COMPARATIVO ENTRE SISTEMAS OPERACIONAIS MÓVEIS – ANDROID X IOS

COMPARATISON BETWEEN MOBILE OPERATING SYSTEMS – ANDROID X IOS

Alexandre Canosa Leite – alecanosa.leite@gmail.com
Faculdade de Tecnologia (FATEC) – Taquaritinga – São Paulo - Brasil
Helena Macedo dos Reis – helena.reis@fatec.sp.gov.br
Faculdade de Tecnologia (FATEC) – Araraquara – São Paulo - Brasil

RESUMO:

Hoje em dia os smartphones estão em alta e entre as principais diferenças, o sistema operacional é uma delas. Este artigo tem como objetivo apresentar uma análise comparativa entre dois sistemas operacionais para dispositivos móveis, Google Android e Apple iOS. O documento vai abordar as principais diferenças entre os sistemas operacionais, considerando as características que cada sistema operacional possui. Este artigo foi estruturado tendo como referência pesquisas bibliográficas e consultas de informações em *sites* oficiais de cada plataforma. Os resultados obtidos foram que cada sistema operacional tem suas características de modo a beneficiar qualquer tipo de usuário, tanto para desenvolvimento quanto para utilização, e também servir de guia para estes desenvolvedores e usuários que estão interessados em possuir smartphones com esses sistemas operacionais. Com base nos resultados obtidos podemos concluir que este artigo poderá ajudar o usuário e o desenvolvedor a obter informações referentes a estes sistemas operacionais possibilitando um bom embasamento sobre os mesmos.

Palavras-chave: Apple iOS. Arquitetura. Google Android.

ABSTRACT:

Nowadays the smartphones are on the rise and the difference between them, it's the operational system, the objective of this article is to show a comparative analysis between two smartphones operational systems, Google Android e Apple iOS. The document will talk about the main differences between two operating systems, considering their own characteristics, this article was structured by a study of case, referencing bibliographic searches and collecting informations on official webs of thosen systems. The results are that each system has its own characteristics that can benefit any type of user, for both development and use, it can be an guide to developers and users that are interested in buy an smartphone with this operational systems. Based on results, we can conclude that this article will help the user and the developer to obtain informations about this operational systems, making possible one good rudiment about them.

Keyword: Apple iOS. Architecture. Google Android.

1 INTRODUÇÃO

A presença dos dispositivos móveis é imprescindível no cotidiano das pessoas, comandando uma ampla área no mercado profissional estes dispositivos podem ser utilizados no meio corporativo para solucionar diversos problemas empresariais (SETTING CONSULTORIA, 2017).

Com essas informações percebe-se que as vendas dos dispositivos móveis não param de crescer. Muitas empresas de grande porte, referente à área de tecnologia e telecomunicações vêm sendo atraídas pelas oportunidades de negócio que são geradas pelo alto uso desses aparelhos. Segundo o Instituto Brasileiro de Geografia e Estatística (IBGE), no ano 2012, 83,3% das empresas utilizavam o celular corporativo ou pessoal para fins de trabalho (EXTRA, 2012), como, por exemplo, para anotar pedidos, acessar e-mails corporativos ou utilizar aplicativos que geravam suporte para uma determinada função de um funcionário.

Um exemplo, seria uma empresa provedora de internet, pois ela forneceria um smartphone para o técnico de campo, em que por meio de um determinado aplicativo poderiam ser enviadas ordens de serviços para o técnico de campo e em tempo real o gestor responsável pudesse verificar o desempenho de seus funcionários possibilitando também a criação de rotas alternativas para que os técnicos cheguem com mais eficiência na casa do cliente para prestar o suporte necessário.

Portanto é evidente que a presença do smartphone no cotidiano das corporações é um tanto essencial e, atualmente, duas empresas destacam-se pelo fornecimento desses dispositivos com seus respectivos sistemas operacionais, Google Android e Apple iOS (EL PAÍS, 2017).

O Google Android, de acordo com o instituto de pesquisa Gartner, lidera em número de vendas com 296,9 milhões de smartphones ativados, totalizando 86,2% de mercado. Em segundo, a Gartner indicou o iOS que é disponibilizado no iPhone com 44 milhões de unidades e 12,9% do mercado (OLHAR DIGITAL, 2016).

Com as informações citadas nos parágrafos anteriores pode-se demonstrar a importância destas duas plataformas no meio empresarial realizando uma análise comparativa entre essas duas principais plataformas de desenvolvimento para dispositivos móveis Google

Android e Apple iOS. Essa escolha foi feita com base no destaque atual dessas empresas e o alto índice de desenvolvedores que são interessados em aprender sobre essas plataformas.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Visão geral da plataforma android

O Android é uma plataforma desenvolvida pela Google voltada para dispositivos móveis, totalmente aberta e livre (Open Source), que foi divulgada em 5 de novembro de 2007. Inicialmente, o sistema Android foi desenvolvido pela Google e atualmente essa plataforma é mantida pela Open Handset Alliance (OHA) (SILVA, 2015).

A OHA é uma comunidade constituída por empresas de grande porte do mercado de telefonia de celulares liderados pela Google com a finalidade de criar padrões abertos para a telefonia móvel (LECHETA, 2016).

Essa comunidade também pode ser considerada produtiva pois proporciona a intenção de padronizar uma plataforma de código aberto e livre para celulares, justamente para atender a todas as expectativas e tendências do mercado atual. As empresas dessa aliança estão trabalhando juntas para oferecer uma plataforma de desenvolvimento que permite aos desenvolvedores implementarem e estenderem as aplicações dos seus dispositivos móveis. Essa aliança tem como objetivo também lançar *handsets* e serviços usando a plataforma Android (GIROLLETE, 2012). Portanto, percebe-se que o sucesso do Android se baseia no investimento da Google, pois ela é uma empresa que segue revolucionando a internet e o grupo chamado OHA, cujo os integrantes são empresas de telefonia como Motorola, LG, Samsung, Sony Ericsson e muitas outras.

O que atraiu o interesse da Google foi o propósito de um projeto que possui uma plataforma aberta aos fabricantes, deixando-o flexível e atualizável, com o intuito de permitir flexibilidade ao usuário (SIMÕES E PEREIRA, 2014).

Ainda sobre Google, e o seu interesse em uma plataforma flexível pode-se constatar que a versatilidade do Android é verídica, pois a praticidade do mesmo permite que o usuário customize o dispositivo ao seu modo, possibilitando realizar personalizações básicas, intermediárias e avançadas. Com isso pode-se perceber que essa parceria, além de render todos esses benefícios citados, diminui o trabalho das empresas que criam dispositivos móveis tendo apenas o poder de foco na criação dos *hardwares* já que o sistema operacional é livre.

2.2 Arquitetura Android

A arquitetura do sistema operacional[1] Android é distribuída em camadas, onde cada uma delas é responsável por gerenciar seus respectivos processos (LECHETA, 2016).

2.1.1 Camada de aplicações do sistema

A camada de aplicações do sistema caracteriza-se por executar aplicações sobre a plataforma, sendo elas nativas do sistema, como gerenciador de contatos, navegador, calendário, etc (Android Developers, 2017). Portanto, essa camada pode ser considerada um suporte padrão onde o usuário pode utilizar os serviços do Android por meio de aplicações nativas. Contudo, a camada de aplicações do Android também se caracteriza por executar aplicações desenvolvidas por terceiros, mostrando o quão flexível o Android pode ser (Android Developers, 2017).

2.1.2 Camada Java API

Esta camada é um conjunto de ferramentas que fornecem vários serviços para os aplicativos, visando reuso de componentes (ANDROID DEVELOPERS, 2017). Sendo assim, ela é responsável por proporcionar facilidade para o desenvolvedor realizar a escrita do código, pois ela traz uma série de códigos já prontos com funções pré-definidas para o desenvolvimento de *sites*, *softwares*, aplicativos, etc.

Referente a essa camada, uma classe que é pré-definida, e muito utilizada é o Activity Manager. Ela gerencia o ciclo de vida de todas as activities, quando iniciar e quando terminá-las, possibilitando o deslocamento de uma activity para outra e assim por diante (PEREIRA E SILVA, 2009). Activity fornece uma tela com a qual os usuários podem interagir para fazer algo (ANDROID DEVELOPERS, 2017).

Contudo, pode-se perceber que uma activity pode ser considerada uma classe que faz a interação do usuário com a interface.

[1] https://developer.android.com/guide/platform/index.html

2.1.3 Camada de bibliotecas

Essa camada presente no Android possui bibliotecas em c/c++, as quais são utilizadas por vários recursos do sistema. Estas bibliotecas podem ser acessadas por frameworks disponibilizados para os desenvolvedores que também podem ser chamadas de Framework APIS. São elas, freetype, system c library, webkit, sqlite, sgl, surface manager, media libraries, libwebcore e 3d libraries (PEREIRA E SILVA, 2009).

Ela fornece as Framework APIS para evidenciar a funcionalidade de algumas dessas bibliotecas nativas aos aplicativos. Por exemplo, é possível acessar OpenGL ES pela OpenGL API da estrutura do Android para complementar a capacidade de desenhar e manipular gráficos 2D e 3D no seu aplicativo (ANDROID DEVELOPERS, 2017).

Portanto, percebe-se que os frameworks nativos do Android têm uma boa aplicabilidade do lado do usuário, fornecendo a ele recursos tanto para desenvolvimento quanto para o usuário que só utiliza o determinado dispositivo com o sistema operacional Android.

2.1.4 Camada de Runtime

A pequena camada do ambiente de execução (Android Runtime) é uma instância da máquina virtual Dalvik, criada para cada aplicação executada no Android (PEREIRA E SILVA, 2009).

O Dalvik funciona da seguinte maneira, ao realizar o desenvolvimento de aplicações para o Android, o desenvolvedor vai utilizar a linguagem e todos os seus recursos normalmente, mas depois que o *bytecode* é compilado, ele é convertido para o formato .dex (Dalvik Executable), que representa a aplicação do Android compilada, então os arquivos .dex e outros recursos como imagens são compactados em um único arquivo com extensão .apk (Android Package File), que representa a aplicação final, pronta para ser distribuída e instalada (LECHETA, 2016).

Com relação ao parágrafo anterior pode-se constatar que o Dalvik traduz o *bytecode* em um código legível pela máquina em questão, código que é de fato executado pelo processador principal. Essa tradução (compilação) ocorre durante a execução do aplicativo, cujo o nome é JIT (*Just In Time*) (LEIVA, 2016).

Com a criação do Android 4.4 (Kitkat) foi desenvolvida a máquina virtual ART (Android Runtime) com o objetivo de substituir a Dalvik que podia ser ativado opcionalmente nas configurações presentes no dispositivo.

A partir do Android 5.0(Lollipop), o ART tornou-se a máquina virtual padrão presente nos dispositivos, substituindo a Dalvik. Um dos benefícios presentes no ART é a compilação *Ahead-of-time* (AOT), o que significa que o código é compilado quando a aplicação é instalada (LEIVA, 2016).

O ART também, além de ter o objetivo de otimizar o código ao máximo com o seu funcionamento para melhorar o desempenho do aplicativo, possui um melhor funcionamento do Garbage Collector (GC) e apresenta melhorias no suporte ao debug de aplicativos (LECHETA, 2016).

Na execução dessas máquinas virtuais, os desenvolvedores ou usuários não são impactados pelo funcionamento dos mesmos, porém a Google afirma que o ART apresenta um desempenho muito melhor (LECHETA, 2016).

2.1.5 Camada de abstração de *hardware* (HAL)

A camada de abstração de *hardware* (HAL) fornece interfaces que mostram as capacidades de *hardware* do dispositivo para a estrutura da API de maior nível. Ela consiste em módulos de biblioteca, que implementam uma interface para um tipo específico de componente de *hardware*, como o módulo de câmera ou bluetooth. Quando um Framework API faz uma chamada para acessar o *hardware* do dispositivo, o sistema Android carrega o módulo da biblioteca para este componente de *hardware* (ANDROID DEVELOPERS, 2017).

Com as informações descritas no parágrafo anterior pode-se constatar que a camada de abstração de *hardware* é responsável pela comunicação entre o *hardware* do dispositivo e o *software*, a partir de uma ação realizada pelo usuário o *software* em questão aciona o *hardware*. Como exemplo pode-se citar, o usuário tirando uma foto ou enviando um SMS.

2.1.6 Camada de kernel linux

O Kernel é responsável por fazer a união entre a parte de *hardware* e *software*. O seu principal objetivo é gerenciar o computador e permitir que os aplicativos sejam executados e façam uso dos recursos que existem. O núcleo também tem que garantir que a memória RAM

seja usada em seu potencial sem riscos (SIMÕES E PEREIRA, 2014), ou seja, tem o intuito de estabelecer uma comunicação um tanto eficaz nos recursos do sistema.

No caso do Android, o kernel foi projetado com base na versão 2.6 do kernel do Linux tendo semelhança em suas funcionalidades, tais como segurança, gestão de memória, gestão de processos, etc (SILVA, 2015).

2.3 Visão geral da plataforma iOS

iOS é um sistema operacional da Apple lançado em 09 de janeiro de 2007 na Macworld Conference & Expo sendo descendente do iOS X, desenvolvido especialmente para iPhone e também utilizado em iPod touch, iPad e Apple Tv.

Ele foi desenvolvido pela Apple e sua execução é restrita aos *hardwares* construídos por ela. Portanto, somente os dispositivos da própria Apple é que executam com sucesso o iOS (MILANI, 2012). Logo, pode-se perceber que os famosos *hackintosh* (computadores não fabricados pela Apple que possuem o sistema operacional iOS), além de ilegal com certeza não possuem a eficiência do sistema operacional híbrido e o usuário também não irá usufruir da capacidade original do sistema operacional.

A sensação de utilização do sistema operacional híbrido acontece no momento em que o usuário começa a interagir com o dispositivo, ou seja, a interação do sistema operacional com o usuário é imediata, pois esse sistema possibilita ao usuário, realizar ações através de gestos, toques na tela, deslizar dos dedos, ampliar fotos, reduzir imagens, digitar mensagens em teclado virtual entre outros (SIMÕES E PEREIRA, 2014).

2.4 Arquitetura iphone iOS

A arquitetura do iOS[2] é concebida de quatro camadas: Cocoa Touch, Media Services, Core Services e Core OS e cada uma delas, disponibiliza um conjunto de frameworks que podem ser utilizados no desenvolvimento de aplicações para dispositivos móveis (Silva, Pires e Neto, 2015). Segundo Anvaari e Jansen (2010), arquitetura do iOS é semelhante a arquitetura básica encontrada no Mac OS X.

---

2 https://developer.apple.com/library/content/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/Introduction/Introduction.html#//apple_ref/doc/uid/TP40007898

Deste modo, quem possui computador de mesa utilizando o sistema operacional leva grande vantagem no manuseio do smartphone que por sua vez possui aplicativos nativos como o navegador safari. Os aplicativos desenvolvidos para o iOS raramente se comunicam diretamente com o *hardware* do dispositivo, ao invés disso, os aplicativos se comunicam com o *hardware* através de um conjunto de interfaces de sistema bem definidas que protegem seu aplicativo de alteração de *hardware* (APPLE, 2017).

2.4.1 Cocoa Touch

Os principais frameworks para o desenvolvimento de aplicações são encontrados na camada Cocoa Touch. Ela determina a infraestrutura para as tecnologias fundamentais, tais como multitarefa, serviço de notificação Apple push e diversos serviços de alto nível do sistema (Rocha e Neto, 2011). Portanto, percebe-se que essa camada é encarregada de oferecer ferramentas de infraestrutura para implementar eventos e aplicações para a interface do iphone. O Cocoa Touch foi desenvolvido em Objective-C e é baseada na Cocoa API (Application Programming Interface) utilizado no Mac OS no desenvolvimento de aplicações desktops.

A Apple realiza várias alterações na camada com objetivo de atender as necessidades de *hardware* nos dispositivos iOS.

2.4.2 Media Services

Essa camada é responsável por fornecer recursos de áudio, vídeo e gráficos e disponibilizar uma série de frameworks que podem ser utilizados no desenvolvimento de aplicações multimídia para dispositivos móveis iOS (MENDONÇA, BITTAR E DIAS, 2011).

Contudo, percebe-se que ela foi projetada para tornar mais prática a implementação de aplicações com o uso do framework IUKit (User Interface Kit) que por sua vez disponibiliza recursos como Core Graphics, Core Animation, OpenGL ES, Texto Core, Image I/O e Framework Assets Library que são tecnologias de implementação de gráficos e animações, tecnologia de áudio que possibilita o dispositivo reproduzir e gravar áudios de alta qualidade com a utilização dos seguintes frameworks: Media Player, AV Foundation e Core Audio, cada uma com funções que variam de alto para baixo nível.

E a tecnologia de vídeo que pode ser utilizada para realizar a reprodução e gravação de conteúdo baseados em vídeos, com ela o determinado desenvolvedor pode utilizar suas tecnologias para capturar e incorporar vídeos em sua aplicação com os frameworks UI ImagePickerController (presente na UIKit), Media Player, Av Fundation e Core Media (ROCHA, MENDES e NETO, 2011).

2.4.3 Core Services

Na camada Core Services existem serviços essenciais do sistema que todos os aplicativos utilizam, entre esses serviços destacam-se os Frameworks Core Foundation e Foundation. Mesmo se esses serviços não forem utilizados diretamente pelo desenvolvedor, partes do sistema são construídas em cima dele (APPLE, 2017).

Portanto, percebe-se que a camada Core Services é responsável pela gestão dos serviços fundamentais que um aplicativo iOS nativo usa. As tecnologias que se destacam nesta camada são grand central dispatch, in-app purchase, SQLite e XML support.

Esta camada também contém tecnologias individuais para suportar recursos como localização, iCloud, redes sociais e redes.

2.5 Camada Core OS

Essa camada inclui características de baixo nível que foram utilizadas na prática de outras tecnologias, em ocasiões onde o determinado desenvolvedor tem a necessidade de lidar nitidamente com a segurança ou comunicação com acessório de *hardware* externo, ele pode fazer isso utilizando os frameworks presentes nessa camada (APPLE, 2017).

Devido a característica de manipulação do *hardware* externo percebe-se que a função da camada Core OS é fornecer suporte ao *hardware* do dispositivo iOS onde o desenvolvedor poderá ter o controle sobre a segurança do dispositivo e comunicação, através dos frameworks Accelerate e External Accessory.

O framework Accelerate disponibiliza uma variedade de interfaces para trabalhar com processamento de sinais digitais, álgebra linear e cálculos de processamento de imagens. Outros recursos que podem ser apresentados são as diversas funções matemáticas que possibilitam realizar trabalhos com cálculos complexos (APPLE, 2017).

A vantagem de utilizar este framework são as interfaces otimizadas para todas as configurações de *hardware* presentes em dispositivos baseados no iOS. Portanto quem desenvolve para iOS tem a possibilidade de escrever o código uma vez e ter a certeza que ele será executado de forma eficiente em todos os dispositivos (APPLE, 2017).

O framework External Accessory é uma tecnologia que possibilita a comunicação da parte lógica do dispositivo com acessórios de *hardware* conectados a ele através de uma interface de comunicação (ROCHA e NETO, 2011).

Os acessórios podem ser conectados através de um conector dock de 30 pinos ou via Bluetooth. Por meio deste framework, o desenvolvedor pode realizar o controle dos acessórios diretamente utilizando comandos suportados por ele.

3 PROCEDIMENTOS METODOLÓGICOS

O trabalho em questão é uma pesquisa qualitativa, com objetivos descritivos, utilizando uma forma de estudo tendo como base materiais e métodos em pesquisas bibliográficas e *sites* oficiais que mostram o funcionamento dos sistemas operacionais.

A teoria imposta foi elaborada em revisões bibliográficas e pesquisas documentais, tendo em vista à análise de livros de vários autores, revistas e demais publicações sobre o tema proposto. Os critérios impostos na realização da comparação entre os sistemas operacionais são arquitetura, aplicativos básicos e o desenvolvimento de aplicações para a sua plataforma. Descrevendo como cada plataforma funciona.

4 RESULTADOS E DISCUSSÕES

A utilização dos sistemas operacionais Android e iOS para dispositivos móveis é referência no comércio de celulares, tanto para quem utiliza seus recursos para realizar determinadas tarefas quanto para quem tem interesse em desenvolver aplicações através de suas IDES (*Integrated Development Environment*), ou seja, devido os seus vários recursos esses sistemas operacionais vêm se destacando na atualidade perante o favoritismo dos consumidores.

O Android é uma plataforma inovadora de desenvolvimento para aplicativos móveis, inclui um sistema operacional baseado em Linux (kernel 2.6), uma interface visual requintada,

GPS, inúmeras aplicações já instaladas e traz um ambiente de desenvolvimento bastante poderoso, inovador e flexível (RUBIN, 2010).

Por ser *open source* permite que o usuário realize a troca do aplicativo do discador, SMS, agenda, calculadora, etc. Além de permitir também a alteração de ícones, tem suporte a *widgets* na tela inicial e até permite a instalação de *launchers* alternativos se o determinado usuário desejar.

O iOS é o sistema operacional exclusivo da Apple, provem do Mac OS X, lançado primeiramente para iPhone e também é usado em outros aparelhos da empresa (APPLE, 2017).

Por ser exclusivo da empresa, o sistema operacional não permite que o usuário faça alterações internas nas aplicações do dispositivo, ou seja, o usuário é obrigado a utilizar funcionalidades nos padrões Apple.

Outra diferença é que os aplicativos da Google podem ser utilizados normalmente nos dispositivos que utilizam o sistema operacional iOS (iPhones), enquanto aplicativos nativos da Apple não são encontrados no Android. E essa característica não se aplica apenas aos aplicativos, realizar a transferência de e-mails, contatos, músicas, filmes, fotos e documentos do Android para o iOS é muito mais fácil do que realizar essa transferência do iOS para o Android.

Portanto percebe-se que o Android é mais flexível que o iOS permitindo o usuário realizar inúmeras modificações no sistema operacional. Porém, essa flexibilidade prejudica a segurança do *software* da Google, por ele não ser tão restrito quanto o sistema operacional da Apple, é mais fácil realizar a instalação de *softwares* não autorizados no dispositivo, possibilitando a instalação de *softwares* maliciosos.

Como no *play store* é possível publicar aplicativos sem uma revisão detalhada do mesmo, fica evidente que é mais fácil para hackers iniciarem um ataque através de *softwares* maliciosos.

Por outro lado na *apple store* o processo de publicação de aplicativos é um tanto burocrática pois a Apple recolhe varias informações sobre a pessoa que está publicando e também faz uma revisão detalhada o determinado aplicativo, proporcionando uma segurança a mais.

Em relação a área de desenvolvimento para aplicativos, que vem crescendo em um ritmo acelerado e a procura por profissionais capacitados é imensa, o Android utilizava a IDE Eclipse que foi até um tempo atrás o ambiente mais utilizado para o desenvolvimento de

aplicações para Android destacando-se pela facilidade de gerenciamento de *plug-ins* (LECHETA, 2016).

Contudo, uma nova ferramenta de desenvolvimento foi criada, o Android Studio, uma IDE criada pela Google e JetBrains para realizar o desenvolvimento de aplicativos. Enquanto para iOS a ferramenta de desenvolvimento sempre foi o XCode.

O Android Studio é um ambiente de desenvolvimento essencial na criação de aplicações para a plataforma Android. Tem como característica a IDE *IntelliJ Community Version*, por consequência, desenvolvedores que vêm da plataforma Eclipse sentem dificuldade para se acostumar com o ambiente, especialmente pelas estruturas visuais de desenvolvimento e teclas de atalho, contudo, para quem está iniciando no desenvolvimento Android, a IDE Android Studio se mostra mais didática, necessitando de menos configurações e aumentando significativamente a produtividade (ANDROID DEVELOPERS, 2017).

O XCode já é um ambiente totalmente integrado ao mac, ou seja, só é possível realizar o desenvolvimento de aplicações no próprio sistema operacional e com a utilização do *hardware* da Apple (Hubsch, 2012). Isso não é bom para o desenvolvedor que reside na região brasileira pois os equipamentos da Apple têm um alto custo não sendo de fácil acesso para quem tem interesse em aprender a desenvolver nesta plataforma, porém o XCode possui diversas ferramentas muito úteis, entre as mais importantes podemos citar o compilador Apple LLVM, que além de compilar o código fornece uma ferramenta inteligente para completar o mesmo.

Para quem está iniciando no desenvolvimento de aplicativos e pretende programar para iOS, a Apple oferece uma quantidade imensa de materiais através do *iTunes University* que disponibiliza uma seção de vídeos para iniciantes onde a didática é colocada em prática e apresentada de modo bem intuitivo, ajudando o desenvolvedor a se familiarizar com plataforma.

O Android não disponibiliza guias em forma de vídeos como Apple faz, porém, os recursos apresentados referentes a desenvolvimento são bem mais estruturados do que o iOS, já que muitos projetos de código aberto Android podem ser utilizados como suporte para o desenvolvimento de novos projetos. No caso do iOS, por causa do acordo de não divulgação adotada pela Apple, não existem muitas fontes em termos de código fonte aberto para iPhone, ou seja, por ser um sistema operacional reservado o desenvolvedor iniciante tem um pouco de dificuldade para começar a desenvolver para essa plataforma.

Os dois sistemas operacionais funcionam através de camadas, cada uma delas é responsável por uma determinada função.

Ambos os sistemas operacionais têm a sua disposição uma camada onde é possível executar aplicações sobre a plataforma, no Android essa camada chama-se Camada de Aplicações e no iOS Cocoa Touch.

No Android, a camada é responsável por executar aplicações nativas do sistema operacional, porém ela também permite executar aplicações feitas por terceiros, uma característica que no iOS não é possível, a camada Cocoa Touch só permite o funcionamento de aplicações nativas do iOS.

5 CONSIDERAÇÕES FINAIS

Neste trabalho, foram relatadas diferenças entre os CORE dos sistemas operacionais móveis Android e iOS, apresentando pontos específicos onde cada sistema operacional se destaca, e retratando suas arquiteturas para o desenvolvedor que tem interesse em desenvolver aplicações para um desses sistemas operacionais.

A principal conclusão é que esses sistemas operacionais são bem distribuídos, cada um com suas funções e características para o usuário e o desenvolvedor, o iOS é reservado e o Android é um tanto restrito possuem frameworks com suas respectivas funcionalidades que por sua vez ficam presentes em suas camadas.

Como tendências futuras é possível constatar que cada sistema operacional irá evoluir com suas funcionalidades de modo a sempre surpreender o usuário e o desenvolvedor com inovações, tanto em funcionalidades para o usuário comum, quanto em novos frameworks para dar um suporte a mais para os desenvolvedores no geral.

REFERÊNCIAS BIBLIOGRÁFICAS:

ANDROID DEVELOPERS. **Conheça o Android Studio**. Disponível em: < https://developer.android.com/studio/intro/index.html >. Acessado em: 13 agosto 2017.

ANDROID DEVELOPERS. **Activities.** Disponível em:< https://developer.android.com/guide/components/activities.html?hl=pt-br >. Acessado em: 11 de novembro de 2017.

APPLE DEVELOPERS. **About the iOS Technologies**. Disponível em:< https://developer.apple.com/library/content/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/Introduction/Introduction.html#//apple_ref/doc/uid/TP40007898 > Acessado em: 10 maio 2017.

APPLE DEVELOPERS. **Cocoa Touch Layer**. Disponível em: <https://developer.apple.com/library/content/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/iPhoneOSTechnologies/iPhoneOSTechnologies.html#//apple_ref/doc/uid/TP40007898-CH3-SW1 > Acessado em: 13 maio 2017.

APPLE DEVELOPERS. **Media Layer**. Disponível em: <https://developer.apple.com/library/content/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/MediaLayer/MediaLayer.html#//apple_ref/doc/uid/TP40007898-CH9-SW4 > Acessado em: 13 maio 2017.

APPLE DEVELOPERS. **Core Services Layer**. Disponível em: <https://developer.apple.com/library/content/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/CoreServicesLayer/CoreServicesLayer.html#//apple_ref/doc/uid/TP40007898-CH10-SW5 > Acessado em: 13 maio 2017.

APPLE DEVELOPERS. **Core OS Layer,** 2017. Disponível em: <https://developer.apple.com/library/content/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/CoreOSLayer/CoreOSLayer.html#//apple_ref/doc/uid/TP40007898-CH11-SW1> Acessado em: 13 maio 2017.

ANVAARI, M; JANSEN, S. - **Evaluating architectural openness in software platforms. In Proceedings of the Fourth European Conference on Software Architecture**, 2010. ANVAARI, M. e JANSEN, S. Companion Volume, ECSA '10, pages 85–92, New York, NY, USA. ACM.

EL PAÍS. **Android já é o sistema operacional mais usado do mundo**. Disponível em: <https://brasil.elpais.com/brasil/2017/04/04/tecnologia/1491296467_396232.html> Acessado em: 03 agosto 2017.

EXTRA. **Empresas brasileiras usam mais o celular para fins de trabalho**. Disponível em:< http://extra.globo.com/noticias/celular-e-tecnologia/empresas-brasileiras-usam-mais-celular-para-fins-de-trabalho-7040799.html >Acessado em: 05 jun. 2017.

GIROLLETE, Rowan Ben-hur Andrighetti. **Android: Visão Geral**. Universidade de Cascavel - Alfa Brasil, Cascavel. Disponível em:< http://www.academicoo.com/texto-completo/android-visao-geral > Acessado em: 14 maio 2017.

HERCULES, José Roberto Leiva. **Análise de Impacto da Máquina Virtual ART em Sistemas Android**. Universidade Federal do Rio Grande do Sul - Instituto de Informática, 2016, Rio Grande do Sul. Disponível em:<http://www.lume.ufrgs.br/handle/10183/150968> Acessado em 05/08/2017.

HUBSCH, Eduardo. **Uma Abordagem Comparativa do desenvolvimento de aplicações para dispositivo Móveis**. Faculdade de Tecnologia de São Paulo, 2012, São Paulo. Disponível em: < http://www.fatecsp.br_dti_tcc_tcc00065> Acessado em: 07 maio 2017.

LECHETA, Ricardo R. - **Google Android -** Aprenda a criar aplicações para dispositivos móveis com Android SDK. / Ricardo R. Lecheta. - 5 ed. ver. e ampl. - São Paulo: Novatec Editora, 2016. p. 28 a 29.

MILANI, A. - **Programando para Iphone e Ipad:** aprenda a construir aplicativos o IOS. 2 ed. São Paulo: Novatec Editora, 2012.

OLHAR DIGITAL. **Saiba quais são as marcas que mais vendem smartphones no mundo**. Disponível em: <https://olhardigital.uol.com.br/noticia/saiba-quais-sao-as-marcas-que-mais-vendem-smartphones-no-mundo/61446> Acessado em: 07 abril 2017.

OTHON BATISTA. **Apostila de Android – Programando Passo a Passo**. Disponível em: <http://othonbatista.com/arquivos/android/apostila-android.pdf> Acessado em: 15 maio 2017.

PEREIRA, Lúcio Camilo Oliva; SILVA, Michel Lourenço – **Android – Para desenvolvedores. /** PEREIRA, Lúcio Camilo Oliva; SILVA, Michel Lourenço. - Rio de Janeiro: Brasport Livro e Multimídia Ltda, 2009. p.2 a 8.

RUBIN, A. **Google's Rubin: Android "a revolution"**. Disponível em <http:// http://news.cnet.com/8301-1023_3-10245994-93.html?tag=mncol /> Acesso em 03 jun. 2017.

SIMÕES, Danielle D.; PEREIRA, Júlio C. **Operacionais Móveis - Android X iOS.** Universidade Paranaense – Unipar, 2014, Paranavaí. Disponível em:< http://web.unipar.br/~seinpar/2014/artigos/graduacao/daniellediassimoes.pdf> Acessado em: 07 maio 2017.

SETTING. **Entenda o poder do celular como ferramenta colaborativa**. Disponível em:<http://www.setting.com.br/blog/entenda-o-poder-celular-como-ferramenta-colaborativa/> Acessado em: 09 agosto 2017.

SILVA, Luquetti B; PIRES, Daniel Facciolo; NETO, Silvio Carvalho. **Desenvolvimento de Aplicações para Dispositivos Móveis: Tipos e Exemplo de Aplicação na plataforma iOS**. . Centro Universitário Municipal de Franca - Uni Facef, 2015, Franca. Disponível em: < http://www.lbd.dcc.ufmg.br/colecoes/wicsi/2015/004.pdf > Acessado em: 19 julho 2017.

ROCHA, Adriano Mendonça; NETO, Roberto Mendes Finzi. **Introdução a Arquitetura Apple iOS**. Universidade Federal de Goiás - Campus Catalão, 2011, Goiás. Disponível em:< http://www.enacomp.com.br/2011/anais/trabalhos-aprovados/pdf/enacomp2011_submission_58.pdf > Acessado em 24/05/2017.