

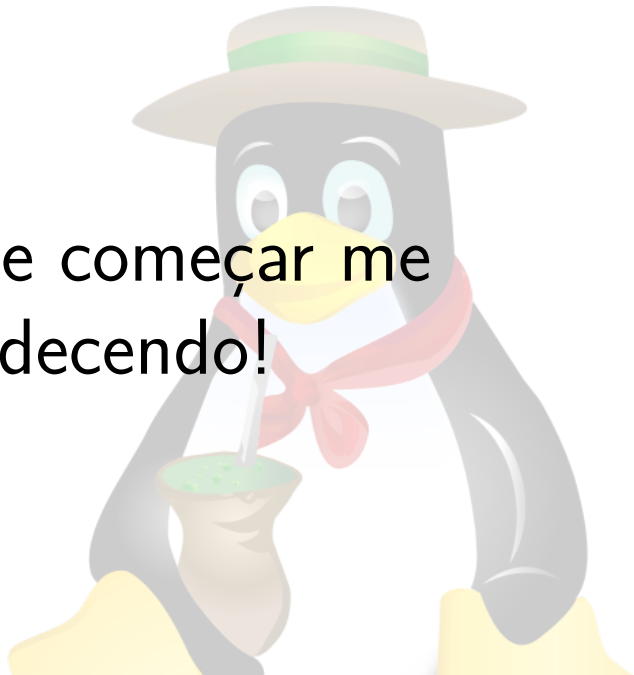


Uma API Resposta

Rafael Guterres Jeffman

2019

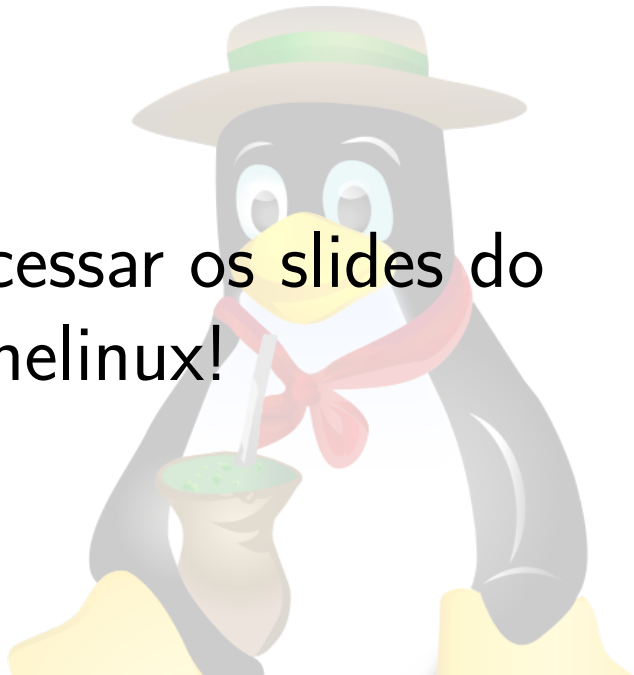
Você pode começar me
agradecendo!



Você tem direito a um certificado
do Tchelinux!



Você pode acessar os slides do
Tchelinux!





O site do evento ficava pronto em
15 minutos!

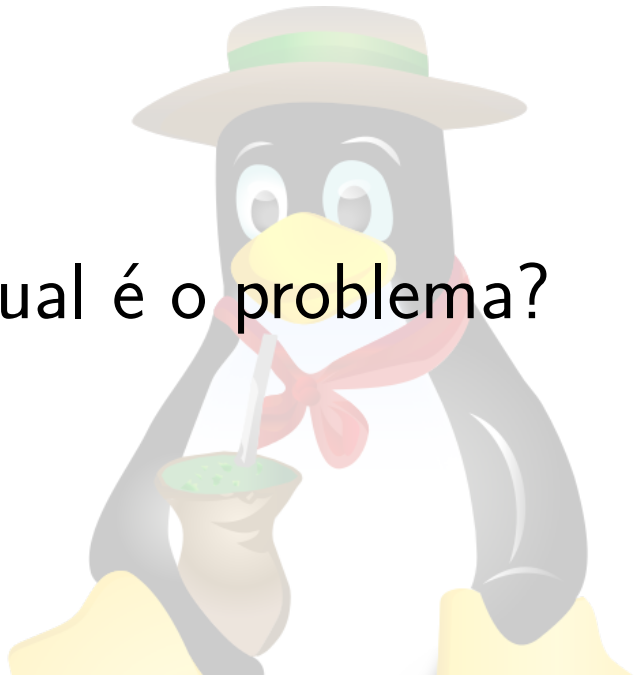
E eu não tenho culpa se desistiram de usar meus scripts.

Então... começamos pelos
agradecimentos!





Mas então, qual é o problema?





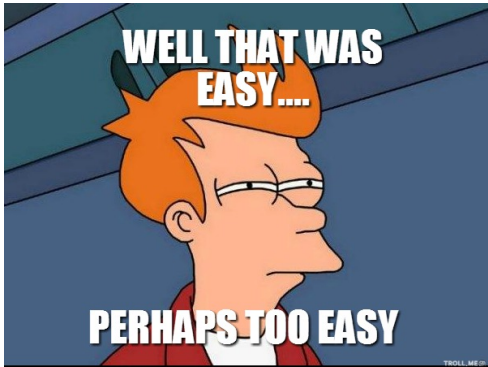
Eu odeio trabalhar.

- E manter diversos **sistemas** só é melhor que fazer tudo a mão.
- E ser o "One Developer To Rule Them All", não escala.
- E ainda tem muita coisa automatizável no Tchelinux...





Eu adoro programar.



- E aprender coisas novas.
- E resolver problemas.
- E poder dizer "ficou mais fácil"!



Limitações do Tchelinux

- Somos todos voluntários
- O mais caro é o tempo.
- Dinheiro é importante.
- Sim, já testamos vários sistemas.





O estado atual



- Sistemas independentes *hackeados* nas horas de folga durante 2-3 dias.
- *Flat file data (JSON)*.
- Certificados com geração quase manual.
- Adição dos slides totalmente manual.
- Quase nada de estatísticas.
- Nada para os usuários.



Por que uma API?

- Escala
- Granularidade
- Integração de todos os sistemas
- Independência da UI
- Livrai-nos de todo PHP...



Design





Desafios Técnicos



- Sem investimento.
- Sem manutenção.
- Sem tempo de desenvolvimento.
- Quatro usuários diferentes.
- Uso em *bursts*, normalmente zero usuários.



Decisões inteligentes

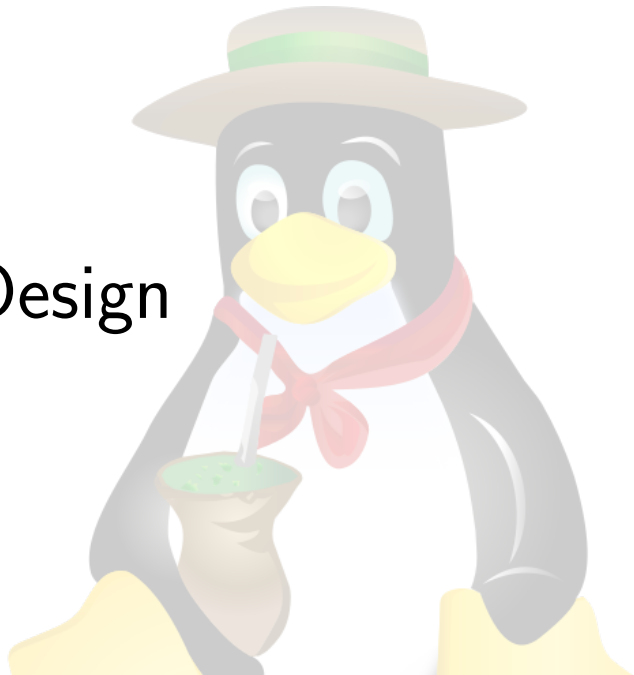
- Automatizar o que for possível.
- 'Not built here' é uma vantagem.
- Participação de outros desenvolvedores.



"A necessidade é a mãe de todas as invenções."

Platão

Design





Serviços

- Gerenciamento de Eventos (dados, site, custos, spam)
- *Call for Papers*
- Inscrições
- Credenciamento
- Geração de Certificados
- Slides das Palestras



Usuários

ERROR
NO KEYBOARD FOUND
PRESS F1 TO CONTINUE

- Coordenação do Tchelinux
- Organizadores de Eventos
- Palestrantes
- Participantes



User Stories com Gherkin

- É fácil.
- Documenta o que se deseja atingir.
- Descreve o comportamento esperado do ponto de vista do usuário.
- É texto com *syntax highlight* e variáveis.



BDD com Behave

- Garante que o comportamento definido está funcionando.
- Exige definição no comportamento do usuário.
- Provê um exemplo de uso da API.
- Rápido para testar o caminho feliz.

Implementação





Decisões Autocráticas

- Python 3
- Flask
- SQLAlchemy e Alembic
- Flake8 e Pylint
- Pytest
- Não dá, mas eu queria o SQLite...





Por que Python 3?

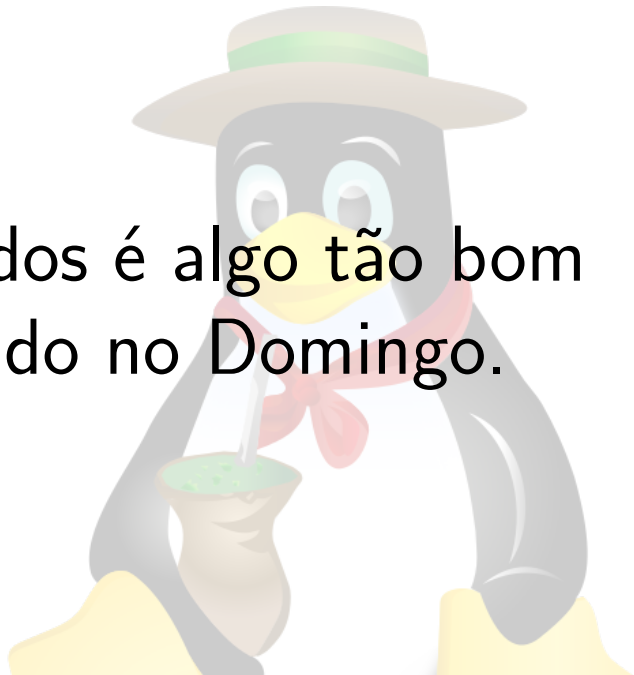
- Porque tem bastante gente pra desenvolver.
- Porque está longe de ser ruim.
- Porque tem tudo que precisamos pronto e com qualidade.
- Porque ninguém deveria estar utilizando Python 2 há anos...



Flask

- Framework extremamente leve para a criação de aplicações WSGI.
- Diversos módulos que facilitam o trabalho:
 - Flask-RESTful
 - Flask-JWT (JSON Web Token)
- Muita coisa disponível
 - Securing REST API: JWT Authentication using Python Flask
 - 2FA - Two Factor Authentication with Flask

Bancos de Dados é algo tão bom
quanto feriado no Domingo.





Data Management

- SQLAlchemy
 - ORM/Framework para persistência em bancos de dados.
 - Ah... os objetos, os lindos objetos!
- Alembic
 - Migração de Dados
 - Implantação, atualização contínua.

Qualidade





"Todo teste de software detecta todos os problemas que foram excitados pelo teste e mais nenhum outro."

Eu mesmo.



Controle de Qualidade

- Apesar das limitações, não testar é muito pior.
- Pytest
 - Testes funcionais e unitários.
 - Ainda não foi tomada uma decisão final.
- Coverage



Qualidade do Código

- PEP8:
 - Flake8: Pode ser bem chato.
 - Pylint: Pode ser impossível de passar.
- Code Review



Code Review

- Realizado nos Pull Requests do Github.
- Utiliza as ferramentas do Github.
- O que verificar:
 - Consistência no código.
 - Simplicidade e facilidade de leitura.
 - Possíveis problemas de manutenção futura.

Deployment





Integração Contínua

- Travis-CI
 - Integra maravilhosamente bem com o Github.
 - Fácilimo de utilizar.
 - Gratuito para projetos *open source*.
 - Gratuito, mas não é aberto.



Integração Contínua

- Jenkins
 - Integra bem com o Github.
 - Requer o uso de um servidor e de configuração trabalhosa.
 - *Open source.*



Integração Contínua

- Hoje, rodamos todos os testes no desenvolvimento e os mesmos testes em CI.
- Com o aumento da demanda, a ideia é executar apenas um subconjunto dos testes em desenvolvimento.
- Por simplicidade, e custo, hoje, utilizamos o Travis.
- No futuro, um pouco distante, é prevista uma migração para o Jenkins.



Disponibilidade da API

- O problema dos custos do Tchelinix
 - Falta braço
 - Falta tempo
 - Falta dinheiro
- Apesar de querermos usar dados estáticos ao máximo, a parte de gerenciamento dos eventos precisa ser dinâmica.

E ainda tem mais...

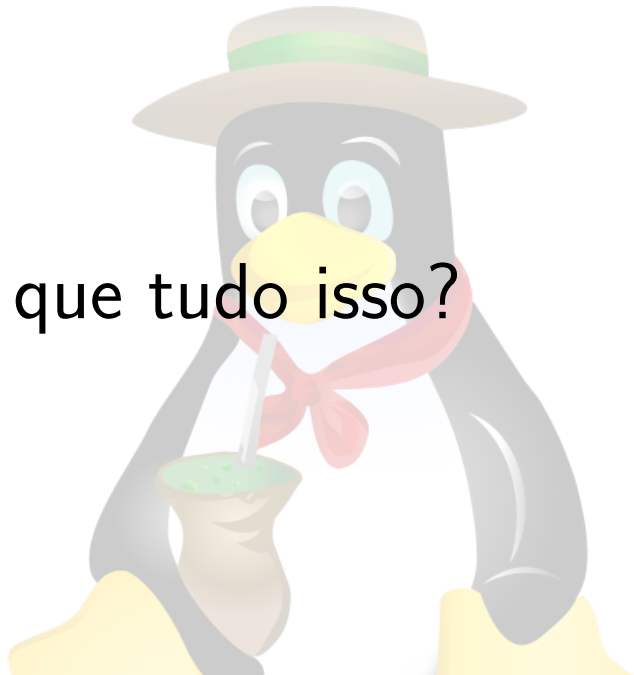




Gerenciamento do Projeto

- Github:
 - Issues
 - Projects
 - Milestones

Mas e por que tudo isso?





Lei Geral de Proteção de Dados Pessoais

- A partir de Agosto de 2020 podem ser aplicadas sanções pela a LGPD
- Não vai nos mandar pra cadeia, mas...
- ... pode nos deixar na m... mais difícil situação.
- Até o nome de vocês a gente precisa proteger!



*"As imagens dessa apresentação foram levemente
surrupiadadas da Internê."*

Os autores das imagens, por favor, me desculpem...



Muito Obrigado!

- `mailto:rafasgj@gmail.com`
- Slides: `https://slides.tchelixux.org`
- Source Code:
`https://github.com/rafasgj/tchelixux-code`
- Certificados:
`https://certificados.tchelixux.org`

