

Roteiro Aula Prática



REPOSITÓRIO DE DADOS

ROTEIRO DE AULA PRÁTICA

NOME DA DISCIPLINA: DESENVOLVIMENTO DE APLICAÇÕES ANDROID

Unidade: U1_ INTRODUÇÃO A PROGRAMAÇÃO DE DISPOSITIVOS MÓVEIS

Aula: A4_ FUNDAMENTOS DE UM APP

OBJETIVOS

Definição dos objetivos da aula prática:

Entender como estruturar um projeto Android e configurar suas propriedades fundamentais.

SOLUÇÃO DIGITAL

Computador com acesso à Internet para uso do Google Colab

O Google Colab, ou Colaboratory, é uma plataforma gratuita baseada na nuvem oferecida pelo Google. Ela fornece um ambiente de notebook interativo e colaborativo que permite a criação e execução de código diretamente no navegador, sem a necessidade de configurar ou instalar qualquer software no seu computador.

<https://colab.google/>

PROCEDIMENTOS PRÁTICOS

Procedimento/Atividade nº 1

Atividade proposta:

Criar uma aplicação Android básica que aborde os conceitos fundamentais de um aplicativo, incluindo o arquivo de manifesto, compatibilidade de dispositivos e recursos de um aplicativo.

Procedimentos para a realização da atividade:

1. Configuração Inicial:

- Crie um novo projeto no Android Studio.
- Configure o arquivo `AndroidManifest.xml`.

2. Compatibilidade de Dispositivos:

- Defina a versão mínima e a versão alvo do SDK no arquivo `build.gradle`.

3. Recursos de um App:

- Adicione recursos ao aplicativo, como strings, cores e layouts.

4. Executar o Aplicativo:

- Compile e execute o aplicativo no emulador ou em um dispositivo físico.
- Verifique se o texto "Hello, World!" é exibido no centro da tela.

Dicas

- Certifique-se de que o Android Studio e os SDKs necessários estejam corretamente instalados.
- Utilize a documentação oficial do Android para esclarecer dúvidas sobre a configuração e uso de arquivos de recurso.
- Faça uso das ferramentas de depuração do Android Studio para identificar e corrigir eventuais problemas.

Checklist:

- Configuração Inicial
- Compatibilidade de Dispositivos
- Recursos de um App
- Executar o Aplicativo

RESULTADOS

Resultados de Aprendizagem:

Espera-se que o aluno seja capaz de entender a criação de um aplicativo android básico.

ESTUDANTE, VOCÊ DEVERÁ ENTREGAR

Descrição orientativa sobre a entrega da comprovação da aula prática:

- Para comprovar a realização da atividade, é necessário entregar um arquivo PDF contendo os códigos criados e uma breve explicação de tudo o que foi desenvolvido.

Unidade: U2_ RECURSOS E COMPONENTES VISUAIS - IONIC FRAMEWORK

Aula: A4_ ARMAZENAMENTO DE DADOS EM DISPOSITIVOS MÓVEIS II

OBJETIVOS

Definição dos objetivos da aula prática:

ler e escrever dados em arquivos JSON no armazenamento interno do dispositivo.

SOLUÇÃO DIGITAL

Computador com acesso à Internet para uso do Google Colab

O Google Colab, ou Colaboratory, é uma plataforma gratuita baseada na nuvem oferecida pelo Google. Ela fornece um ambiente de notebook interativo e colaborativo que permite a criação e execução de código diretamente no navegador, sem a necessidade de configurar ou instalar qualquer software no seu computador.

<https://colab.google/>

PROCEDIMENTOS PRÁTICOS

Procedimento/Atividade nº 1

Atividade proposta:

Criar uma aplicação Android que armazena e recupera dados utilizando o formato JSON.

Procedimentos para a realização da atividade:

1. Configuração Inicial:

- Crie um novo projeto no Android Studio.
- Configure o arquivo `AndroidManifest.xml`.

2. Criação da Interface do Usuário:

- Adicione elementos de interface para inserir e exibir dados.

3. Armazenamento e Recuperação de Dados:

- Implemente a lógica para armazenar dados em formato JSON no armazenamento interno.
- Implemente a lógica para recuperar e exibir esses dados.
- Implemente os métodos `saveData` e `loadData` na `MainActivity` para armazenar e recuperar dados em formato JSON.

4. Executar o Aplicativo:

- Compile e execute o aplicativo no emulador ou em um dispositivo físico.
- Verifique se os dados inseridos são salvos e carregados corretamente.

Dicas

- Utilize o Logcat no Android Studio para depuração.
- Certifique-se de conceder permissões de leitura e escrita no armazenamento, se necessário.
- Familiarize-se com o uso de JSON no Android, pois é um formato comum para armazenamento de dados leves.

Checklist:

- Configuração Inicial
- Criação da Interface do Usuário
- Armazenamento e Recuperação de Dados
- Executar o Aplicativo

RESULTADOS

Resultados de Aprendizagem:

Espera-se que os alunos entendam como utilizar o armazenamento interno e o formato JSON para persistência de dados em aplicativos Android, proporcionando uma base sólida para desenvolvimentos futuros.

ESTUDANTE, VOCÊ DEVERÁ ENTREGAR

Descrição orientativa sobre a entrega da comprovação da aula prática:

Para comprovar a realização da atividade, é necessário entregar

- Um arquivo .zip do projeto todo do aplicativo desenvolvido
- Um arquivo PDF contendo prints que comprovem a execução do código desenvolvido de acordo com os objetivos do roteiro e uma breve explicação de tudo o que foi desenvolvido.

Unidade: U2_ RECURSOS E COMPONENTES VISUAIS - IONIC FRAMEWORK

Aula: A4_ ARMAZENAMENTO DE DADOS EM DISPOSITIVOS MÓVEIS II

Tempo previsto de execução de aula prática: 5h

OBJETIVOS

Definição dos objetivos da aula prática:

ler e escrever dados em arquivos JSON no armazenamento interno do dispositivo.

INFRAESTRUTURA

Instalações – Materiais de consumo – Equipamentos:

NOME DO LABORATÓRIO

- Laboratório de Informática

Equipamentos:

- Computador
- ~ 1 unid. 1 aluno

SOLUÇÃO DIGITAL Infraestrutura mínima necessária para execução.

Computador com acesso à Internet para uso do Google Colab

O Google Colab, ou Colaboratory, é uma plataforma gratuita baseada na nuvem oferecida pelo Google. Ela fornece um ambiente de notebook interativo e colaborativo que permite a criação e execução de código diretamente no navegador, sem a necessidade de configurar ou instalar qualquer software no seu computador.

<https://colab.google/>

PROCEDIMENTOS PRÁTICOS

Procedimento/Atividade nº 1

Atividade proposta:

Criar uma aplicação Android que armazena e recupera dados utilizando o formato JSON.

Procedimentos para a realização da atividade:

1. Configuração Inicial:

- Crie um novo projeto no Android Studio.
- Configure o arquivo AndroidManifest.xml.

2. Criação da Interface do Usuário:

- Adicione elementos de interface para inserir e exibir dados.

3. Armazenamento e Recuperação de Dados:

- Implemente a lógica para armazenar dados em formato JSON no armazenamento interno.
- Implemente a lógica para recuperar e exibir esses dados.
- Implemente os métodos `saveData` e `loadData` na `MainActivity` para armazenar e recuperar dados em formato JSON.

4. Executar o Aplicativo:

- Compile e execute o aplicativo no emulador ou em um dispositivo físico.
- Verifique se os dados inseridos são salvos e carregados corretamente.

Dicas

- Utilize o Logcat no Android Studio para depuração.
- Certifique-se de conceder permissões de leitura e escrita no armazenamento, se necessário.
- Familiarize-se com o uso de JSON no Android, pois é um formato comum para armazenamento de dados leves.

Checklist:

- Configuração Inicial
- Criação da Interface do Usuário
- Armazenamento e Recuperação de Dados
- Executar o Aplicativo

RESULTADOS

Resultados de Aprendizagem:

Espera-se que os alunos entendam como utilizar o armazenamento interno e o formato JSON para persistência de dados em aplicativos Android, proporcionando uma base sólida para desenvolvimentos futuros.

ESTUDANTE, VOCÊ DEVERÁ ENTREGAR

Descrição orientativa sobre a entrega da comprovação da aula prática:

Para comprovar a realização da atividade, é necessário entregar

- Um arquivo `.zip` do projeto todo do aplicativo desenvolvido

- Um arquivo PDF contendo prints que comprovem a execução do código desenvolvido de acordo com os objetivos do roteiro e uma breve explicação de tudo o que foi desenvolvido.

Unidade: U3_ RECURSOS DE HARDWARE PARA DISPOSITIVOS MÓVEIS

Aula: A4_ TESTANDO O APP

Tempo previsto de execução de aula prática: 5h

OBJETIVOS

Definição dos objetivos da aula prática:

Aprender a utilizar o Logcat para detectar e corrigir problemas comuns em uma aplicação Android.

INFRAESTRUTURA

Instalações – Materiais de consumo – Equipamentos:

NOME DO LABORATÓRIO

- Laboratório de Informática

Equipamentos:

- Computador
- ~ 1 unid. 1 aluno

SOLUÇÃO DIGITAL

Infraestrutura mínima necessária para execução.

Computador com acesso à Internet para uso do Google Colab

O Google Colab, ou Colaboratory, é uma plataforma gratuita baseada na nuvem oferecida pelo Google. Ela fornece um ambiente de notebook interativo e colaborativo que permite a criação e execução de código diretamente no navegador, sem a necessidade de configurar ou instalar qualquer software no seu computador.

<https://colab.google/>

PROCEDIMENTOS PRÁTICOS

Procedimento/Atividade nº 1

Atividade proposta:

Detecção e Correção de Problemas no Desenvolvimento Android com Logcat. Neste roteiro você irá reproduzir o passo a passo a seguir para entender como funciona o logcat. Em seguida, irá complementar a atividade com dois desafios.

Procedimentos para a realização da atividade:

Parte 1: Configuração do Ambiente

1. Abrir o Android Studio:

- Certifique-se de que o Android Studio está instalado e configurado corretamente.

2. Criar ou abrir um projeto Android:

- Crie um projeto Android ou abra um projeto existente que você tenha.

3. Adicionar código problemático:

- Insira deliberadamente alguns problemas no código para fins de detecção e correção. Por exemplo:

```
// MainActivity.java
package com.example.logcatdemo;

import android.os.Bundle;
import android.util.Log;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Código problemático
        String text = null;
        Log.d("MainActivity", text.toString()); // Isto causará uma NullPointerException
    }
}
```

Parte 2: Uso do Logcat

4. Executar a aplicação:

- Execute a aplicação no emulador ou em um dispositivo físico.

5. Abrir o Logcat:

- No Android Studio, abra a janela do Logcat. Você pode fazer isso através da aba "Logcat" na parte inferior da interface do Android Studio.

6. Filtrar mensagens de log:

- Use o filtro do Logcat para mostrar apenas as mensagens relevantes. Por exemplo, você pode filtrar pelo tag "MainActivity" para ver apenas as mensagens relacionadas a essa Activity.

```
D/MainActivity: null
```

Parte 3: Detecção e Correção de Problemas

7. Analisar mensagens de erro:

- No Logcat, você verá uma exceção `NullPointerException`. Analise a mensagem de erro e o stack trace para identificar a linha de código problemática.

```
E/AndroidRuntime: FATAL EXCEPTION: main
```

```
Process: com.example.logcatdemo, PID: 1234
```

```
java.lang.NullPointerException: Attempt to invoke virtual method 'java.lang.String  
java.lang.Object.toString()' on a null object reference
```

```
at com.example.logcatdemo.MainActivity.onCreate(MainActivity.java:16)
```

8. Corrigir o problema:

- Corrija o problema no código. Por exemplo, adicione uma verificação para garantir que `text` não seja nulo antes de tentar acessar seu método `toString`.

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {
```

```
    super.onCreate(savedInstanceState);
```

```
    setContentView(R.layout.activity_main);
```

```
    // Código corrigido
```

```
    String text = null;
```

```
if (text != null) {  
    Log.d("MainActivity", text.toString());  
} else {  
    Log.d("MainActivity", "Text is null");  
}  
}
```

9. Executar novamente a aplicação:

- Execute a aplicação novamente para garantir que o problema foi resolvido. Verifique o Logcat para confirmar que a mensagem de erro não aparece mais e que as mensagens de log são exibidas corretamente.

Parte 4: Boas Práticas

10. Adicionar logs para depuração:

- Adicione logs em pontos críticos do código para facilitar a detecção de problemas no futuro. Use diferentes níveis de log (e.g., `Log.v`, `Log.d`, `Log.i`, `Log.w`, `Log.e`) conforme apropriado.

```
Log.v("MainActivity", "Verbose log");  
Log.d("MainActivity", "Debug log");  
Log.i("MainActivity", "Info log");  
Log.w("MainActivity", "Warning log");  
Log.e("MainActivity", "Error log");
```

11. Revisar e limpar logs:

- Certifique-se de revisar e remover logs desnecessários antes de liberar a aplicação para produção. Mantenha apenas os logs que são úteis para manutenção e suporte.

Após este exemplo prático de como usar o Logcat, você tem dois desafios:

1. Introduza outros tipos de erros (e.g., `ArrayIndexOutOfBoundsException`, `ClassCastException`) e use o Logcat para detectá-los e corrigi-los.
2. Crie logs personalizados e experimente diferentes níveis de log para entender como eles podem ser usados em diferentes cenários de depuração.

Checklist:

- Configuração do ambiente
- Uso do logcat
- Detecção e Correção de Problemas
- Resolução dos desafios

RESULTADOS

Resultados de Aprendizagem:

Espera-se que os alunos entendam como praticar a detecção e correção de problemas em uma aplicação Android utilizando o Logcat e a importância de adicionar logs apropriados para facilitar a depuração e a manutenção do código.

ESTUDANTE, VOCÊ DEVERÁ ENTREGAR

Descrição orientativa sobre a entrega da comprovação da aula prática:

Para comprovar a realização da atividade, é necessário entregar

- Um arquivo .zip do projeto todo do aplicativo desenvolvido
- Um arquivo PDF contendo prints que comprovem a execução do código desenvolvido de acordo com os objetivos do roteiro
- Um explicação de tudo o que foi desenvolvido, e o passo a passo realizado para resolver os desafios.

Unidade: U4_ INSTALAÇÃO DE RECURSOS PARA UTILIZAÇÃO DE BANCO DE DADOS

Aula: A4_ COMPONENTES IONIC ENTRADA DE DADOS

Tempo previsto de execução de aula prática: 5h

OBJETIVOS

Definição dos objetivos da aula prática:

Compreender o desenvolvimento de aplicações android com Ionic que permite a entrada e o armazenamento de dados localmente no dispositivo.

INFRAESTRUTURA

Instalações – Materiais de consumo – Equipamentos:

NOME DO LABORATÓRIO

- Laboratório de Informática

Equipamentos:

- Computador
- ~ 1 unid. 1 aluno

SOLUÇÃO DIGITAL Infraestrutura mínima necessária para execução.

- Node.js instalado.
- Ionic CLI instalado (você pode instalar usando `npm install -g @ionic/cli`).
- Editor de código (VSCode recomendado).
- Android Studio (para emulação e testes).

PROCEDIMENTOS PRÁTICOS

Procedimento/Atividade nº 1

Atividade proposta:

Criar uma aplicação simples usando Ionic que permitirá ao usuário inserir dados através de um formulário e armazená-los localmente no dispositivo.

Procedimentos para a realização da atividade:

Passo 1: Criação do Projeto Ionic

1. Abra seu terminal ou prompt de comando.
2. Navegue até o diretório onde deseja criar o projeto.
3. Crie um novo projeto Ionic usando o comando:

```
ionic start DadosApp blank
```

4. Entre no diretório do projeto:

```
cd DadosApp
```

Passo 2: Configuração da Página de Entrada de Dados

1. Abra o projeto no seu editor de código.
2. Navegue até `src/app/home/home.page.html` e substitua o conteúdo pelo seguinte código:

```

html
<ion-header>
  <ion-toolbar>
    <ion-title>
      Entrada de Dados
    </ion-title>
  </ion-toolbar>
</ion-header>

<ion-content>
  <form (ngSubmit)="saveData()">
    <ion-item>
      <ion-label position="floating">Nome</ion-label>
      <ion-input [(ngModel)]="data.name" name="name" required></ion-input>
    </ion-item>
    <ion-item>
      <ion-label position="floating">Idade</ion-label>
      <ion-input type="number" [(ngModel)]="data.age" name="age" required></ion-input>
    </ion-item>
    <ion-button expand="full" type="submit">Salvar</ion-button>
  </form>
</ion-content>

```

3. Em seguida, abra `src/app/home/home.page.ts` e substitua o conteúdo pelo seguinte código:

```

import { Component } from '@angular/core';
import { Storage } from '@ionic/storage-angular';

@Component({
  selector: 'app-home',
  templateUrl: 'home.page.html',
  styleUrls: ['home.page.scss'],
})
export class HomePage {
  data = {
    name: '',
    age: null
  };
}

```

```

constructor(private storage: Storage) {}

async ngOnInit() {
  await this.storage.create();
}

async saveData() {
  await this.storage.set('userData', this.data);
  console.log('Dados salvos:', this.data);
}
}

```

Passo 3: Configuração do Ionic Storage

1. Instale o pacote Ionic Storage:

```
npm install @ionic/storage-angular
```

2. Configure o Ionic Storage em `src/app/app.module.ts`:

```

import { IonicStorageModule } from '@ionic/storage-angular';

@NgModule({
  declarations: [AppComponent],
  entryComponents: [],
  imports: [BrowserModule, IonicModule.forRoot(), AppRoutingModule,
IonicStorageModule.forRoot()],
  providers: [{ provide: RouteReuseStrategy, useClass: IonicRouteStrategy }],
  bootstrap: [AppComponent],
})
export class AppModule {}

```

Passo 4: Testando a Aplicação

1. Execute a aplicação no seu dispositivo Android ou emulador:

ionic capacitor run android

2. Insira os dados no formulário e pressione o botão "Salvar". Verifique o console do navegador para garantir que os dados foram armazenados corretamente.

Passo 5: Modificação do aplicativo:

- Além de permitir a entrada e o armazenamento de dados, agora você deverá implementar a funcionalidade de listar os dados salvos na tela inicial da aplicação.

Checklist:

- Criação do Projeto Ionic
- Configuração da Página de Entrada de Dados
- Configuração do Ionic Storage
- Testando a Aplicação
- Modificação do aplicativo

RESULTADOS

Resultados de Aprendizagem:

Espera-se que os alunos entendam como desenvolver uma aplicação Android com Ionic que permite a entrada e armazenamento local de dados.

ESTUDANTE, VOCÊ DEVERÁ ENTREGAR

Descrição orientativa sobre a entrega da comprovação da aula prática:

Para comprovar a realização da atividade, é necessário entregar

- Um arquivo .zip do projeto todo do aplicativo desenvolvido
- Um arquivo PDF contendo prints que comprovem a execução do código desenvolvido de acordo com os objetivos do roteiro. Além de uma explicação de tudo o que foi desenvolvido.