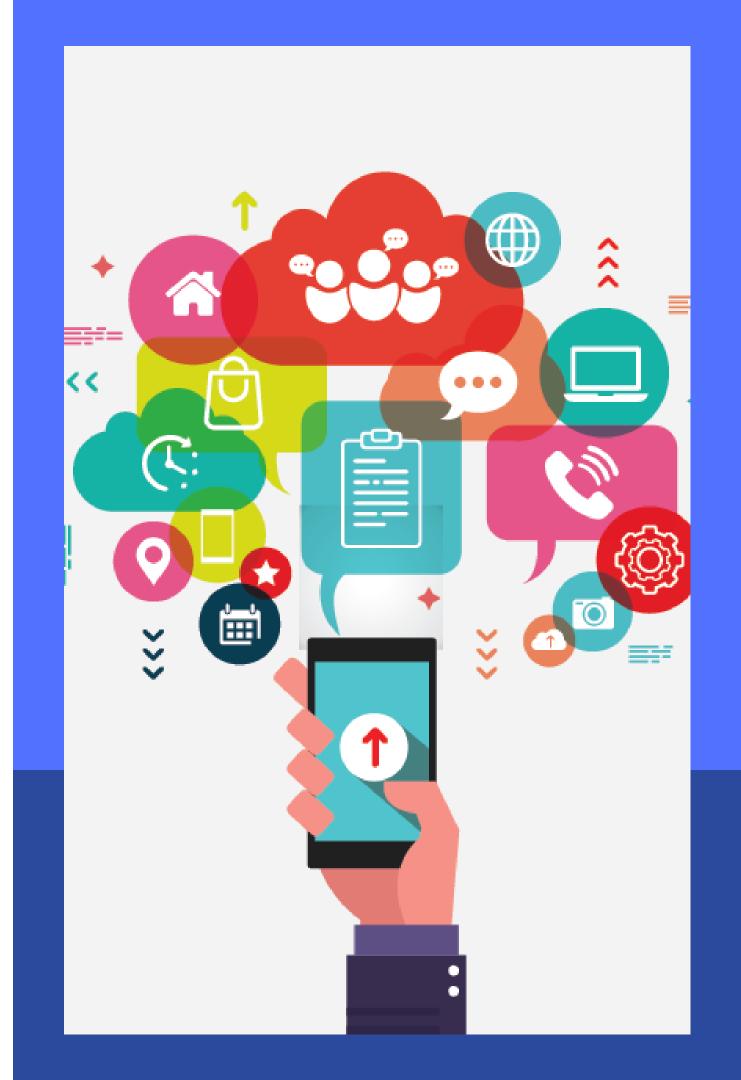
DESENVOLVIMENTO DE APLICAÇÕES MÓVEIS

INTRODUÇÃO

- As aplicações móveis estão em constante crescimento e se tornaram uma parte essencial da nossa vida cotidiana.
- O desenvolvimento de aplicações móveis envolve a criação de software para dispositivos móveis, como smartphones e tablets.
- Nesta apresentação, vamos explorar o que é o desenvolvimento de aplicações móveis e como ele pode impactar nossa vida.



VANTAGENS

ACESSIBILIDADE

As aplicações móveis permitem que os usuários acessem informações e serviços em qualquer lugar e a qualquer momento.

INTEGRAÇÃO

As aplicações móveis podem aproveitar os recursos dos dispositivos, como câmera, GPS, acelerômetro, entre outros.

EXPERIÊNCIA DE USUÁRIO

O desenvolvimento de aplicações móveis permite criar interfaces intuitivas e interativas, proporcionando uma ótima experiência para os usuários.

TECNOLOGIAS PARA DESENVOLVIMENTO

APLICAÇÕES NATIVAS

São desenvolvidas para uma plataforma específica, como iOS (utilizando Swift/Objective-C) ou Android (utilizando Java/Kotlin).

APLICAÇÕES WEB

São acessadas através de um navegador móvel e desenvolvidas utilizando tecnologias web, como HTML, CSS e JavaScript.

APLICAÇÕES HÍBRIDAS

São desenvolvidas utilizando tecnologias web e encapsuladas em um contêiner nativo. Exemplos incluem o uso de frameworks como React Native e Flutter.

DIFERENÇA ENTRE APLICATIVOS NATIVOS E CROSS-PLATFORM

APLICATIVOS NATIVOS

- Aplicativos nativos são desenvolvidos especificamente para uma plataforma específica, como iOS ou Android.
- Eles são escritos em linguagens de programação nativas, como Swift ou Objective-C para iOS, e Java ou Kotlin para Android.
- Aplicativos nativos têm acesso total aos recursos do dispositivo, como câmera, GPS, acelerômetro e notificações.
- Isso permite que os aplicativos nativos ofereçam melhor desempenho e experiência do usuário, aproveitando ao máximo as capacidades do dispositivo.

APLICATIVOS NATIVOS

VANTAGENS

- **Desempenho otimizado:** Aplicativos nativos têm acesso direto aos recursos do dispositivo, proporcionando melhor desempenho e resposta rápida.
- Interface nativa: Os aplicativos nativos seguem as diretrizes de design de cada plataforma, oferecendo uma experiência familiar e consistente para os usuários.
- Acesso total aos recursos: Recursos como câmera, GPS, acelerômetro e notificações podem ser facilmente integrados em aplicativos nativos, proporcionando funcionalidades avançadas.

DESVANTAGENS

- Desenvolvimento separado: Aplicativos nativos exigem o desenvolvimento separado de código para cada plataforma, aumentando o tempo e os custos de desenvolvimento.
- Equipes especializadas: É necessário ter desenvolvedores especializados em cada plataforma, o que pode aumentar a complexidade e os recursos necessários.

APLICATIVOS CROSS-PLATFORM

- Cross-platform mobile permite o desenvolvimento de aplicativos que podem ser executados em várias plataformas, como iOS e Android, usando uma única base de código.
- Frameworks populares, como React Native, Flutter e Xamarin, são amplamente utilizados para o desenvolvimento cross-platform.
- Essas abordagens utilizam linguagens de programação como JavaScript (React Native), Dart (Flutter) ou C# (Xamarin).
- Com o desenvolvimento cross-platform, é possível economizar tempo e recursos, pois uma única equipe pode criar e manter o aplicativo para várias plataformas.

APLICATIVOS CROSS-PLATAFORM

VANTAGENS

- Desenvolvimento mais rápido: Com uma única base de código, é possível desenvolver para várias plataformas simultaneamente, economizando tempo de desenvolvimento.
- Custo efetivo: O desenvolvimento crossplatform requer menos recursos, pois não é necessário contratar equipes separadas para cada plataforma.
- Manutenção simplificada: Atualizações e correções podem ser aplicadas de forma centralizada, beneficiando todas as plataformas simultaneamente.

DESVANTAGENS

- Limitações de desempenho: O desempenho de aplicativos cross-platform pode ser ligeiramente inferior aos aplicativos nativos devido a camadas de abstração e interpretação de código.
- Dependência de frameworks: O uso de frameworks cross-platform pode implicar em limitações ou dependência de recursos específicos do framework.

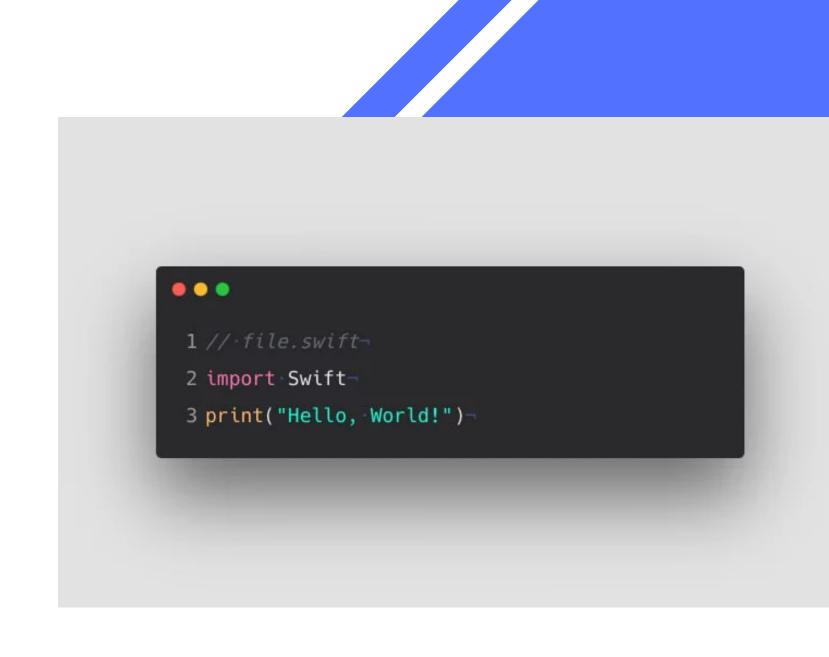
LINGUAGENS DE PROGRAMAÇÃO PARA O DESENVOLVIMENTO DE APLICAÇÕES MÓVEIS

LINGUAGENS DE PROGRAMAÇÃO

- O desenvolvimento de aplicações móveis requer o uso de linguagens de programação específicas para cada plataforma.
- A seguir, vamos explorar algumas das linguagens de programação mais populares para o desenvolvimento de aplicações móveis.

SWIFT

- SWIFT É A LINGUAGEM DE PROGRAMAÇÃO OFICIALMENTE SUPORTADA PELA APPLE PARA O DESENVOLVIMENTO DE APLICATIVOS IOS, MACOS, WATCHOS E TVOS.
- FOI PROJETADA PARA SER SEGURA, RÁPIDA E MODERNA, COMBINANDO RECURSOS DE LINGUAGENS COMO C, OBJECTIVE-C, RUST E OUTRAS.
- SWIFT É CONHECIDA POR SUA SINTAXE SIMPLES E EXPRESSIVA, FACILITANDO A ESCRITA DE CÓDIGO CLARO E LEGÍVEL.



KOTLIN

- KOTLIN É A LINGUAGEM DE PROGRAMAÇÃO OFICIALMENTE SUPORTADA PELO GOOGLE PARA O DESENVOLVIMENTO DE APLICATIVOS ANDROID.
- ELA FOI DESENVOLVIDA PARA SER INTEROPERÁVEL COM JAVA, PERMITINDO QUE DESENVOLVEDORES APROVEITEM O ECOSSISTEMA EXISTENTE.
- KOTLIN É CONHECIDA POR SUA CONCISÃO, SEGURANÇA DE TIPO E RECURSOS AVANÇADOS, COMO A ELIMINAÇÃO DE NULLPOINTEREXCEPTION.

```
fun main() {
    println("Hello, world")
```



- C# É UMA LINGUAGEM DE PROGRAMAÇÃO DA MICROSOFT QUE PODE SER USADA PARA O DESENVOLVIMENTO DE APLICATIVOS MÓVEIS COM O FRAMEWORK XAMARIN.
- XAMARIN PERMITE CRIAR APLICATIVOS NATIVOS PARA IOS, ANDROID E WINDOWS USANDO A LINGUAGEM C# E O ECOSSISTEMA .NET.
- C# É CONHECIDA POR SUA ROBUSTEZ, SEGURANÇA DE TIPO E AMPLA DISPONIBILIDADE DE BIBLIOTECAS E FERRAMENTAS DE DESENVOLVIMENTO.



```
1 //·file.js-
2 alert(·'Hello, World!'·);-
3 -
```

JAVASCRIPT

- JAVASCRIPT É UMA LINGUAGEM AMPLAMENTE UTILIZADA PARA O DESENVOLVIMENTO DE APLICATIVOS MÓVEIS, ESPECIALMENTE COM O USO DE FRAMEWORKS COMO REACT NATIVE E NATIVESCRIPT.
- ESSES FRAMEWORKS PERMITEM QUE OS DESENVOLVEDORES CRIEM APLICATIVOS MÓVEIS USANDO JAVASCRIPT, QUE É ENTÃO TRADUZIDO PARA CÓDIGO NATIVO DURANTE A EXECUÇÃO.
- JAVASCRIPT É UMA LINGUAGEM FLEXÍVEL E PODEROSA, COM UMA VASTA COMUNIDADE E UMA AMPLA GAMA DE BIBLIOTECAS E FRAMEWORKS DISPONÍVEIS.

```
1 // file. is-
2 alert( 'Hello, World!' );
```

DART

- DART É A LINGUAGEM DE PROGRAMAÇÃO USADA PELO FRAMEWORK FLUTTER PARA O DESENVOLVIMENTO DE APLICATIVOS MÓVEIS MULTIPLATAFORMA.
- FLUTTER PERMITE CRIAR APLICATIVOS NATIVOS PARA IOS, ANDROID, WEB E DESKTOP USANDO UMA ÚNICA BASE DE CÓDIGO.
- DART É CONHECIDO POR SUA PERFORMANCE, FACILIDADE DE APRENDIZADO E RECURSOS COMO HOT RELOAD, QUE PERMITEM VISUALIZAR ALTERAÇÕES EM TEMPO REAL DURANTE O DESENVOLVIMENTO.

```
1 // file.dart-
2 void main() {
3 print('Hello, World!');
```

CONCLUSÃO

INTRODUÇÃO

DIFERENÇA ENTRE APLICATIVOS NATIVOS E CROSS-PLATFORM

LINGUAGENS PARA O
DESENVOLVIMENTO DE
APLICAÇÕES



REFERÊNCIAS

- https://edrodrigues.com.br/blog/hello-world-em-20-linguagensde-programacao/
- https://www.teclogica.com.br/desenvolvimento-paradispositivos-moveis-nativo-x-cross-platform/
- https://blog.tecnospeed.com.br/linguagens-mobile/