



Desenvolvimento de Sistemas

Conceitos

Por definição, experiência é o ato de sentir, conhecer ou testemunhar algo. Quando alguém usa ou consome um objeto, um produto e/ou um serviço ou, ainda, interage com eles, surgem entendimentos, percepções e opiniões sobre a utilidade, a eficiência e a facilidade de uso. Assim, a experiência do usuário (ou, em inglês, *user experience* – UX) pode ser entendida como algo tão antigo quanto a própria existência humana.

A UX é uma abordagem humana para o *design* de um produto e tem aplicações físicas e digitais. O ponto crucial da UX envolve a **experiência completa do usuário**, desde o primeiro contato até o último. Já o *design*, no contexto de interfaces, é o **ato de planejar telas e interfaces**, ou seja, de criar o projeto.

Quando a UX se une com o *design* de interfaces, tem-se então a **interface gráfica do usuário** (ou, em inglês, *graphical user interface* – GUI). A GUI trata da utilização de técnicas de *design* e usabilidade voltadas à interação entre homens e máquinas por meio de softwares, sistemas ou aplicações. É um conceito da forma de interação entre o usuário do computador e um programa por meio de uma tela ou uma representação gráfica, visual, com desenhos, imagens etc., geralmente entendido como a “tela” de um programa.

Um bom *design* de interface aperfeiçoa o modo como os usuários utilizam um sistema de computação. As figuras 1 a 3 mostram exemplos de experiências ruins de usuário.



Figura 1 – Caneca com orelhas pontudas que machucam os olhos quando a pessoa a utiliza

Fonte: Farrington (2020)

A screenshot of a web-based form. At the top, there is a date selector showing "January 1 2017". Below it is a section labeled "Primary phone number*:" with a tooltip. The phone number input field is a grid of 10 input boxes, each containing a digit from 0 to 9. The first digit, '0', is highlighted with a blue selection bar. To the right of the input field, there is a label "Secondary phone number:" and another grid of 10 input boxes. At the bottom of the form, there is a note: "Fields with asterisks are required. Make sure your details are correct before you submit this application and we will email you on the progress." There is also a checkbox and a link for terms and conditions.

Figura 2 – Interface forçando o usuário a inserir um número de telefone com uma lista suspensa para cada dígito

Fonte: Farrington (2020)



Figura 3 – Faixa de pedestres com acesso a cadeiras de rodas interrompido por uma segunda calçada

Fonte: Farrington (2020)

Aspectos técnicos e subjetivos

O processo de criação de interfaces ou telas para determinado *software* passa por várias etapas. Quando se abordam aspectos técnicos, está-se falando sobre como, de fato, a interface será construída. É necessário levar em consideração a tecnologia e o *software* a serem utilizados, a utilização ou não da biblioteca de pacotes (como o Java Swing), a composição dos elementos na interface (campos de textos, *labels* – rótulos –, *radio buttons* – botões de opção –, tabelas etc.).

Tais requisitos são aspectos técnicos, que se relacionam menos com a estética e mais com a funcionalidade.

Por outro lado, há também aspectos artísticos e subjetivos na criação de interfaces, utilizando cores, formatos, posicionamentos em prol da beleza das telas ou do sistema em si, mas também transmitindo alguma mensagem ao usuário. Um *site* para um consultório médico, por exemplo, provavelmente terá tons de azul e branco, e não tons predominantes de preto e vermelho.

Considerando ambos os aspectos, que devem trabalhar em conjunto, pode-se praticar a **empatia**, que nada mais é do que se colocar no lugar do usuário e fazer uma reflexão:

- ◆ Consigo entender esta interface?
- ◆ A navegação está fácil?
- ◆ Os elementos estão distribuídos de maneira simples para que eu possa encontrá-los?

Para projetar uma **experiência** no usuário, é necessário **investigação e observação do comportamento do cliente**, para que assim você possa chegar a uma solução satisfatória. Como você conseguirá desenvolver uma interface para um cliente sem nem ao menos saber a área de atuação, os gostos e os potenciais clientes dele? De fato, é necessária uma análise crítica para que o resultado seja excelente e satisfatório.

Ademais, a relação entre arte e *design* é, muitas vezes, mal compreendida, e, embora um resultado de *design* possa ser artístico, os processos por trás dele são bem diferentes.

Aplicações e usabilidade

Tudo começa com a compreensão de quem são os usuários, incluindo seus objetivos, suas habilidades, suas preferências e suas tendências. Vale lembrar que aplicar o conceito de usabilidade em um projeto é garantir a eficiência e a facilidade de interação deste, verificando se existe ou não a necessidade de futuras melhorias.

Depois de entender quem são os usuários, é preciso seguir estes passos ao projetar a interface:

Mantenha a interface simples.

As melhores interfaces são quase invisíveis para o usuário. Então evite elementos desnecessários e utilize uma linguagem clara nos rótulos e nas mensagens.

The screenshot shows a user interface titled "Cadastro Simples com Listas". It features two input fields: "Nome:" and "Idade:", each followed by a text input box. Below these are two buttons: "Adicionar" and "Excluir". At the bottom is a table with two columns: "Nome" and "Idade". The table currently contains one row of data.

Nome	Idade

Figura 4 – Interface gráfica simples

Fonte: Senac EAD (2022)

Estabeleça consistência e use elementos de interface do usuário comuns.

Os usuários, ao identificarem elementos de interface comuns, sentem-se mais à vontade e podem realizar o trabalho mais rapidamente. Também é importante criar padrões de linguagem, *layout* e *design* em todo o *software* para ajudar a melhorar a eficiência. Uma vez que os usuários aprendem como fazer algo, eles devem ser capazes de transferir essa habilidade para outras partes do *software*.

Formulário de Cadastro

Nome	<input type="text"/>	
Contato	<input type="text"/>	
CPF	<input type="text"/>	
Endereço	<input type="text"/>	
Cidade	<input type="text"/>	
Deseja trabalhar na área de TI?	<input type="radio"/> SIM	<input type="radio"/> NÃO
<input type="button" value="Novo"/> <input type="button" value="Salvar"/>		
Nome	Contato	Cidade
<input type="text"/>		

Figura 5 – Exemplo de interface com elementos consistentes e devidamente alinhados

Fonte: Senac EAD (2022)

Seja objetivo no *layout* do *software*.

Considere as relações espaciais entre os itens no *software* e os estruturas com base na importância. A colocação cuidadosa de itens pode ajudar a chamar a atenção para as informações mais importantes.

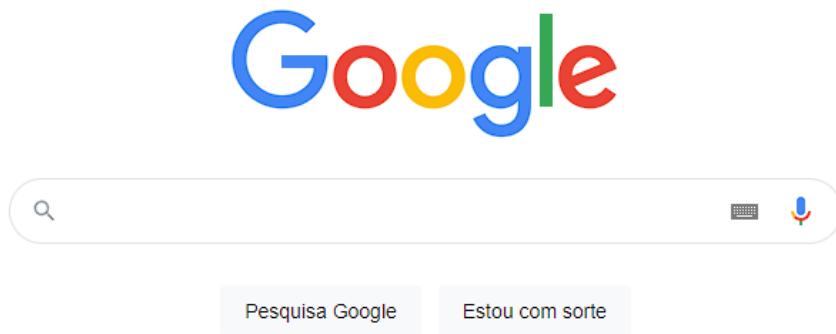


Figura 6 – Exemplo de *layout* simples e objetivo

Fonte: Google (2022)

Use cor e textura estratégicamente.

Você pode direcionar a atenção ou redirecioná-la para os itens usando cor, luz, contraste e textura a seu favor.

Use a tipografia para criar hierarquia e clareza.

Considere cuidadosamente a forma de utilização dos tipos de letra. Diferentes tamanhos, fontes e disposição do texto ajudam a aumentar a legibilidade.

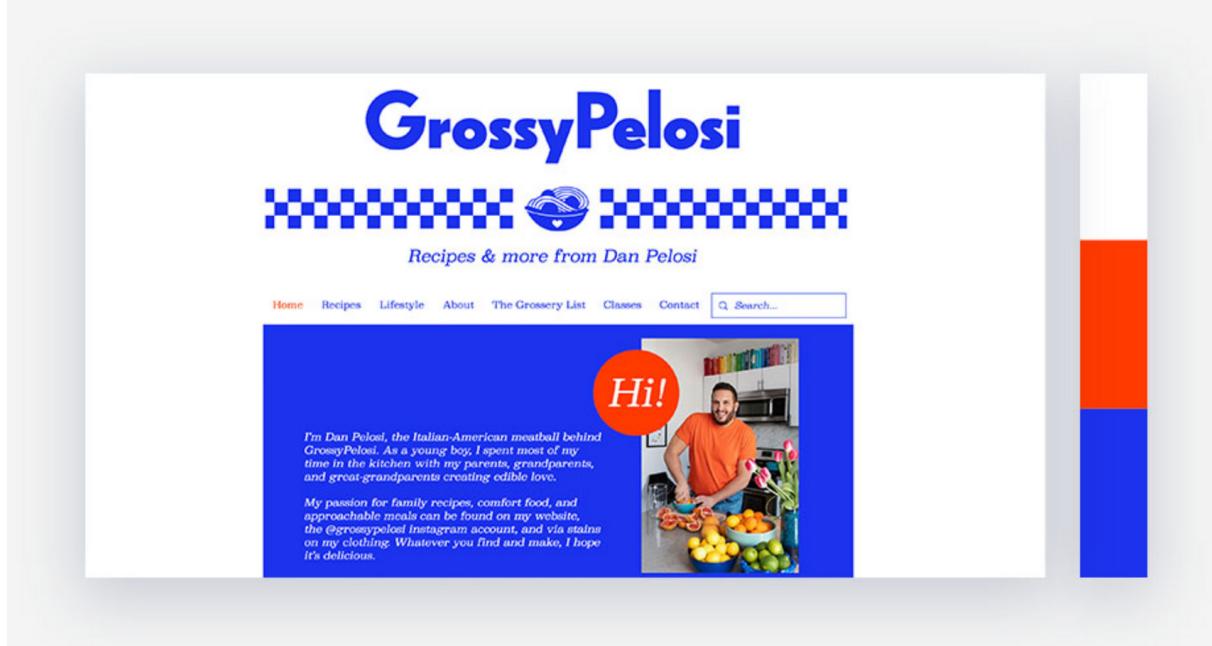


Figura 7 – Exemplo de *layout* que combina cor e tipografia

Fonte: Grossy Pelosi (2022)

Certifique-se de que o sistema comunicará o que está acontecendo.

Sempre informe os usuários sobre localização, ações, mudanças de estado ou erros.

The screenshot shows a user registration form with the following fields and errors:

- Nome Completo:** Ana Silva (correct)
- E-mail:** anasilva.@e0mail.com (incorrect, highlighted with a red border)
- Mensagem de erro para o e-mail:** Este e-mail não é válido
- Telefone:** (empty field)

At the bottom are two buttons: **Cancelar** (Cancel) and **Cadastrar** (Register).

Figura 8 – Formulário de cadastro demonstrando uma validação de dados

Fonte: Senac EAD (2022)

Pense nos padrões.

Pensando cuidadosamente nos objetivos que as pessoas trazem para o *software* e os antecipando, você pode criar padrões que reduzem a carga do usuário. Isso se torna particularmente importante quando se trata de *design* de formulários, pois neles você pode ter a oportunidade de ter alguns campos pré-escolhidos ou preenchidos.

Wireframes e protótipos

As pessoas têm o costume, muitas vezes, de se referir a *wireframes* e protótipos como se ambos fossem termos sinônimos. Então, nada melhor do que conceituar corretamente cada um deles para uma melhor compreensão.

Wireframes

Wireframes são o “esqueleto” de uma interface. Eles representam o primeiro passo do processo de *design* logo após a concepção mental. São rascunhos em preto e branco ou desenhos que representam as telas que os usuários acessarão no sistema ou no *site* e fornecem apenas um contorno dos elementos que estarão presentes. Devem, ainda, ser desprovidos de qualquer *design* visual e são usados por UX *designers* para definir a hierarquia dos itens e a comunicação que estes devem fornecer na interface gráfica baseada nas necessidades do usuário.

Wireframes nada mais são do que um método que ajuda os UX *designers* a projetarem interfaces digitais, como *sites*, aplicativos móveis, *softwares* etc. São uma ferramenta útil para projetar uma interface em um nível estrutural e constituem um dos primeiros passos no processo de *design*. Assim, é muito mais fácil para os *designers* trazerem mudanças com base no *feedback* coletado ao longo do processo.

Portanto, os *wireframes* são o melhor aliado para detectar problemas antecipadamente, mapear funcionalidades e evitar uma rodada interminável de revisões no estágio final, quando cada mudança significa passar por um trabalho complicado e “doloroso”.

O *wireframe* geralmente “entra em cena” após o estágio de pesquisa do usuário, quando são reunidas informações suficientes sobre as necessidades e as motivações do usuário-alvo. No entanto, também pode ser útil durante o processo de descoberta, caso o intuito seja explorar variações ou novas possibilidades no *design*.

Tipos de *wireframe*

Cada *wireframe* requer um nível de habilidade diferente, do mais complexo ao mais básico. A divisão pode ser feita da seguinte forma:

Wireframe de baixa fidelidade

Conhecidos como os *mockups* de *wireframe* mais simples, os *mockups* de baixa fidelidade geralmente são criados manualmente. Mais fácil de criar que os demais, o *wireframe* de baixa fidelidade não requer o uso de cores ou detalhes.



Figura 9 – Exemplo de *wireframe* de baixa fidelidade

Fonte: Senac EAD (2022)

Wireframe anotado

Os *mockups* anotados, considerados um dos tipos de *wireframes* de média complexidade, são adequados para elementos visuais mais organizados e *mockups* de páginas e devem incluir a inserção de descrição e título.

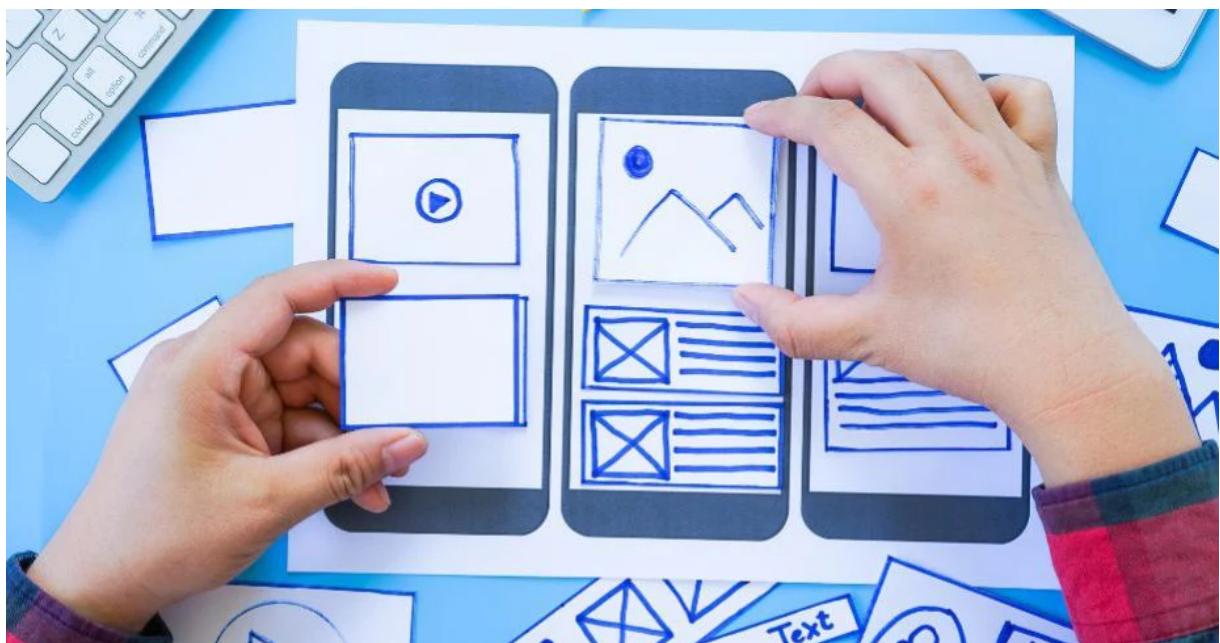


Figura 10 – Exemplo de *wireframe* anotado

Fonte: Senac EAD (2022)

Wireframe de alta fidelidade

É melhor, ao se aproximar da versão final, usar um *software* que forneça interações como cliques de botão para criar modelos de alta fidelidade.

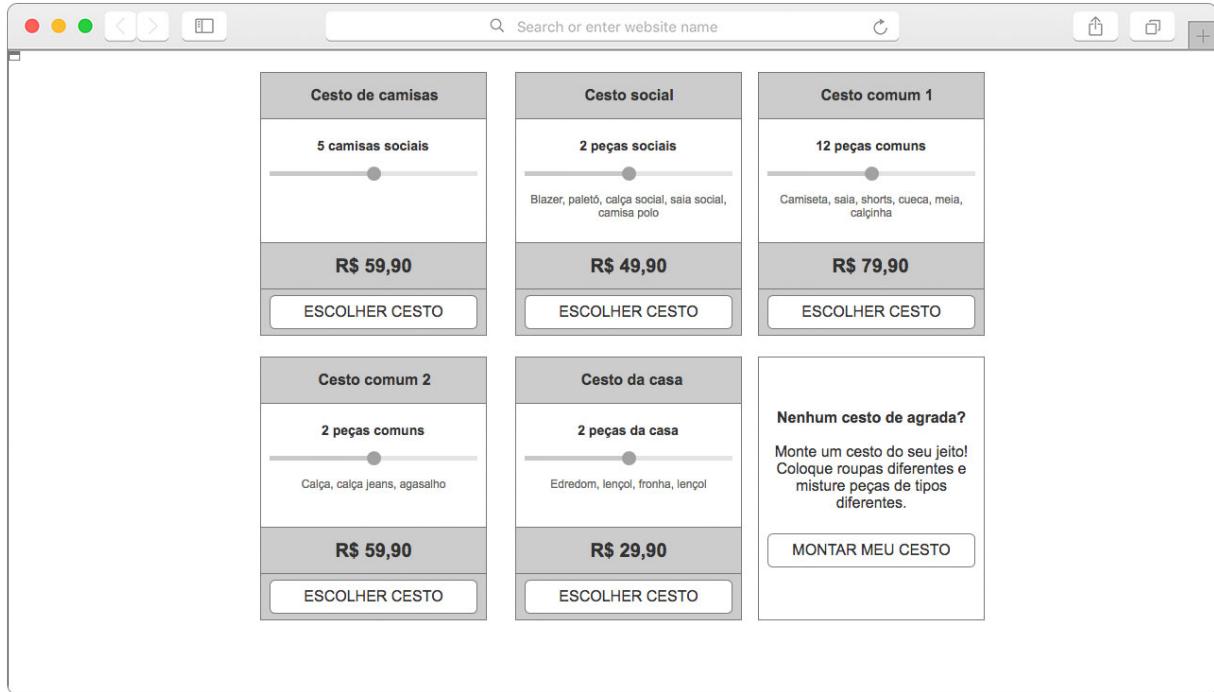


Figura 11 – Exemplo de *wireframe* de alta fidelidade

Fonte: Senac EAD (2022)

Seja qual for o nível que você escolher, nunca esqueça que o *wireframe* não deve ser tão desenvolvido quanto a versão final do *layout*. O que realmente importa é apresentar claramente a estrutura externa. Trata-se de uma abordagem que dá agilidade e confiança aos processos de *design* e desenvolvimento.

Wireframe na prática

Para criar *wireframes*, o **Figma** será a ferramenta utilizada. Então, baixe-o ou utilize-o diretamente no *site* oficial.

Criando uma tela de *login*

Você desenvolverá, passo a passo, um *wireframe* de uma tela de *login*, a qual, ao final, ficará semelhante a esta:

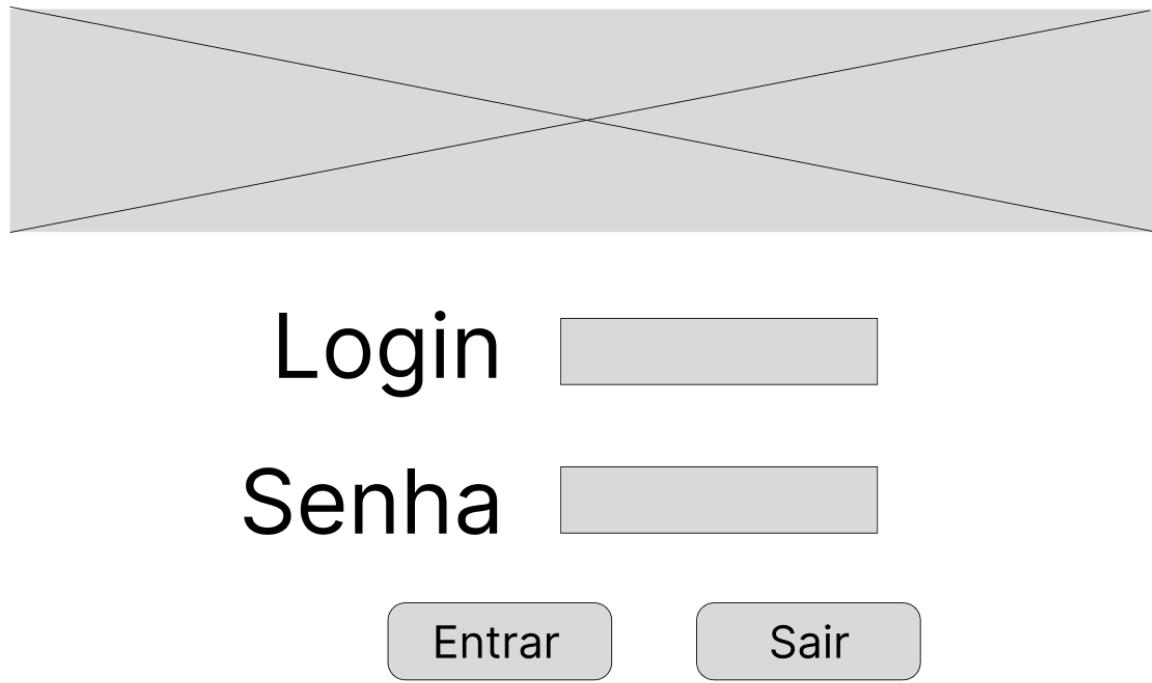


Figura 12 – *Wireframe* de tela de *login*

Fonte: Senac EAD (2022)

Com o Figma devidamente aberto (programa ou modo web), crie um novo *design* clicando em **New design file**, conforme esta imagem:

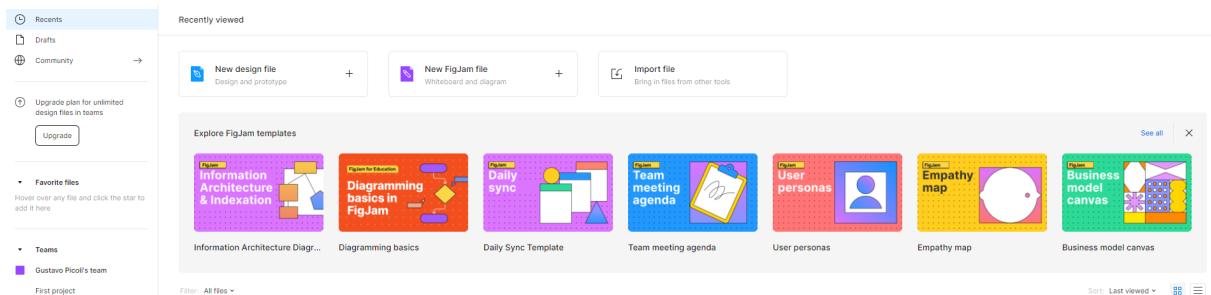
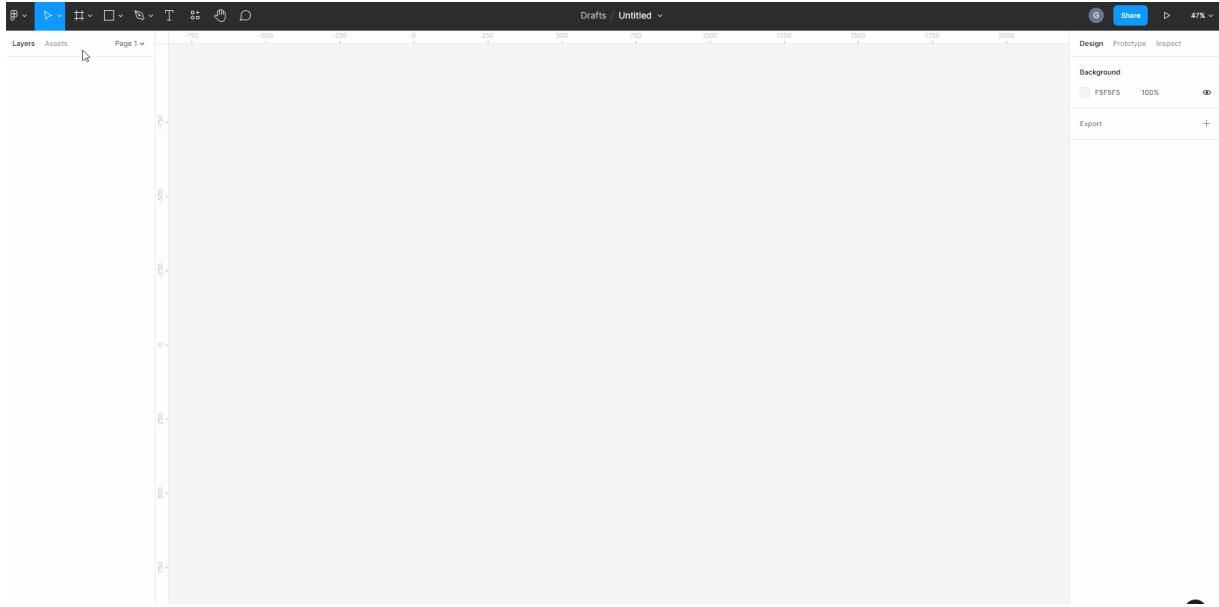


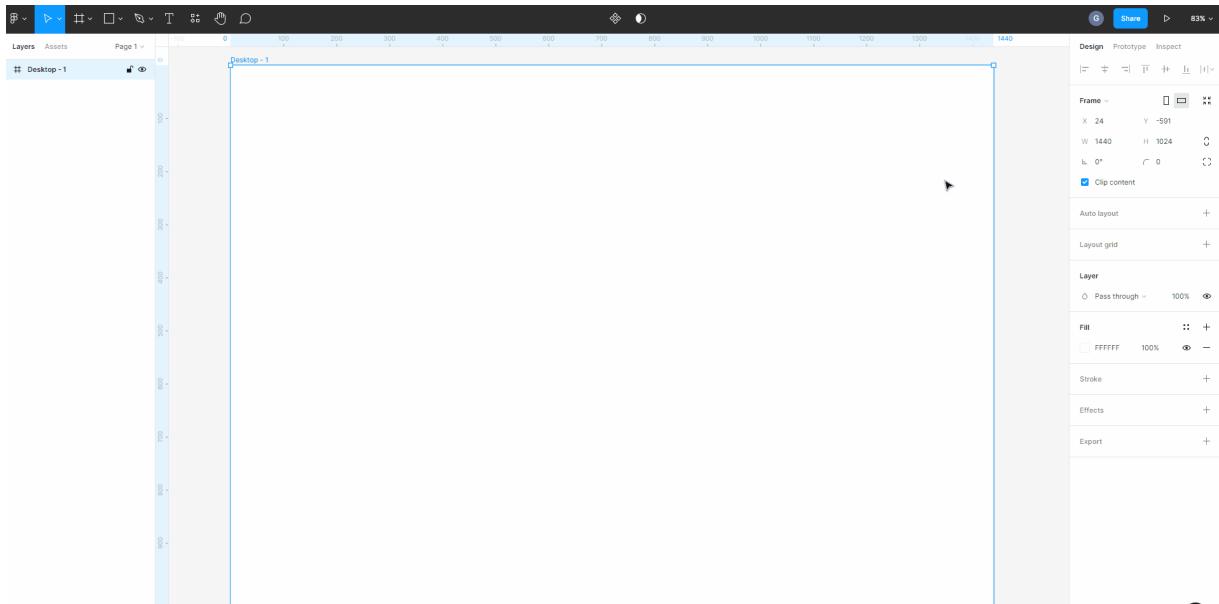
Figura 13 – Tela inicial do Figma

Fonte: Figma (2022)

Após clicar em **New design file**, uma tela em branco abrirá. Agora crie o seu *wireframe*, conforme instruções no *GIF* (*graphics interchange format*) a seguir:

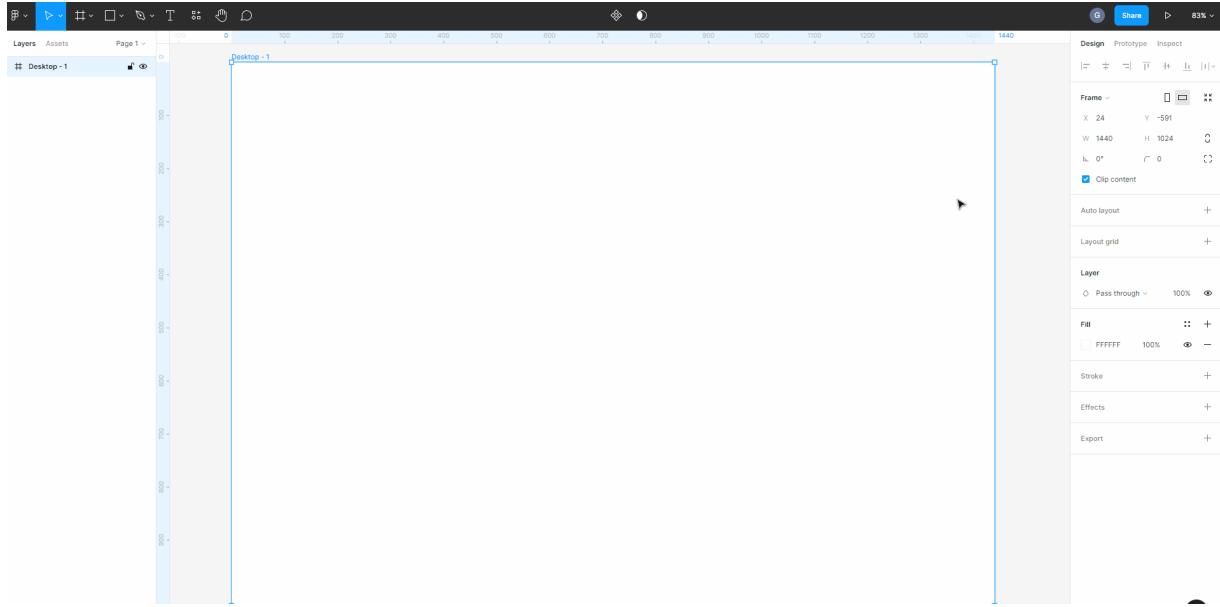


Agora você já tem sua área de trabalho, também chamada de *Canvas*, dentro do Figma. Caso queira alterar o tamanho do *wireframe*, faça o seguinte:

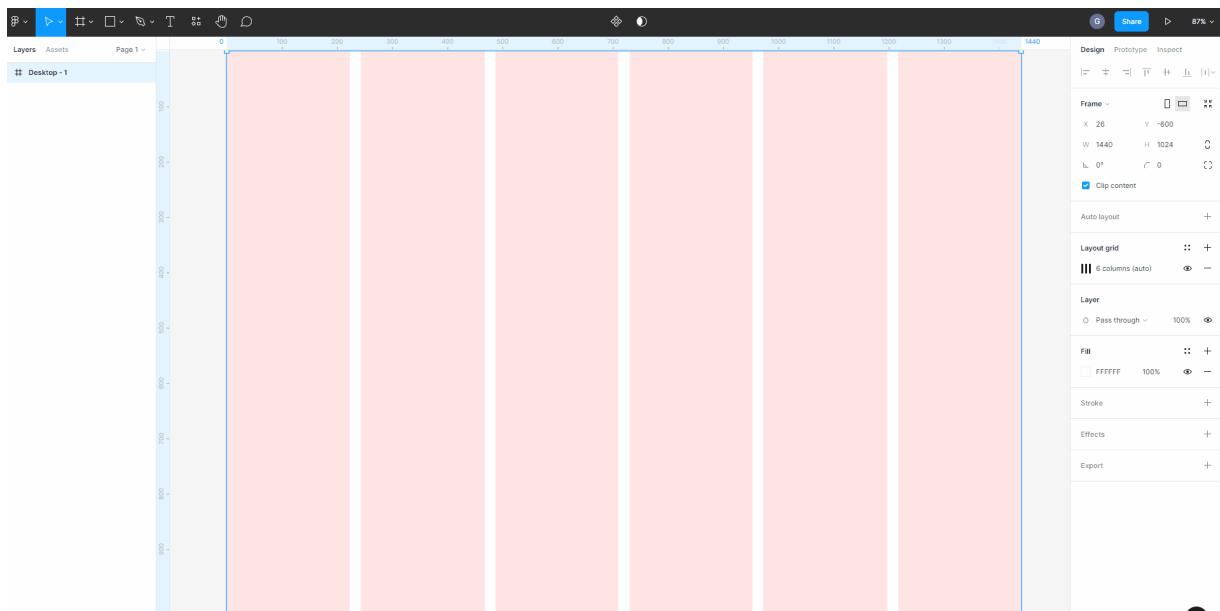


Agora você deve ajustar seu *wireframe*, para que ele mostre uma *grid*. A *grid* tem como principal objetivo auxiliar na ordenação, na distribuição, no alinhamento e na dimensão de imagens, textos, formas e outros elementos. Fique à vontade para personalizar sua *grid*. Cada pessoa prefere um jeito. Faça testes até que esteja do seu gosto.

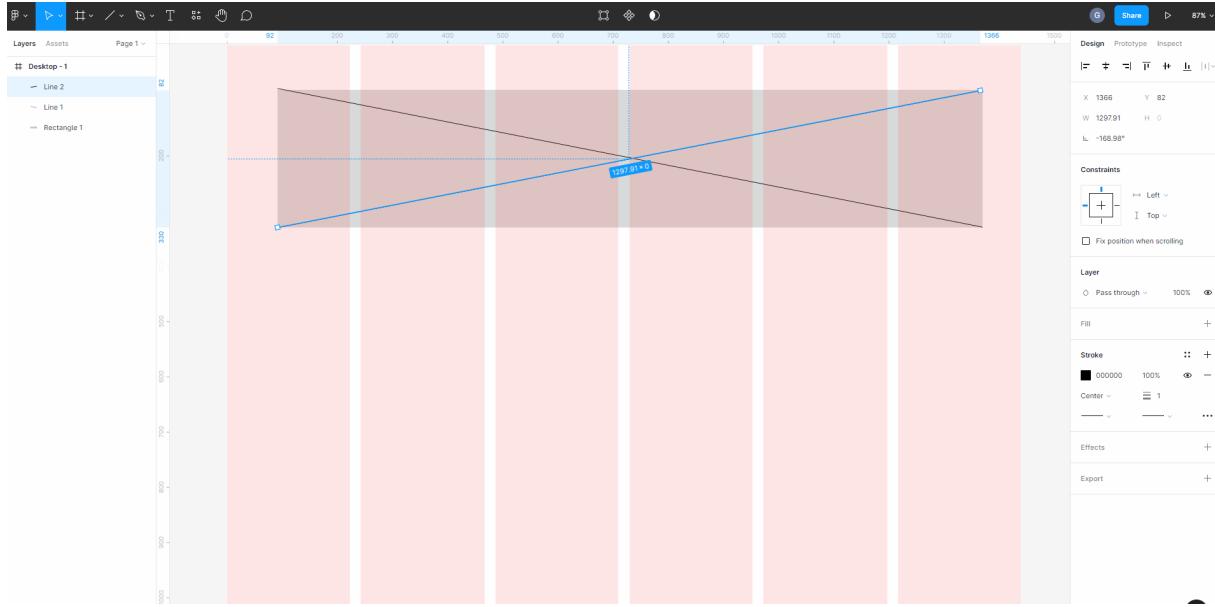
Para este exemplo, a *grid* deve ter seis colunas:



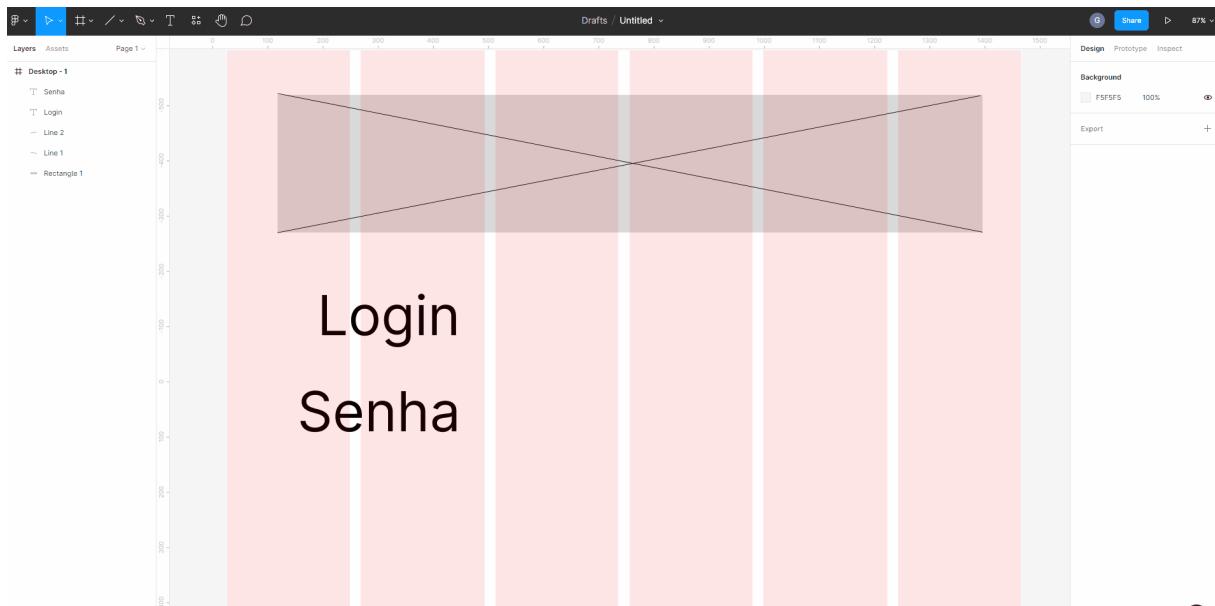
Adicione um espaço para que, futuramente, seja inserida uma imagem:



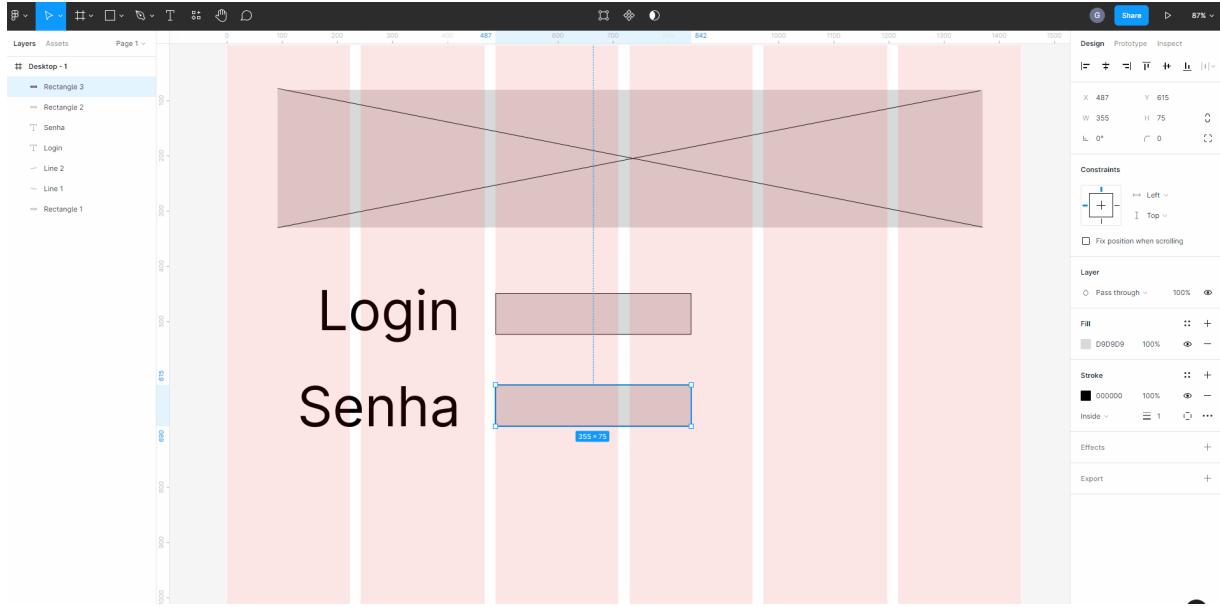
Adicione dois rótulos de texto (*login* e *senha*):



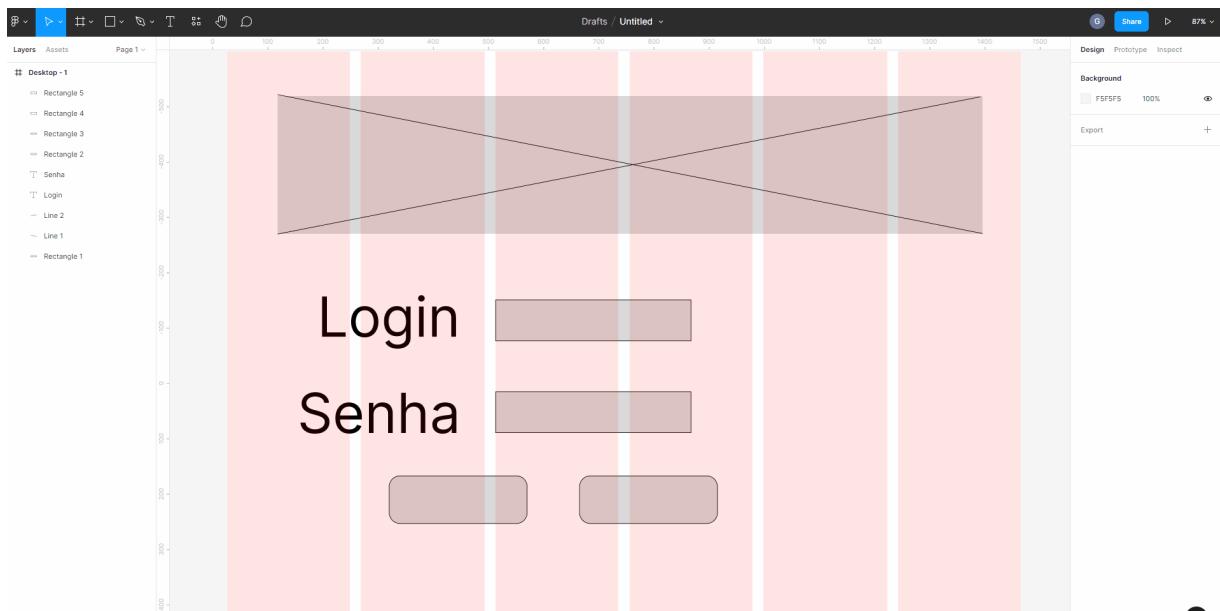
Para dar a impressão de que ali serão programados campos de textos, adicione dois retângulos, como se fossem um campo de texto e um campo de senha:



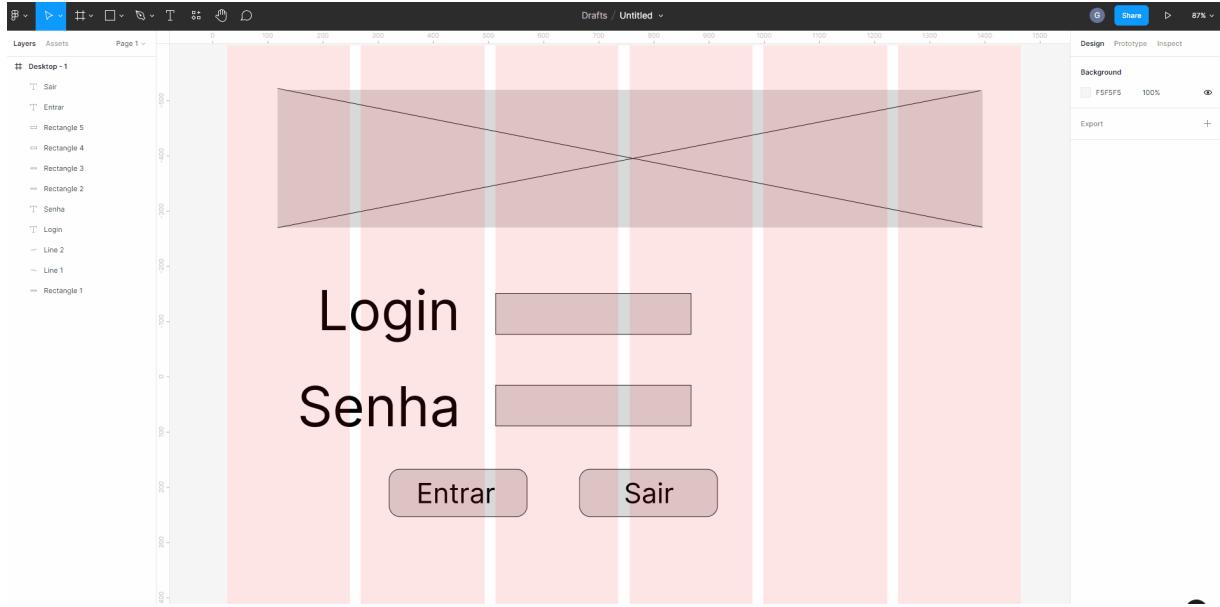
O seu *wireframe* terá também dois botões: **Entrar** e **Sair**.



Os botões precisam de um rótulo, então use a ferramenta de texto para tanto:



Aparentemente, o seu *wireframe* está pronto! Fique à vontade para fazer os ajustes que desejar.



Agora você praticará outro exemplo: um *wireframe* de uma tela de cadastro simples.

Com o Figma devidamente aberto (programa ou modo web), crie um novo *design* clicando em **New design file**, conforme esta imagem:

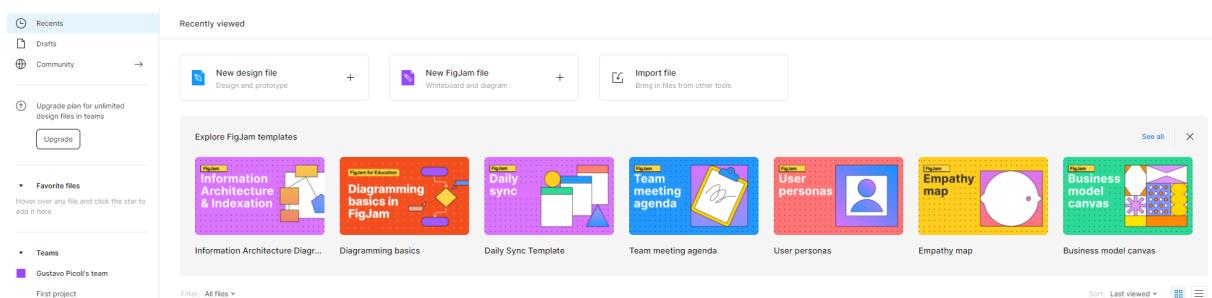
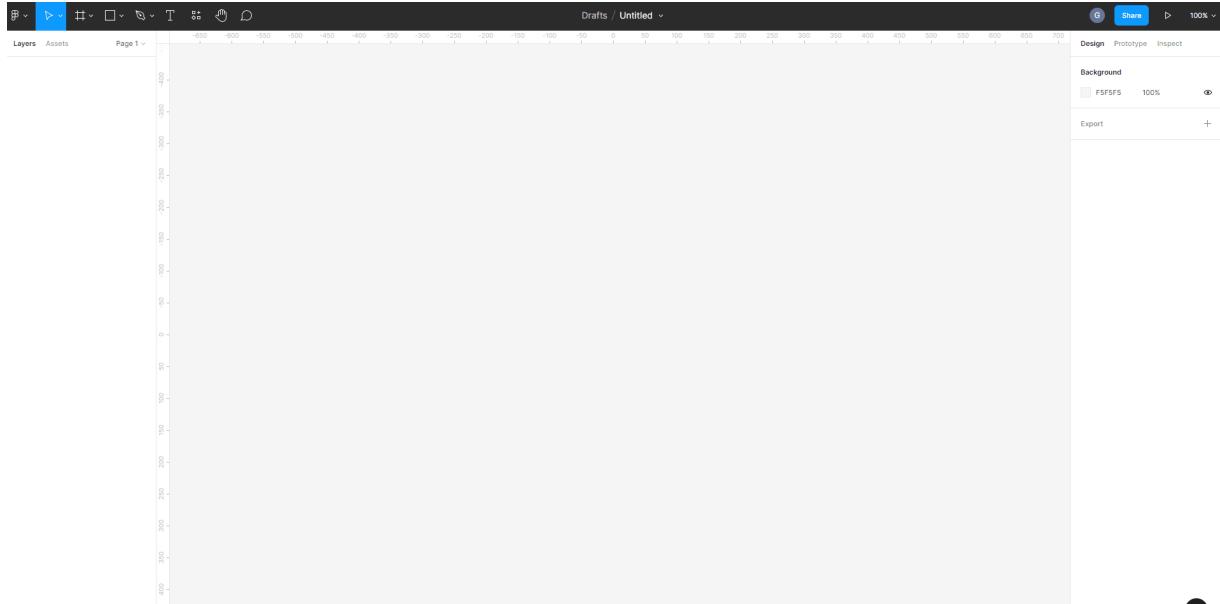


Figura 14 – Tela inicial do Figma

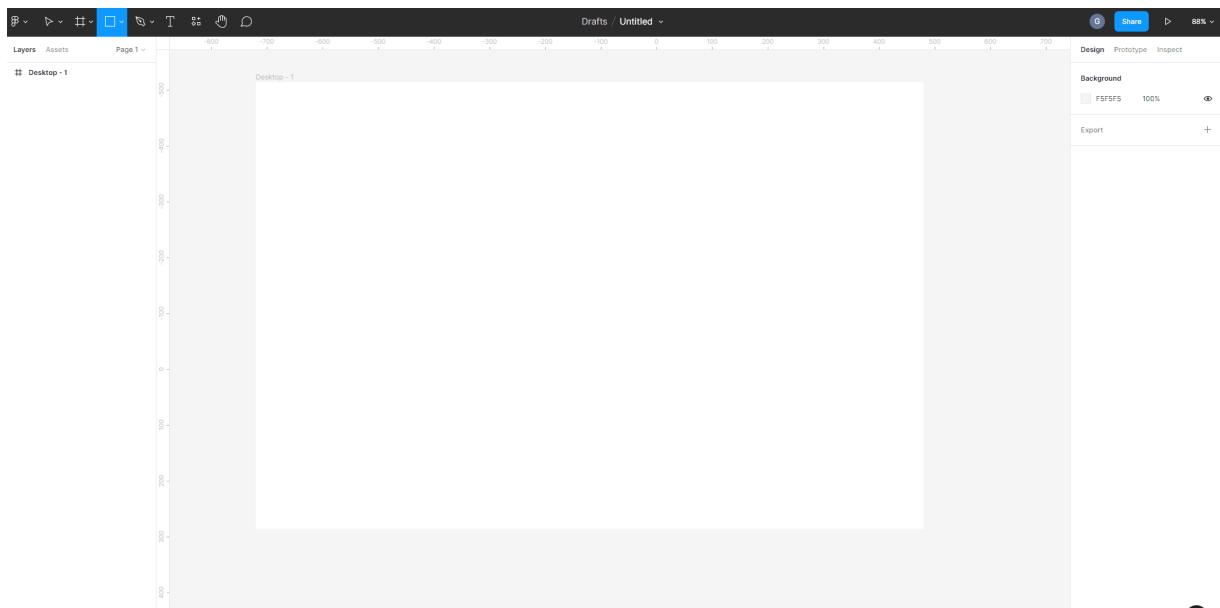
Fonte: Figma (2022)

Após clicar em **New design file**, uma tela em branco abrirá. Agora crie o seu *wireframe*, conforme instruções a seguir:

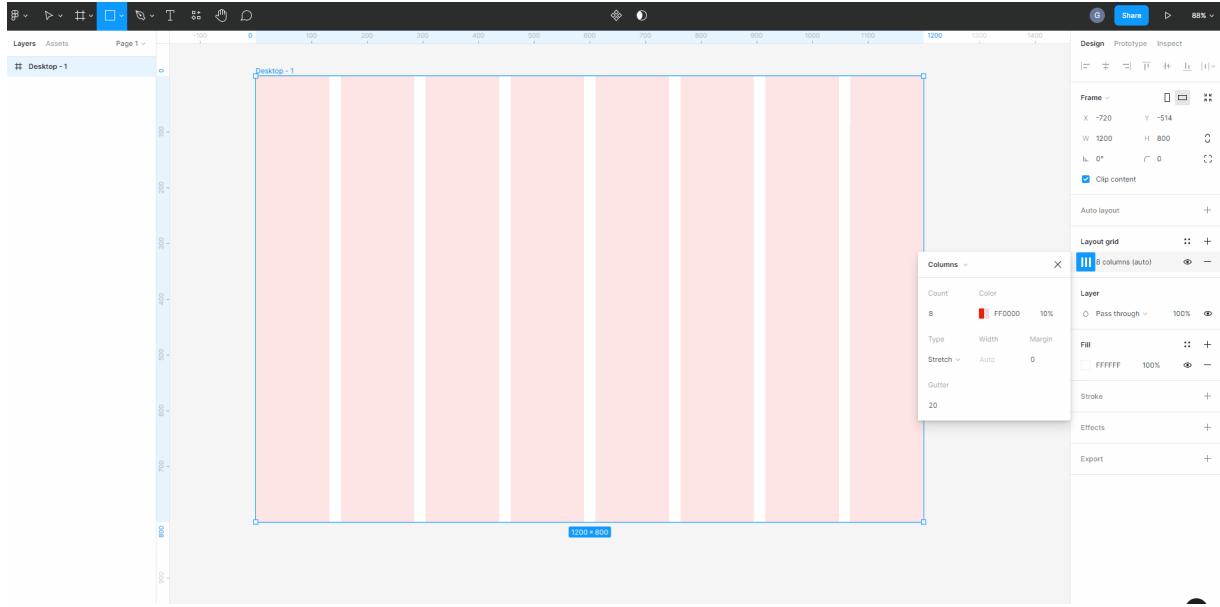


Agora você deve ajustar seu *wireframe*, para que ele mostre uma *grid*. Fique à vontade para personalizar sua *grid*. Cada pessoa prefere um jeito. Faça testes até que esteja do seu gosto.

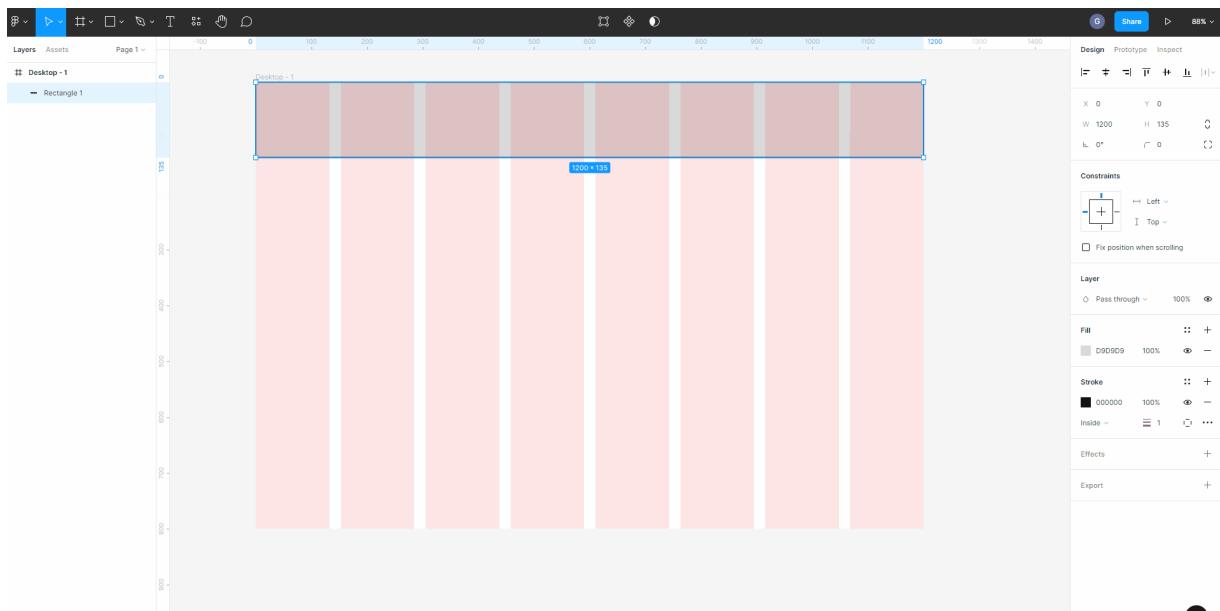
Para este exemplo, a *grid* deve ter oito colunas:



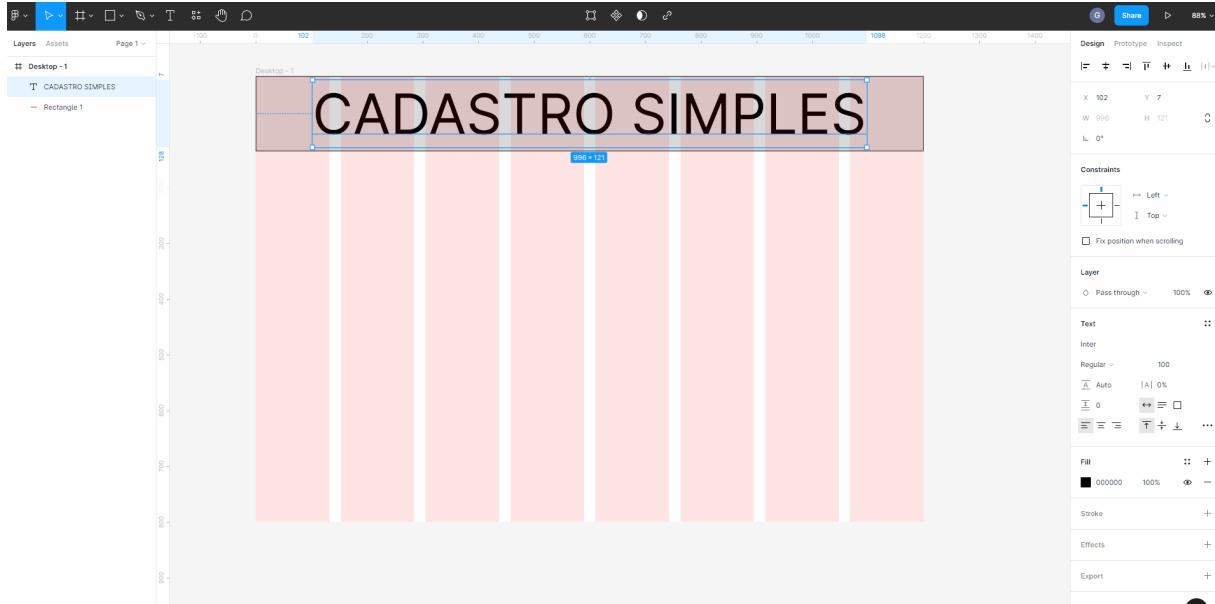
Adicione um retângulo na parte superior da tela, que é o local onde você escreverá. Essa parte pode ser usada também para adicionar uma imagem, ficando a escolha a critério da equipe de desenvolvimento.



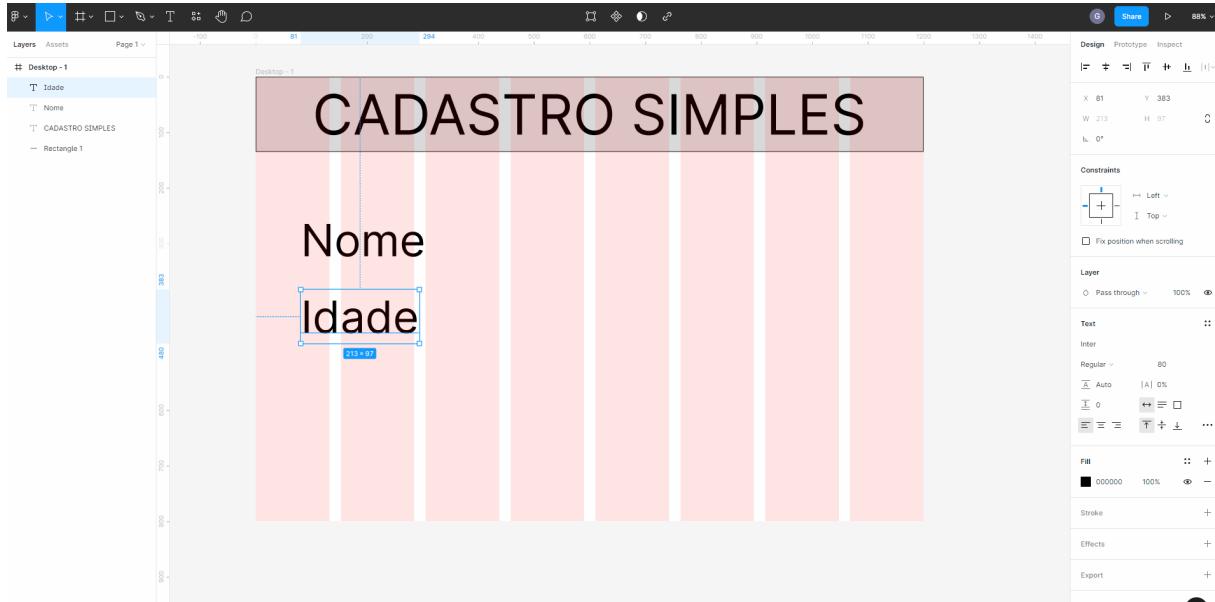
Adicione um rótulo de texto com a escrita “Cadastro simples”:



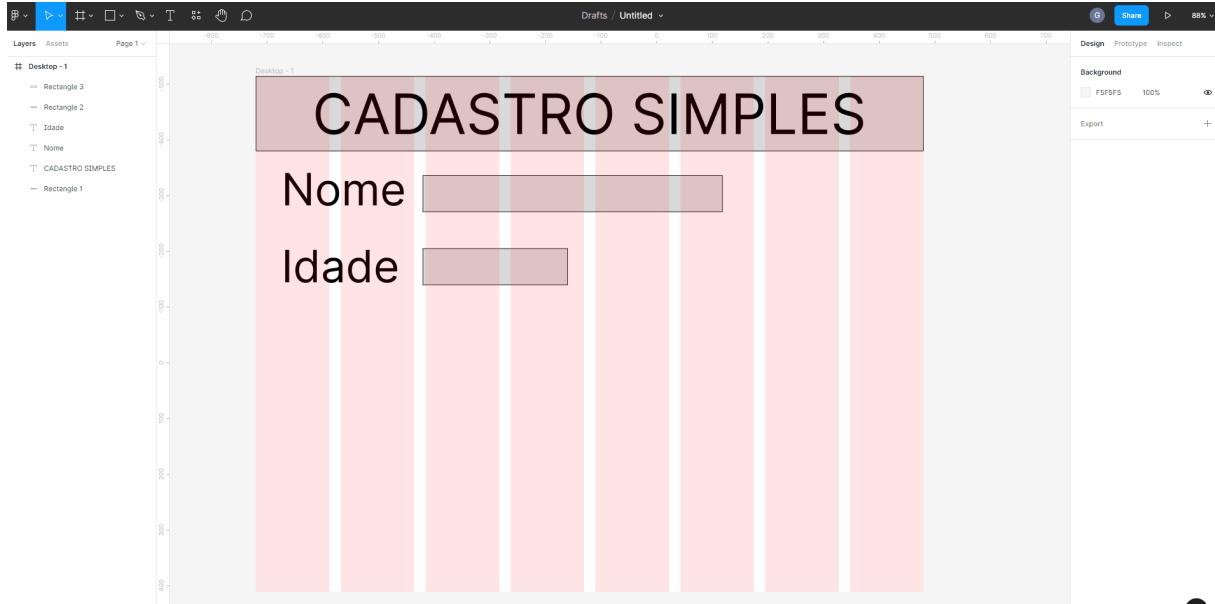
Adicione mais dois rótulos de texto, porém agora com as palavras “Nome” e “Idade”. Fique à vontade para alinhar como quiser:



Para os dois rótulos de texto, adicione, ao lado, dois retângulos. Ainda, para que ambos os retângulos tenham a mesma largura, use a opção de “copiar e colar” e somente ajuste o comprimento do retângulo para o campo “Idade”:

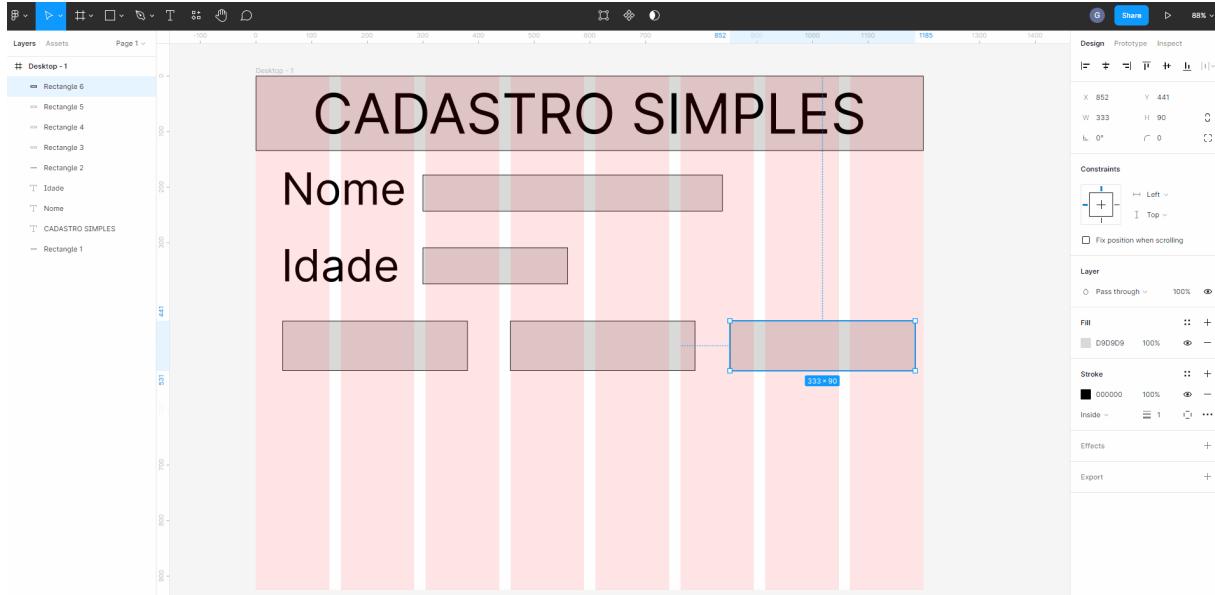


Agora adicione três retângulos, que serão os botões do seu formulário. Uma dica é usar a opção de “copiar e colar”, para que eles fiquem com o tamanho igual. Você também pode adicionar borda, na opção **Stroke** à direita:

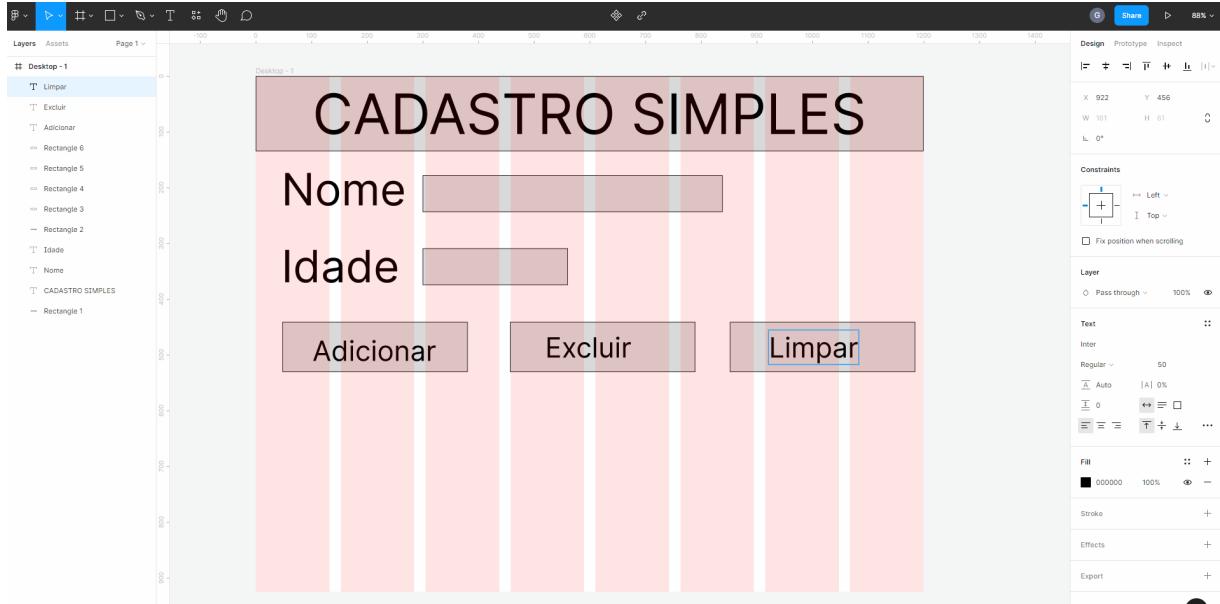


Os botões devem conter um rótulo também. Para tanto, utilize a ferramenta de texto e insira os rótulos por cima dos botões.

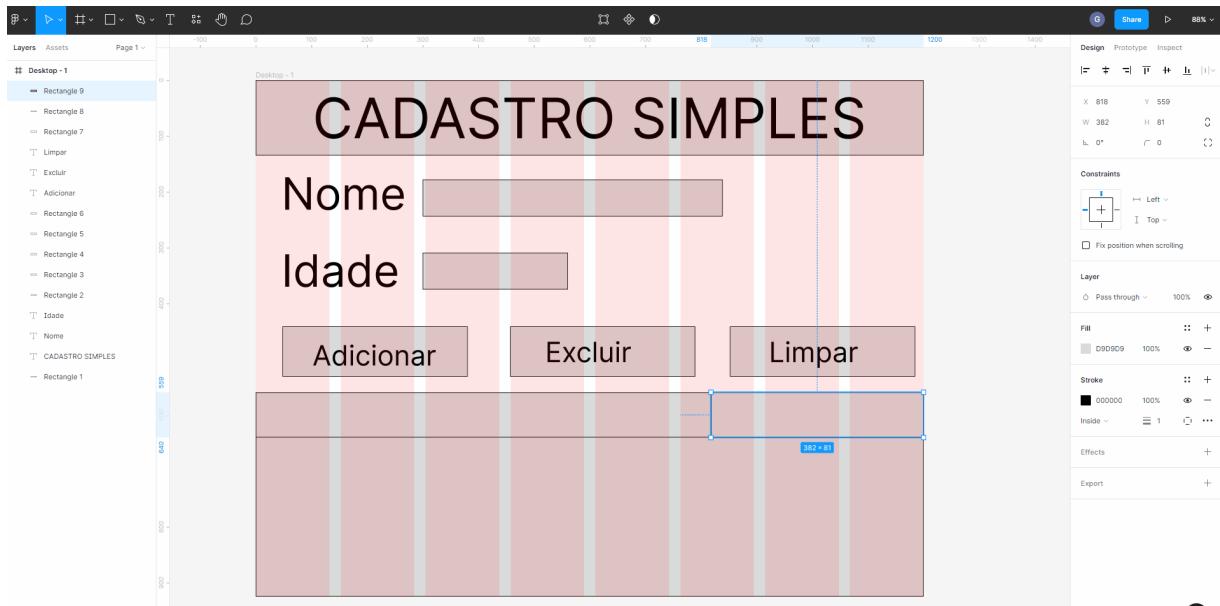
Dica: caso o texto, ao ser inserido em cima do retângulo que representa os botões, fique atrás do retângulo, clique em cima do texto com o botão direito do mouse e escolha a opção **Bring to front**, que significa “trazer para a frente”:



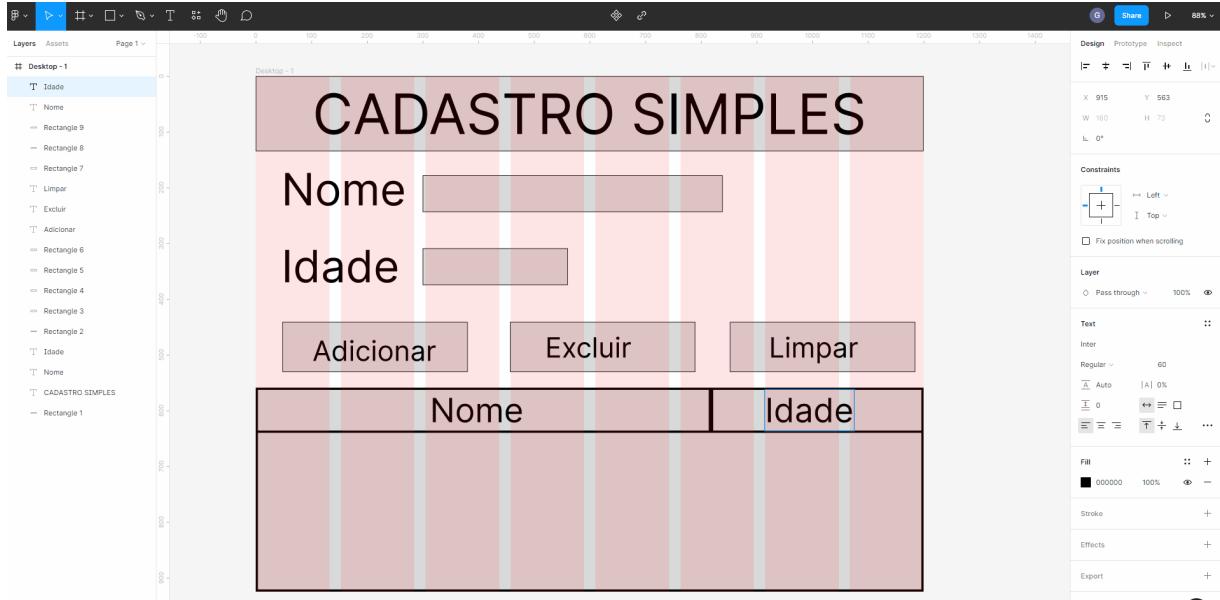
Crie agora uma “lista” na qual os dados serão mostrados após serem adicionados. Também coloque o rótulo na lista:



Coloque campos de texto na lista:



Caso você queira ocultar a *grid*, faça o seguinte:



Por fim, o seu formulário ficará semelhante a este:

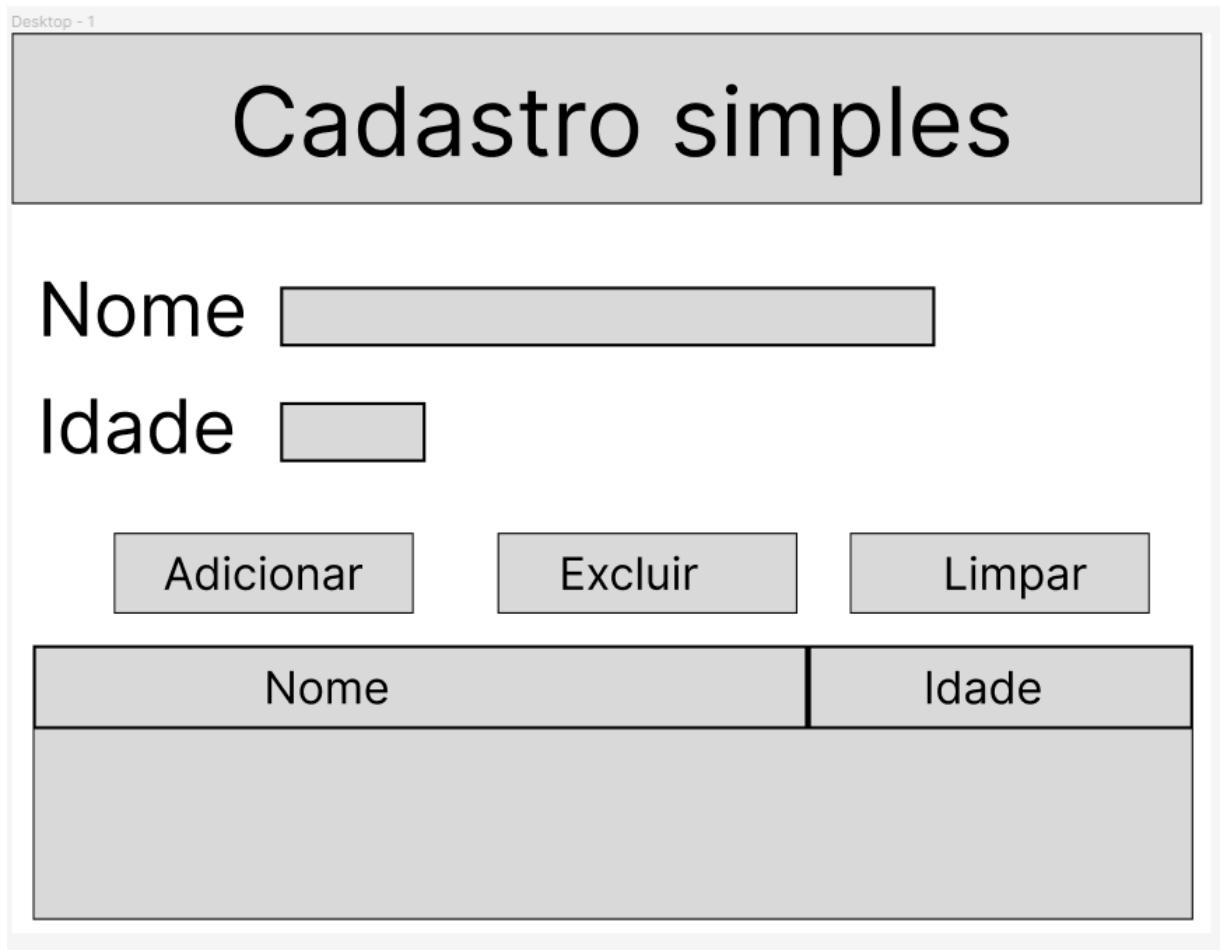


Figura 15 – Wireframe finalizado

Os exemplos desenvolvidos mantiveram a interface o mais simples possível para uma melhor compreensão do usuário. Evitaram-se elementos desnecessários e sem sentido. O *layout* foi objetivo, sempre cuidando a questão relacionada a alinhamento e espaçamento entre os elementos, para que o usuário não tenha uma impressão de “bagunça”. Há uma infinidade de opções de tipografia, e, para a escolha, deve-se pensar em tipografias que sejam legíveis e que agradem o usuário.

Com os conteúdos aprendidos, desenvolva o *wireframe* a seguir, fazendo os devidos ajustes de elementos para uma melhor UX.

Desktop - 1

Calculo do IMC (índice de massa corporal)

PESO

ALTURA

Calcular **Limpar**

RESULTADO

MENOR QUE 18,5 = MAGREZA
ENTRE 18,5 E 24,9 = NORMAL
ENTRE 25,0 E 29,9 = SOBREPESO
ENTRE 30,0 E 39,9 = OBESIDADE
MAIOR QUE 40,0 = OBESIDADE GRAVE

Figura 16 – Modelo de *wireframe* a ser criado

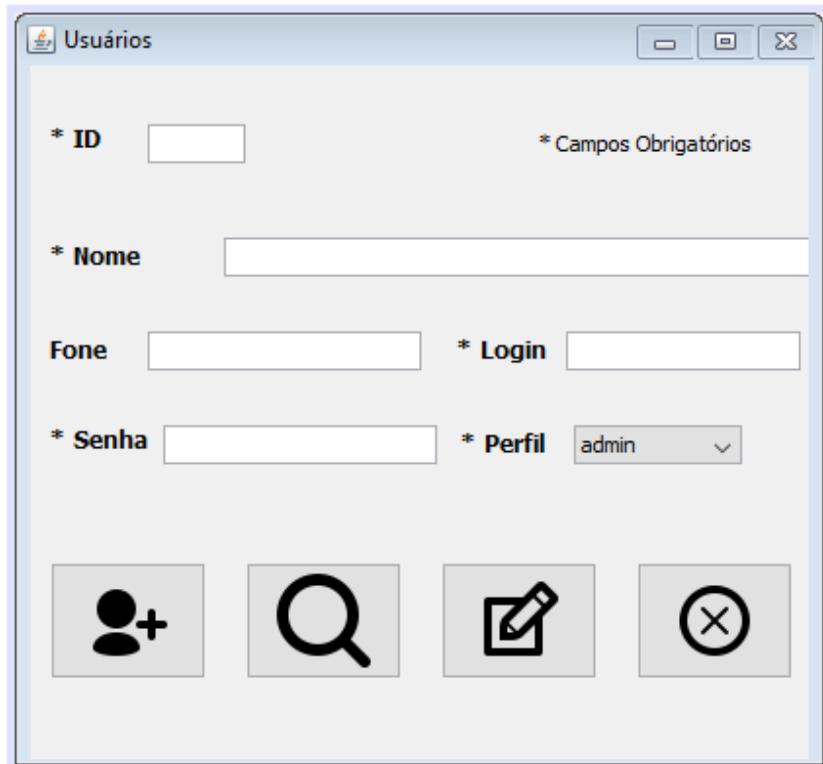
Fonte: Senac EAD (2022)

Protótipos

Um protótipo é uma versão aproximada de um produto que é criado durante o processo de *design*. O objetivo final é dar uma forma unificada e pré-definitiva às descobertas coletadas até este ponto.

A construção de um protótipo permite testar o **fluxo do produto** e a **validade de análises e ideias de design**. Os protótipos são usados então para coletar *feedback* sobre o produto ou o serviço de partes internas, mostrando às partes interessadas como o produto final seria e funcionaria, com foco no público-alvo. Assim, geralmente eles são o ponto final do estágio de projeto e o ponto de partida do estágio de teste.

Os protótipos são fluidos por definição, pois devem ser testados e revisados ao longo do processo com base no *feedback* do usuário. Basicamente, significa lançar o “produto” para o mundo pela primeira vez e ver se tudo sai como planejado.



O formulário "Usuários" contém campos para ID, Nome, Fone, Login, Senha e Perfil. Um link para "Campos Obrigatórios" aparece ao lado do campo ID. Abaixo dos campos, há quatro ícones: adicionar usuário (+), busca (lupa), editar (caneta) e cancelar (X).

Usuários	
* ID	<input type="text"/>
* Campos Obrigatórios	
* Nome	<input type="text"/>
Fone	<input type="text"/>
* Login	<input type="text"/>
* Senha	<input type="password"/>
* Perfil	admin

Figura 17 – Protótipo de uma tela de cadastro de usuário

Fonte: Senac EAD (2022)

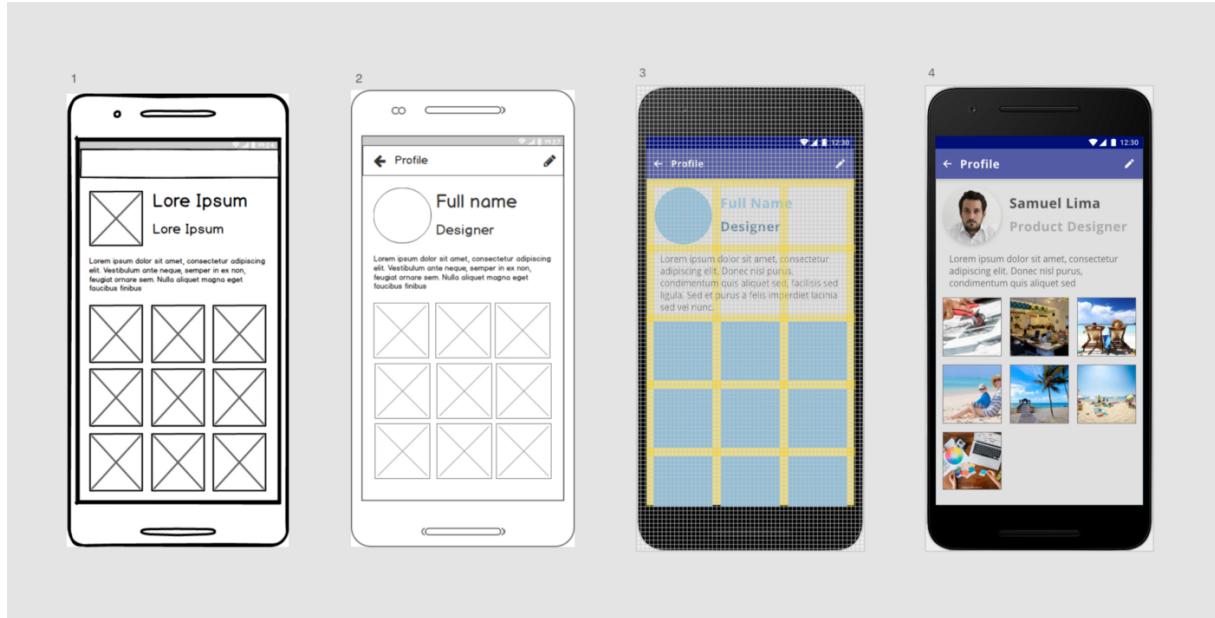


Figura 18 – Protótipo de um aplicativo *mobile*

Fonte: Senac EAD (2022)

Encerramento

Neste conteúdo, você pôde entender um pouco melhor o processo de criação de interface gráfica, sempre respeitando padrões e normas, e também pôde se colocar no lugar do seu cliente, pensando no que foi apresentado para ele, ou seja, na interface gráfica mais agradável. Viu ainda a importância dos *wireframes* nesse processo, pois eles evitam (e muito) o retrabalho. Além disso, o cliente terá uma visão melhor do *software* solicitado e poderá, assim, avaliar melhor o sistema.