



DESENVOLVIMENTO DE APLICATIVOS NATIVOS ANDROID E IOS PARA O RESTAURANTE UNIVERSITÁRIO DA UFRJ

Felipe Podolan Oliveira

Projeto de Graduação apresentado ao Curso de Engenharia de Computação e Informação da Escola Politécnica da Universidade Federal do Rio de Janeiro como parte dos requisitos necessários para a obtenção do grau de Engenheiro de Computação e Informação.

Orientador: Flávio Luis de Mello

Rio de Janeiro
Agosto de 2018

DESENVOLVIMENTO DE APLICATIVOS NATIVOS ANDROID E IOS PARA
O RESTAURANTE UNIVERSITÁRIO DA UFRJ

Felipe Podolan Oliveira

PROJETO SUBMETIDO AO CORPO DOCENTE DO CURSO DE
ENGENHARIA DE COMPUTAÇÃO E INFORMAÇÃO DA ESCOLA
POLITÉCNICA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO
GRAU DE ENGENHEIRO DE COMPUTAÇÃO E INFORMAÇÃO.

Examinadores:

Prof. Nome do Primeiro Examinador Sobrenome, D.Sc.

Prof. Nome do Segundo Examinador Sobrenome, Ph.D.

RIO DE JANEIRO, RJ – BRASIL
AGOSTO DE 2018

Podolan Oliveira, Felipe

Desenvolvimento de aplicativos nativos Android e iOS para o Restaurante Universitário da UFRJ/Felipe Podolan Oliveira. – Rio de Janeiro: UFRJ/POLI – COPPE, 2018.

V, 15 p.: il.; 29, 7cm.

Orientador: Flávio Luis de Mello

Projeto (graduação) – UFRJ/ Escola Politécnica/ Curso de Engenharia de Computação e Informação, 2018.

Referências Bibliográficas: p. 14 – 15.

1. Computação móvel. 2. Android. 3. iOS. I. de Mello, Flávio Luis. II. Universidade Federal do Rio de Janeiro, Escola Politécnica/ Curso de Engenharia de Computação e Informação. III. Título.

Sumário

Lista de Figuras	v
1 Introdução	1
1.1 Tema	1
1.2 Delimitação	1
1.3 Justificativa	1
1.4 Objetivo	2
1.5 Metodologia	2
1.6 Materiais	3
1.7 Descrição	3
2 Conceitos Iniciais	4
2.1 Kit de Desenvolvimento de Software (SDK)	4
2.2 Interface Gráfica (UI)	4
2.3 Experiência do Usuário (UX)	5
2.4 Web Service	6
2.4.1 Rede de computadores	7
2.4.2 API	8
2.5 Criação de projetos para Android e para iOS	10
2.5.1 Android Studio	10
2.5.2 XCode	13
Referências Bibliográficas	14

Lista de Figuras

2.1	Reclamações mais comuns de usuários de aplicações móveis	6
2.2	Utilização de internet via dispositivos móveis versus computadores . .	7
2.3	Arquitetura Cliente/Servidor	7
2.4	Camadas e protocolos de uma rede de computadores	8
2.5	Arquitetura de uma API	9
2.6	Novo projeto no Android Studio - Passo 1	10
2.7	Novo projeto no Android Studio - Passo 2	10
2.8	Novo projeto no Android Studio - Passo 3	11
2.9	Novo projeto no Android Studio - Passo 4	11
2.10	IDE Android Studio	12
2.11	Selecionar dispositivo no Android Studio	12
2.12	Exemplo com Empty Activity	13

Capítulo 1

Introdução

1.1 Tema

O tema deste trabalho é o desenvolvimento de aplicativos móveis. Neste sentido, os problemas a serem resolvidos são a viabilidade em criar aplicações para plataformas móveis, bem como, a capacidade destas aplicações de ampliar a acessibilidade aos serviços oferecidos e de melhorar a experiência de uso dos usuários.

1.2 Delimitação

O objetivo do projeto é atender aos usuários e aos funcionários do Restaurante Universitário do Centro de Tecnologia e de Letras da UFRJ. O escopo de usuários é variável sendo composto pelos alunos, professores e servidores da instituição, bem como por eventuais visitantes que vão à universidade para congressos, semanas esportivas, dentre outros. O projeto também tem capacidade de atender às outras unidades do RU da UFRJ, bem como ser adaptado para outras universidades.

1.3 Justificativa

O Restaurante Universitário da UFRJ tem como objetivo oferecer alimentação de qualidade, equilibrada, e acessível de forma a favorecer a permanência dos estudantes no espaço universitário, permitindo-lhes dedicação integral aos estudos, sendo importante meio de combate à evasão escolar.

Entretanto, apesar de ser benéfico à comunidade universitária, o RU enfrenta alguns problemas, sendo o principal as filas de espera. Pensando nisso, a Decania do Centro de Tecnologia resolveu criar um sistema de agendamento online cujo objetivo é alocar os horários de entrada no RU e, desta forma, tornar as filas, que antes eram físicas e que geravam desgaste aos alunos, em filas virtuais.

Esse projeto funciona desde 2016 na unidade do RU do CT. Os agendamentos são feitos através do website www.ru.ct.ufrj.br. Entretanto, para melhorar ainda mais a experiência dos usuários, bem como para ampliar a acessibilidade aos serviços prestados pelo RU, a Decania do CT resolveu desenvolver aplicativos nativos para smartphone tendo em vista a crescente popularização desta modalidade, notoriamente para as plataformas Android e iOS.

1.4 Objetivo

O objetivo geral é, portanto, desenvolver aplicações móveis nativas em Android (Java) e em iOS (Swift) capazes de fornecer os serviços virtuais do Restaurante Universitário, sendo o principal deles o agendamento de horários para entrar no restaurante. Desta forma, tem-se como objetivos específicos principais:

- Permitir a criação, edição e exclusão de filas virtuais para o acesso ao RU por parte de seus funcionários;
- Permitir o agendamento em filas virtuais de acesso ao RU, bem como sua exclusão ou verificação de status por parte dos alunos;
- Visualização das filas virtuais vigêntes por parte de todos os usuários do RU.

1.5 Metodologia

Este trabalho utilizou os kits de desenvolvimento de software (SDK, do inglês Software Development Kit) oficiais oferecidos pela Google e pela Apple para suas plataformas Android e iOS, respectivamente.

Atendendo às boas práticas de programação e aos conceitos de orientação a objetos, foram desenvolvidos três aplicativos móveis. São eles:

- Aplicativo Android para clientes do RU;
- Aplicativo iOS para clientes do RU;
- Aplicativo Android para funcionários do RU.

As etapas da criação de cada aplicação consistem em:

- Criação de uma interface para cada funcionalidade;
- Criação da lógica de interação com o usuário para alternar entre as interfaces;
- Criação de métodos para a comunicação com o Back-End;
- Atualização dos dados exibidos na interface ou alternância de interface de acordo com a resposta do Back-End.

A proposta deste trabalho é mostrar como cada etapa do projeto foi desenvolvida para os aplicativos para os clientes do RU com respaldo na literatura de desenvolvimento de aplicativos para Android e para iOS. O aplicativo para os funcionários foi desenvolvido apenas para a plataforma Android.

Além disso, são expostas as possíveis tecnologias a serem usadas para a comunicação com o Back-End, como por exemplo, os estilos de arquitetura de serviços REST versus SOAP em APIs. O desenvolvimento da API de comunicação e do Back-End, entretanto, não fazem parte do escopo deste projeto, pois os mesmos já estavam em desenvolvimento por outro aluno desde o início do projeto.

O êxito deste trabalho está centrado na entrega de todas as funcionalidades dos aplicativos.

1.6 Materiais

Foi utilizado um notebook pessoal para o desenvolvimento do projeto. Trata-se de um Macbook, pois é necessário para desenvolvimento nativo iOS e atende às demandas para desenvolvimento Android.

Os softwares utilizados são gratuitos. Para desenvolvimento iOS, foi utilizado o software XCode, e para desenvolvimento Android, o software Android Studio.

1.7 Descrição

O capítulo 2 deste trabalho apresenta alguns conceitos iniciais como interface gráfica (UI), experiência do usuário (UX) e API, discutindo algumas tecnologias da camada de aplicação da internet em sistemas comumente utilizados por aplicações móveis. Também apresenta as etapas de criação de um projeto em ambos os softwares Android Studio e XCode. O capítulo 3, por sua vez, explica mais detalhadamente as interfaces gráficas (ou seja, as "telas") tanto para Android, quanto para iOS, mostrando suas diferenças e possíveis formatos, com respaldo nos conceitos de Activity e de Fragment na plataforma Android e de Controller na plataforma iOS. Também é aprofundada a discussão sobre experiência do usuário, mostrando os conceitos de Intent e FragmentManager para Android e de NavigationController e Segue para iOS. Já o capítulo 4 descreve a implementação dos aplicativos do RU explicando cada componente do projeto bem como as decisões que foram tomadas ao longo de suas implementações. O último capítulo conclui o trabalho e discute possíveis trabalhos futuros.

Capítulo 2

Conceitos Iniciais

2.1 Kit de Desenvolvimento de Software (SDK)

Kit de Desenvolvimento de Software, comumente abreviada para SDK (do inglês, Software Development Kit) é um pacote que contém ferramentas, bibliotecas, códigos fonte, e outros utilitários pré criados. [1] A Google e Apple oferecem respectivamente o Android SDK e o iOS SDK, que possibilitam o desenvolvimento de aplicativos para suas plataformas.

2.2 Interface Gráfica (UI)

Interface gráfica, comumente abreviada para UI (do inglês, User Interface) é a parte da aplicação que o usuário vê e com a qual ele interage. [2] A interface inclui as telas, janelas, controles, menus e etc. [2]

De acordo com um estudo publicado pelo IEEE [3] sobre os motivos pelos quais os usuários mais reclamam de aplicações móveis, a interface gráfica é um dos casos mais frequentes, o que demonstra a necessidade de empenho no desenvolvimento de interfaces agradáveis e intuitivas.

Não é de se estranhar que as empresas desenvolvedoras dos sistemas operacionais cujas plataformas móveis são as mais competitivas no mercado ofereçam maneiras de padronizar as interfaces dos aplicativos feitos para suas plataformas.

De fato, a Google e a Apple oferecem dentro de seus SDKs ferramentas de padronização de interfaces. No caso da Google, esse conjunto de ferramentas é chamado de Material Design [4] e, além de ser oferecido para o Android, também pode ser utilizada nas plataformas Web, Flutter e, inclusive, iOS. Já a Apple, oferece seu kit gráfico, o UIKit [5], apenas para sua própria plataforma.

2.3 Experiência do Usuário (UX)

Um segundo conceito muito importante quando se trata de aplicações móveis é a Experiência do Usuário, comumente abreviada para UX (do inglês, User Experience). Apesar de ser um conceito amplamente disseminado e aceito, ele ainda não é bem definido ou entendido. [6]

Entretanto, para uma certa compreensão, pode-se utilizar a definição de Hassenzahl Tractinsky:

“Experiência de usuário é a consequência do estado interno de um usuário (predisposição, expectativa, necessidade, motivação, humor, etc), das características do sistema designado (complexibilidade, propósito, usabilidade, funcionalidade, etc) e do contexto (ou ambiente) no qual a interação ocorre (configuração organizacional/social, significância da atividade, voluntariedade de uso, etc).” [6]

Pode se dizer que a Experiência do Usuário consiste nas sensações que um produto gera em um usuário que, conseqüentemente, forma opiniões a respeito deste produto com respaldo nessas sensações.

A importância da Experiência do Usuário fica ainda mais clara ao notar que todos os itens mencionados no estudo do IEE [3] resumidos na tabela a seguir podem ser relacionados à UX.

Infere-se da tabela acima que um usuário de aplicações móveis espera que a aplicação

- Não trave;
- Seja compatível com seu dispositivo;
- Atenda somente às funcionalidades esperadas;
- Explícite as cobranças utilizadas na aplicação;
- Possua uma boa interface gráfica;
- Possua boa conectividade com a Internet;
- Atenda às condições de privacidade e ética esperadas da aplicação;
- Consuma o mínimo possível de recursos de hardware;
- Seja interessante;
- Seja responsiva.

Complaint type	Description	Example review
App Crashing	The app often crashed.	"Crashes immediately after starting."
Compatibility	The app had problems on a specific device or an OS version.	"I can't even see half of the app on my iPod Touch."
Feature Removal	A disliked feature degraded the user experience.	"This app would be great, but get rid of the ads!"
Feature Request	The app needed additional features.	"No way to customize alerts."
Functional Error	The problem was app specific.	"Not getting notifications unless you actually open the app."
Hidden Cost	The full user experience entailed hidden costs.	"Great if you weren't forced to buy coins for REAL money."
Interface Design	The user complained about the design, controls, or visuals.	"The design isn't sleek and isn't very intuitive."
Network Problem	The app had trouble with the network or responded slowly.	"New version can never connect to server!"
Privacy and Ethics	The app invaded privacy or was unethical.	"Yet another app that thinks your contacts are fair game."
Resource Heavy	The app consumed too much energy or memory.	"Makes GPS stay on all the time. Kills my battery."
Uninteresting Content	The specific content was unappealing.	"It looks great, but the actual gameplay is boring and weak."
Unresponsive App	The app responded slowly to input or was laggy overall.	"Bring back the old version. Scrolling lags."
Not Specific	The user's comment wasn't useful or didn't point out a problem.	"Honestly the worst app ever."

Figura 2.1: Reclamações mais comuns de usuários de aplicações móveis [3]

2.4 Web Service

Um item notório na publicação do IEEE [3] previamente citada é a conectividade com a Internet. Apesar de a pesquisa ter sido feita nos Estados Unidos, a realidade no Brasil não é muito diferente. Segundo o IBGE [7], em 2014, pela primeira vez, o uso do telefone celular para acessar a Internet ultrapassou o uso de microcomputador nos domicílios brasileiros. [7] Em 2015, esse cenário se repetiu no País. [7] Conforme o mostra gráfico a seguir:

Ao comunicar-se com a Internet um aplicativo passa a fazer parte de um web service.

Um web service é um sistema de software designado a suportar interações máquina-com-máquina sobre uma rede de computadores. [8] Num sistema web existem dois agentes: uma máquina que provém um serviço, conhecida como servidor, e uma máquina que requisita o serviço, chamada de cliente. Existem diversas arquiteturas de comunicação entre o cliente e o servidor, uma delas é através de APIs. [9]

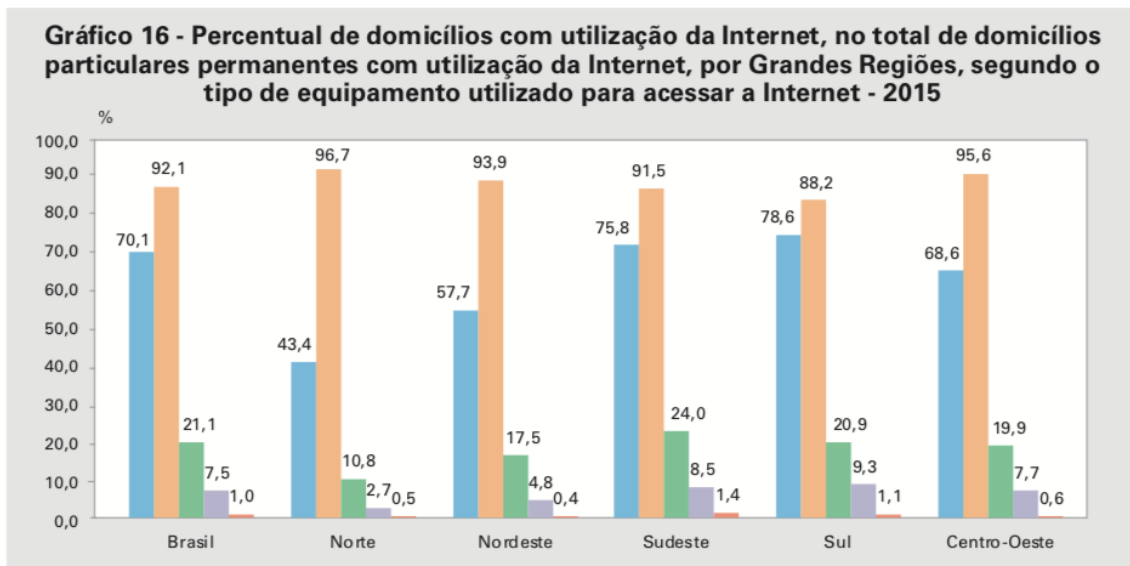


Figura 2.2: Utilização de internet via dispositivos móveis versus computadores [7]



Figura 2.3: Arquitetura Cliente/Servidor [9]

2.4.1 Rede de computadores

Para viabilizar a comunicação entre o cliente e o servidor, um web service necessita de uma rede de computadores. Uma rede de computadores é um conjunto de computadores autônomos interconectados por uma única tecnologia. [9] A World Wide Web (Web), por exemplo, é uma rede de redes de computadores, [9] já que utiliza mais de uma tecnologia em suas conexões. A maioria das redes é organizada como uma pilha de camadas ou níveis, colocadas umas sobre as outras. [9] O objetivo de cada camada é oferecer determinados serviços às camadas superiores, isolando essas camadas dos detalhes de implementação desses recursos. [9]

A camada n de uma máquina se comunica com a camada n de outra máquina. Coletivamente, as regras e convenções usadas nesse diálogo são conhecidas como o protocolo da camada n . [9]

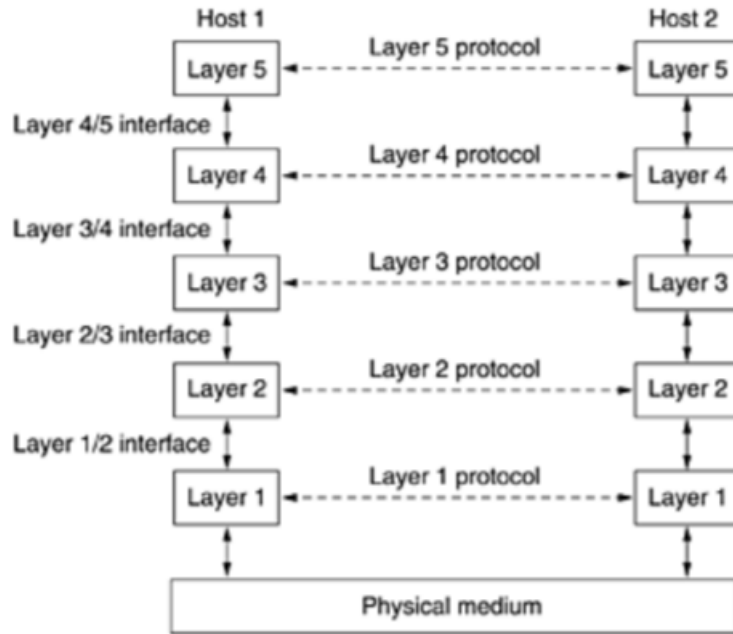


Figura 2.4: Camadas e protocolos de uma rede de computadores [9]

Um protocolo amplamente conhecido é o HTTP que é usado na Camada de Aplicação do modelo de referência TCP/IP e cuja principal função é buscar páginas na World Wide Web. [9]

2.4.2 API

Pessoas usam aplicações via Interfaces Gráficas. Já aplicações usam outras aplicações via API (do inglês, Application Programming Interface). [10] APIs provêm as capacidades essenciais para conectar, estender e integrar software. [10] Com uma mesma API é possível acessar os serviços de um servidor em muitos clientes distintos.

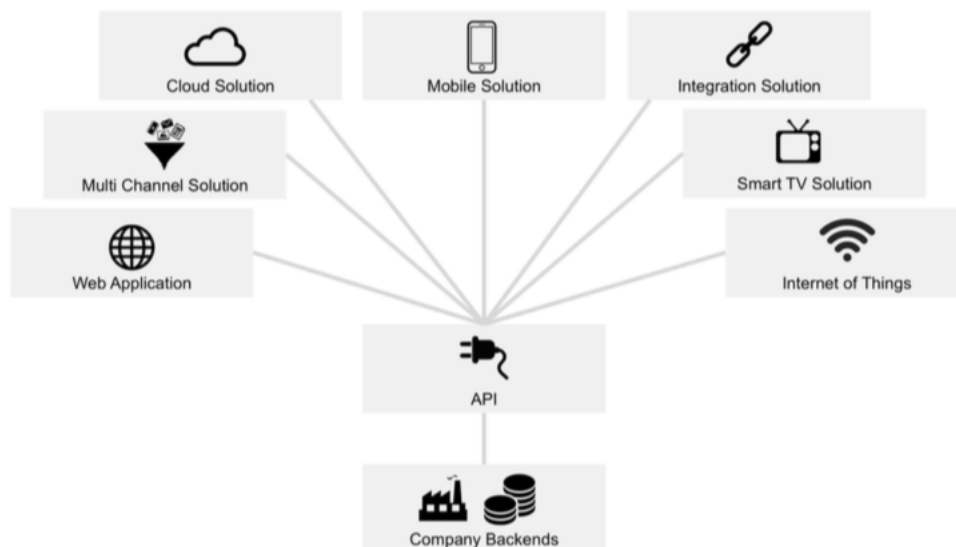


Figura 2.5: Arquitetura de uma API [10]

Existem diversos estilos de arquiteturas de implementação de APIs. Dentre eles, o REST (do inglês Representation State Transfer), o RPC (do inglês, Remote Procedure Call) e o SOAP (do inglês, Simple Object Access Protocol). Sendo hoje em dia as mais comuns o REST e o SOAP. [11]

REST

REST é um estilo de arquitetura para serviços que define um conjunto de restrições e acordos na arquitetura. Um serviço que cumpre as regras REST é chamado de RESTful. [10] O REST foi desenvolvido para fazer uso otimizado de infraestruturas baseadas em HTTP. [10]

Mensagens REST possuem pouco payload, o que é adequado para aplicações móveis. [12]

SOAP

SOAP é o protocolo padrão de interface para web services proposto pelo W3C e é aplicável a qualquer protocolo da camada de aplicação (HTTP, SHTTP, FTP, etc). [11]

Entretanto, as mensagens SOAP contêm muitos metadados e apenas suportam estruturas verbosas de XML para solicitações e respostas. Além disso, devido ao seu grande tamanho, serviços SOAP são considerados complexos para ambos os provedores e consumidores de serviços. [10]

2.5 Criação de projetos para Android e para iOS

2.5.1 Android Studio

O Android Studio é a IDE oficial para desenvolvimento de aplicativos para Android.

Depois de fazer o download e de instalar o software, o primeiro passo para criar uma aplicação Android é escolher a opção: “Start a new Android Studio project” ou equivalente caso o idioma de instalação não seja o inglês.

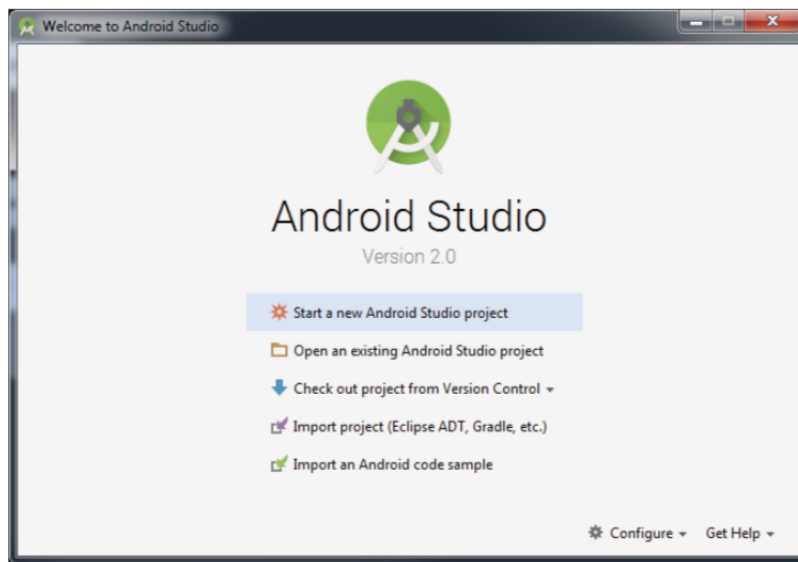


Figura 2.6: Novo projeto no Android Studio - Passo 1 [13]

Em seguida, deve-se nomear a aplicação, bem como escolher o domínio, o diretório onde os arquivos ficarão e o nome do pacote.

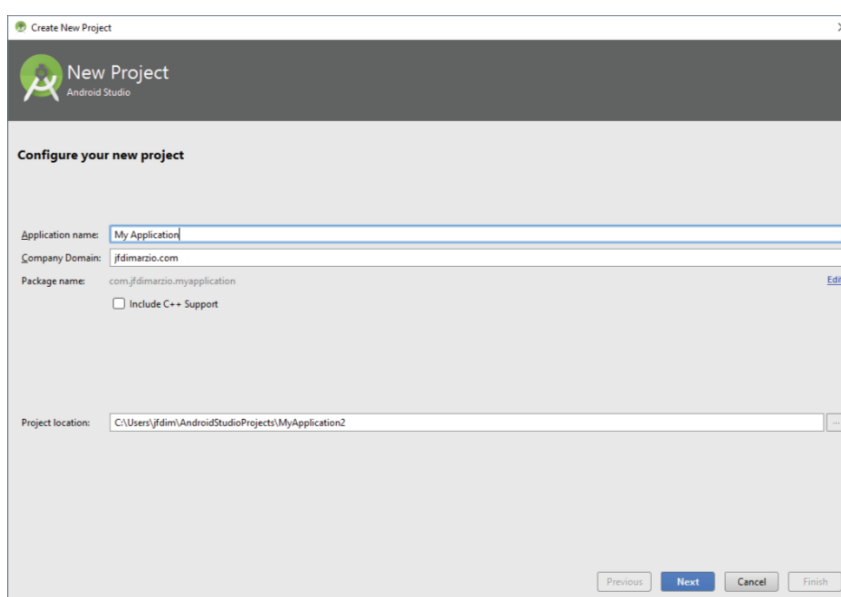


Figura 2.7: Novo projeto no Android Studio - Passo 2 [13]

Após, escolhe-se o SDK mínimo para cada opção que a aplicação será utilizada dentre: smartphones e tablets, wear e etc. O escopo deste trabalho se limita a smartphones e tablets. Quanto menor for a versão do SDK escolhido, maior será o número de dispositivos que suportarão a aplicação.

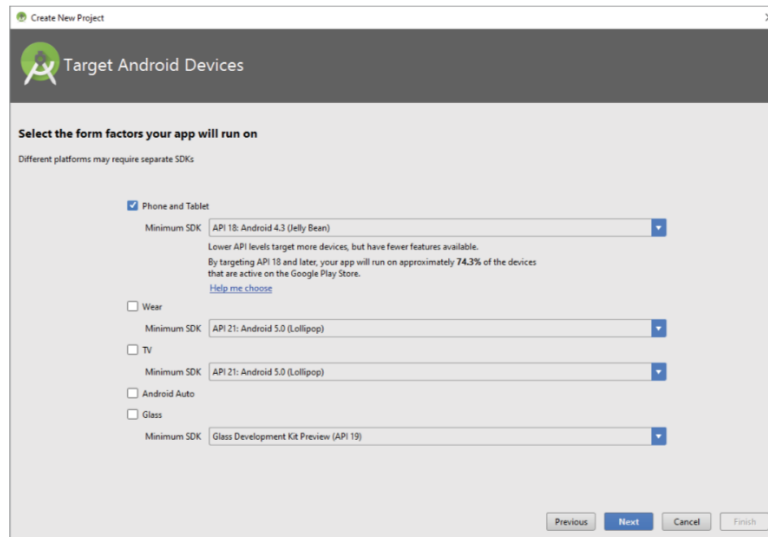


Figura 2.8: Novo projeto no Android Studio - Passo 3 [13]

Feito isso, aparecerão algumas opções de Activity para serem adicionadas à aplicação. O modelo mais básico é o “Empty Activity” e este modelo deve ser o escolhido num primeiro momento até que se obtenha um maior conhecimento da plataforma.

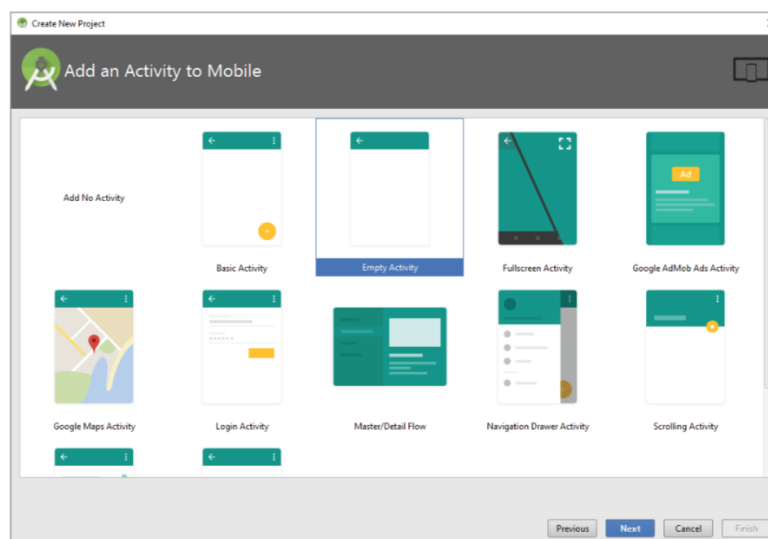


Figura 2.9: Novo projeto no Android Studio - Passo 4 [13]

A IDE será iniciada e alguns arquivos serão criados e podem ser selecionados no navegador à esquerda. Os detalhes de cada um desses arquivos será explicado com

mais detalhes no próximo capítulo.

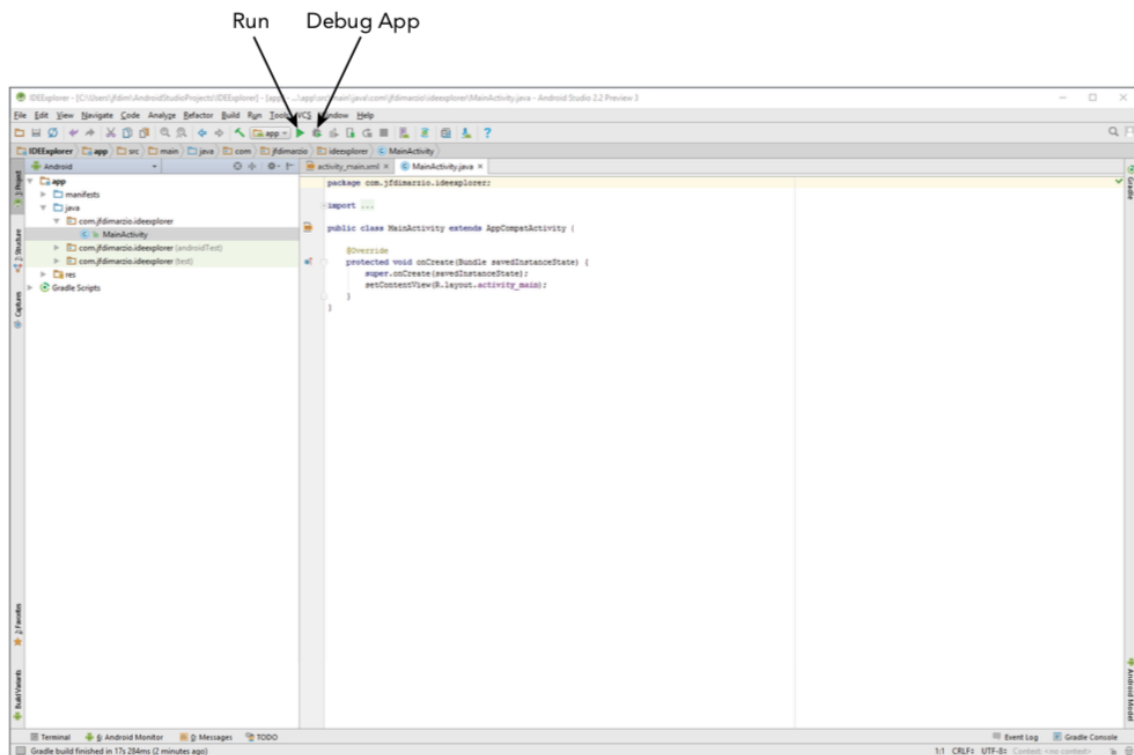


Figura 2.10: IDE Android Studio [13]

O código pré-escrito cria uma Activity vazia. Para visualizá-la basta clicar no botão “Run App”. É possível rodar as aplicações tanto em dispositivos físicos, quanto em dispositivos virtuais.

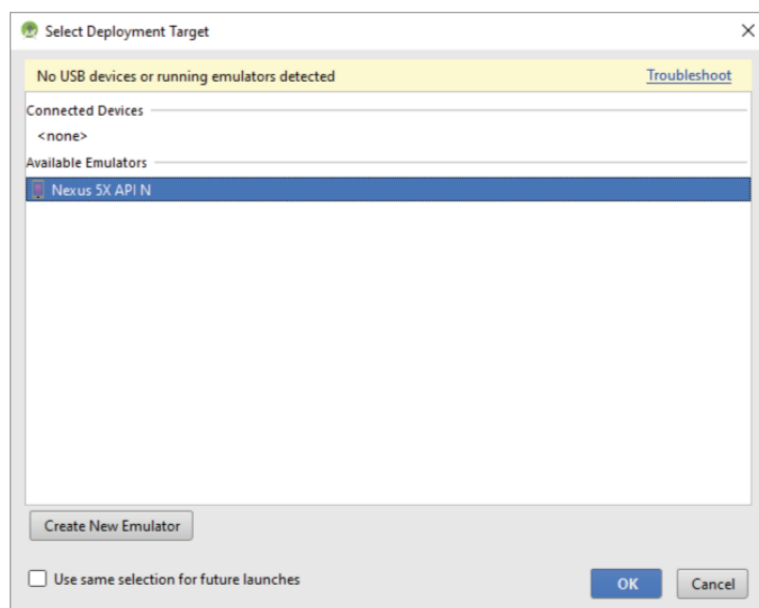


Figura 2.11: Selecionar dispositivo no Android Studio [13]

Compilando o exemplo criado em [13] num dispositivo virtual, obteve-se o seguinte resultado:



Figura 2.12: Exemplo com Empty Activity [13]

2.5.2 XCode

Referências Bibliográficas

- [1] A. Thorn, *Cross-Platform Game Development: Making PC Games for Windows, Linux and Mac*. WORDWARE PUB CO, 2008. [Online]. Available: https://www.ebook.de/de/product/7396496/alan_thorn_cross_platform_game_development_making_pc_games_for_windows_linux_and_mac.html
- [2] P. H. J. Chong, P. L. So, P. Shum, X. J. Li, and D. Goyal, “Design and implementation of user interface for mobile devices,” *IEEE Transactions on Consumer Electronics*, vol. 50, no. 4, pp. 1156–1161, 2004.
- [3] H. Khalid, E. Shihab, M. Nagappan, and A. E. Hassan, “What do mobile app users complain about?” *IEEE Software*, vol. 32, no. 3, pp. 70–77, 2015.
- [4] Google, “Material design,” May 2018. [Online]. Available: <https://material.io>
- [5] Apple, “Apple developer documentation,” May 2018. [Online]. Available: <https://developer.apple.com/documentation>
- [6] E. Law, V. Roto, A. P. O. S. Vermeeren, J. Kort, and M. Hassenzahl, “Towards a shared definition of user experience,” in *CHI’08 extended abstracts on Human factors in computing systems*. ACM, 2008, pp. 2395–2398.
- [7] I. C. de Trabalho e Rendimento, *Acesso a Internet e a televisao e posse de telefone movel celular para uso pessoal, 2015 : Pesquisa Nacional por Amostra de Domicilios*. Rio de Janeiro : IBGE, Instituto Brasileiro de Geografia e Estatística, 2016.
- [8] C. Ferris, “Web services architecture,” *Standard, W3C World*, p. 10, 2004.
- [9] A. S. Tanenbaum, *Redes de Computadores (Em Português do Brasil)*. Campus, 2003.
- [10] M. Biehl, *API Architecture: The Big Picture for Building APIs (API University Series) (Volume 2)*. CreateSpace Independent Publishing Platform, 2015.

- [11] P. A. Castillo, J. L. Bernier, M. G. Arenas, J. Merelo, and P. Garcia-Sanchez, “Soap vs rest: Comparing a master-slave ga implementation,” *arXiv pre-print arXiv:1105.4978*, 2011.
- [12] K. Wagh and R. Thool, “A comparative study of soap vs rest web services provisioning techniques for mobile host,” *Journal of Information Engineering and Applications*, vol. 2, no. 5, pp. 12–16, 2012.
- [13] J. DiMarzio, *Beginning Android Programming with Android Studio*. John Wiley & Sons, 2016.