

Treinamento Teknisa

Módulo UX/UI
Camila Guilherme



Perguntas:
<https://www.sli.do/>
cód. 740854

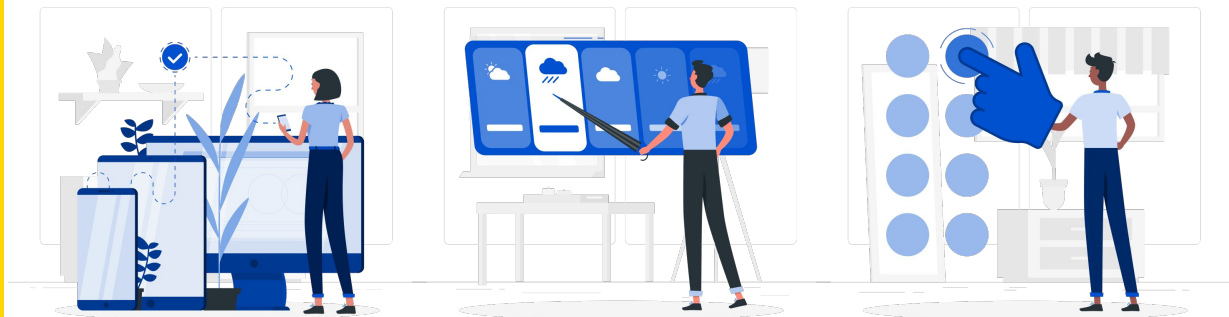
- I. A importância do UX e UI no desenvolvimento
- II. Como interpretar o design de telas e interfaces em arquivos Figma

I. A importância do UX e UI no desenvolvimento

O que é UX?

UX é uma sigla para “User Experience”, que significa “experiência do usuário”.

Na prática a UX engloba mais do que a aplicação que está sendo desenvolvida, pois a experiência do usuário depende, também, de fatores externos, como:

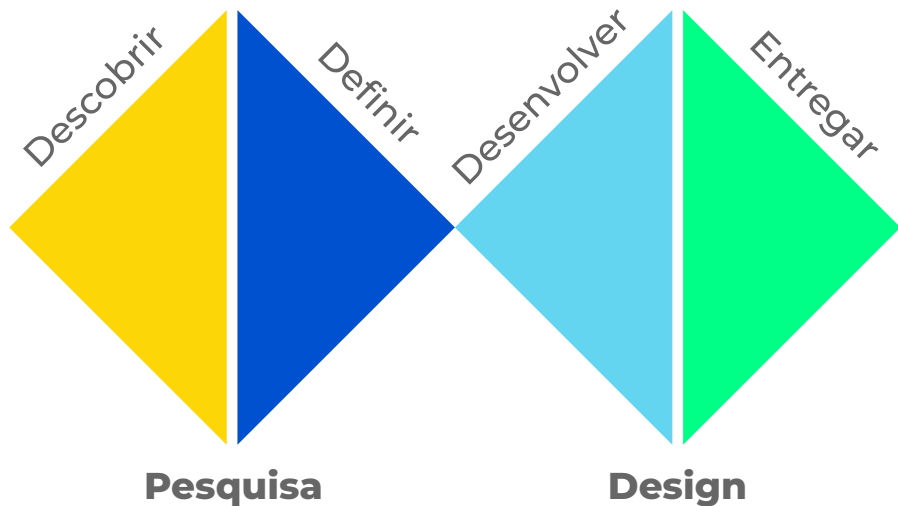


O dispositivo onde a aplicação está sendo executada;

as condições em que esta pessoa está utilizando o dispositivo;

e por quem essa aplicação está sendo utilizada.

Processo de um projeto UX/UI



A pesquisa de UX deve começar juntamente com a definição do produto, e continua mesmo após o produto ser lançado ao público.

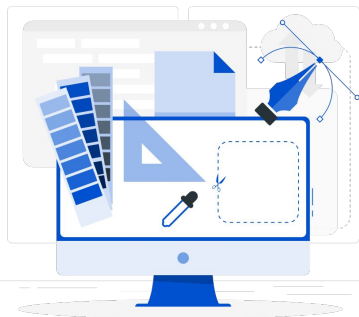
Dentro dessa pesquisa são feitos os levantamentos dos problemas que o produto busca resolver, a definição de soluções ideais e seus competidores no mercado, a proposta de padrões e elementos para interface, os testes de usabilidade, dentre outros.

No processo de UX são utilizadas várias metodologias diferentes, uma delas, ilustrada ao lado, é o "Diamante duplo".

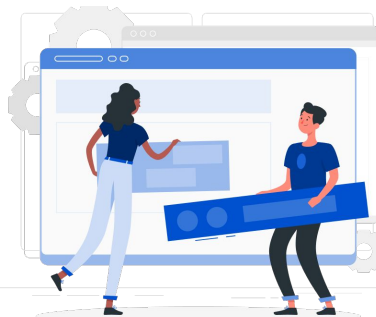
E o que é UI?

UI é uma sigla para “User interface”, que significa “interface do usuário”.

A UI é a parte de uma aplicação utilizada na interação de um usuário com o programa, e é composta por elementos, gráficos ou não, fazem parte, entre outras, do design de UI, as decisões sobre:



A identidade visual,
as cores do aplicativo,
e padrões de forma;



As telas, botões, barras
de navegação, input, e
demais elementos;



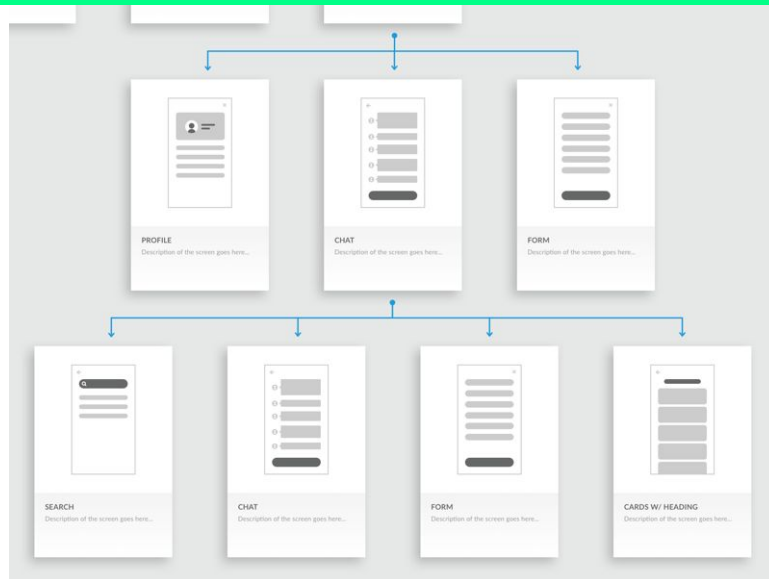
O posicionamento e
tamanho desses
elementos na tela.

E de onde vem UX/UI?

A UI existe em qualquer programa, mesmo que nenhuma pesquisa de UX tenha sido feita durante o desenvolvimento.

Durante o processo de UX são desenhados os fluxos de navegação, a organização da informação na tela, os tipos de input que devem ser utilizados, dentre muitos outros elementos da UI.

Com a pesquisa UX são definidas as melhores formas de montar a interface, para criar a melhor experiência durante a utilização do programa.



Um fluxo de telas é uma das formas de determinar a navegação em uma aplicação, e ajuda a definir todas as telas e elementos necessários para a interação.

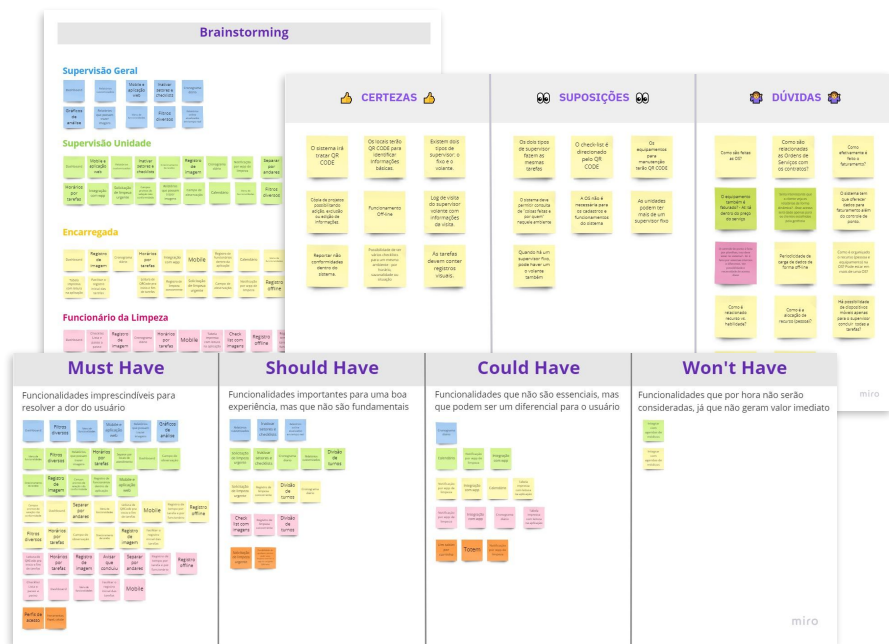
E por quê eu preciso saber disso?

Existem muitos exemplos de programas que “deram certo” sem uma pesquisa UX, mas eles ainda se beneficiam dela pelo aperfeiçoamento e no desenvolvimento de atualizações constantes.

A pesquisa UX é essencial para evitar que sejam criadas soluções que não entregam nenhuma vantagem ou diferencial ao público-alvo, no contexto de um mercado já saturado, ou que um programa venha oferecer uma solução para um problema sem que haja demanda.

A UX foca na usabilidade, otimizando o desenvolvimento pois evita o retrabalho, e busca garantir a qualidade na experiência de interação com um produto antes mesmo de seu lançamento.

O valor da pesquisa UX para o desenvolvimento



Profissionais de UX estudam, dentre outras coisas, o mercado para o qual a solução é voltada, os produtos competidores, a jornada do usuário, e as melhores práticas de usabilidade direcionadas ao público-alvo do projeto, antes mesmo de iniciarem o desenho das telas.

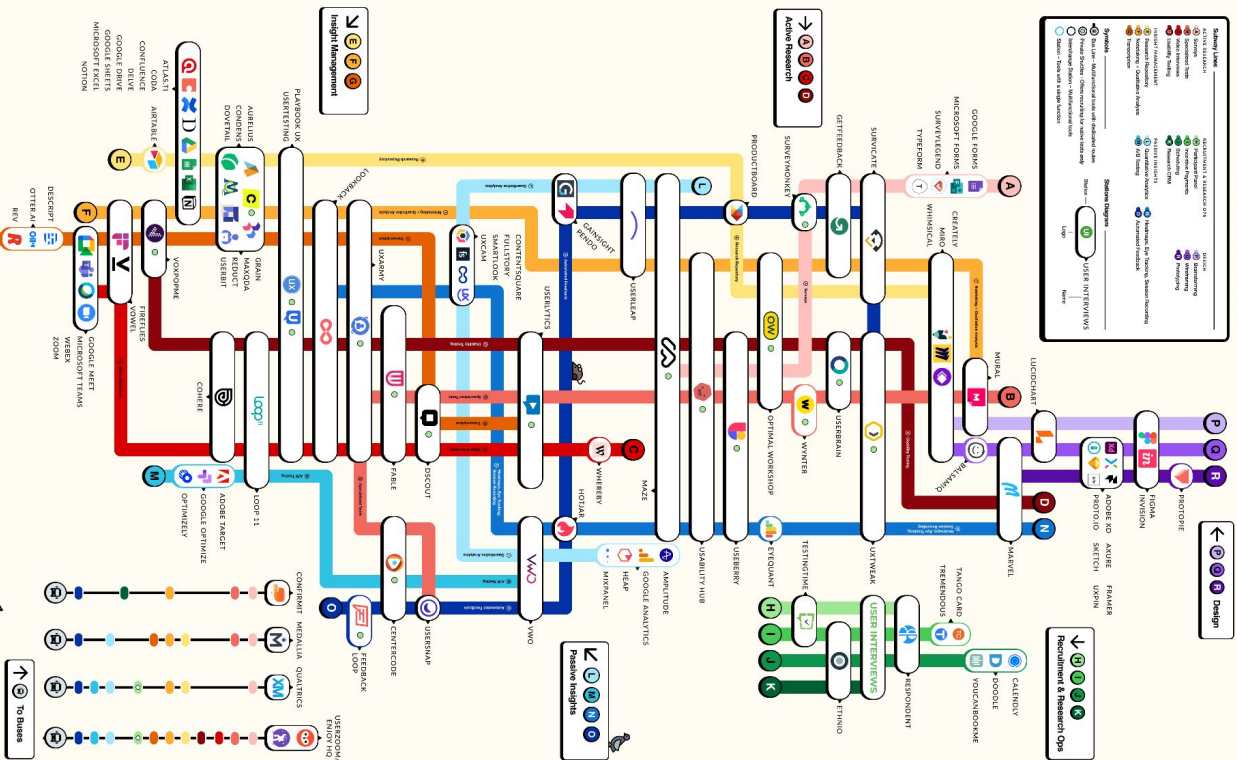
Em aplicações que já estão publicadas a pesquisa UX faz análises quantitativas e qualitativas com os usuários, para melhorar a usabilidade e sugerir novas funcionalidades quando necessário.

Exemplos de atividades da pesquisa UX, brainstorm de funcionalidades, matriz CSD e matriz MoSCoW.

em ao lado mostra
dade de atividades
profissionais de
cutam, e as muitas



Tools Map



Pontos de contato UX/UI x Desenvolvimento

Viabilidade técnica:

Ao definir o problema, deve-se estabelecer se a solução pode ser executada com a tecnologia que está disponível no momento.

Para isso os time de UX/UI e desenvolvimento devem verificar a viabilidade em conjunto, definindo dispositivos, linguagem de programação, e frameworks que serão utilizados.

Neste ponto o desenvolvimento é quem define a tecnologia de acordo com os requisitos de funcionalidades que o UX apresentou.

MVP:

A definição do “Minimum Viable Product”, ou “produto mínimo viável” deve ser também feita em conjunto, pois é necessário que o produto seja montado com o objetivo principal em mente, para que o problema seja solucionado desde a primeira iteração do produto, e que as próximas iterações aconteçam para melhorar e otimizar a solução existente.



Protótipo



MVP



Produto

Biblioteca de componentes:

Uma biblioteca de componentes permite que a equipe de UX monte os protótipos, telas e interações utilizando os componentes que estão disponíveis para a equipe de desenvolvimento.

Isso garante que o produto final possa ser o mais fiel possível ao que foi desenhado. Quando uma equipe de UX/UI utiliza uma biblioteca de componentes certa para o framework de front-end já definido, é necessário que os elementos utilizados no protótipo sejam aqueles existentes e já documentados para desenvolvimento.

Isso não impede que novos componentes sejam desenhados, neste caso a equipe de UX/UI deve tomar a frente das decisões de design do novo componente, tendo em mente, sempre, a viabilidade técnica.

Dados utilizadas na aplicação:

As informações dos dados utilizados na aplicação também devem ser discutidas com a equipe de UX/UI, para tomar as melhores decisões de como demonstrar as informações (tabelas, cards, listas, etc.) e de como elas serão imputadas (select, checkbox, switch, etc...) pois a interação do usuário com esses elementos também faz parte da experiência de uso da aplicação.

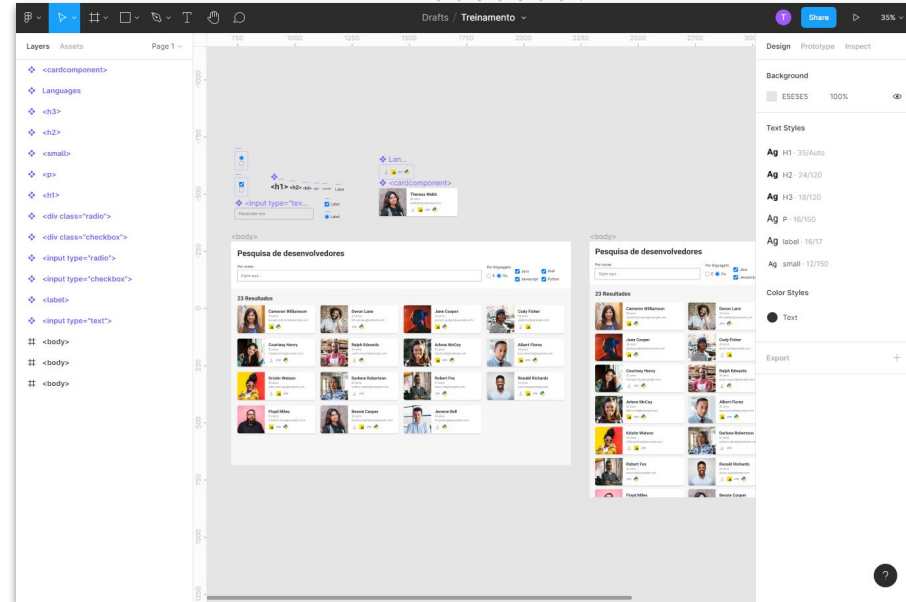
Por esse motivo é importante que o profissional de UX/UI responsável pelo desenho das telas também entenda sobre como os dados serão coletados e utilizados no backend da aplicação.

Definição do wireframes, navegação e protótipos:

A equipe de UX/UI tem à sua disposição várias ferramentas para fazer o “hand off”, a entrega, do projeto, após o desenho dos fluxos, dos wireframes e das telas, é montado o protótipo navegável, muitas vezes em alta definição.

Esse protótipo deve ser utilizado na criação e programação das telas, e as alterações feitas durante o desenvolvimento deverão ser informadas à equipe de UX/UI para que sejam refletidas em projetos consequentes.

Uma destas ferramentas que pode ser utilizada desde a criação do wireframe até a prototipação e entrega é o Figma, que vamos conhecer na próxima etapa.



II. Como interpretar o design de telas e interfaces em arquivos Figma

Um projeto Figma

Vamos começar abrindo um projeto já montado, para acessá-lo você precisará de uma conta na plataforma Figma www.figma.com.

Utilize os links ou os QR codes abaixo para acessar o projeto e o protótipo navegável



Projeto:
<https://www.figma.com/file/h4z36ouFG2dR0TIIBI3S3t/>



Protótipo:
<https://www.figma.com/proto/h4z36ouFG2dR0TIIBI3S3t/>

Navegando em um projeto

Você vai se deparar este projeto, que foi montado especialmente para esse treinamento.

The screenshot shows the Figma design tool interface for a project named "Treinamento" in "View only" mode. The interface includes a left sidebar with a component library, a central canvas with a coordinate grid, and a right sidebar with a comment panel.

Component Library (Left Sidebar):

- Modal
- Button
- Card
- Languages
- H3
- H2
- small
- P
- H1
- Radio
- Checkbox
- Radio
- Checkbox
- Label
- Text
- Mobile
- Tablet
- Desktop

Canvas (Center):

The canvas displays three mobile app mockups for a "Pesquisa de desenvolvedores" (Developer Survey) app. The mockups show a list of developer profiles, each with a profile picture, name, and a "Follow" button. The profiles are arranged in a grid, and the app interface includes a header with the title "Pesquisa de desenvolvedores" and a search bar.

Right Sidebar (Comment Panel):

The right sidebar shows a comment panel with a search bar and a comment input field. The comment input field contains the text: "Give feedback, ask a question, or just leave a note of appreciation. Click anywhere in the file to leave a comment."

Menu do projeto e ferramentas disponíveis

Nome do projeto

Pessoas que estão ativas no projeto, botão de compartilhamento, visualização de protótipo e zoom + opções de visualização

The screenshot shows the Axure RP software interface. At the top, a dark toolbar contains icons for project management and navigation. Below the toolbar, the main workspace is divided into three sections: a left sidebar for project elements, a central canvas for the prototype, and a right sidebar for actions and settings. The left sidebar lists various elements like Modal, Button, Card, Languages, H3, H2, small, P, H1, Radio, Checkbox, Label, Text, Mobile, Tablet, and Desktop. The central canvas displays a prototype of a web application with a search bar, a list of developers, and a modal window. The right sidebar includes a search bar, a comment section, and a list of actions like Comment, Inspect, and Export. The interface is annotated with green arrows and text boxes explaining the different parts.

Page 1 ▾ Páginas do projeto

Modal

Modal

Button

Modal

Card

Languages

H3

H2

small

P

H1

Radio

Checkbox

Radio

Checkbox

Label

Text

Mobile

Tablet

Desktop

Tela (Canvas) do projeto

Treinamento ▾ View only

U Share ▶ 20% ▾

Comment Inspect Export

Search

Give feedback, ask a question, or just leave a note of appreciation. Click anywhere in the file to leave a comment.

Abas de ação: Comentário, Inspeção e Exportação.

Camadas (Layers) da página

◇ Modal

◇ Modal

❖ Button

≡ Modal

❖ Card

❖ Languages

❖ H3

❖ H2

❖ small

❖ P

❖ H1

❖ Radio

❖ Checkbox

❖ Radio

❖ Checkbox

❖ Label

❖ Text

≡ Mobile

≡ Tablet

≡ Desktop

Camadas | Layers

A tela de layers funciona como uma árvore, a posição vertical dos elementos define a posição z dele, ou seja, os elementos na parte de cima da lista sobrepõem todos elementos abaixo.

A posição horizontal indica elementos que estão “dentro” de outros. Elementos que estão no mesmo nível horizontal são elementos “irmãos”, e seguem a regra de sobreposição entre eles.

O símbolo ao lado do layer indica o tipo de camada, por exemplo:

❖ **Componente principal**

◆ **Variante de um componente**

◇ **Instância de um componente**

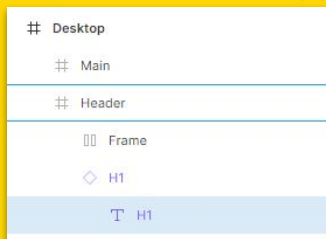
Frame

▢ **Frame com auto layout (horizontal)**

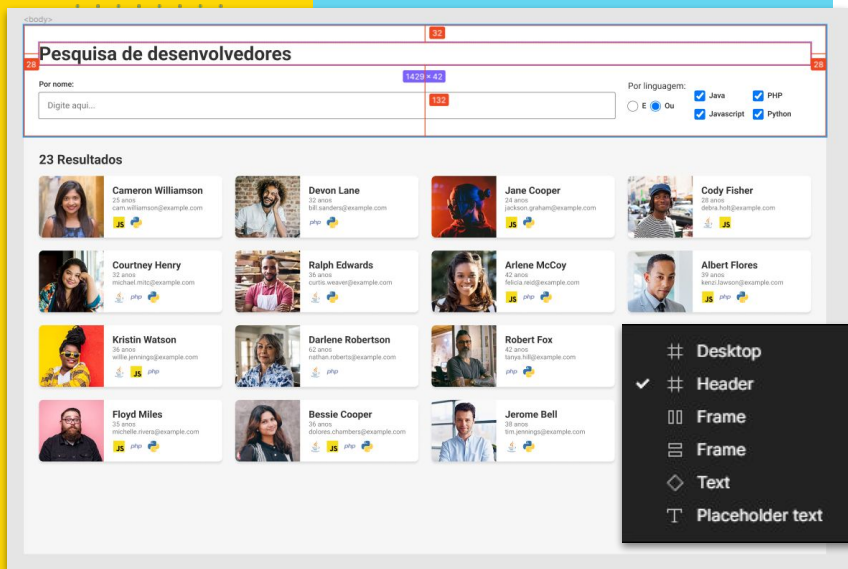
□ **Grupo**

T **Texto**

Tela | Canvas



Com um layer selecionado, ao passar o mouse sobre outro elemento você verá marcadores em vermelho com a distância entre eles.



A tela, ou canvas, é onde estão os elementos visuais. Ao passar o mouse por cima de um elemento você (geralmente) poderá selecionar o elemento mais profundo na posição onde o seu cursor está.

Segure a tecla **CTRL** e **clique** para selecionar a camada de nível mais alto, caso essa camada seja um frame com outros frames dentro, você selecionará o frame no nível inferior.

A forma mais simples de selecionar um frame “superior” na tela, é clicando no nome dele, mas prefira selecioná-lo na lista de camadas

Segure **CTRL** e **clique com o botão direito** para ver uma a lista de elementos que estão na posição do cursor.

CommentInspectExport

T Text H1

Parent component

H1

Properties

Width1429px

Height42px

BlendPass through

↔ Fill container

↕ Hug contents

Content

Pesquisa de desenvolvedores

Typography

AgH1

FontRoboto

Weight700

Stylenormal

Size35px

Line height42px

Colors

Hex

Text

Com um elemento selecionado, neste caso o título da tela “Pesquisa de desenvolvedores”, ao clicar na aba “Inspect”, o Figma informará várias propriedades do elemento selecionado, neste caso:

- **Tipo:** Texto;
- **Componente pai:** H1;
- **Propriedades:**
 - **Tamanho:** 1429x42 px,
 - **Mesclagem:** Passar;
 - **Modelo de dimensionamento:** Preencher container (horizontal) e abraçar conteúdo (vertical).
- **Conteúdo:** Pesquisa de desenvolvedores;
- **Tipografia:** H1 (Roboto, 700, normal, size 35px, line-height 42px)
- **Cores:** Texto (#333333);
- **Código:** CSS (ele possui opção para código Android e iOS também).

ColorsHex

Text

#333333

Code

CSS

<>

```
/* H1 */

position: static;
left: 0%;
right: 0%;
top: 0%;
bottom: 0%;

/* H1 */
font-family: Roboto;
font-style: normal;
font-weight: bold;
font-size: 35px;
line-height: 120%;
/* identical to box height, or 42px */

/* Text */
color: #333333;

/* Inside auto layout */
flex: none;
order: 0;
flex-grow: 1;
margin: 0px 10px;
```

O código

Code

CSS ▾



/* H1 */

position: static;

left: 0%;

right: 0%;

top: 0%;

bottom: 0%;

/* H1 */

font-family: Roboto;

font-style: normal;

font-weight: bold;

font-size: 35px;

line-height: 120%;

/* identical to box height, or

42px */

/* Text */

color: #333333; ■

/* Inside auto layout */

flex: none;

order: 0;

flex-grow: 1;

margin: 0px 10px;

O código que o Figma cria é essencial para que as cores, efeitos e estilos tipográficos do projeto sejam respeitados, para isso o designer cria os estilos que serão reaproveitados em todo o projeto.

Esses estilos são comentados no código, como `/* H1 */` e `/* Text */`, que foram identificados na tela anterior (em tipografia e cores).

Ele também coloca códigos para alguns componentes e propriedades, como o primeiro `/* H1 */`, que é um componente, e `/* Inside auto layout */` que é o código para a utilização de flex, que cria o efeito do auto layout.

Mas é preciso ficar atento, muitos códigos do Figma são supérfluos ou até mesmo incorretos. Neste caso o componente `<h1>`, que é o componente de texto, está com a posição absoluta, esse tipo de posicionamento é muito específico e não deve ser aplicado neste caso.

Auto layout, grid e flex

Essas opções aparecem somente para os designers, que precisam combinar layouts horizontais e verticais para criar grids e tabelas

Muitas vezes o auto layout é utilizado pelos designers de UI para facilitar a criação de layouts responsivos, por isso o código CSS não deve ser utilizado cegamente, sem o conhecimento do protótipo ou do comportamento de cada componente.

O auto layout é “traduzido” pelo Figma pela utilização do Flexbox, mas pode ser também implantado como um Grid. Entretanto aplicação do auto layout não é 100% compatível com Flexbox ou Grid, por isso deve ser interpretado e adaptado pelo desenvolvedor, pois a aplicação deles é diferente e depende de cada caso.

O auto layout deve ser aplicado no componente “pai”, ou “container”, e oferece opções como direção (vertical ou horizontal), distância entre os elementos (gap), preenchimento (padding), posicionamento dos elementos e distribuição, os itens dentro de um auto layout podem preencher o container, crescendo para preencher o espaço disponível no container, e os containers podem abraçar o conteúdo, mudando, assim, de tamanho de acordo com os itens.

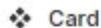
Componentes, instâncias, variações e descrições



Componentes são elementos que servem de modelo e podem ser replicados (criando instâncias) que sempre irão refletir as alterações feitas no componente original. Desta forma é possível criar botões, inputs, e outros elementos, que serão consistentes em todo o projeto.



Um componente pode ser feito utilizando outros componentes dentro dele (como o card ao lado), e todos os componentes podem possuir variações que dependem de propriedades, como , por exemplo, indicar uma mudança de estado, como o “radio” ao lado.



Card

Componente de card para a tabela de resultados. Recebe a foto, o nome, a idade, o e-mail e as linguagens registradas de um desenvolvedor.

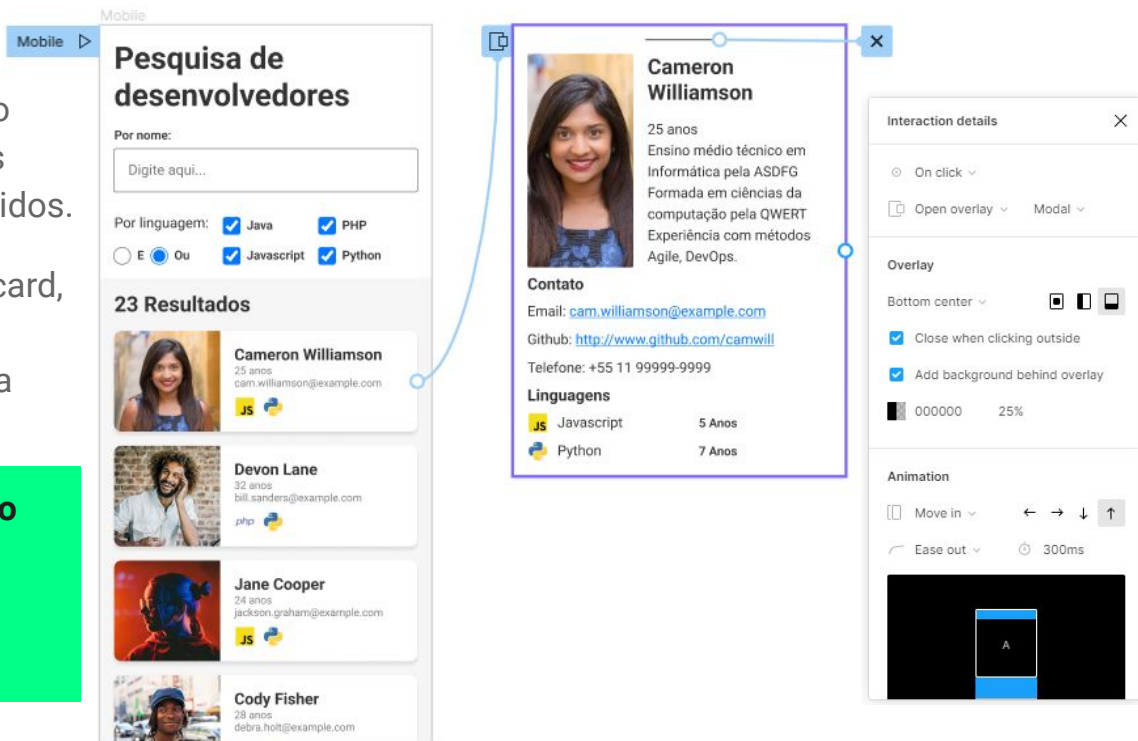
Os componentes possuem um campo para descrição, onde o Designer pode deixar notas de como o componente se comporta e como deve ser utilizado, e um campo de link para a documentação, caso ela exista.

Protótipo

Quando o protótipo de alta-fidelidade é montado, o designer pode entregá-lo com o fluxo de navegação das telas, muitas vezes com animações e comportamento, já definidos.

Neste caso o overlay aparece ao clicar no card, movendo-se de baixo para cima, e fecha quando se clica fora do card ou na pequena barra na parte superior do modal.

Infelizmente a visualização desse protótipo e a animação não são visíveis no modo de inspeção, então é necessário observar o comportamento na tela de navegação.

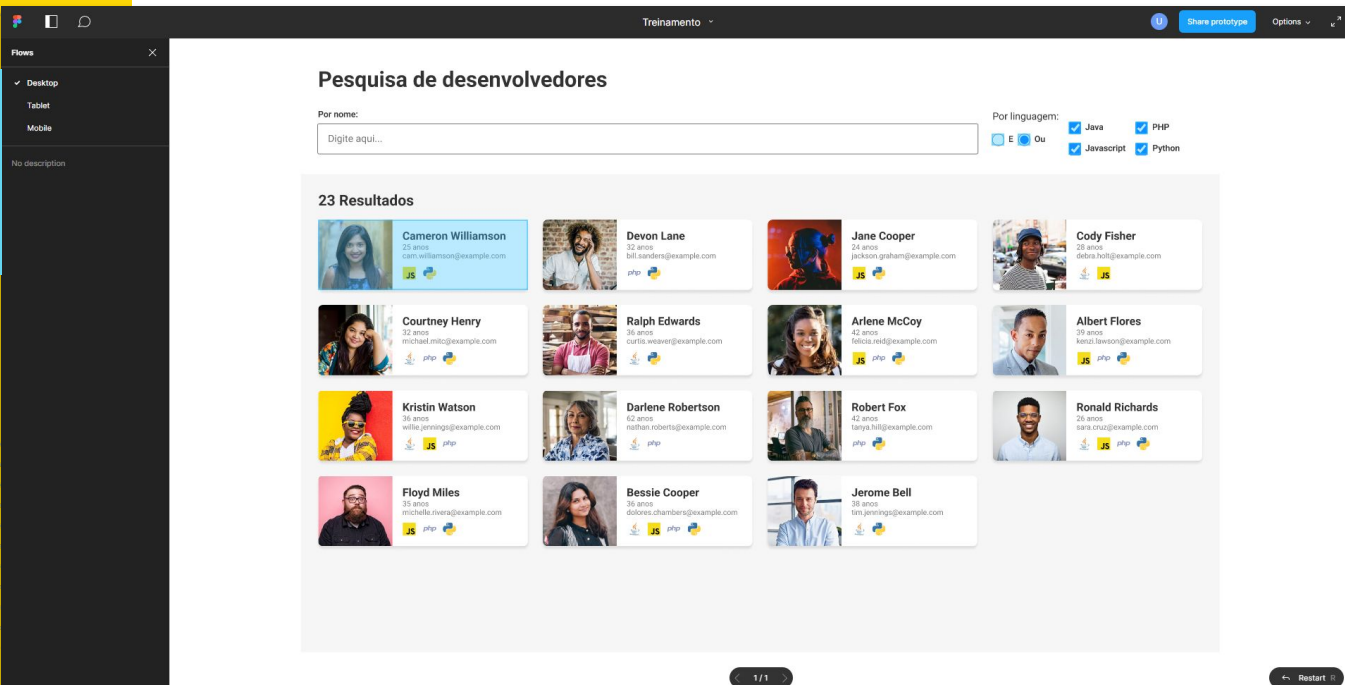


Navegação e fluxos

Essa aba à esquerda só aparecerá caso existam fluxos de navegação definidos

É possível também montar fluxos de navegação diferentes, neste caso, cada fluxo determina um dispositivo.

Ao clicar em qualquer lugar da tela de navegação (que não é um link), uma marcação em azul irá aparecer para indicar que aquela área é interativa.



E agora?

O objetivo final deste módulo é facilitar a interpretação de arquivos do Figma, para isso algumas perguntas devem ter respostas:

1. **Como selecionar, identificar e inspecionar elementos no Figma?**
2. **Como utilizar o painel de inspeção para entender os estilos e propriedades do elemento?**
3. **Como interpretar o protótipo para entender a navegação e as formas de interação com cada um dos elementos?**

Existem muitos termos que são utilizados neste módulo e ficarão mais claros com o conhecimento que será adquirido nos conteúdos à frente. Mas saber onde buscar as informações acima é essencial para a interpretação do design.

Esse projeto será utilizado módulos a seguir, então fiquem livres para copiá-lo e explorá-lo como quiserem.

Boa sorte!