

Relatório de Desenvolvimento: Sistema de Gestão de Tarefas

1. Visão Geral do Projeto

O projeto consiste no desenvolvimento de uma aplicação web de gerenciamento de tarefas com autenticação de usuários, onde cada usuário pode criar, visualizar, editar, excluir e filtrar suas tarefas. A aplicação foi desenvolvida utilizando o framework **Laravel** (PHP) para o back-end e **MySQL** como banco de dados. O front-end foi criado com **HTML5**, **CSS3**, **Bootstrap** e **jQuery** para permitir interações dinâmicas com AJAX. O ambiente foi configurado utilizando **Docker**, permitindo fácil replicação e setup do ambiente. O projeto foi versionado utilizando **Git**, e o histórico de commits foi mantido no GitHub.

2. Organização das Tarefas no Backlog e Sprint

O projeto foi dividido em quatro sprints principais, cada um com foco em um conjunto específico de funcionalidades, você pode ver na área de BackLog e Sprint a divisão detalhada.

3. Principais Desafios Encontrados

a) Configuração do ambiente:

O principal desafio inicial foi configurar corretamente o ambiente Docker, especialmente garantindo a comunicação adequada entre o **PHP/Laravel** e o **MySQL**. Configurações de permissões de arquivos, libs e volumes no Docker também exigiram alguns ajustes.

b) Registro, Autenticação e controle de sessão do usuário:

Configurar o sistema de autenticação de usuários exigiu um cuidado extra, especialmente na proteção das rotas e na implementação de middleware para garantir que apenas usuários logados pudessem acessar as rotas de gerenciamento de tarefas. Foi escolhido utilizar as funcionalidades nativas do Laravel para autenticação via sessão, o que facilitou essa etapa.

c) Interações AJAX no front-end:

A integração do CRUD com **AJAX**, permitindo que as ações de criação, edição, mudança de status e exclusão de tarefas fossem feitas de maneira assíncrona, sem recarregar a página foi um desafio pois envolve algumas tecnologias como trabalhar com requisições HTTP e respostas JSON, jquery e ainda precisa ser pensado algumas tratativas para evitar overloads.

4. Processo de Desenvolvimento e Divisão de Tempo

O desenvolvimento foi distribuído ao longo de quatro dias, com cada dia dedicado a uma parte essencial do projeto, como foi especificado na Sprint.