**Computador, software e hardware**

**Hardwares:** Termo técnico que se refere a parte física de computadores e outros sistemas microeletrônicos.

**Periféricos:** É um “dispositivo auxiliar usado para enviar ou receber informações do computador”.

**Softwares:**

**Firmware:** Armazena todas as informações para que o equipamento funcione corretamente. É armazenado permanentemente em um circuito integrado (Chip) de memória de hardware no momento da fabricação.

**Drivers:** É a ponte entre o sistema operacional e o firmware dos dispositivos eletrônicos. Os drivers são instalados no computador quando há necessidade de se conectar com um novo dispositivo.

**Sistema Operacional:** Responsável pelo gerenciamento de todo o hardware do computador. Ele irá determinar para os programas de execução quando poderá utilizar o processador e por quanto tempo, quanto de memória RAM será usada, gerencia o HD/SSD, etc.

**Programas:**

* Software 🡺 Programa de computador;
* Program 🡺 Programa de computador ou app;

É um conjunto de instruções que descrevem uma tarefa e/ou trabalho especifico no seu computador. Ferramentas desenhadas para o usuário realizar ações.

**Dentro do Computador:**

**Memória RAM:** Espaço temporário de informação do sistema operacional e de aplicativos em uso. Quando uma tarefa em algum software é concluída, os arquivos em uso são movidos para o HD ou SSD para serem armazenadas.

**HD e SSD:** São dispositivos para armazenar dados. Mesmo depois de desligar o computador as informações permanecem guardadas.

**Processador:** É o cérebro do computador. Transformar informações em uma linguagem que o computador entende, Binário (0101001). Transforma dados em informações, por exemplo, carregar páginas, abrir e executar programas entre outras funções. Em partes ele tem responsabilidade sobre a velocidade que os programas são abertos ou executados.

**Internet, roteadores e servidores**

**Internet:** São redes interligadas pelo mundo todo. Uma rede tem como objetivo interligar computadores para fornecer aos usuários acesso a diversas informações. É feita por cabos que conectam o mundo inteiro.

**Rede de computadores e comunicação:** Conexão de dispositivos para emitir a transmissão de dados. A comunicação na internet é feita por meio de protocolos (Conjunto de regras).

**IP e Mac Address:** IP (Internet Protocol) é um número identificador dado ao seu computador, ou roteador, pelo provedor de internet no momento que se conecta à rede.

Mac Address é um número de série identificador gravado no dispositivo de rede. Através do IP que seu computador pode enviar e receber dados na internet.

**Servidores:** É um computador equipado com um ou mais processadores, portas de comunicação, softwares, algum sistema de armazenamento de dados como HDs e/ou SSDs. Fornece serviços a uma rede de computadores, chamada de cliente. Esses serviços podem varias de hospedagem de site, provedor de e-mails, entre outros.

**DNS:** Servidores DNS (Domain Name System) são responsáveis por localizar e traduzir para números IP endereços que digitamos nos navegadores. Em outras palavras o DNS permite acessar sites escrevendo o endereço nominal ao invés de números e pontos (IP).

**HTTP:** Hypertext Transfer Protocol. É um protocolo de transferência de dados. Digitando uma URL em algum navegador, o navegador cria uma requisição HTTP e manda para o servidor correspondente ao IP da URL. Dessa forma o servidor responde através do HTTP enviando todos os dados necessários para exibir o site ao usuário.

HTTPS 🡺 Hypertext Transfer Protocol Secure adiciona uma camada de criptografia, para segurança das informações enviadas ou recebidas de um site.

**Sistema Operacional:**

**O que é:** Interface entre o usuário e máquina.

Programa que irá controlar seu aparelho;

* Gerenciamento de dispositivos;
* Gerenciamento de aplicativos e programas;
* Gerenciamento de tarefa.

Conversar com o computador em linguagem de máquina.

**Características e objetivos:**

1. Fácil entendimento para o usuário:
   1. Experiência do usuário;
2. Controle de Hardware:
3. Uso dos periféricos;
4. Memoria;
5. HD;
6. Gerenciamento de Software:
   1. Programas;

**Grupo de SOs:** Os mais conhecidos.

**Unix:**

* Linux;
* OS X;
* Android;
* iOS.

**Windows:**

* Windows 7;
* Windows 10;
* Windows server.

**Tipos de interfaces do usuário:**

* GUI
  + Graphical User interface;
  + Interface gráfica, elementos gráficos;
    - Cursor do mouse ou touch;
    - Área de trabalho
    - Criar arquivos e pastas.
* CLI
  + Command-Line Interface;
  + Linha de comando;
  + Emitir comandos em texto para o computador;
    - Criar arquivos e pastas

**Módulos de Sistemas Operacionais:**

**Kernel:**

* Componente central, cerne do S.O;
* Primeiro modulo a ser iniciado;
* Permanece executando enquanto o S.O estiver ligado;

RESPONSABILIDADES: Gerenciamento da memória, processos, armazenamento e dispositivos.

**Gerenciador de Processos:**

* Processo é um programa em execução;
* Agendamento de processos (Scheduling);
  + Qual processo está executando agora;
* Thread
  + A divisão de um processo para melhor performance;
  + Um pedacinho do processo;
  + Poderá executar em paralelo.

**Multitarefa:**

* Multitasking;
* Várias tarefas podem ser executadas simultaneamente;
  + Troca muito rápida entre elas;
* Tarefas em segundo plano.

**Gerenciador de Arquivos:**

* File System;
* Organização e armazenamento dos arquivos;
  + Vídeos, imagens, Documentos;
* Diretórios (Pastas);
* Tipos de sistemas de arquivos;
  + FAT, NTFS;
  + Ext3, ext4;
  + HFS+, APFS;
  + São criados quando formatamos o nosso disco, por exemplo.

**Ferramentas de gerenciamento:**

* Tarefas (task manager, activity);
* Pacotes (chocolatey, brew, apt, snap).

**Linguagens de Programação:**

**Codar:**

* Escrever para o computador;
* Seguir regras;
* Linguagem humana x máquina;

**Low-Leve vs High-Level:**

* Low-Level:
  + Significa baixo nível;
  + Mais próximo da máquina, binário;
    - É mais difícil de escrever e de ler;
    - É mais rápida;
  + Existe pouca interferência de tradução;
  + A programação começou pelo baixo nível
* High-Level:
  + Significa alto nível;
  + Mais próxima da comunicação humana;
    - If, else, Function, Object, Class...
  + Leva mais tempo para a máquina traduzir e entender;
    - Compilar ou interpretar;
    - Nada mais que milissegundos;

**Sintaxe:**

* Conjunto de regras de escrita;
* Cada linguagem tem as suas regras;
* Símbolos;
  + 
* Palavras reservadas;
  + 

**Webpage:**

* Página Web;
* Acessado pelo navegador;
* Encontrado pelo endereço universal URL;
* <https://www.google.com.br/>
* Resposta dada pelo servidor é uma cópia da página;
  + HTML – Hyper Text Markup Language;
  + CSS – Cascading Style Sheet;
  + JavaScript;

**Website:**

* Site;
* Agrupamento, estruturado (arquitetado) de várias páginas WEB;
* Pode ser:
  + Estático;
  + Dinâmico;
* Um usuário comum não sabe diferenciar entre estático e dinâmico;

**Website estático:**

* Páginas estáticas
  + A página é sempre a mesma;
  + Alteração direto no código e por quem a criou;
* Não interage com o banco de dados;

**Website dinâmico:**

* Conteúdo da página é dinâmico;
* Interage com banco de dados;
  + Dados + modelo de página = página dinâmica;
* Modificação sem precisar mexer no código;
* Com o mesmo modelo de página, é gerada uma página em tempo real;

**Web Application:**

* Um programa, um software que vive no servidor;
  + Linguagens de programação;
  + Banco de dados;
* Utilizado através de páginas Web e pelo navegador;
* Tem uma complexidade maior que um Website e possui mais requisitos;
* Exemplos: Facebook, Gmail, YouTube, Figma e etc;
* **Desafios:** 
  + Adaptação nos diversos dispositivos (Responsividade);
  + Performance;
  + Acessibilidade;
    - Visual, Auditiva, SEO;

**Front-End x Back-End:**

Request

URL

Cliente

HTML, Css, JS, Img, Json, PDF, etc;

Browser

Response

Servidor

**Front-End:**

* Cliente (Client-Side);
* Browser;
  + Digita a URL e faz um pedido (Request) para o servidor;
* Tecnologias;
  + HTML, Css, JavaScript, Imagens, etc;
  + Frameworks e bibliotecas;

**Back-End:**

* Servidor (Server-Side);
* Recebe o pedido do front-end;
  + Entende e processa o pedido e devolve para o front-end;
  + Regras de negócios da aplicação;
* Tecnologias;
  + PHP, Java, Python, JavaScript, etc;
  + Banco de dados;

**Aplicação Web tradicional x SPA:**

**Aplicação Web tradicional:**

* Recarregar a página para atualizar o conteúdo;
* Front-end e Back-end numa só aplicação;
* Servidor processa e devolve toda a página de uma só vez;

**SPA:**

* Single Page Application;
  + Sensação de estar em uma única página;
* Não precisa recarregar a página para atualizar o conteúdo;
* Front-end e back-end são aplicações diferentes;
* Servidor responde a aplicação front-end;
  + Front-end feito com o uso de framework;
  + Back-end feito no formato de API;

**Alinhamento de Jornada:**

* Sites estáticos;
* Aplicações Web tradicional;
* Single Page Application (SPA);

**Ambiente Dev:**

**Editores Online:**

* Codepen (Recomendável criação de conta);
  + <https://codepen.io/>
* Codesandbox (Recomendável criação de conta);
  + <https://codesandbox.io/>