

# INFORMÁTICA PARA INTERNET

## Conceito e estrutura

*Wireframes* são criados logo no começo do processo de desenvolvimento de uma aplicação *web* ou de um aplicativo. Eles servem para definir a estrutura básica dos layouts das páginas com que o usuário interage antes de se acrescentar o conteúdo em si.

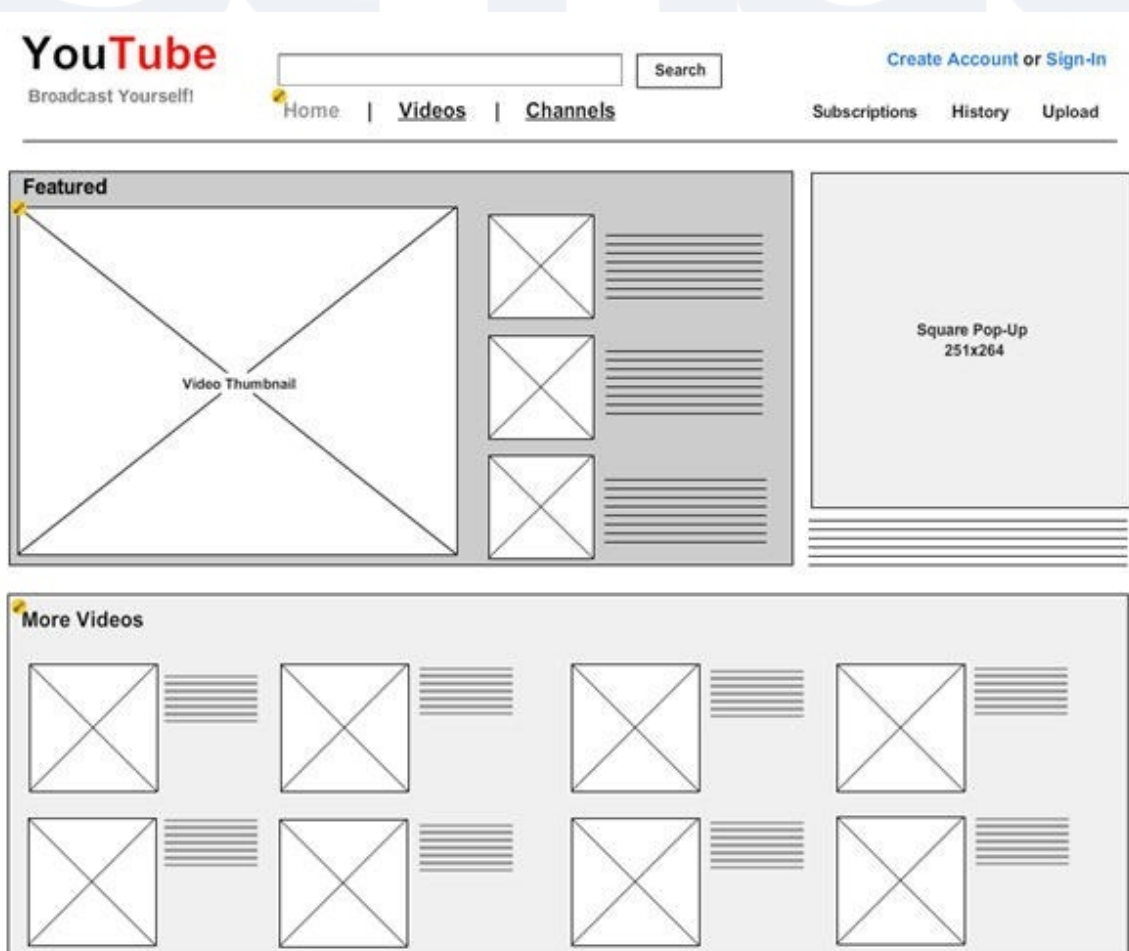


Figura 1 — Wireframe do site YouTube

Fonte: <<https://mockflow.com/>>. Acesso em: 20 set. 2019.

O *wireframe* é um desenho básico que funciona como uma planta baixa do projeto *web* ou do aplicativo que será desenvolvido. Ele é elaborado para organizar os elementos que entrarão na composição do projeto final, transmitindo de forma objetiva como a estrutura do objeto (aplicativo, interface, página *web* etc.), considerando as suas características predefinidas, deverá ser.

Muitos *designers* têm o hábito de criar *wireframes* para os seus projetos. Eles podem ser feitos com ferramentas *on-line* ou rascunhados rapidamente em um caderno de ideias.

O *wireframe* é um processo importante no desenvolvimento *web*, principalmente em projetos muito complexos. Ele deve ser feito da maneira mais evidente e clara possível, mostrando apenas o essencial. O resultado final funciona como um rascunho (sem cores nem imagens), que contém uma representação visual em escala de cinza da estrutura e da funcionalidade de uma única página *web* ou de uma tela de aplicativo móvel.

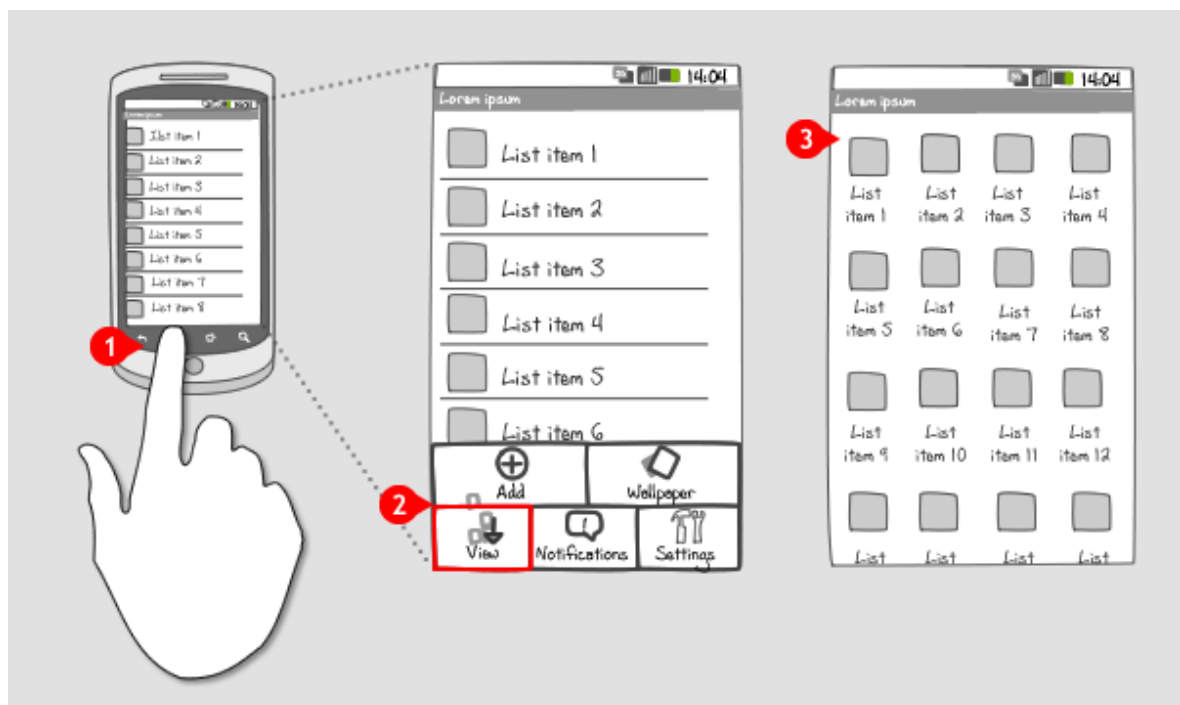


Figura 2 – *Wireframe* de um aplicativo com listas

Fonte: <<https://wireframes.tumblr.com/>>. Acesso em: 20 set. 2019.

A função do recurso é ajudar os desenvolvedores a compreender os requisitos definidos em conjunto com o cliente, considerando encargos e objetos que determinado sistema – que está sendo desenvolvido – contém.

A criação de *wireframes* é especialmente vantajosa quando se está comunicando com os clientes, independentemente se forem internos (o cliente é a própria empresa) ou não. Eles facilitam a visualização de ideias, mais que uma descrição abstrata do que se está planejando para o projeto.

Quer se trate de um aplicativo complexo com diversas funcionalidades, quer se trate de um *site* simples, o *wireframe* será necessário para instruir exatamente o que precisa ser realizado nas diferentes e diversificadas páginas do projeto em desenvolvimento.

Além da necessidade de um *wireframe* na parte de planejamento e comunicação com o cliente, é importante ter em mente que o seu processo de produção é uma vantagem por si só. Por ser muito mais barato, rápido e simples de ser criado, ele pode ser facilmente revisto, adaptado ou descartado.

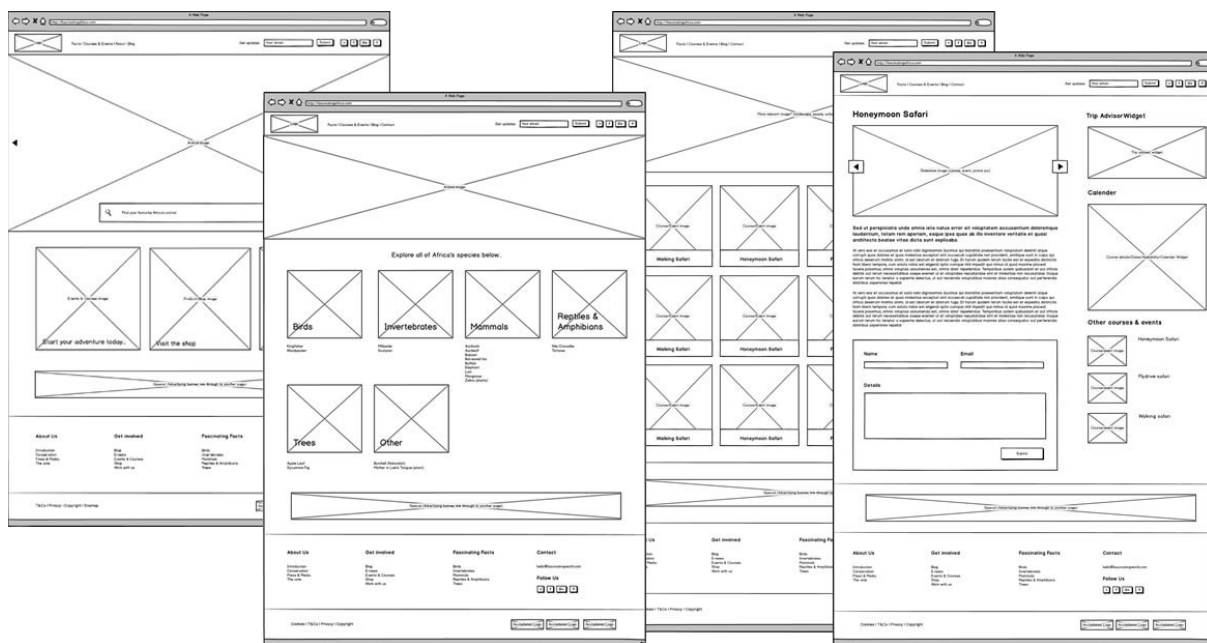


Figura 3 – Wireframes de um site de viagens

Fonte: <[https://medium.com/@robertsmith\\_co/validating-your-product-design-ideas-with-low-fidelity-wireframes-fba03b84af23](https://medium.com/@robertsmith_co/validating-your-product-design-ideas-with-low-fidelity-wireframes-fba03b84af23)>. Acesso em: 18 set. 2019.

Como os *wireframes* acabam gerando organização e uma melhor comunicação entre os profissionais que estão desenvolvendo a aplicação, seu uso acaba resultando em uma economia de tempo e de dinheiro muito grande. Tendo uma definição real de onde cada elemento será posicionado, a equipe de desenvolvimento compreende melhor o trabalho que está sendo executado. A criação de conteúdo é mais objetiva e abre menos margem para erros. Depois da definição e da validação do *wireframe*, ele dificilmente será totalmente refeito ou significativamente modificado.

Tendo isso em mente, podem-se dividir em quatro as metas da ferramenta.

#### 1. Usabilidade

- ◆ Criar um *wireframe* proporciona uma melhor usabilidade, pois é possível planejar antecipadamente cada detalhe do caminho que o usuário percorrerá pelo aplicativo/site.

Em um projeto, uma das estruturas mais importantes é a navegação. Durante a criação do *wireframe*, podem ser levantadas questões como: está fácil para o usuário acessar as informações de que precisa? Ele entende intuitivamente o que deve fazer dentro da aplicação? Como os usuários acessarão as informações? É possível simplificar?

O *wireframing* oferece um olhar objetivo para nomes de *links*, facilidade de uso, navegação e alocação de recursos.

#### 2. Expansividade

- ◆ É corriqueira a necessidade de expansão de um *site*/aplicativo depois da sua finalização. Por isso, a aplicação deve ter a capacidade de acomodar esse crescimento, minimizando impactos na usabilidade, na arquitetura e no *design*.

#### 3. Cumprimento das metas estabelecidas

Os *wireframes* permitem uma visão geral do projeto, mostrando onde os recursos serão posicionados, sua estrutura e divisão. Isso ajuda a definir as expectativas sobre o modo como os recursos deverão ser implementados e torna mais fácil perceber se um recurso deve ou não ser retirado.

#### 4. Feedback

- ◆ Em vez de juntarem todas as funcionalidades, o layout e os elementos criativos em uma única etapa, *wireframes* permitem que esses elementos sejam analisados individualmente. Com isso, os *feedbacks* acontecem no começo do projeto e são mais direcionados.

A criação de *wireframes* é um processo que eventualmente é negligenciado e esquecido. Entretanto, ele é essencial para o desenvolvimento de um projeto *web* ou aplicativo.

## Criando *wireframes*

Criar *wireframes* pode ser tão simples ou tão complexo quanto a equipe do projeto desejar. Os *wireframes* necessitam abranger todas as informações que devem estar dispostas em cada tela/página do projeto.

Pense nos elementos gerais da maioria das páginas *web*: cabeçalhos, barras laterais, conteúdo, busca, *widgets*, rodapés etc. O *site* criado pode ter todos esses campos ou não. Isso depende do que o projeto demanda.

Em suma, um *wireframe* pode ser apenas um rascunho rápido feito durante uma reunião. Ele pode também ser criado digitalmente, mais próximo de um protótipo do que de um rascunho.

O estilo do *wireframe* que será criado dependerá da demanda do projeto, daquilo com que a equipe se sente acostumada ou do que necessita. Projetos mais complicados e complexos demandam *wireframes* mais elaborados e documentados, enquanto projetos menores podem ter *wireframes* mais simples.



Figura 4 – *Wireframe* feito a mão com régua

Fonte: <<https://cacoo.com/blog/8-must-see-wireframe-examples-inspire-next-design/>>.  
Acesso em: 12 set. 2019.

O tamanho da equipe também importa. Se o trabalho envolver uma equipe grande, com diversos profissionais trabalhando em diferentes partes do projeto, o ideal é que o *wireframe* seja o mais completo e explicado possível.

Já em uma equipe pequena, em que todos os envolvidos trabalham com comunicação constante, o *wireframe* pode ser mais simples e menos documentado.

Quando se cria um *wireframe*, é necessário ter em mente que todos os elementos das diferentes e diversas telas/páginas presentes no projeto precisam ser considerados e incluídos. Além disso, é preciso posicionar todos eles nos melhores pontos possíveis.

Para definir os locais ideais, uma série de estudos e testes sobre o assunto acabam sendo feitos. Alguns *designers* fazem rascunho no papel e depois reconstróem-no em *softwares* específicos. Outros iniciam seu processo já em programas. Existem ainda aqueles que preferem a estrutura apenas no papel.

Depois que essas informações já estão definidas na mente dos profissionais envolvidos (o ideal é sempre anotar todos os resultados), cria-se uma representação simples e de baixa fidelidade da página *web* ou do aplicativo.

Separamos as páginas em três partes distintas:



- ◆ O **cabeçalho**, que é a primeira coisa que o usuário verá no topo da página e que geralmente contém informações de identificação do *site*
- ◆ O **corpo da página web**, que concentra as informações disponibilizadas na página
- ◆ E o **rodapé**, que geralmente contém informações menos importantes



Por último, pensemos na navegação. É preciso adicionar botões e *links* para os usuários visitarem as principais seções do *site*/aplicativo.

É usual e recomendado experimentar possibilidades diferentes de *wireframes* antes de bater o martelo sobre a versão final escolhida para o projeto.

A grande vantagem desta etapa é a facilidade e a rapidez com que é possível criar um novo *wireframe*. Trabalhar com *wireframes* é uma maneira eficiente de planejar o *design* de um projeto *web* sem gastar horas trabalhando frente a códigos de projetos de teste. A flexibilidade dos *wireframes* permite trabalhar com diversas ideias de *layout* rapidamente e com pouco custo.

Não fique com receio de experimentar ideias inovadoras nesta etapa do processo de criação. Tenha em mente que *wireframes* podem ser feitos e refeitos rapidamente.

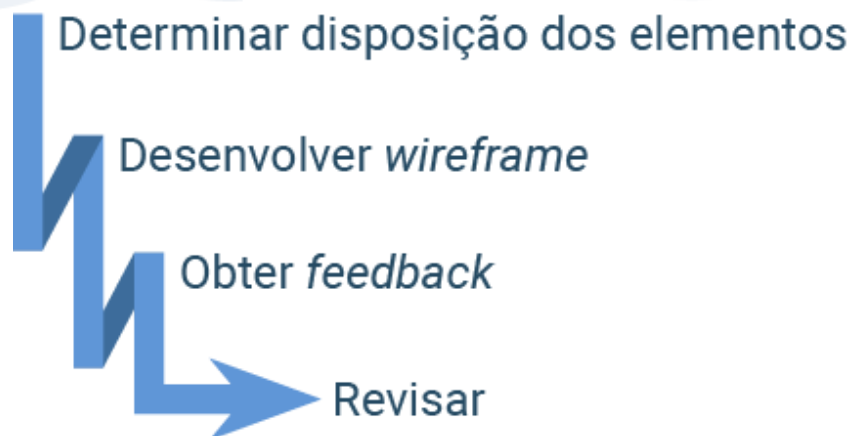


Figura 5 – Passo a passo da criação de um *wireframe*

Uma vez que um *wireframe* está pronto, você deve enviá-lo para outros membros da equipe para que eles o analisem e o validem. Caso a equipe seja pequena, é interessante esperar alguns dias antes de um veredito final.





Quando tudo estiver decidido, é o momento de marcar uma reunião com o cliente e apresentar o *wireframe*.

Ao longo do processo de criação, é altamente recomendado solicitar *feedbacks*, tanto do cliente quanto da equipe. Quando se está imerso em um trabalho, é difícil enxergar os problemas que estão embaixo do nariz.

Quanto mais cabeças pensarem juntas, melhor será o resultado. Logo, em vez de colocar todo o trabalho nas costas de apenas uma pessoa, o ideal é fazer com que o processo seja colaborativo.

Depois de finalizado, você poderá compartilhar o *wireframe* com todos os profissionais que compõem o projeto, definindo com clareza toda a base dele. Para facilitar o entendimento, muitas vezes são adicionadas anotações.

## Wireframe em rascunho x wireframe com programas

Uma importante questão quando trabalhamos com *wireframes* em um projeto é pensar em como ele deve ser visualmente. Existem duas possibilidades: ele pode ser feito em rascunho ou pode ser elaborado em um programa (sendo, neste caso, mais bem elaborado/apresentado).

Clique ou toque para visualizar o conteúdo.

### Wireframe em rascunho



Figura 6 – Wireframe de um aplicativo feito a mão

Fonte: <<https://cacao.com/blog/8-must-see-wireframe-examples-inspire-next-design/>>.

Acesso em: 12 set. 2019.

Caneta e papel são as ferramentas mais básicas para se criar um *wireframe*. Você pode querer usar um lápis para o seu esboço inicial, para apagar as coisas sem ter que começar de novo. Depois de terminar os elementos, você pode reforçar com caneta as linhas para evitar apagar o que não precisa mais ser apagado. Também é útil usar papel quadriculado, pois ele permite criar elementos em proporção.

*Wireframes* em rascunho nem sempre são bem-vistos pelos clientes. Eles podem considerar que o *wireframe* não está finalizado e pedir muitas alterações. Podem também ter dificuldade para entender as telas criadas.

## Wireframe com programas

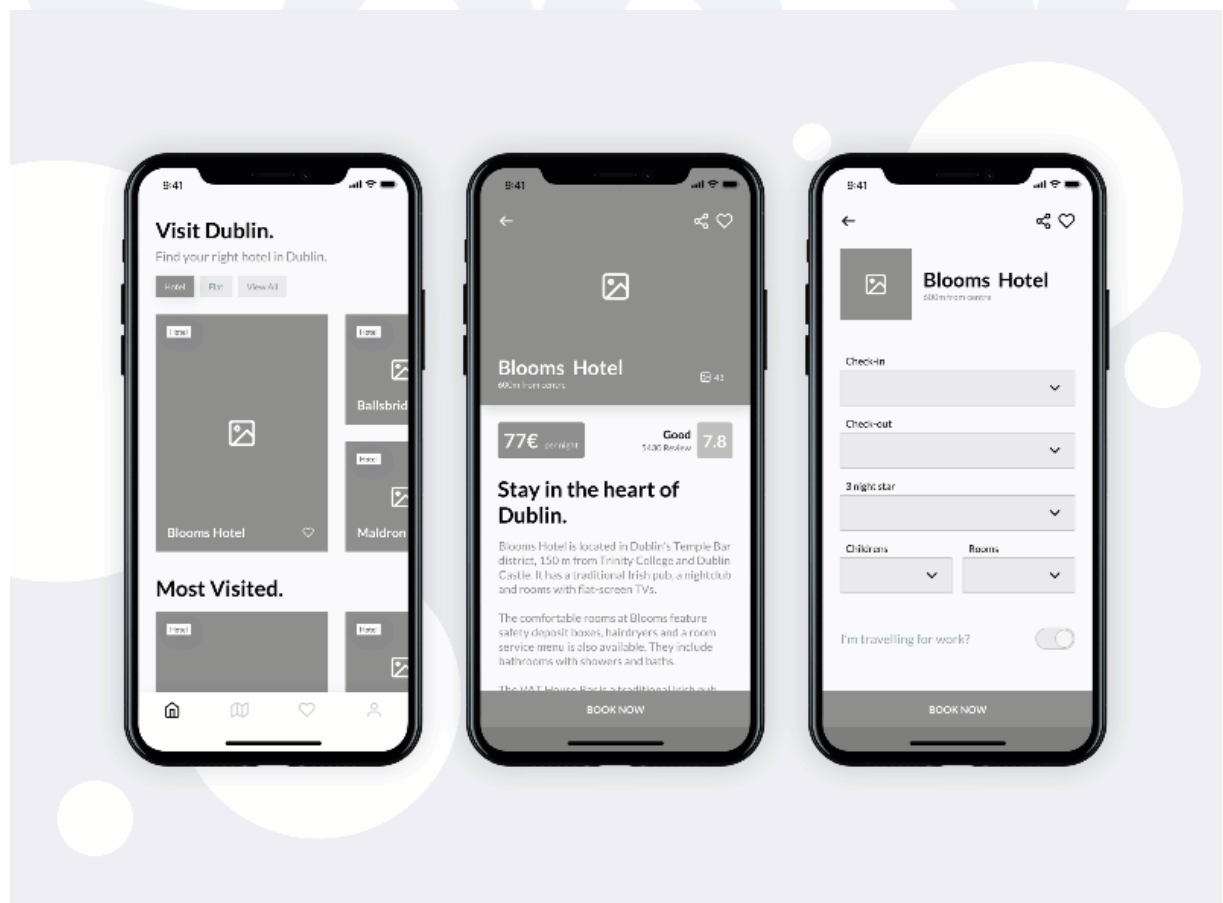


Figura 7 – *Wireframe* feito em software

Fonte: <<https://cacoo.com/blog/8-must-see-wireframe-examples-inspire-next-design/>>.

Acesso em: 12 set. 2019.

Um *wireframe* feito em um programa costuma ajudar o cliente a visualizar melhor as telas do *site*/aplicativo e ter uma ideia melhor de como tudo será estruturado e funcionará. Como os clientes não estão familiarizados com nomenclaturas, esboços e o dia a dia do desenvolvimento, o *wireframe* é uma maneira de tornar tudo mais simples e claro.

Mesmo que um *wireframe* seja feito em tons de cinza, não é um problema adicionar cores e outros recursos gráficos para que o entendimento do cliente – e dos próprios funcionários – seja melhorado.

Em contraste com o melhor aspecto de um *wireframe* criado em computador, aqueles em forma de rascunho são mais convidativos para se obter o *feedback* do cliente, por parecerem ainda estar em processo de criação, indicando a possibilidade de modificação. Um *wireframe* feito em programa passa a ideia de que o processo criativo já foi finalizado.

## *Wireframes* como resultado final do projeto

Assim como todos os membros da equipe precisam estar alinhados com o projeto e as suas características, o cliente também precisa estar a par de tudo. Apresentar *wireframes* é uma forma válida e direta de verificar se ele aprova o andamento do projeto e para colocá-lo em sintonia com o planejamento.



Figura 8 – Wireframe com as telas estruturadas

Fonte: <<https://cacao.com/blog/8-must-see-wireframe-examples-inspire-next-design/>>.

Acesso em: 12 set. 2019.

Como vimos, é muito mais fácil alterar a posição dos elementos quando se está trabalhando com *wireframes* do que quando se está com o projeto no Photoshop ou em páginas já codificadas. Mesmo que não seja impossível modificá-lo nestes casos, gera retrabalho.

## *Design de wireframes*

Algo muito importante quando se está lidando com aplicativos ou aplicações *web* é a experiência do usuário. É fundamental reter o usuário e aumentar a sua satisfação com a aplicação. Para tanto, é preciso foco em melhorar a usabilidade e facilitar a interação entre ele e o produto.

Para se alcançar uma boa experiência de usuário, existem algumas áreas que devem ser trabalhadas, as quais seguem.

## Arquitetura da informação

Hierarquia de informações é a maneira como é organizada a apresentação para o usuário final, posicionando e priorizando informações de maneira a levar o entendimento do funcionamento de um *site* ou aplicativo móvel aos futuros utilizadores. Isso exige muitas tentativas, com a busca do máximo de opções para apresentar as informações.

É importante sempre ter em mente o que se deseja que o usuário final saiba e como as informações devem chegar até ele. Pense que tudo deve ser moderado, de modo a não sobrecarregar o usuário com informações, ao mesmo tempo em que ele não deve ficar desorientado sobre o caminho a seguir.

## *Design* de navegação

O sistema de navegação é composto por diversos elementos que permitem ao usuário transitar de uma tela para outra. O objetivo deste recurso é tornar intuitiva a relação entre os *links* para que os usuários entendam sem instruções diretas as suas opções de navegação.

Tente visualizar o seu público-alvo e tome decisões sobre navegação tendo ele em mente. Isso tornará o seu aplicativo totalmente voltado para essas pessoas. Além disso, como o *design* será moldado desde a sua concepção até a conclusão, o aplicativo provavelmente terá um desempenho melhor com esse público do que com ajustes simples feitos no fim do projeto.

## Design de interface de usuário

O *design* de interface é o recurso que conduz a interação do ser humano com o aspecto virtual. O trabalho vai além do entendimento do problema do usuário e de suas necessidades: o *design* de usuário mescla conhecimento técnico e estético para construir ferramentas que incrementam e melhoram a forma como ele lida com o produto.

Na prática, o *design* de interface diz respeito à parte visual, à usabilidade, à arquitetura da informação, à navegação, à transição de telas. Ele tem o papel fundamental de oferecer soluções amigáveis e intuitivas para o usuário final.

Sempre pense sobre as características que farão os usuários usarem o seu *site*/aplicativo. Quando estiver familiarizado com o seu público-alvo, você poderá começar a expor as suas ideias e a tomar algumas decisões, como as seguintes.



- ◆ Qual é o objetivo principal do aplicativo? O que ele busca facilitar?
- ◆ A qual página se quer que o usuário vá?
- ◆ Como se quer que o usuário interaja com o sistema?

## Ferramentas para criar *wireframes*





Figura 9 – Wireframe feito em software

Fonte: <<https://cacoo.com/blog/8-must-see-wireframe-examples-inspire-next-design/>>.

Acesso em: 12 set. 2019.

O processo de visualizar a versão final de um *site* ou de um aplicativo é muito mais simples quando se cria um *wireframe*. Como sabemos, ele determina o que é necessário fazer antes mesmo do desenvolvimento de códigos e imagens.

Assim como existem inúmeras linguagens de programação, existem inúmeras ferramentas disponíveis para a criação de *wireframe*. Algumas são direcionadas justamente para isso, outras são ferramentas de edição que podem ser utilizadas para criá-los.

Ferramentas de criação de *wireframes* oferecem como vantagens a flexibilidade e a economia de recursos. É possível elaborar ideias, visualizar o escopo geral do projeto e evitar erros.

A escolha por uma ferramenta dependerá de três fatores: preferência pessoal, preferência do resto da equipe e características do projeto. Como tudo no mundo da tecnologia da informação, não existe ferramenta certa ou errada. Pode-se encontrar a ferramenta mais adequada para determinadas necessidades.

*Templates de wireframes* são fundamentais no início da criação de uma nova página *web* ou de um aplicativo. Há *templates* gratuitos de *wireframes on-line*. A maioria dos *softwares* para a criação de *sites* e aplicativos também oferece uma grande variedade de recursos.

*Softwares* que oferecem *templates* e outros recursos geram economia de tempo. Também facilitam o trabalho quando existe a demanda de uma grande quantidade e variedade de *wireframes*. Além disso, muitos produtos de *wireframes* fornecem ícones, *widgets* e temas.

O *software* de *wireframes* deve tornar o desenvolvimento do recurso mais rápido e simples.

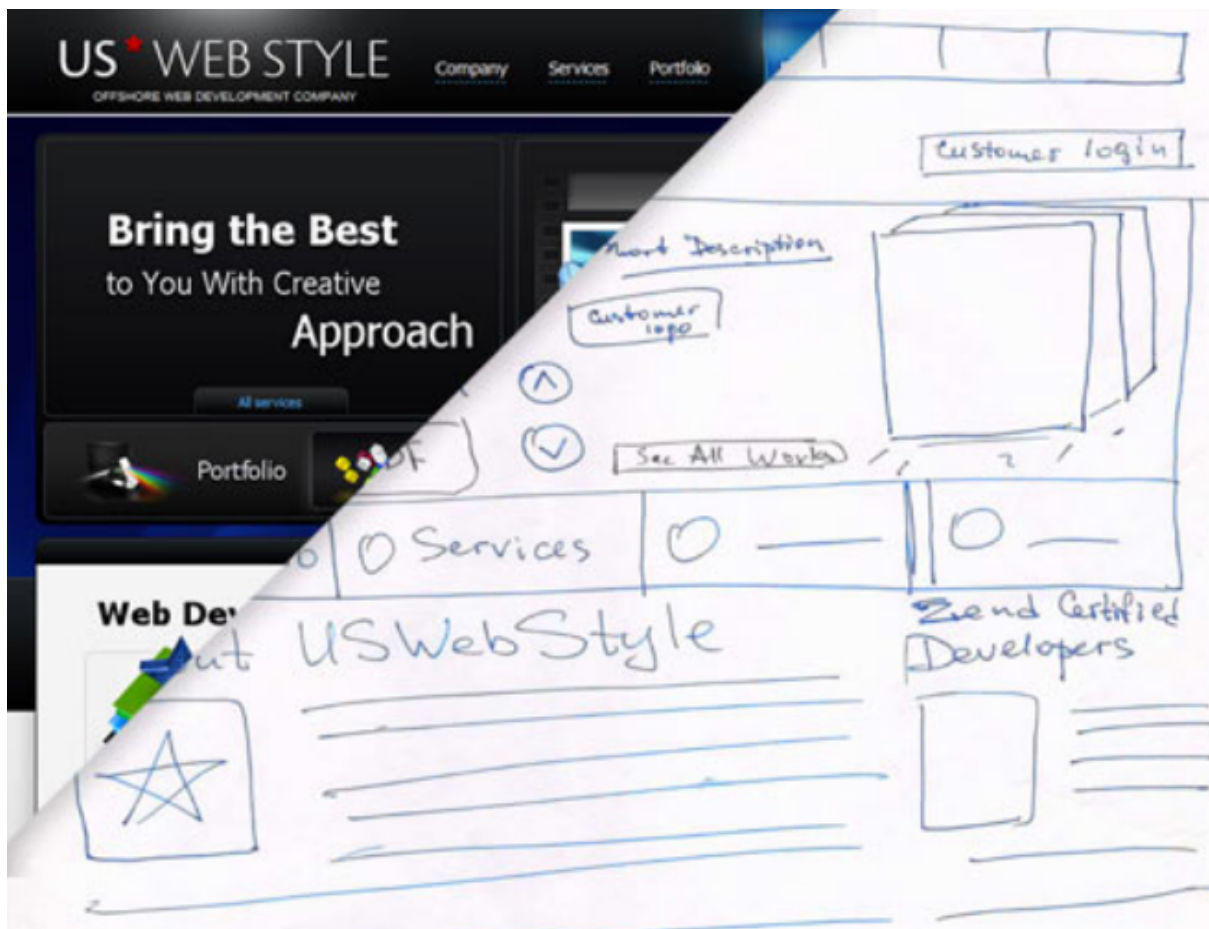


Figura 10 – Comparação entre *wireframe* e projeto finalizado

Fonte:

<[http://www.wireframeshowcase.com/wireframes/detail/uswebstyle\\_rebranding](http://www.wireframeshowcase.com/wireframes/detail/uswebstyle_rebranding)>.

Acesso em: 18 set. 2019.

Como vimos até aqui, a etapa de elaboração de *wireframes* é essencial em qualquer projeto. Por mais que seja necessário ajustar prazos para que essa criação aconteça da maneira mais adequada (com períodos para criação, edição, entrega, *feedback* e revisão), é claramente mais rápido, fácil e barato alterar elementos em um *wireframe* do que em uma estrutura HTML/CSS já pronta.

Se o seu objetivo é evitar alterar o *design* no futuro e criar um projeto criativo, único, bem-elaborado e otimizado, não deixe de usar *wireframes* em seus projetos.

# Prototipagem: conceito e formatos

A ideia de que um projeto será entregue em sua primeira versão é otimista demais. Toda criação, desde a sua concepção até a sua execução, é um processo que demanda tentativas e erros, ajustes e cortes.

Entregar ou lançar um aplicativo/projeto *web* sem validação é muito perigoso. O projeto pode não funcionar da maneira idealizada; o público-alvo pode não se identificar (ou até mesmo não compreender) a utilidade da criação; pode haver sugestões de soluções melhores e mais eficientes.

Para evitar retrabalho, é necessário validar o projeto antes de ele ser lançado. Para isso, existe a prototipagem.

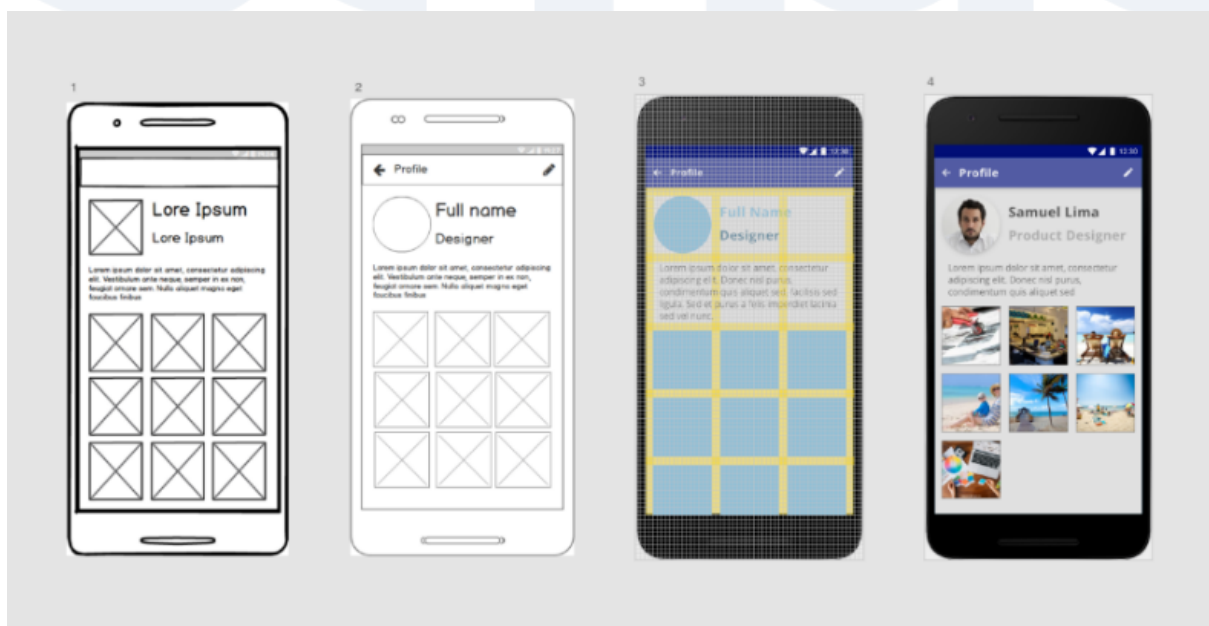


Figura 11 – Evolução de um projeto

Fonte: <<https://brasil.uxdesign.cc/>>. Acesso em: 9 out. 2019.

O protótipo serve para tornar tangível alguma ideia discutida nas reuniões, em que cada profissional interpreta de maneira diferente as funcionalidades listadas.

É na prototipagem que se define claramente como o produto será, independentemente do tipo de protótipo. Também é o momento de aperfeiçoar as ideias. Não é incomum, por exemplo, modificar bastante o projeto enquanto se testa o *wireframe* (cortar funcionalidades desnecessárias, ajustar as interações permitidas ao usuário etc.).

A prototipagem não se restringe apenas ao desenvolvimento de aplicativos e ao desenvolvimento *web*. É um conceito que se aplica tanto a produtos quanto a serviços. Entretanto, os protótipos variam conforme o projeto e a área de atuação.

Quando um projeto tem várias fases e etapas, é muito importante criar protótipos para cada uma delas. Como existem diferentes tipos deles, existe um modelo que se adapta melhor a cada situação.

Pular a etapa de prototipação de um projeto gera muito retrabalho e gastos. O tempo economizado quando se vai diretamente para a execução não será aproveitado. Quanto mais a equipe avançar no projeto, mais trabalhosa será a realização de modificações e correções no código. É muito mais fácil apagar linhas no papel ou modificar um arquivo de imagem do que alterar a programação.

Para ser possível criar o protótipo de um projeto, não é necessário que ele esteja em um estágio avançado. É plenamente recomendado prototipar até mesmo um produto com uma única interação ou funcionalidade. O primeiro protótipo, que servirá para verificar a estrutura geral do projeto, não precisa ser sofisticado. O ideal é que ele tenha um baixo custo e que, a partir dele, as melhorias no produto passem a ser feitas.

O nível de proximidade do protótipo com o produto final, chamado de fidelidade, pode variar de acordo com o seu propósito. Se foi feito para uma verificação mais simples, ele precisa ser diferente do que em uma situação em que a precisão será testada, por exemplo.

## Ciclo da prototipação

Primeiramente, é gerada a ideia de como o projeto deve ser, e desenvolve-se algum tipo de protótipo.

Depois disso, inicia-se uma bateria de testes e de validações para verificar se ele atende as expectativas dos futuros clientes, da empresa contratante e da própria equipe.

Por último, utilizam-se as informações adquiridas com os testes, e modificam-se diversos elementos da sua ideia inicial.

Por fim, um novo protótipo será criado, e a roda girará novamente.

## Ciclo da prototipação

Primeiramente, é gerada a ideia de como o projeto deve ser, e desenvolve-se algum tipo de protótipo.

Depois disso, inicia-se uma bateria de testes e de validações para verificar se ele atende as expectativas dos futuros clientes, da empresa contratante e da própria equipe.

Por último, utilizam-se as informações adquiridas com os testes, e modificam-se diversos elementos da sua ideia inicial.

Por fim, um novo protótipo será criado, e a roda girará novamente.

## *Wireframes x mockup x protótipos*

Os termos **protótipo**, **wireframe** e **mockup** costumam ser utilizados como sinônimos. Entretanto, dentro do mundo do desenvolvimento *web*, são três estágios diferentes que, eventualmente, podem se sobrepor um ao outro um pouco, com características e funções distintas.

Além dos *wireframes*, que já conhecemos bem, vejamos as duas outras alternativas de prototipagem.



## Mockups

O *mockup* é uma representação estática (sem interação), de média a alta fidelidade, do *design* de um projeto. Geralmente, o *mockup* é um protótipo com um *design* bastante próximo do *design* final do produto, podendo até mesmo ser o próprio *design* visual do produto final.

Este estágio está voltado para a definição dos elementos de *design* visual do *site*. São arquivos de imagens e incluem todos os gráficos, a tipografia e outros elementos da página.

Um *mockup* tem dois objetivos principais:

- ◆ Representar a estruturação da informação, tornar possível a visualização do conteúdo e demonstrar as principais funcionalidades de modo estático
- ◆ Focar os profissionais na análise e na revisão da parte visual do projeto, sem uma preocupação direcionada apenas para funcionalidades e interações

Os *mockups* são particularmente úteis quando se quer vender a ideia do produto antes de ele estar pronto para o cliente. Como são essencialmente visuais, os *mockups* não têm a resistência de protótipos em *wireframes* (baixa fidelidade), ao mesmo tempo que proporcionam custo mais baixo e mais rapidez do que um protótipo (alta fidelidade).

*Mockups* são, essencialmente, uma ótima maneira de coletar *feedback*.

## Protótipos



São leiautes semifuncionais das telas e servem para testar características gerais do projeto, buscando maior fidelidade para a futura versão final do *site*/aplicativo.

Nesta etapa, adicionam-se os elementos que possibilitam a interação com elementos do *site*/aplicativo, e simula-se o funcionamento do projeto. Protótipos podem ou não incluir elementos de *design* finalizado.

Um protótipo é uma representação de média a alta fidelidade do produto final e possibilita ao usuário testar as suas interações. O principal objetivo do protótipo é possibilitar testes da interação em um estágio muito próximo ao do produto final, seja com usuários, seja com os próprios desenvolvedores.

Um protótipo é uma simulação da interação final entre o usuário e a interface. Deve-se buscar a similaridade com o produto final, mesmo que ele não seja exatamente idêntico. O principal é que as interações sejam o foco. Elas devem ser construídas com cuidado, de modo a apresentar uma alta similaridade com a experiência final que o usuário do projeto deverá ter.

Por se tratar de um protótipo que pode ser refeito e que precisa reduzir custos ao máximo, a comunicação entre a interface e o funcionamento do *back-end* costuma ser reduzida.

Protótipos são utilizados em seu potencial máximo quando são aplicados em testes com o usuário. A simulação das interações finais permite testar a usabilidade da interface antes de o desenvolvimento iniciar de fato, sem o risco de não funcionarem quando a versão final for lançada.

Criar protótipos exige investimento de dinheiro e de tempo. Nenhum dos tipos de protótipo exclui outro. O ideal é que todos sejam realizados ao longo do projeto.

Entretanto, nem sempre há para o projeto os recursos necessários para realizar todas as etapas. Uma sugestão é criar protótipos que possam ser utilizados no desenvolvimento, o que é eficaz em projetos simples e pequenos.

# Por onde começar?

Existem diferentes ferramentas de prototipagem, muitas especializadas em uma etapa específica. Segue uma pequena lista de programas que podem ser utilizados para criar *wireframes* e protótipos.

Todos estão disponíveis *on-line* e oferecem opções gratuitas.

## Wireframe.cc

É uma ferramenta simples e minimalista para a criação de *wireframes*.

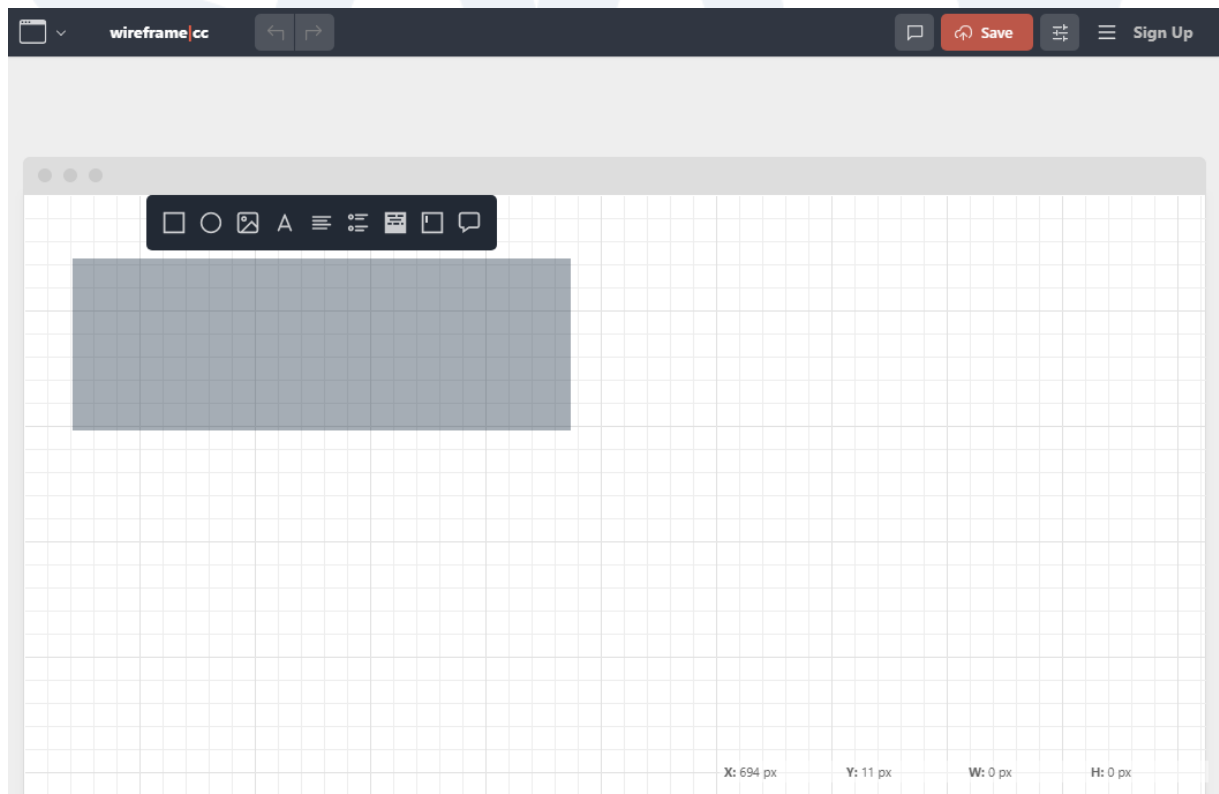


Figura 12 – Wireframe.cc

## Mockflow

É uma das mais usadas ferramentas *on-line* de *wireframes*. Há uma versão gratuita, mas ela se restringe a um único projeto.

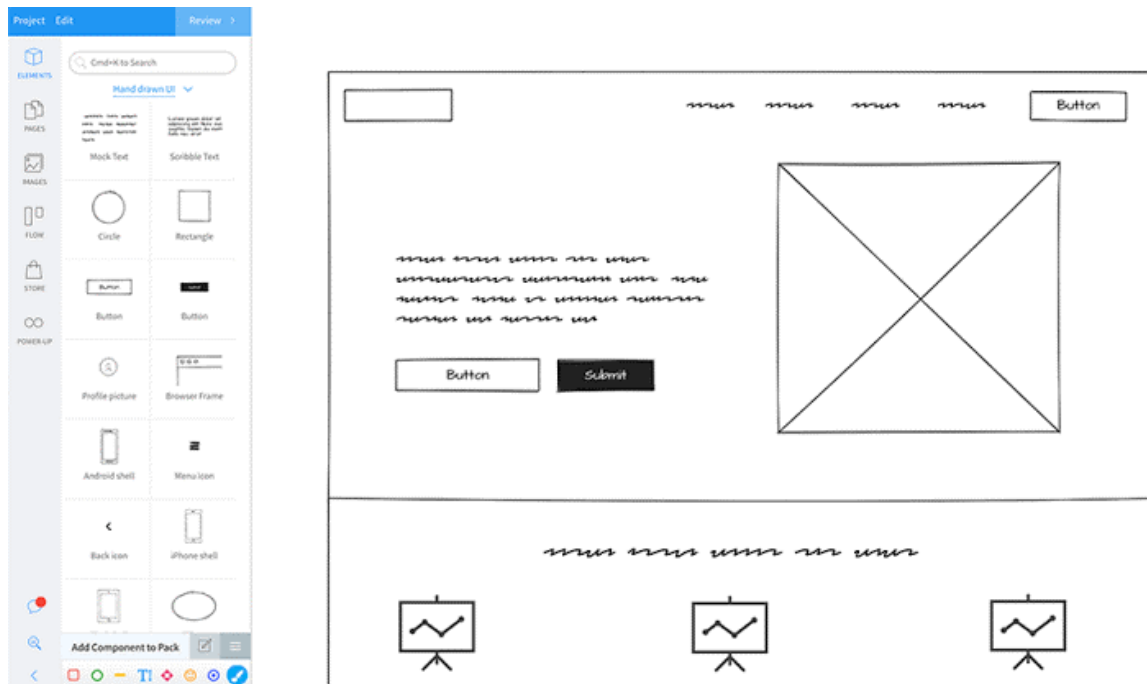


Figura 13 – Mockflow.com

## Figma

Conta com vários recursos de *wireframe* e permite que mais de um *designer* interaja ao mesmo tempo em um leiaute. Também conta com versões pagas e gratuitas (com restrição de três projetos e dois editores).

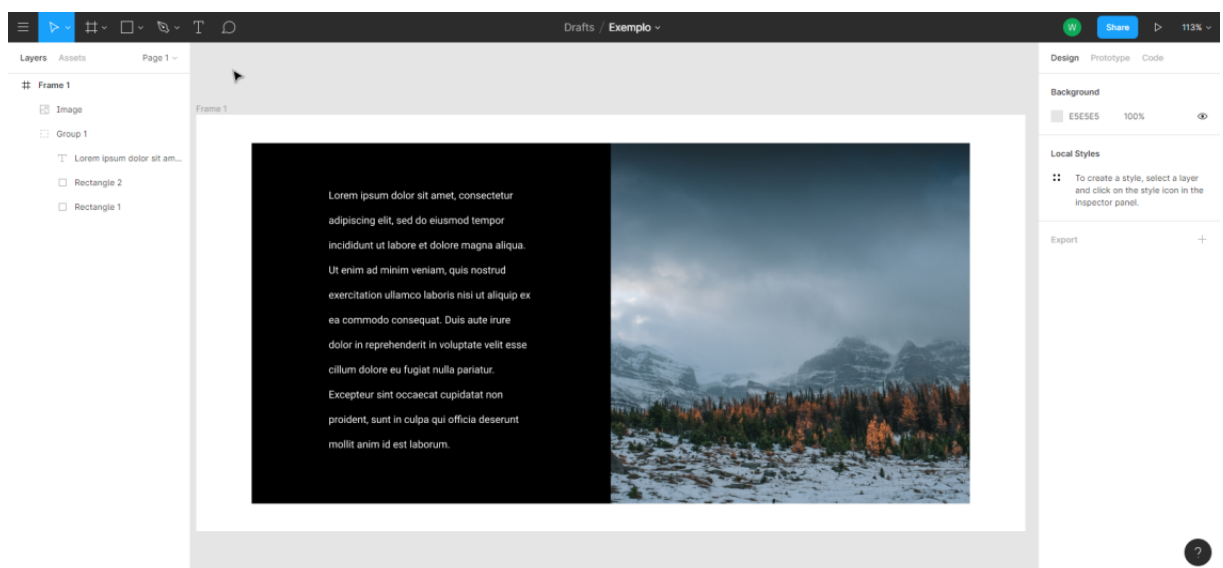


Figura 14 – Figma.com

Assista à videoaula disponível na UC para acompanhar uma prática de criação de *wireframe* usando o Figma.

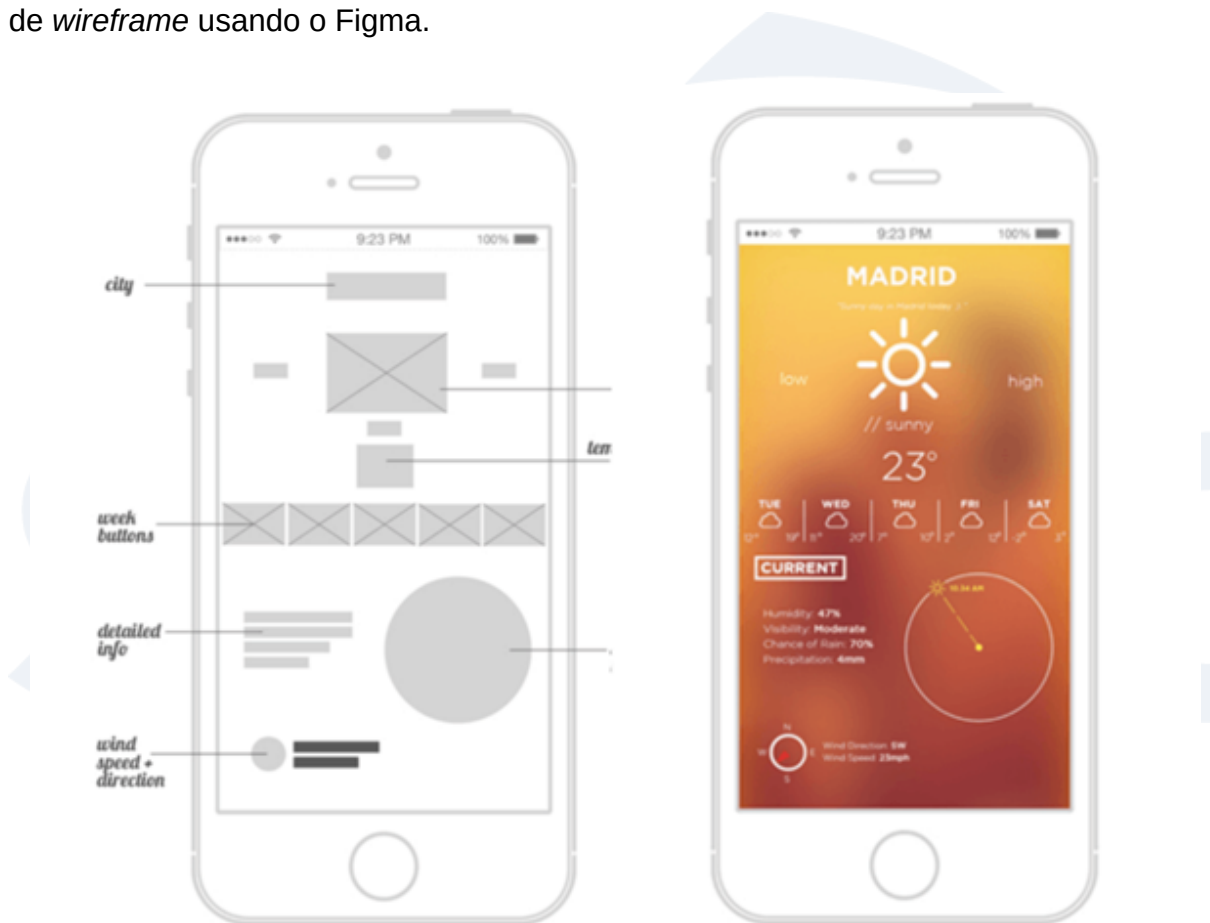


Figura 15 – *Wireframe* e versão finalizada do mesmo projeto

Fonte: <<https://learn.g2.com/wireframe>>. Acesso em: 20 set. 2019.

Feita corretamente e em combinação com testes de usuário, a criação de protótipos consegue pagar seu custo. Geralmente, o *wireframe* vem em primeiro lugar. Depois, vem a etapa de *mockup*; e, se a empresa tiver tempo disponível, cria-se o protótipo.