

# Aplicativos Móveis para Consulta de Informações de Gerência Ágil de Processos utilizando LPS

Bruno Fortes dos Santos, Patrícia Vilain

Departamento de Informática e Estatística  
Universidade Federal de Santa Catarina (UFSC) - Florianópolis, SC - Brasil

bruno.fortes@inf.ufsc.br, vilain@inf.ufsc.br

**Abstract.** *To render the monitoring of projects and their iterations more accessible to customers and agile teams members, this paper proposes the development of three Android applications that access the information generated by web tools for agile project management. Due to the existence of common parts and points of variation among web tools, this work developed a software product line as a mechanism for the development of the proposed applications. This work describes how this software product line was developed.*

**Resumo.** *Para tornar o acompanhamento de projetos e suas iterações mais acessíveis a clientes e membros de equipes ágeis, esse trabalho propõe o desenvolvimento de três aplicativos para Android que acessem as informações geradas por ferramentas web de gerência de projetos ágeis. Devido à existência de partes comuns e pontos de variação entre as ferramentas web, esse trabalho desenvolveu uma linha de produtos de software como mecanismo para desenvolvimento dos aplicativos propostos. Este trabalho descreve o desenvolvimento desta linha de produto de software.*

## 1. Introdução

Os métodos ágeis surgiram como uma alternativa aos métodos tradicionais de desenvolvimento de software e vem se popularizando cada vez mais entre as equipes de desenvolvimento. O desenvolvimento de software em pequenas releases permite um acompanhamento do andamento do produto por parte do cliente, permitindo um feedback sobre as funcionalidades implementadas [Soares 2004].

Diversas ferramentas surgiram para auxiliar as equipes de desenvolvimento na produção de softwares utilizando métodos ágeis. Com o desenvolvimento iterativo e incremental, é importante que toda a equipe possa acompanhar o andamento das iterações, avaliando durante o andamento de cada uma delas se o ritmo está sendo suficiente para produzir uma *release* ao final da iteração. Uma forma de disseminar essas informações é através dos dispositivos móveis dos envolvidos no projeto.

Atualmente os smartphones estão sendo cada vez mais acessíveis, aumentando consideravelmente sua utilização e o surgimento de novas funcionalidades para o mesmo. O sistema operacional Android vem se popularizando e se tornando o principal sistema operacional para smartphones [Constantinou 2013], tendo como importante característica o fato de ser gratuito e de suas aplicações poderem ser desenvolvidas a partir de qualquer computador.

Assim, diferentes aplicativos móveis podem ser desenvolvidos para acessar as informações de diferentes ferramentas de apoio à gestão de projetos que utilizam métodos ágeis. Estas ferramentas possuem características diferentes entre si, mas apresentam vários objetivos em comum. Para lidar com essas diferenças e semelhanças e para convergir para uma série de produtos para dispositivos móveis que satisfaçam as necessidades das equipes de desenvolvimento ágil, a técnica de Linha de Produto de Software (LPS) surge como uma excelente alternativa de desenvolvimento.

O presente trabalho apresenta o desenvolvimento de diferentes aplicações para dispositivos móveis na plataforma Android que proveem acesso às informações geradas por funcionalidades selecionadas a partir de algumas ferramentas de gerenciamento de projetos para equipes ágeis. O desenvolvimento desses aplicativos ocorre utilizando a abordagem de LPS. Para facilitar e simplificar o desenvolvimento, um framework de práticas ágeis estendido para LPS [Souza and Vilain 2013] foi utilizado para viabilizar o desenvolvimento das diferentes aplicações em um curto período de tempo.

Este artigo está organizado da seguinte forma: na seção 2 é abordado o conceito de métodos ágeis e apresentação de algumas ferramentas de apoio à gestão de projetos que utilizam métodos ágeis; na seção 3 é feita a apresentação do conceito de LPS, abordando um framework estendido para LPS; na seção 4 é feita uma contextualização do sistema operacional Android; por fim, a seção 5 apresenta o desenvolvimento da LPS e dos aplicativos Android que acessam as informações de ferramentas para gerência de equipes ágeis.

## **2. Desenvolvimento Ágil de Software**

O desenvolvimento ágil de software surgiu a partir de algumas necessidades não contempladas pelos métodos tradicionais de desenvolvimento de software. Os métodos ágeis surgem valorizando a iteração do cliente com o desenvolvimento e a entrega contínua de partes do software que agreguem valores. O desenvolvimento de forma incremental e entrega de pequenas partes do software permite que novos requisitos sejam incluídos ao longo do processo de desenvolvimento e que o software comece a agregar valor para o cliente antes de sua entrega total [Chin 2004]. O reconhecimento dos métodos ágeis se deu a partir de 2001 com a publicação do “Manifesto para o desenvolvimento ágil de software” [Manifesto Ágil 2012].

Apoiados no desenvolvimento ágil de software surgem diversos métodos ágeis. Abaixo serão descritos os métodos mais conhecidos e utilizados: o Scrum e o Extreme Programming. Esses métodos também costumam ser os mais abordados pelas ferramentas de gerência de métodos ágeis.

### **2.1. Extreme Programming (XP)**

O Extreme Programming (XP) é um método ágil criado na década de 90 por Kent Beck, um dos principais apoiadores do Manifesto Ágil. O método XP surgiu com a proposta de ser utilizado por equipes de dois a dez programadores e com foco no desenvolvimento de softwares com requisitos vagos que se modificam rapidamente [Beck 2004].

O XP contribui para evitar desperdícios com funcionalidades desenvolvidas que podem ser poucas ou nunca utilizadas pelo cliente, através de iterações semanais com a priorização no desenvolvimento de funcionalidades que agreguem mais valores ao

software proposto [Academia Agile 2010]. As tarefas que agregam menor valor são deixadas para o final do projeto, com isso caso ocorra uma diminuição do escopo, as funcionalidades que agregam menor valor deixam de ser desenvolvidas.

## 2.2. Scrum

O Scrum é um framework que fornece práticas iterativas e incrementais para gerenciamento de equipes de desenvolvimento de software [Cohn 2009]. No Scrum as iterações são chamadas de sprints, sendo uma sprint um ciclo onde um conjunto de atividades é executado [Schwaber and Sutherland 2011].

Um projeto Scrum começa com a elaboração do product backlog que contém a lista de requisitos do sistema a ser desenvolvido pelo projeto. Ele é constantemente modificado para identificar o que o produto necessita para ser apropriado, competitivo e eficiente. O desenvolvimento ocorre através de sprints, que são iterações de 30 dias consecutivos, tempo necessário para a equipe produzir uma versão do produto que pode ser entregue e que atenda a um interesse significativo do product owner e dos stakeholders.

## 2.3. Ferramentas de Apoio aos Métodos Ágeis

Para o desenvolvimento desse trabalho foram selecionadas três ferramentas de apoio à gerência de projetos ágeis. Como critérios para seleção foram definidos: (1) a existência de uma versão *free* da ferramenta (independente da quantidade de usuários liberados); (2) possibilidade de instalação da ferramenta na máquina local ou em servidor específico (critério importante na facilitação no processo de desenvolvimento e teste da ferramenta); e (3) possibilidade de acessar os dados gerados pelas funcionalidades selecionadas na ferramenta, via API disponibilizada ou através de acesso direto ao banco de dados. As ferramentas selecionadas foram: tinyPM<sup>1</sup>, Pronto<sup>2</sup> e Agilefant<sup>3</sup>.

Foram identificadas nestas ferramentas funcionalidades que atendessem as principais práticas dos métodos ágeis, como gestão de histórias de usuários, tarefas, iterações e quadro de acompanhamento das iterações.

## 3. Linha de Produtos de Software

Uma Linha de Produtos de Software (LPS) é um conjunto de aplicativos que compartilham em comum uma série de características que satisfazem as necessidades específicas de um determinado segmento de mercado, e que são desenvolvidos a partir de um conjunto base de componentes previamente planejados para esse segmento [Clements and Northrop 2002].

O desenvolvimento de componentes para uma LPS ocorre de forma previamente planejada. Ao invés de se colocar todos os componentes dentro de uma biblioteca sem previsão de quando serão reutilizados posteriormente, na LPS os artefatos de software só são criados quando é identificado que estes serão utilizados em um ou mais produtos da LPS [Krueger 2011].

---

<sup>1</sup> <http://www.tinypm.com/>

<sup>2</sup> <https://github.com/bluesoft/pronto-agile>

<sup>3</sup> <http://agilefant.com/>

O desenvolvimento de uma LPS consiste em duas atividades principais: engenharia de domínio e engenharia de aplicação [Clements, P. and Northrop, L. 2002]. A engenharia de domínio trata do levantamento de requisitos e elaboração dos componentes da LPS, já a engenharia de aplicação consiste no desenvolvido de aplicações, através da seleção de componentes obtidos no processo de engenharia de domínio.

### **3.1. Framework FAP Estendido**

O Framework de Práticas Ágeis (FAP – Framework of Agile Practices), proposto inicialmente em [Fagundes 2005], foi elaborado a partir de uma análise comparativa de diferentes métodos ágeis e reúne as melhores práticas dos métodos ágeis. Porém, o FAP não é adequado ao desenvolvimento de LPS.

Em [Souza and Vilain 2013] práticas utilizadas tradicionalmente no desenvolvimento de LPS foram acrescentadas ao FAP. No FAP estendido para LPS, o processo de desenvolvimento é dividido em dois ciclos, denominados “Engenharia de Domínio” e “Engenharia de Aplicação”, e cada ciclo, por sua vez, é subdividido em atividades menores. Durante o processo de engenharia de domínio, são coletadas as informações e requisitos referentes ao domínio da LPS que se pretende desenvolver e são definidos os requisitos que são comuns a todos os produtos da LPS e os que fazem parte dos pontos de variabilidade. Durante o processo de engenharia da aplicação, um produto específico é desenvolvido a partir da LPS, ou seja, um produto é desenvolvido através da utilização dos artefatos produzidos na engenharia de domínio. Cada um destes dois ciclos apresenta um conjunto de práticas que podem ser selecionadas durante a definição do processo de desenvolvimento da LPS. As práticas obrigatórias, como o próprio nome indica, devem sempre estar presentes em qualquer processo.

## **4. Android**

Segundo o site oficial do projeto Android [Android 2013], o “Android é um software open-source criado para telefones móveis e outros dispositivos”. O projeto é liderado pela Google, que fica encarregada pelo desenvolvimento e manutenção do Android.

O Android teve sua origem através de um grupo de empresas chamado Open Handset Alliance (OHA), liderado pela Google. O Projeto Android tem como objetivo principal “criar um produto real e de sucesso, que melhore a experiência de usuários finais com dispositivos móveis” [Android 2013]. Para alcançar tal objetivo, o projeto Android oferece uma plataforma aberta onde operadoras, fabricante de equipamentos e desenvolvedores possam transformar suas ideias inovadoras em realidade.

O Android é intencionalmente open-source, dessa forma as organizações que compartilham necessidades em comum podem agrupar recursos para contribuir na implementação de um único produto [Android 2013]. Apesar de receber várias contribuições, o Android é visto como um produto único e integral, e não uma distribuição, especificação ou coleção de partes reutilizáveis.

## **5. Desenvolvimento da LPS para Aplicativos Móveis**

Como citado anteriormente, o FAP estendido para LPS [Souza and Vilain 2013] sugere várias práticas a serem adotadas na definição de um processo para a engenharia de domínio e engenharia de aplicação. Nas Figuras 1 e 2 as práticas selecionadas para

serem utilizadas na engenharia de domínio e na engenharia de aplicação da LPS desenvolvida neste trabalho estão identificadas com “X”.

<b>Engenharia de Domínio</b>	Lista de requisitos de cada aplicação (obrigatório)	X
	Modelo de features (obrigatório)	X
	Documentação	
	Projeto da arquitetura do sistema (obrigatório)	X
	Representação da arquitetura de domínio (obrigatório)	X
	Planejamento da iteração (obrigatório)	X
	Duração da iteração (obrigatório)	X
	Distribuição dos módulos para os responsáveis	
	Reunião diária	
	Projeto da iteração	
	Desenvolvimento da iteração (obrigatório)	X
	Escrita dos testes de unidades	X
	Escrita dos testes de aceitação	
	Desenvolvimento coletivo do código	
	Programação em pares	
	Refatoração	
	Programação lado-a-lado	
	Inspeção do código	X

Figura 1. Práticas selecionadas da Engenharia de Domínio.

<b>Engenharia de Aplicação</b>	Lista de requisitos (Obrigatório)	X
	Seleção dos componentes (Obrigatório)	X
	Modelo geral	
	Documentação Inicial	
	Planejamento da iteração (Obrigatório)	X
	Duração de iterações (Obrigatório)	X
	Distribuição de requisitos para os responsáveis (Obrigatório)	X
	Reuniões diárias	
	Estórias do usuário	
	Casos de uso	
	Projeto da iteração	
	Desenvolvimento da iteração (Obrigatório)	X
	Escrita dos testes de unidade	
	Escrita dos testes de aceitação	
	Desenvolvimento coletivo do código	
	Programação em pares	
	Refatoração	
	Programação lado-a-lado	
	Inspeção de código	X
	Reunião de revisão da integração	X
Geração de uma breve documentação		
Entrega do sistema ao cliente	X	

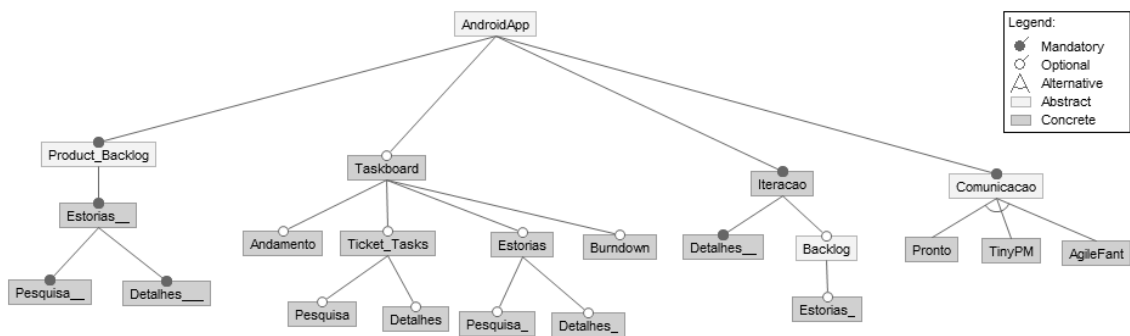
Figura 2. Práticas selecionadas da Engenharia de Domínio.

### 5.1. Engenharia de Domínio

Como indicado na Figura 1, as seguintes práticas foram realizadas: lista de requisitos, modelo de *features*, projeto da arquitetura, representação da arquitetura, planejamento da iteração, duração da iteração, desenvolvimento das iterações, escrita de testes de unidade, e inspeção de código.

Os seguintes requisitos foram identificados para fazerem parte da **lista de requisitos** dos aplicativos da LPS: product backlog (comum), detalhamento de estória do backlog (comum), pesquisa de estórias do backlog (comum), andamento da iteração atual, pesquisa de tickets do taskboard, detalhamento de ticket, pesquisa de estória do taskboard, detalhamento de estória, detalhes da iteração atual, backlog da iteração atual, possibilidade de comunicação com os sistemas Pronto, TinyPM e AgileFant, e interface gráfica para o dispositivo Android.

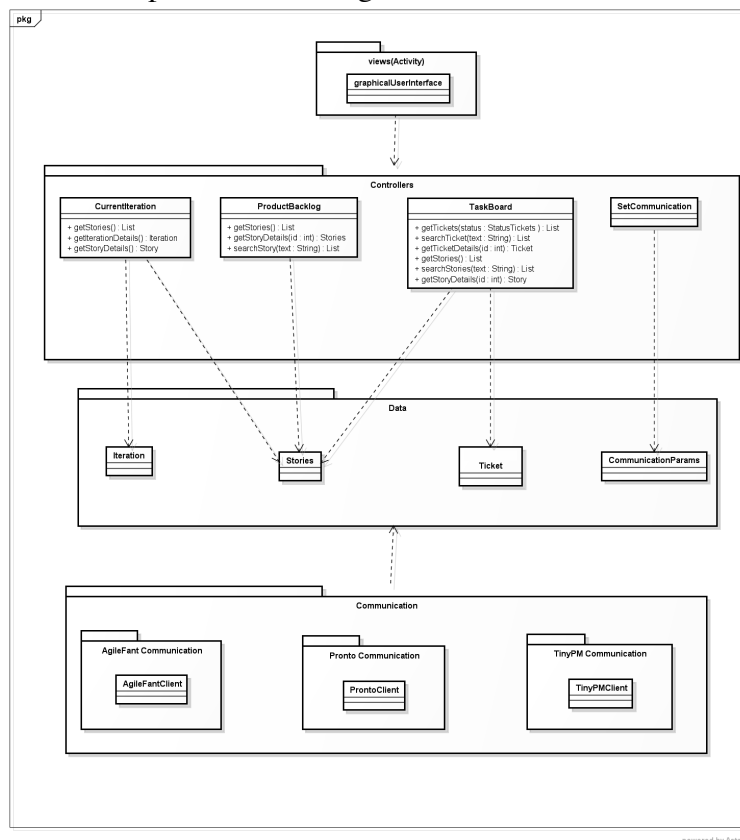
A partir dos requisitos da LPS, foi elaborado o **modelo do features** apresentado na Figura 3.



**Figura 3. Modelo de features.**

Durante o **projeto da arquitetura**, foi definido um conjunto de componentes. Os seguintes componentes foram definidos como pontos comuns da LPS: interface gráfica, product backlog, pesquisa de histórias do product backlog, detalhes de histórias do product backlog, detalhes da iteração atual. Os componentes que farão parte dos pontos de variação da LPS foram os seguintes: taskboard, andamento dos tickets, pesquisa dos tickets do taskboard, detalhe dos tickets do taskboard, backlog da iteração, comunicação.

A **representação da arquitetura** da LPS, considerando o conjunto de componentes definido, é apresentada na Figura 4.



**Figura 4. Representação da arquitetura.**

Após a definição dos componentes e arquitetura da LPS, foi iniciado o desenvolvimento destes componentes e arquitetura através do **desenvolvimento de duas iterações**. Durante o desenvolvimento destas duas iterações foram feitos **testes de unidade** e foi feita a **inspeção do código** desenvolvido para os componentes.

## Iteração 1

Nessa iteração foram desenvolvidas as classes para comunicação com a API de cada uma das ferramentas (TinyPM, Pronto e AgileFant). Cada classe obtém um json com os dados requisitados e armazena esses dados em um banco de dados SQLite local. Os dados retornados são as informações relacionadas à iteração atual, ao taskboard e ao product backlog. A Figura 5 apresenta as classes resultantes da iteração 1.

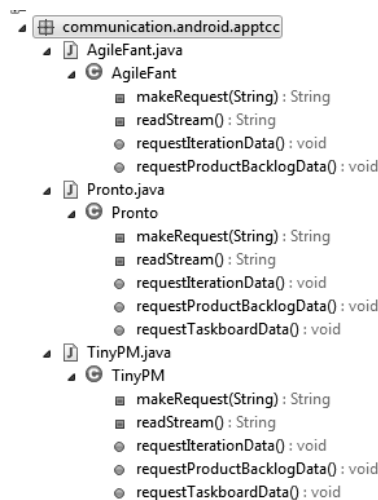


Figura 5. Classes produzidas durante a iteração 1.

## Iteração 2

Na segunda iteração foram desenvolvidas as camadas controller e view. Na camada controller foi realizada a codificação dos objetos CurrentIterations, ProductBacklog e Taskboard. Na camada view foi desenvolvida a interface gráfica, de acordo com o guia de estilos estipulado pela Google.

### 5.2. Engenharia de Aplicação

Nesta etapa foram desenvolvidos os três produtos da LPS. Cada produto é responsável pela comunicação com cada uma das três ferramentas web para gerência de projetos ágeis selecionadas nesse trabalho. Como mostrado na Figura 2, em cada iteração foram realizadas as seguintes práticas: lista de requisitos, seleção dos componentes, planejamento das iterações, duração das iterações, distribuição dos requisitos para os responsáveis, desenvolvimento da iteração, inspeção de código, reunião de revisão da integração, e entrega do sistema ao cliente. A seguir, é mostrado o modelo de *features* de cada aplicação, resultante dos requisitos selecionados.

### Produto AppPronto

O primeiro produto é um aplicativo Android para comunicação com a ferramenta web Pronto, chamado de AppPronto. A Figura 6 apresenta o modelo com as

*features* selecionadas para este produto. A Figura 7 destaca as abas da tela Taskboard sem a presença da opção Burndown, principal ponto de variação desse produto da LPS.

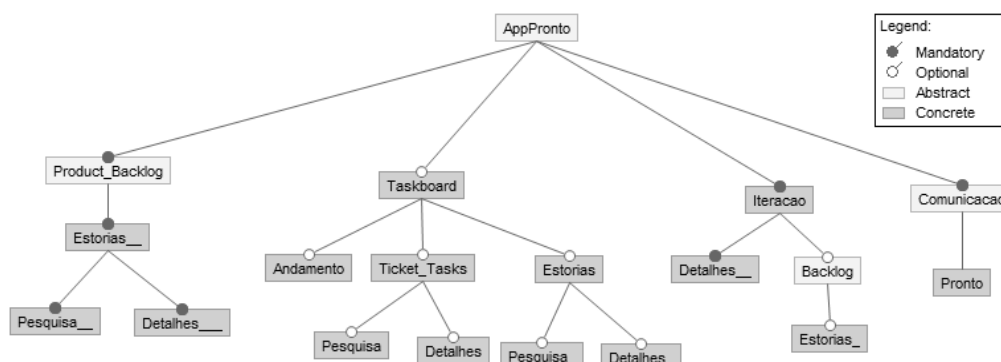


Figura 6. Modelo de *features* do Aplicativo AppPronto.

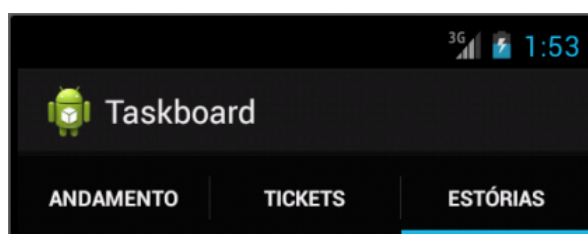


Figura 7. Abas da tela Taskboard do Aplicativo AppPronto.

### Produto AppAgilefant

O segundo produto é um aplicativo Android para comunicação com a ferramenta web Agilefant, chamado de AppAgilefant. Na Figura 8 é apresentado o modelo com as *features* selecionadas para este produto.

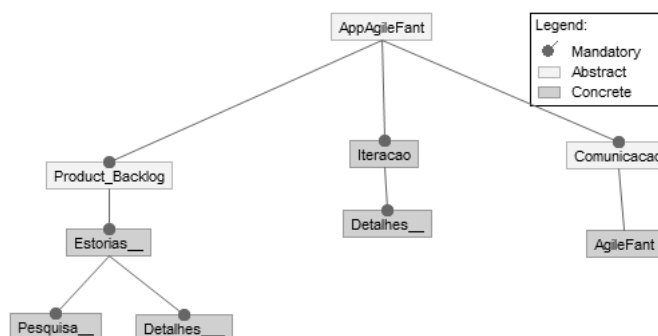


Figura 8. Modelo de domínio do Aplicativo AppAgilefant.

### Produto AppTinyPM

O terceiro produto é um aplicativo Android para comunicação com a ferramenta web TinyPM, chamado de AppTinyPM. A Figura 9 apresenta o modelo com as *features* selecionadas para este produto. A Figura 10 apresenta a interface gráfica do Aplicativo



AppTinyPM. Esta interface sofre pouca alteração em relação à interface gráfica padrão produzida na engenharia de domínio. Isso se deve ao fato do Aplicativo AppTinyPM suportar todos os componentes da LPS.

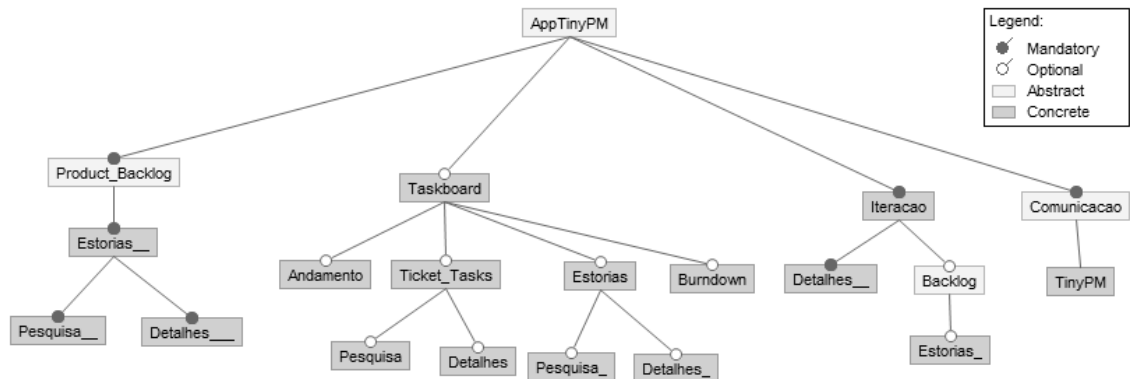


Figura 9. Modelo de domínio do Aplicativo AppTinyPM.

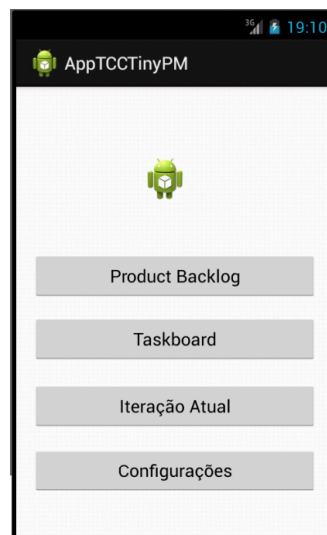


Figura 10. Tela inicial do Aplicativo AppTinyPM.

## 6. Conclusão

Esse trabalho resultou no desenvolvimento de uma LPS composta de três aplicativos para Android, onde cada aplicativo é responsável por acessar os dados gerados por três ferramentas web para gerência de equipes ágeis selecionadas. As ferramentas web selecionadas foram Pronto, tinyPM e Agilefant, por serem ferramentas de código aberto ou por possuírem uma versão gratuita que pudesse ser baixada e instalada em um servidor próprio.

Para desenvolver a LPS e os aplicativos, o processo de desenvolvimento foi definido utilizando o framework FAP estendido para LPS proposto em [Souza and Vilain 2013]. O desenvolvimento da LPS proposto através de iterações permitiu que pequenas entregas fossem realizadas e validadas ao longo do processo de elaboração da mesma. A LPS desenvolvida nesse trabalho reforça a eficiência do framework estendido e da combinação das práticas selecionadas.

A LPS desenvolvida neste trabalho está disponível em <http://www.inf.ufsc.br/~bruno.fortes/>, podendo ser estendida com novas *features* e com o desenvolvimento de novos aplicativos. Para facilitar a extensão desta LPS, é importante ressaltar que durante a modelagem da representação da arquitetura, percebeu-se que para cada *feature* do modelo de *features* poderia existir um controlador correspondente na representação da arquitetura, e para cada ação (opção) da *feature* poderia existir um método correspondente dentro do controlador. Dessa forma, a representação da arquitetura espelha o modelo de *features*, indicando como ele será desenvolvido e garantido que todo ele será atendido pela LPS.

## Referências

- Academia Agile (2010) Práticas Ágeis de Engenharia de Software com XP. 2010.
- Android Developers (2013) Disponível em <<http://source.android.com/>> Acesso em 19 mar. 2013
- Beck, K. (2004) Programação Extrema Explicada, Bookman, 2004.
- Chin, G. (2004) Agile project management: how to succeed in the face of changing project requirements, 2004.
- Clements, P. and Northrop, L. (2002) Software Product Lines: Practices and Patterns. Boston: Addison-Wesley, 2002, 563 p.
- Cohn, M. (2009) Software Development Using Scrum, 2009.
- Constantinou, A., Camilleri, E. and Kapetanakis, M. (2013) Developer Economics Q3 2013: State of the Developer Nation. Londres: Visionmobile, 2013.
- Fagundes, P. B. (2005) Framework Para Comparação e Análise de Métodos Ágeis. 2005. 134f. Dissertação (Mestrado em Ciência da Computação) – Departamento de Informática e Estatística, Universidade Federal de Santa Catarina, Florianópolis, 2005.
- Krueger, C. W. (2011) “The emerging practice of software product line development”. Austin, TX. 2011. Disponível em <<http://mil-embedded.com/PDFs/BigLever.Fall06.pdf>>
- Manifesto Ágil (2012) Manifesto para o desenvolvimento ágil de software. Disponível em <<http://manifestoagil.com.br/>> Acesso em 03 abr. 2012
- Schwaber, K. e Sutherland, J. (2011) The Scrum Guide the official rulebook. 2011. Disponível em: <<http://www.scrum.org/>> acessado em novembro de 2011.
- Soares, M. dos S. (2004) Metodologias Ágeis Extreme Programming e Scrum para o Desenvolvimento de Software. Disponível em: <<http://revistas.facecla.com.br/index.php/reinfo/article/view/146/38>> Acesso em 03 abr. 2012
- Souza, D. S. and Vilain, P. (2013) “Selecting Agile Practices for Developing Software Product Lines”. SEKE 2013.