



Criando um README - Web, Desktop e Mobile



📌 Desenvolvimento de Sistemas: Web, Desktop e Mobile

Este documento tem como objetivo apresentar uma visão geral dos três principais tipos de desenvolvimento de software: **Web, Desktop e Mobile**.

Entender essas diferenças é essencial para escolher a abordagem correta ao iniciar um projeto e também para realizar a manutenção de sistemas já existentes.

Compreender cada ambiente ajuda desenvolvedores e equipes a:

- **otimizar recursos;**
- **melhorar a experiência do usuário;**
- **garantir maior compatibilidade** entre plataformas.

Assim, esse README serve como um guia para aprender quando usar cada tipo de desenvolvimento, e suas vantagens e desvantagens, e as suas principais ferramentas e linguagens.



Desenvolvimento Web

O que é?

O desenvolvimento web é uma área da tecnologia responsável pela criação, codificação e programação de site, aplicativos e seus respectivos elementos de linguagem.

Ele engloba uma série de tarefas, que vão desde a construção de uma página estática simples, como um blog, até a criação de aplicações complexas e dinâmicas, como lojas virtuais, redes sociais e sistemas bancários online.

O que faz o Desenvolvedor Web?

O profissional de Dev ou Web Developer é responsável por **planejar, criar, desenvolver e manter sites e aplicações web**. Sendo assim, esse profissional trabalha com várias linguagens de programação, como HTML, CSS, JavaScript, PHP e outras para dar vida às páginas web conforme as suas necessidades.

Inclusive, além da criação de páginas, o desenvolvedor também pode implementar outros serviços, como soluções de pagamento, segurança, hospedagem e análise.

Desse modo, o desenvolvedor web desenvolve, testa e mantém sites e aplicações web. Isso pode incluir design de interface do usuário, codificação de programa, edição de imagem, gerenciamento de conteúdo e muito mais.

Além disso, o desenvolvedor web normalmente trabalha em equipe com outros profissionais de tecnologia, incluindo designers, engenheiros de software, administradores de sistemas, entre outros, para criar aplicações ricas em recursos que podem ser acessadas pela web.

Por fim, dentro da carreira de desenvolvedor web existem várias especialidades como, por exemplo, **front e back-end**.

📌 Principais linguagens e frameworks

<https://youtu.be/wLEw36KAVfU?si=7Z1ye9CYREreV-21>

- **Linguagens:**

HTML, CSS, JavaScript

- **Frameworks e bibliotecas:**

React, Angular, Vue.js, Node.js



Ferramentas utilizadas

- Código VS Code
- Navegadores (Chrome, Firefox, Edge)
- Git e GitHub para versionamento



Exemplos de projetos

- Comércio eletrônico (Amazon, Mercado Livre)
- Plataformas de redes sociais (Twitter/X, Facebook)
- Sistemas de gestão online (Trello, Notion)

```
    scrollHeight - element.clientHeight + 0.02 ^ w...
    window.scroll(0, scrollHeight);
}
```



Vantagens

- Acessível em qualquer dispositivo com navegador, não requer instalação e não precisa de uma loja de aplicativos;



- Atualizações simples (centralizadas no servidor), e os usuários sempre acessam a versão mais recente sem a necessidade de atualizações de hardware ou software;
- Boa escalabilidade, o que permite que sejam dimensionadas para atender a um grande número de usuários e volumes crescentes de dados;
- Alcance Global, permitindo um público vasto e global, expandindo o mercado e visibilidade da marca;
- A publicação é mais rápida, sem a necessidade de passar por processos de aprovação das lojas de app.

Limitações

- Dependência de conexão com internet, o que pode ser uma barreira se a internet estiver instável ou inexistente;
- Diferenças de compatibilidade entre navegadores;
- Pode ter desempenho inferior aos aplicativos nativos, sendo mais lento em comparação a aplicativos nativos, que são otimizados para dispositivos específicos;
- Vulnerabilidades, os sites podem ser mais suscetíveis a ameaças de segurança, como ataques de scripts maliciosos;
- Fidelização do usuário, a falta de um aplicativo instalado pode resultar em menor fidelização do usuário, já que a experiência pode ser menos envolvente.



Desenvolvimento Desktop

O que é?

O desenvolvimento desktop cria aplicativos que são instalados e executados diretamente em computadores (Windows, Linux, macOS), e não exige um navegador web para funcionar.

Ao contrario dos aplicativos móveis, que priorizam a mobilidade, ou dos aplicativos web, que funcionam em navegadores, os programas desktops são feitos para aproveitar ao máximo o poder de processamento e os recursos do hardware do computador ou seja, é uma aplicação mobile só que para desktop

Linguagens comuns

- C#
- Java
- Python
- C++

IDEs utilizadas

- Estúdio Visual
- Eclipse
- PyCharm
- NetBeans

Exemplos de aplicativos

- Microsoft Office
- Adobe Photoshop
- VLC Media Player



✓ Vantagens

- Melhor desempenho em aplicações pesadas, onde podem usar o poder de processamento, a memória e os recursos do próprio dispositivo;
- Acesso offline garantido, sendo usado mesmo sem conexão com a internet, o que pode tornar ideais para tarefas que exigem disponibilidade contínua;
- Maior controle de hardware, possuem maior facilidade para interagir e integrar-se com periféricos e outros componentes de hardware do computador;
- Segurança e privacidade, os dados podem ser mais seguros, pois são armazenados localmente na máquina do usuário;
- Os requisitos de sistema específicos, tem a capacidade de lidar com tarefas que exigem muito processamento ou uso de telas grandes.

✗ Limitações

- Dependência de plataforma (Windows, Linux, macOS), onde um aplicativo desenvolvido para Windows não funcionará no macOS, por exemplo;
- Atualizações manuais, os usuários precisam baixar e instalar atualizações manualmente, o que pode ser tedioso e um desafio para manter todos os sistemas com os recursos mais recentes;
- Custo de manutenção pode ser aumentado devido à necessidade de espaço de armazenamento, o que pode sobrecarregar o sistema e exigir a modificação do disco rígido;

- mobilidade Reduzida, o acesso fica restrito ao dispositivo onde a aplicação está instalada, exigindo instalação separada para usá-la em outros computadores;
 - O armazenamento é local, o que significa que os arquivos dos usuários ocupam um espaço no dispositivos, o que pode complicar para quem tem pouco armazenamento.
-



Desenvolvimento Mobile

O que é?

É o desenvolvimento de aplicativos para dispositivos móveis (smartphones e tablets). Esse campo abrange todo o ciclo de vida de um app, desde a ideia inicial, design, programação, testes e até o lançamento da aplicação

O objetivo principal do desenvolvimento mobile é fornecer aos usuários uma experiência otimizada e conveniente, permitindo que acessem informações, serviços e entretenimento diretamente de seus dispositivos e de suas mãos

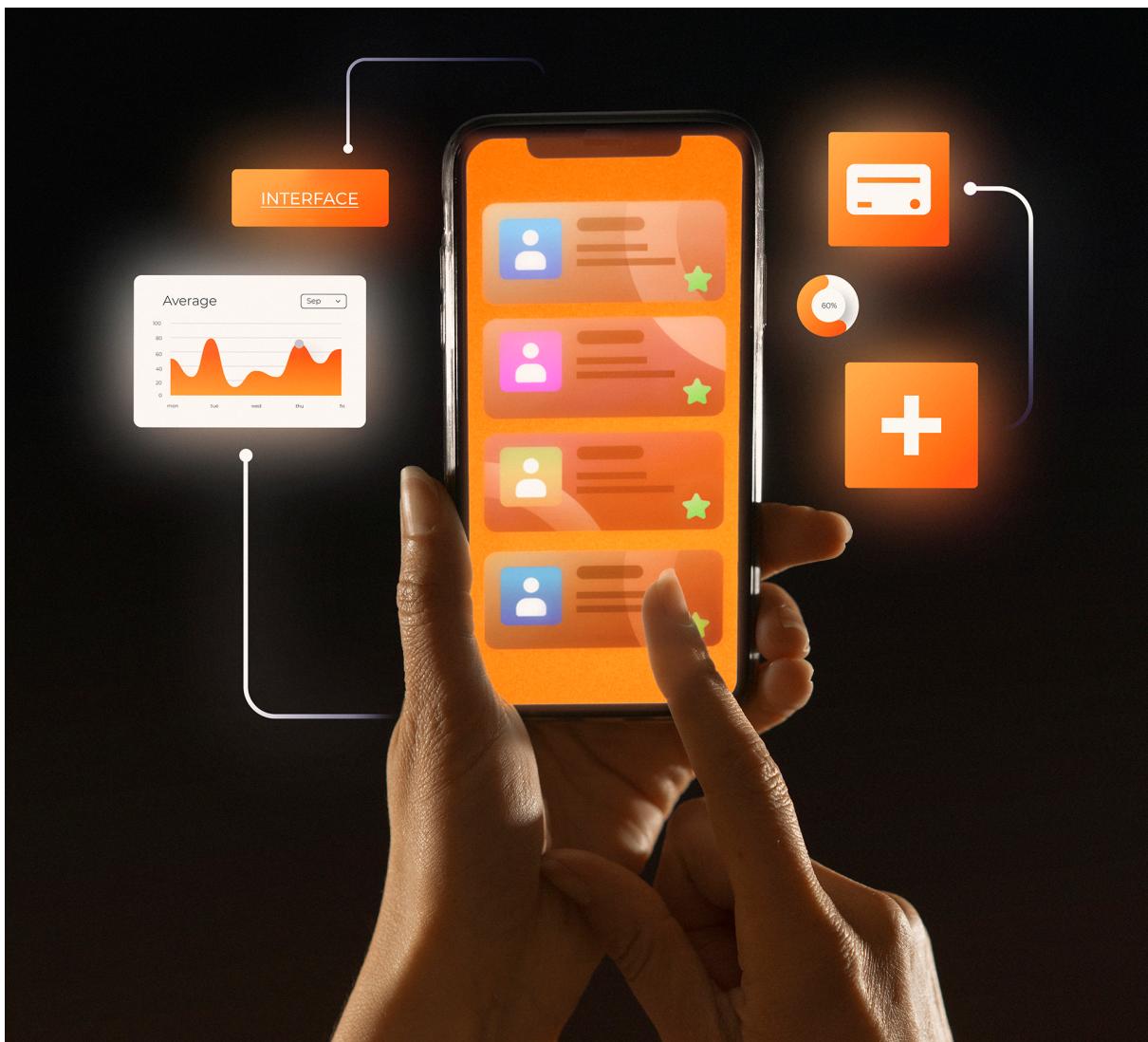
Esses aplicativos podem ser **nativos, híbridos ou PWAs**.

Quais são os principais desafios do desenvolvimento mobile?

Este mercado oferece uma série de possibilidades e tendências, mas enfrenta desafios que se colocam frente aos desenvolvedores e à tecnologia. Conheça quais são eles:

- **Fragmentação de plataformas:** desenvolver para Android, iOS e outros sistemas envolve lidar com diferentes sistemas operacionais, versões e tamanhos de tela. Isso cria complexidade e exige testes abrangentes para os desenvolvedores;
- **Experiência do usuário:** os usuários móveis têm expectativas elevadas quanto à usabilidade e desempenho de aplicações, o que exige a criação de uma experiência de usuário fluida e responsiva;

- **Segurança:** a segurança dos dados do usuário é crítica e exige a implementação de medidas rigorosas para proteger informações sensíveis. À medida que a tecnologia avança, surgem não só novas soluções, mas também novos desafios, brechas e riscos.
- **Ciclos de lançamento rápidos:** o mercado móvel evolui rapidamente e passa por constantes atualizações de sistemas que, por sua vez, também decorrem de novas necessidades e tecnologias. Por isso, desafia o desenvolvedor a ter atenção a este aspecto e a acompanhar essas alterações recorrentes;
- **Particularidades dos dispositivos móveis:** outros desafios do desenvolvimento *mobile* se referem às próprias particularidades dos dispositivos móveis, como instabilidade de conexão com a internet (que depende de vários fatores) e hardwares com memória limitada.



!? Diferenças entre os tipos

- **Nativos:**

feitos para Android (Kotlin/Java) ou iOS (Swift), melhor desempenho, mas mais caro.

- **Híbridos:**

um único código para várias plataformas (React Native, Ionic), equilíbrio entre custo e desempenho.

- **PWAs:**

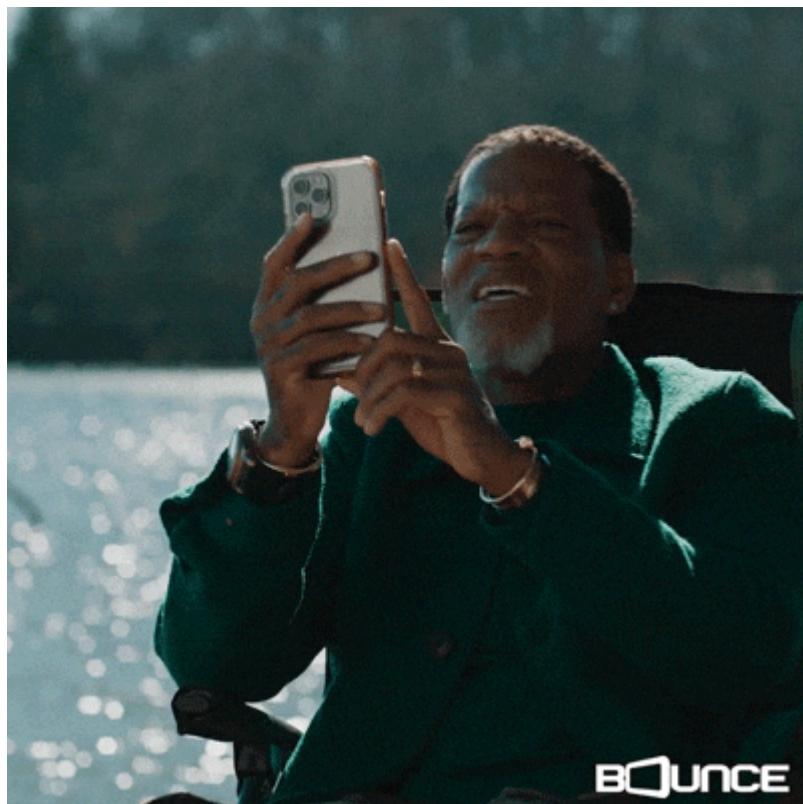
aplicativos progressivos que rodam no navegador, mas funcionam como apps, mais acessível, mas com limitações em recursos avançados.

Ferramentas/Estruturas

- Estúdio Android
- Xcode
- Flutter
- React Native

Exemplos de aplicativos

- WhatsApp
- Instagram
- Uber
- Spotify



Vantagens

- Melhor experiência de usuário (rodam diretamente no sistema) Ao rodarem diretamente no sistema operacional, os aplicativos nativos oferecem um desempenho superior e uma experiência de usuário (UX) mais integrada e fluida.
- Acesso a recursos do dispositivo (GPS, câmera, sensores, bluetooth) O acesso total e direto aos recursos do dispositivo é uma das maiores vantagens dos aplicativos nativos. Essa integração profunda permite criar experiências de usuário mais ricas e personalizadas.
- Maior engajamento O maior engajamento dos usuários em aplicativos nativos é impulsionado por uma combinação de fatores, principalmente a experiência de alta qualidade e as funcionalidades de comunicação mais eficazes.
- Dependência das lojas de aplicativos Os aplicativos nativos são distribuídos exclusivamente por meio das lojas oficiais (App Store e Google Play), o que sujeita o desenvolvedor às políticas, revisões e regras de cada plataforma.
- Ciclo de desenvolvimento mais lento . Criação dupla de funcionalidades: A necessidade de criar e testar o aplicativo para iOS e Android separadamente resulta em um ciclo de desenvolvimento mais longo.

Limitações

- Aprovação em lojas de aplicativos (Google Play, App Store) As lojas de aplicativos, como a Google Play e a App Store, são os canais oficiais e essenciais para a distribuição de apps nativos, mas apresentam desvantagens notáveis para desenvolvedores.
- Fragmentação de dispositivos (tamanhos de tela, sistemas diferentes) A fragmentação de dispositivos é um dos maiores desafios no desenvolvimento de aplicativos nativos, principalmente no ecossistema Android.
- Custos maiores para manter versões separadas (Android/iOS) Ter de manter versões separadas para Android e iOS é uma das principais desvantagens dos aplicativos nativos, gerando custos significativos e complexidade em diversas frentes.

- Dependência de linguagens e ferramentas específicas A dependência de linguagens e ferramentas específicas é uma das principais desvantagens do desenvolvimento de aplicativos nativos, pois exige recursos e conhecimentos especializados para cada plataforma.
- Ciclo de desenvolvimento mais longo A necessidade de criar e manter versões separadas para cada sistema operacional (Android e iOS) faz com que o ciclo de desenvolvimento dos aplicativos nativos seja significativamente mais longo, afetando o tempo de lançamento e a frequência de atualizações.

Comparação Web, Desktop e Mobile

Característica	 Rede	 Área de Trabalho	 Móvel
Acesso	Navegador/Internet	Instalação local	Instalação via loja
Linguagens comuns	HTML, CSS, JS	C#, Java, Python	Kotlin, Swift, Dart
Ferramentas	Código VS, GitHub	Estúdio Visual, Eclipse	Android Studio, Xcode
Exemplos	Documentos Google, Trello	Photoshop, Excel	WhatsApp, Instagram
Vantagens	Escalabilidade, acesso fácil	Alto desempenho, offline	Recursos nativos do dispositivo
Limitações	Depende da internet	Pouca portabilidade	Custos e fragmentação

Conclusão

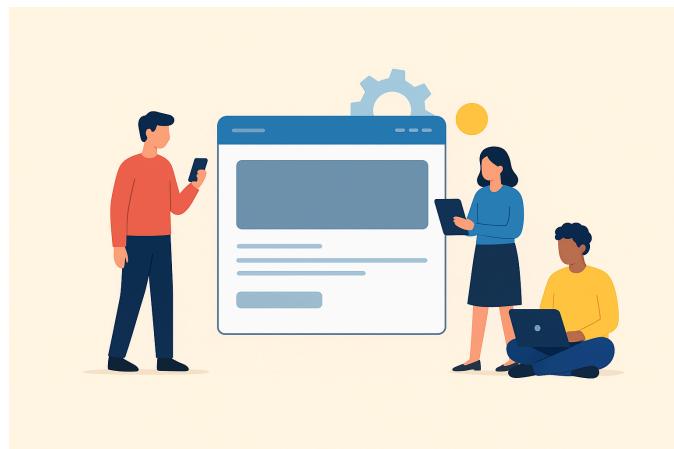
O estudo dos diferentes tipos de desenvolvimento **Web, Desktop e Mobile** evidencia que cada abordagem possui suas próprias **características, vantagens e limitações**.

Enquanto o **desenvolvimento web** se destaca pela **acessibilidade global** e facilidade de atualização, o **desenvolvimento desktop** oferece **alto desempenho e maior controle de hardware**, sendo ideal para aplicações robustas e especializadas. Já o **desenvolvimento mobile** busca entregar

experiências personalizadas e integradas ao cotidiano do usuário, aproveitando os recursos exclusivos dos dispositivos móveis.

Portanto, **não existe uma única solução universalmente superior**: a escolha depende diretamente do **objetivo do projeto, do público-alvo e dos recursos disponíveis**. Muitas vezes, a combinação dessas abordagens (como sistemas híbridos ou integrações entre web, desktop e mobile) representa a estratégia mais eficiente.

Compreender essas diferenças é fundamental para que desenvolvedores, equipes de TI e empresas possam **tomar decisões assertivas**, garantindo **melhor aproveitamento de recursos, maior qualidade na entrega e experiências digitais que atendam às expectativas dos usuários** em diferentes contextos.



😊 Integrantes do Grupo

- Júlio César
- Vitória Beatriz
- Enzzo Gabriel
- Davi Moulin
- Christian