

Questões de Teste Para Desenvolvedor Android

Candidato: Adonai Filipe da Silva Pontes

E-mail: adonaipontes@gmail.com

1. O que é um Fragment ?

R: Fragment é um componente versátil excelente para desenvolvimento de UI dinâmicas e complexas, com Fragment podemos criar pedaços UI com comportamento distinto e reutilizar esses pedaços de UI em diversas Activitys, esse comportamento possibilita o reaproveitamento e centralização de código em UI modulares onde vários fragmentem podem compor uma Activity para montar uma UI.

2. Quais as principais diferenças entre uma Activity e um Fragment? Quando usar um ou outro?

R: A diferença fundamental é que um Fragment necessita de uma Activity para ser apresentado, e mesmo possuindo um ciclo de vida próprio, o ciclo de vida de um Fragment é diretamente impactado pelo ciclo de vida da Activity em que está inserido. O processamento individual de cada ciclo de vida de um Fragment pode ser realizado desde que a Activity em que está inserido esteja no ciclo de vida onResume().

Outra diferença notável é o reaproveitamento de código que pode ser obtido na utilização de Fragments, quebrando a UI em diversos pedaços com interface e comportamentos distintos que podem ser utilizados em diversas Activitys. Tal abordagem possibilita o desenvolvimento de interfaces modulares onde podemos possuir vários Fragments funcionando dentro de uma mesma Activity.

Para o desenvolvimento de interfaces dinâmicas e complexas onde elementos de interface que serão desenvolvidos podem ser reaproveitados posteriormente no projeto o uso de Fragment é o ideal. Já em casos de fluxo único, onde o que será desenvolvido será utilizado apenas em uma única UI, o uso somente de Activity é uma boa pratica.

3. Qual é o ciclo de vida de uma activity?

R: uma Activity possui 7 ciclos de vida sendo eles:

onCreate() – Executado quando uma Activity é criada. Geralmente é o método responsável por carregar os layouts (XML) e outras operações de inicialização. Só é executado 1 vez durante o Clico de Vida da Activity.

onStart() – É chamado logo depois do onCreate() ou quando a Activity que estava em background volta a ter foco. (Depois dela temos onResume, caso a Activity esteja em primeiro plano ou onStop se ela não for visível).

onResume() – Método chamado quando a Activity vai começar a interagir com o usuário (ou retoma foco). Sempre vem depois do onStart, caso a Activity tenha sido iniciada ou se ela estava parada

(onStop). Pode ocorrer direto (vindo depois do onPause) caso a Activity que não estava em primeiro plano, mas ainda estava visível, volte para primeiro plano.

onPause() – Ocorre logo antes da Activity perder o foco, ou seja, quando ir para o Background, mas não foi destruída. Geralmente é usada para parar animações e recursos que estejam consumindo processamento e persistir dados não salvos. Desta forma, se (por falta de recursos) a Activity for destruída, não perderemos essas informações.

onStop() – Ocorre assim que a Activity deixa de aparecer totalmente. Pode ser seguida por onRestart caso a Activity volte para primeiro plano ou por onDestroy caso ela seja destruída pelo usuário.

onRestart() – É chamado quando uma Activity que estava parada volta ao foco. Bem antes do onStart.

onDestroy() – Esse método ocorre logo antes da Activity ser destruída/finalizada.

4. Se você tivesse uma tela no seu APK com problemas de lentidão ao carregar, o que você faria para resolver o problema?

R: Se possuir acesso ao código fonte do aplicativo, realizaria a execução e depuração do mesmo analisando cuidadosamente os avisos e erros em tempo de execução. Analisaria também todas as chamadas de Activitys, Fragments, Classes, Métodos, Serviços e etc. verificando seu retorno e tempo de resposta. Dessa forma conseguiria realizar um diagnóstico preciso identificando o motivo da lentidão. Por fim, poderia me concentrar na resolução do problema.