

# Questionário Desenvolvimento APP

## 1- O que é e como funciona uma Activity?

Uma Activity é uma classe Java que representa uma tela de aplicativo Android. Ela oferece diversos métodos que permitem que você monitore e controle as ações do usuário em seu aplicativo. Todo aplicativo Android é construído a partir de várias atividades que interagem entre si para criar a experiência desejada. Cada Activity possui sua própria tela com a qual o usuário pode interagir. Ao criar uma Activity, você fornece a lógica necessária para controlar como os usuários interagem com a tela e como ela se comporta.

## 2- Como funciona o ciclo de vida de uma Activity?

O ciclo de vida de uma Activity é dividido em oito passos:

1. `onCreate()`: Esta atividade é chamada quando a Activity é criada pela primeira vez. Aqui, você inicializará os recursos essenciais que a Activity precisa para iniciar.
2. `onStart()`: Esta atividade é chamada quando a Activity se torna visível para o usuário. Aqui, você inicializará os recursos que precisam ser exibidos na tela.
3. `onResume()`: Esta atividade é chamada quando a Activity está pronta para receber a entrada do usuário. Aqui, você inicializará os recursos que precisam para receber a entrada do usuário.
4. `onPause()`: Esta atividade é chamada quando a Activity está prestes a ser encerrada. Aqui, você salvará quaisquer dados que não possam ser restaurados quando a Activity é recriada.
5. `onStop()`: Esta atividade é chamada quando a Activity não é mais visível para o usuário. Aqui, você liberará qualquer recurso que não seja mais necessário.
6. `onDestroy()`: Esta atividade é chamada quando a Activity está sendo destruída. Aqui, você liberará todos os recursos que foram inicializados para a Activity.
7. `onRestart()`: Esta atividade é chamada quando a Activity está sendo reiniciada depois de ser parada. Aqui, você restaurará os dados que foram salvos na etapa `onPause`.
8. `onSaveInstanceState()`: Esta atividade é chamada quando a Activity está prestes a ser destruída. Aqui, você salvará qualquer estado da Activity que possa ser restaurado quando ela é recriada.

## 3- O que é e qual a função do Adapter e ViewHolder?

**Adapter** é uma classe que faz a ligação entre os dados a serem exibidos na lista e a exibição da lista. Ela é responsável por inserir os dados nos elementos da lista, como texto, imagens e outros. **ViewHolder** é uma classe que contém as referências para os elementos de exibição da lista. Ele ajuda a otimizar o desempenho, pois armazena as referências dos elementos da lista quando eles são criados para que possam ser reutilizados. Isso evita que os elementos sejam recriados quando o usuário rola a lista. O padrão Adapter é usado para permitir que diferentes classes e APIs sejam usadas juntas, mesmo que não sejam originariamente projetadas para isso, ele faz isso traduzindo as interfaces entre duas classes de modo que elas sejam compatíveis e possam se comunicar entre si, sem alterar os códigos das classes originais.

**ViewHolder** é uma classe auxiliar que permite armazenar os elementos de interface de usuário dentro de uma lista. O ViewHolder foi criado para melhorar o desempenho do aplicativo, pois permite que os elementos de interface de usuário sejam armazenados em memória para serem reutilizados cada vez que o usuário rola a lista. Isso reduz a quantidade de recursos necessários para atualizar a lista, pois os elementos de interface de usuário já estão prontos.

O ViewHolder é um padrão usado para guardar referências a objetos visuais dentro de uma lista ou tabela, de modo a minimizar o número de objetos necessários para recuperar objetos visuais a partir de um adaptador. Os ViewHolders são usados para minimizar as chamadas de sistema de recuperação de objetos visuais.

#### **4- O que é o Retrofit e por que utilizar ao invés de criar as chamadas manualmente?**

Retrofit é uma biblioteca para Android desenvolvida pela Square, que facilita a chamada de serviços de APIs REST.

Ele é usado para tornar as solicitações de rede mais fáceis, rápidas e simples. Com o Retrofit, você pode definir uma interface com anotações que descrevem as solicitações HTTP. Isso facilita a criação e a manutenção de solicitações, além de facilitar o teste do seu código. O Retrofit também possui suporte integrado a diversos formatos de dados, como JSON, XML etc., o que facilita muito a análise da resposta de uma solicitação. Ele também suporta cache de solicitação e resposta, o que ajuda no desempenho. O Retrofit é uma ótima alternativa pois facilita a conexão a um serviço da Web, traduzindo a API em interfaces Java. Ele pode ser usado para obter e enviar dados para serviços da Web e para fazer upload e download de arquivos.

Com ela é possível criar chamadas HTTP de forma simples, rápida e eficiente, evitando a necessidade de codificar toda a lógica de requisição manualmente. Além de Por ser uma biblioteca simples e objetiva, é uma excelente opção para quem deseja implementar serviços de APIs REST em seu projeto.