

Roteiro Aula Prática



**DESENVOLVIMENTO COM
FRAMEWORK NODE.JS**

ROTEIRO DE AULA PRÁTICA

NOME DA DISCIPLINA: DESENVOLVIMENTO COM FRAMEWORK NODE.JS

Unidade: U1 _ FUNDAMENTOS DO NODE.JS

Aula: A2_ CONSTRUINDO UM SERVIDOR WEB BÁSICO

Tempo previsto de execução de aula prática: 2h

OBJETIVOS

Definição dos objetivos da aula prática:

Nesta atividade prática, você irá aprender a criar um servidor HTTP básico usando Node.js que responde a diferentes solicitações com várias mensagens.

INFRAESTRUTURA

Instalações – Materiais de consumo – Equipamentos:

Equipamentos:

(Os materiais devem estar previstos no book)

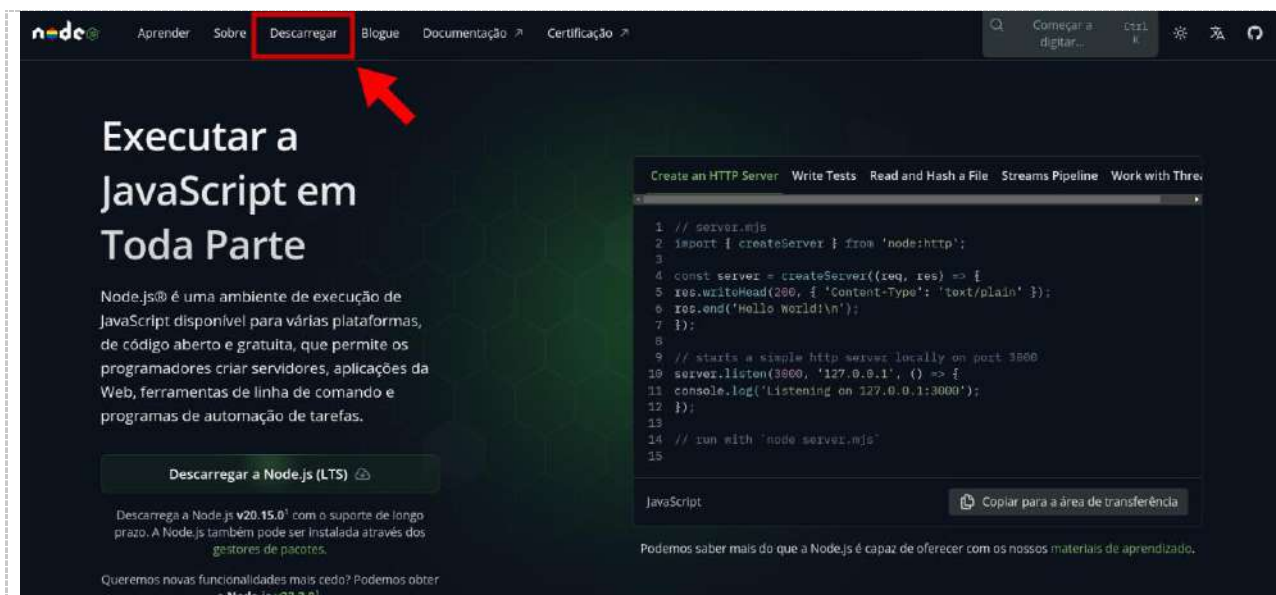
Exemplo:

- Desktop ou Notebook
~ 1 unid. 1 aluno
- Node.js instalado (é possível baixar e instalar em nodejs.org)
- Editor de texto (VSCode, Sublime Text, etc)

SOLUÇÃO DIGITAL

Infraestrutura mínima necessária para execução.

É necessário ter instalado o node.js que pode ser baixado direto do endereço: <nodejs.org>. Acesse o link. Na página que surge, acesse a opção “Descarregar”, como mostra a figura a seguir:



Em seguida, é preciso escolher a versão adequada para o seu computador, seja ele um desktop ou notebook. No exemplo a seguir, está sendo selecionado a versão pré-compilada, ou seja, o executável para instalação, considerando que o computador no qual será instalado está com windows 7+ de 64 bits. Veja o exemplo na imagem a seguir:



Depois de descarregá-lo, basta realizar a instalação conforme orientações que aparecerão na sua tela.

EQUIPAMENTO DE PROTEÇÃO INDIVIDUAL (EPI)

NSA

PROCEDIMENTOS PRÁTICOS

Procedimento/Atividade nº 1

Node.js é uma plataforma de desenvolvimento que permite executar código JavaScript fora do navegador. Construída sobre o motor V8 do Google Chrome, é usada para criar aplicações de rede escaláveis e de alta performance. Node.js utiliza um modelo de I/O não bloqueante e orientado a eventos, o que o torna ideal para aplicações que exigem um grande volume de operações simultâneas, como servidores web e APIs. Sua arquitetura permite desenvolver tanto o front-end quanto o back-end de uma aplicação em JavaScript, promovendo uma maior integração e simplicidade no desenvolvimento. Node.js possui um vasto ecossistema de pacotes, acessíveis através do gerenciador de pacotes npm, facilitando a adição de funcionalidades diversas ao projeto.

Agora que você pôde relembrar o que é o Node.js, você deve utilizá-lo na criação de um servidor básico, HTTP. Esse servidor é muito útil pois é indispensável no desenvolvimento de aplicações. Você precisa criar resposta para 4 situações:

localhost:3000

localhost:3000/sobre

localhost:3000/contato

E uma resposta de 'Página não encontrada' para demais casos.

Ao criar o código que levantará esse servidor, utilize a porta 3000 como porta padrão. Não esqueça de verificar se você não tem outro framework ou biblioteca instalado e configurado em sua máquina, que utilize essa porta. Se for necessário, pode alterá-la.

Atividade proposta:

Criação de servidor HTTP com resposta a 4 requisições diferentes: /, /sobre, /contato, e outra qualquer. Neste último caso, o servidor deve responder "Página não encontrada".

Procedimentos para a realização da atividade:

Passo-a-passo do procedimento para a execução da atividade/procedimento prático.

PROCEDIMENTO PARA INSTALAÇÃO:

1 – Inicializar um projeto Node.js

- Crie uma pasta para o seu projeto e navegue até ela via terminal.
- Execute o comando 'npm init -y' para criar um arquivo 'package.json' padrão.

2 - Instalação de dependências

- Para um servidor básico, não é necessário instalar pacotes adicionais. Para essa tarefa, vamos usar apenas módulos nativos do Node.js.

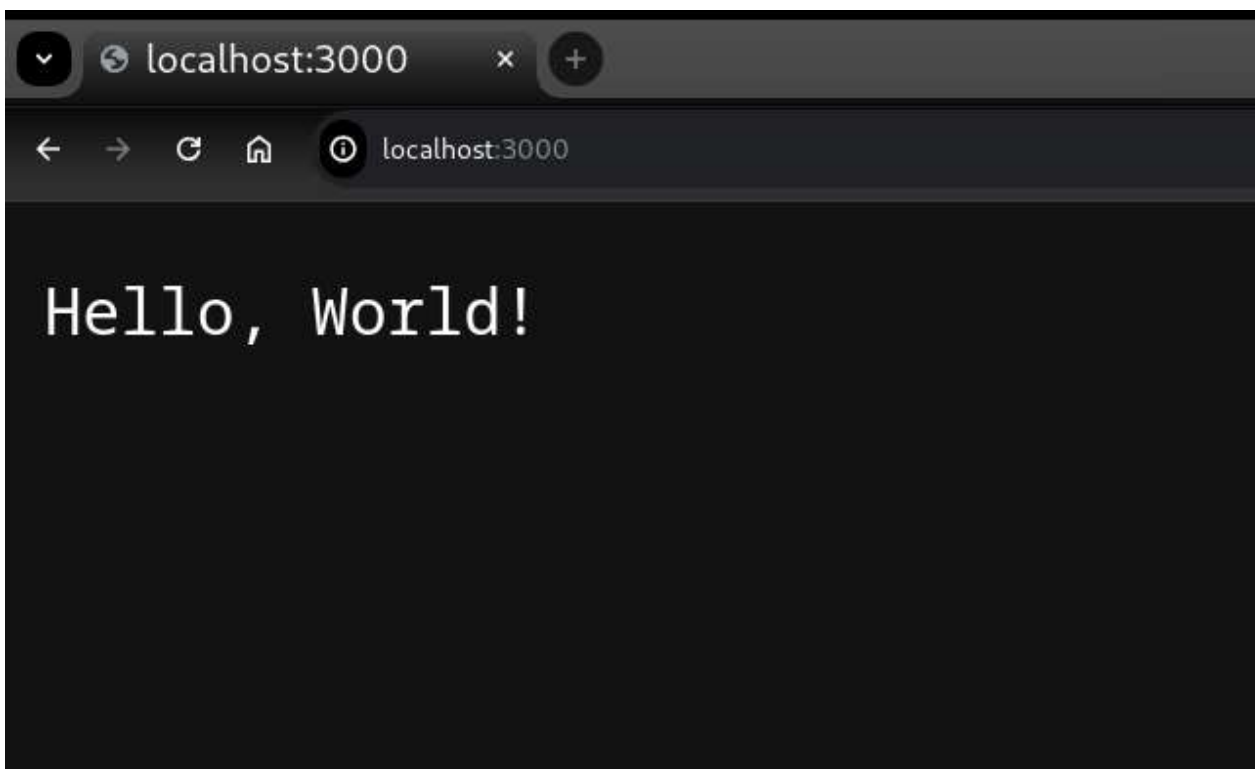
3 – Crie um arquivo chamado 'servidor.js' na raiz do projeto. É nesse arquivo que você deverá inserir o código usado para criar um servidor HTTP.

4 – Para testar seu servidor, após construir o código dentro do arquivo servidor.js, execute-o. Considerando que você esteja utilizando o VSCode, basta abrir, no menu Terminal, a opção "New Terminal".

5 – Execute o comando 'node servidor.js'.

6 – Finalmente, abra seu servidor acessando o endereço que aparecerá após a execução do comando mencionado na etapa anterior. No exemplo, o servidor é <http://localhost:3000>

Resultado:



Checklist:

- 1 - Inicialização do Projeto Node.js
- 2 - Criação do Servidor HTTP
- 3 - Configuração das Respostas [/, /sobre, /contato]
- 4 - Execução do Servidor
- 5 - Testes de Funcionamento
- 6 - Estrutura e Organização do Código
- 7 - Documentação e evidências

RESULTADOS (obrigatório – aparecer para todos)

Resultados de Aprendizagem:

O QUE SE ESPERA QUE O ALUNO TENHA APRENDIDO AO FINAL DA AULA PRÁTICA

Ao realizar essa atividade, espera-se que o aluno tenha aprendido:

Configuração de Projetos Node.js: Inicializar um projeto Node.js e criar um arquivo package.json básico.

Criação de Servidor HTTP: Escrever um servidor HTTP básico utilizando o módulo HTTP nativo do Node.js.

Manipulação de Solicitações e Respostas: Configurar o servidor para responder a diferentes URLs com mensagens específicas.

Execução de Scripts Node.js: Executar o servidor Node.js através do terminal e verificar sua operação.

Testes de Funcionamento: Testar as respostas do servidor em um navegador web, garantindo que as diferentes URLs retornem as respostas esperadas.

Organização de Código: Estruturar o código de forma clara e funcional para melhor compreensão e manutenção futura.

ESTUDANTE, VOCÊ DEVERÁ ENTREGAR

Descrição orientativa sobre a entrega da comprovação da aula prática:

Certifique-se de que seu servidor está funcionando corretamente conforme descrito nos passos de teste.

Envie o arquivo servidor.js e um print screen das respostas das URLs solicitadas (/ , /sobre, /contato, e uma URL desconhecida) para o sistema de entrega da atividade.

REFERÊNCIAS BIBLIOGRÁFICAS

Descrição (em abnt) das referências utilizadas

DIGITALOCEAN. How to create a web server in Node.js with the HTTP module. Disponível em: <https://www.digitalocean.com/community/tutorials/how-to-create-a-web-server-in-node-js-with-the-http-module-pt>. Acesso em: 24 jun. 2024.

Roteiro Aula Prática



**DESENVOLVIMENTO COM
FRAMEWORK NODE.JS**

ROTEIRO DE AULA PRÁTICA

NOME DA DISCIPLINA: DESENVOLVIMENTO COM FRAMEWORK NODE.JS

Unidade: U2 _ IMPLEMENTAÇÃO E DEPURAÇÃO UTILIZANDO O NODE JS

Aula: A1 _ TESTANDO E DEPURANDO APLICAÇÕES NODE.JS

Tempo previsto de execução de aula prática: 2h

OBJETIVOS

Definição dos objetivos da aula prática:

Proporcionar uma experiência prática com a configuração e utilização do framework de testes Mocha em uma aplicação Node.js.

INFRAESTRUTURA

Instalações – Materiais de consumo – Equipamentos:

Equipamentos:

(Os materiais devem estar previstos no book)

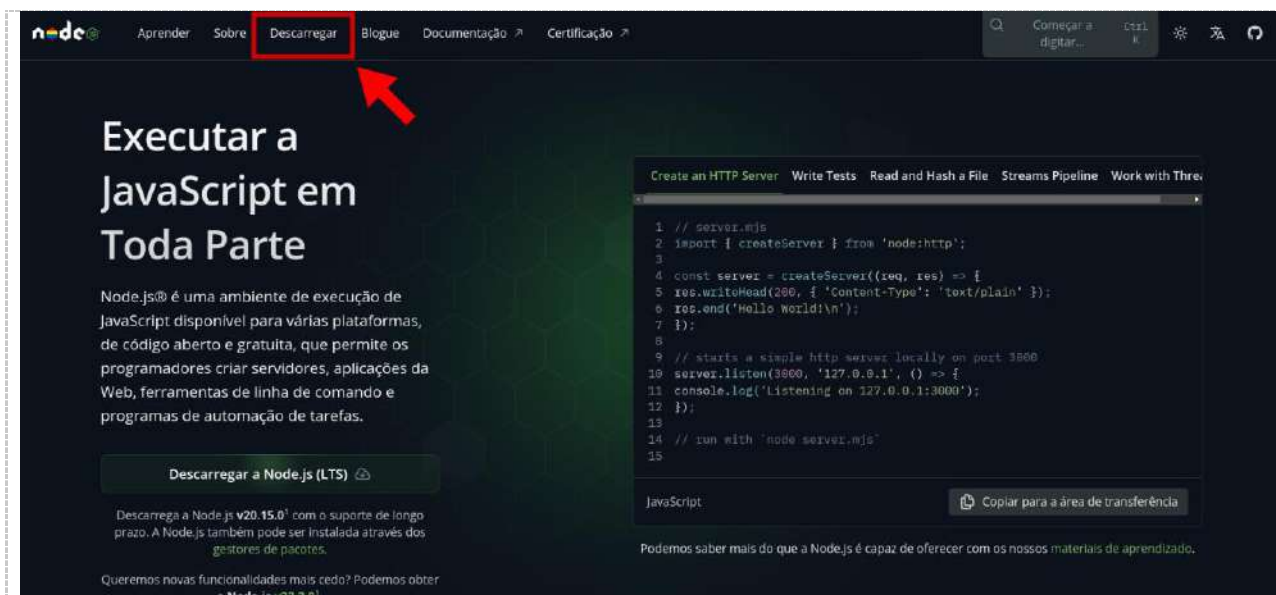
Exemplo:

- Desktop ou Notebook
~ 1 unid. 1 aluno
- Node.js instalado (é possível baixar e instalar em nodejs.org)
- Editor de texto (VSCode, Sublime Text, etc)

SOLUÇÃO DIGITAL

Infraestrutura mínima necessária para execução.

É necessário ter instalado o node.js que pode ser baixado direto do endereço: <nodejs.org>. Acesse o link. Na página que surge, acesse a opção “Descarregar”, como mostra a figura a seguir:



Em seguida, é preciso escolher a versão adequada para o seu computador, seja ele um desktop ou notebook. No exemplo a seguir, está sendo selecionado a versão pré-compilada, ou seja, o executável para instalação, considerando que o computador no qual será instalado está com windows 7+ de 64 bits. Veja o exemplo na imagem a seguir:



Depois de descarregá-lo, basta realizar a instalação conforme orientações que aparecerão na sua tela.

EQUIPAMENTO DE PROTEÇÃO INDIVIDUAL (EPI)

NSA

PROCEDIMENTOS PRÁTICOS

Procedimento/Atividade nº 1

Atividade proposta:

Criação de projeto de teste simples usando mocha para uma aplicação Node.js, utilizando uma função básica de soma.

Procedimentos para a realização da atividade:

Passo-a-passo do procedimento para a execução da atividade/procedimento prático.

PROCEDIMENTO PARA INSTALAÇÃO:

1 - Configuração do Projeto

Crie um novo diretório para o projeto e inicialize um projeto Node.js

2 - Instalação do Mocha

Instale o Mocha como dependência de desenvolvimento

3 - Criação de Funções

Crie um arquivo math.js para implementar funções simples de matemática

4 - Escrevendo Testes com Mocha

Crie um diretório test e dentro dele um arquivo math.test.js para os testes

5 - Executando os Testes

Adicione um script de teste ao package.json para executar os testes com Mocha

Digite o comando para execução do teste

6 - Documentação e entrega

Resultado esperado:

```
Math Functions
  sum
    ✓ should return 5 when adding 2 and 3
    ✓ should return -1 when adding -2 and 1
    ✓ should return 0 when adding 0 and 0

  3 passing (6ms)
```

Checklist:

- 1) Inicializar um projeto Node.js e configurar o Mocha como dependência de desenvolvimento.
- 2) Implementar funções simples (como sum) no arquivo math.js.
- 3) Escrever testes utilizando Mocha no arquivo math.test.js.
- 4) Adicionar um script de teste no package.json para executar os testes com Mocha.
- 5) Executar os testes e verificar se todos os casos passam corretamente.
- 6) Documentação e entrega

RESULTADOS

Resultados de Aprendizagem:

Após completar esta atividade, o aluno deverá ser capaz de configurar um ambiente básico de teste com Mocha para uma aplicação Node.js, implementar testes simples para funções JavaScript e executar os testes para verificar a funcionalidade do código.

ESTUDANTE, VOCÊ DEVERÁ ENTREGAR

Descrição orientativa sobre a entrega da comprovação da aula prática:

Documente a criação dos arquivos, incluindo o teste escritos.

REFERÊNCIAS BIBLIOGRÁFICAS

Descrição (em abnt) das referências utilizadas

MEDIUM. Testes de integração para aplicações Node.js com Mocha e Chai. Medium, [s.d.]. Disponível em: <https://medium.com/desenvolvimento-com-node-js/testes-de-integra%C3%A7%C3%A3o-para-aplica%C3%A7%C3%B5es-node-js-com-mocha-e-chai-610a1ba15e1b>. Acesso em: 25 jun. 2024.

Roteiro Aula Prática



**DESENVOLVIMENTO COM
FRAMEWORK NODE.JS**

ROTEIRO DE AULA PRÁTICA

NOME DA DISCIPLINA: DESENVOLVIMENTO COM FRAMEWORK NODE.JS

Unidade: U3 _ INTERFACE E SEGURANÇA NO NODE.JS

Aula: A1_DESENVOLVIMENTO DE INTERFACES DE USUÁRIO COM NODE.JS

Tempo previsto de execução de aula prática: 2h

OBJETIVOS

Definição dos objetivos da aula prática:

Proporcionar uma experiência prática com a criação de uma aplicação simples para validação de CPF, utilizando tecnologias front-end (HTML5, CSS3 e Javascript).

INFRAESTRUTURA

Instalações – Materiais de consumo – Equipamentos:

Equipamentos:

(Os materiais devem estar previstos no book)

Exemplo:

- Desktop ou Notebook
~ 1 unid. 1 aluno
- Node.js instalado (é possível baixar e instalar em nodejs.org)
- Editor de texto (VSCode, Sublime Text, etc)

SOLUÇÃO DIGITAL

Infraestrutura mínima necessária para execução.

NSA

EQUIPAMENTO DE PROTEÇÃO INDIVIDUAL (EPI)

NSA

PROCEDIMENTOS PRÁTICOS

Procedimento/Atividade nº 1

Atividade proposta:

Criação de projeto de aplicação de validação de campo CPF, muito comum em formulários disponíveis em aplicações web. A mensagem de validação deve aparecer logo abaixo do campo dedicado à digitação desse dado. Quando o CPF estiver incorreto, deve ser exibida uma mensagem de alerta com cor vermelha. Do contrário, deve ser exibido uma mensagem com cor verde.

Procedimentos para a realização da atividade:

Passo-a-passo do procedimento para a execução da atividade/procedimento prático.

PROCEDIMENTO PARA INSTALAÇÃO:

1 - Configuração do Projeto

Crie uma pasta para abrigar os arquivos do projeto

2 - Criação dos arquivos do projeto

Crie os arquivos: cpf.html, cpf.css e cpf.js

4 - Escrevendo os códigos em cada arquivo

Escreva o código do HTML, montando um formulário com um único campo, para CPF

Estilize o HTML com código CSS3

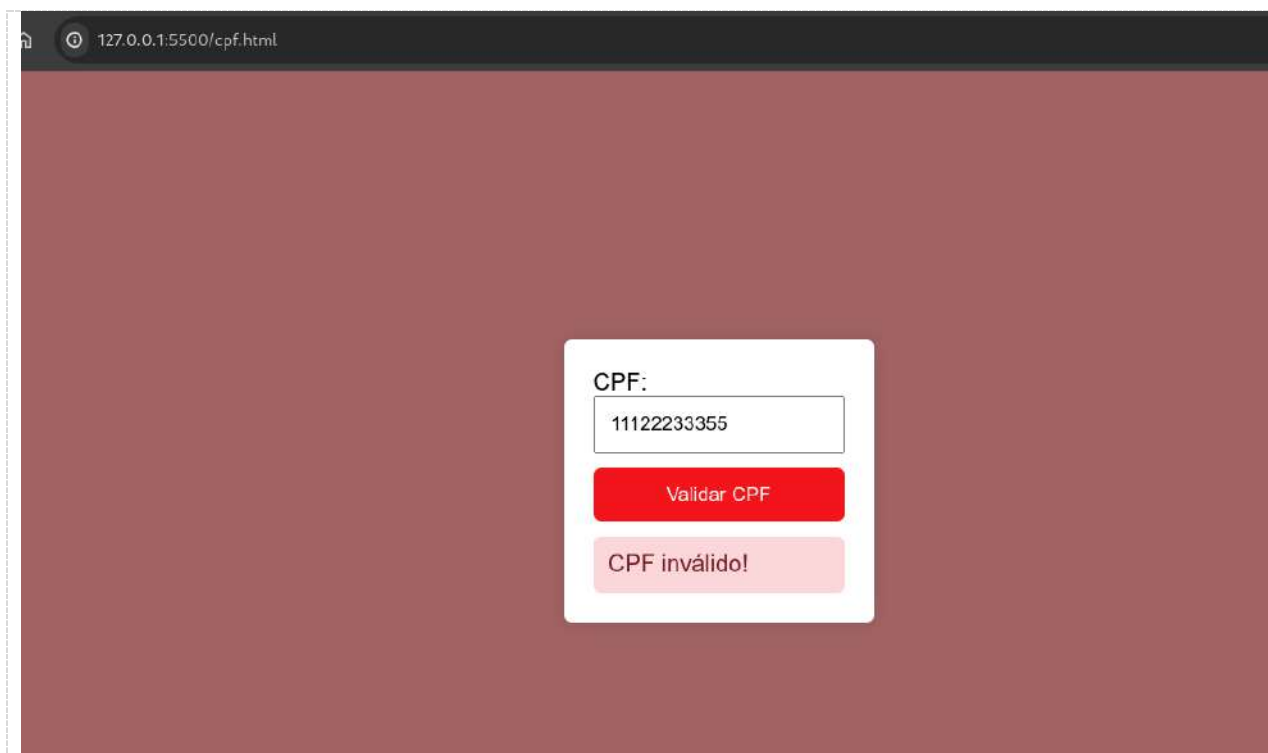
Construa o código de validação de CPF no arquivo cpf.js

5 – Testando a aplicação

Faça um teste com a aplicação e verifique se ela está validando CPF inserido no formulário.

6 - Documentação e entrega

Resultado esperado:



Checklist:

- 1) Configuração do Ambiente
- 2) Estrutura do Projeto
- 3) Criação do HTML
- 4) Estilo CSS
- 5) Lógica de Validação em JavaScript
- 6) Teste de Validação de CPF
- 7) Documentação e entrega

RESULTADOS

Resultados de Aprendizagem:

A atividade de validação de CPF permite aos alunos desenvolver habilidades em HTML, CSS e JavaScript, com foco na criação e estilização de formulários, manipulação do DOM e implementação de algoritmos de validação. Os alunos aprendem a fornecer feedback visual, testar e depurar código, além de praticar boas práticas de codificação e documentação de projetos, preparando-os para desafios mais complexos no desenvolvimento web.

ESTUDANTE, VOCÊ DEVERÁ ENTREGAR

Descrição orientativa sobre a entrega da comprovação da aula prática:

Documente a criação dos arquivos, incluindo o teste escritos.

REFERÊNCIAS BIBLIOGRÁFICAS**Descrição (em abnt) das referências utilizadas**

DEVMEDIA. Validar CPF com JavaScript. Disponível em: <https://www.devmedia.com.br/validar-cpf-com-javascript/23916>. Acesso em: 26 jun. 2024.

Roteiro Aula Prática



**DESENVOLVIMENTO COM
FRAMEWORK NODE.JS**

ROTEIRO DE AULA PRÁTICA

NOME DA DISCIPLINA: DESENVOLVIMENTO COM FRAMEWORK NODE.JS

Unidade: U4 _TESTES UTILIZANDO NODE.JS

Aula: A1 _ESTRATÉGIAS DE TESTES

Tempo previsto de execução de aula prática: 2h

OBJETIVOS

Definição dos objetivos da aula prática:

Configuração de Ambiente Node.js: Entender como configurar um ambiente de desenvolvimento Node.js.

Criação de um Servidor HTTP: Aprender a criar um servidor HTTP básico com Node.js.

Implementação de Rotas: Desenvolver e gerenciar rotas no servidor Node.js.

Teste de Integração: Escrever e executar testes de integração usando Mocha e Chai.

INFRAESTRUTURA

Instalações – Materiais de consumo – Equipamentos:

Equipamentos:

(Os materiais devem estar previstos no book)

Exemplo:

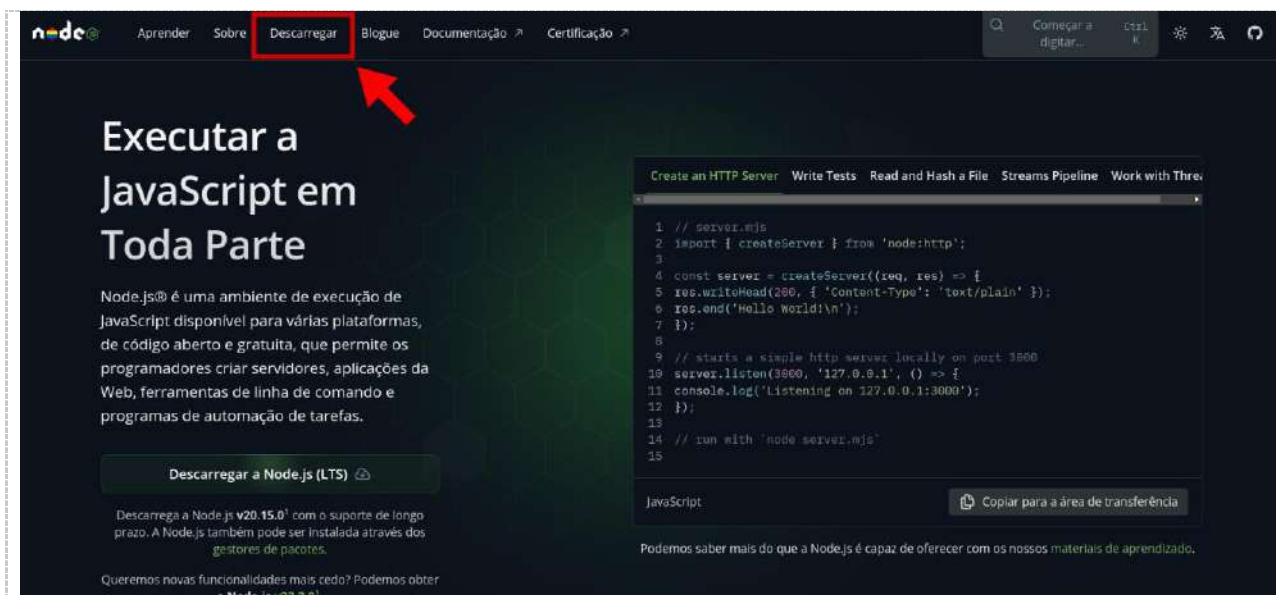
- Desktop ou Notebook
~ 1 unid. 1 aluno
- Node.js instalado (é possível baixar e instalar em nodejs.org)
- Editor de texto (VSCode, Sublime Text, etc)

SOLUÇÃO DIGITAL

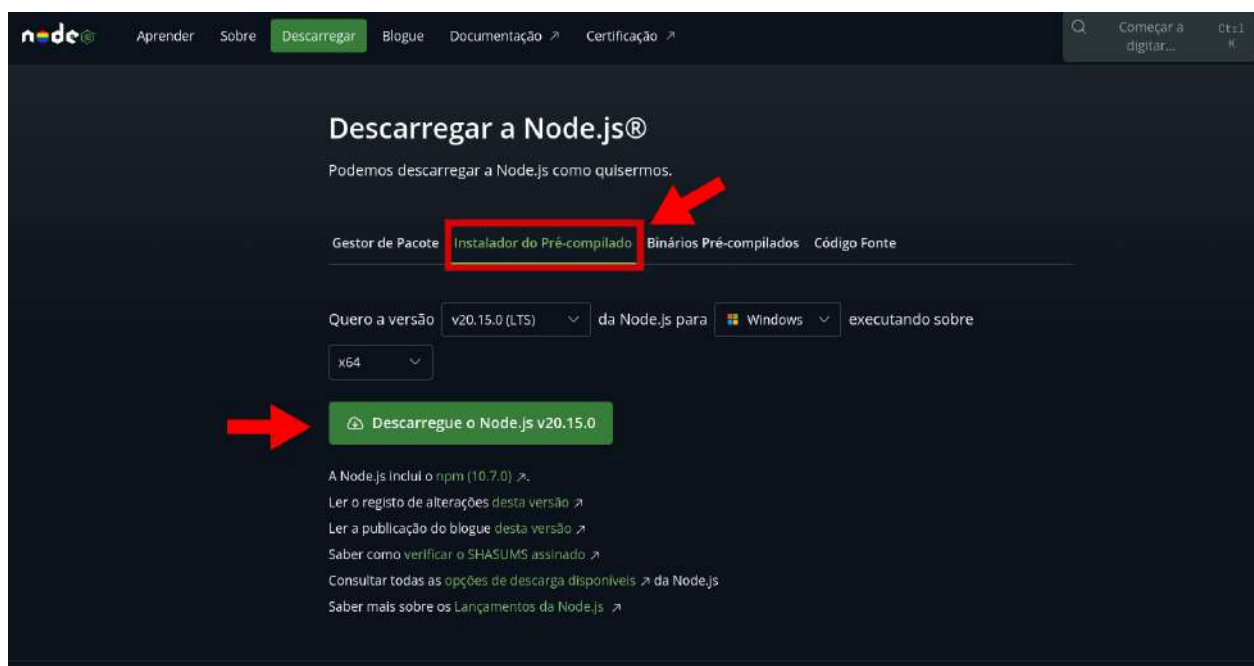
Infraestrutura mínima necessária para execução.

É necessário ter instalado o node.js que pode ser baixado direto do endereço: <nodejs.org>.

Acesse o link. Na página que surge, acesse a opção “Descarregar”, como mostra a figura a seguir:



Em seguida, é preciso escolher a versão adequada para o seu computador, seja ele um desktop ou notebook. No exemplo a seguir, está sendo selecionado a versão pré-compilada, ou seja, o executável para instalação, considerando que o computador no qual será instalado está com windows 7+ de 64 bits. Veja o exemplo na imagem a seguir:



Depois de descarregá-lo, basta realizar a instalação conforme orientações que aparecerão na sua tela.

EQUIPAMENTO DE PROTEÇÃO INDIVIDUAL (EPI)

NSA

PROCEDIMENTOS PRÁTICOS

Procedimento/Atividade nº 1

Atividade proposta:

Nesta atividade, você criará um servidor HTTP simples usando Node.js e escreverá testes de integração para validar o funcionamento das rotas desse servidor. Usaremos a biblioteca Mocha para escrever os testes e a biblioteca Chai para asserções.

Procedimentos para a realização da atividade:

Passo-a-passo do procedimento para a execução da atividade/procedimento prático.

PROCEDIMENTO PARA INSTALAÇÃO:

1 - Configuração do Projeto

Inicialize um novo projeto Node.js.

Instale as dependências necessárias (express, mocha, chai, chai-http).

2 - Criação do Servidor HTTP

Crie um arquivo server.js que configura e inicia um servidor HTTP básico utilizando Express.

Adicione duas rotas: uma rota GET para / que retorna uma mensagem "Hello World" e uma rota POST para /data que aceita um JSON e retorna o mesmo JSON com uma mensagem de sucesso.

3 - Estrutura do Projeto

Crie um arquivo math.js para implementar funções simples de matemática

4 - Escrever Testes de Integração

No arquivo test/integration.test.js, escreva testes de integração para validar o comportamento das rotas do servidor.

Use Mocha como framework de teste e Chai para asserções.

5 - Executar os Testes

Configure um script no package.json para rodar os testes.

Execute os testes e verifique se todos passam.

6 - Documentação e entrega

Resultado esperado:

```
Server running on port 3000
```

```
Server Routes
```

- ```
✓ should GET / and return Hello World
✓ should POST /data and return success message with data
```

```
2 passing (62ms)
```

### Checklist:

- Configuração do Projeto
- Criação do Servidor HTTP
- Estruturação do Projeto
- Escrita dos Testes de Integração
- Execução dos Testes
- Documentação e entrega

## RESULTADOS

### Resultados de Aprendizagem:

Após completar esta atividade, o aluno deverá ser capaz de configurar um ambiente básico de teste com Mocha para uma aplicação Node.js, implementar testes simples para funções JavaScript e executar os testes para verificar a funcionalidade do código.

## ESTUDANTE, VOCÊ DEVERÁ ENTREGAR

### Descrição orientativa sobre a entrega da comprovação da aula prática:

Documente a criação dos arquivos, incluindo os códigos escritos.

## REFERÊNCIAS BIBLIOGRÁFICAS

### Descrição (em abnt) das referências utilizadas

MEDIUM. Testes de integração para aplicações Node.js com Mocha e Chai. Disponível em: <https://medium.com/desenvolvimento-com-node-js/testes-de-integra%C3%A7%C3%A3o-para-aplic%C3%A7%C3%B5es-node-js-com-mocha-e-chai-610a1ba15e1b>. Acesso em: 26 jun. 2024.