

TAWAN GOMES MENEZES

DESENVOLVIMENTO DE UM PROTÓTIPO DE APLICAÇÃO MOBILE
ANDROID PARA AUTOMAÇÃO DO PROCESSO DE ELABORAÇÃO
DE ORÇAMENTO DE VENDA EM UMA DISTRIBUIDORA DE
COSMÉTICOS DA CIDADE DE TEÓFILO OTONI

FACULDADE UNIFICADAS DE TEÓFILO OTONI
TEÓFILO OTONI – MG
2018

TAWAN GOMES MENEZES

DESENVOLVIMENTO DE UM PROTÓTIPO DE APLICAÇÃO MOBILE
ANDROID PARA AUTOMAÇÃO DO PROCESSO DE ELABORAÇÃO
DE ORÇAMENTO DE VENDA EM UMA DISTRIBUIDORA DE
COSMÉTICOS DA CIDADE DE TEÓFILO OTONI

Monografia apresentada ao curso de Sistemas de Informação das
Faculdades Unificadas de Teófilo Otoni como requisito parcial à obtenção do
título de Bacharel em Sistemas de Informação.

Área de Concentração: Desenvolvimento Mobile

Orientador: Prof. Msc Amaury Gonçalves Costa

FACULDADE UNIFICADAS DE TEÓFILO OTONI
TEÓFILO OTONI – MG
2018



FACULDADES UNIFICADAS DE TEÓFILO OTONI
NÚCLEO DE TCC / SISTEMAS DE INFORMAÇÃO
Autorizado pela Portaria 4.012 de 06/123/2004 – MEC

FOLHA DE APROVAÇÃO


A monografia intitulada: *Desenvolvimento de um protótipo de aplicação mobile android para automação do processo de elaboração de orçamento de venda em uma distribuidora de Cosméticos da cidade de Teófilo Otoni,*

elaborada pelo aluno Tawan Gomes Menezes,


foi aprovada por todos os membros da Banca Examinadora e aceita pelo curso de Sistemas de Informação das Faculdades Unificadas de Teófilo Otoni, como requisito parcial da obtenção do título de

BACHAREL EM SISTEMAS DE INFORMAÇÃO.

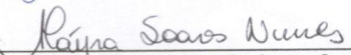
Teófilo Otoni, 5 de dezembro de 2018



Professor Orientador: Amáry Gonçalves Costa



Professor Examinador: Wilbert Viana Barbosa



Professor Examinador: Professor Convidado

LISTA DE FIGURAS

Figura 01 - Camadas da Engenharia de Software.....	24
Figura 02 - Tela inicial do FuidUI.....	35
Figura 03 - Tela do protótipo no FluidUI.....	36
Figura 04 - Criação do projeto no Firebase.....	37
Figura 05 - Tela inicial Andoid Studio.....	38
Figura 06 - Tela inicial Visual Studio Code.....	39
Figura 07 - Verificação da instalação do Composer.....	40
Figura 08 - Painel de controle de XAMPP.....	41
Figura 09 - Versões do Node.js e NPM.....	42
Figura 10 - Versões do Laravel.....	43
Figura 11 - Maneiras de instalar o AdminLTE.....	44
Figura 12 - Versão do Git instalada.....	45
Figura 13 - Modelo de Prototipagem.....	47
Figura 14 - Telas do Protótipo no FluidUI.....	48
Figura 15 - Criação do Projeto 01.....	50
Figura 16 - Criação do Projeto 02.....	50
Figura 17 - Criação do Projeto 03.....	51
Figura 18 - Criação do Projeto 04.....	52
Figura 19 - Estrutura de Classes.....	53

Figura 20 - Estrutura do Layout.....	54
Figura 21 - Criação do Bando de Dados.....	55
Figura 22 - Definição do tipo de aplicação.....	56
Figura 23 - Registro da aplicação.....	57
Figura 24 - Aquisição chave SHA-1 01.....	58
Figura 25 - Aquisição chave SHA-1 02.....	59
Figura 26 - Arquivo de configuração Firebase.....	60
Figura 27 - Tela das tabelas do Banco de Dados.....	64
Figura 28 - Tela dos usuários da aplicação.....	65
Figura 29 - Telas do Protótipo.....	66
Figura 30 - Criando aplicação web.....	68
Figura 31 - Telas do projeto da aplicação web.....	69
Figura 32 - Download do arquivo de configuração.....	70
Figura 33 - Arquivos de configuração.....	71
Figura 34 - Tela de login da aplicação web.....	71
Figura 35 - Telas de orçamentos da aplicação web.....	72
Figura 36 - Visualização do orçamento.....	73
Figura 37 - Teste de login.....	75
Figura 38 - Teste criação de orçamento.....	76
Figura 39 - Teste alteração de orçamento.....	77
Figura 40 - Teste visualização de orçamentos na web.....	78
Figura 41 - Teste verificação de orçamentos.....	79
Figura 42 - Teste confirmação do orçamento.....	80
Figura 43 - Teste orçamento para pedido.....	80

RESUMO

A presente monografia tem como tema o desenvolvimento de um protótipo de aplicação *mobile android* para automação do processo de elaboração de orçamento de venda em uma distribuidora de cosméticos da cidade de Teófilo Otoni; está concentrada na área de desenvolvimento *mobile* e tem como principal objetivo o desenvolvimento de um protótipo de aplicativo móvel capaz de propor uma otimização do processo de elaboração de orçamentos de vendas de uma distribuidora de cosméticos da cidade de Teófilo Otoni. O estudo teve como principais metodologias uma pesquisa descritiva e laboratorial, em razão de entender um determinado caso na empresa no que tange à elaboração de orçamentos, com intuito de estudar e registrar passo a passo para uma melhor compreensão de como seria desenvolvido o protótipo da aplicação. Dando prosseguimento, foram efetuados os testes necessários, apresentando uma forma otimizada de elaboração de orçamento de venda. A conclusão deste trabalho resulta no desenvolvimento do protótipo de um aplicativo móvel que propõe melhorias no processo de elaboração de orçamentos de vendas da distribuidora OLV Cosméticos, e que se torna uma ferramenta bastante útil levando em conta os problemas enfrentados pela empresa e a real importância em que um protótipo é desenvolvido, com propostas de melhorias para processos atuais.

Palavras-Chave: Protótipo; Otimização; Aplicação *mobile*; Orçamento.

SUMÁRIO

INTRODUÇÃO	8
1 EMPRESA OLV COSMETICOS	10
2 REFERENCIAL TEÓRICO	11
2.1 Mobile	11
2.1.1 Aplicativos Nativos	11
2.1.2 Aplicativos Híbridos	12
2.1.3 Android	12
2.2 Linguagem de Programação	13
2.2.1 Programação orientada a Objetos.....	14
2.2.2 Java.....	14
2.3 Plataforma Android	15
2.3.1 Ferramenta de Desenvolvimento Android Studio	15
2.3.2 Activity	15
2.3.3 Intents.....	17
2.3.4 Avd	17
2.4 Web	18
2.4.1. Arquitetura cliente / servidor.....	18
2.4.2 Php.....	19
2.4.3 Servidor Web.....	19
2.4.4 Css	20
2.4.5 Html.....	21
2.4.6 Framework	21
2.4.6.1 <i>Laravel</i>	22
2.5 Visual Studio Code	22
2.6 Engenharia de Software	23

2.6.1 Software	24
2.6.2 Processos de software	25
2.6.3 Modelo de processo de software.....	25
2.6.4 Especificação de requisitos	26
2.6.5 Requisitos.....	27
2.6.6 Engenharia de requisitos.....	27
2.6.7 Prototipagem	28
2.7 Banco de Dados	29
2.7.1 Sistema Gerenciador de Banco de Dados (SGBD).....	30
2.7.2 Linguagem de SGBD	30
2.7.3 SQL	30
2.7.4 NoSql.....	31
2.7.5 Firebase	31
2.7.6 Baas	32
3 DESENVOLVIMENTO DO PROTÓTIPO	33
3.1 Preparação do Ambiente	33
3.2 Ferramentas e softwares utilizados.....	33
3.2.1 FluidUI	34
3.2.2 Firebase	36
3.2.3 Android Studio.....	37
3.2.4 Visual Studio Code	38
3.2.5 Composer	39
3.2.6 XAMPP	40
3.2.7 NPM e Node.js	41
3.2.8 Laravel.....	42
3.2.9 AdminLTE.....	43
3.2.10 Git.....	45
3.3 Metodologia	46
3.4 Construção da base do protótipo	47
3.5 Construção da aplicação	49
3.6 Banco de dados.....	54
3.7 Finalização do Protótipo.....	65
3.8 Painel Administrativo.....	67
4 TESTES E ANALISE DE TESTES	72

CONCLUSÃO	Erro! Indicador não definido.
REFERÊNCIAS.....	86

INTRODUÇÃO

A presente monografia está concentrada na área de desenvolvimento *mobile* e tem como tema o desenvolvimento de um protótipo de uma aplicação *mobile Android* para automação do processo de elaboração de orçamento de venda em uma distribuidora de cosméticos da cidade de Teófilo Otoni. A busca por tecnologias capazes de agilizarem processos empresariais aumenta a cada dia, já que um sistema auxilia em processos internos evitando retrabalho, integrando todas as áreas da empresa, reduzindo os custos, aumentando o lucro, apoiando a tomada de decisão e outros fatores que são indispensáveis para qualquer empresa que almeja o seu próprio crescimento. A monografia tem como pergunta problema: De que forma o desenvolvimento de um protótipo de uma aplicação *mobile* para a criação de orçamentos poderia minimizar os problemas no processo de elaboração de orçamentos de vendas em uma distribuidora de cosméticos da cidade de Teófilo Otoni-MG?

O ato de captação e preenchimento de pedidos da empresa OLV Cosméticos, ainda se faz pelo uso de planilhas eletrônicas para posteriormente ser feito o encaminhamento para o e-mail da empresa. Alguns consultores não praticam o mesmo processo, realizando a transmissão dos produtos solicitados pelo cliente por telefone, em que um funcionário tem a função de recebê-los e criar uma planilha lançando os produtos informados para posteriormente fazer a inserção dos mesmos no sistema, gerando assim orçamentos que após serem analisados pelo setor financeiro, futuramente viriam a ser vendas efetuadas. Com isso observa-se retrabalho, gastos e baixa na produtividade em ambas as partes.

Desse modo a probabilidade de erros em criação de orçamentos aumenta consideravelmente e o tempo gasto para realizar as mesmas funções é desnecessário, visto que a tecnologia está avançada e traz cada vez mais recursos

para a humanidade. A presente monografia aborda o desenvolvimento de um protótipo de aplicativo móvel capaz de propor uma otimização do processo de elaboração de orçamentos de vendas acima citado.

Quanto aos fins a pesquisa se caracteriza como descritiva, pois visa estudar um fato específico na empresa no que tange a elaboração de orçamentos, com intuito de analisar e registrar passo a passo, identificando possíveis falhas e causas dos problemas enfrentados que acarretam atrasos, desentendimentos e diversos fatores que influenciam na produtividade dos colaboradores.

Quanto aos meios a pesquisa se caracteriza primeiramente como bibliográfica, pois baseou-se num levantamento de livros, monografias, dissertações e artigos para o estudo e entendimento da linguagem de programação que foi utilizada para o desenvolvimento, e laboratorial, pois produziu determinado fato, observando e registrando os teste feitos para um melhor aprimoramento do protótipo com intuito de otimizar o processo orçamentário da empresa. Os resultados foram tratados de forma qualitativa.

A monografia está composta de 4 capítulos: o primeiro descreve o problema enfrentado pela empresa; o segundo discute o referencial teórico que abrange subcapítulos, onde está o embasamento para a elaboração da monografia e diversas tecnologias e ferramentas utilizadas como aplicações *mobile*, Linguagens de programação, Plataforma *Android*, *Web*, Engenharia de *Software* e banco de dados; o terceiro apresenta o desenvolvimento do protótipo; o quarto apresenta os testes e análises realizados com o protótipo.

1 EMPRESA OLV COSMETICOS

A empresa OLV Cosméticos está no mercado há muito tempo no ramo de distribuição de cosméticos, atendendo diversas lojas e salões de cabeleireiro em várias regiões do estado de Minas Gerais. Ao longo dos anos a empresa teve um aumento considerável no seu faturamento, o que acarretou adaptações necessárias em seus processos para melhor atender os seus clientes. Observando essa perspectiva viu-se a necessidade de implantação de um sistema que atendesse a demanda crescente de vendas e o grande fluxo de informações que a empresa estava manipulando.

Atualmente a empresa possui um sistema para atender necessidades gerenciais e os processos internos como: controle de estoque, faturamento, cadastros de clientes, análises precisas para tomada de apoio a decisão e entre outras funcionalidades. Observando os processos internos da empresa após a implantação do sistema, foi possível constatar uma total dependência dos consultores de vendas, que muitas das vezes ocasionam desentendimentos e divergências criando um desgaste desnecessário no ambiente de trabalho no que se refere a elaboração de orçamento de vendas, que ainda é um processo que não está totalmente informatizado se comparado a outros processos da empresa.

2 REFERENCIAL TEÓRICO

2.1 Mobile

2.1.1 Aplicativos Nativos

Os aplicativos nativos desenvolvidos especificamente para uma plataforma *Android*, *iOS* ou *Windows Phone*, fazendo com que sejam utilizados de forma mais eficiente os recursos disponíveis no smartphone. O aplicativo deve estar armazenado no banco de dados do próprio aparelho, possibilitando o seu funcionamento mesmo off-line. Com o desenvolvimento nativo, a aplicação poderá explorar recursos mais avançados do aparelho como por exemplo tela sensível ao multitoque e efeitos visuais (TOLEDO; DEUS, 2012).

Ainda na mesma página os autores complementam dizendo que na maioria das vezes o poder de processamento dos smartphones são bem aproveitados, fazendo com que os aplicativos tenham um tempo de resposta melhor, o que possibilita mais agilidade no seu uso. Outro ponto interessante é a possibilidade de integração com outras aplicações existentes no *smartphone* e acesso aos dados presentes no mesmo como por exemplo: câmera e agenda telefônica. Desenvolver aplicativos específicos requer linguagens específicas e em muitos casos pode exigir um alto investimento para o treinamento para as plataformas escolhidas e sua possível duplicação para outras plataformas. Outro problema está relacionado à atualizações de versões, tornando-se necessária a interação do usuário manualmente para receber a mesma aplicação com novos recursos (TOLEDO; DEUS, 2012, p.15).

2.1.2 Aplicativos Híbridos

Lopes (2016, p.2-3) afirma que: “Chamamos de aplicações híbridas porque usam as linguagens da Web para construir aplicativos.” A aplicação híbrida é desenvolvida com intuito de ser multiplataforma e evitar que se desenvolva um aplicativo para cada plataforma existente. Aplicações híbridas podem utilizar os recursos nativos disponíveis da plataforma no qual foi instalado, o seu desenvolvimento é em HTML, CSS e JavaScrip.

Segundo Lopes:

Um aplicativo híbrido bem construído se integra à plataforma da mesma forma que um nativo. Há diferenças de performance apenas em casos muito específicos que exijam realmente bastante processamento no dispositivo (LOPES, 2016, p.6).

Toledo e Deus explicam que desenvolvimento de aplicações híbridas, devido à sua alta popularidade, permite uma proliferação maior, ou seja, a não necessidade de desenvolver para cada plataforma faz com que esses aplicativos tenham uma aceitação maior (TOLEDO; DEUS, 2012, p.15). Analisando o raciocínio dos autores tratados nos tópicos sobre aplicações nativas e híbridas, pode-se perceber que cabe ao desenvolvedor escolher o que melhor irá atender ao propósito do aplicativo, levando em conta os fatores acima tratados.

2.1.3 Android

“O *Android* é o sistema operacional móvel do Google e atualmente é líder mundial nesse segmento” (LECHETA, 2015, p.25). O sucesso do sistema operacional não se deu apenas pela força da Google; existem várias empresas que são gigantes no mercado de mobilidade envolvidas no projeto. O grupo que ajuda no

desenvolvimento se chama OHA (Open Handset Alliance) que é formado por grandes empresas como Intel, LG, HTC, Samsung, Motorola, Sprint Nextel, Sony Ericsson, Acer, Dell, ASUS, Garmin, entre outras (LECHETA, 2015, p.25).

Cada vez mais o mercado competitivo vem utilizando o *mobile*, empresas visam incorporar aplicações moveis em seus sistemas para agilizarem os seus negócios, smartphones e tablets ocupam um importante espaço quando o assunto tratado é a mobilidade. O desenvolvimento *mobile* é um grande pilar na área da tecnologia e, de acordo com pesquisas, é uma das áreas que terá um grande crescimento em pouco tempo. (LECHETA, 2015, p.26).

Segundo Lecheta:

O fato de o *Android* ser de código aberto contribui muito para seu aperfeiçoamento, uma vez que desenvolvedores de todos os lugares do mundo podem contribuir para seu código-fonte, adicionando novas funcionalidades ou simplesmente corrigindo falhas (LECHETA, 2015, p.27).

2.2 Linguagem de Programação

Programa é um conjunto de instruções que dizem ao computador como realizar certas funções. A língua compreendida pelos computadores se nomeia como linguagem de máquina, que é uma serie de *bits* pelos quais o computador compreende o que está sendo solicitado (DOBAY, 2012, p.4).

Linguagem de programação é uma linguagem que faz intermédio entre a linguagem compreendida por humanos e a linguagem de máquinas. Geralmente uma linguagem de programação é composta de:

- Palavras chaves que normalmente são em inglês (*if, while, function*) que vem acompanhadas de alguns comandos da linguagem;
- Conjuntos de símbolos (“;”, #, {, etc.);
- Regras da própria linguagem que dizem como devem ser passadas as instruções e a sua organização dizendo quais as possíveis combinações e utilizações. (DOBAY, 2012, p.4-5).

Os computadores compreendem apenas linguagem de máquina, então para entenderem linguagem de programação utilizam-se compiladores ou interpretadores

que por sua vez fazem o processo de tradução do código para a linguagem de máquina (DOBAY, 2012, p.4-5).

2.2.1 Programação orientada a Objetos

A programação orientada a objetos é um modelo de programação que faz o uso de classes e objetos para representação e processamento utilizando programas de computadores (SANTOS, 2013, p.5).

A orientação a objeto tem como princípio a representação do mundo real e a forma de se interagir com os objetos. A programação orientada a objetos tende a ser mais rápida, visto que o desenvolvimento ocorre em módulos e em blocos de código, trazendo ao programador uma maior qualidade e agilidade no desenvolvimento, pois permite a reutilização de objetos já desenvolvidos e utilizados anteriormente que são facilmente integrados à aplicação. (CLARO; SOBRAL, 2008, p.8)

2.2.2 Java

É uma linguagem orientada a objetos e foi desenvolvida pela *Sun Microsystems*, com a capacidade de gerar grandes aplicações comerciais, *softwares* robustos, completos e independentes para web. Caracteriza-se pela semelhança com linguagem C++ exceto pelas características consideradas complexas (CLARO; SOBRAL, 2008, p.12-13).

Programas escritos na linguagem Java se classificam como *Applets*, Aplicativos ou *Servlets*. Os aplicativos necessitam de um interpretador instalado para funcionarem na máquina. Os *Applets* são programas que fazem o seu carregamento juntamente com páginas HTML e neste caso os *browsers* fazem o papel de interpretador. No caso dos *Servlets* os programas são desenvolvidos em Java e são interpretados por um servidor Web (CLARO; SOBRAL, 2008, p.12-13).

2.3 Plataforma Android

Com o avanço da tecnologia, existem no mercado diversos tipos de smartphones e cada um deles contém uma versão diferente do sistema operacional. O primeiro smartphone com *Android* em sua primeira versão 1.1 foi o HTC G1. “No *Android*, uma versão do sistema operacional é conhecida com plataforma. Podemos dizer então que existem diversas plataforma diferentes do *Android* (1.5, 1.6, 2.x, 3.x, 4.x, 5.x etc.)” (LECHETA, 2015, p.43).

Cordeiro (2017, p.11) afirma que: “Desde 2007, o *Android* vem sofrendo atualizações e a cada versão que é lançada a plataforma fica mais poderosa em termos de processamento e integração com outros aparelhos eletrônicos.

2.3.1 Ferramenta de Desenvolvimento Android Studio

O *Android Studio* é a IDE oficial de desenvolvimento para *Android* e foi anunciado no Google I/O 2013 e se baseia no *IntelliJ IDEA* da *JetBrains*. “O *Android Studio* apresenta alguns diferenciais importantes se comparado ao Eclipse, que antigamente era a ferramenta oficial” (LECHETA, 2015, p.45).

Conforme ensina Lecheta:

Uma grande diferença entre o Eclipse e o *Android Studio* é o processo de compilação dos projetos. No Eclipse cada projeto é compilado do jeito clássico, como qualquer projeto Java dentro do Eclipse. Mas no *Android Studio* a compilação é feita pelo *Gradle*, que é um moderno sistema de builds (LECHETA, 2015, p.45).

2.3.2 Activity

Activity é uma classe que representa a tela da aplicação e é responsável por todos os eventos ocorrentes nesta tela. Para criar uma nova tela na aplicação, necessita-se que uma nova *activity* seja criada e declarada no arquivo

AndroidManifest.xml (LECHETA, p.96).

Lecheta (2015, p.97) afirma que:

A tradução da palavra *activity* para o português é atividade. Então, também podemos dizer que uma *activity* representa uma atividade, ação ou funcionalidade que o usuário pode realizar dentro de sua aplicação.

O ciclo de vida de um *activity* é caracterizado por uma pilha, cada tela que está executando no momento ou seja, uma *activity* estará no topo da pilha, cada nova *activity* iniciado irá para o topo da pilha e as outras *activities* podem estar pausadas ou executando em segundo plano. Caso o sistema necessite de mais poder de processamento e espaço de memória para realizar determinada função exigida, o mesmo poderá encerrar os processos pausados (LECHETA, 2015, p.99-100).

Existem vários métodos da classe *Activity* que se utilizam para controlar os estados da aplicação que são:

- *onCreate(bundle)* : Esse método é obrigatório e é chamado uma única vez para fazer a inicialização do projeto. Nele deve-se chamar o método *setContentView(view)* para que seja configurado o layout da tela.
- *onStart()*: É utilizado quando a *activity* está ficando visível para o usuário. Pode se chamar depois dos métodos *onCreate()* e *onRestart()* dependendo do estado em que a aplicação se encontre
- *onRestart()*: É chamado quando a *activity* foi parada e está sendo iniciada novamente.
- *onResume*: Esse método é utilizado quando a *activity* está no topo da pilha e está sendo executada pelo usuário, geralmente é utilizada para atualização da tela onde são disparadas threads para a verificação junto ao *web service* ou banco de dados atualizando as informações da *activity*.
- *onPause()*: Esse método é chamado quando a *activity* for fechada, seja pelo usuário ao pressionar a tecla Home ou pelo recebimento de uma ligação. O método é acionado e salva o estado da aplicação para quando o usuário retornar, tudo ser chamado pelo método *onResume()*.
- *onStop()*: É chamado após o método *onPause()* mostrando que a *activity* está sendo finalizada, podendo ser chamada novamente pelo método *onRestart()*. Caso

a activity fique parada muito tempo e o sistema operacional necessite de memória, ele irá utilizar o `onDestroy()` para remover a activity da pilha e matar o seu processo.

- `onDestroy`: É chamado para o encerramento da activity, pode ser chamado automaticamente pelo sistema para a liberação de memória ou na própria aplicação para a sua finalização(LECHETA, 2015, p.101-102).

2.3.3 Intents

Lecheta (2015, p.608) diz que uma intent é uma mensagem enviada da aplicação para o sistema operacional com a intenção de a aplicação realizar alguma tarefa como por exemplo:

- Abrir uma nova tela na aplicação;
- Ligar para um número de celular;
- Abrir a câmera e solicitar uma gravação; etc.

Existem dois tipos de intents, as explícitas e implícitas. Intentes explícitas são aquelas que determinam qual activity deverá executar, são direcionadas; já as intents implícitas são genéricas e são enviadas para o sistema operacional e podem ser atendidas por qualquer aplicação que esteja instalada no sistema operacional (LECHETA, 2015, p.606-608).

2.3.4 Avd

O AVD é o emulador virtual do *Android Studio* onde se faz a simulação de um dispositivo móvel, seja smartphone ou tablet, com as configurações desejadas de

resolução e outros ajustes utilizando a mesma plataforma do sistema operacional.

O emulador tem papel fundamental nos testes da aplicação em desenvolvimento, pois auxilia o programador dando diversas opções de configuração e testes sobre a aplicação em si (LECHETA, 2015, p.57-60)

2.4 Web

Segundo Nonato (2015, p.4),” A World Wide Web, também conhecida como Web ou WWW, é um sistema de documentos em hipermídia que são interligados e executados na Internet.”. O intuito principal de sua criação foi fazer com que o compartilhamento de documentos se tornasse mais fácil. Em 21 de março de 1989, Tim Berners-Lee apresentou uma proposta para o gerenciamento dessas informações, onde se descrevia um sistema que hoje seria Web. No final de 1990 Berners-Lee conseguiu publicar a primeira página Web com todas as configurações e algumas páginas que na verdade eram detalhes do seu projeto. (NONATO,2015, p.6-7).

Continuando com o raciocínio do autor Nonato (2015, p.12) A “Web é todo o conteúdo que o usuário final pode acessar na rede. Sendo que, a web já passou por transformações evolutivas - evolução do código e dentre elas existem a: WEB 1.0, WEB 2.0 E WEB 3.0.”

2.4.1. Arquitetura cliente / servidor

O termo cliente/servidor se caracteriza como o cliente sendo um “programa” com tempo de vida finito que, por sua vez, faz solicitações a um servidor que no caso seria um “programa” com tempo de vida indeterminado que faz a comunicação

respondendo as solicitações (FOROUZAN; MOSARRAF, 2013, p.803).

Os autores Forouzan e Mosharraf (2013, p.804) acrescentam que a arquitetura pode ser projetada como servidor interativo ou servidor concorrente. O servidor interativo recebe diversas requisições ao mesmo tempo, mas o seu tratamento é individual ou seja, as solicitações são atendidas uma a uma, enquanto o servidor está ocupado respondendo a um cliente os outros estão esperando o atendimento. Esse tipo de projeção pode se assemelhar a um banco com um único caixa. O servidor concorrente é exatamente o contrário do interativo, pois servidores concorrentes tem a capacidade de atender diversas solicitações ao mesmo tempo, assemelhando-se a um banco com diversos caixas onde o cliente chega e já é automaticamente direcionado para um caixa para ser atendido. O número de atendimentos simultâneos depende exclusivamente dos recursos computacionais do servidor

2.4.2 Php

O PHP foi criado por Rasmus Lerdorf em 1994 especificamente para Web. O PHP pode ser inserido em páginas HTML e é uma linguagem utilizada para criar scripts que serão interpretados pelo servidor e que irá gerar uma saída para o usuário. A linguagem passou por diversas transformações e aperfeiçoamentos ao longo do tempo e significa PHP Hypertext Preprocessor. (WELLING; THOMSON, 2005, p.1).

O PHP se caracteriza como uma ferramenta de pré-processamento de páginas que sejam HTML, podendo alterar o conteúdo da página até mesmo antes de fazer o envio para o navegador; permite a entrada de dados inseridos por um usuário como por exemplo preenchimento de formulários e outras interações (BENTO, 2013, p.3).

2.4.3 Servidor Web

Segundo explica Marcelo (2005, p.3) O servidor Web deu seus primeiros

passos em 1995 quando foi criado pela NCSA (National Center for Computer Applications). O projeto não teve muitas melhorias e acabou ficando estagnado até o momento em que uma parte da NCSA, não contente com o acontecido, desligou-se da empresa e se juntou a um grupo de desenvolvedores que, por sua vez, fizeram diversas alterações no projeto e inovações em cima do código fonte do web server da NCSA. Devido ao grande número de patch adicionados, os desenvolvedores fizeram um trocadilho ao nomear de Apache que finalmente nasceu em abril de 1995 com a versão 0.62.

Marcelo (2005, p.4) continua dizendo que posteriormente ao lançamento, surgiu a Fundação Apache que é reconhecida por manter o servidor web mais popular e líder absoluto na web, contendo os sites mais populares sob o seu domínio, criando uma comunidade de usuários pelo mundo todo.

Continuando o raciocínio do autor Marcelo (2005, p.6) um Servidor Web tem características como:

- Hospedar páginas na web
- Hospedagem e bancos de dados;
- Suporte a páginas em PHP, Java OU Perl;
- Suporte a transações seguras (Secure Sockets Layer);
- Autenticação dos usuários.

2.4.4 Css

O CSS (Cascading Style Sheets) ou folha de estilo é uma tecnologia fundamental para a criação de um conteúdo profissional para internet, onde o CSS irá formatar o conteúdo da página, alterando o fundo da página, os estilos dos textos, imagens ou qualquer tipo de mídia. Alguns editores profissionais como o Dreamweaver da Adobe auxiliam e facilitam muito a criação de CSS nas páginas HTML (QUIERELLI, 2012, p.8-9).

Quierelli (2012, p.10) diz que o CSS além de formatar os estilos de texto e imagens que são exibidas para o usuário, ele tem a capacidade de criar caixas que

podem auxiliar o desenvolvedor no layout da página para uma melhor visualização e disposição dos elementos.

Acompanhando o raciocínio do autor, percebe-se que o CSS veio para modificar e melhorar as páginas antes feitas somente em HTML, dando mais vida e proporcionando aos leitores das páginas uma visualização agradável do conteúdo ali exposto.

2.4.5 Html

Segundo Neves (2004, p.13), o HTML ou *Hyper Text Markup Language* nada mais é que uma linguagem de marcação utilizada para a formatação de textos, possibilitando a criação de ficheiros de textos através de marcas (tags), onde o navegador irá fazer a interpretação e mostrar o resultado ao usuário.

A estrutura utilizada para a criação desse tipo de documento consiste na utilização de marcas(tags) exterior `<html>` que abrangem todo o documento do início ao fim. Dentro dessa tag são inseridos o cabeçalho `<head>` e o corpo do documento denominado `<body>` (NEVES, 2004, p.15). As tags são designações interpretadas pelo navegador que irá exibir na tela a forma como foram organizadas na página. A designação de uma tag se dá pela sua abertura `<head>` e o seu fechamento `</head>`, “...já numa forma mais complexa poderá incluir atributos (ex: `<body bgcolor="LightCyan">`)” que possibilita ao desenvolvedor inserir informações complementares sobre o atributo (NEVES, 2004, p.17-18).

2.4.6 Framework

Segundo Gabardo (2017, p.14) Framework é “Um conjunto de classes ou funções implementadas em uma linguagem de programação específica usadas para auxiliar o desenvolvimento de um software”. Os frameworks são utilizados para diversos fins, mas tem o foco principal em redução de reescrita de códigos adotando a reutilização de métodos e funções já utilizadas no decorrer do desenvolvimento. Além de oferecer diversos recursos como autenticações, elaboração de formulários e autenticações, um framework tem a capacidade de reduzir o tempo gasto em um desenvolvimento pois conta com recursos prontos ou semiprontos, o que facilita a vida do desenvolvedor.

2.4.6.1 Laravel

Gabardo (2017, p.14-15) explica que as aplicações Web com o tempo foram deixando de ser apenas simples páginas em HTML e acabaram se transformando em grandes softwares complexos com diversas funcionalidades e conectadas a APIs com recursos para atender um grande número de dispositivos. “Laravel é um framework PHP MVC sob o paradigma de orientação a objetos”.

Com essa evolução, a demanda por softwares de mais qualidade cresceu consideravelmente, o que fez a concorrência das empresas desenvolvedoras seguir o mesmo caminho. Com isso as empresas buscam as melhores ferramentas para auxiliarem em seus processos. Com Laravel a entrega de projetos é agilizada pois o tempo de escrita do código é minimizado, além de ter a capacidade de uma equipe trabalhar em forma conjunta baseando-se na organização do código, testar soluções em tempo hábil e dar manutenções. (GABARDO, 2017, p.16).

2.5 Visual Studio Code

Segundo o site oficial do Visual Studio¹, o Visual Studio Code é um editor de código-fonte leve e muito poderoso que pode ser executado na área de trabalho, está disponível para Linux, macOS e Windows. Apesar de leve é um software com muitas funcionalidades e dá suporte a diversas linguagens como por exemplo: JavaScript, Node.js e TypeScript além de possuir diversas extensões para outras linguagens como C#, C++, Python, PHP, Java e Go.

2.6 Engenharia de Software

A Engenharia de Software consiste em procedimentos relacionados a produção de software desde a fase inicial até a final. Pressman (2006, p.39) ensina que “Engenharia de software é o estabelecimento e o emprego de sólidos princípios de engenharia de modo a obter software de maneira econômica, que seja confiável e funcione de forma eficiente em máquinas reais.”. Ou seja a engenharia de software segue os princípios da engenharia, auxiliando no processo de desenvolvimento fazendo com que se obtenha um produto final de qualidade.

Pressman (2006, p.39-40) afirma ainda que a Engenharia de software é um processo de produção de software que em todas as suas formas e campos de aplicação deve passar pelos 4 pilares de processos de engenharia, com compromisso organizacional e qualidade baseando-se em:

- Foco na Qualidade: A engenharia de software se sustenta pelo foco na qualidade, e visa o aperfeiçoamento levando ao desenvolvimento de abordagens cada vez mais efetivas;
- Processos: O processo define uma metodologia que será adotada para o prazo de entrega estabelecido e é a base que mantém coesas as camadas de tecnologia e define a base para o controle do gerenciamento do projeto;
- Métodos: Fornecem as informações técnicas para o desenvolvimento do software e se baseia em um conjunto de princípios que governam cada área da

¹ Disponível em: < <https://visualstudio.microsoft.com/pt-br/> >

tecnologia utilizando atividades de modelagem e técnicas descritivas;

- Ferramentas: Fornecem suporte para os processos e métodos de maneira automatizada ou semiautomatizada.

As informações podem ser analisadas na imagem a seguir:

Figura 01: Camadas da Engenharia de Software



Fonte: PRESSMAN, 2006, p.27

2.6.1 Software

É a parte lógica de um computador capaz de utilizar os recursos de hardware e possuindo características diferentes dos mesmos, pois o software não se desgasta. A maioria dos Softwares são feitos sob medida para empresas. O seu processo de produção é a partir de desenvolvimento e planejamento pois não são manufaturados no sentido clássico. Um grande diferencial para empresas é a aquisição de um bom software com interface amigável capaz de gerir um negócio ou capacitar um sistema (PEREIRA JR, p.4).

Pressman (2006. p.31) afirma que:

O software distribui o produto mais importante de nossa era — a informação. Ele transforma dados pessoais (por exemplo, transações financeiras de um indivíduo) de modo que possam ser mais úteis num determinado contexto; gerencia informações comerciais para aumentar a competitividade; fornece um portal para redes mundiais de informação (Internet) e os meios para obter informações sob todas as suas formas.

2.6.2 Processos de software

Processo é definido como conjunto de atividades, ações e tarefas realizadas para a criação de um software com o intuito de criar um resultado de alta qualidade e dentro do prazo estabelecido (PRESSMAN, 2011, p.52-53).

Essas atividades em sua maioria são executadas por engenheiros de software que utilizam quatro atividades fundamentais a todos os processos de software que são:

- Especificação de software: Onde devem ser definidas as suas funcionalidades e restrições;
- Desenvolvimento do software: Desenvolvimento do software atendendo as suas especificações;
- Validação do software: Que garante que o software faz o que o cliente deseja;
- Evolução do software: A constante mudança e evolução do software para atender as necessidades mutáveis do cliente (SOMMERVILLE, 2004, p.5).

2.6.3 Modelo de processo de software

Segundo Sommerville (2004, p.8) “Um modelo de processo de software é uma descrição simplificada de um processo de software, que é apresentada a partir de uma perspectiva específica.”

Pressman (2011, p.59-61) ensina que o modelo em cascata prega uma abordagem sequencial para o desenvolvimento, denominada também de abordagem clássica, que se inicia no levantamento de necessidades do cliente, avançando para as fases de planejamento, modelagem, construção, emprego e no suporte do software já pronto.

O modelo incremental segue sequências lineares onde são liberadas funções do software básicas para posteriormente serem incrementados outros recursos complementares. Após a entrega do primeiro incremento, é desenvolvido um planejamento baseado em futuras modificações no software essencial para melhor

atender as necessidades apresentadas pelo cliente (PRESSMAN, 2011, p. 61-62). “Modelos evolucionários são interativos. Apresentam características que possibilitam desenvolver versões cada vez mais completas do software.” (PRESSMAN, 2011, p. 62). Para o autor, existem dois modelos comuns em processos evolucionários. A prototipação que pode ser utilizado, como técnica possível para qualquer um dos modelos de processo de software, quando o cliente define uma serie de objetivos que o software deverá abranger, mas não identifica quais os requisitos para as funções e recursos. Se inicia com a comunicação, onde o cliente define os objetivos gerais do software, identifica quais requisitos já são conhecidos e especifica quais as áreas mais necessitadas, levando ao desenvolvedor uma interação de prototipação em que o mesmo desenvolve um projeto rápido de partes do sistema como, por exemplo, layout ou formas de exibição, apresentando ao cliente e aguardando o seu feedback para aprimorar o próximo protótipo.

Pressman (2011, p.65) afirma que:

Usando-se o modelo espiral, o software será desenvolvido em uma série de versões evolucionárias. Nas primeiras iterações, a versão pode consistir em um modelo ou em um protótipo. Já nas iterações posteriores, são produzidas versões cada vez mais completas do sistema que passa pelo processo de engenharia.

O modelo espiral se caracteriza por voltas em um conjunto de atividades metodológicas que a engenharia de software prega. No modelo espiral o software evolui a cada volta à medida que o processo avança. Este modelo utiliza-se de prototipação para redução de riscos. Ou seja, o modelo é uma abordagem interativa, onde o cliente faz com que o software evolua a cada volta no espiral até a sua finalização e entrega, e podendo ser adaptado para ser utilizado ao longo da vida do software (PRESSMAN, 2011, p. 66).

2.6.4 Especificação de requisitos

Especificação de requisitos pode se caracterizar como um documento composto de modelos gráficos, matemáticos e descrições em linguagem natural detalhadas de todos os aspectos do software que deveram ser preparadas antes da

criação do mesmo, para auxiliar no processo de desenvolvimento do software (PRESMAN,2011, p.129).

Paula Filho (2000, p.14) afirma que:

Mesmo requisitos documentados podem apresentar problemas. Uma especificação de requisitos podem conter requisitos incompletos, inconsistentes ou ambíguos. Alguns destes problemas decorrem da natureza da própria linguagem natural, que normalmente é usada para expressá-los. Outros decorrem de técnicas deficientes de elaboração dos requisitos.

2.6.5 Requisitos

Os requisitos são as características presentes no software e se classificam em funcionais e não funcionais. Os funcionais demonstram como o software deverá reagir a determinadas ações dos usuários; os não funcionais quantificam as características do comportamento do software (PAULA FILHO,2000, p.13). Paula Filho (2000, p.14) mostra que existem tipos de requisitos que podem ser:

- Requisitos explícitos que estão presentes no documento de especificação de requisitos.
- Requisitos normativos que decorrem de leis, regulamentos ou padrões que determinados produtos devem obedecer.
- Requisitos implícitos que não estão documentados mas são requisitos esperados pelo cliente e que são cobrados por ele.

2.6.6 Engenharia de requisitos

Segundo Paula Filho (2000, p.14) “O conjunto das técnicas de levantamento, documentação e análise forma a engenharia dos requisitos, que é uma das disciplinas da engenharia de software.”

A engenharia de requisitos é um processo que mescla as tarefas necessárias para a elaboração e para manter o documento de requisitos de sistema (SOMMERVILLE,2004, p.103).

Pressman (2011, p.127) ensina que a engenharia de requisitos fornece um mecanismo para entender o que o cliente realmente deseja, analisando a viabilidade e entrando em consenso para a especificação da solução sem ambiguidade e validando as solicitações do cliente de acordo com o desenvolvimento. Ela abrange sete tarefas como: concepção, levantamento, elaboração, negociação, especificação, validação e, por fim, a gestão. Elas podem acontecer paralelamente e são adequadas de acordo com as necessidades do projeto.

2.6.7 Prototipagem

Sommerville (2004, p.145) explica que o protótipo se caracteriza pela versão inicial de um projeto de software, que é utilizado para testar funcionalidades, descobrir a fundo o que o software realmente irá prover e para conhecer mais sobre os problemas e suas possíveis soluções. A prototipação é essencial pois ajuda ao usuário ter uma visão de como poderá vir a ser o software desenvolvido. O protótipo permite o controle de custos e realizações de diversos testes de funcionalidades. “Um protótipo de software apoia duas atividades do processo de engenharia de requisitos:” (SOMMERVILLE, 2004, p.145).

- Levantamento de requisitos: Os protótipos auxiliam os usuários a verificar se o sistema atende realmente a empresa. A partir de experiência pode-se ter novas ideias sobre os requisitos do projeto, identificar pontos negativos e positivos e até mesmo acréscimo de novas funcionalidades;
- Validação de requisitos: Os protótipos tem a capacidade de identificar erros e omissões. O usuário pode ter definido muito bem uma função do sistema, mas após a integração com outras funcionalidades o usuário pode perceber que a sua visão do software estava incorreta, podendo alterar as especificações do mesmo.

O autor continua o raciocínio dizendo: “A prototipação reduz o número de problemas com a especificação de requisitos. Além disso, os custos totais de desenvolvimento poderão ser mais baixos se um protótipo for desenvolvido” (SOMMERVILLE, 2004, p.145)

2.7 Banco de Dados

Segundo Cayres (2015, p.1-2), “Banco de dados: coleção de dados relacionados que podem ser inseridos, atualizados, e recuperados e que possuem um significado implícito.” Os dados podem ser vistos como fatos, conceitos ou instruções, já a informação pode ser vista como esse conjunto de dados devidamente organizados e ordenados. Banco de dados foram projetados com o objetivo de armazenar essas informações, possibilitando o acesso e a manipulação posteriormente de maneira organizada e estruturada.

Cayres (2015, p.2) ensina que: “Banco de dados podem ser caracterizados como: agenda telefônica, catálogo de livros, planilhas orçamentárias etc.” E que os principais desafios de um sistema de banco de dados eram:

- **Gerenciamento de grande quantidade de informação:** Um sistema de banco de dados teria de possibilitar tanto o armazenamento simples de uma lista telefônica quanto um cadastro de passagens de uma empresa aérea sendo capaz de prover a confiabilidade e a segurança em ambos os casos independentemente do volume dos dados;
- **Evitar redundância e inconsistência de dados:** Reduzir ou eliminar redundâncias de dados e diversos lugares onde poderia acontecer a inconsistência de dados, que seria a atualização de um arquivo somente em um determinado lugar;
- **Facilidade de acesso:** Seria a facilidade do acesso aos dados, onde diversos usuários poderiam ter acesso a mesma informação compartilhada;
- **Segurança de dados:** Garantir a segurança de dados por meio de acesso por parte de usuários com login e senha;
- **Garantia de Integridade:** Garantir com que os valores dos dados armazenados satisfaçam as restrições para manutenção de consistência e coerência;

- **Facilidade de Migração:** O sistema deve garantir a possível transferência de dados de um banco para outro, esse ato deu-se o nome de Migração.

2.7.1 Sistema Gerenciador de Banco de Dados (SGBD)

Um SGBD é uma coleção de programas que permite ao usuário manter um banco de dados. É um sistema de software que facilita os processos de definição, construção, manipulação e compartilhamento de bancos de dados entre várias aplicações e usuários. (ELMASRI; NAVATHE, 2005, p.3). Segundo Cayres (2015, p.3) O SGBD é: “é um software que incorpora e facilita as funções de definição, recuperação e alteração de dados em um Banco de Dados.”

2.7.2 Linguagem de SGBD

A Linguagem de Definição de Dados (DDL) é utilizada pelo administrador do banco de dados e pelos projetistas para definir os esquemas conceituais e internos. A Linguagem de Definição de Armazenamento (SDL) é utilizada para o esquema interno ou o para o mapeamento entre os dois esquemas. Para uma arquitetura de três esquemas, existe a Linguagem de Definição de Visões(VDL) para a especificação e os mapeamentos para o esquema conceitual. Para a manipulação dos dados existe a Linguagem de Manipulação de Dados (DML) (ELMASRI, NAVATHE 2005, p.24).

2.7.3 SQL

Segundo ensina Gonçalves (2014, p.23) SQL “é uma linguagem para interface com bancos de dados relacionais, isto é, todos os usuários e programas que desejarem realizar alguma tarefa no banco de dados devem fornecer comandos

escritos nesta linguagem”. No início desenvolvida no laboratório de pesquisa da IBM por volta de 1970 foi chamada de SEQUEL e depois teve seu nome alterado para SQL e permanece até hoje como linguagem padrão para banco de dados relacionais. Com o tempo houve a necessidade de implementações na linguagem que atualmente se encontra na versão SQL3.

O nascimento da linguagem SQL deu-se por necessidade de se realizar operações como: diferença, divisão, junção, intersecção, projeção, união. Produto cartesiano e entre outras, pode-se dizer que a linguagem possui uma gama de comandos como de manipulação, definição e controle dos dados. (GONÇALVES, 2014, p.23).

2.7.4 NoSql

Com o grande aumento no número de dados produzidos, viu-se a necessidade de grande sistemas com muito poder de processamento capazes de manipular esses dados gerados de forma eficiente e escalável. Com isso surgiu o NoSql capaz de dar suporte a esse grande volume de dados estruturados, semiestruturados e não estruturados (VIEIRA, et al. p.6).

Para Peres (2018) NoSql tem características bem diferentes do Sql por ser uma base de dados não relacional. O NoSql foi criado para suprir a necessidade quando os desenvolvedores se deparavam com um grande volume de dados, além de prover uma maior flexibilidade na gerencia, escalabilidade mais simples e projeção para funcionamento em hardware de baixo custo e ágil no que diz respeito ao desenvolvimento, pois não é necessário um modelo de dados tão detalhado como no paradigma relacional. O NoSql foi criado para ter uma melhor performance e uma escalabilidade horizontal.

2.7.5 Firebase

Segundo o site oficial do Firebase¹, este foi desenvolvido pela Google e fornece diversos serviços para desenvolvedores como por exemplo: banco de dados disponíveis em nuvem, serviços de hospedagem, entre outras diversas ferramentas capazes de facilitar o armazenamento, acesso, manipulação de dados, integração com outras aplicações, ferramentas de análises e autenticação. O Firebase é muito utilizado por desenvolvedores *mobile* e para aplicações web, onde se torna fácil a integração dos mesmos com o banco de dados unificado. O desenvolvimento com um *back-end* unificado traz ótimos benefícios para a aplicação, pois não é necessário o gerenciamento da infraestrutura, além de promover agilidade no seu desenvolvimento².

2.7.6 Baas

Segundo Lane (2013, p.3) Baas nada mais é do que um *back-end* como um serviço que ganhou o mercado após o grande aumento de dispositivos móveis. O Baas surgiu para atender a grande demanda existente de criação, gerenciamento e implantação do ciclo de vida dos aplicativos. Ele fornece ao desenvolvedor ferramentas capazes de conectar as aplicações *mobile* ao *back-end* armazenado nas nuvens, disponibilizando recursos como: notificações, gerenciamento de usuários, integração com diversas redes sociais, entre outros exigidos por diversos aplicativos.

O autor continua o raciocínio dizendo sobre os benefícios adquiridos com o uso do Baas como os ganhos e eficiência, tempos mais rápidos no desenvolvimento, redução em problemas de fragmentação de diversas plataformas, além de fornecer uma infraestrutura integrada que lida com segurança, escalabilidade e desempenho, o que faz com que o desenvolvedor foque apenas no desenvolvimento do aplicativo. O Baas tem o seu maior foco em desenvolvimento *mobile*, mas pode ser utilizado para a web devido a sua flexibilidade (LANE, 2013, p.3-4)

² Disponível em: <<https://firebase.google.com/>>

3 DESENVOLVIMENTO DO PROTÓTIPO

3.1 Preparação do Ambiente

O ambiente para o desenvolvimento do protótipo da aplicação contou com as seguintes configurações de hardware: processador processador Intel ® Core™ I5-4440 CPU @3.10GHz 3.10 GHz, Memória RAM 8GB com sistema operacional Windows 10 *Education* de 64bits, processador com base em x64, *Hard Disk* de 1TB, Placa de vídeo ASUS STRIX-GTX1060 Series. Estes componentes foram utilizados para uma melhor experiência de desenvolvimento e realização de testes, não sendo exatamente estas configurações necessárias para o desenvolvimento de qualquer protótipo ou aplicação.

3.2 Ferramentas e softwares utilizados

Foram utilizadas para a elaboração e execução do protótipo algumas ferramentas essenciais ao desenvolvimento, que se tornam comuns no dia a dia de um programador. Essas ferramentas são capazes de auxiliar o processo de produção do código, onde aceleram alguns procedimentos em que o programador poderia levar muito tempo sem o auxílio das mesmas. Antes e durante o desenvolvimento do protótipo foram utilizadas algumas ferramentas como: FluidUI, Firebase, *Android*

Studio, Visual Studio Code, Laravel, Composer, Git, XAMPP, NPM, Noede.js e AdminLTE.

3.2.1 FluidUI

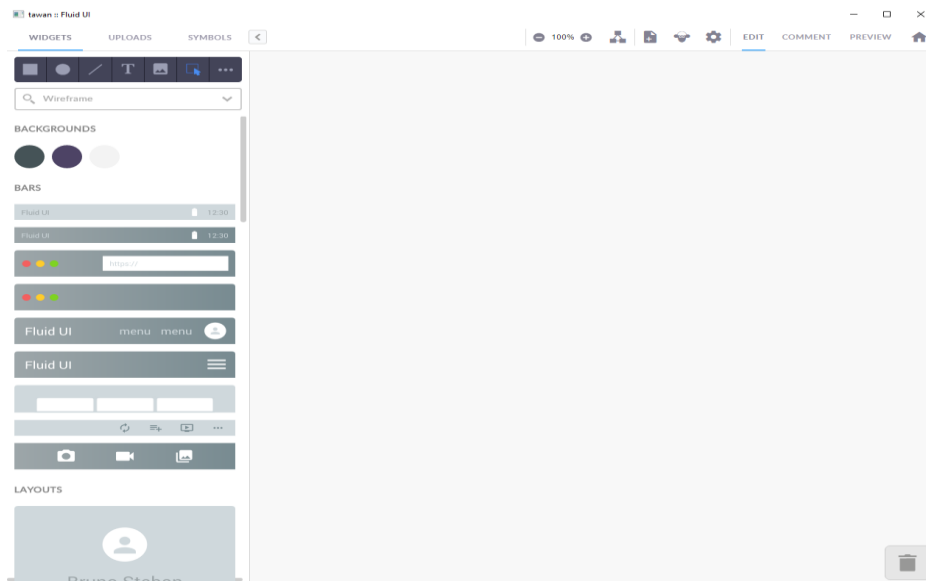
O FluidUI³ é uma ferramenta utilizada para a prototipagem, sendo capaz de proporcionar visualizações interativas, fazendo com que o desenvolvedor tenha uma ideia melhor do que realmente será desenvolvido. Essa ferramenta pode ser utilizada no navegador ou ser efetuado o download⁴ do cliente de desktop. A ferramenta ainda disponibiliza testes do protótipo em qualquer celular ou tablet desde que contenha o aplicativo do player gratuito.

Para o acesso a ferramenta é necessário a criação de uma conta no próprio site, após o download da ferramenta e sua instalação e login efetuado. A tela inicial esta representada pela figura 02.

³ Disponível em: <<https://www.fluidui.com/>>

⁴ Disponível em: <<https://www.fluidui.com/download/>>

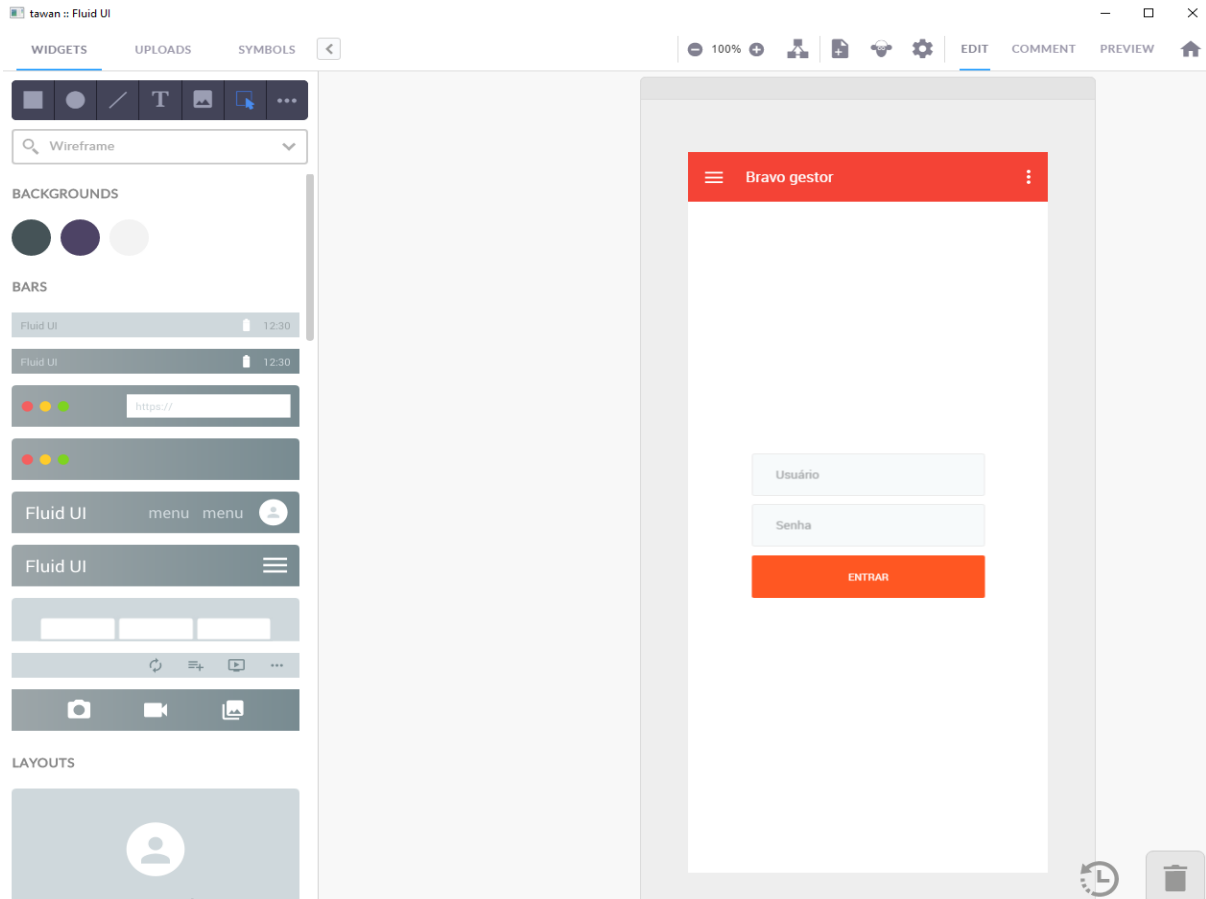
Figura 02: Tela inicial do FuidUI



Fonte: do próprio autor

A ferramenta disponibiliza diversos recursos para a criação de telas, proporcionando uma melhor visualização do protótipo final como pode-se observar na figura 03.

Figura 03: Tela do protótipo no FluidUI



Fonte: do próprio autor

3.2.2 Firebase

O firebase⁵ é um conjunto de ferramentas e recursos disponibilizados pela Google, que proporciona diversos serviços ao desenvolvedor como por exemplo: serviços de hospedagem e banco de dados que foram utilizados na monografia. O acesso a plataforma online pode ser feita através da própria conta da Google.

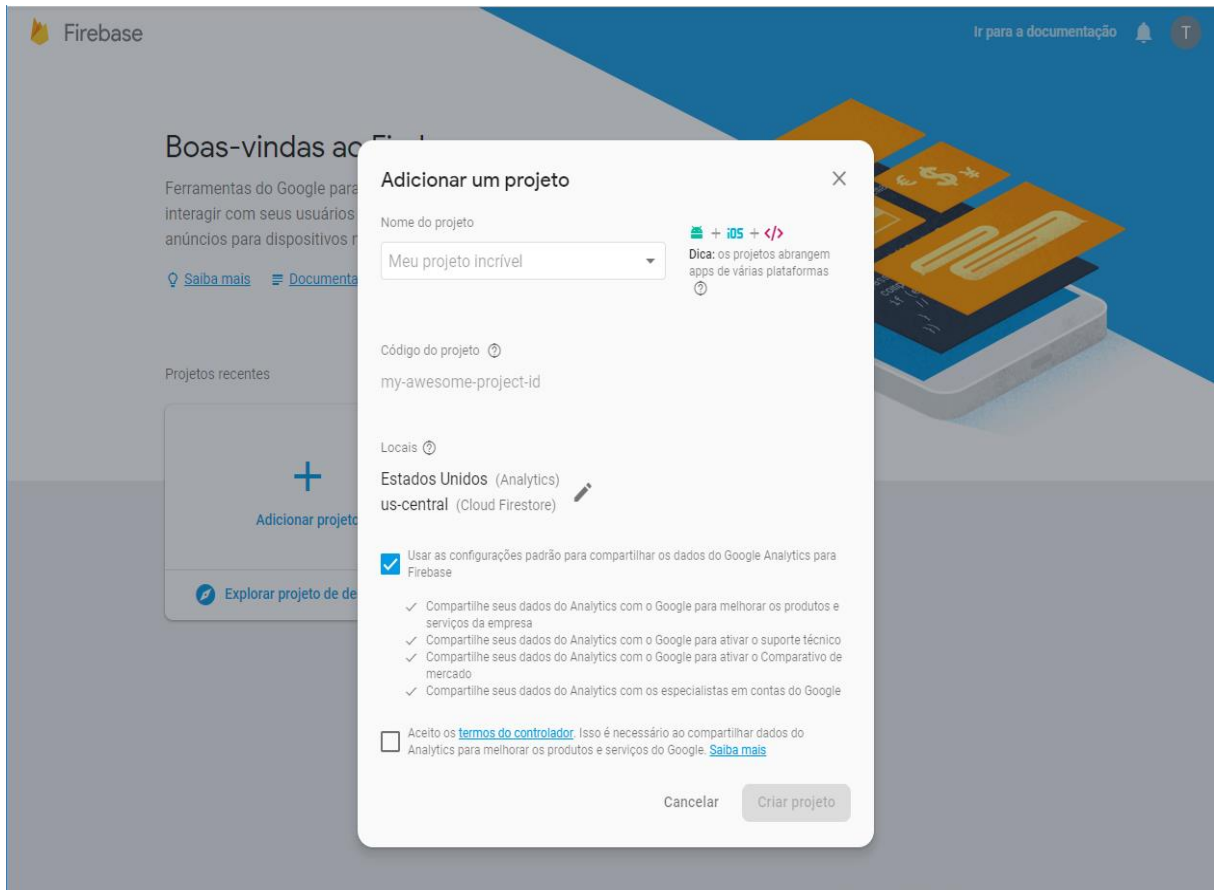
Ao efetuar o login, a parte para criação do projeto se encontra no console⁶

⁵ Disponível em: <<https://firebase.google.com/>>

⁶ Disponível em: <<https://console.firebase.google.com/>>

conforme representa a figura 04, após a criação, o desenvolvedor tem a possibilidade de dar início a confecção do banco de dados.

Figura 04: Criação do projeto no Firebase



Fonte: do próprio autor

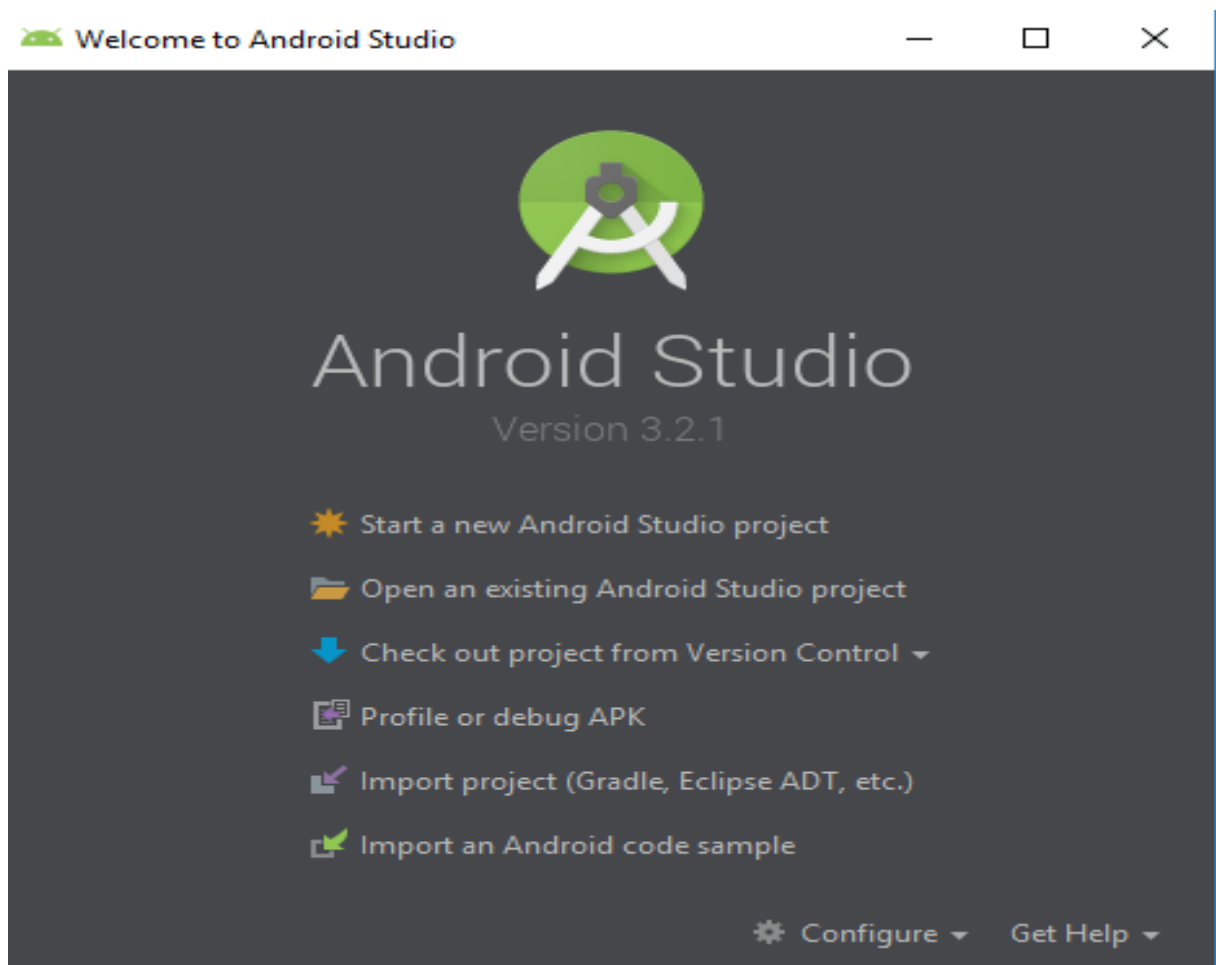
3.2.3 Android Studio

O software utilizado para o desenvolvimento do protótipo do aplicativo móvel, foi o Andoid Studio⁷ na versão 3.2.1 para Windows 64 *bits* que é referência quando se

⁷ Disponível em: < <https://developer.android.com/studio/> >

trata de desenvolvimento *mobile*. Uma grande diferença em comparação com outros softwares de desenvolvimento *android*, é que o *Android Studio* utiliza um moderno sistema de *builds* para a compilação do projeto. Após o download e a sua instalação pode-se verificar a tela inicial como mostra a figura 05.

Figura 05: Tela inicial Andoid Studio

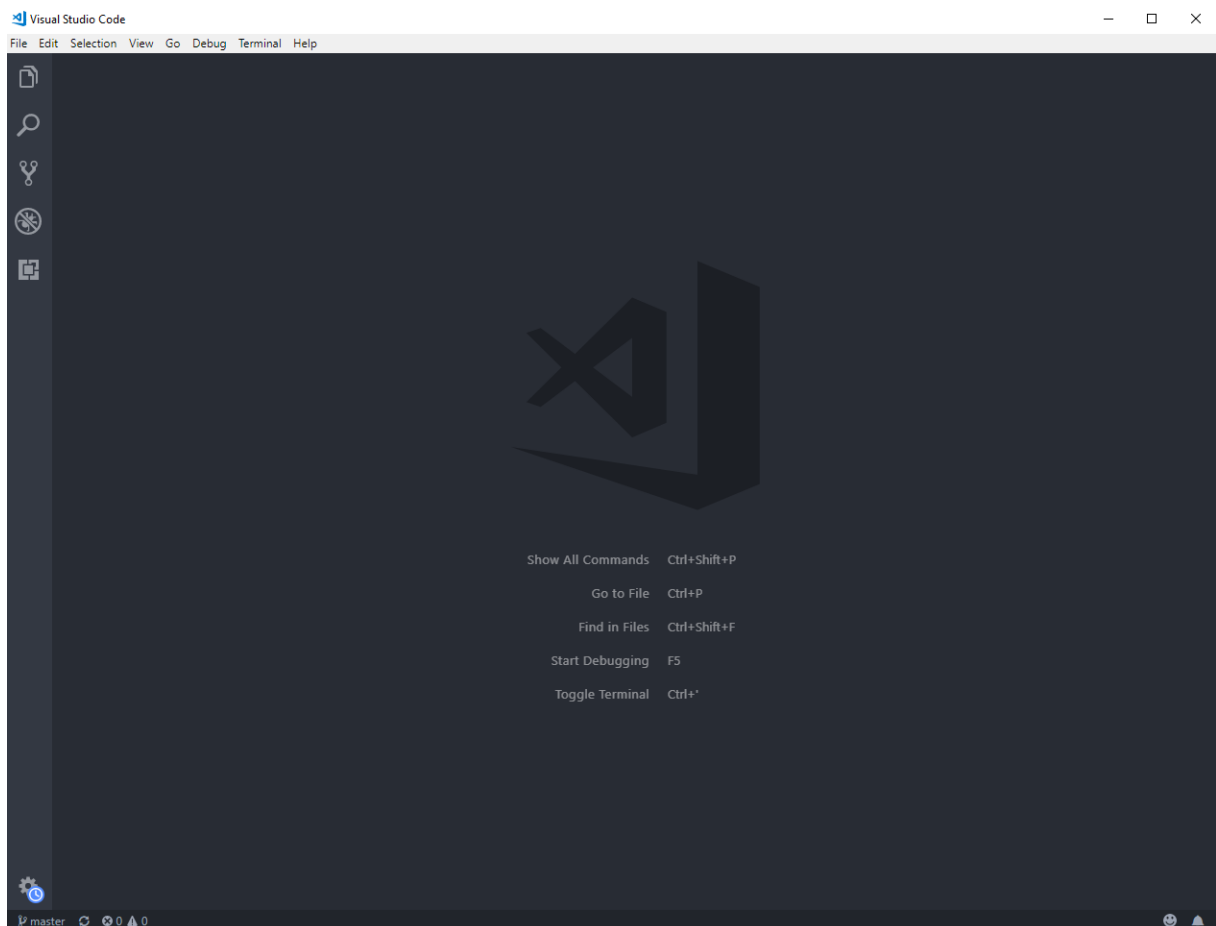


Fonte: do próprio autor

3.2.4 Visual Studio Code

A ferramenta utilizada para o desenvolvimento web foi o editor de texto Visual Studio Code⁸ por ser grátis e de código aberto. A sua versão utilizada foi a 1.28 que tem novos recursos e correções se comparada com outras versões, o Visual Studio Code é um dos mais utilizados pela praticidade e uma gama de extensões disponíveis para serem incorporadas a aplicação. A figura 06 demonstra a tela inicial da ferramenta utilizada.

Figura 06: Tela inicial Visual Studio Code



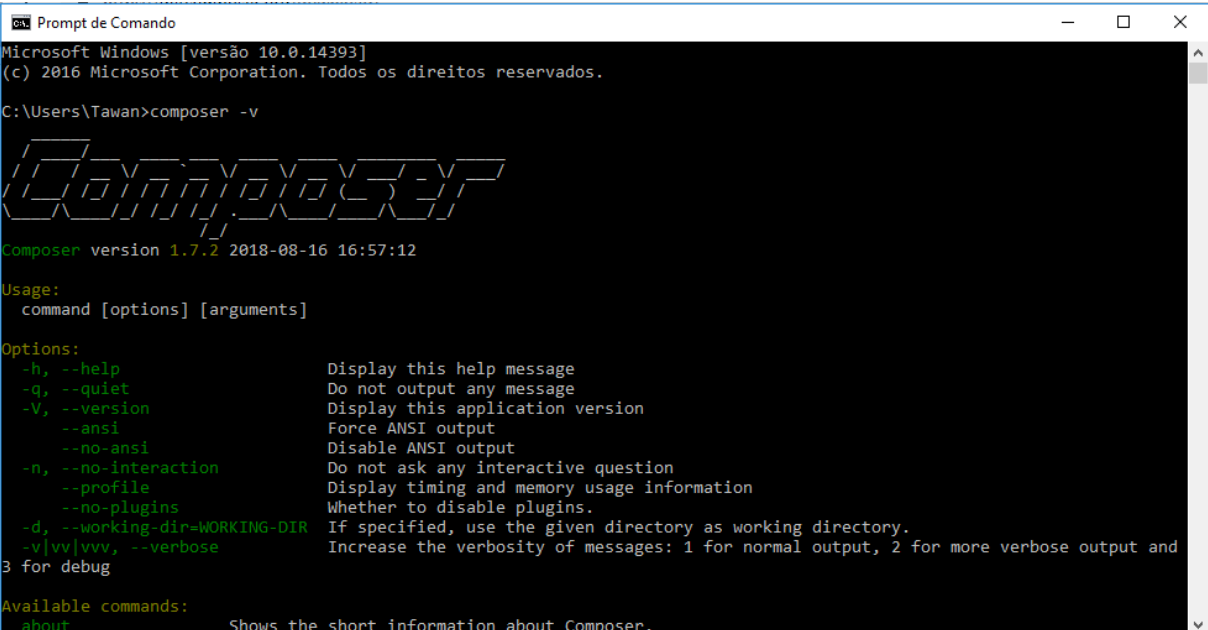
Fonte: do próprio autor

3.2.5 Composer

⁸ Disponível em: <<https://code.visualstudio.com>>

Para o gerenciamento dos pacotes em nível de aplicativo para PHP foi utilizado o software Composer⁹ que disponibiliza uma forma padrão de gerenciamento das dependências do PHP e suas bibliotecas necessárias, podendo facilmente instalar, desinstalar e atualizar qualquer pacote. Após o download e sua instalação, para se verificar se o software foi realmente instalado utiliza-se um comando como mostra na figura 07 para verificar a instalação e a sua versão. Para a monografia foi utilizada a versão 1.7.2.

Figura 07: Verificação da instalação do Composer



```
Microsoft Windows [versão 10.0.14393]
(c) 2016 Microsoft Corporation. Todos os direitos reservados.

C:\Users\Tawan>composer -v

Composer version 1.7.2 2018-08-16 16:57:12

Usage:
  command [options] [arguments]

Options:
  -h, --help                Display this help message
  -q, --quiet               Do not output any message
  -V, --version             Display this application version
  --ansi                    Force ANSI output
  --no-ansi                 Disable ANSI output
  -n, --no-interaction     Do not ask any interactive question
  --profile                 Display timing and memory usage information
  --no-plugins              Whether to disable plugins.
  -d, --working-dir=WORKING-DIR
                           If specified, use the given directory as working directory.
  -v|vv|vvv, --verbose     Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and
  3 for debug

Available commands:
  about                     Shows the short information about Composer.
```

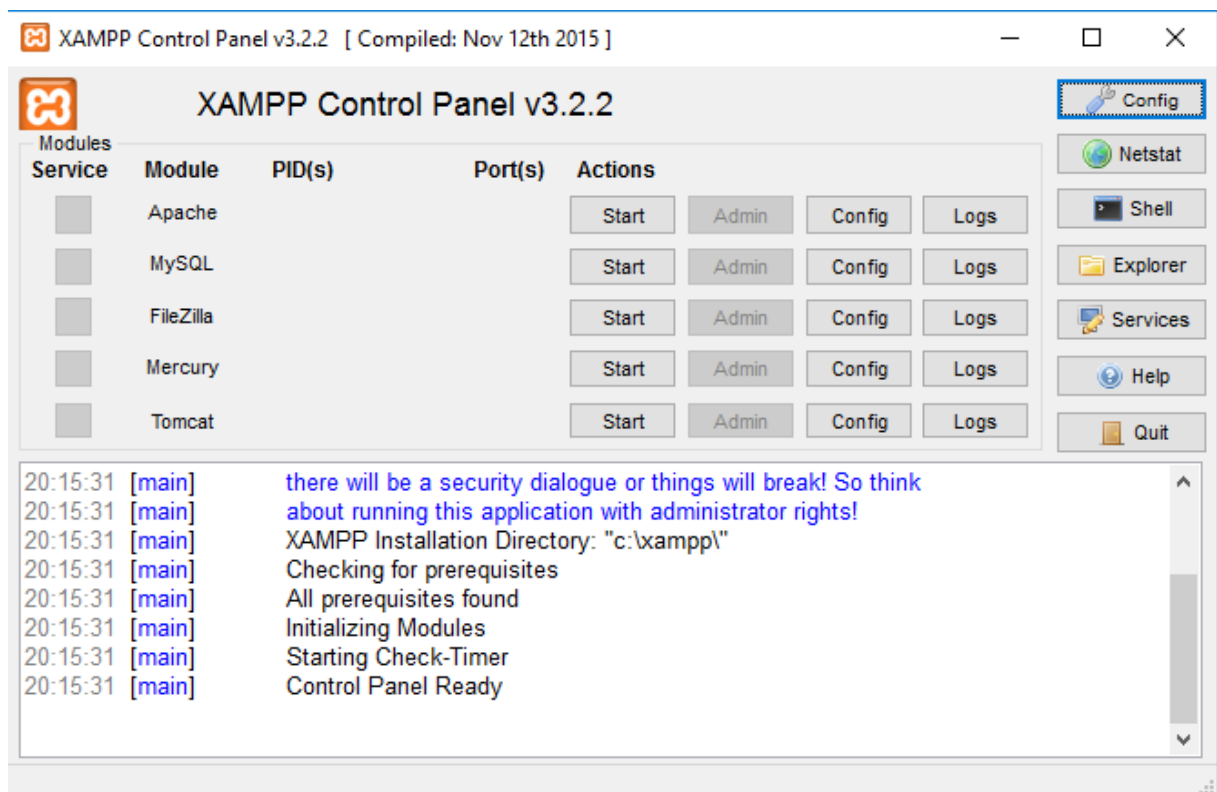
Fonte: do próprio autor

3.2.6 XAMPP

⁹ Disponível em: < <https://getcomposer.org/download/>>

O XAMPP¹⁰ é um servidor que não depende de plataforma, é um software livre que consiste em uma base de dados, servidor web Apache e interpretadores de script como: PHP e Perl. O XAMPP auxilia no desenvolvimento por armazenar arquivos e tornar o endereço local um servidor web. A versão utilizada para a monografia foi a v3.2.2, após a instalação o XAMPP o seu painel de controle é aberto como mostra a figura 08.

Figura 08: Painel de controle de XAMPP



Fonte: do próprio autor

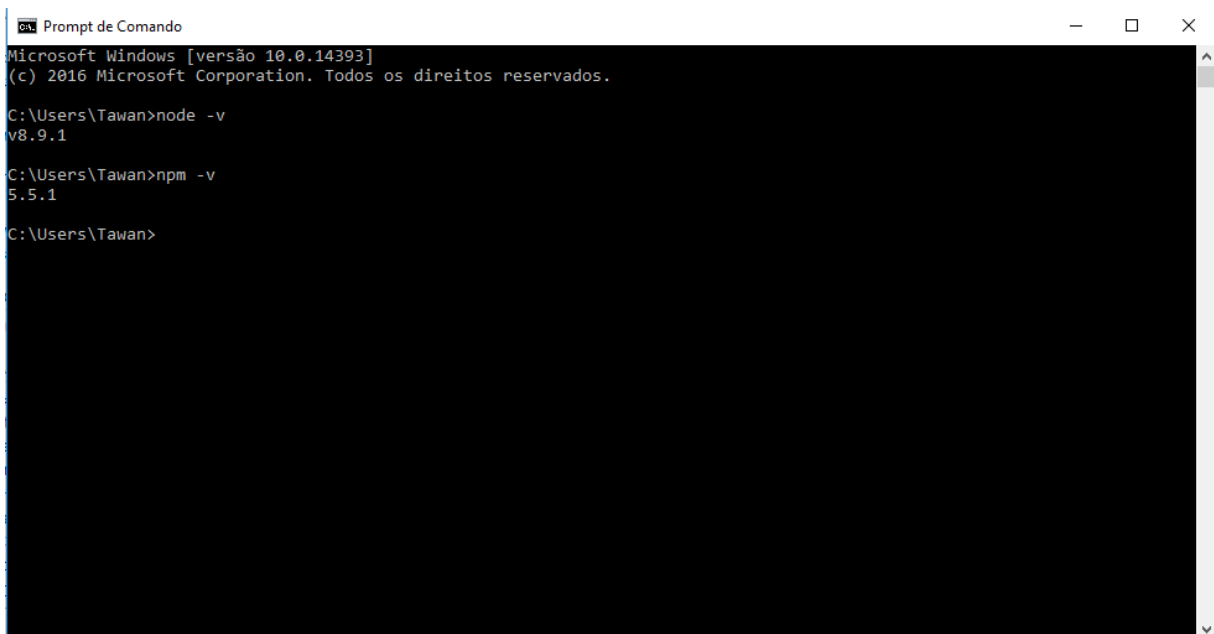
3.2.7 NPM e Node.js

¹⁰ Disponível em: < https://www.apachefriends.org/pt_br/download.html >

O Node.js¹¹ é um interpretador de código JavaScript do lado do cliente para servidores. É utilizado para a aplicação web ter uma maior escalabilidade, tendo a capacidade de manipular diversas conexões ao mesmo tempo.

O NPM é o gerenciador de pacotes padrão da linguagem JavaScript, ele vem instalado juntamente com o Node.js. Após o download e instalação, para a verificação das versões instaladas são utilizados os comandos como mostra a figura 09.

Figura 09: Versões do Node.js e NPM



```
Prompt de Comando
Microsoft Windows [versão 10.0.14393]
(c) 2016 Microsoft Corporation. Todos os direitos reservados.

C:\Users\Tawan>node -v
v8.9.1

C:\Users\Tawan>npm -v
5.5.1

C:\Users\Tawan>
```

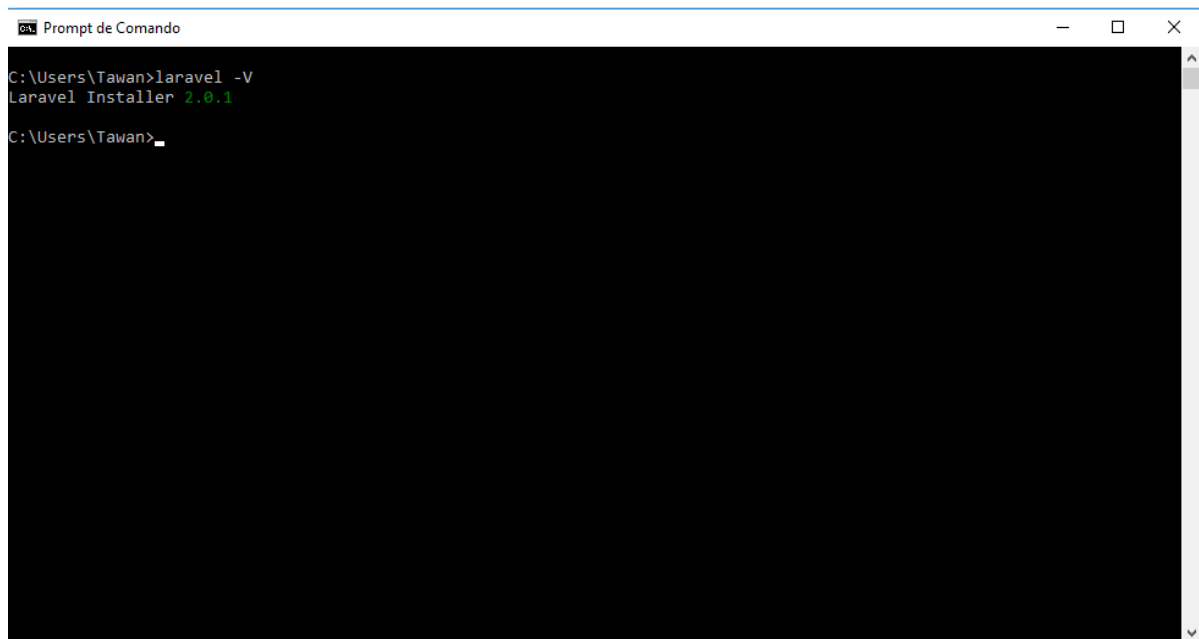
Fonte: do próprio autor

3.2.8 Laravel

¹¹ Disponível em: <<https://nodejs.org/en/download/>>

Para a parte Web foi utilizado o framework PHP Laravel que é livre e *open-source*, é um dos mais utilizados pelos recursos e funcionalidades disponibilizadas, o que traz otimização para projetos de aplicações web. A utilização de um framework traz uma maior produtividade no processo de desenvolvimento visto que, se caracteriza como um modelo de estrutura padrão de desenvolvimento bem definidos. O guia de instalação¹² do Laravel está disponível em seu próprio site e a sua instalação é feita por linha de comando, após seguir o passo a passo, para a verificação da instalação é utilizado o comando como mostra a figura 10.

Figura 10: Versões do Laravel



```
Prompt de Comando
C:\Users\Tawan>laravel -V
Laravel Installer 2.0.1
C:\Users\Tawan>
```

Fonte: do próprio autor

3.2.9 AdminLTE

¹² Disponível em: < <https://laravel.com/docs/5.7> >

O AdminLTE¹³ é um tema gratuito para painel de administração e tem como base o Bootstrap podendo também ser utilizado com o Laravel como um pacote, além de ser um tema simples para a sua implementação, o AdminLTE facilita a vida do desenvolvedor pois conta com diversos recursos prontos de *front-end*. A sua instalação poder ser efetuada de várias maneiras como mostra a figura 11.

Figura 11: Maneiras de instalar o AdminLTE

Baixar

Do Github

Visite a [seção de lançamentos no Github](#) e faça o download da versão mais recente.

[Download do último lançamento](#)

Linha de comando

Via NPM

```
npm install admin-lte --save
```

Via Bower

```
bower install admin-lte
```

Se o bower perguntar qual versão do jQuery usar, escolha a versão 3 ou superior.

Via Composer

```
composer require "almasaeed2010/adminlte=~2.4"
```

Via Git

- Bifurca o repositório ([guia](#)).
- Clone para sua máquina.

```
git clone https://github.com/YOUR_USERNAME/AdminLTE.git
```

Fonte: Site oficial AdminLTE¹⁴

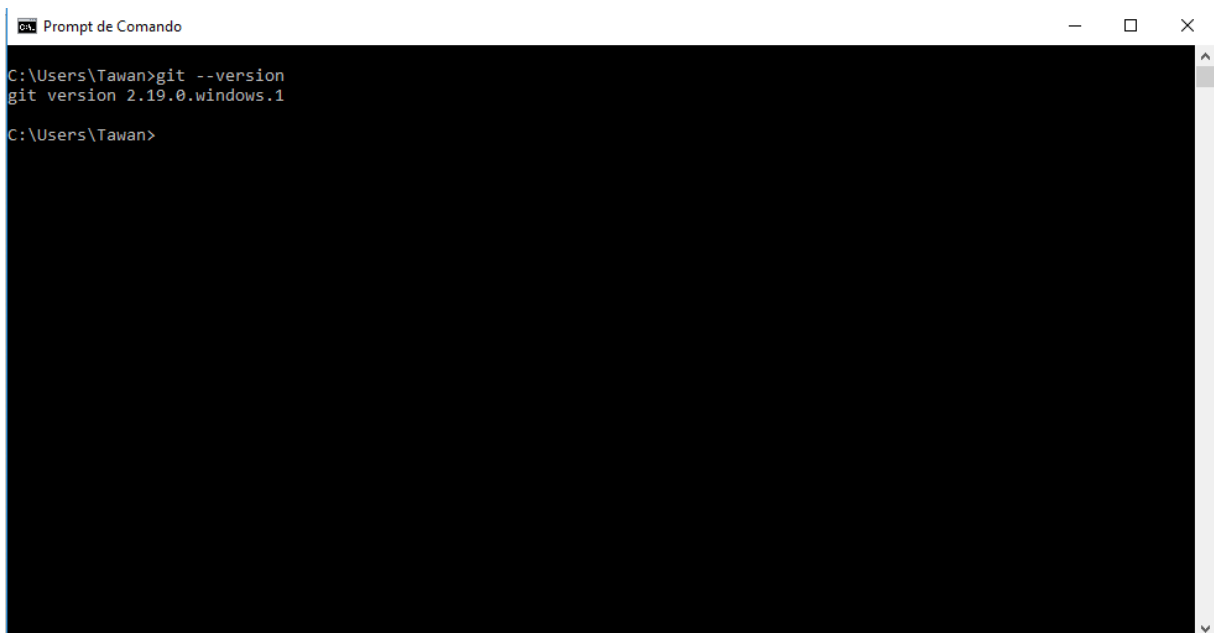
¹³ Disponível em: <<https://adminlte.io/docs/2.4/installation>>

¹⁴ Disponível em: <<https://adminlte.io/docs/2.4/installation>>

3.2.10 Git

Para o controle das versões da aplicação foi utilizado o Git¹⁵ que é um sistema utilizado principalmente no desenvolvimento de softwares, capaz de registrar o histórico de edições de qualquer tipo de arquivo. Após a instalação, para a verificação da versão é utilizado o comando como mostra a figura 12.

Figura 12: Versão do Git instalada



```
Prompt de Comando
C:\Users\Tawan>git --version
git version 2.19.0.windows.1
C:\Users\Tawan>
```

Fonte: do próprio autor

Para o gerenciamento do repositório baseado em Git, foi utilizado o GitLab¹⁶ que permite o desenvolvedor o armazenamento do código em servidor próprio sem a

¹⁵ Disponível em: < <https://git-scm.com/downloads>>

¹⁶ Disponível em: < <https://about.gitlab.com/>>

necessidade de utilizar servidores de terceiros. Para a sua utilização é necessário efetuar o registro no site, podendo ser utilizadas contas da Google, Twitter, GitHub e Bitbucket.

3.3 Metodologia

Para o desenvolvimento do aplicativo, foi utilizada a prototipagem que tem um papel importante na elaboração de um sistema onde os requisitos ainda não estão bem definidos pelo cliente, e para auxiliar o desenvolvedor a compreender melhor o que necessita ser feito. O desenvolvimento foi feito de forma gradativa ou seja, partes da aplicação foram desenvolvidas e logo em seguida efetuados os devidos testes para dar prosseguimento ao processo.

Como o desenvolvedor está inserido no processo de elaboração de orçamentos de venda da empresa OLV Cosméticos, o levantamento de requisitos baseou-se no conhecimento adquirido sobre o processo no cotidiano do desenvolvedor e com diálogos frequentes com os consultores de vendas para saber qual a real necessidade dos mesmos no que tange a elaboração de orçamentos de vendas e sobre melhorias neste processo. A iteração acontece à medida que os testes são efetuados com a obtenção do *feedback* para os ajustes da próxima iteração até que o protótipo se assemelhe ao produto final como mostra a figura 13.

Figura 13: Modelo de Prototipagem



Fonte: PRESSMAN (2010).

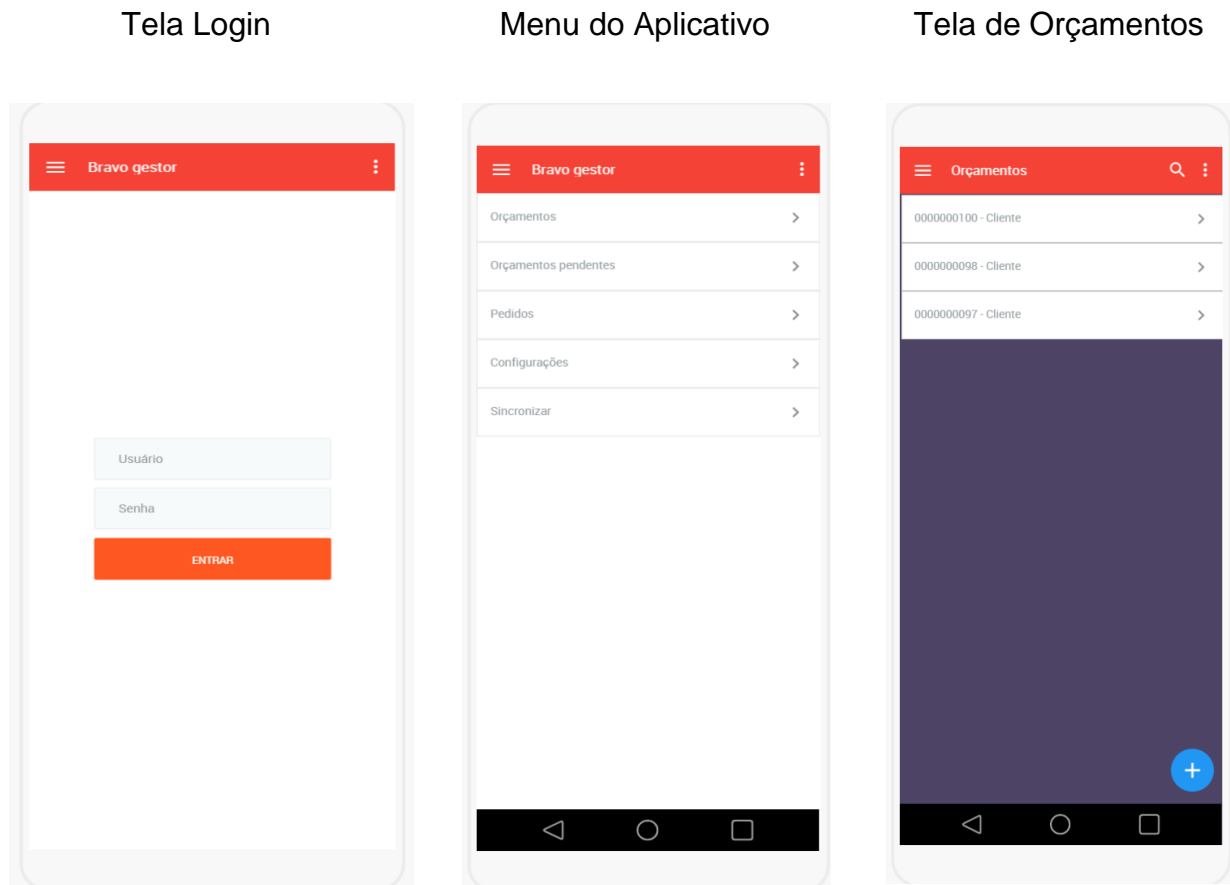
O protótipo tratado na monografia tem a função de mostrar como será a elaboração de orçamentos de venda da empresa OLV Cosméticos, podendo agregar algumas mudanças solicitadas por se tratar de um protótipo.

3.4 Construção da base do protótipo

Primeiramente foi criado um esboço das telas utilizando a ferramenta FluidUI, para uma melhor visualização e definição dos campos e funcionalidades do protótipo. Não sendo exatamente a forma final de como o protótipo ficaria. Essa ferramenta foi

muito útil para que se desse início ao desenvolvimento. A figura 14 mostra um conceito para algumas das telas da aplicação.

Figura 14: Telas do Protótipo no FluidUI



Tela de Criação de Orçamentos

Orçamento/Pedido

ID

ID no bravo gestor

Cliente

Nome do cliente

CPF/CNPJ

Telefone

R\$ total dos produtos

R\$ desconto

R\$ Total do orçamento

Produtos

Tela de produtos

Produtos

123456-Shampoo

123456-Condicionador

123456-Creme

123456-Produto escolhido

R\$ Unitário UN

Quantidade

R\$ Total do produto

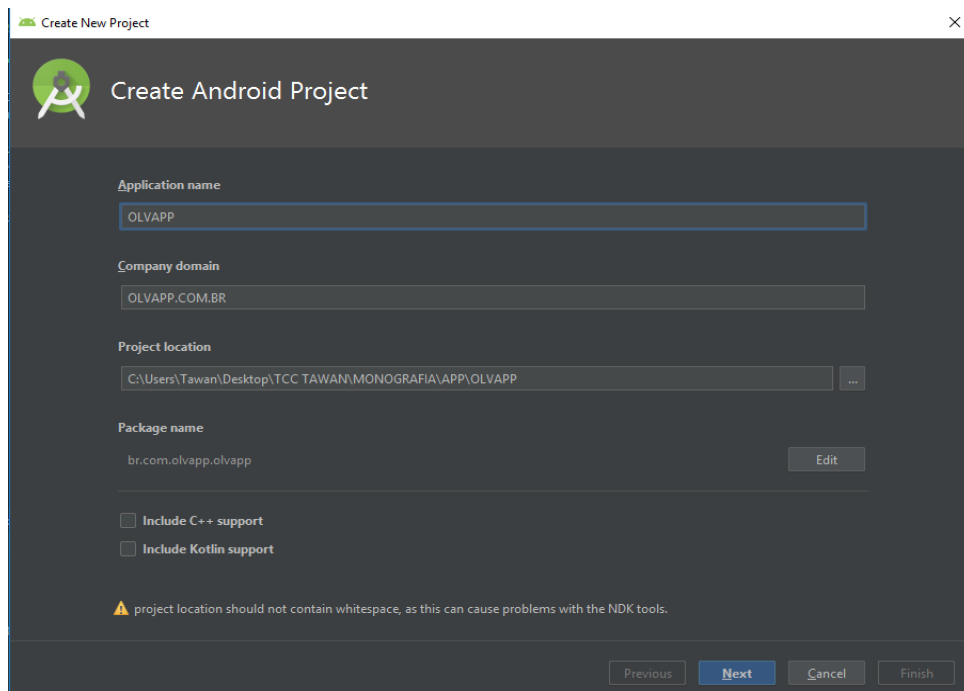
R\$ Total do orçamento

Fonte: do próprio autor

3.5 Construção da aplicação

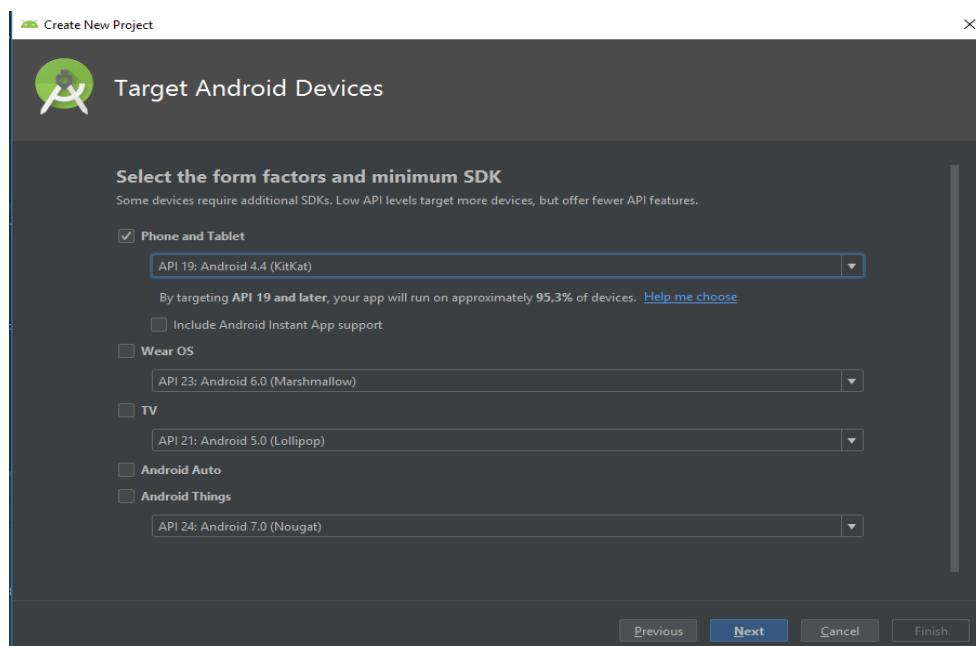
O primeiro passo foi a criação do projeto na ferramenta *Android Studio* com o nome do aplicação, o local onde ficará armazenado na máquina o projeto e a escolha da versão da API que será utilizada baseando-se na porcentagem de utilização dos usuários conforme mostra a figura 15 e 16.

Figura 15: Criação do Projeto 01



Fonte: do próprio autor

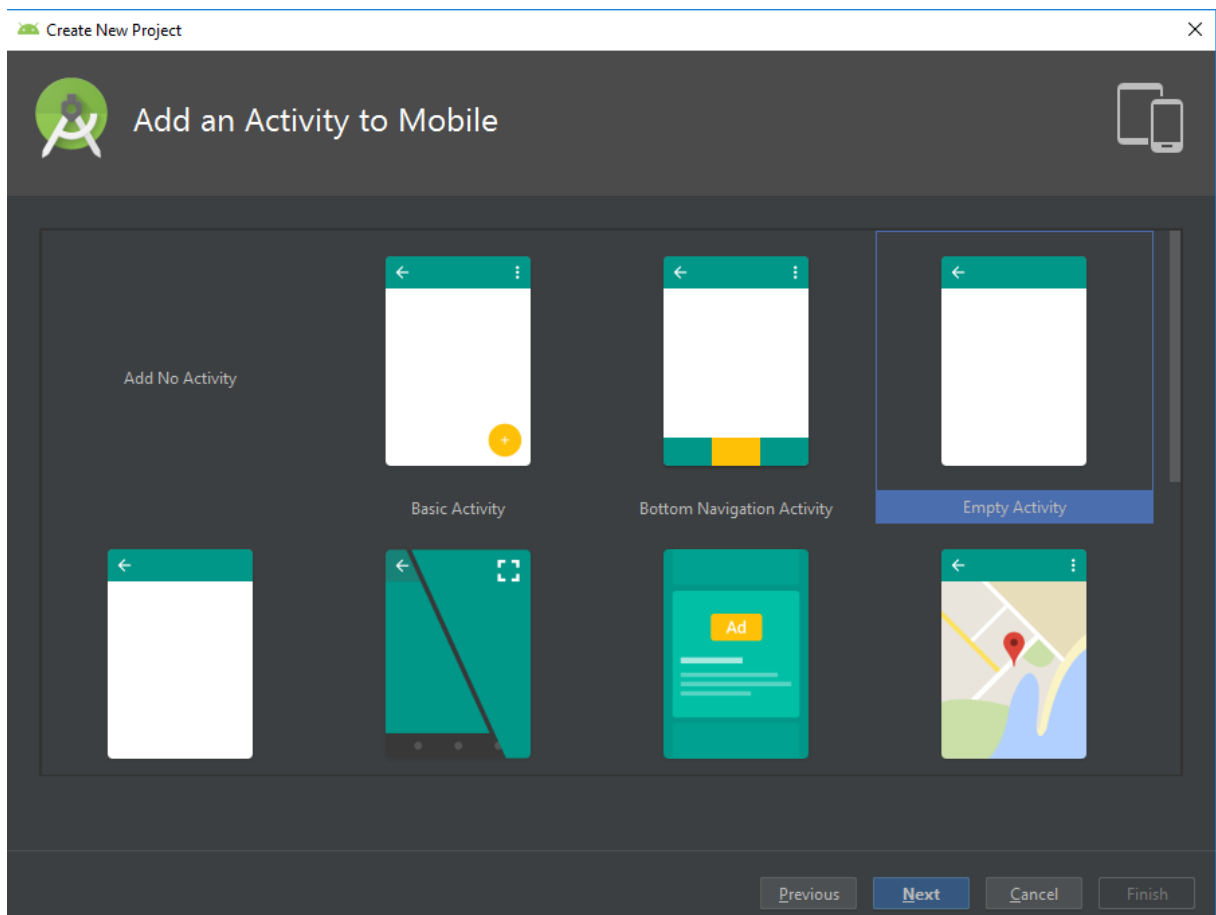
Figura 16: Criação do Projeto 02



Fonte: do próprio autor

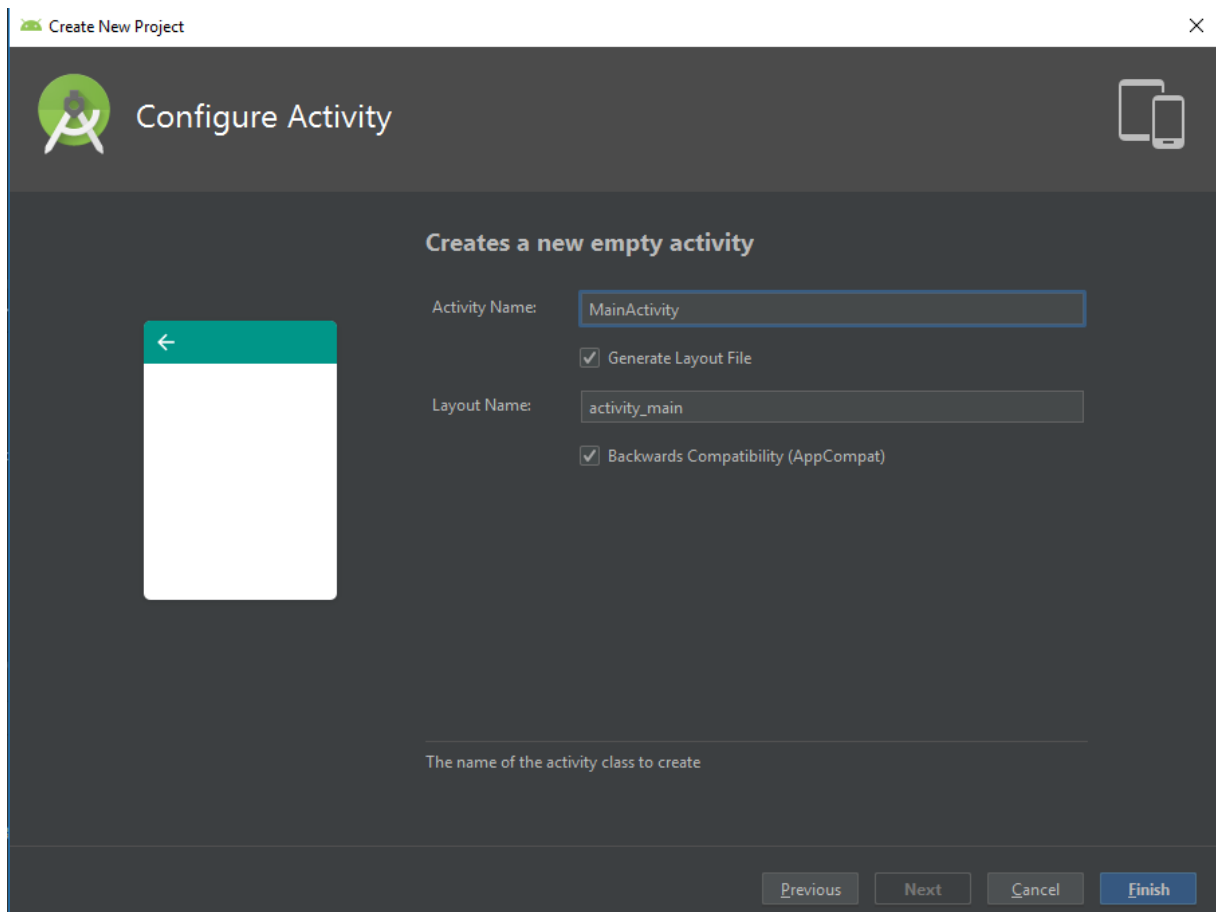
Após essas definições foi escolhido o layout da classe principal da aplicação chamada de *MainActivity* como mostra as figuras 17 e 18.

Figura 17: Criação do Projeto 03



Fonte: do próprio autor

Figura 18: Criação do Projeto 04



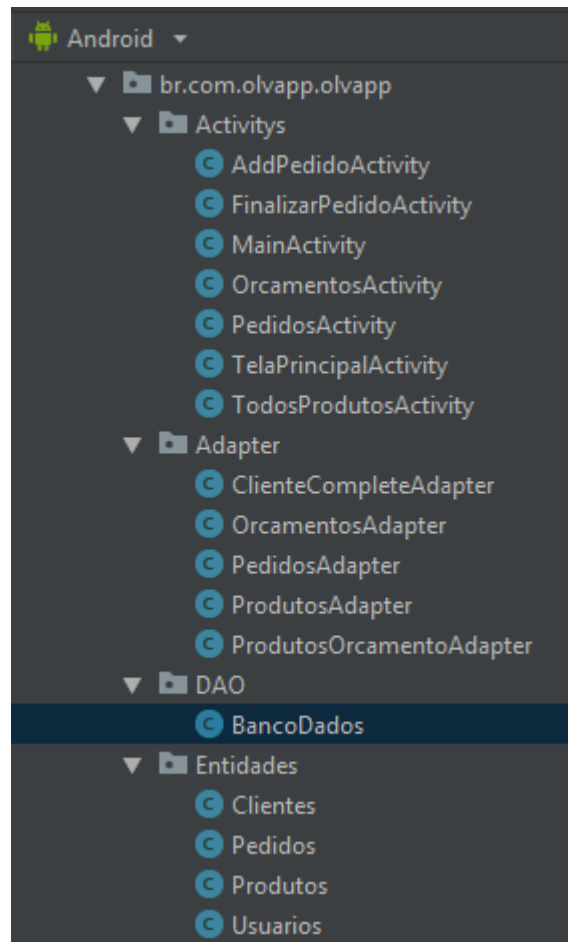
Fonte: do próprio autor

Seguindo com a preparação do protótipo, logo após a definição da tela *MainActivity* foram criadas gradativamente as seguintes telas: (1) *TelaPrincipalActivity*, (2) *OrcamentosActivity*, (3) *PedidosActivity*, (4) *AddPedidoActivity*, (5) *TodosProdutosActivity* (6) *FinalizarPedidoActivity*. Como em várias destas telas existem *arrays* e listas de clientes, produtos, orçamentos e pedidos, foi necessário a criação de *adapters* como: (1) *ClienteCompleteAdapter*, (2) *OrcamentosAdapter*, (3) *PedidosAdapter*, (4) *ProdutosAdapter*, (5) *ProdutosOrcamentoAdapter*, para servir como ponte para os dados destinados às *Views*. Além dos *adapters* foram criadas as entidades: (1) *Clientes*, (2) *Pedidos*, (3) *Produtos*, (4) *Usuarios*, para o acesso aos dados da aplicação foi necessário a criação

da classe (1) *BancoDados*, para a codificação logo após a criação do banco.

Toda essa estrutura do projeto pede ser verificada na figura 19 a seguir.

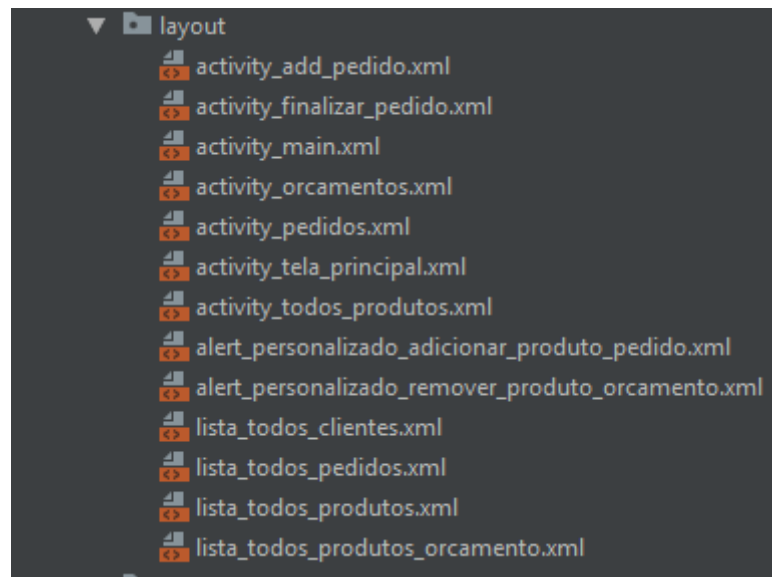
Figura 19: Estrutura de Classes



Fonte: do próprio autor

Os *layouts* foram definidos gradativamente de acordo com a criação das *activities* como pode-se observar na figura 20.

Figura 20: Estrutura do Layout



Fonte: do próprio autor

3.6 Banco de dados

Para que a aplicação tenha sucesso em suas requisições foi necessário a criação de um banco de dados, utilizando a plataforma Firebase disponibilizada pela Google. O firebase por ser um banco de dados NoSQL ou seja, um banco não relacional, não foi elaborado um DER para representação da estrutura do banco de dados. Após o processo de cadastro no site e login, partimos do ponto de criação do projeto do banco de dados representado pela figura 21.

Figura 21: Criação do Bando de Dados

Adicionar um projeto ✕

Nome do projeto

OLVAPP

+ iOS + </>

Dica: os projetos abrangem apps de várias plataformas ?

Código do projeto ?

olvapp-f4537

Locais ?

Estados Unidos (Analytics)

us-central (Cloud Firestore)

Usar as configurações padrão para compartilhar os dados do Google Analytics para Firebase

- ✓ Compartilhe seus dados do Analytics com o Google para melhorar os produtos e serviços da empresa
- ✓ Compartilhe seus dados do Analytics com o Google para ativar o suporte técnico
- ✓ Compartilhe seus dados do Analytics com o Google para ativar o Comparativo de mercado
- ✓ Compartilhe seus dados do Analytics com os especialistas em contas do Google

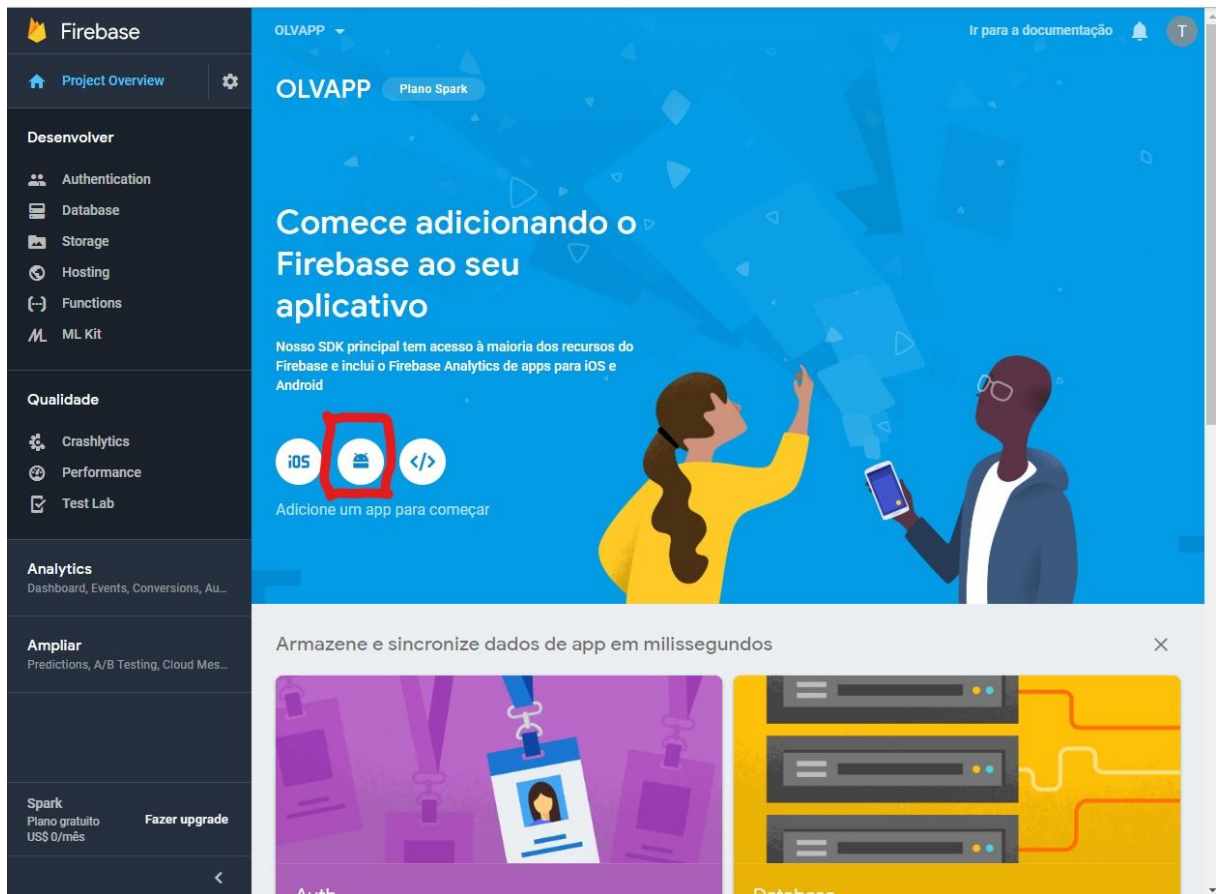
Aceito os [termos do controlador](#). Isso é necessário ao compartilhar dados do Analytics para melhorar os produtos e serviços do Google. [Saiba mais](#)

Cancelar **Criar projeto**

Fonte: do próprio autor

Após a definição do nome do banco de dados e a finalização, irá abrir a tela inicial do projeto, onde foi selecionado o tipo de aplicativo que será utilizado juntamente com a base de dados como mostra a figura 22.

Figura 22: Definição do tipo de aplicação



Fonte: do próprio autor

Foi selecionado a plataforma *Android*, onde será feita a parte de ligação entre o banco de dados e a aplicação conforme a figura 23.

Figura 23: Registro da aplicação

× Adicionar o Firebase ao app para Android

1 Registrar app

Nome do pacote do Android ⓘ

br.com.olvapp.olvapp

Apelido do app (opcional) ⓘ

OLVAPP

Certificado de assinatura de depuração SHA-1 (opcional) ⓘ

[Redacted SHA-1 key]

Necessário para o Dynamic Links, para o Invites e para o Login do Google ou para receber suporte por telefone no Auth. Edite o SHA-1 nas configurações.

Registrar app

2 Fazer o download do arquivo de configuração

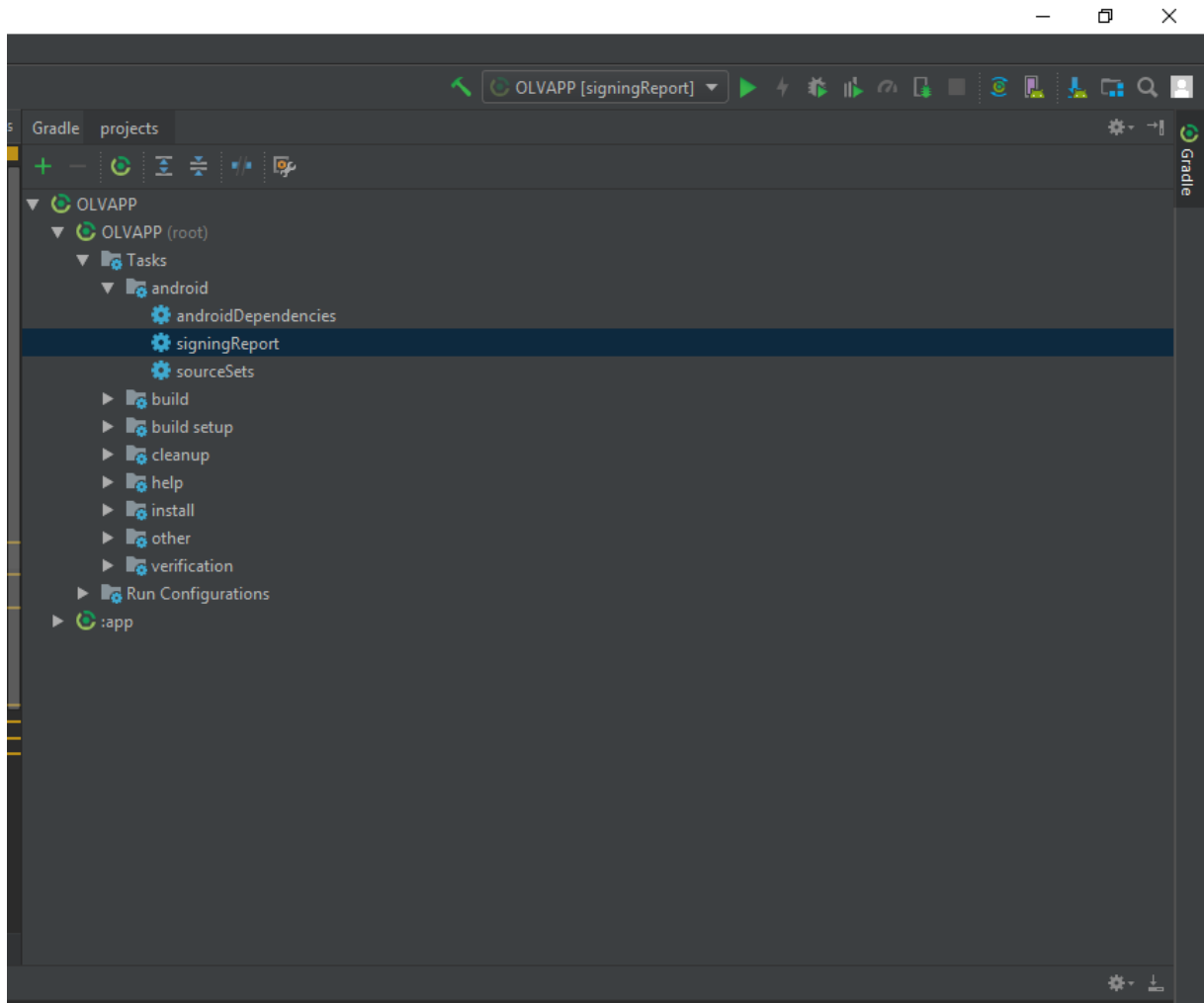
3 Adicionar o SDK do Firebase

4 Execute seu app para verificar a instalação

Fonte: do próprio autor

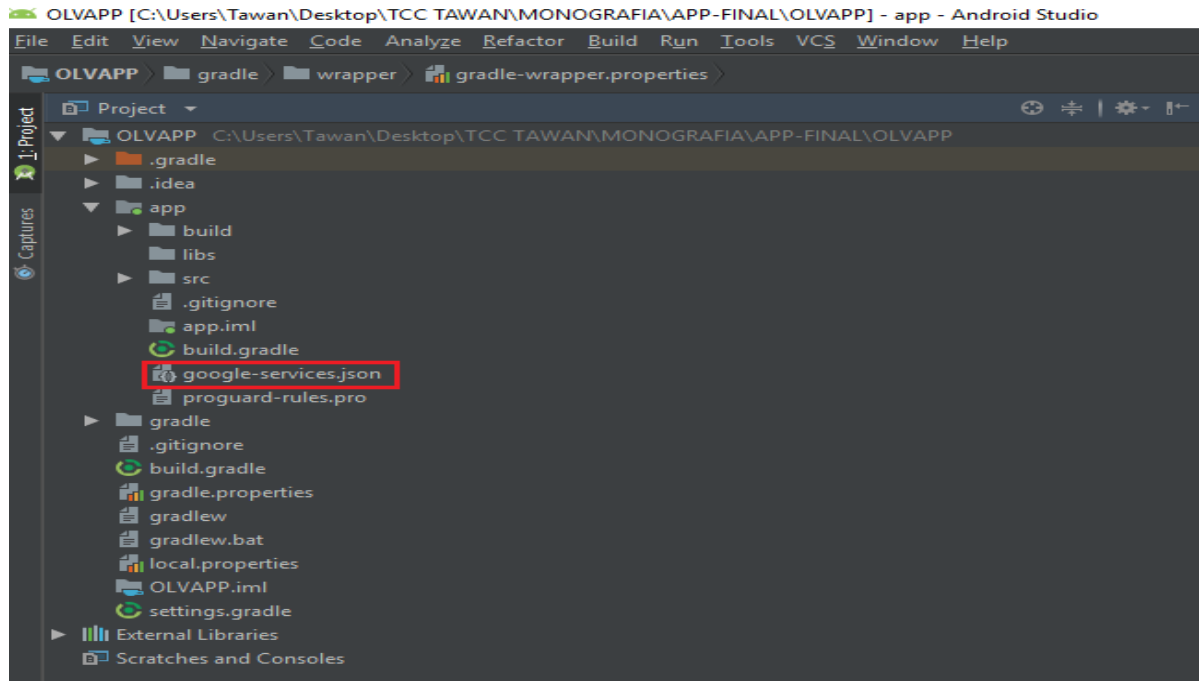
Nos campos solicitados para o registo foi acrescentado o nome do pacote, juntamente com um apelido e a chave SHA-1 da aplicação disponíveis no *Android Studio* conforme mostra a figura 24 e 25 identificando a localização da chave no *Android Studio*.

Figura 24: Aquisição chave SHA-1 01



Fonte: do próprio autor

Figura 26: Arquivo de configuração Firebase



Fonte: do próprio autor

Depois de efetuar o download do arquivo de configurações, é necessário o acréscimo de algumas dependências que serão utilizadas pelo Firebase no projeto *android* como mostra o trecho de código 1 e 2.

Trecho de código 1: Dependências do projeto Android 01

```
...
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    //noinspection GradleCompatible
    implementation 'com.android.support:appcompat-v7:28.0.0-rc02'
    implementation 'com.android.support.constraint:constraint-
layout:1.1.3'
    implementation 'com.google.firebase:firebase-core:16.0.1'
    //noinspection GradleCompatible
    implementation 'com.android.support:recyclerview-v7:27.1.1'
```

```

testImplementation 'junit:junit:4.12'
androidTestImplementation
'com.android.support.test:runner:1.0.2'
androidTestImplementation
'com.android.support.test.espresso:espresso-core:3.0.2'

compile 'com.google.firebase:firebase-database:16.0.1'
compile 'com.google.firebase:firebase-storage:16.0.1'
compile 'com.google.firebase:firebase-auth:16.0.2'
implementation 'com.android.support:design:28.0.0-rc02'

}
apply plugin: 'com.google.gms.google-services'

```

O trecho de código acima representa alguns plug-ins do projeto e outros dos serviços da Google para o *Gradle* do *Android* que irá fazer o carregamento do arquivo que foi disponibilizado para download e foi acrescentado ao projeto como mostrou a figura 26.

Fonte: do próprio autor

Trecho de código 02: Dependências do projeto Android 02

```

buildscript {

    repositories {
        google()
        jcenter()
    }

    dependencies {
        classpath 'com.android.tools.build:gradle:3.2.1'
        classpath 'com.google.gms:google-services:4.0.1'

        // NOTE: Do not place your application dependencies here;
they belong
        // in the individual module build.gradle files
    }
}

```

```

}
allprojects {
    repositories {
        google()
        jcenter()
    }
}

task clean(type: Delete) {
    delete rootProject.buildDir
}

```

Fonte: do próprio autor

Após adicionar as dependências necessárias do firebase na aplicação, foi codificada a classe BancoDados para a utilização do banco criado, a criação dessa classe é de extrema importância para o funcionamento da aplicação, além de ser utilizada em todas as *activities*, é através dela que os dados existentes no firebase poderão ser acessados pelo protótipo. A classe pode ser verificado no trecho de código 03.

Trecho de código 03: Codificação da classe BancoDados

```

package br.com.olvapp.olvapp.DAO;

import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;

public class BancoDados {

    public static DatabaseReference referenceFirebase;
    public static FirebaseAuth autenticacao;

    public static DatabaseReference getFirebase(){

        if(referenceFirebase == null){
            referenceFirebase = FirebaseDatabase.getInstance().getReference();
        }
    }
}

```



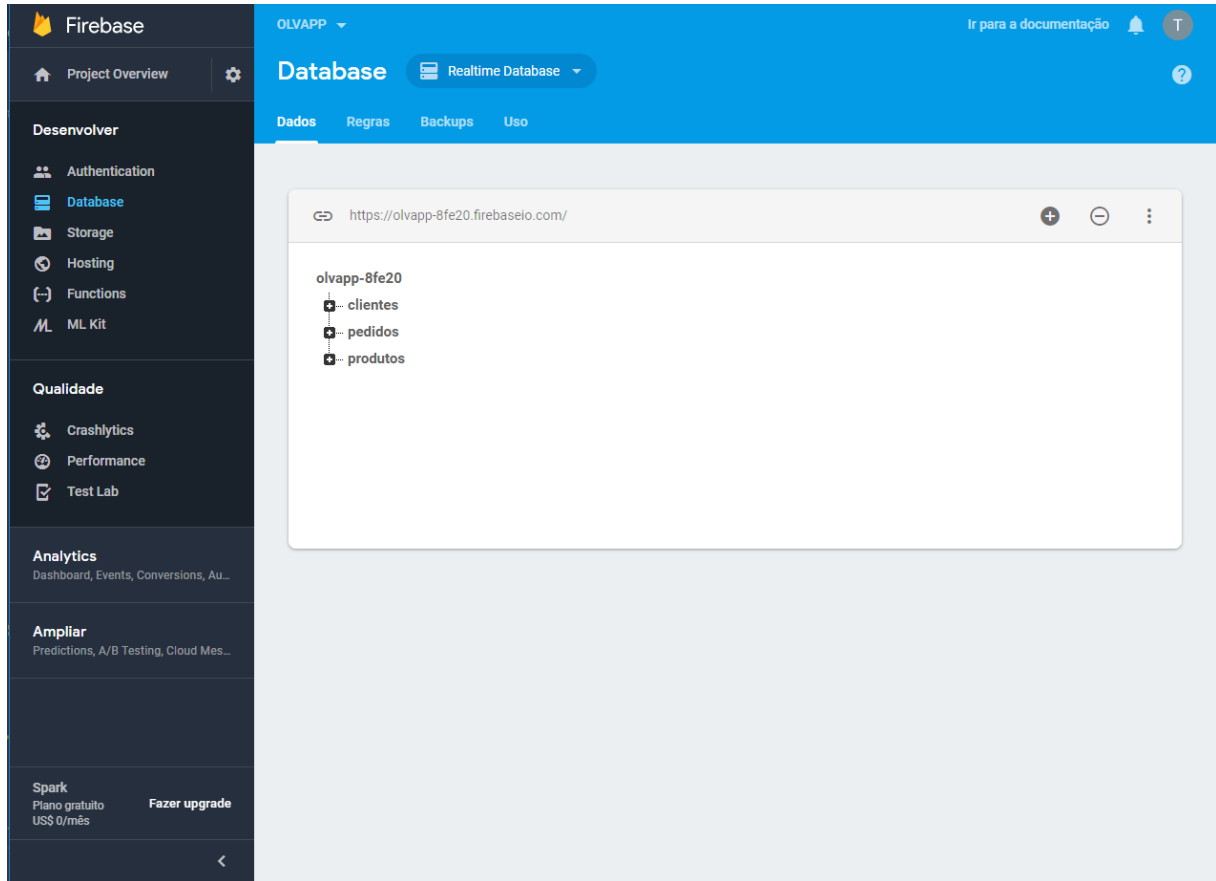
```
    }  
    return referenceFirebase;  
}  
public static FirebaseAuth getAutenticacao(){  
    if(autenticacao == null){  
        autenticacao = FirebaseAuth.getInstance();  
    }  
    return autenticacao;  
}  
}
```

O trecho de código acima representa a classe BancoDados da aplicação utilizada em todo o projeto *android*.

Fonte: do próprio autor

Posteriormente a codificação da classe BancoDados no *android*, foi criado as tabelas do banco de dados no firebase, para acesso ao banco a criação de tabelas e inserção de dados, basta acessar o console com *login* e senha, selecionar o projeto desejado e se dirigir a parte de *database* no menu. A -figura 27 mostra como ficou as tabelas do banco.

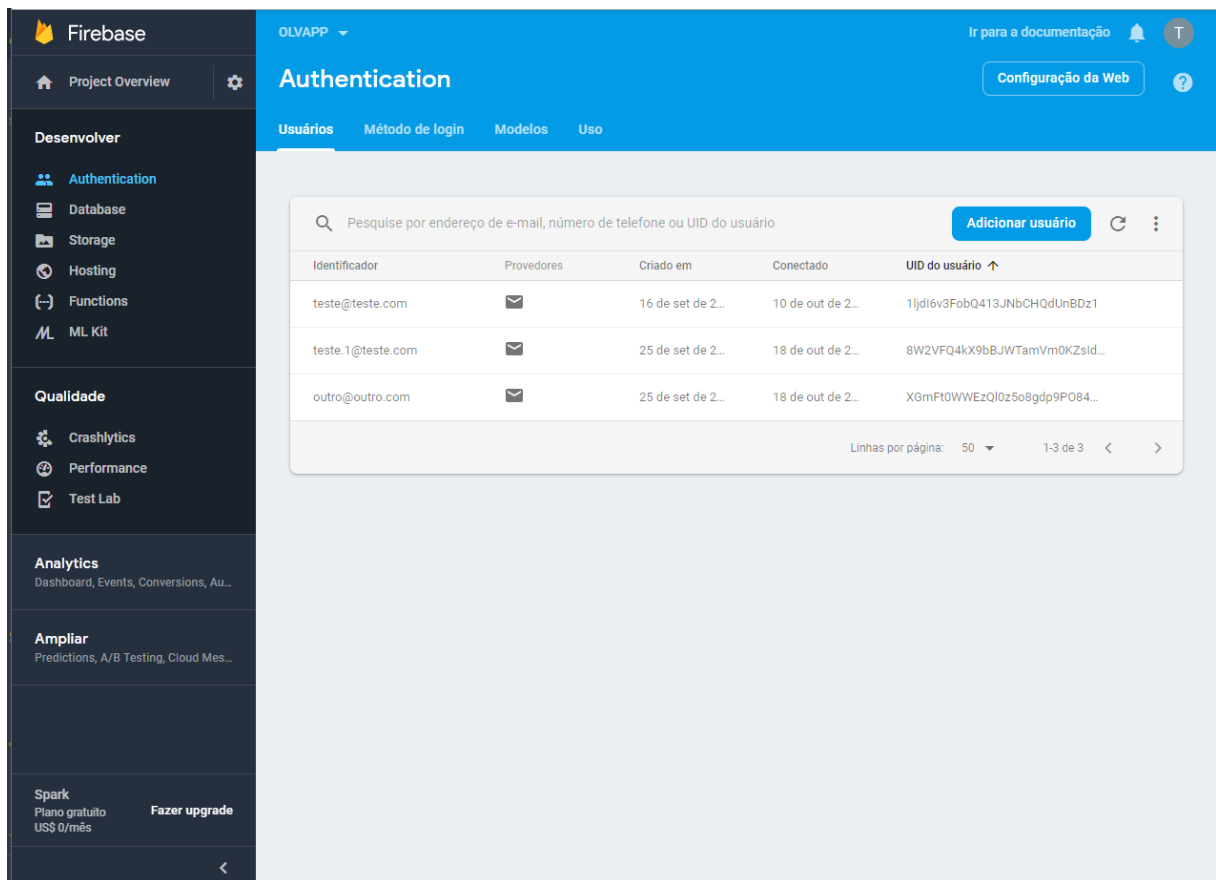
Figura 27: Tela das tabelas do Banco de Dados



Fonte: do próprio autor

A parte do acesso ao protótipo da aplicação utiliza um recurso disponibilizado pela plataforma, que é o método de autenticação do próprio firebase, onde o acesso ao aplicativo só é disponível para usuário com e-mail e senha cadastrados na base como mostra a figura 28.

Figura 28: Tela dos usuários da aplicação



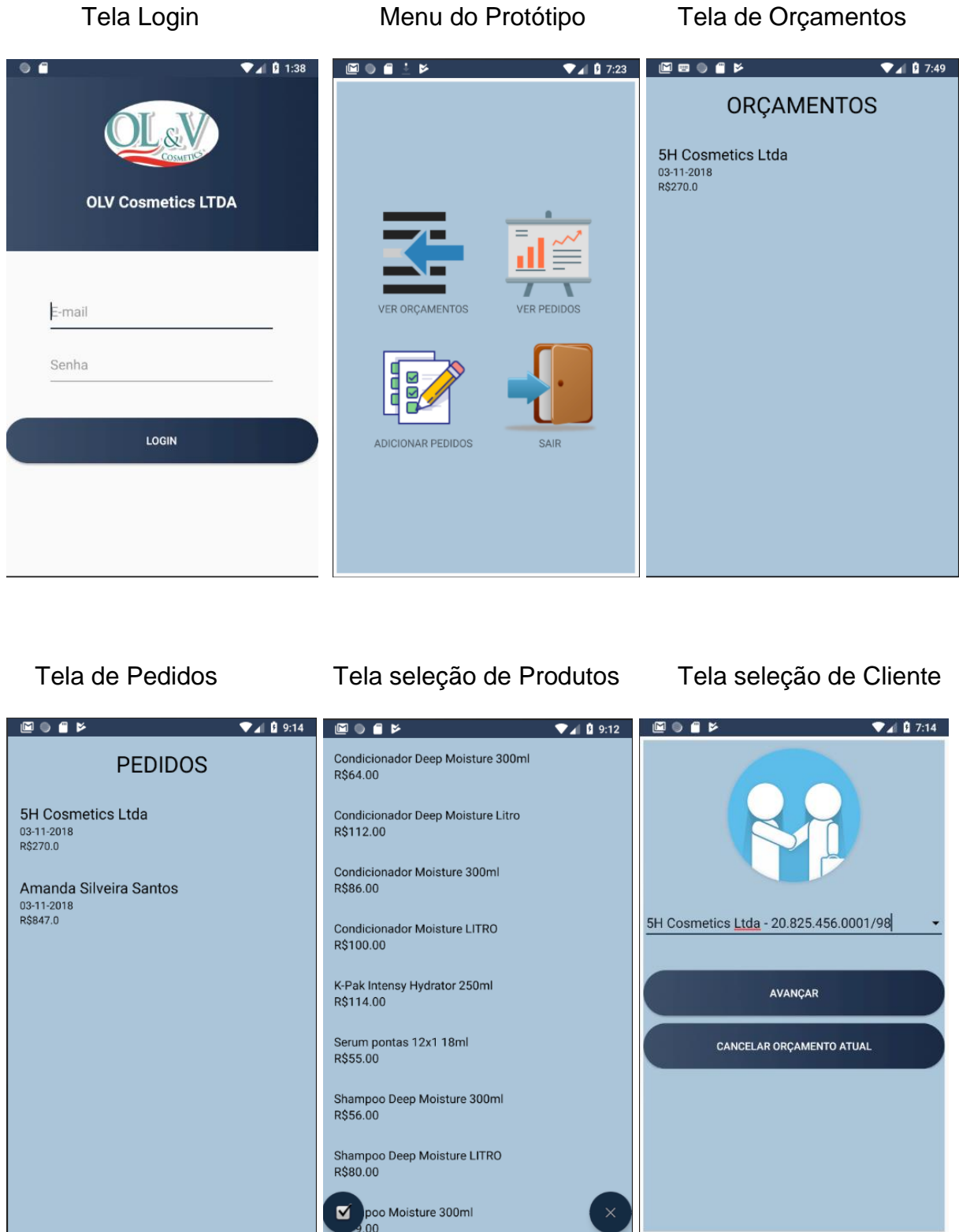
Fonte: do próprio autor

Os usuários podem ser cadastrados através da parte de autenticação no menu do console do firebase

3.7 Finalização do Protótipo

Com a base de dados pronta e o protótipo da aplicação registrado ao firebase, foi dado prosseguimento ao desenvolvimento do protótipo o que resultou no seguinte conteúdo como pode-se observar na figura 29.

Figura 29: Telas do Protótipo



Seleção de quantidade



Finalização do Orçamento



Fonte: do próprio autor

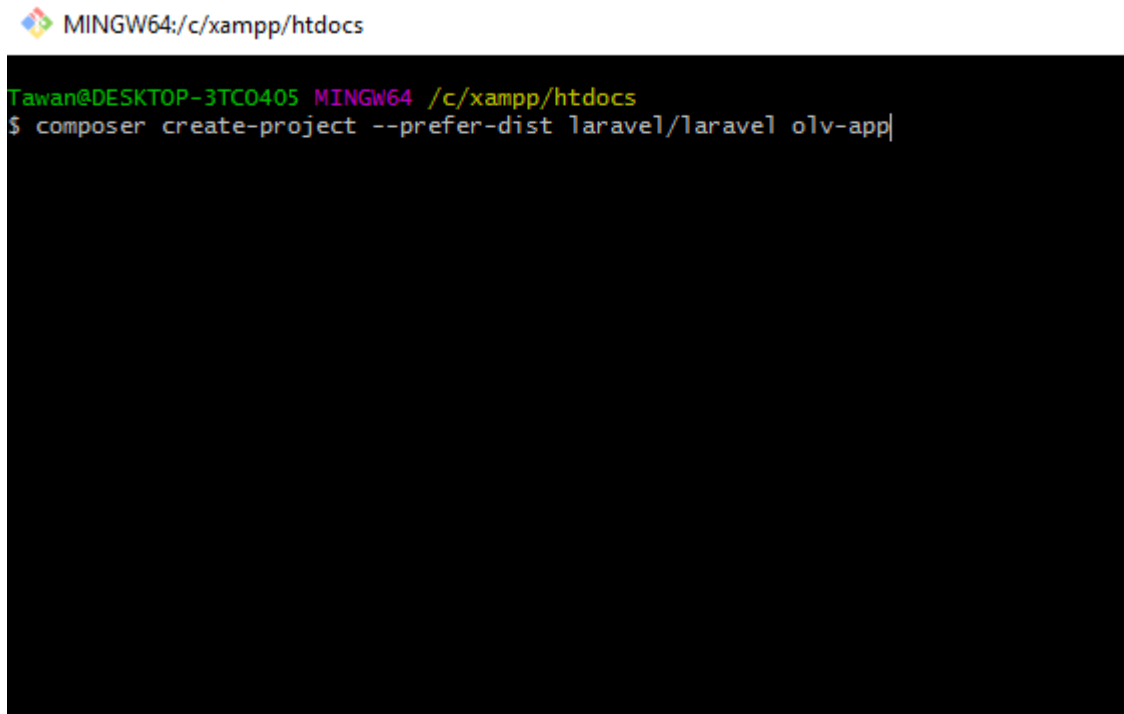
3.8 Painel Administrativo

A empresa ainda não conta com uma API do sistema interno para fazer a comunicação com a aplicação, o que auxiliaria os colaboradores nas realizações de suas tarefas. Desse modo foi desenvolvido um painel administrativo para a visualização e troca de informações dos possíveis orçamentos elaborados pelo protótipo da aplicação. O painel servirá como base de retorno para informar quais orçamentos foram liberados.

O desenvolvimento foi elaborado utilizando ferramentas já citadas anteriormente como: Laravel, AdminLTE, Visual Studio Code, Xampp, Composer e entre outros, a linguagem de programação utilizada foi PHP.

A figura 30 mostra a tela do comando utilizado para criar o projeto web.

Figura 30: Criando a aplicação web

A terminal window with a black background and white text. The title bar at the top reads 'MINGW64:/c/xampp/htdocs'. The prompt is 'Tawan@DESKTOP-3TC0405 MINGW64 /c/xampp/htdocs'. The command entered is '\$ composer create-project --prefer-dist laravel/laravel olv-app|'.

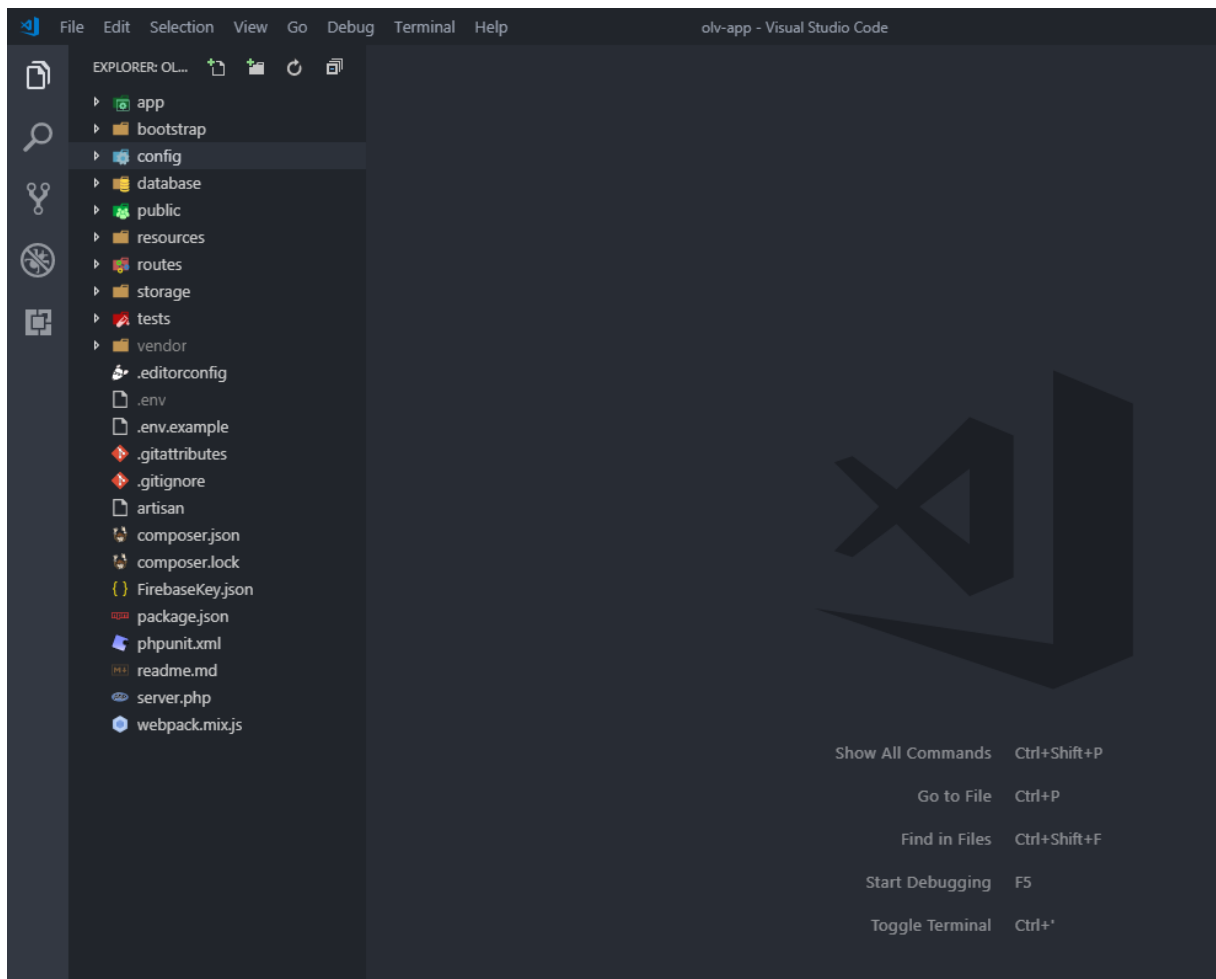
```
MINGW64:/c/xampp/htdocs
Tawan@DESKTOP-3TC0405 MINGW64 /c/xampp/htdocs
$ composer create-project --prefer-dist laravel/laravel olv-app|
```

Fonte: do próprio autor

Após criar a aplicação web com todas as dependências do laravel, foi feito o download do *template* AdminLTE seguindo as configurações disponíveis no site¹⁷. O projeto ficou da seguinte maneira no editor Visual Studio Code como mostra a figura 31.

¹⁷ Disponível em: <<https://adminlte.io/docs/2.4/installation>>

Figura 31: Telas do projeto da aplicação web



Fonte: do próprio autor

Posteriormente foi feita as configurações necessários no arquivo *adminlte.php* que está localizado dentro da pasta *config*. Antes do início do desenvolvimento da aplicação, foi necessário o download do arquivo que faz a conexão com o firebase como mostra a figura 32.

Figura 32: Download do arquivo de configuração

OLVAPP ▾ Ir para a documentação

Configurações

Geral Cloud Messaging Integrações **Contas de serviço** Privacidade dos dados Usuários e permissões

[Gerenciar todas as contas de serviço](#)

SDK Admin do Firebase

Credenciais de legado

Chaves secretas do banco de dados

Admin SDK do Firebase

Sua conta de serviço Firebase pode ser usada para autenticar diversos recursos do Firebase como Database, Storage e Auth de maneira programática pelo Admin SDK unificado. [Saiba mais](#)

Conta de serviço Firebase
firebase-adminsdk-dhte6@olvapp-8fe20.iam.gserviceaccount.com

Snippet de configuração do Admin SDK

Node.js Java Python Go

```
var admin = require("firebase-admin");  
var serviceAccount = require("path/to/serviceAccountKey.json");  
  
admin.initializeApp({  
  credential: admin.credential.cert(serviceAccount),  
  databaseURL: "https://olvapp-8fe20.firebaseio.com"  
});
```

Gerar nova chave privada

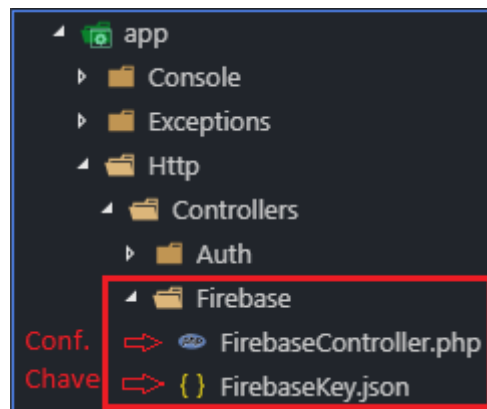
Última chave baixada: 19 de out de 2018 00:15:50

Ativar o Windows
Acesse Configurações para ativar o Windows.

Fonte: do próprio autor

A chave é acrescentada ao projeto como mostra a figura 33, e é criado o arquivo *FirebaseController.php* para fazer o uso da mesma no projeto.

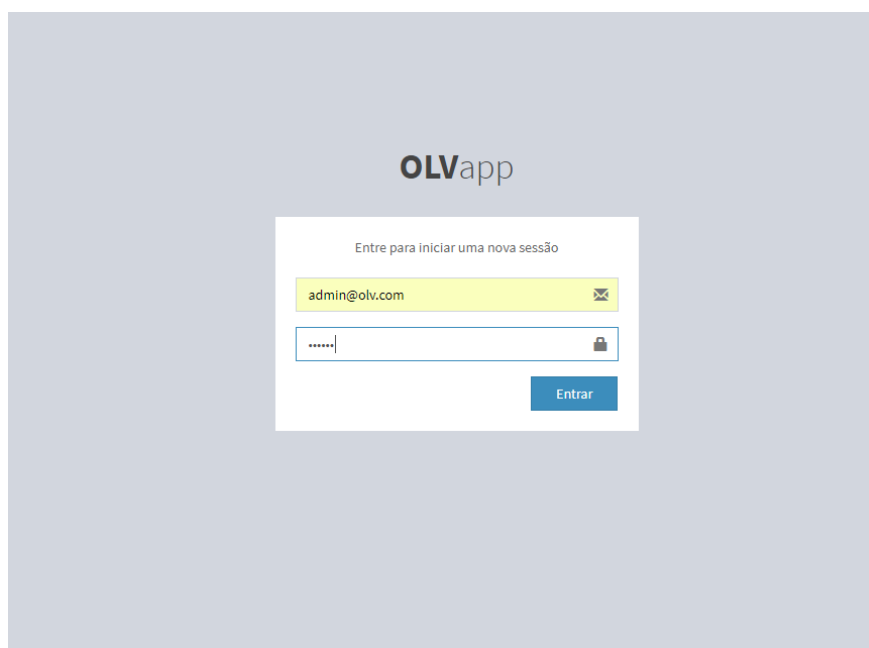
Figura 33: Arquivos de configuração



Fonte: do próprio autor

Feito todas as configurações necessárias para que o projeto web comunique com a base de dados no firebase, deu-se início ao processo de codificação da aplicação o que resultou nas seguintes imagens. A figura 34 mostra a tela de login da aplicação

Figura 34: Tela de login da aplicação web



Fonte: do próprio autor

Após efetuar o login, o administrador tem na tela todos os orçamentos efetuados, tendo a possibilidade de visualiza-los e podendo alterar o status de “Orçamento” para “Pedido” como mostra a figura 35 e 36.

Figura 35: Telas de orçamentos da aplicação web

Header: Sair

Todos os pedidos cadastrados.

Show 10 entries Search:

ID	Nome cliente	Vendedor e-mail	Valor total	Valor do desconto	Metodo de pagamento	Quantidade parcelas	Status	Ação
-LQNEwnDVCWra4JRzBuN	5H Cosmetics Ltda	teste.1@teste.com	1302.0	54	Boleto	3	Orçamento	Visualizar Pedido
-LQNFRX0QIJ3Tbh7-tos	Amanda Silveira Santos	teste.1@teste.com	280.0	80	Cartão	2	Orçamento	Visualizar Pedido

Showing 1 to 2 of 2 entries Previous 1 Next

Fonte: do próprio autor

A tela traz a informações do orçamento como *ID*, *Nome Cliente*, *Vendedor e-mail*, *Valor total*, *Valor do desconto*, *Método de pagamento*, *Quantidade parcelas* e *Status*, além da possibilidade de visualizar o pedido como mostra a figura 36

Figura 36: Visualização do orçamento

Orçamento numero: -LQNEwnDVCWra4JRzBuN

Nome Cliente
5H Cosmetics Ltda

Tipo Pagamento
Boleto

Quantidade de parcelas
3

Produtos

ID	Nome	Marca	Quantidade
16	Condicionador Deep Moisture 300ml	CLAIROL	2
5	Condicionador Deep Moisture Litro	CLAIROL	5
3	Condicionador Moisture 300ml	JOICO	1
13	Condicionador Moisture LITRO	JOICO	3
6	K-Pak Intensy Hydrator 250ml	JOICO	2

Valor do desconto
54

Subtotal
1302,0

VALOR TOTAL
1248,0

Confirma orçamento

Fonte: do próprio autor

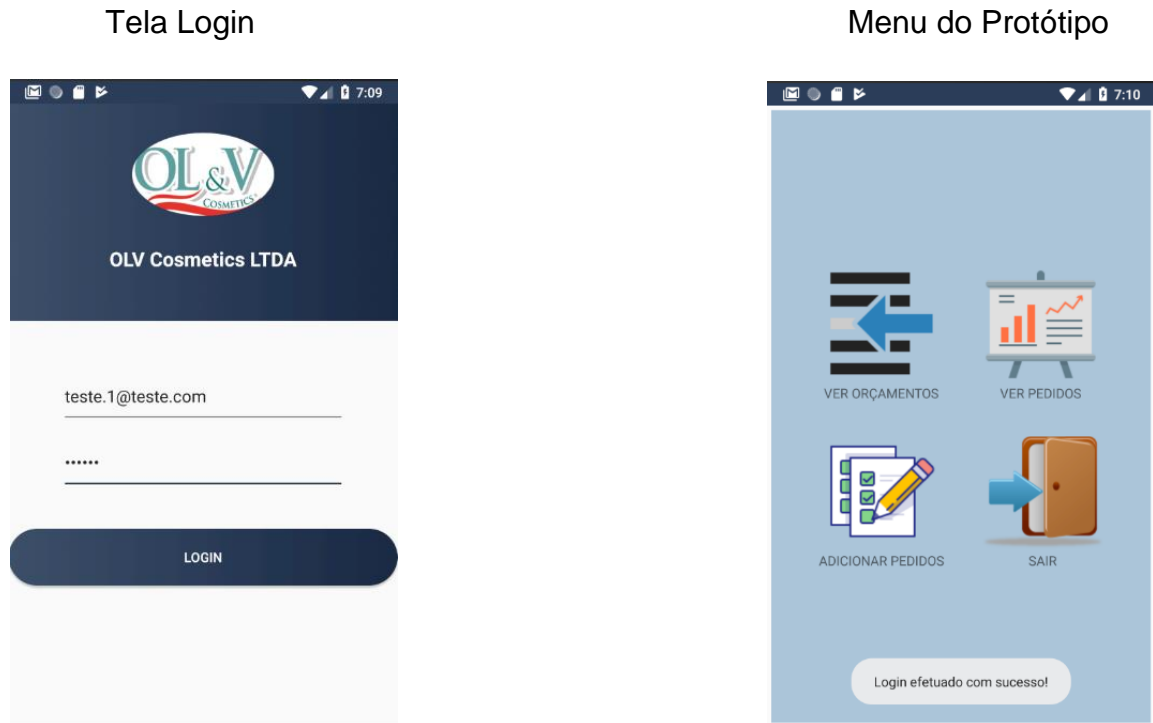
O administrador tem a possibilidade de confirmar o orçamento na tela de visualização, desde que esteja conforme as normas da empresa. Caso haja um confirmação do orçamento, é alterado o status do mesmo no banco de dados atualizando a informação para o protótipo da aplicação mobile e na aplicação web como será tratado no capítulo 4. Caso tenha algo incorreto com o orçamento, a empresa realiza seus devidos procedimentos.

4 TESTES E ANALISE DE TESTES

Esse capítulo tem como propósito mostrar a realização dos testes do protótipo da aplicação e a projeção de ganho que os colaboradores da empresa iram obter. Como os testes serão simulados, os dados utilizados pelo protótipo foram inseridos manualmente no firebase para uma melhor comparação ao banco de dados da empresa. Os dados dos clientes e dos produtos são fictícios, pois são apenas para a realização dos testes.

Dado início aos testes, as informações necessárias para efetuar o login no protótipo são previamente cadastradas no firebase na parte de autenticação onde é feita a verificação de compatibilidade dos dados inseridos para o login na aplicação como mostra a figura 37.

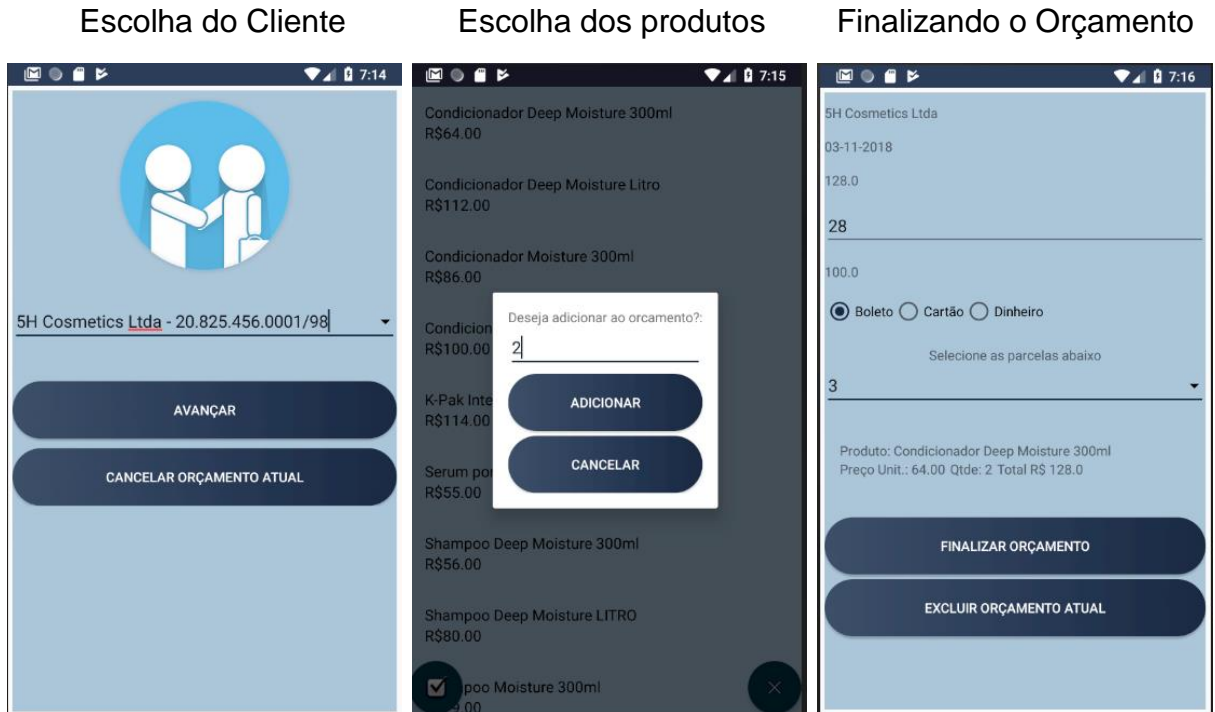
Figura 37: Teste de login



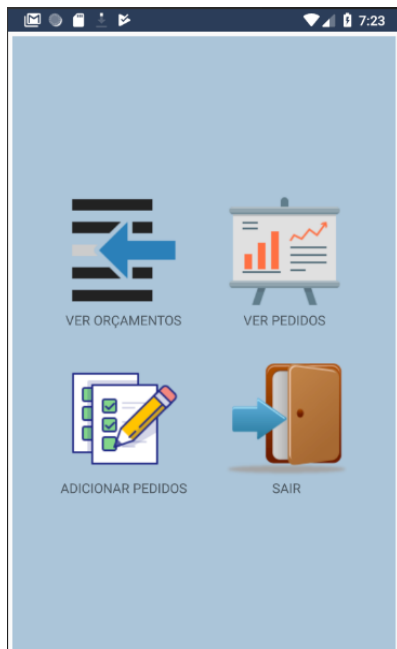
Fonte: do próprio autor

Ao ter acesso ao protótipo, o usuário poderá criar orçamentos como mostra a figura 38, onde se inicia pela escolha do cliente, produtos e finalização como meio de pagamento e desconto concedido

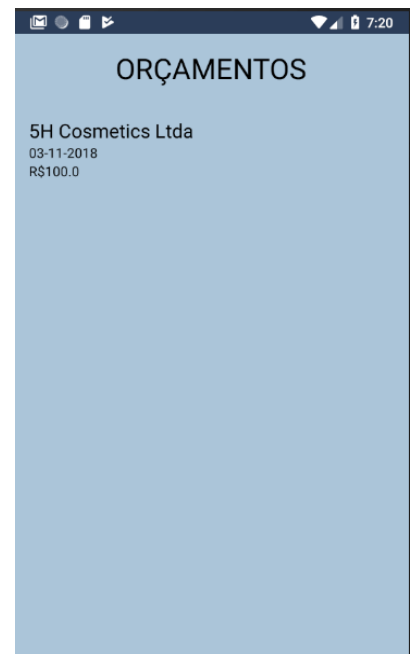
Figura 38: Teste criação de orçamento



Menu



Tela dos Orçamentos criados

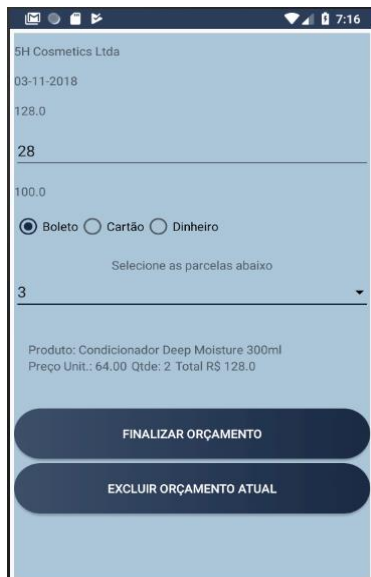


Fonte: do próprio autor

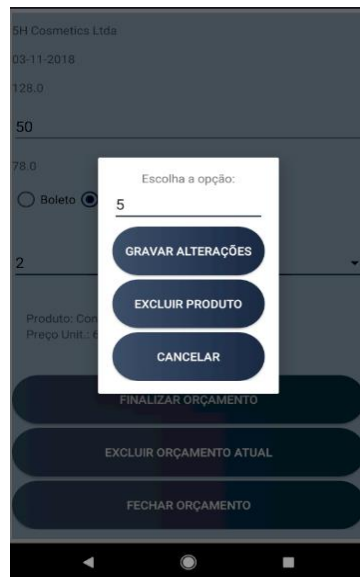
Enquanto o orçamento ainda não foi confirmado pelo administrador como um pedido, o consultor tem a possibilidade de alterar a forma de pagamento, desconto, quantidade dos produtos escolhidos ou até mesmo a exclusão de produtos ou do próprio orçamento como mostra a figura 39.

Figura 39: Teste alteração de orçamento

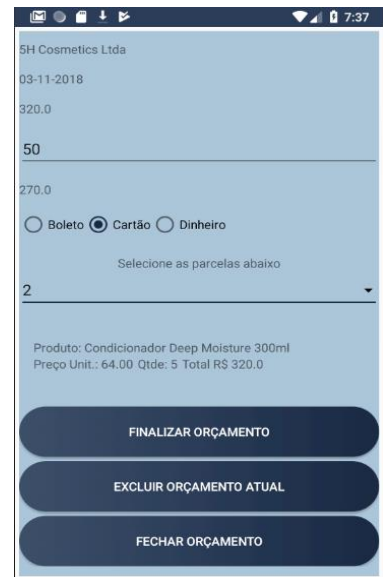
Visualização do Orçamento



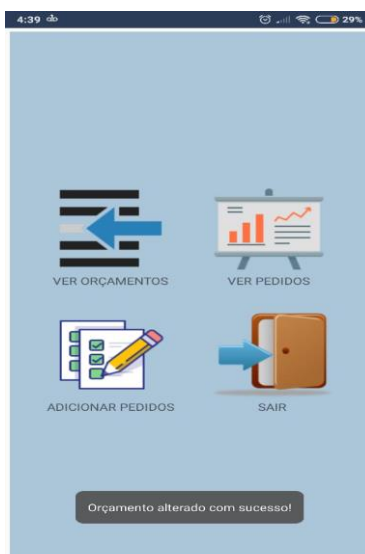
Alteração dos produtos



Confirmando o Orçamento



Menu



Tela dos Orçamentos



Os orçamentos criados podem ser visualizados como mostra a figura 40.

Figura 40: Teste visualização de orçamentos na web

Todos os pedidos cadastrados.

Show entries Search:

ID	Nome cliente	Vendedor e-mail	Valor total	Valor do desconto	Metodo de pagamento	Quantidade parcelas	Status	Ação
- LQNRBT73eIM3xJaBNWO	5H Cosmetics Ltda	teste.1@teste.com	320.0	50	Cartão	2	Orçamento	Visualizar Pedido
ID	Nome Cliente	Vendedor E-mail	Valor total	Valor do desconto	Metodo de Pagamento	Quantidade parcelas	Status	Ação

Showing 1 to 1 of 1 entries Previous **1** Next

Fonte: do próprio autor

Para uma consulta dos produtos solicitados no orçamento, o administrador poderá visualizar de forma mais detalhada como mostra a figura 41, podendo após as devidas verificações confirmar o orçamento.

Figura 41: Teste verificação de orçamentos

Orçamento numero: -LQNRBT73eIM3xJaBNWO

Nome Cliente
SH Cosmetics Ltda

Tipo Pagamento
Cartão

Quantidade de parcelas
2

Produtos

ID	Nome	Marca	Quantidade
16	Condicionador Deep Moisture 300ml	CLAIROL	5

Valor do desconto
50

Subtotal
320.0

VALOR TOTAL
270.0

Confirma orçamento

Fonte: do próprio autor

Logo após a confirmação, o status do orçamento é alterado na base de dados, atualizando assim na aplicação web e no protótipo mobile, o que impossibilita o consultor de vendas efetuar alguma possível alteração como mostra as figuras 42 e 43.

Figura 42: Teste confirmação do orçamento

Todos os pedidos cadastrados.

Show entries Search:

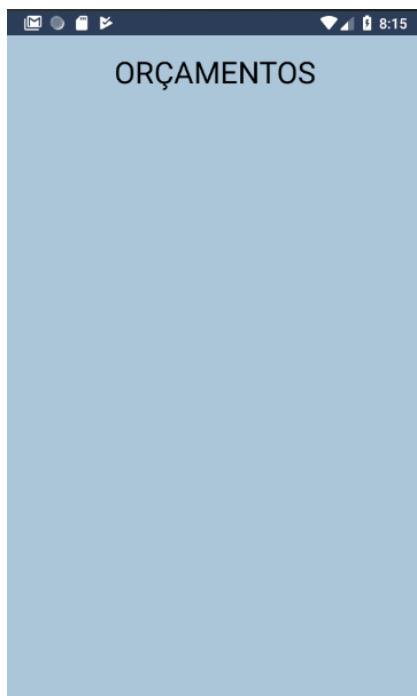
ID	Nome cliente	Vendedor e-mail	Valor total	Valor do desconto	Metodo de pagamento	Quantidade parcelas	Status	Ação
- LQNRBT73eIM3xJaBNWO	5H Cosmetics Ltda	teste.1@teste.com	320.0	50	Cartão	2	Pedido	Visualizar Pedido

Showing 1 to 1 of 1 entries Previous **1** Next

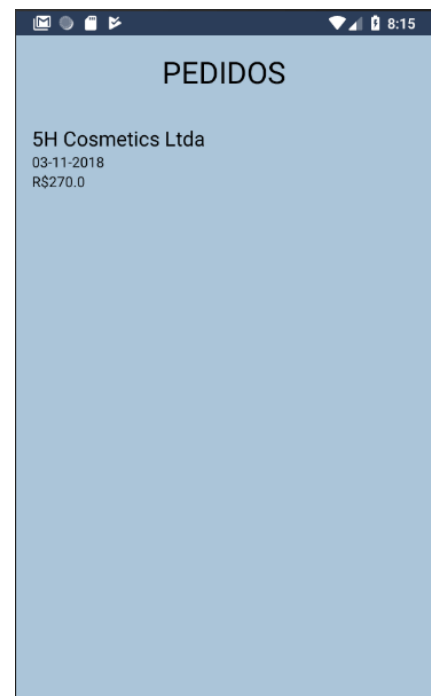
Fonte: do próprio autor

Figura 43: Teste orçamento para pedido

Tela dos Orçamentos



Tela dos Pedidos



Fonte: do próprio autor

Os testes foram realizados juntamente ao orientador da presente monografia, quanto ao uso da aplicação nos testes laboratoriais realizados, o protótipo opera de forma eficiente ao que foi proposto o seu desenvolvimento, com o intuito de trazer melhorias para o processo orçamentário da empresa OLV Cosméticos

Com base nos testes realizados e no contexto atual em que se encontra o processo orçamentário da empresa, foi elaborado dois quadros demonstrativos e comparativos de possíveis ganhos que o protótipo poderá trazer para a mesma.

Quadro 01: Contexto atual da empresa

Característica	Descrição
Processo Manual	Os colaboradores ainda utilizam agendas para atenderem os clientes e anotarem as suas solicitações para posteriormente ser elaborado uma planilha orçamentária.
Retrabalho	Após o de ter o trabalho de anotar em sua agenda todos os produtos solicitados, o colaborador deverá preencher uma planilha eletrônica com os produtos para que depois quando o mesmo tiver acesso à internet, poder enviar para o e-mail da empresa, onde um funcionário recebe e faz a inserção do orçamento ao sistema.
Baixa produtividade	Como o processo ainda é precário, percebe-se que o profissional não produz de acordo com o esperado pelo tempo gasto para elaborar um orçamento e encaminhar a empresa.
Divergências	O processo é feito várias vezes o que acaba dando margem para erros, causando divergências e desentendimentos na empresa.
Definição do processo	Ainda não é definido como é a forma de envio dos orçamentos a empresa, muitos colaborados ainda utilizam o telefone alegando uma maior praticidade no processo.

Fonte: do próprio autor

Quadro 01: Possíveis melhorias com o protótipo

Característica	Descrição
Informatização	Os colaboradores podem elaborar o orçamento no exato momento de atendimento ao cliente, esclarecendo possíveis dúvidas além de enviar o orçamento a empresa no ato.
Único processo e alta disponibilidade	O processo passa a ser único, sem a intervenção de terceiros, além de proporcionar a elaboração de orçamentos mesmo off-line desde que se tenha logado na aplicação previamente com acesso à internet. Logo após retomar com acesso a internet, todos os orçamentos efetuados são instantaneamente enviados para a base de dados.
Alta produtividade	Como o protótipo o consultor de vendas pode atender diversos clientes ao dia pois os orçamentos são enviados instantaneamente ao sistema da empresa. O que possibilita um maior tempo de atendimento visto que não há necessidade de preencher planilhas no final do expediente para encaminhar.
Redução de erros	O processo por ser feito uma única vez, reduz consideravelmente a possibilidade de transcrever um produto erroneamente, visto que os orçamentos são inseridos diretamente no sistema da empresa.
Definição do processo	Como grande parte dos colaboradores alegam que por telefone o processo é mais rápido, um aplicativo traz a praticidade e conforto como ferramenta de trabalho.

Fonte: do próprio autor

O protótipo tem um grande potencial desde que se saiba aplicar no contexto correto, tendo a capacidade de crescimento com implementações futuras devido a sua praticidade e usabilidade.

CONCLUSÃO

O presente trabalho apresentou o desenvolvimento do protótipo de uma aplicação mobile visando melhorias do processo de elaboração de orçamentos de vendas em uma distribuidora de cosméticos da cidade de Teófilo Otoni. Atualmente as empresas investem e tecnologias buscando melhorias em seus processos objetivando o mercado competitivo. Este trabalho procurou responder a seguinte pergunta: De que forma o desenvolvimento de um protótipo de uma aplicação mobile para a criação de orçamentos poderia minimizar os problemas no processo de elaboração de orçamentos de vendas em uma distribuidora de cosméticos da cidade de Teófilo Otoni-MG?

O presente estudo teve como objetivo geral, propor uma otimização do processo de elaboração de orçamentos de vendas de uma distribuidora de cosméticos da cidade de Teófilo Otoni-MG através do desenvolvimento de um protótipo de aplicativo móvel, para que este proposito fosse alcançado, foi necessária a elaboração de objetivos específicos como: Alimentar a aplicação com um banco de dados na nuvem contendo todos as informações necessárias para a elaboração de um orçamento e com sincronização automática pois a aplicação necessita de troca de informações instantâneas com a sua base de dados para que haja uma praticidade na realização do processo orçamentário; Criar mecanismos de monitoramento dos orçamentos realizados identificando os orçamentos aprovados para o consultor visto que o mesmo necessita de um retorno sobre os orçamentos para possíveis questionamentos do cliente; Permitir à distribuidora o acompanhamento dos orçamentos por uma aplicação web, uma vez que a empresa deverá conceder a aprovação dos mesmos para a liberação dos pedidos; Possibilitar a inserção na base

de dados na nuvem pela aplicação *web*, o que permite a empresa cadastros de novos produtos e clientes que serão utilizados pela aplicação *mobile*.

O desenvolvimento da aplicação *web* para o acompanhamento por parte da empresa dos orçamentos elaborados pelos consultores de vendas foi necessário visto que, a empresa ainda não disponibiliza de uma API (*Application Programming Interface*) pronta que fosse capaz de dar suporte à aplicação junto ao sistema da empresa. Dessa maneira, para reproduzir um ambiente simulado do processo orçamentário tratado no estudo foi desenvolvido um painel administrativo para dar suporte o protótipo da aplicação.

De acordo com o tema proposto, pelos testes laboratoriais realizados e análise dos dados obtidos, pode-se dizer que o objetivo apresentado foi alcançado, o protótipo da aplicação trouxe um ganho de praticidade e agilidade para a realização do processo de elaboração de orçamentos de venda. Baseando-se nos dados obtidos dos testes efetuados foi feita a validação e não validação das seguintes hipóteses levantadas durante a elaboração do projeto. A hipótese nula indicava que a aplicação não traria benefícios no processo orçamentário pois necessitaria de aparelhos com grande capacidade de processamento para o seu funcionamento; após análises dos requisitos mínimos necessários para o funcionamento da aplicação a hipótese não foi validada. A hipótese nº 1 afirmava que o uso da aplicação auxiliaria o processo de verificação de orçamentos por parte dos consultores de vendas pois os mesmos teriam acesso a aprovação dos orçamentos efetuados; essa hipótese foi validada através de uma funcionalidade da aplicação que disponibiliza orçamentos autorizados para o consultor. A hipótese nº 2 dizia que a aplicação não otimizaria o processo de elaboração de orçamentos de venda na distribuidora pois os consultores conseguem realizar o seu trabalho de forma ágil e prática sem que seja necessário o uso do aplicativo; essa hipótese não foi validada segundo a comparação elaborada entre o estado atual da empresa e possíveis melhorias que o protótipo poderia trazer a organização. A hipótese nº 3 afirmava que a utilização da aplicação otimizaria o processo de elaboração de orçamentos de venda na distribuidora além de reduzir os erros no processo orçamentário pois daria aos consultores praticidade para realizar as suas tarefas evitando retrabalho; essa hipótese foi validada visto que a aplicação informatizou um processo evitando retrabalho, que antes era feito por meio de caneta e papel deixando-o suscetível a possíveis erros no procedimento.

As empresas que buscam crescimento na qualidade e produtividade,

necessitam de ferramentas que otimizam as suas atividades. Sendo assim, precisam disponibilizar aos seus colaboradores recursos para tal realização. A conclusão do trabalho resultou no desenvolvimento de um protótipo para a elaboração de orçamento de vendas em uma distribuidora de cosméticos da cidade de Teófilo Otoni. O protótipo visa mostrar o ganho de praticidade e agilidade que a ferramenta pode trazer para a empresa, sendo utilizada como utensílio capaz de minimizar erros auxiliando os colaboradores na realização de suas tarefas. A medida que as necessidades da empresa forem surgindo, o protótipo pode vir a se tornar uma aplicação complexa e com várias funcionalidades.

Por fim, o presente projeto proporcionou ganho na formação do desenvolvedor como um maior aprofundamento sobre o desenvolvimento mobile, programação orientada a objetos e a obtenção de um melhor conhecimento sobre a importância da tecnologia no ambiente empresarial. Do ponto de vista social apresentou uma solução para um problema que afeta algumas empresas, podendo ser aplicada com as devidas alterações. O ganho pessoal vem com o conhecimento adquirido sobre a realização do projeto, que servirá como base para uma futura especialização na área em questão.

REFERÊNCIAS

ARRUDA NONATO, Marcos Daniel. *World Wide Web: WWW*. Edição do Autor, 2015. 27p. Disponível em: < <https://www.livrosdigitais.org.br/livro/1447A68L1BQ3T> > Acesso em: 04 Set. 2018

BENTO, Edvaldo Junior. *PHP e MySQL. Desenvolvimento Web*. 1. Ed. São Paulo: Casa do Código, 2013.

BORGES, Luiz Eduardo. *Python para desenvolvedores*. 1. Ed. Rio de Janeiro: Edição do Autor, 2009.

CAYRES, Paulo Henrique. *Modelagem de Banco de Dados*. 1. Ed. Rio de Janeiro: Escola Superior de Redes, 2015.

CLARO, Daniela Barreiro; SOBRAL, Joao Bosco Manguieira. *Programação em Java*. Florianópolis: Pearson Education, 2008. 89p. Disponível em: < <https://www.passeidireto.com/arquivo/6624067/programacao-em-java---daniela-barreiro-claro-e-joao-bosco-manguieira-sobral> >. Acesso em: 15 Mai. 2018.

CORDEIRO, Felipe. *Android Aprendiz: Um guia para iniciantes Crie seu primeiro aplicativo Android*. 3. Ed. São Paulo: Android Pro, 2017. 52 p.

DOBAY, Eduardo S. *Programação em C. Guia de Programação em C*. São Paulo, p. 03-05, ago/2012. Pdf. Disponível em: <<http://fig.if.usp.br/~esdobay/>>. Acesso 14 Mai 2018.

ELMASRI, Ramez; NAVATHE, Shamkant B. *Sistemas de Banco de Dados*. 4ª edição. São Paulo: Pearson Addison-Wesley, 2005.

FIREBASE. Site Oficial. Disponível em: < <https://firebase.google.com/> >. Acesso em: 16 Set. 2018.

FOROUZAN, Behrouz A.; MOSHARRAF, Firouz. *Redes de Computadores. Uma abordagem top-down*. Tradução Técnicas de Marcos A. Simplicio Jr. e Charles Christian Miers. 1ª Ed. Porto Alegre: AMGH Editora Ltda, 2013.

GABARDO, Ademir C. *Laravel para ninjas 1*. Ed. São Paulo: Novatec Editora Ltda, 2017.

GONÇALVES, Eduardo. *PL/SQL: Domine a linguagem do banco de dados Oracle*. Casa do Código, 2015. 431p. Disponível em: < <https://pd.eboek.win/download/pl-sql-br1070314463.html> >. Acesso em: 16 Mai. 2018.

GONÇALVES, Eduardo. *SQL: Uma abordagem de dados Oracle*. Casa do Código, 2014. 357p. Disponível em: < http://galdino.sytes.net:8080/04_Documents/Casa%20do%20Codigo/SQL%20-%20Uma%20abordagem%20para%20bancos%20de%20dados%20Oracle.pdf >. Acesso em: 16 Mai. 2018.

LANE, Kin. *Overview Of The Backend as a Service (BaaS) Space Prepared*. 2013. Disponível em: <<http://www.integrover.com/wp-content/uploads/2014/11/api-evangelist-baas-whitepaper.pdf>>. Acesso em: 16 Set 2018.

LECHETA, Ricardo R. *Google Android: Aprenda a criar aplicações para dispositivos móveis com o Android SDK*. 5. Ed. São Paulo: Novatec, 2015. 1067 p.

LOPES, Sergio. *Aplicações mobile híbridas com Cordova e PhoneGap*. 1. Ed. São Paulo: Casa do Código, 2016.

MARCELO, Antônio. *Apache. Configurando o servidor web para Linux 3*. Ed. São Paulo: Brasport Livros e Multimídia Ltda, 2005.

NEVES, Pedro M C. *O guia prático da HTML*. 1ª Ed. 2004. 19p. Disponível em: <<http://www.centroatl.pt/titulos/tecnologias/imagens/excerto-e-book-ca-oguiapracicodahtml.pdf> > Acesso em: 05 Set. 2018.

PAULA FILHO, Wilson de Pádua. *Engenharia de Software: fundamentos, métodos e padrões*. 1.Ed. São Paulo: LTC, 2000.

PERES, Rita. *SQL VS NOSQL*. 2018. Disponível em: < <https://www.revista-programar.info/artigos/sql-vs-nosql/>>. Acesso em: 16 Mai. 2018.

SANTOS, Rafael. *Introdução à programação orientada a objetos usando Java*. 2. Ed. Rio de Janeiro: Elsevier, 2013.

SOMMERVILLE, Ian. *Engenharia de Software*. 1. Ed. São Paulo: Pearson Education do Brasil, 2004.

TOLEDO, Jan Miszura; DE DEUS, Gilcimar Divino. Desenvolvimento em Smartphones. *Aplicativos Nativos e Web*. Departamento de Computação – Pontifícia Universidade Católica de Goiás, 2012. Disponível em: <<http://www.cpgls.pucgoias.edu.br/7mostra/Artigos/AGRARIAS%20EXATAS%20E%20DA%20TERRA/Desenvolvimento%20em%20Smartphones%20-%20Aplicativos%20Nativos%20e%20Web.pdf>> Acesso em: 16 Set. 2018.

VIEIRA, Marcos Rodrigues et al. *Banco de Dados NoSQL: Conceitos, Ferramentas, Linguagens e Estudos de Casos no Contexto de Big Data*. Disponível em: < http://data.ime.usp.br/sbbd2012/artigos/pdfs/sbbd_min_01.pdf > Acesso em: 16 Set. 2018.

VISUAL STUDIO. Site Oficial. Disponível em: < <https://code.visualstudio.com/docs> > Acesso em: 05 Set. 2018.

WELLING, Luke; THOMSON, Laura. PHP e MySQL. *Desenvolvimento Web*. Tradução de Docware Traduções Técnicas de Adriana Kramer. 3ª Ed. Rio de Janeiro: Campus/Elsevier, 2005.