



Licenciatura em Engenharia Informática

Ano Letivo 2023/2024

Desenvolvimento de Aplicações Web

Relatório Projeto final

Docentes:

Prof. Noélia Susana Costa Correia

Realizado por:

António Afonso, Nº 71351

Gonçalo Figueiredo, Nº 71353

Tomás Rodrigues, Nº 67382

Este projeto detalha o processo de desenvolvimento e as funcionalidades de um sistema de calendário personalizado, concebido para permitir aos utilizadores registar e gerir as suas atividades e lembretes de forma eficiente. Discutimos também como a integração de funcionalidades de lembrete melhora a experiência do utilizador, facilitando a organização diária e a produtividade.

A aplicação web criada serve para os utilizadores criarem atividades, com título e descrição, após criar atividades, é possível adicionar aos lembretes as atividades à sua escolha, é possível atualizar a informação sobre essa atividade e eliminá-la, caso se adicione atividades aos lembretes aparece um pop-up em que se escolhe o intervalo de tempo em que se vai realizar essa atividade e o dia em que se vai realizar. Após serem adicionadas aos lembretes as atividades podem dar-se como concluídas e são removidas dos lembretes.

Functionality and requirements:

Client-Side:

Funcionalidades:

1. Activities.tsx:

fetchActivities: Carrega uma lista de atividades do servidor ou fonte de dados.

filterActivity: Filtra a lista de atividades com base em critérios específicos.

deleteActivity: Remove uma atividade específica da lista ou do servidor.

fetchActivity: Carrega os detalhes de uma atividade específica.

Activities: O componente principal que organiza e exibe atividades.

2. Activity.tsx:

Activity: O componente principal que exibe detalhes de uma atividade específica.

fetchActivity: Carrega os detalhes de uma atividade específica.

3. CreateActivity.tsx:

Create: Cria uma nova atividade e a adiciona à lista ou envia para o servidor.

4. DeleteDialog.tsx:

DeleteDialog: Exibe um diálogo de confirmação para a exclusão de um item.

handleClose: Fecha o diálogo sem realizar a ação.

5. DeleteDialogReminder.tsx:

DeleteDialogReminder: Exibe um diálogo de confirmação para a exclusão de um lembrete.

handleClose: Fecha o diálogo sem realizar a ação.

6. Footer.tsx:

Footer: Renderiza o rodapé do aplicativo ou página.

7. Message.tsx:

Message: Exibe uma mensagem ou alerta para o usuário.

8. MyReminder.tsx:

deleteReminder: Remove um lembrete específico.

fetchReminder: Carrega detalhes de um lembrete específico.

fetchReminders: Carrega uma lista de lembretes.

Reminder: O componente principal que organiza e exibe lembretes.

9. Navigation.tsx:

Navigation: Renderiza a barra de navegação ou menu do aplicativo.

10. ReminderDialog.tsx:

addReminder: Adiciona um novo lembrete à lista.

ReminderDialog: Exibe um diálogo para a criação ou edição de lembretes.

11. UpdateActivity.tsx:

fetchActivity: Carrega os detalhes de uma atividade específica para atualização.

Update: Atualiza os detalhes de uma atividade específica.

Requerimentos:

React: A biblioteca fundamental para construir os componentes da interface do utilizador, conforme indicado pelas importações de 'react'.

React Router DOM: Para navegação dentro da aplicação, usado em ficheiros como Activities.tsx e Navigation.tsx, indicado por importações de 'react-router-dom'.

Material-UI (MUI): Um popular framework de UI para React, como se vê em várias importações como @mui/material/* e @mui/icons-material/*. Este é usado para componentes de UI como botões, cartões, ícones, barras de aplicação e muito mais.

Componentes Locais: Há referências a outros componentes locais dentro do projeto, como ./DeleteDialog, ./ReminderDialog, ./Navigation e ./Footer. Isso indica uma estrutura modular onde os componentes são reutilizados.

Server:

Funcionalidades:

1. Activities.ts

Define uma classe Worker em TypeScript para gerir atividades utilizando a base de dados NeDB. As funcionalidades incluem listar, adicionar, eliminar e atualizar atividades, com cada operação a retornar uma promessa para tratamento assíncrono.

2. Api.ts

É uma função registerRoutes em TypeScript que define rotas de API para operações de atividades e lembretes usando o framework Express. Permite listar, adicionar, eliminar e

atualizar atividades e lembretes numa aplicação web, interagindo com workers específicos para cada tipo de entidade.

3. Main.ts

Inicializa uma aplicação Express em TypeScript, configurando middlewares para processar JSON e definir políticas CORS, bem como inicializa workers para atividades e lembretes. Regista rotas de API através de uma função importada e inicia um servidor para escutar pedidos HTTP na porta 8080.

4. Reminders.ts

Define uma classe Worker em TypeScript para gerir lembretes utilizando a base de dados NeDB, permitindo listar, adicionar e eliminar lembretes com base em IDs de lembretes ou de atividades, cada operação a retornar uma promessa para tratamento assíncrono.

Requerimentos:

Para configurar e executar eficazmente este código, precisaria de um ambiente de desenvolvimento adequado configurado para Node.js e TypeScript, com as dependências específicas instaladas: Express, NeDB e Path. Além disso, familiaridade com o framework Express e NeDB seria benéfica para compreender e trabalhar com os aspetos de roteamento de API e gestão de dados do código.

Layout

A interface do utilizador, que se sobrepõe à imagem em background, é simples e intuitiva, com botões devidamente identificados para realizar as funcionalidades, bem como uma barra de pesquisa saliente. Os elementos da interface são minimalistas e não entram em conflito com a complexidade do fundo, preservando um bom equilíbrio visual e facilidade de uso.

Project set up

Começámos o projeto definindo claramente os objetivos da nossa aplicação web. Progressivamente, dividimos e executamos as funcionalidades através de métodos adequados e melhores práticas. Adaptamo-nos continuamente aos desafios, escolhendo as melhores tecnologias para atingir nossos objetivos. O processo culminou numa aplicação eficaz que cumpre as expectativas e oferece uma experiência rica aos utilizadores.