**Laboratório de Desenvolvimento Web**

*Npm* é o gerenciador de pacotes que vem junto ao Node

*Npm init -y* – Criar um package.json

*Node <arquivo>* – Rodar

*Npm install <módulos>* – Instalar módulos (para mais de um é só separar por espaços)

*Npm i* – Instalar todos os módulos salvos no package.json

Npm start – Iniciar o projeto caso haja o atalho definido no “Scripts” do package.json

    "test": "echo \"Error: no test specified\" && exit 1",

    "start": "node index.js"

**Sintaxe CommonJS**

- na importação, não é necessário especificar a extensão do arquivo

Módulo ‘**mensagem.js’**

function mensagem(nome)

{

    console.log(`Bem-vindo, ${nome}, ao Node com sintaxe do CommonJS`);

}

module.exports = mensagem;

Importação no index.js

const mensagem = require("./mensagem");

mensagem("Kaique");

**Sintaxe ES Modules**

- o ES Modules usa a extensão **.mjs** por padrão, então para utilizar a grafia do ES e ter extensões de arquivo js, é necessário acrescentar no package.json:

"main": "index.js",

"type": "module",

Módulo ‘**math.js’**

export function soma(x, y) // quando importar a soma, será obrigado colocá-la entre chaves

{

    return x + y;

}

export default function subtracao(x, y) // quando importar a subtração com 'default', não precisa colocá-la entre chaves

{

    return x - y;

}

Importação no index.js

import chalk from "chalk";

import {soma} from "./math.js"; // é necessário a extensão do arquivo

import subtracao from "./math.js";

const sum = soma(10, 5);

console.log(`A soma é: ${sum} \n`);

const sub = subtracao(10, 5);

console.log(`A subtração é: ${sub} \n`);

WebSocket cria objetos diferentes para cada máquina que está interagindo com o servidor

**Módulos Úteis**

- chalk: estilização do terminal

- nodemon: continua rodando até que os erros sejam corrigidos, sem precisar interromper a aplicação

- ws: websocket

- fs: ler arquivos