



BACK-END

Desenvolvimento de Software para Internet



FRAMEWORKS PHP

O que é um framework? Definições



- Conjuntos de objetos reutilizáveis que reúnem recursos genéricos que podem ser utilizados em aplicações específicas e com características semelhantes. Essas arquiteturas genéricas (compostas por classes, objetos, componentes, etc.) podem ser estendidas para criar sistemas mais específicos (Viljamaa, 2001; SCHMIDT; GOKHALE; NATARAJAN, 2004)
- Coleção de classes abstratas, objetos e padrões dedicados a resolver determinados problemas, em uma arquitetura flexível e extensível ([Dicionário do Programador](#))



Analogia com coleção de livros ...



VALIDAÇÃO
DE FORMU-
LÁRIO

MANIPULA-
ÇÃO DE
STRINGS

CRIAÇÃO
DE QUERY
NO B.D.

UPLOAD DE
ARQUIVOS
NO PHP

Dentro dos livros estão todos os atributos e métodos (funções) para realizar as tarefas propostas

**FRAMEWORK: PERMITE DESENVOLVER SOFTWARE
DE FORMA MAIS RÁPIDA E SIMPLES**

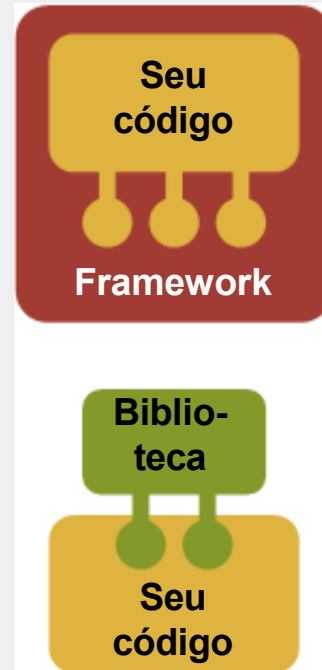
Ícones: Freepik

Diferença entre framework e biblioteca



Biblioteca

- Trabalham de maneira independente ao ambiente que será implementado



Crédito: tomasp.net

Framework

- Contém várias bibliotecas;
- Norteiam todo o desenvolvimento da aplicação



jQuery versus Angular

Crédito: [Stack Overflow](#)



70



Basicamente, jQuery é uma biblioteca Javascript, que *simplifica* o Javascript para atividades comuns no dia-a-dia do desenvolvimento. Excelente para manipular o [DOM](#) com muito menos código que o JS puro.

Exemplo de jQuery:

Selecionando o `<div id="meuidiv"></div>`



Javascript: `document.getElementById('meuidiv')`



jQuery: `$('#meuidiv')`



Já AngularJS é um [framework](#), que trabalha com dados, focando na interação usuário <-> aplicativo. Tem os seguintes pontos principais:

- **Data Binding de mão dupla.** [Exemplo.](#)

Você define um tipo de dados (chamado Model) e quaisquer alterações nestes ocorrem em toda aplicação, onde os mesmos aparecerem.

- **Padrão de desenvolvimento** [\[MVVM\]](#) (semelhante ao [MVC](#)).

- **Template engine integrada**

Ex.: Você recebe de um servidor um objeto [JSON](#) com vários itens e os atribui à um *Model* chamado `Pessoas` no Angular. Esse é o nosso objeto que atrelamos ao model através de um controller do Angular:

```
personas = [  
  {nome: 'Romulo', sobrenome: 'Zoch', caracteristica: 'Lindo'},  
  {nome: 'João', sobrenome: 'Do caminhão', caracteristica: 'feio'}  
]
```

Para listar todos os itens do objeto/model, você pode, por exemplo, fazer o seguinte:

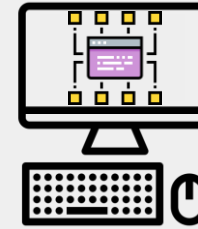
```
<li ng-repeat="pessoa in pessoas" ng-model="pessoas">  
  {{pessoa.nome}} é {{pessoa.caracteristica}} e seu sobrenome é  
  {{pessoa.sobrenome}}  
</li>
```



“From Scratch” versus Framework



- Programador escreve todo código do zero
- Complexidade crescente
- Maior tempo e esforço de desenvolvimento



- Programador começa a partir de uma estrutura existente
- Desenvolvimento simplificado, com reutilização de código pronto
- Menos tempo e esforço



Vantagens de usar um Framework



- **Redução de custos:** funcionalidades implementadas e amplamente testadas permitem ao desenvolvedor focar nas regras do projeto
- **Aumento na produtividade:** reuso, e padrão de desenvolvimento, resultam na facilidade na comunicação da equipe
- **Facilidade de manutenção:** como segue um padrão de projeto, é mais fácil localizar os pontos em que devem ocorrer alterações
- **Mais segurança:** a maioria dos frameworks aplicam regras de prevenção (validações de segurança) previamente testadas



Desvantagens de usar um Framework



- **Alta curva de aprendizagem:** tempo gasto para aprender como funciona e começar a usar o framework, pois são difíceis de explorar
- **Escolha do framework errado:** escolha de uma solução muito complexa para um sistema simples, pode tornar o desenvolvimento complexo
- **Segurança:** se um problema for explorado, ele será amplamente difundido
- **Eles moldam a maneira como você codifica:** uma vez que toda estrutura do código é controlada e definida previamente

É fundamental conhecer bem a linguagem usada pelo framework!



Resumo dos principais frameworks PHP



Característica	Laravel	Symfony	CodeIgniter	Zend Framework (Laminas)
Lançamento	2011	2005	2006	2006
Arquitetura	MVC	MVC	MVC	MVC
Requisitos de Sistema	PHP 7.3+	PHP 7.2+	PHP 5.3.7+	PHP 7.2+
Comunidade Ativa	Sim	Sim	Sim	Sim
Documentação	Extensa e atualizada	Extensa e detalhada	Boa	Boa
Facilidade de Aprendizado	Moderada	Moderada	Fácil	Moderada
Recursos Incluídos	Autenticação, ORM, Filas	Componentes diversos	Segurança, Filtragem	Componentes diversos
Desempenho	Bom	Muito bom	Bom	Bom
Comunidade e Suporte	Ativa	Ativa	Ativa	Ativa
Segurança	Bom	Muito bom	Bom	Bom
Empresas que o Utilizam	BBC, Governo Brasileiro, Toyota Hall of Fame	Spotify, Lufthansa, Leroy Merlin	Harvard University, Bitbucket	Cisco, AirBnb, Magento (primeira versão)
Atualização e Manutenção	Regular	Regular	Eventual	Regular



DÚVIDAS?



Créditos:

Profª: Carolina Sacramento
Carolina.sacramento@fiocruz.br

