|  |  |
| --- | --- |
|  | **Pontifícia Universidade Católica de Minas Gerais**  **Desenvolvimento de Aplicativos para Dispositivos Móveis**  **Disciplina: Computação em Nuvem** |

**Relatório Técnico  
Trabalho Final**

Carlos Teles

Flávio Teixeira

Pedro Novaes

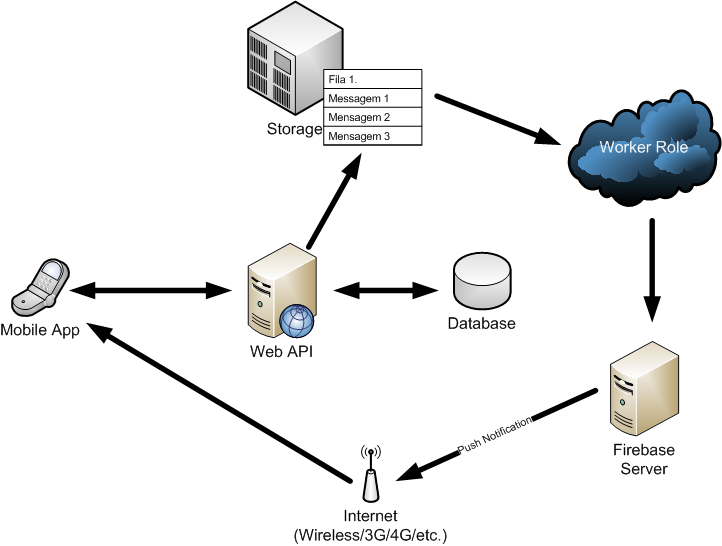
Belo Horizonte

07/2017

# Introdução

Este Relatório tem como objetivo apresentar os requisitos técnicos utilizados para construção do aplicativo móvel, API e Arquitetura do trabalho prático final “Assistente Virtual”.

# Arquitetura



## Descrição

O Aplicativo foi desenvolvido na plataforma *Android Studio,* o mesmo simula a aquisição de artefatos/armas através de um *chat* com uma atendente virtual.

O Aplicativo possui conexão com uma API, onde a mesma envia as características dos itens que estão à venda, como nome, valor, prazo de entrega.

A API recebe as informações do item adquirido e insere o item numa fila para processamento.

A *Worker Role* retira o item da fila, simula o processamento e em seguida envia uma mensagem para o Servidor do Firebase.

O Firabase recebe a requisição e envia um *Push Notification* para todos os participantes do aplicativo avisando-os da aquisição do integrante da equipe.

# API

## Link

A API foi hospedada no Microsoft Azure, sendo o endereço da mesma:

<http://webapiposgraduacao.azurewebsites.net>

## Testes

A API foi construída baseada em três classes Usuarios, Produtos, UsuariosProdutos, sendo assim descritas:

a) Usuarios:

* idusuario;
* equipe;
* nome;
* senha;

A chamada a URL <http://webapiposgraduacao.azurewebsites.net/api/usuarios> retorna os usuários cadastrados no sistema.

b) Produtos:

* idproduto;
* nome;
* prazoentrega;
* preco;

A chamada a URL <http://webapiposgraduacao.azurewebsites.net/api/produtos> retorna os produtos cadastrados no sistema.

a) UsuariosProdutos:

* id;
* idusuario;
* idproduto;
* datacompra;

A chamada a URL <http://webapiposgraduacao.azurewebsites.net/api/usuariosprodutos> retorna a lista de produtos adquiridos pelos usuários.

Para consultar os demais métodos você pode visitar a URL <http://webapiposgraduacao.azurewebsites.net/Help>

# Hospedagem

A API foi hospeda no Azure, utilizando-se do Serviço de Aplicativos, sendo o *deploy* realizado diretamente da IDE Visual Studio 2017.

<http://webapiposgraduacao.azurewebsites.net>

A Worker Role foi hospedada também no Azure, utilizando-se do Serviço de Nuvem, sendo o *deploy* também realizado diretamente da IDE Visual Studio 2017.

http://apicloudserviceapplication.cloudapp.net/

# Escalonamento Horizontal

O gargalo do sistema é o processamento das mensagens (na fila), ou seja, o tempo que o sistema leva para processar os pedidos. A *Work Roler* foi configurada para escalonamento automático baseando-se na quantidade de mensagem que existem na fila. Caso a quantidade exceda 1000 (mil) uma nova instância é criada. A verificação é a cada minuto.