Documentação Técnica e Compartilhamento de Conhecimento
Coleta de Dados da Rede Móvel com Aplicativo React Native

Davi Rocha

24 de julho de 2025

1 Objetivo

Este documento tem como objetivo transmitir o conhecimento adquirido no processo de desenvolvimento de um aplicativo que coleta dados da rede móvel de forma passiva e otimizada, com foco na preservação da experiência do usuário. A proposta é servir como material base para desenvolvedores, pesquisadores ou engenheiros que desejem entender, reproduzir ou evoluir essa solução.

2 Resumo do Projeto

O trabalho consiste no desenvolvimento de um aplicativo em **React Native CLI** com código nativo em **Kotlin**, que realiza coleta periódica de dados como intensidade e qualidade do sinal da rede móvel (`RSRP`, `RSRQ`, `CellID`), além da **localização GPS**. A coleta ocorre a cada 15 minutos, em background, e pode ser exportada para um servidor remoto sem consumo de dados pelo usuário.

3 Tecnologias Utilizadas

- **React Native CLI** — interface do aplicativo e integração com bibliotecas.
- **Kotlin / Android SDK** — coleta nativa de dados de rede e execução em segundo plano.
- **WorkManager** — execução periódica resistente a reinicializações do sistema.
- **TelephonyManager & PhoneStateListener** — APIs para extrair dados da rede celular.
  `https://stackoverflow.com/questions/19805880/get-signal-strength-in-android`
  `https://www.youtube.com/watch?v=hojwvo99584`

- **@react-native-community/geolocation** — biblioteca para localização GPS. `https://github.com/react-native-geolocation/react-native-geolocation`
- **TrafficStats API** — para análise de throughput (volume de dados). `https://developer.android.com/reference/android/net/TrafficStats`
- **AWS EC2 + S3** — infraestrutura para receber os dados coletados.

4 Etapas de Desenvolvimento

4.1 1. Estudo e Escopo Inicial

O projeto teve início com o estudo da tese de Daniel, que já havia realizado uma coleta massiva de dados da rede móvel. Com base nessa análise, decidiu-se focar primeiramente na entrega de um protótipo funcional do app, postergando a modelagem matemática para uma fase posterior.

4.2 2. Criação do App de Testes

Um aplicativo em `React Native CLI` foi iniciado, na mesma versão usada pelo app Algar Varejo. Isso garantiu compatibilidade com o sistema alvo e permitiu testes mais realistas.

4.3 3. Coleta de Dados com TelephonyManager

Foi realizada a integração com a API `TelephonyManager`, que permite o acesso aos dados da rede celular do Android. A coleta incluiu:

- **RSRP** (potência do sinal recebido);
- **RSRQ** (qualidade do sinal recebido);
- **Cell ID** (identificador da torre conectada).

O acesso a essas informações exige permissões como `ACCESS_FINE_LOCATION` e `READ_PHONE_STATE`.

4.4 4. Latitude e Longitude

A geolocalização foi obtida usando a biblioteca `@react-native-community/geolocation`. Para isso, também foi necessário lidar com permissões e falhas comuns de dispositivos Android que bloqueiam a localização em segundo plano.

4.5 5. Execução em Background com WorkManager

Inicialmente, tentou-se usar bibliotecas JS para tarefas em segundo plano. No entanto, essas soluções não garantiam execução confiável com o app fechado ou após reinício do sistema. Assim, optou-se pelo uso do **WorkManager** nativo em Kotlin, que:

- Executa tarefas mesmo com o app fechado;
- Suporta regras de energia e conectividade;
- Foi configurado para rodar a cada 15 minutos com persistência.

4.6 6. Salvamento Local dos Dados

Os dados coletados são salvos em arquivos locais (`.txt` ou `.json`) usando `react-native-fs`. Isso permite inspeção local e exportação posterior para servidores.

4.7 7. Transmissão sem Consumo de Dados — APN + EC2

Para evitar consumo do pacote de dados do usuário, estudou-se o funcionamento da **APN (Access Point Name)**, que é a configuração do ponto de acesso à internet da operadora. A ideia foi:

- Criar uma exceção na APN para não tarifar conexões com um IP fixo;
- Substituir o bucket S3 por uma instância `EC2` com IP fixo público;
- Configurar os dispositivos com uma APN personalizada com essa exceção.

Essa abordagem exige testes com a equipe da operadora (Algar) e validação com os responsáveis do app oficial.

4.8 8. Próximos Passos: Modelagem e Throughput

Com os dados em mãos, o próximo passo será:

- Aplicar modelos como **Processos Gaussianos** e **Redes Neurais** para estimar a cobertura da rede na cidade;
- Estudar formas de medir **throughput** (taxa de transmissão de dados), usando a API `TrafficStats.getMobileRxBytes()` e `getMobileTxBytes()` para obter os bytes transmitidos e recebidos;
- Executar testes apenas em horários estratégicos (ex: durante a madrugada e com o celular carregando), visando economia de bateria e dados.

5 Desafios e Decisões Técnicas

- **SINR foi descartado** — embora inicialmente considerado, foi descartado por orientação técnica de especialista, dado seu baixo valor prático na modelagem inicial.
- **Coleta passiva e energética** — decisões sempre priorizaram baixo consumo e não interferência na experiência do usuário.
- **Integração nativa foi essencial** — soluções apenas em JS se mostraram instáveis em segundo plano.

6 Referências

- **PhoneStateListener (YouTube)**: https://www.youtube.com/watch?v=hojwvo99584
- **TelephonyManager (StackOverflow)**: https://stackoverflow.com/questions/19805880/get-signal-strength-in-android
- **Geolocalização (React Native)**: https://github.com/react-native-geolocation/react-native-geolocation
- **Throughput (Android TrafficStats)**: https://developer.android.com/reference/android/net/TrafficStats
- **AWS EC2/S3**: https://aws.amazon.com/pt/documentation/
- **WorkManager**: https://developer.android.com/topic/libraries/architecture/workmanager

7 Conclusão

Este documento registra, em detalhe, a construção de um sistema capaz de coletar dados da rede móvel de forma passiva e eficiente, visando aplicações futuras em modelagem de cobertura e diagnóstico de qualidade. A abordagem combina tecnologias híbridas (React Native e Kotlin), boas práticas de eficiência energética, e integração com infraestrutura em nuvem.

O conteúdo aqui apresentado pode ser facilmente atualizado e evoluído, tornando-se uma base viva para novas contribuições e pesquisas.