

## # Projeto MAC0218 - USPMais

### ## Nome dos integrantes

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

---

## # Como executar em modo de produção

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
6. Para execução dos nossos teste, basta abrir o terminal iterativo docker após construir a imagem do projeto e subir a aplicação, através do comando ``sudo docker exec -ti uspmais_web_1 /bin/bash``. No terminal, digite ``rspec`` para testar os nossos models e controllers, e digite ``cucumber`` que testará as nossas views.

## # Execução como usuário

Para acessar a versão já online da aplicação hospedada pelo heroku, basta acessar ``https://uspmais.herokuapp.com/``

---

> 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 terceira e última entrega, continuamos usando o [repositório do Github] (<https://github.com/mrfelipenoronha/uspmais>) para organizar e controlar o nosso desenvolvimento. Atraves 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 (incluindo reuniões semanais), havendo grande harmonia entre planejamento e mão na massa.

Agora, antes de entrarmos no escopo do que realmente foi feito, é importante ressaltar que na segunda entrega já deixamos a aplicação operacional, portanto, o foco nessa etapa foi otimizar a aplicação e corrigir pequenos erros. No quesito de otimização, trocamos o Debug manual pelo desenvolvimento de testes automatizados com o auxílio do RSpec e o

Cucumber. Usamos a aplicação no heroku. Novamente, 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.

Não menos importante, para se obter mais informações sobre o que foi feito e como foi feito, pode se consultar o [[arquivo CHANGELOG](https://github.com/mrfelipenoronha/uspmais/blob/master/CHANGELOG.md)] (<https://github.com/mrfelipenoronha/uspmais/blob/master/CHANGELOG.md>) disponível no repositório.

## **## Agradecimentos**

Gostaríamos de agradecer primeiramente ao professor e monitor da disciplina, que deram a oportunidade e o apoio necessário para que pudéssemos aprender sobre a construção de uma aplicação a partir do zero, bem como a possibilidade de vivenciarmos a metodologia ágil, o trabalho em equipe e a formação de conhecimento através da prática.

Também é importante evidenciar o papel da universidade que nos disponibiliza esse ambiente de aprendizado contínuo, tal como o próprio ambiente físico como o fornecimento de equipamentos para que pudéssemos trabalhar e nos reunirmos.

Por fim, temos que agradecer aos vários tutoriais, soluções e pessoas dispostas a nos ajudar, tanto conhecidos como na própria internet, como por exemplo o StackOverFlow.

Muito obrigado.

## **## Conclusão**

Com isso, chegamos ao fim desse trabalho, da mesma maneira que esse semestre. Com os desafios proporcionados que foram surgindo ao longo do semestre, conseguimos superá-los assim como nossas expectativas. Fomos capaz de concluir a aplicação completamente funcional, gerar os testes (automatizados) necessários e upar o nosso site para que possa ser acessado, visto, e, um dia, utilizado por outras pessoas.

## **## Próximos passos ?**

Hoje, podemos nos dizer satisfeitos com o nosso trabalho, porém ainda não alcançamos o nosso objetivo. Começamos com a ideia de distribuir a nossa aplicação para solucionar um problema que discutimos e encontramos no nosso dia a dia no instituto, e agora cabe a nós apresentarmos e dar esse retorno ao IME, ou até talvez, a USP inteira.

## **## Contribuição de cada membro do grupo**

Quanto aos papéis do desenvolvimento ágil, o Felipe e a Daniela se ajudaram sendo ScrumMaster, definindo as reuniões para as quintas-feiras e verificando o andamento do processo. Já o Rodrigo ficou como Product Owner, que organizou o nosso Trello (Kanban) deixando claro quem deveria fazer cada tarefa e quando. Esses papéis não foram exclusivos, foi um aprendizado mútuo e proativo da parte de cada um.

Quanto a implementação, como já mencionado, existe o arquivo CHANGELOG que descreve cada alteração feita no decorrer da produção. Mais resumidamente:

- o grupo todo reunido traçou as metas para essa última fase.
- o Felipe cuidou dos testes automatizados com RSpec, de otimizações e correções de erros.
- o Rodrigo fez algumas otimizações e correções de erros
- a Daniela trabalhou com o Cucumber, além de embelezar através do CSS, otimizar e corrigir erros também.

## **## 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, obtemos êxito em aprender parte da linguagem, tornando possível a realização do projeto.