



Digital  
College

FORMAÇÃO EM

# DESENVOLVEDOR **FULL STACK**

UNIDADE 1:

**FUNDAMENTOS DA PROGRAMAÇÃO WEB**

MÓDULO 1:

> CONCEITOS E FUNDAMENTOS DA WEB >

### Aula 1:

**Temas abordados: Introdução ao desenvolvimento de sistemas e conceitos.**

#Datacenter #computacaoemnuvem #internet #SO #CLI #front-end #back-end.

- **A vida Pré-Internet**

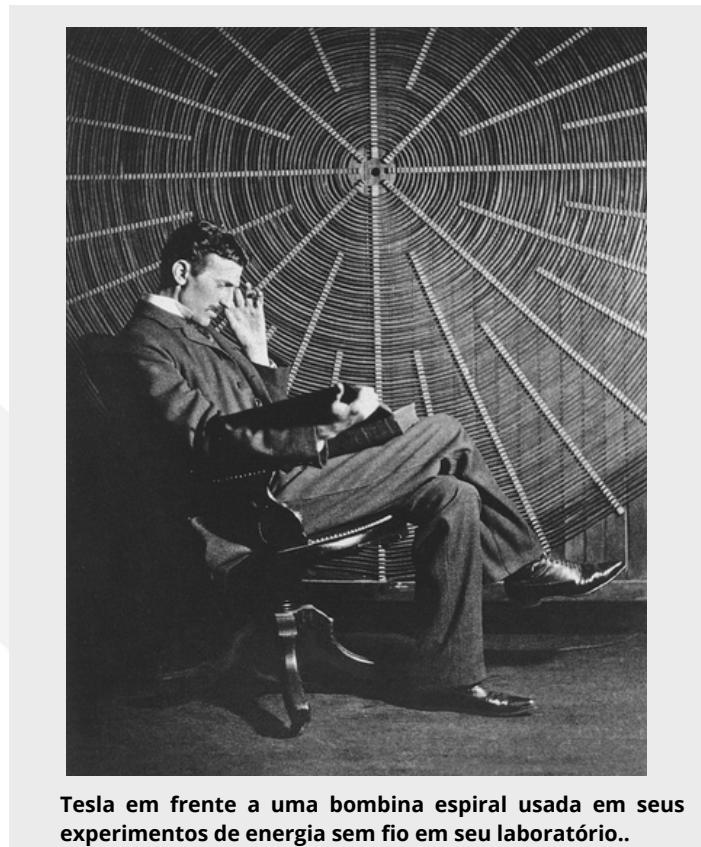
Antes da internet, nos comunicavamos principalmente por telefone, cartas, telefones públicos, jornais impressos, para acesso às notícias, telegramas, fax, este único bastante utilizado no meio corporativo. Acredite se quiser, não era instantâneo, com exceção do telefone, o qual exigia um bom tempo de planejamento e recursos para adquiri-lo, o tempo de espera para poder ter uma resposta exigia tempo e paciência.



Muitos filmes, principalmente, de ficção científica, tinham símbolos de alta tecnologia, como relógios com monitores, se comunicando como em uma videochamada, carros andando sozinhos, casas inteligentes, dentre muitas outras.

Mas a ideia dessas tecnologias serem tão avançadas para o início da década de 90, tinha sido prevista por grandes cientistas como Nikola Tesla. Em uma entrevista, em 1926, para a revista Colliers, Tesla previu, como os editores iriam escrever, "comunicando-se instantaneamente por meio de um simples equipamento de bolso". Suas palavras transmitem uma imagem muito mais grandiosa, e mais precisa, de uma imagem do futuro:

"Quando o sem fio for perfeitamente aplicado o mundo inteiro será convertido em um grande cérebro, o que de fato é... Seremos capazes de nos comunicar um com os outros, instantaneamente, independente da distância. Não apenas isso, mas através da televisão e da telefonia seríamos capazes de ver e ouvir uns aos outros perfeitamente como se estivéssemos frente a frente, apesar da distância de milhares de quilômetros; e os instrumentos pelos quais seremos capazes de fazer isso serão surpreendentemente simples em comparação com nosso telefone atuais. Um homem será capaz de carregá-lo no bolso do colete."



Tesla em frente a uma bombina espiral usada em seus experimentos de energia sem fio em seu laboratório..

Um dos eventos que mais atrai a atenção das pessoas, a copa do mundo de futebol, para que as pessoas pudessem saber quem ganhou, teriam que se dirigir até as agências de notícias e esperar pelos telegramas. Grandes tragédias ambientais, por exemplo, só eram informadas para pessoas em outros países, geralmente, no dia seguinte.

- **O início da internet**

### **O Mundo em Tensão**

O surgimento da internet deu-se em uma dos maiores conflitos não armamentistas da história, a Guerra Fria. No início dos anos 60, as maiores potências, Estados Unidos e União das Repúblicas Socialistas Soviéticas.

Essas grandes superpotências travaram um intenso conflito tecnológico e bélico, e a famosa corrida espacial. Os soviéticos saíram na frente como o envio do primeiro satélite ao espaço, Sputnik 1, em 1957 e quatro anos depois, em 1961, enviaram o primeiro homem ao espaço, em 12 de abril de 1961.

No dia 30 de outubro de 1961, os soviéticos realizaram o teste da maior bomba já criada pelo homem, a Tsar Bomb. O físico Andrei Sakharov, convenceu as autoridades soviéticas de que uma bomba H, termonuclear, de 100 megatons (Mt) era bastante perigosa, cerca de 700 vezes mais potente das lançadas nas cidades de Hiroshima e Nagasaki, com efeitos destrutivos incalculáveis. O projeto então reduziu a potência do artefato pela metade, tendo em vista, portanto, 50 megatons.

Os americanos responderam com o envio do primeiro homem à lua, o projeto Apollo 11, com a “corrida sendo vencida” pelos soviéticos, houve grande cobrança e apoio da sociedade para investimentos na corrida espacial e bélica. Assim como os soviéticos, os americanos desenvolveram diversas armas, bombas atômicas e nucleares superiores às que foram lançadas nas cidades japonesas.

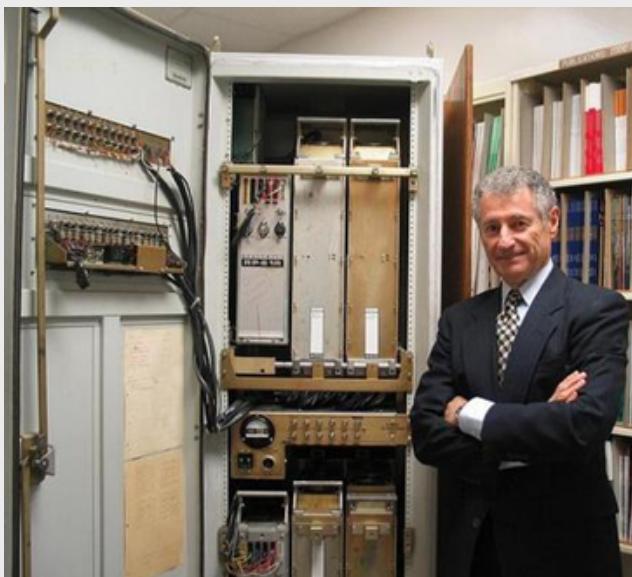
O cenário de tensão, tendo como plano de fundo o alto investimento em tecnologia de ponta, testes com bombas de destruição em massa fez com que os americanos pensassem em uma forma de trafegar seus dados entre as bases militares de maneira segura, confiável e rápida.

### **A Primeira Mensagem na Rede**

No início dos anos 50 os computadores existiam apenas em grandes laboratórios, eram enormes (em um deles cabia até uma pessoa dentro, no caso para fazer a manutenção, já imaginou?) e tinha o objetivo principal de fazer cálculos em larga escala.

Não haviam grandes fabricantes, empresas não possuíam e tão pouco nas casas das pessoas. Era algo inimaginável. Apenas as grandes potências da época possuíam alguma máquina dessas como Estados Unidos, Inglaterra e França, as nações pioneiras na criação da internet.

Como vimos que o mundo vivia em grande tensão, o uso dessas máquinas era destinado principalmente para projetos bélicos, espionagem, contra espionagem. Assim, houve uma necessidade de trafegar informações de grande segredo entre locais distantes e foi em um laboratório do Departamento de Defesa dos Estados Unidos que começaram a desenvolver a **ARPANET** (Advanced Research Projects Agency Network).



**Computador utilizado para enviar as primeiras mensagens da ARPANET, o SDS-90, com o professor Kleinrock ao seu lado.**

Este projeto tinha o intuito de interligar computadores locais em uma rede privada, que mais tarde se tornaria a grande rede global de computadores, comunicando-se de maneira simultânea com várias outras redes. Este conceito, primeiramente conhecido como internetworking, é um dos pontos chaves no surgimento da internet.

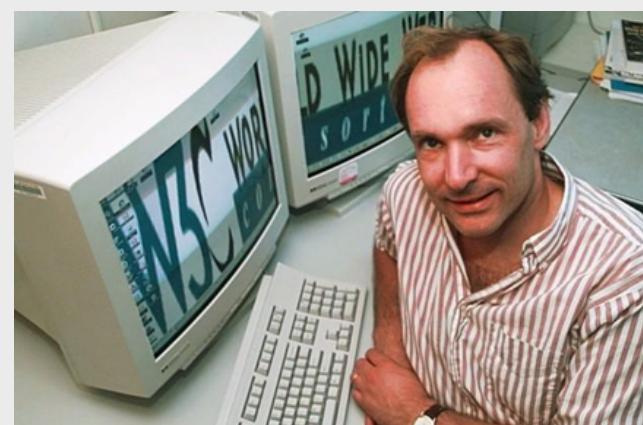
Em 1965, Lawrence Roberts fez dois computadores separados em lugares diferentes "conversarem" um com o outro pela primeira vez. Esta ligação experimental usou uma linha telefônica com um modem acusticamente acoplado e transferiu dados digitais usando pacotes.

Quando a primeira rede de comutação de pacotes foi desenvolvida, Leonard Kleinrock foi a primeira pessoa a usá-la para enviar uma mensagem. Ele usou um computador na UCLA para enviar uma mensagem a um computador em Stanford. Kleinrock tentou digitar 'login', mas o sistema travou depois que as letras 'L' e 'O' apareceram no monitor de Stanford.

Uma segunda tentativa foi bem-sucedida e mais mensagens foram trocadas entre os dois sites. A ARPANET nasceu.

### O www

Foi em 1974 que a abreviação do termo internetworking surgiu pela primeira vez, transformando-se em **internet**.



**Tim Berners Lee, chamado de "O pai da internet"**

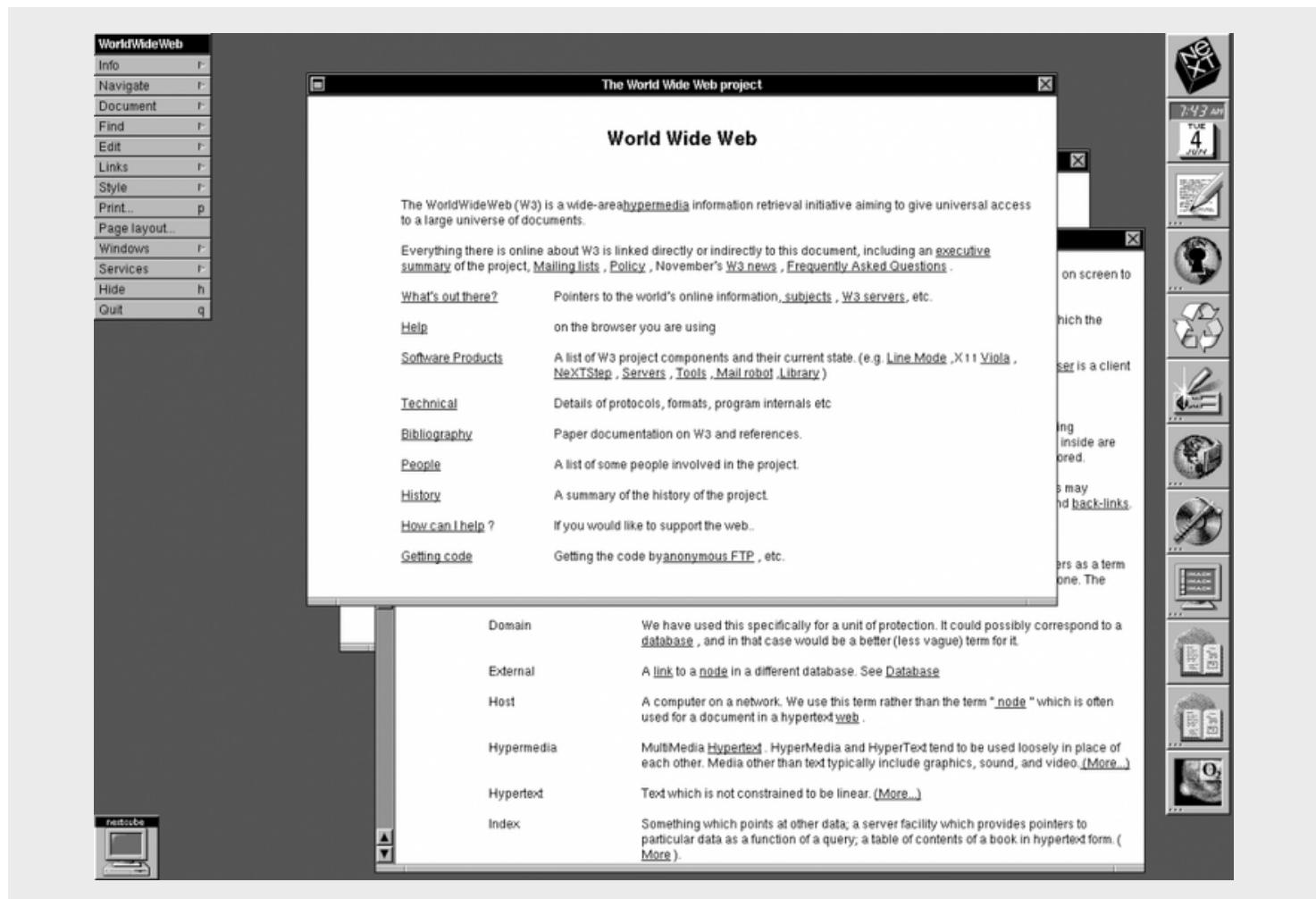
Tempo depois, Tim Berners-Lee, formou-se em física pela The Queen's College em Oxford e alguns anos depois começou a trabalhar para o CERN (European Organization for Nuclear Research, ou Organização Europeia para a Pesquisa Nuclear, em português) e com o propósito de facilitar a comunicação e a troca de informações entre os pesquisadores e instituições, propôs um projeto baseado em hipertextos em 1980.

No dia 13 de março de 1989 que Berners-Lee, entregou a seu supervisor um documento intitulado "Gerenciamento de informações: uma proposta". Com o auxílio do seu colega Robert Cailliau, eles construíram a linguagem global do hipertexto (o "http" dos endereços das páginas de internet) e desenvolveram o primeiro web browser (navegador de rede, como o "dinossauro" do Chrome, Firefox.).

Foi apenas em 1989 que unindo os hipertextos com a internet, nasceu a World Wide Web, que revolucionou e democratizou o uso da internet e tornando-se, como muitos o chamam, o "Pai da internet".

Hoje, Bernes-Lee é pesquisador do Massachusetts Institute of Technology (MIT), nos Estados Unidos, e professor de ciência da computação na Southampton University, ainda lidera o World Wide Web Consortium (a W3C) que coordena o desenvolvimento da web.

Tim criou o HTML (HyperText Markup Language), uma linguagem de marcação usada na criação das páginas e também o HTTP (Hypertext Transfer Protocol) principal protocolo que estabelece as conexões de internet no mundo todo. Também foi o criador do primeiro navegador, ou browser, ainda em modo texto, WorldWideWeb (www), em 1990. Posteriormente, para não confundir-se com sua própria rede, trocou de nome para Nexus.



Em 6 de agosto de 1991, Tim Berners-Lee disponibilizou a primeira página de internet, que ainda está online até hoje, e possui um formato muito similar ao que temos atualmente. A página foi feita em HTML e tentava explicar o conceito por trás do WWW como “uma iniciativa de recuperação de informação hipermídia para dar acesso universal a um grande universo de documentos”.

Este site foi hospedado por uma empresa, Next Computer, empresa fundada por Steve Jobs. Esse computador era conhecido como NEXTcube, tornando-se o primeiro servidor web do mundo.

- **Outros conceitos Importantes**

### Data Centers

Um data center é um conjunto de equipamentos, como modems, roteadores, switches, firewalls, sistemas de armazenamento, servidores e controladores de disponibilização de aplicativos; que possibilita e centraliza as condições necessárias para armazenamento, processamento e disseminação de dados e aplicativos, sendo possível ser em estrutura física **in loco** ou na **nuvem**.

Essas estruturas no meio corporativo são criados para prover a disponibilidade e recursos necessários para garantir alguns serviços essenciais, tais como:

- Armazenamento dos dados, gerenciamento, backup and restauração
- Produtividade das aplicações, tais como serviços de comunicação, por exemplo email

- Garantia de enormes transações para os e-commerce, por exemplo, o período de blackfriday, onde o aumento é exponencialmente maior comparado a outros períodos do ano.
- Fortalecendo as comunidades de jogos online
- Big data, machine learning and inteligência artificial
- Desktops virtuais, comunicações e serviços de colaboração
- Sistemas de gestão empresarial (ERP) e banco de dados
- Ferramentas de gestão do relacionamento com o cliente (CRM)

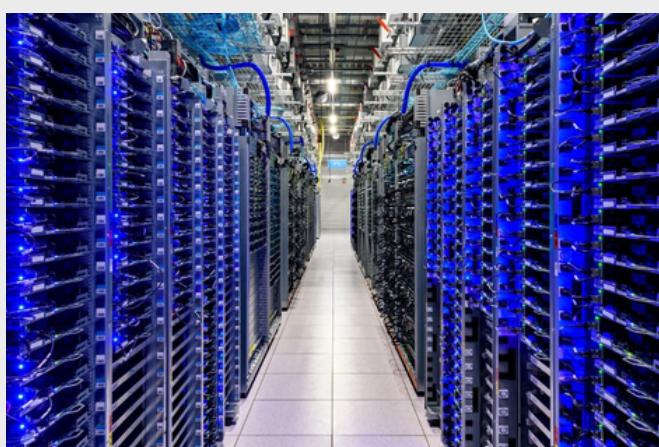
Como esses componentes armazenam e gerenciam dados e aplicativos essenciais para a empresa, a segurança do data center é fundamental no design do data center. Juntos, eles fornecem:

Independentemente de onde os recursos dos data centers estejam alocados eles devem obedecer algumas regras básicas:

Um dos padrões mais adotados para o design e a infraestrutura é o ANSI/TIA-942. Ele inclui os padrões da certificação, que garante a conformidade com uma das quatro categorias de camadas de data center classificadas para níveis de redundância e tolerância a falhas.

- **Camada 1:** infraestrutura básica do local. Um data center de camada 1 oferece proteção limitada contra eventos físicos. Possui componentes de capacidade única e um caminho de distribuição único e não redundante.

- **Camada 2:** infraestrutura do local dos componentes com capacidade redundante. Esse data center oferece proteção melhorada contra eventos físicos. Possui componentes de capacidade redundante e um caminho de distribuição único e não redundante.
- **Camada 3:** infraestrutura do local que pode ser mantida simultaneamente. Esse data center protege contra praticamente todos os eventos físicos, fornecendo componentes com capacidade redundante e vários caminhos de distribuição independentes. Cada componente pode ser removido ou substituído sem interromper os serviços para os usuários finais.
- **Camada 4:** infraestrutura do local tolerante a falhas. Esse data center fornece os mais altos níveis de tolerância a falhas e redundância. Componentes de capacidade redundante e vários caminhos de distribuição independentes permitem a manutenção simultânea e uma falha em qualquer lugar da instalação, sem causar período de inatividade.



### • Nuvem de computadores

Computação em nuvem (cloud computing) é um termo coloquial para a disponibilidade sob demanda de recursos do sistema de computador, especialmente armazenamento de dados e capacidade de computação, sem o gerenciamento ativo direto do utilizador. O termo geralmente é usado para descrever centros de dados disponíveis para muitos utilizadores pela internet. Nuvens em grande escala, predominantes hoje em dia, geralmente têm funções distribuídas em vários locais dos servidores centrais.

Empresas como o Google, Amazon, Microsoft, Oracle e IBM foram as primeiras a iniciar uma grande ofensiva nessa nuvem de informação.

O primeiro serviço na internet a oferecer um ambiente operacional para os usuários foi criado por Fredrik Malmer e utilizou JavaScript, a linguagem de programação que vamos utilizar durante todo o curso.

Segundo o NIST, Instituto Nacional de Padrões e Tecnologia, as cinco características essenciais da computação em nuvem são:

- **Autoatendimento sob demanda** - Um consumidor pode utilizar unilateralmente recursos de computação, como tempo de servidor e armazenamento de rede, conforme necessário, automaticamente, sem exigir interação humana com cada provedor de serviços;
- **Amplo acesso à rede** - Os recursos estão disponíveis na rede e são acedidos por meio de mecanismos padrão que promovem o uso por plataformas heterogêneas (por exemplo, telefones celulares, tablets, notebooks e estações de trabalho);

- **Pool de recursos** - Os recursos de computação do provedor são agrupados para atender a vários consumidores usando um modelo de multilocação, com diferentes recursos físicos e virtuais atribuídos dinamicamente de acordo com a demanda do consumidor;
- **Elasticidade rápida** - Os recursos podem ser provisionados e liberados elasticamente, em alguns casos automaticamente, para escalar rapidamente para fora e para dentro de acordo com a demanda. Para o consumidor, os recursos disponíveis para utilização muitas vezes parecem ilimitados e podem ser apropriados em qualquer quantidade e a qualquer momento;
- **Serviço medido** - Os sistemas em nuvem controlam e otimizam automaticamente o uso de recursos aproveitando um recurso de medição em algum nível de abstração apropriado ao tipo de serviço (por exemplo, armazenamento, processamento, largura de banda e contas de utilizador ativas). O uso de recursos pode ser monitorado, controlado e relatado, fornecendo transparência tanto para o provedor quanto para o consumidor do serviço utilizado.

Existem 3 modelos de implantação da nuvem:

### Nuvem privada

As nuvens privadas são aquelas construídas exclusivamente para um único utilizador (uma empresa, por exemplo). Diferentemente de um data center privado virtual, a infraestrutura utilizada pertence ao utilizador, e, portanto, ele possui total controle sobre como as aplicações são implementadas na nuvem. Uma nuvem privada é, em geral, construída sobre um data center privado.

### Nuvem pública

Uma nuvem é chamada de "nuvem pública" quando os serviços são disponibilizados em uma rede aberta para uso público.

### Nuvem híbrida

Nas nuvens híbridas temos uma composição dos serviços disponibilizados por nuvens públicas, privadas e de terceiros com orquestração entre essas plataformas.

#### • **Sistemas Operacionais**

Do inglês, Operating Systems (OS), são interfaces gráficas que permitem a nós humanos, nos comunicarmos com os componentes, hardware dos computadores.

Outro papel importante é gerenciar os recursos do sistema (identificar qual programa recebe mais recurso de processador e memória criação e uso do sistema de arquivos), fornecendo uma interface entre o computador e o usuário.

```
Award Modular BIOS v6.00PG, An Energy Star Ally
Copyright (C) 1984-2007, Award Software, Inc.

Intel X38 BIOS for X38-DQ6 F4

Main Processor : Intel(R) Core(TM)2 Extreme CPU X9650 @ 4.00GHz(333x12)
<CPUID:0676 Patch ID:0000>
Memory Testing : 2096064K OK

Memory Runs at Dual Channel Interleaved
IDE Channel 0 Slave : WDC WD3200AAJS-00RYA0 12.01B01
IDE Channel 1 Slave : WDC WD3200AAJS-00RYA0 12.01B01

Detecting IDE drives ...
IDE Channel 4 Master : None
IDE Channel 4 Slave : None
IDE Channel 5 Master : None
IDE Channel 5 Slave : None

<DEL>:BIOS Setup <F9>:XpressRecoveryZ <F12>:Boot Menu <End>:Qf Flash
09/19/2007-X38-ICH9-6A790G0QC-00
```

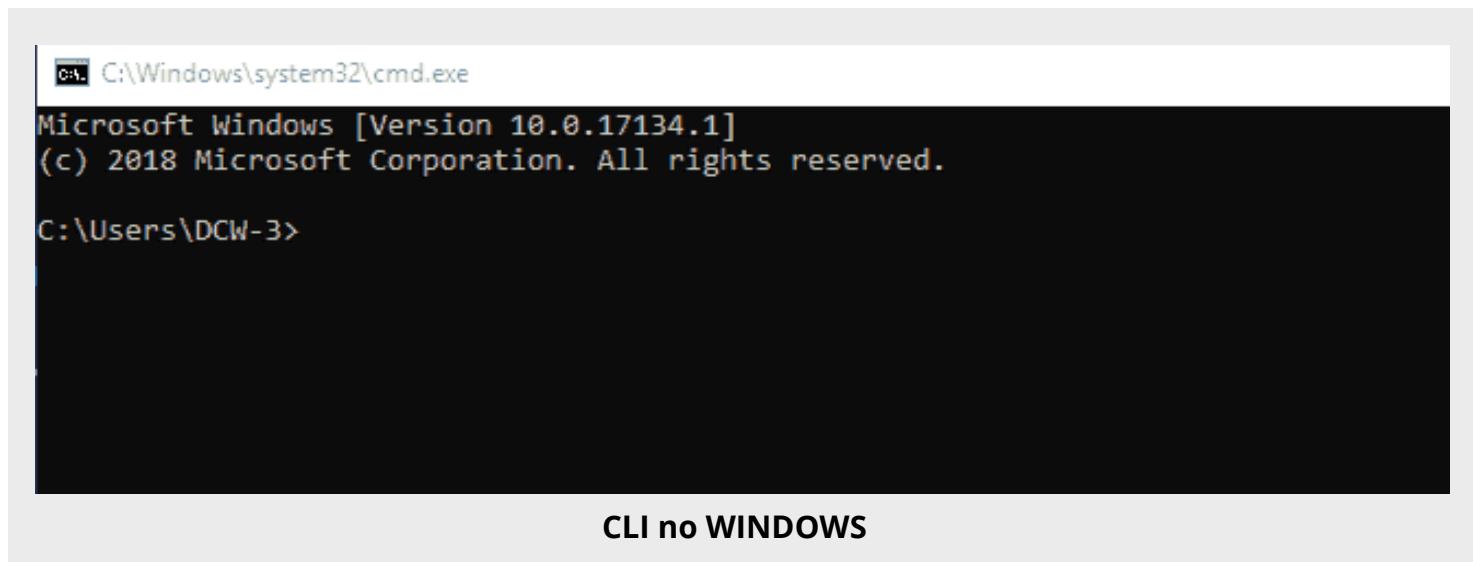
O sistema operacional (vamos chamá-lo de SO) possa ser executado imediatamente após a máquina ser ligada, na maioria dos desktops ele é executado após outro programa, armazenado na em uma memória não-volátil chamado de ROM (Read Only Memory) chamado BIOS, Basic Input Output System (Sistema Básico de Entrada e Saída) num processo chamado "bootstrapping". Nesta etapa, a BIOS faz uma série de verificações do hardware como verificação de novos hardwares, energia, é realizado alguns testes de componentes como monitor, teclado, mouse, e etc.

Após os testes de hardware, a próxima etapa é verificar o boot do sistema, ou seja, se o SO está válido e consistente em alguma unidade de armazenamento, SSD, .M2, rede, HD, entre outros e a partir daí, o sistema operacional tem autonomia para acesso e gestão dos recursos, conforme ações do usuário.

Os computadores atualmente já são adquiridos com o SO instalado, sendo os principais Windows (Microsoft), Linux e Mac (Apple). Mas nem sempre foi assim. Há várias gerações na história dos computadores até chegar na geração atual e que a cada geração os SOs tornam-se melhores do ponto de vista de gestão de recursos, usabilidade, tamanho, performance.

## CLI

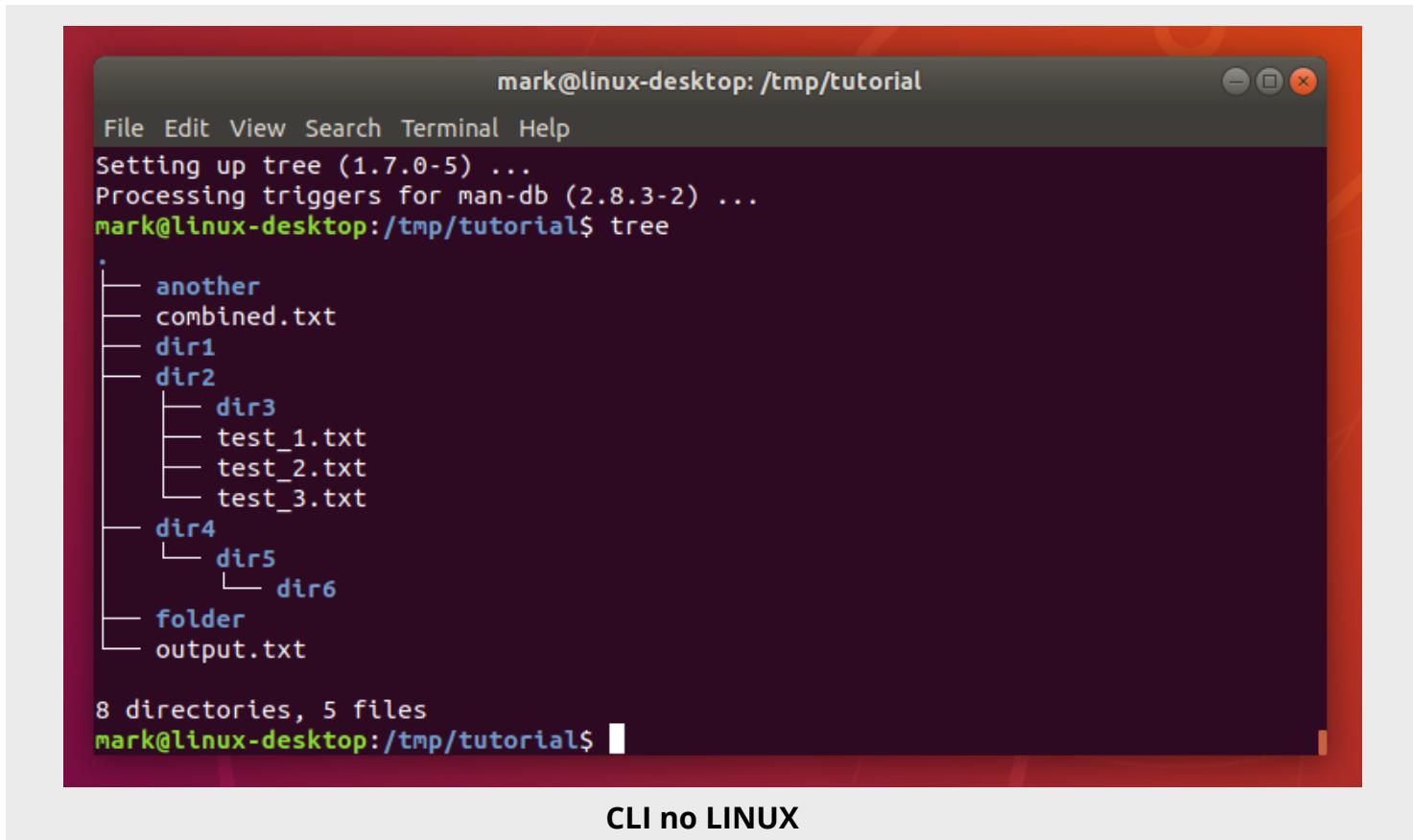
O CLI ou Command Line Interface (Interface de Linha de Comando) é um programa que permite que os usuários digitem comandos de texto dando instruções a uma máquina para realizar alguma função específica.



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.17134.1]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\DCW-3>
```

**CLI no WINDOWS**



```
mark@linux-desktop: /tmp/tutorial
File Edit View Search Terminal Help
Setting up tree (1.7.0-5) ...
Processing triggers for man-db (2.8.3-2) ...
mark@linux-desktop:/tmp/tutorial$ tree
.
├── another
├── combined.txt
├── dir1
├── dir2
│   ├── dir3
│   ├── test_1.txt
│   ├── test_2.txt
│   └── test_3.txt
└── dir4
    └── dir5
        └── dir6
    └── folder
    └── output.txt

8 directories, 5 files
mark@linux-desktop:/tmp/tutorial$
```

**CLI no LINUX**

- **Front-end e Back-end**

Em ciência da computação, front-end, interface frontal ou parte frontal e back-end, parte secundária, parte de suporte ou parte de retaguarda são termos generalizados que se referem às etapas inicial e final de um processo. O front-end é responsável por coligir a entrada do usuário em várias formas e processá-la para adequá-la a uma especificação em que o back-end a possa utilizar.

Algumas tecnologias para web front-end e back-end:

**Front-end**

- React
- XHTML
- HTML5
- CSS
- Javascript
- AJAX
- jQuery
- CFML
- AngularJS
- Angular
- Vue

**Back-end**

- PHP
- C
- C++
- Node.js
- Ruby on Rails
- Python
- Java
- JSP
- .Net
- C#
- VB
- Perl
- Golang

## AULA 2:

**Temas abordados: O navegador.**

#browser #barradeendereços #protocolo #caminhoderede #cache #dns

- **Os browsers**

São programas que, intermediados pelo SO, permitem ao usuário inserir os endereços das páginas de algum site e “navegar” através dos hiperlinks, desde que a conexão com a internet esteja estabelecida.

Para transferir esses dados para você, os navegadores utilizam o protocolo de transferência de hipertexto (HTTP). Ele é o responsável por definir como as informações chegarão aos usuários de forma consistente, em qualquer browser e em qualquer parte do planeta.

Os dez principais navegadores utilizados hoje são: Firefox, Google Chrome, Microsoft Edge, Apple Safari, Opera, Brave, Vivaldi, DuckDuckGo, Chromium e Epic. Atualmente os navegadores modernos conseguem, além de traduzirem os documentos HTML, visualizar arquivos no formato .pdf, visualizar imagens

Estrutura da URL

**https://www.exemplo.com:443/principal.htm  
l?key1=value1&key2=value2**

**Protocolo**

**Sub domínio**

**Domínio**

**Porta**

**Path**

**Inicio Query**

**Parâmetros**

### **Cache**

Uma cache é um bloco de memória para o armazenamento temporário de dados que possuem uma grande probabilidade de serem utilizados novamente.

Uma definição mais simples de cache poderia ser: uma área de armazenamento temporária onde os dados frequentemente acedidos são armazenados para acesso rápido. [1]

[1] <https://aws.amazon.com/pt/caching/>

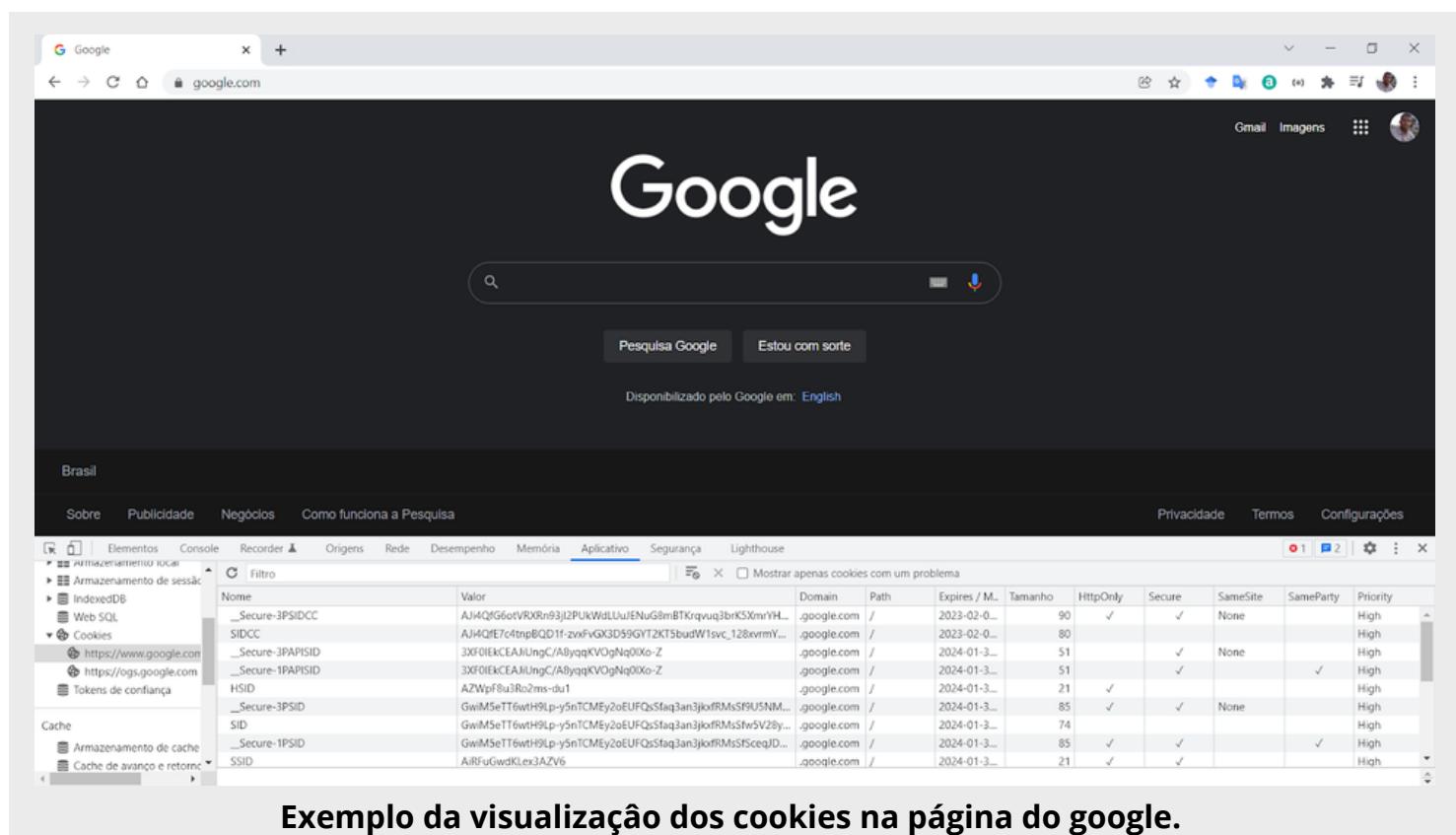
### **Cookies**

Os cookies possuem vários nomes como HTTP cookie, a web cookie, Internet cookie ou browser cookie. O termo se refere aos dados que os navegadores recebem e enviam novamente para os servidores que os correspondem sem alterá-lo. Os cookies não são capazes de executar códigos maliciosos como vírus, malwares, ransomwares entre outros.

Como os dados em um cookie não mudam quando ele vai e volta, não tem como afetar o funcionamento do seu computador.

No entanto, alguns vírus e malwares podem estar disfarçados como cookies. Por exemplo, “supercookies” podem ser uma preocupação potencial de segurança e muitos navegadores oferecem uma maneira de bloqueá-los. Um “cookie zumbi” é um cookie que se recria depois de ser excluído, tornando os cookies zumbis difíceis de gerenciar. Os cookies de rastreamento de terceiros também podem causar problemas de segurança e privacidade, pois tornam mais fácil para as partes que você não consegue identificar ver para onde você está indo e o que está fazendo online. [2]

[2] <https://us.norton.com/internetsecurity-privacy-what-are-cookies.html>

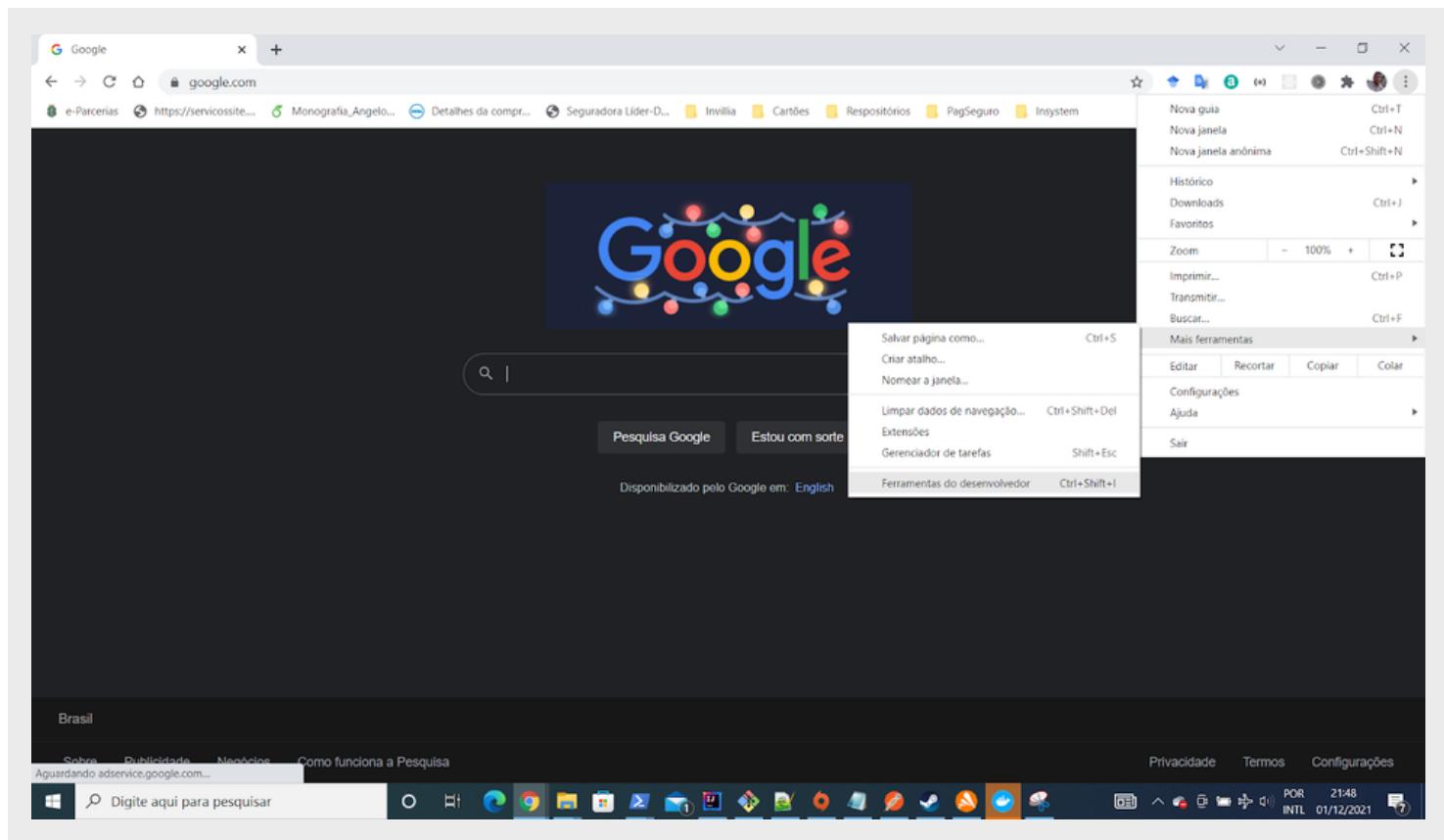


The screenshot shows the Google DevTools Network tab for the URL <https://www.google.com>. The 'Cookies' section is selected, displaying a list of cookies:

Nome	Valor	Domain	Path	Expires / M.	Tamanho	HttpOnly	Secure	SameSite	SameParty	Priority
__Secure-3PSIDCC	AJiAQfG6oIvRXRn93j2PUkWdLUuENuG8mBTkrqvuj3brK5XmrYH...	.google.com	/	2023-02-0...	90	✓	✓	None		High
SIDCC	AJi4QfE7c4tnpBCQD1f-zxvFvGX3DS9gYT2KT5budiW1svc_128xvmvY...	.google.com	/	2023-02-0...	80			None		High
__Secure-3PAPISID	3XF0fEkCEAJUnqC/A8yqqKV0qNq0Xo-Z	.google.com	/	2024-01-3...	51		✓	None		High
__Secure-1PAPISID	3XF0fEkCEAJUnqC/A8yqqKV0qNq0Xo-Z	.google.com	/	2024-01-3...	51		✓		✓	High
HSID	AZWpf8u3Ro2ms-du1	.google.com	/	2024-01-3...	21	✓				High
__Secure-3PSID	GwIM5eTT6wtH9lp-y5nTCMEy2oEUFQsStaq3an3jkofRMssf9USNM...	.google.com	/	2024-01-3...	85	✓	✓	None		High
SID	GwIM5eTT6wtH9lp-y5nTCMEy2oEUFQsStaq3an3jkofRMssf9USNM...	.google.com	/	2024-01-3...	74					High
__Secure-TPSID	GwIM5eTT6wtH9lp-y5nTCMEy2oEUFQsStaq3an3jkofRMssf5ceaJD...	.google.com	/	2024-01-3...	85	✓	✓		✓	High
SSID	AiRFuGwdkLex3AZV6	.google.com	/	2024-01-3...	21	✓	✓			High

**Exemplo da visualização dos cookies na página do google.**

### Ferramentas de Desenvolvedor



Caminho para acessar as Ferramentas do Desenvolvedor no Google Chrome

Todo navegador web moderno inclui um poderoso conjunto de ferramentas para desenvolvedores. Essas ferramentas fazem muitas coisas, desde inspecionar o HTML, CSS, JavaScript e quais recursos foram requeridos até mostrar quanto tempo a página precisou para carregar.

### DNS

O Domain Name Service é um serviço, um pouco desconhecido, que é responsável por fazer a internet funcionar da maneira como estamos acostumados a utilizar.

O DNS (Domain Name System – Sistema de nome de domínio) converte nomes de domínio legíveis por humanos (por exemplo, [www.digitalcollege.com.br](http://www.digitalcollege.com.br)) em endereços IP legíveis por máquina (por exemplo, 192.0.2.44).

Todos os computadores da internet, abrangendo de smartphones ou laptops a servidores que distribuem conteúdo para grandes websites do comércio, se encontram e se comunicam entre si usando números. Esses números são conhecidos como endereços IP. Ao abrir um navegador e acessar um site, você não precisará lembrar-se de um longo número nem digitá-lo. Em vez disso, você poderá informar um nome de domínio, como exemplo.com, e ainda assim encontrar o que deseja. [3]

No início da internet as poucas máquinas que existiam eram todas acessíveis via IP. Então, para acessar alguma máquina, bastava apenas saber o endereço IP do servidor e pronto.

Imagine se atualmente tivéssemos ter que saber o número de IP de cada página que vamos acessar no navegador? Acho que não ficaríamos tanto tempo navegando. Assim, é mais humano saber o nome do que um conjunto de números aleatórios. [3]

[3]<https://aws.amazon.com/pt/route53/what-is-dns/>

## HTML

A Linguagem de Marcação de HiperTexto, do contrário que muitos pensam, não é uma linguagem de programação e sim, como o próprio nome já diz, uma linguagem de marcação de blocos para construção de páginas da web que podem ser visualizadas, por exemplo, pelos navegadores (browsers) ou via CLI (Interface Command Line - Interface de Linha de comando).

O processo de renderização é feito por um sistema chamado rendering engine, este sistema recebe o que está vindo da camada de rede, interpreta e transforma em elementos visuais que serão carregados na tela para o usuário.

Para fazer isso, a rendering engine passa pelos passos de parse de HTML para construção da árvore DOM (DOM tree, do inglês Document Object Model), construção da Render tree, Layout da Render tree e Painting da Render tree. A montagem da árvore é realizada através de um criador de tokens, chamado de **tokenizer** que é responsável por identificar cada tag ou texto dentro do conjunto de caracteres e fazer a construção da árvore. A especificação do HTML mostra como cada token deve ser identificado e como eles devem ser inseridos na árvore do DOM.

Se durante o processo de construção da árvore o parse identificar um HTML mal formatado ou incompleto ele o conserta, por exemplo, se um documento não possuir a tag <body>, ele recebe um ao passar pela máquina de estados que está implementada no processo de parse.

A estrutura criada a partir da criação da árvore é chamada de DOM Tree, possuindo a finalidade de representar o HTML em forma de árvore, para possibilitar a manipulação dos elementos isoladamente, por isso, chamamos de DOM Tree, ou árvore do DOM.

Vamos ver o código abaixo:

```
<html>
  <head>
    <link href="style.css"
      rel="stylesheet">
  </head>
  <body>
    <p>Minha <span>primeira</span>
      página!</p>
    <div></div>
  </body>
</html>
```

- **<html>**: tag principal
- **<head>**: tag configurações da página
- **<body>**: tag do conteúdo a ser exibido na página

## CSS

Assim como o HTML o CSS é interpretado utilizando tokens e também geram um modelo de árvores. O processo de criação de tokens é exibido como regular expressions, ou regex. O modelo de árvore que é gerado depois da criação dos tokens é chamado de CSSOM, que significa CSS Object Model, ou modelo de objetos CSS, em português. É uma árvore assim como a DOM Tree, o que facilita para o navegador no momento de aplicar as regras.

Vamos ver o código abaixo:

```
body { font-size: 16px }
p { font-weight: bold }
span { color: red }
p span { display: none }
img { float: right }
```

No CSS, as regras são hierárquicas, ou seja, uma regra aplicada à tag <body> é aplicada também para todos os elementos filhos. Essa regra pode ser sobreposta por eles, por isso a árvore é tão importante.

## Incluindo Arquivos externos

Durante o parse, o browser pode encontrar referências no HTML à arquivos externos como de código em .js(JavaScript, tag <script>), .jpeg, .png ou outro tipo de arquivo de imagem (tag <img>).

Esse carregamento pode ser feito via link do arquivo no file system ou por um link onde o arquivo será baixado. Uma boa prática é que esses arquivos estejam em um servidor de CDN(Content Delivery Network), ou seja, uma rede de entrega de conteúdo que está distribuída em vários servidores ao redor do mundo, sendo que o download é feito pelo servidor mais próximo, melhorando a performance para requisitar os arquivos.

## AULA 3

- #MODELOOSI #HOSTS #IP #DNS #ROTEADORES

### O que é o modelo OSI?

O modelo Interconexão de Sistemas Abertos (OSI) é um modelo conceitual criado pela Organização Internacional de Normalização que permite que diversos sistemas de comunicação se comuniquem usando protocolos padronizados. Em poucas palavras, o OSI fornece um padrão para que diferentes sistemas de computadores possam se comunicar.

O modelo OSI pode ser visto como uma linguagem universal para Redes de computadores. É baseado no conceito de dividir um sistema de comunicação em sete camadas abstratas, empilhadas umas sobre as outras.

imagem modelo osi 1

Cada camada do modelo OSI lida com uma tarefa específica e se comunica com as camadas acima e abaixo dela.

Por que o modelo OSI é importante?

Embora a internet moderna não siga estritamente o modelo OSI (segue mais de perto um conjunto mais simples de protocolos da internet), o modelo continua sendo muito útil para solucionar problemas de Rede. Seja uma pessoa que não consegue colocar seu notebook na internet ou um site que esteja inativo para milhares de usuários, o modelo OSI pode ajudar a resolver e isolar a fonte do problema. Se o problema puder ser reduzido a uma camada específica do modelo, muito trabalho desnecessário poderá ser evitado.

### Quais são as sete camadas do modelo OSI?

As sete camadas de abstração do modelo OSI podem ser definidas como se segue, de cima para baixo:

Imagen modelo osi 2

### 7. Camada de aplicativos

Essa é a única camada que interage diretamente com os dados do usuário. Os softwares aplicativos, como navegadores web e clientes de e-mail, dependem da camada de aplicação para iniciar as comunicações. Mas é preciso deixar claro que os softwares aplicativos clientes não fazem parte da camada de aplicação, que, na verdade, é responsável pelos protocolos e manipulação de dados dos quais o software depende para apresentar dados significativos ao usuário. Os protocolos da camada de aplicação incluem o HTTP e o SMTP (Simple Mail Transfer Protocol, um dos protocolos que permite a comunicação por e-mail).

Imagen modelo osi 3

## 6. Camada de apresentação

Essa camada é a principal responsável pela preparação dos dados para que possam ser usados pela camada de aplicação; em outras palavras, a camada 6 torna os dados apresentáveis para que os aplicativos os consumam. A camada de apresentação é responsável pela tradução, criptografia e compactação dos dados.

Dois dispositivos de comunicação que se comunicam podem usar métodos de codificação diferentes; por isso, a camada 6 é responsável pela tradução dos dados de entrada em uma sintaxe que a camada de aplicação do dispositivo receptor possa entender.

Se os dispositivos se comunicarem por meio de uma conexão criptografada, a camada 6 será responsável por adicionar a criptografia na extremidade do remetente e decodificar a criptografia na extremidade do destinatário, podendo, assim, apresentar dados não criptografados e legíveis à camada de aplicação. Finalmente, a camada de apresentação também é responsável por compactar os dados recebidos da camada de aplicação antes de entregá-los à camada 5. Isso ajuda a aumentar a velocidade e a eficiência da comunicação ao minimizar a quantidade de dados que serão transferidos.

imagem modelo osi 4

## 5. Camada de sessão

Essa é a camada responsável pela abertura e fechamento da comunicação entre os dois dispositivos. O tempo decorrido entre o momento em que a comunicação é aberta e fechada é conhecido como "sessão". A camada de sessão garante que a sessão permaneça aberta pelo tempo necessário para transferir todos os dados que estão sendo trocados e, em seguida, fecha imediatamente a sessão para evitar o desperdício de recursos.

A camada de sessão também sincroniza a transferência de dados com pontos de verificação. Por exemplo, se um arquivo de 100 megabytes estiver sendo transferido, a camada de sessão poderá definir um ponto de verificação a cada 5 megabytes. No caso de uma desconexão ou falha após a transferência de 52 megabytes, a sessão pode ser retomada a partir do último ponto de verificação, o que significa que apenas mais 50 megabytes de dados precisam ser transferidos. Sem os pontos de verificação, a transferência inteira teria que começar novamente do zero.

imagem modelo osi 5

#### **4. Camada de transporte**

A camada 4 é responsável pela comunicação de ponta a ponta entre os dois dispositivos. Isso inclui pegar os dados da camada de sessão e dividi-los em porções chamadas segmentos antes de enviá-los para a camada 3. A camada de transporte no dispositivo receptor é responsável por remontar os segmentos em dados que a camada de sessão possa consumir. A camada de transporte também é responsável pelo controle de fluxo e pelo controle de erros. O controle de fluxo determina uma velocidade de transmissão ideal para garantir que um remetente com uma conexão rápida não sobrecarregue um receptor com uma conexão lenta. A camada de transporte executa o controle de erros no lado do receptor, garantindo que os dados recebidos estejam completos e solicitando uma retransmissão caso não estejam.

imagem modelo osi 6

#### **3. Camada de Rede**

A camada de rede é responsável por facilitar a transferência de dados entre duas redes diferentes. Se os dois dispositivos que estão se comunicando estiverem na mesma rede, a camada de rede será desnecessária. A camada de rede divide os segmentos da camada de transporte em unidades menores denominadas pacotes no dispositivo remetente e remonta esses pacotes no dispositivo receptor. A camada de rede também encontra o melhor caminho físico para que os dados cheguem ao seu destino, o que é conhecido como "roteamento".

imagem modelo osi 7

#### **2. Camada de enlace de dados**

A camada de enlace de dados é muito semelhante à camada de rede, a não ser pelo fato de que a camada de enlace de dados facilita a transferência de dados entre dois dispositivos na MESMA rede. A camada de enlace de dados pega os pacotes da camada de rede e os divide em pedaços menores denominados "quadros". Como a camada de rede, a camada de enlace de dados também é responsável pelo controle de fluxo e pelo controle de erros na comunicação (a camada de transporte faz o controle de fluxo e o controle de erros para comunicações inter-rede).

imagem modelo osi 8

#### **1. Camada física**

Essa camada inclui o equipamento físico envolvido na transferência de dados, como cabos e comutadores. Essa também é a camada em que os dados são convertidos em um fluxo de bits, que é uma sequência de 1s e 0s. A camada física de ambos os dispositivos também precisa aceitar de comum acordo uma convenção de sinal para que se possa distinguir os 1s dos 0s em ambos os dispositivos.

Como os dados fluem através do modelo OSI

Para que informações legíveis por humanos sejam transferidas por uma Rede de um dispositivo para outro, os dados devem percorrer as sete camadas do modelo OSI na ordem decrescente no dispositivo que os envia e, em seguida, percorrer as sete camadas na ordem crescente na extremidade que os recebe.

Por exemplo: o Angelo quer enviar um e-mail ao Daniel. O Angelo escreve sua mensagem no aplicativo de e-mail do seu notebook e, em seguida, pressiona "enviar". Seu aplicativo de e-mail passa sua mensagem de e-mail para a camada de aplicação, que seleciona um protocolo (SMTP) e passa os dados para a camada de apresentação. A camada de apresentação compacta os dados que, em seguida, chegam à camada de sessão, que inicia a sessão de comunicação.

Em seguida os dados chegam à camada de transporte do remetente, onde são segmentados; esses segmentos são divididos em pacotes na camada de rede e os pacotes, por sua vez, são divididos em quadros na camada de enlace de dados. A camada de enlace de dados a seguir entrega esses quadros à camada física, que converte os dados em um fluxo de bits de 1s e 0s e os envia por meio de uma mídia física, como um cabo.

Assim que o computador do Daniel recebe o fluxo de bits por meio de uma mídia física (como o seu wi-fi), os dados fluem através da mesma série de camadas em seu dispositivo, mas na ordem inversa. Primeiro, a camada física converte o fluxo de bits de 1s e 0s em quadros, que são passados para a camada de enlace de dados. A camada de enlace de dados remonta os quadros em pacotes para a camada de rede. A camada de rede cria segmentos remontando os pacotes para a camada de transporte, que remonta os segmentos em um simples dado.

Os dados em seguida fluem para a camada de sessão do receptor, que os transmite para a camada de apresentação e em seguida encerra a sessão de comunicação. A camada de apresentação então remove a compactação e passa os dados brutos para a camada de aplicação. A camada de aplicação alimenta o software de e-mail do Daniel com dados legíveis por humanos, permitindo que ela leia o e-mail do Angelo na tela do seu notebook.

- **Introdução ao TCP/IP**

### **Um visão geral do protocolo TCP/IP**

Para que os computadores de uma rede possam trocar informações entre si é necessário que todos os computadores adotem as mesmas regras para o envio e o recebimento de informações. Este conjunto de regras é conhecido como Protocolo de comunicação. Falando de outra maneira podemos afirmar: "Para que os computadores de uma rede possam trocar informações entre si é necessário que todos estejam utilizando o mesmo protocolo de comunicação". No protocolo de comunicação estão definidas todas as regras necessárias para que o computador de destino, "entenda" as informações no formato que foram enviadas pelo computador de origem. Dois computadores com diferentes protocolos instalados, não serão capazes de estabelecer uma comunicação e nem serão capazes de trocar informações.

Antes da popularização da Internet existiam diferentes protocolos sendo utilizados nas redes das empresas. Os mais utilizados eram os seguintes:

- **TCP/IP**
- **NETBEUI**
- **IPX/SPX**
- **Apple Talk**

Se colocarmos dois computadores ligados em rede, um com um protocolo, por exemplo o TCP/IP e o outro com um protocolo diferente, por exemplo NETBEUI, estes dois computadores não serão capazes de estabelecer comunicação e trocar informações entre si. Por exemplo, o computador com o protocolo NETBEUI instalado, não será capaz de acessar uma pasta ou uma Impressora compartilhada no computador com o protocolo TCP/IP instalado.

À medida que a Internet começou, a cada dia, tornar-se mais popular, com o aumento exponencial do número de usuários, o protocolo TCP/IP passou a tornar-se um padrão de fato, utilizando não só na Internet, como também nas redes internas das empresas, redes estas que começavam a ser conectadas à Internet. Como as redes internas precisavam conectar-se à Internet, tinham que usar o mesmo protocolo da Internet, ou seja: TCP/IP.

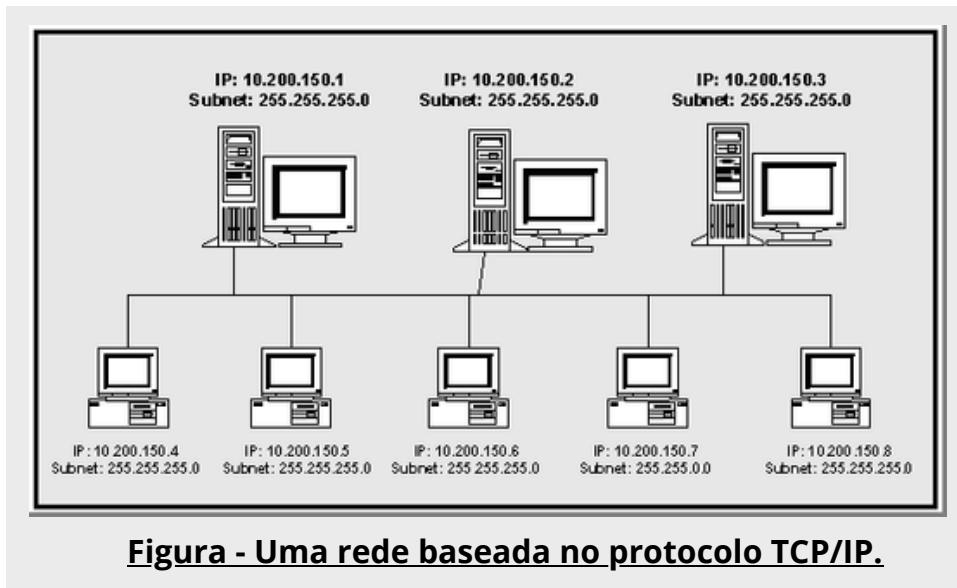
Dos principais Sistemas Operacionais do mercado, o UNIX sempre utilizou o protocolo TCP/IP como padrão. O Windows dá suporte ao protocolo TCP/IP desde as primeiras versões, porém, para o Windows, o TCP/IP somente tornou-se o protocolo padrão a partir do Windows 2000.

O que temos hoje, na prática, é a utilização do protocolo TCP/IP na esmagadora maioria das redes. Se durante a instalação, o Windows detectar a presença de uma placa de rede, automaticamente será sugerida a instalação do protocolo TCP/IP.

Agora passaremos a estudar algumas características do protocolo TCP/IP. Veremos que cada equipamento que faz parte de uma rede baseada no TCP/IP tem alguns parâmetros de configuração que devem ser definidos, para que o equipamento possa comunicar-se com sucesso na rede e trocar informações com os demais equipamentos da rede.

### **Configurações do protocolo TCP/IP para um computador em rede**

Quando utilizamos o protocolo TCP/IP como protocolo de comunicação em uma rede de computadores, temos alguns parâmetros que devem ser configurados em todos os equipamentos que fazem parte da rede (computadores, servidores, hubs, switchs, impressoras de rede, etc). Na Figura a seguir temos uma visão geral de uma pequena rede baseada no protocolo TCP/IP:



**Figura - Uma rede baseada no protocolo TCP/IP.**

No exemplo da Figura 1 temos uma rede local para uma pequena empresa. Esta rede local não está conectada a outras redes ou à Internet. Neste caso cada computador da rede precisa de, pelo menos, dois parâmetros configurados:

- **Número IP**
- **Máscara de sub-rede**

O Número IP é um número no seguinte formato: x.y.z.w

Ou seja, são quatro números separados por ponto. Não podem existir duas máquinas, com o mesmo número IP, dentro da mesma rede. Caso eu configure um novo equipamento com o mesmo número IP de uma máquina já existente, será gerado um conflito de Número IP e um dos equipamentos, muito provavelmente o novo equipamento que está sendo configurado, não conseguirá se comunicar com a rede. O valor máximo para cada um dos números (x, y, z ou w) é 255.

Uma parte do Número IP (1, 2 ou 3 dos 4 números) é a identificação da rede, a outra parte é a identificação da máquina dentro da rede. O que define quantos dos quatro números fazem parte da identificação da rede e quantos fazem parte da identificação da máquina é a máscara de sub-rede (subnet mask). Vamos considerar o exemplo de um dos computadores da rede da Figura 1:

- **Número IP: 10.200.150.1**
- **Máscara de Sub-rede: 255.255.255.0**

As três primeiras partes da máscara de sub-rede (subnet) iguais a 255 indicam que os três primeiros números representam a identificação da rede e o último número é a identificação do equipamento dentro da rede. Para o nosso exemplo teríamos a rede: **10.200.150**, ou seja, todos os equipamentos do nosso exemplo fazem parte da rede **10.200.150** ou, em outras palavras, o número IP de todos os equipamentos da rede começam com **10.200.150**.

Neste exemplo, onde estamos utilizando os três primeiros números para identificar a rede e somente o quarto número para identificar o equipamento, temos um limite de 254 equipamentos que podem ser ligados neste rede. Observe que são 254 e não 256, pois o primeiro número - 10.200.150.0 e o último número - 10.200.250.255 não podem ser utilizados como números IP de equipamentos de rede. O primeiro é o próprio número da rede: **10.200.150.0** e o último é o endereço de Broadcast: **10.200.150.255**. Ao enviar uma mensagem para o endereço de Broadcast, todas as máquinas da rede receberão a mensagem. Nas próximas partes deste tutorial, falaremos um pouco mais sobre Broadcast.

Com base no exposto podemos apresentar a seguinte definição:

**"Para se comunicar em uma rede baseada no protocolo TCP/IP, todo equipamento deve ter, pelo menos, um número IP e uma máscara de sub-rede, sendo que todos os equipamentos da rede devem ter a mesma máscara de sub-rede".**

No exemplo da figura anterior observe que o computador com o IP 10.200.150.7 está com uma máscara de sub-rede diferente da máscara de sub-rede dos demais computadores da rede. Este computador está com a máscara: 255.255.0.0 e os demais computadores da rede estão com a máscara de sub-rede 255.255.255.0. Neste caso é como se o computador com o IP 10.200.150.7 pertencesse a outra rede. Na prática o que irá acontecer é que este computador não conseguirá se comunicar com os demais computadores da rede, por ter uma máscara de sub-rede diferente dos demais. Este é um dos erros de configuração mais comuns.

Se a máscara de sub-rede estiver incorreta, ou seja, diferente da máscara dos demais computadores da rede, o computador com a máscara de sub-rede incorreta não conseguirá comunicar-se na rede.

Na Tabela a seguir temos alguns exemplos de máscaras de sub-rede e do número máximo de equipamentos em cada uma das respectivas redes.

Tabela: Exemplos de máscara de sub-rede.

MÁSCARA	Número de equipamentos na rede
255.255.250	254
255.255.0.0	65.534
255.0.0.0	16.777.214

Quando a rede está isolada, ou seja, não está conectada à Internet ou a outras redes externas, através de links de comunicação de dados, apenas o número IP e a máscara de sub-rede são suficientes para que os computadores possam se comunicar e trocar informações.

A conexão da rede local com outras redes é feita através de links de comunicação de dados. Para que essa comunicação seja possível é necessário um equipamento capaz de enviar informações para outras redes e receber informações destas redes. O equipamento utilizado para este fim é o Roteador. Todo pacote de informações que deve ser enviado para outras redes deve, obrigatoriamente, passar pelo Roteador.

Todo pacote de informação que vem de outras redes também deve, obrigatoriamente, passar pelo Roteador. Como o Roteador é um equipamento de rede, este também terá um número IP. O número IP do roteador deve ser informado em todos os demais equipamentos que fazem parte da rede, para que estes equipamentos possam se comunicar com as redes externas. O número IP do Roteador é informado no parâmetro conhecido como Default Gateway. Na prática quando configuramos o parâmetro Default Gateway, estamos informando o número IP do Roteador.

Quando um computador da rede tenta se comunicar com outros computadores/servidores, o protocolo TCP/IP faz alguns cálculos utilizando o número IP do computador de origem, a máscara de sub-rede e o número IP do computador de destino (veremos estes cálculos em detalhes nas próximas lições deste curso). Se, após feitas as contas, for concluído que os dois computadores fazem parte da mesma rede, os pacotes de informação são enviados para o barramento da rede local e o computador de destino captura e processa as informações que lhe foram enviadas. Se, após feitas as contas, for concluído que o computador de origem e o computador de destino, fazem parte de redes diferentes, os pacotes de informação são enviados para o Roteador (número IP configurado como Default Gateway) e o Roteador é o responsável por achar o caminho (a rota) para a rede de destino.

Com isso, para equipamentos que fazem parte de uma rede, baseada no protocolo TCP/IP e conectada a outras redes ou a Internet, devemos configurar, no mínimo, os seguintes parâmetros:

- **Número IP**
- **Máscara de sub-rede**
- **Default Gateway**

Em redes existem outros parâmetros que precisam ser configurados. Um dos parâmetros que deve ser informado é o número IP de um ou mais servidores DNS – Domain Name System. O DNS é o serviço responsável pela resolução de nomes. Toda a comunicação, em redes baseadas no protocolo TCP/IP é feita através do número IP. Por exemplo, quando vamos acessar o site: <https://digitalcollege.com.br/>, tem que haver uma maneira de encontrar o número IP do servidor onde fica hospedado o site. O serviço que localiza o número IP associado a um nome é conhecido como Servidor DNS. Por isso a necessidade de informarmos o número IP de pelo menos um servidor DNS, pois sem este serviço de resolução de nomes, muitos recursos da rede estarão indisponíveis, inclusive o acesso à Internet.

As configurações do protocolo TCP/IP podem ser definidas manualmente, isto é, configurando cada um dos equipamentos necessários com as informações do protocolo, como por exemplo o Número IP, Máscara de sub-rede, número IP do Default Gateway, número IP de um ou mais servidores DNS e assim por diante.

Esta é uma solução razoável para pequenas redes, porém pode ser um problema para redes maiores, com um grande número de equipamentos conectados. Para redes maiores é recomendado o uso do serviço DHCP – Dynamic Host Configuration Protocol. Os parâmetros são fornecidos quando o equipamento é inicializado e podem ser renovados em períodos definidos pelo Administrador. Com o uso do DHCP uma série de procedimentos de configuração podem ser automatizados, o que facilita a vida do Administrador e elimina uma série de erros.

Você pode verificar, facilmente, as configurações do protocolo TCP/IP que estão definidas para o seu computador Windows.

Para isso siga os seguintes passos:

1. Faça o logon com uma conta com permissão de Administrador.
  2. Abra o Prompt de comando: Iniciar -> Programas -> Acessórios -> Prompt de comando.
  3. Na janela do Prompt de comando digite o seguinte comando: **ipconfig /all** e pressione Enter.
  4. Serão exibidas as diversas configurações do protocolo TCP/IP, conforme indicado a seguir, no exemplo obtido a partir de um dos meus computadores que eu uso na rede da minha casa:

## Configuração de IP do Windows

```
Nome do host . . . . . : servidor01
Sufixo DNS primário. . . . . : groza.com
Tipo de nó . . . . . : híbrido
Roteamento de IP ativado . . . . . : não
Proxy WINS ativado . . . . . : não
Lista de pesquisa de sufixo DNS. . . . . : groza.com
```

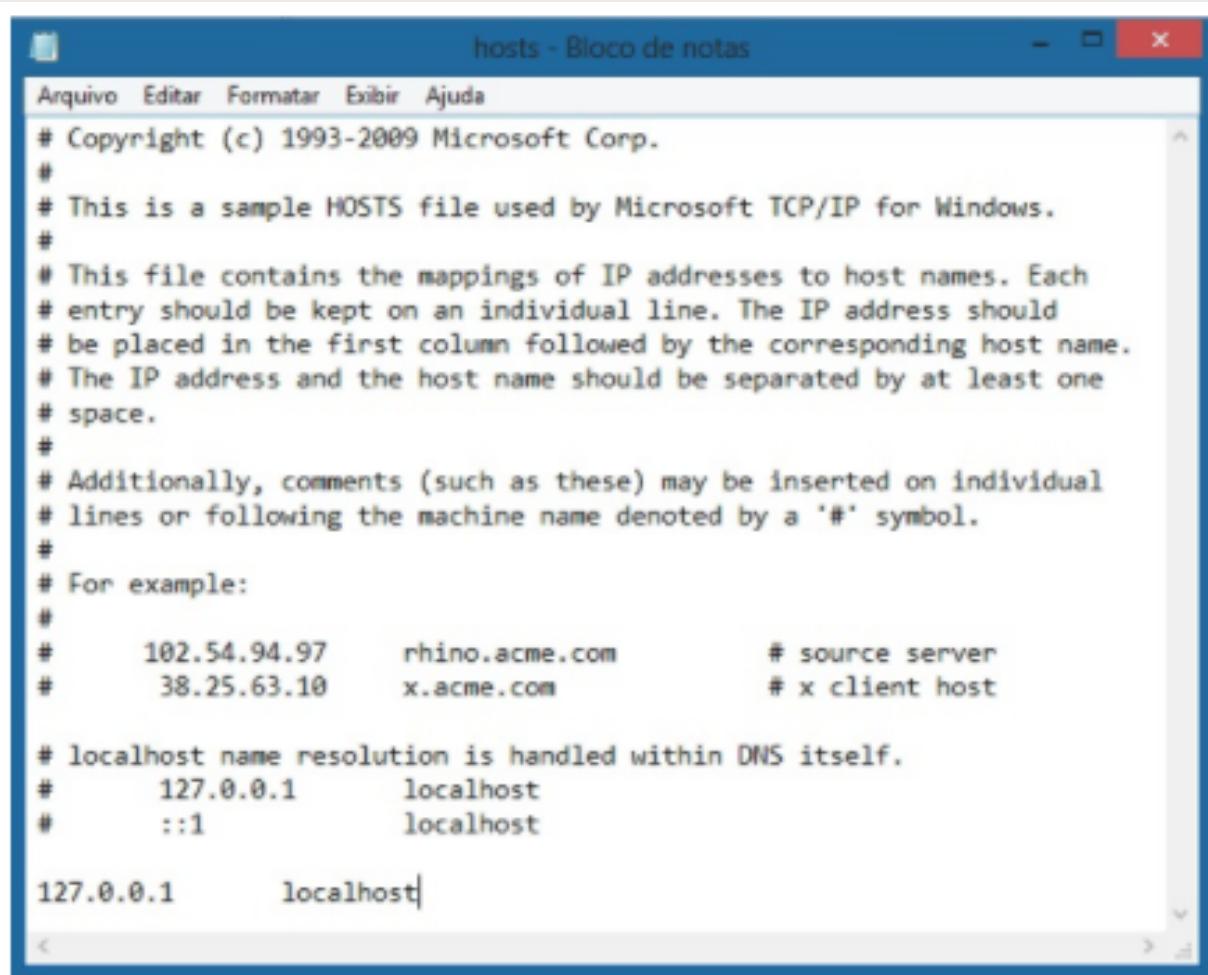
Adaptador Ethernet Conexão local:

O comando ipconfig exibe informações para as diversas interfaces de rede instaladas – placa de rede, modem, etc. No exemplo anterior temos uma única interface de rede instalada, a qual é relacionada com uma placa de rede Realtek RTL8139 Family PCI Fast Ethernet NIC. Observe que temos o número IP para dois servidores DNS e para um servidor WINS. Outra informação importante é o Endereço físico, mais conhecido como MAC-Address ou endereço da placa. O MAC-Address é um número que identifica a placa de rede. Os seis primeiros números/letras são uma identificação do fabricante da placa e os seis últimos uma identificação da placa. Não existem duas placas com o mesmo MAC-Address, ou seja, este endereço é único para cada placa de rede.

No exemplo da listagem a seguir, temos um computador com duas interfaces de rede. Uma das interfaces é ligada a placa de rede (Realtek RTL8029(AS) PCI Ethernet Adapter), a qual conecta o computador a rede local. A outra interface é ligada ao fax-modem (WAN (PPP/SLIP) Interface), o qual conecta o computador à Internet. Para o protocolo TCP/IP a conexão via Fax modem aparece como se fosse mais uma interface de rede, conforme pode ser conferido na listagem a seguir:

## O Arquivo hosts

Um dos passos na tradução do endereço IP (numérico) para um nome (alfanumérico) é o arquivo hosts, no qual a localização depende do sistema operacional. Lembrando que a funcionalidade não muda, apenas a sua localização no sistema de arquivos.



The screenshot shows a Windows Notepad window titled "hosts - Bloco de notas". The window contains the following text:

```
# Copyright (c) 1993-2009 Microsoft Corp.  
#  
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.  
#  
# This file contains the mappings of IP addresses to host names. Each  
# entry should be kept on an individual line. The IP address should  
# be placed in the first column followed by the corresponding host name.  
# The IP address and the host name should be separated by at least one  
# space.  
#  
# Additionally, comments (such as these) may be inserted on individual  
# lines or following the machine name denoted by a '#' symbol.  
#  
# For example:  
#  
#      102.54.94.97      rhino.acme.com      # source server  
#      38.25.63.10      x.acme.com          # x client host  
  
# localhost name resolution is handled within DNS itself.  
#      127.0.0.1      localhost  
#      ::1            localhost  
  
127.0.0.1      localhost|
```

- **GNU/Linux:** /etc/hosts.
- **Windows:** C:\Windows\System32\drivers\etc\hosts.

O conceito do arquivos hosts é bem simples, basicamente um de para do número IP para um determinado domínio.

```
root@cassia-virtual-machine: /home/cassia
GNU nano 2.2.6          Arquivo: /etc/hosts

127.0.0.1      localhost
127.0.1.1      cassia-virtual-machine

# The following lines are desirable for IPv6 capable hosts
::1      ip6-localhost ip6-loopback
fe80::1 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

[ 9 linhas lidas ]
^G Obter Ajuda ^O Gravar    ^R Ler o Arq ^W Pág Anter ^K Recort Txt ^C Pos Atual
^X Sair      ^J Justificar ^H Onde está? ^V Próx Pág ^U Colar Txt ^T Para Spell
```

## AULA 4

- #http #get #post #put #delete #statuscodes #TCP #UDP #HTTPS #TLS #SSL  
#certificado #chavepública #autoridadecertificadora #trocadechaves

### Um pouco sobre Pacotes e sobre os protocolos de Transporte

O TCP/IP, na verdade, é formado por um grande conjunto de diferentes protocolos e serviços de rede. O nome TCP/IP deriva dos dois protocolos mais importantes e mais utilizados, que são os seguintes:

- **IP:** É um protocolo de endereçamento, um protocolo de rede. Eu me arriscaria a afirmar que as principais funções do protocolo IP são endereçamento e roteamento, ou de uma maneira mais simples, fornecer uma maneira para identificar unicamente cada máquina da rede (endereço IP) e uma maneira de encontrar um caminho entre a origem e o destino (Roteamento).
- **TCP:** O TCP é um protocolo de transporte e executa importantes funções para garantir que os dados sejam entregues de uma maneira confiável, ou seja, sem que os dados sejam corrompidos ou alterados.

### TCP – Uma Visão Geral

O Transmission Control Protocol (TCP) é, sem dúvidas, um dos mais importantes protocolos da família TCP/IP. É um padrão definido na RFC 793, "Transmission Control Protocol (TCP)", que fornece um serviço de entrega de pacotes confiável e orientado por conexão. Ser orientado por conexão, significa que todos os aplicativos baseados em TCP como protocolo de transporte, antes de iniciar a troca de dados, precisam estabelecer uma conexão.

Na conexão são fornecidas, normalmente, informações de logon, as quais identificam o usuário que está tentando estabelecer a conexão.

Algumas características do TCP:

- **Garante a entrega de datagramas IP:** Esta talvez seja a principal função do TCP, ou seja, garantir que os pacotes sejam entregues sem alterações, sem terem sido corrompidos e na ordem correta. O TCP tem uma série de mecanismos para garantir esta entrega.
- **Executa a segmentação e reagrupamento de grandes blocos de dados enviados pelos programas e garante o sequenciamento adequado e entrega ordenada de dados segmentados:** Esta característica refere-se a função de dividir grandes arquivos em pacotes menores e transmitir cada pacote separadamente. Os pacotes podem ser enviados por caminhos diferentes e chegar fora de ordem. O TCP tem mecanismos para garantir que, no destino, os pacotes sejam ordenados corretamente, antes de serem entregues ao programa de destino.

- Verifica a integridade dos dados transmitidos usando cálculos de soma de verificação:** O TCP faz verificações para garantir que os dados não foram alterados ou corrompidos durante o transporte entre a origem e o destino.
- Envia mensagens positivas dependendo do recebimento bem-sucedido dos dados. Ao usar confirmações seletivas, também são enviadas confirmações negativas para os dados que não foram recebidos:** No destino, o TCP recebe os pacotes, verifica se estão OK e, em caso afirmativo, envia uma mensagem para a origem, confirmando cada pacote que foi recebido corretamente. Caso um pacote não tenha sido recebido ou tenha sido recebido com problemas, o TCP envia uma mensagem ao computador de origem, solicitando uma retransmissão do pacote. Com esse mecanismo, apenas pacotes com problemas terão que ser reenviados, o que reduz o tráfego na rede e agiliza o envio dos pacotes.
- Oferece um método preferencial de transporte de programas que devem usar transmissão confiável de dados baseada em sessões, como bancos de dados cliente/servidor e programas de correio eletrônico:** Ou seja, o TCP é muito mais confiável do que o UDP (conforme mostrarei mais adiante) e é indicado para programas e serviços que dependam de uma entrega confiável de dados.

### Funcionamento do TCP

O TCP baseia-se na comunicação ponto a ponto entre dois hosts de rede. O TCP recebe os dados de programas e processa esses dados como um fluxo de bytes. Os bytes são agrupados em segmentos que o TCP numera e sequencia para entrega. Estes segmentos são mais conhecidos como “**Pacotes**”.

Antes que dois hosts TCP possam trocar dados, deve primeiro estabelecer uma sessão entre si. Uma sessão TCP é inicializada através de um processo conhecido como um tree-way handshake (algo como Um Aperto de Mão Triplo). Esse processo sincroniza os números de sequência e oferece informações de controle necessárias para estabelecer uma conexão virtual entre os dois hosts.

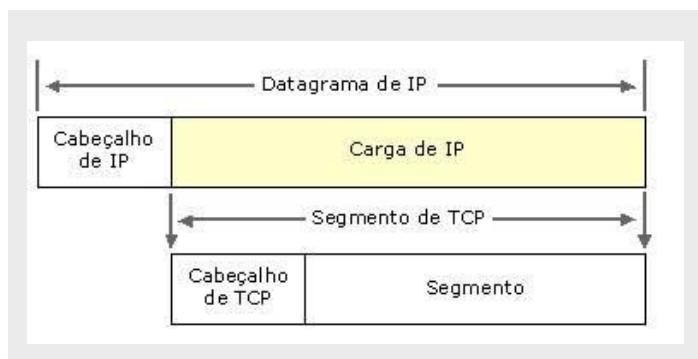
De uma maneira simplificada, o processo de tree-way handshake, pode ser descrito através dos seguintes passos:

- O computador de origem solicita o estabelecimento de uma sessão com o computador de destino:** Por exemplo, você utiliza um programa de FTP (origem) para estabelecer uma sessão com um servidor de FTP (destino).
- O computador de destino recebe a requisição, verifica as credenciais enviadas (tais como as informações de logon e senha) e envia de volta para o cliente, informações que serão utilizadas pelo cliente, para estabelecer efetivamente a sessão. As informações enviadas nesta etapa são importantes, pois é através destas informações que o servidor irá identificar o cliente e liberar ou não o acesso.

- O computador de origem recebe as informações de confirmação enviadas pelo servidor e envia estas confirmações de volta ao servidor. O servidor recebe as informações, verifica que elas estão corretas e estabelece a sessão. A partir deste momento, origem e destino estão autenticados e aptos a trocar informações usando o protocolo TCP. Se por algum motivo, as informações enviadas pela origem não estiverem corretas, a sessão não será estabelecida e uma mensagem de erro será enviada de volta ao computador de origem.

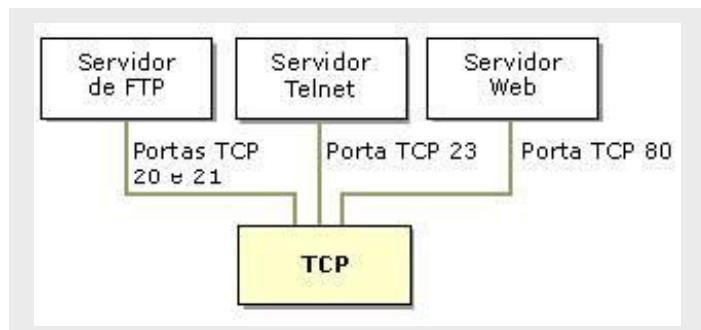
Depois de concluído o three-way handshake inicial, os segmentos são enviados e confirmados de forma seqüencial entre os hosts remetente e destinatário. Um processo de handshake semelhante é usado pelo TCP antes de fechar a conexão para verificar se os dois hosts acabaram de enviar e receber todos os dados.

Os segmentos TCP são encapsulados e enviados em datagramas IP, conforme apresentado na figura a seguir, obtida na ajuda do Windows 2000 Server:



### O conceito de Portas TCP

Os programas TCP usam números de porta reservados ou conhecidos, conforme apresentado na seguinte ilustração, da ajuda do Windows 2000 Server:



### O que é uma Porta TCP?

Bem, sem entrar em detalhes técnicos do TCP/IP, vou explicar, através de um exemplo prático, o conceito de porta. Vamos imaginar um usuário utilizando um computador com conexão à Internet. Este usuário, pode, ao mesmo tempo, acessar um ou mais sites da Internet, enviar um e-mail, está jogando no seu console através da Internet e assim por diante.

### Como o sistema sabe para qual dos programas se destina cada um dos pacotes que estão chegando no computador?

Quando o pacote chega no seu computador, o sistema lê no pacote o número da porta e sabe para quem encaminhar o pacote. Por exemplo, o protocolo HTTP, utilizado para o transporte de informações de um servidor Web até o seu navegador, opera, por padrão, na porta 80. Os pacotes que chegarem, destinados à porta 80, serão encaminhados para o navegador. Se houver mais de uma janela do navegador aberta, cada uma usando diferentes páginas, o sistema inclui informações, além da porta, capazes de identificar cada janela individualmente.

Com isso, quando chega um pacote para a porta 80, o sistema identifica para qual das janelas do navegador se destina o referido pacote.

Pesquisa:

<http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml>

### UDP - Uma Visão Geral

O User Datagram Protocol (UDP) é um padrão TCP/IP e está definido pela RFC 768, "User Datagram Protocol (UDP)." O UDP é usado por alguns programas em vez de TCP para o transporte rápido de dados entre hosts TCP/IP. Porém o UDP não fornece garantia de entrega e nem verificação de dados. De uma maneira simples, dizemos que o protocolo UDP manda os dados para o destino; se vai chegar ou se vai chegar corretamente, sem erros, só Deus sabe. Pode parecer estranho esta característica do UDP, porém você verá que em determinadas situações, o fato de o UDP ser muito mais rápido do que o TCP (por não fazer verificações e por não estabelecer sessões), o uso do UDP é recomendado.

O protocolo UDP fornece um serviço de pacotes **sem conexão** que oferece entrega com base no melhor esforço, ou seja, UDP não garante a entrega ou verifica o seqüenciamento para qualquer pacote. Um host de origem que precise de comunicação confiável deve usar TCP ou um programa que ofereça seus próprios serviços de seqüenciamento e confirmação.

As mensagens UDP são encapsuladas e enviadas em datagramas IP, conforme apresentado na seguinte ilustração, da ajuda do Windows 2000 Server:



### Portas UDP

O conceito de porta UDP é idêntico ao conceito de portas TCP, embora tecnicamente, existam diferenças na maneira como as portas são utilizadas em cada protocolo. A idéia é a mesma, por exemplo, se um usuário estiver utilizando vários programas baseados em UDP, ao mesmo tempo, no seu computador, é através do uso de portas, que o sistema operacional sabe a qual programa se destina cada pacote UDP que chega.

O lado do servidor de cada programa que usa UDP escuta as mensagens que chegam no seu número de porta conhecido. Todos os números de porta de servidor UDP menores que 1.024 (e alguns números mais altos) são reservados e registrados pela Internet Assigned Numbers Authority (IANA, autoridade de números atribuídos da Internet).

Cada porta de servidor UDP é identificada por um número de porta reservado ou conhecido. A tabela a seguir mostra uma lista parcial de algumas portas de servidor UDP conhecidas usadas por programas baseados em UDP padrão.

## Métodos de requisição HTTP

O protocolo HTTP define um conjunto de métodos, ou verbos, de requisição responsáveis por indicar a ação a ser executada para um dado recurso. Cada um deles implementa uma semântica diferente, mas alguns recursos são compartilhados por um grupo deles, como por exemplo, qualquer método de requisição pode ser do tipo safe, idempotent ou cacheable.

- **GET:** Solicita a representação de um recurso específico, devendo retornar apenas dados.
- **HEAD:** Solicita uma resposta de forma idêntica ao método GET, porém sem conter o corpo da resposta.
- **POST:** Submeter uma entidade a um recurso específico, frequentemente causando uma mudança no estado do recurso ou efeitos colaterais no servidor.
- **PUT:** Substitui todas as atuais representações do recurso de destino pela carga de dados da requisição.
- **DELETE:** Remove um recurso específico.
- **CONNECT:** Estabelece um túnel para o servidor identificado pelo recurso de destino.
- **OPTIONS:** Descreve as opções de comunicação com o recurso de destino.
- **TRACE:** Executa um teste de chamada loop-back junto com o caminho para o recurso de destino.
- **PATCH:** Aplica modificações parciais em um recurso.

## Códigos de status de respostas HTTP

Ao solicitarmos determinados recursos usando os métodos HTTP, obtemos uma informação chamada de códigos de status (status code). Os códigos de status das respostas HTTP indicam se uma requisição HTTP foi corretamente concluída.

As respostas são agrupadas em cinco classes:

1. **Respostas de informação** (100-199),
2. **Respostas de sucesso** (200-299),
3. **Redirecionamentos** (300-399)
4. **Erros do cliente** (400-499)
5. **Erros do servidor** (500-599).

Vamos descrever abaixo alguns dos status codes padrão, caso seu recurso retorne um código diferente dos descritos abaixo é por ele não ser padrão e pode ter sido configurado no servidor o qual respondeu à requisição do seu recurso.

### Respostas de sucesso

- **GET:** O recurso foi buscado e transmitido no corpo da mensagem.
- **HEAD:** Os cabeçalhos da entidade estão no corpo da mensagem.
- **PUT ou POST:** O recurso descrevendo o resultado da ação é transmitido no corpo da mensagem.
- **TRACE:** O corpo da mensagem contém a mensagem de requisição recebida pelo servidor.

**200 OK** - A Requisição foi bem sucedida. O significado do sucesso varia de acordo com o método HTTP:

**201 Created** - A requisição foi bem sucedida e um novo recurso foi criado como resultado. Esta é uma típica resposta enviada após uma requisição POST.

**202 Accepted** - A requisição foi recebida mas nenhuma ação foi tomada sobre ela. Isto é uma requisição não-comprometedora, o que significa que não há nenhuma maneira no HTTP para enviar uma resposta assíncrona indicando o resultado do processamento da solicitação. Isto é indicado para casos onde outro processo ou servidor lida com a requisição, ou para processamento em lote.

**203 Non-Authoritative Information** - O conjunto de meta-informações retornadas não é o conjunto exato disponível no servidor de origem, mas coletado de uma cópia local ou de terceiros. Exceto essa condição, a resposta de 200 OK deve ser preferida em vez dessa resposta.

**204 No Content** - Não há conteúdo para enviar para esta solicitação, mas os cabeçalhos podem ser úteis. O user-agent pode atualizar seus cabeçalhos em cache para este recurso com os novos.

**205 Reset Content** - Esta requisição é enviada após realizada a solicitação para informar ao user agent redefinir a visualização do documento que enviou essa solicitação.

**206 Partial Content** - Esta resposta é usada por causa do cabeçalho de intervalo enviado pelo cliente para separar o download em vários fluxos.

### Respostas de erro do Cliente

**400 Bad Request** - Essa resposta significa que o servidor não entendeu a requisição pois está com uma sintaxe inválida.

**401 Unauthorized** - Embora o padrão HTTP especifique "unauthorized", semanticamente, essa resposta significa "unauthenticated". Ou seja, o cliente deve se autenticar para obter a resposta solicitada.

**402 Payment Required** - Este código de resposta está reservado para uso futuro. O objetivo inicial da criação deste código era usá-lo para sistemas digitais de pagamento, porém ele não está sendo usado atualmente.

**403 Forbidden** - O cliente não tem direitos de acesso ao conteúdo, portanto o servidor está rejeitando dar a resposta. Diferente do código 401, aqui a identidade do cliente é conhecida.

**404 Not Found** - O servidor não pode encontrar o recurso solicitado. Este código de resposta talvez seja o mais famoso devido à frequência com que acontece na web.

**405 Method Not Allowed** - O método de solicitação é conhecido pelo servidor, mas foi desativado e não pode ser usado. Os dois métodos obrigatórios, GET e HEAD, nunca devem ser desabilitados e não devem retornar este código de erro.

**406 Not Acceptable** - Essa resposta é enviada quando o servidor da Web após realizar a negociação de conteúdo orientada pelo servidor, não encontra nenhum conteúdo seguindo os critérios fornecidos pelo agente do usuário.

**407 Proxy Authentication Required** -

Semelhante ao 401, porém é necessário que a autenticação seja feita por um proxy.

**408 Request Timeout** - Esta resposta é enviada por alguns servidores em uma conexão ociosa, mesmo sem qualquer requisição prévia pelo cliente. Ela significa que o servidor gostaria de derrubar esta conexão em desuso. Esta resposta é muito usada já que alguns navegadores, como Chrome, Firefox 27+, ou IE9, usam mecanismos HTTP de pré-conexão para acelerar a navegação. Note também que alguns servidores meramente derrubam a conexão sem enviar esta mensagem.

**409 Conflict** - Esta resposta será enviada quando uma requisição conflitar com o estado atual do servidor.

**410 Gone** - Esta resposta será enviada quando o conteúdo requisitado foi permanentemente deletado do servidor, sem nenhum endereço de redirecionamento. É esperado que clientes removam seus caches e links para o recurso. A especificação HTTP espera que este código de status seja usado para "serviços promocionais de tempo limitado". APIs não devem se sentir obrigadas a indicar que recursos foram removidos com este código de status.

**411 Length Required** - O servidor rejeitou a requisição porque o campo Content-Length do cabeçalho não está definido e o servidor o requer.

**Respostas de erro do Servidor**

**500 Internal Server Error** - O servidor encontrou uma situação com a qual não sabe lidar.

**501 Not Implemented** - O método da requisição não é suportado pelo servidor e não pode ser manipulado. Os únicos métodos exigidos que servidores suportem (e portanto não devem retornar este código) são GET e HEAD.

**502 Bad Gateway** - Esta resposta de erro significa que o servidor, ao trabalhar como um gateway a fim de obter uma resposta necessária para manipular a requisição, obteve uma resposta inválida.

**503 Service Unavailable** - O servidor não está pronto para manipular a requisição. Causas comuns são um servidor em manutenção ou sobrecarregado. Note que junto a esta resposta, uma página amigável explicando o problema deveria ser enviada. Estas respostas devem ser usadas para condições temporárias e o cabeçalho HTTP Retry-After: deverá, se possível, conter o tempo estimado para recuperação do serviço. O webmaster deve também tomar cuidado com os cabeçalhos relacionados com o cache que são enviados com esta resposta, já que estas respostas de condições temporárias normalmente não deveriam ser postas em cache.

**504 Gateway Timeout** - Esta resposta de erro é dada quando o servidor está atuando como um gateway e não obtém uma resposta a tempo.

**505 HTTP Version Not Supported** - A versão HTTP usada na requisição não é suportada pelo servidor.

**506 Variant Also Negotiates** - O servidor tem um erro de configuração interno: a negociação transparente de conteúdo para a requisição resulta em uma referência circular.

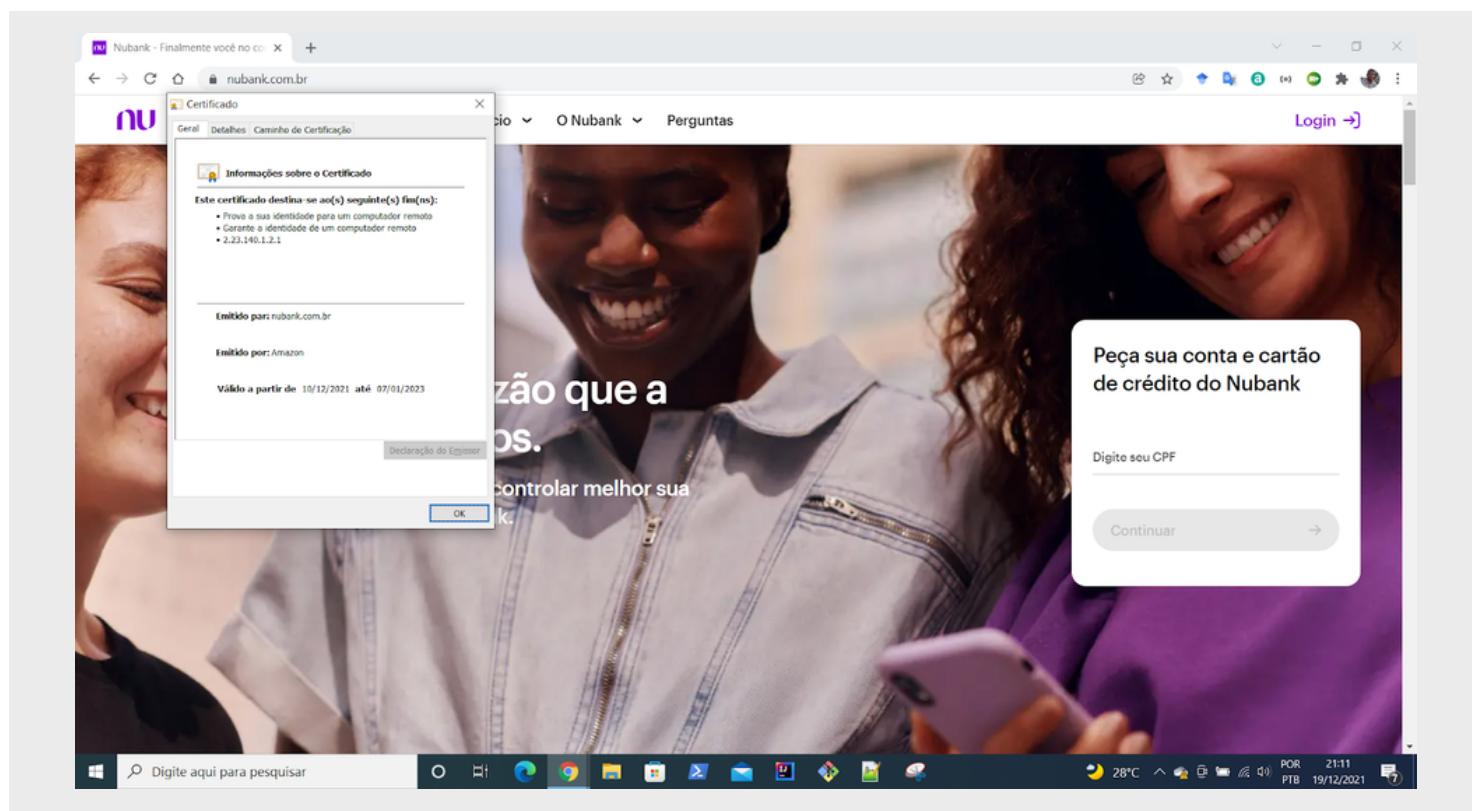
**507 Insufficient Storage** - O servidor tem um erro interno de configuração: o recurso variante escolhido está configurado para entrar em negociação transparente de conteúdo com ele mesmo, e portanto não é uma ponta válida no processo de negociação.

**508 Loop Detected** - O servidor detectou um looping infinito ao processar a requisição.  
510 Not Extended - Exigem-se extensões posteriores à requisição para o servidor atendê-la.

**511 Network Authentication Required** - O código de status 511 indica que o cliente precisa se autenticar para ganhar acesso à rede

## HTTPS

Quando navegamos em um site temos como premissa a segurança e privacidade, assim o HTTPS (Protocolo de Transferência de Hipertexto Seguro) é um protocolo que protege a integridade e a confidencialidade dos dados entre o computador do usuário e o site, portanto a recomendação do uso do HTTPS para proteger a conexão dos usuários ao seu site, qualquer que seja o conteúdo dele.



É através do protocolo Transport Layer Security (TLS) que os dados são protegidos e enviados pelo HTTPS. O TLS fornece três camadas principais de proteção:

- **Criptografia:** Para proteger de invasores os dados enviados e recebidos são criptografados. Desta forma, não é possível acompanhar as atividades em várias páginas ou roubar informações do usuário.
- **Integridade dos dados:** não é possível modificar nem corromper os dados durante a transferência (intencionalmente ou não) sem que isso seja detectado.
- **Autenticação:** prova que os usuários estão se comunicando com o site certo. Protege contra ataques "man-in-the-middle" e aumenta a confiança do usuário, o que beneficia os negócios.

### Um pouco sobre Certificados Digitais

De uma maneira simples, o Certificado Digital é a versão eletrônica da sua identificação de usuário na rede (usuário e senha). O Certificado Digital é como se fosse a "carteira de identidade" do usuário na rede.

Um certificado de chave pública, geralmente chamado somente de certificado, é uma declaração assinada digitalmente que vincula o valor de uma chave pública à identidade da pessoa, dispositivo ou serviço que contém a chave privada correspondente.

Os Certificados Digitais podem ser emitidos para uma série de funções, tais como autenticação de usuário na Internet, autenticação de um servidor Web, correio eletrônico seguro (S/MIME), IPSec, para utilização com o protocolo Transaction Layer Security (TLS, segurança de camada de transação) e assinatura de códigos.

Os certificados digitais tem que ser emitidos por uma Autoridade Certificadora (CA – Certificate Authority). Uma opção é usar uma autoridade certificadora externa, como por exemplo a VeriSign, que é uma empresa especializada em segurança e em certificação digital ([www.verisign.com](http://www.verisign.com)).

### Certificados e Autoridades de Certificação

Todo certificado é emitido por uma Autoridade de Certificação (CA – Certificate Authority). A autoridade de certificação, a partir de agora denominada apenas CA, é responsável pela verificação sobre a veracidade dos dados do usuário que está requisitando o certificado. Por exemplo, qualquer usuário pode solicitar um certificado para utilizar na Internet. Para obter o certificado ele precisa utilizar os serviços de uma CA, como por exemplo a VeriSign ([www.verisign.com](http://www.verisign.com)).

Uma autoridade de certificação é uma entidade encarregada de emitir certificados para indivíduos, computadores ou organizações, sendo que os certificados é que confirmam a identidade e outros atributos do usuário do certificado, para outras entidades. Uma autoridade de certificação aceita uma solicitação de certificado, verifica as informações do solicitador (incluindo aí uma série de documentos e comprovantes, os quais devem ser apresentados pelo solicitante do certificado) e, em seguida, usa sua chave privada para aplicar a assinatura digital no certificado.

A autoridade de certificação emite então o certificado para que o usuário do certificado o use como uma credencial de segurança dentro de uma infra-estrutura de chave pública (PKI). Uma autoridade de certificação também é responsável por revogar certificados e publicar uma lista de certificados revogados (CRL).

## AULA 5 e 6

#agile #manifestoagil #xp #scrum #kanban #12factor

### Manifesto para Desenvolvimento Ágil de Software

Estamos descobrindo maneiras melhores de desenvolver software, fazendo-o nós mesmos e ajudando outros a fazerem o mesmo. Através deste trabalho, passamos a valorizar:

1. Indivíduos e interações mais que processos e ferramentas.
2. Software em funcionamento mais que documentação abrangente.
3. Colaboração com o cliente mais que negociação de contratos.
4. Responder a mudanças mais que seguir um plano.

Ou seja, mesmo havendo valor nos itens à direita, valorizamos mais os itens à esquerda.

<https://agilemanifesto.org/iso/ptbr manifesto.html>

### Programação extrema

Programação extrema (do inglês eXtreme Programming), ou simplesmente XP, é considerada uma metodologia ágil e se ajusta bem a projetos de software com requisitos vagos e em constante mudança. Para isso, adota a estratégia de constante acompanhamento e realização de vários pequenos ajustes durante o desenvolvimento de software.

### O XP possui algumas características marcantes que são:

- Feedback constante.
- Abordagem incremental.
- Encoraja a comunicação entre as pessoas envolvidas.

Os cinco valores fundamentais são: comunicação, simplicidade, feedback, coragem e respeito. A partir desses valores, possui como princípios básicos: feedback rápido, presumir simplicidade, mudanças incrementais, abraçar mudanças e trabalho de qualidade.

Dentre as variáveis de controle em projetos (custo, tempo, qualidade e escopo), há um foco explícito em escopo. Para isso, recomenda-se a priorização de funcionalidades que representam maior valor possível para o negócio. Desta forma, caso seja necessário a diminuição de escopo, as funcionalidades menos valiosas serão adiadas ou canceladas.

A XP incentiva o controle da qualidade como variável do projeto, pois o pequeno ganho de curto prazo na produtividade, ao diminuir qualidade, não é compensado por perdas (ou até impedimentos) a médio e longo prazo.

Processo / Atividades metodológicas

Segundo Pressman, o XP prefere uma abordagem orientada a objetos como paradigma de desenvolvimento e envolve 4 atividades metodológicas:

- Planejamento
- Projeto ("Designing")
- Codificação
- Testes
- Práticas

Para aplicar os valores e princípios durante o desenvolvimento de software, XP propõe uma série de práticas. Há uma confiança muito grande na sinergia entre elas, os pontos fracos de cada uma são superados pelos pontos fortes de outras.

Jogo de Planejamento (Planning Game): O desenvolvimento é feito em interações semanais. No início da semana, desenvolvedores e clientes reúnem-se para priorizar as funcionalidades. Essa reunião recebe o nome de Jogo do Planejamento e nelas já devem estar criadas antecipadamente pelos usuários as User Stories (história dos usuários). Nessa reunião, o cliente identifica prioridades e os desenvolvedores as estimam. O cliente é essencial neste processo e assim ele fica sabendo o que está acontecendo e o que vai acontecer no projeto. Como o escopo é reavaliado semanalmente, o projeto é regido por um contrato de escopo negociável, que difere significativamente das formas tradicionais de contratação de projetos de software. Ao final de cada semana, o cliente recebe novas funcionalidades, completamente testadas e prontas para serem postas em produção.

**Fases pequenas (Small Releases):** A liberação de pequenas versões funcionais do projeto auxilia muito no processo de aceitação por parte do cliente, que já pode testar uma parte do sistema que está comprando. As versões chegam a ser ainda menores que as produzidas por outras metodologias incrementais, como o RUP.

**Metáfora (Metaphor):** Procura facilitar a comunicação com o cliente, entendendo a realidade dele. O conceito de rápido para um cliente de um sistema jurídico é diferente para um programador experiente em controlar comunicação em sistemas em tempo real, como controle de tráfego aéreo. É preciso traduzir as palavras do cliente para o significado que ele espera dentro do projeto.

**Design Simples (Simple Design):** Simplicidade é um princípio da XP. Projeto simples significa dizer que caso o cliente tenha pedido que na primeira versão apenas o usuário "teste" possa entrar no sistema com a senha "123" e assim ter acesso a todo o sistema, você vai fazer o código exato para que esta funcionalidade seja implementada, sem se preocupar com sistemas de autenticação e restrições de acesso. Um erro comum ao adotar essa prática é a confusão por parte dos programadores de código simples e código fácil. Nem sempre o código mais fácil de ser desenvolvido levará a solução mais simples por parte do projeto. Esse entendimento é fundamental para o bom andamento do XP. Código fácil deve ser identificado e substituído por código simples.

**Testes de Aceitação (Customer Tests):** São testes construídos pelo cliente e conjunto de analistas e testadores, para aceitar um determinado requisito do sistema.

**Semana de 40 horas (Sustainable Pace):**

Trabalhar com qualidade, buscando ter ritmo de trabalho saudável (40 horas/semana, 8 horas/dia), sem horas extras. Horas extras são permitidas quando trouxerem produtividade para a execução do projeto. Outra prática que se verifica neste processo é a prática de trabalho energizado, onde se busca trabalho motivado sempre. Para isto, o ambiente de trabalho e a motivação da equipe devem estar sempre em harmonia.

**Propriedade Coletiva (Collective Ownership):** O código fonte não tem dono e ninguém precisa solicitar permissão para poder modificar o mesmo. O objetivo com isto é fazer a equipe conhecer todas as partes do sistema.

**Programação Pareada (Pair Programming):** É a programação em par/dupla num único computador. Geralmente a dupla é formada por um iniciante na linguagem e outra pessoa funcionando como um instrutor. Como é apenas um computador, o novato é que fica à frente fazendo a codificação, e o instrutor acompanha ajudando a desenvolver suas habilidades. Desta forma o programa sempre é revisto por duas pessoas, evitando e diminuindo assim a possibilidade de defeitos. Com isto busca-se sempre a evolução da equipe, melhorando a qualidade do código fonte gerado.

**Padronização do Código (Coding Standards):** A equipe de desenvolvimento precisa estabelecer regras para programar e todos devem seguir estas regras. Desta forma parecerá que todo o código fonte foi editado pela mesma pessoa, mesmo quando a equipe possui 10 ou 100 membros.

**Desenvolvimento Orientado a Testes (Test Driven Development):** Primeiro crie os testes unitários (unit tests) e depois crie o código para que os testes funcionem. Esta abordagem é complexa no início, pois vai contra o processo de desenvolvimento de muitos anos. Só que os testes unitários são essenciais para que a qualidade do projeto seja mantida.

**Refatoração (Refactoring):** É um processo que permite a melhoria continua da programação, com o mínimo de introdução de erros e mantendo a compatibilidade com o código já existente. Refatorar melhora a clareza (leitura) do código, divide-o em módulos mais coesos e de maior reaproveitamento, evitando a duplicação de código-fonte;

**Integração contínua (Continuous Integration):**

Sempre que produzir uma nova funcionalidade, nunca esperar uma semana para integrar à versão atual do sistema. Isto só aumenta a possibilidade de conflitos e a possibilidade de erros no código fonte. Integrar de forma contínua permite saber o status real da programação.

<https://www.agilealliance.org/glossary/>

- **Os 12 Fatores**

[https://12factor.net/pt\\_br/](https://12factor.net/pt_br/)

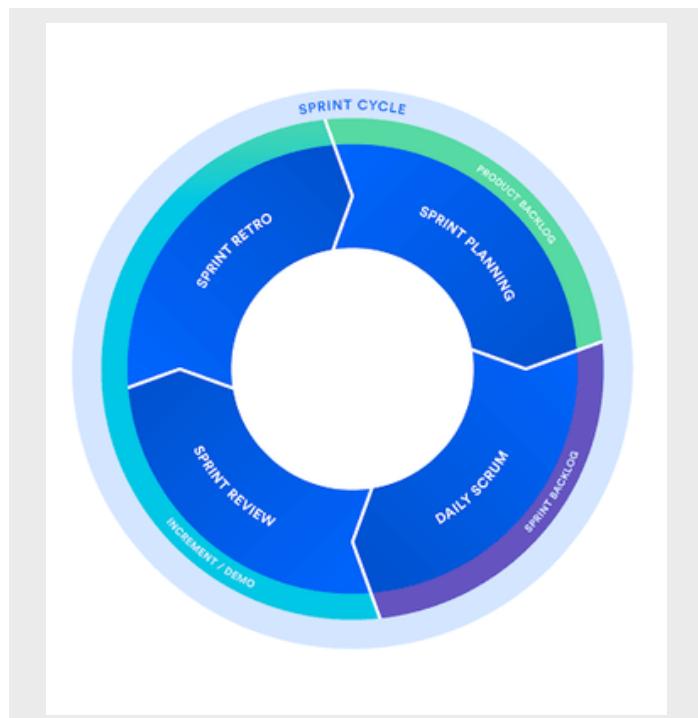
### Scrum

O Scrum é uma estrutura que ajuda as equipes a trabalharem juntas. Semelhante a uma equipe de rugby (de onde vem o nome) treinando para o grande jogo, o Scrum estimula as equipes a aprenderem com as experiências, a se organizarem enquanto resolvem um problema e a refletirem sobre os êxitos e fracassos para melhorarem sempre.

Embora o Scrum seja mais usado pelas equipes de desenvolvimento de software, os princípios e as lições dessa estrutura podem ser aplicados a todos os tipos de trabalhos em equipe. Esse é um dos motivos de o Scrum ser tão popular. Muitas vezes considerado uma estrutura de gestão de projetos de agilidade, o Scrum descreve um conjunto de reuniões, ferramentas e cargos que atuam juntos para ajudar as equipes a organizarem e gerenciarem o trabalho.

Em geral, as pessoas pensam que o Scrum e a agilidade são a mesma coisa porque o Scrum é centrado na melhoria contínua, que é o princípio fundamental da agilidade. No entanto, o Scrum é uma estrutura para concluir tarefas, enquanto a agilidade é uma forma de pensar. Você não consegue "se tornar ágil" do nada, afinal é necessária a dedicação de toda a equipe para mudar a forma de pensar sobre como entregar valor aos clientes. Contudo, você pode usar uma estrutura como o Scrum como ajuda para começar a pensar dessa forma e para praticar o desenvolvimento dos princípios de agilidade na comunicação e no trabalho do dia a dia.

A estrutura do Scrum é heurística; ela é baseada no aprendizado contínuo e na adaptação aos fatores variáveis. O Scrum reconhece que a equipe não sabe tudo no início de um projeto e que evolui de acordo com a experiência. Ele é estruturado para ajudar as equipes a se adaptarem naturalmente às mudanças e aos requisitos do usuário, com repriorização integrada no processo e ciclos curtos de liberação para que sua equipe aprenda e melhore constantemente.



Embora o Scrum seja estruturado, ele não é rígido por completo. Ele pode ser adaptado às necessidades de qualquer empresa. Há diversas teorias sobre o modo exato de como as equipes do Scrum devem trabalhar para se tornarem bem-sucedidas.

Vamos começar identificando os três artefatos do Scrum. Um artefato é algo que produzimos, como uma ferramenta para resolver problemas. No Scrum, os três artefatos são um backlog do produto, um backlog do sprint e um incremento com aquilo que você define como "concluído".

O backlog do produto é a principal lista do trabalho que precisa ser feita e é mantida pelo proprietário do produto ou gerente de produtos. É uma lista dinâmica de recursos, requisitos, aprimoramentos e correções que atua como a entrada para o backlog do sprint. Basicamente, ela é a "Lista de afazeres" da equipe. O backlog do produto é sempre revisto, priorizado e mantido pelo proprietário do produto porque, conforme aprimoramos o conhecimento ou o mercado muda, os itens podem não ser mais relevantes ou os problemas podem ser resolvidos de outras formas.

O backlog do sprint é a lista de itens, histórias de usuários ou correções de bugs selecionada pela equipe de desenvolvimento para a implementação no ciclo atual de sprint. Antes de cada sprint, durante a reunião de planejamento de sprint (que abordaremos posteriormente neste artigo), a equipe escolhe quais itens funcionarão para o sprint a partir do backlog do produto. Um backlog do sprint pode ser flexível e se desenvolver durante um sprint. No entanto, a meta fundamental do sprint, ou seja, o que a equipe deseja alcançar com o sprint atual, não pode ser comprometida.

Incremento (ou meta de sprint) é o produto final utilizável, proveniente de um sprint. Talvez você não escute a palavra "incremento" por aí, visto que ela costuma ser citada como a definição de "Concluído" dada pela equipe, como um marco, a meta de sprint ou, até mesmo, uma versão completa ou um epic lançado. Depende apenas de como as equipes definem "Concluído" e como você define suas metas de sprint.

Um dos componentes mais conhecidos da estrutura do Scrum é o conjunto de eventos sequenciais, cerimônias ou reuniões que as equipes do Scrum executam com frequência. É nas cerimônias que a gente vê as maiores diferenças entre as equipes.

Organizar o backlog: algumas vezes conhecido como "preparação do backlog", esse evento é responsabilidade do proprietário do produto. As principais tarefas do proprietário é orientar o produto em direção à visão do produto e acompanhar constantemente o mercado e o cliente. Dessa forma, ele mantém a lista usando o feedback dos usuários e da equipe de desenvolvimento para ajudar a priorizar e manter a lista clara e pronta para ser trabalhada a qualquer momento.

Planejamento de sprints: o trabalho que será realizado (escopo) ao longo do sprint atual é planejado durante essa reunião por toda a equipe de desenvolvimento. A reunião é conduzida pelo mestre do Scrum e é nela que a equipe decide a meta de sprint. Histórias de uso específicas são, então, acrescentadas ao sprint a partir do backlog do produto. Essas histórias sempre se alinham à meta e também são aceitas pela equipe do Scrum como sendo viáveis para a implementação durante o sprint.

No final da reunião de planejamento, cada membro do Scrum precisa esclarecer o que pode ser apresentado no sprint e como o incremento pode ser entregue.

Sprint: um sprint é o período real em que a equipe do Scrum trabalha em conjunto para concluir um incremento. A duração mais comum de sprint é de duas semanas, embora algumas equipes prefiram uma semana por ser mais fácil de realizar um escopo ou um mês por ser mais fácil de entregar um incremento de valor.

Todos os eventos, desde o planejamento à retrospectiva, ocorrem durante o sprint. Assim que um determinado intervalo de tempo é estabelecido para o sprint, ele precisará permanecer consistente durante todo o período de desenvolvimento. Isso ajuda a equipe a aprender com experiências passadas e a aplicar esse insight aos sprints futuros.

Scrum diário ou reunião rápida diária: é uma reunião diária bem rápida que ocorre na mesma hora (em geral, pela manhã) e local para manter a simplicidade. Muitas equipes tentam concluir a reunião em 15 minutos, mas é apenas uma diretriz. Ela também é chamada de "reunião rápida diária" para enfatizar que precisa ser breve. A meta do Scrum diário é fazer com que todos os integrantes da equipe estejam atualizados com as mesmas informações e alinhados com a meta do sprint para chegarem a um planejamento para as próximas 24 horas.

A reunião rápida é o momento de exprimir qualquer preocupação que você tenha a respeito de cumprir a meta do sprint ou quaisquer bloqueadores.

Uma forma comum de conduzir uma reunião rápida é solicitar que cada membro da equipe responda a três perguntas sobre o cumprimento da meta do sprint:

- O que eu fiz ontem?
- O que eu planejo fazer hoje?
- Há algum obstáculo?

No entanto, a gente já viu reuniões que logo se transformaram em pessoas lendo as agendas para falarem sobre o dia anterior ou seguinte. A teoria por trás da reunião rápida é que ela deixa a conversa fiada para uma reunião diária. Dessa forma, a equipe pode focar o trabalho no restante do dia.

Análise de sprint: no final do sprint, a equipe se reúne para uma sessão informal a fim de ver uma demonstração do incremento ou inspecioná-lo. A equipe de desenvolvimento mostra os itens de backlog que estão "concluídos" para as partes interessadas e aos colegas de equipe para que eles possam dar o feedback. O proprietário do produto pode decidir se vai lançar ou não o incremento, embora, na maioria das vezes, o incremento seja lançado.

É também nessa reunião de análise que o proprietário do produto reformula o backlog com base no sprint atual. Esse backlog pode orientar a próxima sessão de planejamento de sprint.

Retrospectiva de sprint: a retrospectiva é o momento em que a equipe se reúne para documentar e discutir o que funcionou e o que não funcionou em um sprint, em um projeto, nas pessoas ou nos relacionamentos, nas ferramentas ou, até mesmo, em determinadas cerimônias. A ideia é criar um local em que a equipe possa focar o que foi bem e o que precisa melhorar para a próxima vez, sem ficar ressaltando o que deu errado.

Uma equipe de Scrum precisa de três funções específicas: proprietário do produto, Scrum master e equipe de desenvolvimento. E, como as equipes de Scrum são multifuncionais, a equipe de desenvolvimento inclui testadores, designers, especialistas em experiência do usuário, engenheiros de operações e, acima de tudo, de desenvolvedores.

#### O proprietário do produto do Scrum

Os proprietários do produto são os campeões para o seu produto. Eles têm como foco compreender os negócios, o cliente e os requisitos do mercado, priorizando o trabalho a ser feito pela equipe de engenharia, adequadamente. Proprietários do produto eficazes:

Criam e gerenciam o backlog do produto.

Estabelecem uma parceria estreita com os negócios e a equipe para garantir que todos compreendam os itens de trabalho no backlog do produto.

Orientam claramente a equipe sobre quais recursos entregar em seguida.

Decidem quando lançar o produto com uma predisposição para entrega mais frequente.

Nem sempre o proprietário do produto é o gerente de produtos. Os proprietários do produto têm como foco garantir que a equipe de desenvolvimento agregue o máximo de valor para os negócios. Além disso, é importante que o proprietário do produto seja uma pessoa física. Nenhuma equipe de desenvolvimento quer orientação mista de vários proprietários do produto.

#### O Scrum master

Os mestres do Scrum são os campeões de Scrum em suas equipes. Eles orientam a equipe, os proprietários do produto e os negócios durante o processo de Scrum e procuram maneiras de melhorar a prática.

Um mestre do Scrum eficaz comprehende profundamente o trabalho realizado pela equipe e pode ajudá-la a otimizar a transparência e o fluxo de entrega. Como facilitador principal, ele agenda os recursos necessários (humanos e logísticos) para planejamento de sprint, reuniões rápidas, revisão de sprint e retrospectiva de sprint.

#### A equipe de desenvolvimento do Scrum

As equipes do Scrum mais eficazes são unidas, compartilham o mesmo local e normalmente são compostas por cinco, seis ou sete membros. Uma maneira de resolver o tamanho da equipe é usar a famosa "regra das duas pizzas", cunhada por Jeff Bezos, CEO da Amazon (a equipe deve ser pequena o suficiente para dividir duas pizzas).

Os membros da equipe têm diferentes conjuntos de competências, que são passadas de um para o outro para que nenhum deles se torne um obstáculo para a entrega do trabalho. Equipes fortes do Scrum se organizam e abordam os projetos impondo claramente o "nós". Todos os membros da equipe se ajudam para garantir a conclusão bem-sucedida do sprint.

A equipe do Scrum direciona o plano para cada sprint. Eles fazem a previsão de quanto trabalho acreditam que podem concluir durante a iteração usando sua velocidade histórica como um guia. Manter fixa a duração da interação fornece feedback importante para a equipe de desenvolvimento no seu processo de estimativa e entrega, que, por sua vez, faz previsões cada vez mais precisas ao longo do tempo.

### Kanban

O Scrum e o Kanban usam métodos visuais, como o painel do Scrum ou o painel Kanban para monitorar o progresso do trabalho. Os dois enfatizam a eficiência e a divisão de tarefas complexas em partes menores de trabalho gerenciável, mas as abordagens em direção a essa meta são diferentes.

O Scrum foca iterações menores de duração fixa. Assim que o período de um sprint é finalizado, as histórias ou as entradas de backlog do produto que podem ser implementadas durante esse ciclo de sprint são, então, determinadas. No Kanban, entretanto, o número de tarefas ou de trabalho em progresso (limite de WIP) a ser implementado no ciclo atual é fixado desde o início. O tempo que se leva para implementar esses recursos é, então, calculado de trás para frente.

O Kanban não é tão estruturado como o Scrum. Além do limite de WIP, ele é bastante aberto a interpretações. No entanto, o Scrum tem diversos conceitos categóricos aplicados como parte da implementação, tais como análise de sprint, retrospectiva, Scrum diário etc. Ele também insiste na multidisciplinaridade. Isto é, a equipe do Scrum consegue não depender de membros externos para alcançar os objetivos. Montar uma equipe multidisciplinar não é tarefa simples. Nesse sentido, o Kanban é mais fácil de adaptar, ao passo que o Scrum pode ser considerado uma mudança fundamental no processo de reflexão e no funcionamento de uma equipe de desenvolvimento.

<https://www.scrum.org/>



Digital  
College

ENSINO DE HABILIDADES DIGITAIS

[digitalcollege.com.br](http://digitalcollege.com.br)