 1983-2664. Disponível em: <<http://www.latec.ufrj.br/revistas/index.php?journal=educaonline&page=article&op=view&path%5B%5D=144&path%5B%5D=263>>. Acesso em: 10 mar. 2012

FAYAD, M. Schmidt, D. **Object-Oriented Application Frameworks.** Communications of the ACM, New York, v. 40, n.10, p. 32-38, Oct. 1997. Disponível em: <<http://dl.acm.org/citation.cfm?id=262798>>. Acesso em: 06 abr. 2013.

GAZETA DO POVO, Tecnologia. **Governo desonera smartphones de até R\$ 1.500.** Disponível em:

<<http://www.gazetadopovo.com.br/tecnologia/conteudo.phtml?tl=1&id=1361531&tit=Governo-desonera-smartphones-de-ate-R-1500>>. Acesso em: 09 abr. 2013.

GUIMARÃES, Célio C. **Fundamentos de Bancos de Dados: modelagem, projeto e linguagem SQL**. Editora Unicamp - 2º ed. Campinas, SP, p. 19-23, 2008.

IEEE. **Standard for Software and System Test Documentation 829-2008**. Disponível em: <<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true&arnumber=5983353>>. Acesso em: 15 mar. 2013.

INTERACTION DESIGN FOUNDATION. **Mock-ups**. 2010. Disponível em: <<http://www.interaction-design.org/encyclopedia/mock-ups.html>>. Acesso em: 15 mar. 2013.

JSON. **Introducing JSON**. 2013. Disponível em: <<http://www.json.org/>>. Acesso em: 22 de abr. 2013.

MACORATTI, José C. **Padrões de Projeto: O modelo MVC - Model View Controller** Disponível em: <http://www.macoratti.net/vbn_mvc.htm>. Acesso em: 18 mar. 2013.

LABORATÓRIO DE UTILIZABILIDADE DA INFORMÁTICA, LabIUtil. **Ensaio de interação**. Universidade Federal de Santa Catarina. 2013. Disponível em: <http://www.labiutil.inf.ufsc.br/hiperdocumento/unidade3_3_2_3_1.html>. Acesso em: 15/08/2012.

LEITE, Umbelina R.; TAMAYO, Álvaro; GÜNTHER, Hartmut. **Organização do uso do tempo e valores de universitários**. Avaliação Psicológica, 2003, 57-66. Disponível em: <http://pepsic.bvsalud.org/scielo.php?script=sci_arttext&pid=S1677-04712003000100007>. Acesso em: 03 abr. 2012.

LORS, Chris. **Evolution of the Eletronic Calendar: Introducing Social Calendarig**. Disponível em: <<http://chrislord.net/files/paper.pdf>> Acesso em: 12 mar.2012.

MICROSOFT. **Metro Design Language of Windows Phone 7**. Tutorials. Disponível em: <<http://www.microsoft.com/design/toolbox/tutorials/windows-phone-7/metro/>>. Acesso em: 06 jul. 2012.

MOÇO, Sueli de S. **O Uso De Cenários Como Uma Técnica De Apoio Para Avaliações Ergonômicas De Softwares Interativos**. 1996. 130 f. Tese (Mestrado em Engenharia de Produção) – Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Santa Catarina. 1996. Disponível em: <<http://www.eps.ufsc.br/disserta96/sueli/cap4/cap4.htm> >. Acesso em: 15 fev. 2013.

OISHI, Lindsay. **Google apps goes to school**. Technology & Learning, 2007, ISSN 2177-3866. Disponível em: <<http://www.techlearning.com/article/working-together/44268>> Acesso em: 10 mar. 2012.

OLIVEIRA, Osvaldo Luiz de; BARANAUSKAS, M. Cecília C. **Análise Glossemática da Estrutura das Linguagens de Interface Humano-Computador**. Portal Hjelmslev Glossemática, 27 out. 2008. Disponível em: <http://glossematics.org/forum/pdfs/OLIVEIRA_Analise.pdf>. Acesso em: 02 jul. 2012.

PELLANDA, E. **Comunicação móvel: das potencialidades aos usos e aplicações**. Em Questão, Brasil, v. 15, n. 1, 2009. Disponível em: <<http://www.revistas.univerciencia.org/index.php/revistaemquestao/article/view/6627/6010>>. Acesso em: 25 jun. 2012.

PERES, Gilberto; TRIBUG, Emmanuel X.; GERALDO, Tatiana. **“Fatores que Determinam a Aquisição de Smartphones pelos Consumidores”**; Seminários de Administração, Setembro 2010, ISSN 2177-3866. Disponível em: <<http://www.ead.fea.usp.br/semead/13semead/resultado/trabalhosPDF/842.pdf>>. Acesso em: 10 mar. 2012.

PLAY, Google Store. **School Assistant**. Disponível em: <<https://play.google.com/store/apps/details?id=com.school>>. Acesso em: 06 mar. 2013.

PLAY, Google Store. **My Class Schedule**. Disponível em: <<https://play.google.com/store/apps/details?id=de.rakuun.MyClassSchedule.free>>. Acesso em: 06 mar. 2013.

PREECE, Jeniffer; ROGERS, Yvonne; SHARP, Hellen. **Design de Interação: além da interação homem-computador**. Editora Bookman - 2º ed. Porto Alegre, RS, p. 28-266, 2008.

R.S. Pressman, **Engenharia de Software**, Rio de Janeiro: McGraw Hill, 5ª edição, 2002.

SACCOL, Amarolinda Zanela; REINHARD, Nicolau. **Tecnologias de informação móveis, sem fio e ubíquas: definições, estado-da-arte e oportunidades de pesquisa**. Rev. adm. contemp., Curitiba, v. 11, n. 4, Dec. 2007. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1415-6552007000400009&lng=en&nrm=iso>. Acesso em: 25 jun. 2012.

SAUVÉ, Jacques P. **Diagrama de Atividades**. 2013. Disponível em: <http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/uml/diagramas/atividades/diag_atividades.htm>. Acesso em: 23 abr. 2013.

SHIN, D.; SHIN, Y.; CHOO, H.; BEOM, K. **Smartphones as smart pedagogical tools: Implications for smartphones as u-learning devices**. Computers in Human Behavior, Volume 27, Issue 6, November 2011, 2207-2214, ISSN 0747-5632, 10.1016/j.chb.2011.06.017. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0747563211001233>>. Acesso em: 09 mar. 2012.

SILVINO, Alexandre Magno Dias; ABRAHÃO, Júlia Issy. **Navegabilidade e inclusão digital: usabilidade e competência**. RAE electron., São Paulo, v. 2, n. 2, Dec. 2003. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1676-56482003000200002&lng=en&nrm=iso>. Acesso em: 23 mar. 2013.

SOMMERVILLE, I. Engenharia de Software. 8. ed. São Paulo: Pearson Addison-Wesley, 2007.

TECHTUDO, Notícias. **Vantagens e Problemas do Windows Phone 7**. Disponível em: <<http://www.techtudo.com.br/artigos/noticia/2010/12/5-vantagens-e-5-problemas-do-windows-phone-7.html>>. Acesso em: 06 jun. 2012.

TECHTUDO, Notícias. **Tudo Sobre Windows Phone**. Disponível em: <<http://www.techtudo.com.br/tudo-sobre/windows-phone.html>>. Acesso em: 06 jul. 2012.

TECNOBLOG. **Windows Phone**. Disponível em: <<http://tecnoblog.net/sobre/windows-phone/>>. Acesso em: 06 abr. 2012.

TIECHER, Anderson. **Identificar os atributos de produto, no mercado de smartphones, que agreguem valor na percepção dos estudantes da escola de administração da Universidade Federal do Rio Grande do Sul**. Porto Alegre, 2011. Disponível em: <<http://www.lume.ufrgs.br/bitstream/handle/10183/33341/000787035.pdf?sequence=1>>. Acesso em: 03 abr. 2012.

TI, Profissionais, Notícias. **Qual é o melhor Banco de Dados: ORACLE ou SQL SERVER?**. Disponível em: <<http://www.profissionaisiti.com.br/2012/01/qual-e-o-melhor-banco-de-dados-oracle-ou-sql-server/>>. Acesso em: 06 abr. 2013.

WIKIPEDIA. **Metro (design language)**. Disponível em: <[http://en.wikipedia.org/wiki/Metro_\(design_language\)](http://en.wikipedia.org/wiki/Metro_(design_language))>. Acesso em: 10 abr. 2012.

APÊNDICES

APÊNDICE A – QUESTIONÁRIO ENSAIOS DE INTERAÇÃO

Ensaio de Interação – MAAS – Data: ___/___/_____ Perfil: Aluno

QUESTIONÁRIO

Este Ensaio de Interação tem o objetivo de avaliar a interface do protótipo MAAS diretamente com os seus usuários, para fins de coleta de informações positivas e negativas, opiniões e expectativas, sobre a interface e utilidade do produto.

- a) O software é capaz de ser aprendido sem treinamento? (Somente com as dicas fornecidas pelo próprio software).
- b) Avaliar a dificuldade de uso de 1 a 5 (1 fácil – 5 difícil):
- c) O vocabulário está adequado?
- d) Este software lhe ajudaria a organizar os seus compromissos acadêmicos?
- e) Conhecimentos prévios de informática:
 - a. Tem convivência com informática?
 - b. Utiliza email pessoal e/ou redes sociais? Se sim, Com que frequência?
- f) Avalie a importância ter compromissos de disciplina em uma agenda móvel:
- g) Avalie importância ter eventos acadêmicos em uma agenda móvel:
- h) Avalie importância de Visualizar Agenda:
- i) Avalie as dificuldades de Visualizar Agenda:
- j) Gostaria de deixar algum comentário ou sugestão?

Ensaio de Interação – MAAS – Data: ___/___/_____ Perfil: Professor

QUESTIONÁRIO

Este Ensaio de Interação tem o objetivo de avaliar a interface do protótipo MAAS diretamente com os seus usuários, para fins de coleta de informações positivas e negativas, opiniões e expectativas, sobre a interface e utilidade do produto.

- a) O software é capaz de ser aprendido sem treinamento? (Somente com as dicas fornecidas pelo próprio software).
- b) Avaliar a dificuldade de uso de 1 a 5 (1 fácil – 5 difícil):
- c) O vocabulário está adequado?

- d) Este software lhe ajudaria a organizar os seus compromissos acadêmicos?
- e) Conhecimentos prévios de informática:
 - a. Tem convivência com informática?
 - b. Utiliza email pessoal e/ou redes sociais? Se sim, Com que frequência?
- a) Avalie a importância e dificuldades de Cadastrar um Compromisso:
- b) Avalie a importância e dificuldades de Cadastrar Evento:
- c) Avalie a importância e dificuldades de Editar Evento:
- d) Avalie a importância e dificuldades de Editar Compromisso:
- e) Avalie a importância e dificuldades de Cancelar Compromisso:
- f) Avalie a importância e dificuldades de Cancelar Evento:
- g) Avalie a importância de Visualizar Agenda:
- h) Avalie a dificuldades de Visualizar Agenda:
- i) Gostaria de deixar algum comentário ou sugestão?

APÊNDICE B – PROCEDIMENTOS DE TESTES

Procedimentos de testes do servidor web.

PROCEDIMENTOS DE TESTES			
T001 – Efetuar <i>Login</i>			
Pré-condições	Passo	Ação	Resultado Esperado
1. O usuário deve ter acesso à URL em que o sistema está hospedado. 2. O sistema deve estar disponível. 3. O usuário deve possuir <i>login</i> e senha para acesso ao sistema.	1	O usuário digitar a url http://200.134.10.40/MaasPortal/View/Login.aspx no navegador web de sua preferencia.	O MAAS exibe sua tela de <i>login</i> .
	2	O usuário deve preencher os campos “ <i>Usuário</i> ” e “ <i>Senha</i> ” e então clicar no botão “ <i>Entrar</i> ”.	O MAAS autentica o usuário, verificando na base de dados se <i>login</i> e senha existem, e então exibe a tela Agenda; Importante: Verificar se os dados de <i>login</i> existem na base de dados.

T002 – Cadastrar Compromisso			
Pré-condições	Passo	Ação	Resultado Esperado
<p>1. O usuário deve ter acesso à URL em que o sistema está hospedado.</p> <p>2. O sistema deve estar disponível.</p> <p>3. O usuário deve possuir <i>login</i> e senha para acesso ao sistema, o mesmo deve estar logado.</p>	1	O usuário deve clicar em um dia do calendário	O MAAS exibe uma lista com todos os eventos e compromissos do dia.
	2	O usuário deve selecionar um grupo (Departamento, Curso ou Disciplina).	O MAAS filtra a lista exibindo apenas os eventos e compromissos do grupo selecionado.
	3	O usuário deve clicar no botão “ <i>Novo</i> ”.	O MAAS preenche o campo “ <i>Status</i> ” da área de detalhes com o status “ Ativo ”.
	4	O usuário deve selecionar o radio button “ <i>Compromisso</i> ”	O MAAS habilita os campos da área de detalhes com exceção dos campos “ <i>Status</i> ”, “ <i>Grupo</i> ” e “ <i>Id Grupo</i> ”.
	5	Preencher todos os campos e clicar no botão “ <i>Salvar</i> ”.	O MAAS exibe a mensagem de confirmação “Compromisso Cadastrado com Sucesso” e desabilita todos os campos da área de detalhes.
	6	Clicar no botão “ <i>Ok</i> ” da mensagem	O MAAS exibe o novo compromisso na lista de atividades. Importante: Todos os dados salvos devem constar na base de dados.

PROCEDIMENTOS DE TESTES			
T003 – Cadastrar Evento			
Pré-condições	Passo	Ação	Resultado Esperado
1. O usuário deve ter acesso à URL em que o sistema está hospedado. 2. O sistema deve estar disponível. 3. O usuário deve possuir <i>login</i> e senha para acesso ao sistema, o mesmo deve estar logado.	1	O usuário deve clicar em um dia do calendário	O MAAS exibe uma lista com todos os eventos e compromissos do dia.
	2	O usuário deve selecionar um grupo (Departamento, Curso ou Disciplina).	O MAAS filtra a lista exibindo apenas os eventos e compromissos do grupo selecionado.
	3	O usuário deve clicar no botão “ <i>Novo</i> ”.	O MAAS preenche o campo “ <i>Status</i> ” da área de detalhes com o status “ Ativo ”.
	4	O usuário deve selecionar o radio button “ <i>Evento</i> ”	O MAAS habilita os campos da área de detalhes com exceção dos campos “ <i>Status</i> ”, “ <i>Grupo</i> ” e “ <i>Id Grupo</i> ”.
	5	Preencher todos os campos e clicar no botão “ <i>Salvar</i> ”.	O MAAS exibe a mensagem de confirmação “Evento Cadastrado com Sucesso” e desabilita todos os campos da área de detalhes.
	6	Clicar no botão “ <i>Ok</i> ” da mensagem	O MAAS exibe o novo evento na lista de atividades. Importante: Todos os dados salvos devem constar na base de dados.

PROCEDIMENTOS DE TESTES			
T004 – Editar Compromisso			
Pré-condições	Passo	Ação	Resultado Esperado
1. O usuário deve ter acesso à URL em que o sistema está hospedado. 2. O sistema deve estar disponível. 3. O usuário deve possuir <i>login</i> e senha para acesso ao sistema, o mesmo deve estar logado.	1	O usuário deve clicar em um dia do calendário	O MAAS exibe uma lista com todos os eventos e compromissos do dia.
	2	O usuário deve selecionar um compromisso para edição.	O MAAS preenche os campos da área de detalhes com os dados da atividade escolhida, todos os campos devem estar desabilitados.
	3	O usuário deve clicar no botão “ <i>Editar</i> ”.	O MAAS habilita os campos da área de detalhes com exceção dos campos “ <i>Status</i> ”, “ <i>Grupo</i> ” e “ <i>Id Grupo</i> ”.
	4	Preencher todos os campos editáveis, clicar no botão “ <i>Salvar</i> ”.	O MAAS exibe a mensagem de confirmação “Compromisso Atualizado com Sucesso” e desabilita todos os campos da área de detalhes.
	5	Clicar no botão “ <i>Ok</i> ” da mensagem.	Importante: Todos os dados alterados devem constar na base de dados.

PROCEDIMENTOS DE TESTES			
T005 – Editar Evento			
Pré-condições	Passo	Ação	Resultado Esperado
1. O usuário deve ter acesso à URL em que o sistema está hospedado. 2. O sistema deve estar disponível. 3. O usuário deve possuir <i>login</i> e senha para acesso ao sistema, o mesmo deve estar logado.	1	O usuário deve clicar em um dia do calendário	O MAAS exibe uma lista com todos os eventos e compromissos do dia.
	2	O usuário deve selecionar um evento para edição.	O MAAS preenche os campos da área de detalhes com os dados da atividade escolhida, todos os campos devem estar desabilitados.
	3	O usuário deve clicar no botão “ <i>Editar</i> ”.	O MAAS habilita os campos da área de detalhes com exceção dos campos “ <i>Status</i> ”, “ <i>Grupo</i> ” e “ <i>Id Grupo</i> ”.
	4	Preencher todos os campos editáveis, clicar no botão “ <i>Salvar</i> ”.	O MAAS exibe a mensagem de confirmação “Evento Atualizado com Sucesso” e desabilita todos os campos da área de detalhes.
	5	Clicar no botão “ <i>Ok</i> ” da mensagem.	Importante: Todos os dados alterados devem constar na base de dados.

PROCEDIMENTOS DE TESTES			
T006 – Visualizar Compromisso			
Pré-condições	Passo	Ação	Resultado Esperado
1. O usuário deve ter acesso à URL em que o sistema está hospedado. 2. O sistema deve estar disponível. 3. O usuário deve possuir <i>login</i> e senha para acesso ao sistema, o mesmo deve estar logado.	1	O usuário deve clicar em um dia do calendário	O MAAS exibe uma lista com todos os eventos e compromissos do dia.
	2	O usuário visualiza todos os eventos e compromissos do dia. O usuário deve selecionar um grupo (Departamento, Curso ou Disciplina).	O MAAS filtra a lista exibindo apenas os eventos e compromissos do grupo selecionado.
	3	O usuário Visualiza os eventos e compromissos do grupo. O usuário deve clicar em um compromisso da lista.	O MAAS preenche todos os campos da área de detalhes com os dados do compromisso escolhido.

PROCEDIMENTOS DE TESTES			
T007 – Visualizar Evento			
Pré-condições	Passo	Ação	Resultado Esperado
1. O usuário deve ter acesso à URL em que o sistema está hospedado. 2. O sistema deve estar disponível. 3. O usuário deve possuir <i>login</i> e senha para acesso ao sistema, o mesmo deve estar logado.	1	O usuário deve clicar em um dia do calendário	O MAAS exibe uma lista com todos os eventos e compromissos do dia.
	2	O usuário visualiza todos os eventos e compromissos do dia. O usuário deve selecionar um grupo (Departamento, Curso ou Disciplina).	O MAAS filtra a lista exibindo apenas os eventos e compromissos do grupo selecionado.
	3	O usuário Visualiza os eventos e compromissos do grupo. O usuário deve clicar em um evento da lista.	O MAAS preenche todos os campos da área de detalhes com os dados do evento escolhido.

PROCEDIMENTOS DE TESTES			
T008 – Cancelar Compromisso/Evento			
Pré-condições	Passo	Ação	Resultado Esperado
1. O usuário deve ter acesso à URL em que o sistema está hospedado. 2. O sistema deve estar disponível. 3. O usuário deve possuir <i>login</i> e senha para acesso ao sistema, o mesmo deve estar logado.	1	O usuário deve clicar em um dia do calendário	O MAAS exibe uma lista com todos os eventos e compromissos do dia.
	2	O usuário deve selecionar um compromisso para edição.	O MAAS preenche os campos da área de detalhes com os dados da atividade escolhida, todos os campos devem estar desabilitados.
	3	O usuário deve clicar no botão “ <i>Cancelar</i> ”.	O MAAS exibe a mensagem de diálogo “Deseja cancelar o registro?”. Com as opções OK e Cancel .
	4	Clicar no botão “ <i>Ok</i> ” da mensagem.	O MAAS exibe a mensagem de confirmação “Cancelado com Sucesso!” e altera o campo “ <i>Status</i> ” de Ativo para Cancelado. Importante: Todos os dados alterados devem constar na base de dados.

Procedimentos de testes do dispositivo móvel.

PROCEDIMENTOS DE TESTES			
T001 – Efetuar <i>Login</i>			
Pré-condições	Passo	Ação	Resultado Esperado
1. O usuário deve ter instalado em seu <i>smartphone</i> o aplicativo. 3. O usuário deve possuir número de matrícula e senha para acesso ao sistema.	1	O usuário executa a aplicação em seu <i>smartphone</i> .	O MAAS exibe sua tela de <i>login</i> , se esta for a primeira vez que é utilizado.
	2	O usuário deve preencher os campos “ <i>Matrícula</i> ” e “ <i>Senha</i> ” e então clicar no botão “ <i>Entrar</i> ”.	O MAAS autentica o usuário, verificando na base de dados do servidor se o número de matrícula e senha existem, e então exibe a tela Agenda;

PROCEDIMENTOS DE TESTES			
T002 – Navegar Entre os Meses			
Pré-condições	Passo	Ação	Resultado Esperado
1. O usuário deve possuir <i>login</i> e senha para acesso ao sistema; 2. O usuário deve estar logado no sistema 3. O usuário deve ter acesso à internet; 4. A tela mostrada deve ser a tela principal.	1	O usuário deve clicar no botão contendo a flecha para a esquerda ou direita;	O MAAS redesenha a tela principal para apresentar o mês anterior ou posterior ao mostrado inicialmente;

PROCEDIMENTOS DE TESTES			
T003 – Expandir Compromissos/Eventos do Dia			
Pré-condições	Passo	Ação	Resultado Esperado
1. O usuário deve possuir <i>login</i> e senha para acesso ao sistema, o mesmo deve estar logado;	1	O usuário deve clicar em um dia do calendário;	O MAAS exibe uma lista com todos os eventos e compromissos marcados para aquele dia.

PROCEDIMENTOS DE TESTES			
T004 – Expandir Detalhes do Compromisso			
Pré-condições	Passo	Ação	Resultado Esperado
1. O usuário deve possuir <i>login</i> e senha para acesso ao sistema, o mesmo deve estar logado 2. Deve haver algum compromisso ou evento associado a este usuário	1	O usuário deve clicar em um compromisso listado na tela;	O MAAS exibe uma tela contendo todos os detalhes do compromisso selecionado.

PROCEDIMENTOS DE TESTES			
T005 – Expandir Detalhes do Evento			
Pré-condições	Passo	Ação	Resultado Esperado
1. O usuário deve possuir <i>login</i> e senha para acesso ao sistema, o mesmo deve estar logado 2. Deve haver algum compromisso ou evento associado a este usuário	1	O usuário deve clicar em um evento listado na tela;	O MAAS exibe uma tela contendo todos os detalhes do evento selecionado.