Trabalho Prático - Livraria JavaScript

Filipa Barreiro
Informática Web
Universidade da Beira Interior
Covilhã, Portugal
nº 44446

Larissa Goularte
Informática Web
Universidade da Beira Interior
Covilhã, Portugal
nº: 43070

I. Introduction

A. Enquadramento

O presente relatório terá sido realizado no âmbito da Unidade Curricular (UC) **Scripting**, disciplina dirigida aos alunos de segundo ano de licenciatura em Informática Web, leccionada pelo professor Doutor Paul Andrew Crocker. O conceito, planeamento e desenvolvimento da proposta deste trabalho de grupo, visa a aplicação de conceitos teóricos retidos ao longo de percurso no primeiro semestre, proporcionando uma enorme oportunidade de implementar este tipo de pesquisa para fins curriculares, e possivelmente, profissionais.

B. Motivação

Vivemos num mundo onde as ciências computacionais crescem todos os dias, sendo uma das maiores áreas de investigação, estudo e empreendedorismo. É fundamental dinamismo, curiosidade e iniciativa como vínculos à pesquisa e obtenção de uma maior diversidade de informação, bem como ao desenvolvimento de um método eficaz e ritmado de trabalho. Para tal, projetos como este são considerados essenciais em cursos dentro da área, pois motivam o aluno a gerir o seu tempo e capacidades de forma a que consiga alcançar os objetivos propostos, desenvolvendo o seu intelecto e *skills*.

Uma das grandes motivações para a realização deste trabalho terá sido a aplicação de conhecimentos relativamente mais aprofundados adquiridos nesta UC. Aquando a inicialização do desenvolvimento do tema, este revelou-se desafiante, interessante e bastante enriquecedor para os nossos futuros como profissionais na área das tecnologias, sendo por si só, uma enorme potencial de inovação e escalabilidade, sendo estes os principais fatores de motivação para a implementação do trabalho.

C. Objetivos

Õ objetivo principal deste trabalho, recai no desenvolvimento de uma livraria em JavaScript, onde será possível comprar, vender livros e ainda visual o crédito da loja e stock.

D. Organização do documento

De modo a refletir todo o trabalho que terá sido desenvolvido, este documento encontra-se estruturado da seguinte forma:

1) O primeiro capítulo – **Introdução** – Apresenta o projeto, a motivação para a sua escolha, o enquadramento para o mesmo, os seus objetivos e a respetiva organização do documento.
2) O segundo capítulo – **Desenvolvimento** – Apresenta o desenvolvimento do projeto e como ele funciona.
3) O terceiro capítulo – **Tecnologias e Ferramentas Utilizadas** – Descreve os conceitos mais importantes no âmbito deste projeto, as tecnologias utilizadas no desenvolvimento da aplicação.
4) o quarto capítulo – **Funcionalidades JavaScript** – Apresenta as funcionalidades de JavaScript e como estas se relacionam no trabalho.
5) o quinto capítulo – **Conclusão** – contempla as conclusões finais e o trabalho futuro do dado projeto.

II. Desenvolvimento do trabalho

A. Introdução

O trabalho realizado tem 2 versões. A primeira é uma versão simples que utiliza como DataStore apenas a memória do Servidor, enquanto que a segunda versão é mais rica em funcionalidades e que utiliza MongoDB, que permite a persistência dos dados usando documentos semelhantes a JSON com esquemas.

B. Funcionamento do programa

A API começa com uma página inicial http://localhost:3000/, e nem todas as funcionalidades estão visíveis ao utilizador.

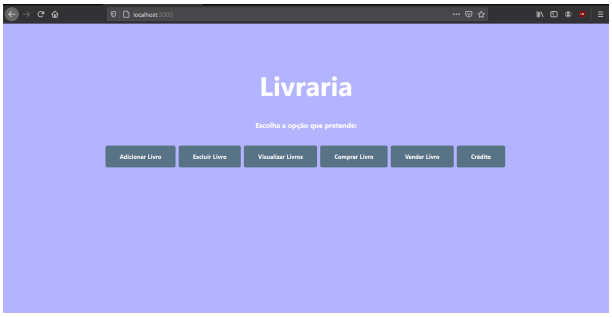

Fig. 1. Página inicial

Na primeira versão (memória do servidor, sem MongoDB), adicionar um livro faz-se em http://localhost:3000/adicionar, através do método POST. Cada objeto criado é armazenado num array loja.livros. A loja é uma variável com o nome "Livraria". O servidor responde com o envio do objeto para a página.

Para eliminar um livro recorre-se ao URL http://localhost:3000/delete, em que "titulo" corresponde ao título do livro. Ou seja, um livro é eliminado pelo seu título através de DELETE. Esta funcionalidade não pode ser realizada através de um formulário, como a funcionalidade adicionar livro. Na versão 1 todas as páginas HTML são chamadas a partir do método GET.

Em http://localhost:3000/livros é possível ver a lista de objetos criados aquando da iniciação de cada sessão da API, através do método GET. Com o método PUT, em http://localhost:3000/livros/:titulo é possível ver os dados do objeto cujo título é substituído no URL.

Comprar livros faz-se através do URL http://localhost:3000/comprar. O utilizador deverá escrever o título do livro e a quantidade que pertende comprar. O servidor irá responder ao pedido com o envio de uma mensagem de sucesso, ou pelo, contrário, que ocorreu um erro ou que o livro não existe. A resposta é enviada em http://localhost:3000/comprado e ambas foram lançadas com o método GET.

Para vender livros basta aceder ao URL http://localhost:3000/vender. Aqui o utilizador insere o título do livro, a quantidade e o formato. Se o livro existir e estiver em stock, é lançada uma mensagem de sucesso (com GET) para http://localhost:3000/vendido. Caso contrário, para o mesmo endereço, é lançada uma mensagem de erro: ou que o stock é insuficiente ou o livro não existe.

Na versão 2, em que foi utilizado o MongoDB, existe um ficheiro livro.js com a documentação dos objetos que serão manipulados na base de dados. Cada objeto Livro tem um título, autor, data e editora. Esta versão é mais sensível aos formatos dos livros. Por esse motivo, existe um ficheiro livroFormato.js, que guarda os objetos numa base de dados à parte do mongo. Cada formato tem um título com referência a Livro, um preço de compra, um preço de venda, um formato e um stock. Existe, também, um ficheiro loja.js que armazena informações da loja: o nome e o crédito (saldo, diferença entro o lucro e o débito).

Para chamar as páginas HTML e EJS na suganda versão, foram usados apenas métodos USE. Para criar um novo objeto Livro, é necessário aceder http://localhost:3000/adicionar, preencher o formulário e enviar. Para eliminar uma um objeto Libro da base de dados, o utilizador deverá dirigir-se a http://localhost:3000/delete e colocar o título que pretende eliminar no formulário. Caso o título exista, o objeto Livro (e objetos LivroFormato se tiver) será eliminado. Uma mensagem de sucesso será lançada.

É possível pesquisar um título pelo nome do mesmo. Para isso, basta ir a http://localhost:3000/pesquisa/?titulo=nome, em que "nome" é substituído pelo título que se quer pesquisar. O endereço http://localhost:3000/pesquisa/ necessita do parâmetro nome, ou é enviada uma mensagem que o título não existe. O mesmo acontece quando é introduzido de facto um nome que não existe.

Para listar todos os títulos, o URL é http://localhost:3000/all. Através de um ficheiro .ejs, é possível ver para cada livro na base de dados o nome do mesmo, do autor e o ano em que foi lançado.

Para que o utilizador compre um livro, tem de se dirigir ao endereço http://localhost:3000/comprar. Aqui, o utilizador seleciona o livro que pretende comprar e o formato. De seguida, insere a quantidade que pretende e define o preço de compra e de venda. Ou seja, fornece o stock de um livro de um certo formato. Se esse formato já estiver disponível, a quantidade inserida é adicionada ao stock. Se não, é deixado disponível. Ao crédito da loja é descontado o preço total (quantidade vezes o preço de compra inseridos). A confirmação é feita em http://localhost:3000/comprado e é enviada uma mensagem que confirma a realização da compra.

Para vender um livro, neste caso a um potencial cliente, o endereço é http://localhost:3000/vender. O utilizador neste caso apenas tem de selecionar o livro, a quantidade e o formato. Caso o formato exista (LivroFormato) em stock suficiente, a venda é um sucesso, ao crédito da loja é acrescentado o preço total (quantidade vezes o preço de venda) e é enviada uma mensagem de sucesso. Caso o formato exista mas em stock insuficiente, é enviada uma mensagem a relatar esse mesmo caso e o mesmo acontece quando o formato não estiver disponível para venda. A confirmação é feita em http://localhost:3000/vendido.

Finalmente, em http://localhost:3000/info é possível ver as informações da loja através de um ficheiro .ejs, ou seja, o nome e o crédito.

III. Ferramentas Utilizadas

No seguinte capitulo, serão apresentadas breve e sucintamente as tecnologias e ferramentas utilizadas na implementação do projeto desenvolvido. Durante a elaboração deste projeto foram utilizadas diversas tecnologias e ferramentas tais como, *Postman*, *MongoDB* e *NodeJS*. Todos estes estiveram sempre presentes tendo sido necessário a pesquisa por novas formas de codificação e implementação, permitindo a sua aprendizagem e capacidade de interiorizar o funcionamento de algumas bibliotecas, *frameworks* e linguagens utilizadas no desenvolvimento do dado projeto.

IV. Comandos de Linux Utilizados

Uma linguagem script é qualquer linguagem que for usada para escrever um script e que permite uma interpretação mais complexa dos nossos comandos.

O Node é basicamente uma ferramenta que executa o JavaScript fora do navegador. Quando instalado, o Node.js é chamado pelo terminal, não por um browser.

V. Funcionalidades do JavaScript

Uma linguagem script é qualquer linguagem que for usada para escrever um script e que permite uma interpretação mais complexa dos nossos comandos.

O Node é basicamente uma ferramenta que executa o JavaScript fora do navegador. Quando instalado, o Node.js é chamado pelo terminal, não por um browser.

Há várias funcionalidades disponibilizadas para o Node.js, fazendo com que seja posível fazer coisas com JavaScript que não se conseguem fazer com o ambiente de trabalho e o navegador, como por exemplo o acesso a arquivos.

O Node.js possibilitou a criação de várias outras ferramentas para usar JavaScript fora do navegador, e hoje em dia é um conhecimento obrigatório para quem quer trabalhar com JavaScript.

Express é uma estrutura de aplicativo da web para o Node.js e é projetado para construir aplicativos da Web e APIs. E considerado a estrutura de servidor padrão para o Node.js. Ele permite a passagem facilitada do navegador para back-end e economiza trabalho de programação.

EJS é a sigla para Embedded JavaScript templates, que é uma linguagem de templates que permite a criação de HTML com apenas JavaScript. A utilização do EJS simplifica a vida do developper, que poderá utilizar esta linguagem em todo o desenvolvimento da aplicação.

Como foi dito anteriormente, o MongoDB é um banco de dados orientado a documentos, que usa documentos semelhantes a JSON com esquemas. É possível receber a entrada no navegador com JavaScript, transportar a demanda para o back-end ainda com a linguagem e, por fim, salvar um objeto no MongoDB ainda com o JavaScript.

VI. Conclusão

A. Conclusão Principal

Tendo concluído este projeto final da UC de Scripting, é possível observar o quão importante foi o desenvolvimento deste relatório, não obstante da parte prática. Através da análise deste relatório é possível obter toda a informação relativa ao processo de desenvolvimento e o estado da aplicação final.

B. Trabalho Futuro

Como a finalização desta proposta de projeto é possível afirmar que grande parte das funcionalidades que inicialmente teriam sido projetadas para serem implementadas ficaram concluídas. Porém, tendo em conta que a maioria das funcionalidades estão presentes na aplicação existem alguns aspetos que gostaria de melhorar num futuro próximo. Alguns destes aspetos seriam:

- Relatório de vendas;
- Alteração de um ou mais dados sem ser necessária a eliminação do objeto por completo.