





Barbara Nery
Comunidad

Desarrollador, profesor y acróbata :D

habla sobre:

#sharepoint #liderazgo #javascript #itil

 Barbara Nery

 _barbara.js

 barbarasemacento

 barbarasemacento



Menu

AULA 1

Slide 04: Introdução e conceitos.

Slide 07: Responsivo x PWA x Mobile.

Slide 10: Nativo x Híbrido.

Slide 16: Custom coding x Low-code x No-code.

Slide 19: Tipos de plataformas Low-code e No-code para desenvolvimento.

AULA 2:

Slide 21: Slide: Introdução ao Glide.

AULA 3:

Slide 23: Construção do Aplicativo no Glide – Passo a passo.

AULA 4:

Slide 31: Flutter.

Mobile: De tendência à Necessidade!

O desenvolvimento mobile é um tipo de rotina de criação de soluções de TI voltadas para tablets, smartphones e outros dispositivos móveis.

A experiência do usuário pode ser considerada como a referência fundamental para desenvolver um aplicativo visando seu sucesso. É pensando nela que um programador é capaz de levantar questões pertinentes sobre, por exemplo, funcionalidade e design que guiarão o projeto no rumo certo.



Mobile: Pra começar...

Bora fazer?!

(Necessidade)

Estruturação
de ideias



Quando fazer?

(Cronograma, estimativas e prazos)

Planejamento do
layout e recursos



Especificações
do aplicativo

3



4



Criação do
design

Como fazer?

*(Arquitetura, plataforma e
linguagens)*

Escolha da
tecnologia



Desenvolvimento
do app

6



7



Prototipação

Nascendo...

(Testando...)

O que fazer?

(Requisitos)

O jeito certo de fazer!

(Usabilidade)

Fazendo...

("Codando")

Mobile: O jeito certo de fazer!

Utilização com eficiência, com aprendizado fácil e simples. A usabilidade pode se resumir a essas características, pois trata-se da capacidade de um sistema em fazer com que os usuários consigam desempenhar as tarefas sem dificuldades.

Em outras palavras, é um conjunto de técnicas envolvendo recursos visuais e programação, com o objetivo de dar ao app a capacidade intuitiva necessária para que os usuários entendam as funcionalidades.

Talvez o seu *requisito*² ou o seu *prazo*³ podem ser fatores determinantes para essa escolhas.

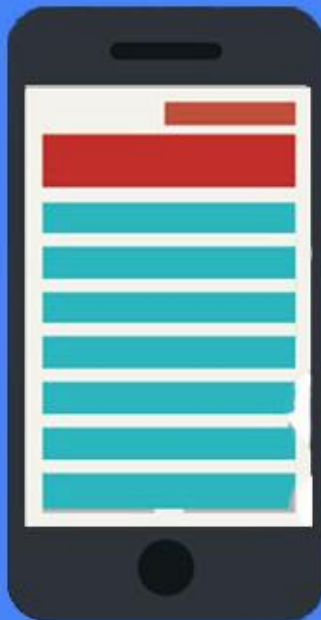
- * *Entender como o seu público manuseia o smartphone - Design estratégico!*
- * *Optar por plataformas mobile first !*
- * *Reduzir clicks, funcionalidades e conteúdo!*
- * *Consistencia, metáforas, design, visualização....*



Responsivo, são aqueles que adaptam o tamanho das suas páginas (alteração do layout) ao tamanho das telas que estão sendo exibidos, como as telas de celulares e tablets.



PWA, sigla para **Progressive Web App**, é um site desenvolvido para parecer e se comportar como um aplicativo nativo independente da plataforma em que é utilizado.



Mobile, possuem comportamento específico para o device. Pode atingir um desempenho muito melhor e tem capacidade para utilizar os recursos do dispositivo. A experiência do usuário é muito superior.



Qual a diferença entre o Web e app Mobile ?

1. Não exige download;
2. Facilidade de encontrar;
3. Formas de acesso.



1. Facilidade de acesso;
2. Notificações Push;
3. Funcionamento offline;

Cada uma das soluções possui vantagens. Assim, a escolha mais adequada depende das prioridades definidas pelos gestores da *empresa*. Um indicador que pode ajudar na decisão é perguntar:

Como o consumidor busca seu produto ou serviço? e qual problema seu App deverá resolver?

Mobile: Como fazer!

Agora é a vez de desenvolver, certo? Mas você já sabe qual a tecnologia usar ou escolher? Talvez o seu *requisito*² ou o seu *prazo*³ podem ser fatores determinantes para essa escolhas.

Além dos itens 2 e 3 da nossa lista, que ajudam na escolha, também temos:

- * *Dispositivo* (IOS, Android ou os dois);
- * *Plataforma* (No-Code, Low Code ou Custom);
- * *Linguagem* (Xamarin, Flutter, React Native....);
- * *Arquitetura* (MVC-Model View Controller, MVP-Model View Presenter, V&PER -View Interactor Presenter Entity Router....);



Nativo



Vs

Híbrido



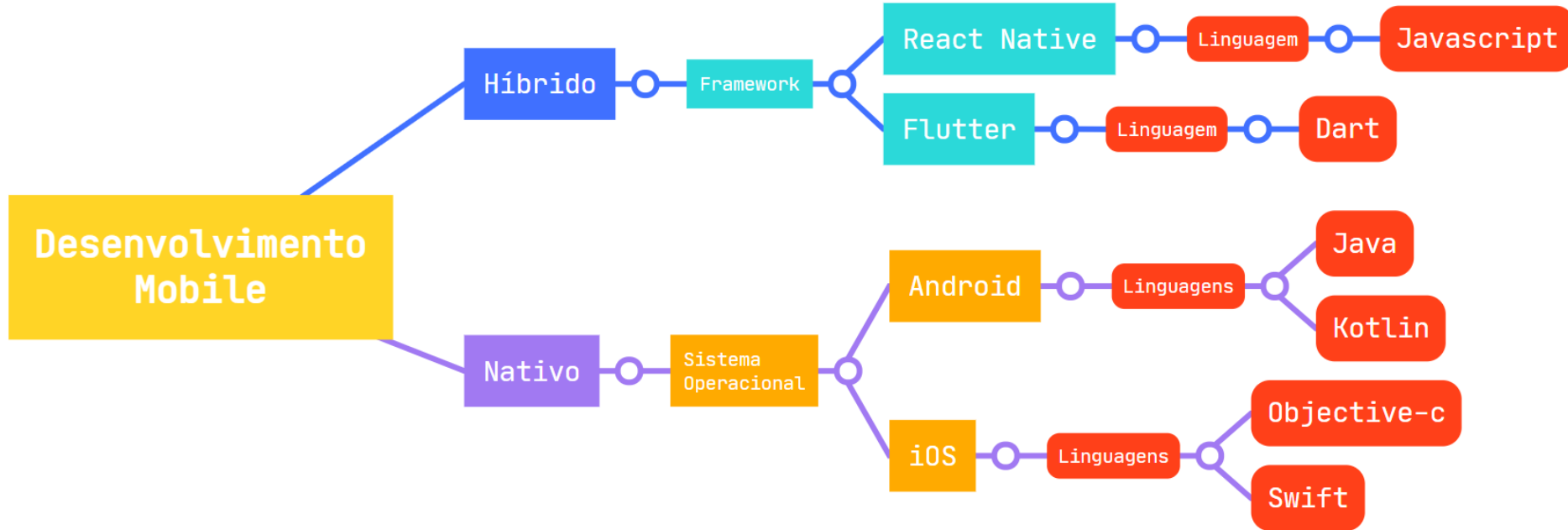
Nativo ou Híbrido?

Um app **nativo** costuma a ser desenvolvido para uma plataforma específica, explorando assim todas as potencialidades daquele sistema operacional. O **híbrido**, no entanto, consegue unir o **nativo** com a linguagem **da** web, podendo ser utilizado tanto **em** um como no outro.

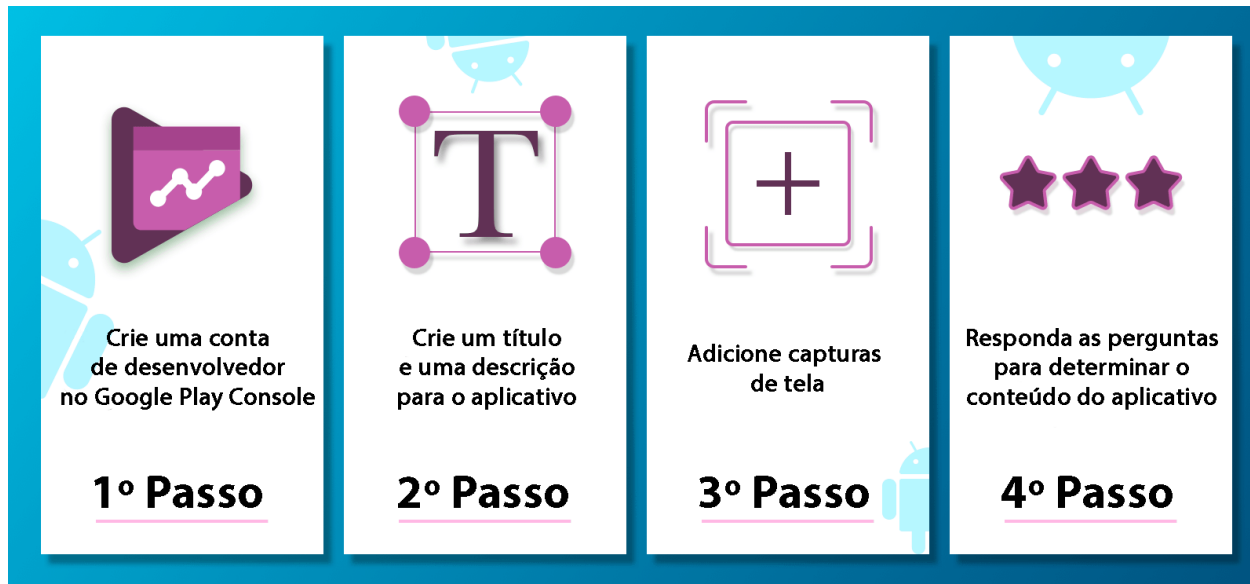


	Vantagens	Desvantagens
Apps Nativos	<ul style="list-style-type: none">• Acesso às funcionalidades do aparelho (Câmera, GPS)• Apps baixados diretamente pelas lojas• Sistema mais específico• Funcionamento offline	<ul style="list-style-type: none">• Custo bem maior• Aprovação da loja / linguagem específica• Criação em diferentes plataformas
Web app	<ul style="list-style-type: none">• Custo mais baixo• Página da web• Acessado de qualquer browser / Funciona para Todas as plataformas	<ul style="list-style-type: none">• Funcionamento apenas online• Não publica nas lojas• Não utilizam as funcionalidades do aparelho
Apps Híbridos	<ul style="list-style-type: none">• Funcionalidades semelhante aos nativos• Custo mais baixo Funciona para todas as plataformas• Apps baixados diretamente pelas lojas• Opção mais barata para manter o app nas lojas	<ul style="list-style-type: none">• Performance e usabilidade um pouco inferior ao app nativo• Design mais restrito

Mobile: Como fazer - Dispositivos e Linguagens!



Mobile: Como fazer – Publicar apps!



Mobile: Como fazer – Publicar apps!



Mobile: Como fazer – Publicar apps!



Mobile: Como fazer - Plataformas!

Agora já escolhemos para que tipo de dispositivo desenvolver e a usabilidade, vamos falar de um tema novo e promissor que é a escolha da plataforma para entregar o seu produto.



Mobile: Quantos profissionais são necessários para criar um aplicativo? Você Sabe? *(Exercício)*

- 1) Requisitos (P.Q);
- 2) Gerente de Projetos (Scrum);
- 3) Experiencia do Usuário;
- 4) Arquiteto de Soluções e o Engenheiro de software;
- 5) Front, Back e Banco de Dados;
- 6) QA ou Teste.

Mobile: Como fazer - Plataformas!

Sim! Podemos entregar um produto de valor com 0 códigos! Cada vez mais empresas de tecnologia como Maicrosof investem na tecnologia 'no' e 'low' code.

Low-code (pouco código) e **no-code** (sem código) são tecnologias que buscam a utilização de menos - ou nenhum - código no desenvolvimento de produtos e serviços. Há alguns anos atrás, era necessária muita digitação e criação de linhas de código para construir um software, por exemplo.

- 1) *Flexibilidade para criar aplicativos customizados;*
- 2) *Produtividade para a equipe;*
- 3) *Autonomia para a empresa.*

*** Existem planos free e planos pagos*

Mobile: No-Code e Low-Code para Mobile!

 bubble

Glide

Adalo

 PhoneGap

 AppGyver

 AppSheet

Mobile: Agora é a sua vez!



Vamos conversar? O que você quer saber?

Aula 2 - No code

Aula 3 - Low Code

Aula 4 - Code

Baixar : <https://www.appgyver.com/> e

<https://www.appgyver.com/>

Fazer um "iFood"

Mobile: Aula 2 !

Hoje vamos fazer um aplicativo 100% No-code, com o auxílio do [Glider](https://www.glideapps.com/templates) que é uma plataforma online de desenvolvimento de Web app e App mobile. (<https://www.glideapps.com/templates>)

Para habilitar o recurso de pagamentos online no aplicativo (o famoso carrinho) você pode utilizar o [Stripe](https://stripe.com/br) para viabilizar o pagamento online (<https://stripe.com/br>) - Quem usa o plano free do Glider, destina 10% da sua venda para o pagamento do Stripe e que é pró usa 2% do valor de venda.

Para criar a extensão APK (aplicativos) vamos utilizar o auxílio do [kodular](https://www.kodular.io/) (<https://www.kodular.io/>)

Caso você queira disponibilizar seu app na Google Play, basta acessar <https://play.google.com/console/about/> , fazer o cadastro e pagar uma taxa de 25 dólares (20/09/22) disponibilizando para a Google Play as imagens do aplicativo nas seguintes dimensões:

- Ícone: 512x512px PNG
- Prints das telas: 1080x2065px PNG (prints)
- Banner: 1024x500px PNG

Mobile: Aula 3 !

Agora que já conhecemos o Glider vamos fazer nosso aplicativo:

Premissas:

O Glide é uma plataforma online, então criar a sua conta com o e-mail [google](#) em:

<https://www.glideapps.com/templates>

Acesso ao app criado hoje:

<https://go.glideapps.com/join/electric-tiger-19bb21>

Todo conteudo está disponivel em:

https://github.com/barbarasemacento/Toti-Aula_Mobile

****As imagens deverão estar em um servidor de imagem (Estou usando o próprio GitHub)**

Mobile: Aula 3 !

Vamos criar nosso aplicativo baseado em uma planilha excel, então vamos imaginar que cada aba do nosso excel será uma tela do nosso aplicativo, ok?

Para cada componente na tela (Botões, imagens...) será original de uma coluna desse excel.

Então se quiser fazer um **botão na Home**, você deverá criar uma **aba no excel chamada Home** e uma **coluna no excel chamada botão**, beleza?!

Esse excel deverá estar disponível no mesmo google drive que o endereço de e-mail utilizado para logar no Glider.

Mobile: Aula 3 !

Para esse aplicativo vamos precisar de quatro telas , ou abas de excel, para começar:

Home				Botão	
https://raw.githubusercontent.com/barbarasemacento/TotiMobileGlider/main/logoToti.jpg					
◀ ▶	Home	Categoria	Produtos	Pedido	⊕

Categoria	Imagem
Fast food	https://raw.githubusercontent.com/barbarasemacento/TotiMobileGlider/main/fotosAlimentos/fastFood.jpg
Fitness	https://raw.githubusercontent.com/barbarasemacento/TotiMobileGlider/main/fotosAlimentos/fitness.jpg
Arabe	https://raw.githubusercontent.com/barbarasemacento/TotiMobileGlider/main/fotosAlimentos/arabe.jpg
Brasileira	https://raw.githubusercontent.com/barbarasemacento/TotiMobileGlider/main/fotosAlimentos/brasileira.jpg
Japonesa	https://raw.githubusercontent.com/barbarasemacento/TotiMobileGlider/main/fotosAlimentos/japonesa.jpg
Venezuelana	https://raw.githubusercontent.com/barbarasemacento/TotiMobileGlider/main/fotosAlimentos/venezuelana.jpg

◀ ▶	Home	Categoria	Produtos	Pedido	⊕	⋮
-----	------	-----------	----------	--------	---	---

Mobile: Aula 3 !

Para esse aplicativo vamos precisar de quatro telas , ou abas de excel, para começar:

Categoria	Produto	Imagem	Descrição	Preço
Fast food	Batata Frita	https://raw.githubusercontent.com/barbarasemacento/TotiMobileGlider/main/fotosAlimentos/fastFood/batataFrita.jpg	350 gramas - Serve 1 Pessoa	10
Fast food	Hamburger	https://raw.githubusercontent.com/barbarasemacento/TotiMobileGlider/main/fotosAlimentos/fastFood/hamburger.jpg	351 gramas - Serve 1 Pessoa	10
Fitness	Purê de Inhame	https://raw.githubusercontent.com/barbarasemacento/TotiMobileGlider/main/fotosAlimentos/fitness.jpg	352 gramas - Serve 1 Pessoa	10
Fitness	Purê de Abóbora	https://raw.githubusercontent.com/barbarasemacento/TotiMobileGlider/main/fotosAlimentos/fitness/pure.jpg	353 gramas - Serve 1 Pessoa	10
Arabe	Cuzcuz Marroquino	https://raw.githubusercontent.com/barbarasemacento/TotiMobileGlider/main/fotosAlimentos/arabe/cuzcuzMarroquino.jpg	354 gramas - Serve 1 Pessoa	10
Arabe	Esfiha	https://raw.githubusercontent.com/barbarasemacento/TotiMobileGlider/main/fotosAlimentos/arabe/esfiha.jpg	355 gramas - Serve 1 Pessoa	10
Brasileira	Tacaca	https://raw.githubusercontent.com/barbarasemacento/TotiMobileGlider/main/fotosAlimentos/brasileira/tacaca.jpg	356 gramas - Serve 1 Pessoa	10
Brasileira	Acarajé	https://raw.githubusercontent.com/barbarasemacento/TotiMobileGlider/main/fotosAlimentos/brasileira/acaraje.jpg	357 gramas - Serve 1 Pessoa	10
Japonesa	Temaki	https://raw.githubusercontent.com/barbarasemacento/TotiMobileGlider/main/fotosAlimentos/japonesa/temaki.jpg	358 gramas - Serve 1 Pessoa	10
Japonesa	Sukiaki	https://raw.githubusercontent.com/barbarasemacento/TotiMobileGlider/main/fotosAlimentos/japonesa.jpg	359 gramas - Serve 1 Pessoa	10
Venezuelana	Hallaca	https://raw.githubusercontent.com/barbarasemacento/TotiMobileGlider/main/fotosAlimentos/venezuelana/hallaca.jpg	360 gramas - Serve 1 Pessoa	10
Venezuelana	Tequeños	https://raw.githubusercontent.com/barbarasemacento/TotiMobileGlider/main/fotosAlimentos/venezuelana/teque%C3	361 gramas - Serve 1 Pessoa	10

Home

Categoria

Produtos

Pedido

+

Dono do pedido	Produto	Imagem do Produto	Qty	Preço	Total	e-mail	Endereço	Entrega?	Forma de pagamento	Troco

Home

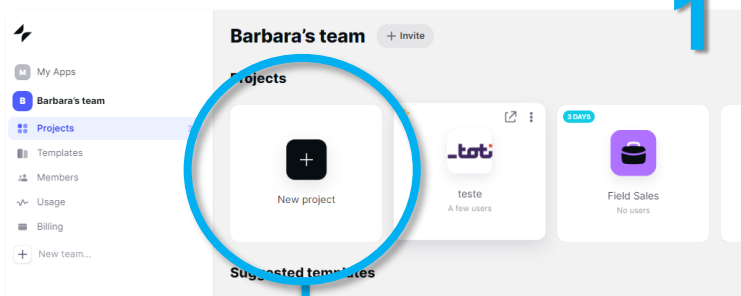
Categoria

Produtos

Pedido

+

Mobile: Aula 3 ! No Glide:

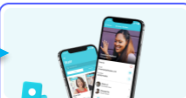


Create a new project

Name your project

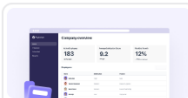
Toti Food

Choose a type



Glide App

An app optimized for phone & tablet, that users can add to their homescreen.



Glide Page

A powerful, data-driven website that adapts to screens of all sizes.

Continue

2

Select a source



Glide Tables

Start with Glide's built-in data source. Best for performance.



Google Sheets

Pick a Google Sheet from your Google Drive.



Excel

Select an Excel Workbook as your data source.



Airtable

Select an Airtable Base as your data source.



Import a file from your computer

XLSX, CSV, or ODS files. Maximum size 5MB.

Back

Continue

3

Select a file

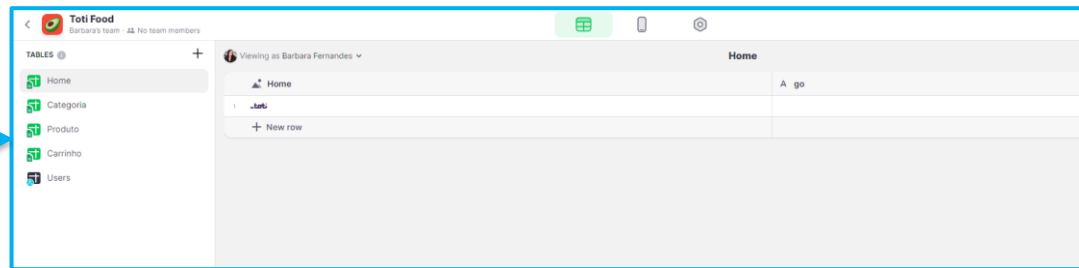
Name	Owner	Last modified
TotiFood	me	12:11 AM
Dados	me	Sep 28, 2022
Sheets for Copy of Delivery tutor	me	Sep 23, 2022
CompanyMarket	me	Sep 21, 2022
Restaurant	me	Sep 21, 2022
Formulário sem título (repostas)	me	Aug 2, 2022

Select

Cancel

4

Mobile: Aula 3 !



Edição das planilha excel

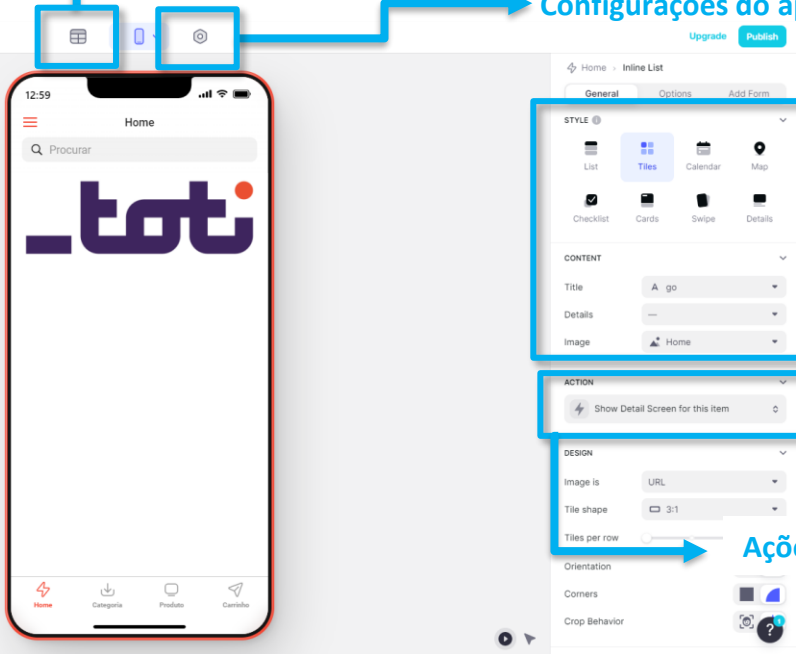
Configurações do app, como cor, nome e icone...

Editar componentes

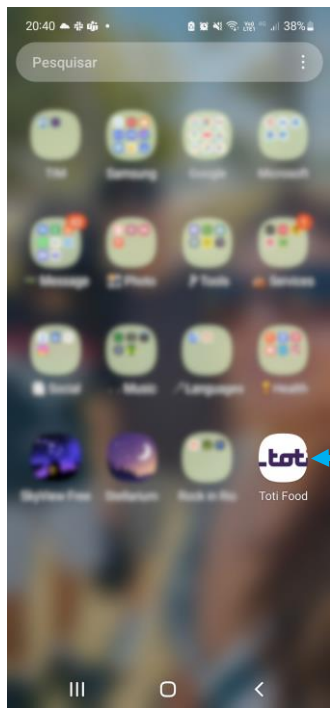
Ações como salvar ou deletar itens

Adicionar novas páginas

Adicionar novos componentes



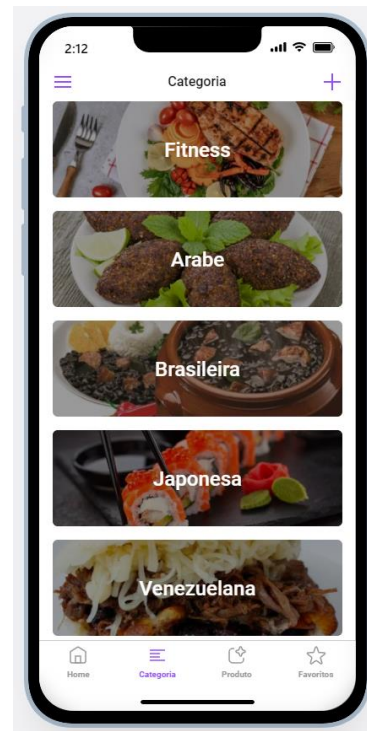
Mobile: Aula 3 - O nosso Toti Food!



Aplicativo no aparelho celular.

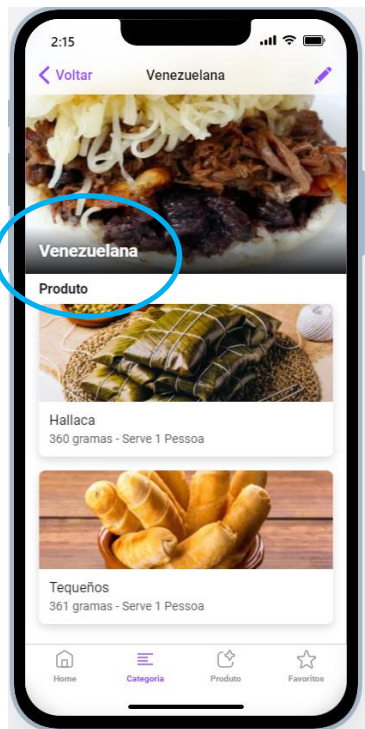


Home do nosso aplicativo.



Menu - Categorias de refeição.

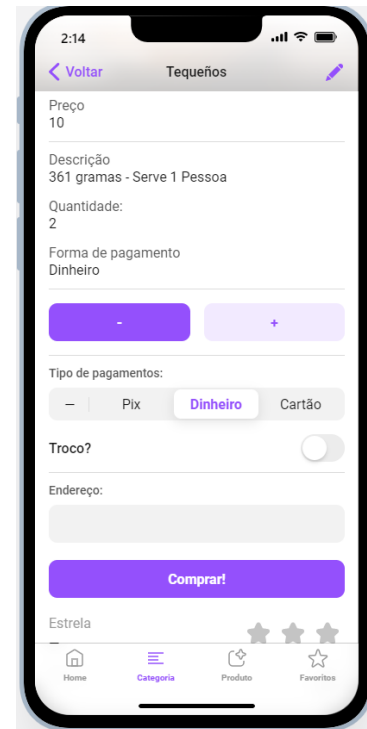
Mobile: Aula 3 - O nosso Toti Food!



Tipos de refeição categorizados.



Detalhes da refeição escolhida.



Formas de pagamento.

Mobile: Aula 3 - Referências!

Bora turbinar nosso conhecimento em Low e no-Code?

Glide Academy - Harry:

<https://www.youtube.com/c/GlideAcademyHarry>

CRIE APLICATIVOS MOBILE SEM CÓDIGO COM XANO (Back-end - BD) + APPGYVER (similar ao Glider - Front):

<https://www.youtube.com/watch?v=IOfczrQ2PH4&list=LLNO4duwbfrIzwA8jKLQFmGg&index=53>

Como Criar APPs Nativos Android e iOS sem Programação com Adalo – Introdução:

<https://www.youtube.com/watch?v=K2oIPXBXINU&list=LLNO4duwbfrIzwA8jKLQFmGg&index=39>

Mobile: Aula 4 - Flutter!



Mobile: Aula 4 - Flutter!

Flutter é um framework (ferramenta) de desenvolvimento com foco multiplataforma em dispositivos móveis. Criado pela Google, é bastante utilizado no mercado e, mais recentemente, permite a criação de aplicações para desktop (Linux, Windows e macOS). Vamos entender mais sobre isso?

Na prática, o Flutter é uma ferramenta para desenvolver aplicativos em diferentes plataformas - Android e iOS - ao mesmo tempo com um único código. Logo, seus principais benefícios são a versatilidade, menor curva de aprendizado e agilidade.

Mas... O que é um framework?

O framework pode ser simplesmente definido como uma ferramenta. Em uma definição mais aprofundada, trata-se de um pacote de códigos de bibliotecas pré-prontas que facilitam o desenvolvimento do projeto.

Mobile: Aula 4 - O Flutter é uma linguagem de programação?

A resposta é simples: não! O Flutter não pode ser considerado uma linguagem de programação como Dart, Java e .net; trata-se de um framework, conforme vimos.

Qual linguagem o Flutter utiliza?

O Flutter utiliza o **Dart**, também criada pela Google, uma linguagem otimizada para dispositivos clientes (ou client-side) multi plataforma.

O objetivo dela é ser a mais produtiva possível e também a mais flexível para servir como base para diversos frameworks de desenvolvimento de aplicativos.

Mobile: Aula 4 - Mas... O que vem a ser o Dart?

O Dart é uma linguagem de programação fortemente tipada inicialmente criada pela Google em 2011. A missão inicial do Dart era substituir o JavaScript para desenvolvimento de scripts em páginas web. Porém, com a evolução da linguagem e com o passar dos anos, ela hoje pode ser considerada uma linguagem multi-paradigma, embora a linguagem apresente fortes estruturas típicas de linguagens orientadas a objeto.

A sintaxe do Dart

O Dart possui uma sintaxe com estilo baseado no C. Isso faz com que sua sintaxe seja muito similar à linguagens atualmente populares, como Java e C#. Porém, o Dart tenta reduzir um pouco os ruídos característicos de linguagens baseadas no C.

Um simples “Hello World” em Dart poderia ser escrito da seguinte forma:

```
main() {  
  print('Hello World!');  
}
```

Mobile: Aula 4 - Comparação Flutter vs Dart:

	Flutter	Dart
Descrição	SDK de UI de código aberto	Linguagem de programação do lado do cliente para aplicativos da web e móveis
Categoria	Framework	Linguagem de programação
Linguagem de programação	Dart	Dart
Data de Lançamento Inicial	2017	2013
Desenvolvedor	Google	Goole
Open Source	Sim	Sim
Free to Use	Sim	Sim
Licença	BSD 3-Clause	BSD 3-Clause
Vantagens	Mesma IU em várias plataformas Desempenho nativo Motor de Renderização Próprio	Fácil de aprender Alta performance Estabilidade
Aplicativos populares	Philips MGM Resorts ByteDance	Flutter

Mobile: Aula 4 - Criação:



O Flutter é um framework desenvolvido pela Google para criação de aplicações móveis multiplataforma (Android e iOS) utilizando uma única base de código e muito clamado por sua facilidade em desenvolver aplicações com uma interface intuitiva e agradável. Para a criação destas interfaces, o Flutter utiliza um conceito de **Widgets**, como veremos neste artigo.

Mas... O que são widgets?

Toda interface de um aplicativo desenvolvido com o Flutter é criada com base em um conjunto de Widgets. Basicamente, um widget é um componente visual para definir a interface de um aplicativo.

Pense em uma tela do aplicativo como um LEGO, onde cada widget é uma pequena peça e, ao final, este conjunto de peças representará uma interface completa:

Mobile: Aula 4 - Criação > Widgets:

Flutter divide seus widgets em duas categorias:

Layout:

Widgets responsáveis por determinar a organização e posicionamentos de outros widgets. Estes widgets servem para delimitar áreas em nossas telas que serão preenchidos por outros widgets;

Scaffold: Um widget responsável por criar um layout “padrão” para o app, contendo uma appBar e o conteúdo da tela.

Stack: Widget responsável por “empilhar” widgets na tela do topo ao final da interface.

Container: Widget responsável por comportar novos widgets de layout. O Container é muito utilizado para comportar widgets em sua estrutura separando-os com bordas.

Interface:

Widgets responsáveis por criar componentes visuais que serão exibidos para os usuários. Estes widgets servem para determinar os componentes que comporão a interface do aplicativo.

*O Flutter possui centenas de widgets de interfaces documentados e prontos para serem utilizados em nossos aplicativos. Destes, podemos dar destaque para 2 conjuntos de widgets amplamente utilizados, o **Material** e o **Cupertino**. Basicamente, o Material é responsável por estilizar os widgets de interface com base nos padrões do **Material Design da Google**. Já o Cupertino é responsável por estilizar os widgets de interface com base nos **padrões de design do iOS**.*

Mobile: Aula 4 - Mas... O que é 'Material Design da Google'?

Em 2014, o Google desenvolveu uma nova versão — até então — do seu sistema operacional Android e criou uma nova metodologia de design, que foi batizada de Material Design.

Hoje em dia essa é uma das maiores tendências do design.

Pensado para ser fluido, natural, intuitivo e de simples compreensão, o Material Design possui diversas particularidades e fundamentos, que serão explicados ao longo deste post.

Então sem mais delongas, vamos aos conceitos essenciais!

Função:

O Material Design tem como objetivo sintetizar os conceitos clássicos de um bom design com a inovação e possibilidades trazidas com a tecnologia e a ciência.

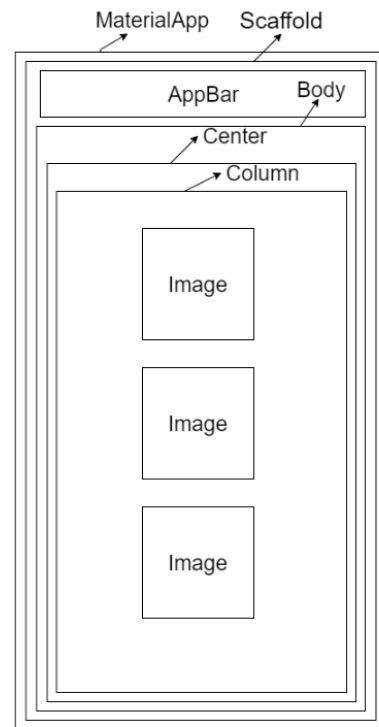
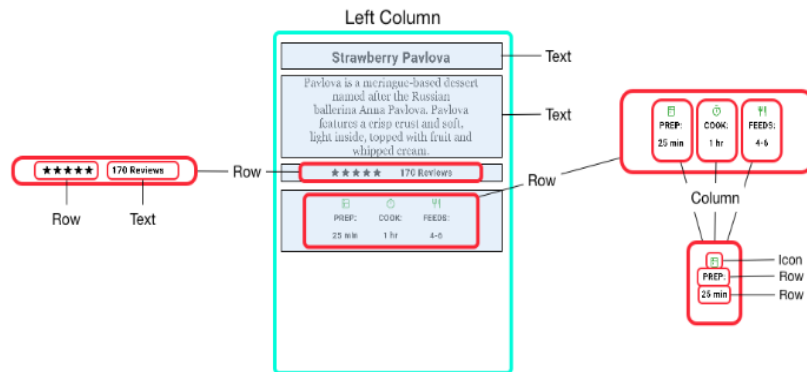
Ele proporciona uma experiência uniforme através de diversas plataformas diferentes, sejam smartphones, computadores ou relógios inteligentes.

Mobile: Aula 4 - ... E ...'Padrões de design do iOS'?

O padrão de delegação esta entre os padrões mais comuns de desenvolvimento em iOS e OS X. É um padrão simples que é fortemente usada pelos frameworks da Apple e mesmo o mais simples aplicativo iOS aproveita da delegação para funcionar. Vamos começar olhando à definição da delegação.

Layout + Interface:

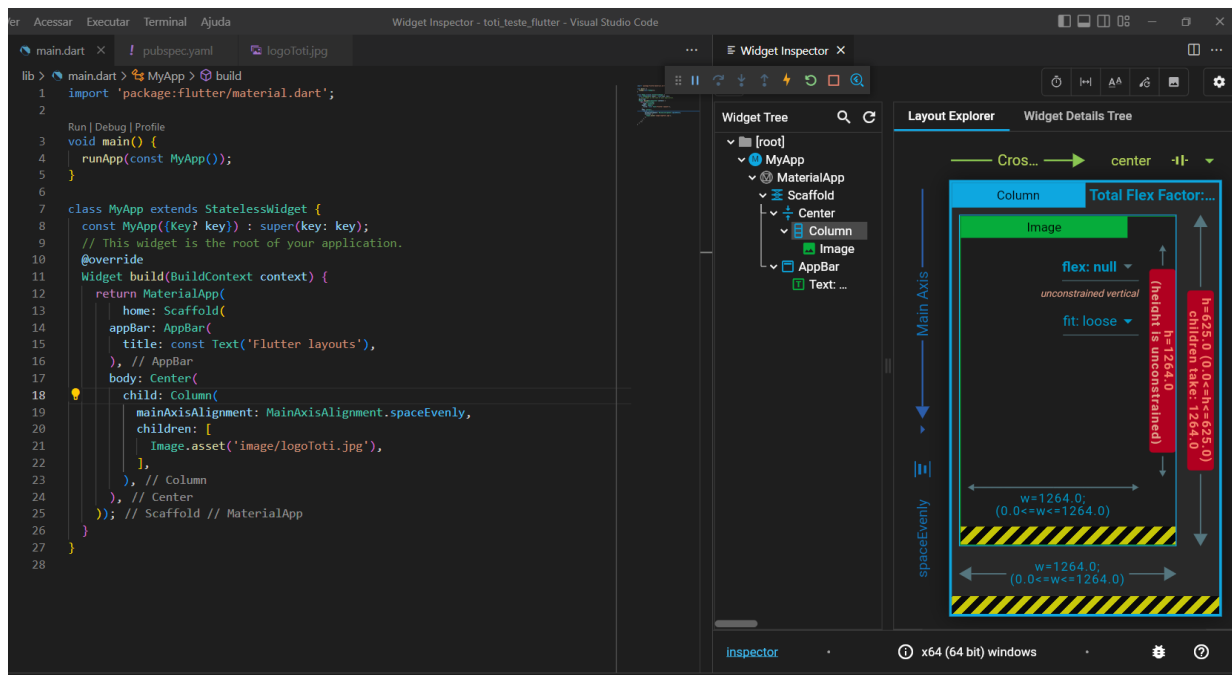
Como vimos, os widgets de layout e interface são utilizados em conjunto. Basicamente, a construção de interfaces são compostos por widgets de interface inseridos em widgets de layout, como pode ser visto na imagem abaixo:



Mobile: Aula 4 - Flutter , uma “linguagem” visual:

Esse é uma imagem de um código no VS code. A janela à esquerda esta a “**Widgets Inspector**” que nos ajuda visualmente a entender o conceito de **Widgets** e a estrutura de desenvolvimento em Flutter ;)

```
Import;  
MyApp;  
MaterialApp;  
Scaffold;
```



Mobile: Aula 4 - Como utilizar o Flutter?

Vamos considerar que você tenha quase tudo o que é necessário para começar o desenvolvimento com Flutter (ambiente de desenvolvimento com emulador e IDE).

Como podemos **iniciar a escrita da nossa aplicação?**

Precisamos **instalar**, no computador, as seguintes ferramentas:

- 1) Flutter SDK:** o arquivo que permite utilizar a ferramenta;
- 2) Android Studio e/ou xCode:** contém as ferramentas de build para testar e subir a aplicação em dispositivos Android e iOS (além de servirem como IDEs);
- 3) IDE:** é o programa onde você vai escrever o código. Há diversas opções de IDE no mercado para escolher: Visual Studio Code (que utilizaremos neste artigo), Android Studio, xCode;
- 4) Emulador:** necessário para rodar e testar o aplicativo que pretendemos criar.

<https://docs.flutter.dev/get-started/install>

Mobile: Aula 4 - Atenção ao fazer um app !

Quais são as principais diferenças entre os formatos de arquivos abertos e os proprietários?

Formatos abertos:

domínio público usada para a descrição e armazenamento de dados digitais, **livre de restrições legais** para seu uso. O principal objetivo dos formatos abertos é garantir o acesso aos dados a longo prazo sem inseguranças jurídicas ou restrições devidas às especificações técnicas: em outras palavras, garantir a interoperabilidade. Outro objetivo do formato aberto é incentivar a concorrência, ao invés de permitir que um único fabricante detenha o controle sobre um formato proprietário.

Formatos proprietários:

Os formatos proprietários são, na verdade, controlados e definidos por **interesses privados**.

Com o formato proprietário, o usuário corre o risco de perder todas as informações contidas no arquivo caso o software pare de funcionar ou fosse descontinuado, já que não teria outro sistema capaz de processá-las. Todas as informações coletadas nos arquivos públicos (órgãos, instituições, administração, etc.) se tornariam ilegíveis e inutilizáveis em muito pouco tempo.

A título de exemplo, o formato MP3 é proprietário mas também aberto porque, se bem patenteado, suas especificações são públicas. Tornar públicas as especificações técnicas é uma maneira de difundir o formato e torná-lo um padrão internacional.

Mobile: Aula 4 - Atenção ao fazer um app !

Tipos	Formatos
Texto	.odt
Planilha	.ods
Apresentação	.odp
Áudio	.mp3, .FLAC, .ogg
Vídeos	.mkv, .webM, .mp4(codec x264)
Webpage	HTML5
e-book	epub
Fórmula Matemática	MathML
Imagens	PNG, SVG

Exemplos de formatos proprietários:

- **DWG** - AutoCad Drawings
- **SWF** - Shockwave Flash
- **DOC** - formato proprietário
- **PPT** - formato proprietário de apresentação de slides





MATONDO!

(OBRIGADO)

_toti



@toti.diversidade



/school/toti-diversidade



/toti.diversidade

[_https://totidiversidade.com.br](https://totidiversidade.com.br)