

[Descrição](#)

[Público-Alvo/Intended User](#)

[Funcionalidades/Features](#)

[Protótipo de Interfaces do Usuário](#)

[Main Activity](#)

[Detail Activity](#)

[Main Activity \(sw600dp-land\)](#)

[Considerações Chave/Key Considerations](#)

[Persistência de Dados](#)

[Bibliotecas adicionais](#)

[Outras considerações](#)

[Próximos Passos: Tarefas Necessárias](#)

[Tarefa 1: Configuração do Projeto/Project Setup](#)

[Tarefa 2: Implementar o backend no GCE](#)

[Tarefa 3: Implementar aplicativo Android](#)

[Tarefa 4: Configurar teste integrado - aplicativo e backend](#)

Usuário do GitHub: guskuma

# Notifique-me

## Descrição

Informe sua equipe de forma fácil e centralizada através de envio de notificações em tempo real. Notificações com conteúdo rico que permite interação através de ações rápidas configuráveis (ligação, envio de mensagem, abertura de links, etc)

## Público-Alvo/Intended User

Principalmente mantenedores de sistemas que necessitam de informações disparadas de forma centralizada e automatizada, com conteúdo rico e suporte a ações rápidas.

Servirá para notificar:

- Informações (sobre servidores ou serviços) - Ação rápida: abrir link
- Relatórios (sobre funcionalidades) - Ação rápida: compartilhar o relatório
- Problemas na execução de batch-jobs. Ação rápida: ligar para o responsável

## Funcionalidades/Features

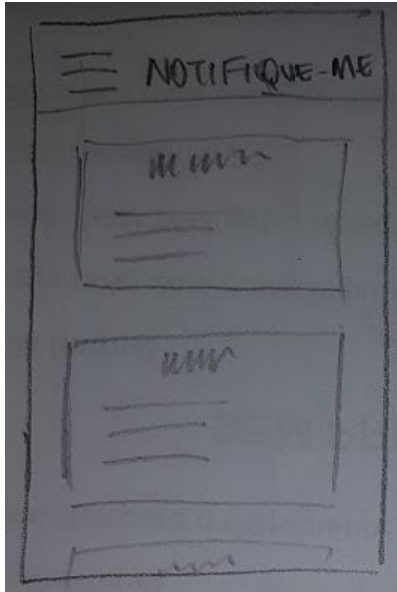
- Enviar de notificações através de API
- Receber notificações com conteúdo rico
- Realizar ações rápidas que foram configuradas na própria notificação

O aplicativo será composto de um backend no GCE (repasse, via FCM, de mensagens) e um aplicativo Android para recebimento e interpretação das mensagens, transformando-as em notificações de forma coerente para cada tipo de mensagem e ação disponível.

## Protótipo de Interfaces do Usuário

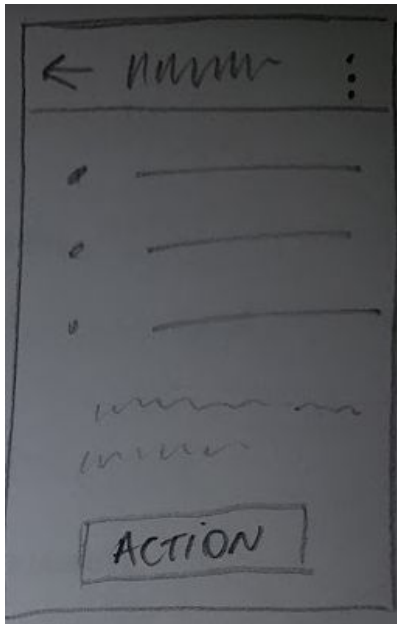
Elas podem ser feitas a mão (tire uma foto dos seus desenhos e os insira neste fluxo), ou usando um programa como o Photoshop ou Balsamiq.

## Main Activity



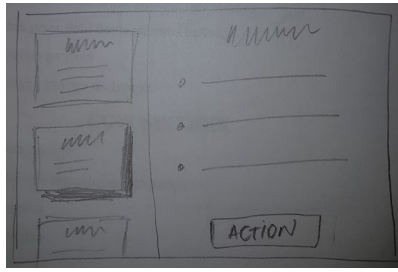
Tela principal do sistema que contém a lista (RecyclerView) de notificações em formato de Cards. Ao clicar em algum dos Cards a Activity de detalhe é apresentada.

## Detail Activity



Esta tela apresenta os detalhes da notificação selecionada pelo usuário na tela principal. Aqui é possível ver todo o conteúdo da notificação e também acionar o botão de ação (FloatingActionButton).

## Main Activity (sw600dp-land)



Quando for o caso de um tablet em landscape, a lista e o detalhe da notificação selecionada serão apresentados ao mesmo tempo, lado a lado.

## Considerações Chave/Key Considerations

### Persistência de Dados

O aplicativo fará a persistência de dados através de uso da biblioteca DBFlow para facilitar a interação com banco SQLite. Não haverá ContentProvider já que não haverá necessidade de exposição de dados para aplicativos externos.

### Bibliotecas adicionais

Serão usadas as bibliotecas ButterKnife para injeção de *Views*, Parceler para agilizar a implementação da interface *Parcelable*, o Retrofit para chamadas *REST* mais robustas com menos código, o Timber para facilitar o *debugging* da aplicação e também o Google Play Services - Mobile Ads para apresentação das propagandas na versão gratuita.

### Outras considerações

O aplicativo será composto de poucas *Activities* mas a ideia é que nas próximas iterações (depois do curso) mais tipos de mensagens e, portanto, mais tipos de visualizações (*Fragments*) sejam incluídos no aplicativo - conforme novas necessidades forem surgindo para minha equipe de desenvolvimento.

## Próximos Passos: Tarefas Necessárias

### **Tarefa 1: Configuração do Projeto/Project Setup**

1. Criação do projeto do aplicativo Android
2. Configuração do FCM
3. Criação do backend no GCE
4. Configuração de versões paga/gratuita
5. Configuração do Gradle
6. Submissão do projeto inicial para o GitHub

### **Tarefa 2: Implementar o backend no GCE**

1. Criação de biblioteca Java que conterá os tipos de mensagem e parâmetros possíveis nas mensagens enviadas do backend para o app. Esta biblioteca será compartilhada entre backend e app - que se comunicarão no padrão JSON
2. Implementar o backend para receber os parâmetros e informações da mensagem e repassar, via FCM, para o aplicativo

### **Tarefa 3: Implementar aplicativo Android**

1. Implementar a camada de persistência para mensagens recebidas
2. Implementar a UI MainActivity, que deverá listar as mensagens já recebidas;
1. Implementar a UI dos Fragments, que apresentarão os diferentes tipos de mensagens;
2. Implementar a UI das notificações, que apresentarão os diferentes tipos de mensagens e suas ações;
3. Criar layouts de variações (para tablets, landscape, etc)
4. Implementar as propagandas do Google Ad Mobile

### **Tarefa 4: Configurar teste integrado - aplicativo e backend**

1. Implementar o teste integrado para subir o backend e o aplicativo e enviar mensagens