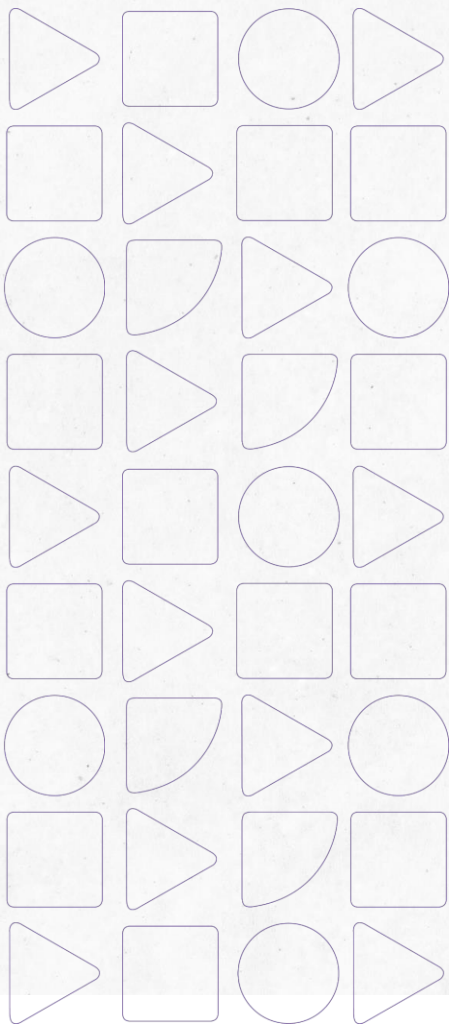


# Sistema operacional: *software* Aplicativos

**Disciplina:** Tecnologia e Sistemas de Informação



## Conteúdo:

Sistema operacional: *software*

*Aplicativos*

## Habilidades:

- Compreender a importância dos *softwares* para as máquinas;
- Analisar os tipos de *softwares*;
- Compreender as etapas de desenvolvimento de um *software*.
- Conhecer os tipos de aplicativos e as suas aplicações;
- Compreender o algoritmo por trás de um aplicativo.

Já falamos sobre os  
*hardwares*. Agora, é  
a vez dos *softwares*.

---





## O que é *software*?

*Softwares* são programas de computador que contêm instruções e dados que permitem que um computador execute tarefas específicas. Eles podem incluir:

- aplicativos;
- sistemas operacionais;
- *drivers* de dispositivos e outras ferramentas.

Imagine que um **computador** é como uma **casa**. Sendo assim, o *hardware* seria a estrutura física da casa, com as suas paredes, portas, janelas, os seus pisos e o seu telhado. Ou seja, é tudo aquilo que você pode ver e tocar.

Já o *software* seria a decoração, os móveis, os eletrodomésticos e tudo o que torna a casa funcional e útil. **Eles são as instruções e os dados que dão vida ao *hardware*, permitindo que ele execute tarefas específicas.**



## Classificações do *software*

Existem classificações de *software* para ajudar a organizar e categorizar os diferentes tipos de programas de computador com base em suas características e finalidades. Essas classificações são úteis para fins de comunicação, compreensão e seleção adequada do *software* necessário para atender a determinadas necessidades.

Temos algumas subdivisões:

- classificação por licença;
- classificação pela finalidade;
- classificação pela plataforma;
- classificação pelo modo de execução.



## Classificação por licença

- **Softwares proprietários:** são criados e distribuídos por empresas e os usuários podem usá-los apenas com licenças limitadas. Por exemplo, o **Microsoft Windows, Microsoft Office e o Adobe Photoshop;**
- **Softwares livres e de código aberto:** são desenvolvidos de forma colaborativa, o que significa que os usuários podem acessar e alterar o código-fonte. Por exemplo, o **Linux, o MySQL, o Python, o Mozilla Firefox, o Android e o LibreOffice.**



## Classificação pela finalidade

- Os *softwares* que **controlam e gerenciam o *hardware*** de um computador são conhecidos como **sistemas operacionais**;
- ***Softwares de produtividade*** são programas como editores de texto, planilhas e programas de apresentação, **usados para realizar tarefas específicas e aumentar a eficiência do usuário**;
- Os gráficos, as imagens e as ilustrações podem ser criados e editados usando ***softwares de design gráfico***;
- Por sua vez, os ***softwares de edição de vídeo*** são principalmente destinados a editar e manipular vídeos;
- Os ***softwares de banco de dados*** são usados para armazenar, administrar e recuperar grandes quantidades de informações;
- E os ***softwares de navegação na web*** são os programas usados para navegar na internet.



## Classificação pela plataforma

- ***Softwares desktop***: são projetados para funcionar em computadores pessoais ou estações de trabalho;
- ***Softwares web***: são executados em servidores remotos e podem ser acessados por meio de navegadores *web*;
- ***Softwares para dispositivos móveis***: podem funcionar em *tablets*, *smartphones* e outros dispositivos móveis;
- ***Softwares embarcados***: são integrados a sistemas eletrônicos e dispositivos específicos, como equipamentos médicos, eletrodomésticos e sistemas automotivos.





## Classificação pelo modo de execução

- **Nativos:** são compilados para uma arquitetura de *hardware* específica e podem ser executados diretamente no sistema operacional;
- **Interpretados:** podem ser executados usando um interpretador de código sem a necessidade de uma compilação prévia;
- **Virtuais:** são executados em uma máquina virtual que simula um ambiente de *hardware* e sistema operacional.



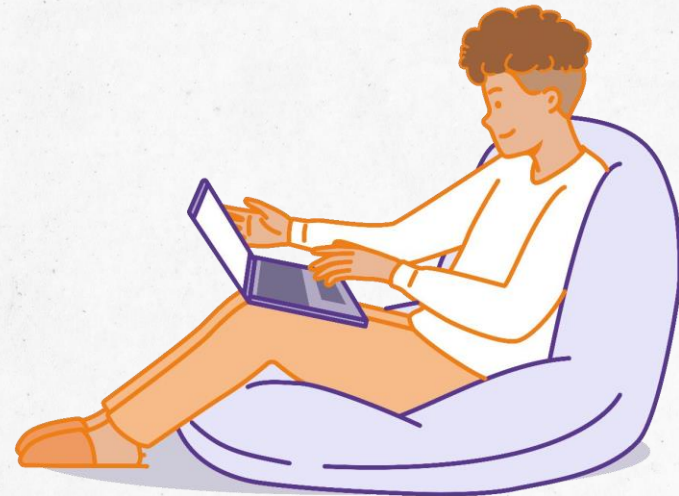


Existem alguns tipos de *softwares* como os de *design* gráfico e edição de imagens, os de edição de vídeo, de segurança, de banco de dados e de navegação na *web*.

# Aplicativos de produtividade

Esses *softwares* são projetados para auxiliar a realização de tarefas específicas e aumentar a **produtividade dos usuários**. Eles incluem editores de texto, planilhas eletrônicas, programas de apresentação, aplicativos de gerenciamento de projetos, entre outros.

Exemplos populares são o **Microsoft Office (Word, Excel, PowerPoint)**, **Google Docs**, **LibreOffice** e **Adobe Acrobat**.



## Softwares de *design* gráfico e edição de imagens

São utilizados para criar e editar gráficos, imagens, ilustrações e *designs*. Eles fornecem ferramentas avançadas para manipular e aprimorar elementos visuais, como cores, formas, texturas e efeitos.

Exemplos conhecidos são: **Adobe Photoshop**, **Adobe Illustrator**, **CorelDRAW** e **GIMP**.

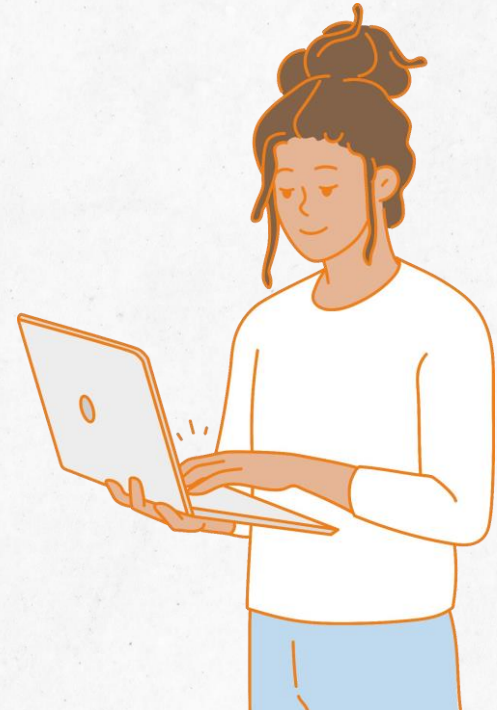




## Softwares de edição de vídeo

São usados para editar, cortar, adicionar efeitos visuais, áudios e realizar outras manipulações em vídeos. Eles são amplamente utilizados em produções audiovisuais profissionais e também por entusiastas do vídeo.

Exemplos populares incluem: **Adobe Premiere Pro, Final Cut Pro, Sony Vegas e DaVinci Resolve.**





## Softwares de segurança

Tem a função de proteger os computadores e sistemas contra ameaças, como **vírus**, **malware**, **spyware** e ataques cibernéticos. Eles incluem antivírus, *firewalls*, *softwares* de proteção de dados e criptografia.

Exemplos conhecidos são: **Avast**, **Norton**, **McAfee** e **Kaspersky**.



## Softwares de banco de dados

São usados para armazenar, gerenciar e recuperar grandes volumes de dados. Eles fornecem estruturas para organizar informações de forma eficiente e possibilitam consultas e manipulações dos dados.

Exemplos populares incluem: **MySQL, Oracle Database, Microsoft SQL Server e PostgreSQL.**



## Softwares de navegação na web

**São utilizados para acessar e navegar na internet.** Eles permitem a visualização de páginas da *web*, o acesso aos *sites*, a busca de informações e a interação com conteúdo *on-line*.

Exemplos comuns são: **Google Chrome, Mozilla Firefox, Microsoft Edge e Safari.**



## Aplicativos

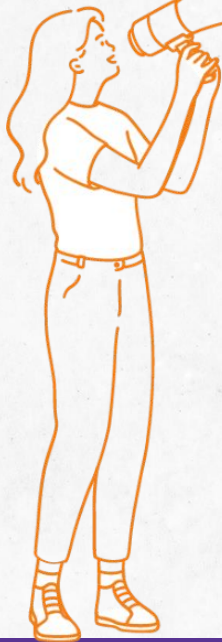
Os aplicativos têm feito muito sucesso devido à evolução dos dispositivos móveis.

Eles são programas de *software* projetados para executar tarefas específicas em dispositivos como, editar fotos, mandar mensagens, editores de texto, slides, planilhas e afins.



## Importante!

Existem **aplicativos *web*** (para computadores) e **aplicativos *mobile*** (compatíveis com celulares, *tablets* e afins). Ambos são importantes e desempenham uma melhor função no aparelho para qual ele foi criado.





# Diferentes tipos de aplicativos

Existem diferentes tipos de aplicativo e eles se moldam ao sistema.

- Aplicativos **mobile** podem ser desenvolvidos para as plataformas iOS ou Android;
- Enquanto os aplicativos **web** podem ser acessados em diferentes sistemas operacionais, como Windows, Mac ou Linux.

**Cada plataforma possui suas especificidades e impacta na forma como os aplicativos são desenvolvidos.**



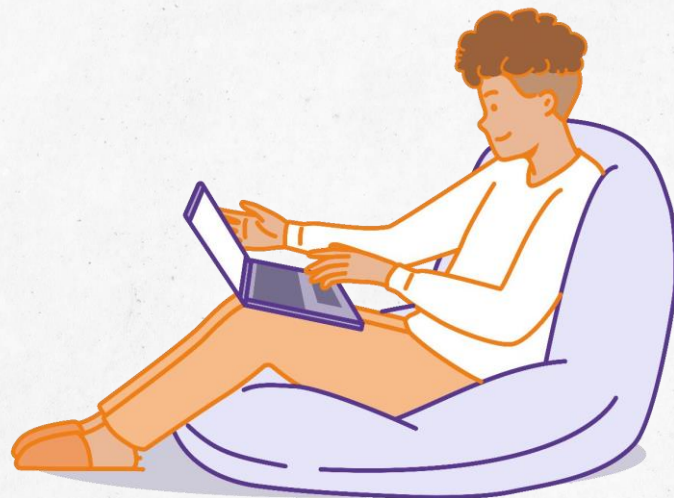
## Sistemas e suas especificidades

- **Aplicativos *Mobile* (iOS):** têm como especificidades o desenvolvido para dispositivos Apple, como iPhones e iPads. Usam a linguagem de programação Swift ou Objective-C;
- **Aplicativos *Mobile* (Android):** têm especificidades destinadas aos dispositivos Android, como smartphones e tablets. São desenvolvidos principalmente em Java ou Kotlin.
- **Aplicativos *Web* (Linux):** têm como especificidade ser acessível em sistemas operacionais Linux através de navegadores da *web*. Podem ser desenvolvidos usando uma variedade de tecnologias, como HTML, CSS e JavaScript.



## Sistemas e suas especificidades

- **Aplicativo Web (Mac):** têm como especificidade o acesso em sistemas macOS por meio de navegadores. São desenvolvidos com tecnologias *web* padrão, como HTML, CSS e JavaScript;
- **Aplicativo Web (Windows):** têm como especificidades ser projetado para sistemas operacionais Windows, acessível por navegadores. São desenvolvidos usando tecnologias *web* comuns.



▶ ◻ ○ ▶ ◻ ○ ▶ ◻ ○ ▶ ◻ ○ ▶ ◻ ○ ▶ ◻ ○ ▶

**Agora, nós vamos aprender sobre os aplicativos compactadores e descompactadores.**





## (Des)Compactadores

Os aplicativos compactadores e descompactadores são *softwares* desenvolvidos para reduzir o tamanho de arquivos (compactação) e restaurar os arquivos compactados para o formato original (descompactação).







# WinRar

O WinRar é um aplicativo que **serve para compactar ou descompactar arquivos** em qualquer computador, com suporte a diversos formatos.

- Com ele, **podemos reduzir o tamanho de um arquivo ou até mesmo dividi-lo em vários pedaços** para facilitar o envio ou o compartilhamento;
- Quanto ao item de descompactação, o programa suporta vários formatos como, RAR, ZIP, 7-ZIP, ACE, ARJ, BZ2, CAB, GZIP, ISSO, JAR, LZH, TAR, UUE, XZ e Z;
- Além disso, para efetuar uma compactação, o aplicativo é capaz de trabalhar com os tipos ZIP, RAR e RAR 5.

Esse *app* oferece diversos recursos com um ótimo desempenho e está disponível também em português.



## 7-Zip

Por sua vez, o **7-Zip é um programa de compressão de arquivos com código aberto.**

É um aplicativo de **fácil manuseio**, que apresenta um ótimo desempenho. O *app* utiliza o 7z, um formato de compressão próprio que melhora os resultados do ZIP em até 50%.

Ele também possui diversas opções para solucionar e **efetuar uma compactação e descompactação** rápida e prática, sendo compatível com **vários formatos**, como 7z, ZIP, GZIP, BZIP2, TAR, RAR, ISO, LZH, LZMA etc.

## Tente responder

Mas, como vocês acham que os *softwares* são  
construídos?



## Código algoritmo

Todos os programas nascem a partir de um código algoritmo conhecido como **linguagem de programação**.

Esta ordenação de códigos permite a criação e a concepção de um programa e, a partir destes padrões, os *softwares* começam a surgir.



## Composição do *software*

A sua composição geralmente trabalha com recebimento de dados (*input*). Após este recebimento, o programa procura realizar os seus processamentos e, por final, ele realiza a saída de dados ou resultados (*output*).





## Para entender melhor:

Por exemplo, em um *software* de **processamento de texto**, o **input** poderia ser o texto digitado pelo usuário. Já o **output** seria o texto formatado, visualizado na tela ou impresso.



## Vamos treinar! É *input* ou *output*?

1

Digitar uma pergunta no mecanismo de busca do navegador.

2

Pressionar o botão de chamada de um elevador.

3

O micro-ondas aquece o alimento pelo tempo especificado e emite um sinal sonoro quando termina.

4

Digitar um número de telefone em um celular.

5

A televisão liga ou desliga e a imagem e o som são exibidos na tela.



## Fechamento

Definam os conceitos a seguir, em uma única palavra:

- aplicativo *web*;
- aplicativo *mobile*;
- compactar;
- WinRAR;
- Zip.





Dê um *play* no conhecimento!



Pseudocódigo – vamos lembrar?





# Prática: Exemplos de pseudocódigo.

# Vamos praticar?

Dividam-se em cinco grupos!

## Primeiro momento

 15 minutos

Cada grupo deve pesquisar as seguintes linguagens de programação e como elas são utilizadas no desenvolvimento de um *software*:

1. Java;
2. JavaScript;
3. Python;
4. C++;
5. Ruby.

## Segundo momento

 15 minutos

Apresentem o que foi recolhido e expliquem como elas são utilizadas no desenvolvimento de *softwares*.



▶ ◻ ○ ▶ ◻ ○ ▶ ◻ ○ ▶ ◻ ○ ▶ ◻ ○ ▶ ◻ ○ ▶

# Vamos lá?

Vamos aprender sobre desenvolvimento de *software*!



## É hora de praticar – Relatório e Protótipo da Startup

Vamos realizar um projeto de sistema utilizando a Statup criada no grupo nas aulas de Empreendedorismo!

Vocês devem montar um escopo lógico para um projeto de desenvolvimento de *software*. O projeto precisa ter:

1. **Introdução** – Apresentação do projeto e seu propósito e explicação sobre o contexto em que o *software* será utilizado.
2. **análise de Requisitos** – Identificação das necessidades e funcionalidades esperadas do *software*, levantamento de requisitos essenciais para o desenvolvimento do sistema, definição dos principais atores e suas interações com o *software*.





- Indique as principais funcionalidades do sistema;
- Utilize o Canva ou o Google Slides para demonstrar como ficaria a interface básica do *software*.

Pode fazer direto:

## **Canva Sites**

<https://www.canva.com/websites/templates/>

## **Google Sites**

<https://sites.google.com/>



## É hora de praticar

Nós iremos construir o aplicativo  
utilizando **pseudocódigo!**

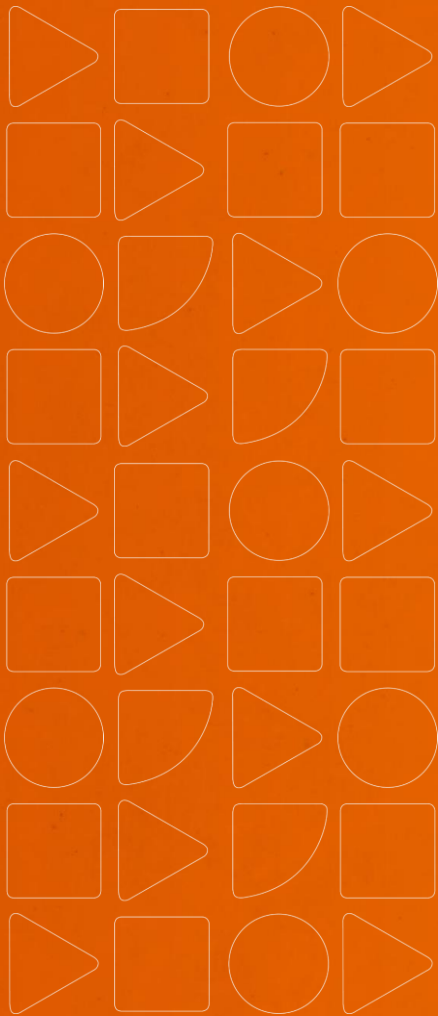
**Modelo base: Sistema Acadêmico**



## É hora de apresentar!

Cada grupo terá um limite de **seis minutos** para apresentar seus respectivos protótipos de aplicativo com representação das funcionalidades em pseudocódigo.





## Referências Bibliográficas

PROZ EDUCAÇÃO. *Apostila de Tecnologia e Sistema de Informação*. 2023.