

Unidade 1

Introdução e conceitos



Introdução da Unidade

Neste módulo inicial vamos abordar a visão histórica sobre a Computação Móvel, incluindo as principais definições, conceitos, classificação e relevância. Veremos a evolução dos dispositivos móveis como os primeiros telefones celulares, os primeiros computadores móveis, como o *Apollo Guidance Computer*, até o surgimento dos primeiros *smartphones*. Também abordaremos os ambientes de programação para dispositivos móveis, e alguns conceitos relevantes, como exemplo aplicações responsivas, PWA (*Progressive Web Application*) entre outros.

Objetivos

- Apresentar a visão histórica sobre dispositivos e aplicativos móveis;
- Abordar as diferentes formas de desenvolvimento de aplicativos móveis;
- Fazer uma introdução aos conceitos mais relevantes no desenvolvimento de aplicativos móveis;
- Construir o protótipo das aplicações móveis que serão usadas no decorrer da disciplina para exercitar os conteúdos.

Conteúdo programático

Aula 01 - Conceitos iniciais sobre desenvolvimento de aplicativos

Aula 02 – Prototipagem de Aplicações Móveis



Você poderá também **assistir às videoaulas** em seu celular! Basta apontar a câmera para os **QRCodes** distribuídos neste conteúdo.

Pode ser necessário instalar um aplicativo de leitura QRcode no celular e efetuar login na sua conta Gmail.

Aula 1 – Conceitos iniciais sobre desenvolvimento de aplicativos

Olá pessoal, creio que estão ansiosos para desenvolver seu primeiro aplicativo para dispositivos móveis, e logo faremos isso, mas antes, temos que entender alguns conceitos básicos para poder iniciar nossas atividades da forma correta.

Em primeiro lugar, temos que entender o que é um aplicativo, o que é um dispositivo móvel e o que são linguagens de programação. Também abordaremos os conceitos de prototipagem de *software* e desenvolvimento rápido de aplicações (RAD), desta forma, quando estes termos forem abordados durante as aulas, teremos condições de compreender melhor o conteúdo e onde buscar mais detalhes, quando necessário.

Dispositivo móvel

Um dispositivo móvel, designado popularmente em inglês por *handheld* é um computador de bolso habitualmente equipado com uma pequena tela (*output*) e um teclado de letras ou numérico em miniatura (*input*). No caso dos PDAs (*Personal Digital Assistants*), o *output* e o *input* combinam-se numa tela sensível ao toque (*touchscreen*). Estes dispositivos possuem sistema operacional e são capazes de rodar **aplicativos móveis** podendo se comunicar sem fio com outros dispositivos ou pela internet, começaram a ser produzidos na década de 1990.

Curiosidades

Em 3 de abril de 1973 (dia da estreia do Globo Repórter), liderado por Martin Cooper, a Motorola apresentou e fez a primeira ligação de um telefone celular, com o DynaTAC 8000, que só chegou a ser comercializado em 1983 por cerca de US\$ 4.000,00 (quatro mil dólares). Este celular marcou a primeira geração.

Figura 1 - Martin Cooper em 2007, utilizando o DynaTAC de 1973



Fonte: disponível em: https://pt.wikipedia.org/wiki/Motorola_DynaTAC. Acesso em: 15 mar. 2021.

Veja mais informações no link abaixo:

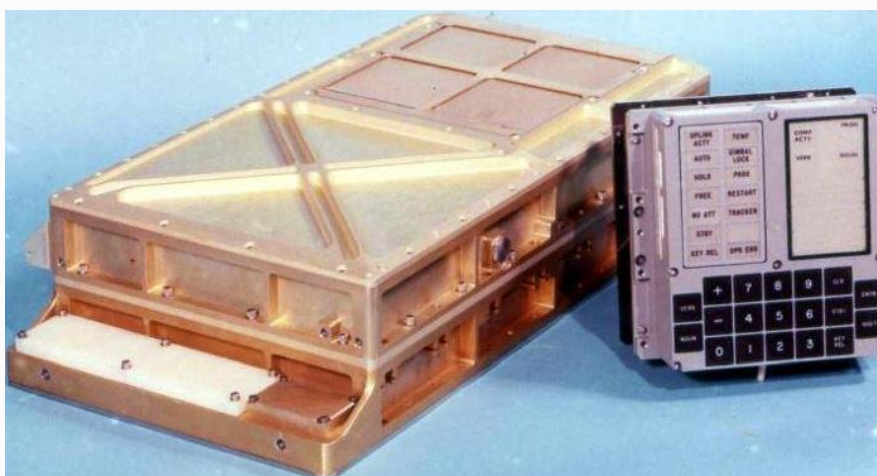
Disponível em: https://pt.wikipedia.org/wiki/Telefone_celular. Acesso em: 15 mar. 2021.

Entre os dispositivos móveis mais comuns podemos citar *Smartphone*, PDA, Celular, Console portátil, Ultra Mobile PC, *Ultrabook*, *Notebook*, *Netbook*, *Laptop*, Coletor de dados, e *Smartwatch*, porém a cada dia surgem novos tipos de dispositivos móveis.

Curiosidades

O **Apollo Guidance Computer (AGC)** pode ser considerado como um dos primeiros dispositivos móveis eletrônicos, semelhante aos nossos *smartphones* ou *notebooks*. O computador de bordo da Apollo 11, possuía um total de 2KB de memória RAM e 36 KB de memória ROM. É até difícil comparar o AGC ao poder computacional dos dispositivos móveis de hoje, um Galaxy J8, por exemplo, tem 4 mil vezes mais capacidade de processamento, já um iPhone com 512GB tem mais de sete milhões de vezes a sua capacidade de armazenamento.

Figura 2 - Apollo Guidance Computer



Fonte: disponível em: https://pt.wikipedia.org/wiki/Apollo_Guidance_Computer. Acesso em: 15 mar. 2021.

Você pode baixar um simulador do AGC em:

Disponível em: <http://www.ibiblio.org/apollo/>. Acesso em: 15 mar. 2021.

Veja mais informações no link abaixo:

Disponível em: https://pt.wikipedia.org/wiki/Dispositivo_m%C3%B3vel. Acesso em: 15 mar. 2021.

Videoaula 1

Agora assista à videoaula que aborda um pouco da história dos dispositivos móveis.

Disponível em: <https://www.youtube.com/watch?v=tA9gervQTD0>. Acesso em: 15 mar. 2021.



Videoaula 1

Utilize o QRcode para assistir!

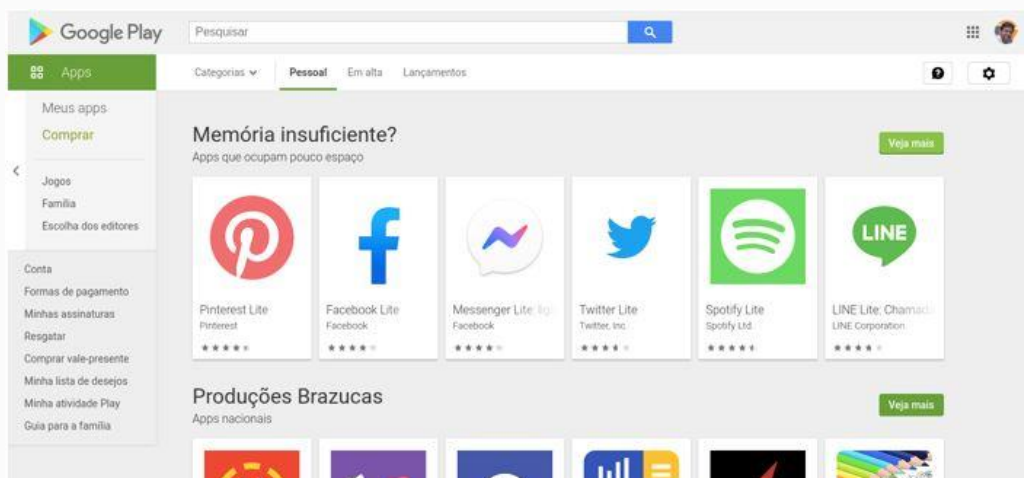
Agora assista à videoaula que aborda um pouco da história dos dispositivos móveis.



Aplicativo móvel

Aplicativo móvel ou App, é um *software* desenvolvido para ser instalado em um dispositivo eletrônico móvel, como um PDA, telefone celular, *smartphone* ou um leitor de MP3. Esta aplicação pode ser instalada no dispositivo, ou se o aparelho permitir descarregada pelo usuário através de uma loja on-line, tais como:

Figura 3 – Google Play



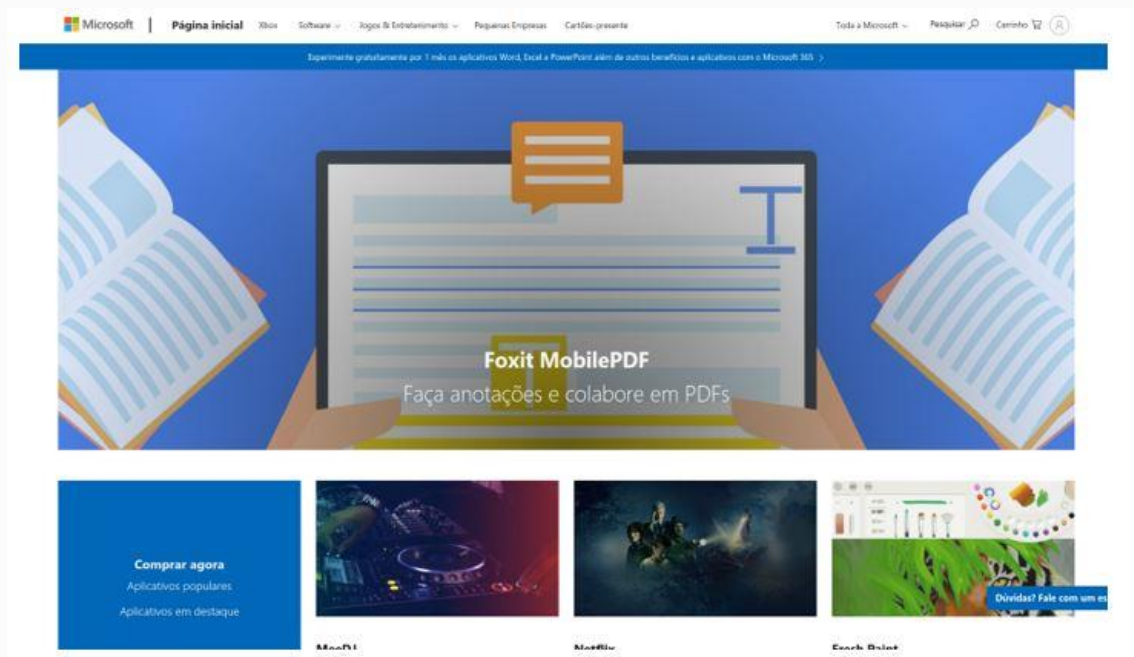
Fonte: disponível em: <https://play.google.com/store/apps>. Acesso em: 15 mar. 2021.

Figura 4 - App Store



Fonte: disponível em: <https://www.apple.com/br/ios/app-store/>. Acesso em: 15 mar. 2021.

Figura 5 - Windows Phone Store

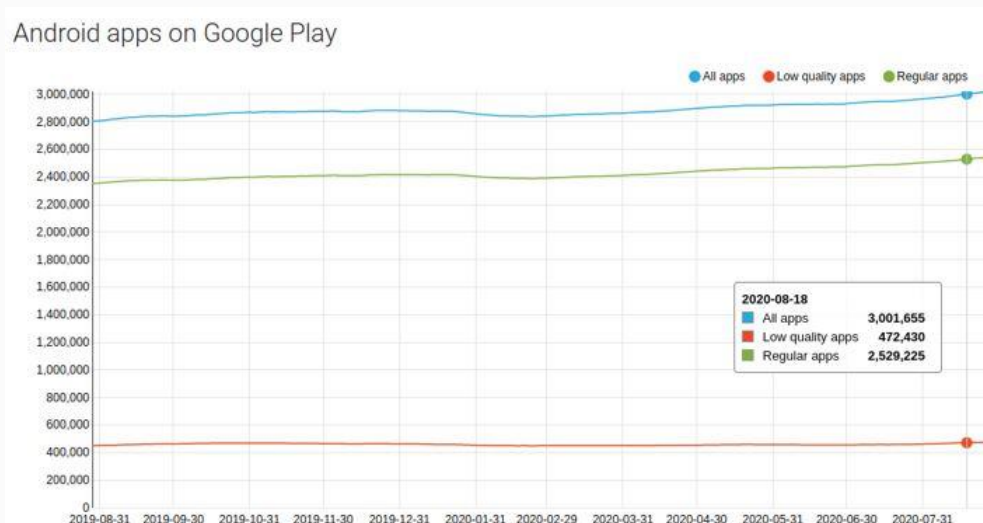


Fonte: disponível em: <https://www.microsoft.com/pt-br/store/apps/windows-phone>. Acesso em: 15 mar. 2021.

Curiosidades

Em agosto de 2020 a *Google Play* superou 3 milhões de aplicações disponíveis para o *Android*, porém esta marca não é novidade para o *Google*, em junho de 2017 a marca de 3 milhões de Apps foi superada, chegando a mais de 3.5 em meados de 2018, porém, no final de 2018, devido às mudanças nas versões do Android aliadas a modificações na política sobre a disponibilidade de aplicações, a quantidade de aplicações caiu significativamente.

Figura 6 - Aplicações para Android no Google Play



Fonte: disponível em: <https://www.appbrain.com/stats/number-of-android-apps>. Acesso em: 15 mar. 2021.

Veja mais informações abaixo:

Disponível em: https://en.wikipedia.org/wiki/Google_Play. Acesso em: 15 mar. 2021.

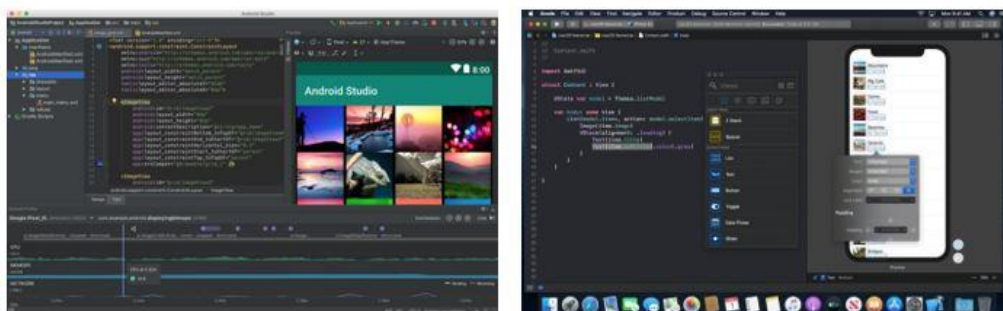
Apesar da *Google Play* e *Apple App Store* serem as lojas de aplicativos mais conhecidas, em grande parte devido a sua relação com o Android e IOS respectivamente, não são as únicas, a *Wikipédia* tem uma lista das lojas de aplicativos atualizada periodicamente, saiba mais no link abaixo:

Disponível em: https://en.wikipedia.org/wiki/List_of_mobile_app_distribution_platforms. Acesso em: 15 mar. 2021.

Podemos classificar um aplicativo móvel por diversos aspectos tecnológicos, estruturais, comerciais, entre outros. Em relação à tecnologia de desenvolvimento temos aplicativos nativos, *Web Apps* e Aplicativos híbridos.

Aplicativos nativos: as aplicações desenvolvidas especificamente para uma plataforma, como o iOS (*Objective-C* ou *Swift*) ou o Android (*Java* ou *Kotlin*), são chamadas de aplicativos nativos. Isto é, uma aplicação criada para a plataforma Android não funciona no iOS. Aplicações para Android são programadas nativamente em *Java*, ou *Kotlin* e nas aplicações para iOS, utiliza-se a linguagem *Objective-C* ou *Swift*. Nos aplicativos nativos é possível utilizar os recursos existentes no *smartphone*, como câmera, GPS etc. O WhatsApp, o Facebook e o Uber são exemplos de aplicativos nativos.

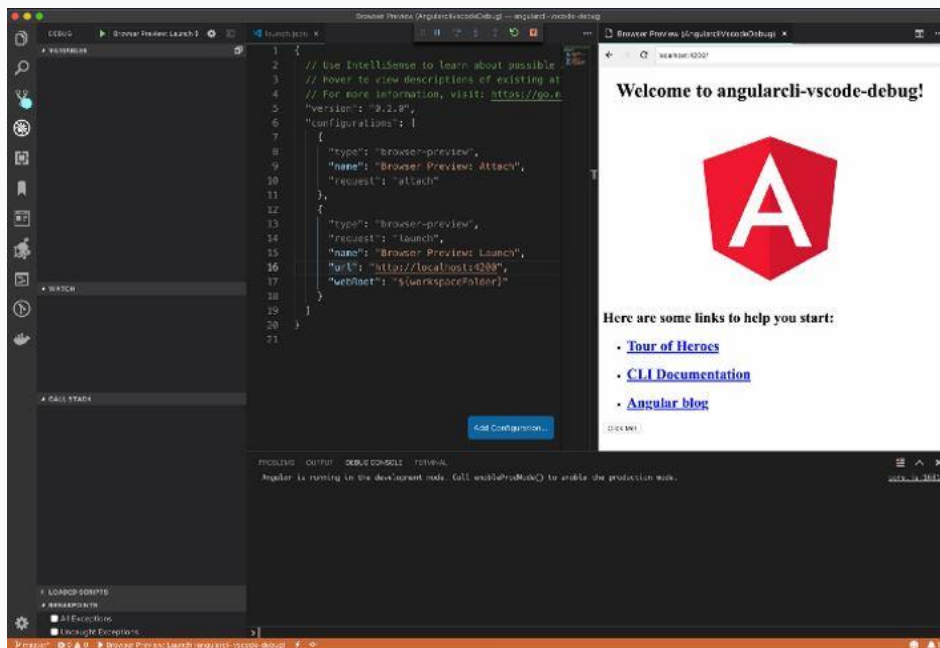
Figura 7 - Android Studio 1.2.1.1. e Xcode 11



Fonte: disponível em: <https://pt.wikipedia.org>. Acesso em: 15 mar. 2021.

Web Apps: os *web Apps* são executados pelo navegador e, uma vez que o programa reconhece que o usuário esteja acessando o site através de um *Smartphone*, então se adapta a ele. Não estão disponíveis para instalação nas *Apps Stores*. São desenvolvidos utilizando HTML5, CSS e Javascript. Estes aplicativos não são recomendáveis para empresas ou ideias que querem proporcionar uma grande qualidade do aplicativo para os usuários. Geralmente fábricas de aplicativos usam essas tecnologias.

Figura 8 - Visual Studio Code com um código em Angular



Fonte: disponível em: <https://github.com/auchenberg/vscode-browser-preview>. Acesso em: 15 mar. 2021.

Aplicativos híbridos: é uma junção de um aplicativo nativo e um *Web App*, gerando assim uma aplicação multiplataforma. Estão disponíveis nas *Apps Stores* diversos aplicativos construídos com esta abordagem, *Netflix* e *Steam* são exemplos de aplicativos híbridos. São desenvolvidos normalmente usando *frameworks* baseados em HTML5, CSS e Javascript, como: IONIC, React Native, Xamarin, Adobe PhoneGap, Flutter, Corona SDK, JQuery Mobile, Native Scripts, Mobile Angular UI, entre outros.

Em geral os aplicativos híbridos são baseados na metodologia de desenvolvimento **PWA** (*Progressive Web App*), um modelo de desenvolvimento de aplicações combina recursos oferecidos pelos mais modernos navegadores, com as vantagens de uso de um smartphone. Veja mais em: https://pt.wikipedia.org/wiki/Progressive_web_app.

Linguagens de Programação

A linguagem de programação é um método padronizado, formado por um conjunto de regras sintáticas e semânticas, de implementação de um código fonte - que pode ser compilado e transformado em um programa de computador, ou usado como script interpretado - que informará instruções de processamento ao computador. Permite que um programador especifique precisamente quais os dados que o computador irá atuar, como estes dados serão armazenados ou transmitidos e quais ações devem ser tomadas de acordo com as circunstâncias. Linguagens de programação podem ser usadas para expressar algoritmos com precisão.

Existem milhares de linguagens de programação, a Wikipédia mantém algumas listas de linguagens de programação atualizadas periodicamente como: Lista de linguagens de programação, *Lists of programming languages*, *Timeline of programming languages*.

Também podemos ver a História das linguagens de programação, que conceitualmente surgiram antes mesmo do equipamentos eletrônicos que hoje chamamos de computadores, entre 1842 e 1843, Ada Lovelace, considerada uma das primeiras programadoras da história, escreveu o primeiro algoritmo para ser processado por uma máquina, a máquina analítica de Charles Babbage.

Videoaula 2

Agora assista ao vídeo que aborda os conceitos de Aplicativos Móveis e linguagens de programação.

Disponível em: <https://www.youtube.com/watch?v=0-RMUgpcH6c>. Acesso em: 15 mar. 2021.



Videoaula 2

Utilize o QRcode para assistir!

Agora assista ao vídeo que aborda os conceitos de Aplicativos Móveis e linguagens de programação.



Aula 2 – Prototipagem de Aplicações Móveis

A prototipagem de *software* é um processo iterativo de geração de modelos de *software* que faz parte da análise do ciclo de vida do desenvolvimento de sistemas. É a atividade de desenvolvimento de uma versão inicial do sistema baseada no atendimento dos requisitos ainda pouco definidos, permitindo a descoberta de falhas difíceis de serem encontradas na comunicação verbal. Um protótipo simula a aparência e funcionalidade do *software* permitindo que os clientes, analistas, desenvolvedores e gerentes “vejam” como os requisitos do sistema estão sendo entendidos, permitindo interagir, avaliar, alterar e aprovar as características mais marcantes na interface e funções.

O protótipo de um sistema pode ser construído usando diversas ferramentas e técnicas, mas basicamente podemos dividir em prototipação descartável e prototipação evolutiva. Para criar um **protótipo descartável** podemos usar apenas lápis e papel, ou ferramentas criadas para esta finalidade, como o Balsamiq Mockups, Figma, Justinmind, Proto.io, Fluidui, entre outras. Já o **protótipo evolutivo** normalmente é criado usando uma IDE (*Integrated Development Environment*) ou um *framework* baseada nos conceitos de RAD (*Rapid Application Development*) que permita construir a estrutura visual da aplicação de forma simples e posteriormente implementar a lógica de funcionamento de forma gradativa.

Em ambos os casos, na elaboração de um protótipo é importante ter em mente as principais características e funcionalidades da aplicação em questão. Para isso aplicamos os conceitos de IHC (Interação humano-computador), para conceber uma interface de usuário que seja simples, intuitiva e que permita a execução de todas as funcionalidades propostas, respeitando padrões de ergonomia, desta forma a experiência do usuário durante o contato com o aplicativo tem mais condições de garantir critérios de usabilidade e acessibilidade. Veja mais em: Design universal, Interface gráfica do utilizador, Acessibilidade web.

A definição de fidelidade de protótipo

Segundo Jerry Cao em seu artigo “*A Guide to Prototype Fidelity*” (disponível em: <https://www.webfx.com/blog/web-design/design-mockup-fidelity/>), a chave para um *mockup* de sucesso pode ser encontrada na definição do nível de fidelidade que será usado no estágio de prototipagem.

Começar com papel e caneta para trabalhar os primeiros rabiscos é uma vitória fácil. Esses esquetes são bastante genéricos e pouco fiéis. À medida que a fidelidade desses esquetes aumenta, eu recomendo que você use uma ferramenta simples, como *balsamiq* ou *moqups*. Deixe ferramentas mais profissionais como *Photoshop* e similares para quando você for trabalhar em desenhos de alta fidelidade. A chave aqui é encontrar o *timing* do esquete de baixa para alta fidelidade.

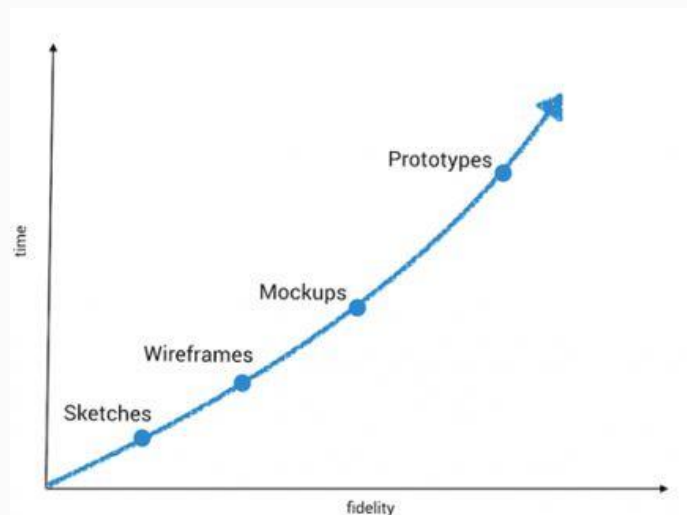
Figura 9 – Tipos de esquete



Fonte: disponível em: <https://www.webfx.com/blog/web-design/design-mockup-fidelity/>. Acesso em: 15 mar. 2021.

Ambos, seja começando muito cedo ou muito tarde nos *mockups* de alta fidelidade, vão gastar recursos (físicos e de tempo), e você vai arriscar acabar com um protótipo medíocre.

Figura 10 - Recursos



Fonte: disponível em: <https://www.webfx.com/blog/web-design/design-mockup-fidelity/>. Acesso em: 15 mar. 2021.

Enquanto fazemos esquetes e *wireframes*, nossa tendência é pensar em soluções antes mesmo de levantar o “problema real”.

O pedido do cliente “faça a logo maior” não é a identificação de nenhum problema.

É uma proposta de solução, apesar de você não estar oferecendo nenhuma informação do problema para a solução. Questionar com vários “porquês” pode ajudar nessas situações.

Hoje, somos treinados para pensar em soluções e em muitas situações essas ideias podem ser muito úteis. Porém, não é realmente de grande ajuda quando você está no meio do processo criativo, desenhando *wireframes* e *mockups*.

E a maioria dessas soluções vem em forma de requisições, como “podemos mudar a cor de vermelho para azul”.

Então, se você está buscando muito a “solução” dos seus colegas e clientes, tente levantar alguns porquês. Por exemplo, “você pode me dizer qual é o desafio que você está tentando resolver com isso?”.

Ao ir fundo e descobrir o real problema, você não só vai ter um resultado melhor, mas vai tornar a sua vida de designer e desenvolvedor mais fácil.

Da mesma forma, encontrar uma boa relação entre a criação do *mockup* sem se ater a detalhes desnecessários, porém, coerentes o suficiente para representar corretamente a ideia além de causar uma boa impressão nos membros da equipe torna-se um desafio.

Claro, não há necessidade de apresentar imagens padrão ou ilustrações finalizadas nesses *mockups*. Há uma grande quantidade de imagens e vídeos de bancos de imagens disponíveis que ajudarão você a tornar seus *mockups* mais vívidos, definitivamente fazendo a diferença na sua apresentação.

Indicação de Leitura

Leia o artigo “*Resources of royalty-free images for your tech blog*” de Thomas Peham.

Disponível em: <https://medium.com/web-development-resources/resources-of-royalty-free-images-for-your-tech-blog-b725b251f656>. Acesso em: 15 mar. 2021.

Neste artigo existem diversas indicações de bancos de vídeo e imagem.

Use a ferramenta certa para cada etapa do trabalho, nem sempre uma única ferramenta consegue dar a agilidade necessária para todas as etapas, por exemplo:

- **Sketches:** usar papel e quadros (*Flip Chart* ou um Quadro Branco) para coletar as primeiras ideias. Evernote ou o próprio GDocs podem ser ótimos assistentes para agrupar essas ideias.
- **Wireframes e mockups:** quando você estiver trabalhando com *wireframes* e *mockups*, Balsamiq Mockups, wireframe.cc, Mockplus e Figma são ferramentas muito úteis!

- **Protótipos:** para trazer seus *mockups* à vida real, ferramentas como InVision, Usersnap e CodePen ajudam você na construção dos seus protótipos ao mesmo tempo em que garantem que nenhuma ideia ou *feedback* tenha ficado perdido para trás.

Videoaula 1

Agora assista à videoaula que aborda um pouco sobre a elaboração de protótipos e que apresenta algumas ferramentas disponíveis para isso.

Disponível em: <https://youtu.be/fjZfNTEAaEU>. Acesso em: 15 mar. 2021.



Videoaula 1

Utilize o QRcode para assistir!

Agora assista à videoaula que aborda um pouco sobre a elaboração de protótipos e que apresenta algumas ferramentas disponíveis para isso.



Existem muitas ferramentas destinadas a auxiliar no processo de construção do protótipo de sua aplicação e a cada dia novas ferramentas são criadas, o mais importante é entender o conceito e definir as ferramentas mais adequadas para seu fluxo de trabalho.

Em uma pesquisa rápida no *Google* pelos termos “*wireframe tools*”, “*mockup tools*” ou “*software prototype tools*” são apresentadas centenas de ferramentas com este objetivo, segue uma pequena lista com algumas que são muito interessantes:

- *MockFlow* (<https://www.mockflow.com/>);
- *Cacoo: Online Diagram and Flowchart Software* (<https://cacoo.com/home>);
- *FluidUI.com* (<https://www.fluidui.com/>);
- *NinjaMock online wireframe and mockup tool* (<https://ninjamock.com/>);
- *Pencil Project* (<https://pencil.evolus.vn/>);
- *mockplus.com* (<https://www.mockplus.com/free-wireframing-tool/>);
- *Wireframe.cc* (<https://wireframe.cc/>);
- *Sketch* (<https://www.sketch.com/>);
- *Adobe XD* (<https://www.adobe.com/in/products/xd.html>);
- *HotGloo | Wireframe UX Prototyping Tool* (<https://www.hotgloo.com/>);
- *Froont – responsive web design tool* (<https://froont.com/>);

- UXPin | UI Design and Prototyping Tool (<https://www.uxpin.com/>);
- Axure RP 9 (<https://www.axure.com/>);
- Framer (<https://www.framer.com/>);
- UX-App (<https://www.ux-app.com/>);
- InVision - Free (<https://www.invisionapp.com/cloud/prototype>);
- Gravit Designer (<https://www.designer.io/en/>);
- Xara Designer Pro (<https://www.xara.com/us/designer-pro/>);
- Lucid App (<https://lucid.app/users/>);
- Pidoco - Online Wireframe and UX Prototyping Tool (<https://pidoco.com/en>);
- Figma (<https://www.figma.com/files/recent>);
- FluidUI.com (<https://www.fluidui.com/>);
- Avocode App (<https://avocode.com/>);
- InDesign (<https://www.adobe.com/products/indesign.html>);
- Moqups (<https://moqups.com/>).

Construção dos nossos primeiros protótipos

Para construção dos nossos primeiros protótipos vamos usar a ferramenta *Balsamiq Mockups* (<https://balsamiq.com/>) em sua versão *on-line*. Disponível em: (<https://balsamiq.cloud/>).

O *Balsamiq Mockups* é um *software* que visa aprimorar a criação de *mockups*, ou seja, o estágio inicial de desenvolvimento de produtos, aplicativos, páginas etc.

Para iniciar as atividades, vamos criar uma conta no *Balsamiq* (<https://balsamiq.cloud/>). Use seu e-mail (@edu.unifil.br) para acessar, assim você poderá solicitar um acesso acadêmico para sua conta da seguinte forma:

Figura 11 – Conta no Balsamiq



Fonte: disponível em: <https://balsamiq.cloud/>. Acesso em: 15 mar. 2021.

Ao criar sua conta com o e-mail (@edu.unifil.br) você estará apto para receber o acesso educacional para uso em sala de aula. A liberação do acesso pela *Balsamiq* pode demorar alguns dias, enquanto isso você pode usar o período de demonstração gratuita.

Videoaula 2

Agora assista à videoaula na qual construímos o nosso primeiro protótipo de uma aplicação básica.

Disponível em: <https://youtu.be/5Fej4XORFhc>. Acesso em: 15 mar. 2021.



Videoaula 2

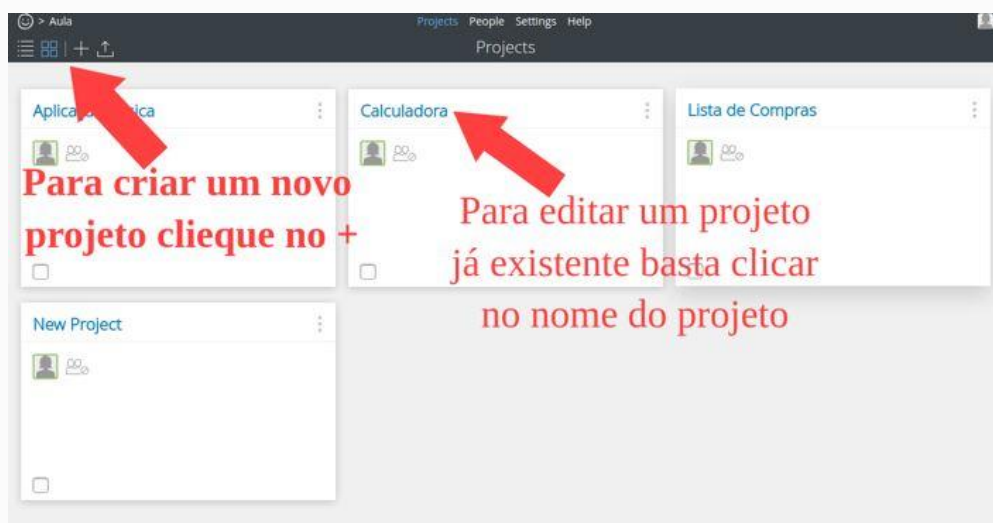
Utilize o QRcode para assistir!

Agora assista à videoaula na qual construímos o nosso primeiro protótipo de uma aplicação básica.



Podemos usar o *Balsamiq Mockups* para desenhar a interface (telas) de aplicações *Web*, *Desktop* além de Aplicações para Dispositivos Móveis.

Figura 12 - Projeto no Balsamiq Mockups

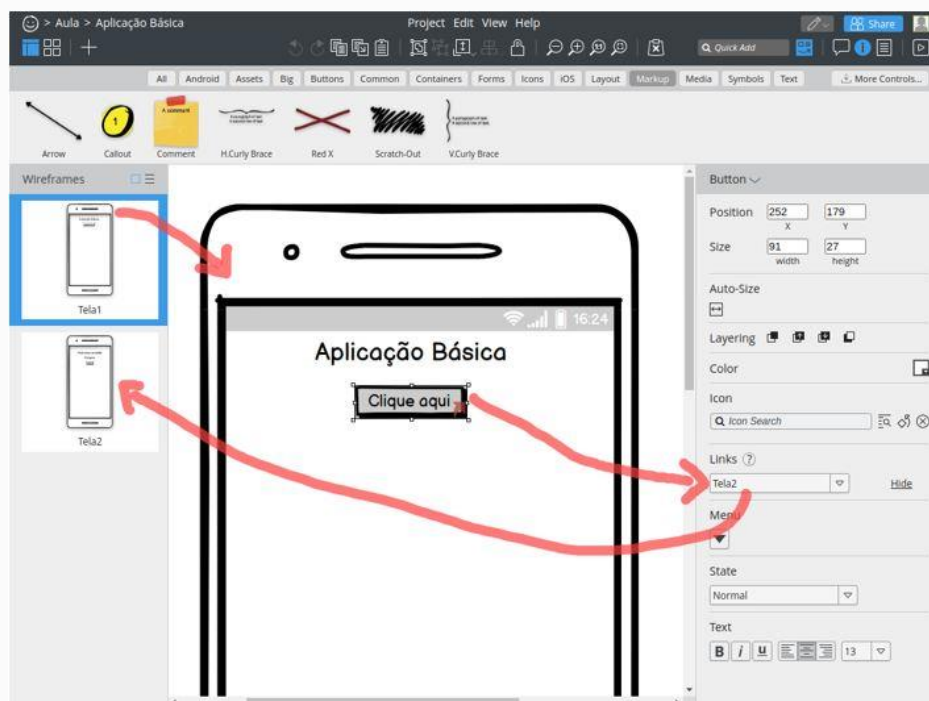


Fonte: Balsamiq (2021)

Temos diversos recursos para facilitar o trabalho durante a criação dos protótipos, mas lembre-se que o *Balsamiq Mockups* é destinado a criação de *Sketches*, *Wireframes* ou *Mockups*, com “Baixa Fidelidade”, ele não é a ferramenta adequada para validar detalhes “finos” da aplicação. O objetivo neste momento é a validação das funcionalidades, características de comportamento e principais elementos durante o desenvolvimento.

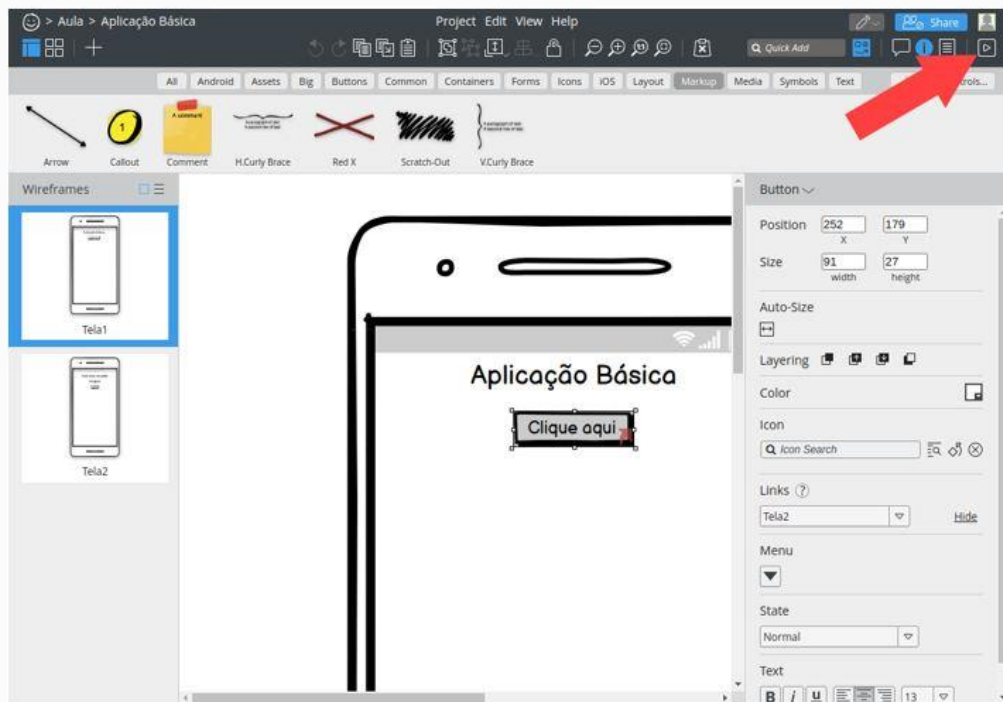
Como vimos em nossa segunda videoaula da aula 02, podemos simular o comportamento da aplicação utilizando o recurso “Links”. Este recurso pode ser muito útil quando nossa aplicação tem diversas telas e temos que demonstrar como irá funcionar a navegação entre estas telas.

Figura 13 – Navegação entre telas



Fonte: Balsamiq (2021)

Figura 14 – Aplicação básica



Fonte: Balsamiq (2021)

O recurso “*Full Screen Presentation*” é muito interessante durante uma reunião de apresentação da proposta, pois permite que o comportamento de navegação “simulada” seja visto pelos participantes, simplificando o entendimento e contribuindo para identificar pontos de ajustes necessários.

Videoaula 3

Agora assista à videoaula na qual construímos o protótipo de uma calculadora.

Disponível em: <https://youtu.be/NckkcC8KBIM>. Acesso em: 15 mar. 2021.



Videoaula 3

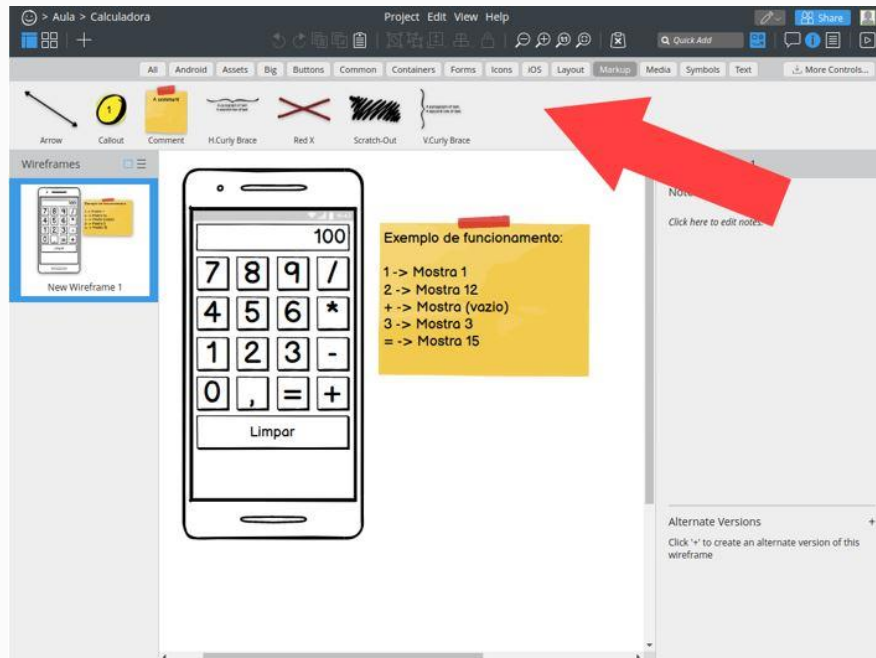
Utilize o QRcode para assistir!

Agora assista à videoaula na qual construímos o protótipo de uma calculadora.



O uso de anotações, marcações e outros elementos também pode contribuir de forma significativa com o entendimento do comportamento esperado para a aplicação.

Figura 15 – Protótipo de uma calculadora



Fonte: Balsamiq (2021)

Desta forma, vemos que o processo de prototipação ajuda a entender o propósito do *software* que será desenvolvido. Quando ele é bem executado, fica mais fácil compreender o conceito do planejamento, avaliar a viabilidade das decisões tomadas e realizar ajustes. Assim, as chances de o resultado final estar dentro das expectativas do cliente serão muito maiores.

Com a construção de protótipos, pequenas melhorias e ajustes serão feitos logo no começo da criação do sistema, evitando que a equipe gaste tempo e recursos com a construção de uma aplicação que não atenderá às necessidades de quem vai usar esta aplicação.

Temos muitos benefícios ao criar protótipos para nossas aplicações antes de iniciar o trabalho de construção de código, dentre eles podemos citar:

- A possibilidade de reduzir riscos no desenvolvimento do sistema ao definir objetivos com precisão;
- A facilidade de visualizar o modo como a solução será utilizada pelo usuário, permitindo um planejamento mais eficaz de cada recurso;
- A criação de um fluxo constante de *feedbacks* entre a empresa e o seu cliente, melhorando a relação entre toda a equipe;

- A otimização do investimento na criação da ferramenta com a eliminação de erros;
- A maior agilidade na entrega de resultados por meio de um planejamento melhor estruturado.

Por fim, vale ressaltar que a criação de um sistema é um processo complexo. Em meio a várias etapas, portanto, pode ser que erros sejam cometidos e que o produto final precise de ajustes para atingir os objetivos esperados pelo cliente.

Para se prevenir de falhas e garantir que todos estejam alinhados, uma série de abordagens são utilizadas. Há quem opte por *feedbacks* e reuniões constantes. A comunicação, nesse sentido, permite um trabalho contínuo para evitar que as funcionalidades não sejam projetadas corretamente.

A prototipação, nesse cenário, é importante por ser um meio rápido e bastante eficiente de validar uma ideia. As abordagens e conceitos empregados na criação do sistema serão visualizados junto com outros fatores que impactam diretamente na experiência do uso, evitando riscos.

Para garantir mais eficiência, porém, o tipo de protótipo deve ser pensado com cautela. É necessário avaliar cada necessidade e o perfil do usuário de forma a identificar o que de melhor essa técnica pode oferecer.

Em outras palavras, após identificar quais são os tipos de prototipação que melhor correspondem às suas demandas, se mantenha sempre alinhado com a equipe de desenvolvimento. Afinal de contas, um simples detalhe prototipado incorretamente pode gerar um trabalho complexo, acarretando custos inesperados e atrasos.

Encerramento da Unidade

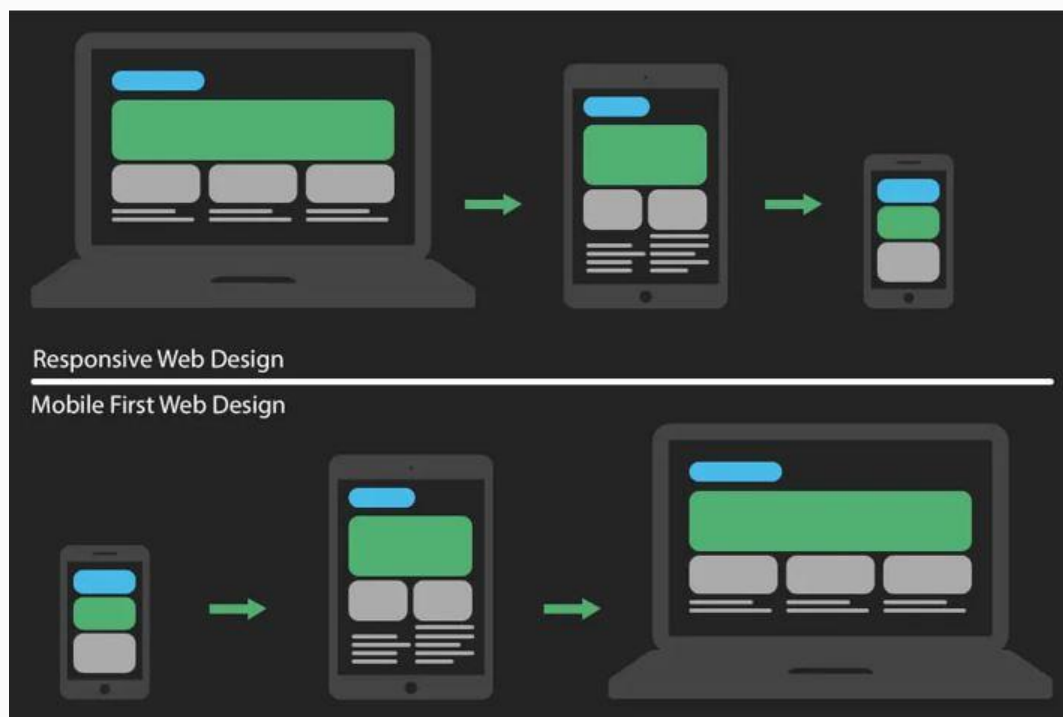
Como vimos nesta unidade, um dispositivo móvel foi originalmente caracterizado como um computador de bolso habitualmente equipado com uma pequena tela e um teclado em miniatura, este conceito evoluiu passando pelos PDAs, onde as telas sensíveis ao toque começaram a surgir até chegar ao que hoje conhecemos por *smartphones* e *tablets*.

As evoluções tecnológicas nos dispositivos móveis na parte de *hardware* (equipamento físico) também foram significativas na parte de *software* (Sistema Operacional, programas, aplicações), com isso, os ambientes de desenvolvimento de aplicativos móveis mudaram muito nos últimos anos.

Outro fator importante é que o volume de pessoas que usam dispositivos móveis, principalmente *smartphones*, para realizar a maioria das tarefas no seu dia a dia tem crescido ano após ano.

Este fator cria inúmeras discussões sobre como conceber a construção de aplicações e sistemas, como exemplo, os conceitos de Responsive Web Design e Mobile First Web Design.

Figura 16 - *Responsive Web Design* e *Mobile First Web Design*



Fonte: disponível em: <https://portal.clint.digital/mobile-first-conceitos-e-como-aplicar/>. Acesso em: 15 mar. 2021.

Independente da abordagem, a relevância da criação de qualquer solução de *software* considerando seu uso em dispositivos móveis torna-se cada vez mais necessária, assim, conhecer os conceitos e tecnologias para o desenvolvimento de aplicativos móveis passa a compor o conjunto de habilidades (*skills*) de um bom profissional da computação.

Em nossas próximas unidades veremos como aplicar estes conceitos na prática e construir nossas primeiras aplicações móveis.

Referências

FÉLIX, Rafael; SILVA, Everaldo Leme. **Arquitetura para computação móvel**. 2. ed. São Paulo: Pearson Education do Brasil, 2019.

FLATSCHART, Fábio. **HTML 5: Embarque Imediato**. Rio de Janeiro: Editora Brasport, 2011.

LEE, Valentino; SCHNEIDER, Heather; SCHELL, Robbie. **Aplicações móveis: arquitetura, projeto e desenvolvimento**. São Paulo: Pearson Makron Books, 2005.

MARINHO, Antonio Lopes (org.). **Desenvolvimento de aplicações para Internet**. São Paulo: Pearson Education do Brasil, 2016.

SEGURADO, Valquiria Santos (org.). **Projeto de interface com o usuário**. 1. ed. São Paulo: Pearson Education do Brasil, 2015.

SILVA, Diego (org.). **Desenvolvimento para dispositivos móveis**. São Paulo: Pearson Education do Brasil, 2016.

SOMMERVILLE, Ian. **Engenharia de software**. Tradução Luiz Claudio Queiroz. Revisão técnica Fábio Levy Siqueira. 10. ed. São Paulo: Pearson, 2018. 768 p.

Publicações da SBC (<http://www.sbc.org.br/index.php?Itemid=196>)

Portal Periódicos. CAPES (<http://www.periodicos.capes.gov.br>)

Google Acadêmico (<http://scholar.google.com.br>)

SciELO (<http://www.scielo.br>)

arXiv.org (<http://arxiv.org>)

ACM Digital Library (<http://dl.acm.org>)

IEEE Xplore (<http://ieeexplore.ieee.org/Xplore/home.jsp>)

ScienceDirect (<http://www.sciencedirect.com>)



UNIFIL.BR