



UNIVERSIDADE EM TRANSFORMAÇÃO: INTEGRALIZANDO SABERES E EXPERIÊNCIAS

2 A 6 DE SETEMBRO/2019



Marque a opção do tipo de trabalho que está inscrevendo:

() Resumo () Relato de Experiência () Relato de Caso

CROSS-PLATFORM MESSAGING: UM ESTUDO SOBRE FERRAMENTA PARA SOLUÇÃO DE MENSAGENS ENTRE PLATAFORMAS

AUTOR PRINCIPAL: Marcelo Felipe Guarani Fernandes

CO-AUTORES: Profa. Dra. Anubis Graciela de Moraes Rossetto

ORIENTADOR: Prof. Me. Élder Francisco Fontana Bernardi

UNIVERSIDADE: Instituto Federal de Educação, Ciência e Tecnologia Sul-rio-grandense, Câmpus Passo Fundo.

INTRODUÇÃO

O Firebase Cloud Messaging (FCM) é uma solução de mensagens *cross-platform*, que permite enviar notificações sem nenhum custo, (Firebase, 2019). O termo *cross-platform*, fundamenta em implementar aplicações que funcionem em inúmeras plataformas, tais como: IOS nativo, Android, Desktop e Progressive Web Apps (PWA). Para pleno funcionamento do serviço, é necessário que haja conexão via *wi-fi* ou dados móveis, pois é por meio dela que o FCM notificará ao usuário.

O objetivo deste trabalho é desenvolver uma maneira de que a aplicação não perca sua funcionalidade em um cenário onde não haja conexão com a rede de dados. A hipótese proposta é que caso a aplicação esteja *off-line*, uma SMS (Short Message Service), seja enviada com o conteúdo da notificação. Essa solução se faz necessária em um cenário onde é vital que a aplicação seja funcional, mesmo sem conexão com a Internet.

Para isso, é explorado o desenvolvimento de uma aplicação híbrida, produzida em Ionic Framework na versão 3, que consiste em um kit de ferramentas de interface do usuário, para desenvolvimento de aplicações *desktop e mobile* utilizando linguagens para implantação *web (HTML, CSS e JavaScript)*, e a estrutura do *app* será construída em Angular, uma plataforma escrita em *TypeScript*, que implementa as funções principais para execução do serviço.



UNIVERSIDADE EM TRANSFORMAÇÃO: INTEGRALIZANDO SABERES E EXPERIÊNCIAS

2 A 6 DE SETEMBRO/2019



DESENVOLVIMENTO:

Existem dois componentes principais para envio e recebimento de notificações, o *Cloud Functions*, onde o desenvolvedor pode criar funções que são acionadas por produtos do próprio Firebase; ou uma aplicação cliente IOS, *Android* ou *Web* para recebimento das mensagens.

A primeira tarefa é obter o *token* de permissão do FCM para um determinado dispositivo. É preciso lidar com isso de maneira diferente em cada plataforma. Em especial, o *Android* exige a solicitação de permissão antes de tudo. Essa função retorna uma *promise*, então utilizamos *async/await*. Ou seja, deve-se criar uma função *async*, e retornar em uma variável *await* o retorno desta função.

Um usuário pode ter vários dispositivos registrados para notificações, porém é preciso de um documento que preserve o relacionamento entre o usuário e o dispositivo. Além disso, um *token* é apenas uma string, portanto, pode ser usado como o *ID* do documento para garantir que cada *token* tenha apenas um documento.

Notificações *push* são projetadas para funcionar somente quando o aplicativo está rodando em segundo plano; elas são recebidas diretamente no dispositivo do usuário e a exibição do seu conteúdo e as ações são gerenciadas pelo sistema operacional de cada plataforma.

Pensando em um cenário onde não haja conexão com a rede de dados; notificações via FCM se tornam impossíveis pois não há como acessar as *Cloud Functions*. Cabe destacar, que em alguns casos é necessário que a aplicação continue operando de forma ativa mesmo sem acesso a Internet. Tal aspecto comumente não é levado em consideração em aplicações projetadas para auxiliar na segurança física do usuário, onde existe, por exemplo, um “botão do pânico” que notifica aos monitores de determinado usuário seu pedido de ajuda, apenas quando há conexão com a rede de dados. Entretanto, pode ser necessário fazer uso deste recurso quando não há conectividade, e para isso podemos utilizar o envio de SMS de cada dispositivo como alternativa, a fim de driblar essa falha.

CONSIDERAÇÕES FINAIS:

Após efetuar testes da aplicação, o FCM de fato enviou uma mensagem de notificação ao dispositivo à partir da nuvem como mostra a Figura 1. Porém no modo *off-line*, quando se faz necessário o envio da mesma mensagem via SMS, não se obteve sucesso pois o sistema operacional *Android* não deu permissão, conforme mostrado na Figura 2. Segundo documentação do Google, tal falha ocorreu pela falta de registro da aplicação na plataforma de aplicações da (Google, 2019). Tal procedimento será realizado ao concluir-se o desenvolvimento da aplicação onde o mecanismo testado



UNIVERSIDADE EM TRANSFORMAÇÃO: INTEGRALIZANDO SABERES E EXPERIÊNCIAS

2 A 6 DE SETEMBRO/2019



será implantado. Assim feito o registro, poder-se-á aferir se de fato o procedimento foi bem-sucedido, embora todas as evidências indiquem que sim.

REFERÊNCIAS

ANGULAR, Google. **Architecture overview.** Disponível em: <<https://angular.io/guide/architecture>>. Acesso em: 01 abr. 2019.

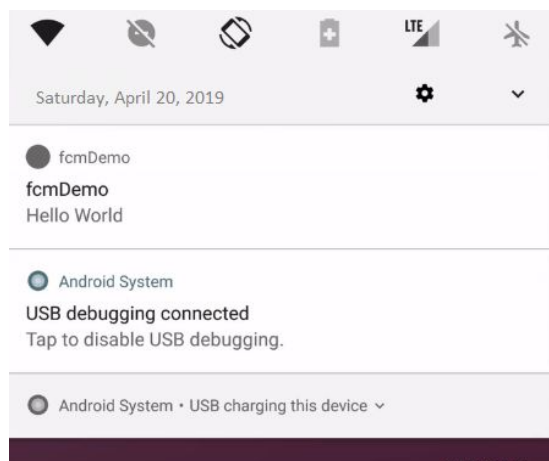
IONIC FRAMEWORK, Documentation. **Ionic Native.** Disponível em: <<https://ionicframework.com/docs/v3/native/>>. Acesso em: 03 abr. 2019.

IONIC FRAMEWORK, Documentation. **What is Ionic Framework?** Disponível em: <<https://ionicframework.com/docs/intro>>. Acesso em: 01 abr. 2019.

FIREBASE, Google. **Firestore Cloud Messaging.** Disponível em: <<https://firebase.google.com/docs/firestore/?hl=pt-br>>. Acesso em: 10 abr. 2019.

GOOGLE, Google Play. **Declarar permissões do seu app.** Disponível em: <<https://support.google.com/googleplay/android-developer/answer/9214102>>. Acesso em: 12 abr. 2019.

ANEXOS

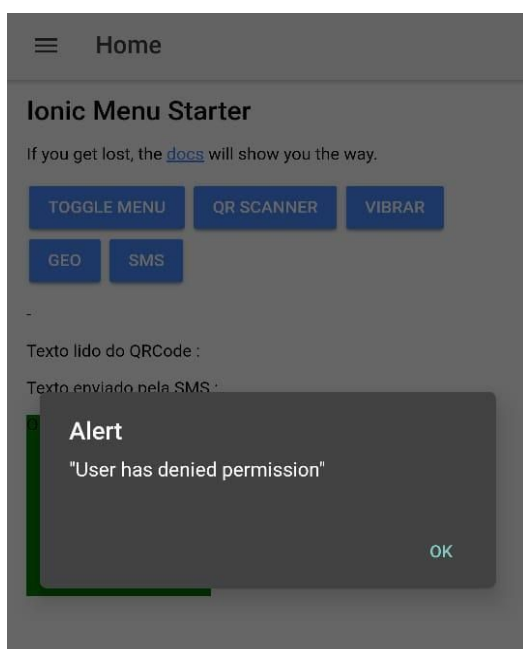


(Figura 1. Fonte própria)



UNIVERSIDADE EM TRANSFORMAÇÃO: INTEGRALIZANDO SABERES E EXPERIÊNCIAS

2 A 6 DE SETEMBRO/2019



(Figura 2. Fonte própria)