Introdução

Este projeto desenvolvido pelos alunos de Tecnologia e Programação de Sistemas de Informação tem como objetivo de criar uma aplicação de adoção de animais de estimação utilizando as tecnologias de servidor para a gestão das mesmas. Utilizando o **NodeJs** como servidor aplicacional e o **MySql** para persistência dos dados. Ou seja, que contenha as operações GET,POST,PUT,DELETE sobre um SGBD (Sistema de Gestão de Base de Dados) MySql.

Este relatório foi dividido em duas partes:

**Manual Técnico**: no qual servirá para quem for instalar o produto e possui as seguintes secções:

* Breve descrição das tecnologias base.
* Lista de Tecnologias utilizadas e para que efeito; Ex: javascript jquery,mysql,

**Manual de Utilizador:** no qualservirá para um utilizador do mesmo (nas várias perspetivas ou papéis) e contem as seguintes informações:

* Lista das várias funcionalidades. Ou seja, a gestão, a consulta , criação e eliminação dos usuários, animais e organizações
* Cada funcionalidade indica as suas características e toda a informação relevante.

Base de Dados

Foi utilizado o programa **MySQL** para a criação da base de dados para o projeto. Foram criadas as seguintes Tabelas:

* **Usuario**

|  |  |
| --- | --- |
| Nome | Tipo de dado |
| id\_usuario | Int (Primary Key) |
| nome | Varchar(50) |
| email | Varchar(50) |
| morada | Varchar(50) |
| idade | Tinyint |
| telefone | Varchar(15) |
| sexo | Varchar(10) |
| nome\_usuario | Varchar(20) |
| senha | Varchar(25) |

* **Animal**

|  |  |
| --- | --- |
| Nome | Tipo de dado |
| id\_animal | Int (Primary Key) |
| tipo | Varchar(15) |
| raca | Varchar(15) |
| cor | Varchar(15) |
| idade | Tinyint |
| sexo | Varchar(10) |
| localidade | Varchar(20) |

* **Organização**

|  |  |
| --- | --- |
| Nome | Tipo de dado |
| id\_organizacao | Int (Primary Key) |
| nome\_organizacao | Varchar(15) |
| localidade | Varchar(20) |

* **Login**

|  |  |
| --- | --- |
| Nome | Tipo de dado |
| Id\_login | Int (Primary Key) |
| Usuario\_nome | Varchar(50) |
| Usuario\_senha | Varchar(50) |
| Id\_usuario | Int (Foreign Key) |

Tecnologias Utilizadas

Para a realização deste projeto foram utilizadas as seguintes tecnologias:

* **HTML** - (Linguagem de Marcação de Hipertexto) é o bloco de construção mais básico da web. Define o significado e a estrutura do conteúdo da web.
* **CSS** - Cascading Style Sheets (CSS) é um mecanismo para adicionar estilo (cores, fontes, espaçamento, etc.) a um documento web.
* **JavaScript** - (frequentemente abreviado como JS) é uma linguagem de programação interpretada estruturada, de script em alto nível com tipagem dinâmica fraca e multiparadigma (protótipos, orientado a objeto, imperativo e, funcional).
* **NodeJS** - Node.js é um software de código aberto, multiplataforma, baseado no interpretador V8 do Google e que permite a execução de códigos JavaScript fora de um navegador web.



Figura 3- NodeJS

HTML

Link: <https://www.w3schools.com/html/>

Neste projeto foi utilizado o HTML para a construção da nossa página Web, juntamento com o CSS (aparência/apresentação) e o JavaScript (funcionalidade/comportamento de uma página da web).

Foram utilizados os métodos GET,POST,PUT,DELETE para os usuários, animais e organizações. As figuras a seguir mostram a utilização destes métodos.

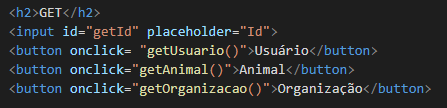


Figura 4 – GET

Foi utilizado o elemento **Form** para a construção do formulário, juntamento com o atributo **Button** que enviar os dados recolhidos no formulário para o local indicado e o **Input** que é usado para criar controles interativos para formulários baseados na web para receber dados do usuário.

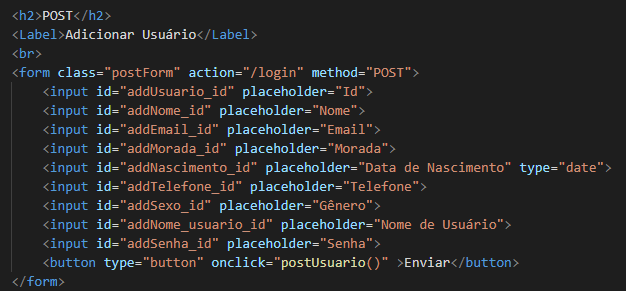


Figura 5- POST – Adicionar Usuário

O utilizador insere nos inputs as informações referentes ao que é pedido, após a inserção das informações, prime o botão para enviar essas informações para a base de dados onde fica guardado as informações dos Usuários. A seguir temos o Post da Criação dos animais, funcionando da mesma maneira.

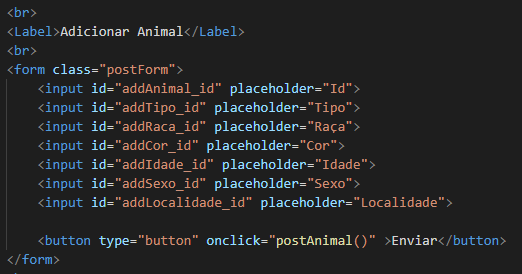
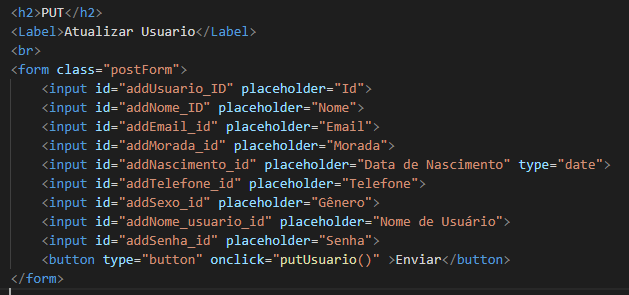


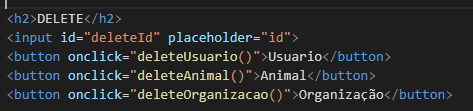
Figura 6 – POST - Adicionar Animal

Foi o utilizado o método PUT, para fazer a atualização dos dados que já existiam, no exemplo abaixo verificamos a atualização do Usuário, no qual o utilizador pode alterar qualquer informação, para salvar a alteração ele deve premir o botão enviar para atualizar na base de dados.



*Figura 7 – PUT – Atualizar Usuário*

Foi o utilizado o método DELETE, para fazer a eliminação dos elementos que já existiam, para fazer a eliminação, ele deve escolher o ID e qual o tipo de elemento que pretende eliminar, após isso deve premir o botão referente ao elemento para eliminar o elemento da base de dados.



*Figura 8 - DELETE*

CSS

Link: <https://www.w3schools.com/css/>

O CSS foi utilizado para alterar a cor do fundo, margem, fonte e espaçamento entre parágrafos e outras coisas. Não foi tão utilizado para a resolução do nosso projeto.



JavaScript

Link: <https://www.w3schools.com/js/>

Utilizando o JavaScript, foram criadas as funções: getUsuario, getAnimal, getOrganizacao, postUsuario, postAnimal, postOrganizacao, putUsuario, putAnimal, putOrganizacao, deleteUsuario, deleteAnimal, deleteOrganizacao.

Abaixo temos um exemplo de cada tipo de função que foi referida acima. O processo é semelhante em quase todos, só mudando o elemento que vai ser trabalhado.

Funções

* getUsuario() - Esta função recebe o id que foi inserido pelo utilizador e vai buscar na base de dados toda a informação referente a esse ID. No caso a informação referente ao usuário.



* postAnimal() - Esta função recebe as informações que foram inseridas pelo utilizador e vai enviar para a base de dados toda a informação para ser armazenada. No caso abaixo a informação referente a criação de um Animal.



* putOrganizacao() - Esta função recebe as informações que foram inseridas pelo utilizador e vai atualizar as informações antigas e enviar novamente para a base de dados para ser armazenada. No caso abaixo a informação referente a atualização de uma organização.



* deleteUsuario() - Esta função recebe o id que foi inserido pelo utilizador e vai apagar toda a informação referente a esse ID. No caso a informação referente ao usuário.



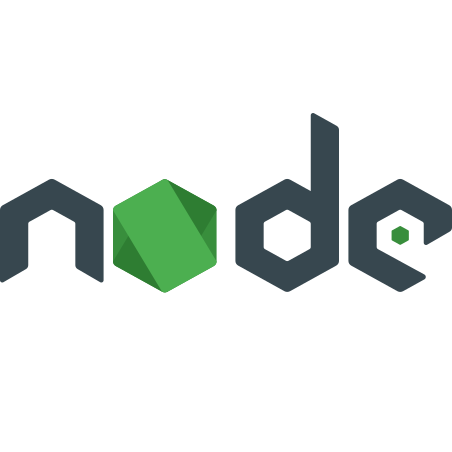
NodeJS

Link: w3schools.com/nodejs/

Node.js é uma plataforma para construir aplicativos de servidor rápidos e escaláveis ​​usando JavaScript. Node.js é o tempo de execução e npm é o gerenciador de pacotes para módulos Node.js.

O Visual Studio Code oferece suporte para as linguagens JavaScript e TypeScript prontas para usar, bem como para depuração Node.js. No entanto, para executar um aplicativo Node.js, você precisará instalar o tempo de execução Node.js em sua máquina.

Utilizamos o NodeJS e o Express para criar nosso servidor web.



Package.json

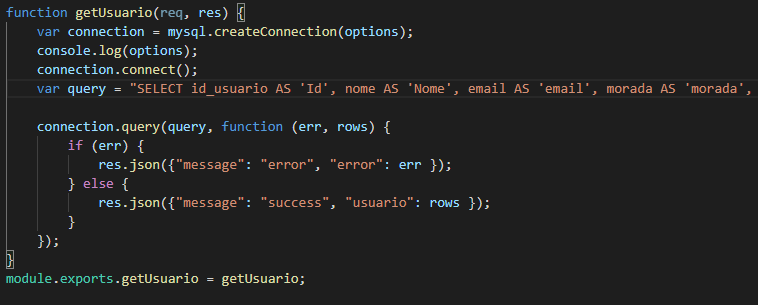
O package.json é um repositório central de configuração para ferramentas criado para o NodeJS.



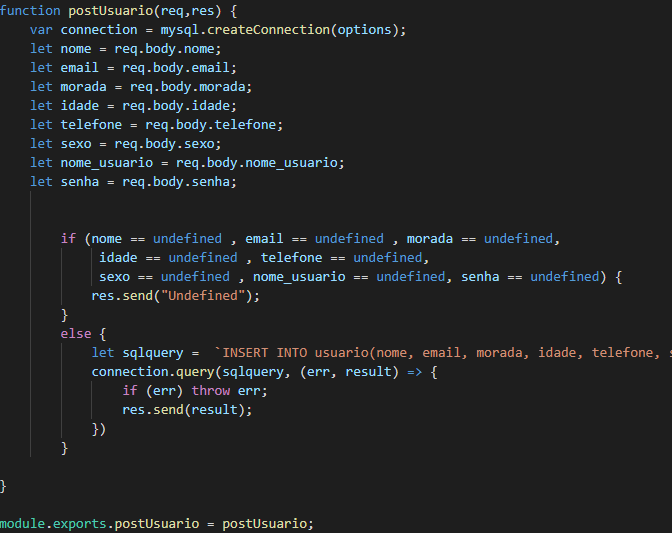
Request-handler

O request-handler é um script que é usado para definições das funções que serão executadas em resposta a cada pedido (uma por cada caminho). O request-handler utiliza o mysql para fazer a gestão dos dados. Foram criadas varias funções para a recolha de dados, como : getUsuario, getAnimais, getOrganizacao, etc. As funções recebem os dados do mysql e com isso utilizam e fazem as verificações e em seguida exportam os objetos. Conseguimos ver isso com as funções abaixo.

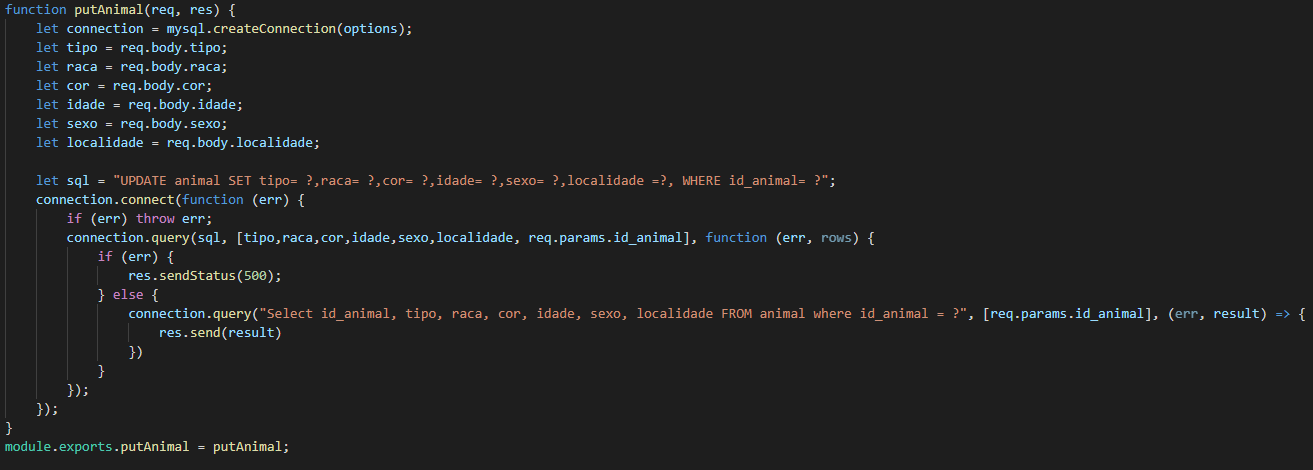
**Função - getUsuario**



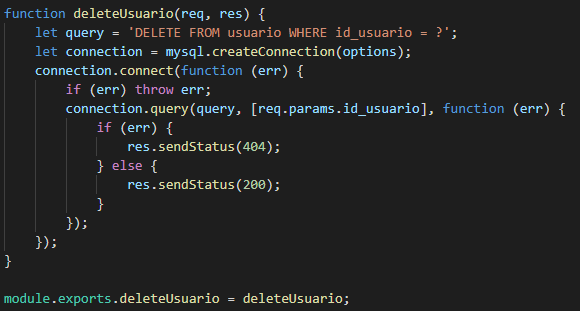
**Função - postUsuario**



**Função - putAnimal**



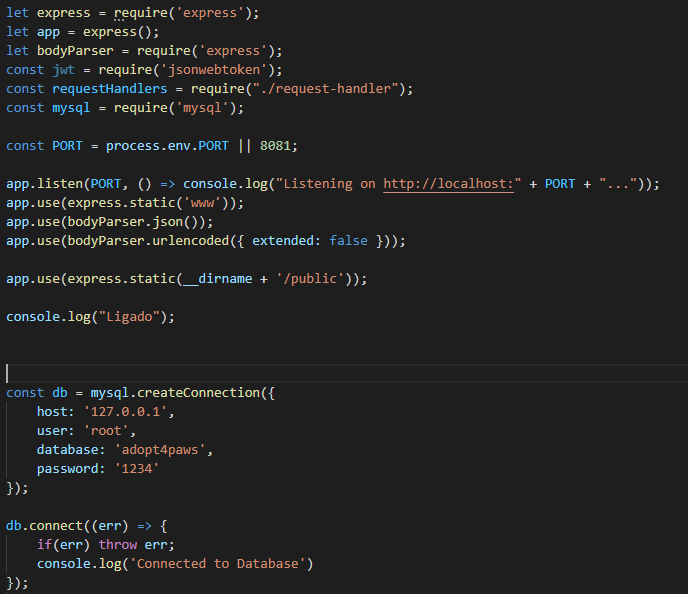
**Função - deleteUsuario**



Express

Express é uma estrutura de aplicativo da web Node.js mínima e flexível que fornece um conjunto robusto de recursos para desenvolver aplicativos da web e móveis. Ele facilita o rápido desenvolvimento de aplicativos da Web baseados em Node. A seguir estão alguns dos principais recursos da estrutura Express -

* Permite configurar middlewares para responder a solicitações HTTP.
* Define uma tabela de roteamento que é usada para realizar diferentes ações com base no método HTTP e URL.
* Permite renderizar páginas HTML dinamicamente com base na passagem de argumentos para modelos



**Server.js**



**Server.js**

Para a utilização do Express precisamos chamar o módulo express, como foi usado na linha 1 (Figura Server.js) **let express = require (“express”);**

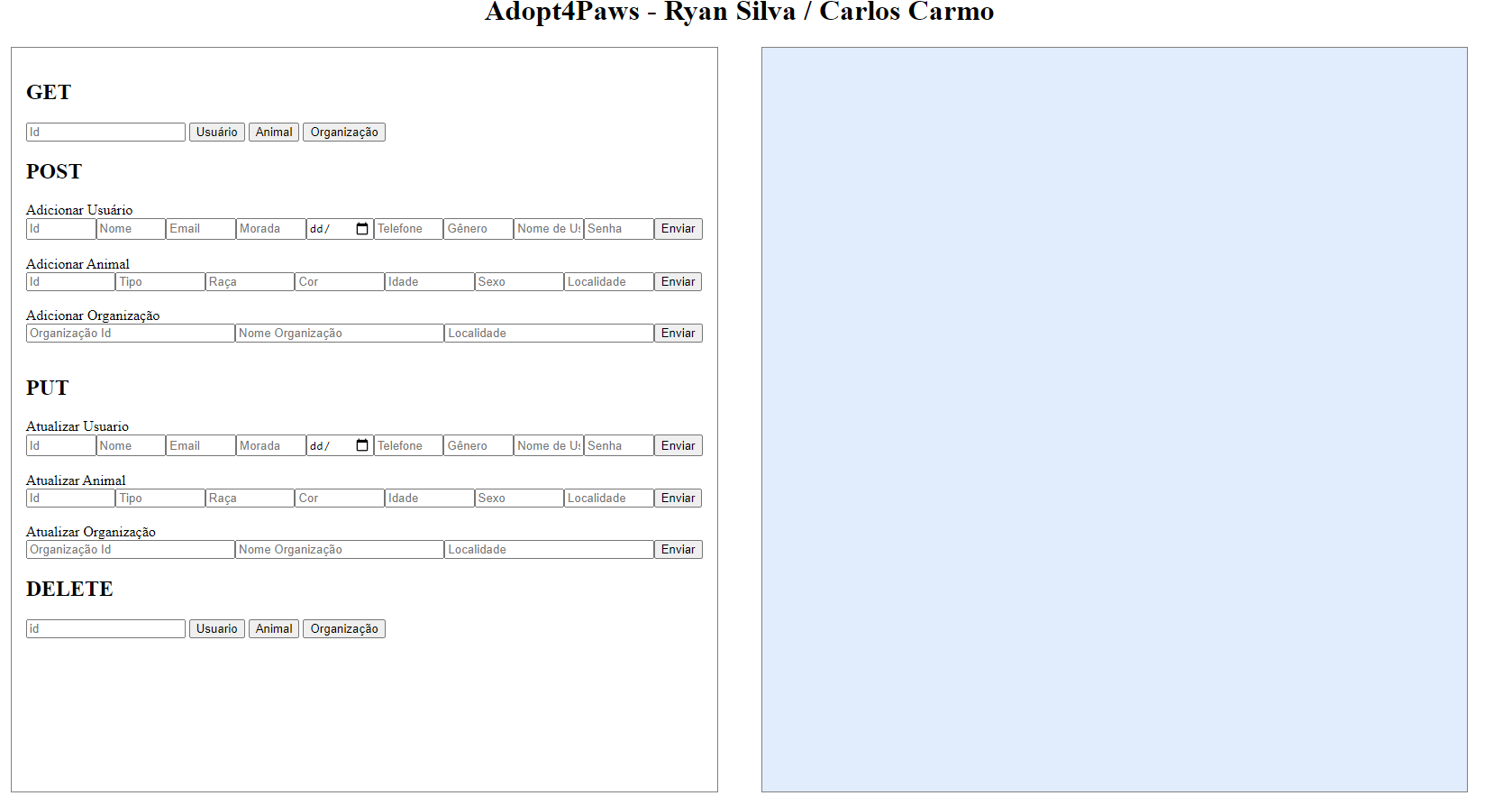
Foi utilizado o roteamento que é como um aplicativo responde a uma solicitação do cliente para um ponto de extremidade específico, que é um URI (ou caminho) e um método de solicitação HTTP específico (GET, POST e assim por diante).

* Get para chama a função para receber os dados referentes ao pedido , como exemplo: app.get(“/id\_usuario”,requestHanldlers.getUsuario).
* Post para chamar a função e para enviar os dados referentes ao pedido , como exemplo: app.post(“/”,requestHanldlers.postUsuario).
* Put para chamar a função e para enviar os dados referentes ao pedido , como exemplo: app.post(“/id\_usuario”,requestHanldlers.putUsuario).
* Delete para chama a função e para eliminar os dados referentes ao pedido , como exemplo: app.delete(“/id\_usuario”,requestHanldlers.deleteUsuario).

A função app.listen(8081, function() ) do Express inicia um socket UNIX e escuta as conexões em um caminho fornecido. Ou seja quando o servidor for ligado, será ligado na **localhost: 8081**. Não é possível iniciar o Express sem dizer onde escutar.

Manual de Utilizador

Página Principal

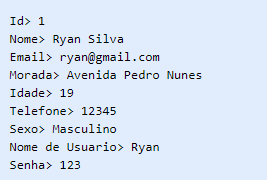
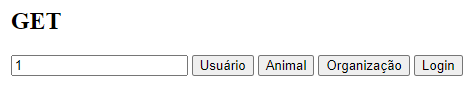


Na Página Principal apresentam-se 2 Formulários:

* Formulário a esquerda – Neste formulário podemos tanto: Ver, Adicionar, atualizar ou Eliminar um dos elementos que possuem, no caso os elementos são: usuários, animais e organização.
* Formulário a direita – Neste formulário conseguimos verificar todas as ações que foram realizadas no formulário a esquerda, como por exemplo os GET,POST,PUT,DELETE referente aos elementos.

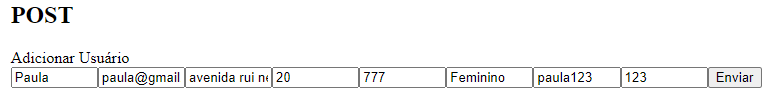
GET

* O utilizador insere o ID referente ao seu desejo, após isso prime o botão referente ao elemento que quer observar. Após isso será mostrado no formulário a direita o seu pedido.



POST

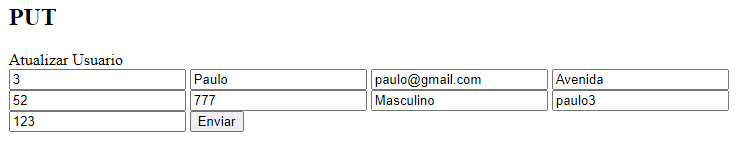
* O utilizador insere as informações referente ao elemento, respeitando o tipo de dados, após a inserção da informação, prime o botão enviar para que a informação inserida seja enviada para a base de dados.





PUT

* O utilizador pode atualizar as suas informações que foram inseridas anteriormente, atualizando os campos que deseja , prime o botão enviar para atualizar a informação na base de dados.



DELETE

* O utilizador insere o ID referente ao seu desejo de eliminação, após isso prime o botão referente ao elemento que quer eliminar. Após isso o elemento será apagado.





Conclusão

Neste relatório abordamos os Manuais Técnicos e de Utilizador que possuíram os seguintes tópicos: Base de Dados, Técnicas Utilizadas, HTML, CSS, NodeJS, JavaScript, Express e como utilizar o nosso servidor web.

Cumprimos os objetivos que nos foram propostos, porém tivemos dificuldades em algumas parte, como na montagem do servidor web e na ligação com o MySQL, porém conseguimos resolver esses problemas.

Este trabalho foi muito importante para a nossa compreensão sobre a matéria de PIS uma vez que nos permitiu compreender melhor e desenvolver nossas competências de investigação.