Seminário 1 — Dart e Fundamentos Flutter

Objetivo do encontro

Dar ao estudante a base mínima de **Dart** para conseguir acompanhar e praticar com tranquilidade os **Fundamentos de Flutter** no mesmo seminário.

Estrutura sugerida

1. Microtarefas de Dart (DartPad/CLI).
2.
   - **Parte A — Dart** : linguagem, null-safety e async.
   - **Parte B — Flutter**: árvore de widgets, layout, navegação e tema.
3. Checklist + encaminhamentos.

PARTE 1 — Curadoria com IA

Caderno de Pesquisa com 12 seções curtas (4–6 frases + 1–2 exemplos + 3 erros comuns + mini-checklist cada):

Seções 1-4

1. O que é Dart (tipagem estática com inferência, var/final/const)
2. **Null safety** (?, !, ??, ??=, late, parâmetros nomeados required)
3. **Funções e parâmetros** (posicionais/nomeados; arrow; closures)
4. **Coleções** (List/Set/Map; *spread* ...; map/where/fold; for/if em literais)

Seções 5-8

1. **POO em Dart** (classes, construtores nomeados/factory, get/set, ==/hashCode)
2. **Generics** (List<T>, Map<K,V>)
3. **Enums aprimorados** e **extensions** (métodos utilitários)
4. **Mixins/abstract/implements** (quando usar; diferença de herança/implementação)

Seções 9-12

1. **Assíncrono** (Future, async/await, try/catch, noções de Stream)
2. **Erros/exceções** (criar, lançar e capturar)
3. **Fundamentos Flutter** (Stateless/Stateful, build)
4. **Layout/Navegação/Tema** (Row/Column/Expanded; push/pop; ThemeData)

Prompt curto para a IA (cole na sua ferramenta):

> "Produza um caderno conciso com as 12 seções acima. Em cada seção: (a) defina; (b) 1–2 exemplos mínimos; (c) 3 erros comuns; (d) mini-checklist. Use exemplos que serão reutilizados no Flutter (lista de produtos, contador, detalhes por id)."

PARTE 2 — Mapa Conceitual

Nó central: "Dart + Flutter — Fundamentos para o 1º App" **Ramos:**

Dart

Tipos/Null Safety → Funções → Coleções → POO/Generics → Async/Future

Flutter

Widgets (Stateless/Stateful) → Layout (Row/Column/Expanded, ListView) → Navegação (push/pop, rotas) → Tema

PARTE 3 — Quiz/Flashcards

Total: 40–60 itens. Sugestão de distribuição inicial: **Dart (20–30)** + **Flutter (20–30)**. Sempre com **gabarito, justificativa e nível**.

Amostra (Dart):

1. (Básico) final vs const: const é avaliado em tempo de compilação (Correta)
2. (Básico) O operador que fornece valor padrão quando nulo: ?? (Correta)
3. (Inter.) Em parâmetros **nomeados**, como tornar obrigatório? required (Correta)
4. (Inter.) Converter num para double de forma segura: (valor as num).toDouble() (Correta)
5. (Avanç.) Melhor forma de aguardar I/O e tratar erro: await + try/catch (Correta)

Amostra (Flutter):

1. **Nível:** Básico — **Tipo:** Múltipla escolha **Pergunta:** Quando usar um StatelessWidget? A) Quando a UI nunca muda após o build **(Gabarito)** B) Quando há animação contínua C) Quando consome uma Stream que atualiza a UI D) Quando precisa armazenar um contador local **Justificativa:** StatelessWidget serve para UI imutável; estados/animações exigem StatefulWidget ou gerenciadores de estado.
2. **Nível:** Básico — **Tipo:** Múltipla escolha **Pergunta:** Qual método dispara o redesenho em StatefulWidget? A) didUpdateWidget B) setState **(Gabarito)** C) initState D) build **Justificativa:** setState notifica o framework da mudança e agenda novo build().
3. **Nível:** Básico — **Tipo:** Múltipla escolha **Pergunta:** Expanded funciona corretamente quando o pai é: A) Stack B) Row, Column ou Flex **(Gabarito)** C) ListView D) Center **Justificativa:** Expanded depende de um contêiner flexível (Flex) para calcular espaço.
4. **Nível:** Intermediário — **Tipo:** Múltipla escolha **Pergunta:** Para ir a outra tela **mantendo** a atual na pilha, use: A) Navigator.pushReplacement B) Navigator.push **(Gabarito)** C) Navigator.pushNamedAndRemoveUntil D) Navigator.popAndPushNamed **Justificativa:** push empilha a nova rota sem remover a anterior.
5. **Nível:** Intermediário — **Tipo:** Múltipla escolha **Pergunta:** Melhor forma de passar dados simples para uma rota nomeada é: A) Variável global B) Navigator.pushNamed(context, '/details', arguments: obj) **(Gabarito)** C) setState com valor compartilhado D) InheritedWidget sempre **Justificativa:** arguments/onGenerateRoute são o caminho idiomático para parâmetros de rota.
6. **Nível:** Intermediário — **Tipo:** Múltipla escolha **Pergunta:** Erro comum ao usar ListView dentro de Column: A) Usar ListTile B) Falta de altura limitada (ex.: não envolver com Expanded/SizedBox) **(Gabarito)** C) Usar itemBuilder D) Ter menos de 3 itens **Justificativa:** Sem restrição de altura, ocorre overflow/layout indefinido.
7. **Nível:** Intermediário — **Tipo:** Múltipla escolha **Pergunta:** Onde declarar **tema global** da aplicação? A) Em cada Widget manualmente B) Dentro de qualquer Scaffold C) Em MaterialApp(theme: ...) **(Gabarito)** D) Em main() via print **Justificativa:** O tema global é fornecido por MaterialApp.
8. **Nível:** Avançado — **Tipo:** Múltipla escolha **Pergunta:** Uso correto de setState é envolver: A) Toda a árvore de widgets B) Apenas a mutação que altera a UI **(Gabarito)** C) Chamadas de rede longas D) Tudo que não muda o estado **Justificativa:** Limitar o escopo reduz *rebuilds* e efeitos colaterais.
9. **Nível:** Avançado — **Tipo:** Verdadeiro/Falso **Afirmação:** Evitar *rebuilds* desnecessários melhora performance e legibilidade. **Gabarito: Verdadeiro Justificativa:** Menos trabalho de renderização e UI mais previsível.
10. **Nível:** Avançado — **Tipo:** Múltipla escolha **Pergunta:** Quando considerar Provider/Riverpod (em vez de só setState)? A) Somente para estilizar textos B) Quando o estado é efêmero e local a um único widget C) Quando o estado cruza múltiplas telas/árvores e precisa ser compartilhado **(Gabarito)** D) Nunca; setState sempre basta **Justificativa:** Estado compartilhado/global e reatividade entre múltiplos ramos pedem gerenciadores dedicados.

PARTE 4 — Slides do Seminário

Bloco A — Dart (6–7 slides)

1. Por que Dart (tipagem + *hot reload* no Flutter)
2. Tipos, var/final/const, **null safety**
3. Funções (nomeados/posicionais), coleções e operadores úteis (..., ?., ??)
4. POO essencial (classe, construtores, fromMap/toMap)
5. Assíncrono (Future, async/await)
6. Mini-desafio: transformar lista de mapas em modelos e filtrar/exibir
7. Checklist "Dart pronto para Flutter"

Bloco B — Flutter (8–10 slides)

1. Por que Flutter (multiplataforma + produtividade + hot reload)
2. Árvore de widgets (composição > herança)
3. Stateless vs Stateful (quando usar cada um)
4. Ciclo de vida do `State` (initState → build → setState → dispose)
5. Layout essencial (Row/Column/Flex/Expanded, espaçamento e overflow)
6. Listas e Cards (ListView.builder, ListTile, Card)
7. Navegação (push/pop, rotas nomeadas, passagem de argumentos)
8. Tema e estilos (ThemeData, ColorScheme, Theme.of)
9. Boas práticas iniciais (estrutura por features, `const`, keys, lints, evitar rebuilds)
10. Encerramento + demo guiada (lista → detalhes, contador com `setState`, troca rápida de tema)

PARTE 5 — Exercícios práticos

1

Ex. 1 (Dart, 10 min): modelo e transformação

```
class Product {
  final String id;
  final String name;
  final double price;

  const Product({
    required this.id,
    required this.name,
    required this.price
  });

  factory Product.fromMap(Map m) => Product(
    id: m['id'] as String,
    name: m['name'] as String,
    price: (m['price'] as num).toDouble(),
  );

  Map toMap() => {
    'id': id,
    'name': name,
    'price': price
  };
}

extension Currency on num {
  String asBRL() => 'R\$ ${toStringAsFixed(2)}';
}
```

2

Ex. 2 (Dart, 10 min): assíncrono básico

```
Future> loadProducts() async {
  await Future.delayed(const Duration(milliseconds: 400));
  return [
    Product(id: '1', name: 'Pizza', price: 29.9),
    Product(id: '2', name: 'Suco', price: 8.0),
  ];
}
```

3

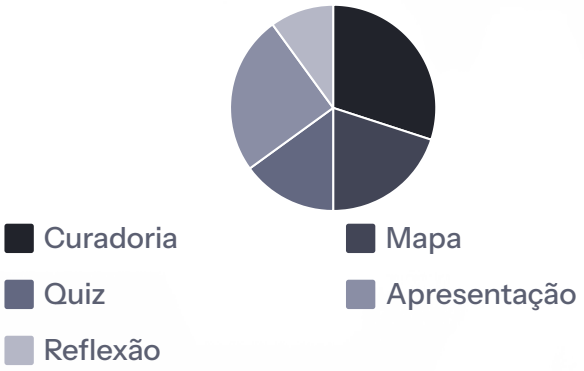
Ex. 3 (Flutter, 15–20 min): reutilizar no UI

- ListView.builder exibindo name e price.asBRL()
- Botão que incrementa um contador com setState
- Navegação simples para DetailsPage(id)

Entregáveis

- **Caderno de Pesquisa** (agora com blocos Dart + Flutter)
- **Mapa Conceitual** (inclui ramo Dart)
- **Quiz/Flashcards** (com itens de Dart e Flutter)
- **Slides** (Bloco A Dart + Bloco B Flutter)
- **Reflexão** (½–1 pág.) — "O que em Dart mais me ajudou no Flutter hoje?"

Rubrica (inalterada)



Checklists rápidos

Dart

- [ ] dart --version ok (ou DartPad funcionando)
- [ ] Entendo var/final/const
- [ ] Sei usar ?, ??, required
- [ ] Sei criar class, factory fromMap, toMap
- [ ] Sei escrever async/await com try/catch

Flutter

- [ ] Stateless vs Stateful
- [ ] Row/Column/Expanded sem *overflow*
- [ ] ListView.builder com dados do modelo
- [ ] Navigator.push com argumento
- [ ] ThemeData aplicado no MaterialApp