### **Relatório Técnico 1. Configuração do Projeto Android**

* O projeto foi criado no Android Studio.
* Foi definido o pacote básico como com.example.cepapp.
* O SDK mínimo e as dependências necessárias foram configuradas, como bibliotecas de rede.

### **2. Integração com a API ViaCEP**

* A API ViaCEP foi escolhida para buscar informações de endereço baseadas no CEP.
* **Retrofit** foi provavelmente utilizado para fazer chamadas RESTful, devido à sua popularidade e simplicidade de uso em projetos Android. A biblioteca facilita a configuração de URLs e o parsing de dados JSON.

### **3. Estrutura de Pacotes**

* Criados pacotes para **atividades** (activities), **redes** (networking), e **modelos** (models), que organizam o código de forma modular.
* **activities:** Contém as telas e controladores de interface, como a Activity principal para capturar o CEP do usuário.
* **networking:** Responsável pela configuração do Retrofit ou Volley, incluindo a interface de serviço API que contém o método de requisição HTTP.
* **models:** Define as classes de dados que representam o retorno da API, como objetos Endereco com campos como logradouro, bairro, localidade e UF.

### **4. Lógica de Requisição e Manipulação de Dados**

* A Activity principal captura o CEP inserido pelo usuário e, ao pressionar um botão, envia o CEP para a API.
* A requisição é feita via método GET para a API de ViaCEP. Por exemplo: https://viacep.com.br/ws/{cep}/json/.
* O Retrofit (ou Volley) processa a resposta da API, convertendo o JSON retornado em um objeto Java.

### **5. Interface de Usuário (UI)**

* A UI foi criada usando **XML layouts**. Elementos principais incluem:
  + Um campo de entrada (EditText) para o CEP.
  + Um botão (Button) para disparar a busca.
  + TextViews ou CardViews para exibir o logradouro, bairro, cidade, estado.
* Ao receber uma resposta da API, a tela exibe os dados formatados, ou uma mensagem de erro caso o CEP não seja válido.

### **6. Tratamento de Erros**

* Implementação de tratamento de exceções para gerenciar casos como CEPs inválidos ou falhas de rede.
* Exibição de mensagens de erro claras para o usuário, como "CEP não encontrado" ou "Erro de conexão".

### **7. Testes e Validação**

* Testes foram realizados com diferentes CEPs para garantir a exatidão dos dados e o bom funcionamento da API.
* O aplicativo foi validado para verificar a estabilidade e a correta exibição das informações recebidas da API.