

SOFTWARE

PROFº JANSSEN LEITE.

SOFTWARE

O que significa, **para você**, o termo **Sofwtare**?



SOFTWARE



Resposta

É um conjunto de **instruções, dados ou programas** usados para operar **computadores e executar tarefas específicas**.

Ou seja, a **parte Lógica** do computador.

SOFTWARE

Já ouviu falar **sobre algum desses termos?**

Números Binários Software Sistemas
Operacionais Bytes codificação
Processamento Compilação Terabytes Bits
Gigabytes Apps Kilobytes Windows
Android MacOs GoogleOs Linux



**Em uma escala de 0 a 10. Qual nota
você se daria em conhecimentos de
Software?**

SOFTWARE

Tipos de Software

Basicamente a informática se divide em **três tipos principais de software**, cada um com suas **funções específicas**.

ENTENDA DE UMA VEZ POR TODAS

O SISTEMA BINÁRIO

SOFTWARE

Pergunta

O que é um sistema Binário?

Resposta

Sistema que utiliza 0 e 1 para representar informações!

SOFTWARE

Pergunta

Como é chamada a menor unidade de informação na computação?

Resposta

Bit.

SOFTWARE

Pergunta

Como é chamado a unidade básica de armazenamento de dados na computação, composta por um conjunto de 8 bits?

Resposta

Byte.

SOFTWARE

Existem diferentes formas de **software**, desde simples scripts que **automatizam tarefas repetitivas** até **complexos sistemas operacionais** que gerenciam todos os recursos de um **computador**.



SOFTWARE



Tipos de Software



SOFTWARE

Tipos de Software

Software de Sistema

O software de sistema é projetado para **gerenciar e controlar o hardware do computador** que inclui:

- ▶ Sistemas Operacionais;
- ▶ Drivers;
- ▶ Firmware;



SOFTWARE

Tipos de Software

Software de Sistema

► Sistemas Operacionais:

Como **MacOS**, **Linux**, **Windows**, **iOS**,
Android, BSB... que atuam como uma
interface entre o **usuário** e o **hardware**.
Eles **gerenciam** recursos como **memória**,
processador, e **dispositivos de entrada e saída**.



SOFTWARE

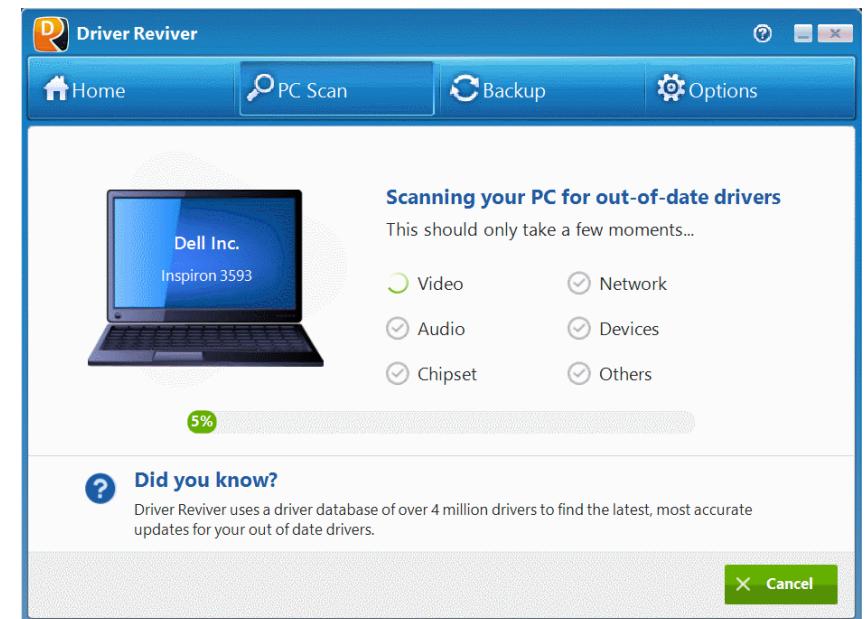
Tipos de Software

Software de Sistema

► Drivers:

Que permitem que o **sistema operacional se comunique com o hardware**.

Por exemplo, **drivers de impressora** permitem que diferentes aplicativos enviem documentos para serem impressos.



SOFTWARE

Tipos de Software

Software de Sistema

► Firmware:

Um **tipo específico de software** que está **embutido em hardware** para controlar dispositivos específicos. Um exemplo clássico é o **BIOS** (Sistema Básico de Entrada/Saída), que **inicializa e testa os componentes de hardware** durante a **inicialização do computador**.



SOFTWARE

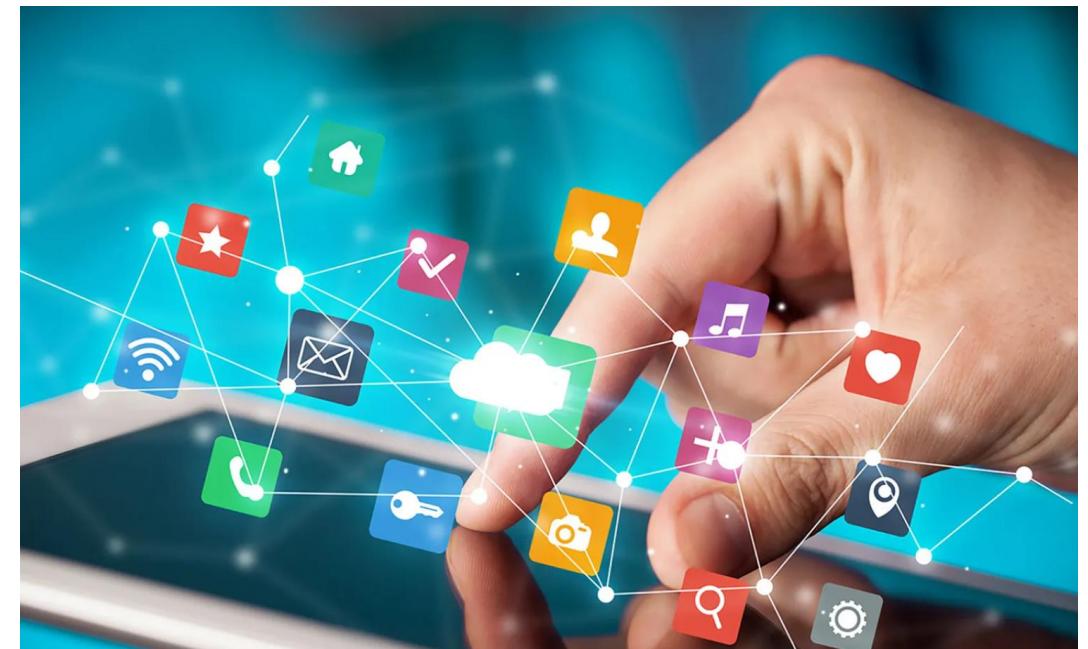
Tipos de Software

Software de Aplicação

O **software de aplicação** permite que os usuários realizem **tarefas específicas**.

Exemplos incluem:

- ▶ Software de Produtividade;
- ▶ Aplicativos de Comunicação;
- ▶ Jogos Eletrônicos;
- ▶ Software de Designer Gráfico;



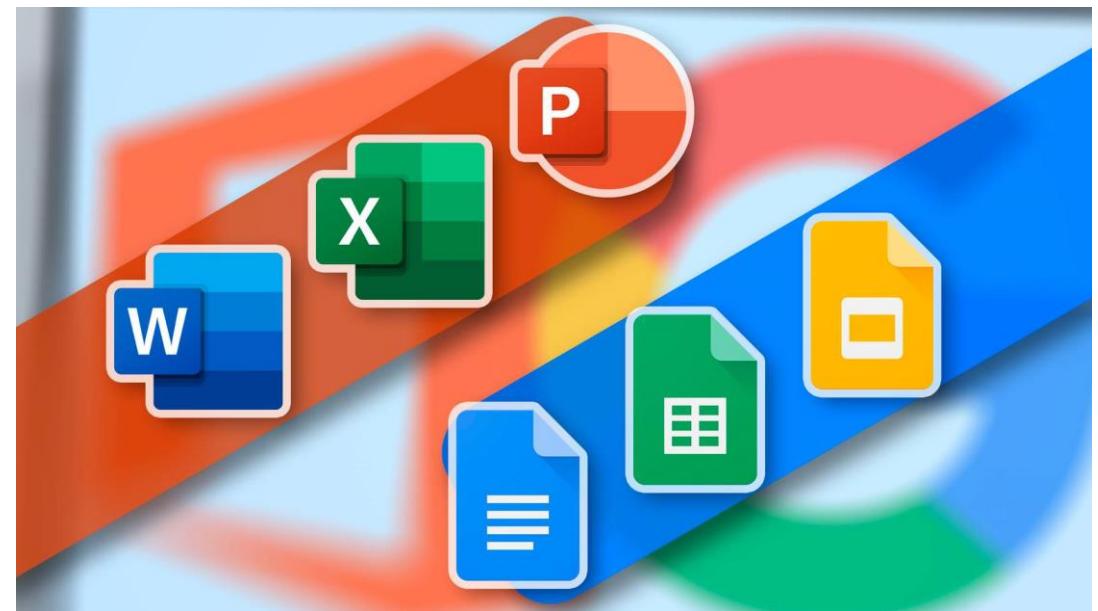
SOFTWARE

Tipos de Software

Software de Aplicação

► Software de Produtividade:

A exemplo o **Microsoft Office**, **Google Docs**, que facilitam a **criação e edição de documentos, planilhas e apresentações**.



SOFTWARE

Tipos de Software

Software de Aplicação

► Aplicativos de Comunicação:

A exemplo o WhatsApp, Zoom, que possibilitam **a troca de mensagens, chamadas de vídeo e conferências online**, essencial em tempos de **trabalho remoto e educação a distância**.



SOFTWARE

Tipos de Software

Software de Aplicação

► Jogos Eletrônicos

A exemplo o **Fortnite**, **Minecraft**, que proporcionam entretenimento e **interação social entre jogadores**. Os jogos eletrônicos também estão se tornando **plataformas de aprendizado**, onde jogadores **desenvolvem habilidades cognitivas** e de **resolução de problemas**.



SOFTWARE

Tipos de Software

Software de Aplicação

► Software de Designer Gráfico

A exemplo o **Adobe Photoshop**, **Illustrator**, **CorelDRAW**, que são ferramentas indispensáveis para designers e artistas. Eles permitem a **criação de gráficos**, ilustrações e manipulação de **imagens em alta qualidade**.



SOFTWARE

Tipos de Software

Software de Programação

Este tipo de software fornece ferramentas aos programadores para desenvolver outros softwares.

Que Inclui:

- ▶ Editores de Código;
- ▶ Compiladores;
- ▶ Ambientes de Desenvolvimento Integrado (IDE);



SOFTWARE

Tipos de Software

Software de Programação

► Editores de Código

A exemplo o **Visual Studio Code**, **Sublime Text**, que oferecem funcionalidades avançadas como destaque de **sintaxe**, **autocompletar e depuração**. Eles são essenciais para o **desenvolvimento eficiente de software**.



SOFTWARE

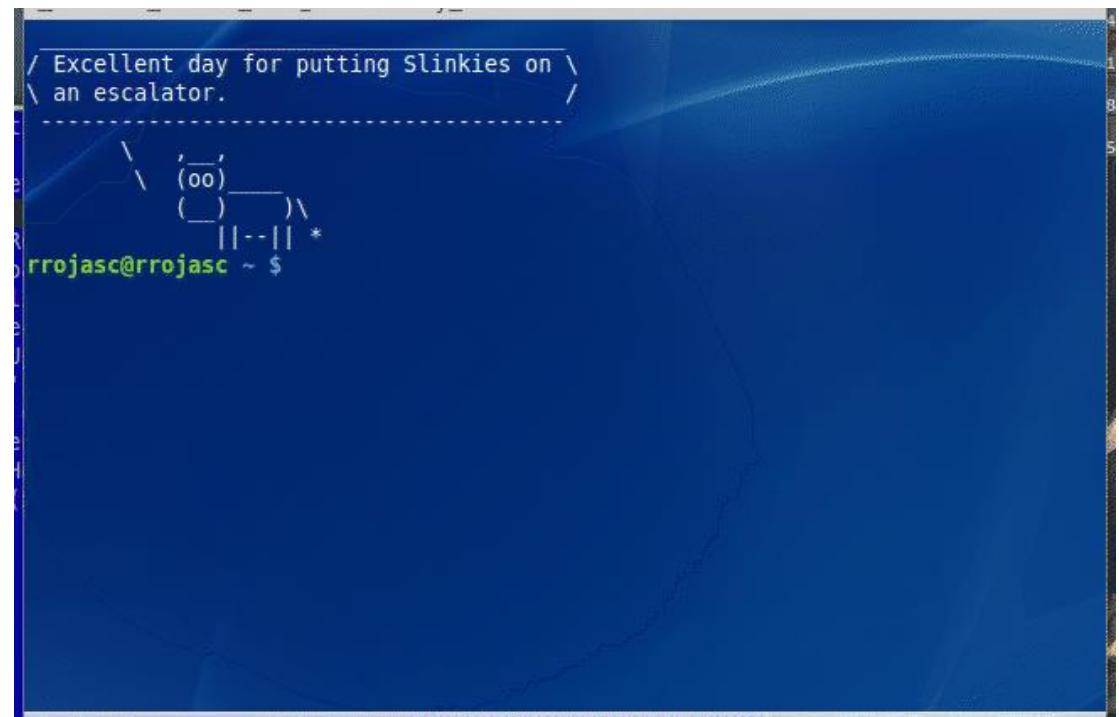
Tipos de Software

Software de Programação

► Compiladores

Traduzem o **código-fonte em código executável**.

Por exemplo, o GCC (GNU Compiler Collection) é **amplamente utilizado para compilar programas escritos em C e C++.**



SOFTWARE

Tipos de Software

Software de Programação

- Ambientes de Desenvolvimento Integrado (IDE)

A exemplo o **Eclipse**, **IntelliJ IDEA**, que fornecem um conjunto de ferramentas abrangente para **desenvolvimento de software**, incluindo **editores de código**, **depuradores** e **sistemas de controle de versão integrados**.



The image shows a screenshot of a software development environment. On the left, there's a sidebar with various project files and settings. The main area is a code editor with dark-themed syntax highlighting. The code is written in Ruby and RSpec, specifically for a Rails application, as indicated by the file path and the code itself. A blue button labeled "desenvolvimento" is visible in the top right corner of the code editor area.

```
6 require 'spec_helper'
7 require 'rspec/rails'
8
9 require 'capybara/rspec'
10 require 'capybara/rails'
11
12 Capybara.javascript_driver = :webkit
13 Category.delete_all; Category.create!
14 Shoulda::Matchers.configure do |config|
15   config.integrate do |with|
16     with.test_framework :rspec
17     with.library :rails
18   end
19 end
20
21 # Add additional requires below this line if you have external dependencies
22
23 # Requires supporting files within the same directory as this file if you don't
24 # want them to be automatically required.
25 # run as spec files by default. You can also pass an array of file names
26 # in .spec.rb will both be required.
27 # end twice. It is recommended that you do not name files test*.rb as they
# will be automatically required and cause conflicts. Instead, name them
# feature*.rb or integration*.rb
```

SOFTWARE



Componentes do Software

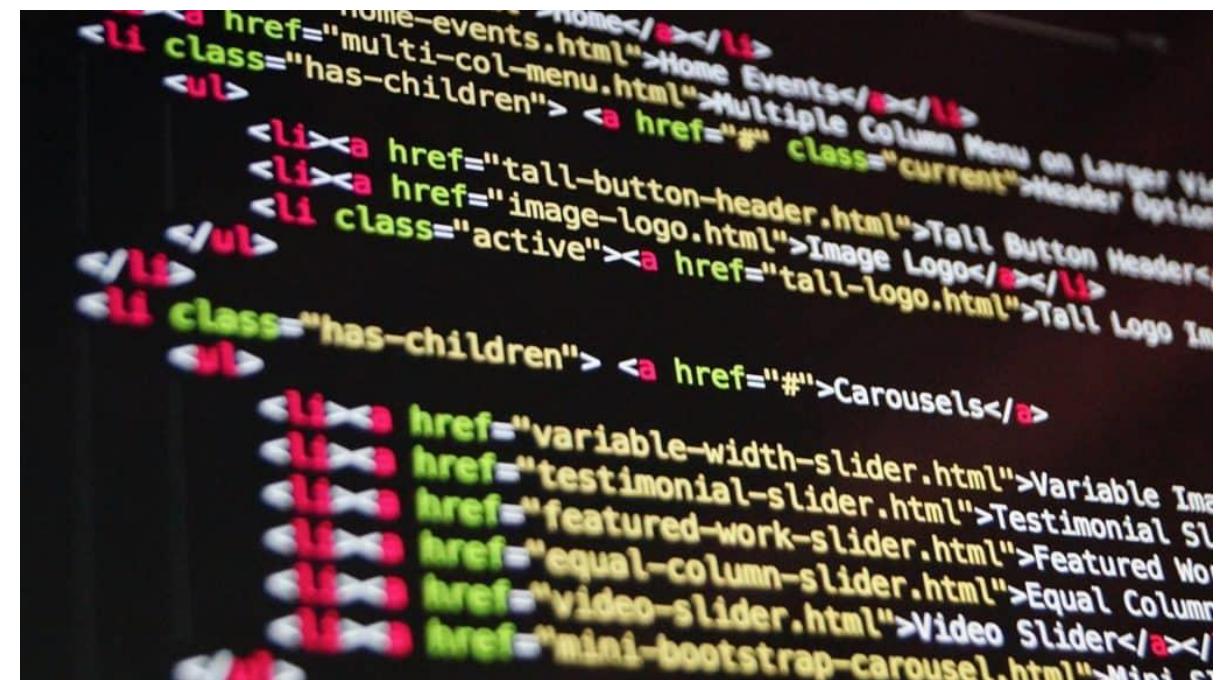


SOFTWARE

Componentes de Software

Código Fonte

O código fonte é o **conjunto de instruções** escritas por programadores em uma **linguagem de programação** como Java, C++, Python, etc.



```
<li><a href="#">Home</a></li>
<li><a href="multi-col-menu.html">Home Events</a></li>
<li class="has-children"> <a href="#" class="current">Header Option</a>
    <ul>
        <li><a href="tall-button-header.html">Tall Button Headers</a>
        <li><a href="image-logo.html">Image Logo</a></li>
    </ul>
<li class="has-children"> <a href="#">Carousels</a>
    <ul>
        <li><a href="variable-width-slider.html">Variable Width Sliders</a>
        <li><a href="testimonial-slider.html">Testimonial Sliders</a>
        <li><a href="featured-work-slider.html">Featured Work Sliders</a>
        <li><a href="equal-column-slider.html">Equal Column Sliders</a>
        <li><a href="video-slider.html">Video Slider</a></li>
        <li><a href="mini-bootstrap-carousel.html">Mini Carousels</a></li>
    </ul>
</li>
```

SOFTWARE

Componentes de Software



Código Fonte

Esse código é posteriormente **compilado ou interpretado** para ser **executado pelo computador**. A **qualidade** e a clareza do código fonte são **vitais para a manutenção e evolução do software**.

A screenshot of the Visual Studio Code interface. The left sidebar shows a file tree with a folder named 'ALURA' containing files like 'forca.h', 'palavras.txt', 'teste.c', 'teste.exe', 'teste2.c', and 'teste2.exe'. The main editor area displays a C program named 'teste.c' with the following code:

```
C > C teste.c > main()
1 #include <stdio.h>
2
3 int main(){
4     printf("Olá, mundo!");
5 }
```

The status bar at the bottom right indicates the code is in Python 3.9.6 64-bit mode, with line 5, column 2, and other details like spaces: 4, CRLF: C, Win32.

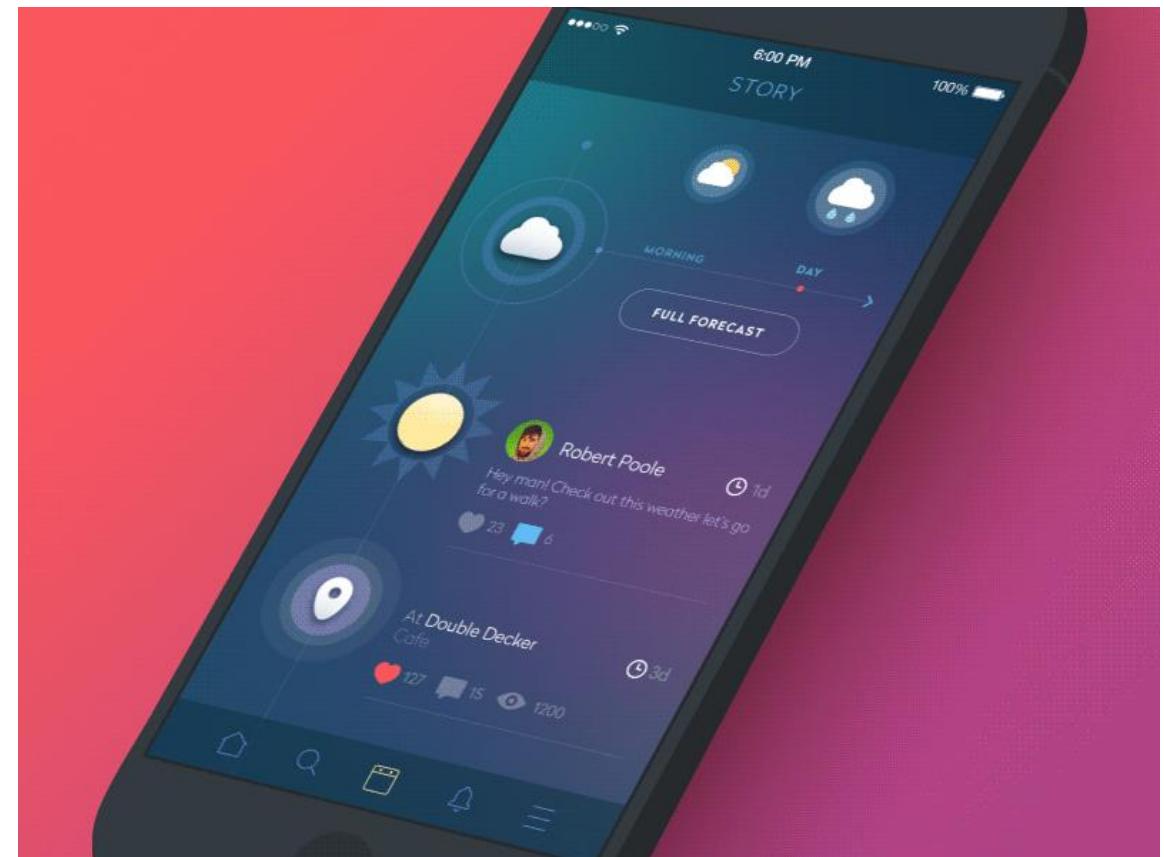
Pressionando F6

SOFTWARE

Componentes de Software

Interface do usuário (UI – User Interface)

A interface do usuário é a parte do **software** com a qual os **usuários** interagem.



SOFTWARE

Componentes de Software

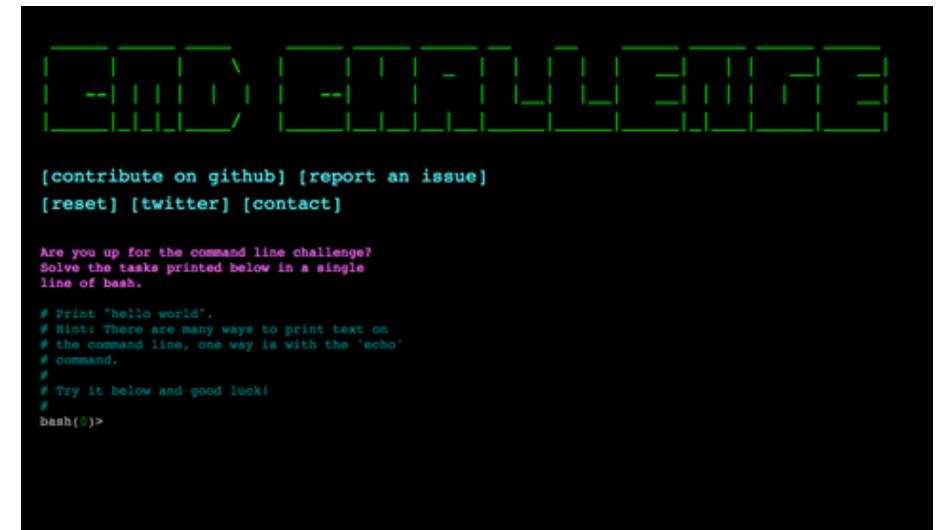
Interface do usuário (UI)

Pode ser:

Interface Gráfica (**GUI** – Graphic User Interface) Linha de Comando (CLI)



Linha de Comando (**CLI** – Command Line Interface)

A screenshot of a terminal window with a black background and white text. At the top, there are two large, faint green rectangular shapes. Below them, the text '[contribute on github] [report an issue]' and '[reset] [twitter] [contact]' is displayed in green. In the center, there is a message in white text: 'Are you up for the command line challenge? Solve the tasks printed below in a single line of bash.' Below this, several command-line hints and a prompt are shown in white text:

```
# Print "hello world".
# Hint: There are many ways to print text on
# the command line, one way is with the 'echo'
# command.
#
# Try it below and good luck!
#
bash(0)>
```

SOFTWARE

Componentes de Software

Data Base – Banco de Dados.

Muitos **softwares** utilizam **bancos de dados** para **armazenar**, gerenciar e recuperar dados de forma eficiente.



SOFTWARE

Componentes de Software

Data Base – Banco de Dados.

Dentre esses incluem **MySQL**, **PostgreSQL**, **MongoDB**.

Um **banco de dados** bem projetado é essencial para o **desempenho** e **escalabilidade do software**.

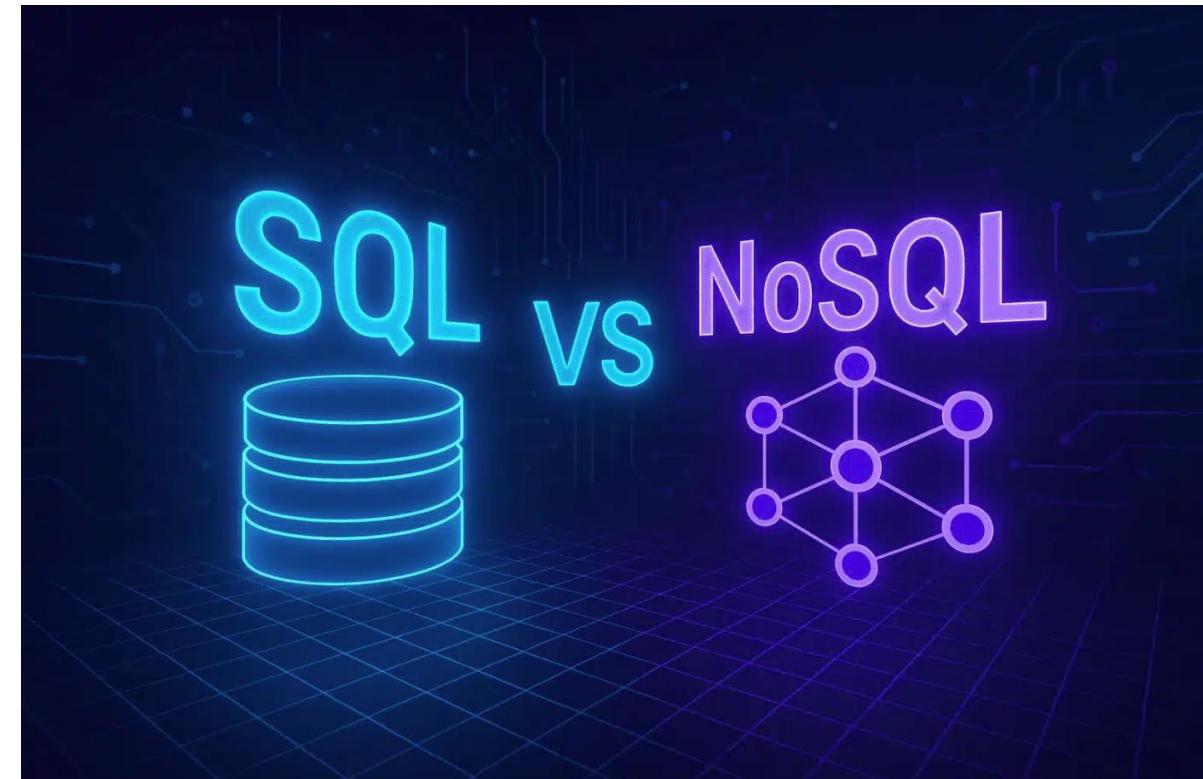


SOFTWARE

Componentes de Software

Data Base – Banco de Dados.

Relacionais e
Não-Relacionais
(NoSQL).



SOFTWARE

Componentes de Software

Data Base – Banco de Dados.

Banco de Dados - Relacional.

É um tipo de banco de dados que **organiza informações estruturadas em tabelas** (chamadas de relações) compostas por linhas e colunas.

The screenshot shows the phpMyAdmin interface for a MySQL database named 'tutorial_meusite'. The left sidebar lists various tables under the schema 'tutorial_meusite'. The right panel displays the 'jos_components' table. The table has the following columns: id, name, link, menuid, parent, admin_menu_link, admin_menu_alt, and option. The data in the table is as follows:

	1	Banners		0	0	Banner Management	com_banne
	2	Banners		0	1	Active Banners	com_banne
	3	Clients		0	1	Manage Clients	com_banne
	4	Web Links	option=com_weblinks	0	0	Manage Weblinks	com_weblin
	5	Links		0	4	View existing weblinks	com_weblin
	6	Categories		0	4	Manage weblink categories	com_weblink_categorie
	7	Contacts	option=com_contact	0	0	Edit contact details	com_contact
	8	Contacts		0	7	Edit contact details	com_contact
	9	Categories		0	7	Manage contact categories	com_contact_categories
	10	Polls	option=com_poll	0	0	Manage Polls	com_poll
	11	News Feeds	option=com_newsfeeds	0	0	News Feeds Management	com_newsf
	12	Feeds		0	11	Manage News Feeds	com_newsf
	13	Categories		0	11	Manage Categories	com_newscat

SOFTWARE

Componentes de Software

Data Base – Banco de Dados.

Banco de Dados – Não-Relacional(NoSQL).

É um tipo de banco de dados projetado para alta **flexibilidade, escalabilidade horizontal** e **alto desempenho**

```
{  
  "id": 55,  
  "Pais": "Brasil",  
  "Regiao": "América do Sul",  
  "Populacao": 201032714,  
  "PrincipaisCidades": [  
    {  
      "NomeCidade": "São Paulo",  
      "Populacao": 1182876,  
    },  
    {  
      "NomeCidade": "Rio de Janeiro",  
      "Populacao": 6323037,  
    }  
]  
}
```

SOFTWARE

Componentes de Software

Data Base – Banco de Dados.

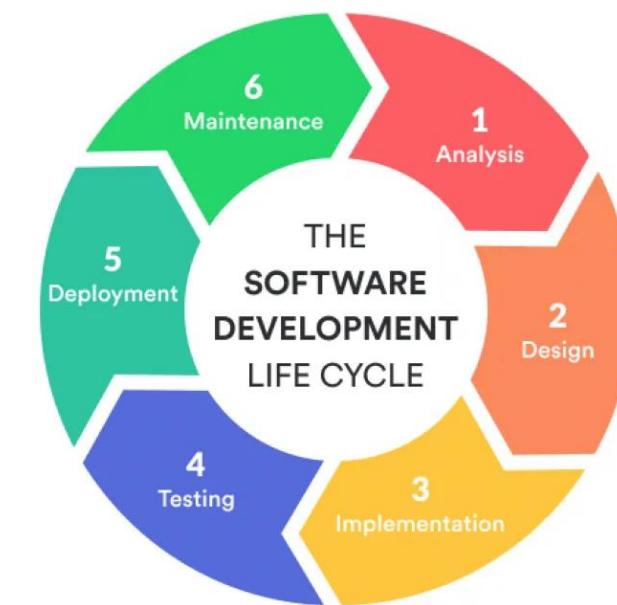
Relacional.

Bancos de dados relacionais como MySQL são amplamente utilizados em **aplicações web**.

Não-Relacional (NoSQL).

Bancos de dados NoSQL como MongoDB são preferidos para grandes volumes de dados não estruturados.

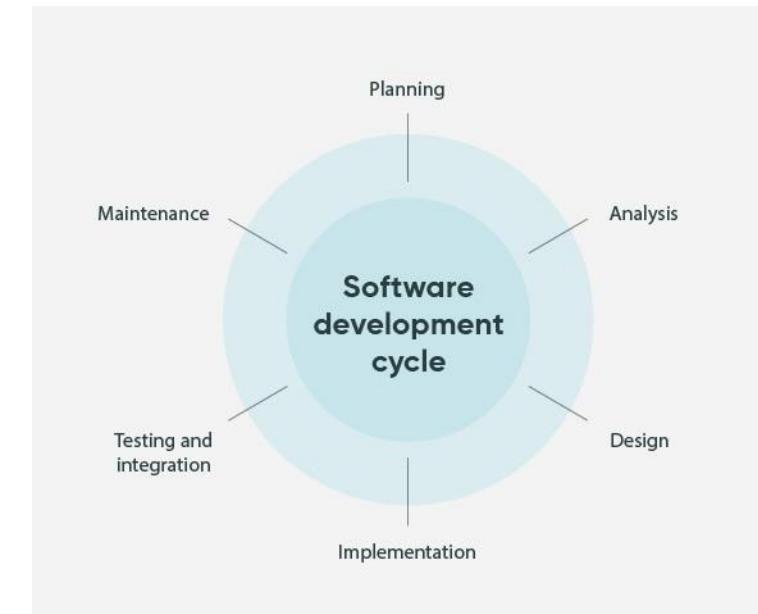
Ciclo de Vida do Desenvolvimento de Software (SDLC)



SOFTWARE

Ciclo de Vida do Desenvolvimento de Software (SDLC)

O **SDLC** é um processo estruturado **utilizado para desenvolver software de alta qualidade.**



SOFTWARE

Ciclo de Vida do Desenvolvimento de Software (SDLC)

1. Planejamento:

Identificação de requisitos e definição do escopo do projeto. Nesta fase, **stakeholders** discutem as **necessidades e expectativas para o software**.



SOFTWARE

Ciclo de Vida do Desenvolvimento de Software (SDLC)

2. Análise de Requisitos:

Detalhamento dos **requisitos funcionais** e **não funcionais**. Os requisitos são documentados e analisados para garantir que **todos os aspectos necessários sejam cobertos**.

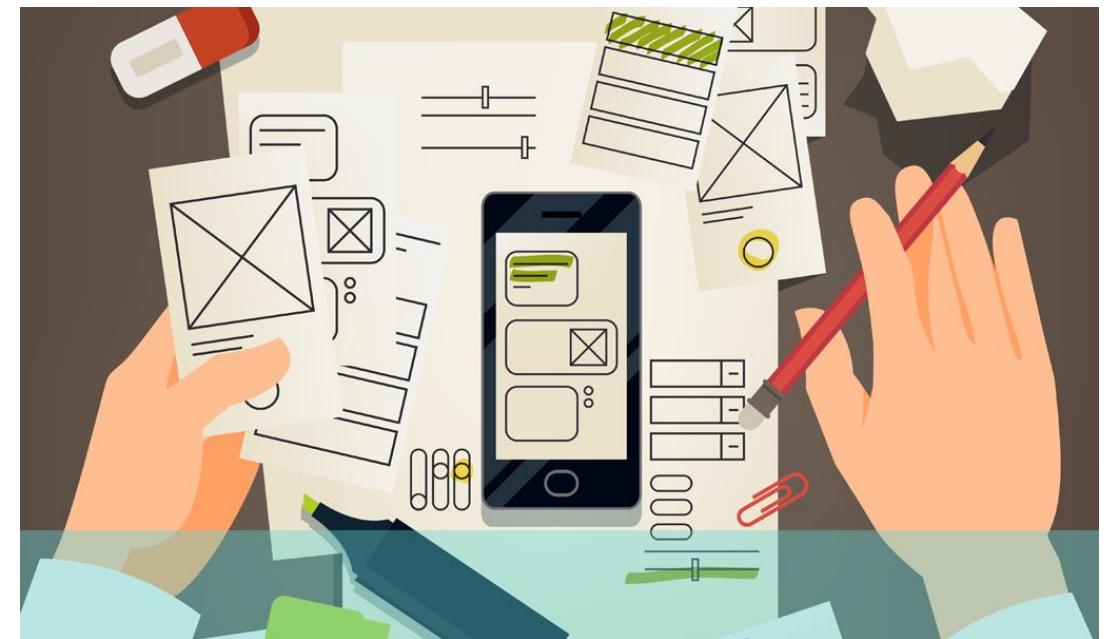


SOFTWARE

Ciclo de Vida do Desenvolvimento de Software (SDLC)

3. Design:

Criação da arquitetura do software. Nesta fase, os desenvolvedores **criam diagramas de fluxo, esquemas de banco de dados e outros documentos de design.**



SOFTWARE

Ciclo de Vida do Desenvolvimento de Software (SDLC)

4. Implementação:

Codificação e desenvolvimento do software. Os programadores **escrevem o código-fonte e desenvolvem as funcionalidades** conforme especificado.

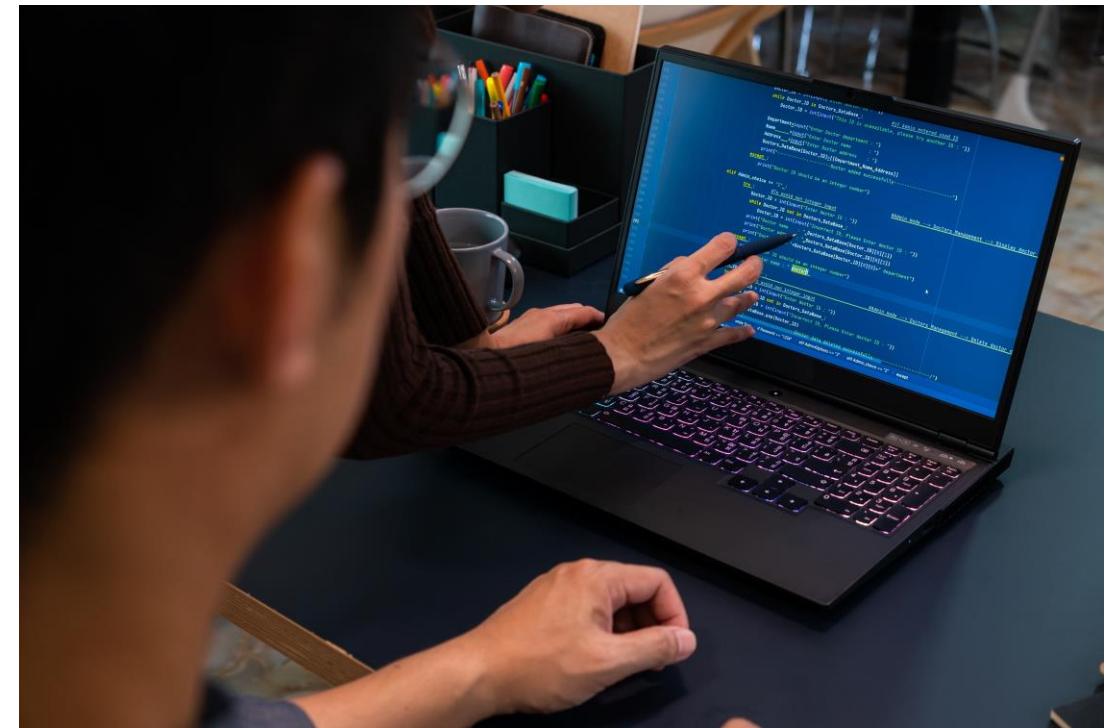


SOFTWARE

Ciclo de Vida do Desenvolvimento de Software (SDLC)

5. Teste:

Verificação e validação do software para garantir que ele funcione conforme esperado. **Testes de unidade, integração e sistema** são realizados para **identificar e corrigir bugs**.

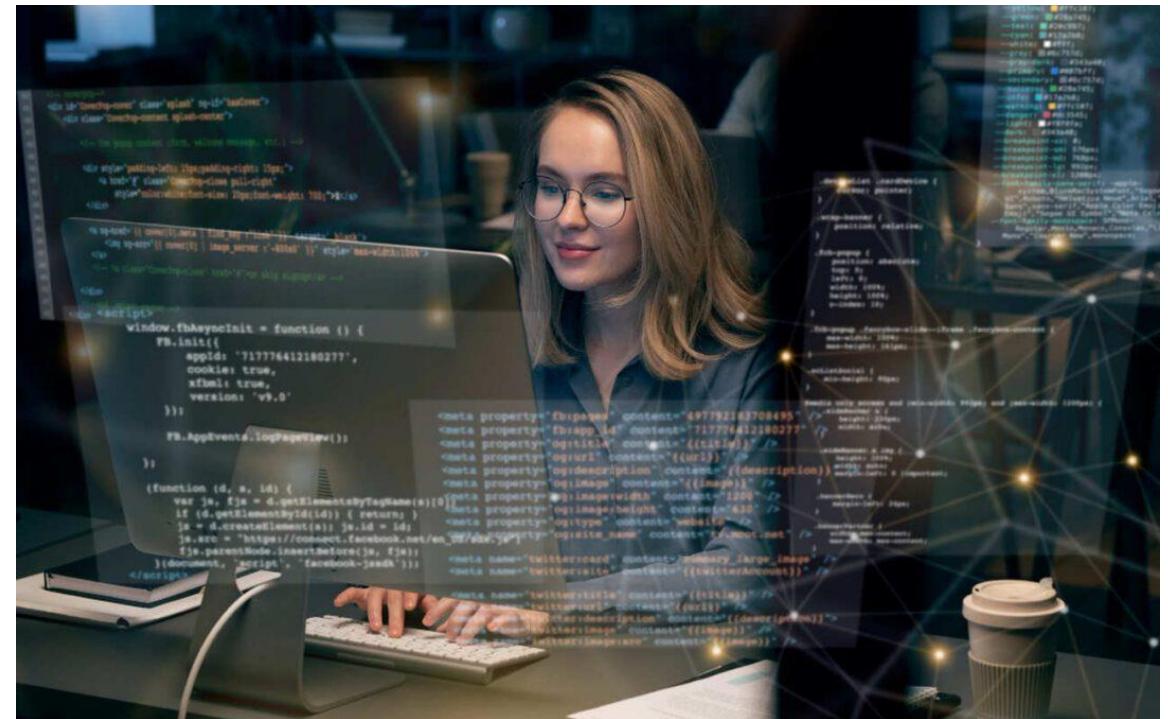


SOFTWARE

Ciclo de Vida do Desenvolvimento de Software (SDLC)

6. Implantação:

Lançamento do software para os usuários finais. O software é **instalado em servidores de produção e disponibilizado para os usuários**.

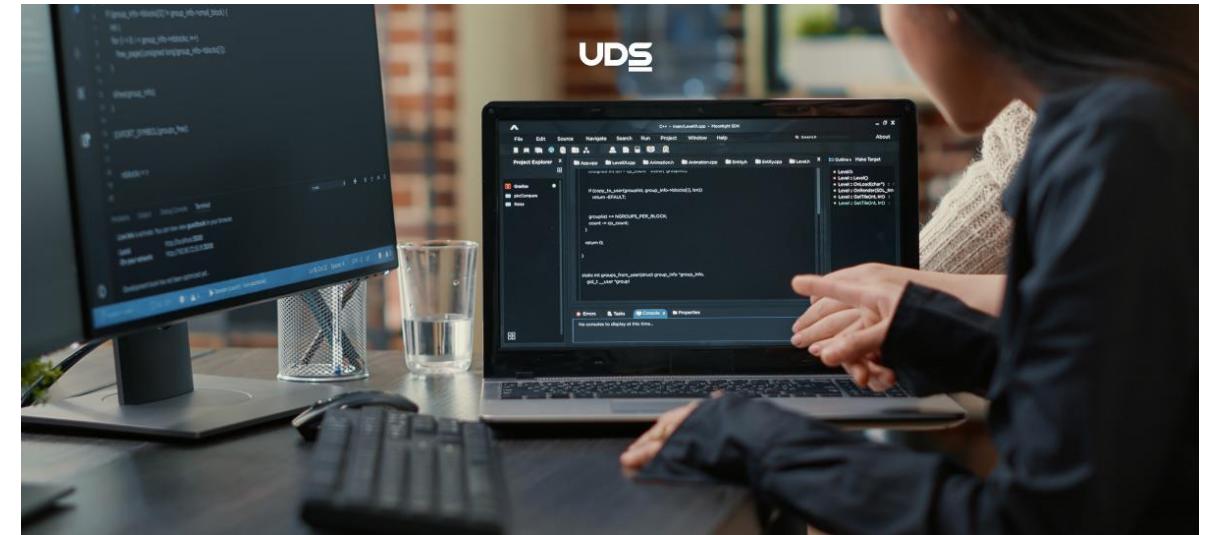


SOFTWARE

Ciclo de Vida do Desenvolvimento de Software (SDLC)

7. Manutenção:

Correção de bugs e atualizações contínuas. O software é **monitorado para desempenho e atualizações** são feitas conforme necessário para **adicionar novas funcionalidades ou melhorar a segurança**.



SOFTWARE

Ciclo de Vida do Desenvolvimento de Software (SDLC)

Mapa Mental SDLC



SOFTWARE



Metodologias de Desenvolvimento



SOFTWARE

Metodologia de Desenvolvimento

Existem várias **metodologias** para gerenciar o processo de **desenvolvimento de software**.

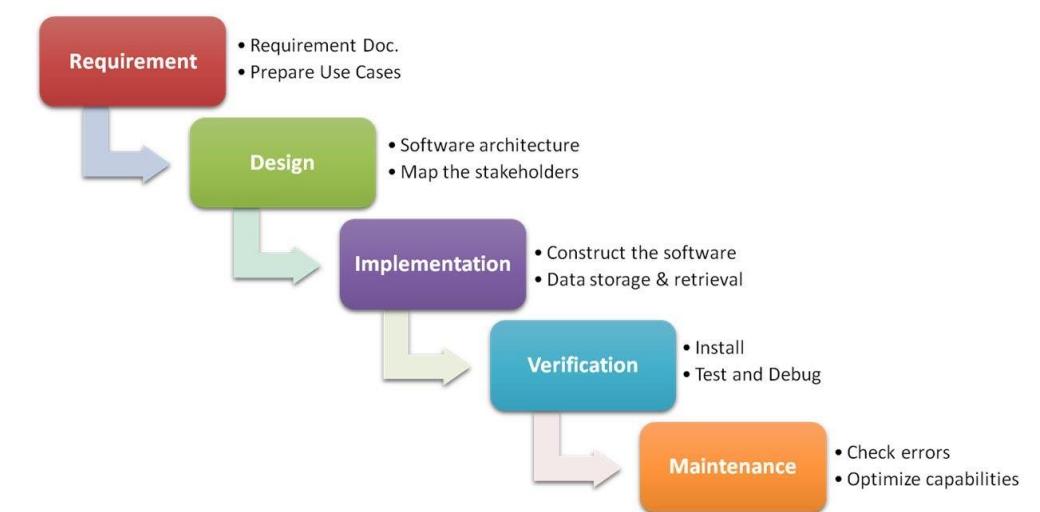


SOFTWARE

Metodologia de Desenvolvimento

Waterfall

Um **modelo linear e sequencial** onde **cada fase deve ser completada antes de passar para a próxima**. É fácil de entender e gerenciar, **mas pode ser inflexível** para mudanças.

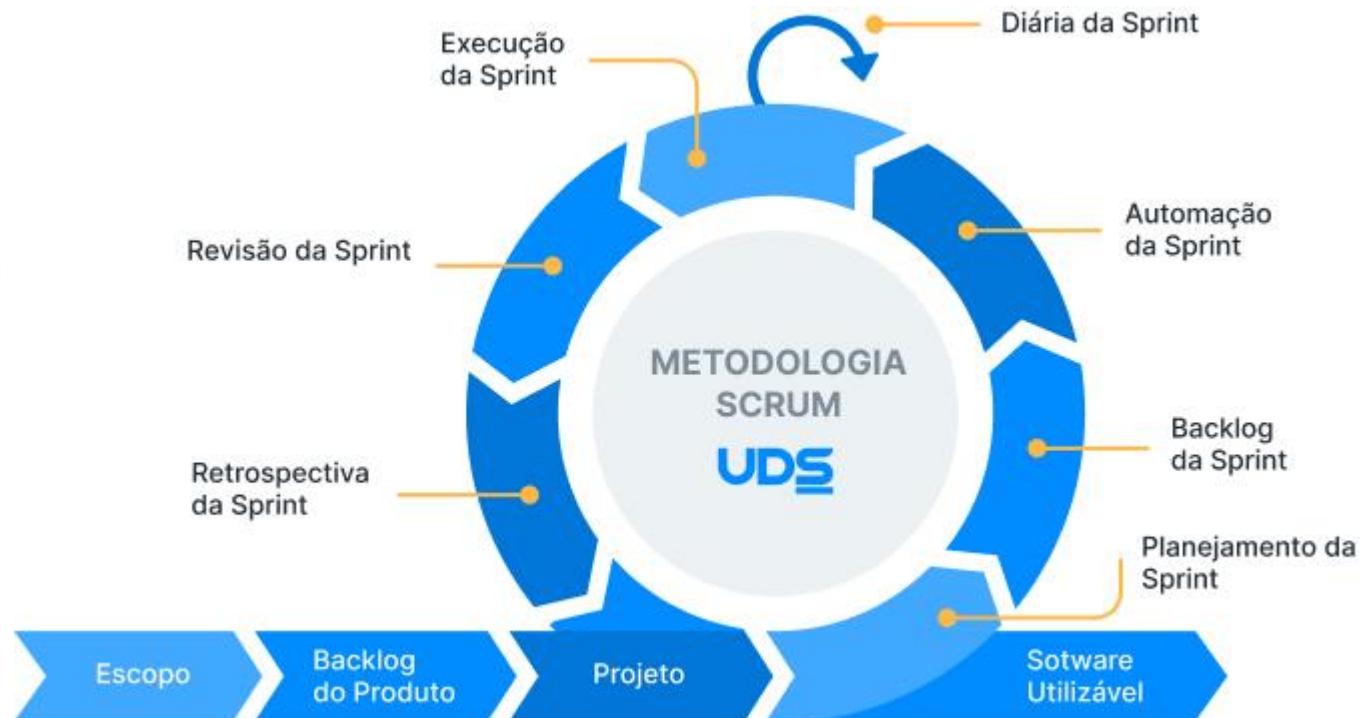


SOFTWARE

Metodologia de Desenvolvimento

Agile

Um modelo **iterativo e incremental** que promove **flexibilidade e resposta rápida a mudanças**. Equipes Agile trabalham em **sprints curtos**, entregando incrementos funcionais do software em ciclos rápidos.

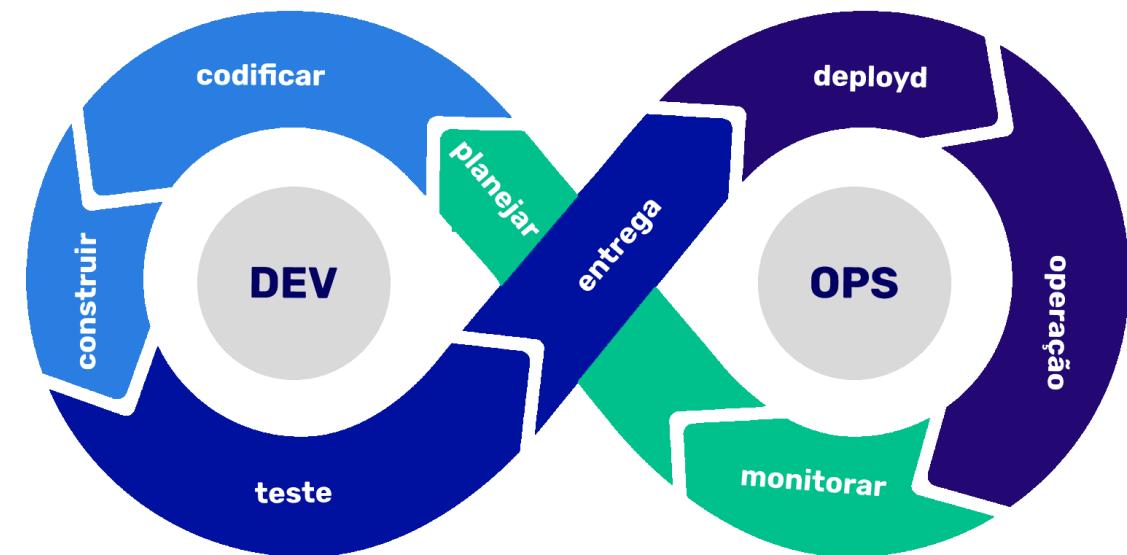


SOFTWARE

Metodologia de Desenvolvimento

DevOps

Uma prática que **combina** o **desenvolvimento de software (Dev)** e as **operações de TI (Ops)** para melhorar a eficiência e a qualidade. DevOps enfatiza a automação e a integração contínua, reduzindo o tempo de entrega de novas funcionalidades.



SOFTWARE



Desafios do Desenvolvimento de Software

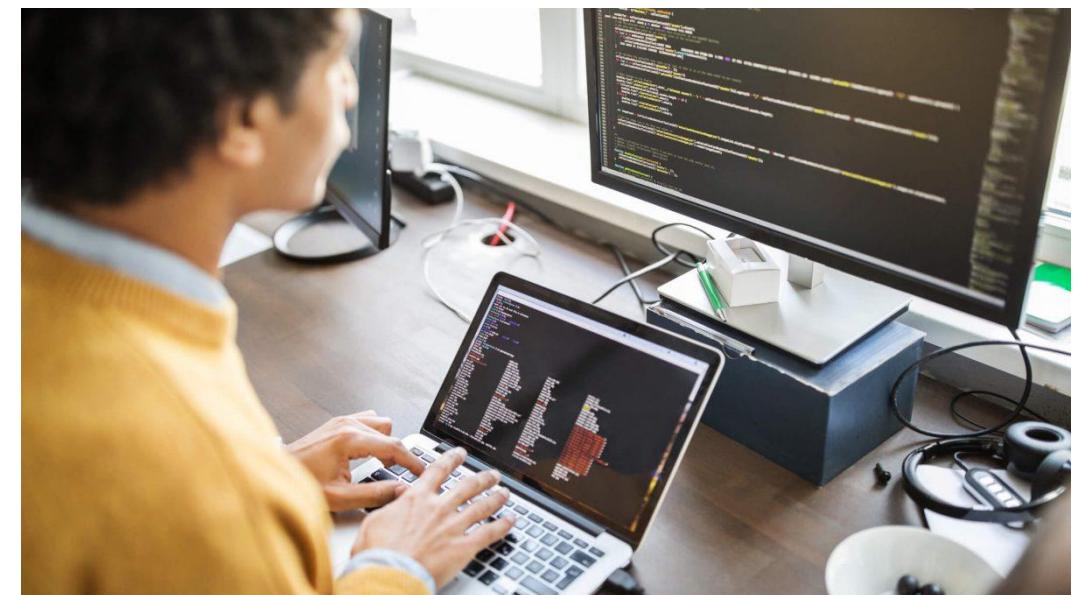


SOFTWARE

Desafio do Desenvolvimento de Software

Desenvolver um software é uma tarefa complexa que enfrenta inúmeros desafios, onde cada **etapa** do processo **apresenta obstáculos únicos** que exigem **soluções inovadoras e eficientes**.

Listamos algumas dessas dificuldades:



SOFTWARE

Desafio do Desenvolvimento de Software

Segurança

Garantir a **segurança** do software contra ameaças de **malware, ataques cibernéticos e violações de dados**, é uma preocupação constante.

Desenvolvedores precisam **implementar** medidas como **criptografia de dados** e autenticação multifator, para proteger informações sensíveis.



SOFTWARE

Desafio do Desenvolvimento de Software

Compatibilidade

Garantir que o software seja **compatível** com diferentes **sistemas operacionais**, dispositivos e ambientes é um desafio constante. **Testes de compatibilidade** são essenciais para **assegurar que o software funcione corretamente em várias plataformas**.



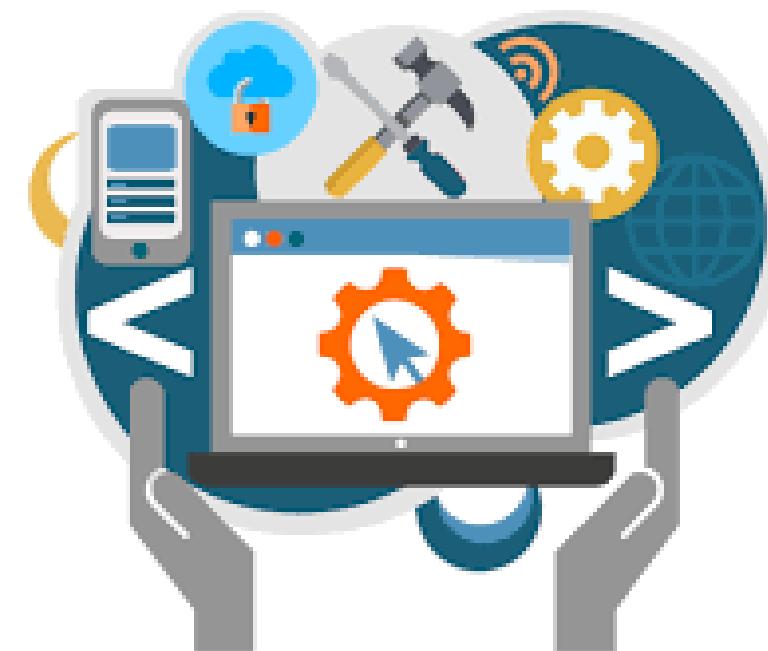
SOFTWARE

Desafio do Desenvolvimento de Software

Manutenção e Atualização

Manter o **software atualizado** e livre de bugs requer um esforço contínuo, especialmente em um ambiente tecnológico em constante evolução.

Empresas precisam investir em equipes de suporte e desenvolvimento para **garantir que o software continue a atender às necessidades dos seus usuários**.



SOFTWARE



Tendências para o Futuro de Software



Tendências para o Futuro de Software

O futuro do software promete avanços revolucionários, com **tecnologias emergentes** que abrirão novas fronteiras e possibilitarão aplicações mais inteligentes e integradas em todos os setores.



SOFTWARE

Tendências para o Futuro de Software

Cultura as a Service (SaaS e HaaS)

A tendência de **Software as a Service (SaaS)** e **Hardware as a Service (HaaS)** está moldando o acesso à tecnologia. Essa solução oferece equipamentos e programas por meio de assinaturas, eliminando a necessidade de instalações locais e simplificando a manutenção e atualizações.



Hardware as a Service (HaaS)

SOFTWARE

Tendências para o Futuro de Software

Inteligência Artificial e Machine Learning

A integração de IA e ML no software está transformando a maneira como as aplicações funcionam, tornando-as mais inteligentes e capazes de aprender e se adaptar. Assistentes virtuais como Siri e Alexa utilizam IA para entender e responder a comandos de voz.



SOFTWARE

Tendências para o Futuro de Software

Computação em Nuvem

O software baseado em nuvem permite que os usuários acessem aplicações e dados de qualquer lugar, promovendo a flexibilidade e a colaboração. Serviços como **Google Drive** e **Dropbox** permitem que os usuários armazenem e compartilhem arquivos na nuvem.



SOFTWARE

Tendências para o Futuro de Software

Internet das Coisas (IoT)

A IoT está expandindo o alcance do software para **controlar e monitorar dispositivos conectados**, criando novas oportunidades e desafios.

Dispositivos domésticos inteligentes já utilizam software para fornecer controle remoto e automação.



SOFTWARE

Tendências para o Futuro de Software

Realidade Virtual e Aumentada

A **VR** e **AR** estão abrindo novos horizontes para o desenvolvimento de software, especialmente em áreas como jogos, educação e treinamento. Aplicativos como Google Earth VR permitem que os usuários explorem o mundo em realidade virtual, enquanto ferramentas de AR são utilizadas em manutenção industrial para sobrepor informações digitais em equipamentos físicos.



FELIZ 7 PLAY

REALIDADE VIRTUAL X AUMENTADA

Conexão Divina

