

Criação e desenvolvimento de um software

Nomes:

Euller Henrique

Bruno Matos

André Oliveira

Mateus Herrera G. Borges

Wellington Silva

Henrique Costa Fernandes

Carlos Henrique



O desenvolvimento de um software é praticamente dividido em 3 etapas:

- Requisitos
- Viabilidade
- Código

Requisitos

Levantamento de requisitos

(Requisitos funcionais e não funcionais)

- Essa parte do desenvolvimento possui o objetivo de fazer com os desenvolvedores e clientes compreendam o problema para pensarem em como será a solução.
- Os clientes e desenvolvedores tentam levantar e priorizar as necessidades dos futuros usuários do software, isto é, os requisitos.



Requisitos

Análise de requisitos

- Nessa etapa os desenvolvedores realizam um estudo aprofundado dos dados adquiridos anteriormente.



Viabilidade

- É realizada uma extensa análise para determinar se o software é viável.
- Viabilidade técnica : Avaliação dos recursos técnicos disponíveis.
- Viabilidade operacional : Avaliação da eficácia da solução proposta.
- Viabilidade econômica: Avaliação do custo para desenvolver o software.



Código

Primeira etapa:
Criar Algoritmos

Algoritmos

- Instruções concisas e precisas
- São utilizados em qualquer linguagem de programação (JAVA, C , C++, HASKELL).
- Precisa respeitar determinadas regras para ser interpretado corretamente :
 1. Nenhuma linha pode ser ambígua;
 2. Os comandos devem estar nos lugares certos;
- Podem ser escritos em pseudocódigo, fluxograma ou em alguma linguagem de programação.



Pseudocódigo

- O pseudocódigo serve para que qualquer pessoa que não tenha conhecimentos avançados de programação consiga entender determinado algoritmo.

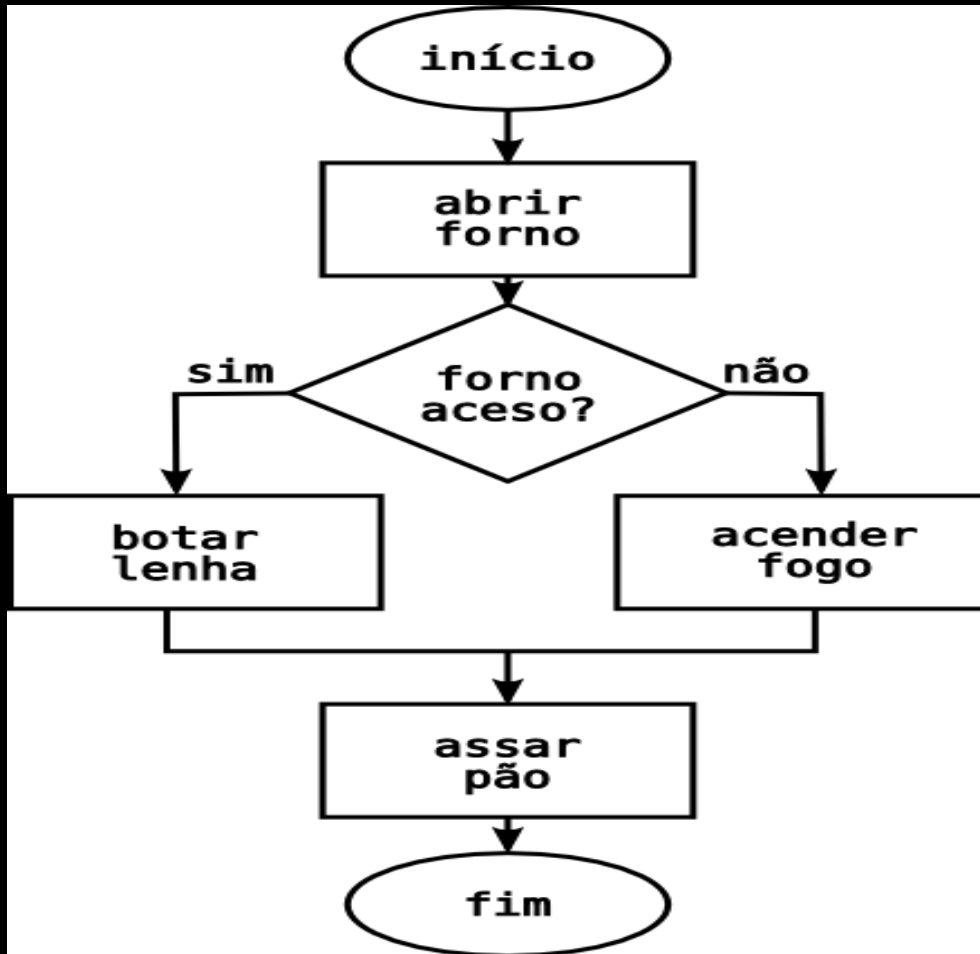


MMC

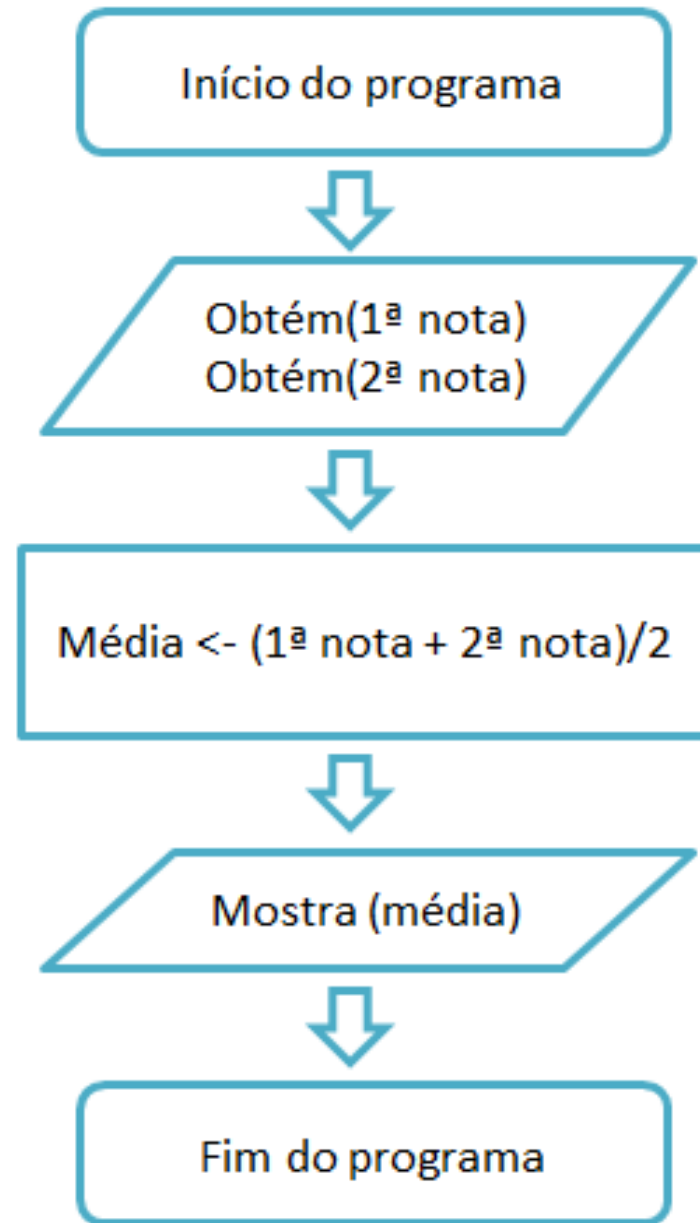
```
1 leia o número natural N1
2 leia o número natural N2
3 resto <- resto (N1/N2)
4 X1 <- N1
5 X2 <- N2
6 enquanto resto != 0 faça
7   N1 <- N2
8   N2 <- resto
9   resto <- resto(N1/N2)
10 fim enquanto
11 MDC <- N2
12 MMC <- X1 * X2 / MDC
12 escreva " MMC =", MMC
```

Fluxograma

- O fluxograma é visual e intuitivo, tornando-se assim um jeito ainda mais fácil de se entender um algoritmo.



<http://nivealseguinte.blogspot.com>



Algoritmo em C

MMC



```
1# include <stdio.h>
2 int main (void) {

3  unsigned int N1 = 0;
4  unsigned int N2 = 0;
5  unsigned int resto = 0;
6  unsigned int X1 = 0;
7  unsigned int X2 = 0;
8
9  scanf ("%u", & N1);
10 scanf ("%u", & N2);
11 resto = (N1 % N2);
12 X1 = N1;
13 X2 = N2;

14 while (resto != 0 ) {
15  N1 = N2;
16  N2 = resto;
17  resto = (N1%N2);
18 }
19 MDC = N2;
20 MMC = X1 *X2 / MDC;
21 printf ("MMC = %u\n", MMC);
22 return 0;
22 }
```

CÓDIGO

Terceira etapa :

Determinar a linguagem que será utilizada e codificar o software nesta linguagem



Conceito

- Linguagem de programação é o meio que se utiliza para "conversar" com o computador, de forma que seja possível interagir e programar a máquina para que ela realize determinadas tarefas com precisão.
 - Existem diversas linguagens, entretanto todas possuem o mesmo objetivo, isto é, possibilitar a interação entre o homem e a máquina de maneira eficiente por meio de códigos.
- Como cada linguagem foi feita para atender uma necessidade específica, não é possível caracterizar em "melhores e piores", devem ser utilizadas de acordo com a necessidade da situação.

Exemplos:

C

```
#include <stdio.h>
int main()
{
    // printf() displays the string inside quotation
    printf("Hello, World!");
    return 0;
}
```



Assembly

```
DATA SEGMENT
    MESSAGE DB "HELLO WORLD!!!$"
ENDS
CODE SEGMENT
    ASSUME DS:DATA CS:CODE
START:
    MOV AX,DATA
    MOV DS,AX
    LEA DX,MESSAGE
    MOV AH,9
    INT 21H
    MOV AH,4CH
    INT 21H
ENDS
END START
```

**ASSEMBLY
LANGUAGE**

CÓDIGO

Terceira etapa:
compilar

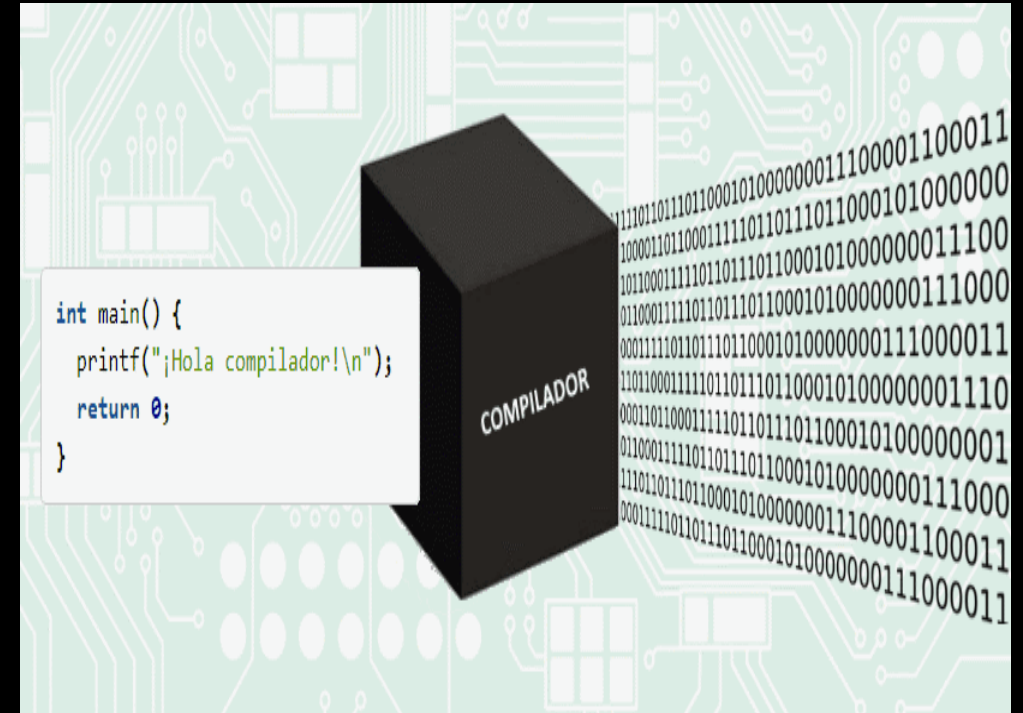
Conceito

- Compilador é um software que traduz um programa escrito em linguagem de alto nível para uma linguagem de baixo nível.

Como funciona

O processo de compilação ocorre em 6 fases:

- 1. Análise Léxica;
- 2. Análise Sintática;
- 3. Análise Semântica;
- 4. Geração de código intermediário;
- 5. Otimização de código;
- 6. Geração de código objeto;



1. Análise Léxica

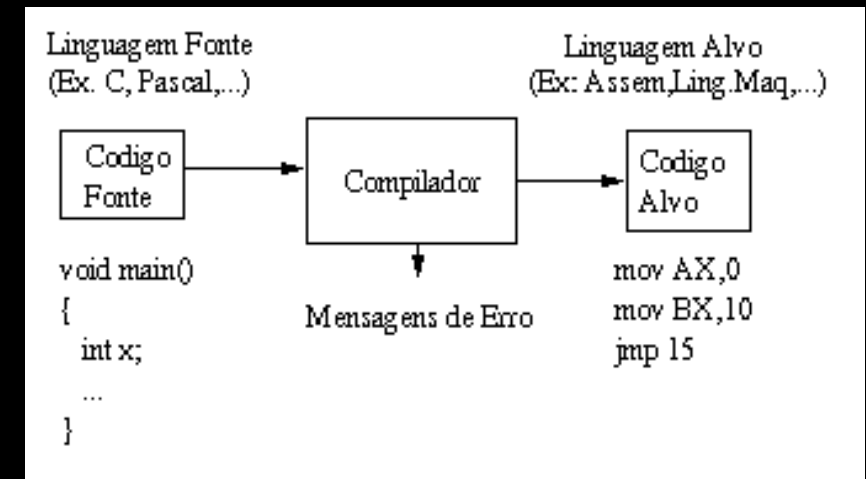
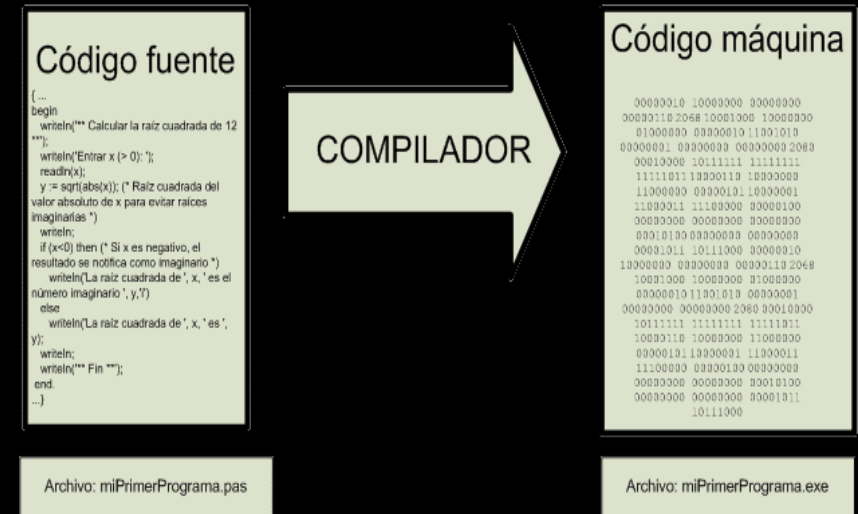
- Nesta fase é lida cada carácter do código fonte e verifica se eles pertencem ao alfabeto da linguagem, comentários e espaços em branco são descartados, os caracteres são convertidos em tokens com base em uma tabela de símbolos, formando uma sequência de tokens.

2. Análise Sintática

- A análise sintática analisa se a sequência de tokens formada na análise léxica forma comandos válidos à linguagem de programação.

3. Análise Semântica

- Nesta fase da compilação os comandos e instruções são analisados para garantir a funcionalidade, a integridade e a coerência do programa para que possa ser convertido para baixa linguagem.



4. Geração do código intermediário

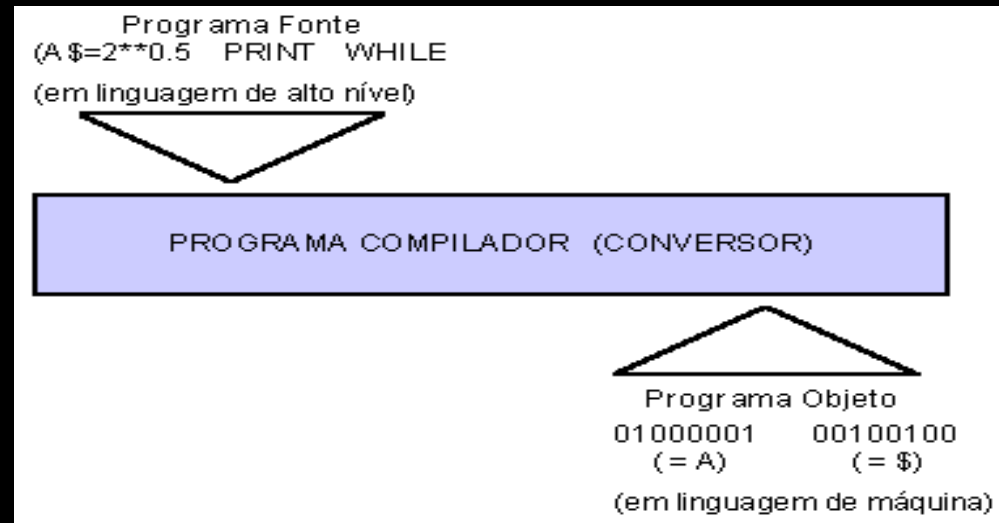
- Nesta fase é gerada uma sequência de código, que posteriormente gerará o código objeto.

5. Otimização de código

- Nesta fase o código é otimizado em termos de consumo de memória e velocidade de execução.

6. Geração do código objeto

- Esta é a última fase do processo de compilação, onde após a verificação de que não há erros sintáticos e semânticos o compilador gera, a partir do código intermediário, o código objeto final.



CÓDIGO

Opcionais :

O software pode ser feito em um IDE

- **Conceito**

- IDE é um software em que é possível digitar códigos fonte de maneira simplificada.

- **Funcionalidades**

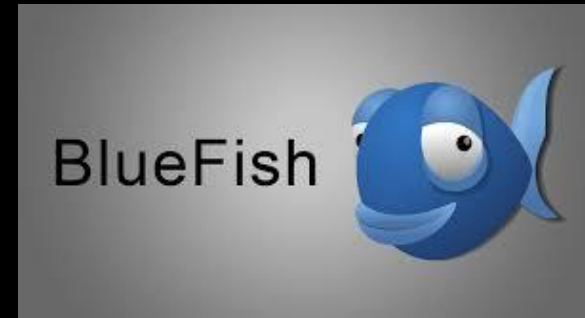
- A principal função da IDE é facilitar e agilizar o processo de digitação de código fonte.

- **Principais Funcionalidades**

- Editor de Texto;
- GUI Builder;
- Debugger;
- Compilador

- **Exemplos de IDE**

- Bluefish (Open Source);



- Microsoft Visual Studio (Microsoft);



- Xcode (Apple).



Código

Opcionais:
o código pode ser
Open Source

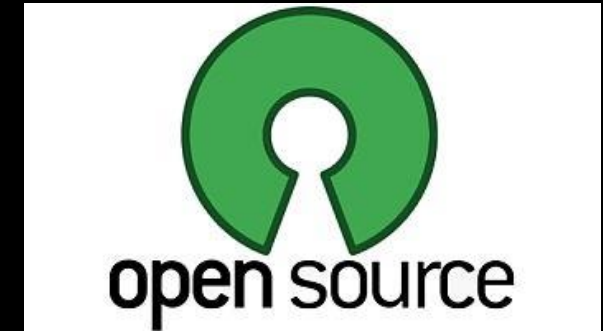
- Conceito

Modelo de desenvolvimento livre, que permite à qualquer pessoa analisar, modificar, consultar e distribuir o código para que outros possam fazer a mesma coisa, sem cobrar uma taxa por isso.

- História

O termo surge em uma reunião em 1998 entre pessoas que futuramente se tornaram referências no assunto, entre eles Todd Anderson, Chris Peterson, Jon “Maddog” Hall, Larry Augustin, Sam Ockman e Erick Raymond.

Com o cunho desse termo, fica claro que o objetivo dos desenvolvedores era evitar questões éticas, como o direito autoral e fornecer para as empresas um produto comercial.

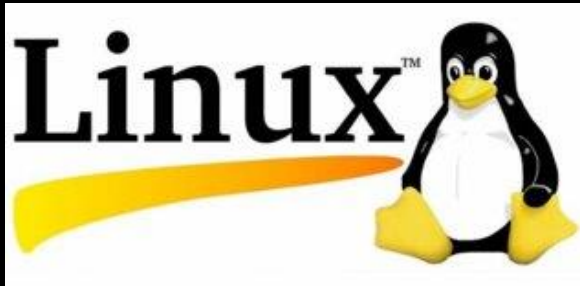


- Mercado

Um estudo da Oxford mostrou que 80% das empresas que foram entrevistadas usavam o softwares de código aberto, entre outras grandes empresas de tecnologia do mercado como Microsoft, IBM, HP, Intel e Dell.

Muito se acreditava que por ter seu código fonte distribuído gratuitamente o esperado era que as pessoas iriam estragar o produto, como hackers tentando criar janelas para quebrar a segurança de algum software.

Mas o que ocorre é o contrário, um caso que pode ser colocado é o do Linux, que antes de ser lançado oficialmente passou por uma versão *beta* aberto, onde milhares de pessoas puderam adquiri-la, identificar seus erros, sugerir e fazer melhorias antes de seu lançamento oficial.



Open Source x Software Livre

- O software livre garante 4 liberdades ao usuário:

1. Liberdade de executar o programa como desejar, para qualquer propósito.
2. A liberdade de estudar como o programa funciona, e adaptá-lo a suas necessidades.
3. A liberdade de redistribuir cópias de modo que você possa ajudar o próximo.
4. Liberdade de distribuir cópias de suas versões modificadas a outros, de modo que toda comunidade possa beneficiar.



- Diferença

A diferença é que Open Source simplesmente quer dizer que o código fonte do programa está aberto para consulta, e dependendo da **vontade** do criador para distribuição e redistribuição sob determinadas características.

O que vai determinar essa vontade é a licença que acompanha o programa.

O Software Livre implica a não propriedade do software, o Open Source pode ter um dono, como por exemplo os drivers da Nvidia que são Open Source hoje em dia, com o código fonte podendo ser acessado por terceiros mas somente quem vai promover alterações nele é a própria Nvidia.

CÓDIGO

Opcionais : o código pode ter
API

- **O que é API**

é um conjunto de rotinas e padrões de programação para acesso a um aplicativo de software ou plataforma baseado na Web.

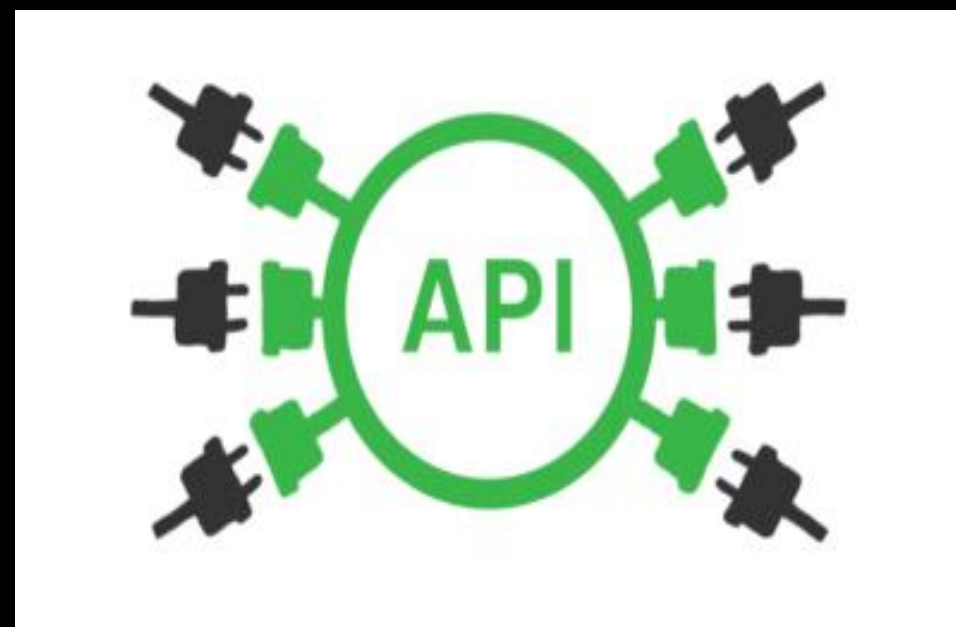
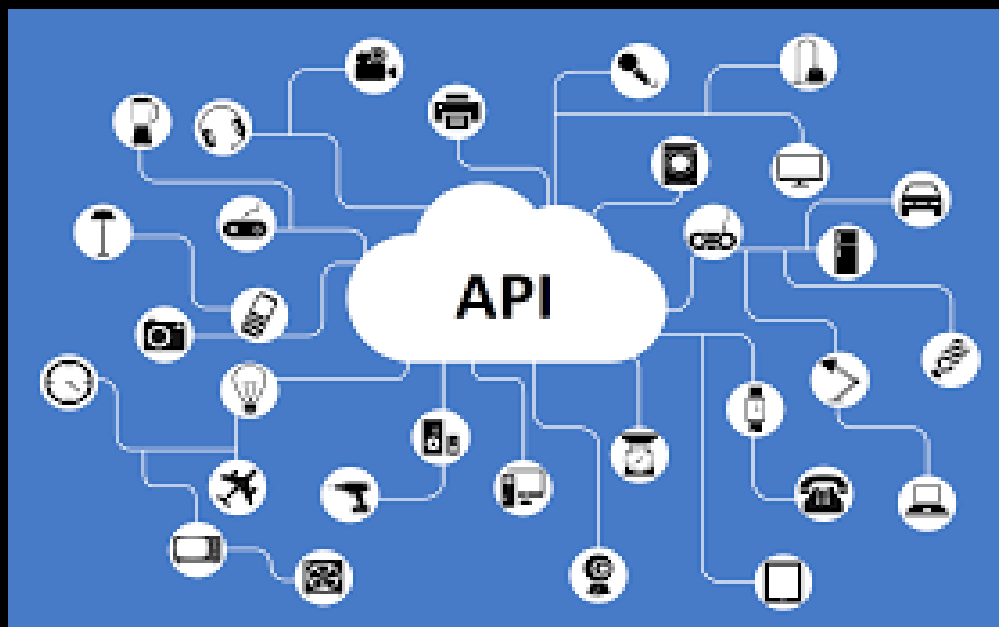
- **Como é usada**

Esta interface é o conjunto de padrões de programação que permite a construção de aplicativos e a sua utilização de maneira não tão evidente para os usuários

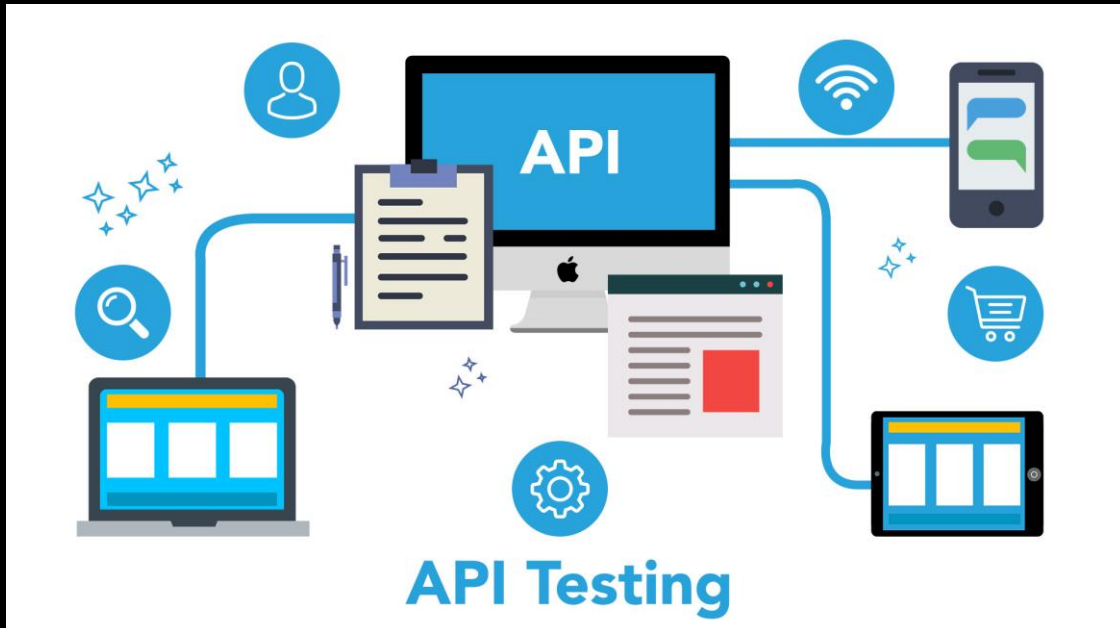


•API de maneira simples

Uma vida sem padrões definitivamente seria mais complicada e podemos usar diversas comparações para mostrar como as APIs facilitam diversos processos nesse sentido. Uma analogia interessante é com o sistema elétrico de uma casa. Da mesma forma que um aparelho utiliza a eletricidade da concessionária através de uma tomada elétrica, uma aplicação pode utilizar dados ou funcionalidades de um serviço através de uma API. Para ilustrar melhor, se não houvesse plugues, teríamos que ligar os cabos dos aparelhos eletrônicos diretamente nas estruturas elétricas da casa. Para isso, seria necessário possuir as ferramentas específicas para retirar todos os cabos para fora da parede, emendar com o cabo do aparelho a ser utilizado e guardar tudo de novo. Isso sem contar a necessidade de conhecer as especificidades de cada tipo de cabo elétrico e aparelho. Seria uma bagunça imensurável!



- O API Manager é uma plataforma completa de gestão de API's que visa acelerar o desenvolvimento de integrações entre sistemas e facilitar o processo de externalização de serviços para uso interno, externo e/ou de parceiros.
- Toda a plataforma foi construída pensando no conceito de atendimento multicanal, facilitando e reduzindo os custos do desenvolvimento de user interfaces como web, mobile, ura, sms, entre outros de forma centralizada e organizada. Para tanto foi criada uma camada ágil conectada aos sistemas internos da empresa onde com apenas um click a empresa pode fazer um deploy de uma nova versão das API's e disponibilizá-la para milhares de developers.



CÓDIGO

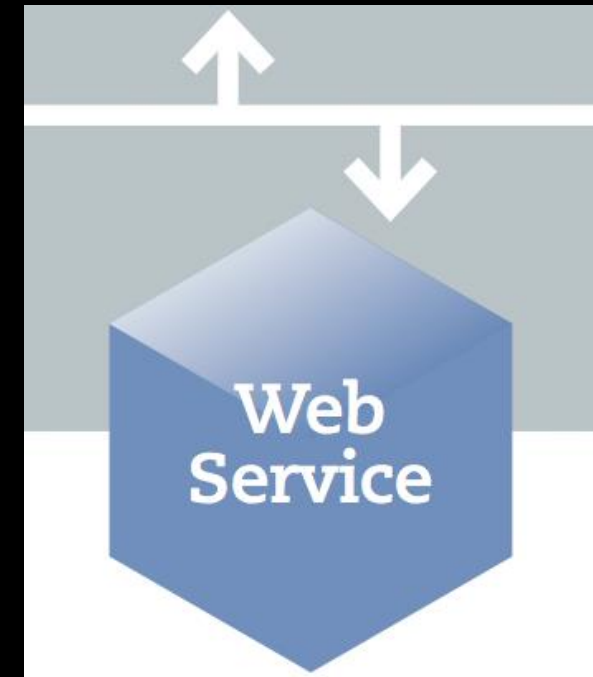
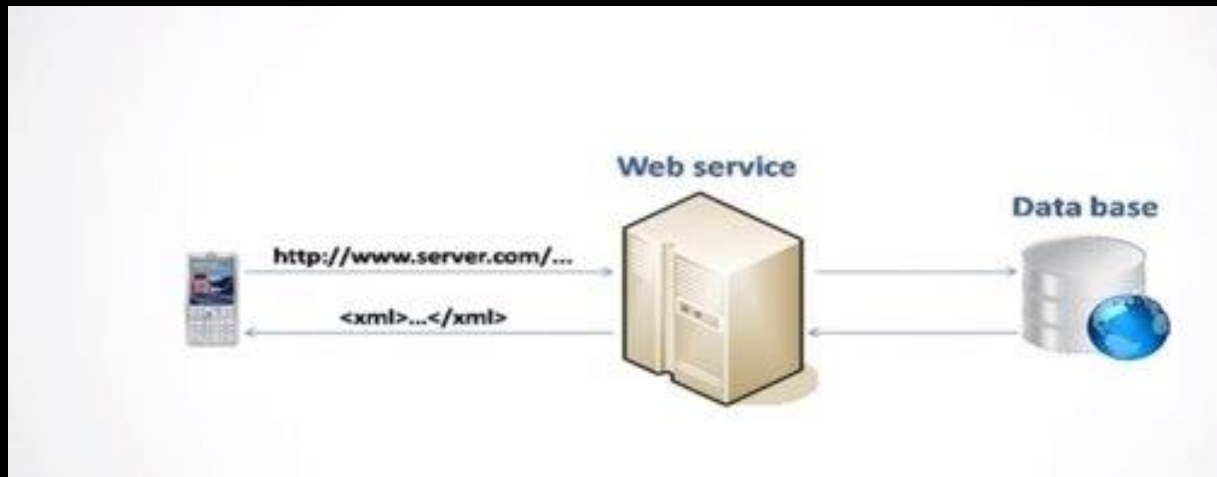
Opcionais :
o código pode ter Web
Service

- O Web Service é utilizado para integração entre sistemas, possibilitando interações independente do sistema operacional ou linguagem de programação dos mesmos. Isso é possível pois apesar de cada sistema ou aplicação utilizar sua própria linguagem, essa interação é traduzida para uma linguagem universal.



Como funciona:

- A aplicação solicita uma operação. O Web Service se comunica com o banco de dados e retorna o que foi requisitado pela aplicação em uma linguagem universal. A aplicação recebe os dados, faz uma interpretação e os converte para a sua própria linguagem.



SOAP E REST

- O Web Service depende de uma linguagem intermediária para se comunicar com o sistema que faz a requisição, como o SOAP (Simple Object Access Protocol) e o REST (Representational State Transfer).



- O SOAP utiliza XML para enviar mensagens e, geralmente, serve-se do protocolo HTTP para transportar os dados. Associado ao protocolo SOAP está o documento WSDL (Web Service Definition Language) que descreve a localização do Web Service e as operações que dispõe, fornecendo a informação necessária para que a comunicação entre sistemas seja possível.
- O REST, protocolo mais recente, tem por objetivo simplificar o acesso aos Web Services. Baseia-se no protocolo HTTP e permite utilizar vários formatos para representação de dados, como JSON, XML, RSS, etc. Tem como grandes vantagens sua flexibilidade, já que não limita os formatos de representação de dados e performance, uma vez que é um protocolo ágil e com a capacidade de transmitir dados diretamente via protocolo HTTP.

Sedex

- Sites de compras utilizam Web Service para localizarem o CEP, calcularem valor do frete e o tempo de entrega de uma mercadoria. Ao informar o CEP ou o nome da rua a aplicação solicita ao Web Service que busque no banco de dados do Sedex as informações de endereço, calculo do frete ,se o produto for enviado via Sedex, e o tempo de entrega.

⊖ Calcular Frete

CEP Origem

CEP

CEP Destino

CEP

Tipo de Envio

SEDEX

SEDEX

SEDEX a Cobrar

SEDEX 10

SEDEX Hoje

PAC

obrigatório

Peso (kg)

Kg

Total

Calcular Frete



Benefícios

- Alguns dos benefícios do Web Service são:
- Integração de informação e sistemas, já que seu funcionamento depende essencialmente de tecnologia XML/JSON e protocolos HTTP, simplificando a comunicação entre sistemas independente de plataformas e linguagens de programação
- Redução de tempo de desenvolvimento e custos já que os sistemas não são construídos do zero e pode-se incluir funcionalidades com facilidade
- Maior segurança pois o Web Service evita a comunicação direta com a base de dados.

