

UNIVERSIDADE FEDERAL DE SANTA CATARINA - UFSC

GERENCIAMENTO DE PROJETOS EM DISPOSITIVOS MÓVEIS
UMA EVOLUÇÃO AO DOTPROJECT

Glauco Silva Neves

Florianópolis - SC

2011/1

UNIVERSIDADE FEDERAL DE SANTA CATARINA – UFSC

DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

CURSO DE CIÊNCIAS DA COMPUTAÇÃO

GERENCIAMENTO DE PROJETOS EM DISPOSITIVOS MÓVEIS

UMA EVOLUÇÃO AO DOTPROJECT

Glauco Silva Neves

Trabalho de conclusão de curso
apresentado como parte dos requisitos
para obtenção do grau de Bacharel em
Ciências da Computação.

Florianópolis - SC

2011/1

Glauco Silva Neves

GERENCIAMENTO DE PROJETOS EM DISPOSITIVOS MÓVEIS

UMA EVOLUÇÃO AO DOTPROJECT

Trabalho de conclusão de curso apresentado como parte dos requisitos para obtenção do grau de Bacharel em Ciências da Computação.

Orientadora: Prof. Dr. rer. nat. Christiane Gresse von Wangenheim, PMP

Banca examinadora

Prof. Ms. José Eduardo De Lucca

Ms. Mathias Henrique Weber

SUMÁRIO

LISTA DE FIGURAS	7
LISTA DE QUADROS	9
LISTA DE REDUÇÕES	10
RESUMO.....	11
1 INTRODUÇÃO	12
1.1 CONTEXTO	12
1.2 PROBLEMA	13
1.3 OBJETIVOS	14
1.3.1 Objetivo Geral.....	14
1.3.2 Objetivos Específicos	15
1.3.3 Limitações	15
1.4 METODOLOGIA	15
1.5 JUSTIFICATIVA	17
1.6 ESTRUTURA DO TRABALHO	18
2 FUNDAMENTAÇÃO TEÓRICA	20
2.1 GERENCIAMENTO DE PROJETOS	20
2.2 SOFTWARE PARA GERENCIAMENTO DE PROJETOS	26
2.2.1 dotProject	34
2.3 APLICAÇÕES MÓVEIS	39
2.3.1 Android	44
3 REVISÃO DO ESTADO DA ARTE	46

3.1	BASECAMP	48
3.2	CONNECTWISE	49
3.3	GTDAGENDA	50
3.4	RALLY	51
3.5	DISCUSSÃO	52
4	DESENVOLVIMENTO DE UM APLICATIVO MÓVEL INTEGRADO AO DOTPROJECT	56
4.1	ANÁLISE DE REQUISITOS	56
4.1.1	Casos de Uso	58
4.2	DESIGN	61
4.2.1	Arquitetura	61
4.2.2	Design de Interface	62
4.3	IMPLEMENTAÇÃO	64
4.4	TESTES	65
5	AVALIAÇÃO	67
5.1	DEFINIÇÃO DA AVALIAÇÃO	67
5.1.1	Objetivo da Avaliação	67
5.1.2	Perguntas da Avaliação	68
5.1.3	Design do Estudo	68
5.1.4	Materiais da Avaliação	69
5.2	EXECUÇÃO DA AVALIAÇÃO	69
5.3	RESULTADOS DA AVALIAÇÃO	72

5.4	AMEAÇAS À VALIDADE	74
6	DISCUSSÃO	75
7	CONCLUSÃO	76
	REFERÊNCIAS	78
	APÊNDICE A – TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO	81
	APÊNDICE B – DESCRIÇÃO DA TAREFA	82
	APÊNDICE C – SCRIPT DE INTRODUÇÃO	84
	APÊNDICE D – ESCALA DE USABILIDADE DO SISTEMA(SUS)	85
	APÊNDICE E – QUESTIONÁRIO DEMOGRÁFICO	86
	APÊNDICE F – ROTEIRO DO PESQUISADOR	88
	APÊNDICE G – TREINAMENTO	90
	APÊNDICE H – RESULTADO DO SUS	93
	APÊNDICE I – RESULTADO DO QUESTIONÁRIO DEMOGRÁFICO	96
	APÊNDICE J – TEMPOS DA AVALIAÇÃO	100
	APÊNDICE K – CÓDIGO FONTE	101

LISTA DE FIGURAS

Figura 1: Conexão entre os grupos de processos.....	22
Figura 2: Interação entre os grupos de processos.....	23
Figura 3: Funcionalidades fundamentais em um SGP segundo as organizações.	28
Figura 4: Softwares de apoio ao gerenciamento de projetos mais utilizados. .	30
Figura 5: Importância das características de um SGP para gerenciar um projeto.	32
Figura 6: Impressão sobre as características dos SGPs utilizados.....	33
Figura 7: Gráfico de Gantt gerado pelo dotProject.....	36
Figura 8: Menu de projetos.	36
Figura 9: Lista de tarefas.....	37
Figura 10: Página de projetos (esquerda) e diagrama de Gantt (direita) na tela do emulador de 3.3 polegadas.....	39
Figura 11: Exemplos de dispositivos móveis.....	40
Figura 12: Capacidades e funcionalidades de um dispositivo móvel.....	41
Figura 13: Capacidades e funcionalidades de um dispositivo móvel.....	42
Figura 14: Porcentagem de venda de <i>smartphones</i> em 2010 (esquerda) e a previsão para 2011 (direita).....	44
Figura 15: Gráfico baseado no número de dispositivos Android que acessaram o <i>Android Market</i> no período de 18 de abril a 2 de maio de 2011.	45
Figura 16: Interface do Timy (esquerda) e Beacon (direita).....	49
Figura 17: Interface do ConnectWise Mobile mostrando agenda (esquerda) e menu principal (direita).....	50
Figura 18: Interface do Gtdagenda, mostrando a caixa de mensagens.	51

Figura 19: Interface do RallyDroid.....	52
Figura 20: Arquitetura do sistema.....	62
Figura 21: Participantes durante a avaliação.	70
Figura 22: Resultado da pergunta 4 do questionário demográfico.....	71
Figura 23: Resultado da pergunta 6 do questionário demográfico.....	71
Figura 24: Resultado da pergunta 9 do questionário demográfico.....	71
Figura 25: Tempo de execução por tarefa e média.....	72
Figura 26: Pontuação do SUS por participante.....	73

LISTA DE QUADROS

Quadro 1: Processos mapeados em grupos e áreas do conhecimento.....	25
Quadro 2: Diferenças entre dispositivos móveis e computadores.....	42
Quadro 3: Dados da pesquisa.....	47
Quadro 4: Comparação das ferramentas.....	54
Quadro 5: Caso de uso detalhado – Autenticar usuário.	58
Quadro 6: Caso de uso detalhado – Visualizar projetos	59
Quadro 7: Caso de uso detalhado – Visualizar tarefas atrasadas.....	60
Quadro 8: Caso de uso detalhado - Entrar em contato com responsável por tarefa atrasada.....	61
Quadro 9: Telas do aplicativo e descrição.	63
Quadro 10: Resultado dos testes de fluxo de casos de uso.	65
Quadro 11: Resultado dos testes de requisitos não funcionais.	66
Quadro 12: Comparação das ferramentas identificadas no estado da arte e da ferramenta desenvolvida.....	75

LISTA DE REDUÇÕES

SIGLAS

PMI	<i>Project Management Institute</i>
GP	Gerenciamento de projetos
SGP	Software para gerenciamento de projetos
PMBOK	<i>Project Management Body of Knowledge</i>
PERT	Técnica de Programa, Avaliação e Revisão
GPL	<i>General Public License</i>
SDK	<i>Software Development Kit</i>
ADT	<i>Android Development Toolkit</i>
IDE	<i>Integrated Development Environment</i>

ABREVIações

Result.	Resultados
Analís.	Analísados
Seleci.	Selecionados
Flux.	Fluxo
Sit.	Situação
Prin.	Principal
Alt.	Alternativo

RESUMO

O mercado está se tornando mais globalizado e competitivo. Conseguir produzir mais em menos tempo e gastando menos recursos é essencial. Esse é um dos motivos que torna o gerenciamento de projetos necessário. A aplicação de ferramentas computacionais faz parte do gerenciamento de projetos. Essas ferramentas auxiliam principalmente os gerentes a planejar, monitorar e controlar seus projetos. Com a popularização e potencialização dos *smartphones* a partir de 2005 diversas empresas do setor de tecnologia da informação têm portado suas aplicações para plataformas móveis. Empresas desenvolvedoras de softwares para gerência de projetos também parecem caminhar nesse sentido. Porém, voltado à comunidade de software livre ainda não há um movimento tão forte nesta direção.

Uma das ferramentas de código aberto mais populares é o dotProject, que por ser uma ferramenta *open source* conta com a comunidade para estendê-la. O objetivo desse trabalho é estender o dotProject para dispositivos móveis, desenvolvendo um aplicativo móvel para suportar o monitoramento e controle de projetos.

O projeto é executado em quatro etapas bem definidas. Primeiramente é realizada a fundamentação teórica dos conceitos de gerência de projetos, softwares para gerenciar projetos e dispositivos móveis. A segunda etapa é a revisão do estado da arte com a finalidade de comparar e identificar as principais funcionalidades dos aplicativos disponíveis. A terceira etapa é a construção do aplicativo. E a quarta e última etapa é a avaliação do aplicativo desenvolvido.

Espera-se conceber um aplicativo para monitorar e controlar projetos que estenderá o dotProject. O benefício é auxiliar o gerente de projeto a rastrear, revisar e regular o progresso e o desempenho do projeto a qualquer hora e em qualquer lugar.

1 INTRODUÇÃO

Neste capítulo são apresentados o contexto do trabalho, o problema a ser resolvido, os objetivos a serem alcançados, a metodologia utilizada, a justificativa e a estrutura do trabalho.

1.1 CONTEXTO

O mercado está se tornando mais globalizado e competitivo, com maiores demandas de serviços e produtos (HORINE, 2009). Ao mesmo tempo aumenta a expectativa dos clientes (PORTNY, 2007). Nesse contexto as empresas precisam se adaptar para serem mais eficientes e produtivas. Requisitos solicitados tornam-se obsoletos de um dia para o outro e as empresas devem responder rapidamente (HORINE, 2009). Torna-se essencial conseguir produzir mais em menos tempo e gastando menos recursos (ROTHMAN, 2007). Dessa forma é necessário gerenciar projetos de forma eficiente.

O *Project Management Institute* (PMI) (2008, p.6) define Gerenciamento de Projetos (GP) como sendo “a aplicação do conhecimento, habilidades, ferramentas e técnicas às atividades do projeto a fim de atender aos seus requisitos”. Conforme esta definição, uma parte importante do GP são as ferramentas. Os Softwares para Gerenciamento de Projetos (SGP) são ferramentas computacionais que auxiliam principalmente os gerentes a planejar e controlar seus projetos. Existem de diversos tipos, gratuitos ou pagos, de licença proprietária ou livre, com escopo e funcionalidades variadas, tanto como aplicações *desktop* ou sistemas *web*. Por exemplo, o SGP *Microsoft Project* (<http://www.microsoft.com/project>), que é proprietário, possui versões para *desktop* e sistema *web*, e é um dos mais utilizados (LOCKWOOD, 2008). Uma das ferramentas livres e de código aberto mais

populares é o dotProject (<http://www.dotproject.net>). O dotProject possui funcionalidades de gerenciamento de tarefas, tempo, comunicação e compartilhamento. Além de ser gratuito e ter seu código aberto, ele possui a vantagem de ser acessível através de internet, não necessitando a instalação em cada máquina utilizada.

As empresas desenvolvedoras de software vivem hoje num momento de convergência digital. Com o mesmo serviço ou produto disponível em diversos meios digitais como celular, televisão, internet, entre outros (IBCD, 2011).

É observado o crescimento de uso de dispositivos móveis no mundo inteiro, incluindo principalmente o número crescente de celulares também no Brasil onde já existem mais celulares do que habitantes brasileiros (ANATEL, 2010). Celulares estão cada vez mais poderosos com melhores processadores e telas maiores. Esses celulares são conhecidos como *smartphones* e já possuem mais de 12 milhões de usuários no Brasil (ANATEL, 2010).

Dentro deste novo cenário, algumas funcionalidades dos SGPs também podem ser estendidas para aplicações móveis, principalmente para o monitoramento e controle de projetos.

1.2 PROBLEMA

A partir de 2005 os *smartphones* começaram a ganhar mercado, trazendo uma experiência mais atrativa ao usuário (CONSTANTINO, CAMILLERI e KAPETANAKIS, 2010). Surgiu então o fenômeno das lojas de aplicativos, e novas plataformas apareceram: BlackBerry, iOS e Android. A *App Store* da Apple já possui mais de 350 mil aplicativos, e é a maior seguida do *Android Market* com mais de 200 mil aplicativos.

Com a popularização da internet móvel e com a melhoria das funcionalidades e usabilidade dos celulares, diversas empresas do setor de tecnologia da informação têm portado suas aplicações para plataformas móveis (MEHTA, 2008).

Empresas desenvolvedoras de softwares para gerência de projetos também parecem caminhar nesse sentido. Por exemplo, a ConnectWise (<http://www.connectwise.com>), empresa premiada em soluções gerenciais, já possui versões móveis para Android, iPhone e *web* móvel. Entre as principais funcionalidades estão: alocação de recursos, agendamento de compromissos, controle de gastos e acesso a informações de contatos.

Porém, voltado à comunidade de software livre ainda não há um movimento tão forte nesta direção. Como é o caso da ferramenta dotProject, que é uma das mais populares¹, e não possui um aplicativo móvel que estenda a ferramenta.

Dado esse contexto, então a pergunta de pesquisa deste trabalho é se é possível estender uma(s) funcionalidade(s) do dotProject para aplicações móveis?

1.3 OBJETIVOS

Com base no contexto e no problema são definidos o objetivo geral e os objetivos específicos.

1.3.1 Objetivo Geral

O objetivo geral desse trabalho é desenvolver um aplicativo móvel para suportar atividades do monitoramento e controle de projetos a ser integrado ao software dotProject.

¹ Possui mais de 850 mil downloads desde a versão 2.0 (dado levantado no dia 27/11/2010)

² Possui mais de 2 mil downloads semanais através do SourceForge (dado levantado no dia

1.3.2 Objetivos Específicos

Os objetivos específicos desse trabalho são:

1. Analisar o contexto do trabalho, estudando as áreas de gerenciamento de projetos, softwares para gerência de projetos e dispositivos móveis.
2. Realizar o levantamento do estado da arte, analisando aplicações móveis já existentes neste contexto.
3. Desenvolver um protótipo funcional de um aplicativo móvel a ser integrado ao dotProject para a plataforma Android.
4. Avaliar o protótipo desenvolvido.

1.3.3 Limitações

Não faz parte do escopo deste trabalho estender todas as funcionalidades do dotProject para o aplicativo móvel. São estendidas algumas atividades do grupo de monitoramento e controle de projetos. Somente as funcionalidades selecionadas através do estudo são implementadas. Também não faz parte deste trabalho desenvolver o aplicativo para ser integrado em outros softwares para gerenciamento de projetos, além do dotProject.

O trabalho visa somente à plataforma Android, por ser uma plataforma livre, que possui a menor curva de aprendizado entre as plataformas (CONSTANTINOU, CAMILLERI e KAPETANAKIS, 2010) e é uma das mais utilizadas atualmente com 35% do mercado de *smartphones* no primeiro quarto de 2011 (GARTNER, 2011).

1.4 METODOLOGIA

O presente trabalho é realizado em quatro etapas:

- **Etapas 1. Fundamentação teórica:** É realizado um estudo da área de gerência de projetos, principalmente levantando os conceitos definidos pelo

PMI. Posteriormente é feita uma análise dos softwares para gerenciar projetos, onde são discutidas as principais características e é aprofundado o estudo da ferramenta dotProject. Por fim são descritos os dispositivos móveis, suas particularidades e as principais plataformas. As atividades desta etapa são:

A1.1 Fundamentar a teoria de GP

A1.2 Fundamentar os SGPs

A1.3 Fundamentar os dispositivos móveis

- **Etapa 2. Revisão do estado de arte:** É feita uma análise sistemática, de acordo com Kitchenham (2004), com a finalidade de comparar e identificar as principais funcionalidades de softwares para gerência de projetos em dispositivos móveis utilizando a plataforma Android. As atividades desta etapa são:

A2.1 Definição do estudo

A2.2 Execução do estudo

A2.3 Extração e análise de informação

- **Etapa 3. Concepção e desenvolvimento do protótipo:** Com base na análise feita na revisão do estado da arte é concebido um novo aplicativo para a plataforma Android. É realizada uma iteração de desenvolvimento, incluindo a análise de requisitos, projeto, implementação e testes. Em paralelo à análise de requisitos e projeto é realizada a engenharia de usabilidade.

A3.1 Analisar o contexto

A3.2 Iteração 1

A3.2.1 Análise de requisitos

A3.2.2 Projeto

A3.2.3 Implementação

A3.2.4 Testes

- **Etapa 4. Avaliação:** O aplicativo móvel desenvolvido é avaliado em relação à utilidade, eficiência, eficácia e satisfação. É realizado um teste de usabilidade envolvendo representantes do grupo de usuários (pessoas com conhecimento em gerência de projetos) coletando dados via observação e por meio de questionário.

A4.2 Definição da avaliação

A4.3 Planejamento dos testes de usabilidade

A4.5 Execução dos testes de usabilidade

A4.7 Análise dos dados

1.5 JUSTIFICATIVA

O dotProject é uma das ferramentas para gerenciamento de projetos de código aberto e gratuita mais utilizadas no mundo². O projeto conta com a participação da comunidade para estender a ferramenta além das funcionalidades do núcleo do dotProject. O intuito desse trabalho é estender o dotProject para dispositivos móveis e verificar a contribuição de possuir essa ferramenta em celulares.

A contribuição científica vem através do conhecimento documentado na revisão do estado da arte e do modelo conceitual das funcionalidades para suportar atividades do monitoramento e controle no contexto de dispositivos móveis. O resultado da revisão sistemática da literatura apresenta uma visão geral de aplicações móveis utilizadas em conjunto com sistemas *web* de gerenciamento de projetos e

² Possui mais de 2 mil downloads semanais através do SourceForge (dado levantado no dia 27/11/2010)

posteriormente a análise das funcionalidades suportadas. É também uma contribuição o modelo conceitual construído para dispositivos móveis que suporte o monitoramento e controle alinhados ao PMBOK. Indicando possíveis formas como ferramentas desse tipo em dispositivos móveis podem suportar a gerência de projetos.

A contribuição de impacto tecnológico vem através do desenvolvimento de um aplicativo para a plataforma Android integrado ao dotProject. O código do aplicativo é aberto e está disponível no Apêndice K.

A contribuição social vem através de um aplicativo de código aberto que possa ser usado, copiado, estudado e redistribuído sem restrições pela comunidade do software livre.

As empresas podem se beneficiar utilizando o modelo conceitual descrito nesse projeto para estender a ferramenta de gerenciamento que utiliza. Ou utilizar o aplicativo desenvolvido para realizar o monitoramento e controle do projeto mesmo quando não tiver fácil acesso a um computador. E dessa forma procura-se obter uma melhora na taxa de sucesso dos projetos.

1.6 ESTRUTURA DO TRABALHO

O presente trabalho está dividido em quatro capítulos descritos a seguir.

No capítulo dois é feita a fundamentação teórica do trabalho onde são apresentados os principais conceitos da área de gerência de projetos de acordo com o PMI. São abordados os processos que fazem parte do ciclo de um projeto e áreas de conhecimentos envolvidas. Também são descritos os SGPs, abordando principalmente suas funcionalidades. E por fim a descrição de aplicações móveis.

No capítulo três é apresentada a revisão do estado da arte de aplicativos móveis de SGPs analisando principalmente os *features* oferecidos.

No capítulo quatro é descrita a solução com base em tudo que foi discutido anteriormente. É feita uma análise de requisitos, modelagem, desenvolvimento e teste de um novo aplicativo.

No capítulo cinco o sistema passa por um teste de usabilidade. São descritos a definição da avaliação, a execução e os resultados obtidos.

O capítulo seis faz uma discussão de tudo que foi apresentado.

O capítulo sete conclui o trabalho, mostrando tudo o que foi feito, quais foram as contribuições e sugestões de trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Nesse capítulo são apresentados conceitos de gerenciamento de projetos, softwares para gerenciamento de projetos e aplicativos móveis que são relevantes no contexto do presente trabalho.

2.1 GERENCIAMENTO DE PROJETOS

Para entender o que é gerenciamento de projetos precisa-se primeiramente definir o que é um projeto. Segundo o PMI (2008, p.5) *“um projeto é um esforço temporário empreendido para criar um produto, serviço ou resultado exclusivo”*. O termo temporário quer dizer que o projeto possui um ciclo de vida com início e final determinados. O projeto termina quando seus objetivos forem alcançados ou quando existirem motivos para não continuá-lo, por exemplo, por falta de tempo. A exclusividade do projeto está no fato do produto, serviço ou resultado possuírem características únicas como custo, tempo, tarefas entre outras (PMI, 2008). Por fim, o projeto possui uma elaboração progressiva que de acordo com o PMI (2004, p.6) *“significa desenvolver em etapas e continuar por incrementos”*, ou seja, as características do resultado do projeto vão sendo incrementadas continuamente enquanto as etapas do projeto vão sendo executadas.

Conforme o PMI (2008, p.6) o gerenciamento de projetos *“é a aplicação do conhecimento, habilidades, ferramentas e técnicas às atividades do projeto a fim de atender aos seus requisitos”*. A aplicação do conhecimento diz respeito aos processos que devem ser aplicados e integrados para efetivação do gerenciamento de projetos. O PMI (2008, p.37) define processo como sendo *“um conjunto de ações e atividades inter-relacionadas realizadas para alcançar um produto, resultado ou serviço pré-especificado”*. A aplicação das habilidades e ferramentas é feita nas

entradas de um processo e resultam em uma ou mais saídas que alimentam outros processos. Os processos estão divididos em cinco grupos de processos de gerenciamento de projetos (PMI, 2008):

- **Iniciação:** Contém processos para definição de um novo projeto ou nova fase do projeto. É elaborado um documento chamado de termo de abertura de projeto. No termo de abertura são definidos um escopo inicial e um orçamento inicial. Também são identificadas as partes interessadas no projeto, gerente, equipe, cliente, patrocinadores e etc. O projeto se torna oficialmente autorizado quando o termo de abertura é assinado.
- **Planejamento:** Esse grupo consiste dos processos realizados para determinar o escopo total do projeto, definir e refinar os objetivos e planejar as atividades para alcançar esses objetivos. Como resultado desse grupo de processos tem-se o plano de gerenciamento de projeto que deve contar com a participação das partes interessadas.
- **Execução:** Heerkens (2002, p.6) disse que o *“gerenciamento de projetos está relacionado ao fato de projetos serem realmente sobre pessoas fazendo as coisas”*. Nesse grupo os processos estão relacionados às atividades que são executadas a fim de cumprir os objetivos. O gerente é responsável por coordenar pessoas e recursos de acordo com o plano de gerenciamento de projetos.
- **Monitoramento e controle:** Nesse grupo reúnem-se os processos responsáveis por rastrear, revisar e regular o progresso e o desempenho do projeto. Os processos desse grupo visam monitorar o bom andamento do projeto, verificando se o projeto está dentro do escopo, prazo, orçamento e caso não esteja, iniciam-se e gerenciam-se ações corretivas.

- **Encerramento:** Esse grupo possui dois processos responsáveis pela formalização da completude do projeto ou fase. Será então verificado se os objetivos foram alcançados e se o cliente aceita a entrega do projeto. É também aqui que são discutidas as lições aprendidas.

Os resultados que um processo produz geralmente são consumidos na entrada de outro processo, formando assim, conexões entre os processos, que se pode elevar ao nível dos grupos de processos conforme a Figura 1 (PMI, 2008).

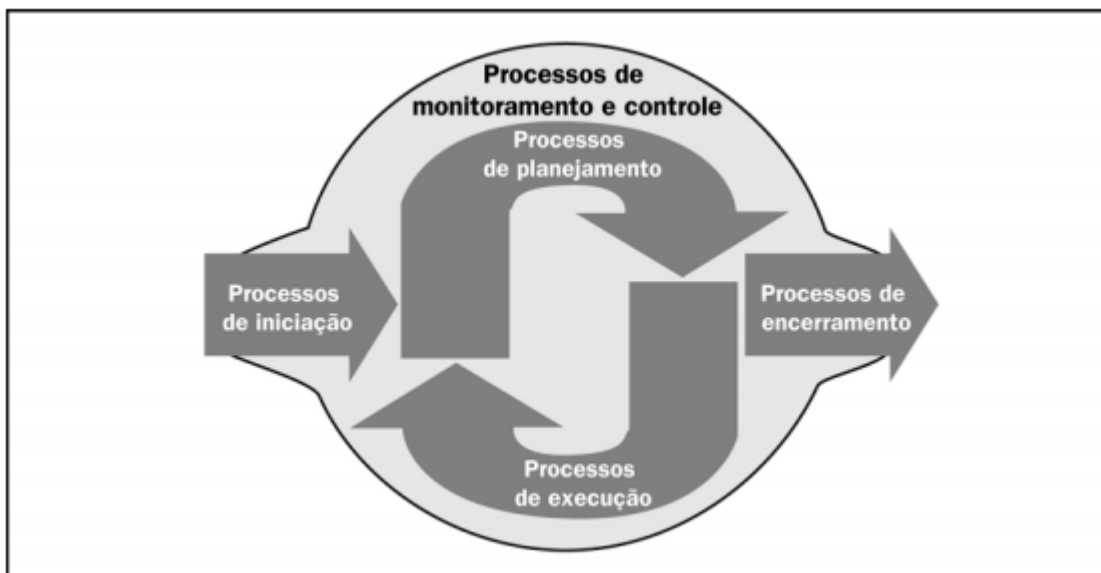


Figura 1: Conexão entre os grupos de processos

Fonte: Guia PMBOK®. Um Guia do Conjunto de Conhecimentos em Gerenciamento de Projetos. 3ª edição. Pensilvânia: *Project Management Institute*, 2004.

Os grupos de processos, porém, de acordo com o PMI (2008, p.40) “*raramente são eventos distintos ou únicos; eles são atividades sobrepostas que ocorrem em diversos níveis de intensidade durante todo o projeto*” como é mostrado na Figura 2.

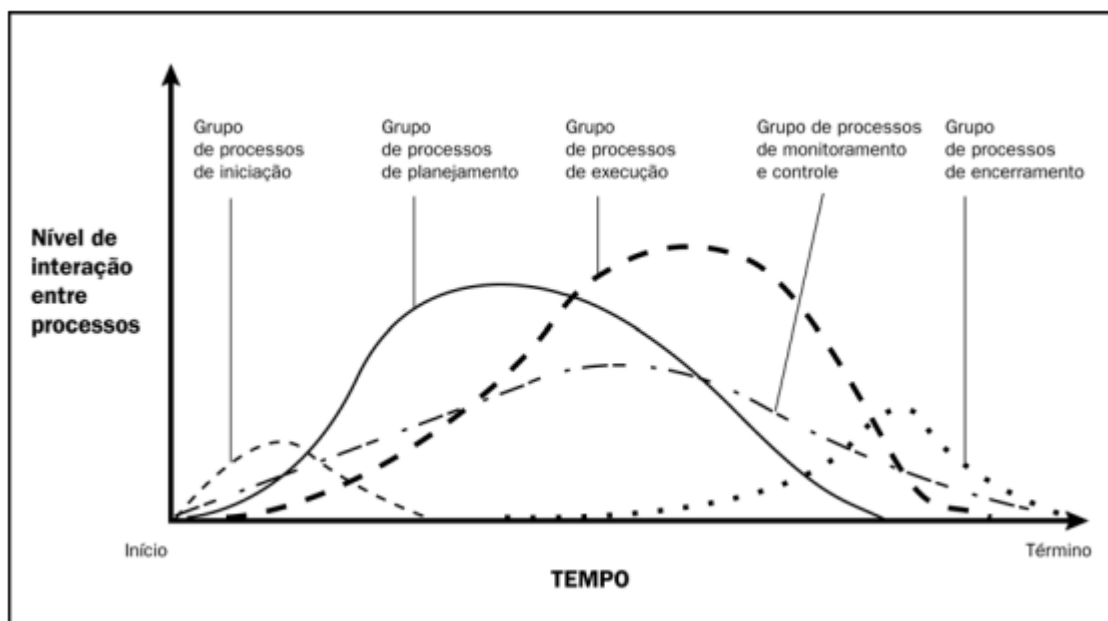


Figura 2: Interação entre os grupos de processos

Fonte: Guia PMBOK®. Um Guia do Conjunto de Conhecimentos em Gerenciamento de Projetos. 3ª edição. Pensilvânia: *Project Management Institute*, 2004.

Além da divisão por grupos, os processos também são organizados em nove áreas de conhecimento (PMI, 2008), sendo elas:

- **Integração:** Os processos dessa área são os responsáveis por identificar, combinar, unificar e coordenar os processos de todas as outras áreas do conhecimento.
- **Escopo:** Inclui os processos que definem o que será e o que não será feito no projeto com a finalidade de garantir que o produto, serviço ou resultado só possuirá as características e funções especificadas no projeto.
- **Tempo:** Processos relacionados às atividades e a duração delas fazem parte dessa área de conhecimento. O objetivo é gerenciar o tempo para que o projeto encerre no prazo certo.

- **Custos:** Envolve todos os processos relacionados ao orçamento do projeto, desde as estimativas até o controle de gastos. O projeto deve ser entregue dentro do orçamento combinado.
- **Qualidade:** Contém os processos responsáveis por determinar, garantir e controlar as políticas e procedimentos de qualidade executados durante todo o projeto.
- **Recursos humanos:** Abrange os processos relacionados à equipe do projeto. Segundo o PMI (2004, p.199) a equipe do projeto “*é composta de pessoas com funções e responsabilidades atribuídas para o término do projeto*” e essa área de conhecimento é responsável por formá-la, desenvolvê-la e gerenciá-la.
- **Comunicações:** De acordo com o PMI (2004, p.10) essa área “*descreve os processos relativos à geração, coleta, disseminação, armazenamento e destinação final das informações do projeto de forma oportuna e adequada*” e é fundamental para o gerente de projetos, visto que boa parte do seu tempo é utilizada em contato com as partes interessadas.
- **Riscos:** Inclui os processos responsáveis por identificar os riscos e por fazer um planejamento desses riscos buscando facilitar a ocorrência de eventos positivos e dificultar a ocorrência dos negativos, mas tendo um plano de resposta para caso ocorram.
- **Aquisições:** Agrupa os processos que tem relação com produtos, serviços e resultados que podem ser adquiridos para o projeto e que não fazem parte da equipe do projeto.

O quadro 1 mostra os 42 processos definidos pelo PMBOK mapeados nos cinco grupos de processos de gerenciamento de projetos e nas nove áreas de conhecimento.

Quadro 1: Processos mapeados em grupos e áreas do conhecimento.

		Grupos de processos				
		Iniciação	Planejamento	Execução	Monitoramento e controle	Encerramento
Áreas do conhecimento	Integração	Desenvolver o termo de abertura do projeto	Desenvolver o plano de gerenciamento do projeto	Orientar e gerenciar a execução do projeto	Monitorar e controlar o trabalho do projeto Realizar o controle integrado de mudanças	Encerrar o projeto ou fase
	Escopo		Coletar os requisitos Definir o escopo Criar a EAP		Verificar o escopo Controlar o escopo	
			Definir as atividades Sequenciar as atividades Estimar os recursos das atividades Estimar as durações das atividades Desenvolver o cronograma		Controlar o cronograma	
			Estimar os custos Determinar o orçamento		Controlar os custos	
	Qualidade		Planejar a qualidade	Realizar a garantia de qualidade	Realizar o controle da qualidade	
	Recursos humanos		Desenvolver o plano de recursos humanos	Mobilizar a equipe do projeto Desenvolver a equipe do projeto Gerenciar a equipe do projeto		
		Identificar as partes interessadas	Planejar as comunicações	Distribuir as informações Gerenciar as expectativas das partes interessadas	Reportar o desempenho	
	Riscos		Planejar o gerenciamento de riscos Identificar os riscos Realizar a análise qualitativa de riscos Realizar a análise quantitativa de riscos Planejar as respostas aos riscos		Monitorar e controlar os riscos	
			Planejar as aquisições Planejar contratações	Conduzir as aquisições	Administrar as aquisições	Encerrar as aquisições
	Aquisições					

Fonte: Adaptado de PMBOK® *Guide. A Guide to the Project Management Body of Knowledge*. 4ª edição. Pensilvânia: *Project Management Institute*, 2008. P.43.

Esses processos são então um subconjunto do conjunto de conhecimentos em gerenciamento de projetos e são reconhecidos pelo PMI como boas práticas. A aplicação correta desses conceitos pode aumentar a taxa de sucesso dos projetos,

porém os projetos são únicos e é necessário bom senso para saber o que não deve ser aplicado (PMI, 2008).

Como definido no início desse capítulo, faz parte do gerenciamento de projetos a utilização de ferramentas. O capítulo a seguir define um dos tipos de ferramenta que são os softwares para gerência de projetos.

2.2 SOFTWARE PARA GERENCIAMENTO DE PROJETOS

Softwares para Gerenciamento de Projetos (SGPs) são ferramentas que auxiliam principalmente o gerente do projeto. SGPs podem ser usados para diversos fins como planejar, rastrear, monitorar, relatar e comunicar. Existem variados SGPs com diferentes custos, funcionalidades e usabilidades. Kerzner (2001, p.706) define uma lista das principais funções oferecidas pelos SGPs, descritas a seguir:

- **Planejamento, rastreamento e monitoramento.** De acordo com Kerzner (2001, p.706) *“são os recursos mais comuns fornecidos para o planejamento e rastreamento das tarefas, recursos e custos do projeto”*. Tipicamente as informações podem ser vistas em tipologias de rede como o método do caminho crítico, o método do diagrama de precedência e Técnica de Programa, Avaliação e Revisão (PERT). Dessa forma é fácil visualizar as tarefas e suas características como seu tempo de início e fim, seus recursos alocados e seus custos. Também é possível acompanhar a situação do projeto e compará-lo com o que foi planejado inicialmente.
- **Relatórios.** Usuários podem gerar, modificar ou criar relatórios, padronizados ou não, através do SGP. Esses relatórios incluem uma série de diagramas de Gantt, diagramas de rede, sumários tabulares e gráficos de negócio. Kerzner (2001, p.707) cita alguns exemplos de relatórios: *“relatório do custo orçado do*

trabalho planejado; relatório do custo do trabalho realizado; relatório dos gastos reais versus planejados; análise do valor agregado; [e outros]”.

- **Calendário do projeto.** Permite a visualização do projeto em semanas considerando os dias úteis. Outros períodos sem trabalho como feriados e férias também podem ser considerados.
- **Análise de cenários (*what-if analysis*).** É possível simular modificações no projeto e o SGP compara as alternativas mostrando riscos positivos e negativos. Por exemplo, pode-se avaliar a iniciação de uma tarefa mais tarde e ver se ocorreram impactos no orçamento ou tempo do projeto.

Um estudo de *benchmarking* em gerenciamento de projetos realizado no Brasil mostrou as funcionalidades fundamentais de um SGP do ponto de vista de 300 organizações brasileiras (Figura 3). O cronograma foi considerado o mais importante, com 81%, é uma das funções principais de um SGP, descritas acima, que permite o planejamento e o controle de execução do trabalho em função do tempo. Além das funcionalidades principais de um SGP podem-se ver outras características que são consideradas importantes como, por exemplo, o acesso via web com 56%.



Figura 3: Funcionalidades fundamentais em um SGP segundo as organizações.

Fonte: Estudo de Benchmarking em Gerenciamento de Projetos Brasil 2009, Project Management Institute – Chapters Brasileiros.

Observando essas outras características Blokdiijk (2007) define cinco tipos de SGP:

- **Desktop:** precisam ser instalados no computador como qualquer outro programa. Normalmente são usados em projetos mais complexos.
- **Baseados na web (web-based):** podem ser utilizados a partir de um navegador. A grande vantagem é não precisar instalar o aplicativo, podendo acessá-lo de qualquer computador com acesso a internet.
- **Pessoal:** utilizado para gerenciar tarefas pessoais ou projetos caseiros.
- **Único usuário:** onde somente uma pessoa pode acessar, modificar e atualizar o SGP.
- **Colaborativo:** onde diversos usuários utilizam o software.

Bereaux (2008) refina a divisão das categorias *desktop* e baseados na *web* em:

- **Open-source:** softwares que podem ser adquiridos sob uma licença gratuita ou que estão sob um domínio público.
- **Proprietário:** softwares que precisam ser comprados. Suas licenças podem ser individuais ou da empresa.

A Figura 4 mostra os SGPs mais utilizados pelas empresas que participaram do *benchmarking*. Seguindo o refinamento de Bereux podem-se separar os softwares pesquisados da seguinte forma:

- SGPs baseados na web:
 - *Open-source:* dotProject
 - Proprietário: MS Project Server, Oracle Primavera Systems, Project Builder, SAP PS, IBM RPM, CA Clarity, Compuware Changeoint
- SGPs desktop:
 - *Open-source:* OpenProj
 - Proprietário: MS Project, WBS Chart Pro, Planview

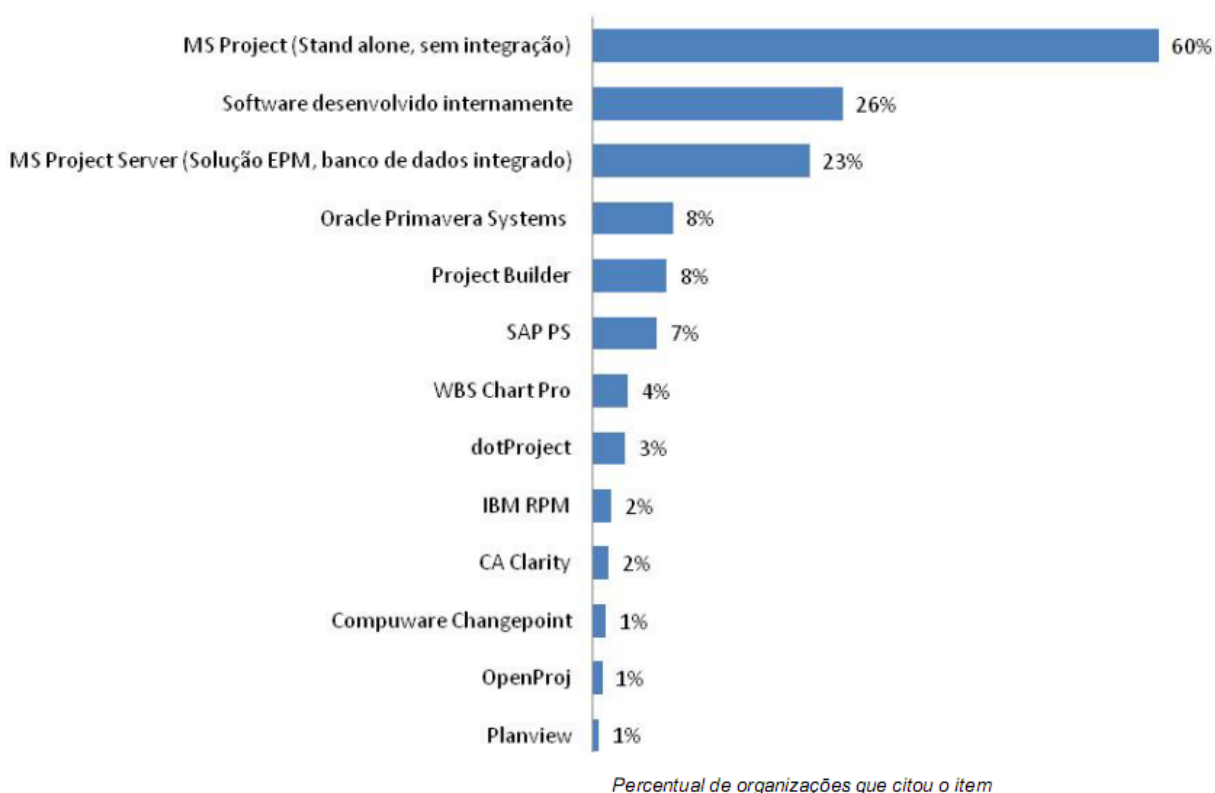


Figura 4: Softwares de apoio ao gerenciamento de projetos mais utilizados.

Fonte: Estudo de Benchmarking em Gerenciamento de Projetos Brasil 2009, Project Management Institute – Chapters Brasileiros.

Além das características de auxílio à gerência de projetos, outras qualidades do SGP devem ser consideradas. Levine (2002), Davidson (2000) e Heerkens (2002) propõem alguns critérios que influenciam na seleção de um SGP:

- **Compatibilidade com outros sistemas:** Algumas funções podem ser executadas em outros sistemas e deve existir conexão com o SGP. Como por exemplos ferramentas de controle de versão e de *bug tracking*.
- **Funcionalidades:** Devem-se considerar necessidades atuais e prever necessidades futuras, porém sem confundir necessidades futuras adquirindo o software com milhares de funções que nunca serão utilizadas.
- **Custo:** Deve se preocupar de acordo com o poder da ferramenta que se necessita, pois um SGP pode chegar a custar dezenas de milhares de

dólares (HEERKENS, 2002). Outro custo que deve ser levado em conta é o do ciclo do SGP, pois existem os gastos com treinamento, suporte, hardware e outros.

- **Facilidade de uso:** Normalmente um SGP com mais funcionalidades será mais complexo, exigindo mais treinamento. Devem ser considerados a existência de tutoriais, suporte ao consumidor, boa usabilidade, manuais e outras características que auxiliam na utilização, porém quanto mais intuitivo melhor.

Uma pesquisa realizada com 605 gerentes de projeto, a maioria afiliados ao PMI, perguntou a eles a importância de uma série de características de um SGP (Figura 5), e posteriormente a impressão que eles têm dessas mesmas características nos SGPs que utilizam (Figura 6). Mais de 90% responderam que consideram importante ou muito importante que o sistema deva ser fácil de utilizar, no entanto vê-se que menos da metade deles tem uma impressão positiva da facilidade do uso dos SGPs que utilizam. E entre as características menos importantes do SGP estariam a facilidade de uso para os patrocinadores, onde mais de 40% responderam não achar muito importante e menos de 20% têm uma impressão positiva.

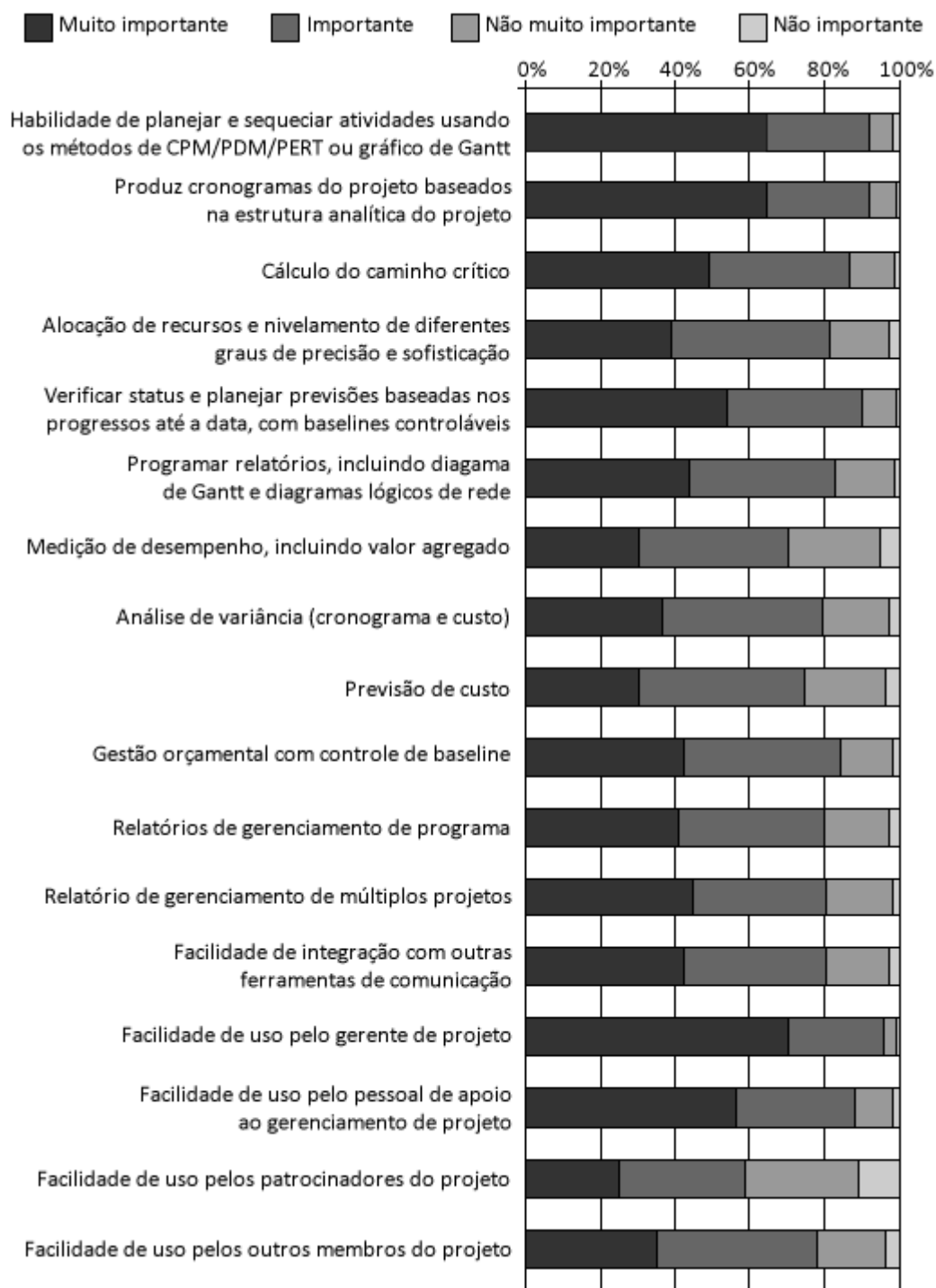


Figura 5: Importância das características de um SGP para gerenciar um projeto.

Fonte: Adaptado de Lockwood, Arden. The Project Manager's Perspective on Project Management Software Packages. 2008.

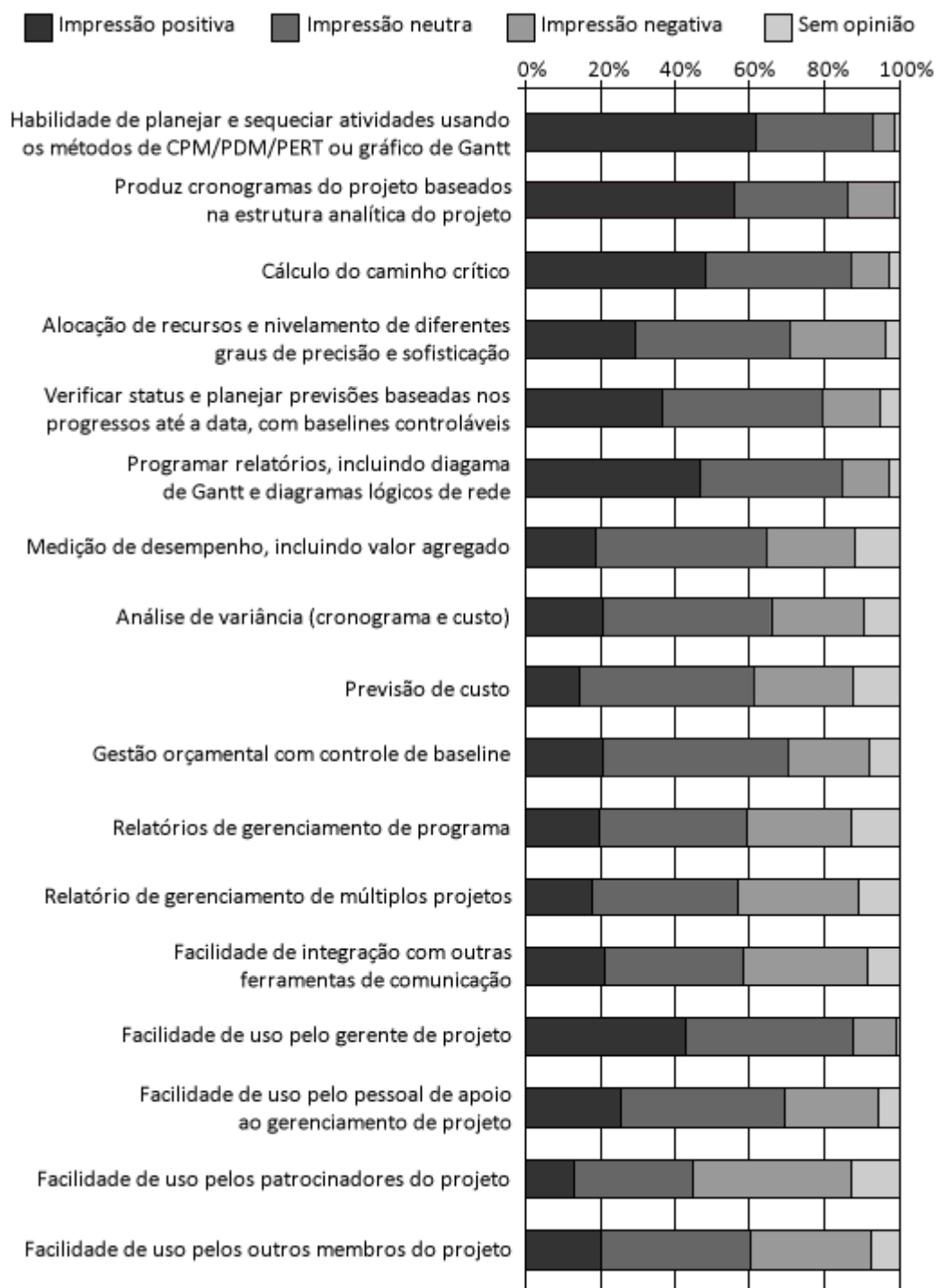


Figura 6: Impressão sobre as características dos SGP's utilizados.

Fonte: Adaptado de Lockwood, Arden. The Project Manager's Perspective on Project Management Software Packages. 2008.

Kleim e Ludin (1998, p.162) ressaltam que, a pesar dos benefícios dessas ferramentas, existem diversos desafios em utilizá-las como: *“fazer as pessoas entenderem o propósito do software; reduzir a curva de aprendizado; aumentar a*

receptividade das saídas; [e] produzir informação confiável ao invés de lixo”. Existem também desvantagens na utilização de SGPs, principalmente os iniciantes na área que acreditam que gerenciar um projeto é apenas utilizar um SGP (HEERKENS, 2002). Nem mesmo a mais poderosa ferramenta de gerenciamento substitui as habilidades interpessoais de um gerente (KERZNER, 2001). Por isso é importante reforçar que o SGP é somente uma ferramenta, que tem como objetivo auxiliar o gerenciamento do projeto, se for bem utilizada.

Existem diversos tipos de ferramentas com diversas funcionalidades, porém a facilidade de uso do software é uma das características mais valorizadas pelos usuários.

Dentre as ferramentas baseadas na *web* e de software livre o dotProject é um dos mais utilizados. Por ser uma das ferramentas de código disponível com maior presença no mercado foi a ferramenta escolhida para ser utilizada no trabalho. O próximo capítulo apresenta as características dessa ferramenta.

2.2.1 dotProject

dotProject (<http://www.dotproject.net>) é um SGP que assim como outros SGPs, fornece ferramentas para o gerenciamento de tarefas, tempo, comunicação e compartilhamento. Além disso, o dotProject também é um *framework*, pois permite que suas funcionalidades sejam estendidas. É um SGP baseado na *web* e é *open-source*. A licença utilizada é a GNU *General Public License* (GPL), que permite: a utilização do software para qualquer fim; acesso ao código-fonte; redistribuição de cópias e a edição do código, desde que ele continue sendo aberto.

Possui mais de 900 mil downloads desde a versão 2.0, sendo que a última versão 2.1.5, de janeiro de 2011, possui quase 50 mil downloads (DOTPROJECT, 2011). O projeto possui registro no SourceForge desde 28 de fevereiro de 2001.

Segundo a documentação do dotProject (2010) a instalação básica traz os módulos do núcleo que são o mínimo necessário para gerenciar um projeto. São 17 módulos descritos a seguir em ordem alfabética:

1. **Arquivos:** permite o armazenamento e versionamento de arquivos por projetos.
2. **Backup:** permite a criação de uma cópia de segurança da base de dados, essa copia é um arquivo XML.
3. **Busca inteligente:** permite a busca de conteúdo através de todos os módulos.
4. **Calendário:** fornece as funcionalidades de edição de eventos e várias formas de visualização do calendário como em miniatura, por dia, por semana e por mês.
5. **Contatos:** fornece o armazenamento de informação de contatos relacionados ao projeto.
6. **Departamentos:** é um subconjunto de empresas, possibilita o agrupamento de usuários ou projetos dentro de uma empresa.
7. **Empresas:** são as entidades que vão agrupar projetos e usuários.
8. **Fóruns:** similar aos fóruns de discussões da web que utiliza a estrutura de tópicos.
9. **Gráfico de Gantt:** gráfico que mostra as tarefas do projeto e a duração delas. (Figura 7).

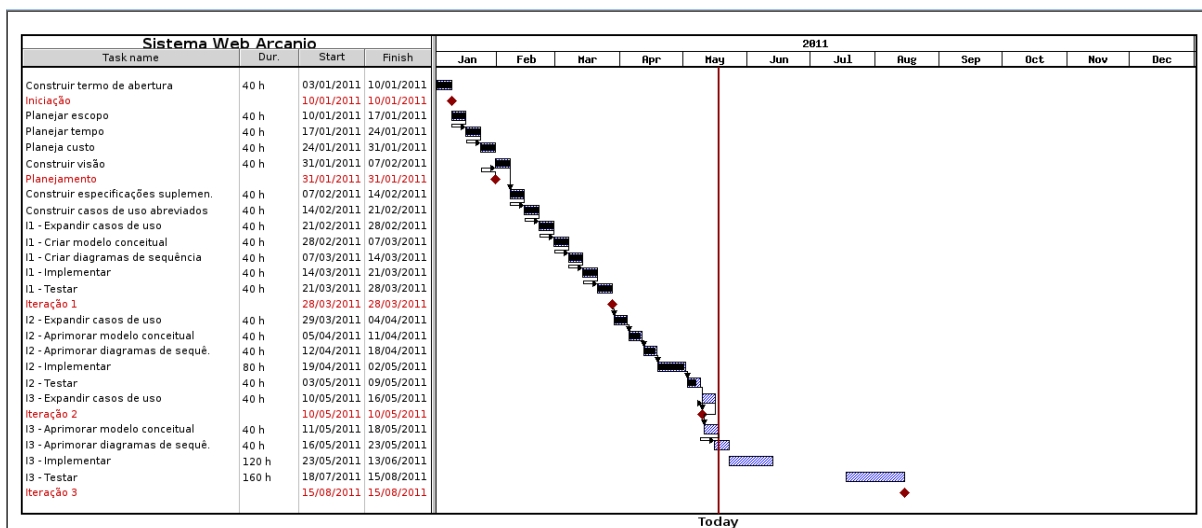


Figura 7: Gráfico de Gantt gerado pelo dotProject.

10. **Histórico:** permite a auditoria através de logs de acessos e mudanças.

11. **Links:** gravação de links importantes ou frequentemente utilizados.

12. **Projetos:** os projetos agrupam as tarefas que precisam ser executadas.

(Figura 8)

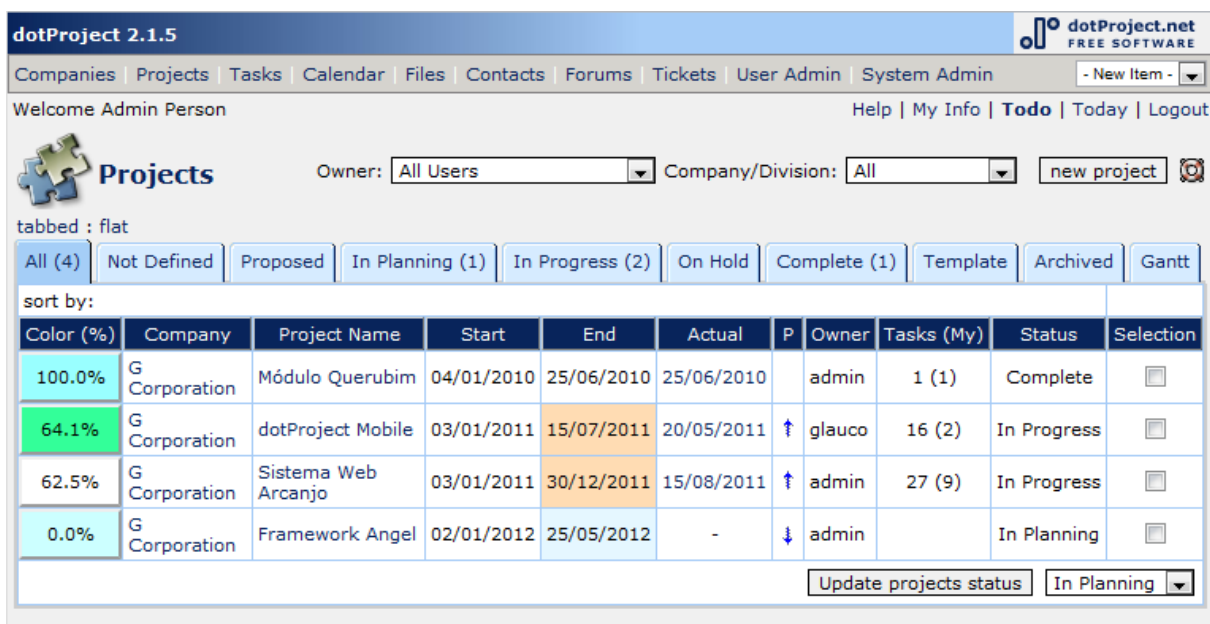


Figura 8: Menu de projetos.

13. **Recursos:** aonde são incorporados recursos não humanos, como salas ou equipamentos.

14. **Relatórios:** permite a geração de relatórios de andamento do projeto.

15. **Sistema administrativo:** inclui todas as configurações das opções entre elas a de definir o papel dos usuários do sistema e as permissões.

16. **Tarefas:** unidades de trabalho que mostram o trabalho que deve ser feito, quanto tempo deve ser gasto. Também é possível atualizá-las e acompanhar seu progresso (Figura 9).

Tasks

Tasks (Inactive)

Forums

Gantt Chart

Task Logs

Events

Files

Expand All : Collapse All

Show: ☐ Incomplete Tasks Only

	Pin	New Log	Work	P	Task Name	Task Creator	Assigned Users	Start Date	Duration	Finish Date	
		Log	100%		Construir termo de abertura	admin	admin (100%)	03/01/2011 08:45 am	5 days	10/01/2011 08:45 am	
		Log	100%		Iniciação	admin	admin (100%)	10/01/2011 08:45 am	0 hours	10/01/2011 09:45 am	
		Log	100%		Planejar escopo	admin	admin (100%)	10/01/2011 08:45 am	5 days	17/01/2011 08:45 am	
		Log	100%		Planejar tempo	admin	admin (100%)	17/01/2011 08:45 am	5 days	24/01/2011 08:45 am	
		Log	100%		Planeja custo	admin	admin (100%)	24/01/2011 08:45 am	5 days	31/01/2011 08:45 am	
		Log	100%		Planejamento	admin	admin (100%)	31/01/2011 09:45 am	0 hours	31/01/2011 09:45 am	
		Log	100%		Construir visão	gabriel	gabriel (100%)	31/01/2011 09:45 am	5 days	07/02/2011 08:45 am	
		Log	100%		Construir especificações suplementares	gabriel	gabriel (100%)	07/02/2011 08:45 am	5 days	14/02/2011 08:45 am	
		Log	100%		Construir casos de uso abreviados	gabriel	gabriel (100%)	14/02/2011 08:45 am	5 days	21/02/2011 08:45 am	
		Log	100%		I1 - Expandir casos de uso	gabriel	gabriel (100%)	21/02/2011 08:45 am	5 days	28/02/2011 08:45 am	
		Log	100%		I1 - Criar modelo conceitual	gabriel	gabriel (100%)	28/02/2011 08:45 am	5 days	07/03/2011 08:45 am	
		Log	100%		I1 - Criar diagramas de sequência	gabriel	gabriel (100%)	07/03/2011 08:45 am	5 days	14/03/2011 08:45 am	
		Log	100%		I1 - Implementar	rafael	rafael (100%)	14/03/2011 08:45 am	5 days	21/03/2011 08:45 am	
		Log	100%		I1 - Testar	miguel	miguel (100%)	21/03/2011 08:45 am	5 days	28/03/2011 08:45 am	
		Log	100%		Iteração 1	admin	admin (100%)	28/03/2011 08:15 am	0 hours	28/03/2011 05:00 pm	
		Log	100%		I2 - Expandir casos de uso	gabriel	gabriel (100%)	29/03/2011 08:00 am	5 days	04/04/2011 05:00 pm	
		Log	95%		I2 - Aprimorar modelo conceitual	gabriel	gabriel (100%)	05/04/2011 08:00 am	5 days	11/04/2011 05:00 pm	
		Log	95%		I2 - Aprimorar diagramas de sequência	gabriel	gabriel (100%)	12/04/2011 08:00 am	5 days	18/04/2011 05:00 pm	
		Log	95%		I2 - Implementar	rafael	rafael (100%)	19/04/2011 08:00 am	10 days	02/05/2011 05:00 pm	
		Log	70%	↑	I2 - Testar	miguel	miguel (100%)	03/05/2011 08:00 am	5 days	09/05/2011 05:00 pm	
		Log	100%		Iteração 2	admin	admin (100%)	10/05/2011 08:00 am	0 hours	10/05/2011 08:00 am	
		Log	0%	↑	I3 - Expandir casos de uso	gabriel	gabriel (100%)	10/05/2011 08:00 am	5 days	16/05/2011 05:00 pm	
		Log	0%		I3 - Aprimorar modelo conceitual	gabriel	gabriel (100%)	11/05/2011 08:30 am	5 days	18/05/2011 08:30 am	
		Log	0%		I3 - Aprimorar diagramas de sequência	gabriel	gabriel (100%)	16/05/2011 08:30 am	5 days	23/05/2011 08:30 am	
		Log	0%		I3 - Implementar	rafael	rafael (100%)	23/05/2011 08:30 am	15 days	13/06/2011 08:30 am	
		Log	0%		I3 - Testar	miguel	miguel (100%)	18/07/2011 08:30 am	20 days	15/08/2011 08:30 am	
		Log	0%		Iteração 3	admin	admin (100%)	15/08/2011 08:30 am	0 hours	15/08/2011 05:00 pm	
Summaries:							03/01/2011	140 hours	15/08/2011		
Key:	=Future Task =Started and on time =Should have started =Overdue =Done										Expand All : Collapse All

Figura 9: Lista de tarefas.

17. **Usuário administrador:** utilizado para gerenciar o dotProject.

O dotProject também é um *framework*, permitindo o desenvolvimento de novos módulos e outras customizações como por exemplo o tema. Além dos módulos do núcleo, existem módulos adicionais que foram desenvolvidos pela comunidade, e não pela equipe de desenvolvimento do dotProject. O código fonte do dotProject está disponível nos controladores de versão Github e Subversion. É feito nas

linguagens PHP e JavaScript. Utiliza o banco de dados MySQL e precisa de um servidor *web* para sua instalação.

De acordo com Jordan (2007, p.12) o dotProject é uma boa escolha para organizações que precisam de um SGP que *“não tenha taxas, tenha uma generosa licença de uso, seja estável, funcione nos grandes browsers, tenha uma comunidade que preste suporte, tenha permissões granulares, e seja escalável”*. Pode ser aplicado em diversos ambientes como escritórios, gerenciamento pessoal, empresas e escolas (DOTPROJECT, 2010). Possui a vantagem de SGPs baseados na *web* que é a utilização através do browser, não necessitando de instalação e possibilitando o acesso remoto (HORINE, 2009). As permissões granulares fornecem outra vantagem de SGPs baseados na *web* citada por Horine (2010, p.358), que é a maior visibilidade e abertura do SGP, onde todos os interessados (*stakeholders*) do projeto podem acessar o dotProject, cada um com uma permissão diferenciada.

Apesar de ser uma ferramenta com uma boa representatividade no mercado, ainda não foi visto nenhum esforço em lançar uma versão do dotProject para dispositivos móveis. E apesar de ser um sistema *web*, que pode ser acessado dos navegadores de dispositivos móveis, sua interface não é otimizada para telas pequenas, como é mostrado na Figura 10. Essa falta de otimização pode fazer com que o usuário final tenha dificuldades de acessar o conteúdo que deseja obter. Outras desvantagens que podem ocorrer: formatação incorreta dos dados; navegador não ser capaz de renderizar todos os recursos do site; ter que colocar e retirar *zoom*; ou perder muito tempo com a rolagem do site.

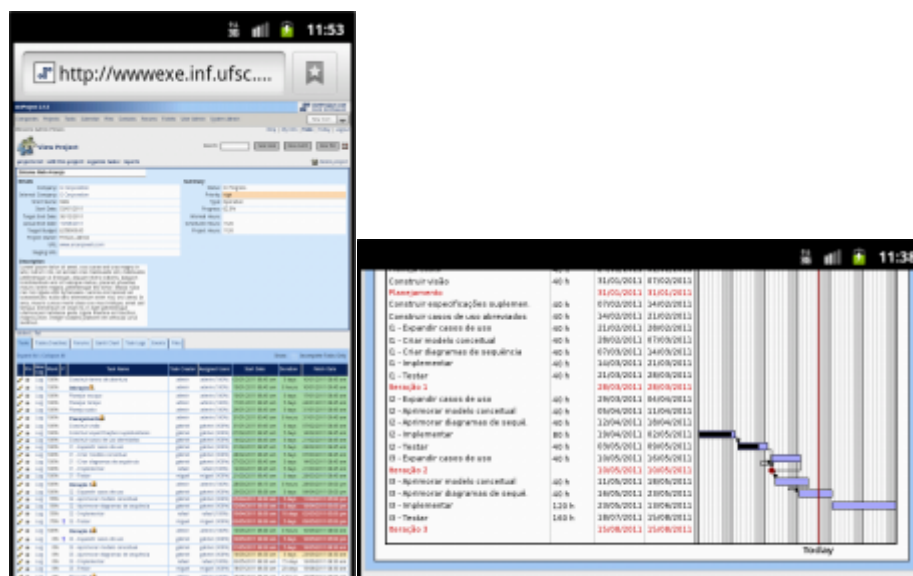


Figura 10: Página de projetos (esquerda) e diagrama de Gantt (direita) na tela do emulador de 3.3 polegadas.

O próximo capítulo apresenta os aplicativos móveis que é o formato pretendido para a extensão do dotProject.

2.3 APLICAÇÕES MÓVEIS

Primeiramente é necessário definir o que são dispositivos móveis e suas características antes de entender o que diferenciam suas aplicações. Firtman (2010) define o dispositivo móvel como sendo aquele que possui as seguintes características:

- **Portátil:** Que possa ser carregado sem muito esforço, como um molho de chaves ou uma carteira.
- **Pessoal:** Cada pessoa tem seu próprio dispositivo, com suas customizações e normalmente não se compartilha sua utilização.
- **Companheiro:** É possível estar com o dispositivo o tempo todo.

- **Fácil de usar:** No sentido de condições de uso, como por exemplo, não é necessário utilizar o dispositivo sentado, ou num lugar com ótima iluminação, ou ter que esperar dois minutos para que o sistema esteja pronto para uso.
- **Conectado:** Deve ser possível conectar na internet quando necessário.

A definição de Firtman (2010) inclusive desconsidera alguns celulares como dispositivos móveis por não possuírem a última característica de ser conectado. Além dos celulares, também são considerados dispositivos móveis: *tablets*, por exemplo, o iPad; *e-readers*, por exemplo, o Kindle; e outros aparelhos com as características de dispositivos móveis como, por exemplo, o iPod Touch (Figura 11).



Figura 11: Exemplos de dispositivos móveis.

Outras capacidades e funcionalidades de dispositivos móveis podem ser divididas em três classes, conforme Figura 12 (FITZEK, REICHART, 2007):

- **Interface com o usuário.** Diferentes tipos de entrada e saída de dados como alto-falante, microfone, câmera, tela, sensor e teclado. Alguns tipos de sensores são de: movimento, localização, iluminação entre outros.

- **Interface de comunicação.** Divididas em duas: a interface celular como GSM e 3G; e a interface de curto alcance como *wi-fi*, *bluetooth* e infravermelho.
- **Recursos internos.** Componentes da estrutura do dispositivo como unidade de armazenamento, unidade de processamento central e bateria.



Figura 12: Capacidades e funcionalidades de um dispositivo móvel.

Fonte: Adaptado de Fitzek, Frank & Reichart, Frank. Mobile Phone Programming. 2007.

As interfaces de comunicação não estão disponíveis em todos os dispositivos e nem o tempo todo (FITZEK, REICHART, 2007). As comunicações de curto alcance geralmente precisam de outros dispositivos como roteadores wireless ou outros celulares. A cobertura das redes de comunicação também não atinge todos os lugares, principalmente a rede 3G que é uma tecnologia recente. Até mesmo a rede de voz sofre interferências, por exemplo, dentro de elevadores.

Comparados com os computadores, os recursos internos dos dispositivos móveis possuem algumas limitações. O espaço de armazenamento é reduzido, a CPU não é tão rápida e a bateria limita o tempo de uso do dispositivo (FITZEK, REICHART, 2007). Todas essas características influenciam nas aplicações desenvolvidas, pois

não podem ser tão pesadas que não caibam nos dispositivos ou que não possam ser processados em tempo razoável. Aplicações que utilizem muito a CPU vão diminuir o tempo de vida da bateria, algo que deve ser evitado.

Essas limitações na maior parte do tempo ficam invisíveis ao usuário, por isso o ponto principal no desenvolvimento de aplicações móveis é a usabilidade do sistema (MIKKONEN, 2007). Firtman (2010) ressalta as principais diferenças entre a usabilidade do computador e de dispositivos móveis, conforme quadro 2.

Quadro 2: Diferenças entre dispositivos móveis e computadores.

Característica	Dispositivo Móvel	Computador
Tamanho da tela	Menor do que 4 polegadas	Maior do que 15 polegadas
Resolução da tela	Não existe padrão e varia bastante. Celulares com tela sensível ao toque possuem resoluções variando de 240x480 até 640x960	Monitores <i>full</i> HD possuem resolução de 1920 x 1080
Orientação da tela	Vertical, horizontal e quadrada (Figura 13)	Horizontal
Métodos de entrada	Teclado numérico, teclado QWERTY, teclado virtual na tela, toque, reconhecimento de voz	Teclado e mouse



Figura 13: Capacidades e funcionalidades de um dispositivo móvel.

Fonte: Adaptado de Firtman, Maximiliano. Programming the Mobile Web. 2010.

Dispositivos móveis são utilizados em diferentes contextos do que computadores pessoais e *notebooks* (MEHTA, 2008). Normalmente utiliza-se o computador sentado confortavelmente, em um ambiente bem iluminado, em um tempo dedicado para essa utilização (MIKKONEN, 2007). Enquanto o celular é utilizado em diversas situações, entre uma tarefa e outra, para ter um acesso rápido a alguma informação (MIKKONEN, 2007).

Além das características físicas particulares dos dispositivos móveis, ainda existem diversas marcas, modelos e plataformas que precisam ser levados em conta no desenvolvimento. Cada uma possui suas próprias ferramentas, linguagem de programação e uma série de dispositivos que a suportam. As principais no mercado são:

- **iOS:** Sistema operacional proprietário da Apple utilizado em iPhones, iPad e iPods.
- **Android:** Sistema operacional *open-source* desenvolvida pela Google.
- **Symbian:** Atualmente *open-source*, foi desenvolvida pela Nokia.
- **Java:** Java não é um sistema operacional, mas diversos sistemas operacionais tem suporte para essa plataforma.
- **BlackBerry:** Desenvolvida pela Research In Motion. É um sistema operacional que visa o mercado corporativo.
- **Windows:** Existem o Windows Mobile e, mais recentemente, o lançamento do Windows Phone 7.

E ainda outras como o Web OS da Palm, o Meego da Intel e Nokia, o Bada da Samsung.

Segundo o Gartner (2011) o Android foi responsável por 22,7% da venda de *smartphones* para usuários finais em 2010 (Figura 14, esquerda) e deve dominar 38,5% das vendas em 2011 (Figura 14, direita). Possui a menor curva de aprendizado e utiliza a linguagem Java para desenvolvimento (CONSTANTINOU, CAMILLERI e KAPETANAKIS, 2010).

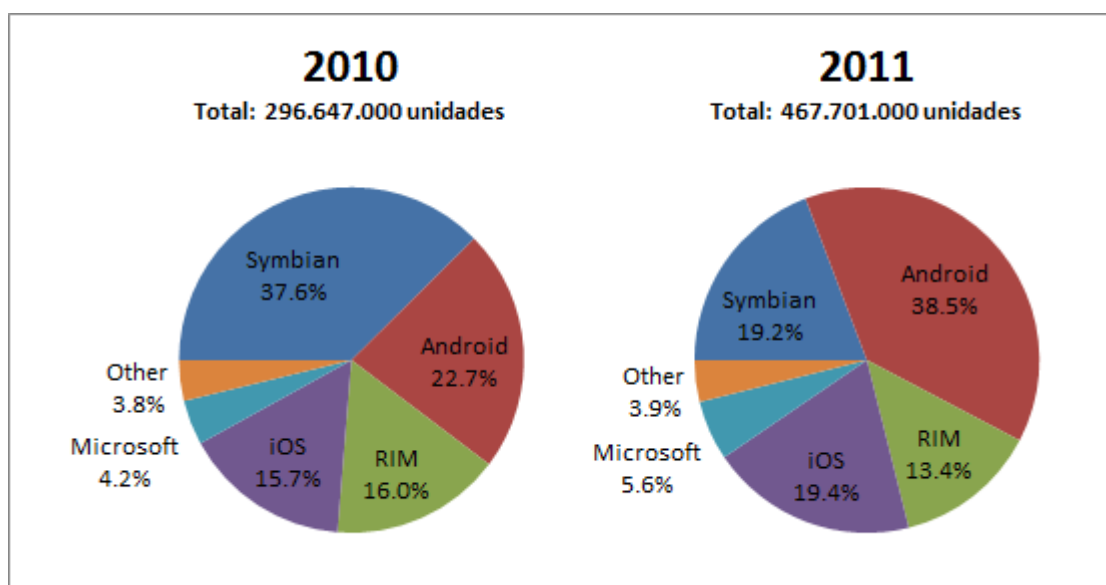


Figura 14: Porcentagem de venda de *smartphones* em 2010 (esquerda) e a previsão para 2011 (direita).

Fonte: Adaptado de Gartner. Gartner Says Android to Command Nearly Half of Worldwide Smartphone Operating System Market by Year-End 2012. 2011.

2.3.1 Android

Android é um sistema operacional independente de marcas e modelos. É *open-source* e roda no Linux. É um sistema multi-tarefa baseado no conceito de máquinas virtuais que executam *bytecode*. A linguagem principal dessa plataforma é o Java (ANDROID DEVELEPORS, 2011). Não é uma obrigatoriedade, mas atualmente todos os dispositivos Android possuem tela sensível ao toque. Vários desses dispositivos possuem GPS, bússola digital e acelerômetro.

O Android tem sofrido um problema de fragmentação, pois por ser um sistema de código livre, ele pode ser alterado. As construtoras têm customizado as versões de Android lançadas e não acompanham o lançamento das novas versões do sistema operacional. A cada duas semanas a Google divulga um gráfico baseado no número de dispositivos Android que acessaram o *Android Market* durante aquelas duas semanas. A predominância do mercado atual é da versão 2.2, porém a versão 2.1 ainda ocupa uma grande fatia do montante (Figura 15). Apesar da versão 3.0 já estar no mercado, os desenvolvedores têm que ficar atentos à situação do mercado.

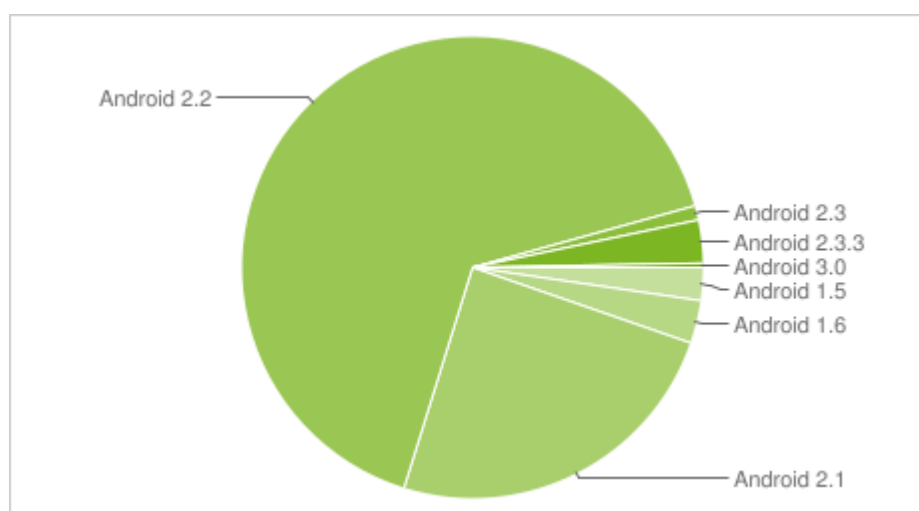


Figura 15: Gráfico baseado no número de dispositivos Android que acessaram o *Android Market* no período de 18 de abril a 2 de maio de 2011.

Fonte: *Android Developers. Platform Versions*. 2011.

Por seguir a filosofia de ser open-source, por possuir ferramentas gratuitas e por estar crescendo no mercado, foi o sistema escolhido para receber a versão móvel do dotProject.

No próximo capítulo é descrita a revisão do estado da arte, que avalia os softwares para gerência de projetos para dispositivos móveis de plataforma Android.

3 REVISÃO DO ESTADO DA ARTE

Para levantar o estado da arte é realizada uma revisão sistemática da literatura com o objetivo de identificar, avaliar e interpretar as pesquisas importantes na área de desenvolvimento de aplicações móveis de gerenciamento de projetos. Somente foram levados em conta os softwares que possuam versão móvel para a plataforma Android, por ser o alvo desse trabalho. A revisão foi realizada seguindo as fases e tarefas propostas por Kitchenham (2004), que são:

1. Primeira fase: planejar a revisão
 - 1.1. Identificar a necessidade da revisão
 - 1.2. Desenvolver o protocolo de revisão
2. Segunda fase: conduzir a revisão
 - 2.1. Identificar as fontes de busca relevantes
 - 2.2. Selecionar os estudos primários
 - 2.3. Estudar a qualidade da avaliação
 - 2.4. Extração e monitoração dos dados
 - 2.5. Sintetizar os dados
3. Terceira fase: documentar a revisão

O objetivo é encontrar aplicativos para dispositivos móveis para gerenciar projetos e revisá-los a fim de responder as seguintes questões:

- É relevante a utilização de software para gerenciamento de projetos via dispositivos móveis?
- Quais funcionalidades são suportadas via dispositivos móveis?
- Quais os benefícios trazidos?

- Quais questões de usabilidade devem ser levadas em conta no desenvolvimento?

As fontes utilizadas para a busca foram a IEEEXplore (<http://ieeexplore.ieee.org>), a ACM Digital Library (<http://portal.acm.org>), Source Forge (<http://sourceforge.net>), Google Code (<http://code.google.com>) e Google (<http://www.google.com>). A busca foi realizada no dia 02 de novembro de 2010 e os dados da pesquisa estão no quadro 3.

Quadro 3: Dados da pesquisa.

Base de dados	String de busca	Result.	Analís.	Seleci.
IEEEXplore	"Abstract": "project management" AND (mobile OR cellphone OR phone)	71	71	0
ACM Digital Library	(Abstract: "project management") and (Abstract: mobile or Abstract: cellphone or Abstract: phone)	18	18	0
SourceForge	"project management" AND mobile	5	5	0
Google Code	("project management software" OR "project management application") AND (mobile OR phone OR cellphone)	8	8	1
Google	("project management software" OR "project management application") mobile	4.520.000	100	3

Todos os resultados da busca foram analisados pelo autor em termos de sua relevância adotando três critérios de exclusão.

1. Resultados que pelo título e resumo não mostraram nenhuma relação com dispositivos móveis e/ou gerência de projetos. Entendeu-se por resumo nas fontes IEEEXplore e ACM Digital Library o *abstract* dos artigos resultantes. E nas fontes SourceForge, Google Code e Google a descrição dos itens resultantes.
2. Resultados que não apareciam entre os 100 mais relevantes segundo o motor de busca das fontes utilizadas.

3. Resultados que não possuíam versão para a plataforma Android e que não possuíam integração com algum SGP baseado na *web* ou *desktop*.

Aplicando estes critérios de exclusão restaram 4 resultados das buscas originais. Estes resultados foram examinados através do site da ferramenta, imagens, vídeos e documentação. As ferramentas encontradas são descritas a seguir.

3.1 BASECAMP

Basecamp (<http://basecamphq.com>) é um SGP proprietário baseado na *web* para empreendedores, *freelancers*, pequenos negócios e grupos dentro de grandes organizações. A ferramenta foca na comunicação, compartilhamento de documentos, gerenciamento de tarefas e reutilização de modelos de projeto. Está disponível em diversas linguagens entre elas inglês, português e espanhol. As versões para celular são extensões e não foram feitas pela desenvolvedora do Basecamp. Existem diversas extensões para várias plataformas móveis, duas delas são para Android: Beacon e Timy. Os aplicativos estão disponíveis em inglês. As funções contidas nessas extensões são:

- **Projetos:** Visualização de diferentes projetos (Figura 16, direita), e para cada projeto acesso a um resumo das informações do projeto, como mensagens, tarefas e comentários.
- **Mensagens:** Adição de mensagens que podem ser visualizadas pelos membros do projeto.
- **Comentários:** Comentários em mensagens e metas do projeto.
- **Tarefas:** Adição de grupos de tarefas e tarefas, atribuição de responsabilidade pela tarefa.

- **Metas:** Metas do projeto com definição de data e responsabilidade de entrega.
- **Sincronização de dados:** Os dados são sincronizados com a versão *web* quando existe conexão com a rede.
- **Horas trabalhadas:** Contador de horas de trabalho (Figura 16, esquerda).

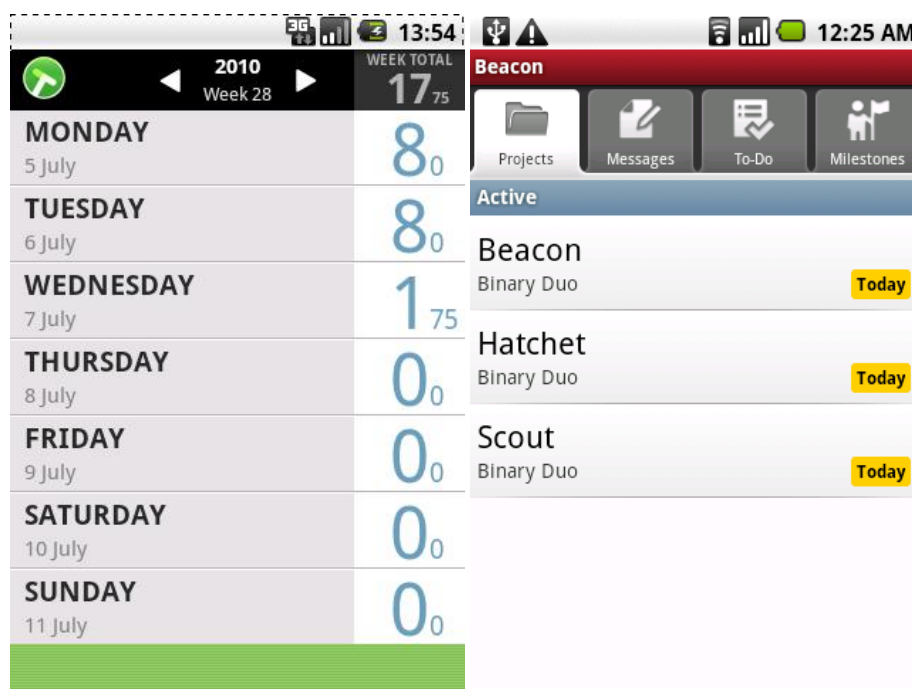


Figura 16: Interface do Timy (esquerda) e Beacon (direita).

3.2 CONNECTWISE

ConnectWise (<http://www.connectwise.com>) é uma solução completa para negócios na área de tecnologia da informação, incluindo ferramentas para gerenciar projetos. É um software proprietário e *desktop*. A versão móvel é um adicional que pode ser comprado separadamente. Possui versões para *web* móvel e aplicativos para iPhone e Android. O aplicativo está disponível em inglês. O aplicativo possui armazenamento de dados locais permitindo a utilização mesmo sem conexão com a internet. Outras funcionalidades são:

- **Atividades:** Visualização e criação de novas atividades com descrição da tarefa, tempo de duração e atribuição de responsáveis.
- **Agenda:** Visualização por dia da semana com as tarefas do dia (Figura 17, esquerda).
- **Lista de contatos:** Visualização de contatos relacionados ao projeto.

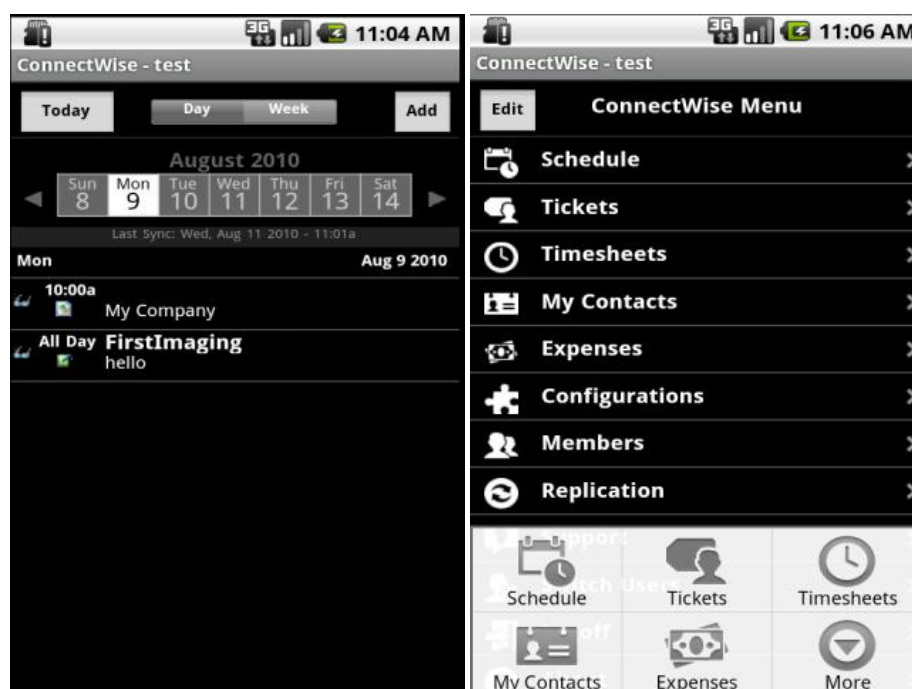


Figura 17: Interface do ConnectWise Mobile mostrando agenda (esquerda) e menu principal (direita).

3.3 GTDAGENDA

Gtdagenda (<http://www.gtdagenda.com>) é um SGP proprietário baseado na *web* com funcionalidades padrão como definição de objetivos, lista de projetos, divisão por tarefas e ações, utilização de contextos que define onde ou como a tarefa deve ser cumprida. As versões móveis estão disponíveis nas plataformas *web* móvel e Android. O aplicativo está disponível em inglês. Através do aplicativo é possível:

- **Projetos:** Visualização de projetos.
- **Tarefas:** Visualização de tarefas.

- **Sincronização de dados:** Os dados são sincronizados com a versão web quando existe conexão com a rede.
- **Mensagens:** Mensagens recebidas dentro do sistema (Figura 18).

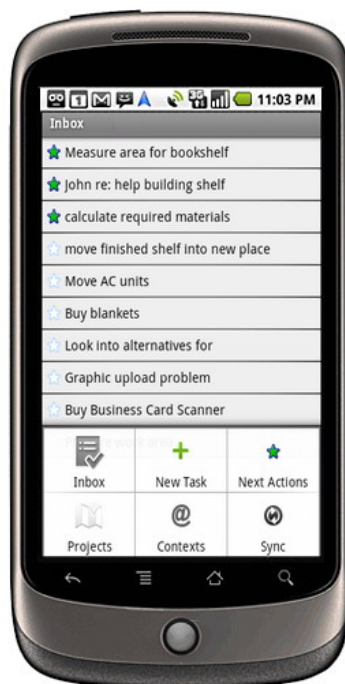


Figura 18: Interface do Gtdagenda, mostrando a caixa de mensagens.

3.4 RALLY

Rally (<http://www.rallydev.com>) é um SGP proprietário baseado na *web* para gerenciar projetos de desenvolvimento ágil. Entre suas funcionalidades estão: geração de relatórios; quadro de estórias, de tarefas e Kanban; e integração com outras ferramentas. Existem versões móveis para iPhone e Android. A versão para Android está sendo desenvolvida por um funcionário da empresa Rally, porém não é um componente oficial. O aplicativo está disponível em inglês. Ainda está em desenvolvimento, possui código aberto, e até agora possui as seguintes funcionalidades:

- **Planning poker:** para que os membros do projeto façam uma estimativa de esforço por tarefa (Figura 19).

- **Tarefas:** Visualização de tarefas.
- **Situação da iteração:** Visualização do andamento do projeto.

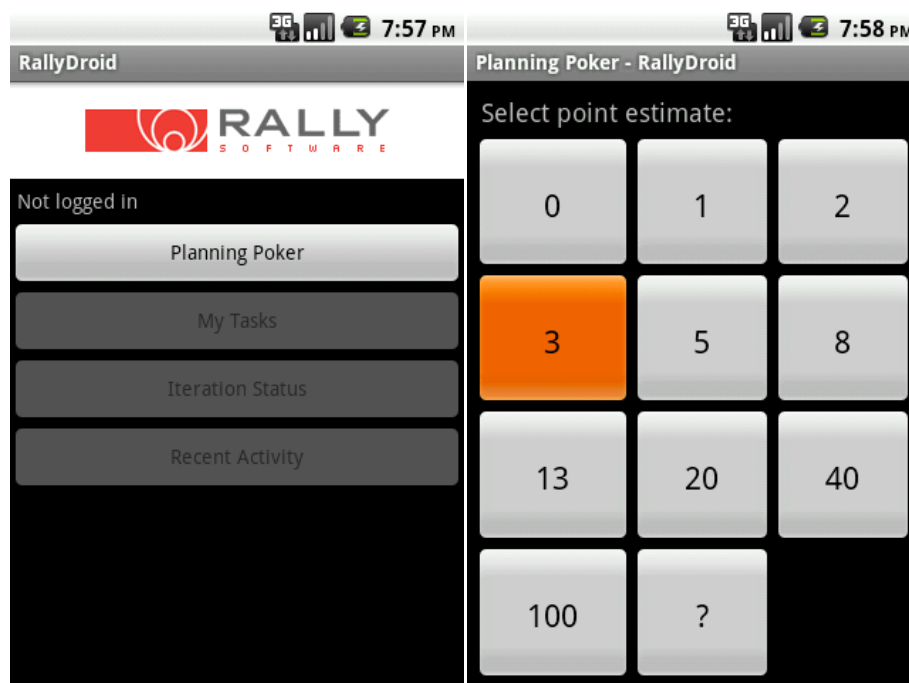


Figura 19: Interface do RallyDroid.

3.5 DISCUSSÃO

Poucos aplicativos foram encontrados, provavelmente pelo fato da plataforma Android ser nova e ter aparecido no mercado no último ano.

Além dos aplicativos que foram selecionados para revisão, existiam outros aplicativos móveis de gerenciamento de projetos para outras plataformas, como iOS e *web* móvel. Os aplicativos para *web* móvel não possuíam uma usabilidade tão atrativa quanto à dos aplicativos para Android e iOS, pois utilizavam uma tecnologia com recursos limitados para *web* móvel. Os aplicativos para iOS possuem a restrição de só funcionar em iPhone, iPods ou iPads, prendendo o usuário a um tipo específico de aparelho da construtora Apple. Por outro lado a plataforma Android é mais flexível e está presente em alguns aparelhos um pouco mais acessíveis e de diversas construtoras.

Não foram encontrados aplicativos móveis para os SGPs mais utilizados descritos no capítulo 2.2. Os mais próximos achados foram aplicativos para Windows Mobile que permitiam a visualização de arquivos gerados pelo MS Project da Microsoft, como por exemplo, o Pocket Plan (<http://www.twiddlebit.com>).

Com base nesta avaliação as informações relevantes referente às questões citadas anteriormente são descritas a seguir.

Quanto à relevância da utilização de software para gerenciamento de projetos via dispositivos móveis, como não foi encontrado nenhum trabalho científico, não ficou clara a sua relevância.

As funcionalidades suportadas pelos aplicativos analisados foram identificadas e apresentadas no quadro 4. Todos são relacionados a um software de gerenciamento de projetos *web* ou *desktop*. As características mais relevantes foram selecionadas, e nenhum dos aplicativos possui todas elas. Uma das características consideradas mais relevantes é a sincronização de dados, pois permite que o usuário consiga manipular os dados mesmo não possuindo uma conexão de rede.

Quadro 4: Comparação das ferramentas.

Aplicativo	Timy	Beacon	ConnectWise Mobile	Gtdagenda	Rallydroid
Empresa	Nascom	Binary Duo	ConnectWise	David Allen	Rally
Site	www.timy.be	www.binaryduo.com/beacon	www.connectwise.com/connectwise-mobile-1.aspx	www.gtdagenda.com/android.php	code.google.com/p/rallydroid
SGP integrado	Basecamp	Basecamp	ConnectWise	Gtdagenda	Rally
Língua	Inglês	Inglês	Inglês	Inglês	Inglês
Disponível no Android Market	Sim	Sim	Sim	Não	Não
Licença	Proprietário	Proprietário	Proprietário	Proprietário	Open source
Preço	Grátis	Pago	Grátis	Não disponível	Grátis
Versão	1.1.1	1.0.7	1.3	Informação não disponível	Em desenvolvimento
Features					
Sincronização de dados	Sim	Sim	Sim	Sim	Não
Projetos (visualização/edição)	Sim	Sim	Sim	Sim	Não
Tarefas (visualização/edição)	Não	Sim	Sim	Sim	Sim
Cronograma (visualização/edição)	Não	Não	Sim	Não	Não
Quadro de horários	Sim	Não	Sim	Não	Não
Contatos	Não	Sim	Sim	Não	Não
Mensagens (para comunicação com a equipe)	Não	Sim	Não	Sim	Não

Quanto aos benefícios, os aplicativos móveis se mostraram como um adicional ao sistema *web* ou *desktop*. No SGP são feitos a iniciação e o planejamento do projeto, enquanto a componente móvel auxilia a fazer a monitoria e controle do projeto. Em relação ao monitoramento e controle em geral os aplicativos analisados focam em: monitorar e controlar o trabalho do projeto, controlar o cronograma e reportar o desempenho.

Na questão de usabilidade não foi possível realizar um estudo aprofundado visto que havia algumas limitações como aplicativos pagos ou que nem estavam disponíveis no *Android Market*.

Assim como os SGPs, os aplicativos para gerenciar projetos em dispositivos móveis se mostraram mais atuantes no grupo de processos de monitoramento e controle. As áreas de conhecimento mais abordadas são principalmente as de tempo e comunicação, e um pouco de escopo. Esses aplicativos basicamente focam nas funcionalidades de: rastreamento e monitoramento de projetos e tarefas; cronograma do projeto; e comunicação com membros da equipe.

4 DESENVOLVIMENTO DE UM APLICATIVO MÓVEL INTEGRADO AO DOTPROJECT

Com base nos resultados da revisão sistemática de literatura, foi desenvolvido o aplicativo móvel dotProject Mobile.

Com respeito às aplicações móveis é importante considerar a utilidade das aplicações sendo desenvolvidas. Nem todas as aplicações fazem sentido em serem estendidas para dispositivos móveis. Dentro desta questão serão analisados possíveis cenários de uso de um aplicativo móvel complementando um SGP baseado na *web*.

Conforme observado no estado da arte, o uso de dispositivos móveis no gerenciamento de projetos faz sentido principalmente no monitoramento e controle de projetos.

Nesse capítulo é apresentada a proposta da solução do problema baseado no que foi exposto anteriormente. São feitas uma análise de requisitos, a modelagem conceitual, a implementação e os testes do protótipo.

4.1 ANÁLISE DE REQUISITOS

A idéia é desenvolver um aplicativo que estenda o dotProject para dispositivos móveis. Os dispositivos móveis alvos são os *smartphones* que possuem o sistema operacional Android com versão 2.1 ou superior, por abranger a maior parte do mercado, como foi visto anteriormente. Esse sistema será um complemento à versão *web* do dotProject visando auxiliar o gerente de projetos na área de monitoramento e controle.

O aplicativo será utilizado pelo gerente de projetos em situações que este não está em frente ao seu computador e notebook. Nessas situações, a solução proposta nesse trabalho visa possibilitar o gerente de projeto a monitorar e controlar projetos via celular, um dispositivo hoje sempre presente.

Para este protótipo apenas a plataforma Android será contemplada por possuir uma baixa curva de aprendizado e por estar bem inserida no mercado. Porém futuramente poderá ser estendida para outras plataformas como o iOS ou Windows Phone 7. A intenção desse trabalho é desenvolver um primeiro protótipo para a plataforma Android para possibilitar a avaliação do conceito em si, verificando a questão de utilidade de suporte de ferramentas de GP via celulares.

Através da revisão do estado da arte foram identificadas as seguintes atividades do grupo de monitoramento e controle que são relevantes em termos de utilidade para dispositivos móveis:

Monitorar e controlar o trabalho do projeto: Segundo o PMBOK, é o processo utilizado para coletar, medir e disseminar informações sobre o desempenho e avaliar as medições e as tendências para efetuar melhorias no processo. Faz parte desse processo a emissão de relatórios de andamento, medição do progresso e previsão.

Reportar o desempenho: Segundo o PMBOK é o processo necessário para coletar e distribuir informações sobre o desempenho. Isso inclui relatório de andamento, medição do progresso e previsão.

As vantagens previstas por um sistema móvel são a possibilidade do gerente sempre estar informado sobre o projeto, podendo monitorá-lo e controlá-lo mesmo quando estiver longe de sua estação de trabalho. No caso de tarefas atrasadas o

gerente terá a possibilidade de prontamente entrar em contato com o responsável pela tarefa, seja por uma ligação ou por e-mail.

O primeiro protótipo permite aos usuários a visualização rápida das características do projeto, das tarefas atrasadas e a comunicação com os responsáveis pelas mesmas. Essas funcionalidades foram escolhidas, pois vão permitir que o gerente usufrua das vantagens de possuir informações sobre o projeto com mais mobilidade e vai auxiliá-lo a monitorar e controlar o andamento do projeto.

4.1.1 Casos de Uso

Conforme as atividades de monitoramento e controle identificados como úteis a serem suportados por ferramentas de GP via celulares, foca-se no presente trabalho nos seguintes casos de uso.

Autenticar usuário: O gerente precisa se autenticar para poder ter acesso às informações do servidor (descrito de forma detalhada no quadro 5).

Quadro 5: Caso de uso detalhado – Autenticar usuário.

Ator primário	Gerente de projeto
Pré-condições	Informações do servidor, login e senha. Acesso a internet. Aplicativo instalado no dispositivo.
Pós-condições	Gerente autenticado no sistema.
Fluxo principal	1. Usuário entra com informações de autenticação a. URL do servidor onde o dotProject está instalado b. Usuário c. Senha 2. Sistema autentica usuário 3. Sistema mostra uma lista de projetos
Fluxo alternativo	1a. Sistema já possui as informações de autenticação salvas 2a. Sistema não autentica usuário, por informações incorretas 1. Usuário abre o menu de opções 2. Usuário seleciona “definições” 3. Usuário reinsere informações de autenticação 3a. Sistema não mostra lista pois não existem projetos no servidor

Requisitos não funcionais	Celular com Android versão 2.1 ou superior Conexão com a internet
Frequência	Sempre que o aplicativo for utilizado

Visualizar projetos: O gerente de projetos precisa visualizar informações do projeto, porém encontra-se longe da sua estação de trabalho. Ele pega seu *smartphone*, verifica se há conexão com a internet e abre o aplicativo do dotProject. Ele se autentica, caso não o tenha feito anteriormente. Escolhe um dos projetos que gostaria de visualizar. Uma tela com as informações do projeto abre e o gerente pode navegar por ela (descrito de forma detalhada no quadro 6).

Quadro 6: Caso de uso detalhado – Visualizar projetos

Ator primário	Gerente de projeto
Partes interessadas	Gerente de projetos: Quer visualizar informações importantes e de forma resumida do projeto. Quer visualizá-las de forma rápida e clara. Empresa: Quer que o gerente esteja sempre informado do andamento dos projetos para que eles sejam entregues dentro do tempo, orçamento e escopo.
Pré-condições	Usuário autenticado.
Pós-condições	Gerente informado sobre as informações do projeto.
Fluxo principal	1. Usuário escolhe projeto 2. Sistema mostra lista de opções 3. Usuário escolhe visualizar projeto 4. Sistema mostra informações do projeto a. Nome do projeto b. Companhia c. Companhia interna d. Nome pequeno e. Data de início f. Data de fim planejado g. Orçamento planejado h. Dono do projeto i. URL j. Estado do projeto k. Prioridade l. Tipo m. Progresso n. Horas trabalhadas

	o. Horas agendadas p. Horas do projeto q. Descrição
Requisitos não funcionais	Rapidez no acesso à informação (menos de 1 segundo)
Frequência	Quase sempre que o aplicativo for utilizado

Visualizar tarefas atrasadas: O gerente de projetos quer verificar se existem tarefas atrasadas, porém encontra-se longe da sua estação de trabalho. Ele pega seu *smartphone*, verifica se há conexão com a internet e abre o aplicativo do dotProject. Ele se autentica, caso não o tenha feito anteriormente. Escolhe um dos projetos que gostaria de verificar. Uma tela com as tarefas atrasadas aparece e o gerente pode visualizar quem é o responsável por ela (Descrito de forma detalhada no quadro 7).

Quadro 7: Caso de uso detalhado – Visualizar tarefas atrasadas.

Ator primário	Gerente de projeto
Partes interessadas	Gerente de projetos: Quer visualizar informações importantes sobre tarefas atrasadas, tais como nome da tarefa e responsável por ela. Empresa: Quer que o gerente esteja monitorando as tarefas para que elas não atrasem.
Pré-condições	Usuário autenticado.
Pós-condições	Gerente informado sobre a existência de tarefas atrasadas.
Fluxo principal	1. Usuário escolhe projeto 2. Sistema mostra lista de opções 3. Usuário escolhe tarefas atrasadas 4. Sistema mostra duas abas a. Tarefas atrasadas b. Tarefas que deviam ter começado 5. Para cada aba sistema mostra uma lista de tarefas com as informações a. Nome da tarefa b. Responsável pela tarefa
Caminhos alternativos	5a. Sistema não mostra lista, pois não existem tarefas atrasadas
Requisitos não funcionais	Rapidez no acesso à informação (menos de 1 segundo)
Frequência	Quase sempre que o aplicativo for utilizado

Entrar em contato com responsável por tarefa atrasada: O gerente de projetos quer entrar em contato com o responsável por uma (ou mais) tarefa(s) atrasada(s). O gerente navega pelo projeto escolhido, visualiza a lista de tarefas atrasadas e acessa o menu de contexto dela. O gerente escolhe entre ligar ou mandar e-mail e entra em contato com o responsável (Descrito de forma detalhada no quadro 8).

Quadro 8: Caso de uso detalhado - Entrar em contato com responsável por tarefa atrasada.

Ator primário	Gerente de projeto
Partes interessadas	Gerente de projetos: Quer entrar em contato com o responsável por alguma tarefa atrasada para saber o motivo do atraso. Empresa: Quer que o gerente tome alguma atitude quando alguma tarefa estiver atrasada.
Pré-condições	Usuário autenticado. Tarefa atrasada selecionada.
Pós-condições	Gerente informado sobre o motivo do atraso.
Fluxo principal	1. Usuário acessa o menu de contexto da tarefa 2. Sistema mostra as opções a. Telefone b. E-mail 3. Usuário escolhe uma opção 4. Sistema abre o aplicativo do celular de acordo com a opção selecionada 5. Usuário entra em contato com responsável por tarefa atrasada
Requisitos não funcionais	Celular que possa efetuar chamadas Aplicativo que envie e-mail Acesso a internet
Frequência	Somente quando houver tarefas atrasadas

4.2 DESIGN

A seguir, a descrição da arquitetura e da interface do aplicativo.

4.2.1 Arquitetura

O aplicativo utiliza a componente Activity da plataforma Android, que representa uma única tela com interface de usuário (<http://developer.android.com/guide/topics/fundamentals/activities.html>). A Activity

DotProject é responsável por gerenciar informações relevantes às outras Activities. As outras Activities estendem DotProject. O aplicativo se conecta com o servidor do dotProject através do protocolo HTTP, que já é suportado por ele, de acordo com os comentários no código fonte do dotProject. Dessa forma é possível autenticar-se perante o servidor e posteriormente requisitar páginas HTML que são processadas e têm seus dados extraídos para que cada Activity receba as informações do servidor e formate os dados para mostrar na tela. Para esse protótipo não foi utilizado o protocolo HTTPS, pois não era mencionado na documentação.

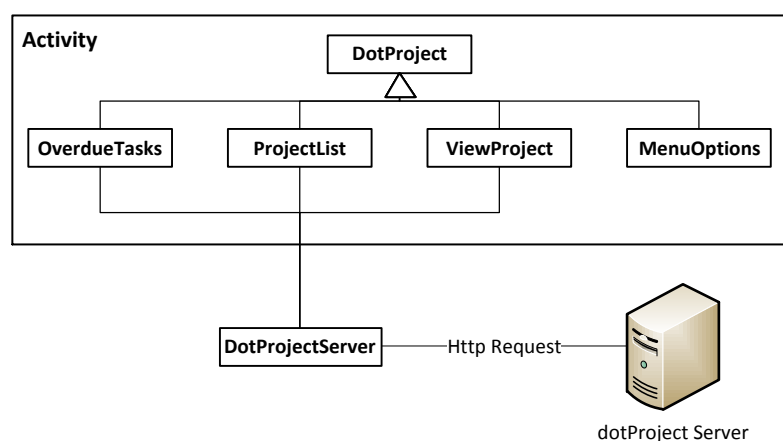


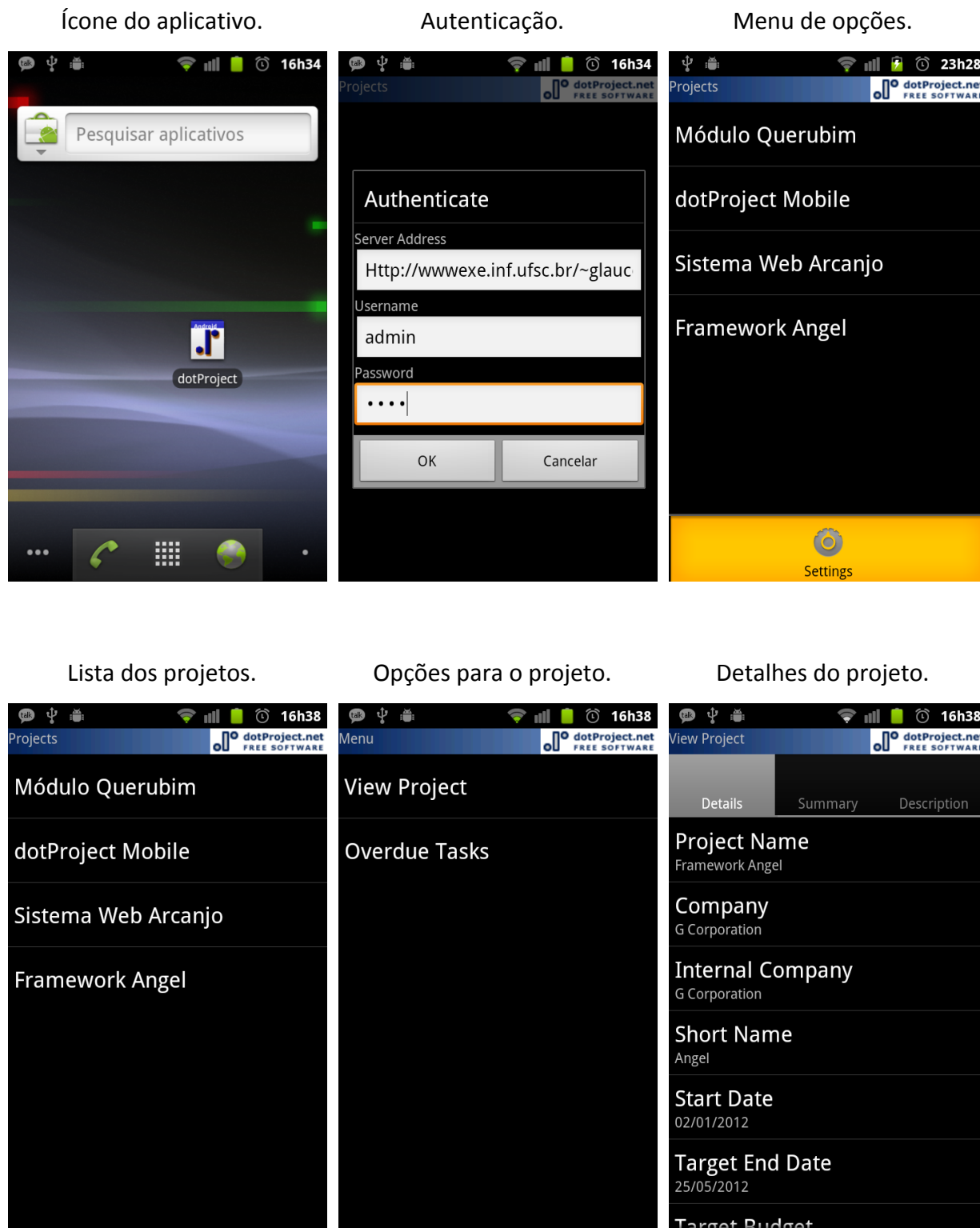
Figura 20: Arquitetura do sistema.

4.2.2 Design de Interface

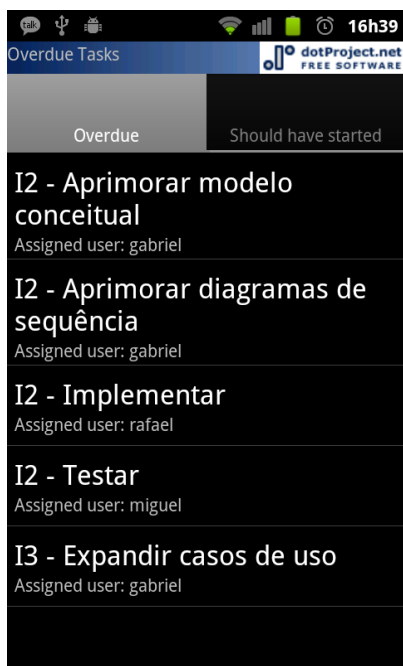
Nessa seção é apresentado o design de interface das telas. A interface foi construída utilizando objetos View e ViewGroup da plataforma Android (<http://developer.android.com/guide/topics/ui/index.html>). As subclasses de View oferecem objetos como botões e campos de textos. As subclasses de ViewGroup oferecem objetos de layout, que são responsáveis pela organização de objeto na tela, por exemplo de forma linear, tabular ou relativa. Os objetos View tem propriedades que definem seu tamanho e posicionamento na tela. A interface de usuário de uma atividade é definida através de uma hierarquia de objetos View e ViewGroup. Para descrever essa hierarquia são utilizados arquivos XML, dessa

forma, oferecendo uma melhor legibilidade. Cada Activity do projeto tem sua interface definida em um arquivo XML.

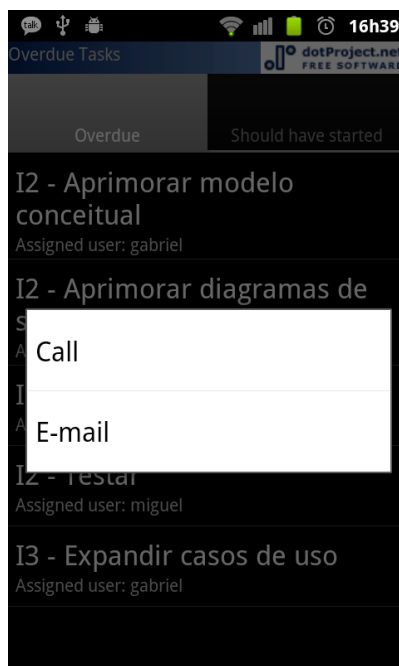
Quadro 9: Telas do aplicativo e descrição.



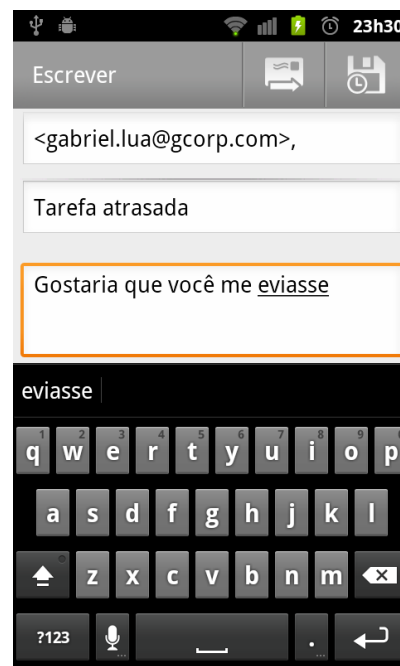
Tarefas atrasadas.



Opções de contato.



E-mail.



4.3 IMPLEMENTAÇÃO

O aplicativo foi desenvolvido utilizando as ferramentas disponíveis no *Software Development Kit* (SDK) da plataforma Android. A utilização dessas ferramentas foram feitas através do *Integrated Development Environment* (IDE) Eclipse utilizando o *plugin Android Development Tools* (ADT). A linguagem utilizada foi o Java.

A versão alvo do desenvolvimento foi a 2.1, pois ainda possui grande fatia do mercado e as versões mais novas têm total suporte a ela. Para depurar o aplicativo foram utilizados emuladores que fazem parte do SDK do Android e um celular Google Nexus S com versões 2.3.1 e 2.3.4 do Android.

No lado do servidor foi instalado o dotProject em um servidor Suse da UFSC.

O código fonte está disponível no Apêndice K.

4.4 TESTES

Os testes foram construídos a partir dos casos de uso definidos na análise de requisitos. O quadro 10 mostra a análise dos casos de uso e o quadro 11 mostra o resultado referente aos requisitos não funcionais.

Quadro 10: Resultado dos testes de fluxo de casos de uso.

Nº	Flux.	Caso de uso	Dados de teste	Pré requisitos	Passos	Resultados esperados	Sit.
1	Prin.	Autenticar usuário	Autenticar no servidor "wwwexe.inf.ufsc.br / ~glaucosn/dotproject/index.php" com usuário "admin" e senha "pass"	Existir projetos "Sistema Web Arcanjo" e "Framework Angel" no servidor	Inserir endereço no campo de texto; inserir usuário no campo de texto; inserir senha no campo de texto; clicar no botão "ok"	Lista com os projetos "Sistema Web Arcanjo" e "Framework Angel"	ok
2	Alt. 1a	Autenticar usuário	Autenticar no servidor com informações já inseridas previamente	Informações do servidor salvas no celular e existir projetos "Sistema Web Arcanjo" e "Framework Angel" no servidor	Abrir o aplicativo	Lista com os projetos "Sistema Web Arcanjo" e "Framework Angel"	ok
3	Alt. 2a	Autenticar usuário	Autenticar no servidor "wwwexe.inf.ufsc.br / ~glaucosn/dotproject/index.php" com usuário "admin" e senha "senhaerrada" e depois com senha "pass"	Usuário "admin" não possuir senha "senhaerrada" e existir projetos "Sistema Web Arcanjo" e "Framework Angel" no servidor	Acessar o menu, selecionar "settings", inserir informações corretas e clicar no botão "ok"	Lista com os projetos "Sistema Web Arcanjo" e "Framework Angel"	ok
4	Alt. 3a	Autenticar usuário	Autenticar no servidor "wwwexe.inf.ufsc.br / ~glaucosn/dotproject/index.php" com usuário "admin" e senha "pass"	Não existir projetos no servidor "wwwexe.inf.ufsc.br / ~glaucosn/dotproject/index.php"	Inserir endereço no campo de texto; inserir usuário no campo de texto; inserir senha no campo de texto; clicar no botão "ok"	Lista sem projetos	ok

5	Prin.	Visualizar projetos	Visualizar informações do projeto "Sistema Web Arcanjo"	Existir projeto "Sistema Web Arcanjo"	Escolher projeto "Sistema Web Arcanjo", escolher opção de visualizar projeto, navegar pelas abas "Details", "Summary", "Description"	Visualizar as informações do projeto "Sistema Web Arcanjo"	ok
6	Prin.	Visualizar tarefas atrasadas	Visualizar tarefa atrasada "I2 - Aprimorar modelo conceitual" do projeto "Sistema Web Arcanjo"	Existir tarefa atrasada "I2 - Aprimorar modelo conceitual" no projeto "Sistema Web Arcanjo"	Escolher projeto "Sistema Web Arcanjo", escolher opção de visualizar tarefas atrasadas, visualizar tarefa "I2 - Aprimorar modelo conceitual"	Visualizar tarefa "I2 - Aprimorar modelo conceitual" e responsável "gabriel"	ok
7	Alt. 5a	Visualizar tarefas atrasadas	Visualizar que não existem tarefas atrasadas no projeto "dotProject Mobile"	Não existir tarefas atrasadas no projeto "dotProject Mobile"	Escolher projeto "dotProject Mobile", escolher opção de visualizar tarefas atrasadas, não aparecer nada	Não exibir nenhuma tarefa	ok
8	Prin.	Entrar em contato com responsável por tarefa atrasada	Entrar em contato com o "gabriel", responsável pela tarefa atrasada "I2 - Aprimorar modelo conceitual" do projeto "Sistema Web Arcanjo"	Existir tarefa atrasada "I2 - Aprimorar modelo conceitual" com responsável "gabriel"	Acessar o menu de contexto da tarefa; Selecionar e-mail; Escrever e-mail; Enviar e-mail	E-mail enviado para o gabriel, responsável pela tarefa "I2 - Aprimorar modelo conceitual"	ok

Quadro 11: Resultado dos testes de requisitos não funcionais.

Nº	Requisito não funcional	Situação
1	Rodar em Android 2.1 ou superior	Testado no emulador para versão 2.1 e no aparelho Nexus S com versão 2.3.1 e 2.3.4
2	Estar conectado a internet	Testado com diferentes conexões wireless
4	Efetuar chamada	Confirmado na avaliação do sistema
5	Enviar e-mail	Confirmado na avaliação do sistema

5 AVALIAÇÃO

A avaliação é a etapa final do desenvolvimento do protótipo realizando um teste de usabilidade envolvendo representantes do público-alvo do gerenciamento de projetos em dispositivos móveis. Como uma das principais preocupações no desenvolvimento deste tipo de aplicativo é a usabilidade (LOOCKWOOD, 2008) e a utilidade de um suporte desse via celular, o foco da avaliação é tanto na utilidade quanto na usabilidade. A avaliação é realizada com base no processo de estudos empíricos proposto pelo (WOHLIN, 2000).

5.1 DEFINIÇÃO DA AVALIAÇÃO

Para a realização da avaliação é necessário definir os objetivos, perguntas e materiais utilizados.

5.1.1 Objetivo da Avaliação

O objetivo é avaliar a utilidade e usabilidade da tarefa “Visualizar informações de um projeto em planejamento” que compreende os casos de uso “Autenticar usuário” e “Visualizar projetos” e da tarefa “Enviar e-mail para responsável por tarefa atrasada” que compreende os casos de uso “Visualizar tarefas atrasadas” e “Entrar em contato com responsável por tarefa atrasada”. Conforme a ISO/IEC 9241 entende-se a usabilidade em termos de eficácia, eficiência e satisfação no contexto da aplicação da avaliação. Nesse contexto, identificaram-se os seguintes requisitos de usabilidade:

Tarefa 1 - Visualizar informações de um projeto em planejamento

- Eficácia: Visualizar informações importantes sobre um projeto em andamento.
- Eficiência: Completar a tarefa em no máximo 4 minutos.
- Satisfação: 90% dos usuários ficarem satisfeitos ou muito satisfeitos.

Tarefa 2 - Enviar e-mail para responsável por tarefa atrasada

- Eficácia: Conseguir enviar um e-mail para o responsável pela tarefa atrasada.
- Eficiência: Completar a tarefa em no máximo 5 minutos.
- Satisfação: 90% dos usuários ficarem satisfeitos ou muito satisfeitos.

5.1.2 Perguntas da Avaliação

Com esta avaliação tem-se como objetivo verificar se os requisitos de usabilidade foram cumpridos, analisando se os usuários tem facilidade de executar as tarefas “Visualizar informações de um projeto em planejamento” e “Enviar e-mail para responsável por tarefa atrasada” e se o sistema é agradável a ponto de motivar seu uso. Ao final, espera-se que as seguintes perguntas possam ser respondidas:

- Na tarefa 1, o participante conseguiu visualizar as informações do projeto?
- Na tarefa 2, o participante conseguiu enviar o e-mail?
- A execução das tarefas é eficiente?
- As expectativas dos usuários em relação às tarefas foram satisfeitas?
- A execução das tarefas se mostra intuitiva e os usuários não têm dificuldade em terminá-la?

5.1.3 Design do Estudo

Para realizar a avaliação é realizado um teste de usabilidade conduzido da seguinte forma: É feita a leitura do Script de Introdução e então é dado o Termo de Consentimento Livre e Esclarecido para o participante ler e assinar. Em seguida é feito um treinamento na plataforma Android. As tarefas são lidas e as dúvidas sanadas. A gravação de áudio e vídeo é iniciada e o participante executa as tarefas. Após o termino das tarefas é entregue a Escala de Usabilidade do Sistema e o

Questionário demográfico para o participante responder. E ao final o participante é questionado sobre o que achou do aplicativo.

5.1.4 Materiais da Avaliação

Os materiais utilizados são os seguintes:

- Termo de consentimento livre e esclarecido (Apêndice A)
- Descrição da tarefa (Apêndice B)
- Script de introdução (Apêndice C)
- Escala de Usabilidade do Sistema (SUS) (Apêndice D)
- Questionário demográfico (Apêndice E)
- Roteiro do pesquisador (Apêndice F)
- Treinamento (Apêndice G)

5.2 EXECUÇÃO DA AVALIAÇÃO

A avaliação foi executada no dia 13 de maio de 2011, entre as 14:00 e 17:30 horas, na sala de reuniões do INCOD - UFSC. Foram utilizadas duas câmeras para capturar o rosto do participante e a tela do aplicativo, Figura 21. Também foi registrado o áudio durante a execução das tarefas. O *smartphone* utilizado para a avaliação foi um Google Nexus S com a versão 2.3.4 do Android. Também foi utilizada a versão 2.1.5 do dotProject, instalado no espaço que o aluno possui nos servidores da UFSC com endereço <http://wwwexe.inf.ufsc.br/~glaucosn/dotproject>. O público-alvo foram pessoas que possuíam algum conhecimento na área de gerenciamento de projetos. A avaliação contou com 7 participantes, selecionados entre alunos do departamento de Informática e Estatística da UFSC. Cada teste durou em média 20:34 minutos.

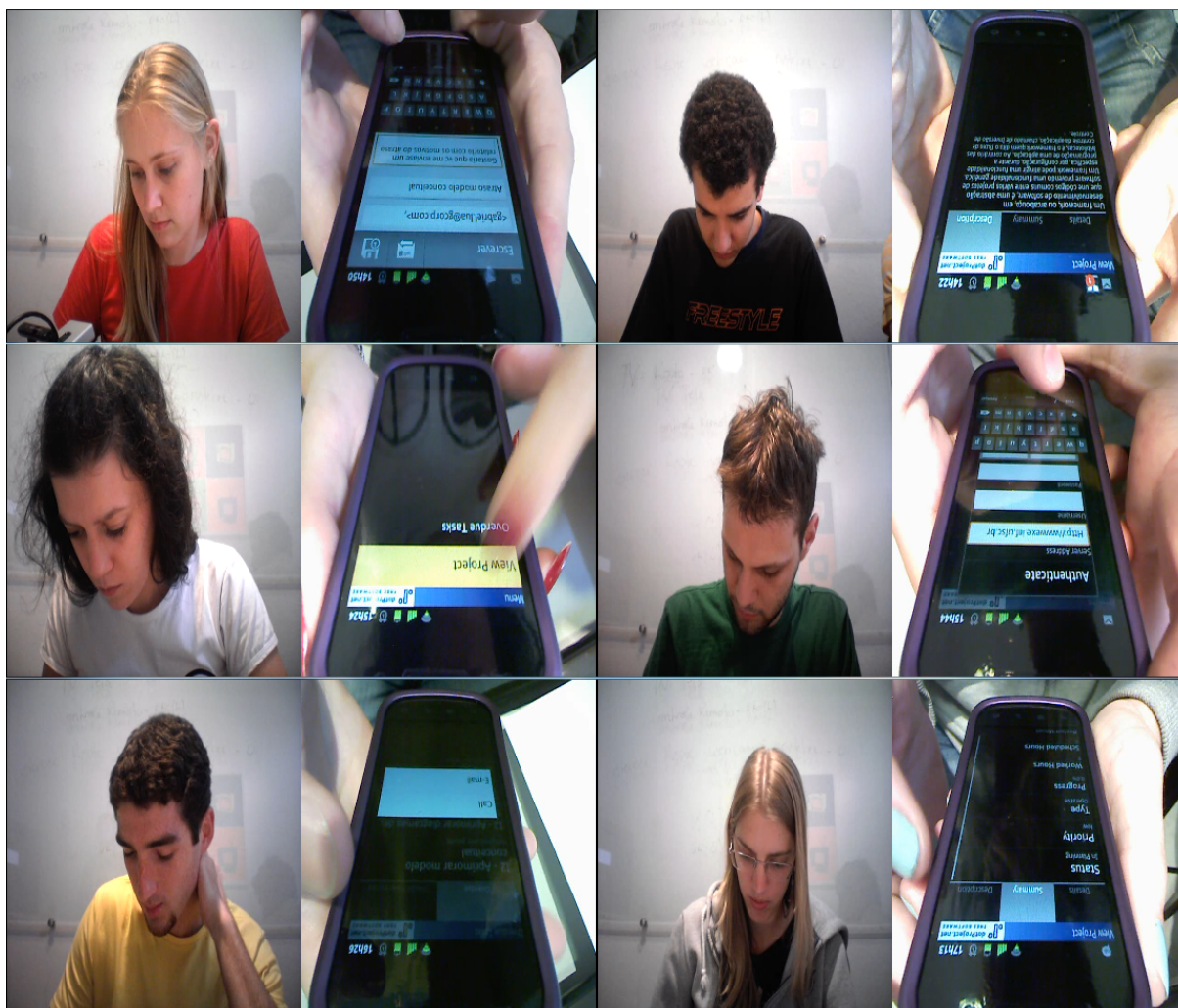


Figura 21: Participantes durante a avaliação.

Os participantes da avaliação utilizam a internet diariamente, possuem celulares e na sua maioria, *smartphone*, conforme Figura 22. Utilizam o celular para diversas funções, como fazer ligações, enviar mensagens de texto, jogar, tirar fotos e entrar na internet, como mostra a Figura 23. Apenas um dos participantes não tinha nenhuma experiência com a plataforma Android. Todos os participantes tinham algum conhecimento em gerenciamento de projetos, e a maioria já havia utilizado o dotProject, conforme Figura 24. Todos os participantes estão na faixa etária de 18 a 25 anos e possuem ensino superior completo ou incompleto. As respostas do Questionário demográfico estão disponíveis no Apêndice I.

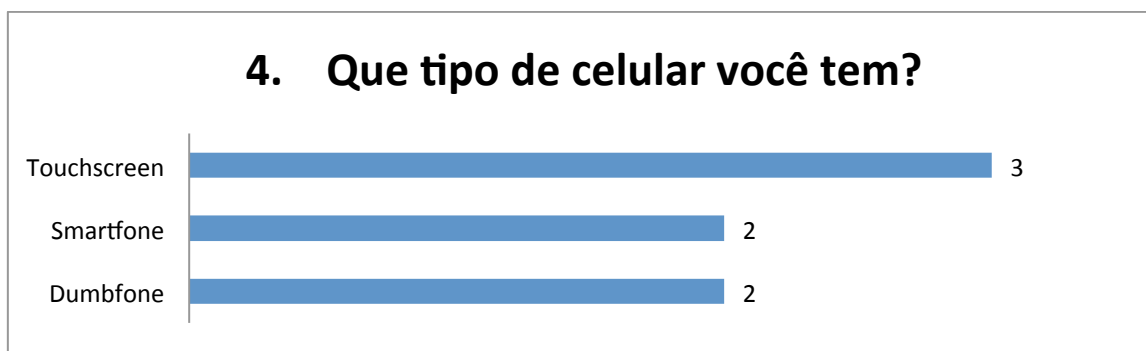


Figura 22: Resultado da pergunta 4 do questionário demográfico.

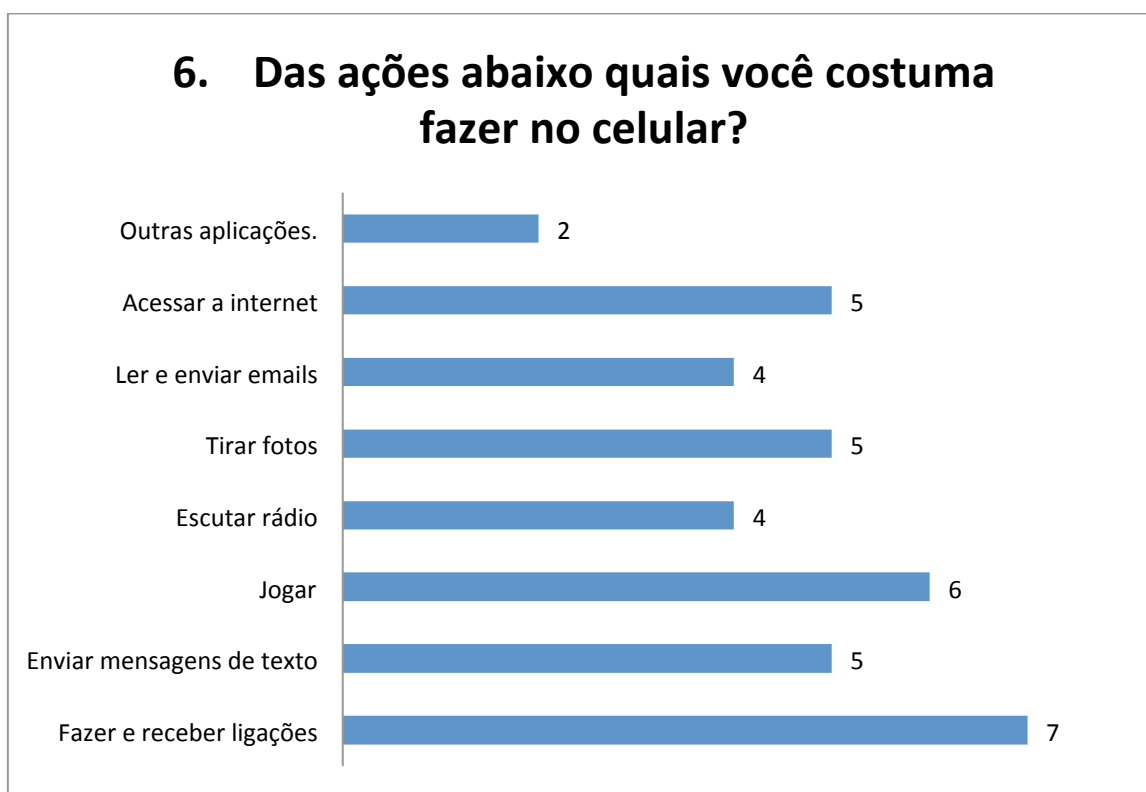


Figura 23: Resultado da pergunta 6 do questionário demográfico.

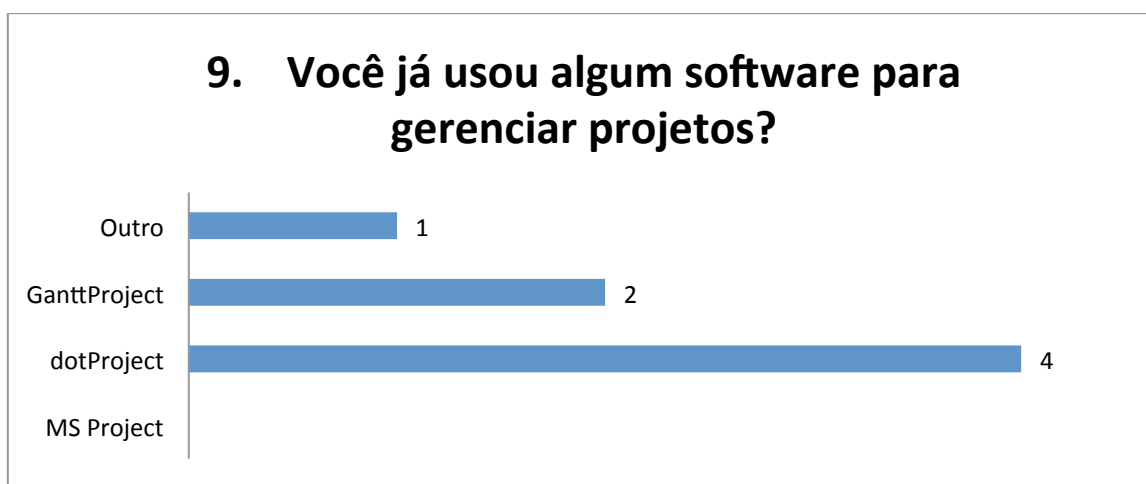


Figura 24: Resultado da pergunta 9 do questionário demográfico.

5.3 RESULTADOS DA AVALIAÇÃO

Os dados coletados foram analisados em relação às perguntas de pesquisa definidas. A análise resultou nas seguintes respostas:

- Na tarefa 1, o participante conseguiu visualizar as informações do projeto?
 - Todos os participantes navegaram entre as informações do projeto e fizeram a leitura da descrição do projeto.
- Na tarefa 2, o participante conseguiu enviar o e-mail?
 - Todos os participantes conseguiram enviar um e-mail para um responsável por alguma tarefa atrasada. No entanto, dois participantes erraram em contatar o responsável pela tarefa solicitada.
- A execução das tarefas é eficiente?
 - Os participantes levaram em média 2:52 minutos para executar a tarefa 1 e em média 2:06 minutos para executar a tarefa 2, conforme mostra a Figura 25. Esse tempo ficou dentro da expectativa de eficiência.

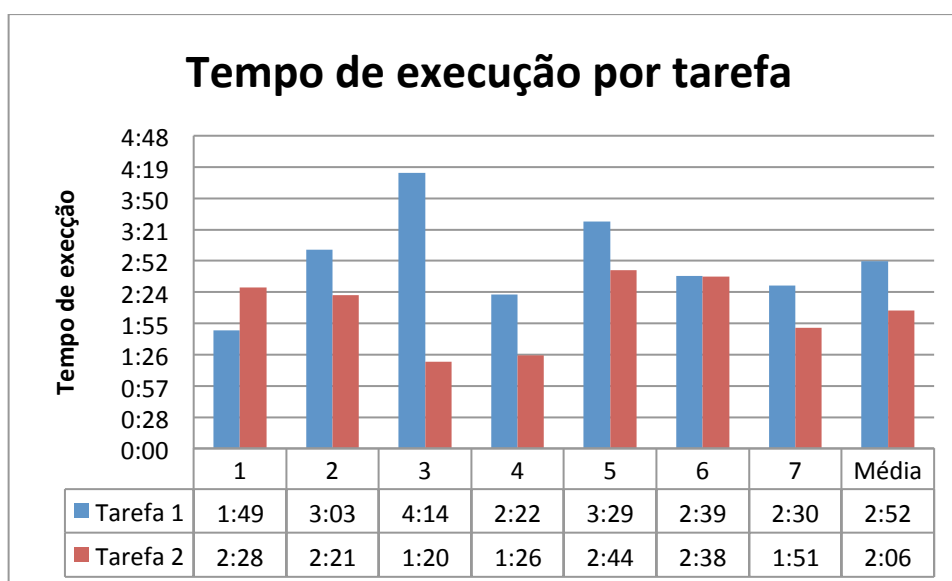


Figura 25: Tempo de execução por tarefa e média.

- As expectativas dos usuários em relação às tarefas foram satisfeitas?

- Com pontuação superior a 77,5 no SUS é possível considerar que o aplicativo atinge de forma satisfatória as expectativas dos usuários.

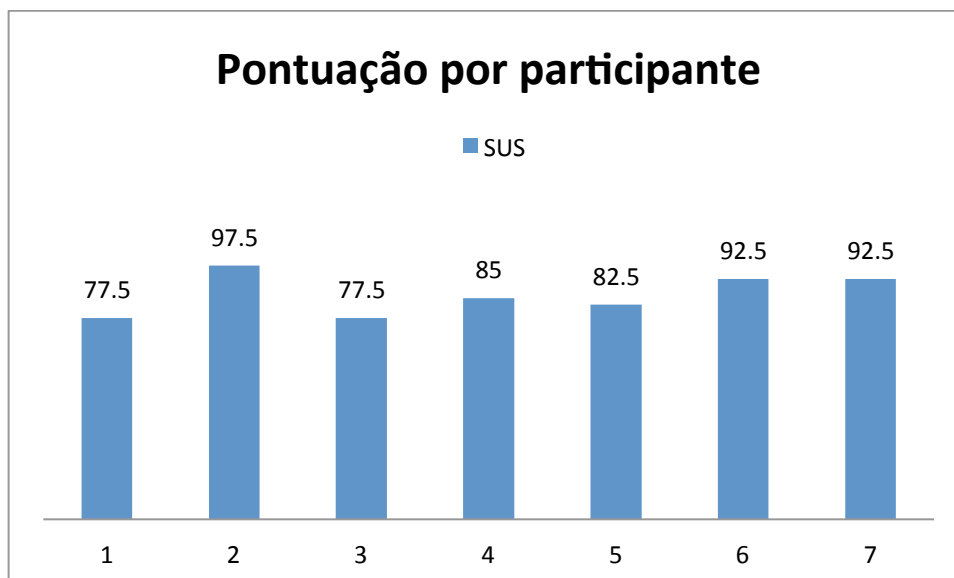


Figura 26: Pontuação do SUS por participante.

- A execução das tarefas se mostra intuitiva e os usuários não tem dificuldade em terminá-la?
 - Na primeira tarefa alguns participantes que erraram na inserção dos dados de autenticação sentiram falta de um aviso pelo erro de conexão, o que é uma falha do sistema em notificar o usuário da situação do sistema. O tempo de resposta também foi citado por alguns participantes, que sentiram uma demora entre o toque na tela e a mudança de estado no software. Na segunda tarefa alguns participantes se sentiram frustrado ao clicar em uma tarefa e não obter mais detalhes sobre a mesma. Mostra o nome da tarefa e o nome do responsável não se mostrou suficiente para atingir as expectativas dos participantes.

Com essa análise ficou claro que o sistema possui algumas falhas de usabilidade, mas no contexto geral possui uma boa usabilidade. Os pontos fracos puderam ser identificados e não são considerados impactantes no desenvolvimento, podendo ser implantados de forma simples e sem comprometer o sistema. As melhorias seriam um sistema de notificações ao usuário, para que ele entenda melhor o estado do sistema. E a adição de detalhamento das tarefas atrasadas, para que o gerente possua mais informações sobre a tarefa.

5.4 AMEAÇAS À VALIDADE

As ameaças à validade do teste podem ocorrer, pois o teste contou com um pequeno número de participantes. Os participantes não cobriam uma grande variação de faixa etária e conhecimento em gerência de projetos. O estudo teria sido mais relevante se houvesse a participação de gerentes de projetos efetivos.

6 DISCUSSÃO

A partir da avaliação realizada obteve-se uma primeira indicação que o aplicativo poderá ser útil e ter boa usabilidade tendo uma boa aceitação entre os participantes do teste de usabilidade. O aplicativo pode ser útil para auxiliar o gerente disponibilizando informações importantes do projeto a qualquer hora e em qualquer lugar. Também auxiliando o gerente a contatar de forma rápida os responsáveis por tarefas atrasadas. Então o grupo de processos que ganha com isso é o de monitoramento e controle e as atividades monitorar e controlar o trabalho do projeto e reportar o desempenho.

Em comparação com os outros aplicativos identificados no estado da arte pode-se analisar o quadro 12. Porém não é possível comparar em termos de aceitação pela falta de informação.

Quadro 12: Comparação das ferramentas identificadas no estado da arte e da ferramenta desenvolvida.

Aplicativo	Timy	Beacon	ConnectWise Mobile	Gtdagenda	Rallydroid	dotProject Mobile
Empresa	Nascom	Binary Duo	ConnectWise	David Allen	Rally	-
Features						
Sincronização de dados	Sim	Sim	Sim	Sim	Não	Sim
Visualização de projetos	Sim	Sim	Sim	Sim	Não	Sim
Edição de projetos	Sim	Sim	Sim	Sim	Não	Não
Visualização de tarefas	Não	Sim	Sim	Sim	Sim	Sim
Edição de tarefas	Não	Sim	Sim	Sim	Sim	Não
Visualização de cronograma	Não	Não	Sim	Não	Não	Não
Edição de cronograma	Não	Não	Sim	Não	Não	Não
Quadro de horários	Sim	Não	Sim	Não	Não	Não
Contatos	Não	Sim	Sim	Não	Não	Sim
Mensagens (para comunicação com a equipe)	Não	Sim	Não	Sim	Não	Sim

7 CONCLUSÃO

O objetivo geral desse trabalho é desenvolver um aplicativo móvel para suportar atividades do monitoramento e controle de projetos a ser integrado ao software dotProject.

Esse trabalho fornece fundamentação teórica sobre: gerenciamento de projetos de acordo com o PMBOK; sobre características de software para gerenciamento de projetos, incluindo as principais funcionalidades; e sobre a área de dispositivos móveis, com foco no sistema Android.

Foi realizado um estudo do estado da arte a fim de analisar as ferramentas para gerenciar projetos a partir de dispositivos móveis. Esse estudo mostrou que há a tendência de levar funcionalidades de sistemas de GP para o dispositivo móvel, porém foram identificados somente 4 aplicativos especificamente para Android, e nenhum deles integrado ao sistema dotProject.

Nesse contexto foi proposta, a extensão do dotProject, por ser um dos SGPs mais utilizados e de código e uso livre. Identificando funcionalidades úteis via celulares. Um protótipo foi construído e avaliado.

A principal contribuição do trabalho é a demonstração da utilidade e usabilidade de suporte de ferramentas de GP via celulares e da extensão de uma ferramenta amplamente utilizada e sob a licença GNU GPL de código livre.

Dessa forma espera-se que o monitoramento e controle do projeto possam ser feitos de forma mais efetiva. Contribuindo para que a taxa de sucesso de projetos aumente.

Para trabalhos futuros sugere-se a correção das falhas de usabilidade, apontadas na avaliação, de não existir mensagens de erro quando não se consegue conectar no servidor, e de não existir uma tela com mais descrições sobre as tarefas atrasadas, especificando a data das tarefas atrasadas. E o desenvolvimento das funcionalidades: sincronização de dados, edição de projetos, edição de tarefas, visualização de cronograma, que são funcionalidades já propostas em outros aplicativos. E uma avaliação em larga escala voltada aos usuários do dotProject envolvendo um maior número de gerentes de projetos de diversos contextos.

REFERÊNCIAS

ANATEL (Brasília - DF) (Org.). **Brasil ultrapassa um celular por habitante**. Disponível em: <

<http://www.anatel.gov.br/Portal/exibirPortalNoticias.do?acao=carregaNoticia&codigo=21613> >. Acesso em: 18 nov. 2010.

BEREAUX, Elle. **The Complete Guide to Project Management for New Managers and Management Assistants: How to Get Things Done in Less Time**. [s. L]: Atlantic Publishing Company, 2008.

BLOKDIJK, Gerard. **Project Management 100 Success Secrets**. [S. L]: Lulu.com, 2007.

CONSTANTINOU, Andreas; CAMILLERI, Elizabeth; KAPETANAKIS, Matos. **Mobile Developer Economics 2010 and Beyond**. Londres: Visionmobile, 2010.

DAVIDSON, Jeff. **10 Minute Guide to Project Management**. Indianapolis: Alpha Books, 2000. 4 v. (Management Series).

DOTPROJECT. **DotProject**. Disponível em: <
<http://sourceforge.net/projects/dotproject> >. Acesso em: 27 nov. 2010.

DOTPROJECT. **What is Included**. Disponível em: <
http://docs.dotproject.net/index.php?title=Main_Page >. Acesso em: 27 nov. 2010.

FIRTMAN, Maximiliano. **Programming the Mobile Web**. Sebastopol: O'reilly Media, 2010.

FITZEK, Frank; REICHART, Frank. **Mobile phone programming: and its Application to Wireless Network**. Dordrecht: Springer, 2007.

GARTNER. **Gartner Says Android to Command Nearly Half of Worldwide Smartphone Operating System Market by Year-End 2012.** Disponível em: < <http://www.gartner.com/it/page.jsp?id=1622614> >. Acesso em: 18 de mai. 2011.

HEERKENS, Gary R. **Project Management.** [S. L.]: Briefcasebooks, 2002.

HORINE, Gregory M. **Absolute Beginner's Guide to Project Management.** 2. ed. Estados Unidos: Que, 2009.

IBCD. **A Revolução Digital leva à Convergência.** Disponível em: < http://www.ibcd.com.br/conv_digital.htm >. Acesso em: 18 de mai. 2011.

JORDAN, Lee. **Project Management with dotProject:** Implement, Configure, Customize, and Maintain your dotProject Installation. Birmingham: Packt Publishing, 2007.

KERZNER, Harold. **Project Management:** A Systems Approach to Planning, Scheduling, and Controlloing. 7. ed. [S. L.]: John Wiley & Sons, Inc, 2001.

KITCHENHAM, Barbara. **Procedures for Performing Systematic Reviews.** Keele, 2004.

KLEIM, Ralph L.; LUDIN, Irwin S. **Project Management Practioner's Handbook.** [s. L.]: Amacom Books, 1998.

LEVINE, Harvey A. **Pratical Project Management:** Tips, Tactics, and Tools. New York: John Wiley & Sons, Inc, 2002.

LOCKWOOD, Arden. **The Project Manager's Perspective on Project Management Software Packages.** Avignon, 2008.

MEHTA, Nirav. **Mobile Web Development.** Birmingham: Packt, 2008.

MIKKONEN, Tommi. **Programming Mobile Devices: An Introduction for Practitioners**. Chichester: Wiley, 2007.

PORTNY, Stanley E. **Project Management for Dummies**. 2. ed. Indianapolis: Wiley, 2007.

PROJECT MANAGEMENT INSTITUTE (Org.). **A Guide to the Project Management Body of Knowledge: PMBOK® Guide**. 4. ed. Newtown Square: Project Management Institute, Inc, 2008.

PROJECT MANAGEMENT INSTITUTE (Org.). **Um Guia do Conjunto de Conhecimentos em Gerenciamento de Projetos: Guia PMBOK®**. 3. ed. Newtown Square: Project Management Institute, Inc, 2004.

ROTHMAN, Johanna. **Manage It!: Your Guide to Modern, Pragmatic Project Management**. Dallas: The Pragmatic Booksheld, 2007.

WOHLIN, C. et all. **Experimentation in Software Engineering: An introduction**. Kluwer Academic Publishers, USA, 2000.

APÊNDICE A – TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

Eu estou sendo convidado(a) a participar do teste de usabilidade de um protótipo de sistema da área de gerenciamento de projetos, desenvolvidos no contexto do Trabalho de Conclusão de Curso intitulado: Gerenciamento de Projetos em Dispositivos Móveis: Uma Evolução ao Dotproject, realizado pelo aluno Glauco Neves, sendo orientado pela Prof.^a. Dra. rer. nat. Christiane A. Gresse von Wangenheim, PMP, cujo objetivo é avaliar e melhorar a usabilidade do protótipo desenvolvido. A minha participação no referido projeto será no sentido de auxiliar na identificação de pontos fortes e fracos no design de interface deste protótipo.

Recebi esclarecimentos sobre a pesquisa e estou ciente de que minha privacidade será respeitada, ou seja, meu nome será mantido em sigilo.

Eu autorizo a gravação de áudio e vídeo durante os testes de usabilidade e entendo que as gravações de áudio e vídeo serão utilizadas somente para os fins desta pesquisa e não serão divulgados fora do contexto desta pesquisa.

Fui informado(a) de que posso me recusar a participar do estudo, ou retirar meu consentimento a qualquer momento, sem precisar justificar.

É assegurada a assistência durante toda a pesquisa.

Manifesto meu livre consentimento em participar.

Florianópolis, 13 de Maio de 2011

Nome e assinatura do participante

Nome e assinatura do pesquisador

APÊNDICE B – DESCRIÇÃO DA TAREFA

CONTEXTO

Você é um gerente de projetos de software de uma empresa. Você está numa viagem de negócios, e não tem tido muito tempo para acessar seu notebook. Nesse momento você está no aeroporto esperando seu voo e você gostaria de acessar o dotProject para alcançar dois objetivos: Ler a descrição de um projeto que ainda está em planejamento para obter informações para tratar com um cliente; e acompanhar o desenvolvimento de outro projeto, vendo se existem tarefas atrasadas, e caso exista, entrar em contato com o responsável pela mesma.

TAREFA 1 – VISUALIZAR INFORMAÇÕES DE UM PROJETO EM PLANEJAMENTO

Utilizando um smartphone Android com o aplicativo do dotProject instalado, se autentique no servidor utilizando as seguintes informações:

- Server address:
<http://wwwexe.inf.ufsc.br/~glaucosn/dotproject/index.php>
- Username: admin
- Password: pass

Caso nenhum projeto apareça na tela, abra o menu de opções e utilize a opção “settings” para verificar se as credenciais de autenticação estão corretas.

Acesse o projeto “Framework Angel” e navegue pelas informações do projeto. Por último leia a descrição do projeto.

TAREFA 2 - ENVIAR E-MAIL PARA RESPONSÁVEL POR TAREFA ATRASADA

Ainda com o aplicativo aberto, navegue de volta aos projetos e acesse o projeto “Sistema Web Arcanjo” navegue pela tarefas atrasadas. Para a tarefa atrasada “I2 – Aprimorar modelo conceitual” acesse o menu de contexto e selecione a opção de e-mail. Envie uma mensagem para o responsável da tarefa como por exemplo:

- Título: Atraso no modelo conceitual
- Mensagem: Gostaria que você me enviasse um relatório sobre os motivos do atraso.

APÊNDICE C – SCRIPT DE INTRODUÇÃO

Oi [nome do participante]. Eu sou Glauco Neves e estou fazendo meu Trabalho de Conclusão do curso de Ciências da Computação na UFSC com o tema Gerenciamento de Projetos em Dispositivos Móveis: Uma Evolução ao Dotproject. O objetivo geral desse projeto é desenvolver um aplicativo móvel para suportar o monitoramento e controle de projetos a ser integrado ao software dotProject. Estou realizando este teste para analisar a eficiência, eficácia e satisfação dos usuários no uso do protótipo do meu sistema.

O objetivo é testar a interface do sistema e não você. Se você tiver dificuldades em usar o sistema, provavelmente outras pessoas também terão. Este teste é simplesmente um meio para avaliar o design da interface e identificar questões que precisam ser melhoradas. Este teste não deverá levar mais do que 30 minutos, mas você não precisa concluí-lo caso sinta-se desconfortável, você pode parar a qualquer momento.

Se sentir-se à vontade, você pode falar em voz alta durante o teste o que me possibilitará entender seu ponto de vista. Ao final farei algumas perguntas em relação à execução da tarefa, a fim de encontrar possíveis pontos fracos em minha proposta.

Você tem alguma dúvida?

Primeiro gostaria que você lesse e confirmasse este termo de consentimento livre e esclarecido.

Podemos começar!

APÊNDICE D – ESCALA DE USABILIDADE DO SISTEMA(SUS)

	Discordo Totalmente				Concordo Totalmente
1. Eu usaria esse sistema frequentemente.					
	1	2	3	4	5
2. Eu achei o sistema desnecessariamente complexo.					
	1	2	3	4	5
3. Eu achei o sistema fácil de usar.					
	1	2	3	4	5
4. Eu precisaria de suporte técnico pessoal para ser hábil em usar o sistema.					
	1	2	3	4	5
5. Eu achei que as funções do sistema estavam bem integradas.					
	1	2	3	4	5
6. Eu achei muitas inconsistências no sistema.					
	1	2	3	4	5
7. Eu acho que a maioria das pessoas irá aprender rapidamente a usar o sistema.					
	1	2	3	4	5
8. Eu achei o sistema muito incômodo de usar.					
	1	2	3	4	5
9. Eu me senti muito confiante usando o sistema.					
	1	2	3	4	5
10. Eu precisei aprender algumas coisas antes de conseguir usar o sistema.					
	1	2	3	4	5

APÊNDICE E – QUESTIONÁRIO DEMOGRÁFICO

INTERNET

1. Você costuma acessar a Internet? (trabalho, em casa, lan house)
 - ☐ Sim
 - ☐ Não
2. Com que frequência você acessa a internet?
 - ☐ Diariamente
 - ☐ Uma vez por semana
 - ☐ Uma vez por mês
 - ☐ Nunca

CELULAR

3. Você possui celular?
 - ☐ Sim (quantos?)
 - ☐ Não
4. Que tipo de celular você tem?
 - ☐ Dumbfone (possui funções básicas de ligações e mensagens, possui apenas teclado numérico)
 - ☐ Smartfone (celular com alguns aplicativos, teclado QWERTY)
 - ☐ Touchscreen (qual a plataforma? (Symbian, IOS, Android))
5. Qual marca? Se não souber a marca, pode deixar em branco.
6. Das ações abaixo quais você costuma fazer no celular? Pode marcar mais de uma.
 - ☐ Fazer e receber ligações
 - ☐ Enviar mensagens de texto
 - ☐ Jogar
 - ☐ Escutar rádio
 - ☐ Tirar fotos
 - ☐ Ler e enviar e-mails
 - ☐ Acessar a internet
 - ☐ Outras aplicações. Quais?
7. Quanta experiência com a plataforma Android você considera ter?
 - ☐ Nenhuma (Nunca havia utilizado)
 - ☐ Pouca (Já tinha visto ou utilizado pouco)
 - ☐ Média (Utilizo de vez em quando)

- ☐ Muita (Utilizo ou já utilizei frequentemente)

GERENCIAMENTO DE PROJETOS

8. Quanta experiência com gerenciamento de projetos você considera ter?

- ☐ Nenhuma
- ☐ Pouca (Apenas o que aprendi em aulas ou sendo gerenciado)
- ☐ Média (Aplico de vez em quando)
- ☐ Muita (Aplico ou já apliquei frequentemente)

9. Você já usou algum software para gerenciar projetos?

- ☐ MS Project
- ☐ dotProject
- ☐ GanttProject
- ☐ Outro – Qual:

DEMOGRÁFICAS

10. Sexo

- ☐ Masculino
- ☐ Feminino

11. Faixa etária

- ☐ 18-25 anos
- ☐ 26-45 anos
- ☐ 46-60 anos
- ☐ Mais de 61

12. Escolaridade

- ☐ Ensino fundamental incompleto
- ☐ Ensino fundamental completo
- ☐ Ensino médio incompleto
- ☐ Ensino médio completo
- ☐ Ensino superior incompleto
- ☐ Ensino superior completo
- ☐ Possui pós-graduação, mestrado, doutorado...

13. Possui alguma deficiência ou dificuldade:

- ☐ Motora
- ☐ Visual
- ☐ Auditiva

APÊNDICE F – ROTEIRO DO PESQUISADOR

Data: _____ N° do Participante: _____

Tempo total: De _____ à _____

Tempo tarefa 1: De _____ à _____ - Tempo tarefa 2: De _____ à _____

Sucesso tarefa 1: ☐ Parcial ☐ Total - Sucesso tarefa 2: ☐ Parcial ☐ Total

COMPORTAMENTO VERBAL	TAREFA 1	TAREFA 2
☐ Comentário fortemente positivo		
☐ Outro comentário positivo		
☐ Comentário fortemente negativo		
☐ Outro comentário negativo		
☐ Sugestão		
☐ Pergunta		
☐ Variação da expectativa		
☐ Confuso		
☐ Frustrado		

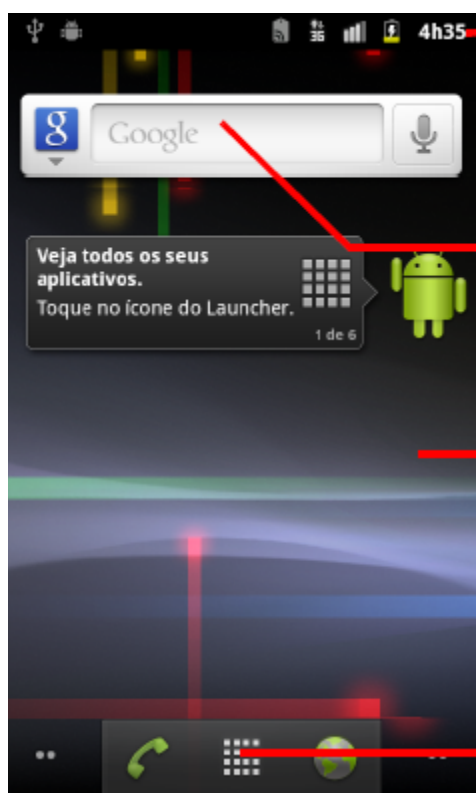
COMPORTAMENTO VERBAL	NÃO-	TAREFA 1	TAREFA 2
☐ Careta/Infeliz			
☐ Sorrindo/Rindo/Feliz			
☐ Surpreso/Inesperado			
☐ Concentrado			
☐ Impaciente			
☐ Variação de expectativa			
☐ Inclinado próximo à tela			
☐ Se remexendo na cadeira			
☐ Suspirando			
☐ Massageando cabeça/pescoço			

ERROS COMETIDOS

Erro 1:		☐ Simples	☐ Moderado
		☐ Grave	☐ Gravíssimo
Erro 2:		☐ Simples	☐ Moderado
		☐ Grave	☐ Gravíssimo
Erro 3:		☐ Simples	☐ Moderado
		☐ Grave	☐ Gravíssimo

APÊNDICE G – TREINAMENTO

PÁGINA INICIAL



A Barra de status mostra o horário, a potência do sinal, o status da bateria e outras informações. Ela também exibe ícones de notificação.

Widgets são aplicativos que podem ser utilizados diretamente na tela "Página inicial".

Toque nos itens na tela "Página inicial" para abri-los. Toque e mantenha pressionado um espaço vazio para adicionar um atalho para um aplicativo, um widget etc.

Toque no ícone do Iniciador para abrir o Iniciador e visualizar todos seus aplicativos.

TELA DE TOQUE

Tocar

Para realizar ações em itens na tela, como ícones de aplicativos e de configurações, para digitar letras e símbolos usando o teclado virtual ou para pressionar botões virtuais, basta tocá-los com seu dedo.

Tocar e manter pressionado

Toque e mantenha pressionado um item na tela. Toque nele e não levante seu dedo até ocorrer uma ação. Por exemplo, para abrir um menu de personalização da tela "Página inicial", basta tocar em uma área vazia na tela "Página inicial" até o menu abrir.

BOTÕES DO CELULAR

Botão	Pressionar	Pressionar e manter pressionado
Voltar ↶	Abre a tela anterior na qual você estava trabalhando. Se o teclado virtual estiver aberto, fecha o teclado.	
Menu ≡	Abre um menu com itens que afetam a tela ou o aplicativo atual.	
Página inicial ↗	Abre a tela "Página inicial". Se você estiver visualizando a tela "Página inicial" estendida para a esquerda ou para a direita, abre a tela central da "Página inicial".	Abre a tela com os últimos aplicativos utilizados.
Pesquisar 🔍	Na tela "Página inicial", abre a Pesquisa do Google para pesquisar no seu telefone e na web. Em muitos aplicativos, abre uma caixa de pesquisa para pesquisar dentro do aplicativo.	Abre a Pesquisa do Google por voz.
Botão liga/desliga	Desativa a tela.	Abre um menu com opções para o Modo para avião, para o Modo silencioso e para desligar o telefone.

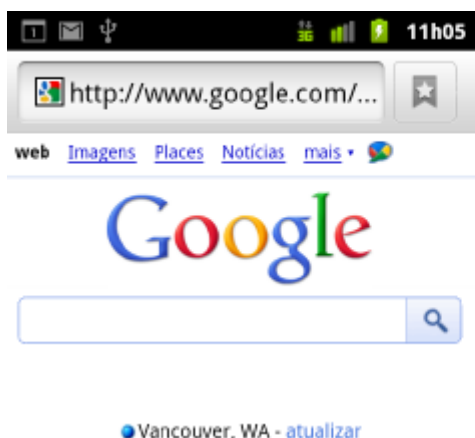
MENUS

Há dois tipos de menus no Android: menus de opções e menus de contexto.

Menus de opções

Os menus de opções contêm ferramentas que se aplicam às atividades da tela ou aplicativo atual, não a qualquer item específico na tela. Abra os menus de opções pressionando o botão Menu ≡. Nem todas as telas possuem menus de opções. Se você pressionar Menu ≡ em uma tela que não possui menu de opções, nada será exibido.

Em algumas telas, o número de itens do menu de opções é maior que suportado no menu principal. Toque em Mais para abrir um menu de itens adicionais.

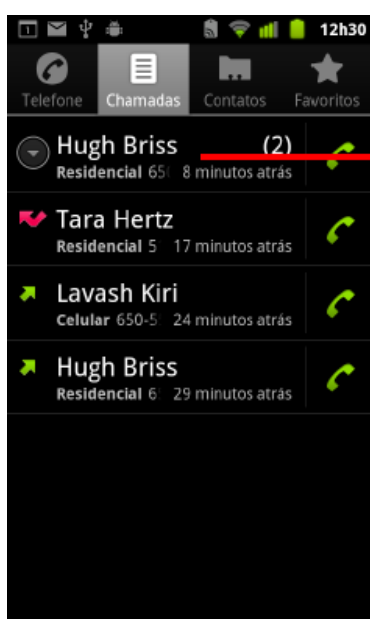


Os menus de opções contêm itens associados à tela atual ou ao aplicativo atual como um todo.

Toque para abrir mais itens do menu.

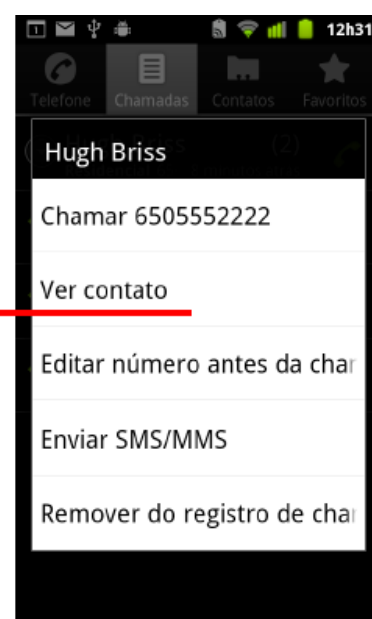
Menus de contexto

Os menus de contexto contêm ferramentas que se aplicam a um item específico na tela. Toque e mantenha pressionado um item na tela para abrir um menu de contexto. Nem todos os itens possuem menus de contexto. Se você tocar e mantiver pressionado um item que não possui menu de contexto, não acontecerá nada.



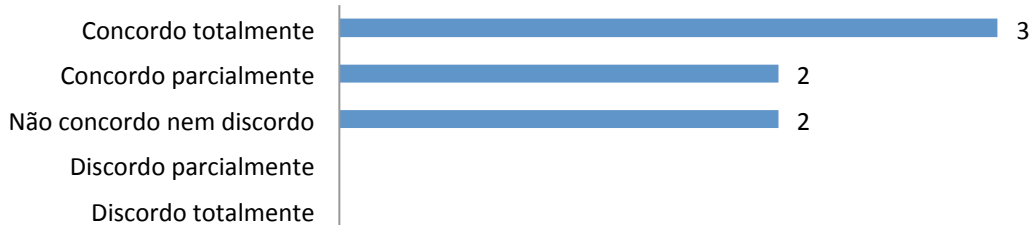
Ao tocar e manter pressionados alguns itens de uma tela...

... um menu de contexto é exibido.



APÊNDICE H – RESULTADO DO SUS

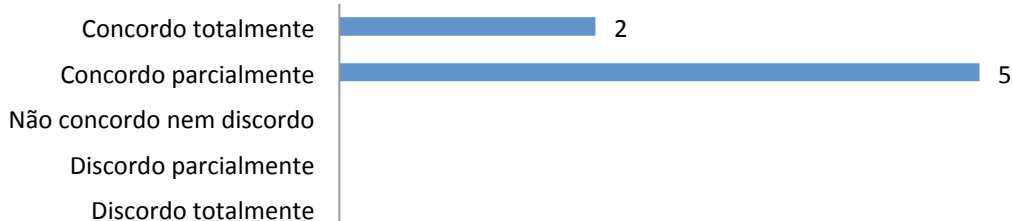
1. Eu usaria esse sistema frequentemente.



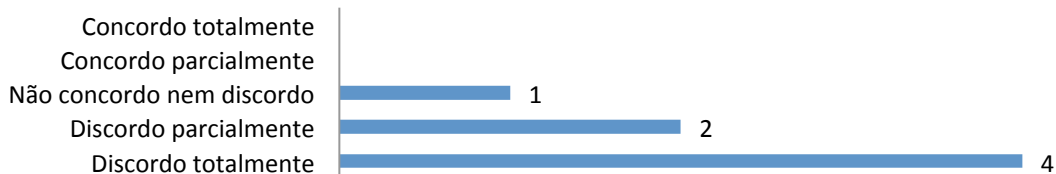
2. Eu achei o sistema desnecessariamente complexo.



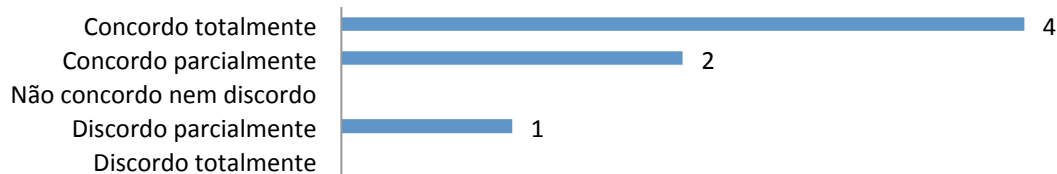
3. Eu achei o sistema fácil de usar.



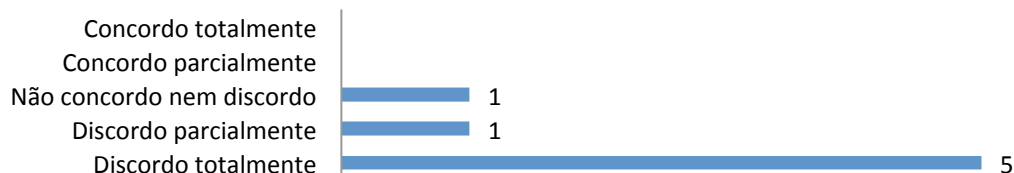
4. Eu precisaria de suporte técnico pessoal para ser hábil em usar o sistema.



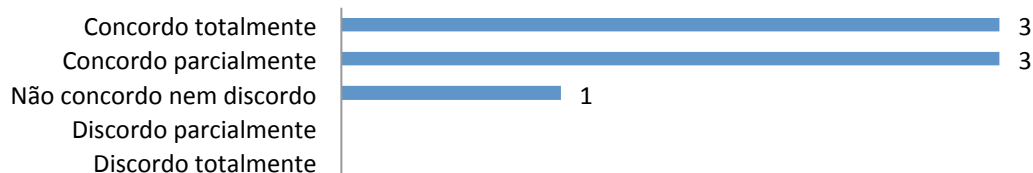
5. Eu achei que as funções do sistema estavam bem integradas.



6. Eu achei muitas inconsistências no sistema.



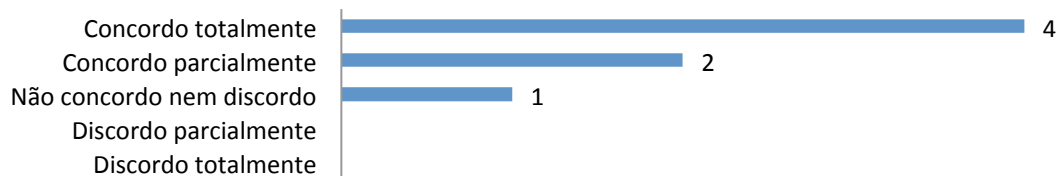
7. Eu acho que a maioria das pessoas irá aprender rapidamente a usar o sistema.



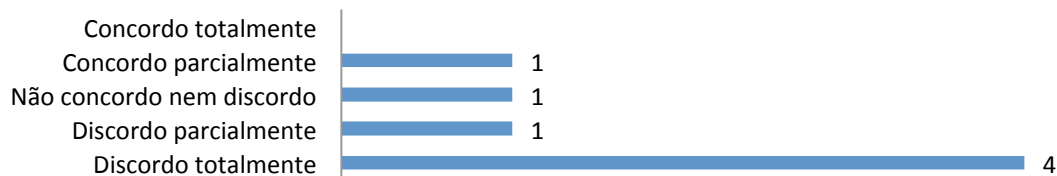
8. Eu achei o sistema muito incômodo de usar.



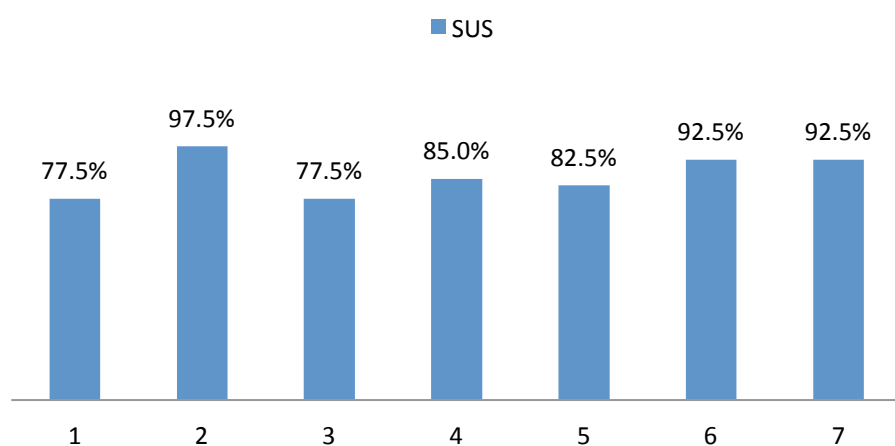
9. Eu me senti muito confiante usando o sistema.



10. Eu precisei aprender algumas coisas antes de conseguir usar o sistema.

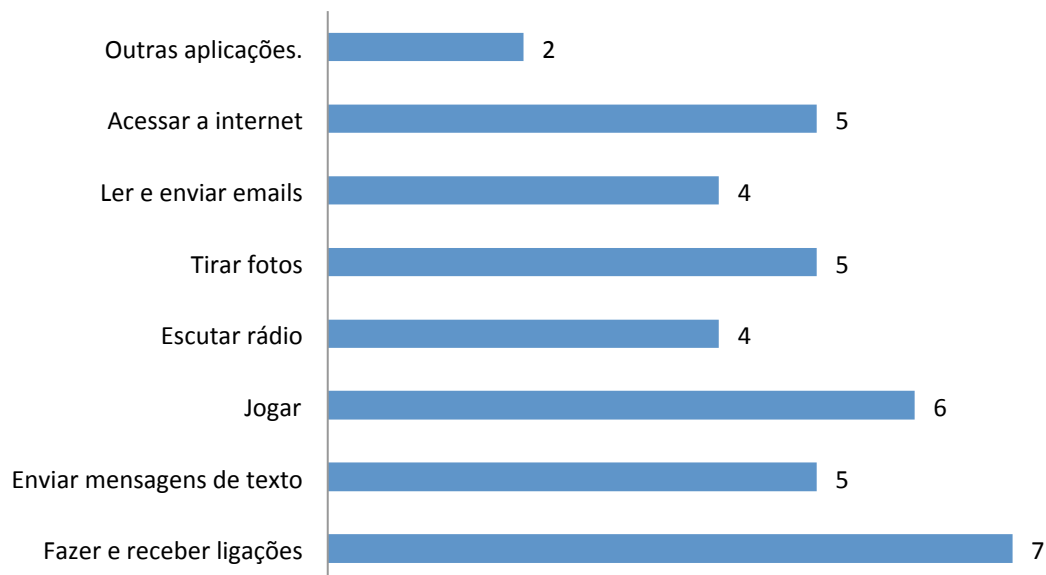


Pontuação por participante

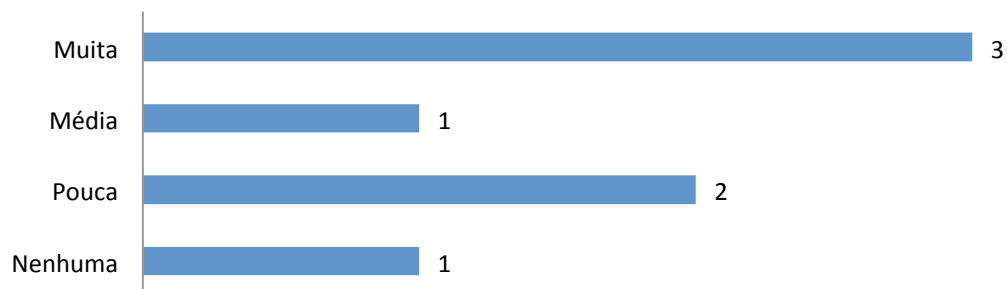


APÊNDICE I – RESULTADO DO QUESTIONÁRIO DEMOGRÁFICO**1. Você costuma acessar a Internet?****2. Com que frequência você acessa a internet?****3. Você possui celular?****4. Que tipo de celular você tem?**

6. Das ações abaixo quais você costuma fazer no celular?



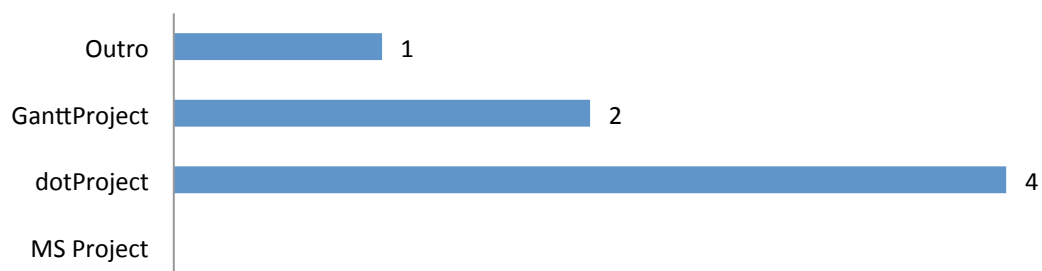
7. Quanta experiência com a plataforma Android você considera ter?



8. Quanta experiência com gerenciamento de projetos você considera ter?



9. Você já usou algum software para gerenciar projetos?



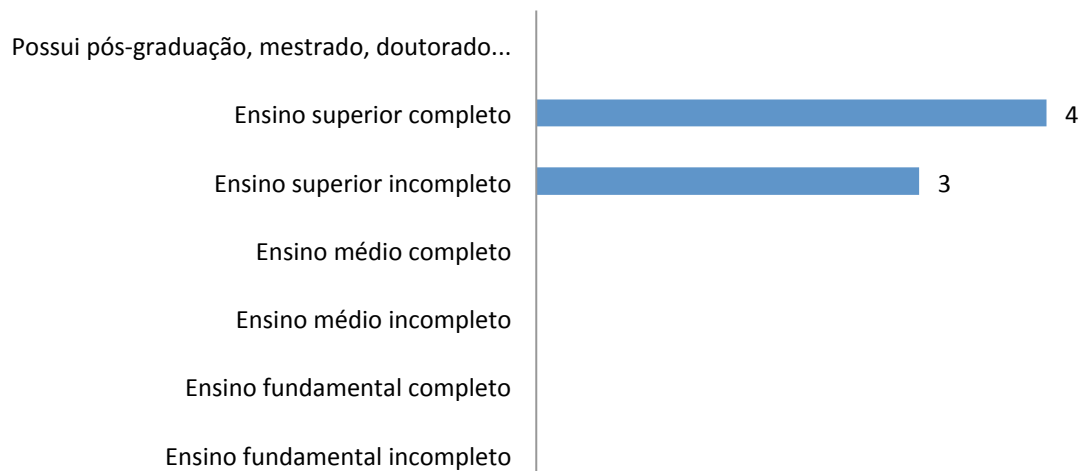
10. Sexo



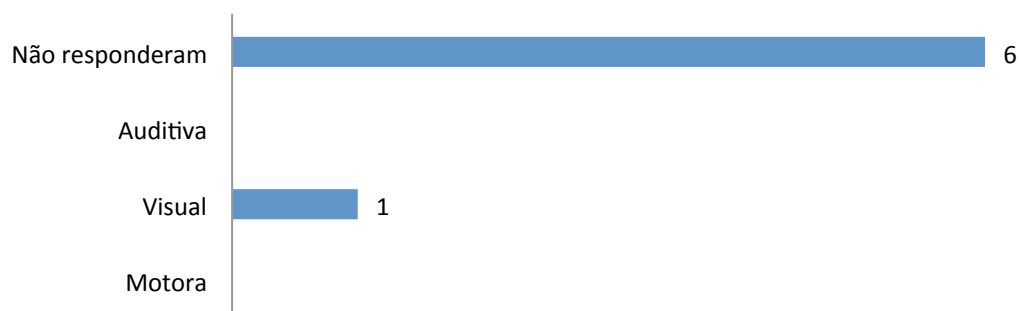
11. Faixa etária



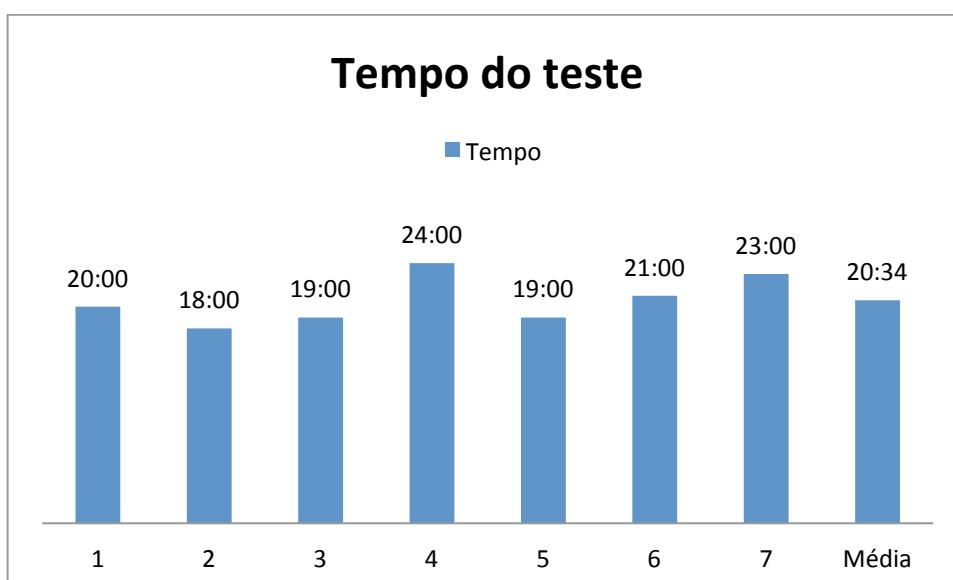
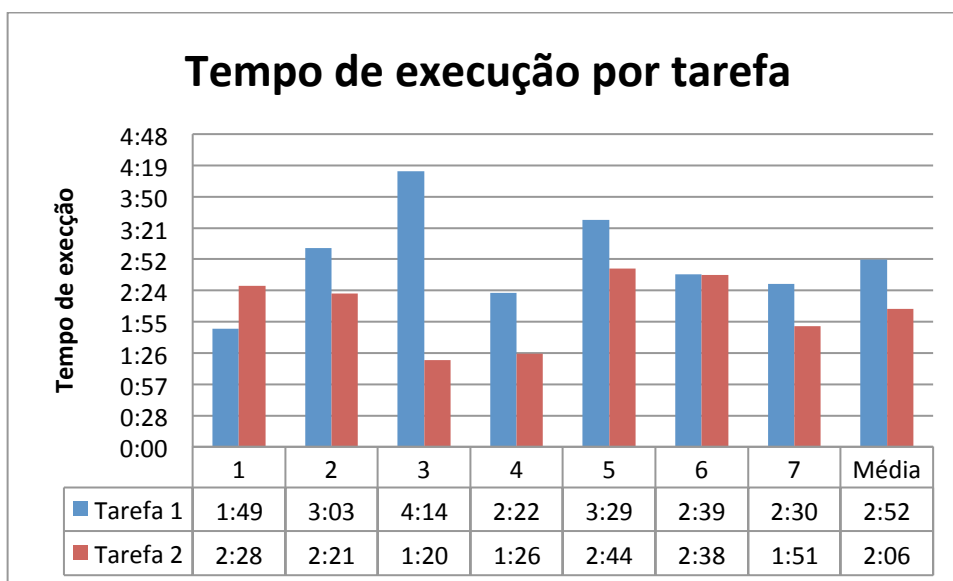
12. Escolaridade



13. Possui alguma deficiência ou dificuldade:



APÊNDICE J – TEMPOS DA AVALIAÇÃO



APÊNDICE K – CÓDIGO FONTE

Pacote br.ufsc.dotproject.activity

Classe DotProject.java

```
package br.ufsc.dotproject.activity;

import br.ufsc.dotproject.service.DotProjectServer;
import android.app.Activity;

public class DotProject extends Activity {

    public static DotProjectServer mDPInterface;

    public static final String DP_PREFERENCES = "Preferences";

    public static final String DP_PREFERENCES_SERVER_ADDRESS = "Nada";
    public static final String DP_PREFERENCES_USERNAME = "Username";
    public static final String DP_PREFERENCES_PASSWORD = "Password";

    public static final String DP_PREFERENCES_PROJECT_ID = "";

}
```

Classe MenuOptions.java

```
package br.ufsc.dotproject.activity;

import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;

public class MenuOptions extends DotProject {

    SharedPreferences mPreferences;

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_menu);
        mPreferences = getSharedPreferences(DP_PREFERENCES,
Context.MODE_PRIVATE);
        final int id = mPreferences.getInt(DP_PREFERENCES_PROJECT_ID, -
1);

        mDPInterface.setProjectPage(id);
        final String[] mOptions = { "View Project", "Overdue Tasks" };
        final ListView mListViewProjects = (ListView)
findViewById(R.id.listview_menu);
        mListViewProjects.setAdapter(new ArrayAdapter<String>(this,
android.R.layout.simple_list_item_1, mOptions));
        mListViewProjects.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
```



```

        mTabSpecShouldHaveStarted.setIndicator("Should have started");

        mTabSpecShouldHaveStarted.setContent(R.id.listview_should_have_starte
d_tasks);
        mTabHost.addTab(mTabSpecShouldHaveStarted);

        mTabHost.setCurrentTabByTag("overdue");

        initOverdueTaksList();
        initShouldHaveStartedTaksList();
    }

    void initOverdueTaksList() {
        int id = mPreferences.getInt(DP_PREFERENCES_PROJECT_ID, -1);
        final ArrayList<HashMap<String, String>> list =
mDPInterface.requestOverdueTasksList(id);
        ListView mListViewProject = (ListView)
findViewById(R.id.listview_overdue_tasks);
        SimpleAdapter notes = new SimpleAdapter(this, list,
android.R.layout.simple_list_item_2, new String[] {
            "line1", "line2" }, new int[] { android.R.id.text1,
android.R.id.text2 });
        mListViewProject.setAdapter(notes);
        registerForContextMenu(mListViewProject);
        mListViewProject.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
            public boolean onItemClick(AdapterView<?> parent,
View itemClicked, int position, long id) {
                final HashMap<String, String> mMap =
(HashMap<String, String>) list.get(position);
                final int user_id =
Integer.valueOf(mMap.get("line3"));
                mDPInterface.setUserPage(user_id);
                phone = mDPInterface.requestPhone();
                email = mDPInterface.requestEmail();
                return false;
            }
        });
    }

    void initShouldHaveStartedTaksList() {
        int id = mPreferences.getInt(DP_PREFERENCES_PROJECT_ID, -1);
        final ArrayList<HashMap<String, String>> list =
mDPInterface.requestShouldHaveStartedTasksList(id);
        ListView mListViewProject = (ListView)
findViewById(R.id.listview_should_have_started_tasks);
        SimpleAdapter notes = new SimpleAdapter(this, list,
android.R.layout.simple_list_item_2, new String[] {
            "line1", "line2" }, new int[] { android.R.id.text1,
android.R.id.text2 });
        mListViewProject.setAdapter(notes);
        registerForContextMenu(mListViewProject);
        mListViewProject.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
            public boolean onItemClick(AdapterView<?> parent,
View itemClicked, int position, long id) {
                final HashMap<String, String> mMap =
(HashMap<String, String>) list.get(position);
                final int user_id =
Integer.valueOf(mMap.get("line3"));
                mDPInterface.setUserPage(user_id);
            }
        });
    }

```

```

        phone = mDPInterface.requestPhone();
        email = mDPInterface.requestEmail();
        return true;
    }
    });
}

@Override
public void onCreateContextMenu(ContextMenu menu, View v,
ContextMenuItem menuInfo) {
    super.onCreateContextMenu(menu, v, menuInfo);
    getMenuInflater().inflate(R.menu.inflater_tasks_context, menu);
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.item_call:
            Uri mTelephone = Uri.parse("tel:" + phone);//99165832");
            Intent call = new Intent(Intent.ACTION_CALL, mTelephone);
            startActivity(call);
            return true;
        case R.id.item_email:
            Uri mEmail = Uri.parse(email);//"mailto:glaucosn@gmail.com");
            Intent email = new Intent(Intent.ACTION_SENDTO, mEmail);
            startActivity(email);

            // Intent it = new Intent(Intent.ACTION_SEND);
            // it.putExtra(Intent.EXTRA_EMAIL, email);
            // Log.e("Task", email);
            // it.putExtra(Intent.EXTRA_SUBJECT, task);
            // it.putExtra(Intent.EXTRA_TEXT, "The email body text");
            // it.setType("text/plain");
            // startActivity(Intent.createChooser(it, "Choose Email
            Client"));

            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}
}

```

Classe ProjectsList.java

```

package br.ufsc.dotproject.activity;

import java.util.ArrayList;
import java.util.HashMap;

import android.app.AlertDialog;
import android.app.Dialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.SharedPreferences.Editor;
import android.os.Bundle;

```

```

import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.SimpleAdapter;
import br.ufsc.dotproject.service.DotProjectServer;

public class ProjectsList extends DotProject {

    SharedPreferences mPreferences;
    static final int AUTHENTICATE_DIALOG_ID = 0;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_projects_list);

        mPreferences = getSharedPreferences(DP_PREFERENCES,
Context.MODE_PRIVATE);
        if (!mPreferences.contains(DP_PREFERENCES_SERVER_ADDRESS)) {
            showDialog(AUTHENTICATE_DIALOG_ID);
        } else {
            init();
        }
    }

    void init() {
        String server =
mPreferences.getString(DP_PREFERENCES_SERVER_ADDRESS, null);
        String username =
mPreferences.getString(DP_PREFERENCES_USERNAME, null);
        String password =
mPreferences.getString(DP_PREFERENCES_PASSWORD, null);
        mDPInterface = new DotProjectServer(server, username,
password);
        mDPInterface.login();
        initProjectsList();
    }

    void initProjectsList() {
        final ArrayList<HashMap<String, String>> mProjectsList =
mDPInterface.requestProjectsList();
        final ListView mListViewProjects = (ListView)
findViewById(R.id.listview_projects_list);
        final SimpleAdapter mSimpleAdapter = new SimpleAdapter(this,
mProjectsList,
            android.R.layout.simple_list_item_1, new String[] {
"line1", "line2" }, new int[] { android.R.id.text1,
                                android.R.id.text2 });

        mListViewProjects.setAdapter(mSimpleAdapter);
        mListViewProjects.setChoiceMode(ListView.CHOICE_MODE_SINGLE);
        mListViewProjects.setTextFilterEnabled(true);
        mListViewProjects.setOnItemClickListener(new
AdapterView.OnItemClickListener() {

```

```

        public void onItemClick(AdapterView<?> parent, View
itemClicked, int position, long id) {
            final Editor mEditor = mPreferences.edit();
            final HashMap<String, String> mMap =
(HashMap<String, String>) mProjectsList.get(position);
            final int project_id =
Integer.valueOf(mMap.get("line2"));
            mEditor.putInt(DP_PREFERENCES_PROJECT_ID,
project_id);
            mEditor.commit();
            startActivity(new Intent(ProjectsList.this,
MenuOptions.class));
        }
    });
}

/*
 * (non-Javadoc) Handling Authenticate Dialog
 */
@Override
protected Dialog onCreateDialog(int id) {
    switch (id) {
        case AUTHENTICATE_DIALOG_ID:
            final LayoutInflater mInflater = (LayoutInflater)
getSystemService(Context.LAYOUT_INFLATER_SERVICE);
            final ViewGroup mViewGroup = (ViewGroup)
findViewById(R.id.dialog_authenticate);
            final View mLayout =
mInflater.inflate(R.layout.dialog_authenticate, mViewGroup);
            final AlertDialog.Builder mBuilder = new
AlertDialog.Builder(this);
            mBuilder.setView(mLayout);
            mBuilder.setTitle(R.string.title_authenticate_dialog);

            final EditText mServerAddress = (EditText) mLayout
.findViewById(R.id.edittext_dialog_authenticate_server_address);
            final EditText mUsername = (EditText)
mLayout.findViewById(R.id.edittext_dialog_authenticate_username);
            final EditText mPassword = (EditText)
mLayout.findViewById(R.id.edittext_dialog_authenticate_password);

            mServerAddress.setText(mPreferences.getString(DP_PREFERENCES_SERVER_A
DDRESS, "Http://"));

            mUsername.setText(mPreferences.getString(DP_PREFERENCES_USERNAME,
""));

            mPassword.setText(mPreferences.getString(DP_PREFERENCES_PASSWORD,
""));

            mBuilder.setPositiveButton(android.R.string.ok, new
DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int
which) {
                    final Editor mEditor = mPreferences.edit();

                    mEditor.putString(DP_PREFERENCES_SERVER_ADDRESS,
mServerAddress.getText().toString());

```

```

        mEditor.putString(DP_PREFERENCES_USERNAME,
mUsername.getText().toString());
        mEditor.putString(DP_PREFERENCES_PASSWORD,
mPassword.getText().toString());
        mEditor.commit();

        ProjectsList.this.dismissDialog(AUTHENTICATE_DIALOG_ID);
        init();
    }
    });

        mBuilder.setNegativeButton(android.R.string.cancel,    new
DialogInterface.OnClickListener() {
        public void  onClick(DialogInterface  dialog,  int
whichButton) {

            ProjectsList.this.dismissDialog(AUTHENTICATE_DIALOG_ID);
        }
    });

        final        AlertDialog        mAuthenticateDialog        =
mBuilder.create();
        return mAuthenticateDialog;
    }
    return super.onCreateDialog(id);
}

/*
 * (non-Javadoc) Handling Options Menu
 */
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    super.onCreateOptionsMenu(menu);
    getMenuInflater().inflate(R.menu.inflater_settings, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    super.onOptionsItemSelected(item);
    showDialog(AUTHENTICATE_DIALOG_ID);
    return true;
}
}

```

Classe ViewProject.java

```

package br.ufsc.dotproject.activity;

import java.util.ArrayList;
import java.util.HashMap;

import android.content.Context;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.widget.ListView;
import android.widget.SimpleAdapter;
import android.widget.TabHost;
import android.widget.TabHost.TabSpec;
import android.widget.TextView;

```

```

public class ViewProject extends DotProject {

    SharedPreferences mPreferences;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_view_project);
        mPreferences = getSharedPreferences(DP_PREFERENCES,
Context.MODE_PRIVATE);

        TabHost mTabHost = (TabHost)
findViewById(R.id.tabhost_view_project);
        mTabHost.setup();

        TabSpec mTabSpecDetails = mTabHost.newTabSpec("details");
        mTabSpecDetails.setIndicator("Details");
        mTabSpecDetails.setContent(R.id.listview_details);
        mTabHost.addTab(mTabSpecDetails);

        TabSpec mTabSpecSummary = mTabHost.newTabSpec("summary");
        mTabSpecSummary.setIndicator("Summary");
        mTabSpecSummary.setContent(R.id.listview_summary);
        mTabHost.addTab(mTabSpecSummary);

        TabSpec mTabSpecDescription = mTabHost.newTabSpec("description");
        mTabSpecDescription.setIndicator("Description");
        mTabSpecDescription.setContent(R.id.textview_description);
        mTabHost.addTab(mTabSpecDescription);

        mTabHost.setCurrentTabByTag("details");

        initProjectDetails();
        initProjectSummary();
        initProjectDescription();
    }

    void initProjectDetails() {
        int id = mPreferences.getInt(DP_PREFERENCES_PROJECT_ID, -1);
        ArrayList<HashMap<String, String>> list =
mDPInterface.requestProjectDetails(id);
        ListView mListViewProject = (ListView)
findViewById(R.id.listview_details);
        SimpleAdapter notes = new SimpleAdapter(this, list,
android.R.layout.simple_list_item_2, new String[] {
            "line1", "line2" }, new int[] { android.R.id.text1,
android.R.id.text2 });
        mListViewProject.setAdapter(notes);
    }

    void initProjectSummary() {
        int id = mPreferences.getInt(DP_PREFERENCES_PROJECT_ID, -1);
        ArrayList<HashMap<String, String>> list =
mDPInterface.requestProjectSummary(id);
        ListView mListViewProject = (ListView)
findViewById(R.id.listview_summary);
        SimpleAdapter notes = new SimpleAdapter(this, list,
android.R.layout.simple_list_item_2, new String[] {

```

```

        "line1", "line2" }, new int[] { android.R.id.text1,
        android.R.id.text2 });
        mListViewProject.setAdapter(notes);
    }

    void initProjectDescription() {
        int id = mPreferences.getInt(DP_PREFERENCES_PROJECT_ID, -1);
        String description =
        mDPInterface.requestProjectDescription(id);
        TextView mTextViewDescription = (TextView)
        findViewById(R.id.textview_description);
        mTextViewDescription.setText(description);
    }
}

```

Pacote br.ufsc.dotproject.service

Classe DotProjectServer.java

```

package br.ufsc.dotproject.service;

import java.io.IOException;
import java.io.UnsupportedEncodingException;
import java.util.ArrayList;
import java.util.HashMap;

import org.apache.http.HttpEntity;
import org.apache.http.NameValuePair;
import org.apache.http.client.HttpClient;
import org.apache.http.client.ResponseHandler;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.impl.client.BasicResponseHandler;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.message.BasicNameValuePair;

import android.util.Log;

public class DotProjectServer {

    int nextIndex;
    String server;
    String username;
    String password;
    String projectPage;
    String userPage;
    final HttpClient mHttpClient = new DefaultHttpClient();
    final ResponseHandler<String> mResponseHandler = new
    BasicResponseHandler();

    public DotProjectServer(String server, String username, String
    password) {
        this.server = server;
        this.username = username;
        this.password = password;
    }

    public void login() {

```

```

        final      ArrayList<NameValuePair>      mParams      =      new
ArrayList<NameValuePair>();
        mParams.add(new BasicNameValuePair("login", "login"));
        mParams.add(new BasicNameValuePair("username", username));
        mParams.add(new BasicNameValuePair("password", password));
        HttpEntity mEntity = null;
        try {
            mEntity = new UrlEncodedFormEntity(mParams);
        } catch (final UnsupportedEncodingException e) {
        }
        final HttpPost mPost = new HttpPost(server);
        mPost.addHeader(mEntity.getContentType());
        mPost.setEntity(mEntity);
        try {
            mHttpClient.execute(mPost);
        } catch (final IOException e) {
        }
    }

    public ArrayList<HashMap<String, String>> requestProjectsList() {
        final HttpGet mGet = new HttpGet(server + "?m=projects");
        String teste = "";
        try {
            teste = mHttpClient.execute(mGet, mResponseHandler);
        } catch (final IOException e) {
        }
        ArrayList<HashMap<String, String>> list = new
ArrayList<HashMap<String, String>>();
        int a, b, c;
        String mAuxiliar = "<a
href=\"?m=projects&a=view&project_id=";
        a = teste.indexOf(mAuxiliar);
        while (a != -1) {
            HashMap<String, String> item = new HashMap<String,
String>();
            String id;
            if (Character.isDigit(teste.charAt(a + mAuxiliar.length()
+ 1))) {
                id = teste.substring(1, 2);
            } else {
                id = Character.toString(teste.charAt(a +
mAuxiliar.length()));
            }
            b = teste.indexOf(">", a) + 1;
            c = teste.indexOf("</a>", b);
            a = teste.indexOf(mAuxiliar, c);
            item.put("line1", teste.substring(b, c).trim());
            item.put("line2", id);
            list.add(item);
        }
        return list;
    }

    public ArrayList<HashMap<String, String>> requestProjectDetails(int
project_id) {
        ArrayList<HashMap<String, String>> list = new
ArrayList<HashMap<String, String>>();
        String mAuxiliar = "<span style=\"color:#000000; font-
weight:bold\">";
        int length = mAuxiliar.length();
        int a = projectPage.indexOf(mAuxiliar);

```

```

int b = projectPage.indexOf("</span>", a + length);
mAuxiliar = projectPage.substring(a + length, b).trim();
HashMap<String, String> item = new HashMap<String, String>();
item.put("line1", "Project Name");
item.put("line2", mAuxiliar);
list.add(item);

mAuxiliar = "

```

```

a = projectPage.indexOf(mAuxiliar, b);
b = projectPage.indexOf("<span >", a + length);
if (b != -1) {
    int c = projectPage.indexOf("</span>", b);
    mAuxiliar = projectPage.substring(b + 7, c).trim();
    item = new HashMap<String, String>();
    item.put("line1", "Actual End Date");
    item.put("line2", mAuxiliar);
    list.add(item);
} else {
    b = a;
}

mAuxiliar = "Target Budget:</td>";
length = mAuxiliar.length();
a = projectPage.indexOf(mAuxiliar, b);
b = projectPage.indexOf("</td>", a + length);
mAuxiliar = projectPage.substring(a + length + 23, b).trim();
item = new HashMap<String, String>();
item.put("line1", "Target Budget");
item.put("line2", mAuxiliar);
list.add(item);

mAuxiliar = "Project Owner:</td>";
length = mAuxiliar.length();
a = projectPage.indexOf(mAuxiliar, b);
b = projectPage.indexOf("</td>", a + length);
mAuxiliar = projectPage.substring(a + length + 23, b).trim();
item = new HashMap<String, String>();
item.put("line1", "Project Owner");
item.put("line2", mAuxiliar);
list.add(item);

mAuxiliar = "URL:</td>";
length = mAuxiliar.length();
a = projectPage.indexOf(mAuxiliar, b);
int aux1 = projectPage.indexOf("<a href=", a);
int aux2 = projectPage.indexOf(">", aux1);
b = projectPage.indexOf("</A>", aux2);
mAuxiliar = projectPage.substring(aux2 + 1, b).trim();
item = new HashMap<String, String>();
item.put("line1", "URL");
item.put("line2", mAuxiliar);
list.add(item);

nextIndex = b;

return list;
}

public ArrayList<HashMap<String, String>> requestProjectSummary(int
project_id) {
    ArrayList<HashMap<String, String>> list = new
ArrayList<HashMap<String, String>>();

    String mAuxiliar = "Status:</td>";
    int length = mAuxiliar.length();
    int a = projectPage.indexOf(mAuxiliar, nextIndex);
    int b = projectPage.indexOf("</td>", a + length);
    mAuxiliar = projectPage.substring(a + length + 36, b).trim();
    HashMap<String, String> item = new HashMap<String, String>();

```

```

item.put("line1", "Status");
item.put("line2", mAuxiliar);
list.add(item);

mAuxiliar = "Priority:</td>";
length = mAuxiliar.length();
a = projectPage.indexOf(mAuxiliar, b);
int aux1 = projectPage.indexOf(">", a + length);
b = projectPage.indexOf("</td>", aux1);
mAuxiliar = projectPage.substring(aux1 + 1, b).trim();
item = new HashMap<String, String>();
item.put("line1", "Priority");
item.put("line2", mAuxiliar);
list.add(item);
Log.e("LOG", mAuxiliar);

mAuxiliar = "Type:</td>";
length = mAuxiliar.length();
a = projectPage.indexOf(mAuxiliar, b);
b = projectPage.indexOf("</td>", a + length);
mAuxiliar = projectPage.substring(a + length + 36, b).trim();
item = new HashMap<String, String>();
item.put("line1", "Type");
item.put("line2", mAuxiliar);
list.add(item);

mAuxiliar = "Progress:</td>";
length = mAuxiliar.length();
a = projectPage.indexOf(mAuxiliar, b);
b = projectPage.indexOf("</td>", a + length);
mAuxiliar = projectPage.substring(a + length + 36, b).trim();
item = new HashMap<String, String>();
item.put("line1", "Progress");
item.put("line2", mAuxiliar);
list.add(item);

mAuxiliar = "Worked Hours:</td>";
length = mAuxiliar.length();
a = projectPage.indexOf(mAuxiliar, b);
b = projectPage.indexOf("</td>", a + length);
mAuxiliar = projectPage.substring(a + length + 36, b).trim();
item = new HashMap<String, String>();
item.put("line1", "Worked Hours");
item.put("line2", mAuxiliar);
list.add(item);

mAuxiliar = "Scheduled Hours:</td>";
length = mAuxiliar.length();
a = projectPage.indexOf(mAuxiliar, b);
b = projectPage.indexOf("</td>", a + length);
mAuxiliar = projectPage.substring(a + length + 36, b).trim();
item = new HashMap<String, String>();
item.put("line1", "Scheduled Hours");
item.put("line2", mAuxiliar);
list.add(item);

mAuxiliar = "Project Hours:</td>";
length = mAuxiliar.length();
a = projectPage.indexOf(mAuxiliar, b);
b = projectPage.indexOf("</td>", a + length);
mAuxiliar = projectPage.substring(a + length + 36, b).trim();

```

```

        item = new HashMap<String, String>();
        item.put("line1", "Project Hours");
        item.put("line2", mAuxiliar);
        list.add(item);

        return list;
    }

    public String requestProjectDescription(int project_id) {
        String mAuxiliar = "<strong>Description</strong>";
        int length = mAuxiliar.length();
        int a = projectPage.indexOf(mAuxiliar, nextIndex);
        int b = projectPage.indexOf("<td class=\"hilite\">", a +
length);
        int c = projectPage.indexOf("</td>", b);
        return projectPage.substring(b + 19, c).trim();
    }

    public ArrayList<HashMap<String, String>> requestOverdueTasksList(int
project_id) {
        ArrayList<HashMap<String, String>> list = new
ArrayList<HashMap<String, String>>();
        HashMap<String, String> item;// = new HashMap<String,
String>();

        String mAuxiliar = "<a
href=\"./index.php?m=tasks&a=view&task_id=";
        int a = projectPage.indexOf(mAuxiliar, nextIndex), b, c, d, e,
f, g;
        while (a != -1) {
            b = projectPage.indexOf("color:#", a);
            String comp = projectPage.substring(b + 7, b + 13);
            if (comp.equals("cc6666")) {
                c = projectPage.indexOf(">", a) + 1;
                d = projectPage.indexOf("</a>", c);
                e = projectPage.indexOf("<a
href=\"?m=admin&a=viewuser&user_id=", d);
                f = projectPage.indexOf(">", e) + 1;
                g = projectPage.indexOf("</a>", f);
                item = new HashMap<String, String>();
                item.put("line1", projectPage.substring(c,
d).trim());
                item.put("line2", "Assigned user: " +
projectPage.substring(f, g).trim());
                item.put("line3", projectPage.substring(e + 45, f -
2).trim());
                list.add(item);
            }
            a = projectPage.indexOf(mAuxiliar, b);
        }
        return list;
    }

    public ArrayList<HashMap<String, String>>
requestShouldHaveStartedTasksList(int project_id) {
        ArrayList<HashMap<String, String>> list = new
ArrayList<HashMap<String, String>>();
        HashMap<String, String> item;// = new HashMap<String,
String>();

```

```

        String mAuxiliar = "<a href=\"./index.php?m=tasks&a=view&task_id=";
        int a = projectPage.indexOf(mAuxiliar, nextIndex), b, c, d, e, f, g;
        while (a != -1) {
            b = projectPage.indexOf("color:#", a);
            String comp = projectPage.substring(b + 7, b + 13);
            if (comp.equals("ffeebb")) {
                c = projectPage.indexOf(">", a) + 1;
                d = projectPage.indexOf("</a>", c);
                e = projectPage.indexOf("<a href=\"?m=admin&a=viewuser&user_id=", d);
                f = projectPage.indexOf(">", e) + 1;
                g = projectPage.indexOf("</a>", f);
                item = new HashMap<String, String>();
                item.put("line1", projectPage.substring(c, d).trim());
                item.put("line2", "Assigned user: " + projectPage.substring(f, g).trim());
                item.put("line3", projectPage.substring(e + 45, f - 2).trim());
                list.add(item);
            }
            a = projectPage.indexOf(mAuxiliar, b);
        }
        return list;
    }

    public String requestPhone(){
        String mAuxiliar = "<td align=\"right\" nowrap>Phone:</td>";
        int length = mAuxiliar.length();
        int a = userPage.indexOf(mAuxiliar);
        int b = userPage.indexOf("<td class=\"hilite\" width=\"100%\">", a + length);
        int c = userPage.indexOf("</td>", b);
        return userPage.substring(b + 32, c).trim();
    }

    public String requestEmail(){
        String mAuxiliar = "Email:</td>";
        int length = mAuxiliar.length();
        int a = userPage.indexOf(mAuxiliar);
        int b = userPage.indexOf("<a href=\"", a + length);
        int c = userPage.indexOf(">", b);
        return userPage.substring(b + 9, c).trim();
    }

    public void setProjectPage(int project_id) {
        final HttpGet mGet = new HttpGet(server + "?m=projects&a=view&project_id=" + project_id);
        try {
            projectPage = mHttpClient.execute(mGet, mResponseHandler);
        } catch (final IOException e) {
        }
    }

    public void setUserPage(int user_id){
        final HttpGet mGet = new HttpGet(server + "?m=admin&a=viewuser&user_id=" + user_id);
        try {

```

```

        userPage = mHttpClient.execute(mGet, mResponseHandler);
    } catch (final IOException e) {
    }
}
}

```

Outros recursos

activity_menu.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <RelativeLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
        <ImageView
            android:scaleType="matrix"
            android:src="@drawable/titlegrad"
            android:layout_alignParentLeft="true"
            android:layout_alignParentTop="true"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
        <TextView
            android:text="@string/title_menu"
            android:layout_alignParentLeft="true"
            android:layout_alignParentTop="true"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
        <ImageView
            android:src="@drawable/dp_icon"
            android:layout_alignParentRight="true"
            android:layout_alignParentTop="true"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
    </RelativeLayout>
    <ListView
        android:id="@+id/listview_menu"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true" />
</LinearLayout>

```

activity_overdue_tasks.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <RelativeLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">

```

```

<ImageView
    android:scaleType="matrix"
    android:src="@drawable/titlegrad"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
<TextView
    android:text="@string/title_overdue_tasks"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
<ImageView
    android:src="@drawable/dp_icon"
    android:layout_alignParentRight="true"
    android:layout_alignParentTop="true"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
</RelativeLayout>
<TabHost
    android:id="@+id/tabhost_overdue_tasks"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <LinearLayout
        android:orientation="vertical"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent">
        <TabWidget
            android:id="@android:id/tabs"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content" />
        <FrameLayout
            android:id="@android:id/tabcontent"
            android:layout_width="fill_parent"
            android:layout_height="fill_parent">
            <ListView
                android:id="@+id/listview_overdue_tasks"
                android:layout_width="fill_parent"
                android:layout_height="fill_parent" />
            <ListView
                android:id="@+id/listview_should_have_started_tasks"
                android:layout_width="fill_parent"
                android:layout_height="fill_parent" />
        </FrameLayout>
    </LinearLayout>
</TabHost>
</LinearLayout>

```

activity_projects_list.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <RelativeLayout
        android:layout_width="wrap_content"

```

```

        android:layout_height="wrap_content">
        <ImageView
            android:scaleType="matrix"
            android:src="@drawable/titlegrad"
            android:layout_alignParentLeft="true"
            android:layout_alignParentTop="true"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
        <TextView
            android:text="@string/title_projects_list"
            android:layout_alignParentLeft="true"
            android:layout_alignParentTop="true"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
        <ImageView
            android:src="@drawable/dp_icon"
            android:layout_alignParentRight="true"
            android:layout_alignParentTop="true"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
    </RelativeLayout>
    <ListView
        android:id="@+id/listview_projects_list"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true" />
</LinearLayout>

```

activity_tabs.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <TextView
        android:text="test" />
</LinearLayout>

```

activity_view_project.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <RelativeLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
        <ImageView
            android:scaleType="matrix"
            android:src="@drawable/titlegrad"
            android:layout_alignParentLeft="true"
            android:layout_alignParentTop="true"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
        <TextView

```

```

        android:text="@string/title_view_project"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <ImageView
        android:src="@drawable/dp_icon"
        android:layout_alignParentRight="true"
        android:layout_alignParentTop="true"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</RelativeLayout>
<TabHost
    android:id="@+id/tabhost_view_project"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <LinearLayout
        android:orientation="vertical"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent">
        <TabWidget
            android:id="@android:id/tabs"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content" />
        <FrameLayout
            android:id="@android:id/tabcontent"
            android:layout_width="fill_parent"
            android:layout_height="fill_parent">
            <ListView
                android:id="@+id/listview_details"
                android:layout_width="fill_parent"
                android:layout_height="fill_parent" />
            <ListView
                android:id="@+id/listview_summary"
                android:layout_width="fill_parent"
                android:layout_height="fill_parent" />
            <TextView
                android:id="@+id/textview_description"
                android:layout_width="fill_parent"
                android:layout_height="fill_parent" />
        </FrameLayout>
    </LinearLayout>
</TabHost>
</LinearLayout>

```

dialog_authenticate.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/dialog_authenticate"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <TextView
        android:text="@string/dialog_authenticate_textview_server_address"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent" />
    <EditText

```

```

        android:text="http://"
        android:id="@+id/edittext_dialog_authenticate_server_address"
        android:maxLines="1"
        android:inputType="textUri"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent" />
<TextView
    android:text="@string/dialog_authenticate_textview_username"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" />
<EditText
    android:id="@+id/edittext_dialog_authenticate_username"
    android:maxLines="1"
    android:inputType="text"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" />
<TextView
    android:text="@string/dialog_authenticate_textview_password"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" />
<EditText
    android:id="@+id/edittext_dialog_authenticate_password"
    android:maxLines="1"
    android:inputType="textPassword"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" />
</LinearLayout>

```

strings.xml

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string
        name="app_name">dotProject</string>
    <string
        name="title_projects_list">Projects</string>
    <string
        name="title_authenticate_dialog">Authenticate</string>
    <string
        name="dialog_authenticate_textview_server_address">Server
Address</string>
    <string
        name="dialog_authenticate_textview_username">Username</string>
    <string
        name="dialog_authenticate_textview_password">Password</string>
    <string
        name="title_item_settings">Settings</string>
    <string
        name="title_menu">Menu</string>
    <string
        name="menu_option_view_project">View Project</string>
    <string
        name="menu_option_overdue_tasks">Overdue Tasks</string>
    <string
        name="title_view_project">View Project</string>
    <string
        name="title_overdue_tasks">Overdue Tasks</string>
    <string
        name="item_call">Call</string>
    <string

```

```

        name="item_email">E-mail</string>
</resources>

```

AndroidManifest.xml

```

<?xml version="1.0" encoding="utf-8"?>
<manifest
    xmlns:android="http://schemas.android.com/apk/res/android"
    package="br.ufsc.dotproject.activity"
    android:versionCode="1"
    android:versionName="1.0">
    <uses-sdk
        android:minSdkVersion="7" />
    <uses-permission
        android:name="android.permission.INTERNET"></uses-permission>
    <uses-permission
        android:name="android.permission.CALL_PHONE"></uses-permission>
    <application
        android:icon="@drawable/icon"
        android:label="@string/app_name"
        android:debuggable="true"
        android:theme="@android:style/Theme.NoTitleBar">
        <activity
            android:name="ProjectsList"
            android:label="@string/app_name"
            android:screenOrientation="portrait">
            <intent-filter>
                <action
                    android:name="android.intent.action.MAIN" />
                <category
                    android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name="DotProject"></activity>
        <activity
            android:name="MenuOptions"
            android:screenOrientation="portrait"></activity>
        <activity
            android:name="ViewProject"
            android:screenOrientation="portrait"></activity>
        <activity android:name="OverdueTasks"
            android:screenOrientation="portrait"></activity>
    </application>
</manifest>

```