

Modernizando projetos legados usando APIs

Elton Minetto

<http://eltonminetto.net>
@eminetto



Tweet

**Sam Newman**

@samnewman



Don't forget - having legacy software is often a sign of success. Your business was successful to last long enough for your software to become legacy.

12/11/17 06:09

Case Winker

CONHEÇA A WINKER

A MAIS COMPLETA PLATAFORMA PARA INTEGRAR,
GERENCIAR E OTIMIZAR CONDOMÍNIOS.

Quer saber mais? Entraremos em contato com você.

Experimente

Acesse **sua conta** se você já é cliente



Características

- Desenvolvido com Yii 1.x e Lumen
- ~5 anos de desenvolvimento
 - ~160k linhas de código
 - ~70% de cobertura de testes
 - PHP 5.X
- Estrutura usando containers e AWS

Problemas

- Crescimento da empresa gera novos desafios de velocidade de entrega
 - Framework antigo e difícil de manter

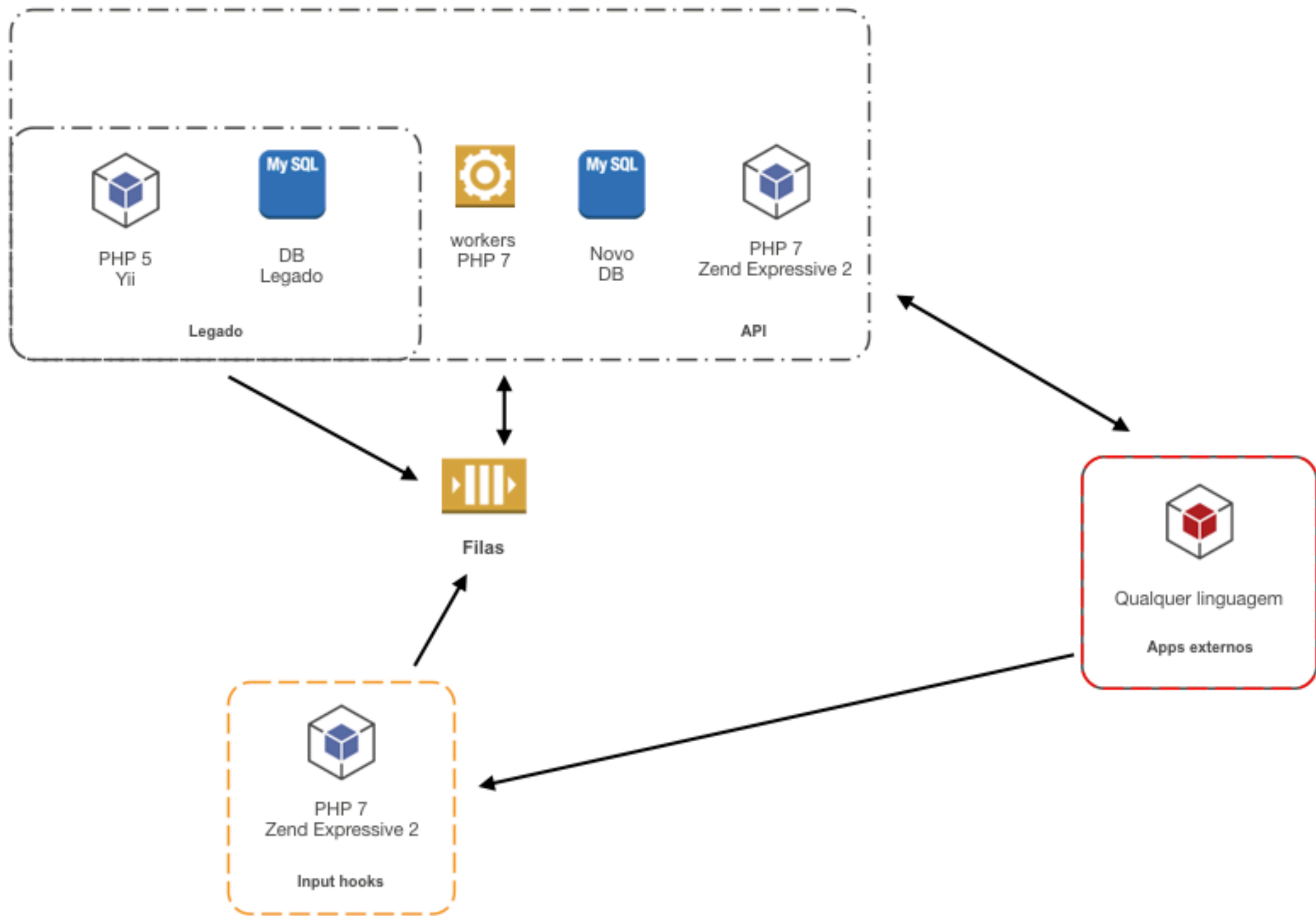
Requisitos do projeto

- Modernizar a stack
- Lançar uma API pública para aumentar o alcance do negócio
 - Lançar um portal de desenvolvedores
 - Transformar o projeto em uma plataforma

Arquitectura propuesta

"[...]developers have used **virtualized machine images** or **container images** with Docker containers to create immutable images of legacy systems and their configurations. The intent is to contain the **legacy in a box** for developers to run locally and remove the need for rebuilding, reconfiguring or sharing environments"

<https://www.thoughtworks.com/pt/radar/techniques>



Ferramentas usadas

- Zend Expressive 2
 - Docker
- AWS Elastic Beanstalk
 - AWS SQS
- Bernard - abstração de filas
- Fractal - transformação de dados
- Monolog + Sentry - centralização de logs

Ferramentas usadas

- Doctrine - ORM
- API Blueprint
- aglio - documentação da API
 - dredd - testes de API
 - drakov - mock de API
- phpunit - testes unitários/integração
 - prooph - event bus

Estágio Atual do projeto

- Integração entre o legado e a nova API implementada
- Lançado o primeiro aplicativo externo conectado à nova arquitetura
 - Legado rodando com PHP 7

Lições aprendidas

- Testes unitários e de integração são cruciais
- Containers para isolar as versões tornaram o processo mais fácil
- Conflitos entre Amazon Elastic Beanstalk + Docker Compose foram mais complexos do que o esperado
 - Middlewares (Zend Expressive 2) aceleram o desenvolvimento

Lições aprendidas

- Começar pela definição da API foi uma boa decisão
 - API Blueprint foi uma boa decisão
 - Bernard foi uma boa decisão
- Trabalhar com filas/processamento paralelo é complexo

Próximos passos

- Lançamento do Portal de Desenvolvedores
- Desenvolvimento do componente *Input hooks*

Perguntas

Contato

<http://eltonminetto.net>

<http://coderochr.com>

<http://codenation.com.br>

<http://asemanaphp.com.br>