



# Desenvolvimento de Sistemas

---

## Experiência do usuário

### Introdução

Você já leu ou ouviu, em algum momento, a expressão *user experience* – UX (ou, em tradução livre, “experiência do usuário”). Ela ganhou popularidade no mundo do *design* nos últimos tempos, mas não está limitada a essa área, sendo também aplicada, por exemplo, à área de tecnologia da informação (TI), principalmente no setor de desenvolvimento de *software* (*web*, *mobile* ou *desktop*).

A UX é importante por considerar a interação do cliente ou do usuário com os produtos, com os equipamentos ou com os aplicativos em todos os processos de interação, pensando nas suas experiências, para tornar o uso das soluções mais eficientes e principalmente para surpreender e encantar o público.

Dada a importância do tema, serão apresentados, de forma simples, os conceitos, os fundamentos, as etapas, os segmentos de aplicação, as técnicas de identificação e heurísticas e ainda alguns *cases* de sucesso.

## Fundamentos

### O que é experiência do usuário?



*A experiência do usuário inclui todas as emoções, crenças, preferências, respostas físicas e psicológicas, comportamentos e realizações do usuário que ocorrem antes, durante e após o uso. (...) Existem três fatos que influenciam a experiência do usuário: o sistema, o usuário e o contexto de uso. (BRASIL, 2011)*

Essa definição, encontrada na ABNT NBR ISO 9241-210, traz recomendações centradas no ser humano para todo o ciclo de vida de sistemas interativos computacionais. A norma é destinada àqueles que gerenciam processos de projeto e se preocupam com a forma com que componentes, tanto de *hardware* quanto de *software*, podem aprimorar a interação entre humano e sistema.

Pesquise na Internet o vídeo *Don Norman: o Termo "UX"*, disponível no YouTube pelo NNgroup. No vídeo, Donald Arthur Norman define "UX". Para muitos especialistas, ele foi o criador da expressão na década de 1990. O áudio do vídeo está em inglês com legendas em português.

É possível sintetizar a expressão "*user experience*" como o conjunto das interações ativas, passivas e secundárias que os usuários experimentam e vivenciam nos *softwares* e nos aplicativos desenvolvidos por você ou pela empresa em que você atua. Muito provavelmente, quanto mais significativa e melhor for a experiência, maiores serão a satisfação, o engajamento e a fidelização do cliente e o *net promoter score* (o quanto o cliente indica o produto para outras pessoas).

Confira um breve estudo de caso para entender melhor:

Um sistema de controle de estoques de uma loja de venda de equipamentos eletrônicos somente é executado ("roda") em rede interna acessando uma única base de dados no servidor. Essa mesma base de dados é consultada para a finalização da compra, ou seja, para a emissão da nota fiscal pelos atendentes.

João estava almoçando perto da loja e, ao retornar do seu intervalo de almoço, vê na vitrine uma televisão que desejava muito com um desconto alto. A partir daí, ocorrem os seguintes passos:

- 1 João entra na loja para comprar o produto.
- 2 O atendente localiza a televisão.
- 3 João e o vendedor realizam os testes.
- 4 O atendente inicia o processo de emissão da nota fiscal, digitando o nome e o CPF (cadastro de pessoa física) de João.
- 5 Quando da leitura do código de barras do produto, o sistema sinaliza que o produto não está cadastrado.
- 6 O atendente informa que pedirá para o responsável cadastrar o produto para realizar a venda.
- 7 O atendente retorna com a informação de que o responsável está no intervalo do almoço e demorará uns 40 minutos para retornar.

**Como você acha que foi a experiência do João? Ele comprou o produto? O resultado foi positivo para a empresa? A imagem passada foi boa?**

Percebe-se que o sistema estava funcionando perfeitamente, sem falha ou *bugs*. No entanto, a **experiência do cliente foi péssima**, resultando em prejuízos financeiros e para a imagem da empresa.

## Objetivos



É preciso descobrir e criar formas de oferecer um serviço diferenciado para obter mais satisfação e fidelização dos clientes por meio do estudo do comportamento humano.

A ideia central não é apenas **atender às expectativas do cliente, e sim superá-las**, por meio de processos fluidos, livres de falhas e assertivos, sem gerar confusões na relação com o produto ou o serviço, proporcionando significado, encantamento e desejo de manter o vínculo com você ou com a sua empresa.

Para perceber a importância da UX, veja um exemplo prático:

- 1** Imagine que você quer comprar um tênis novo pela Internet.
- 2** Após fazer uma busca no Google, você clica no anúncio que mais gostou.
- 3** Você é então direcionado para a loja virtual, mas a página não carrega, a navegação não é intuitiva e as informações estão desorganizadas.

**Como foi sua experiência? Muito provavelmente foi péssima. Qual foi sua atitude? É bem provável que você tenha saído imediatamente do *site* e acessado outra loja.**

Se o cliente não tem uma boa experiência com produtos e serviços diversos procurará algo melhor na concorrência. No entanto, no caso de uma experiência encantadora, o cliente será fidelizado e se tornará um promotor da sua marca e do seu produto, recomendando-os para outras pessoas (PATEL, 2022). Logo, em um mercado competitivo, o foco no cliente e nas experiências dele é determinante para o sucesso de um produto, uma marca, um *software* ou um aplicativo.

Ciente da importância do assunto, conheça a seguir os elementos da UX.

## Elementos da UX

Para ajudar os clientes a entenderem por que eles devem ir além da usabilidade, Morville (2004) criou um diagrama em formato de colmeia, orientando como uma boa UX deve ser.



## Figura 1 – Elementos da UX

Fonte: Adaptado de Morville (2004)

### Útil

Os profissionais da área devem ter coragem e criatividade para questionar se os produtos e/ou os sistemas são úteis e, com base nas respostas, criar soluções inovadoras.

### Usável

A facilidade de uso é vital, pois a utilização apenas de métodos e perspectivas de interação entre usuário e máquina, centrados tão somente na interface, não corresponde a todas as dimensões do *design* para o *desktop*, a *web* ou o *mobile*. Logo, para Morville (2004), “usabilidade é necessária, mas não suficiente”.

### Desejável

A busca da eficiência deve considerar o poder e o valor da imagem, a identidade, a marca e outros elementos característicos do emocional das pessoas.

### Valorosa

Os sistemas precisam agregar valor (não necessariamente monetário) aos negócios.

### Localizável

O *software* deve ter boa navegabilidade e objetos localizáveis, de modo que os usuários encontrem o que precisam mais facilmente.

## Acessível

Considerar a acessibilidade para pessoas com deficiência (PCDs) é fundamental. Se construções precisam de elevadores, rampas, braile, sons, cores, por que os sistemas não precisariam de acessibilidade?

## Crível

É necessário utilizar elementos da construção do *design* da aplicação que influenciam a confiança dos usuários (certificados de segurança, criptografia, segurança da informação etc.).

Agora que você já sabe como uma boa UX deve ser, chegou o momento de conhecer as etapas do processo de desenvolvimento pelas quais uma aplicação deve passar.

## Etapas

Rodrigues (c2020) elenca sete etapas, descritas a seguir, que devem estar presentes no desenvolvimento de projetos orientados à UX.

### 1ª etapa: *briefing*

Geralmente, *briefing* é um documento gerado com base em uma reunião na qual são identificadas várias informações importantes para o desenvolvimento de uma ideia ou de um projeto. O nível de detalhamento deve ser suficiente, sem ser muito extenso, para evitar erros de interpretação ou equívocos entre as pessoas que vão utilizá-lo.

É importante que o objetivo a ser atingido com o projeto esteja bem definido. Também devem constar o problema a ser resolvido e as principais funcionalidades. Assim, pode-se concluir que o *briefing* é o guia para todo o projeto.

Aliás, uma reunião de *briefing* bem realizada responde aos seguintes questionamentos:

Qual é o problema a ser resolvido?

Quais são os possíveis usuários do produto?

Qual é o contexto em que estão inseridos?

Quais são as possíveis soluções?

Como se chegou a tal problema?

Quais são as principais funcionalidades?

## 2ª etapa: investigação

A etapa de investigação identifica o **público-alvo** do produto em desenvolvimento, sendo necessário, então, observar, registrar, analisar e identificar as características mais relevantes que distinguem um grupo de pessoas de outro.

Para obter essas respostas, é possível utilizar duas abordagens distintas:

### Análise quantitativa

Busca coletar dados mensuráveis sobre a opinião de um grande número de usuários sobre determinada ideia ou projeto. Pode ser realizada por meio de questionários. Por exemplo, é possível identificar a faixa etária dos possíveis consumidores do produto, os estados e as cidades onde o produto é mais consumido ou utilizado.

### Análise qualitativa



Deve ser realizada em conjunto com o usuário, pois os dados são obtidos por meio de observação e entrevista sobre os aspectos que o motivam a utilizar o *software* ou sobre o porquê ele utiliza o sistema de determinada maneira. A análise qualitativa permite identificar o que resolver e como resolver com base na experiência do usuário.

### 3ª etapa: *benchmarking*

A etapa de *benchmarking* analisa os **concorrentes**. Sendo assim, podem-se identificar as melhores ferramentas ou práticas que estão sendo utilizadas para poder implementá-las no projeto ou na empresa ou para melhorar alguma estratégia de negócio com o objetivo de tornar-se mais competitivo no mercado. Ainda é possível identificar o que a concorrência entrega para os usuários, quais são os pontos fortes, os pontos fracos e, principalmente, as oportunidades de melhoria (inovação).

### 4ª etapa: personas

As personas são personagens fictícios criados para representar diferentes tipos de usuários com base nas informações extraídas nas etapas de investigação e *benchmarking*. Existem algumas ferramentas gratuitas, entre as quais merecem destaque a **Make My Persona** e o **Gerador de Personas**. Elas são facilmente encontradas fazendo uma busca no Google. Experimente criar uma persona!

### 5ª etapa: prototipagem

A prototipagem é uma etapa importantíssima, pois nela é validado o produto por meio **da simulação do seu funcionamento**. Inclusive, podem ser criados três níveis de fidelidade:

#### Baixa fidelidade

É a prototipação com baixo grau de detalhamento utilizada para apresentar apenas funcionalidades, ou seja, neste nível não são consideradas as interfaces de interação com os usuários.

## Média fidelidade



Busca definir e validar o conteúdo da interface por meio de simulações não muito complexas, conhecidas como *wireframes*. Este nível permite simular o comportamento esperado com um pouco mais de similaridade e proximidade ao mínimo produto viável (tradução do inglês *minimum viable product* – MVP).

## Alta fidelidade

Permite validar as interações e as experiências do usuário com a aplicação em desenvolvimento.

## 6ª etapa: testes de usabilidade

O teste de usabilidade é um teste de **qualidade** que objetiva medir o grau de satisfação e de *performance* com relação à interface do produto. O intuito é identificar, além da quantidade, quais são os erros que podem ser cometidos e qual é a possibilidade de realização das tarefas. Os testes devem ser realizados pelo público-alvo sob a observação do analista, que observa, ouve e anota tudo. Deve estar claro para os participantes que é o *software* que está sendo testado, e não o usuário.

## 7ª etapa: implementação

A implementação é o momento final do processo, na qual é realizada a entrega do produto. Essa etapa ainda permite realizar alguma alteração com base nas informações elencadas nas etapas anteriores. Lembre-se de que as pessoas mudam seus comportamentos e seus gostos, o que impacta diretamente a UX.

# Técnicas de identificação e heurísticas



Como já apresentado anteriormente, projetar e desenvolver um sistema sem pensar na UX e na UI (*user interface*, ou, em tradução livre, “interface do usuário”) resultará em experiências desagradáveis. Portanto, a experiência e a usabilidade devem ser consideradas antes, durante e após o desenvolvimento de um projeto de *software*.

Pensar na UX, durante todas as fases do projeto, é projetar uma interface que não cause insegurança, que deixe claros os resultados das ações e que garanta a realização, de forma simples e eficiente, de todas as tarefas pelos usuários. De uma maneira mais simples, é fazer com que “o usuário consiga utilizar a aplicação sem a necessidade de ter um manual de instruções em mãos” (MOMA, 2017).

Quanto ao termo “heurística”, tema que também será tratado neste tópico, veja o que diz o dicionário:

**Heurística:** “1 arte de inventar, de fazer descobertas; ciência que tem por objeto a descoberta dos fatos; heurética [...] 1.2 INF método de investigação baseado na aproximação progressiva da solução de um dado problema” (HOUAISS, 2022).

Para projetar uma boa interface, que proporcione uma excelente experiência, existem dez heurísticas criadas por Jakob Nielsen:

## Visibilidade do *status* do sistema

De uma maneira geral, as pessoas são muito dependentes da visão, mas, em um ambiente natural, outros sentidos (audição, olfato, tato) complementam as informações sobre o que acontece ao redor. Quando se migra para um ambiente estritamente digital, a dependência da visão é evidentemente maior. Por esse motivo, é importantíssimo que a interface a ser desenvolvida deixe claro para o usuário onde ele estava, onde ele está e para onde ele pode ir.

A figura a seguir mostra, à esquerda, a barra de navegação que indica a localização (**Página Inicial**) e as possibilidades de locais para onde o usuário pode ir (**Novo** e **Abrir**, entre outras). À direita, estão **Nova Apresentação em Branco** ou modelos pré-formatados (*templates*) e, abaixo, as últimas apresentações criadas.

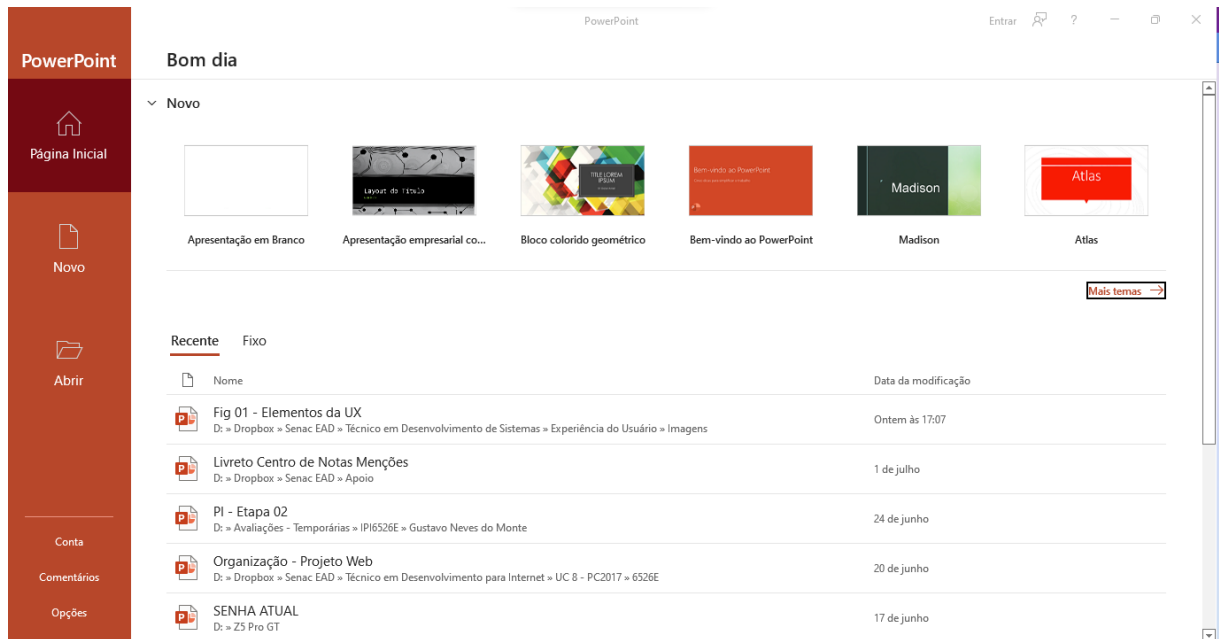


Figura 2 – Tela de boas-vindas do Microsoft PowerPoint 2021

Fonte: Microsoft PowerPoint (2022)

## Compatibilidade entre o sistema e o mundo real

É necessário considerar também que a aplicação a ser desenvolvida precisa se comunicar com o usuário, ou seja, deve falar “a mesma língua” não apenas em idioma (inglês, português), mas também na linguagem cotidiana utilizada por ele. Dessa forma, a escolha correta dos símbolos e dos ícones é um fator importante que ajuda na compatibilidade e na compreensão das informações.

A figura a seguir mostra a tela de boas-vindas do *desktop* para declaração do Imposto sobre a Renda da Pessoa Física (IRPF). Observe a presença de ícones de utilização global e de compreensão universal para abrir, excluir e criar uma nova declaração. Você consegue identificá-los?

Figura 3 – Tela de boas-vindas do IRPF 2022

Fonte: BRASIL (2022)

## Controle e liberdade para o usuário

Dar controle e liberdade para o usuário significa possibilitar a ele desfazer ações que possam ter sido realizadas acidental ou intencionalmente, ou seja, é uma “saída de emergência”. Isso transmite confiança ao usuário para utilizar a aplicação, pois o deixa seguro com relação à possibilidade de reverter uma ação indesejada.

Um exemplo clássico é a exclusão indesejada ou precipitada de informações e/ou arquivos. Logo, deve-se criar um mecanismo de recuperação, mesmo que temporário, para que o usuário possa desfazer a ação.

## Consistência e padronização

Uma das heurísticas mais simples e muitas vezes negligenciadas na construção de interfaces é a de consistência e padronização. O foco deve estar em manter um padrão replicável em todas as telas do sistema em produção, para que o usuário não precise aprender a interagir em diferentes telas do *software*.

Como exemplo, pode-se fazer referência ao Pacote Office da Microsoft, pois a forma como são apresentadas as interfaces do Word, do Excel e do PowerPoint é muito semelhante, permitindo identificar qual delas está sendo utilizada somente pela sua cor e/ou pelos seus ícones. Veja:

Figura 4 – Barras de ferramentas do Word 2021, do PowerPoint 2021 e do Excel 2021

Fonte: Adaptado de Pacote Office da Microsoft (2022)

## Prevenção de erros



Normalmente, os usuários cometem dois tipos de erro: o “deslize” e o “engano”. O primeiro ocorre quando um usuário pretende realizar uma ação, mas, por não estar totalmente focado na ação desejada, realiza outra. Já o segundo ocorre por falta de compreensão da informação ou por interpretação equivocada da informação.

A utilização de caixas de confirmação, por exemplo, é uma excelente prática para prevenir erros:

Figura 5 – Caixa de confirmação de aplicação *desktop* Java

Fonte: Java (2022)

## Reconhecimento em vez de memorização

É mais adequado entregar meios de reconhecimento de padrões ao usuário do que obrigá-lo a memorizar várias informações enquanto utiliza a aplicação. Isso porque o cérebro é muito bom no reconhecimento de padrões, então expor dicas, ações e opções familiarizará o usuário com a aplicação.

Pegue uma folha de papel. Escreva nela a senha do seu *e-mail* (memorização).

Sem acessar o Microsoft Word, escreva o processo de como salvar o arquivo que você está trabalhando (reconhecimento).

Não há resposta certa ou errada. Apenas reflita sobre o assunto e entenda o significado de reconhecimento e memorização.

## Eficiência e flexibilidade de uso

É fundamental criar interfaces úteis tanto para usuários leigos quanto para experientes. Sempre considere que leigos necessitam de informações mais detalhadas para realizar tarefas, mas, à medida que reconhecem a interface e se familiarizam com ela, opções de interação mais rápidas serão necessárias.

Os exemplos mais clássicos são os atalhos de teclado para as ações de copiar (**Ctrl + C**), colar (**Ctrl + V**) e alternar entre janelas abertas (**Alt + Tab**), presentes nos sistemas operacionais.

## Estética e *design* minimalistas

A heurística referente à estética e ao *design* minimalistas orienta a manter visível somente as informações estritamente necessárias, pois, quanto maior for a quantidade de informações, mais difícil será para o usuário tomar uma decisão. Informações e/ou ações secundárias podem ser colocadas em segundo plano, utilizando-se de recursos como barra de ferramentas, menus e abas.

Um bom exemplo é o MySQL Workbench:

Figura 6 – Tela de aplicação do MySQL Workbench

Fonte: MySQL Workbench (2022)

## Reconhecimento, diagnóstico e recuperação de erros

O usuário precisa poder corrigir um erro, caso ele ocorra. Avisos nos campos de formulário, caso estes não tenham sido preenchidos corretamente, são uma excelente medida. Veja:

Figura 7 – Formulário de *login* do Gmail

Fonte: Gmail (2022)

## Ajuda e documentação

É muito importante que a ajuda e a documentação constem no sistema, ainda que sejam pouco utilizadas. Porém, caso o usuário precise sanar alguma dúvida, ele pode consultá-las. Além disso, um *software* é composto de seus arquivos e sua documentação.

A figura a seguir demonstra o acesso à ajuda e à documentação no editor Visual Studio Code da Microsoft, ferramenta muito utilizada por desenvolvedores de *software*.

Figura 8 – Menu de ajuda do Visual Studio Code

Fonte: Visual Studio Code (2022)

## Segmentos de aplicação de UX

A UX vai além do desenvolvimento de aplicações computacionais. Ela pode ser aplicada a todo segmento de negócio que deseja fidelizar e manter seus clientes, bem como ser provido por eles. Além disso, a UX ganhou muita força e aplicação nas últimas décadas, quando as empresas entenderam sua importância e seu retorno positivo para os negócios e começaram a investir massivamente na sua utilização.

Então, confira a seguir alguns *cases* de sucesso apresentados por Vale (2020).

### Disneyland

A Disneyland foi projetada com apenas um único objetivo: **encantar o cliente por meio da melhor experiência da vida dele.**

(objetos/figura\_09.jpg)

Figura 9 – Disneyland

Fonte: UOL (2022)

Quando se visita o parque, não é possível ver os prédios em volta, para reter o foco e a atenção do cliente. Apesar de existirem vários funcionários vestidos de Mickey, o visitante nunca verá dois ao mesmo tempo passeando pelo parque. Além disso, a pulseira digital que dá acesso ao local permite realizar compras e muito mais.

### Lego (blocos de montar)



Em 2004, a Lego quase foi à falência, pois os brinquedos que produzia eram voltados apenas para crianças que logo cresciam e não queriam mais brincar com os blocos. Surgiu aí a necessidade de reinvenção. Os projetistas tradicionais foram substituídos por psicólogos, antropólogos e uma grande equipe diversa e multidisciplinar, os quais passaram a aplicar a UX no desenvolvimento de novos produtos.

Entre esses produtos, estão as linhas temáticas Star Wars, Jurassic Park, Technic, Mindstorms (robótica), Harry Potter, Marvel, DC etc., despertando desejos em crianças, jovens e adultos.

(objetos/figura\_10.png)

Figura 10 – Produtos temáticos da Lego

Fonte: Senac EAD (2022)

## Apple


Uma das primeiras pessoas que perceberam que a experiência poderia mudar a vida das pessoas foi Steve Jobs. Ele acreditava que o computador poderia ser acessível a todas as pessoas. Então juntou seu pensamento inovador e empreendedor com as ideias de Douglas Engelbart (inventor do *mouse*) e, juntos, eles criaram a interface gráfica dos computadores que evoluiu muito no decorrer do tempo e é utilizada até hoje.

(objetos/figura\_11.png)

Figura 11 – Macintosh 1 e produtos atuais da Apple

Fonte: Senac EAD (2022)

## Encerramento

 Com base no conteúdo estudado, você pôde perceber a importância do desenvolvimento de projetos, principalmente os de *software*, orientados à UX. Também conseguiu aprender que, por mais que as interações do usuário se concentrem no *front-end*, o *back-end* deve acompanhar e seguir todas as técnicas e todas as etapas para que o *software* possa estar sempre disponível e funcione de forma rápida e eficaz.