

FASE 1

Em um mundo empresarial em constante evolução, onde a rapidez e a eficiência são mais valorizadas do que nunca, a gestão de estoque emerge como uma das pedras angulares de um negócio bem-sucedido. Independentemente do tamanho ou setor da empresa, a capacidade de gerenciar com precisão e eficiência os produtos em estoque determina, em grande parte, sua lucratividade e reputação no mercado.

Parabéns! Você foi contratado para desenvolver um sistema de controle de estoque, uma das ferramentas mais valiosas e essenciais no mundo dos negócios modernos. Após várias reuniões entre os analistas de requisitos e a área de negócio foram levantadas as seguintes histórias de usuário:

Feature: Cadastro de Fornecedor

Como gerente de compras eu quero cadastrar fornecedores no sistema para que possa gerenciar e rastrear todas as informações relevantes sobre os fornecedores com os quais faço negócios.

1º cenário: Cadastrando um novo fornecedor com sucesso

Dado que estou na página de cadastro de fornecedores e não existe um fornecedor com o mesmo CNPJ já cadastrado, quando eu inserir todas as informações válidas do fornecedor e clicar no botão “Cadastrar” então o fornecedor deve ser adicionado ao sistema e uma mensagem de “Fornecedor cadastrado com sucesso!” deve ser exibida.

2º cenário: Tentando cadastrar um fornecedor com CNPJ já existente

Dado que estou na página de cadastro de fornecedores e já existe um fornecedor com o mesmo CNPJ no sistema, quando eu tentar cadastrar um novo fornecedor com esse CNPJ então o sistema não deve permitir o cadastro e uma mensagem de “Fornecedor com esse CNPJ já está cadastrado!” deve ser exibida.

3º cenário: Tentando cadastrar um fornecedor com informações inválidas

Dado que estou na página de cadastro de fornecedores, quando eu inserir informações inválidas ou deixar campos obrigatórios em branco e clicar no botão “Cadastrar” então o sistema não deve permitir o cadastro e mensagens de erro relevantes devem ser exibidas ao lado dos campos inválidos.

Protótipo Descritivo: Tela de Cadastro de Fornecedor

Cabeçalho:

Título: “Cadastro de Fornecedor”

Botão de voltar ou fechar (caso esteja em um modal)

Formulário de Cadastro:

Nome da Empresa (obrigatório):

Campo de texto

Placeholder: “Insira o nome da empresa”

CNPJ (obrigatório):

Campo de texto formatado para CNPJ

Placeholder: "00.000.000/0000-00"

Endereço (obrigatório):

Campo de texto longo

Placeholder: "Insira o endereço completo da empresa"

Telefone (obrigatório):

Campo de texto formatado para números de telefone

Placeholder: "(00) 0000-0000"

E-mail (obrigatório):

Campo de texto para e-mail

Placeholder: "exemplo@fornecedor.com"

Contato Principal (obrigatório):

Campo de texto

Placeholder: "Nome do contato principal"

Feature: Cadastro de Produtos

Como gerente de estoque, eu quero cadastrar produtos no sistema para que possa gerenciar e rastrear todas as informações relevantes sobre os produtos disponíveis no estoque.

1º cenário: Cadastrando um novo produto com sucesso

Dado que estou na página de cadastro de produtos e não existe um produto com o mesmo código de barras já cadastrado. quando eu inserir todas as informações válidas do produto e clicar no botão "Cadastrar" então o produto deve ser adicionado ao sistema e uma mensagem de "Produto cadastrado com sucesso!" deve ser exibida.

2º cenário: Tentando cadastrar um produto com código de barras já existente

Dado que estou na página de cadastro de produtos e já existe um produto com o mesmo código de barras no sistema, quando eu tentar cadastrar um novo produto com esse código de barras então o sistema não deve permitir o cadastro e uma mensagem de "Produto com este código de barras já está cadastrado!" deve ser exibida.

3º cenário: Tentando cadastrar um produto com informações inválidas

Dado que estou na página de cadastro de produtos, quando eu inserir informações inválidas ou deixar campos obrigatórios em branco e clicar no botão "Cadastrar" então o sistema não deve permitir o cadastro e mensagens de erro relevantes devem ser exibidas ao lado dos campos inválidos.

Protótipo Descritivo: Tela de Cadastro de Produto

Cabeçalho:

Título: "Cadastro de Produto"

Botão de voltar ou fechar (caso esteja em um modal)

Formulário de Cadastro:

Nome do Produto (obrigatório):

Campo de texto

Placeholder: "Insira o nome do produto"

Código de Barras:

Campo de texto numérico

Placeholder: "Insira o código de barras"

Descrição (obrigatório):

Campo de texto de várias linhas

Placeholder: "Descreva brevemente o produto"

Quantidade em Estoque:

Campo de texto numérico

Placeholder: "Quantidade disponível"

Categoria (obrigatório):

Menu suspenso (dropdown) com categorias pré-definidas, como "Eletrônicos", "Alimentos", "Vestuário", etc.

Opção "Outro" para especificar uma categoria não listada

Data de Validade (se aplicável):

Campo de data

Placeholder: "Selecione a data de validade"

Imagem do Produto (se aplicável):

Campo de upload de arquivo

Opção para carregar uma imagem representativa do produto

Feature: Associação de Fornecedor a Produto

Como gerente de estoque, eu quero associar fornecedores aos produtos para que possa rastrear de quais fornecedores estou adquirindo produtos específicos e gerenciar melhor as relações de compra.

1º cenário: Associando um fornecedor a um produto com sucesso

Dado que estou na página de detalhes de um produto específico e tenho uma lista de fornecedores cadastrados, quando eu selecionar um fornecedor da lista e clicar no botão "Associar Fornecedor" então o fornecedor deve ser associado a esse produto e uma mensagem de "Fornecedor associado com sucesso ao produto!" deve ser exibida.

2º cenário: Tentando associar um fornecedor que já está associado ao produto

Dado que estou na página de detalhes de um produto específico e o fornecedor já está associado a este produto, quando eu tentar associar novamente o mesmo fornecedor então o sistema não deve permitir a ação e uma mensagem de "Fornecedor já está associado a este produto!" deve ser exibida.

3º cenário: Desassociando um fornecedor de um produto

Dado que estou na página de detalhes de um produto específico e o produto tem fornecedores associados, quando eu selecionar um fornecedor associado e clicar no botão “Desassociar Fornecedor” então o fornecedor deve ser desassociado do produto e uma mensagem de “Fornecedor desassociado com sucesso!” deve ser exibida.

Protótipo Descritivo: Tela de Associação de Fornecedor a Produto

Cabeçalho:

Título: “Associação de Fornecedor a Produto”

Botão de voltar ou fechar (caso esteja em um modal)

Detalhes do Produto:

Nome do Produto: Campo de texto somente leitura com o nome do produto

Código de Barras: Campo de texto somente leitura com o código de barras

Descrição: Campo de texto somente leitura com a descrição breve do produto

Imagem do Produto (se aplicável): Miniatura da imagem do produto

Associação de Fornecedor:

Seleção de Fornecedor:

Menu suspenso (dropdown) com a lista de fornecedores cadastrados

Placeholder: “Selecione um fornecedor”

Botão “Associar Fornecedor”: Para confirmar a associação do fornecedor selecionado ao produto

Fornecedores Associados:

Lista ou tabela mostrando os fornecedores atualmente associados ao produto. Cada entrada da lista tem:

Nome do Fornecedor

CNPJ

Botão “Desassociar”: Permite remover a associação do fornecedor ao produto

FASE 2

ETAPA 1 - Preparando o ambiente de desenvolvimento Backend

Passo 1 - Configuração das Ferramentas

1. Faça a instalação do Visual Studio Code disponível para download em: <https://code.visualstudio.com/>
2. Faça a instalação do node.js disponível para download em: <https://nodejs.org/en>. Recomendamos baixar a versão LTS (Long Term Support) que, embora seja mais antiga, é bem mais estável.
3. Faça a instalação do Insomnia (ou Postman, se preferir), disponível para download em: <https://insomnia.rest/download>
4. (opcional) Caso queira utilizar um framework node.js, recomendamos utilizar o nest.js (<https://nestjs.com/>). A instalação é bem simples, siga o roteiro disponível em: <https://docs.nestjs.com/>

Com o Visual Studio Code e o Node.js instalados você está pronto para começar a desenvolver, porém precisamos pensar em um desenvolvimento moderno onde vários desenvolvedores possam trabalhar juntos em seu projeto bem como, quem sabe, a criação de uma esteira de CI/CD (se não souber o que é CI/CD não se preocupe, ainda vamos falar disso em disciplinas futuras).

Se você ainda não tem uma conta no GitHub (<https://github.com/>), entre no site e crie a sua conta gratuitamente. Ter um perfil público no GitHub pode ser uma ferramenta valiosa no processo de recrutamento para várias vagas relacionadas à tecnologia, especialmente para desenvolvedores. Nossa sugestão é que deixe este projeto público para que ele possa se somar a seu portfólio.

Após realizarmos a instalação e criarmos a nossa conta no github, está na hora de juntar tudo e iniciarmos o nosso projeto. Vamos iniciar configurando a integração do seu Visual Studio com o Github. Essa é uma configuração um pouquinho “mais complicada” então sugiro que assista ao vídeo abaixo:

<https://www.youtube.com/watch?v=7cNP3AE49Bg>

Passo 2 - Iniciando um novo projeto NodeJS

Agora temos tudo instalado então vamos criar o nosso projeto. Após configurar o github e o VisualStudio code, siga o roteiro abaixo para criar o primeiro projeto em node.js:

1. Abra o VisualStudio Code e selecione a pasta que realizou o primeiro commit no github.
2. Usando o Visual Studio crie o arquivo README.md com o seguinte conteúdo:

FACULDADE GRAN (<https://faculdade.grancursosonline.com.br/>)

Projeto Disciplina Projeto Integrador

3. Abra o Terminal no Visual Studio e Inicialize um novo projeto Node.js:

```
npm init
```

Isso iniciará um assistente que o guiará pela criação de um arquivo package.json. Você pode pressionar “Enter” para aceitar os padrões ou fornecer suas próprias informações. O arquivo package.json contém meta-informações sobre o seu projeto e dependências.

4. Crie um novo arquivo para o seu aplicativo chamado app.js

5. Abra o arquivo app.js no Visual Studio Code e adicione o seguinte código:

```
const http = require('http');

const server = http.createServer((req, res) => {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Olá, Mundo!')
});

const PORT = 3000;
server.listen(PORT, () => {
  console.log(`Servidor rodando em http://localhost:${PORT}/`);
});
```

Este é um servidor HTTP simples que responde com “Olá, Mundo!” para qualquer solicitação.

6. Volte para o terminal e execute o seu projeto:

```
node app.js
```

7. Acesse o servidor:

Abra seu navegador e navegue até <http://localhost:3000/>. Você deve ver “Olá, Mundo!” exibido.

8. Agora que temos a nossa primeira aplicação rodando, faça um commit no git para salvar o conteúdo.

```
git add .
git commit -m "Primeiro commit"
git push
```

ETAPA 2 - Construção dos Artefatos no Backend

O backend, muitas vezes referido como o “lado do servidor”, é a espinha dorsal de qualquer aplicação web. Ele interage com o banco de dados, processa a lógica de negócios e envia as informações necessárias para o frontend para serem exibidas ao usuário. Neste projeto iremos construir 3 controllers conforme detalhado abaixo. Utilize o Insomnia para testar o seu backend bem como o SQLite como banco de dados inicial, apenas para testar a sua aplicação.

1. Controlador de Produto:

Este controlador é responsável por gerenciar as operações relacionadas aos produtos, como criar, ler, atualizar e deletar (CRUD). No contexto da nossa aplicação, ele interagirá com o banco de dados para armazenar informações sobre cada produto, como nome, descrição, preço e código de barras.

2. Controlador de Fornecedor:

Similar ao controlador de produto, o controlador de fornecedor gerencia as operações relacionadas aos fornecedores. Ele cuida dos detalhes como nome do fornecedor, CNPJ, endereço e contato.

3. Controlador de Associação Produto/Fornecedor:

Este é um controlador especializado que lida com a relação entre produtos e fornecedores. Sabendo que um produto pode ser fornecido por vários fornecedores e um fornecedor pode oferecer vários produtos, essa relação é muitas vezes chamada de “muitos para muitos”. O controlador de associação permite associar um produto a um fornecedor, desassociá-los e consultar quais produtos são fornecidos por um determinado fornecedor (e vice-versa).

Testando com o Insomnia:

Uma vez que o backend esteja operacional, é vital testar suas rotas e lógica para garantir que tudo funcione conforme o esperado. O Insomnia é uma ferramenta que permite fazer requisições HTTP para seu servidor, tornando mais fácil testar, organizar e reutilizar as chamadas para seu backend.

ETAPA 3 - Preparando o ambiente de desenvolvimento e Construção dos Artefatos no Frontend

Passo 1 - Configuração das Ferramentas

Iremos utilizar para o desenvolvimento frontend e backend da mesma IDE, ou seja, o Visual Studio Code. Os passos abaixo já foram realizados quando da configuração do projeto Backend, então você pode pular caso já tenha executado.

1. Faça a instalação do Visual Studio Code disponível para download em: <https://code.visualstudio.com/>

2. Faça a instalação do node.js e o npm disponível para download em: <https://nodejs.org/en>. Recomendamos baixar a versão LTS (Long Term Support) que, embora seja mais antiga, é bem mais estável.

3. (opcional) Caso queira utilizar um framework react.js, recomendamos utilizar o next.js (<https://nextjs.org/>). A instalação é bem simples, siga o roteiro disponível em: <https://nextjs.org/docs/getting-started/installation>

Com o Visual Studio Code e o Node.js instalados você está pronto para começar a desenvolver, porém precisamos pensar em um desenvolvimento moderno onde vários desenvolvedores possam trabalhar juntos em seu projeto bem como, quem sabe, a criação de uma esteira de CI/CD (se não souber o que é CI/CD não se preocupe, ainda vamos falar disso em disciplinas futuras).

Se você ainda não tem uma conta no GitHub (<https://github.com/>), entre no site e crie a sua conta gratuitamente. Ter um perfil público no GitHub pode ser uma ferramenta valiosa no processo de recrutamento para várias vagas relacionadas à tecnologia, especialmente para desenvolvedores. Nossa sugestão é que deixe este projeto público para que ele possa se somar a seu portfólio.

Após realizarmos a instalação e criarmos a nossa conta no github, está na hora de juntar tudo e iniciarmos o nosso projeto. Vamos iniciar configurando a integração do seu Visual Studio com o Github. Essa é uma configuração um pouquinho “mais complicada” então sugiro que assista ao vídeo abaixo:

<https://www.youtube.com/watch?v=7cNP3AE49Bg>

Passo 2 - Iniciando um novo projeto ReactJs

Agora temos tudo instalado então vamos criar o nosso projeto. Após configurar o github e o VisualStudio code, siga o roteiro abaixo para criar o primeiro projeto em node.js:

1. Abra “Prompt de Comando” do Windows e acesse a pasta onde irá inicializar o seu projeto. Se sua máquina for Linux ou Mac você deve adaptar a execução deste passo.

2. Inicialize um novo projeto React.js:

```
npx create-react-app nome-do-seu-projeto
```

3. Abra o VisualStudio Code e selecione a pasta que foi criada no passo anterior.

4. Crie ou altere o arquivo README.md com o seguinte conteúdo:

FACULDADE GRAN (<https://faculdade.grancursosonline.com.br/>)

Projeto Disciplina Projeto Integrador

5. No Visual Studio abra a aba Terminal e inicie o servidor de desenvolvimento:

```
npm start
```

Este comando iniciará o servidor de desenvolvimento e abrirá automaticamente o seu novo aplicativo React no navegador.

Esta é uma boa hora para configurar o git neste novo projeto e realizar um commit

DICA: Instalar Extensões Úteis no Visual Studio Code

Para facilitar o desenvolvimento com React, você pode instalar algumas extensões no Visual Studio Code:

ESLint: Ajuda a encontrar e corrigir problemas no seu código JavaScript.

Prettier - Code formatter: Formata automaticamente o seu código para mantê-lo consistente.

Simple React Snippets: Fornece snippets de código para React.

Para instalar extensões, clique no ícone de extensões na barra lateral esquerda (ou pressione Ctrl+Shift+X), procure as extensões desejadas e clique em "Install".

Passo 3 - Construção dos Artefatos no Frontend

O frontend, muitas vezes referido como o "lado do cliente", é responsável por interagir com o usuário da aplicação, portanto aspectos como usabilidade e experiência devem ser observados quando de sua construção. Neste projeto iremos construir 3 páginas conforme detalhado abaixo que deverão consumir o backend construído na atividade anterior.

1. Página de Produto:

Esta página é responsável por gerenciar as operações relacionadas aos produtos, como criar, ler, atualizar e deletar (CRUD). No contexto da nossa aplicação, ela irá interagir com o backend para armazenar informações sobre cada produto, como nome, descrição, preço e código de barras.

2. Página de Fornecedor:

Similar à página de produto, a página de fornecedor gerencia as operações relacionadas aos fornecedores. Ele cuida dos detalhes como nome do fornecedor, CNPJ, endereço e contato.

3. Página de Associação Produto/Fornecedor:

Esta página lida com a relação entre produtos e fornecedores. Sabendo que um produto pode ser fornecido por vários fornecedores e um fornecedor pode oferecer vários produtos, essa relação é muitas vezes chamada de "muitos para muitos". A página de associação permite associar um produto a um fornecedor, desassociá-los e consultar quais produtos são fornecidos por um determinado fornecedor (e vice-versa).