**Relatório do Projeto I**

**Nome:**

**Disciplina:**

Este projeto visa demonstrar técnicas de desenvolvimento de aplicações web utilizando o modelo *request* e *response* (requisição e resposta – em que um cliente faz uma requisição para um servidor e ele responde) dos verbos GET e POST partes do protocolo HTTP ,vide pág. 2, também conhecido como modelo cliente-servidor. Para isso foi criada uma aplicação que permite realizar o gerenciamento da base de dados, emulada por um arquivo JSON (JavaScript Object Notation), de uma determinada livraria digital, podendo realizar cadastro, remoção e venda dos itens dessa base.

**Introdução**:

O referente tema está relacionado ao desenvolvimento de um uma aplicação web utilizando os verbos GET e POST do protocolo HTTP para gerenciamento de uma livraria, de modo que o operador do sistema poderá adicionar novos livros, tanto físico como digital, e realizar a venda e a exclusão destes. Inicialmente o sistema dispoẽ de uma rota principal onde está o index.handlebars (/) (uma rota está diretamente relacionada a URL, ou seja, também funciona como um localizador de recurso entre outros) que por meio de um botão direciona o usuário para realização do cadastro dos novos livros onde deverão ser informados dados como título, autor, editora e preço. A rota /cadastro também possui três botões “Cadastrar”, o qual realiza a adição do livro na base de dados, “Ir para livros”, que direciona o usuário à rota /livros, onde será possível vendê-los e excluí-los e outro para retornar à página inicial (Página inicial). Nessa rota, de maneira resumida, será possível visualizar informações como quantidade de livros no estoque, valor total do estoque e valor total de vendas, bem como, excluir e vender qualquer das obras.

**Objetivos**:

Tem-se como principal objetivo, com a realização do projeto, demonstrar as técnicas de manipulação dos verbos GET e POST do protocolo HTTP no modelo *request* e *response* proporcionando dessa forma a possibilidade da manipulação do estoque e controle de caixa de uma determinada livraria digital.

**Atividades desenvolvidas**:

Durante o desenvolvimento das atividades foram utilizadas ferramentas que possibilitaram a criação do projeto de aplicação web, foram elas:

**Ferramentas:**

* Linguagem de programação JavaScript;
* Node.js;
* Biblioteca do npm;
* Framework Express;
* Protocolo HTTP com os verbos GET e POST;
* View engine handlebars;
* Fonte css aplicada do Bootstrap;
* Linguagem de marcação HTML;
* VS Code como ambiente de desenvolvimento;

A linguagem de programação escolhida para o projeto foi o JS (JavaScript) por ser simples e uma das principais linguagens de desenvolvimento web e de software. O JavaScript é categorizado como linguagem de criação de scripts ou uma linguagem interpretada sendo utilizada pela maioria dos navegadores de internet.

Somente é possível trabalhar com *JavaScript* fora dos navegadores através da plataforma do Node.js, que é o interpretador JavaScript. Ele é o responsável por desvincular o JavaScript dos navegadores, dessa forma, ele oferece a possibilidade da criação de aplicações *back-end*. O Node.js é basicamente composto pelo motor V8 da *Google*, responsável por interpretar os códigos JS, e a Libuv, biblioteca que fornece características *back-end* para o Node.

Outra característica do Node é que ele possui um dos maiores ecossistemas de bibliotecas, módulos e *plugins* do mundo, o gerenciador de pacotes NPM (*Node Package Manager*), que potencializa a utilização da linguagem no *back-end*.

No intuito de poder trabalhar com os verbos GET e POST do modelo cliente-servidor do protocolo HTTP (o protocolo HTTP (*hypertext transfer protocol*)

* o protocolo de transferência de hipertexto, que permite a comunicação entre o cliente e o servidor através do modelo *request* e *response* em que é feita uma requisição ao servidor e ele responde ao cliente. O protocolo HTTP se encontra na 7ª camada, camada de aplicação, do modelo OSI (*Open System Interconnection*)) foi usado o módulo Express do npm. O Express é um *framework* minimalista com um conjunto de poderosas ferramentas que possibilitam trabalhar de forma ágil. Ele é o principal *framework* e a principal ferramenta para criar aplicações web e trabalhar no *back-end* com o node.

Com a finalidade de renderizar os arquivos HTML (*Hypertext Market Language* – Linguagem de Marcação de Hipertexto) será usado o *Express-handlebars*, ele é um *template engineering,* ou seja, um motor devisualização. Ele traz muitas funcionalidades ao html como estruturas condicionais, de repetição e até mesmo exibir dados no HTML que estão armazenados em variáveis JS.

As folhas de estilo CSS tem como fonte a base do *Bootstrap*. O *Bootstrap* é um *framework front-end* que fornece estruturas de CSS para a criação de sites e aplicações responsivas de forma rápida e simples.

**Desenvolvimento:**

O usuário poderá acessar a aplicação por meio do navegador de internet acessando o endereço de url [http://localhost:3000](http://localhost:3000/cadastro/). Ao acessar essa rota, que faz uso do verbo GET, o cliente realizará uma requisição ao servidor que está hospedado que a hospeda, como resposta virá uma página web.

Em primeiro plano tem-se uma tela inicial com dois botões, através desses botões ele poderá ser direcionado ou para a rota /livros ou /cadastro a qual se pode adicionar livros em uma base de dados representada por um arquivo JSON. Nessa rota (página web), pode-se encontrar os campos título, autor, editora e preço. Logo abaixo se localizam dois botões, onde poderá cadastrar um novo livro e ir para a rota /livros. Quando o usuário clicar no botão “Cadastrar” um formulário será enviado de volta ao servidor, por meio do verbo POST (por uma questão de segurança foi utilizado o método POST com ele os dados trafegam pelo corpo da requisição, uma vez que com o GET, os dados trafegam pela url), esse formulário será armazenado na memória do Node.js.

No momento em que o usuário não tiver mais livros para cadastrar ele poderá visualizar os livros que foram cadastrados, além daqueles que estavam pré cadastrados por meio do arquivo .json, clicando no botão “Ir para livros”. Ao clicar nesse botão será feita mais uma requisição *(request)* ao servidor e ele responderá *(response)* pelo método GET direcionando-o a rota /livros.

A rota “livros” é basicamente composta por uma lista de livros com seus respectivos dados, além de dois botões: um para vender e outro para remover. Uma barra superior poderá ser vista. Nela contém informações como o valor total do estoque e o valor total de vendas, um botão para voltar a cadastrar novos livros e um campo de pesquisa para localizar livros na base de dados (ainda não implementado).

Todo livro que for vendido ou excluído, serão passados por parâmetro, ou seja, pela url, o seu valor e seu id (identificador único). O valor para ser somado ao estoque de venda e o id para para sua exclusão. Para isso, foram criadas duas rotas por meio do método GET chamadas /vender e /deletar. Já na exclusão, o valor enviado ao servidor será decrementado do valor total do estoque e o id para a remoção do item.

**Conclusões**:

Pode-se concluir que os resultados foram satisfatórios, uma vez que o Node.js é uma das ferramentas mais poderosas em termos de aplicações back-end, além de proporcionar toda a facilidade na implementação ele é muito leve, usa pouca memória RAM e faz um melhor aproveitamento da CPU. O Node.js está perto do topo em uma lista de frameworks e APIs, capaz de suportar um imenso número de requisições por minuto (em torno de 420.000 requisições por minuto). Diante disso foi possível comprovar o funcionamento do modelo cliente-servidor manipulando uma base de dados na memória, requisitar (*request*) arquivos HTML do servidor e receber os arquivos no lado cliente (*response*), receber dados por parâmetro e pelo corpo da requisição através dos métodos GET e POST do protocolo HTTP.