Sistema de Monitoramento de Localização Android

Granith - Documentação Técnica

Versão 1.0

Equipe de Desenvolvimento

22 de agosto de 2025

Conteúdo

1 Introdução

Este documento apresenta a documentação técnica do Sistema de Monitoramento de Localização Android da aplicação Granith. O sistema é projetado para monitorar a localização de funcionários em tempo real, utilizando tecnologia de geofences para detectar entrada e saída de áreas específicas.

1.1 Objetivo

O sistema tem como objetivo principal:

- Monitorar a localização de funcionários em tempo real
- Detectar entrada e saída de geofences (áreas geográficas delimitadas)
- Sincronizar dados offline com Firebase Firestore
- Fornecer controle administrativo por empresa
- Garantir funcionamento em background de forma eficiente

1.2 Tecnologias Utilizadas

- **Android SDK**: Plataforma de desenvolvimento
- **Firebase Firestore**: Banco de dados NoSQL em nuvem
- **Google Play Services**: Serviços de localização
- **FusedLocationProviderClient**: Cliente para obtenção de localização
- **Java**: Linguagem de programação principal

2 Arquitetura do Sistema

2.1 Visão Geral

O sistema é composto por três atividades principais e um serviço de foreground que trabalham em conjunto para fornecer monitoramento contínuo de localização:

1. **InputActivity**: Tela de autenticação e seleção de usuário
2. **MainActivity**: Tela principal com verificação de permissões
3. **LocationForegroundService**: Serviço principal de monitoramento

2.2 Diagrama de Componentes

Camada	Componente	Responsabilidade
UI	InputActivity	Autenticação de usuários
UI	MainActivity	Verificação de permissões
Service	LocationForegroundService	Monitoramento de localização
Data	Firebase Firestore	Armazenamento de dados
Local	SharedPreferences	Cache local

3 Componentes Principais

3.1 LocationForegroundService

O `LocationForegroundService` é o componente central do sistema, responsável pelo monitoramento contínuo de localização e detecção de geofences.

3.1.1 Características Principais

- **Execução em Foreground**: Mantém o monitoramento ativo mesmo com o app em background
- **Otimização de Bateria**: Ajusta a precisão baseada na proximidade com geofences
- **Sincronização Offline**: Armazena eventos localmente quando sem conexão
- **Multi-empresa**: Suporte a múltiplas empresas com dados isolados

3.1.2 Estados de Precisão

O serviço implementa três estados de precisão para otimizar o consumo de bateria:

Estado	Condição	Configuração
HIGH	Próximo a geofence	Intervalo: 60s, Prioridade: HIGH_ACCURACY
LOW	Dentro de geofence ou distante	Intervalo: 120s, Prioridade: BALANCED_POWER
VERY_LOW	Muito distante de todas	Intervalo: 120s, Prioridade: LOW_POWER

3.1.3 Constantes de Configuração

```
1 private static final long SYNC_INTERVAL_MS = 5 * 60 * 1000; // 5 minutos
2 private static final long STALE_CHECK_INTERVAL = 30 * 60 * 1000; // 30
    minutos
3 private static final long DUPLICATE_WINDOW_MS = 2 * 60 * 1000; // 2
    minutos
4 private static final long AUTO_EXIT_THRESHOLD_MS = 24 * 60 * 60 * 1000;
    // 24 horas
5 private static final long EVENT_CLEANUP_AGE_MS = 7 * 24 * 60 * 60 *
    1000; // 7 dias
```

Listing 1: Constantes do LocationForegroundService

3.2 Sistema de Confirmação de Eventos

O sistema implementa um mecanismo robusto para confirmação de eventos de entrada/-saída:

3.2.1 Parâmetros de Confirmação

```
private static final class ConfirmationParams {
    static final int REQUIRED_UPDATES_ONLINE = 3;
    static final long MIN_INTERVAL_ONLINE_MS = 2 * 60 * 1000; // 2
    minutos
    static final int REQUIRED_UPDATES_OFFLINE = 3;
    static final long MIN_INTERVAL_OFFLINE_MS = 2 * 60 * 1000; // 2
    minutos
}
```

Listing 2: Parâmetros de Confirmação

3.2.2 Fluxo de Confirmação

1. **Detecção**: Sistema detecta possível entrada/saída
2. **Contagem**: Incrementa contador de atualizações
3. **Verificação**: Verifica se atingiu número mínimo de confirmações
4. **Tempo**: Valida se passou tempo mínimo entre primeira e última detecção
5. **Confirmação**: Gera evento confirmado se critérios atendidos

3.3 InputActivity

A `InputActivity` gerencia a autenticação e seleção de usuários com suporte multi-empresa.

3.3.1 Funcionalidades

- **AutoComplete**: Campo com sugestões automáticas de funcionários
- **Multi-empresa**: Busca funcionários de todas as empresas usando collectionGroup
- **Aprovação em Tempo Real**: Monitora aprovações administrativas
- **Sistema de Sessões**: Cria sessões de login rastreáveis

3.3.2 Fluxo de Autenticação

1. Usuário digita nome (com autocomplete)
2. Sistema verifica existência do funcionário
3. Verifica aprovação administrativa na empresa
4. Se não aprovado, cria solicitação de acesso
5. Monitora aprovação em tempo real
6. Configura empresa no CompanyService
7. Navega para MainActivity

3.4 MainActivity

A `MainActivity` é responsável pela verificação de permissões e inicialização do sistema.

3.4.1 Permissões Verificadas

- **ACCESS_FINE_LOCATION**: Localização precisa
- **ACCESS_COARSE_LOCATION**: Localização aproximada
- **ACCESS_BACKGROUND_LOCATION**: Localização em background (Android 10+)
- **CAMERA**: Para escaneamento de QR codes
- **POST_NOTIFICATIONS**: Notificações (Android 13+)

3.4.2 Otimizações de Sistema

- **Battery Optimization**: Solicita desabilitação para o app
- **Auto-start**: Configura reinicialização automática
- **Background Restrictions**: Verifica restrições de background

4 Estrutura de Dados

4.1 Coleções Firebase

4.1.1 Estrutura Multi-empresa

O sistema utiliza uma estrutura hierárquica no Firebase:

```
companies/
  {companyId}/
    employees/
      {employeeId}
    geofences/
      {geofenceId}
    geofence_events/
      {eventId}
    system_events/
      {eventId}
    permission_requests/
      {requestId}
    sessions/
      {sessionId}
```

Listing 3: Estrutura de Coleções Firebase

4.2 Modelos de Dados

4.2.1 GeofenceData

```
public class GeofenceData {
    private String name;
    private double latitude;
    private double longitude;
    private float radius;
    private String codigoObra;
    private boolean active;
}
```

Listing 4: Modelo GeofenceData

4.2.2 Evento de Geofence

```
Map<String, Object> geofenceRecord = new HashMap<>();
geofenceRecord.put("employeeId", employeeId);
geofenceRecord.put("employeeName", userName);
geofenceRecord.put("geofenceName", geofence.getName());
geofenceRecord.put("eventType", "entry" | "exit" | "auto_exit");
geofenceRecord.put("latitude", location.getLatitude());
geofenceRecord.put("longitude", location.getLongitude());
geofenceRecord.put("timestamp", System.currentTimeMillis());
geofenceRecord.put("isOfflineSync", false);
```

Listing 5: Estrutura de Evento

5 Funcionalidades Especiais

5.1 Modo Offline

O sistema mantém funcionalidade completa mesmo sem conexão à internet:

5.1.1 Armazenamento Local

- **Eventos de Geofence**: Armazenados em JSON no SharedPreferences
- **Eventos de GPS**: Status de GPS armazenado separadamente
- **Cache de Geofences**: Dados de geofences salvos localmente

5.1.2 Sincronização

- **Automática**: Sincroniza a cada 5 minutos quando online
- **Detecção de Duplicatas**: Evita eventos duplicados na sincronização
- **Limpeza**: Remove eventos com mais de 7 dias automaticamente

5.2 Monitoramento de Sistema

5.2.1 Eventos de GPS

O sistema monitora mudanças no status do GPS:

```
private void recordGpsStatusEvent(String message, boolean gpsEnabled) {
    Map<String, Object> gpsStatusEvent = new HashMap<>();
    gpsStatusEvent.put("eventType", gpsEnabled ? "gps_enabled" : "
    gps_disabled");
    gpsStatusEvent.put("message", message);
    gpsStatusEvent.put("timestamp", System.currentTimeMillis());
    // ...
}
```

Listing 6: Monitoramento GPS

5.2.2 Detecção de Bateria

- **Estado da Bateria**: Monitora níveis baixos e críticos
- **Modo de Economia**: Detecta ativação do modo de economia de energia
- **Adaptação**: Ajusta comportamento baseado no estado da bateria

5.3 Sistema de Saída Automática

Para evitar funcionários "presos"em geofences:

- **Threshold**: 24 horas sem detecção de saída
- **Verificação**: A cada 30 minutos verifica entradas obsoletas
- **Ação**: Gera evento de "Saída Automática"
- **Localização Fictícia**: Usa coordenadas da geofence para o evento

6 Considerações de Performance

6.1 Otimizações Implementadas

6.1.1 Precisão Adaptativa

- **Proximidade**: Aumenta precisão quando próximo a geofences
- **Distância**: Reduz precisão quando distante de todas as geofences
- **Interior**: Mantém precisão baixa quando dentro de geofences

6.1.2 Gestão de Recursos

- **WakeLock Parcial**: Mantém CPU ativo apenas quando necessário
- **Handlers Otimizados**: Usa handlers com timeouts apropriados
- **Cleanup Automático**: Remove dados antigos periodicamente

6.2 Monitoramento de Estado

6.2.1 BroadcastReceivers

```
// Monitoramento de conectividade
private class NetworkReceiver extends BroadcastReceiver

// Monitoramento de GPS
private class GpsStatusReceiver extends BroadcastReceiver

// Monitoramento de bateria e dispositivo
private class DeviceStateReceiver extends BroadcastReceiver
```

Listing 7: Receivers Implementados

7 Instalação e Configuração

7.1 Dependências

7.1.1 Firebase

```
implementation 'com.google.firebase:firebase-firestore'
implementation 'com.google.firebase:firebase-auth'
```

Listing 8: Dependências Firebase

7.1.2 Google Play Services

```
implementation 'com.google.android.gms:play-services-location'
implementation 'com.google.android.gms:play-services-maps'
```

Listing 9: Dependências Google Play

7.2 Permissões no Manifest

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"
    />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION
    " />
<uses-permission android:name="android.permission.
    ACCESS_BACKGROUND_LOCATION" />
<uses-permission android:name="android.permission.FOREGROUND_SERVICE" />
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.POST_NOTIFICATIONS" />
```

Listing 10: Permissões Necessárias

8 Troubleshooting

8.1 Problemas Comuns

8.1.1 Serviço Não Inicia

- Verificar se todas as permissões foram concedidas
- Confirmar que otimização de bateria está desabilitada
- Verificar se Firebase está configurado corretamente

8.1.2 Eventos Não Sincronizam

- Verificar conectividade de internet
- Confirmar configuração do Firebase
- Verificar logs para erros de autenticação

8.1.3 Alta Consumo de Bateria

- Verificar se precisão adaptativa está funcionando
- Confirmar que geofences estão sendo carregadas corretamente
- Verificar se WakeLock está sendo liberado adequadamente

9 Conclusão

Este sistema de monitoramento de localização oferece uma solução robusta e escalável para empresas que necessitam rastrear funcionários em tempo real. Com suporte offline, otimizações de bateria e estrutura multi-empresa, o sistema atende às necessidades tanto de pequenas quanto grandes organizações.

9.1 Próximos Passos

- Implementação de relatórios analíticos
- Adição de notificações push para administradores
- Integração com sistemas de RH existentes
- Desenvolvimento de dashboard web para administração