**Disciplina**: Programação

**Prof**.: Gilmar Gomes do Nascimento

**Aluno:** Francisco Wagner de Souza Silva

**Estudo Dirigido 27/04/22**

**- Pesquisar sobre Controladores de versão de código, busque exemplos;**

Um sistema de controle de versão (como o próprio nome já diz) tem a finalidade de gerenciar diferentes versões de um documento. Com isso ele te oferece uma maneira muito mais inteligente e eficaz de organizar seu projeto, pois é possível acompanhar um histórico de desenvolvimento, desenvolver paralelamente e ainda te oferecer outras vantagens, como exemplo, customizar uma versão, incluir outros requisitos, finalidades especificas, layout e afins sem mexer no projeto principal ou resgatar o sistema em um ponto que estava estável, isso tudo sem mexer na versão principal. Atualmente, os sistemas de controle de versão são classificados em dois tipos: **Centralizados** e **distribuídos.**

### Alguns exemplos de ferramentas de controle de versão são: CVS, Subversion, TFS, GIT, Mercurial.

### - Pesquise sobre IDEs e as principais usadas em Javascript;

Um ambiente de desenvolvimento integrado (IDE) é um software para criar aplicações que combina ferramentas comuns de desenvolvimento em uma única interface gráfica do usuário (GUI). Um IDE geralmente consiste em:

* **Editor de código-fonte**: é um editor de texto que auxilia na criação de código de software por meio de funcionalidades como destaque da sintaxe com indicadores visuais, recurso de preenchimento automático específico da linguagem e verificação de bugs durante a criação.
* **Automação de compilação local**: são utilitários que automatizam tarefas simples e repetíveis durante a criação de uma compilação local do software usada pelo desenvolvedor. São tarefas como compilação de código-fonte em código binário, criação de pacotes de código binário e execução de testes automatizados.
* **Debugger**: é um programa usado para testar outros programas e mostrar graficamente a localização do bug no código original.
* As principais IDEs usadas em Javascript são:
* Eclipse
* Visual Studio Code
* NetBeans
* Atom

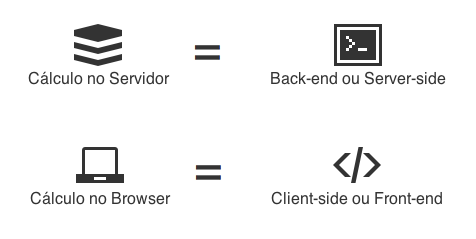
**- Linguagens lado cliente/servidor;**

Existem dois tipos de linguagens para desenvolvermos para internet: as linguagens **client-side** e as linguagens **server-side**. Você pode ver estes nomes meio diferentes em outros lugares como por exemplo, front-end, se referindo às linguagens client-side e back-end se referindo às linguagens server-side. Não se preocupe, são a mesma coisa.

As linguagens server-side são linguagens que o SERVIDOR entende. Isso quer dizer que vai escrever um código onde o servidor vai processá-lo e então vai mandar para o seu navegador a resposta.

As linguagens client-side são linguagens onde apenas o seu NAVEGADOR vai entender. Quem vai processar essa linguagem não é o servidor, mas o seu browser.

Por exemplo: se criarmos um script em linguagem back-end (PHP, Asp, Java, Ruby, etc) que apenas calcula a soma de 2 + 2, será o SERVIDOR (ou back, o server) que calculará este resultado. Se fizermos esse cálculo em alguma linguagem front-end, como o JavaScript, quem calculará essa conta é o BROWSER do usuário. Por isso o termo client ou front.



Exemplos de linguagens client-side:

* HTML (xHTML, HTML4, HTML5...)
* CSS (CSS2.1, CSS3...)
* JavaScript

Exemplos de linguagens server-side:

* PHP
* ASP
* Java
* Ruby
* Python

- **Navegadores e as ferramentas de teste e desenvolvimento**

Todo navegador web moderno inclui um conjunto avançado de ferramentas para desenvolvedores. Essas ferramentas fazem muitas, desde o HTML, CSS e JavaScript carregados e quantos recursos são necessários para a página apresentar para carregar.

A Automação de navegador web é um processo em que determinados passos no navegador são realizados repetidamente para garantir o funcionamento correto da funcionalidade dos aplicativos web. Ela pode ser aplicada para testes de controle de qualidade no processo de desenvolvimento e para o controle sobre a acessibilidade de informações do sistema e o desempenho durante a implementação. O segundo se torna cada vez mais importante quando as tendências atuais do mercado indicam que ter apenas um bom serviço não é suficiente; o serviço também deve ser altamente acessível e eficaz. Com ferramentas de automação do navegador web, é possível verificar a acessibilidade e o desempenho ao executar periodicamente alguns cenários de transação para determinados serviços.

## Alguns exemplos de ferramentas de automação de teste são: **Selenium, TestComplete, Telerik Test Studio, Robotium, Watir, HPE Unified Functional Testing, Ranorex, Cucumber, Visual Studio Test Professional, TestingWhiz.**