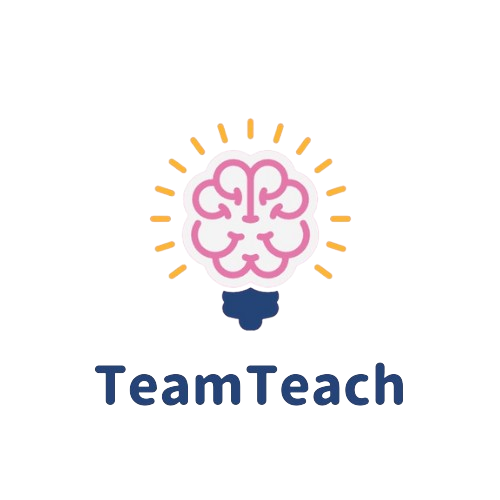


Agrupamento de Escolas Manuel Teixeira Gomes

Técnico de Gestão e Programação de Sistemas Informáticos

Relatório da Prova de Aptidão Profissional

2021/2024

Portimão, março de 2024

Ivan Kelya Balan | Nº12 | 12ºF

Professor Orientador: Tiago Teodoro

# **Índice**

[Relatório de PAP 1](#_Toc1928718213)

[Índice 1](#_Toc377429317)

[Índice de ilustrações 2](#_Toc1199715751)

[Agradecimentos 3](#_Toc1259068277)

[Introdução 4](#_Toc1153486928)

[Fundamentação 5](#_Toc134029507)

[Metodologias 6](#_Toc1063148426)

[Software 8](#_Toc608204518)

[Visual Studio Code 9](#_Toc1797486482)

[WampServer 10](#_Toc412082296)

[Git 10](#_Toc166626408)

[GitHub 12](#_Toc1924905140)

[MySQL 12](#_Toc812296207)

[JavaScript 13](#_Toc427882440)

[Fases do projeto 13](#_Toc1982371367)

[1ª Fase 14](#_Toc882675953)

[2ª Fase 14](#_Toc1516319309)

[3ª Fase 14](#_Toc453916546)

[4ª Fase 14](#_Toc1320650240)

[5ª Fase: 15](#_Toc1885350910)

[6ª Fase 15](#_Toc191456013)

[Mapa de Gantt 16](#_Toc1886662829)

[DER do projeto de PAP 17](#_Toc1697972183)

[Modelo Relacional 18](#_Toc1694891127)

**Índice de ilustrações**

*Figura – 1 Estrutura do projeto (Backend) - 7*

*Figura – 2 Logo do Visual Studio Code – 8*

*Figura – 3 Logo do WampServer – 9*

*Figura – 4 Logo do Git – 9*

*Figura - 5 Logo do GitHub – 10*

*Figura - 6 Logo do MySQL – 10*

*Figura - 7 Logo do JavaScript – 11*

# **Agradecimentos**

Gostaria de expressar a minha gratidão a todos os que apoiaram e ajudaram a concretizar com êxito o meu projeto. Sem a vossa ajuda, este sucesso teria sido muito mais difícil de alcançar.

Em primeiro lugar, gostaria de agradecer ao Professor Tiago Teodoro e ao Tiago Lopes pela sua orientação como bons professores e mentores.

Agradeço também ao meu colega Andrii Yanitskyi, que partilhou a sua experiência e ideias, contribuindo para a diversidade e profundidade do meu projeto.

Não posso deixar de mencionar a minha família, pois sem a sua compreensão e apoio esta viagem teria sido muito mais difícil. As vossas palavras de incentivo e o vosso apoio constante foram essenciais para mim.

Este projeto tornou-se realidade graças a vós e estou profundamente grata pelo vosso contributo. Obrigada pela vossa bondade, ensinamentos e apoio!

# **Introdução**

O objetivo deste projeto é criar plataformas em linha onde os professores possam partilhar os seus planos de aulas e apresentações, bem como trocar experiências e ferramentas de aprendizagem.

TeamTeach é uma plataforma em linha concebida para professores de todas as disciplinas. Permite aos professores partilharem os seus planos de aula e apresentações, trocarem materiais e ferramentas úteis e criarem equipas para trabalharem em conjunto em projetos educativos.

O projeto visa apoiar o desenvolvimento profissional dos professores e melhorar a qualidade da educação em geral.

# **Fundamentação**

TeamTeach é um projeto criado com o objetivo de facilitar a troca de conhecimentos e experiências entre professores de diferentes disciplinas e especializações. A inspiração para criar esta plataforma foi o professor José Maria Rasquinho. Foi ele que me deu a ideia base para conseguir realizar este projeto.

Foi perceptível que os professores eram obrigados a gastar muito tempo a desenvolver aulas e apresentações do zero, o que já é um grande peso visto que já passam tanto tempo dedicados a esta profissão. Por isso, decidi desenvolver uma plataforma online onde os professores possam compartilhar os seus materiais e experiências, facilitando o processo de preparação das aulas.

O objetivo é fazer do TeamTeach uma ferramenta integral para professores. Acredito que a partilha de conhecimento e experiência ajudará os professores a melhorar a qualidade da educação e a criar aulas mais interessantes e eficazes. Espero que o meu projeto contribua para o desenvolvimento profissional dos professores e melhore o nível de ensino em geral.

# **Metodologias**

**Devias de falar da metodologia Kanban também.**

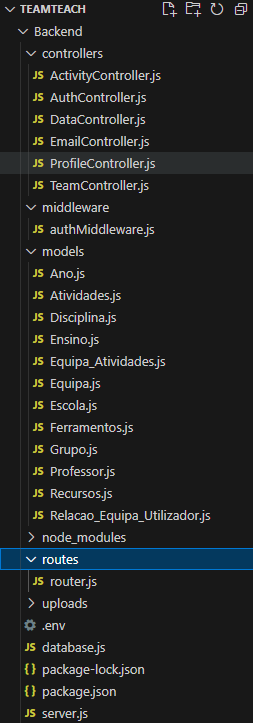
Neste projeto, utilizei uma API baseada em *Node.js*. Criei o lado do servidor da minha aplicação utilizando Node.js e a estrutura Express.js. Desenvolvi uma API para interagir com o lado cliente da aplicação, definindo rotas (*routes*), modelos (*models*) e controladores (*controllers*) para diferentes pedidos.

Os conceitos de *router*, *models* e *controllers* desempenham um papel importante na organização do código do servidor numa aplicação Web.

As *routes* são a forma de definir como a aplicação deve responder aos vários pedidos do cliente. Especificam que URLs devem ser associados a que manipuladores para servir os pedidos e devolver as respostas adequadas.

Os *models* representam os dados na nossa aplicação. Definem a estrutura dos dados e os métodos para os manipular, como a leitura, a escrita e a atualização. Normalmente, representam tabelas de bases de dados ou outros objetos de dados na aplicação.

Os *controllers* são onde reside a lógica comercial da aplicação. Recebem pedidos das rotas, interagem com os modelos apropriados para efetuar operações nos dados e devolvem as respostas ao cliente. Os controladores ajudam a partilhar a responsabilidade entre as rotas e os modelos, tornando o código mais modular e fácil de gerir.



*Figura – 1 Estrutura do projeto (Backend)*

# **Software**

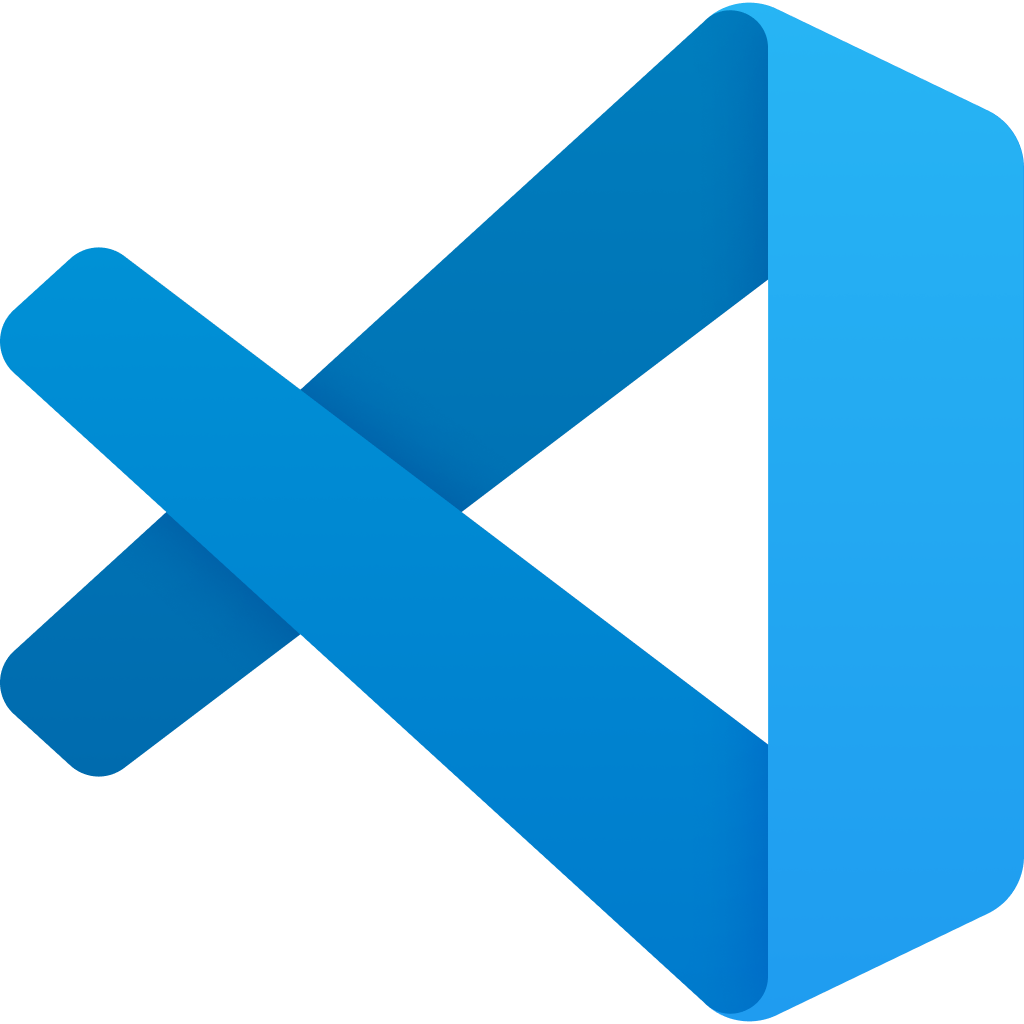
Neste projeto, foram utilizadas diversas ferramentas, aplicações e linguagens de programação, tais como:

Ferramentas / Aplicações:

* Visual Studio Code 2024
* WampServer
* Git
* GitHub

Linguagens de programação:

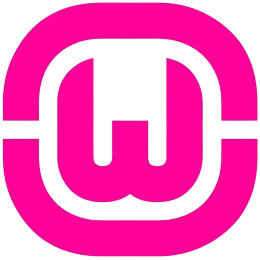
* MySQL
* JavaScript



*Figura – 2 Logo do Visual Studio Code*

## **Visual Studio Code**

O Visual Studio Code (VS Code) é um editor de código livre e de fonte aberta da Microsoft. Fornece muitas funcionalidades para o desenvolvimento de software, tais como realce de sintaxe, preenchimento automático, integração com o Git, depuração e extensibilidade através de plug-ins.

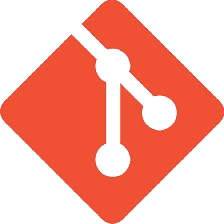


*Figura – 3 Logo do WampServer*

## **WampServer**

O WampServer é um pacote de software onde já se encontra tudo o que é necessário para criar sítios Web no computador local. Inclui um servidor Web, uma base de dados e PHP, o que me permite configurar rapidamente um ambiente local para desenvolver aplicações Web no Windows.

É cómodo porque não tenho de instalar cada componente separadamente - está tudo pronto a usar de uma só vez.

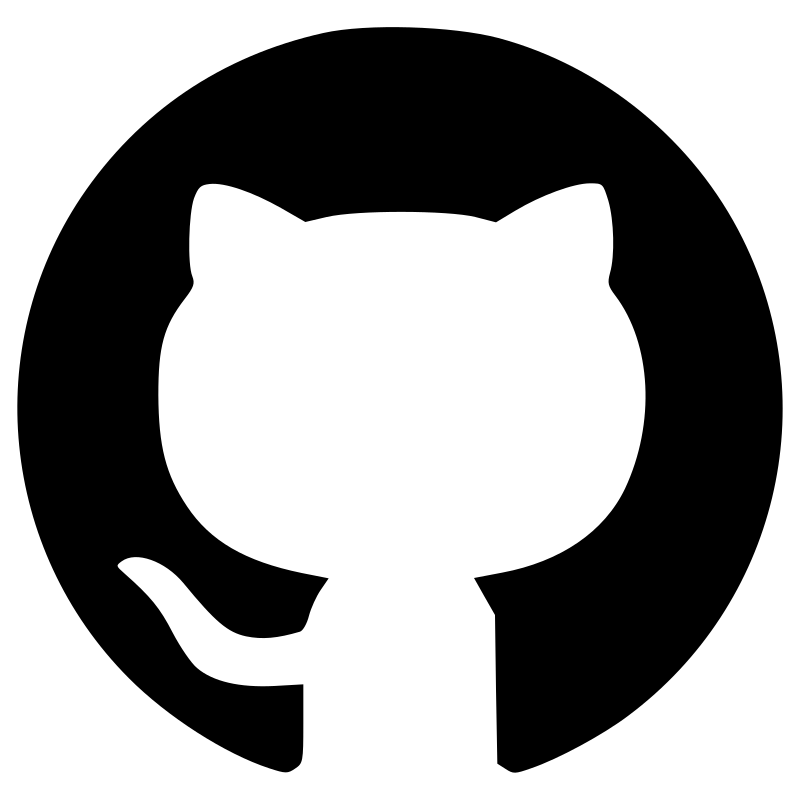


*Figura – 4 Logo do Git*

## **Git**

O Git é um sistema de controlo de versões que ajuda a acompanhar as alterações do código. Posso utilizá-lo para guardar diferentes versões dos meus ficheiros, alternar entre elas, anular alterações ou fundir diferentes versões.

Isto é muito útil para colaborar em projetos com outros programadores e para manter a integridade do meu código.



*Figura - 5 Logo do GitHub*

## **GitHub**

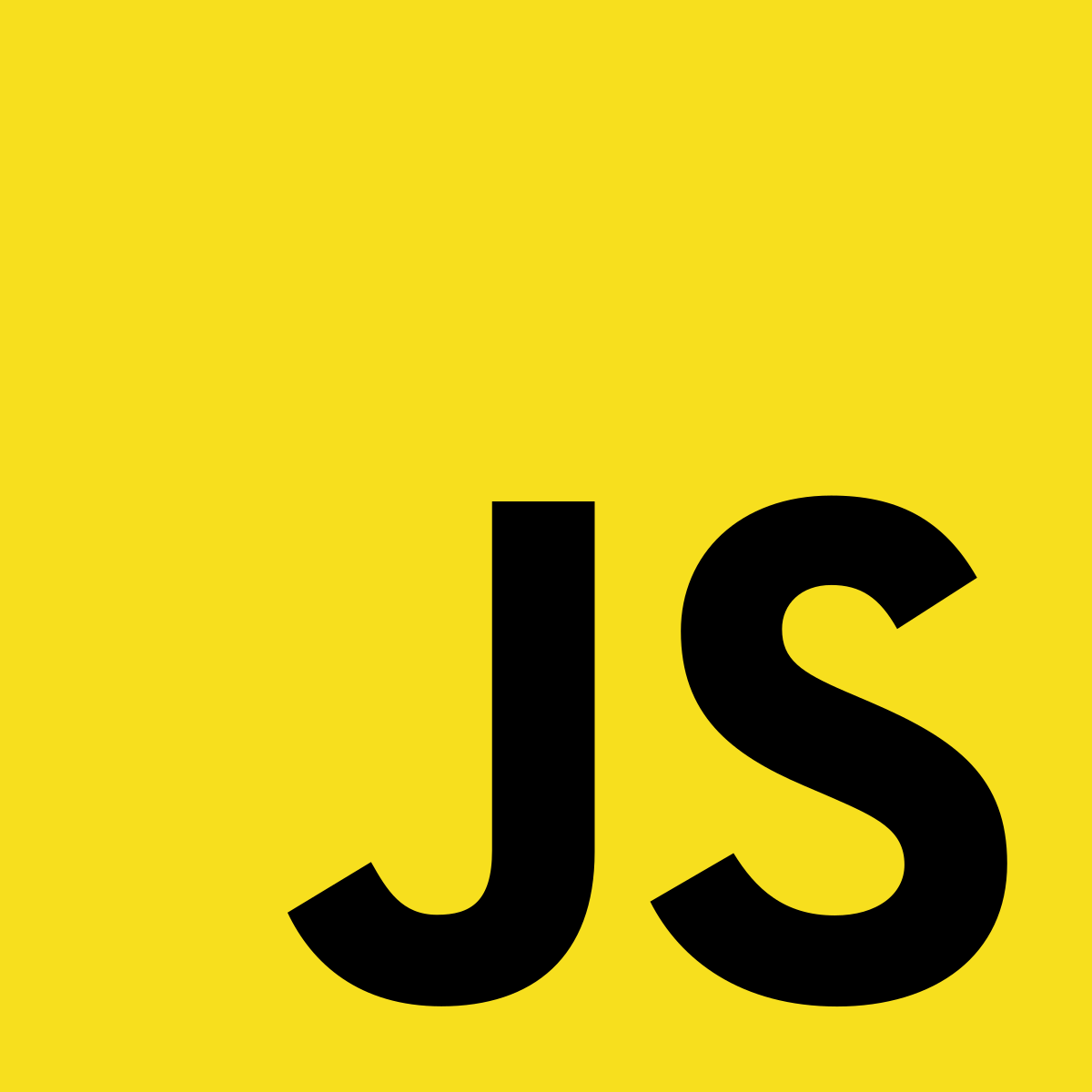
O GitHub é uma plataforma baseada na Web para armazenar e colaborar em projetos de software. Posso carregar o código no GitHub utilizando o *Git* para o armazenar na nuvem e partilhá-lo com outros programadores. Isto ajuda-me a controlar as versões do meu código, a acompanhar as alterações e a sugerir correções de erros nos projetos.



*Figura - 6 Logo do MySQL*

## **MySQL**

O MySQL é um sistema de gestão de bases de dados relacionais que utilizo para armazenar dados nas minhas aplicações Web. Posso utilizá-lo para criar bases de dados, tabelas e executar consultas para armazenar, atualizar e recuperar informações. O MySQL é amplamente utilizado no desenvolvimento Web devido à sua fiabilidade, desempenho e facilidade de utilização.



*Figura - 7 Logo do JavaScript*

## **JavaScript**

JavaScript (JS) é uma linguagem de programação que é utilizada para tornar as páginas Web interativas. Pode ser utilizada para criar animações, efetuar validação de formulários, tratar eventos e muito mais.

# **Fases do projeto**

## **1ª Fase**

* Aprender JavaScript
* ~~Revisar~~ Rever os conceitos básicos da linguagem: sintaxe, variáveis, tipos de dados e operadores. Trabalhar com funções e objetos.
* Compreender o funcionamento dos eventos e o tratamento das entradas do utilizador.

Nesta primeira fase do trabalho, realizei uma pesquisa sobre a linguagem de programação JavaScript. Aprendi os conceitos básicos da linguagem, incluindo a sintaxe e os tipos de dados. Aprendi também os conceitos de funções e objetos em JavaScript.

## **2ª Fase**

* Aprender React.js
* Dominar os conceitos básicos do React: componentes, JSX, estado e propriedades.
* Trabalhar com formulários e eventos em React. Roteamento usando React Router. Configurando o estado no React usando os hooks useState, useEffect.
* Envio de pedidos para o servidor usando Axios.

Nesta fase do trabalho, concentrei-me na aprendizagem da estrutura React.js. Aprendi os conceitos básicos do React, incluindo a criação de componentes, o trabalho com JSX e a gestão de estados. Também aprendi a trabalhar com formulários e a lidar com eventos em React, bem como a gerir o encaminhamento de aplicações utilizando o React Router. Além disso, comecei a enviar pedidos para o servidor utilizando a biblioteca Axios, que me permite interagir com o lado do servidor da aplicação.

## **3ª Fase**

* Aprender Node.js e Express.js
* Dominar os conceitos básicos do Node.js: trabalhar com módulos, sistema de ficheiros.
* Criar um servidor simples usando Express.js.
* Trabalhar com rotas e processar pedidos HTTP.
* Interação com base de dados: MySQL.

Nesta fase do trabalho, aprendi Node.js e a framework Express.js. Aprendi os conceitos básicos do Node.js, incluindo o trabalho com módulos e o sistema de ficheiros. Também aprendi a criar servidores simples utilizando Express.js e a lidar com pedidos HTTP.

## **4ª Fase**

* Desenvolvimento de aplicações Back-End e Front-End
* Criação da estrutura básica do projeto para Back-End e Front-End.
* Integração do Front-End (React) com o Back-End (Express.js).
* Trabalhando com autenticação e autorização.
* Processamento de formulários e interação com bases de dados. Otimizar o desempenho e a segurança da aplicação.

Nesta fase do trabalho, estava a desenvolver uma aplicação completa. Criei a estrutura básica do projeto para o Front-End e o Back-End e depois integrei-os uns com os outros. A autenticação, o processamento de formulários e a interação com a base de dados foram aspectos fundamentais desta fase.

## **5ª Fase:**

* Otimização e aperfeiçoamento
* Otimizar o código e as consultas à base de dados.
* Acrescentar funcionalidades e melhorias adicionais.
* Trabalhar no design e na interface do utilizador.
* Testes e correção de erros.
* Preparação do relatório.

Nesta fase do trabalho, otimizei o código e as consultas à base de dados para melhorar o desempenho da aplicação. Também acrescentei funcionalidades e melhorias adicionais para tornar a aplicação mais funcional e fácil de utilizar. O trabalho de design e de interface do utilizador foi também uma parte importante desta fase, bem como os testes e a correção de erros antes de finalizar o projeto.

## **6ª Fase**

* Conclusão do projeto.
* Implementação da aplicação num servidor real.
* Testes de desempenho e segurança.
* Análise da versão e feedback.
* Conclusão do projeto e avaliação das lições aprendidas.

Na última fase do trabalho, concluí o projeto implementando a aplicação num servidor real e testando o seu desempenho e segurança. Depois disso, analisei a versão e obtive feedback. Concluí o projeto e tirei conclusões da experiência, o que me permitiu crescer como profissional e programador.

# **Calendarização**

# **Representação da base de dados**

## **Mapa de Gantt**

## **DER do projeto de PAP**

# **Modelo Relacional**