Atividade 02 - Resumo da estrutura de um projeto android

Manifesto:

Em resumo, o arquivo AndroidManifest.xml desempenha um papel central na definição do comportamento, das permissões e das características de um aplicativo Android. Ele é essencial para o funcionamento adequado e a interação com o sistema operacional Android.
As funções do manifesto são:

- Declarar os componentes do aplicativo, como atividades, serviços e receptores de transmissão.
-
- Especificar as permissões necessárias para acessar recursos e informações do dispositivo.
-
- Configurar informações importantes, como o nome do pacote, versão e classe principal da atividade.
-
- Definir como o aplicativo responde a ações de outros aplicativos, usando filtros de intenções.
-
- Gerenciar dependências de bibliotecas externas.
-
- Personalizar a aparência do aplicativo, incluindo ícone e título.
-
- Adicionar configurações de segurança, como permissões personalizadas e políticas de segurança de rede.

Código Fonte:

O código-fonte é o esqueleto de um aplicativo Android. Depois de escrito o código fonte passa por um processamento junto dos outros recursos e vira o aplicativo em si, que é a lista de tarefas que o dispositivo vai executar.

Recursos:

Recursos em um projeto Android são ativos e dados usados pelo aplicativo, como imagens, layouts, strings de texto, cores, arquivos de áudio, instruções de animação, entre outros. Eles são armazenados na pasta "res" e são fundamentais para definir a aparência, o comportamento e a funcionalidade do aplicativo. Recursos permitem a internacionalização, personalização e organização eficaz do projeto, facilitando o desenvolvimento de aplicativos Android bem projetados e consistentes.

Layouts: Os layouts XML definem a estrutura e a aparência da interface do usuário (UI) do aplicativo. Eles especificam como os elementos da UI, como botões, campos de texto e imagens, são organizados na tela.

Imagens e ícones: As imagens e ícones são recursos gráficos usados para elementos visuais do aplicativo, como ícones de aplicativos, imagens de fundo e ícones de botões.

Strings: Strings de texto são armazenadas em arquivos XML de recursos de strings. Isso permite que o texto do aplicativo seja facilmente traduzido para diferentes idiomas e mantido em um único local.

Valores dimensionais: Valores dimensionais definem tamanhos, margens, preenchimentos e outras medidas que são usadas para dimensionar elementos na UI.

Cores: Recursos de cores definem cores que são usadas em toda a interface do usuário do aplicativo, permitindo uma consistência visual.

Arquivos de áudio e vídeo: Arquivos de áudio e vídeo são recursos usados para reproduzir mídia, como músicas ou vídeos dentro do aplicativo.

Arquivos de animação: Os arquivos de animação definem animações personalizadas que podem ser aplicadas a elementos da UI ou outros objetos do aplicativo.

Arquivos de layout de menu: Esses arquivos definem a estrutura dos menus do aplicativo, incluindo itens de menu e grupos de itens.

Arquivos de valor: Esses arquivos podem ser usados para armazenar valores constantes, como valores booleanos, inteiros ou arrays, que podem ser referenciados em várias partes do código do aplicativo.

Arquivos de configuração: Os arquivos de configuração, como o AndroidManifest.xml, armazenam informações essenciais sobre o aplicativo, como permissões, atividades principais e configurações de inicialização.

Recursos de fonte: Você pode incluir fontes personalizadas que serão usadas para estilizar o texto em seu aplicativo.

Gradle:

O Gradle é uma ferramenta de automação de compilação e gerenciamento de dependências usada em projetos Android. Ele automatiza tarefas de compilação, gerencia bibliotecas externas, oferece flexibilidade de configuração, suporta construções incrementais e é altamente integrado com IDEs como o Android Studio. O Gradle usa arquivos de script de build (geralmente "build.gradle") para definir as configurações do projeto e é essencial para criar aplicativos Android de forma eficiente e personalizada.

build.gradle (Projeto): Define configurações globais para o projeto, como a versão do Gradle e os repositórios de classe Gradle.

build.gradle (Módulo): Contém configurações específicas do módulo, incluindo dependências, plugins, tarefas personalizadas e configurações de compilação.

settings.gradle: Define a estrutura do projeto, incluindo quais módulos estão incluídos no projeto e sua relação hierárquica.

gradle.properties: Armazena propriedades e variáveis personalizadas usadas no projeto, como chaves de API ou configurações específicas do ambiente.

buildSrc (diretório): Este diretório permite criar um módulo especial chamado "buildSrc" para compartilhar código de build personalizado e tarefas entre os scripts de build do Gradle.