



CENTRO UNIVERSITÁRIO DE BRASÍLIA – UnICEUB
PROGRAMA DE INICIAÇÃO CIENTÍFICA

GIUGLIANO SANTOS SEVERO

**UMA ANÁLISE DO IMPACTO DAS EMPRESAS QUE ATUAM EM ECOSISTEMAS
DE SOFTWARE**

BRASÍLIA

2018



GIUGLIANO SANTOS SEVERO

**UMA ANÁLISE DO IMPACTO DAS EMPRESAS QUE ATUAM EM ECOSISTEMAS
DE SOFTWARE**

Relatório final de pesquisa de Iniciação Científica
apresentado à Assessoria de Pós-Graduação e
Pesquisa.

Orientação: Rodolfo Vasconcelos Correia Lima de
Andrade

BRASÍLIA

2018

AGRADECIMENTOS

Primeiramente agradeço a Deus que permitiu que este momento fosse vivido por mim, trazendo alegria aos meus pais e a todos que contribuíram para a realização deste trabalho.

A esta universidade e todo seu corpo docente, além da direção e a administração, que realizam seu trabalho com tanto amor e dedicação, trabalhando incansavelmente para que nós, alunos, possamos contar com um ensino de extrema qualidade.

Agradeço a este meu orientador, Rodolfo Vasconcelos Correia Lima de Andrade pela paciência, dedicação e ensinamentos que possibilitaram que eu realizasse este trabalho.

Agradeço de forma especial ao meu pai Welliton e à minha mãe Maria Goretti, por não medirem esforços para que eu pudesse levar meus estudos adiante.

Agradeço minha namorada Victória pela paciência e companheirismo durante todo o período da graduação e pesquisa, por ter me ouvido falar inúmeras vezes sobre o tema desta pesquisa.

“Pense diferente: as pessoas loucas o suficiente para achar que podem mudar o mundo são as que, de fato, mudam”

(Steve Jobs)

UMA ANÁLISE DO IMPACTO DAS EMPRESAS QUE ATUAM EM ECOSISTEMAS DE SOFTWARE

Giugliano Santo Severo – UniCEUB, PIC Institucional, aluno bolsista
giusevero@hotmail.com

Rodolfo Vasconcelos Correia Lima de Andrade – UniCEUB, professor orientador
rodolfo.andrade@ceub.edu.br

Ecosistema de software (ECOS) é um conceito que representa a integração de softwares funcionando de forma conjunta, além das ações integradas dos atores envolvidos no processo. Estes atores são empresas parceiras ou até mesmo desenvolvedores independentes. Aplicações para usuário final normalmente são desenvolvidas utilizando softwares específicos que são conhecidos como Ambientes de Desenvolvimento Integrado, no inglês Integrated Development Environment (IDE), que são softwares que constroem outros softwares. Para desenvolvedores de aplicações para dispositivos móveis uma IDE é utilizada na construção como ferramenta principal, unificando design e código de funcionamento de uma plataforma em um único software. Este trabalho analisa como essas IDEs e seus novos recursos influenciam o ciclo de vida do ecossistema durante o tempo em que se mantém no mercado até sua renovação, considerando a qualidade dos recursos que o desenvolvedor possa utilizar de maneira que aumente quantidade de parceiros do ecossistema e impacte em todas as fases do ciclo de vida. Foram selecionados os ecossistemas do iPhone, tendo como sistema operacional o iOS, e do Windows Phone, pois são ecossistemas conhecidos e suas IDEs estão no mercado há bastante tempo como ferramentas de desenvolvimento para suas plataformas base, o MacOS e o Windows.

Palavras-Chave: Ecosistema de software, iOS, Windows Phone, IDE.

LISTA DE FIGURAS

Figura 1 - Quota de mercado dos sistemas operacionais móveis de 2010 - 2017	21
---	----

LISTA DE ABREVIACOES

Apps - Mobile Applications	12
DS-SECO - Dynamic Strategies for Software Ecosystems	11
IDE - Integrated Development Environment.....	5
SDK - Software Development Kit	12
SECO - Software Ecosystem	11

SUMÁRIO

INTRODUÇÃO.....	9
FUNDAMENTAÇÃO TEÓRICA	10
METODOLOGIA	16
RESULTADOS E DISCUSSÃO	24
CONSIDERAÇÕES FINAIS	25
REFERÊNCIAS	26

INTRODUÇÃO

Com a evolução da indústria de software, foi observado um fenômeno de interdependência entre os mesmos, denominado Ecosistema de Software, em que grandes companhias provém ferramentas para a participação de terceiros em seus sistemas para que triunfem sobre os ecossistemas concorrentes. Inspirado sobre a noção de Ecossistemas naturais, os Ecossistemas de software também parte do princípio que seus organismos constituintes possuem relação uns com os outros de maneira hierárquica.

Muito se tem discutido sobre a composição de um Ecosistema de software, as estratégias de manutenibilidade e relação entre os envolvidos como visto em [7], [4] e [6], a eficácia de tais propostas não foram constatadas, apenas observadas, sem uma métrica de qualidade exibida, [10] afirma que há ainda muita pesquisa a ser feita no que se trata de qualidade dentro do ecossistema de software.

Este trabalho visa ir além das estratégias observadas e propostas e buscar entender como as ferramentas que possibilitam os terceiros a produzirem aplicações que integram o ecossistemas impactam a qualidade e sustentabilidade do ecossistema tornando-o mais ou menos atrativos para desenvolvedores e usuários.

FUNDAMENTAÇÃO TEÓRICA

Ecosistema natural

Ecosistemas de software ou Software Ecosystems (SECOs) possuem grande importância na indústria de software, como um conceito que abrange as complexas relações entre as múltiplas companhias que produzem software e as distribuem.

Ecosistemas de software são primariamente baseadas no conceito biológico de ecossistemas naturais, de acordo com [1] há similaridades entre ambos ecossistemas naturais e de software a respeito de como eles trabalham com seus participantes internos e como o ambiente define ou é definido pelos organismos que interagem um com o outro dentro do ecossistema. A noção de recursos básicos como água e comida em um ecossistema natural são tempo, código, arquitetura, etc para um ecossistema de software.

As interações entre atores em um ecossistema natural como simbiose podem ser vistas de uma maneira similar como a colaboração entre as companhias dentro de um ecossistema de software, mas com algumas ressalvas em relação aos atores do SECO, as empresas podem colaborar em um ecossistema de software elas podem ser concorrentes no mercado, como por exemplo: Facebook and Twitter são ambas plataformas de rede social e ambas produzem aplicativos para o ecossistema do iPhone. Os participantes migrando de e para um ecossistema, assim como o ciclo de vida de um ecossistema são conceitos intrínsecos para ambos ecossistemas naturais e de software. Em [2], é possível inferir que há fortes similaridades entre ecossistemas naturais e de software mas com diferenças entre como o ambiente lida com os participantes e como eles agem.

Ecosistemas de negócios

A segunda abordagem é que um SECO é baseado em ecossistema de negócios. Podemos ver que [3] faz uso do conceito de ecossistemas naturais mas com mais definições para o mundo dos negócios, assim dizendo que não há mais empresas que crescem sozinhas no mercado, significando que há uma comunidade envolvendo companhias que interagem uma com as outras e com indivíduos que provém produtos e serviços para os consumidores que são parte do ecossistemas. Em [3] nós também podemos ver uma explicação provida como um ecossistema de negócios trabalha estágios evolucionários, que são: Nascimento,

Expansão, Liderança e Auto-renovação. Esses quatro estágios compreendem o ciclo de vida do ecossistema de negócios.

O estágio de Nascimento começa com a produção de um novo produto ou serviço que consumidores querem e como vão atingir o público, provido por uma companhia, que com a prosperidade de um ecossistema, irá liderá-lo no início, mas logo outros jogadores irão integrar o ecossistema. Expansão foca em como o produto irá atingir novos mercados ou substituir outros produtos no mercado, tudo isso através das relações entre os jogadores de nicho e o líder de plataforma. Liderança significa que o produto que atingiu o mercado e mostrou sucesso e rentabilidade, liderando em primeiro lugar, podem encontrar disputas por outros ecossistemas e conflitos internos por parte dos jogadores de nicho que integram seu ecossistema, então como o lugar do líder de plataforma está ameaçado por outras companhias dentro e fora do ecossistema, se deve tomar cuidado para que o líder da plataforma ou o software central para que não seja derrubado por um jogador que compõem o ecossistema. Auto- renovação é como a companhia traz inovações para seu produto trabalhando com parceiros para continuar atrativo enquanto o ambiente muda com o tempo, assim como as necessidades do consumidores assim como um nova tecnologia surgindo. Durante a auto-renovação, o ecossistema pode mudar completamente, a nível de software ou estratégias utilizadas por companhias para sobreviver.

O ecossistema de negócios possui jogadores que, atuando através dos estágios, eles são: líderes de plataforma, jogadores de nicho e jogadores de conexão, como explicado por [2]. Líderes de plataforma são as companhias que provém primeiramente o produto para viabilizar o ecossistema que nascerá e crescerá através do tempo. Jogadores de nicho são aqueles que utilizam os produtos feitos pelo líder de plataforma para atingir um nicho de mercado que inicialmente o produto não estava direcionado, assim como a produção de recursos para o produto que aumentam sua usabilidade, tornando-o mais atrativo durante o ciclo de vida do ecossistema. Jogadores de conexão são aqueles que fazem a conexão entre diferentes ecossistemas sem importar o número de ecossistemas envolvidos, sendo um jogador de conexão significa que recursos podem ser compartilhados através do ecossistemas, o exemplo dado por [9] é o software *middleware* que integra diferentes tipos de tecnologias, jogadores de conexão não é definido como um papel mas sim como uma propriedade.

Ecosistema de software

O conceito de Ecosistema de Software (ECOS ou SECO) vem da indústria de desenvolvimento de software como uma maneira de entender como um software interage com outros e como pessoas interessadas são afetadas por essas interações. Um exemplo de ECOS que é bem comum na literatura é ecossistema do iPhone. Esse ecossistema tem como líder de plataforma a Apple, que concebeu a Apple Store, a qual possui aplicações (apps) feitas pela Apple e outros desenvolvedores. Spotify é um exemplo de jogador de nicho, que representa um aplicativo para a plataforma iPhone provida pela Apple. A Apple projeta o iPhone mas não o produz o mesmo, assim como é vendido por empresas de telecomunicações e varejistas especializados. Em relação aos serviços de suporte, isso também é terceirizado.

Como descrito por [6], a indústria do software começou com apenas uma única empresa produzindo um produto e vendendo diretamente ao consumidor, sem qualquer interação com outras empresas, como parceiros de distribuição, pois eram vistos como competição por um quota do mercado. Agora, a indústria do software evoluiu para uma rede complexa de interações entre empresas que contribuem com visões diferentes e valores ao produto. A parceria que foi trazida pelo SECO desfaz o antigo padrão que uma companhia trabalha sozinha com um produto desde a sua construção até sua distribuição, pois agora a produção de software é fragmentada devido as organizações que oferecem serviços complementares e funcionalidades, visando o aumento de valor, do software produzido pelo líder de plataforma.

Ecosistemas de software é definido em [5] como *“um conjunto de atores funcionando como uma unidade e interagindo com um mercado compartilhado por softwares e serviços, juntos com relações entre eles. Essas relações são frequentemente apoiadas por uma plataforma tecnológica comum ou mercado e operam através da troca de informação, recursos e artefatos”*.

Em [11], um ecossistema de software é definido como *“consistindo de uma série de soluções de software que permitem, suportam e automatiza as atividades e transações pelos atores aos ecossistemas sociais ou de negócios e as organizações que provém essas soluções”*.

Como visto em [4], um ecossistema de software possui mais que a definição dada por [5] e [11], por possuir uma estrutura similar ao Ecosistema de negócios com quatro estágios: nascimento, expansão, liderança e auto renovação.

Até agora, a compreensão sobre um ECOS, dado por [4], [5] e [11], são as relações entre os produtores de software, provedores de serviço e empresas de tecnologia, que geram soluções para produto de software comum que atenda as necessidades e desejos dos consumidores, com a básica estrutura do ecossistema de negócio, baseada no nascimento do produto de software até sua fase de auto renovação (morte), compreendendo a expansão de mercado e a luta pelo seu contínuo lugar como líder do seu próprio ecossistema, tudo isso associado com a companhias que contribuem para a saúde e manutenção do ecossistema, operando com recursos e conhecimento.

Em relação aos recentes trabalhos relacionados a Ecossistemas de Software que compreendem toda a construção do ECOS são sobre o desenvolvimento de um ecossistema de software, estudado por [7], como sustentar e manter os parceiros que provém soluções e alcance de mercado explicado por [6], mantendo sua saúde a partir da identificação de como participantes, recursos e ordenação do ecossistema estão influenciando o ECOS como dito por [8] e as estratégias que servem de guia para as companhias dentro do ECOS como explicado por [4]. Considerando o desenvolvimento de um Ecossistema de Software, [7] diz que não um caminho exato a ser seguido para se criar um ecossistema de sucesso, mas há alguns passos, dentro de cada passo há uma série de atividades que devem ser cumpridas, esses passos incluem pré análise, design e avaliação. A pré análise consiste em pensar no escopo e fazer projeções em qual é o foco do ecossistema, a partir do entendimento onde o ecossistema irá agir para entregar o produto para o consumidor e como os participantes irão integrar o ECOS durante seu ciclo de vida, depois do estágio de pré análise vem o design do ecossistema que é como o ecossistema será constituído com os parceiros em três tipos de design: o *infraestruturado* que foca no produto que virá a ser a estrutura de tecnologia do ecossistema como por exemplo o ecossistema da IDE Eclipse, que todo ecossistema se baseia na construção do Eclipse; o *ator* que consiste em uma série de parceiros poderosos para formar a estrutura básica do ecossistema e finalmente o *negocial* que foca na oportunidade de mercado que a estrutura básica para esse ecossistema é possuir incentivo para que as empresas participem dele; avaliação é onde ocorre o monitoramento das reações do ecossistemas para entender como proceder com o ecossistema atuando como um passo iterativo, uma característica similar ao estágio de auto renovação do ECOS.

Faça uma comparação com o desenvolvimento de um ECOS, o DS-SECO proposto por [4] vem com um modelo para analisar a evolução do ecossistema através do ciclo de vida

e como tomar decisões estratégicas para continuar dentro do ecossistema, para os jogadores de nichos, ou adaptar o ecossistema ao mercado. A DS-SECO une o conceito de Strategy Dynamics, que fornece insights e compreende como o desempenho de um negócio evolui através do uso de uma estratégia para ganhar vantagem no mercado, com a estrutura ECOS, para permitir que as empresas se adaptem trajetória temporal utilizada, identificando recursos, analisando o fluxo de recursos e respondendo às questões dinâmicas: Por que o desempenho histórico do negócio seguiu o caminho temporal que ele possui? Onde o caminho do desempenho futuro nos levará no caso de continuarmos como estamos? Como podemos alterar esse futuro para melhor? Com aqueles que uma empresa dentro da economia deve determinar o comportamento a seguir para ter sucesso no ECOS.

A manutenção do ECOS deve levar em conta, já que não há propósito em criar um ecossistema insustentável, como [6] foco na gestão de parceiros dentro do ECOS, resultando em quatro níveis de gestão que incluem: seleção de parceiros que agreguem valor direto para o software ou atuar como membro para alcançar uma faixa mais alta do mercado; como lidar com cada parceiro individual para ver quão benéfico será o seu papel no ecossistema; fazer um programa de parceria para gerenciar a grande quantidade de parceiros e lidar com eles simultaneamente e, por último, produzir uma rede entre os parceiros para colaborar uns com os outros por meio de condições para melhorar o ecossistema. Quanto a [8] vê não apenas a relação de parceria, mas toda a arquitetura do ecossistema de software e propõe uma estrutura para medir a saúde dos componentes dentro do ECOS, uma vez que a definição de produto e atores dentro do ECOS são diferentes do ecossistema natural. No ECOS, há uma entidade que orquestra o ecossistema aplicando regras, normalmente essa entidade é a líder da plataforma, diferente do ecossistema natural que não possui uma figura central que regule o funcionamento do ecossistema.

Quanto a esses estudos recentes, podemos ver que há um foco nas estratégias de desenvolvimento e manutenção de um ecossistema, em [7] são apresentados passos para desenvolver um ECOS, fazendo uma pré análise, desenho, monitoramento e avaliação do mesmo, como esses dois últimos passos são iterativos e são aplicados após a implementação do ecossistema como uma maneira de analisá-lo. O ecossistema em curso feito será avaliado e poderá mostrar que procederá de uma maneira diferente a que foi desenhado, para isso [4] propõe um modelo que consiste na análise do momento atual do ecossistema e toma decisões estratégicas para lidar com as mudanças dentro e otimizar o ECOS baseando-se em

parâmetros de saúde, tempo e recursos, onde um dos recursos considerados são parceiros a serem mantidos, como [6] mostrar uma maneira de manter e gerenciar aqueles através de níveis de série que são discutidos no papel .

METODOLOGIA

Questão de pesquisa e modelo proposto

Como dito em [4], ecossistemas de software pode mudar através do tempo e serem adaptados, mas como observar a qualidade do ecossistema durante as mudanças a medida que o ciclo de vida ocorre? Como as ferramentas disponibilizadas pelos líderes de plataforma para os jogadores de nicho afetam a qualidade se tratando de rentabilidade e atratividade tanto para os jogadores de nicho quanto aos consumidores? Como abordado por [10] há trabalho a ser feito ainda sobre qualidade em ecossistemas de software, com seu foco em aspectos do desenvolvimento de software de uma maneira generalizada, este trabalho irá focar nas ferramentas de desenvolvimento, as IDEs, e qual seu impacto na qualidade do ecossistema de maneira que afetem tanto para os jogadores de nicho fazerem produtos melhores quanto para os consumidores desejarem tais produtos.

Os estudos mencionados anteriormente são relacionados aos recursos providos pelos líderes de plataforma e estratégias utilizados por eles, para os jogadores de nicho, [6] mostra para nós como criar e manter as relações entre eles e os líderes de plataforma, mas não mostra como a falta de jogadores de nicho pode fazer o ecossistema não viável ou o excesso dos mesmos pode saturá-lo.

Assim como o modelo DS-SECO nos explica o ecossistema do iPhone a partir de seu modelo através do tempo, enquanto [6] nos traz uma análise em 4 estágios em como manter as relações entres os parceiros e [10] foca na maturidade do ecossistema de desenvolvimento de software. A partir da combinação das ideias destes trabalhos propomos uma análise de como as ferramentas de desenvolvimento afetam tanto as relações com os jogadores de nicho quanto a rentabilidade e atratividade para os consumidores. A medida que o ciclo de vida se desenvolve novos recursos são adicionados às IDEs podendo atrair mais jogadores de nicho que assim podem trazer ou não novos consumidores para fazerem parte do ecossistema.

Execução da pesquisa

Este trabalho irá abordar o ecossistema do iOS e do Windows Phone, a partir da comparação desses ecossistemas nós iremos mostrar qual foi a influência que suas

ferramentas de desenvolvimento (IDE) e recursos proveram para o sucesso e declínio das plataformas além de mostrar como o relacionamento com os jogadores de nicho foi estabelecido durante o ciclo de vida dos ecossistemas. Baseado na linha temporal do próprio ecossistema e nas ferramentas de cada plataforma iremos estudar o XCode para o iOS e o Visual Studio para o Windows Phone.

Ecossistema do iPhone(iOS)

Como estudado por [4], o ecossistema do iOS surgiu em 2007, em Agosto do mesmo ano a Apple apresenta o XCode 3 que inclui o SDK para o iOS 2.0, trazendo um recurso que constrói Apps para os usuários do iPhone. Infelizmente devido a falta de informações oficiais sobre o XCode 3, utilizaremos o Wikipédia apenas como referência se tratando da linha do tempo da IDE, para fins de estudo iremos iniciar do XCode 4 e seus recursos descritos em [11], [4] afirma que o estágio de auto-renovação do iPhone foi concebido em 2010 quando surgiu o iOS 4.0.

O XCode 4 surgiu em Março de 2011 contendo o SDK para o iOS 4..3 presente até então nos iPhones 3GS e 4, iPad 1 e 2 além do iPod Touch, na terceira e quarta geração segundo [13], o XCode 4 apresentou recursos como o Gerenciamento de Repositório GIT dando aos desenvolvedores ferramentas para trabalhar com repositórios remotos e em equipe, trazendo mais flexibilidade ao trabalho, recursos como o *storyboards* também foram integrados à IDE, esse recurso permite que o desenvolvedor construa a interface do usuário e consiga definir a transição delas a partir dos comandos e gatilhos de transição, o simulador do iPhone que contido dentro do XCode agora possui um simulador de localização para que se possa criar apps com melhores sistemas de geolocalização. Outra fase de auto-renovação do ecossistema do iPhone foi a concepção do iOS 5 que continha uma inteligência artificial como assistente virtual, a Siri, além da completa integração com a nuvem em um novo sistema chamado *iCloud*. O XCode 4 abrangeu do iOS 4.3 ao 6.1.

O XCode 5 trouxe consigo o SDK do iOS 7, além das funcionalidades de seus antecessores a nova IDE possui um gerenciador de código-fonte e editor de versão próprio além do organizador de repositórios já constituinte, integração contínua foi outro recurso a ser implementado na IDE, provendo assim uma maneira mais organizada de liberar

atualizações do app em construção. O iOS 7 marcou também uma fase de auto-renovação do iPhone devido ao lançamento do novo sistema de desbloqueio do smartphone com o Touch ID, inserindo hardware e software de leitura de impressão digital, o ciclo de vida do XCode 5 foi curto, foi utilizado apenas para o desenvolvimento de aplicações do iOS 7 e seus anteriores, a partir do iOS 8 foi utilizado o XCode 6.

Para o XCode 6 as diferenças apresentadas foram outras, uma nova linguagem foi apresentada aos desenvolvedores, o Swift, com todas as características de seus anteriores mas agora com o suporte para a nova linguagem, um novo recurso integrado ao Swift foi o chamado *Playground* em que o resultado de uma linha de código escrita é exibido imediatamente segundo [15], assim como seu antecessor seu ciclo de vida foi curto tendo como tempo de vida 9 meses, entre setembro de 2014 até junho de 2015 e só suportava até as versões mais novas do iOS 8.

Em setembro de 2015 a Apple lança o XCode 7 agora com o SDK do iOS 9 além dos kits de desenvolvimento de seus novos produtos, o Apple Watch e Apple TV, contendo assim os SDK do watchOS e tvOS, provendo assim uma maneira de construir aplicações tanto universais quanto que conversem entre si, além desses recursos, o *Crash Log* também foi inserido na nova versão do XCode, que provém uma maneira para analisar erros e interrupções do funcionamento do app que foram observados durante o uso do mesmo pelo usuário.

O iOS 10 foi lançado em setembro de 2016, anunciado pela Apple como “o maior lançamento do iOS de todos os tempos” segundo [16], trouxe para o desenvolvedor o SiriKit, um kit de desenvolvimento para a Siri, fazendo com que os desenvolvedores pudessem utilizar a assistente pessoal diretamente em seus apps e não apenas como o inicializador dos mesmos quando o usuário o pedisse. O SDK do iOS 10 veio embutido no XCode 8 lançado no mesmo mês que o sistema operacional, seu ciclo de vida foi de um ano com o lançamento do iOS 11 e o XCode 9.

Para fins de estudo a linha temporal do ecossistema do iPhone junto com o XCode se encerrará no iOS 10 e XCode 8, isso será explicado devidamente na Ação de pesquisa.

Ecossistema do Windows Phone

O Ecossistema do Windows Phone se inicia com o Windows Phone 7 lançado em 2010 mas a história da Microsoft com smartphones data de 1996, segundo [17], com o lançamento do Windows CE que foi lançado para os chamados *Handheld PCs*(Computadores de mão) após isso houve uma lista de SOs produzidos para dispositivos portáteis, mas para fins de pesquisa iremos começar com o Windows Phone 7, seu SDK veio como uma extensão do Visual Studio 2010, que é a IDE proprietária da Microsoft utilizada para criar aplicações com as tecnologias providas por ela, o Windows Phone 7 trouxe um novo layout de smartphone para concorrer com seus grandes adversários na época, o Android e o iPhone, utilizava uma linguagem de design da própria Microsoft chama Metro UI, que consistia em tipografia e ícones simplificados conhecidos como *Live Tiles*. O Visual Studio 2010 trouxe consigo os recursos de desenvolvimento na nuvem utilizando o *Windows Azure Tools*, que cria integração entre a aplicação que está sendo desenvolvida e o sistema de nuvem da Microsoft, podendo testar a aplicação e disponibilizar o serviço diretamente na web. Um sistema de programação paralela foi incorporado também para entender como uma aplicação se comporta quando há um processador com múltiplos núcleos.

O Visual Studio 2012 chega trazendo o SDK do Windows Phone 8 e Windows 8, com a nova proposta da Microsoft de unificar o desenvolvimento de aplicações, ou seja, aplicativos feitos para o Windows Phone 8 poderiam ser executados no Windows 8, tendo a unificação dos layouts, a utilização dos *Live Tiles*. O novo SDK trouxe também novos emuladores de Windows Phone, contando agora com 3 tamanhos de tela WVGA, WXGA e 720p, para que o desenvolvedor possa ver o design do seu *Tile* em diferentes tamanhos.

O Windows Phone 8.1 já foi uma nova atualização do sistema operacional lançada em 2014, sendo considerada uma fase de auto-renovação, com diferenças visíveis para os usuários do sistema operacional, seguindo a ideia de unificação das plataformas, a assistente digital ativada por voz da Microsoft foi incluída na versão 8.1 do SO além da adição de um painel de notificação chamado *Action Center* na parte superior da tela. Para os desenvolvedores as mudanças não foram tão significativas, foram adicionados novos recursos de tarefas em segundo plano como: verificar a intensidade do sinal bluetooth, mudança da conexão do celular, Geolocalização e *Push Notification*. O novo SDK foi lançado junto a nova versão do SO, sua IDE alvo foi o Visual Studio 2013.

Em 2015 a Microsoft lança o Windows Phone 10 e o Windows 10, a última versão do SO mobile veio com a nova forma de união entre os dispositivos, o UWP (Universal Windows

Platform). O SDK do Windows Phone 10 teve como alvo o Visual Studio 2015, sua maior funcionalidade agregada foi o UWP, utilizando agora o XAML como linguagem para construir o design para todos os dispositivos e o C# como linguagem de controle de funcionalidades da aplicação. O UWP permitiu a criação de uma única aplicação para o Windows 10, Windows Phone 10 e Xbox One.

Ação de pesquisa

Iremos comparar a linha do tempo das IDEs junto com as de seus sistemas operacionais alvo, para entender como os seus recursos causaram impacto nos ciclos de vidas de cada um. A Microsoft encerrou oficialmente o lançamento de novos recursos para o Windows Phone em outubro de 2017 segundo [22], encerrando assim o ciclo de vida do mesmo, porém sua IDE, o Visual Studio continua em funcionamento oferecendo os recursos do UWP para os desenvolvedores. A Apple continua a desenvolver o iOS quase que anualmente como vimos na linha do tempo que foi explicada anteriormente, para continuação da pesquisa encerraremos com o XCode 8 devido ao mesmo ter o ciclo de vida renovado em meados de 2017.

Em 2010 antes do lançamento do XCode 4, a Apple possuía 43.185 desenvolvedores exclusivos de aplicativos para o iOS, tendo mais 1.412 desenvolvedores construindo aplicações para as duas plataformas *mobile* da época, o iPhone e o Android, segundo [21], no mesmo ano a Apple vendeu em torno de 40 milhões de unidades do iPhone, no ano seguinte após o lançamento do XCode 4 e do iOS 4.3 viu esse número aumentar para 72,29 milhões de unidades vendidas no mundo segundo, segundo [23], ao final de 2011 a Apple afirma que possuía 425.000 aplicativos em sua loja, a *App Store*, é o que afirma [24]. Para o Windows Phone, em 2010 não foram encontrados números exatos de seus desenvolvedores, para o ano de 2011 a própria Microsoft disponibilizou um infográfico contendo informações, havia 50.000 aplicativos disponíveis para download na sua loja de aplicativos a *MarketPlace* e 80.000 desenvolvedores registrados, segundo [25].

Há no primeiro ano uma diferença entre os Ecossistemas, há quase o dobro de desenvolvedores registrados na Microsoft que na Apple, isso se deve ao fato de que a Microsoft trouxe os desenvolvedores de aplicação do sistema operacional Symbian para o

Windows Phone com a construção de um *toolkit*, para que se pudesse realizar a portabilidade dos aplicativos do Symbian para o Windows Phone 7. O Visual Studio 2010 trouxe consigo as ferramentas de debug e escalabilidade em nuvem do Azure enquanto o XCode 4 trouxe o GIT como nova ferramenta de versionamento, tornando mais rápido o compartilhamento de código e integração entre uma equipe de desenvolvimento além de um mapa do comportamento do app, para entender como o usuário se comporta ao utilizar a aplicação, o *Storyboard*. Ao final de 2011 o iOS estava em 22,29% dos smartphones presentes no mundo enquanto o Windows Phone estava em apenas 0,54% assim como é mostrado na Figura 1.

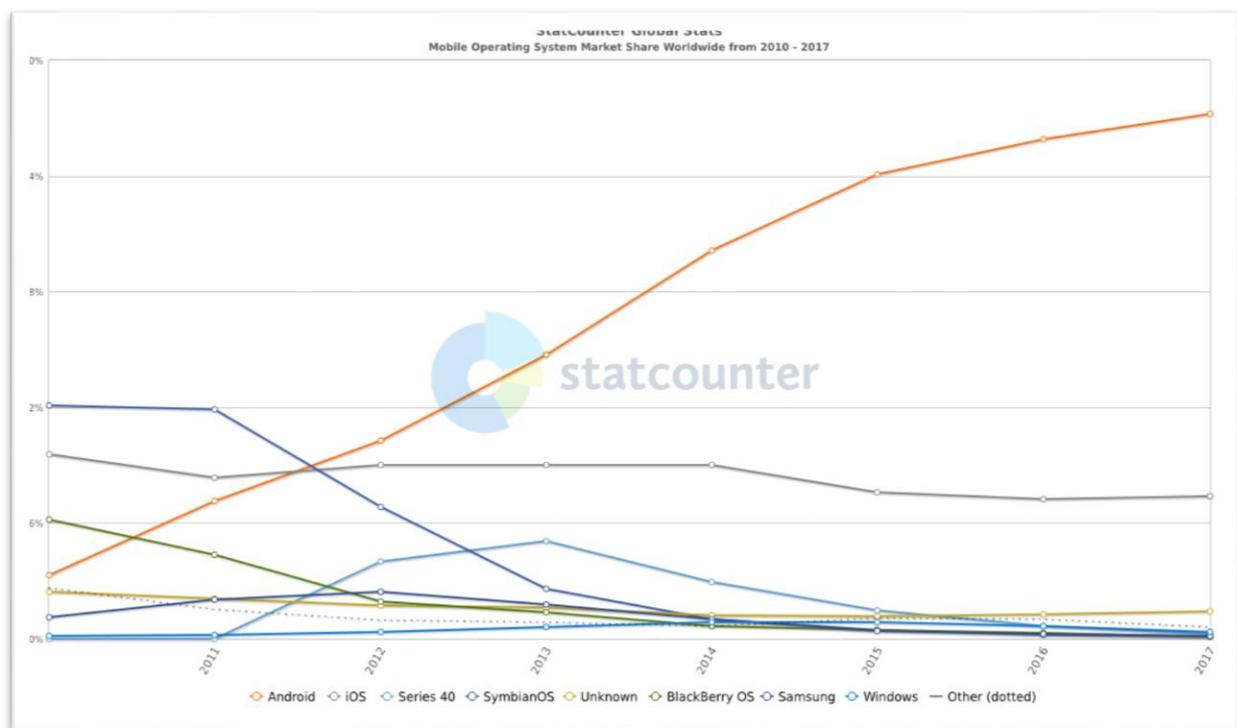


Figura 1 - Quota de mercado dos sistemas operacionais móveis de 2010 - 2017

Em 2012 o XCode 4, lançado em 2011, continua sendo a IDE utilizada pelos desenvolvedores, a Apple o manteve atualizando recursos e inserindo pequenas modificações como suporte a gestos da tela, por mais que houvesse mudança de hardware e novas versões do iOS, apenas os SDK foram atualizados para o XCode até o lançamento de sua versão 5, o iOS agora possui em torno de 24% do mercado de smartphones do mundo. Para a Microsoft sua mudança foi a estratégia de unificar suas plataformas o Visual Studio 2012 possuía o SDK do Windows Phone 8 e do Windows 8, porém seus projetos ainda eram criados de maneira

separadas, tendo assim que criar dois projetos para as plataformas, a Microsoft, em 2012, atinge 0,96% do mercado de sistemas operacionais móveis ativos.

O novo iOS 7 é lançado em 2013 junto a um novo iPhone que traz consigo o TouchID, a tecnologia de desbloqueio de tela e aplicativos a partir da impressão digital, o XCode 5 que incorpora o SDK do novo iOS já traz o recurso para que os desenvolvedores possam utilizar o TouchID mas termina o ano com a mesma quota do mercado que o ano anterior, já o Windows Phone vai para 1,59% mantendo a versão 8 do seu sistema operacional móvel, mas agora com o Visual Studio 2013 como IDE de desenvolvimento, não houve avanços significativos do Visual Studio 2012 para o Visual Studio 2013 se tratando de desenvolvimento mobile, suas maiores diferenças foram em relação ao desenvolvimento de sistemas para internet.

No ano de 2014 a Apple traz o XCode 6 com o SDK do iOS 8 com o que viria ser uma forma da Apple para simplificar o desenvolvimento para seus dispositivos, a linguagem *Swift*, a Apple continua com o *Objective-C*, mas trouxe todos os recursos do XCode agora para a nova linguagem,[27] lista uma série de razões pelas quais a nova linguagem é considerada ótima para o desenvolvimento de aplicativos para iOS mas que não irá substituir a já utilizada até o momento, porém não aumentou sua quota de mercado mas no ano, segundo [28], foram estimados 263.000 desenvolvedores para o iOS. Para a Microsoft, 2014 marcou o lançamento do Windows 8.1 e Windows Phone 8.1 que mantinham a ideia de um app para todas as plataformas Microsoft, porém ainda não havia estabilidade de suas aplicações entre as duas plataformas, porém houve aumento da quota do mercado em que se encontrava a 2,35%.

Em 2015 O XCode 7 é lançado com o SDK do iOS 9 trazendo recursos para testes e *crash* de aplicações, além da atualização da linguagem *Swift* para o 2.0 e recursos para o *Objective-C*, a App Store atinge 1,4 milhões de aplicativos em sua loja porém sua participação no mercado cai para 20,2%. A Microsoft lança o Windows 10 e Windows Phone 10, juntamente com a nova IDE Visual Studio 2015, agora com uma plataforma estável para a unificação dos aplicativos entre suas plataformas o UWP, tornando apps universais para todos os dispositivos que estiverem com alguma versão do Windows 10 instalada porém mesmo com esse avanço decresce para 2,29%.

No ano de 2016 a Apple lança o iOS 10, com ele o XCode 8 e o SDK do novo SO móvel, segundo [29] a App Store possuía 2 milhões de aplicativos registrados, ainda foi disponibilizado o *Swift Playgrounds*, um aplicativo para incentivar o desenvolvimento de aplicações para iOS, mesmo com tais recursos de sua IDE e incentivos se tratando de

aprendizagem sua participação no mercado continuou caindo, agora para 19,29%. Para a Microsoft não houve tantos avanços em relação ao Visual Studio ou Windows Phone mas sua fatia do mercado caiu para 1,75%.

Em 2017 após o anúncio de encerramento de novos recursos para o Windows Phone a quota de mercado do mesmo cai para 0,89%, enquanto a quota do iOS cresce timidamente para 19,65%

RESULTADOS E DISCUSSÃO

Durante as comparações dos ciclos de vida do ecossistemas móveis não foram encontrados indícios de que os recursos das IDE causaram impactos diretos aos ecossistemas, houve aumento na quantidade de apps em cada loja pois era o esperado como fase de expansão do ciclo de vida do ecossistema porém suas quotas no mercado não se comportaram de maneira similar, o Windows Phone cresceu de maneira progressiva porém em sua última auto renovação seus números diminuíram tanto no ano da renovação do ecossistema assim como no ano posterior até que fosse totalmente encerrada sua produção de recursos. Para o iOS aconteceu algo similar, após o lançamento da linguagem *Swift* em 2014 sua quota de mercado caiu após permanecer estável no do lançamento da linguagem.

Durante a pesquisa foram encontrados indícios que apontam a relação entre as empresas e os desenvolvedores externos, os jogadores de nicho, como um dos fatores para os acontecimentos de flutuação nas quotas de mercado, [31] exhibe alguns problemas se tratando do Windows Phone, como em alguns países operadoras de telefones simplesmente não anunciavam a venda de smartphones contendo o sistema operacional da Microsoft, a baixa velocidade com que a Microsoft introduz recursos para o Windows Phone, recursos os quais estão já disponíveis em outras plataformas, [32] realça o fato de a Google, com um vasto portfólio de aplicações não estar enviando seus aplicativos para os usuários do Windows Phone, uma vez que esses aplicativos se fazem presentes na App Store.

CONSIDERAÇÕES FINAIS

Entendendo a análise realizada, os resultados encontrados não foram de acordo com o que a pesquisa propunha inicialmente pois foi retomado o fato das estratégias impactarem de maneira visível na qualidade e sustentabilidade do ciclo de vida do ecossistema. Este estudo proveu uma análise histórica de 2010 até 2017 quando desde o nascimento do Windows Phone e a primeira renovação do iPhone até 2017 quando foi anunciado o fim do Windows Phone, apresentando várias dos recursos que suas IDEs obtiveram através dos anos para que houvesse facilidade na construção do código e atratividade para continuar sendo um desenvolvedor para a plataforma, uma vez que o impacto dessas IDEs não pode ser confirmado nos ciclos de vida de cada ecossistema, o estudo é útil como forma de afirmação que as estratégias escolhidas pelos líderes de plataforma causam impacto direto no ciclo de vida de maneira a fazê-lo continuar sendo um ecossistema atuante ou acabar com o mesmo. Futuros trabalhos podem entender se o CMMI associado as estratégias de software podem ser aplicados aos ecossistemas de software, uma vez que o CMMI lida com qualidade do processo de software.

REFERÊNCIAS

- [1] D. Dhungana, I. Groher, E. Schludermann, and S. Biffi. (2010). “Software ecosystems vs. natural ecosystems: learning from the ingenious mind of nature”. In Proceedings of the Fourth European Conference on Software Architecture: Companion Volume, pages 96–102.
- [2] S. Jansen and M. Cusumano. (2012) “Defining Software Ecosystems: A Survey of Software Platforms and Business Network Governance”. In Proc. of the 4th Int’l Workshop on Software Ecosystems (IWSECO ’12), pages 41–58.
- [3] Moore, James. (1999). “Predators and Prey: A New Ecology of Competition”. Harvard business review. 1.75-86.
- [4] R. Andrade, C. Alves, G. Valença. (2015) “An analysis of dynamic strategies during the lifecycle of software ecosystems: The DS-SECO Model”. Brazilian Congress on Software: Theory and Practice/WDES, Belo Horizonte, Brazil, pp. 57–64.
- [5] Jansen, S., Finkelstein, A. and Brinkkemper, S. (2009) “A sense of community: a research agenda for Software Ecosystems”. In: Proc 31st International Conference on Software Engineering, pp 187-190.
- [6] A, Avila. O, Terzidis. (2016) “Management of Partner Ecosystems in the Enterprise Software Industry”. In 8th International Workshop on Software Ecosystems (IWSECO), At Dublin, Ireland.
- [7] P, Tyrväinen. M, Hämäläinen. K, Manikas. (2016) “Designing, Developing, and Implementing Software Ecosystems: Towards a Step-wise Guide”. In 8th International Workshop on Software Ecosystems (IWSECO), At Dublin, Ireland.
- [8] K, Marius Hansen. K, Manikas.(2013) “Reviewing the Health of Software Ecosystems – A Conceptual Framework Proposal”. In 5th International Workshop of Software Ecosystems (IWSECO), At Potsdam, Germany.
- [9] S.Jansen, M. A. Cusumano, S. Brinkkemper.(2013) “Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry”. Edward Elgar Publishing
- [10] K, van Baarsen. S, Jansen. S, España. (2017) “Measuring Tool and Resource Maturity in Developer Ecosystems”. In 9th International Workshop on Software Ecosystems (IWSECO), At Helsinki, Finland.
- [11] Developer Apple (2016), https://developer.apple.com/library/archive/documentation/Xcode/Conceptual/WhatsNewXcode-Archive/Articles/Introduction.html#//apple_ref/doc/uid/TP40004635-DontLinkElementID_1. Acessado em 10/07/2018
- [12] Wikipedia (2018), <https://en.wikipedia.org/wiki/Xcode>. Acessado em 10/07/2018

[13] Apple (2011), <https://www.apple.com/newsroom/2011/03/02Apple-Introduces-iOS-4-3/>. Acessado em 10/07/2018

[14] The Verge (2011), <https://www.theverge.com/2011/12/13/2612736/ios-history-iphone-ipad>. Acessado em 10/07/2018

[15] Apple developer (2016), https://developer.apple.com/library/archive/documentation/Xcode/Conceptual/WhatsNewXcode-Archive/Articles/xcode_6_0.html#/apple_ref/doc/uid/TP40014509-SW1. Acessado em 10/07/2018

[16] Apple (2016), <https://www.apple.com/newsroom/2016/06/apple-previews-ios-10-biggest-ios-release-ever/>. Acessado em 10/07/2018

[17] Mobiforge, <https://mobiforge.com/timeline/windows-phone-history>. Acessado em 12/07/2018

[18] Microsoft MSDN blogs (2009), <https://blogs.msdn.microsoft.com/joy/2009/10/11/visual-studio-2010-features-and-advantages/>. Acessado em 12/07/2018

[19] Microsoft MSDN blogs (2012), <https://blogs.msdn.microsoft.com/visualstudio/2012/10/30/introducing-windows-phone-sdk-8-0/>. Acessado em 12/07/2018

[20] Windows Central (2014), <https://www.windowscentral.com/developers-leak-new-features-windows-phone-81-sdk>. Acessado em 12/07/2018

[21] Mashable (2010), <https://mashable.com/2010/07/02/ios-android-developer-stats/#dT3dgpPrS5qB>. Acessado em 15/07/2018

[22] The Verge (2017), <https://www.theverge.com/2017/10/9/16446280/microsoft-finally-admits-windows-phone-is-dead>. Acessado em 15/07/2018

[23] Statista (2018), <https://www.statista.com/statistics/276306/global-apple-iphone-sales-since-fiscal-year-2007/>. Acessado em 15/07/2018

[24] Apple (2011), <https://www.apple.com/newsroom/2011/07/07Apples-App-Store-Downloads-Top-15-Billion/>. Acessado em 17/07/2018

[25] The Windows Club (2012), <https://www.thewindowsclub.com/infographic-windows-phone-2011-retrospective>. Acessado em 17/07/2018

[26] The Windows Club (2012), <https://www.thewindowsclub.com/download-windows-phone-toolkit-symbian-app-developers>. Acessado em 17/07/2018

- [27] Infoworld (2015), <https://www.infoworld.com/article/2920333/mobile-development/swift-vs-objective-c-10-reasons-the-future-favors-swift.html>. Acessado em 17/07/2018
- [28] Quora (2018), <https://www.quora.com/How-many-iOS-developers-exist-worldwide>. Acessado em 30/07/2018
- [29] Apple (2016), <https://www.apple.com/newsroom/2016/10/apple-opens-first-ios-developer-academy-in-naples/>. Acessado em 30/07/2018
- [30] Apple (2016), <https://www.apple.com/newsroom/2016/09/swift-playgrounds-now-available-on-the-app-store/>. Acessado em 30/07/2018
- [31] Forarena (2016), <https://www.fonearena.com/blog/77660/there-are-six-major-problems-with-windows-phone-and-apps-isnt-one-of-them.html>. Acessado em 30/07/2018
- [32] The Verge (2017), <https://www.theverge.com/2017/10/10/16452162/windows-phone-history-glorious-failure>. Acessado em 30/07/2018