

Projeto MAC0218 - USPMais

Nome dos integrantes

- Daniela Favero, 10277443
 - Felipe Noronha, 10737032
 - Rodrigo Vidotti, 10687896
-

Como executar

1. `git clone https://github.com/mrfelipenoronha/uspmais.git` - Clona o repositório do github
2. `sudo docker-compose build` - Constrói a imagem do projeto atual
3. `sudo docker-compose up` - Subindo a aplicação
4. Acesse `http://localhost:3000`
5. `sudo docker-compose down` - Para a execução

Em alguns momentos o docker-compose up não funciona de primeira, sendo necessário que o container seja finalizado e inicializado novamente para o funcionamento correto.

O que foi feito

Nessa segunda entrega, usamos um [repositório do Github](#) para organizar e controlar o nosso desenvolvimento. Através da criação de *branches* pudemos dividir o workflow entre criação de novas features e correção de bugs. Além disso, usando técnicas de desenvolvimento ágil vistas em aula, conseguimos manter um ótimo ritmo de desenvolvimento, havendo grande harmonia entre planejamento e mão na massa.

Agora, entrando no escopo do que realmente foi feito. Nessa parte construímos todo o esqueleto do projeto e definimos suas principais funcionalidades, através da criação de novas visões, rotas, modelos, e muito mais. Para manter a consistência entre os diferentes ambientes de desenvolvimento, usamos o Docker, uma ferramenta muito robusta e prática que nos possibilitou

um nível incrível de compatibilidade entre as versões feitas por cada integrante do grupo.

Explicando o esqueleto

Nesse esqueleto, criamos:

1. 4 controladores (event, session, static_pages e user);
2. 3 modelos (event, host, user);
3. Diversas views para cada controlador;
4. E muito mais, que não cabem na descrição desse sucinto relatório.

Assim que a página inicial é acessada, é possível, através do menu superior, que o usuário faça diversas interações com o sistema, como login e sign-in. Ademais, caso um usuário esteja *logado*, é possível que ele crie novos eventos, além de poder realizar edições nele mesmo e em seus eventos antigos. Já um usuário que não está *logado* pode visualizar todos os eventos, e, até mesmo, outros usuários. Em nosso sistema, definimos que usuários seriam as organizações criadoras de eventos.

Não menos importante, para se obter mais informações sobre o que foi feito e como foi feito, pode-se consultar o [arquivo CHANGELOG](#) disponível no repositório.

Próximos passos

Para um futuro próximo, iremos tomar os seguintes passos acerca do projeto:

1. Implementar testes automatizados;
2. Deixar o sistema mais orgânico, aumentando a fluidez do usuário;
3. Traduzir erros e mensagens que aparecem.

Principais dificuldades encontradas

A principal, e basicamente única, dificuldade que encontramos foi a de usar ruby/rails. A linguagem não é nem um pouco clara para estudantes dos 3º semestre, já a bibliografia e as próprias notas de aula são muito vagas e inconsistentes. Todavia, com ajuda de pessoas que já tinham cursado a matéria e da maravilhosa rede mundial de computadores, obtivemos sucesso em aprender parte da linguagem, tornando possível a realização do projeto.