

Publicação de Aplicação e Mercado

Neste módulo, vamos abordar os passos necessários para preparar um aplicativo Flutter para publicação. Discutiremos as verificações e ajustes finais que precisam ser feitos antes de publicar o aplicativo, incluindo a configuração do arquivo de manifesto, otimização do desempenho, gerenciamento de permissões e preparação de ícones e imagens de lançamento. A preparação adequada garante que o aplicativo esteja pronto para ser submetido às lojas de aplicativos sem problemas.

Preparação para Publicação

Verificações Finais

1. Verificação de Dependências

- Certifique-se de que todas as dependências estão atualizadas e compatíveis.
- Execute `flutter pub get` para garantir que todas as dependências estejam instaladas corretamente.

2. Remoção de Código de Depuração

- Remova qualquer código de depuração, como logs e breakpoints.
- Certifique-se de que o aplicativo está configurado para o modo de lançamento (release mode).

Configuração do Arquivo `pubspec.yaml`

1. Atualização de Informações Básicas

Atualize o nome do aplicativo, descrição, versão e outras informações relevantes:

name: my_flutter_app

description: Um novo aplicativo Flutter.

version: 1.0.0+1

2. Configuração de Ícones e Imagens de Lançamento

- Adicione ícones e imagens de lançamento personalizadas no diretório assets.

- Atualize o arquivo pubspec.yaml para incluir os recursos:

flutter:

assets:

- assets/icons/

- assets/images/

Otimização de Desempenho

1. Minificação de Código

- Utilize ferramentas de minificação para reduzir o tamanho do código.

- Certifique-se de que a configuração de minificação está habilitada no arquivo build.gradle para Android:

```
buildTypes {
```

```
release {  
  
    minifyEnabled true  
  
    shrinkResources true  
  
    proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'),  
    'proguard-rules.pro'  
  
}  
  
}
```

2. Verificação de Performance

- Utilize o Flutter DevTools para monitorar o desempenho do aplicativo e identificar áreas que precisam de otimização.
- Execute testes de desempenho para garantir que o aplicativo funcione de maneira eficiente.

Gerenciamento de Permissões

1. Verificação de Permissões Necessárias

- Verifique todas as permissões que o aplicativo solicita e remova as que não são necessárias.
- Atualize o arquivo AndroidManifest.xml para Android e o arquivo Info.plist para iOS para incluir apenas as permissões necessárias.

2. Solicitação de Permissões no Código

- Garanta que o aplicativo solicita permissões de maneira adequada no código.

- Utilize pacotes como `permission_handler` para gerenciar permissões de forma eficiente:

```
import 'package:permission_handler/permission_handler.dart';
```

```
Future<void> requestPermission() async {  
  
  if (await Permission.location.request().isGranted) {  
  
    // Permissão concedida  
  
  }  
  
}
```

Nesta aula, veremos os passos necessários para publicar um aplicativo Flutter na Google Play Store. Abordaremos o processo de criação de uma conta de desenvolvedor no Google Play, a preparação do aplicativo para upload, a configuração de assinaturas e a submissão do aplicativo para revisão. Seguindo esses passos, você poderá disponibilizar seu aplicativo para milhões de usuários de dispositivos Android.

Publicando no Google Play Store

Criação de Conta de Desenvolvedor no Google Play

1. Registro na Google Play Console

- Acesse Google Play Console e registre-se como desenvolvedor.
- Pague a taxa única de inscrição (cerca de \$25).

Preparação do Aplicativo para Upload

1. Gerar APK/Bundle de Aplicativo

- Execute o comando para gerar o APK:

```
flutter build apk --release
```

- Ou, para gerar um bundle de aplicativo (recomendado):

```
flutter build appbundle --release
```

2. Assinatura do APK/Bundle

- Configure a assinatura do aplicativo no arquivo key.properties:

```
storePassword=your-store-password
```

```
keyPassword=your-key-password
```

```
keyAlias=your-key-alias
```

```
storeFile=path/to/your-keystore.jks
```

- Atualize o arquivo build.gradle com as configurações de assinatura:

```
gradle
```

Copiar código

```
android {
```

```
...
```

```
    signingConfigs {
```

```
        release {
```

```
            keyAlias keystoreProperties['keyAlias']
```

```
            keyPassword keystoreProperties['keyPassword']
```

```
        storeFile file(keystoreProperties['storeFile'])

        storePassword keystoreProperties['storePassword']
    }

}

buildTypes {

    release {

        signingConfig signingConfigs.release
    }

}

}
```

Submissão do Aplicativo na Google Play

1. Criar um Novo Projeto no Google Play Console

- Acesse a Google Play Console e crie um novo projeto para seu aplicativo.
- Preencha as informações básicas, como nome, descrição e categoria do aplicativo.

2. Upload do APK/Bundle

- Na seção de gerenciamento de versões, faça o upload do APK ou bundle gerado.
- Preencha as informações de lançamento, incluindo notas de versão e direcionamentos de público-alvo.

3. Preencher Detalhes da Loja

- Adicione capturas de tela, ícones e descrições detalhadas.
- Configure os preços e países onde o aplicativo estará disponível.

4. Submeter para Revisão

- Revise todas as informações e submeta o aplicativo para revisão.
- Acompanhe o status da revisão na Google Play Console.

Nesta aula, aprenderemos como publicar um aplicativo Flutter na Apple App Store. Abordaremos o processo de criação de uma conta de desenvolvedor da Apple, preparação do aplicativo para upload, configuração de certificados e perfis de provisionamento, e submissão do aplicativo para revisão. Publicar na Apple App Store exige conformidade com diretrizes rigorosas, garantindo alta qualidade e segurança para os usuários de dispositivos iOS.

Publicando na Apple App Store

Criação de Conta de Desenvolvedor da Apple

1. Registro no Apple Developer Program

- Acesse Apple Developer Program e registre-se como desenvolvedor.
- Pague a taxa anual de inscrição (cerca de \$99).

Preparação do Aplicativo para Upload

1. Configuração de Certificados e Perfis de Provisionamento

- Gere um certificado de distribuição no Apple Developer Portal.

- Crie um perfil de provisionamento de distribuição associado ao seu certificado e ao ID do aplicativo.

2. Configuração do Projeto Xcode

- Abra o projeto iOS do aplicativo Flutter no Xcode (ios/Runner.xcworkspace).
- Configure as informações do aplicativo, como nome, identificador de pacote e equipe de desenvolvimento.
- Selecione o perfil de provisionamento e certificado corretos para o modo de release.

3. Gerar Arquivo IPA

- No Xcode, selecione o destino Generic iOS Device.
- Vá para Product > Archive para criar um arquivo de arquivamento.
- Após a conclusão da construção, use o Organizer para exportar o arquivo IPA para upload.

Submissão do Aplicativo na Apple App Store

1. Criar um Novo Projeto no App Store Connect

- Acesse App Store Connect e crie um novo projeto para seu aplicativo.
- Preencha as informações básicas, como nome, descrição e categoria do aplicativo.

2. Upload do IPA usando Transporter

- Use o aplicativo Transporter para fazer o upload do arquivo IPA para a App Store Connect.

- Verifique se há erros durante o upload e corrija quaisquer problemas relatados.

3. Preencher Detalhes da Loja

- Adicione capturas de tela, ícones e descrições detalhadas.
- Configure os preços e países onde o aplicativo estará disponível.

4. Submeter para Revisão

- Revise todas as informações e submeta o aplicativo para revisão.
- Acompanhe o status da revisão no App Store Connect.

Nesta aula, discutiremos estratégias de marketing e monitoramento do mercado para aplicativos móveis. Abordaremos técnicas para promover seu aplicativo, como otimização para lojas de aplicativos (ASO), campanhas de marketing digital e uso de redes sociais. Além disso, veremos como monitorar o desempenho do aplicativo no mercado, analisar feedback dos usuários e realizar melhorias contínuas.

Marketing e Monitoramento do Mercado

Estratégias de Marketing

1. Otimização para Lojas de Aplicativos (ASO)

- Utilize palavras-chave relevantes no título e descrição do aplicativo.
- Adicione capturas de tela atraentes e vídeos de demonstração.
- Obtenha avaliações e classificações positivas de usuários.

2. Campanhas de Marketing Digital

- Utilize anúncios pagos no Google Ads e redes sociais para alcançar um público maior.
- Colabore com influenciadores para promover seu aplicativo.
- Utilize email marketing para engajar usuários existentes.

3. Uso de Redes Sociais

- Crie perfis dedicados nas principais redes sociais (Facebook, Instagram, Twitter) para promover o aplicativo.
- Compartilhe atualizações, dicas e interaja com os usuários para criar uma comunidade em torno do aplicativo.

Monitoramento do Desempenho

1. Ferramentas de Análise

- Utilize ferramentas como Google Analytics, Firebase Analytics e App Annie para monitorar o desempenho do aplicativo.
- Acompanhe métricas como downloads, engajamento, retenção e receita.

2. Feedback dos Usuários

- Leia e responda às avaliações e comentários dos usuários nas lojas de aplicativos.
- Utilize ferramentas de feedback dentro do aplicativo para coletar sugestões e identificar problemas.

3. Melhorias Contínuas

- Lançar atualizações regulares com base no feedback dos usuários e nas análises de desempenho.

- Corrija bugs, adicione novas funcionalidades e melhore a experiência do usuário para manter os usuários engajados.

Exemplo de Estratégia de Marketing

1. Campanha de Anúncios no Google Ads

- Crie uma campanha de anúncios segmentada para o público-alvo do seu aplicativo.
- Utilize palavras-chave relevantes e anúncios visuais atraentes para aumentar os downloads.
- Monitore o desempenho da campanha e ajuste as configurações conforme necessário para maximizar o retorno sobre o investimento (ROI).

Esses exemplos e explicações fornecem uma base sólida para iniciantes em Flutter aprenderem a publicar e promover seus aplicativos no mercado. Para mais detalhes, consulte a documentação oficial do Flutter: [Flutter Documentation](#).

Materiais Extras

Você pode realizar o download do arquivo contendo os materiais extras utilizados ao longo das aulas por meio do seguinte link: <https://drive.google.com/file/d/1mg7lqMI8Pt2zl0rHlsFS0Qew00YN-sEX/view?usp=sharing>.

Conteúdo Bônus

Vídeo “Como Publicar Aplicativos Flutter na Google Play”: Apresentado por João Victor no canal Flutterando TV, este vídeo tutorial detalha o processo de publicação de aplicativos Flutter na Google Play Store, abordando desde a configuração inicial até a submissão final.

Referências Bibliográficas

BOYLESTAD, R. L.; NASHELSKY, L. Dispositivos Eletrônicos e Teoria de Circuitos. 11. ed. Pearson, 2013.

DEITEL, P. J.; DEITEL, H. M. Ajax, Rich Internet Applications e Desenvolvimento Web para Programadores. Pearson, 2008.

DUARTE, W. Delphi para Android e iOS: Desenvolvendo Aplicativos Móveis. Brasport, 2015.

FELIX, R.; SILVA, E. L. da. Arquitetura para Computação Móvel. 2. ed. Pearson, 2019.

LEE, V.; SCHNEIDER, H.; SCHELL, R. Aplicações Móveis: Arquitetura, Projeto e Desenvolvimento. Pearson, 2005.

MARINHO, A. L.; CRUZ, J. L. da. Desenvolvimento de Aplicações para Internet. 2. ed. Pearson, 2019.

MOLETTA, A. Você na Tela: Criação Audiovisual para a Internet. Summus, 2019.

SILVA, D. (Org.) Desenvolvimento para dispositivos móveis. Pearson, 2017.

Ir para exercício