

UCD: Abreviação para User Centered Design. É o ciclo de desenvolvimento de um software centrado na utilização de seu usuário. É dividido em 5 etapas: Compreender o Contexto de Uso, Especificar Requisitos dos Usuários, Construir Soluções de Design, Protótipo e Avaliar Design *ante* Requisitos dos Usuários. As duas primeiras etapas são focadas em levantar requisitos através de perguntas em entrevistas, questionários e etc para se entender as maiores necessidades de um usuário que o sistema poderá resolver, enquanto as outras 3 etapas são focadas na realização do projeto em si a partir dos requisitos.

Personas: São criadas para se entender melhor como o usuário irá interagir com o software para resolver suas dores. Para isso se cria o perfil delas, com nome, idade, ocupação, cidade, personalidade e objetivos. Então inserimos as personas em cenários, que correspondem à histórias sobre elas, tudo isso feito de acordo com os requisitos levantados. Por exemplo, se uma equipe for fazer um software que organize os estudos de um usuário para encaixar no cronograma de provas e vestibulares dele, poderia-se criar a seguinte persona:

Eduardo é um jovem de 17 anos, mora em Minas Gerais e estuda em colégio particular. É apenas estudante e está começando seu período de vestibular, no qual ele vai tentar cursar medicina. Porém, no meio disso ainda há as provas e os trabalhos que ele precisa realizar no colégio dele. Ele tem medo de não conseguir focar no colégio e no vestibular ao mesmo tempo e acabar com desempenho baixo em algumas de suas notas. Logo, ele está enfrentando dificuldades para se organizar nos dois campos. As necessidades dele são se organizar com uma agenda que não seja exaustiva em questão de estudo e conseguir dividir os conteúdos das matérias para não se deixar acumular nada.

Logo, a partir do requisito do sistema criou-se uma persona que será um possível usuário do software que será criado. Na hora de criar a persona não se esquecer de colocar seu nome, suas informações pessoais, suas dores e suas necessidades.

Protótipos: É a concepção de fragmentos do sistema, vai permitir com que haja a conversão dos requisitos de um projeto em algo mais tangível. Se prototipa um sistema para identificar de forma mais rápida as falhas e para conseguir fazer uma avaliação visual do sistema.

Existem 3 tipos de protótipo: o de baixa, o de média e o de alta fidelidade. Desse modo, há uma evolução nos protótipos, dos menos até os mais detalhados.

O protótipo de baixa fidelidade é o que mais possui detalhes ausentes, e é feito no papel através de desenhos a lápis do que será o projeto final. Esse protótipo é realizado no começo do desenvolvimento e será feito em Mockups (apenas desenhos das interfaces, ajudando em brainstorming) ou em Storyboards (permitindo simulação de fluxos). Como são feitos em papel, esse tipo de protótipo é chamado de Non-Computer, e é o tipo mais barato e descartável.

Os protótipos de média e alta fidelidade já dependem de um computador, ou seja, são Computer-Based. Nesses casos, são protótipos que permitem um desenvolvimento e uma edição mais fácil e rápida das telas, sendo recomendados quando já se tem um design básico decidido.

O protótipo de média fidelidade é feito normalmente em powerpoint ou no Balsamiq. É marcado por já começar a ter características mais próximas do modelo final e algumas funcionalidades, mas não possui todas as interatividades possíveis no sistema. Ou seja, esse protótipo é uma continuação do de baixa fidelidade e é criado para refinar o layout e inserir os primeiros botões com comportamentos básicos.

Por fim, o protótipo de alta fidelidade normalmente é feito no Figma e é o mais completo dos três. Utilizado já em uma fase mais avançada do desenvolvimento, ele apresenta uma interatividade completa com todos os botões que serão usados no software e já possui a paleta de cores necessárias para o projeto, ou seja, é essencial a consistência visual nesse protótipo seja a partir das cores, das fontes ou do tamanho dos botões. É o mais próximo de como será a versão final do software.

Análise de Tarefas: É uma técnica na qual se prevê os caminhos que uma pessoa vai utilizar para realizar um objetivo dentro do software. Pode ser feito de forma escrita ou de forma gráfica, sempre apresentando uma hierarquia que vai das primeiras até as últimas ações, sempre em ordem.

Exemplo Escrito:

Pegar um livro emprestado na biblioteca

0. Para pegar emprestado um livro da biblioteca

1. ir até a biblioteca
2. encontrar o livro
 - 2.1 acessar acervo da biblioteca
 - 2.2 acessar tela de busca
 - 2.3 entrar com o critério da busca
 - 2.4 identificar o livro desejado
 - 2.5 anotar localização do livro
3. ir até a estante certa e pegar o livro
4. levar o livro ao balcão de empréstimos

Exemplo Gráfico:

