



“Sistema de gerenciamento de módulos”

Por

Victor de Azevedo Nunes

Trabalho de Graduação



Universidade Federal da Bahia
ceapgmat@ufba.br
wiki.dcc.ufba.br/PMCC/

SALVADOR, Abril/2017



Universidade Federal da Bahia

Departamento de Ciência da Computação

Programa Multiinstitucional de Pós-graduação em Ciência da Computação

Victor de Azevedo Nunes

“Sistema de gerenciamento de módulos”

Trabalho apresentado ao Programa de Programa Multiinstitucional de Pós-graduação em Ciência da Computação do Departamento de Ciência da Computação da Universidade Federal da Bahia como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: *Ivan do Carmo Machado*

SALVADOR, Abril/2017

Eu dedico esta dissertação...

Agradecimentos

Meus agradecimentos...

*Walking on water and developing software from a specification are
easy if both are frozen*

—EDWARD V BERARD

Resumo

Meu resumo

Palavras-chave: palavras chave

Abstract

My abstract...

Keywords: key words...

Sumário

Lista de Figuras	xvii
Lista de Tabelas	xix
Lista de Acrônimos	xxi
1 Introdução	1
1.1 Motivação	1
1.2 Problema	1
1.3 Objetivos	1
1.4 Metodologia	2
1.5 Resultados Esperados	3
1.6 Fora de Escopo	3
1.7 Estrutura do Trabalho	5
2 Referencial Teórico	7
2.1 Software como serviço	7
2.2 Aplicações web	8
2.3	8
Apêndice	10

Lista de Figuras

1.1	Ranking das linguagens de programação no Stack Overflow e Github . .	3
1.2	Infográfico da WebhostFace, exibindo a popularidade dos Frameworks PHP em 2015	4
1.3	Gráfico do Google Trends exibindo as pesquisas por ferramentas front-end	5

Lista de Tabelas

Lista de Acrônimos

ALM Application Lifecycle Management

1

Introdução

1.1 Motivação

Otimizar o desenvolvimento de um software é uma tarefa árdua e bastante estudada. A área de Engenharia de software estuda maneiras de melhorar o desenvolvimento de software através de bons processos e práticas de desenvolvimento. Qualquer melhoria durante a construção de um sistema, normalmente acaba por prover uma economia no orçamento de desenvolvimento e/ou manutenção. Assim, é possível inferir que melhorias no processo de desenvolvimento de sistemas, pode agradar desde programador, com um trabalho facilitado, até o cliente final com redução de custo.

1.2 Problema

A repetição de código durante o desenvolvimento de um software, acaba por aumentar a sua complexidade e tempo de desenvolvimento. O impacto dessa má prática atinge fortemente a manutenibilidade do software, gerando grande transtorno diante de uma alteração que poderia ser simples caso estivesse bem projetado desde o começo. Assim, o reuso de código sempre é buscado num projeto. A abordagem do reuso pode inclusive, abranger mais de um projeto numa organização, fazendo que até mesmo módulos inteiros sejam reaproveitados.

1.3 Objetivos

Seguindo a boa prática de sempre projetar código que atenda ao maior número de casos que for possível, este trabalho tem como objetivo reutilizar o módulo de controle

de acesso de um sistema, já que o mesmo está presente sem desvios significativos nos de uma organização. Assim, proposta é desenvolver tal módulo como um serviço, e deixando-o separado de qualquer sistema e gerando a possibilidade de qualquer sistema consumir tal módulo como um serviço. Dessa forma, ao projetar um novo software para uma organização por exemplo, ao invés de incluir o módulo de controle de acesso, projetaria-se o código para consumir o serviço de controle de acesso que seria um projeto à parte. Tal situação também tem a vantagem de ter interoperabilidade entre linguagens de programação, já que realizaria a comunicação através de JSON, seguindo padrões atuais. Assim, teria-se uma constância no módulo de acesso independentemente da possível necessidade de migração de tecnologia.

1.4 Metodologia

Baseando-se nas tecnologias gratuitas em alta no cenário atual do desenvolvimento web, dispomos de algumas opções eficientes para a implementação da solução. Dentre as possibilidades, considerando a facilidade para futura manutenção e continuidade do projeto, tende-se a optar por uma tecnologia popular. Como linguagem de programação, adota-se o PHP. A escolha é fundamentada de acordo com a pesquisa da RedMonk de 2015 [1], que evidencia o uso das linguagens de programação de acordo com as discussões no StackOverflow e repositórios no GitHub. É possível constatar a popularidade do PHP no cenário atual com o gráfico da pesquisa citada, na qual o PHP é apresentado na terceira colocação, apenas atrás do líder JavaScript e do segundo colocado, o Java.

Entretanto, não seria inteligente desenvolver um sistema completo sem o auxílio de um framework. Dentre os frameworks disponíveis para PHP, hoje o destaque está com o Laravel, que se encontra no topo dentre os mais utilizados no momento.

A WebHostFace, uma empresa de hospedagem, compilou várias estatísticas para criar um infográfico mostrando os frameworks PHP mais populares de 2015. Utilizando informações sobre os próprios clientes, o Google Trends, estatísticas de repositórios do GitHub e a pesquisa do SitePoint “Best PHP Frameworks 2015”, a WebHostFace elaborou o seguinte infográfico:

Como pode ser verificado no gráfico [2], tem-se a evidência que o Laravel em 2015 teve a maior popularidade em projetos pessoais e tem a maior comunidade entre os concorrentes, o que o torna uma boa escolha para a escrita de um software que será continuado por terceiros.

Para elaborar os recursos de interface e integrar ao back-end PHP do sistema, será

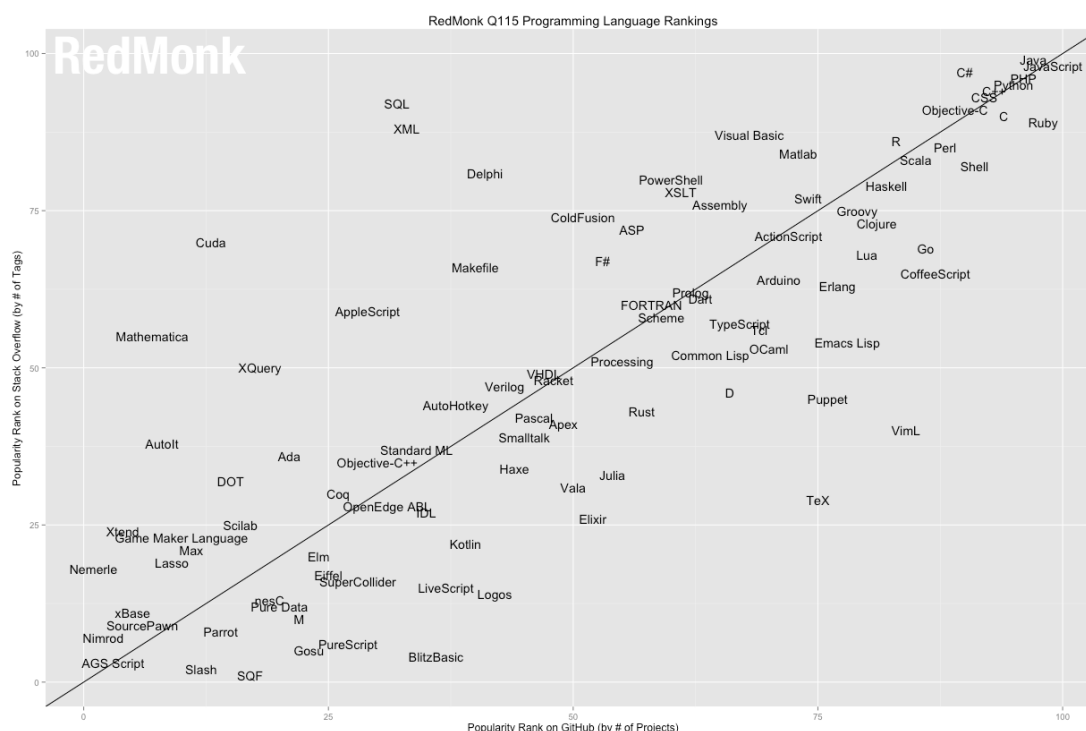


Figura 1.1 Ranking das linguagens de programação no Stack Overflow e Github

adotado o já conhecido AngularJS, ferramenta sólida e conhecida no aspecto em questão.

Dados coletados via Google Trends em [3](#), que propõe comparações entre termos pesquisados, revela a popularidade do AngularJs diante de alguns dos principais concorrentes. O gráfico abaixo evidencia o cenário.

1.5 Resultados Esperados

Realizando o uso do explanado módulo de controle de acesso como serviço, espera-se que exista ganho real no desenvolvimento dos próximos softwares. Tal ganho pode ser representado em tempo, orçamento, complexidade de código ou outra visão de acordo com diferentes abordagens.

1.6 Fora de Escopo

Diante da possibilidade de gerência dos sistemas, módulos e funcionalidades contratadas, seria possível estender o sistema aqui proposto, complementando-o com uma abordagem financeira, controlando o acesso do cliente mediante o pagamento do que

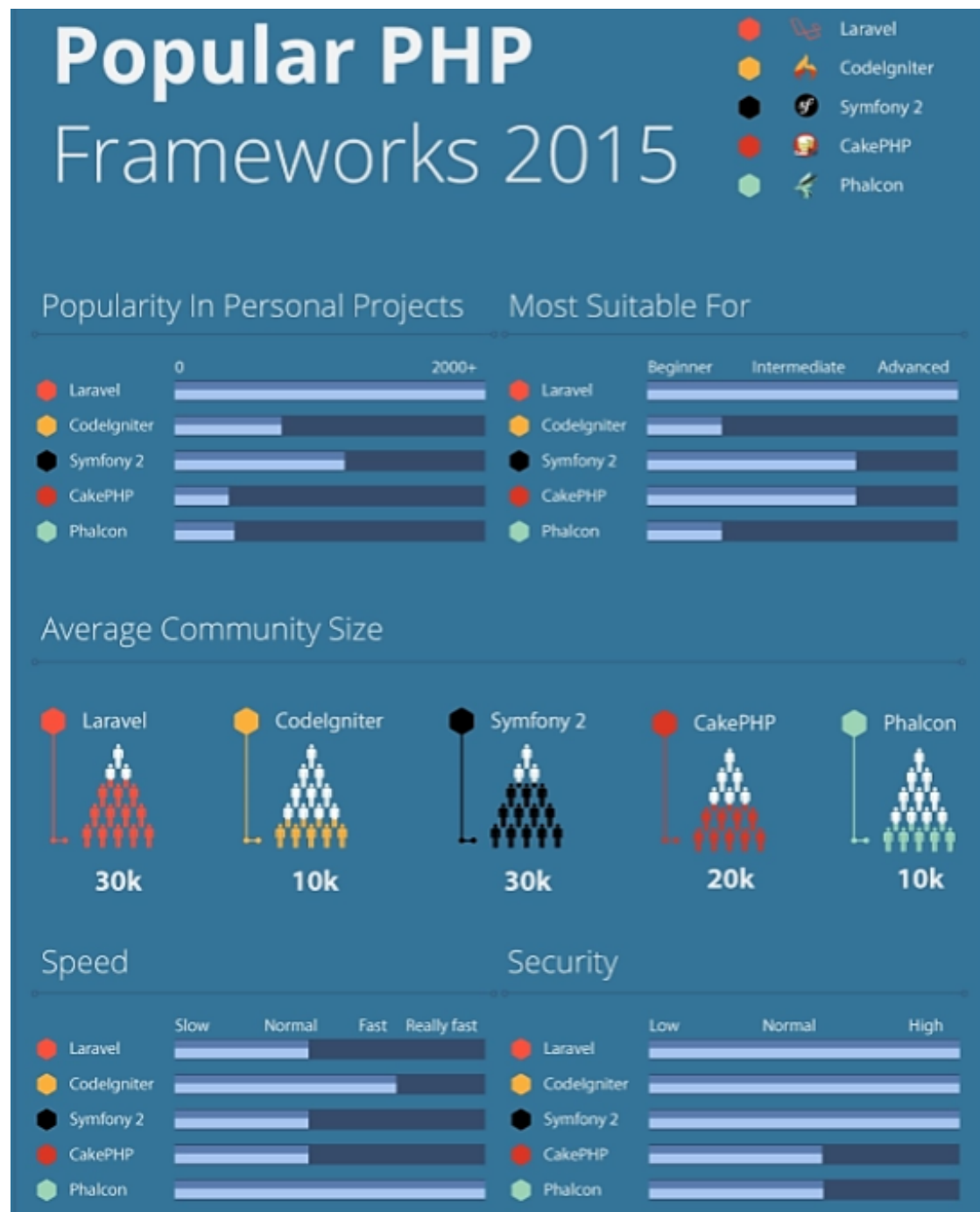


Figura 1.2 Infográfico da WebhostFace, exibindo a popularidade dos Frameworks PHP em 2015

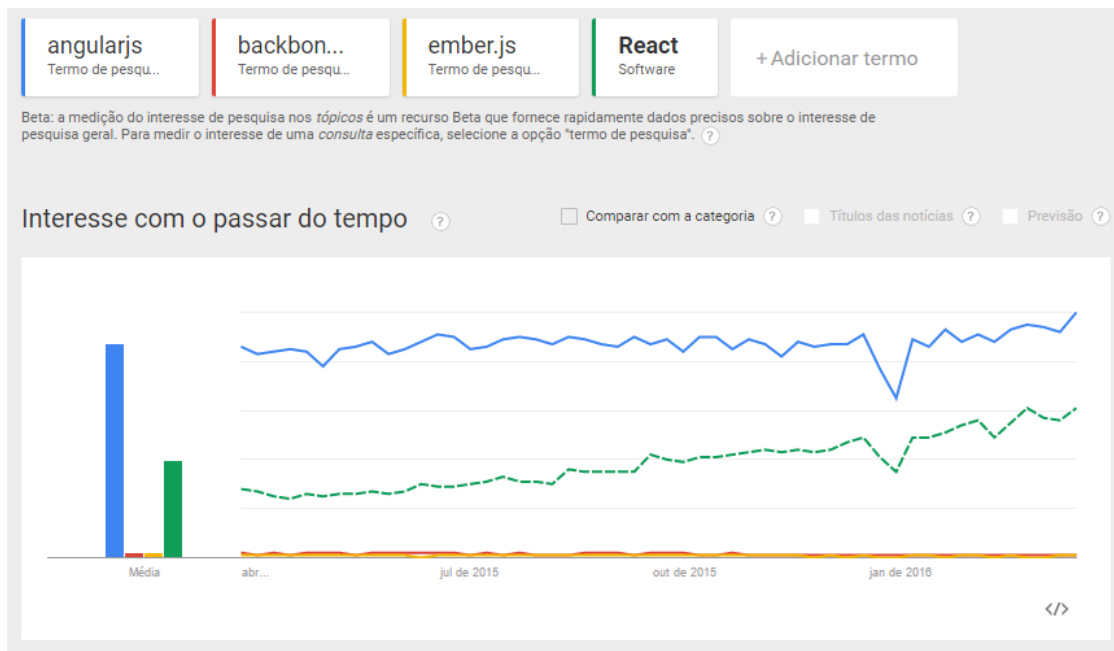


Figura 1.3 Gráfico do Google Trends exibindo as pesquisas por ferramentas front-end

está contratado pelo cliente. Também existe a possibilidade de realizar o controle por número de usuários que estão logados no sistema contratado ou por módulo, mediante uma evolução na implementação. Uma possível solução para a última situação seria o sistema consumidor enviar uma notificação(post) ao sistema de gerenciamento de módulo num determinado intervalo de tempo, fazendo a sinalização de que o usuário está logado no sistema. O sistema de gerenciamento de módulos responderia internamente à sinalização, mantendo registro de usuário ativo. Caso o usuário final saia do sistema, enviaria o sinal de logout e caso fechasse o navegador, a sessão seria encerrada por tempo de inatividade (falta de envio da notificação de uso).

1.7 Estrutura do Trabalho

Na seção 2, é apresentado o referencial teórico, com um embasamento sobre os assuntos a serem tratados neste trabalho.

2

Referencial Teórico

2.1 Software como serviço

Um trabalho recente realizado por Lechessa et al. [4] apresenta uma pesquisa qualitativa sobre os fatores determinantes para adoção ou não de um SaaS voltado para ERP na África do Sul. Esses autores indicam que os principais fatores determinantes para adoção desse mecanismo de software são sua fluidez quanto à rede e a segurança. Esses fatores estão presentes na aplicação desenvolvidas neste trabalho de conclusão de curso.

O artigo de [5] orienta como o avanço da computação em nuvem não é apenas um problema do ponto de vista da segurança, pois muitos trabalhos na literatura apenas a área apenas como mais um ponto de vulnerabilidade. Os autores de [5] realizaram estudos exploratórios junto a empresas usuárias de serviços em computação em nuvem e consideram que a perspectiva de SaaS também pode fortalecer a segurança nas aplicações de cloud computing, pois o software de autenticação compartilhado por várias aplicações em nuvem, oferece uma melhor padronização e consequente facilidade de prevenção a erros de vulnerabilidade específicas de cada módulo da pesquisa. Esse ponto de vista é muito importante para qualquer trabalho de ponta na área de SaaS.

A definição de SaaS encontra-se muito bem elaborada em um dos trabalhos listados na literatura. Segundo La e Chun [6], o princípio da definição de Software como um Serviço (Software as a Service - SaaS) é um serviço complementar para aplicações da computação em nuvem (cloud computing). As duas áreas estão interligadas, no entanto, não se confundem, pois o SaaS deve ser entendido como um mecanismo de suporte às soluções existentes na cloud. Os SaaS existem justamente para maximizar o reuso de serviços repetidos e não centrais em uma aplicação remota.

2.2 Aplicações web

Para atender à fomentada demanda de aplicativos web, é necessário adotar métodos de desenvolvimentos que sejam ágeis, eficientes e de fácil manutenção. [7] Propõe o uso do modelo MVC (Model, View e Controller) no atual desenvolvimento para softwares web. O modelo apresentado tornou-se um padrão popular e divide o software em camadas com propósito definido, tornando-o de mais fácil manutenção.]

2.3

Referências Bibliográficas

- [1] RedMonk, “Gráfico de linguanges - redmonk.” <http://goo.gl/vDSGLx>, 2016.
- [2] W. Face, “Popular frameworks php 2015.” <https://goo.gl/DjUREE>, 2016.
- [3] Google, “Googletrends - frameworks front-ent mais buscados em 2015.” <https://goo.gl/VUhIrv>, 2016.
- [4] M. Lechesa, L. F. Seymour, and J. Schuler, “Erp software as service (saas): Factors affecting adoption in south africa.,” in *CONFENIS* (C. Møller and S. S. Chaudhry, eds.), vol. 105 of *Lecture Notes in Business Information Processing*, pp. 152–167, Springer, 2011.
- [5] R. Rai, G. Sahoo, and S. Mehfuz, “Securing software as a service model of cloud computing: Issues and solutions.,” *CoRR*, vol. abs/1309.2426, 2013.
- [6] H. La and S. Kim, “A Systematic Process for Developing High Quality SaaS Cloud Services,” in *Cloud Computing* (M. Jaatun, G. Zhao, and C. Rong, eds.), vol. 5931 of *Lecture Notes in Computer Science*, ch. 25, pp. 278–289, Berlin, Heidelberg: Springer Berlin / Heidelberg, 2009.
- [7] Y. Ping, K. Kontogiannis, and T. C. Lau, “Transforming legacy web applications to the mvc architecture,” in *Software Technology and Engineering Practice, 2003. Eleventh Annual International Workshop on*, pp. 133–142, Sept 2003.

Apêndice

