



NFC

Neste módulo, exploraremos os fundamentos da tecnologia Near Field Communication (NFC). Discutiremos o que é NFC, como funciona, e suas principais aplicações. NFC é uma tecnologia de comunicação sem fio que permite a troca de dados entre dispositivos próximos. É amplamente utilizada em pagamentos móveis, autenticação e compartilhamento de dados. Ao final desta aula, você terá uma compreensão básica da tecnologia NFC e seu funcionamento.

Fundamentos de NFC

O que é NFC?

NFC (Near Field Communication) é uma tecnologia de comunicação sem fio de curto alcance que permite a troca de dados entre dispositivos compatíveis quando estão próximos um do outro, geralmente a uma distância de até 4 cm. É uma tecnologia derivada do RFID (Radio Frequency Identification).

Como Funciona o NFC?

NFC funciona por meio de indução eletromagnética entre duas antenas: uma no dispositivo emissor e outra no receptor. Existem dois modos de operação principais:

Modo Ativo: Ambos os dispositivos geram seus próprios campos eletromagnéticos.

Modo Passivo: Apenas um dispositivo gera o campo eletromagnético, enquanto o outro usa esse campo para se comunicar.

Aplicações Comuns do NFC

Pagamentos Móveis: Utilizado em serviços como Google Pay e Apple Pay.

Autenticação: Para acessar prédios, sistemas de transporte, etc.

Compartilhamento de Dados: Troca de contatos, fotos, e outros tipos de dados entre dispositivos.

Benefícios do NFC

Conveniente: Permite trocas rápidas de dados com uma simples aproximação.

Seguro: Ideal para aplicações que requerem segurança, como pagamentos.

Versátil: Pode ser usado em uma variedade de cenários, desde pagamentos até identificação e automação.

Nesta aula, abordaremos como configurar NFC no seu aplicativo Flutter. Veremos os passos necessários para integrar o NFC ao Flutter, incluindo a adição de dependências, configuração do AndroidManifest.xml, e a inicialização do pacote NFC no Flutter. Esta configuração é essencial para habilitar a leitura e escrita de tags NFC no seu aplicativo.

Configurando NFC no Flutter

Passos para Configurar NFC

1. Adicionar Dependências ao pubspec.yaml

dependencies:

flutter:

sdk: flutter

nfc_manager: latest_version

2. Configurar AndroidManifest.xml

Adicione as permissões necessárias e configure o filtro de intenções para o NFC.

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"

    package="com.example.nfc_app">

    <uses-permission android:name="android.permission.NFC" />

        <uses-feature            android:name="android.hardware.nfc"
android:required="true" />

    <application

        android:label="nfc_app"

        android:icon="@mipmap/ic_launcher">

        <activity

            android:name=".MainActivity"

            android:launchMode="singleTop">

            <intent-filter>

                <action android:name="android.nfc.action.NDEF_DISCOVERED" />

                <category android:name="android.intent.category.DEFAULT" />

                <data android:mimeType="text/plain" />

            </intent-filter>
```

```
<intent-filter>
```

```
    <action android:name="android.nfc.action.TECH_DISCOVERED" />
```

```
</intent-filter>
```

```
<intent-filter>
```

```
    <action android:name="android.nfc.action.TAG_DISCOVERED" />
```

```
</intent-filter>
```

```
<meta-data
```

```
    android:name="android.nfc.action.TECH_DISCOVERED"
```

```
    android:resource="@xml/nfc_tech_filter" />
```

```
</activity>
```

```
</application>
```

```
</manifest>
```

3. Criar o arquivo res/xml/nfc_tech_filter.xml

Este arquivo define os tipos de tags NFC que seu aplicativo pode ler.

```
<resources xmlns:xliff="urn:oasis:names:tc:xliff:document:1.2">
```

```
    <tech-list>
```

```
        <tech>android.nfc.tech.NfcA</tech>
```

```
        <tech>android.nfc.tech.NfcB</tech>
```

```
        <tech>android.nfc.tech.NfcF</tech>
```

<tech>android.nfc.tech.NfcV</tech>

<tech>android.nfc.tech.Ndef</tech>

<tech>android.nfc.tech.NdefFormatable</tech>

<tech>android.nfc.tech.MifareClassic</tech>

<tech>android.nfc.tech.MifareUltralight</tech>

</tech-list>

</resources>

4. Inicializar o NFC no Flutter

```
import 'package:flutter/material.dart';
```

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

```
void main() {
```

```
  runApp(MyApp());
```

```
}
```

```
class MyApp extends StatelessWidget {
```

```
  @override
```

```
  Widget build(BuildContext context) {
```

```
    return MaterialApp(
```

```
      home: HomeScreen(),
```

```
    );
```

```
}
```

```
}
```

```
class HomeScreen extends StatefulWidget {
```

```
  @override
```

```
  _HomeScreenState createState() => _HomeScreenState();
```

```
}
```

```
class _HomeScreenState extends State<HomeScreen> {
```

```
  @override
```

```
  void initState() {
```

```
    super.initState();
```

```
    NfcManager.instance.isAvailable().then((isAvailable) {
```

```
      if (isAvailable) {
```

```
        print('NFC está disponível');
```

```
      } else {
```

```
        print('NFC não está disponível');
```

```
      }
```

```
    });
```

```
}
```

```
  @override
```

```
Widget build(BuildContext context) {  
  
  return Scaffold(  
  
    appBar: AppBar(  
  
      title: Text('Configuração NFC'),  
  
    ),  
  
    body: Center(  
  
      child: Text('NFC Configurado!'),  
  
    ),  
  
  );  
  
}
```

Explicação: Este exemplo mostra como configurar o NFC no Flutter, incluindo a verificação de disponibilidade do NFC no dispositivo.

Nesta aula, aprenderemos como ler tags NFC usando Flutter. Veremos como utilizar o pacote `nfc_manager` para detectar e ler dados de tags NFC. Abordaremos a configuração do listener de NFC e a leitura de diferentes tipos de tags. Esta aula fornecerá as habilidades necessárias para integrar a funcionalidade de leitura de NFC no seu aplicativo Flutter.

Leitura de Tags NFC

Passos para Ler Tags NFC

1. Adicionar Dependências ao `pubspec.yaml`

dependencies:

flutter:

 sdk: flutter

 nfc_manager: latest_version

2. Configurar AndroidManifest.xml

Certifique-se de que as permissões e filtros de intenções estão configurados corretamente (conforme explicado na Aula 02).

3. Código para Leitura de Tags NFC

```
import 'package:flutter/material.dart';

import 'package:nfc_manager/nfc_manager.dart';

void main() {

  runApp(MyApp());

}

class MyApp extends StatelessWidget {

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      home: HomeScreen(),

    );
```



```
}
```

```
}
```

```
class HomeScreen extends StatefulWidget {
```

```
  @override
```

```
  _HomeScreenState createState() => _HomeScreenState();
```

```
}
```

```
class _HomeScreenState extends State<HomeScreen> {
```

```
  String _nfcData = 'Nenhuma tag lida';
```

```
  @override
```

```
  void initState() {
```

```
    super.initState();
```

```
    NfcManager.instance.startSession(onDiscovered: (NfcTag tag) async {
```

```
      setState(() {
```

```
        _nfcData = tag.data.toString();
```

```
      });
```

```
      NfcManager.instance.stopSession();
```

```
    });
```

```
}
```

```
  @override
```

```
Widget build(BuildContext context) {  
  
  return Scaffold(  
  
    appBar: AppBar(  
  
      title: Text('Leitura de Tags NFC'),  
  
    ),  
  
    body: Center(  
  
      child: Text('Dados da tag: $_nfcData'),  
  
    ),  
  
  );  
  
}
```

Explicação: Este exemplo mostra como iniciar uma sessão NFC e ler os dados de uma tag NFC quando ela é detectada.

Nesta aula, exploraremos como escrever dados em tags NFC usando Flutter. Veremos como utilizar o pacote `nfc_manager` para detectar uma tag NFC e escrever dados nela. Discutiremos os tipos de dados que podem ser escritos e como formatar a tag para escrita. Esta aula permitirá que você adicione a funcionalidade de escrita de NFC ao seu aplicativo Flutter.

Escrita em Tags NFC

Passos para Escrever em Tags NFC

1. Adicionar Dependências ao `pubspec.yaml`

dependencies:

flutter:

sdk: flutter

nfc_manager: latest_version

2. Configurar AndroidManifest.xml

Certifique-se de que as permissões e filtros de intenções estão configurados corretamente (conforme explicado na Aula 02).

3. Código para Escrita em Tags NFC

```
import 'package:flutter/material.dart';

import 'package:nfc_manager/nfc_manager.dart';

void main() {

  runApp(MyApp());

}

class MyApp extends StatelessWidget {

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      home: HomeScreen(),

    );
```

```
}
```

```
}
```

```
class HomeScreen extends StatefulWidget {
```

```
  @override
```

```
  _HomeScreenState createState() => _HomeScreenState();
```

```
}
```

```
class _HomeScreenState extends State<HomeScreen> {
```

```
  @override
```

```
  void initState() {
```

```
    super.initState();
```

```
}
```

```
  Future<void> _writeNfcTag() async {
```

```
    NfcManager.instance.startSession(onDiscovered: (NfcTag tag) async {
```

```
      Ndef? ndef = Ndef.from(tag);
```

```
      if (ndef == null) {
```

```
        print('Tag não é compatível com NDEF');
```

```
        NfcManager.instance.stopSession(errorMessage: 'Tag não é  
compatível com NDEF');
```

```
      return;
```

```
}

NdefMessage message = NdefMessage([

    NdefRecord.createText('Olá, NFC!'),

]);

try {

    await ndef.write(message);

    print('Escrita bem-sucedida');

    NfcManager.instance.stopSession();

} catch (e) {

    print('Falha na escrita: $e');

    NfcManager.instance.stopSession(errorMessage: e.toString());

}

});

}

@override

Widget build(BuildContext context) {

    return Scaffold(

        appBar: AppBar(

            title: Text('Escrita em Tags NFC'),
```

```
),  
  
body: Center(  
  
  child: ElevatedButton(  
  
    onPressed: _writeNfcTag,  
  
    child: Text('Escrever em Tag NFC'),  
  
  ),  
  
),  
  
);  
  
}  
  
}
```

Explicação: Este exemplo mostra como iniciar uma sessão NFC para detectar uma tag e escrever uma mensagem de texto nela. Se a tag não for compatível com NDEF, uma mensagem de erro será exibida.

Esses exemplos e explicações fornecem uma base sólida para iniciantes em Flutter aprenderem a integrar e utilizar a tecnologia NFC em seus aplicativos. 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

Para aprofundar seus conhecimentos em NFC (Near Field Communication) e suas aplicações práticas, recomendamos o seguinte material:

Nome da obra: "NFC Basics"

Autor: Android Developers

Plataforma: Documentação Oficial do Android

Este conteúdo gratuito oferece uma introdução detalhada sobre como implementar e utilizar a tecnologia NFC em dispositivos Android, incluindo exemplos de código e melhores práticas para o desenvolvimento de aplicativos que aproveitam a comunicação por proximidade.

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