



Centro Universitário de Brasília – CEUB
Diretoria de Educação a Distância – DIREAD
Curso Superior de Análise e Desenvolvimento de Sistemas

Danielle Rodrigues Campos RA: 72401626

Sistematização: Engenharia e Projeto de Software

Professor: Romes Heriberto Pires de Araujo

Brasília - DF

Abril / 2025

INTRODUÇÃO

Organizar um casamento é uma das fases mais importantes e especiais na vida de um casal. No entanto, esse momento também pode se tornar bastante estressante por conta da quantidade de tarefas que precisam ser feitas antes do grande dia. São muitos detalhes para lembrar: contratar fornecedores, organizar a recepção, cuidar da lista de convidados, escolher vestido, maquiagem, decoração e muito mais. Com tanta coisa acontecendo ao mesmo tempo, é fácil se perder ou esquecer de algo importante.

Pensando nisso, este trabalho tem como objetivo desenvolver um protótipo funcional chamado “**NossoDia – To Do List de Casamento**”, que tem como objetivo ajudar na organização de todas essas tarefas.

Durante o desenvolvimento, serão aplicados os principais conceitos da Engenharia de Software, passando por etapas como **levantamento de requisitos, modelagem com diagramas UML, implementação com HTML, CSS e JavaScript**, além da **fase de testes**. Todo o processo também será documentado.

A proposta do sistema é unir organização com facilidade de uso, sempre pensando na experiência do usuário. Dessa forma, o casal poderá se concentrar em viver esse momento com leveza, amor e tudo planejado.

DEFINIÇÃO DO PROBLEMA:

Este protótipo atende a casais que estão prestes a realizar um grande sonho: o seu casamento.

É extremamente importante ter uma organização adequada e precisa para a realização de grandes eventos como esse. São muitas contratações, diversos documentos e várias pessoas envolvidas em um dia que precisa acontecer conforme as expectativas dos noivos, para que esse momento seja sempre lembrado com muito amor e alegria.

Existem muitas tarefas a serem realizadas até o grande dia e, devido à quantidade de coisas que precisam ser organizadas, muitos casais acabam se perdendo em meio a tantos compromissos. Organizar o casamento, recepção, lista de convidados, roupas, maquiagem e tantos outros compromissos, não é uma tarefa fácil sem o auxílio de alguma ferramenta própria para isso.

O To Do List “NossoDia” surge como uma opção de organização, eficiência e praticidade para planejar esse momento especial. Ele auxiliará de diversas formas, organizando os compromissos e oferecendo uma visão mais clara e estruturada das tarefas, facilitando o planejamento do grande dia para os dois.

LEVANTAMENTO DE REQUISITOS DO PROJETO:

- Requisitos Funcionais

1. Cadastro de Tarefas:

O sistema permite ao usuário criar novas tarefas que precisam ser realizadas no planejamento do casamento.

Exemplos: *Contratar cerimonialista, reservar o salão de festas, comprar o vestido da noiva, agendar prova de maquiagem, escolher as alianças, entre outros.*

2. Edição de Tarefas:

O usuário poderá editar o conteúdo das tarefas cadastradas, como o nome da tarefa, data ou prioridade.

3. Atualização de Status da Tarefa:

O sistema deve permitir que o usuário altere o status da tarefa para:

- **A Fazer**
- **Urgente**
- **Pendente**
- **Concluída**

4. Definição de Datas para Tarefas:

Cada tarefa poderá conter uma data de vencimento ou data prevista para execução, permitindo o gerenciamento do cronograma.

5. Visualização da Lista de Tarefas:

O sistema apresenta ao usuário a lista completa de tarefas cadastradas, com suas respectivas informações: título, status, data e comentários.

6. Exclusão de Tarefas:

O sistema permitirá ao usuário deletar tarefas da lista, com confirmação para evitar exclusões acidentais.

7. Comentários em Tarefas:

O sistema oferece a funcionalidade de adicionar observações ou comentários em cada tarefa.

Exemplo: *“Falar com a cerimonialista até sexta”, “Confirmar data do ensaio fotográfico”.*

- Requisitos Não Funcionais

1. **Interface Responsiva e Amigável:**

A aplicação adapta-se a diferentes tamanhos de tela (desktop, tablet, smartphone) e garantir uma experiência de uso intuitiva.

2. **Alto Desempenho e Tempo de Resposta Rápido:**

Todas as interações do sistema (criar, editar, deletar, atualizar tarefas) d ocorrem em tempo real, com tempo de resposta inferior a 1 segundo.

3. **Código Limpo e Bem Estruturado (Clean Code):**

O código-fonte deverá seguir boas práticas de desenvolvimento, com indentação adequada, nomes de variáveis semânticos e comentários explicativos.

4. **Design Agradável e Coerente:**

O layout da aplicação deve apresentar cores suaves, tipografia clara e ícones que facilitem a navegação, priorizando a estética e a usabilidade.

MODELAGEM DO PROJETO (UML):

Diagrama Caso de Uso:

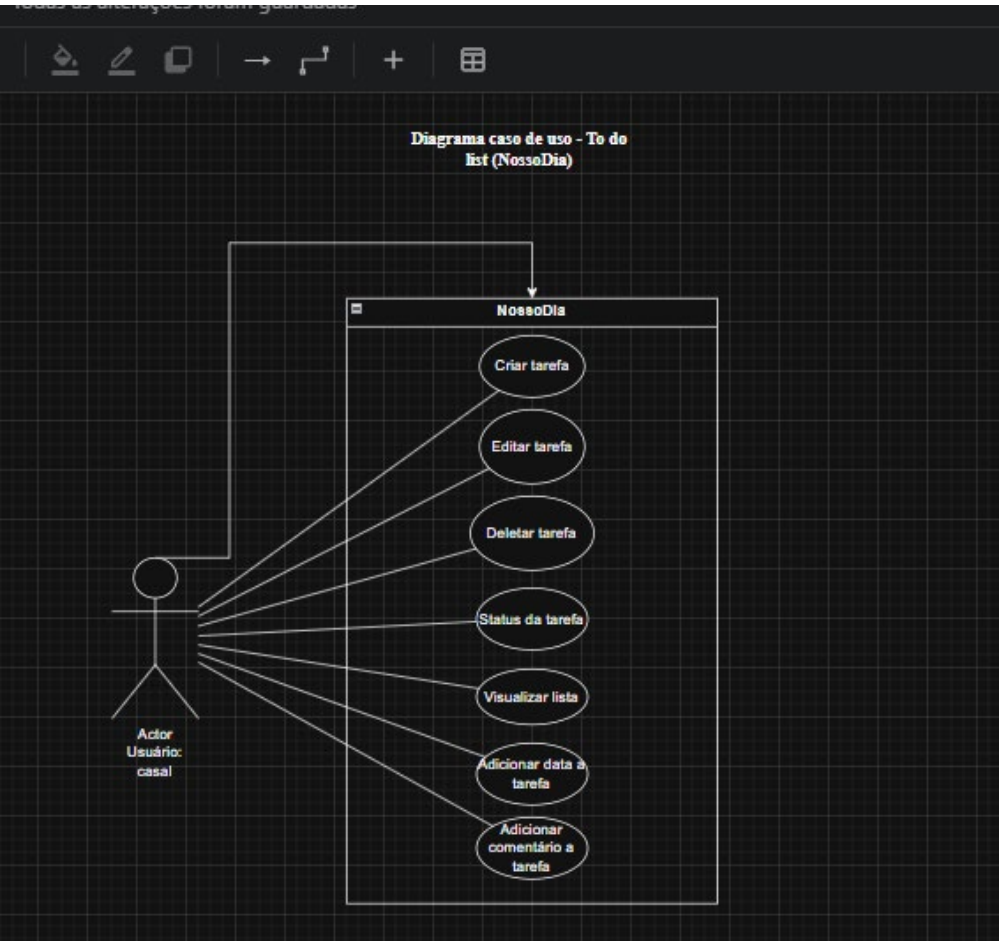


Diagrama de classes:

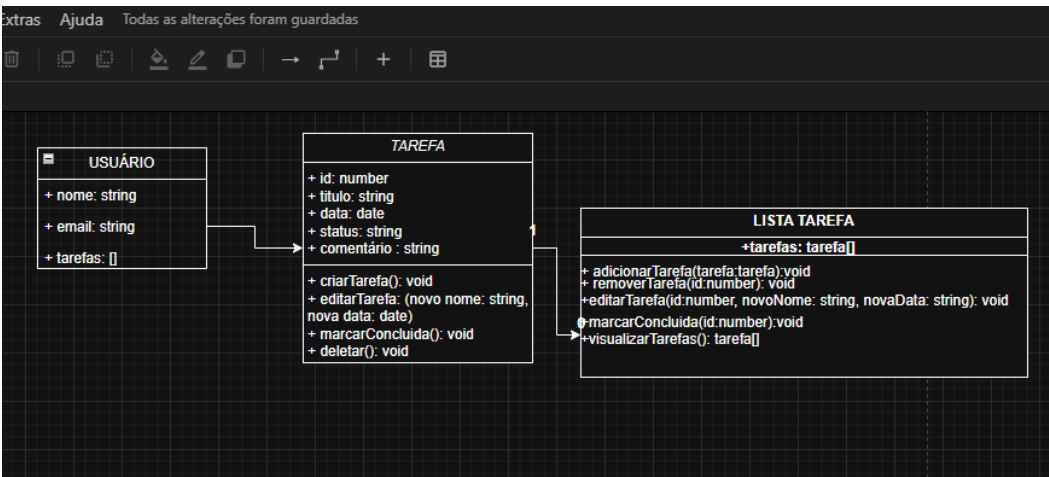
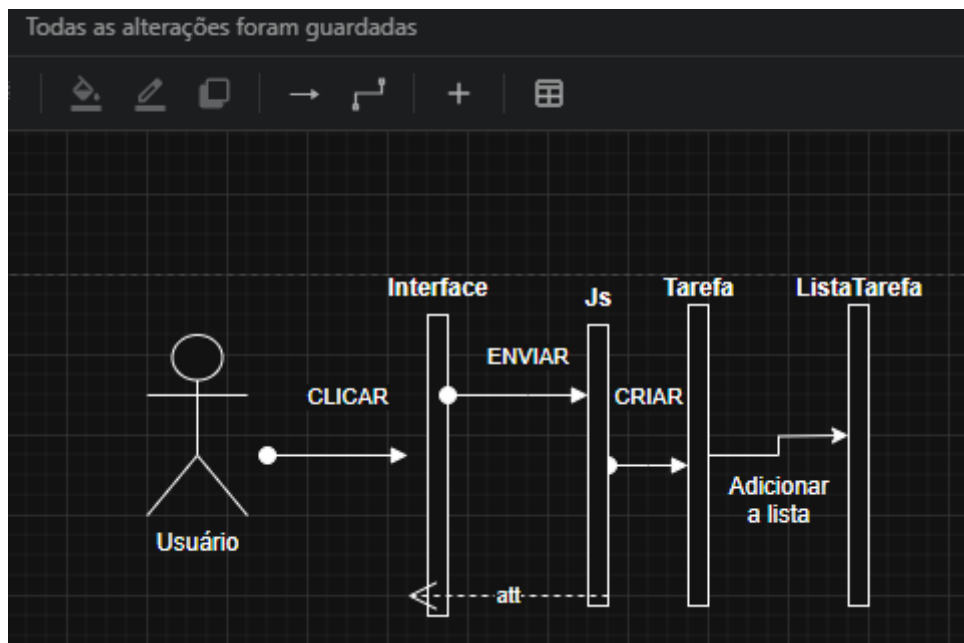


Diagrama de sequência:



Implementação Inicial: Desenvolvendo um protótipo funcional básico:

O protótipo foi desenvolvido com as seguintes tecnologias:

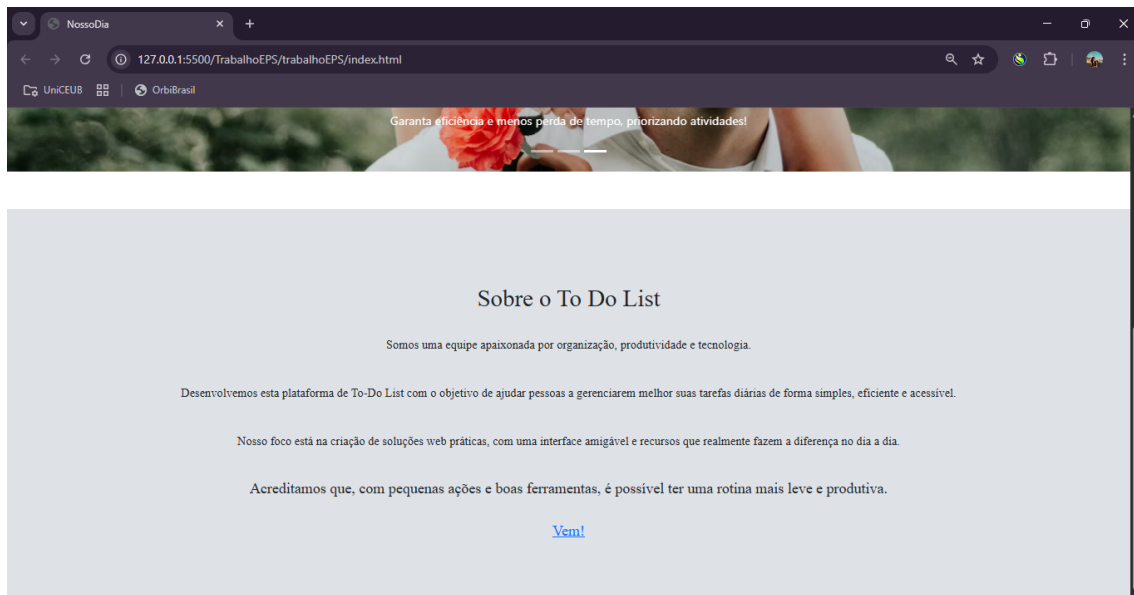
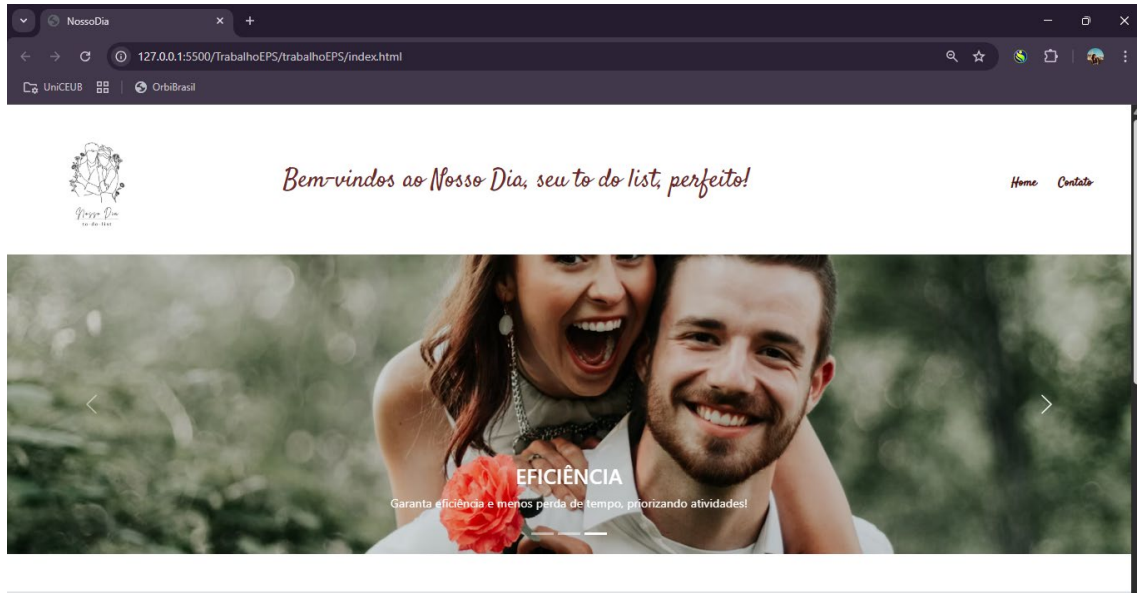
- **HTML:** Responsável por estruturar as páginas.
- **CSS:** Usado para criar o design das páginas.
- **JavaScript:** Lida com toda a lógica da aplicação (CRUD de tarefas usando localStorage).

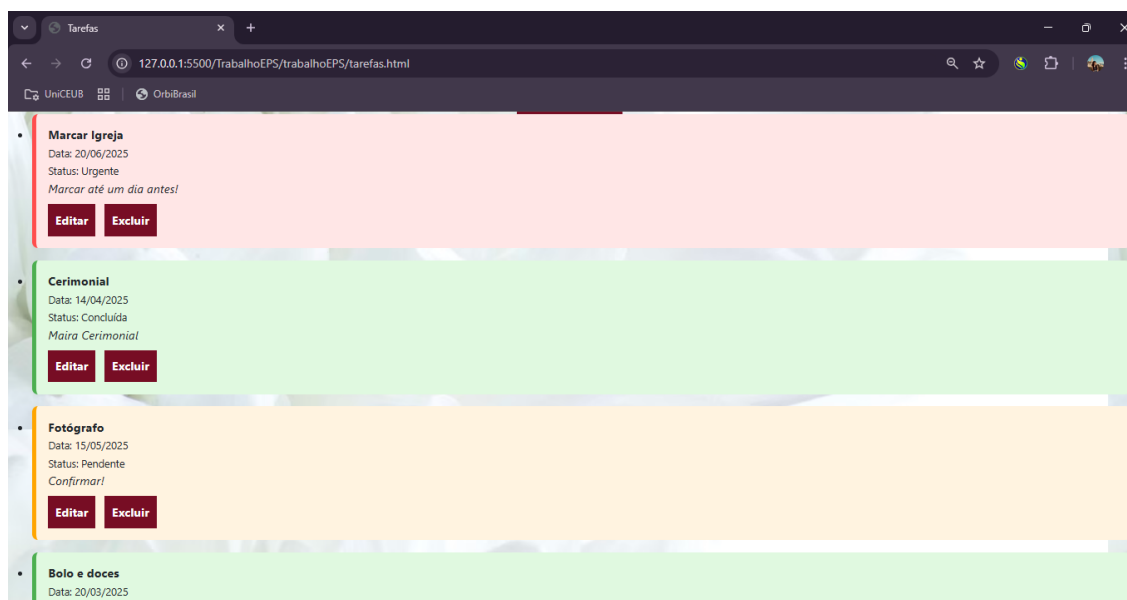
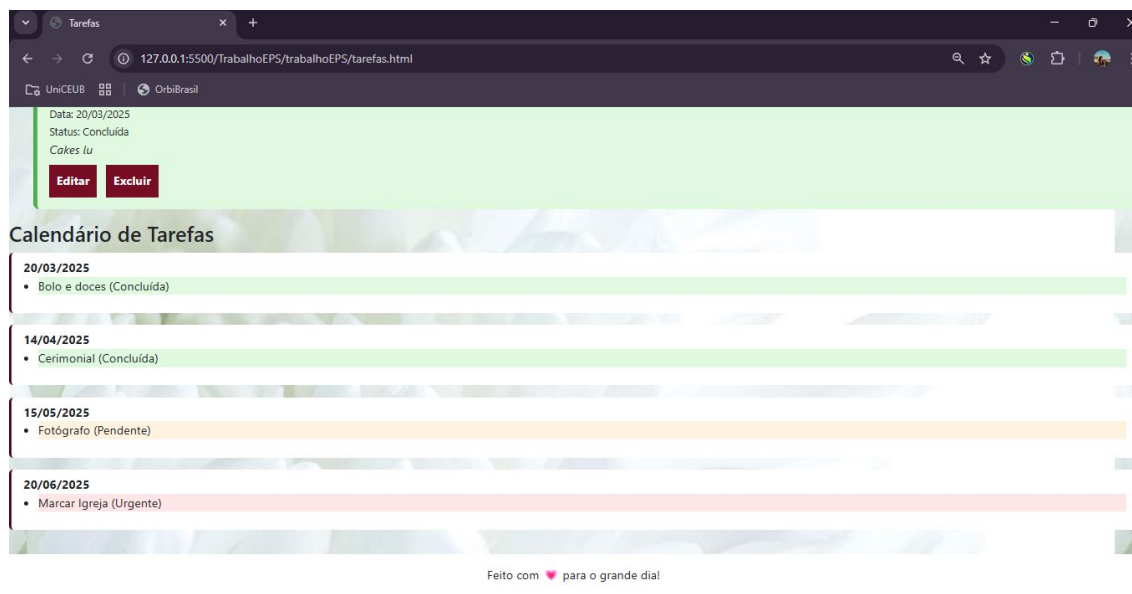
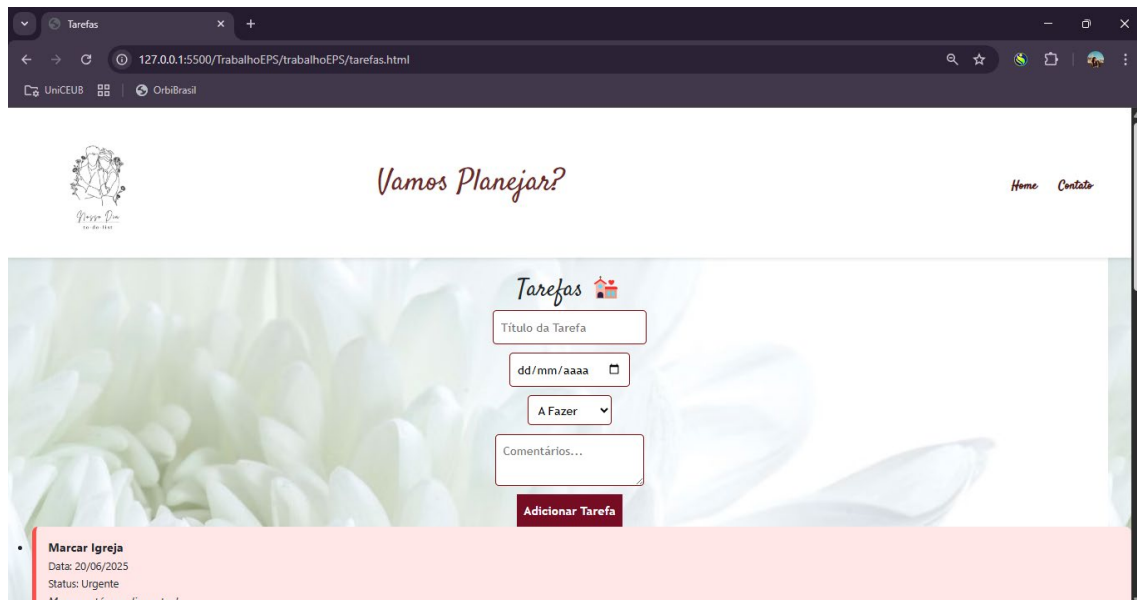
Funcionalidades Implementadas:

- Input para **adicionar tarefas** com título, status, data e comentários.
- Botões de **edição** e **exclusão** das tarefas.
- Checkbox (visual) para diferenciar os status (A Fazer, Urgente, Pendente, Concluída).
- Tarefas são exibidas também por **data no calendário**.

- **Persistência local:** tudo salvo no localStorage, mesmo após atualizar a página.

Imagens do Protótipo:





Link para o projeto: <https://github.com/danicps/Trabalho---EPS>

Fase de Testes:

Teste Funcional

Nesse projeto, realizei verificações manuais com o objetivo de garantir que cada funcionalidade principal estivesse funcionando como esperado. As funcionalidades testadas incluem:

- **Adição de tarefas:** preenchimento do formulário com título, data, status e comentário, seguido do envio e inclusão na lista.
- **Visualização das tarefas:** as tarefas adicionadas aparecem corretamente na lista principal e no calendário.
- **Edição de tarefas:** ao clicar em "Editar", os dados são carregados no formulário e, após nova submissão, atualizados corretamente.
- **Exclusão de tarefas:** ao clicar em "Excluir", a tarefa é removida da lista e do localStorage.
- **Persistência de dados:** ao atualizar ou fechar a página, as tarefas permanecem salvas.

Todos os testes funcionais foram bem-sucedidos.

Teste de Regressão

O teste de regressão foi realizado sempre que alterações no código eram implementadas. O objetivo foi garantir que novas funcionalidades ou ajustes não quebrassem funcionalidades que já estavam funcionando corretamente.

Por exemplo, após a adição da função de edição de tarefas, os seguintes testes foram reexecutados:

- Verificação de adição e exclusão de tarefas.
- Atualização do calendário após mudanças.

- Verificação da consistência dos dados no localStorage.

Nessa fase não foi encontrado nenhum erro.

Teste de Desempenho

Para avaliar o desempenho, realizei a criação manualmente de 20 tarefas simulando diferentes datas e status. A análise considerou os seguintes pontos:

- **Tempo de carregamento da página:** não apresentou lentidão perceptível mesmo com volume elevado de tarefas.
- **Renderização do calendário:** o agrupamento por data e exibição no DOM ocorreu de forma correta.
- **Operações de adição e exclusão:** permaneceram instantâneas independentemente da quantidade de dados.

Concluiu-se que, o sistema apresenta um bom desempenho.

Teste de Segurança

Essa aplicação não utiliza back-end, foram realizadas algumas análises de segurança local para garantir que o usuário não consiga executar código malicioso através dos campos de entrada.

As principais medidas observadas incluem:

- **Armazenamento local:** os dados são salvos apenas no localStorage, sem envio externo.
- **Não execução de HTML nos campos:** os dados são tratados como texto e não como código HTML, evitando a execução acidental de scripts (proteção contra XSS).
- **Simulação de dados inválidos:** a aplicação se manteve estável mesmo com inputs incorretos ou ausentes.

Por se tratar de um projeto local e da faculdade, não foram aplicadas técnicas avançadas de segurança, como criptografia ou autenticação. Para fins acadêmicos e de uso pessoal, os testes foram considerados bons.

Desafios para desenvolver o projeto:

Fazer esse projeto de Engenharia e projeto de software, trouxe muitos desafios. Um dos maiores foi criar o script. No começo, foi bem difícil entender como fazer as tarefas serem salvas, exibidas e editadas corretamente. Tive que pesquisar bastante, testar várias vezes e corrigir muitos erros até tudo funcionar como deveria.

Outra parte que encontrei muitas dificuldades foi fazer os diagramas UML. Eu tive que pensar bastante para transformar o que eu tinha em mente em diagramas que realmente fizessem sentido. Foi a primeira vez que usei UML, nunca tinha feito diagramas, o único que tinha feito até o momento era o BPMN. Foi muito importante pro meu aprendizado, consegui ter mais clareza desse assunto na prática, desenvolvendo os diagramas.

O CSS também não foi muito fácil, eu gosto que tudo fique organizado como eu planejo. Fazer um layout que fosse bonito foi um desafio. Tive que mexer bastante até chegar num resultado que eu achei legal e que fosse fácil de usar.

Mesmo assim, eu sei que o projeto ainda pode melhorar muito. Tem várias ideias que poderiam ser adicionadas, mas como ainda estou aprendendo, foquei no que eu consegui fazer até aqui.

Esse trabalho me ajudou a entender melhor como funciona a engenharia de software, como aplicar ela no desenvolvimento de sistemas e como ela é tão importante para qualquer profissional que deseja fazer um projeto bom desde o início.