

**UNIVERSIDADE MAURÍCIO DE NASSAU**

**FLUTTER DESCOMPLICADO**

Tudo o que você precisa saber para começar

Recife - PE

2024

## Sumário

<b>Equipe .....</b>	<b>3</b>
<b>Cronograma .....</b>	<b>4</b>
<b>Semana 1 .....</b>	<b>4</b>
<b>Semana 2 .....</b>	<b>4</b>
<b>Semana 3 .....</b>	<b>4</b>
<b>Semana 4 .....</b>	<b>4</b>
<b>Semana 5 .....</b>	<b>4</b>
<b>Pesquisa .....</b>	<b>5</b>
<b>Flutter .....</b>	<b>5</b>
<b>Linguagem Dart .....</b>	<b>5</b>
<b>Como o Flutter funciona? .....</b>	<b>5</b>
<b>Quais são as vantagens do Flutter? .....</b>	<b>5</b>
<b>E quais são as desvantagens do Flutter? .....</b>	<b>6</b>
<b>Quem está usando Flutter? .....</b>	<b>7</b>
<b>Funcionalidades do Flutter .....</b>	<b>7</b>
<b>Referências .....</b>	<b>9</b>
<b>Agradecimentos .....</b>	<b>10</b>

## **Equipe**

Laysa Freitas

Luísa Silva

Taciane Cunha

Williame Gabriel

# Cronograma

## **Semana 1**

Configuração do Ambiente de Desenvolvimento

Introdução ao Flutter e Dart

## **Semana 2**

Design da Interface do Usuário (UI)

Implementação das Funcionalidades Básicas

## **Semana 3**

Desenvolvimento da Lógica de Contador

Armazenamento e Histórico

## **Semana 4**

Testes e Debugging

Ajustes Finais e Documentação

## **Semana 5**

Revisão Final e Apresentação

# Pesquisa

## Flutter

O Flutter é um framework desenvolvido pelo Google na linguagem Dart para a criação de aplicativos multiplataforma, web e mobile e, em breve, desktop. Diferentemente de outras soluções, como Ionic e React Native, o Flutter não é executado em uma WebView (Ionic) e nem utiliza componentes nativos através de uma "ponte" (React Native). Ao invés disso gera código verdadeiramente nativo, tanto para Android quanto para iOS.

## Linguagem Dart

Dart é a linguagem do Flutter e o ponto de partida para os seus estudos. É possível aprender Flutter e ao longo do caminho ir adquirindo familiaridade com o Dart, principalmente se você conhece JavaScript, dada a proximidade da sintaxe dessas duas linguagens. Contudo, ao começar do início com o Dart você estará muito melhor preparado para entender o Flutter e tirar proveito dos seus recursos.

## Como o Flutter funciona?

O Flutter, similar à engines utilizadas em jogos como a OpenGL e renderiza os elementos do aplicativo (chamados de Widgets) diretamente em um Canvas. Dessa forma, consegue executar o mesmo aplicativo tanto em aplicativos Android quanto em aplicativos iOS. Além disso, por conta dessa solução, também é capaz de executar animações em até 120 fps (frames por segundo).

## Quais são as vantagens do Flutter?

Agora que você já sabe o que é Flutter, vamos falar sobre as vantagens de optar por ele?

Esse framework se destaca entre as outras tecnologias, principalmente por ser baseado em Dart, uma linguagem recente, mas que está se tornando cada vez mais popular, e que é fácil de aprender.

- Agilidade no desenvolvimento

Por ter muitos widgets próprios, os aplicativos desenvolvidos em Flutter são mais rápidos, já que a sua interface permite rodar animações em 60 fps, além de ser compilado para código nativo.

- Suporte do Google

O Flutter possui um suporte oficial disponibilizado pelo Google, assim como todas as outras ferramentas disponibilizadas pela empresa. Assim, a sua documentação está sempre atualizada e é fácil encontrar as informações necessárias para o desenvolvimento de um projeto.

- Menos código

No Flutter, o desenvolvimento é feito com um código único (na linguagem Dart), o que poupa tempo dos desenvolvedores e torna todo o time mais produtivo.

- Fácil instalação e configuração

O setup de instalação e configuração do Flutter é simples até mesmo em Linux. Com poucos comandos, é possível verificar a existência de problemas e consertá-los. Além disso, o Flutter também torna prática a atualização de plugins, criação de projetos e correção de erros.

## **E quais são as desvantagens do Flutter?**

Apesar de muitos benefícios, o Flutter também traz algumas desvantagens. Saiba quais são!

- É um framework novo

Por ser relativamente novo no mercado, o Flutter ainda oferece menor quantidade de ferramentas do que outros frameworks já consolidados, como Xamarin e React Native.

- Dart não é uma linguagem muito popular

Apesar de ser uma excelente linguagem de programação, Dart não é muito utilizada no mercado. Normalmente, os desenvolvedores preferem trabalhar com outras opções.

- Incompatibilidade de recursos

Como o Flutter é um software do Google, alguns recursos estão disponíveis apenas para Android, o que restringe as opções para desenvolvedores iOS.

## **Quem está usando Flutter?**

Algumas empresas já estão criando cases de sucesso com Flutter. Confira!

Realtor.com

Tencent

The New York Times

Square

Google Assistant

## **Funcionalidades do Flutter**

- Desenvolvimento Multiplataforma:

Aplicações Android e iOS: Criação de aplicativos nativos para ambas as plataformas a partir de uma única base de código.

Web: Suporte para desenvolvimento de aplicações web.

Desktop: Suporte em desenvolvimento para Windows, macOS e Linux (ainda em fase de amadurecimento).

- Widgets Reativos:

Widgets Personalizáveis: Um extenso conjunto de widgets que seguem o Material Design para Android e o estilo Cupertino para iOS.

Construção Declarativa: Interface de usuário é construída de forma declarativa, o que significa que a UI é descrita como uma função do estado.

- Desempenho:

Compilação Nativa: Usa compilação Ahead-of-Time (AOT) para código nativo, resultando em aplicativos de alto desempenho.

Smooth Animations: Capacidade de criar animações e transições suaves e de alta performance.

- Hot Reload:

Atualizações Imediatas: Permite que você veja as mudanças no código imediatamente sem reiniciar o aplicativo, acelerando o processo de desenvolvimento.

#### -Gerenciamento de Estado:

Diversas Abordagens: Suporta várias técnicas e bibliotecas para gerenciamento de estado, como Provider, Riverpod, Bloc, MobX e outros.

#### -Customização e Design:

Design Personalizado: Capacidade de criar e personalizar a interface do usuário de acordo com as necessidades específicas do aplicativo.

Temas e Estilos: Facilita a aplicação de temas e estilos consistentes em toda a aplicação.

#### -Internacionalização e Localização:

Suporte Multilíngue: Facilita a tradução e localização de aplicativos para diferentes idiomas e regiões.

#### -Integração com APIs e Serviços:

Conectividade: Suporte para integração com APIs REST, WebSocket, bancos de dados e outros serviços externos.

#### -Ferramentas e Plugins:

Ecosistema de Plugins: Grande número de plugins e pacotes disponíveis para adicionar funcionalidades como autenticação, integração com dispositivos e muito mais.

#### -Desenvolvimento Ágil:

Arquitetura Flexível: Suporte para vários padrões e práticas de arquitetura, ajudando a criar aplicativos escaláveis e manuteníveis.

#### -Acessibilidade:

Suporte a Acessibilidade: Inclui recursos para tornar os aplicativos acessíveis a pessoas com deficiência.

#### -Testes:

Testes Abrangentes: Suporte para testes unitários, de widget e de integração, facilitando a garantia da qualidade do aplicativo.



## Referências

Alura

## **Agradecimentos**

Expresso à equipe minha mais sincera gratidão ao comprometimento demonstrado ao longo da execução do projeto. A contribuição de cada um foi fundamental.