

# Fundamentos em Nuvem

## Unidade 5

### Laboratórios Práticos e Exercícios

#### AVALIAÇÃO FINAL DA BASE – MATERIAL COMPLEMENTAR

Para começar a criar seu projeto Node.js, abra o terminal. Se você estiver usando o Windows, pode fazer isso pressionando as teclas “Windows” + “R”, o que abrirá o menu “Executar”. Na caixa de texto, digite “CMD” e clique em “OK” para abrir o terminal. Com o terminal aberto, você pode navegar até a área de trabalho digitando “cd desktop”.

Em seguida, é hora de criar a pasta onde o seu projeto será armazenado. Para isso, digite “mkdir nome-da-pasta-do-seu-projeto”, substituindo “nome-da-pasta-do-seu-projeto” pelo nome desejado para o seu projeto. Caso já tenha uma pasta criada, basta navegar até ela usando o comando “cd nome-da-pasta”.

Dentro da pasta do projeto, você precisa iniciar um novo projeto Node.js. Isso pode ser feito configurando o arquivo “package.json” com o comando “npm init -y”. Este arquivo é fundamental para gerenciar as dependências e informações do seu projeto Node.js.

Após configurar o projeto, é necessário instalar o framework Express, que será utilizado para criar o servidor. Para isso, execute o comando “npm install express”. O Express facilita a criação e gestão de rotas, simplificando o desenvolvimento do back-end.

Com o framework instalado, o próximo passo é desenvolver a modelagem do banco de dados. Vamos começar definindo duas tabelas principais:

- **Tabela 'profiles':**
  - user\_id: Número inteiro obrigatório que representa a chave estrangeira que conecta com a tabela 'users'.
  - data\_nascimento: Campo de data para registrar a data de nascimento do usuário.
  - foto\_perfil: Texto com até 255 caracteres para armazenar o caminho da foto de perfil do usuário.
- **Tabela 'users':**
  - id: Número inteiro obrigatório que serve como identificador único para cada usuário.
  - username: Texto com até 255 caracteres e obrigatório, usado para o nome de usuário.
  - password: Texto com até 255 caracteres e obrigatório, armazena a senha do usuário.
  - email: Texto com até 255 caracteres e obrigatório, guarda o endereço de e-mail do usuário.
  - telefone: Texto com até 100 caracteres, opcional, para o número de telefone do usuário.
  - created\_at: Marca de data e hora que registra o momento de criação do registro na tabela.

Depois de modelar as tabelas, conecte seu projeto Node.js ao banco de dados MySQL. Essa conexão permite que seu aplicativo interaja com o banco de dados, realizando operações como inserção e consulta de dados.

Agora, é hora de desenvolver o back-end do projeto. Crie um arquivo chamado “server.js”, que será responsável por configurar o servidor e definir as rotas. Esse arquivo inicializará o servidor, conectará ao banco de dados e configurará as rotas para as diferentes funcionalidades do seu aplicativo.

Para a implementação das rotas de registro e login de usuários, defina as rotas “/register” e “/login”. Utilize bibliotecas para hash de senhas e autenticação de usuários, garantindo a segurança dos dados. Essas rotas permitirão que novos usuários se registrem e façam login no seu aplicativo de forma segura.

Finalmente, é importante cuidar da parte visual do projeto. Crie arquivos HTML para as telas de login e cadastro. Utilize CSS para estilizar essas telas, aplicando estilos como tamanhos de fonte, cores e layout responsivo. O uso de Flexbox pode ajudar a criar layouts mais flexíveis e adaptáveis a diferentes tamanhos de tela.

## Organizando seu Trabalho para Upload na Plataforma de Ensino

Para garantir que seu trabalho seja enviado corretamente na plataforma de ensino, siga estas orientações:

### 1. Estrutura do Arquivo:

- **Nome do Arquivo:** Utilize um nome descritivo que identifique claramente o conteúdo do trabalho, seguindo as convenções de nomenclatura da plataforma. Exemplo: `AtividadeFinal_ModuloFundamentosNuvem_NomeCompleto.pdf`.
- **Formato do Arquivo:** Salve o trabalho no formato PDF para garantir a formatação original e compatibilidade com diferentes dispositivos.

### 2. Conteúdo do Arquivo:

- **Página de Título:** Inclua uma página de título com as seguintes informações:
  - **Nome do Aluno:** Seu nome completo conforme consta na plataforma.
  - **Título da Atividade:** Atividade Final de Módulo: Implementação de Cadastro e Login de Usuários.
- **Seção 1: Documentação do Projeto:**
  - **Diagrama de Fluxo de Dados:** Insira o diagrama de fluxo de dados do sistema de cadastro e login, utilizando ferramentas adequadas como Lucidchart, Draw.io ou outra ferramenta similar.
  - **Diagrama de Classes:** Inclua o diagrama de classes do sistema de cadastro e login, utilizando ferramentas adequadas como Lucidchart, Draw.io ou outra ferramenta similar.
  - **Manual do Usuário:** Apresente o manual do usuário com as instruções passo a passo para cadastro de novos usuários e login no sistema.
- **Seção 2: Implementação do Sistema:**
  - **Código-fonte:** Anexe o novo versionamento (página de login e estrutura de banco de dados).
- **Seção 3: Demonstração do Sistema:**

- **Capturas de Tela:** Inclua capturas de tela do sistema de cadastro e login, abaixo da captura de tela inclua o código fonte e uma breve explicação do código.
- **Atualização o seu código no GitHub Pages:** para obter um link funcional (seu site poderá ser visualizado em público).

### 3. Envio do Arquivo:

- Acesse a plataforma de ensino e localize o local designado para enviar a atividade final.
- Siga as instruções da plataforma para realizar o upload do arquivo PDF organizado conforme as orientações acima.
- Certifique-se de que o arquivo enviado esteja completo e atenda a todos os requisitos da atividade.

### Dicas:

- Comece planejando o sistema de cadastro e login antes de iniciar a implementação.
- Utilize ferramentas de versionamento de código para controlar as alterações no código (Git e GitHub).
- O sistema deve ser seguro e proteger as informações dos usuários.
- Faça backups frequentes do seu projeto.

Ao concluir esta atividade com sucesso, você terá demonstrado suas habilidades e conhecimentos em Fundamentos em Nuvem, além de ter aprimorado seu projeto integrador com uma funcionalidade essencial.

Então, mãos à obra!