



UNIVERSIDADE FEDERAL DO AMAZONAS
INSTITUTO DE COMPUTAÇÃO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

MSECO Watcher: Uma Ferramenta de apoio ao Monitoramento de Saúde de Ecossistemas Móveis

Marcos Gabriel Costa

Manaus – AM
Fevereiro 2015

Marcos Gabriel Costa

MSECO Watcher: Uma Ferramenta de apoio ao
Monitoramento de Saúde de Ecossistemas Móveis

Monografia de Graduação apresentada ao Instituto de
Computação da Universidade Federal do Amazonas
como requisito parcial para obtenção do grau de bacharel
em Ciência da Computação.

Orientador

D.Sc. Arilo Claudio Dias Neto

Universidade Federal do Amazonas – UFAM

Instituto de Computação – IComp

Manaus-AM

Fevereiro 2015

MSECO Watcher: Uma Ferramenta de apoio ao Monitoramento de Saúde de Ecossistemas Móveis

Autor: Marcos Gabriel Costa

Orientador: D. Sc. Arilo Claudio Dias Neto

RESUMO

Cada vez mais pessoas adquirem smartphones e utilizam aplicativos baixados pelas suas respectivas lojas de aplicativos. Nesse contexto, um ecossistema de software móvel (MSECO) consiste numa comunidade que trabalha para promover um sistema operacional móvel, como o *Windows Phone* e o *iOS*, através da publicação de aplicativos em suas lojas. Tendo como principais contribuintes o *keystone*, que toma decisões e fornece o sistema operacional, os evangelistas, que trabalham na promoção da plataforma e auxílio na publicação, e os desenvolvedores de aplicativos, que desenvolvem novas soluções de software, a maneira de verificar se este ecossistema está saudável consiste no monitoramento dos aplicativos publicados e dos seus publicadores. Tendo em vista essa necessidade, a ferramenta MSECO Watcher é proposta para auxiliar no acompanhamento desses ecossistemas móveis. Este trabalho descreve as funcionalidades desta ferramenta. Utilizando um algoritmo que percorre a *Windows Phone Store* inserindo em um documento as informações de aplicativos publicados nas lojas, uma das principais funções da ferramenta consiste na atualização e inserção dos dados automaticamente através do upload desse documento gerado. Além disso, através de um histórico dos dados inseridos, exibe na forma de gráfico, informações de desempenho referente aos downloads e ratings de aplicativos, da quantidade de aplicativos publicados por um desenvolvedor, e a quantidade de desenvolvedores e evangelistas de um ecossistema.

Palavras-chave: Ecossistema Móvel, Saúde de Ecossistema, Ferramenta Case.

MSECO Watcher: A Tool to Support the Monitoring of Health in Mobile Ecosystems

Author: Marcos Gabriel Costa

Advisor: D. Sc. Arilo Claudio Dias Neto

ABSTRACT

The number of people that purchase smartphones and use applications downloaded by their respective application stores is growing in the last years. In this context, a mobile software ecosystem (MSECO) is a community working to promote a mobile operating system, such as Windows Phone and iOS, through the submission/publication of applications in their stores. The principal contributors of the ecosystem are the keystone, who makes decisions and provides the operating system, the evangelists, who works on the platform promotion and the support for publishing, and application developers, who builds new software solutions. Thus, the way to check if the ecosystem is healthy consists in monitoring applications and its publishers. Based on this scenario, *MSECO Watcher* tool is proposed to help the management of these mobile ecosystems. This work describes the features of this tool. Using an algorithm that runs the Windows Phone Store and creates a document with information regarding applications published in the stores, one of the main functions of the tool is to update and insert data automatically by uploading this document. In addition, using a historic of data inserted into the tool, it displays in graphical form, performance information related to download and ratings of the applications, the number of applications published by a developer, and the amount of developers and evangelists of an ecosystem.

Keywords: Mobile Ecosystem, Ecosystem Health, CASE Tool.

Lista de Figuras

Figura 1 - Nível geral de turbulência e a complexidade das suas relações com os outros no ecossistema.	12
Figura 2 - Fatores e determinantes da saúde de um ecossistema de negócios apresentado por Iansiti e Lavien	14
Figura 3 - Relação entre a classificação de Iansiti e Levien (2002), as variáveis de medição selecionadas e o conceito de Hartigh et al. (2006) da saúde de Ecossistemas de Negócios	15
Figura 4 - Pesquisa em Saúde de Ecossistemas de Softwares Livres – A Legenda representa a ocorrência da métrica em um dos projetos de pesquisa	16
Figura 5 - Diagrama de Caso de Uso do Sistema	19
Figura 6 - Tela de Login	23
Figura 7 - Tela do Keystone	23
Figura 8 - Tela de Detalhamento de Ecossistema	24
Figura 9 - Tela de Evangelista	25
Figura 10 - Tela de Desenvolvedor	25
Figura 11 - Tela de Aplicativo	26
Figura 12 - Conjunto da Exibição dos Gráficos de Desempenho do Sistema	27
Figura 13 - Tela do Tipo de Gráfico	27
Figura 14 - Tela de Exibição de Gráfico	28
Figura 15 - Tela de Cadastro de Ecossistema	29
Figura 16 - Tela de Cadastro de Desenvolvedor	29
Figura 17 - Tela de Cadastro de Aplicativo	30
Figura 18 - Tela de Alterar Senha	31
Figura 19 - Planilha CSV utilizada para o recolhimento de dados	32
Figura 20 - Tela de Upload de Documento	32
Figura 21 - Documento CSV de entrada para o algoritmo de varredura	33

Figura 22 - Gráficos de Desempenho de um aplicativo que recebeu uma grande quantidade de downloads.....	35
Figura 23 - Gráficos de Desempenho de um aplicativo que parou de crescer	35
Figura 24 – Gráfico de Desempenho de um desenvolvedor que evoluiu na publicação de novos aplicativos	36
Figura 25 – Gráfico de Desempenho para um desenvolvedor que parou de publicar novos aplicativos	37
Figura 26 - Gráfico de Desempenho de um evangelista relacionado ao número de desenvolvedores acrescentados ao ecossistema.....	37

Lista de Tabelas

Tabela 1 - Descrição das Funcionalidades da Ferramenta, aonde <i>A</i> é o administrador, <i>K</i> é o <i>keystone</i> , <i>E</i> é o evangelista e <i>D</i> o desenvolvedor	22
--	----

Índice

1. Introdução.....	9
2. Referencial Teórico	10
2.1 Ecosystemas de Software.....	10
2.2 Papéis no Ecosystema	11
2.3 Saúde de um Ecosystema.....	13
2.4 Métricas para medir a saúde de um Ecosystema	14
3. A Ferramenta <i>MSECO Watcher</i> para Gerenciamento de Ecosystemas Móveis.....	17
3.1 Tecnologias Utilizadas.....	17
3.2 Coleta de requisitos da ferramenta	17
3.3 Papéis da Ferramenta	19
3.4 Informações armazenadas na ferramenta.....	20
3.5 Funcionalidades	21
3.5.1 Telas dos Atores e exibição de informações de desempenho e gráficos	21
3.5.2 Cadastros e manutenção	28
3.5.3 Atualização de dados do sistema através de um documento externo.....	31
4. Estudo de Caso.....	33
4.1 Análise dos Gráficos de Desempenho.....	34
5. Considerações Finais.....	38
6. Referências.....	38

***MSECO Watcher*: Uma Ferramenta de apoio ao Monitoramento de Saúde de Ecossistemas Móveis**

Marcos Gabriel Costa¹, Arilo Dias Neto¹

¹Instituto de Computação – Universidade Federal do Amazonas (UFAM)

Caixa Postal nº 6.200 – 69.077– 000 – Manaus – AM – Brasil

ext-marcos.g.costa@microsoft.com, arilo@icomp.ufam.edu.br

1. Introdução

Um ecossistema de software consiste em uma plataforma e aplicações construídas em cima dessa plataforma que ampliam os produtos com funcionalidades desenvolvidas por desenvolvedores externos (BOSCH, 2009). Isso significa dizer que todo ecossistema possui uma plataforma e fornece produtos e serviços construídos para facilitar seu uso por parte de desenvolvedores externos.

Em um ecossistema móvel, a plataforma é o sistema operacional de um dispositivo móvel e ele consiste em um sistema de evolução cooperativa de aplicativos móveis, desenvolvedores e usuários, que formam complexos relacionamentos preenchendo nichos, competindo e cooperando, de uma forma similar a ecossistemas biológicos (LIN e YE, 2009). O ambiente de desenvolvimento desses aplicativos envolve, principalmente, desenvolvedores, aplicativos e lojas de aplicativos, onde a loja é um ambiente utilizado para publicar, armazenar e acessar aplicativos (BENTLEY, 2012). Além disso, pode-se definir a figura de um evangelista, que representa a ligação entre a empresa responsável pela plataforma e os desenvolvedores (FONTÃO et al., 2014).

A saúde de um ecossistema mostra a qualidade do seu desempenho e sua propensão para o crescimento. Assim como no ser humano, quando um ecossistema não está saudável, ele possui mais chances de perder produtividade e acabar (ou morrer).

O Ecossistema Móvel da Microsoft é um exemplo de um ecossistema móvel baseado em aplicativos que utiliza o sistema operacional *Windows Phone* como plataforma e a *Windows Phone Store*¹ (WPS) como loja de aplicativos. Seu objetivo é promover a publicação de aplicativos na loja e, visando esta meta, a missão dos seus evangelistas é fornecer cursos e treinamentos para desenvolvedores sobre o sistema operacional e o processo de publicação, além de apresentar ou construir ferramentas que os auxiliem nessa tarefa. Através da realização de entrevistas com evangelistas desse ecossistema móvel da Microsoft, pode-se observar a maneira que o acompanhamento da saúde daquele ecossistema era realizado. Através de uma planilha, os dados dos aplicativos de todos os desenvolvedores eram armazenados e atualizados para que os evangelistas responsáveis por estes pudessem controlar a sua saúde, verificando a quantidade de downloads, a avaliação dos aplicativos, a frequência de publicação dos seus desenvolvedores, entre outras métricas.

Assim, visando a otimização desse processo de monitoramento, foi criada a ferramenta *MSECO Watcher* com o intuito de auxiliar o acompanhamento da saúde do ecossistema pelo evangelista e o *keystone* através do controle dos dados dos aplicativos e desenvolvedores, além da visualização do histórico desses dados, para mostrar as tendências dos números obtidos. Apesar de a ferramenta ter sido baseada em um ecossistema que promove a plataforma

¹ <http://www.windowsphone.com/pt-br/store>

Windows Phone, ela pode ser utilizada no acompanhamento de qualquer outro sistema que utilize uma plataforma móvel como base.

Além desta introdução, o artigo está organizado da seguinte forma: a seção 2 apresenta o que já existe na literatura relacionado a ecossistemas de software, com enfoque nas métricas de monitoramento da saúde. A seção 3 apresenta a ferramenta, batizada de *MSECO Watcher*, desenvolvida para apoiar o monitoramento de ecossistemas móveis a partir da coleta e análise de métricas oriundas das lojas de aplicativos, mostrando desde o levantamento de requisitos até as suas funcionalidades. A seção 4 mostra um estudo de caso para exemplificar a utilização da ferramenta. Finalmente, a seção 5 mostra a conclusão do artigo e os trabalhos futuros.

2. Referencial Teórico

2.1 Ecossistemas de Software

Um Ecossistema Biológico pode ser definido como uma unidade natural constituída de parte não viva e de parcela viva que interagem ou se relacionam entre si, formando um sistema estável. Baseando-se nos aspectos desses ecossistemas naturais, foi criado o Ecossistema de Negócios (Em inglês, *Business Ecosystem* – BECO). Assim como os biológicos, esses ecossistemas são caracterizados por um grande número de participantes interligados que dependem uns dos outros para a sua eficácia e sobrevivência; além disso, ambos admitem a presença de polos cruciais que assumem a função de um líder (ou *keystone*), responsável por manter o sistema saudável (IANSITI; LEVIEN, 2004).

O conceito de BECO, que serviu como base para a definição dos Ecossistemas de Software (Em inglês, *Software Ecosystems* – SECO), foi introduzido por Moore (1993) como um conjunto de organizações que desenvolvem suas capacidades ao redor de uma inovação, trabalhando coopera e competitivamente no suporte de novos produtos, na satisfação das necessidades dos clientes e na incorporação de inovações. Utilizando como base os relacionamentos existentes na biologia, Moore (1993) ainda destaca que a auto-organização, emergência e co-evolução² são conceitos também presentes nos BECO, aonde o capital, interesse do cliente, e talento gerado por uma inovação passam a ser condensados, assim como as espécies fazem com os recursos naturais.

Assim como os sistemas em que foi baseado, um SECO utiliza-se da ideia de um grupo de atores, empresas ou pessoas que contribuem para o sistema, coordenados por um *keystone*, trabalhando na resolução de problemas de software. Ele pode ser definido como um conjunto de soluções de software que permitem, apoiam e automatizam as atividades e transações dos atores do ecossistema social ou de negócios associados e as organizações que fornecem essas soluções (BOSCH, 2009). Sendo assim, os SECO visam o trabalho conjunto em torno de uma solução que acabam contribuindo para os atores do ecossistema como um todo.

A utilização de SECO pelas empresas veio como uma evolução da técnica de Linha de Produto de Software (LPS) (CLEMENTS; NORTHROP, 2001), uma abordagem bem sucedida de reaproveitamento interno de software que foi amplamente adota por empresas desse ramo. Baseando-se na ideia de linhas de produção aonde são montadas partes de produtos que posteriormente serão utilizados nos produtos variantes, a abordagem LPS visa agilizar a construção de sistemas através da criação de uma biblioteca com partes comuns que poderão ser reutilizadas, como por exemplo, a parte que corresponde ao *login*, que é utilizada pela grande maioria dos softwares.

² Na biologia, representa a evolução simultânea de duas espécies que mantém um relacionamento próximo. A evolução de uma espécie é parcialmente dependente da evolução da outra, reciprocamente.

Essas empresas passaram a permitir a contribuição de desenvolvedores externos nas plataformas que eram, até então, fechadas (BOSCH, 2009). Algumas delas utilizaram-se tanto desta prática que deixaram de funcionar como unidades independentes que podem fornecer produtos separados, mas tornaram-se dependentes de outros fornecedores de software para componentes vitais e infraestruturas, por exemplo, sistemas operacionais, bibliotecas, lojas de componentes e plataformas (JANSEN et al., 2009).

Segundo (BOSCH, 2009), uma empresa que utiliza a técnica LPS deveria estar interessada em mover-se para o modelo de SECO por dois motivos: i) pode-se perceber que a quantidade de funcionalidade que precisa ser desenvolvida para satisfazer as necessidades de seus clientes é muito maior do que aquilo que pode ser construído em um período de tempo razoável e com um investimento que ofereça um retorno aceitável, e, ii) a tendência de customização em massa impulsiona a necessidade de um grande investimento de Pesquisa e Desenvolvimento para aplicações de sucesso.

Barbosa e Alves (2009) identificaram, por meio de um estudo de mapeamento, o favorecimento à co-evolução, inovação e aumento da atratividade para novos atores como sendo os benefícios mais comuns na utilização de SECO. A co-evolução corresponde ao efeito que a evolução de um ator no ecossistema tem sobre os demais. Assim, para que um ator evolua, é necessário que outros evoluam junto com ele. A inovação surge a partir do momento em que diversos atores passam a trabalhar na mesma solução e a possibilidade de trabalho em uma nova plataforma atrai a atenção de novos atores para contribuir com a solução. Outros benefícios encontrados por Santos et al. (2012) foram: possibilidade de redução de custos pelas empresas, além do apoio a tomada de decisões de arquitetura, compartilhamento de conhecimento e comunicação dos requisitos entre os atores.

Algumas das dificuldades na utilização de SECO são relacionadas à modelagem de ecossistemas, os desafios arquiteturais, a heterogeneidade das licenças e gestão da evolução do software (SANTOS et al., 2012). Bosch (2009) ainda ressalta os desafios na aplicação de processos de maturidade, como o CMMi (CMMI PRODUCT TEAM, 2010), já que desenvolvedores externos dificilmente poderão estar sujeitos a seguir obrigatoriamente algum tipo de processo.

2.2 Papéis no Ecossistema

Como já visto anteriormente, um ecossistema é composto pelo conjunto de atores que trabalham nas soluções do software. Jansen et al. (2009) afirma que dois papéis podem ser incorporados pelas empresas em segurança durante algum período de tempo, o *keystone* e o *niche player*, e outro que tem demonstrado ser extremamente perigoso em longo prazo, o *dominator*.

Um exemplo de *keystone* encontrado na natureza são os jacarés-americanos, que habitam áreas úmidas nos Estados Unidos. Eles fazem tocas para manter suas caldas aquecidas e quando se movem esses buracos transformam-se em habitats secos e húmidos para outros organismos. Além disso, são responsáveis pelo controle da dinâmica da vegetação através da redução da população de pequenos mamíferos, especialmente a lontra, que pode acabar com toda a vegetação caso não seja impedida.

Assim como os jacarés-americanos, o *keystone* de um SECO representa apenas uma pequena parte de toda a biomassa, mas seu papel é de extrema importância. Segundo Iansiti e Levien (2004), o papel do *keystone* é melhorar a saúde geral de seus ecossistemas, fornecendo um conjunto estável e previsível de bens comuns. Outra definição de seu papel é o provimento de uma tecnologia padrão ou plataforma que seja a base de (parte do) ecossistema (JANSEN

et al., 2009). Iansiti e Levien (2004) ainda ressaltam que a estratégia de um *keystone* tem duas partes: criação de valor dentro do ecossistema e o compartilhamento desse valor para os participantes. Isso significa dizer que seu papel é o fornecimento de uma base (plataforma, tecnologia ou inovação) para o ecossistema, além de permitir o crescimento dos atores através do compartilhamento dos conhecimentos e valores encontrado. O sucesso ou a falha em seu papel fará com que novos membros sejam, respectivamente, atraídos ou retraídos.

A principal função de um *niche player* pode ser definida como a de desenvolver capacidades especializadas que o diferenciam de outros na rede (IANSITI; LEVIEN, 2004). Seu papel é evoluir a sua capacidade em áreas específicas e criar valor com a tecnologia fornecida pelo *keystone*, fazendo com que o ecossistema cresça. Eles representam a maior parte dos atores envolvidos e também são os que mais geram valor ao time.

O papel do *dominator* visa o controle e posse de todo, ou grande parte, do ecossistema em que está. Ele pode dominar completamente o sistema e ser o maior responsável em retirar valor, o que impede completamente um sistema de crescer, podendo até matar o ecossistema, já que não há troca de conhecimento. Ou ele pode representar uma pequena parte do ecossistema e contribuir com pouquíssimo, ou nenhum, valor, mas que retira o máximo que pode do ecossistema em si. Ecossistemas que usam essa estratégia tendem a não sobreviver em longo prazo e por causa da ausência de vantagens. Autores são cada vez menos atraídos para ecossistemas que possuem um *dominator*.

A Figura 1³, retirada de (IANSITI e LEVIEN, 2004), mostra os níveis de turbulência, inovação e complexidade em relação aos papéis dentro de um ecossistema.

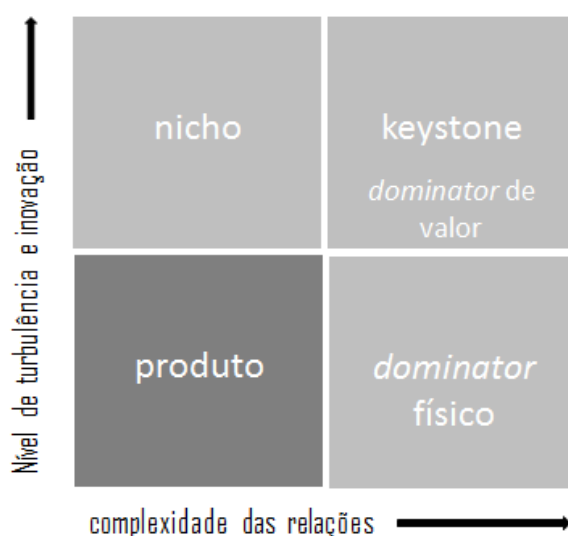


Figura 1 - Nível geral de turbulência e a complexidade das suas relações com os outros no ecossistema.

Hagel et al. (2008) categorizam os papéis de um SECO em *shapers* e *participants*. *Shapers* teriam o mesmo papel de um *keystone*, definido anteriormente. Os autores ainda definem três aspectos que ele precisa ter:

³ A palavra “produto” foi traduzida da palavra *commodity* na figura original, que significa, em tradução literal, mercadoria. Comumente, *commodity* é usada para descrever uma classe de produtos para os quais existe demanda, mas que são fornecidos sem diferenciação qualitativa no mercado.

- i) **Visão:** fornecimento de foco, identificação de oportunidades e descrição das forças fundamentais para os outros atores da SECO.
- ii) **Plataforma:** deve prover desenvolvimento do sistema, fomentar a especialização, e aumentar o número de participantes.
- iii) **Comportamento:** deve ter alta credibilidade na SECO, limitar o risco para novos membros e mostrar que é esperado um grande comprometimento por parte dos atores.

Os *participants* são equivalentes aos *niche players* e podem ser divididos em: *hedgers*⁴, *disciples* e *influencer*. Um *hedger* participa de dois SECO concorrentes, a fim de minimizar riscos. Ele pode ser caro de ser mantido devido ao trabalho duplicado. Um *disciple* representa um ator comprometido a apenas uma plataforma e que normalmente divulga amplamente a tecnologia do *keystone*. O problema com esse ator pode ocorrer quando a plataforma acaba não sendo adotada. E o último papel, *influencer*, consiste em um ator que não apenas participa ativamente do SECO, mas também exerce uma grande influência sobre o *keystone*, apresentando solicitação de recursos, organizando conferências, etc.

2.3 Saúde de um Ecossistema

Como a maioria dos conceitos utilizados na especificação de BECO e SECO, a ideia de saúde também é originada nos estudos dos Ecossistemas Biológicos. Constanza (1992) definiu um ecossistema natural saudável como sendo aquele que é estável e sustentável, ou seja, que mantém a sua organização e autonomia ao longo do tempo e que é resistente ao estresse. Outros fatores que indicam a saúde de um ecossistema são sua produtividade e a variedade de espécies que o compõe. Manikas e Hansen (2013) ainda listam quatro maneiras em que a estrutura de um ecossistema (número de organismos, biomassa, etc.) e as suas funções (atividade, decomposição, etc.) podem estar conectadas: i) extremamente conectadas, aonde as mudanças de um resultam em mudanças no outro; ii) mudanças na estrutura não afetam funções, iii) mudanças nas funções não afetam a estrutura, e, iv) estruturas e funções completamente desconectadas.

Utilizando as características encontradas nos ecossistemas naturais, Iansiti e Levien (2004) introduziram o conceito de saúde no contexto de BECO afirmando que para que um ecossistema funcionasse efetivamente, cada domínio crítico para a entrega de um produto deveria ser saudável. Hartigh (2006) afirma que, assim como na biologia, a saúde de um ecossistema é relacionada à longevidade do sistema e a propensão para o crescimento.

No contexto de SECO, Berk et al. (2010) propôs um modelo estratégico para um SECO baseado na saúde, aonde ele fazia uma analogia entre a saúde de um ecossistema à saúde de um ser humano. No modelo proposto, a saúde era influenciada pelos seguintes fatores: a biologia do ecossistema, o estilo de vida, o ambiente, e a intervenção de organizações que cuidam da saúde.

Um importante ponto a ser considerado sobre a saúde em SECO é o apresentado por Jansen (2014), que diz que a saúde de um ecossistema não pode ser um parâmetro para o sucesso comercial de um produto, já que afirmar que um ecossistema é saudável significa dizer que ele está ganhando mais desenvolvedores, atividades e novos projetos comparado a outros. O que pode ser dito é que um ecossistema mais saudável, em um conjunto de ecossistemas de mesmo tamanho e domínio, é a melhor escolha quando um ator está pensando em se juntar a um ecossistema.

⁴ Hedger é um conceito utilizado em economia e corresponde a um investidor que utiliza operações defensivas para evitar riscos.

2.4 Métricas para medir a saúde de um Ecossistema

Ao definir o conceito de saúde em BECO, Iansiti e Levien (2002) também definiram as primeiras medidas para a verificação da saúde destes. Segundo eles, existem três medidas críticas de saúde, tanto para BECO quanto para Ecossistemas Biológicos, que são:

- Produtividade, ou a eficiência com que um ecossistema transforma tecnologia e outros materiais de inovações em novos produtos, com custos reduzidos.
- Robustez, ou a capacidade de sobrevivência de um ecossistema mediante a rupturas e mudanças de tecnologia.
- Criação de nichos, ou a habilidade de aumentar a diversidade através da criação de novas funcionalidades significativas, ou nichos.

Além de definir a produtividade, robustez e a criação de nichos como determinantes na medição da saúde de um ecossistema, Iansiti e Levien (2002) ainda definiram uma lista de fatores que compõem cada um desses determinantes. Esses fatores foram resumidos em uma tabela por Hartigh et al. (2006), de acordo com a Figura 2.

Produtividade	Robustez	Criação de Nichos
<ul style="list-style-type: none">- Total fator de produtividade- Melhoramento da produtividade- Entrega de Inovações	<ul style="list-style-type: none">- Taxas de sobrevivência- Persistência da estrutura- Previsibilidade- Tecnologia Obsoleta Limitada- Continuidade	<ul style="list-style-type: none">- Variedade- Criação de Valor

Figura 2 - Fatores e determinantes da saúde de um ecossistema de negócios apresentado por Iansiti e Lavien

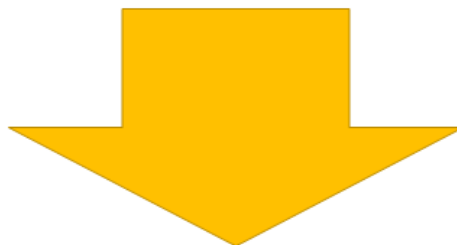
Para a produtividade, os fatores são: i) Total fator de produtividade, que analisa quanto dos fatores de produção são convertidos em produtos úteis; ii) Melhoramento da produtividade no decorrer do tempo; e iii) Entrega de inovações, que é a habilidade de adaptação do ecossistema e na entrega de novas tecnologias e ideias para seus membros.

Para a robustez, os fatores são: i) Taxas de sobrevivência, sobrevivência dos atores no decorrer do tempo; ii) Persistência da estrutura, quanto as relações entre os autores permanecem inalteradas; iii) Previsibilidade, medida de quanto o núcleo do ecossistema permanece inalterado mesmo com os choques nos relacionamentos dos atores; iv) Tecnologia obsoleta limitada, se o ecossistema limita o investimento em tecnologias que podem se tornar obsoletas no decorrer do tempo; e, v) Continuidade, em que avalia a medida em que os produtos crescem gradualmente com a tecnologia ao invés de mudarem abruptamente.

Os fatores da criação de nichos são: i) Variedade, que representa o número de novas opções, produtos e tecnologias que estão sendo criados dentro de um ecossistema em um determinado período de tempo; e, ii) Criação de valor, que mede o valor total que essas novas opções criadas têm para o ecossistema.

Usando como base os fatores definidos por Iansiti e Levien (2002), Hartigh et al. (2006) ainda criou uma nova lista de métricas para BECO, mostrada na Figura 3.

Produtividade	Robustez	Criação de Nichos
<ul style="list-style-type: none"> - Total fator de produtividade - Melhoramento da produtividade - Entrega de Inovações 	<ul style="list-style-type: none"> - Taxas de sobrevivência - Persistência da estrutura - Previsibilidade - Tecnologia Obsoleta Limitada - Continuidade 	<ul style="list-style-type: none"> - Variedade - Criação de Valor































Saúde do Parceiro	Saúde da Rede
<ul style="list-style-type: none"> - EBIT/ Total de ativos - Renda total/total de ativos - Liquidez - Solvência & solvência t-1 - Lucros acumulados / total de ativos - Crescimento total de ativos - Capital de giro / total de ativos 	<ul style="list-style-type: none"> - Número de parcerias - Visibilidade no Mercado - Covariância entre a variedade de parceiros e o mercado

Figura 3 - Relação entre a classificação de Iansiti e Levien (2002), as variáveis de medição selecionadas e o conceito de Hartigh et al. (2006) da saúde de Ecossistemas de Negócios

Eles transformaram os determinantes em dois: saúde dos parceiros, que é uma medição financeira e de longo prazo sobre a força de um parceiro de gestão e a sua capacidade de exploração da tecnologia e das oportunidades oferecidas dentro do ecossistema; e a saúde da rede ou do ecossistema, que mede o quanto um parceiro está embutido no ecossistema, além da sua influência na rede. Cada um deles também possuem seus fatores.

Uma métrica encontrada na literatura criada para Ecossistemas de Software Livres (OSS) e para ecossistemas em geral foi a proposta por Jansen (2014), que realizou uma ampla pesquisa na área de Saúde de Ecossistemas para reunir as principais medições encontradas na literatura. Essas métricas podem ser encontradas na Figura 4.

Utilizando como base os determinantes de Iansiti e Levien (2002), as métricas foram divididas em Nível Teórico, Nível de Rede e Nível de Projeto. O primeiro nível representa o nível teórico em que a pesquisa foi baseada. O segundo mostra as métricas encontradas na literatura que podem ser utilizadas especificamente para medir a saúde de OSS. No terceiro nível foram criadas métricas para a saúde de projetos em uma visão abrangente que, como foi coletada de múltiplos projetos presentes em um ecossistema, podem ser utilizadas para medir a saúde de ecossistemas em geral.

Teoria	Produtividade	Robustez	Criação de Nichos
	<ul style="list-style-type: none"> - Total fator de produtividade - Melhoramento da produtividade - Entrega de Inovações 	<ul style="list-style-type: none"> - Taxas de sobrevivência - Persistência da estrutura - Previsibilidade - Tecnologia Obsoleta Limitada - Continuidade 	<ul style="list-style-type: none"> - Variedade - Criação de Valor
Nível de Rede	Produtividade	Robustez	Criação de Nichos
	<ul style="list-style-type: none"> - Novos projetos relacionados    - Download de novos projetos - Conhecimento adquirido sobre o ecossistema - Eventos 	<ul style="list-style-type: none"> - Número total de projetos ativos     - Conectividade de projetos/Coesão  - Consistência do núcleo da rede  - Links de saída para outros SECO - Custo de mudança para outros SECO 	<ul style="list-style-type: none"> - Variedade em projetos 
Nível de Projeto	Produtividade	Robustez	Criação de Nichos
	<ul style="list-style-type: none"> - KLOC/ período de tempo   - Novos tickets - Novos downloads  - Criação de conhecimento e artefatos   - Tempo de resposta na lista de emails  - Tempo de concerto de bugs - Spin-offs e forks  - Novas parcerias - Novas patentes - Uso 	<ul style="list-style-type: none"> - Parcerias e incorporação - Maturidade da organização - Patrocínio comercial - Contribuições e doações - Satisfação do contribuinte - Contribuintes ativos     - Avaliação e reputação do contribuinte - Multi-homers - Conectividade dos contribuintes - Interesse: visitas na página, estatísticas de pesquisa   - Compartilhamento de mercado  - Custo de mudanças - Uso e lealdade do usuário - Satisfação e avaliação do usuário - Qualidade do artefato 	<ul style="list-style-type: none"> - Variação no tipo de contribuinte  - Variação nas aplicações do projeto - Suporte de linguagens naturais  - Variedade no suporte à tecnologias  - Variedade no desenvolvimento de tecnologias  - Múltiplos mercados

 Lucassen et al., 2003  Hoving, Slot, e Jansen, 2013  Goeminne & Mens, 2013  van Lingen, Palomba, e Lucassen, 2013

Figura 4 - Pesquisa em Saúde de Ecossistemas de Softwares Livres – A Legenda representa a ocorrência da métrica em um dos projetos de pesquisa.

Focando no terceiro nível, pode-se notar a presença de métricas que já são largamente utilizadas para a determinação da qualidade de alguns projetos. Na coluna de produtividade, existem algumas métricas como: o número de linhas do código; o número de downloads do produto, que mostra o quanto os usuários finais estão realmente usando o produto; o tempo de conserto de um defeito, que indica em quanto tempo os problemas do software são resolvidos; e o conhecimento e criação de artefatos, que mostra a quantidade de manuais, guias e documentos que reúnem o conhecimento adquirido que é gerada. Convém observar a advertência dada por Jensen (2014) a respeito das linhas de código: “o crescimento de linhas de código é altamente enganoso: a qualidade de um projeto pode melhorar significativamente quando uma grande parte do código morto é removido, quando duplicatas são substituídas, e quando um projeto é reformulado de forma significativa”.

A robustez foi dividida em três partes: organização, atores e usuários finais. Pode-se observar a métrica de satisfação do usuário e sua classificação, que no contexto de aplicativos móveis podem ser retirados da própria loja, como *AppStore* ou *Windows Phone Store*, por exemplo, além da sua lealdade aos produtos do ecossistema; o número de atores ativos, sua

satisfação e sua reputação nos mostra como está a produtividade do ator no ecossistema e se ele está satisfeito com os desafios que tem recebido. Por fim, o nível de criação de nichos mostra se tem se dado a liberdade suficiente para usuários e atores criarem variedade de nichos dentro do ecossistema.

As métricas que envolvem a saúde de ecossistemas são de vital importância para a tomada de decisão. Jansen (2014) destaca que os coordenadores de SECO exigem as informações decorrentes das métricas de saúde a fim de tomar as melhores decisões sobre as suas estratégias. Além disso, o autor recomenda que as empresas que produzem software deveriam verificar regularmente a saúde dos ecossistemas que estão em volta das plataformas que eles são dependentes, já que grandes mudanças nestes ecossistemas podem afetar grandemente o sucesso na produção dos seus produtos de software.

3. A Ferramenta *MSECO Watcher* para Gerenciamento de Ecossistemas Móveis

3.1 Tecnologias Utilizadas

A ferramenta *MSECO Watcher* é um sistema *web* para o acompanhamento de ecossistemas móveis. Para a construção da ferramenta, foi utilizada a linguagem de programação PHP juntamente com a linguagem de marcação HTML, além do CSS para a formatação da sua interface gráfica. Para o armazenamento dos dados manipulados pela ferramenta, foi utilizada a linguagem SQL através do SGBD PostgreSQL (POSTGRESQL, 2015).

A linguagem PHP é largamente utilizada e especialmente adequada para o desenvolvimento de sistemas *web* e pode ser embutida dentro de código HTML. Seu código é interpretado do lado do servidor pelo módulo PHP, que também gera a página *web* a ser utilizada no lado do cliente.

HTML é uma linguagem de marcação utilizada para o desenvolvimento de páginas na *web*. Este tipo de linguagem é constituído de códigos que delimitam conteúdos específicos, segundo uma sintaxe própria, que no HTML são as *tags*. Seus documentos podem ser interpretados pelos navegadores de internet.

A Linguagem SQL é a linguagem de pesquisa declarativa padrão para bancos de dados relacionais. Sua principal característica consiste na simplicidade, no sentido de que uma consulta SQL especifica a forma do resultado e não o caminho para chegar a ele. Através de um Sistema de Gerenciamento de Banco de Dados (SGBDs) é possível a manipulação de banco de dados. Todas as operações realizadas no banco de dados podem ser solicitadas ao SGBD utilizando esta linguagem. Neste trabalho, foi adotado o SGBD PostgreSQL.

CSS (*Cascading Style Sheets*) é uma linguagem para estilos que define o layout de documentos HTML. Ele define como serão exibidos os elementos contidos no código de uma página da Internet, e sua maior vantagem é efetuar a separação entre o formato e o conteúdo de um documento. Isso significa que o desenvolvedor pode criar uma ligação para uma página que contém os estilos ao invés de colocar essa formatação dentro do código de desenvolvimento. O CSS é suportado por todos os navegadores atuais.

3.2 Coleta de requisitos da ferramenta

Os requisitos para a construção da ferramenta foram coletados através de entrevistas com os evangelistas de Ecossistema Móvel da Microsoft, que eram responsáveis pelo acompanhamento de um grupo de desenvolvedores que deveriam publicar regularmente aplicativos na *Windows Phone Store*. Como resultado desses encontros, foi possível entender

o processo que já era realizado para o controle e monitoramento do desempenho das pessoas pertencentes ao ecossistema e reunir ideias para o sistema que os ajudaria nessa tarefa.

Uma loja de aplicativos móveis é um local que reúne aplicativos, gratuitos e pagos, feitos para uma plataforma mobile específica. Nela, o usuário pode pesquisar aplicações e baixá-las para seus dispositivos móveis. Cada aplicativo possui um nome, uma descrição, uma imagem e uma categoria, dentre outros atributos. Além disso, através dessa loja, cada usuário pode fornecer uma classificação para um aplicativo que varia de uma a cinco estrelas, sendo uma estrela a pior classificação e cinco a melhor. A média de todas as classificações de um aplicativo é chamada de *rating*.

O processo utilizado anteriormente era a manutenção de uma planilha contendo uma lista dos aplicativos publicados pelos desenvolvedores que estavam sob a responsabilidade de cada um dos evangelistas do ecossistema e um resumo referente a esses evangelistas. A lista mostrava todos os aplicativos publicados contendo as suas informações (quantidade de downloads, *rating* e data da última atualização), além das informações do publicador (nome, e-mail e *publisher name*⁵). Para cada evangelista, também era exibido um resumo com as quantidades de aplicativos publicados, de publicadores e de treinamentos realizados. Essas informações eram atualizadas automaticamente, de acordo com novas inserções na lista. Os dados eram fornecidos pelos próprios desenvolvedores, que tinham a tarefa de enviar semanalmente as informações de seus aplicativos para que os evangelistas atualizassem a planilha. Os problemas encontrados nessa abordagem foram a quantidade tempo gasto para a colheita das informações de todos os aplicativos e a ausência da entrega das informações no dia acordado.

Visando a resolução destes problemas, um dos evangelistas do Ecossistema Móvel da Microsoft construiu um algoritmo, chamado de *algoritmo de varredura* (externo ao escopo deste trabalho), que percorre a *Windows Phone Store* (WPS) através do *publisher name* dos desenvolvedores na loja e reúne as informações de todos os aplicativos publicados por estes em um documento *CSV*⁶, onde cada linha do arquivo exibe os dados correspondentes a um aplicativo. Porém, devido a grande quantidade de aplicativos, a versão corrente do algoritmo requer entre uma hora e meia a duas horas para reunir todos os dados.

Através das entrevistas e da explicação do processo existente, foi possível chegar à conclusão das funcionalidades do sistema. Uma delas seria a visualização da situação de cada um dos desenvolvedores: sua média de *ratings*, seu número total de downloads e a quantidade de aplicativos criados e seu histórico de publicações, para que fosse possível analisar a tendência dos dados daquele desenvolvedor. Outra funcionalidade identificada como necessária seria mostrar os dados específicos de cada aplicativo para que se pudesse verificar se a média do desenvolvedor era causada por alguns poucos que possuíam valores muito baixos de *rating* e downloads. Além dessas, mostrar o tempo de última atualização do aplicativo e manter as redes sociais como um meio de comunicação dos desenvolvedores foram outras funções acrescentadas. A utilização do documento *CSV* gerado pelos evangelistas para a otimização na inserção de informações na ferramenta foi outra funcionalidade que desejou-se acrescentar.

⁵ Nome (apelido) usado pelos publicadores em seus cadastros nas lojas de aplicativos.

⁶ *Comma-separated values*, ou valores separados por vírgula, é um tipo de arquivo contendo uma quantidade de registros que são separados por quebras de linhas de algum tipo (comumente mas não necessariamente vírgulas), sendo que cada registro é composto por campos que também possuem caracteres separadores entre si.

Tendo esses requisitos como base, viu-se que a ferramenta poderia ser utilizada não somente para o acompanhamento deste ecossistema específico, mas para qualquer ecossistema móvel, visto que o algoritmo de varredura poderia ser adaptado para outras lojas de aplicativos ou o documento CSV poderia ser gerado manualmente com os dados de aplicativos de outras plataformas.

3.3 Papéis da Ferramenta

Como pode ser visto no Diagrama de Caso de Uso da ferramenta, mostrado na Figura 5, o sistema possui quatro tipos de atores: administrador, *keystone*, evangelista e desenvolvedor.

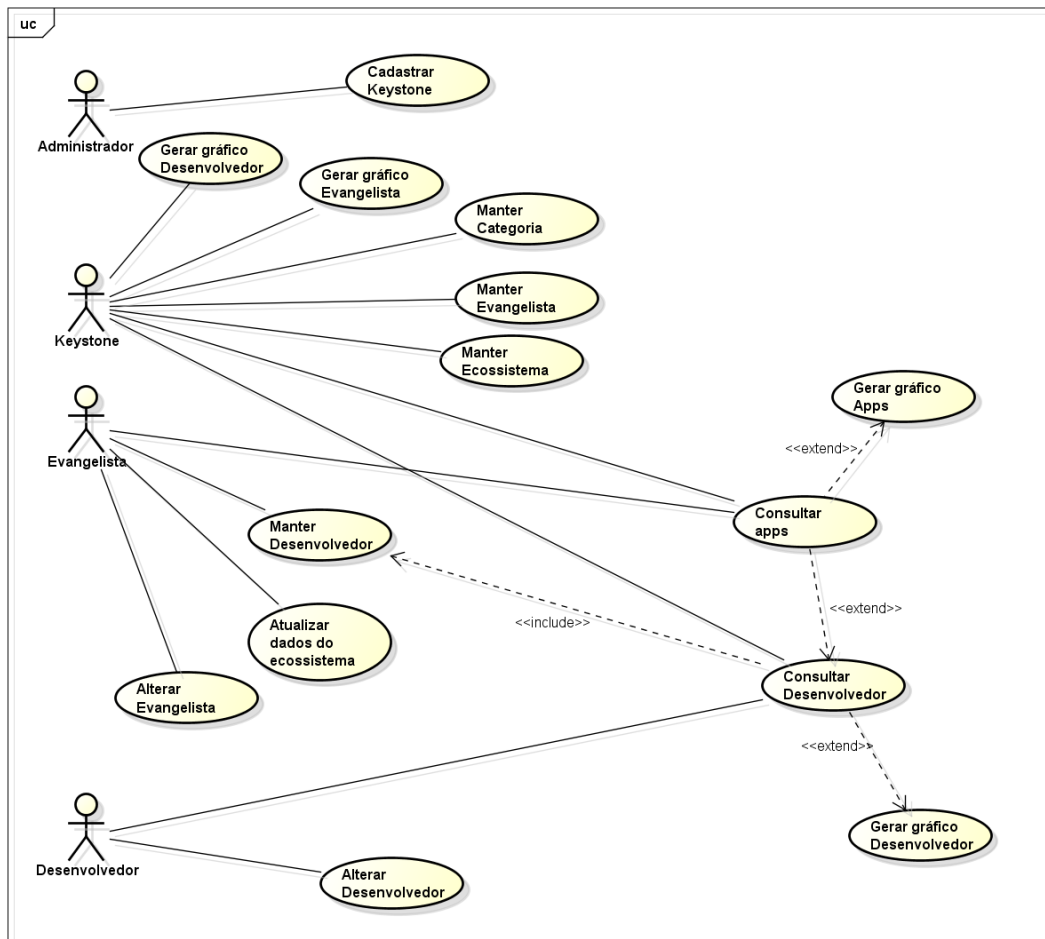


Figura 5 - Diagrama de Caso de Uso do Sistema

O **administrador** foi criado apenas para o cadastro do *keystone* do sistema. Ele somente será utilizado para delegar o sistema para alguém, e não terá acesso a nenhuma outra funcionalidade.

O **keystone** é para um ecossistema o que um gerente é para um projeto. É dele a tarefa de criar estratégias de crescimento, ou modificá-las, quando estas não estão surtindo efeito e acompanhar o trabalho dos evangelistas. Ele também fornece as ferramentas e documentações necessárias para que o trabalho seja feito. Por isso, ele possui a visão geral do ecossistema, com as suas informações e gráficos de desempenho, e também pode analisar informações de cada um dos evangelistas pertencentes a ele, além dos desenvolvedores abaixo destes.

O **evangelista** é um papel que possui similaridades com a de *influencer*, visto anteriormente, sendo que a principal diferença é que ele constitui um elemento interno da

organização, provendo a ligação entre esta e os desenvolvedores, ao invés de ser apenas um desenvolvedor engajado no crescimento do ecossistema. Ele participa de treinamentos, palestras e competições de desenvolvimento visando a expansão de um Ecossistema e formação de novos desenvolvedores de aplicativos (FONTÃO et al., 2014). Além disso, é responsável pelo acompanhamento da produtividade de um grupo de publicadores de aplicativos (chamados de desenvolvedores no sistema). Por isso, o acompanhamento do histórico de seus desenvolvedores é importante, para reconhecer a queda de desempenho e, assim, poder agir para modificar esses dados e garantir a saúde do ecossistema. No sistema, ele terá acesso às informações de todos os desenvolvedores pelo qual é responsável e às suas informações de desempenho, baseadas nesses desenvolvedores. Além disso, ele será o responsável pela atualização dos dados referentes ao ecossistema.

O **desenvolvedor** representa os publicadores de aplicativos do ecossistema. São os dados deles e de seus aplicativos que serão a fonte de alimentação e análise de todo o sistema. Porém, como essas informações serão úteis apenas para o evangelista e para o *keystone*, a ele só é permitida a visualização das suas próprias informações e dos aplicativos que ele publicou.

3.4 Informações armazenadas na ferramenta

O objetivo da ferramenta *MSECO Watcher* é a exibição de dados que mostrem o desempenho de um ecossistema que busca a publicação de aplicativos em uma loja de aplicativos móveis, permitindo assim o controle da saúde desse ecossistema.

Para a ferramenta, as métricas utilizadas para a verificação do desempenho de um aplicativo na loja são a sua quantidade de downloads e seu *rating*, pois elas representam a quantidade de usuários que o utilizam e a satisfação destes em relação ao produto. Por isso, essas informações serão armazenadas para cada aplicativo cadastrado no sistema. Além disso, informações básicas como o nome, descrição, categoria, link e ícone destes também serão incluídas para a identificação. Outro dado incluído é a data da última atualização, que representa o dia mais recente em que o desenvolvedor publicou uma nova versão do aplicativo na loja. Essa informação é importante, pois mostra a quantidade de tempo que um aplicativo não recebe atualizações, o que, caso seja um tempo muito longo, pode ser uma das causas do seu mau desempenho.

As informações de quantidade de downloads e *rating* também foram utilizadas para medição de desempenho do desenvolvedor, evangelista e ecossistema. Esses atributos são calculados utilizando como base os dados dos aplicativos. Ou seja, o desenvolvedor armazenará a soma do número de downloads de todos os aplicativos publicados por ele, bem como a média de *rating* desses, assim como o evangelista, que irá armazenar a soma do número de downloads e a média de *rating* de todos os aplicativos publicados pelos desenvolvedores que estão sob a sua responsabilidade, de igual modo acontece com o ecossistema. Atributos que armazenam dados sobre a quantificação do ecossistema também foram inseridos: a quantidade geral de aplicativos, para todos eles; a quantidade de desenvolvedores para o evangelista e o ecossistema; e a quantidade de evangelistas apenas para o ecossistema. As informações de quantidade são relevantes para mostrar o crescimento do ecossistema em relação à criação de aplicativos e atração de novos membros, métricas importantes para a verificação da saúde do ecossistema.

O acompanhamento dos downloads e do *rating* dos aplicativos são métricas para o controle da saúde de ecossistemas apresentadas por Jansen (2014) no nível de projeto, chamada de novos downloads e satisfação e avaliação pelo usuário. Além disso, o acompanhamento na criação de novos aplicativos pelos desenvolvedores pode ser definido como a produtividade apresentada por Iansiti e Levien (2002), já que sua definição é a eficiência com que um

ecossistema transforma tecnologia e outros materiais de inovações em novos produtos, com custos reduzidos. Aplicando a ecossistemas móveis, isso significa dizer que é a eficiência com que os desenvolvedores transformam a tecnologia, que é o sistema operacional móvel, em novos produtos, que são os aplicativos. As últimas métricas utilizadas, que são a frequência do acréscimo de novos desenvolvedores e evangelistas ao ecossistema, podem ser comparadas à métrica de saúde definida por Hartigh et al. (2006) como taxa de sobrevivência, que é a sobrevivência dos atores no decorrer do tempo, aonde os atores são as pessoas que contribuem para o crescimento da plataforma base do ecossistema.

Alguns atributos são comuns a todos os atores do sistema: nome, e-mail principal, e-mail secundário e senha. A principal razão por trás desta similaridade é o *login*, aonde o usuário deve informar o e-mail e a senha. Porém, o desenvolvedor possui alguns atributos exclusivos. O seu *publisher name*, chamado de *nickname* no sistema, representa um nome único que um publicador possui na loja, que aparecerá em todos os seus aplicativos. Algumas de suas informações pessoais e das redes sociais também são atributos do desenvolvedor incluídos no sistema. O número de telefone, o link do *facebook* e o sua conta no *twitter* são informações que podem ser inseridas para esse ator, apesar de não serem obrigatórias.

Para o histórico, as informações de desempenho são armazenadas por data. Sendo assim, para cada nova atualização do evangelista no sistema, os novos valores são armazenados para os aplicativos, desenvolvedores, evangelistas e ecossistemas, como uma nova instância, juntamente com o dia dessa atualização. Através desses dados será possível a criação dos gráficos de desempenho.

Uma última informação armazenada pelo sistema são as referentes à categoria dos aplicativos. Pensando na possível mudança nas categorias existentes nas lojas, foi permitido aos atores o controle desses dados, podendo cadastrar novas categorias, e alterar as já existentes. O nome e a descrição são os atributos armazenados neste caso.

3.5 Funcionalidades


As funcionalidades presentes na ferramenta estão descritas na Tabela 1. É importante ressaltar que para as três últimas funcionalidades da tabela existe uma hierarquia entre os atores, exceto o administrador. Isso significa dizer que se o desenvolvedor é capaz de visualizar os dados e gráficos do seu desempenho e do de seus aplicativos, o evangelista responsável por esse desenvolvedor também é capaz de visualizar essas informações, além das suas próprias e, consequentemente, o *keystone* acima do evangelista consegue visualizar todas as anteriores, bem como as suas próprias informações. Porém, existem funcionalidades exclusivas de cada ator, que não são herdadas pelos que estão acima dele. Como por exemplo, a atualização dos dados do sistema através de um documento, que apenas o evangelista pode realizar. É importante ressaltar que um ator não pode cadastrar outro ator do mesmo tipo, por isso algumas funcionalidades de cadastro são separadas da manutenção na tabela, como é o caso do cadastro de desenvolvedores, só permitida ao evangelista.

3.5.1 Telas dos Atores e exibição de informações de desempenho e gráficos

A ferramenta inicia com uma tela de *Login*, onde o usuário deve identificar seu tipo (*keystone*, evangelista ou desenvolvedor) e inserir um e-mail e senha que já tenham sido cadastrados (Figura 6). Caso o usuário seja o administrador do sistema, ele deverá escolher o tipo *keystone* e inserir um e-mail e senha únicos para o seu ator. De acordo com o tipo de usuário, o sistema pode ser redirecionado para quatro telas distintas.

Tabela 1 - Descrição das Funcionalidades da Ferramenta, aonde *A* é o administrador, *K* é o keystone, *E* é o evangelista e *D* o desenvolvedor

Funcionalidade	Descrição	Atores
Cadastro de keystones	Inserção do keystone do sistema	<i>A</i>
Manutenção de Keystone	Alteração dos dados de um Keystone	<i>K</i>
Cadastro e Manutenção de Ecossistemas	Cadastro e alteração de ecossistemas	<i>K</i>
Cadastro de Evangelistas	Cadastro de novos evangelistas em ecossistemas existentes na ferramenta	<i>K</i>
Consultar Ecossistemas	Listar todos os ecossistemas presentes na ferramenta.	<i>K</i>
Consultar Evangelistas de um Ecossistema	Listagem de todos os evangelistas que fazem parte de um determinado ecossistema.	<i>K</i>
Cadastro de Desenvolvedores	Inserção de novos desenvolvedores no sistema.	<i>E</i>
Cadastro e Manutenção de Aplicativo	Criação e alteração de um aplicativo	<i>E</i>
Cadastro e Manutenção de Categoria	Criação, alteração e exclusão de uma categoria de aplicativo	<i>E</i>
Atualização de dados do sistema através de um documento	Permite utilizar as informações de um documento enviado ao sistema para cadastro de novos desenvolvedores e aplicativos, além da inserção de novos dados no histórico.	<i>E</i>
Consultar Desenvolvedores de um Evangelista	Permite a visualização de todos os desenvolvedores que estão debaixo da responsabilidade de um determinado evangelista.	<i>K, E</i>
Manutenção de Evangelista	Alteração dos dados de um Evangelista	<i>K, E</i>
Manutenção de Desenvolvedor	Alteração dos dados de um Desenvolvedor	<i>E, D</i>
Consultar Aplicativos de um Desenvolvedor	Permite a visualização de todos os aplicativos cadastrados na ferramenta que foram publicados por um determinado desenvolvedor	<i>K, E, D</i>
Atualização de dados do sistema	Ação em que os dados podem ser recontados e atualizados	<i>K, E, D</i>
Visualização de Gráficos de Desempenho	Permite aos atores a visualização dos gráficos de desempenho das informações permitidas a eles	<i>K, E, D</i>
Visualização de Informações referentes ao desempenho	Permite aos atores a visualização de suas próprias informações de desempenho, além das informações dos atores e ecossistemas por que é responsável	<i>K, E, D</i>



Acesso

Selecione a conta que voce deseja logar e insira seus dados:

☒ Keystone
 ☐ Evangelista
 ☐ Desenvolvedor

Email:

Senha:

Figura 6 - Tela de Login

A tela do administrador só lhe permite fazer o cadastro do *keystone* do sistema. Caso ele já tenha sido cadastrado, ele pode alterar seus dados de login. Essa função é importante para a primeira atribuição e para caso ocorra algum imprevisto em relação ao acesso do *keystone*, o único capaz de criar evangelistas e ecossistemas na ferramenta.

A tela do *keystone* lista todos os ecossistemas criados por este exibindo algumas de suas informações, que incluem: nome, descrição, quantidade de evangelistas participantes, quantidade geral de downloads e média de *rating*, sendo que as duas últimas são calculadas com base nos aplicativos deste ecossistema. Para cada um deles é possível a exibição das suas informações detalhadas caso o *keystone* clique no botão Visualizar (Figura 7).

Bem Vindo(a), Keystone					
Alterar Senha Alterar Perfil Log Off					
<input type="button" value="Cadastrar Novo Ecossistema"/>					
 Ecossistemas					
Nome	Descrição	Downloads	Rating	Quantidade de Evangelistas	Visualizar
Google Mobile Ecosystem	Ecossistema Móvel do Google para a plataforma Android	0	0	0	<input type="button" value="Visualizar"/>
Apple Mobile Ecosystem	Ecossistema Móvel da Apple para a plataforma mobile iOS	0	0	0	<input type="button" value="Visualizar"/>
MSECO Microsoft	Microsoft Mobile Ecosystem - Ecossistema Móvel da Microsoft para a plataforma Windows Phone	1103	0.00506038	3	<input type="button" value="Visualizar"/>

Figura 7 - Tela do Keystone

A exibição detalhada de um ecossistema mostra, além das informações já mostradas na tela do *keystone*, as suas quantidades de desenvolvedores e aplicativos. Essa tela também exibe uma tabela com detalhes de todos os evangelistas participantes do ecossistema, contendo nome, e-mail, quantidade de *downloads*, média de *rating* e quantidade de desenvolvedores (Figura 8).

Para cada evangelista exibido, também é possível a visualização de seus dados. Caso o *keystone* clique no botão de detalhamento de algum deles, ele é redirecionado para a tela daquele evangelista. Além dessas opções, também é possível o cadastro de novos evangelistas para esse ecossistema, a alteração dos dados do ecossistema e a exibição dos seus gráficos de desempenho, função que será detalhada mais adiante.



MSECO Microsoft

Microsoft Mobile Ecosystem - Ecossistema Móvel da Microsoft para a plataforma Windows Phone

Downloads: 1103
Rating: 0.00506038
Quantidade de Evangelistas: 3
Quantidade de Desenvolvedores: 8
Quantidade de Aplicativos: 506

Cadastrar Evangelista
 Alterar Dados do Ecossistema
 Gerar Gráficos do Ecossistema

 **Evangelistas**

Nome	E-mail	Downloads	Rating	Quantidade de Devs	Visualizar
Awdren Fontão	awdren@microsoft.com	0	0	0	Visualizar
Maurílio	maurilio@microsoft.com	0	0	0	Visualizar
Jackson	jackson@microsoft.com	1103	0.00506038	8	Visualizar

Figura 8 - Tela de Detalhamento de Ecossistema

A tela de evangelista exibe os dados referentes a ele, além de exibir uma lista dos desenvolvedores que estão sob a sua responsabilidade. Os seus dados exibidos incluem o nome, e-mail principal, e-mail secundário (caso ele tenha cadastrado), suas quantidades de downloads, desenvolvedores e aplicativos, e sua média de *rating* (Figura 9).

Para cada desenvolvedor exibido na lista, é mostrado o nome, *nickname*, os números de download e *rating* e sua quantidade de aplicativos publicados, com a possibilidade de detalhamento para cada um deles. Através de botões exibidos no cabeçalho ainda é fornecido ao evangelista as funcionalidades do cadastro de novos desenvolvedores, de manutenção de categoria, de exibição dos seus gráficos de desempenho, e de atualização e inserção de dados no sistema através do *upload* de um documento, função descrita mais adiante. Das funções destacadas anteriormente, apenas a da geração dos gráficos e exibida ao *keystone* quando ele visualiza esta tela, além da opção de alterar os dados do evangelista. Caso o evangelista expanda as informações de um dos desenvolvedores, a tela daquele desenvolvedor será exibida.

Bem Vindo(a), **Keystone**

Alterar Senha Alterar Perfil Log Off

E-mail Principal: jackson@microsoft.com

Downloads: 1103

Rating: 0.00506038

Quantidade de Devs: 8

Quantidade de Apps: 506

Jackson

Gerar Gráficos de Desempenho



Desenvolvedores

Nome	Nickname	Downloads	Rating	Quantidade de Apps	Visualizar
Rafael Godoi Vasques	Rafael G Vasques	0	0	1	Visualizar
Fabio Massaretto	fmassaretto	0	0.0153061	98	Visualizar
Lucas Cirilo Consentino	lucas cirilo	0	0.0487805	41	Visualizar
Mario Ribeiro	crazy4apps	1100	0.209677	62	Visualizar

Figura 9 - Tela de Evangelista

Assim como as demais, a tela do desenvolvedor exibe os seus dados, permitindo a ele também o acesso aos seus gráficos de desempenho. Alguns de seus atributos sempre serão exibidos, sendo estes: nome, e-mail principal, *nickname*, quantidade de downloads, número de aplicativos publicados e média de *rating* (Figura 10). O e-mail secundário, número de telefone e páginas do *facebook* e *twitter* são atributos optativos, e só serão exibidos caso tenham sido cadastrados.

Bem Vindo(a), **Keystone**

Alterar Senha Alterar Perfil Log Off

Fabio Massaretto

E-mail Principal: fabiomassaretto@outlook.com

Nickname: fmassaretto

Downloads: 0

Rating: 0.0153061

Quantidade de Apps: 98

Gerar Gráficos do Desenvolvedor

Voltar



Aplicativos

Nome	Descrição	Downloads	Rating	Dias sem atualização	Visualizar
VelvetRevolver	Velvet Revolver	0	0	128	Visualizar
MedalofHonor	Noticias, dicas e videos sobre Medal of Honor.	0	0	130	Visualizar
GTA-5	Noticias, dicas e videos sobre Grand Theft Auto V.	0	0	128	Visualizar
Alter-Bridge	Aplicativo com noticias e videos sobre a banda de rock Alter Bridge.	0	0	129	Visualizar

Figura 10 - Tela de Desenvolvedor

A tela também exibe uma lista com todos os aplicativos publicados por ele, com as informações de nome, descrição, número de downloads e *rating*, e o tempo passado desde a sua última atualização, além de um botão que permite a abertura da tela de informações daquele aplicativo. Caso tenha se passado mais de 50 dias desde a atualização, os dias são mostrados em vermelho na lista. Caso seja o evangelista o ator a visualizar a tela do desenvolvedor, as opções para alteração de dados do desenvolvedor e cadastro de um novo aplicativo serão exibidas.

A tela com a exibição de um aplicativo, chamada de Tela de Aplicativo, exibe o seu ícone, nome, descrição, nome do desenvolvedor, categoria, quantidade de downloads e *rating*, tempo desde a última atualização (que também aparece em vermelho caso seja maior que 50 dias) e o *link* para a visualização do aplicativo na loja (Figura 11). Além disso, permite a exibição dos seus gráficos de desempenho de downloads e *rating*. A um evangelista ainda é exibidas a função de alteração dos dados deste aplicativo.

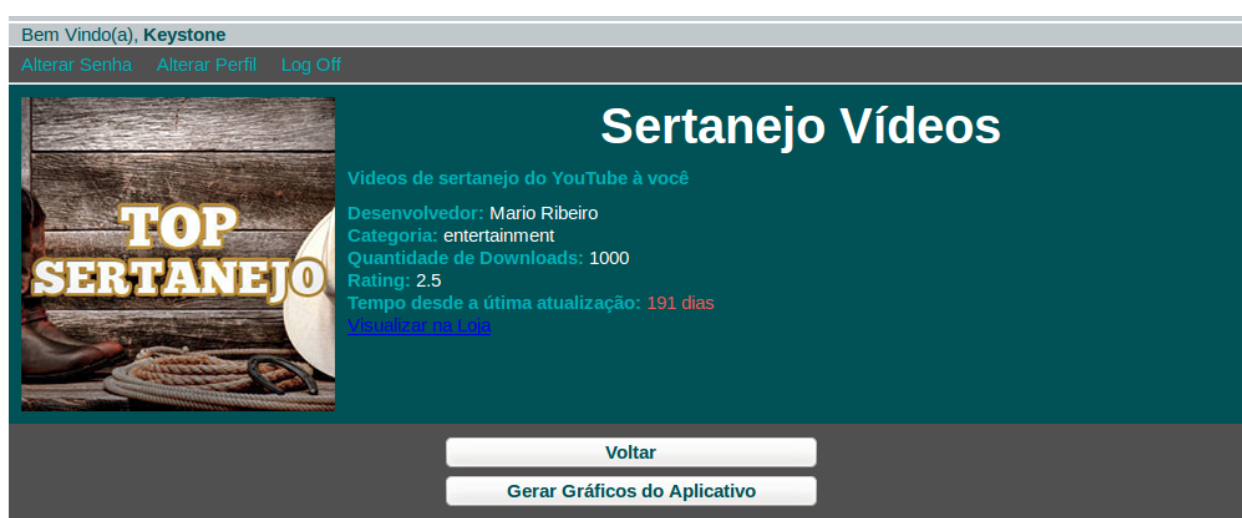


Figura 11 - Tela de Aplicativo

A tela de exibição dos gráficos de desempenho funciona da mesma maneira para cada uma das entradas. A única diferença reside na quantidade de gráficos exibidos, que difere entre o evangelista, desenvolvedor, aplicativo e ecossistema. Os gráficos exibindo o desempenho de downloads e *rating* são mostrados para todos. Já o que exibe a evolução na criação de novos aplicativos é mostrado apenas para os desenvolvedores, evangelistas e ecossistemas, já que é uma informação incompatível para o aplicativo. O evangelista e o ecossistema ainda possuem o gráfico da quantidade de desenvolvedores e o ecossistema possui o da quantidade de evangelistas. A Figura 12 abaixo resume a exibição dos gráficos.

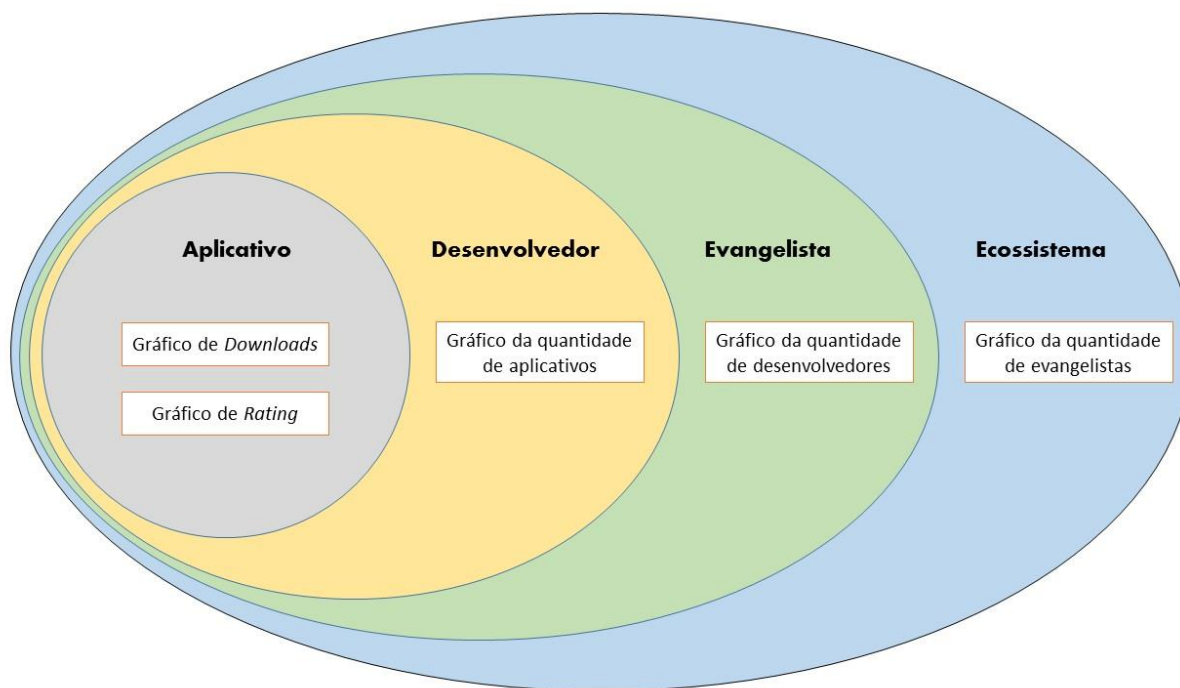


Figura 12 - Conjunto da Exibição dos Gráficos de Desempenho do Sistema

Os gráficos podem ser exibidos de duas maneiras: contendo todos os dados existentes no histórico ou apenas os dados inseridos durante um período de tempo fornecido. O usuário pode escolher qual gráfico deseja gerar na tela exibida na Figura 13 abaixo. Selecionando a opção *Dados Gerais* e clicando em *Gerar Gráficos*, os gráficos exibidos mostrarão todos os dados do histórico. Selecionando a opção *Dados entre um Intervalo*, inserindo as datas inicial e final e apertando no botão *Gerar Gráficos*, os gráficos exibirão apenas os dados contidos no intervalo de tempo fornecido.

Figura 13 - Tela do Tipo de Gráfico

A Figura 14 abaixo mostra os gráficos de desempenho gerado para um ecossistema, que exibe todos os gráficos possíveis da ferramenta.



Figura 14 - Tela de Exibição de Gráfico

3.5.2 Cadastros e manutenção

O *keystone* é o responsável pelo cadastro e manutenção de ecossistemas (Figura 15). Ao criar um novo ecossistema, este já é inserido com o identificador deste ator. Ele também pode alterar as informações básicas de todos os ecossistemas criados.

Bem Vindo(a), **Keystone**
[Alterar Senha](#) [Alterar Perfil](#) [Log Off](#)



Cadastro de Ecosystema

* Nome:

Descrição:

Figura 15 - Tela de Cadastro de Ecosystema

Também cabe ao *keystone* o cadastro de novos evangelistas. No entanto, o evangelista tem a permissão para alterar os seus dados. O mesmo ocorre com o cadastro do desenvolvedor: o evangelista pode cadastrá-lo manualmente, preenchendo o formulário da Figura 16, ou o sistema o cadastra automaticamente, através do *upload* de um documento com os dados, mas o desenvolvedor pode alterar os seus dados pessoais.



Cadastro de Desenvolvedor

* Nome:

* Nickname:

* Email Principal:

Email Secundário:

Telefone:

Facebook:

Twitter:

Figura 16 - Tela de Cadastro de Desenvolvedor

Vale ressaltar que qualquer informação referente ao desempenho, exceto a de aplicativos, não pode ser alterada por um ator, já que são calculadas automaticamente com base nos dados dos aplicativos.

Uma das responsabilidades do evangelista consiste no cadastro e manutenção das categorias dos aplicativos. Ele pode alterar os seus dados ou excluir, caso esta não esteja relacionada a nenhum aplicativo cadastrado. O sistema também pode cadastrar as categorias automaticamente através da atualização por um documento.

Outra funcionalidade permitida apenas ao evangelista é o cadastro e manutenção de aplicativos, e este é o único caso aonde um ator pode inserir e atualizar dados relacionados ao desempenho, como *rating* e número de *downloads* (Figura 17). Após uma nova inserção ou atualização de um aplicativo, todas as informações de desempenho relacionadas a ele serão atualizadas: do desenvolvedor, do evangelista e do ecossistema.



Figura 17 - Tela de Cadastro de Aplicativo

Excetuando o administrador, todos os atores podem alterar os seus dados pessoais e a sua senha. Estas opções são exibidas em todas as telas do sistema para facilitar o seu alcance. A tela de mudança de senha, descrita na Figura 18, valida a atualização através da inserção da senha atual e da repetição da nova senha.



Figura 18 - Tela de Alterar Senha

3.5.3 Atualização de dados do sistema através de um documento externo

A funcionalidade de atualização de dados através de um documento externo representa a base de atualização de dados do sistema, pois através dela os dados dos aplicativos, e consequentemente os dados do resto do ecossistema, serão mantidos e será possível o monitoramento e acompanhamento por meio dos gráficos que descrevem o histórico dos dados.


Ela consiste na atualização e criação de informações para o banco de dados através da leitura de um documento carregado pelo evangelista. O documento é uma planilha no formato CSV que é gerada automaticamente através do *Algoritmo de Varredura*, mas também pode ser construída manualmente pelo evangelista. Seu formato é apresentado na Figura 19, que contém as colunas: *Name*, *DVLUP Name* e *E-mail*, que representam o nome, *publisher name* e e-mail do publicador, respectivamente; *App Link*, *App Name*, *Star Rating*, *Downloads*, *Updated*, *Category*, *Description* e *Image URL*; que representam o *link*, nome, *rating*, quantidade de *downloads*, data da última atualização, categoria, descrição e *link* da imagem, respectivamente, referentes ao aplicativo.

Name	DVLUP Name	E-mail	App Link	App Name	Star Rating	Downloads	Updated	Category	Description	Image URL
Alan Webber	alanalves	alan_webber.alves@hotmail.com								
Alessandro M	alessandropino	alessandro@	http://www.wi	Guia Campeõe	3	0	07/28/2014	entertain	Guia dos cam	http://cdn.m
Alexsandro T	alextrindade	alextrindad	http://www.wi	CantorCristao	4.2	0	07/31/2014	music	vic Hinário Cant	http://cdn.m
Alexsandro T	alextrindade	alextrindad	http://www.wi	Emergência	0	0	08/21/2014	health	fii Tenha de mai	http://cdn.m
Alexsandro T	alextrindade	alextrindad	http://www.wi	Harpa Cristã Ev	4.6	0	08/22/2014	music	vic Hinário Harp	http://cdn.m
Alexsandro T	alextrindade	alextrindad	http://www.wi	PreparaEnem	5	0	09/24/2014	education	Sabe aquele i	http://cdn.m
Allan Monte	allan_mlpe	allan_mlpe	http://www.wi	Clube do Hard	0	0	09/26/2014	entertain	Fique por der	http://cdn.m
Allan Monte	allan_mlpe	allan_mlpe@hotmail.com								
Allan Monte	allan_mlpe	allan_mlpe	http://www.wi	JavaScript Fáci	0	0	09/26/2014	education	Com os video	http://cdn.m
Allan Monte	allan_mlpe	allan_mlpe	http://www.wi	BestMovies	0	0	09/26/2014	entertain	Confira a play	http://cdn.m
Allan Monte	allan_mlpe	allan_mlpe	http://www.wi	Santa Cruz FC	0	0	09/26/2014	sports	Torce para o	http://cdn.m
Allan Monte	allan_mlpe	allan_mlpe	http://www.wi	TV Coral	0	0	09/26/2014	sports	Fique por der	http://cdn.m
Allan Monte	allan_mlpe	allan_mlpe	http://www.wi	RoupaApp	0	0	09/26/2014	music	vic Confira na tel	http://cdn.m
Allan Monte	allan_mlpe	allan_mlpe	http://www.wi	Skank app	0	0	09/26/2014	music	vic Curta o som c	http://cdn.m
Allan Monte	allan_mlpe	allan_mlpe	http://www.wi	MundoCanibal	0	0	09/26/2014	entertain	Dê muitas gai	http://cdn.m
Allan Monte	allan_mlpe	allan_mlpe	http://www.wi	ParalamasApp	0	0	09/27/2014	music	vic Confira o con	http://cdn.m
Ana Flávia Re	anaflavia	aflaviarm@	http://www.wi	UnpCalc	3	0	07/25/2014	education	Aplicativo pa	http://cdn.m
Ana Lúcia T. d	analuciasilva	ana.silva@c	http://www.wi	Minha Banda F	0	0	10/24/2014	music	vic Aplicativo fei	http://cdn.m
Ana Lúcia T. d	analuciasilva	ana.silva@c	http://www.wi	Mundo da Foto	0	0	10/23/2014	photo	Aplicativo fei	http://cdn.m
Ana Lúcia T. d	analuciasilva	ana.silva@c	http://www.wi	My Photos	0	0	10/23/2014	photo	Application n	http://cdn.m
Ana Lúcia T. d	analuciasilva	ana.silva@c	http://www.wi	QuatroOperac	0	0	10/25/2014	personal	f Aplicativo fei	http://cdn.m
Ana Lúcia T. d	analuciasilva	ana.silva@c	http://www.wi	TurismoBR	0	0	10/22/2014	travel	na Aplicativo fei	http://cdn.m
Anchiêta Aca	protonsnet	protonsnet	http://www.wi	Space Adventu	0	0	08/23/2014	games	Shooting gam	http://cdn.m
Anderson Bar	ablma	lpsolidsnake	http://www.wi	Mecha Invasio	4	0	07/31/2014	entertain	The Kingdom	http://cdn.m
Anderson Bar	ablma	lpsolidsnake	http://www.wi	IronMaiden Qu	3	0	10/19/2013	entertain	Are you a har	http://cdn.m
Anderson Mo	anderson_guit	anderson_g	http://www.wi	Cauê Moura St	0	0	09/07/2014	entertain	Aplicativo de	http://cdn.m
Anderson Mo	anderson_guit	anderson_g	http://www.wi	Guit Videos	0	0	08/25/2014	music	vic Um canal par	http://cdn.m
Anderson Mo	anderson_guit	anderson_g	http://www.wi	Stainki App	0	0	08/25/2014	business	O primeiro A	http://cdn.m
Anderson Mo	anderson_guit	anderson_g	http://www.wi	Wakko Purossi	0	0	09/03/2014	entertain	CHUAAA!!! A	http://cdn.m

Figura 19 - Planilha CSV utilizada para o recolhimento de dados

O evangelista carrega o documento para o sistema, através da tela da Figura 20, fornecendo a data referente à construção do documento para que esta seja armazenada. O sistema, por sua vez, faz a leitura do nome, e-mail e *nickname* (ou *publisher name*), referentes ao desenvolvedor; e nome, *rating*, categoria, data da última atualização, número de downloads, link do aplicativo, descrição e link da imagem do aplicativo, referentes ao aplicativo. Para cada linha do documento, a leitura é realizada e são feitas verificações antes do cadastro ou atualização dos dados.

Alterar Senha
Alterar Perfil
Log Off



Upload de Documento

Selecione a data do Documento:

Escolha o arquivo:

Selecione arquivo...
Nenhum arquivo selecionado.

Upload

Voltar

Figura 20 - Tela de Upload de Documento

Utilizando o *nickname* do desenvolvedor como parâmetro, devido à sua unicidade, a primeira verificação é se este já está cadastrado no sistema. Caso não esteja, um novo usuário é criado com as informações deste desenvolvedor, inserindo apenas as informações obrigatórias, que são: nome, *nickname*, e-mail e senha, que recebe o valor do *nickname*. Após isso, o identificador do usuário é recuperado para que ele possa ser relacionado com o aplicativo.

Na sequência, o sistema verifica se o aplicativo já está cadastrado, utilizando o seu nome como parâmetro. Igualmente ao que ocorre ao desenvolvedor, caso o aplicativo não exista, ele é criado preenchendo todas as informações recuperadas, inclusive o identificador do desenvolvedor. Caso contrário, as informações do aplicativo são atualizadas. Para cada novo aplicativo lido pelo sistema, os dados do desenvolvedor são atualizados e após todas as instâncias terem sido lidas o sistema atualiza os dados do evangelista e do ecossistema.

É importante ressaltar que todas as informações, criadas ou atualizadas, são inseridas nos históricos do desenvolvedor, aplicativo, evangelista e ecossistema, para que possam ser usadas para exibição dos gráficos de desempenho.

4. Estudo de Caso

Através do algoritmo de varredura, foram colhidas informações de desenvolvedores reais que publicaram aplicativos na WPS e estavam sendo acompanhados pelos evangelistas do Ecossistema Móvel da Microsoft.

O algoritmo recebe como entrada o documento no formato CSV, mostrado na Figura 21. O documento exibe três colunas: *Name*, com o nome dos desenvolvedores que se deseja recuperar os aplicativos da loja; o *App Link*, listando o *link* de um aplicativo publicado pelo desenvolvedor da coluna anterior; e *Name of the App*, com os nomes dos aplicativos cujos links estão na coluna anterior.

Name	App Link	Name of the App
Mario Ribeiro	http://www.windowsphone	Sertanejo Vídeos
Fabio Massaretto	http://www.windowsphone	VelvetRevolver
Fabricio Aparecido Messias	http://www.windowsphone	Ultimo Lance
Lucas Cirilo Consentino	http://www.windowsphone	Coringão
Lucas Gabriel Chiaretti	http://www.windowsphone	Respostas
Lucas Pereira dos santos	http://www.windowsphone	acer
Rafael Godoi Vasques	http://www.windowsphone	Divisão
Renan Donha S. de Almeida	http://www.windowsphone	Folha Ambiente

Figura 21 - Documento CSV de entrada para o algoritmo de varredura

O algoritmo vai até a página fornecida pelo *link* recuperado da coluna *App Link* do documento, armazena as informações deste aplicativo e guarda o *publisher name* do desenvolvedor e a quantidade de páginas de aplicativos que este desenvolvedor possui. A WPS, ao listar os aplicativos de um desenvolvedor, os exibe em páginas, sendo 48 aplicativos por página. A importância do armazenamento do número de páginas consiste em saber quando parar de percorrer a loja em busca dos aplicativos do desenvolvedor.

Após isso, ele percorre cada uma das páginas e armazena as informações de cada aplicativo encontrado. Ao fim, armazena todas essas informações em um documento CSV que é exportado para o mesmo diretório aonde o algoritmo se encontra. A única informação que o algoritmo não consegue recuperar da loja é a quantidade de downloads, já que ela é exibida

exclusivamente para o publicador⁷. Sendo assim, estes dados são fornecidos ao evangelista pelo próprio desenvolvedor, para que este atualize o documento.

Com a impossibilidade de contato com os publicadores utilizados como base para este experimento, não foi possível a atualização das informações de downloads do documento com as informações reais. Sendo assim, outros valores foram inseridos visando exibir dois casos cuja análise pode representar uma informação valiosa para o evangelista: quando um aplicativo recebe uma grande quantidade de downloads inesperadamente e quando um aplicativo estanca após um determinado número de downloads. Além disso, para simular as informações referentes à quantidade de aplicativos, os aplicativos de alguns desenvolvedores foram sendo incluídos gradativamente em cada *upload* de documento.

Diariamente, durante cinco dias, o algoritmo foi rodado e cada um dos cinco documentos gerados foi guardado. Após isso, as modificações ditas anteriormente foram feitas. Tendo os documentos prontos, eles foram carregados para a ferramenta variando do período do primeiro até o quinto dia de fevereiro de 2015. Após isso, os gráficos de desempenho da ferramenta foram exibidos, filtrando as informações no intervalo de tempo de um a cinco de fevereiro de 2015.

4.1 Análise dos Gráficos de Desempenho

Para a análise dos resultados referentes aos dados de desempenho, foram recolhidos os Gráficos de *Rating* e Gráficos de Downloads de dois aplicativos inseridos na ferramenta através do upload dos arquivos descrito anteriormente. Além disso, os Gráficos de Quantidade de Aplicativos de dois desenvolvedores, também inseridos pelo processo e o Gráfico de Quantidade de Desenvolvedores para o evangelista que realizou a inserção dos dados.

Os gráficos exibidos na Figura 22 são referentes a um aplicativo e mostram o desempenho deste em relação ao rating, através do Gráfico de *Rating* do lado esquerdo e em relação à quantidade de downloads, com o Gráfico de Downloads, do lado direito.

Ao analisar o primeiro gráfico, pode-se observar uma constante queda na avaliação de *rating* do aplicativo, e um crescimento brusco no número de downloads. A queda na avaliação pode indicar que os usuários não têm se agradado com o aplicativo, e pode alertar o evangelista ou o próprio desenvolvedor para a procura da causa deste desagrado do público, que pode ser uma nova versão do aplicativo com muitos erros, por exemplo.

Já o brusco crescimento do número de downloads leva ao questionamento do que pode ter ocorrido durante a quarta e a quinta entrada, para que esse aplicativo tivesse uma evolução tão significativa. Caso algo tenha ocorrido, como a divulgação do aplicativo em um site, isto pode fornecer ao evangelista novas técnicas para ajudar no crescimento de downloads dos aplicativos de seus desenvolvedores, contribuindo para o ecossistema como um todo.

A análise dos dois gráficos em conjunto, por sua vez, pode revelar que, apesar do grande número de downloads, o aplicativo não está agradando aos usuários, o que pode prever uma interrupção no crescimento desse aplicativo caso algo não seja feito.

⁷ A ausência da quantidade de downloads na página dos aplicativos não é uma exclusividade da *Windows Phone Store*. Outras lojas, como a *App Store* da *Apple*, também não exibem essa informação. Apenas o *Google Play* exibe algo relacionado ao download, que consiste em um intervalo em que o aplicativo está, como de 10 a 50 mil downloads, por exemplo.

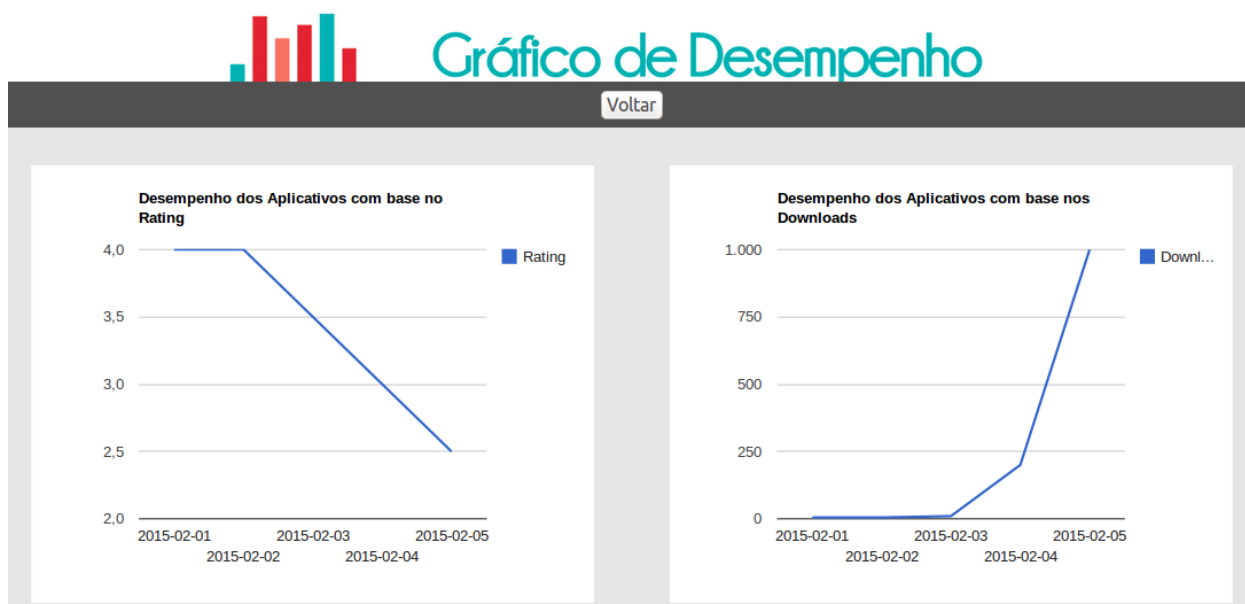


Figura 22 - Gráficos de Desempenho de um aplicativo que recebeu uma grande quantidade de downloads

A Figura 23 exibe os gráficos de desempenho de outro aplicativo e mostram o desempenho deste em relação ao rating, através do Gráfico de *Rating* do lado esquerdo e em relação à quantidade de downloads, com o Gráfico de Downloads do lado direito.

O Gráfico de *Rating* mostra que houve uma evolução muito grande na avaliação dos usuários sobre o aplicativo, pois na primeira entrada o seu *rating* era 3 e na última passou a ser 4.5, saindo de uma nota mediana para uma que fica apenas meia estrela longe da nota máxima.

Em relação aos downloads, por outro lado, este aplicativo tem-se mostrado estagnado. Observando o gráfico, pode-se notar que ele cresceu entre as três primeiras entradas, mas a partir de então não apresentou mais evolução. Deve-se procurar a causa dessa estagnação para que se possam tomar iniciativas visando a continuidade do seu crescimento. A ausência de atualizações recentes pode ser uma causa para esta situação.

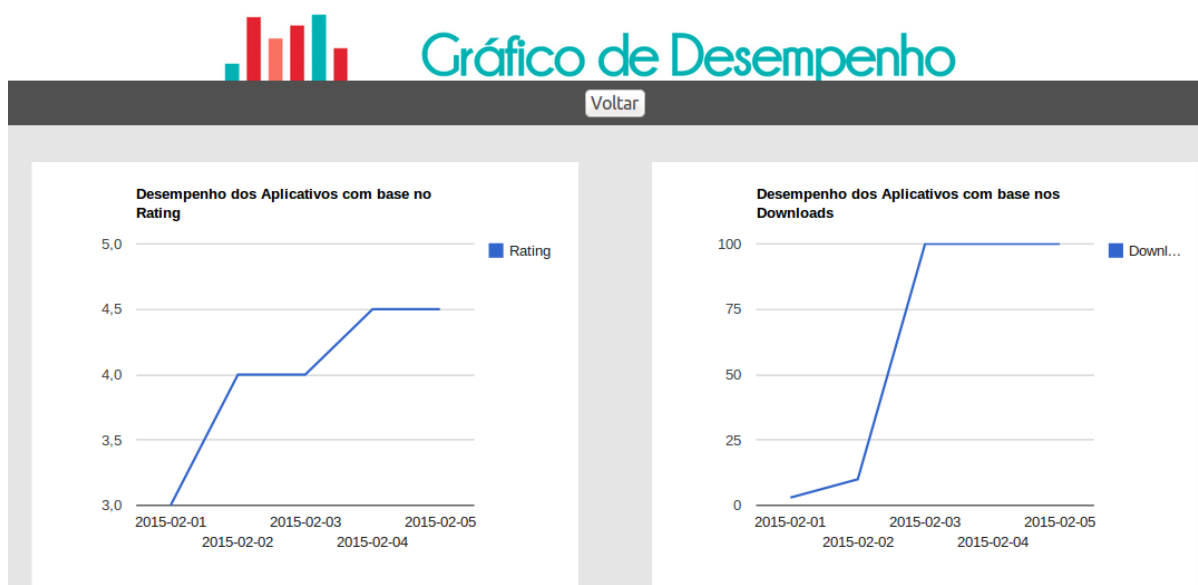


Figura 23 - Gráficos de Desempenho de um aplicativo que parou de crescer

O gráfico da Figura 24 mostra o desempenho de um desenvolvedor na criação de novos aplicativos. Analisando este gráfico, pode-se verificar um constante crescimento na publicação de novos aplicativos por esse publicador, não somente pelo fato da curva ser crescente, mas também pela razão de acréscimo de aplicativos entre as entradas crescer constantemente, ou seja, o número de aplicativos acrescentado entre a primeira e a segunda data é menor do que o adicionado entre a segunda e a terceira, e o aplicativo entre a segunda e terceira data é menor que o entre a terceira e a quarta. Isso mostra que este desenvolvedor tem publicado cada vez mais. Sendo assim, as técnicas de programação e publicação dele podem ser incorporadas ao ecossistema, aumentando a produtividade dos demais.

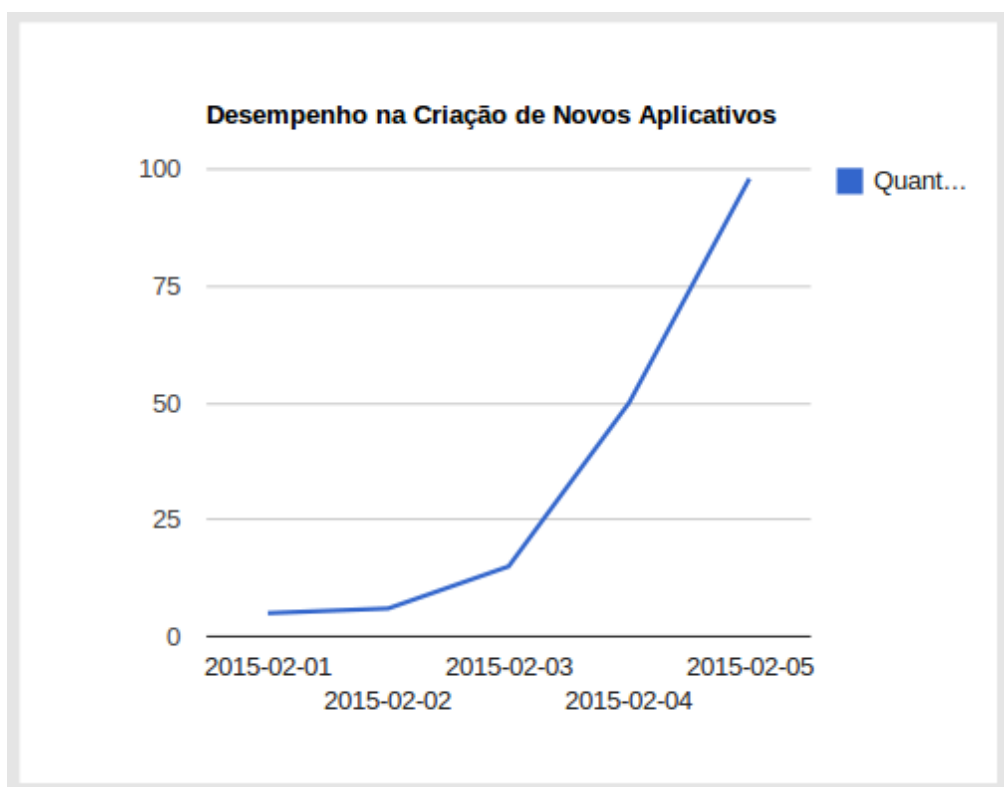


Figura 24 – Gráfico de Desempenho de um desenvolvedor que evoluiu na publicação de novos aplicativos

O gráfico exibido na Figura 25 também exibe as informações relacionadas à quantidade de aplicativos publicados por um desenvolvedor durante o período fornecido. Algo que pode ser percebido através da imagem é que o desenvolvedor parou de publicar aplicativos depois que ultrapassou o número de 40 publicações. Os motivos para essa parada podem ser diversos, dentre eles pessoais, mas também pode ser a ausência de motivação ou de ferramentas. Se esse for o caso, cabe ao evangelista realizar um melhor acompanhamento desta pessoa, através do fornecimento dos itens ausentes ou de estratégias de motivação.

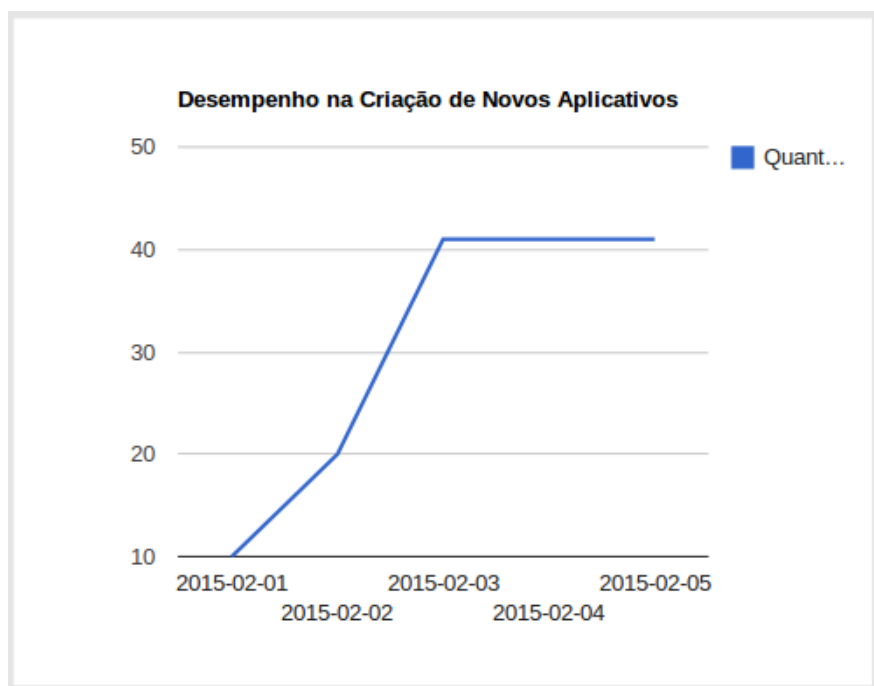


Figura 25 – Gráfico de Desempenho para um desenvolvedor que parou de publicar novos aplicativos

Os gráficos exibidos na Figura 26 são referentes ao desempenho de um evangelista e mostram a frequência de criação de novos aplicativos pelos seus desenvolvedores, no gráfico da esquerda, e a frequência de acréscimo de novos desenvolvedores através desse evangelista, no gráfico da direita.

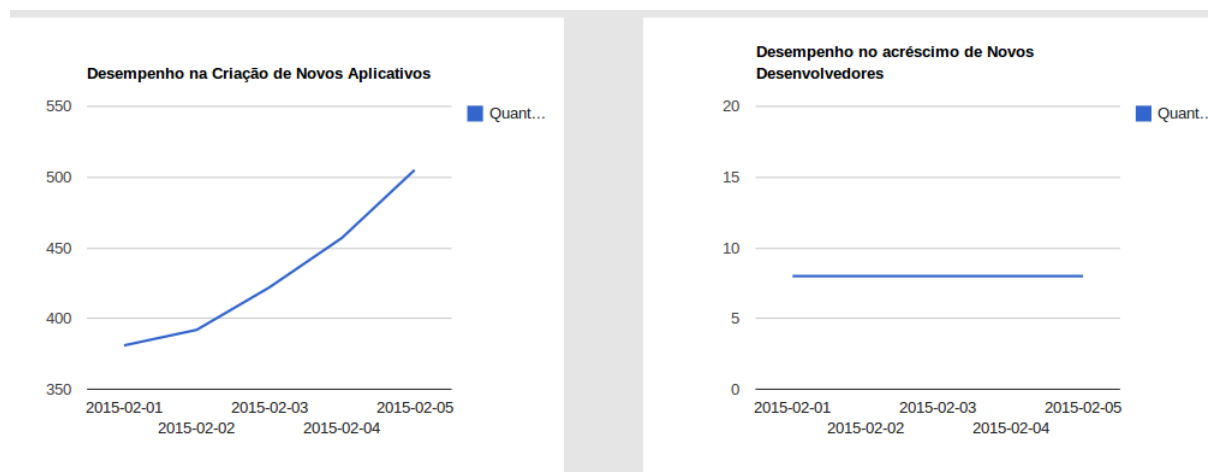


Figura 26 - Gráfico de Desempenho de um evangelista relacionado ao número de desenvolvedores acrescentados ao ecossistema

O primeiro gráfico mostra uma constante evolução na criação de novos aplicativos por parte dos desenvolvedores sob a responsabilidade desse evangelista. Porém, o segundo gráfico mostra que o evangelista não acrescentou nenhum novo desenvolvedor ao ecossistema durante o período exibido. Isso não é necessariamente ruim, já que nenhum deles foi perdido, mas a ausência de atribuição de novas pessoas no ecossistema pode ser um fator contrário à saúde do ecossistema. Jansen (2014) afirma que o número de contribuintes ativos é uma das métricas relacionadas aos contribuintes mais importantes para a saúde de um ecossistema. E no ecossistema móvel relatado neste trabalho, os principais contribuintes são os desenvolvedores

que irão contribuir diretamente para a publicação de aplicativos nas lojas. Dito isso, acrescentar novos desenvolvedores ao ecossistema, ou seja, aumentar o número de contribuintes ativos contribui para a saúde deste ecossistema.

Após a análise destes gráficos, pode-se observar que a interpretação das informações fornecidas por eles pode contribuir bastante no acompanhamento da saúde de um ecossistema móvel, por meio da detecção de problemas que estejam ocorrendo e que possam estar atrapalhando a produtividade ou mesmo a previsão daqueles que podem ocorrer. Ela também possibilita a identificação dos contribuintes que têm se destacado e a possibilidade do compartilhamento para o ecossistema das técnicas e informações utilizadas por ele.

5. Considerações Finais

Através da utilização da ferramenta *MSECO Watcher*, espera-se otimizar os processos de acompanhamento da saúde dos ecossistemas móveis em geral, pois apesar dela ser baseada em um sistema operacional específico, pode ser facilmente utilizada em ecossistemas que utilizem outra plataforma mas que mantenham a mesma estrutura de monitoramento, com *keystone*, evangelista e desenvolvedor.

Como trabalhos futuros, pode-se destacar a incorporação do algoritmo de varredura à ferramenta, permitindo que ele busque as informações sem a necessidade de *upload* de um arquivo externo, além do acréscimo de outras funcionalidades que auxiliem os evangelistas na sua tarefa: como a exibição dos resultados de treinamentos e cursos fornecidos por eles, que podem ser a quantidade de aplicativos gerados e o número de desenvolvedores ativos que foram acrescentados ao ecossistema por meio desses eventos; um sistema de lembretes que avise ao evangelista quando um desenvolvedor ou um aplicativo estão com o desempenho muito baixo, através de limites definidas por eles; e a inserção da função de exclusão dos dados com um tratamento para aquelas informações que só deseja-se esconder para que não participem na contabilização dos dados de desempenho.

6. Referências

- BARBOSA, O.; ALVES, C. A Systematic Mapping Study on Software Ecosystems. In: Proc. of the 3rd IWSECO, 2nd ICSOB, Brussels, Belgium, 15-26, 2011.
- BENTLEY, P.J.: How to be a Successful App Developer: Lessons from the Simulation of an App Ecosystem. GECCO '12 Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference. pp. 129–136. ACM New York, NY, USA, 2012.
- BERK van den, JANSEN I., LUINENBURG S. Software ecosystems: a software ecosystem strategy assessment model. In: Proceedings of the Fourth European Conference on Software Architecture: Companion Volume. ECSA '10, New York, 2010, 127–134.
- BOSCH, J., From Software Product Lines to Software Ecosystem. In: Proc. of 13th SPLC, San Francisco, USA, 1-10, 2009.
- CLEMENTS, P.; NORTHROP, L. Software Product Lines: Practices and Patterns. 3 ed. Boston: Addison-Wesley, 2001. 608p. ISBN: 0-201-70332-7.
- CMMI PRODUCT TEAM. CMMI for Services - Version 1.3, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, Technical Report CMU/SEI-2010-TR-034, 2010.
- COSTANZA, R. Toward an operational definition of ecosystem health. Ecosystem health: New goals for environmental management, 1992, 239–256.

- FONTÃO, A., BONIFÁCIO B., DIAS-NETO A., et al. MSECO Skill: Construção de Competências de Desenvolvedores em Ecossistemas de Software Móvel. In: XVII Iberoamerican Conference on Software Engineering, Pucón, Chile, 2014.
- HARTIGH, E. D., TOL M., VISSCHER W. The health measurement of a business ecosystem. In: Proceedings of the European Network on Chaos and Complexity Research and Management Practice Meeting, 2006.
- IANSITI, M., LEVIEN, R. Keystones and Dominators: Framing the operational dynamics of business ecosystem; Working Paper, 2002.
- IANSITI, M., R. LEVIEN. Strategy as Ecology, In: Harvard Business Review. 2004.
- J. H. III, J. S. BROWN, AND L. DAVISON. Shaping strategy in a world of constant disruption. Harvard Business Review, 2008.
- JANSEN, S., BRINKKEMPER, S., FINKELSTEIN, A.: Business Network Management as a Survival Strategy: A Tale of Two Software Ecosystems, In: Proceedings of the 1st Workshop on Software Ecosystems, Virginia, USA, 2009.
- JANSEN, S. Measuring the Health of Open Source Software Ecosystems: Moving Beyond the Scope of Project Health, In: Information and Software Technology Journal, special issue on Software Ecosystems, 2014.
- LIN, F.L.F., YE, W.Y.W.: Operating System Battle in the Ecosystem of Smartphone Industry. 2009 Int. Symp. Inf. Eng. Electron. Commer, 2009.
- MANIKAS, K., HANSEN, K. M. Reviewing the health of software ecosystems: A conceptual framework proposal. In: Proceedings of the International Workshop on Software Ecosystems, 2013.
- MOORE, J.F. Predators and Prey: A New Ecology of Competition. Harvard Business Review, 1993.
- POSTGRESQL. pgAdmin PostgreSQL Tools. Disponível em: <
<http://www.pgadmin.org/index.php>> Acesso em: 15 fev. 2015.
- SANTOS, R. et al. Software Ecosystems: Trends and Impacts on Software Engineering. Holanda, 2006.