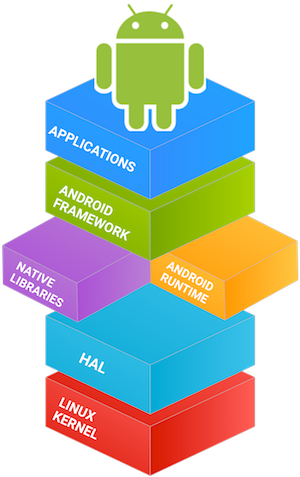
****

*Olá, tudo bem com vocês?*

*Espero que estejam bem, muitos aqui já devem me conhecer, mas para aqueles que não me conhecem, eu sou a Ana!*

*Com um imenso prazer, eu compartilharei TODOS os meus resumos, anotações e dicas baseados nas aulas do Bootcamp everis Kotlin Developer / Docs / Comunidades / Artigos / Vídeos / Plataformas de estudos / outros, para agregar o conhecimento de todos, inclusive o meu com o Feedback de vocês!*

*Cabe ressaltar que, assim como muitos, iniciei meus estudos de programação este ano e sei o quão é difícil se orientar nesse novo “mundão” de Dev’s. Pensando nisso, quero ajudá-los a se direcionarem no Bootcamp e nesta incrível e desafiadora jornada!*

***Obs.: Disponibilizarei os resumos em doses fracionadas.***

*Espero que vocês gostem!*

*Ass.: Ana Guerra*

**Introdução ao tema:**

É extremamente necessário conhecer a área da tecnologia que você deseja se aprofundar. Neste caso, é importante que aos poucos você aprimore o seu conhecimento e este documento diz respeito a base principal para iniciar a sua jornada.

**Como baixar o que você precisa:**

Inicialmente, caso se interesse por fazer Download da ferramenta (IDE) utilizada pela instrutora neste módulo, siga o seguinte vídeo passo a passo:

Intellij IDEA e Java: <https://www.youtube.com/watch?v=Ag0uQDMTxMI>

**Como participar da aula sem baixar essa ferramenta?**

Agora não é necessário você ter o Intellij IDEA em sua máquina. Entretanto, ao decorrer do Bootcamp será extremamente necessário fazer o download do tópico anterior. Por enquanto, use o Kotlin por aqui:

Kotlin online: <https://play.kotlinlang.org/>

**Por que baixar uma IDE e o que é isso?**

Para ter muito mais conhecimento do tema e do módulo do mesmo Bootcamp, “ Lógica Essencial da Programação”, acesse o meu artigo na plataforma da Digital Innovation One:

<https://web.digitalinnovation.one/articles/vem-ver-a-base-da-programacao?back=%2Farticles&page=1&order=oldest>

**O que você aprenderá no módulo *Introdução à Arquitetura da Plataforma Android* ?**

1. O Sistema Operacional Android;

. O que é o Android;

. Interface Android;

. O que é o Kernel Linux?

. O que é uma Plataforma Android?

2. Arquitetura da Plataforma Android:

. System APS;

. Java APIs;

. Native C/C++ Libraries:

. Android Run Time (ART);

. Chamada de Abstração de Hardware (HAL);

. Kernel do Linux.

3. Atualizações Dinâmicas do Sistema (DSU);

. O que é Project Treble?

. O que é Imagem Genérica do Sistema (GSI)?

. Atualizações Dinâmicas do Sistema (DSU)?

4. Segurança do Sistema;

. Objetivo;

. Dicas de Segurança;

5. Segurança com Dados:

. Qual a finalidade da Segurança de dados;

. Como obter uma excelente Segurança de Dados utilizando a Biblioteca Security?

6. Segurança com Dados em outras versões do Android;

7. Segurança com HTTPS e SSL;

. Conceito;

. O que é SSL (Secure Sockets Layer)?

. Exemplo de HTTPS;

8. Safety Net;

9. Criptografia;

. O que é uma criptografia?

. Complexidade de Implementação;

. Funcionalidades Suspensas;

10. Resolver problemas encontrados pelo Google Play;

. O que é o Programa App Security Improvement?

. Como funciona o Programa App Security Improvement?

11. Aplicações Potencialmente Nocivas (PHAs);

. Por que o termo PHAs é usado?

. Dentro do Dispositivo Android;

. O que o Google considera como “aplicativo potencialmente nocivo”?

12. Seja o primeiro a saber;

13. Android App Bundle;

. O que é o Android App Bundle?

. Recursos.

****

**INTRODUÇÃO À ARQUITETURA DA PLATAFORMA ANDROID – Jamile Ferreira**

**- O que é um Hardware?**

. É a parte física de um computador.

**- O que é um Sistema Operacional (SO)?**

. É um Software ou um conjunto de Softwares cuja função é administrar e gerenciar os recursos de um sistema, desde os componentes de hardware e sistemas de arquivos a programas terceiros, estabelecendo uma relação entre o computador e o usuário pela interface de interação;

. Interação do humano com o computador (entrada)→ gerenciamento e administração das interações→ Saída correspondente ao comando da entrada;

. Tipos de sistemas operacionais: Windows, Ubuntu, Mac Os, Android;

. Quando o dispositivo estiver ligado, o SO será o primeiro a ser executado e ele irá gerenciar o aparelho, tanto o Hardware, quanto os softwares e os periféricos plugados;

**- Sistema Operacional para celulares:**

. O estopim do Sistema Operacional que conhecemos hoje ocorreu por conta da elevada demanda do mercado por smartphones com funções mais variadas;

Ex: A Nokia havia adotado um dos grandes avanços do Sistema Operacional, ocorreu com o surgimento do Symbian, um Sistema Operacional mobile aberto (com o código livre para que as pessoas pudessem desenvolver aplicativos).



***O SISTEMA OPERACIONAL* ANDROID:**

**- O que é o Android?**

. O Android é o Sistema Operacional do Google para dispositivos móveis baseados no núcleo Kernel Linux;

. O Sistema Operacional Android possui a sua interface de usuário baseada na manipulação direta (dispositivos móveis sensíveis ao toque, como smartphones, relógios, carros e tablets);

. O código do Sistema Operacional Android é disponibilizado pelo Google sob licença de código aberto;

. O Android é muito popular entre empresas de tecnologia que buscam um software pronto, de baixo custo e personalizável para dispositivos de alta tecnologia.

**- Interface Android:**

. A interface do Sistema Operacional Android é composta por objetos virtuais. Estes objetos, funcionam como “entrada” para a inserção de comandos desejados pelo usuário;

. O objetivo principal da Interface do sistema operacional Android é proporcionar uma maior fluidez e sensação imediata da resposta do sistema pelo comando fornecido;

**- O que é o KERNEL LINUX? (Kernel = núcleo)**

. O Kernel Linux é o Núcleo do Sistema Operacional Android;

. É uma “ponte” entre os aplicativos do Sistema Operacional Android com o Hardware do computador (conectar o Hardware com o Software);

. O Kernel Linux tem como objetivo gerenciar os recursos do sistema (comunicação do Hardware com Software);

. O Kernel Linux é um núcleo composto por sistemas operativos que tem a finalidade de melhorar a manutenção e a modularidade do sistema operativo;

**- O que é uma Plataforma Android?**

. A plataforma Android é um ambiente online que conecta um consumidor com vários produtores, permitindo uma relação de troca, muito além da simples compra e venda. Plataformas virtuais são usadas para lazer, trabalho e entretenimento.

------------------------------------------------------------------------------------------------------------------

**- Informações a mais da aula:**

. Os aplicativos desenvolvidos para Android são feitos em Java e Kotlin;

. Kotlin é uma linguagem de programação nativa a Android;

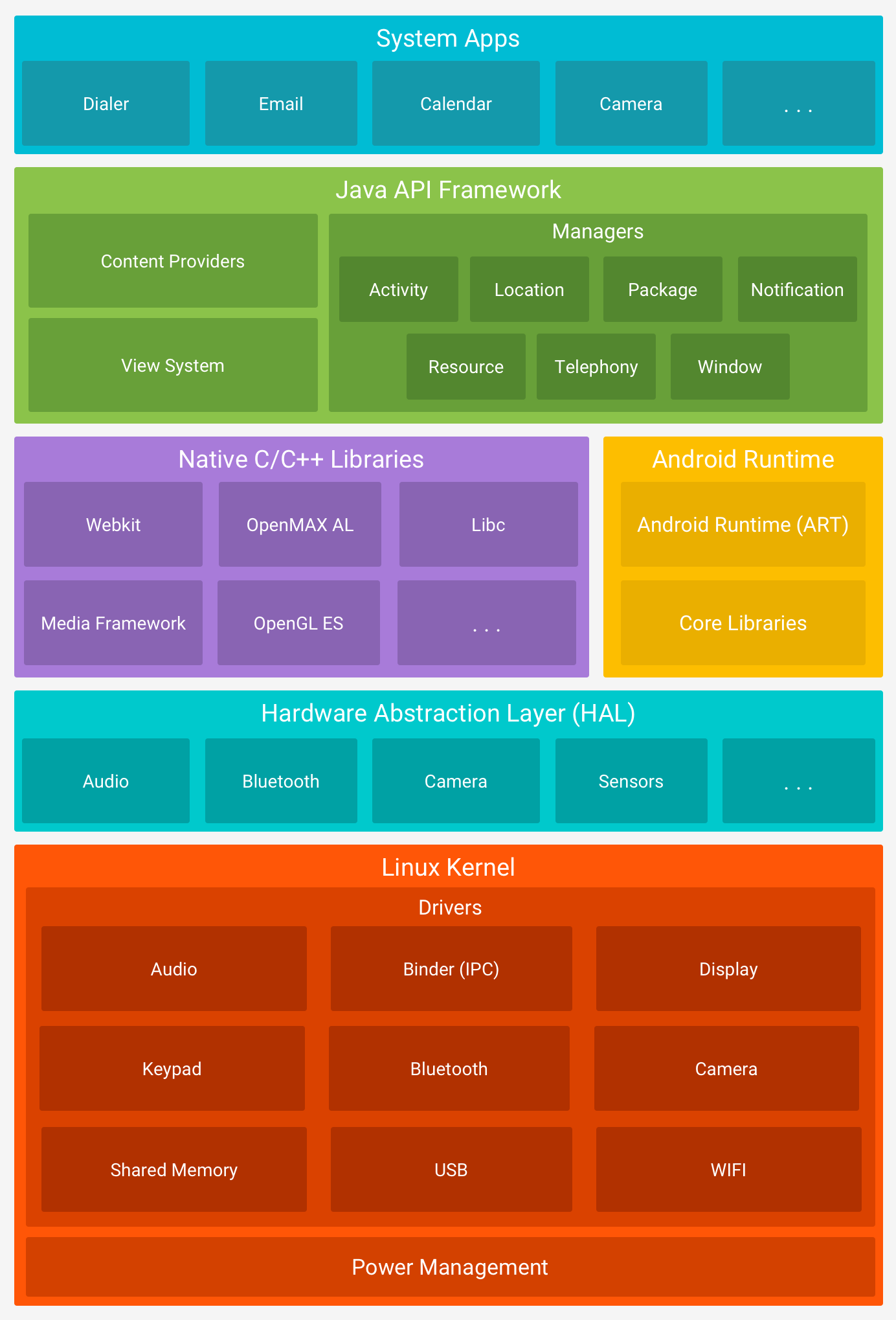
. Kernel e Android

**- Explicação, o que é um Software?**

. Software é a parte lógica do aparelho;

**-Explicação, o que é um Hardware?**

. O aparelho celular é o Hardware, peças que não fazem nada sem o Software.

Componentes da plataforma Android – pilha de software.

***Arquitetura* da Plataforma Android**

. O Android é uma pilha de software com base em Linux de código aberto para diversos dispositivos e fatores de forma.

**1) System APS:**

. O Android vem com os aplicativos principais para e-mail, envio de SMS, calendários, navegador de internet, contatos etc;

. Os aplicativos iniciais fornecidos pelo Android podem ser substituídos por aplicativos terceirizados, como por exemplo o app de e-mail, teclado, SMS e assim por diante;

**2) Java API: (IMPORTANTE)**

. As APIs (Interface de Programação de Aplicativos): é um conjunto de rotinas e padrões de programação para acesso a um aplicativo de software ou plataforma baseado na Web;

. APIs formam os blocos de programação que precisamos para criar os aplicativos Android simplificando a reutilização de componentes e serviços de sistema modulares e principais:

a) Sistema de Visualização: Rico e extensivo útil para programar a IU (user interface) de um aplicativo, com listas, grades, caixas de texto;

b) Gerenciador de Recursos: fornece uma organização de recursos sem código como Strings e gráficos localizados;

c) Gerenciador de Notificação: permite que todos os aplicativos emita alertas personalizados;

d) Gerenciador de Atividade: gerencia a atividade de cada aplicativo e cria uma pilha de navegação;

e) Provedores de conteúdo: permite que aplicativos acessem outros aplicativos.

**3) Native C/C++ Libraries:**

. Android Runtime (ART) e Hardware Abstraction Layer (HAL) são implementados por código nativo que exige as bibliotecas nativas programadas em C/C++;

**3.1) Android Run time (ART):**

. Cada aplicativo pode executar o próprio processo com uma instância do Android Run time;

. Android Run Time foi projetado para executar várias máquinas virtuais em dispositivos de baixa memória

. Algumas funções principais dos Android Run Time:

1. Compilação “ahead-of-time” (AOT) E “just-in-time” (JIT);

– compilação antes do tempo e na hora certa;

2. Coleta de lixo (GC) otimizada;

3. Relatório de erros;

4. Controle para monitorar campos específicos.

**4) Camada de abstração de hardware (HAL):**

. A HAL fornece interfaces padrão que expõem as capacidades de hardware do dispositivo;

. A HAL consiste em módulos de biblioteca, que dão acesso a um componente de hardware, como câmera ou Bluetooth;

**5) Kernel do Linux (Hardware do Sistema):**

. Kernel do Linux é o fundador da plataforma Android;

. Benefícios para o Android por usar o Kernel do Linux:

1. Aproveitar os recursos de segurança principais;

2.Os Fabricantes dos dispositivos podem desenvolver Drivers (comunicação do sistema com o hardware);

3. Integração de um melhor desenvolvimento da Bateria.

.

***Atualizações* Dinâmicas do Sistema (DSU):**

. OBJETIVO: Aprimorar o Sistema Android;

**- O que é Project Treble?**

. É um recurso que foi integrado no Android 8.0 (Android Oreo) para que as fabricantes atualizem o software de seus aparelhos de uma forma mais rápida;

1) Como funcionavam as atualizações antes do Project Treble?

. Antigamente, a fabricante pegavam o código Android, modifica de acordo o seu interesse e depois o deixava-o compatível com o Hardware;

. “Problema” o processo para passar deixar o código compatível ao Hardware poderia trazer inúmeros defeitos.

2) Como funcionam, atualmente, as atualizações com o Project Treble?

. Agora as outras fabricantes pegam o código do Android e o modificam de acordo o seu interesse e depois apenas atualizam o seu Sistema Operacional;

. Simplificou um processo de deixar o código compatível ao Hardware.

OBS: Esse ponto é importante para justificar o motivo de uma fabricante lançar um Sistema Operacional atualizado e logo em seguida as outras também o atualizarem.

**- O que é Imagem Genérica do Sistema (GSI)?**

. Atualmente, os aparelhos que possuem suporte ao Project Treble podem ter suas imagens (Sistemas Operacionais) baixados, sem que haja a necessidade de configurar o código no seu Hardware;

. A PlayStore fornece essas imagens.

**- O que são as Atualizações Dinâmicas do Sistema (DSU)? (LEMBRAR das partições A e B)**

. As atualizações Dinâmicas do Sistema são um recurso do sistema lançado no Android 10 que faz o seguinte:

1) Faz o Download de uma Imagem Genérica do Sistema (GSI), ou uma outra imagem do sistema Android, para o seu dispositivo;

2) Cria uma nova partição Dinâmica;

3) Carrega a Imagem Genérica do Sistema (GSI) para a nova partição;

4) Inicializa a Imagem Genérica do Sistema (GSI) como um Sistema Operacional (SO) convidado no dispositivo;

. As Atualizações Dinâmicas do Sistema (DSU) permitem alternar facilmente entre a imagem do sistema atual e a Imagem Genérica do Sistema para que você possa testá-la sem correr risco de corromper a imagem do sistema atual.. Partição do tipo sistema A e B realiza a atualização do sistema, mesclando as atuações… Uma hora A atualiza, enquanto B está desligada e depois vice e versa. Lembrando que dá para mexer no celular enquanto ele está atualizando. Antigamente não dava e tinha um maior risco do celular não voltar a ligar depois da atualização;



Imagem Pura do Sistema (Imagem= Sistema em si)

| OBS: TODAS IMAGENS GENÉRICAS DO SISTEMA (GSI) TÊM QUE SER ASSINADAS PELO GOOGLE!!!!! PARA PROVAR QUE É UM SISTEMA CONFIÁVEL. |
| --- |



***Segurança* do Sistema**

- Qual a finalidade da Segurança do Sistema?

. Prevenir invasões ao sistema e roubo de dados dos usuários;

**1) Práticas Recomendadas Para Segurança De Apps;**

- Objetivo:

. Deixar o Aplicativo mais seguro ajuda a preservar a confiança do usuário e a integridade do dispositivo;

. Para tornar um Aplicativo seguro siga os passos:

1) Aplicar Comunicação Segura:

- Segurança na comunicação do seu App para outro App e do seu App para um site;

- Mostrar um Seletor de App;

- Aplicar Permissões Baseadas em Assinaturas;

- Cancelar Permissão de Acesso aos Provedores de Conteúdo do seu App;

- Pedir credenciais antes de exibir informações Confidenciais;

- Aplicar Medidas de Segurança de Rede;

. Usar tráfego SSL;

. Adicionar um Tráfico de Segurança de Rede;

. Criar o Próprio Gerenciador de Confiança;

- Usar objetos WebView com cuidado;

. Usar canais de mensagem HTML;

2). Oferecer as permissões corretas;

- Usar intents para adiar permissões;

- Compartilhar os Dados de Forma Segura Entre os Apps;

3) Armazenar os Dados de Forma Segura;

- Armazenar dados privados no armazenamento interno;

- Guardar dados no armazenamento externo com base no caso de uso;

. Verificar a validade de dados;

. Acessar arquivos específicos do app;

. Acessar arquivos compartilhados;

. Verificar a validade dos dados;

- Armazenar somente dados não confidenciais em arquivos cache;

. Usar SharedPreferences no modo privado;

4)Manter os serviços e as dependências atualizados;

- Verificar o provedor de segurança do Google Play Services;

- Atualizar todas as dependências do App;

**2) Dicas de Segurança:**

- Objetivo:

. Diminuir a frequência e o impacto dos problemas de segurança de aplicativos;

. Para evitar os problemas de segurança siga os passos:

1) Armazenamento de Dados:

- É a preocupação de segurança mais comum para um aplicativo Android;

- O objetivo é ter conhecimento se os dados estão sendo acessados por outros Apps, ou não;

I. Armazenamento Interno:

. Arquivos criados no armazenamento interno pode ser acessados apenas pelo próprio App;

. O ideal é criptografar até os arquivos internos;

II. Armazenamento Externo:

. Arquivos com armazenamento externo, como cartões SD não são confiáveis porque podem ser modificados por outros Apps e pode ser removido por um usuário;

. Caso alguma fonte externa deseje se conectar a cabo com o dispositivo, é requisitado um desbloqueio para entrar;

. Observe que atualmente quase nenhum celular possui mais entrada de cartão, para prevenir entradas indesejadas no dispositivo.

III. Provedores de conteúdo:

. São requisitados acessos ao Aplicativo que deseja utilizar dados do dispositivo, como imagens, acesso à câmera, acesso ao microfone.

. Ex: Quando o Instagram pede para usar a sua câmera;

2) Usar Permissões:

- Sandbox é uma solução usada para testar, executar e bloquear programas e códigos potencialmente maliciosos antes que eles afetem uma aplicação, sistema ou rede;

- As permissões são usadas com a finalidade de liberar o acesso, ou não, aos dados e recursos dos demais Apps;

I. Solicite Permissões:

. Objetivo é minimizar o número de solicitações desejadas;

. Use a ideia de: “se uma permissão não é necessária para o funcionamento do seu app, não é preciso solicitá-la;

. Preferenciar o projeto do App que não precise exigir permissões;

II. Criar Permissões:

. Objetivo é definir o menor número de permissões e atender os requisitos de segurança;

. Ao em vez de criar uma permissão, o ideal é usar as permissões já estabelecidas por outros desenvolvedores;

| **OBS:** Acesse os links dos Documentos do Android para melhor entendimento e aprofundamento dos conteúdos. |
| --- |
| [**https://developer.android.com/training/articles/security-tips?hl=pt-br**](https://developer.android.com/training/articles/security-tips?hl=pt-br) |
| **https://developer.android.com/topic/security/data?hl=pt-br** |
| [**https://developer.android.com/training/articles/security-ssl?hl=pt-br**](https://developer.android.com/training/articles/security-ssl?hl=pt-br) |



**Segurança com Dados**

**- Qual a finalidade da Segurança de dados?**

. A Segurança de Dados possui a mesma finalidade que a Segurança do Sistema. Ou seja, prevenir invasões ao sistema e roubo de dados dos usuários;

- Tipos de Segurança de dados:

I. Segurança Forte que equilibra ótima criptografia e bom desempenho:

. Este nível de Segurança é utilizado em Aplicativos destinados ao consumidor, como WhatsApp e Instagram.

II. Segurança Máxima é adequado para apps empresariais que exigem Keystore no Hardware presença de usuário para fornecer acesso a chaves;

Ex: Apps do Banco.

**- Como obter uma excelente Segurança de Dados utilizando a Biblioteca Security?**

1) Gerenciamento de Chaves:

. Conjunto de Chaves: Serve para criptografar um arquivo ou dados compartilhados;

. Chave Primária: criptografa todo o conjunto de chaves.

2) Classes Incluídas na Biblioteca:

. Fornece dados mais seguros em repouso;

3) Incluir a Biblioteca no seu projeto:

. Usar o comando build.gradle;

4) Ler Arquivos:

. Usar o comando Encrypted File para ler o conteúdo de um Arquivo de forma mais segura;

5) Gravar Arquivos:

. Pode-se utilizar Encrypted File ou uns comandos do DOCs;

6) Editar Preferências compartilhadas:

. Usar Encrypted SharedPreferences para editar o conjunto de preferências compartilhadas de um usuário de forma mais segura;

| **OBS:** Acesse no Link dos Docs para melhor entendimento do conteúdo e aprofundamento da matéria: |
| --- |
| https://developer.android.com/topic/security/data |

**Segurança com Dados em outras versões do Android**

**- Os respectivos passos são idênticos ao de Segurança Android;**

1) Gerenciamento de Chaves:

2) Classes Incluídas na Biblioteca:

3) Incluir a Biblioteca no seu Projeto:

4) Ler Arquivos:

5) Gravar Arquivos:

6) Editar preferências Compartilhadas:



**Segurança com HTTPS e SSL**

**- Conceito:**

- O que é HTTPS (Hypertext Protocol Secure → HTTP + SSL)?

. Protocolo de Transferência de Hipertexto Seguro, é uma implementação do Protocolo HTTP sobre uma camada adicional de segurança;

- O que é SSL (Secure Sockets Layer)?

. Agora conhecido como TLS (Transport Layer Security) estabelece comunicação criptografada entre clientes e servidores;

. Porque é necessário a segurança neste quesito? A comunicação entre os Apps e os sites, por TLS, pode ser interceptada por entidades mal-intencionadas, por isso é necessário seguir algumas instruções para manter as comunicações do seu App segura:

- Exemplo de HTTPS:

- Problemas comuns ao verificar certificados de servidor:

. Autoridade de certificação desconhecida;

. Certificado de servidor auto assinado;

. Autoridade de certificação intermediária ausente;

- Problemas comuns com a verificação do nome do Host:

- Avisos sobre o uso direto do SSLSocket;

- Lista de Proibições;

- Fixação:

- Certificados do cliente;

- Nogotofail: Uma ferramenta de teste de segurança do tráfego de rede.

| **OBS:** Continue a leitura e o aprofundamento no conteúdo pelos DOCs |
| --- |
| [https://developer.android.com/training/articles/security-ssl#HttpExample](https://developer.android.com/training/articles/security-ssl#HttpsExample) |
| <https://developer.android.com/training/articles/security-config#CertificatePinning> |
| https://developer.android.com/training/articles/security-gms-provider |



**SafetyNet**

**- O que é Safety Net?**

. O SafetyNet oferece um conjunto de serviços e APIs (Interface de Programação de Aplicações) que ajudam a proteger seu app contra ameaças à segurança, incluindo adulteração de dispositivos, URLs incorretas, apps potencialmente nocivos e usuários falsos.



**Criptografia**

**- O que é Criptografia?**

. Conjunto de princípios e técnicas para cifrar a escrita, torná-la ininteligível para os que não tenham acesso às convenções combinadas

. Utiliza-se o Sistema Android Keystore e a identificação de um provedor (fornecedor de um conteúdo);

| **OBS:** Esta aplicação serve para todas as versões do Android. |
| --- |

**- Escolha de um Algoritmo recomendado:**

. Quando não possuir compatibilidade com um sistema de terceiros para proteger o Seu App use um dos seguintes algoritmos:

| **Classe** | **Recomendação** |
| --- | --- |
| Cipher | AES no modo CBC ou GCM com chaves de 256 bits (como **AES/GCM/NoPadding**) |
| MessageDigest | Família SHA-2 (por exemplo, **SHA-256**) |
| Mac | HMAC da família SHA-2 (por exemplo, **HMACSHA256**) |
| Assinatura | Família SHA-2 com ECDSA (por exemplo, **SHA256withECDSA**) |

**- Complexidade de Implementação:**

. Resumo da Mensagem OAEP:

. O RSA e a OAEP são parametrizadas por dois resumos de mensagens diferentes: o resumo “principal” e o resumo MGF1;

. Para Android, o SHA-1 é usado para o Resumo Principal;

. Mais controle sobre os resumos que o App usa, solicite uma criptografia OAEP Padding;



**Funcionalidades Suspensas:**

. O que não devemos mais usar nos Apps:

1) Algoritmo do Bouncy Castle:

2) Criptográficas Baseadas em Senhas sem IV;

3) Provedor de Criptografia;

| **OBS:**Olhar nos DOCs: |
| --- |
| https://developer.android.com/guide/topics/security/cryptography#impl-complexities |



**Resolver problemas encontrados pelo Google Play**

**- O que é o *Programa App Security Improvement:***

. É um serviço fornecido aos desenvolvedores de aplicativos para melhorar a segurança e correção de erros em seus Apps;

**- Como Funciona o Programa App Security Improvement?**

1) O aplicativo é verificado em relação a proteção e segurança para correção de erros e possíveis problemas de segurança;

2) Caso o App tenha alguma falha de segurança, o Google Play avisa, o mais rápido, possível e por e-mail, o fabricante para manter a integridade de seus usuários;

3) Para confirmar se o problema foi resolvido, basta fazer um upload da versão do app no Google Play Console;



**Aplicações Potencialmente Nocivas (PHAs)**

**- Por que o termo PHAs é usado?**

. Serve para diferenciar um vírus que corrompe dados do computador de um Cavalo de Tróia que rouba os dados de segurança dele.

**- Dentro do Dispositivo Android:**

. Os vírus não se “espalham” para outros celulares, como um vírus tradicional. Eles atuam em lugares específicos do celular.

**- O que o Google considera como “aplicativo potencialmente nocivo”?**

1) Fraude de Cobrança e Contratação de Serviços:

. Ocorre em SMS, Chamadas e Serviços;

. Aplicativo que manda mensagem, liga ou assina serviços junto à operadora de maneira automática ou sem informar o usuário da maneira mais clara o possível;

2) Phishing:

. Golpes por e-mails falsos que são “cópias” de empresas que exigem dados bancários, por exemplo;

3) Escalação de Privilégio e rooting:

. São aplicativos que acessam dados ou permissões do sistema de forma irregular;

. Aplicativos que interferem nas permissões administrativas para evitar desinstalação;

4) Ransomware:

. Vírus que bloqueiam o uso do aparelho;

5) Spam:

. Aplicativos que enviam mensagens não solicitadas para os contatos do aparelho;

6) Programas Espiões (Spyware)

. Um programa que transfere os dados do usuário sem a sua solicitação;

7) Negação de Serviço:

. Aplicativos que colocam o telefone celular a serviço de ataques de negação de serviços distribuídos;

8) Downloader Hostil:

. Um aplicativo nocivo baixar outros aplicativos maliciosos;

9) Cavalos de Tróia:

. “O presente de Grego” (lembra de história);

. O cavalo de troia é um aplicativo malicioso que se disfarça de algo útil e benigno.

10) Backdoor:

. “Porta dos fundos” é um aplicativo que libera a entrada de outros aplicativos e o acesso ao dispositivo;

11) Ameaças de outras Plataformas:

. Google contabiliza códigos que podem atacar outras plataformas que não o Android;

12) Fraude de instalação e cliques:

. Aplicativos com anúncios falsificando os demais cliques e instalando aplicativos comissionados;

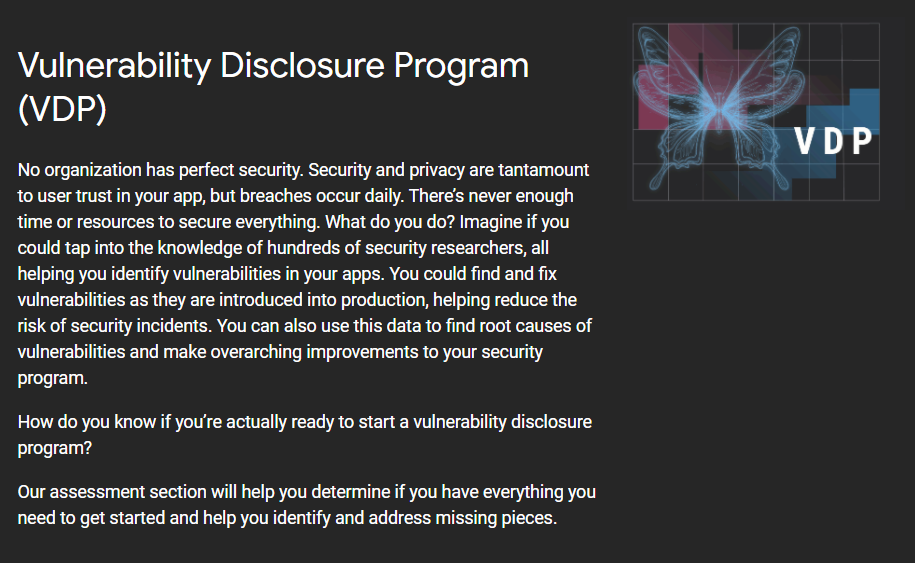
**Seja o primeiro a saber**

**Acesse este Site para saber mais sobre:**

**https://developers.google.com/android/play-protect/starting-a-vdp**

- Quando você estiver em um nível mais avançado de programação, passe a ajudar a comunidade Dev a se proteger dos demais ataques de segurança usando este link;

- Aliás, estudar tudo sobre Ethical Hacker;



**Android App Bundle**

**- O que é o Android App Bundle?**

. É o novo formato de publicação oficial do Android, que oferece uma maneira mais eficiente de criar e lançar apps;

. O Android App Bundle traz mais facilidade para uma ótima experiência em um app menor, o que pode gerar mais instalações e pode reduzir o número de desinstalações;

**- Recursos:**

1) Gerenciar Versões com mais Eficiência:

. Inclui todo o código compilado, recursos e bibliotecas nativas do App;

2) Vantagens de um App menor:

. Download mais leve para os usuários, sem o código não utilizado e os recursos necessários para outros dispositivos;

. Acompanhe o tamanho do App no Google Play Console;

3) Personaliza o envio de recursos:

. Personalize e controle a experiência do usuário, quando e como forem necessários . Ou seja, o tamanho do App não precisa aumentar, ele pode apenas Atualizar e trocar ou modificar os recursos desejados;

4) Acelerar a velocidade de Engenharia:

. Coexistência do App principal e de Apps modulares, onde são feitas as modificações desejadas e depois adicionadas ao App principal;

5) Menor tempo de Criação:

. Os sistemas de criação são otimizados , como o Android Studio, para a utilização em Apps Modulares (Cópia do App original). Estes sistemas de criação são mais rápidos, e menores. Ou seja, menor tempo de produção, e maior tempo de programação;

| **OBS: Acesse SEMPRE os DOCs** |
| --- |
| **https://developer.android.com/platform/technology/app-bundle** |