Questão 01:

**Src** – É onde ficara todas as classes .java (Exemplo: HelloWorld.java);

**Gen** – É onde ficará as classes .java geradas pelo projeto (Exemplo: R.java), se alterar manualmente alguma dessas classes a IDE para de gerar elas automaticamente;

**Assests** – Nesta pasta você adicionara todos os arquivos não compiláveis vinculados ao projeto;

**Bin** – São os arquivos resultantes do seu projeto quando compilado.

**Libs** – É nessas pasta que você adiciona as bibliotecas adicionais do seu projeto.

**Res** – Contém os arquivos de recursos associados ao seu projeto.

Questão 02:

A classe R está contida no diretório gen, esta classe é responsável por fazer a comunicação entre os arquivos .xml e os arquivos .java. Sem essa classe não seria possível buscar os valores de variáveis estáticas no arquivo string.xml por exemplo.

Questão 03:

O arquivo AndroidManifest.xml é o arquivo de configuração central do Android, alguns elementos validos que podem conter são: <activity>, <application>, <meta-data>, <permission>, <permission-group>, <uses-configuration>, <uses-feature>, <uses-library>, <uses-permission>, <uses-sdk>.

Questão 04:

Questão 05:

O sistema operacional android possui basicamente 5 componentes principais:

• Applications: Os aplicativos estão no topo da pirâmide da arquitetura do sistema operacional android, o componente é composto pelo conjunto de aplicações nativas do mesmo. Dentre as aplicações pode–se citar: cliente de e-mail, despertador, calendário, jogos, mapas, browser etc.

• Application Framework: O componente de framework nativo disponibiliza aos desenvolvedores as mesmas Applications Programming Interface (APIs) utilizadas para a criação de aplicações originais do sistema operacional android, permitindo que o programador tenha o mesmo acesso ao sistema que os aplicativos da camada de aplicativos possuem. Este framework foi criado para abstrair a complexidade e simplificar a reutilização de procedimentos. Esse componente funciona como um link para as bibliotecas do sistema operacional que serão acessadas através de APIs contidas no framework.

• Libraries: Essas bibliotecas são responsáveis por fornecer funcionalidades para manipular o áudio, vídeo, gráficos, banco de dados e browser. Algumas bibliotecas são a Bionic, a OpenGL/ES para trabalhar com interface gráfica, e a SQLite para trabalhar com banco de dados.

• Android Runtime: Permite que cada thread rode sua própria instância da MV (máquina virtual). Embora no desenvolvimento de aplicativos seja utilizada a linguagem Java, as aplicações não são executadas em uma máquina virtual Java tradicional, e sim em outra chama de Dalvik especialmente otimizada para dispositivos móveis. A Dalvik foi construída pelos engenheiros da Google, para obter um consumo mínimo de memória e isolamento de processos.

• Kernel Linux: O componente do Kernel é baseada em um Kernel do sistema operacional Linux, mas muitas modificações foram feitas para otimizar memória e tempo de processamento das aplicações. Essas modificações incluem novos dispositivos de drivers, adições no sistema de gerenciamento de energia e um sistema que possibilita terminar processos de maneira criteriosa quando há pouca memória disponível. O componente atua também como responsável pela abstração entre o hardware e os aplicativos e é responsável pelos sérvios principais do sistema operacional android, como o gerenciamento de memória e de processos.