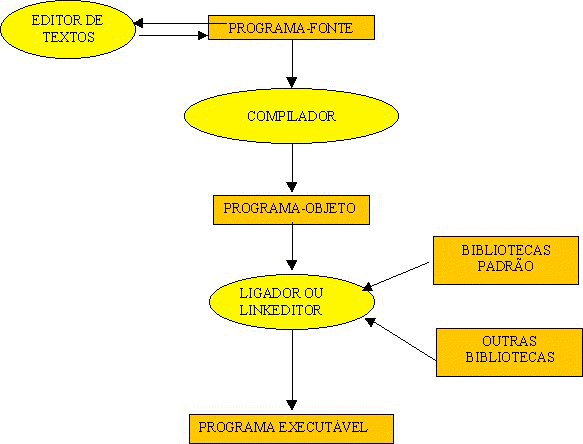
**Conceitos Básicos de Programação**

O processo de programação inicia-se com a edição de um **programa-fonte** e termina com a geração de um **programa-executável**.

Um programa-fonte deve ser criado em um editor de textos (sem formatação). Após a criação, este programa é submetido a um **compilador** que analisa o código-fonte e o converte este para um **código-objeto** (ou **programa-objeto**). O código-objeto é a versão em **linguagem de máquina** do programa-fonte.

Se o programa contiver chamadas a funções das **bibliotecas**   (função cosseno, por exemplo) o **ligador** junta o programa-objeto com a(s) respectiva(s) biblioteca(s) e gera um **código-executável** (ou **programa-executável**).



**Principais Termos:**

* **Código-Fonte:** criado em um editor de textos, contendo os comandos da linguagem de programação (C, Pascal...). Serve como entrada para o compilador.
* **Código-Objeto:**criado pela conversão do código-fonte em linguagem de máquina. É gerado pelo compilador. Só é criado quando não há erros no código-fonte. (Extensão do código-objeto: .OBJ)
* **Ligador ou Linkeditor:**"junta" o código-objeto com as bibliotecas necessárias para gerar o programa-executável. (Extensões das bibliotecas: .DLL ou .LIB)
* **Programa Executável:**código que pode ser executado pelo sistema operacional. (Extensão do programa-executável: .EXE)
* **Tempo de Compilação:**durante o processo de conversão entre código-fonte e código-objeto.
* **Tempo de Execução:**após a ativação do programa executável.

**As 10 linguagens de programação mais populares.**

Apesar de existir uma grande variedade de linguagens de programação, algumas podem ser mais adequadas para certos negócios ou empresas. É preciso cuidado, pois você pode se tornar um mestre em uma linguagem específica, mas, se ela não for muito requisitada no mercado de trabalho, isso significa que poderá ter dificuldades em arranjar um emprego na área.

Sendo assim, se você está buscando alternativas para conseguir se adequar às necessidades do mercado de trabalho, confira neste artigo 10 das linguagens de programação mais utilizadas atualmente. Dessa maneira, você poderá escolher algumas para aprender e conseguir garantir uma boa posição - e um bom salário!

**1. Java**

A linguagem de programação mais solicitada de longe é o Java. No topo da maioria dos índices especializados na medição da popularidade, o Java se caracteriza por ser portável, ou seja, é possível compilar um programa em Java de maneira fácil para todo tipo de aparelho. Vale lembrar também que o Java é a linguagem mais usada para a criação de aplicativos Android.



Outro ponto positivo do Java é a sua escalabilidade, ou seja, a capacidade de adaptar seu programa à medida que ele cresce em número de utilizações, além de sua retrocompatibilidade, já que um código feito em uma versão antiga continua a ser reconhecida pelas versões atuais.

**2. JavaScript**

O JavaScript ainda é amplamente utilizado em aplicações web e tem ganhado espaço no desktop/mobile, sendo bastante usado para criar interatividade. Apesar de ser uma linguagem mais antiga em comparação à maioria das que serão listadas aqui, o JavaScript é bastante requisitado e parte desse sucesso se deve a sua simplicidade.

**3. Python**

Considerada a linguagem de mais fácil aprendizado, a Python continua a ser uma das mais populares no mercado, mesmo que tenha sido lançada há quase 30 anos atrás (em 1989). É um dos códigos de mais fácil leitura e é bastante utilizado para desenvolvimento web e *machine learning*.



**4. C**

Talvez a mais conhecida entre as linguagens de programação - principalmente pelas suas variantes C++ e C# -, a linguagem C também é uma das mais antigas já lançadas. Sua principal vantagem está também na facilidade de *portar*um programa para outro tipo de dispositivo. Vale notar também que a linguagem C, desde cedo, foi adotada por gigantes como Microsoft e Linux, entre outros.

Apesar de antigo, aprender C traz boas vantagens ao desenvolvedor, já que funciona em quase todo tipo de sistema e não exige muito das máquinas. Por conta dessa pouca exigência de performance, a linguagem C é bastante usada para criar softwares para aparelhos pequenos e dispositivos que contam com a Internet das Coisas (IoT).

**5. C++**

Um dos principais motivos pelo qual o C é uma das linguagens mais populares também se dá pela própria popularidade de suas variantes. O C++ é uma versão mais atual do C - embora também já tenha certa idade - e é bastante utilizado no desenvolvimento de softwares mais pesados, como sistemas integrados (CRM), aplicações que promovem interação entre cliente e servidor ou jogos para computador, entre outros.



**6. C#**

Outra variação da linguagem C que é bastante popular no mercado. Porém, anda caindo em desuso em relação a anos anteriores. Assim como o C++, é mais complexo de se aprender que outras linguagens como Python e JavaScript. Por outro lado, ainda é uma linguagem bastante requisitada na área de desenvolvimento de games, se tornando essencial para quem planeja entrar nesse mercado.

**7. Ruby**

Se está a procurar trabalho em uma startup, o Ruby é a linguagem perfeita para conseguir uma vaga na área. Usada na construção de serviços mundialmente reconhecidos como o Airbnb e o Twitter, a linguagem Ruby se caracteriza pela sintaxe de fácil leitura, permitindo que um desenvolvedor escreva menos código para que suas aplicações funcionem.

Através do *framework*web Ruby on Rails, a linguagem permite o lançamento de aplicações web em uma velocidade bem maior que em outras linguagens. O lado negativo do Ruby é que ele é uma linguagem difícil de escalar, ou seja, complicada de manter a medida que sua aplicação cresce em número de usuários, já que ele utiliza bastante processamento para compensar erros no código.



**8. PHP**

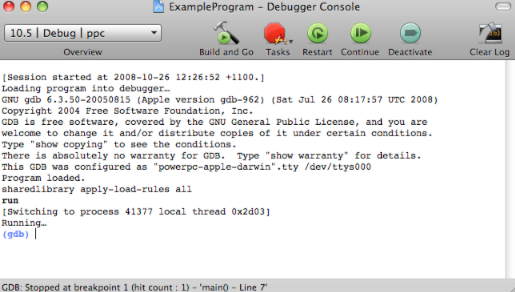
Usado majoritariamente em aplicações web, a linguagem PHP é útil para incluir funções a uma página que o HTML não é capaz de suportar. A linguagem também é utilizada para integração entre informações de sua página e banco de dados MySQL, por exemplo. Sites como o Yahoo e a versão web do Facebook são mantidas em PHP.

**9. R**

Essa linguagem vem ganhando popularidade nos últimos anos, devido à sua utilidade na área de *big data*, que tem recebido mais vagas recentemente e tende a crescer mais nos próximos. Também é considerada uma linguagem fácil de operar e aprender.

**10. Objective-C**

Uma variação do mundialmente conhecido C#, mas específico para construir aplicativos para iPhone e iPad. Ele não chega a ter grandes variações, mas é necessário para quem possui interesse em garantir uma vaga em empresas de desenvolvimento mobile. Além disso, a linguagem é a mais popular para quem constrói aplicativos para os aparelhos da Apple.



<https://www.devmedia.com.br/top-10-linguagens-de-programacao-mais-usadas-no-mercado/39635>

# **Código fonte**

Código fonte vem de um termo em inglês “source code”, sendo um conjunto de palavras ou símbolos escritos de forma ordenada, contendo instruções em linguagens específicas de programação.   
  
Há várias formas de entender as linguagens, algumas são complicadas, outras interpretadas. Mas o importante é que todas sejam entendidas para se transformarem em programas executáveis.  
  
O Código fonte é um conjunto de palavras que formam linhas de comandos onde precisam estar dentro da padronização da linguagem escolhida. Além disso, tem de obedecer aos critérios de execução, caso contrário, a programação não sairia como desejada.   
  
Nos dias atuais, com a diversidade de linguagem, o código pode ser escrito de forma modular, podendo um mesmo conjunto de códigos ser compartilhado por diversos programas ou até mesmo linguagens.  
  
Concluindo, o código fonte são linhas de programação que informam um software em sua forma original. Normalmente o programador inicia o código com a linguagem C++ ou Visual Basic.  
  
Para que o programador execute o software ele precisa converter as linhas programadas em linguagem de máquina, que é a única que o computador consegue entender.  
  
Quando você compra um programa qualquer, na realidade, está comprando a linguagem de máquina e não o código fonte.  
  
A plataforma Windows possui o código fonte fechado. Mas alguns outros programas possuem o código fonte aberto. É o caso do Linux e outros sistemas operacionais que são conhecidos como código aberto.  
  
O código fonte pode alterar a forma como o software funciona, pode adicionar recursos, remover outros, enfim, o programador que está trabalhando no código fonte pode adaptar o software às suas necessidades.

<https://www.portaleducacao.com.br/conteudo/artigos/educacao/codigo-fonte/66618>

**Código Fonte x Executável x Interpretador**

O código fonte é o programa escrito numa linguagem de programação que é facilmente entendida pelos humanos. Essa linguagem o hardware, a CPU do computador, não entende.

Aí tem um software muito esperto chamado "compilador" (ele tem um irmão chamado interpretador). Esse compilador ele entende essa linguagem usada no código fonte. E mais! Ele sabe a linguagem que o hardware entende, que a CPU consegue processar. Então o compilador lê o código fonte e gera um "código executável", que é o programa agora escrito na língua que as CPUs entendem. A linguagem de máquina, como se diz. Tem uns humanos que programam direto nela, não precisam do compilador/interpretador. Esses seres humanos têm problemas (mentira!! São programadores top mesmo, raríssimos, supercompetentes e meio mágicos, são os top nerds).

Então a diferença é essa:

* Código fonte é entendido pelo humano (e pelo compilador) – criado pelo programador
* Código executável é entendido pelo computador (e pelo compilador) – criado pelo compilador

**E o Interpretador?**

Tanto o compilador como o interpretador são tradutores.

Só que o compilador examina o programa todo e traduz o programa inteiro para o código executável.

Já o interpretador é meio que uma tradução simultânea. Ele vai lendo o programa e vai mandando a CPU fazer as coisas. Tipo tradução em tempo real do Oscar. (JVM, Framework)

Cada um tem suas vantagens e desvantagens.

Leia mais em Brainly.com.br - <https://brainly.com.br/tarefa/13087289#readmore>

# **O conceito de Usabilidade**

[](https://www.portaleducacao.com.br/conteudo/artigos/financas/o-conceito-de-usabilidade/52918)

Usabilidade é característica daquilo que é funcional e simples. É tornar óbvio o óbvio, tendo em vista as necessidades do usuário e o contexto em que está inserido.  
  
 **A norma ISO 9241-11, define a usabilidade como**: a efetividade, a eficiência e a satisfação com que usuários específicos atingem objetivos específicos em ambientes particulares.  
  
  
A usabilidade na internet une-se aos estudos da Interação Humano-Computador (IHC) e seu foco principal é a interação dos usuários com websites e sistemas da web. As interfaces devem ser desenvolvidas, de modo a garantir a qualidade e a eficiência de um site.  
  
  
A usabilidade é um estudo que envolve a engenharia de software, desenvolvimento de telas, e estudo de uso da aplicação em si. Este termo indica o quanto o software, site ou uma tela qualquer pode ter facilidade de uso ou não. Saber construir uma tela de modo que ela seja relativamente fácil de usar para qualquer tipo de usuário, experiente ou inexperiente com a Internet, não é algo simples e requer muito estudo.  
  
  
Tornar um site simples, para usuários inexperientes e ao mesmo tempo interessantes para usuários já acostumados com a web, não é tarefa fácil e exige do desenvolvedor cuidados redobrados para conseguir atingir de maneira eficaz todo o público sem restrição.  
  
  
Entre outros comportamentos, vale destacar que os usuários fazem um scan das páginas a procura do que precisam ou de uma indicação rápida do que fazer, assim como nós quando andamos por uma rua olhamos coisas que nos atrai, ignorando grande parte dos outros elementos.  
  
  
Os princípios de usabilidade vêm sendo muito comentados nos dias atuais, no entanto a teoria ainda não se juntou à prática, pois muitos sites da Web ainda não seguem esses princípios, seja por inabilidade técnica ou por simples desinteresse comercial.   
  
  
Porém alguns profissionais já se atentaram para os inúmeros benefícios trazidos por meio da implementação desses princípios. Um desenvolvedor de website que se preocupa com a usabilidade deve estar atento para manter o usuário no caminho desejado, sem confusões e quebras, que poderiam gerar grande frustração, e consequentemente a saída do usuário do site.

# O que é Usabilidade?

A usabilidade é apenas um dos conceitos que abrangem a experiência de uso de pessoas com produtos, tendo também algumas limitações.

por [Talita Pagani](https://tableless.com.br/authors/talita-pagani) 22/08/2011 [15 comentários](https://tableless.com.br/o-que-e-usabilidade/#disqus_thread)~ 5 min. / 949 palavras

Já vimos em um [artigo anterior](https://tableless.com.br/estabelecendo-uma-metodologia-agil-para-avaliacao-de-usabilidade) como a usabilidade beneficia o desenvolvimento web. Mas, o que de fato compreende a usabilidade? Nem tudo que se refere à experiência de uso está dentro do escopo dela. Sim, a usabilidade tem limitações e lacunas que são complementadas por outros paradigmas, como o design de interação. Porém, podemos considera-la a base da pirâmide que sustenta o projeto de uma aplicação com boa experiência de uso.  Neste artigo, veremos brevemente o que faz parte do conceito de usabilidade e qual o papel que ela tem no processo de desenvolvimento de aplicações.

## Afinal, o que é usabilidade?

A Usabilidade é “filha” da Interação Humano-Computador (IHC) e “neta” da Engenharia de Software, carregando esse legado ao longo de sua evolução, podendo ser definida como o **grau de facilidade** com que o usuário consegue interagir com determinada interface.

Partindo da IHC, a usabilidade aborda a forma como o usuário se comunica com a máquina e como a tecnologia responde à interação do usuário, considerando as seguintes habilidades, de acordo com a norma [ISO 9241](https://en.wikipedia.org/wiki/ISO_9241):

* **Facilidade de aprendizado:** a utilização do sistema requer pouco treinamento;
* **Fácil de memorizar:** o usuário deve lembrar como utilizar a interface depois de algum tempo;
* **Maximizar a produtividade:** a interface deve permitir que o usuário realize a tarefa de forma rápida e eficiente;
* **Minimizar a taxa de erros:** caso aconteçam erros, a interface deve avisar o usuário e permitir a correção de modo fácil;
* **Maximizar a satisfação do usuário:** a interface deve dar-lhe confiança e segurança.

Partindo da Engenharia de Software, a usabilidade é englobada dentro da qualidade e visa garantir uma parte da eficiência e eficácia do sistema. A eficiência refere-se a uma interação produtiva entre o usuário e o sistema, permitindo a realização de tarefas com menor esforço sob uma experiência agradável. A eficácia pode ser entendida como a capacidade do sistema e da interface possibilitarem ao usuário a completude da tarefa e o alcance de seus objetivos no sistema.

## A importância da usabilidade no desenvolvimento de projetos

A usabilidade se encaixa em qualquer tipo de projeto de interface, tendo amplitude diferente de acordo com a criticidade do projeto, ou seja, quanto mais crítico for o sistema, maiores serão as perdas caso ele não seja de fácil utilização e proporcione satisfação. Ela deve ser pensada desde o planejamento do projeto, até a etapa de desenvolvimento e teste.

As empresas têm consciência da importância da usabilidade. Porém, muitas ainda a veem como um fator que só deve ser levado em conta durante o desenvolvimento se houver tempo e recurso, como se ela representasse um custo adicional, fora do que é essencial. Mas as empresas têm muito mais a perder ao minimizar a usabilidade dessa forma. De acordo com Cybis, Betiol e Faust (2007):

“Dependendo da frequência com que o software é empregado, os prejuízos para as empresas podem também ser expressivos, não só em decorrência do absenteísmo e da rotatividade do pessoal, mas também pela baixa produtividade, competitividade e menor retorno de investimento. Sistemas difíceis de usar implicam em erros e perda de tempo, fatores que se multiplicam com a frequência das tarefas e o número de usuários. A perda de dados e informações pode implicar na perda de clientes e de oportunidades. Acontecimentos deste tipo causam desde uma resistência ao uso do sistema até a sua subutilização e abandono completo, com o devido consentimento da empresa. O barato terá custado caro.”

A usabilidade pode ser abordada de diferentes formas ao longo do projeto:

[](https://raw.githubusercontent.com/diegoeis/tableless-static-images/master/2011/07/Slide1.png)

Técnicas utilizadas para garantir e verificar a usabilidade em diferentes etapas

## Lacunas a serem preenchidas

Apesar de a usabilidade ser responsável por assegurar grande parte dos quesitos de experiência do usuário, ela possui duas grandes limitações:

1. Embora considere a satisfação do usuário, trata apenas dessa satisfação perante a realização de uma determinada tarefa, concentrando-se na função, sem considerar fatores físicos, ambientais e emocionais inerentes ao contexto de utilização do sistema. Mais do que ser de fácil utilização, aprendizagem e permitir completude da tarefa, uma boa experiência de uso está baseada em uma interação agradável, considerando a forma como as pessoas percebem a interação com o sistema;
2. O contexto usualmente analisado pela usabilidade é mais restrito: abrange apenas a visão do usuário. Para garantir uma boa experiência de uso, é necessário abranger a visão de diferentes stakeholders: do usuário, da organização e da equipe de desenvolvimento, cada qual com requisitos e necessidades diferentes frente ao sistema