



Universidade Federal de Uberlândia
Graduação em Sistemas de Informação
GSI003 - Introdução a Sistemas de Informação

Development

Bruno Caixeta Marques
Cássio de Souza Campos
Evandro Lopes
Gabriel Euripedes Vieira Correa
Jorge Luiz da Silva Madeira
Matheus dos Santos Mendes
Thomas Braga Costa

Índice

1. Algoritmos
2. Linguagens de Programação
3. Compiladores
4. IDE
5. Frameworks
6. API
7. Webservice
8. Unificação dos Temas
9. Referências

1. Algoritmos

Definição

Um algoritmo é uma sequência de comandos finitos que levam a resolver um problema.

Exemplos de algoritmos encontrados no dia a dia :

- Receitas culinárias.
- Manuais de instruções.

Características

- Devem ser finitos.
- Devem ser claro e direto, não podem ser ambíguo.
- Devem ser ordenados, não podem pular passos.

Formas de representação

- Descrição narrativa
- Fluxograma
- Pseudocódigo

Decisão narrativa

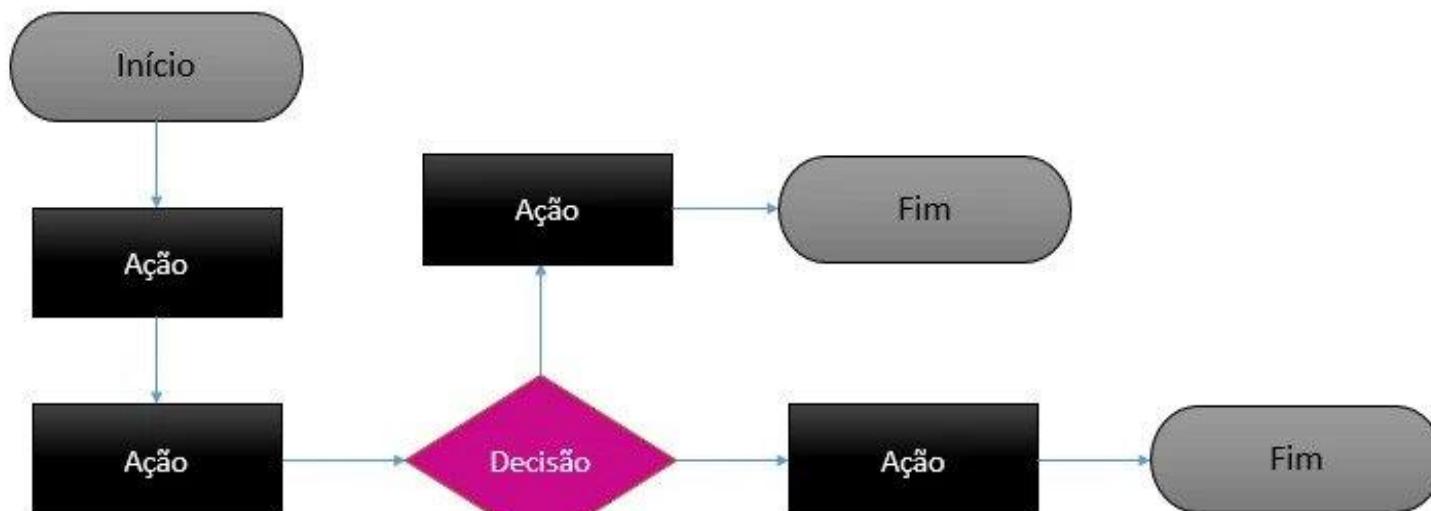
Algoritmo expresso em linguagem natural.

Exemplo, receita de bolo

- 1. Misture os ingredientes
- 2. Unte a forma com manteiga
- 3. Despeje a mistura na forma
- 4. Se houver coco ralado
- 5. Então despeje sobre a mistura
- 6. Leve a forma ao forno
- 7. Enquanto não corar
- 8. Deixe a forma no forno
- 9. Retire do forno
- 10. Deixe esfriar

Fluxograma

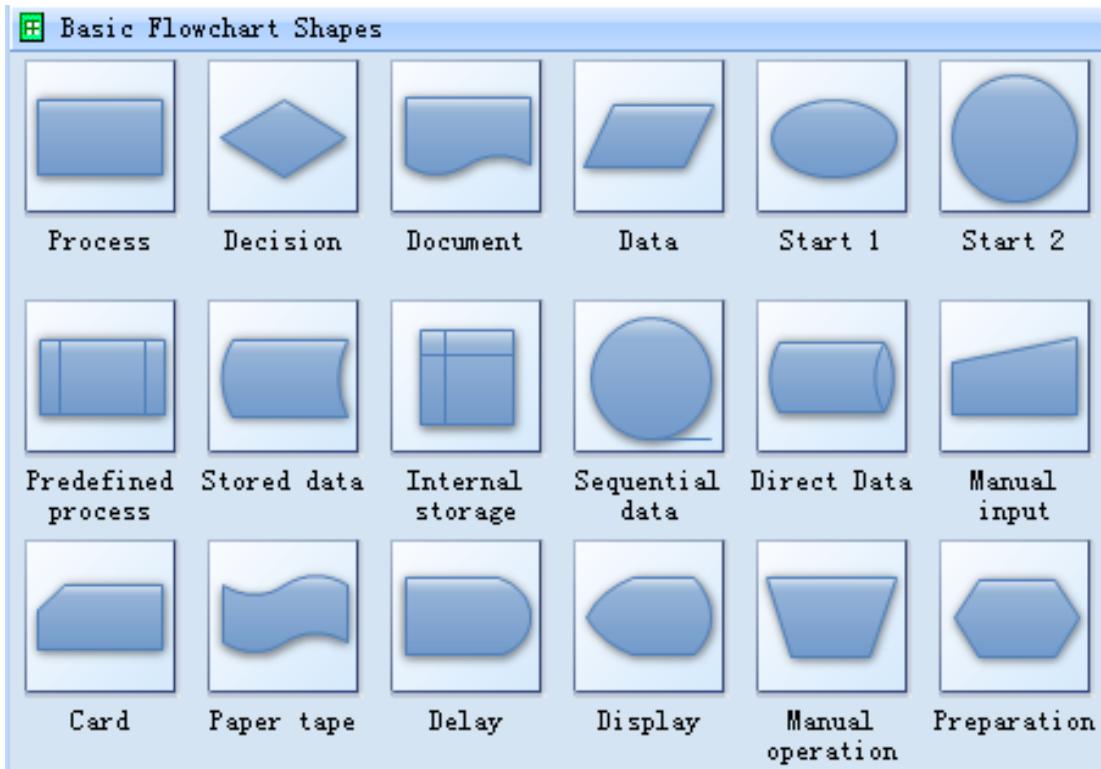
Representação gráfica do algoritmo onde formas geométricas implicam comandos



<https://blog.smlbrasil.com.br/5-passos-para-criacao-de-um-fluxograma/>

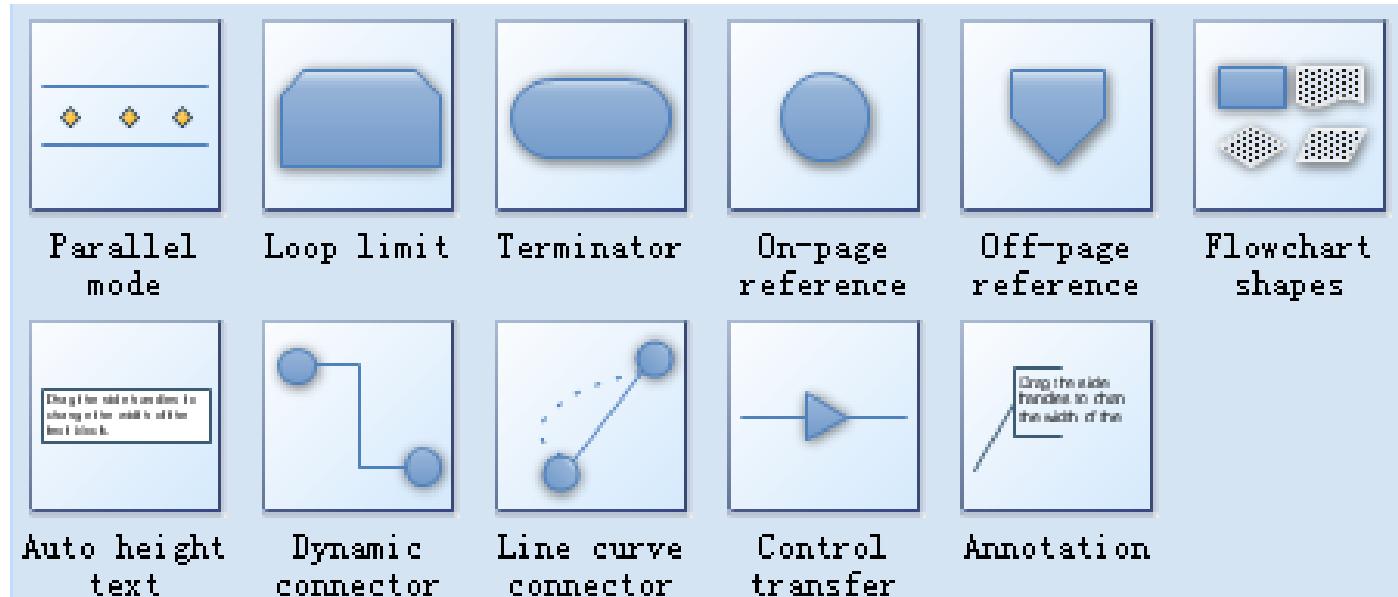
Formas básicas
de um
fluxograma :

Fluxograma



<https://www.edrawsoft.com/pt/flowchart-symbols.html>

Fluxograma



<https://www.edrawsoft.com/pt/flowchart-symbols.html>

Pseudocódigo

O pseudocódigo utiliza uma linguagem natural simples.

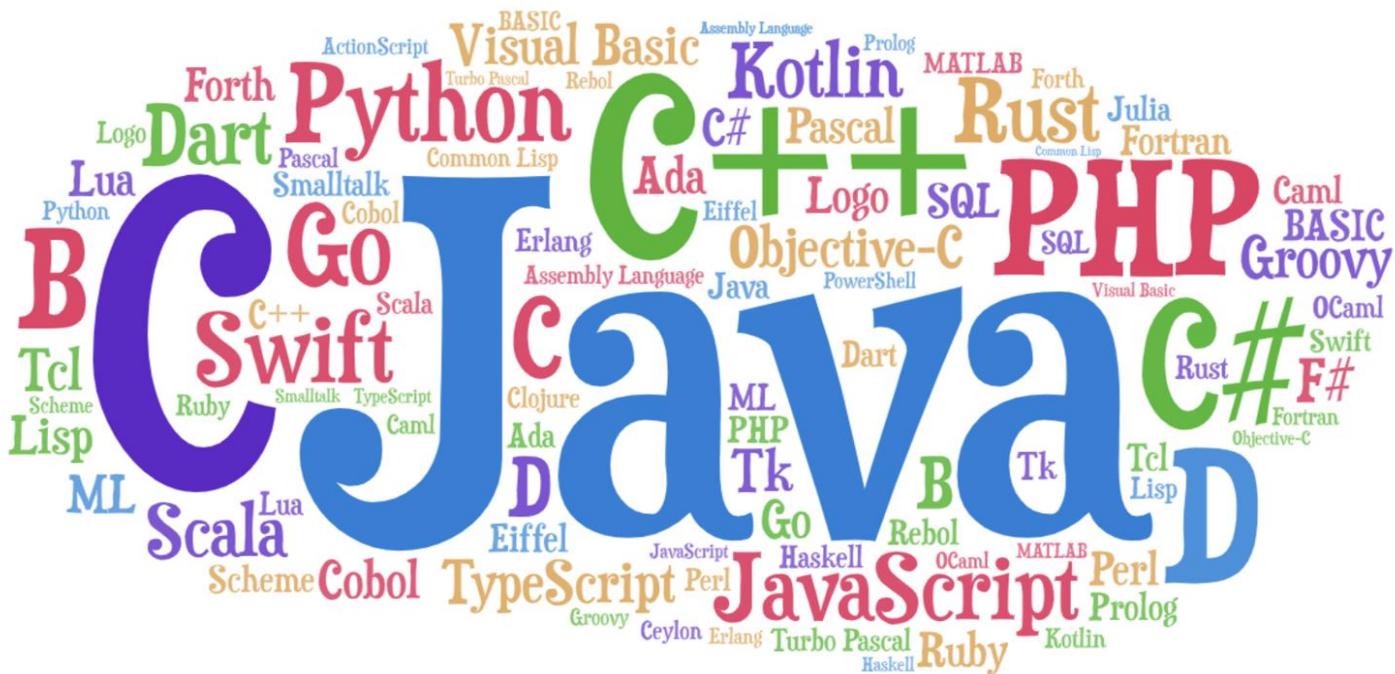
Programa para calcular a média das notas.

Exemplo de
algoritmo para
calcular a media
das notas :

```
DECLARA nota1,nota2,media:NUMERO  
  
ESCREVA "Digite a nota1:"  
LEIA nota1  
ESCREVA "Digite a nota2:"  
LEIA nota2  
  
media = (nota1 + nota2)/2  
  
ESCREVA "Sua média foi:"  
ESCREVA media  
  
SE media >= 7 ENTÃO  
    ESCREVA "APROVADO"  
SE media < 4 ENTÃO  
    ESCREVA "REPROVADO"  
SE media >= 4 E media < 7 ENTÃO  
    ESCREVA "Precisa fazer prova final"
```

<http://producao.virtual.ufpb.br/books/edusantana/pseudocodigolivro/livro/livro.chunked/ch02s04.html>

2. Linguagens de Programação



Conceito de linguagem de programação

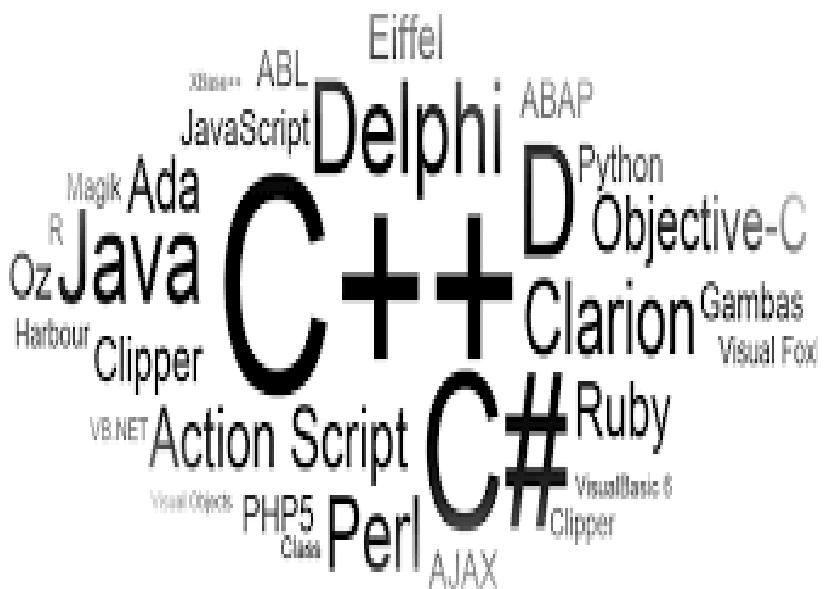
Linguagem de Programação é uma linguagem escrita e formal que especifica um conjunto de instruções e regras usadas para gerar programas (software). Um software pode ser desenvolvido para rodar em um computador, dispositivo móvel ou em qualquer equipamento que permita sua execução. Existem várias linguagens e elas servem para muitos propósitos. Alguns óbvios, como criar um software, outros menos, como controlar um carro ou uma torradeira.

Para que servem as linguagens de programação?

Um computador só é capaz de entender 1's e 0's e efetuar uma sequência de passos programados via hardware. É pra isso que existem as linguagens de programação, para facilitar a comunicação entre programador e hardware, o programador escreve instruções em uma linguagem bem próxima da que as pessoas usam para se comunicar, depois um segundo programa traduz o que o programador escreveu para sequências compostas por 0 e 1 (Compiladores)

Linguagens de programação de alto nível

São linguagens com um nível de abstração relativamente elevado, longe do código de máquina e mais próximo à linguagem humana. Desse modo, as linguagens de alto nível não estão diretamente relacionadas à arquitetura do computador.



Linguagens de baixo nível

Tratam-se de linguagens de programação que compreendem as características da arquitetura do computador. Assim, utiliza-se somente instruções do processador, para isso é necessário conhecer os registradores da máquina. Nesse sentido, as linguagens de baixo nível estão diretamente relacionadas com a arquitetura do computador. Um exemplo é a linguagem Assembly, que trabalha diretamente com os registradores do processador, manipulando dados.

Linguagem

Baixo nível – Mais próximo da linguagem de máquina



Alto nível – Mais próximo da linguagem natural

<https://image.slidesharecdn.com/njw6enwhrgevugee2kvg-signature-60ecb8dc276f4019baabfbf1cdf0096d8f7e393113770f6c5cbb37bed01471f3-poli-140917063629-phpapp01/95/processo-de-desenvolvimento-de-software-programao-e-linguagens-fortemente-x-fracamente-tipadas-6-638.jpg?cb=1410936672>

CÓDIGO ASSEMBLY

NOTA: ; , usado para fazer comentários em programas assembly

```
.MODEL SMALL ;modelo de memória  
.STACK ;espaço de memória para instruções do programa na pilha  
.CODE ;as linhas seguintes são instruções do programa  
    mov ah,01h ;move o valor 01h para o registrador ah  
    mov cx,07h ;move o valor 07h para o registrador cx  
    int 10h ;interrupção 10h  
    mov ah,4ch ;move o valor 4ch para o registrador ah  
    int 21h ;interrupção 21h  
END ;finaliza o código do programa
```

Este programa assembly muda o tamanho do cursor.

CÓDIGO DE MÁQUINA

LG Ei	111 00000000 IIII	(Leia Cartão e guarde em Ei)
COP Ei Ej	111 0001 IIII JJJJ	(Copie Ei em Ej)
VÀ Ei	111 00000001 IIII	(Vá para Ei)
IMP Ei	111 00000010 IIII	(Imprima Ei)
PARE	111 00000011 0000	(Interrompa execução)
SOM Ei Ej Ek	yyy IIII JJJJ KKKK	(Some Ei e Ej e guarde em Ek)
SUB Ei Ej Ek	yyy IIII JJJJ KKKK	(Subt Ei e Ej e guarde em Ek)
MUL Ei Ej Ek	yyy IIII JJJJ KKKK	(Mult Ei e Ej e guarde em Ek)
DIV Ei Ej Ek	yyy IIII JJJJ KKKK	(Div Ei e Ej e guarde em Ek)

PASCAL

```
program Hello;
var mensagem : string;
begin
  mensagem := 'Hello World!';
  write(mensagem);
End.
```

JAVA

```
public class Main {
  public Main(){
    System.out.println("Hello World");
  }
  public static void main(String [] args){
    Main m =new Main();
  }
}
```

C

```
#include <stdio.h>
#include <stdlib.h>

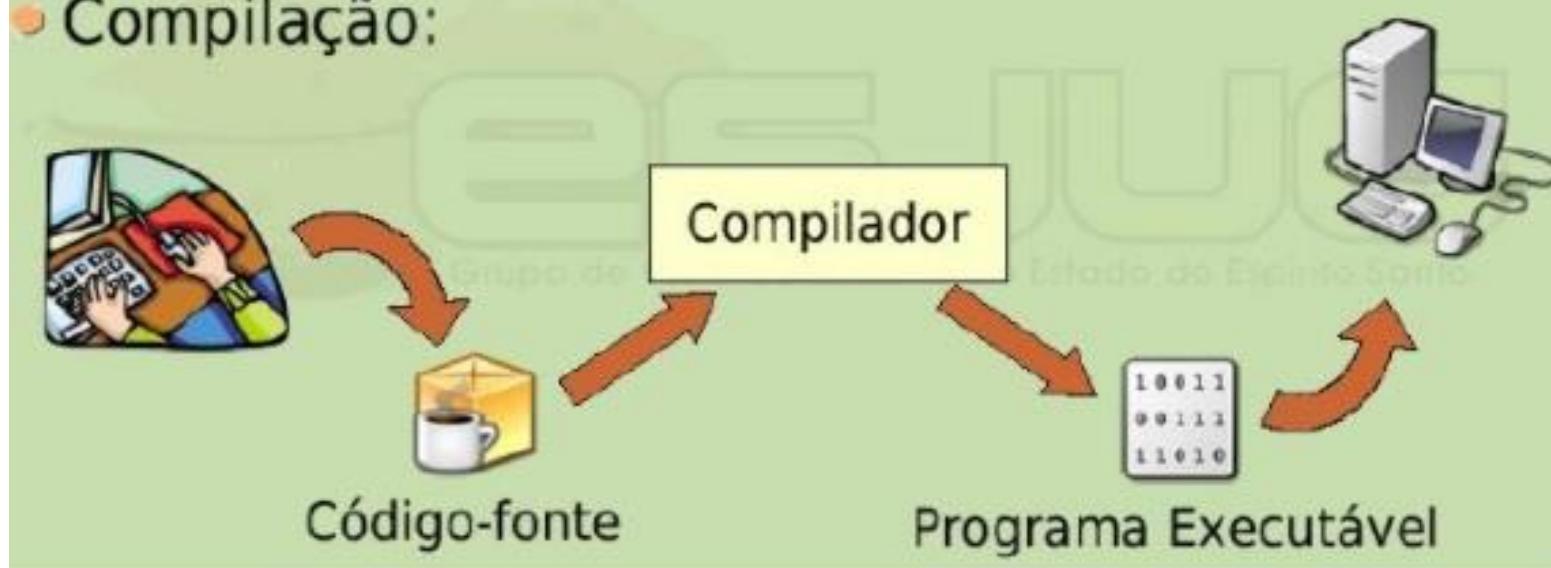
int main()
{
  printf("HELLO WORLD!!!");
  return(0);
}
```

COBOL

```
function Hello (){
  alert("Hello World!")
}
```

3. COMPILADORES

- Compilação:



<https://pt.slideshare.net/top100elearning/aula-de-introduo-java>

O que é um Compilador ?

- É um programa que lê um código escrito em uma linguagem (Linguagem fonte) e faz a tradução para um programa equivalente (Linguagem Simbólica)

Compilador e suas 6 etapas



<https://www.tiespecialistas.com.br/principios-de-um-compilador/>

Programas Relacionados a Compilação

- Editor
- Depurador
- Pré-Processador
- Carregador

Tipos de Compiladores

- Single - Pass
- Multi - Pass
- Load - And - Go
- Debugging
- Optimizing

Vantagens X Desvantagens de um Compilador

Compiladores

	Vantagens	Desvantagens
	Execução mais rápida	Várias etapas de tradução
	Permite estruturas de programação mais completas para a sua execução	Programação final é maior necessitando mais memória
	Permite a otimização do código fonte	Processo de correção de erros e depuração é mais demorado

<http://codemastersufs.blogspot.com/2015/10/compiladores-versus-interpretador>

4. IDE (Integrated Development Environment)

- Definição e características
- Prós e Contras do uso de uma IDE
- Diferença entre IDE e editor de texto
- Demonstração

Definição

- Ambiente de Desenvolvimento Integrado
- Agilidade no processo de desenvolvimento
- Combinação de ferramentas em uma única interface
- Codificação, depuração, execução, testes e correções

Características

- Editor de Código
- Compilador
- Linker
- Debugger
- Testes automatizados
- Refactoring

Prós e Contras

Prós

- Facilidade de desenvolvimento
- Facilidade de verificação e correção de erros
- Ferramentas em um único ambiente
- Aumento da produtividade

Prós e Contras

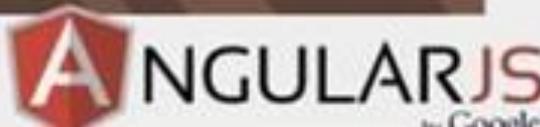
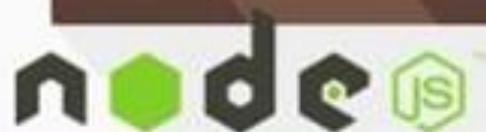
Contras

- Alguns ambientes e features são pagos
- Dependência do desenvolvedor
- Curva de Aprendizado
- Ambientes possivelmente lentos (Ex: Android Studio)

Diferenças entre IDE e Editores de Texto

- Editores são mais simples
- IDE's possuem ferramentas e features não abordadas em editores de texto
- Exemplos de editores
 - Sublime Text
 - Bloco de notas
 - Notepad++
 - Atom

5.



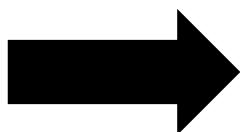
Tela de login

Usuário

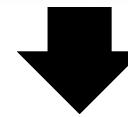
Senha

Entrar

Recuperar
senha



Quantas vezes foram
e serão necessárias?



Conjunto de funções que
auxiliam na agilidade de
construção do código.

FRAMEWORK x BIBLIOTECA

FRAMEWORK

Funções e componentes pré-estabelecidos para solução de **problemas específicos**

Ex: Tela de login

Contém

BIBLIOTECA

Funções pré-estabelecidas que resolvem um **tipo específico** de problema

Ex: Cálculos que remetem a dias/tempo.

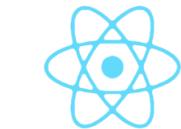
Chama cód.

Chama bib.

Seu código



JavaScript



React

- Todas as funcionalidades são inclusas.
- O fluxo do Angular chama o nosso código
 - ↓
 - Fluxo/estrutura diferente:
error
- Funções para a criação de componentes e criação de estados.
- As funções são chamadas no código.
 - ↓
 - Estrutura e fluxo da aplicação
- Outras funcionalidades (roteamento, animações, internacionalização): outras bibliotecas

Estrutura pronta é o principal diferencial!

Vantagens e desvantagens dos Frameworks



Menos bugs

- Testado
- Comunidade ativa

Facilidade de aprendizado

- Documentação extensa
- Fóruns de suporte.

Padronização de Código

- Padrão imposto a todos
- Futuros profissionais terão menos dificuldades

Redução de custos

- Tempo de produção reduzido.



Dependência

- Muito foco no uso do framework e não na linguagem usada para criá-lo
- Retrabalho caso haja migrações

Complexidade de modificação

- Várias funções interligadas causa dificuldade para alterações

Podem deixar o sistema pesado

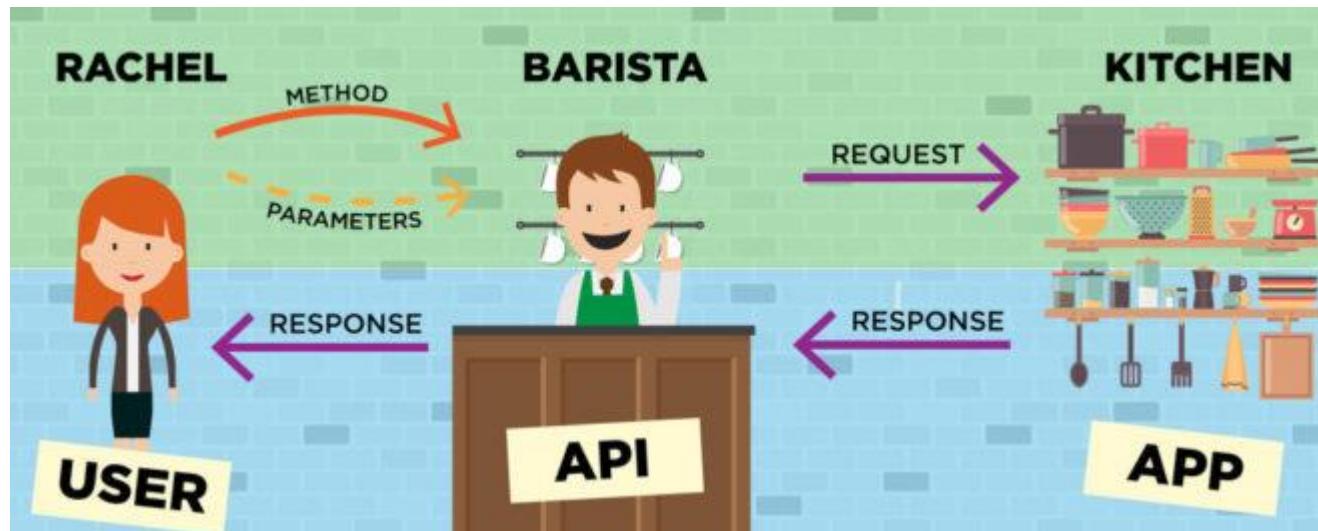
- Frameworks com funções desnecessárias para o produto podem deixá-lo pesado

Criando o próprio Framework

- Grandes empresas de TI
- Necessidades específicas
- Profundo conhecimento da linguagem

- Documentação
- Testes
- Correções e atualizações

6. API (Application Programming Interface)



O que são APIs

- API é um conjunto de definições e protocolos usado no desenvolvimento e na integração de software de aplicações, e assim possibilitando a troca de dados entre sistemas totalmente distintos.
- Ou seja um meio pelo qual dois sistemas se comunicam. Semelhante a um intérprete, que viabiliza a comunicação entre pessoas que não falam um mesmo idioma.

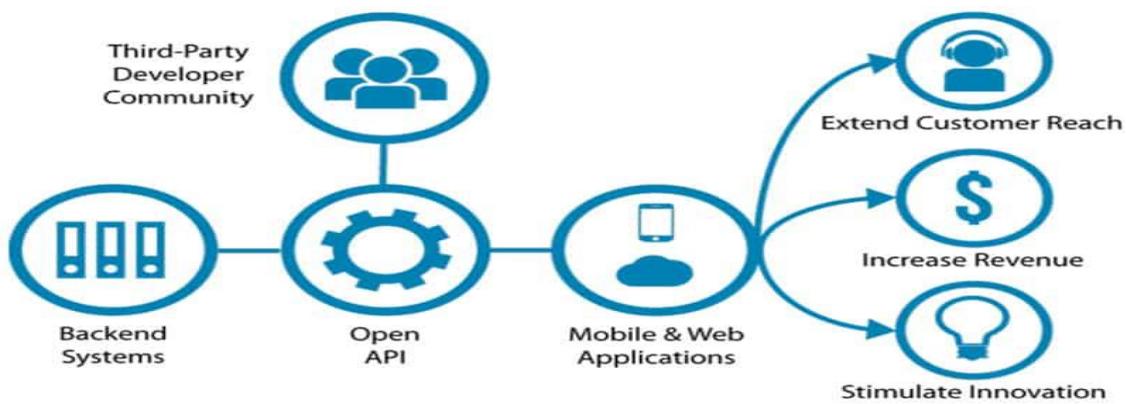
Vantagens no uso de APIs

- Adicionar diferentes funções a sites e aplicativos.
- Facilitar a segurança de dados, pois proporciona uma troca de informações de forma segura.
- Redução no volume de dados.
- Automatização de processos.
- Possibilidade de monetização por acessos.

APIs Públcas x APIs privadas

APIs Públcas:

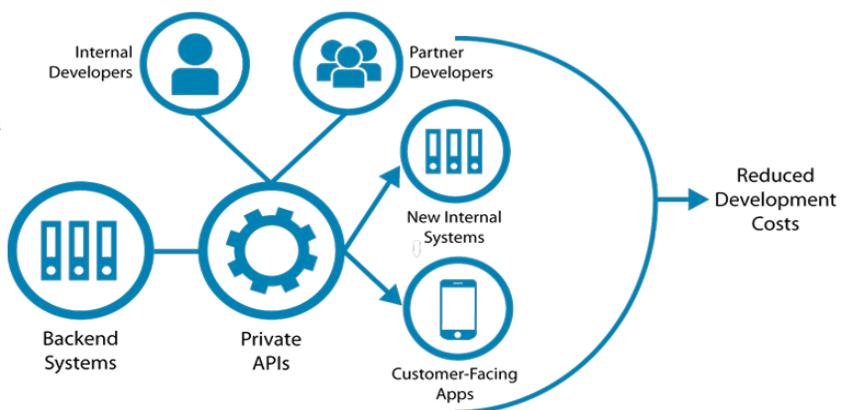
- Disponibilizada para outras empresas e desenvolvedores.
- Terceiros podem desenvolver aplicações que interajam com a sua API.



APIs Públcas x APIs privadas

APIs Privadas:

- Usada apenas internamente pelos seus próprios desenvolvedores ou parceiros específicos.
- Reduzem o tempo e os recursos necessários em diversos procedimentos:
 - integração de sistemas internos
 - construção de novos sistemas
 - otimização da comunicação interna



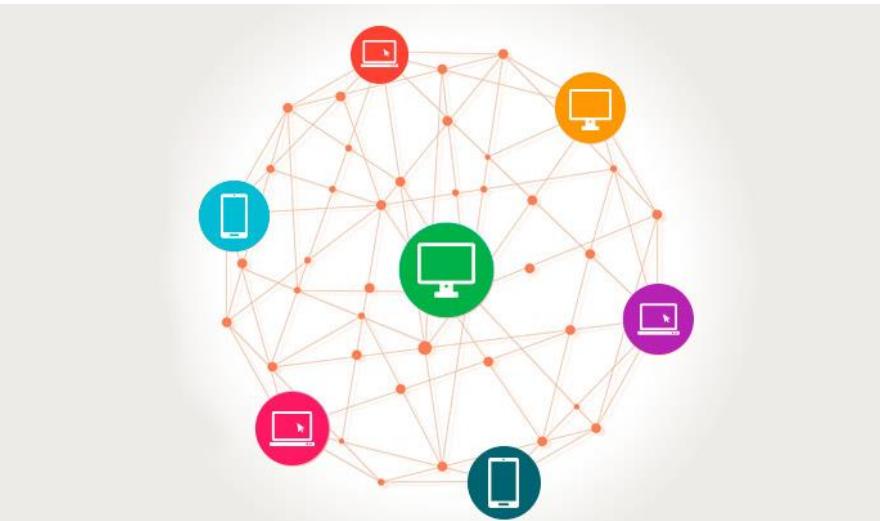
Exemplo de APIs

The screenshot displays the Tibia website's account management page. At the top, there is a navigation bar with links for Twitch, YouTube, Fankit, and a player count of 22,185 Players Online. Below the navigation bar, the main content area is titled "Account Management". It contains several sections:

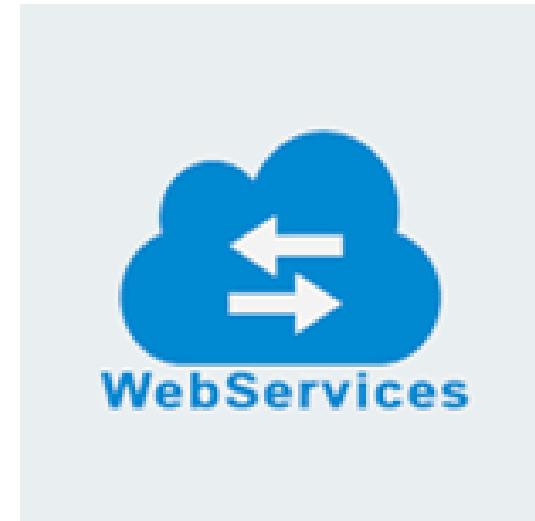
- Important Information!**: A message stating that account names have been abolished and login requires email address and password.
- Account Login**: A form for entering Email Address and Tibia Password. A red box highlights the "Use Authenticator" link next to the password field.
- Facebook Linkup**: A section prompting users to log into Facebook to connect their Tibia account or play Tibia via Facebook. A red box highlights the "Log in" button.
- Networks**: A section showing social media metrics: 263K people recommend Tibia, a "Share" button, a "Like" button, and a "Follow @Tibia" button.
- TRAILER**: A video player showing the "Tibia - Official Trailer 2016".
- Screenshots**: A grid of Tibia screenshots.

<https://www.tibia.com/account/?subtopic=accountmanagement>

7. Webservice



<https://www.opensoft.pt/wp-content/uploads/2016/06/05.jpg>



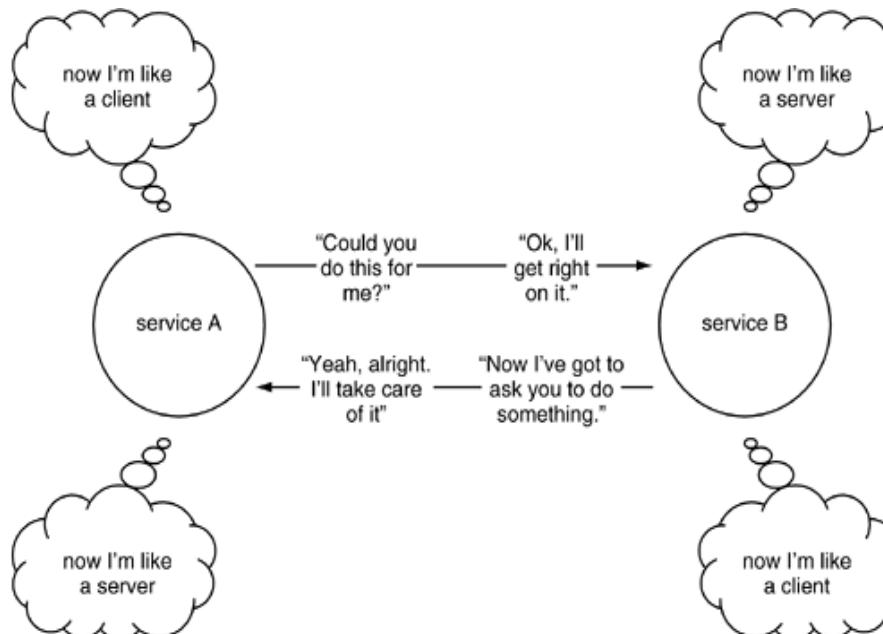
<http://ninelinux.com.br/wp-content/uploads/2019/01/e94e96eb-9.png>

Definição

- Serviço utilizado na comunicação entre diferentes sistemas e aplicações.
- Permite a compatibilidade entre aplicações novas e pré-existentes e sistemas desenvolvidos em plataformas distintas
- Traduz para uma linguagem universal

Como funciona?

- Uma aplicação solicita uma operação
- O Webservice converte os dados em “linguagem universal” e envia para outra aplicação
- Essa aplicação recebe os dados, converte em sua própria linguagem, realiza a operação e repete o processo

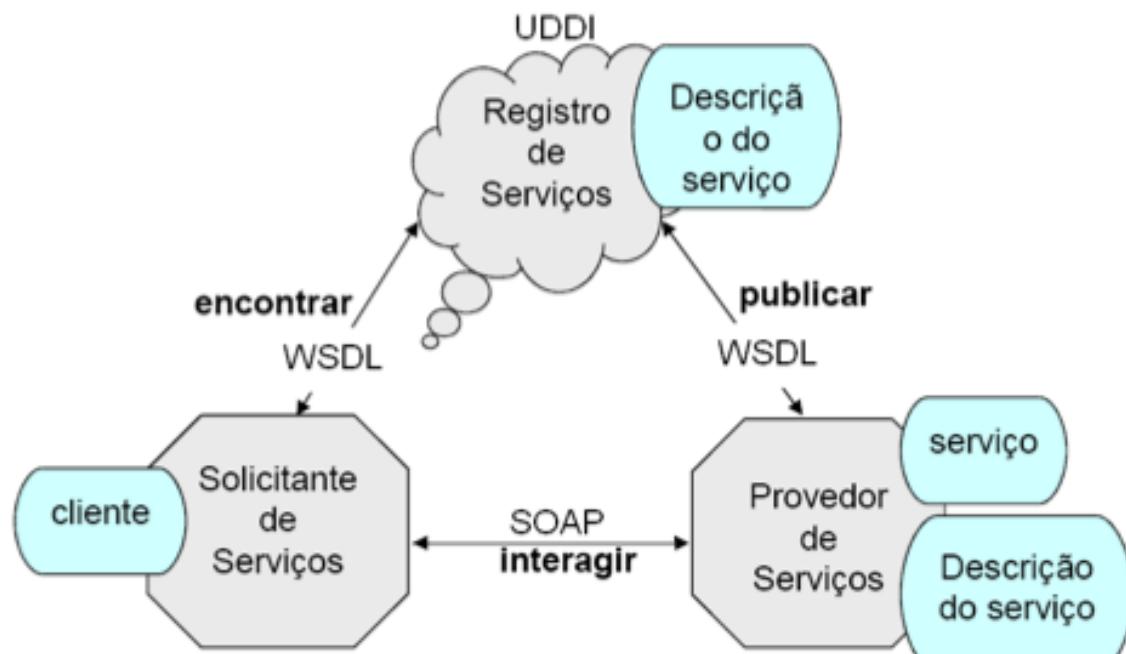


https://www.devmedia.com.br/imagens/webmobile/edicoes/1/wm1-7-uddi_arquivos/image002.gif

Características

- XML (eXtensible Markup Language)
- SOAP (Simple Object Access Protocol)
- WSDL (Web Services Description Language)
- UDDI (Universal Discovery Description and Integration)
 - REST (Representational State Transfer)

<https://i1.wp.com/www.diegomacedo.com.br/wp-content/uploads/2012/05/Componentes-b%C3%A1sicos-da-arquitetura-do-Web-Service.png?w=473&ssl=1>

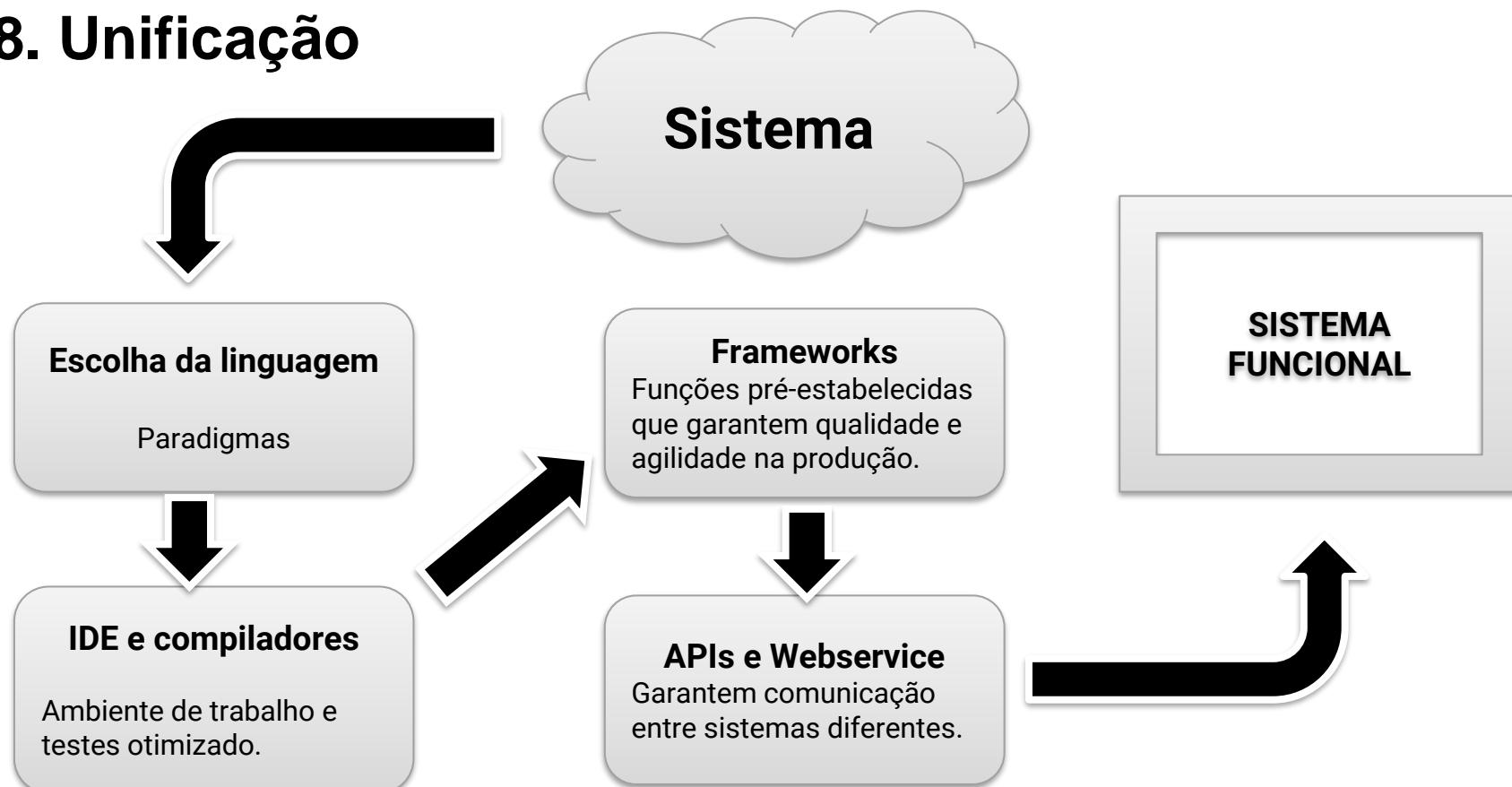


```
<?xml version="1.0"?>
<quiz>
<questão>
Qual o segundo nome de
Sofia Amundsen?
</questão>
<resposta>
Amundsen
</resposta>
<!-- Nota: Precisamos
perguntar mais.-->
</quiz>
```

XML

<https://upload.wikimedia.org/wikipedia/commons/thumb/8/8a/XML-pt.svg/200px-XML-pt.svg.png>

8. Unificação



9. REFERÊNCIAS

Algoritmos

<https://www.slideshare.net/elainececiliagatto/aed-formas-de-representao-de-algoritmos>

<https://www.ic.unicamp.br/~juliana/cursos/mc102/AulaAlgoritmos.pdf>

<https://www.embarcados.com.br/pseudocodigo/>

https://edisciplinas.usp.br/pluginfile.php/2984350/mod_resource/content/1/aula1_algoritmo%20e%20pseudocodigo.pdf

<https://pt.wikipedia.org/wiki/Algoritmo>

<https://rockcontent.com.br/blog/algoritmo/#:~:text=Alguns%20exemplos%20de%20algoritmos%20que,na%20receita%20culin%C3%A1ria%2C%20por%20exemplo>

Linguagem

https://cic.unb.br/~homero/tutorial%20icc/cap1_16.php

<http://evelinesolucoes.com.br/?p=928>

<https://pt.slideshare.net/natanaelsimoes/08-desenvolvimento-programao-linguagens-39189062>

9. REFERÊNCIAS

API

<https://maplink.global/blog/o-que-e-api/>

<https://programathor.com.br/blog/o-que-e-api/>

<https://www.redhat.com/pt-br/topics/api/what-are-application-programming-interfaces>

<https://vertigo.com.br/o-que-e-api-entenda-de-uma-maneira-simples/>

<https://vertigo.com.br/api-publica-ou-privada-como-saber/>

Webservice

<https://www.devmaster.com.br/web-services/2873>

<https://www.soawebervices.com.br/como-funciona.aspx>

https://www.w3schools.com/xml/xml_services.asp

<https://www.tecmundo.com.br/programacao/1762-o-que-e-xml-.htm>

<https://pt.wikipedia.org/wiki/REST>

<https://www.mma.gov.br/informmma/item/872-soap-simple-object-access-protocol.html>

<https://www.devmaster.com.br/wsdl-simplifique-a-integracao-de-dados-via-web-service/3006>

9. REFERÊNCIAS

Compiladores

<https://blog.geekhunter.com.br/metodos-de-traducao-compiladores-ou-interpretadores/>

<https://www.youtube.com/watch?v=2iR8HVecKKg&feature=youtu.be>

<http://www.dca.unicamp.br/cursos/EA876/apostila/HTML/node37.html>

IDE

<https://www.treinaweb.com.br/blog/o-que-e-uma-ide-ambiente-de-desenvolvimento-integrado/>

https://pt.wikipedia.org/wiki/Ambiente_de_desenvolvimento_integrado

<https://www.lucascaton.com.br/2017/12/30/qual-a-diferenca-entre-terminal-ide-e-editor-de-textos/>

Framework

<https://gaea.com.br/entenda-o-que-e-framework/>

<https://www.treinaweb.com.br/blog/para-que-serve-um-framework/>

<https://uilblognews.com/o-que-e-framework-e-porque-eu-deveria-aprender/>

<https://www.treinaweb.com.br/blog/qual-a-diferenca-entre-framework-e-biblioteca/>



Agradecimentos!

