

# **Framework para Desenvolvimento em front-ends HTML**

## **Semantic UI**

O Semantic UI é um framework de desenvolvimento rápido que te ajuda a criar layouts responsivos utilizando o HTML de uma forma simples e fácil.

### **As principais características**

- Validador de formulário;
- Busca auto sugestiva a partir de json;
- Preloadings que sobrepõe conteúdos;
- Div's sticky, onde a div acompanha o scroll automaticamente (isso é muito bom);
- Menus (horizontal, vertical, responsivo, mobile, oculto, iconificados, etc);
- Modal (com sub-modal, com opção de confirmação, com background black);
- Tooltips fantásticos;
- Inúmeras transições;
- Customização em less.

### **Seus principais pontos fortes e fracos**

#### **Prós:**

- Todo o pacote Semantic-UI é bem organizado, com todos os componentes organizados com sua própria folha de estilo e arquivo JavaScript. Dessa forma, você pode carregar apenas os componentes necessários para cada página, minimizando o tempo de carregamento e o tamanho do arquivo.
- Os elementos oferecem uma enorme quantidade de personalização, muito além de um framework como o bootstrap.
- Usa nomes de classes semânticas para o seu estilo, facilitando a compreensão até mesmo para iniciantes
- A interface semântica tem um design futurista e bonito. Muitos ficarão satisfeitos com o design, especialmente quando a Semantic UI é usada como uma estrutura CSS.
- Ele inclui vários componentes de interface do usuário que você pode precisar para quase qualquer tipo de projeto em que você possa trabalhar; E, claro, você sempre pode escolher apenas o que precisa.
- A documentação é fácil de usar, bem escrita e tem muitos exemplos, cada um com seus códigos-fonte.
- Existem várias implementações oficiais da UI semântica para muitas bibliotecas populares, frameworks e CMS. Tais como [Angular](#) ou [Wordpress](#) .
- Fácil de usar

## **Contras:**

- Os pacotes são muito maiores quando comparados ao Twitter Bootstrap ou Zurb Foundation. A UI semântica é realmente muito grande e seria melhor usar módulos e componentes específicos, em vez da coisa toda.
- Muitos recursos da UI semântica usam a personalização de Javascript, como no Modal. Isso é diferente do Bootstrap, que pode adicionar o Modal apenas com a personalização dos atributos HTML. Os desenvolvedores que planejam usar a Semantic UI devem estar familiarizados com o Javascript ou com o JQuery para obter o máximo disso.
- Contém muitas imprecisões da interface do usuário, como posicionamento incorreto, não é possível misturar classes etc.
- Uma vez que você quer fazer algo que não é mencionado no documento - prepare-se para passar uma hora, depois desista e implemente uma "solução alternativa" personalizada.

## **Quem desenvolveu e quem mantém o software**

Jack Lukic é o criador da UI semântica. Além disso, Jack Lukic teve 1 trabalho anterior como co-fundador da Myfav.es.

Semantic UI é uma estrutura de interface de usuário de código aberto que é um dos 30 principais projetos de código aberto mais populares de todos os tempos no GitHub.

## **Quando o software foi lançado**

Foi criado por Jack Lukic e lançado em 2012

## **Históricos de duas versões**

### **Versão 0.1.0 - 25 de setembro de 2013**

#### **Novo**

- -Adicionado novo conjunto de ícones de fontes usando a porta Font Awesome
- -Adicionado ícone antigo definido como um conjunto de ícones básicos separados
- Variação de entrada de fluido adicionada
- -Aumento do tamanho dos rótulos de canto
- -Adiciona variação de grade relaxada com calhas aumentadas

## Erros

- -Adiciona variações de item de menu acolchoadas horizontalmente e verticalmente para permitir que itens de menu removam o preenchimento
- -Adicionado correções para o menu tabular especialmente com conteúdo anexado
- Carregadores -UI agora posicionados automaticamente em todas as circunstâncias, mesmo com texto longo

## Atualizações

- Documentação atualizada para a barra lateral, transição e validação de formulários

Esta é uma das primeiras versões e ainda está disponível para download no site do github:

<https://github.com/Semantic-Org/Semantic-UI/releases/tag/0.1.0>

## Versão 2.4.0 - 17 de setembro de 2018

### Novos componentes

- Placeholder - Adicionado ui placeholder que pode ser usado para mostrar onde o conteúdo aparecerá em breve.
- Novo tipo de interface do usuário
- Segmento - Adicionado novo ui placeholder segment de espaço reservado de interface do usuário usado para reservar espaço para interface do usuário quando o conteúdo está ausente ou vazio.

### Principais aprimoramentos

- Dropdown - Adicionadas dropdowns clearable . Quando clearable: true é especificado, um (X) aparecerá para limpar a seleção suspensa # 2072
- Modal / Dimmer - Modals e dimmers agora incluem uma nova configuração, useFlex cujo padrão é auto . Os modais e dimmers reverterão automaticamente para o uso de layouts não flexíveis quando houver problemas de layout com o uso do flexbox. Os modais retornarão à posição JS quando detachable: false é usado ou com IE11 / Edge (os elementos posicionados totalmente dentro de contêineres flexíveis no IE se comportam de maneira diferente).

## Erros Críticos

- Modal - Corrigido problema em que a scrolling modal não permitia a rolagem com dispositivos de toque. # 6449
- Label - Corrigido problema onde basic label estava aparecendo incorretamente Obrigado @lasley / @ColinFrick # 6582 # 6440
- Menu / Dropdown - left menu fixo dentro do ui menu usuário seria exibido horizontalmente como flex # 6359

## Erros

- Dimmer - O Dimmer agora define a variation em tempo de execução, para suportar a troca de tempo de execução entre top aligned e middle aligned usando .dimmer('setting', 'variation', 'top aligned')
- Dropdown - Corrigido problema onde onChange quando usado com action: hide estaria faltando o terceiro param \$item # 6555
- Flag - Add uk alias para o united kingdom Obrigado @PhilipGarnero # 6531
- Ícone - Correções faltando disk outline icon alias # 6556
- Lista - Corrigido o problema em que o content lista não ocuparia 100% de largura quando usado junto com o img ou o icon
- Menu / menu suspenso - Correções na margem do item suspenso não obedecidas dentro do labeled icon menu # 6557
- Modal - Correção da variável de tema @mobileTopAlignedMargin não foi implementada
- Modal - Modal agora removerá a blurring após o desnivelamento, para evitar problemas com a position: fixed # 6520

## Pequenas alterações

- Dropdown - inline dropdown - inline dropdown close icon padrão margem direita padrão espaçamento ligeiramente modificado.

## Enderenço do site

<https://semantic-ui.com/>

## Link para Download

<https://github.com/Semantic-Org/Semantic-UI/releases/tag/2.4.1>

## Instalação

### Instalando o NodeJS

Se você não está familiarizado com a configuração do NodeJS você pode seguir os passos abaixo

### instalar o executável

Baixar o NodeJS executável.

## Instalar Gulp

Semantic UI utiliza Gulp para fornecer ferramentas de linha de comando na construção de versões temáticas da biblioteca com apenas os componentes que você precisa.

Gulp é um módulo NPM que precisa ser instalado globalmente

```
npm install -g gulp
```

## Instalando o Semantic UI

Semantic UI está disponível em um pacote de mesmo nome no NPM

```
npm install semantic-ui --save  
cd semantic/  
gulp build
```

## Inclua dentro do seu HTML

```
<link rel="stylesheet" type="text/css" href="semantic/dist/semantic.min.css">  
<script  
  src="https://code.jquery.com/jquery-3.1.1.min.js"  
  integrity="sha256-hVVnYaiADRTO2PzUGmuLJr8BLUSjGIZsDYGmIJLv2b8=" crossorigin="anonymous"></script>  
<script src="semantic/dist/semantic.min.js"></script>
```

## Atualizando via NPM

O script de instalação do Semantic vai automaticamente atualizar o Semantic UI para a última versão enquanto preserva seu site os temas empacotados.

```
npm update
```

## Próximos passos

Tudo pronto!

Muito bem! O Semantic UI agora está pronto para ser usado.

## Como usar?

### Para compilar todos os arquivos:

```
gulp build
```

### Para ver as alterações

```
gulp watch
```

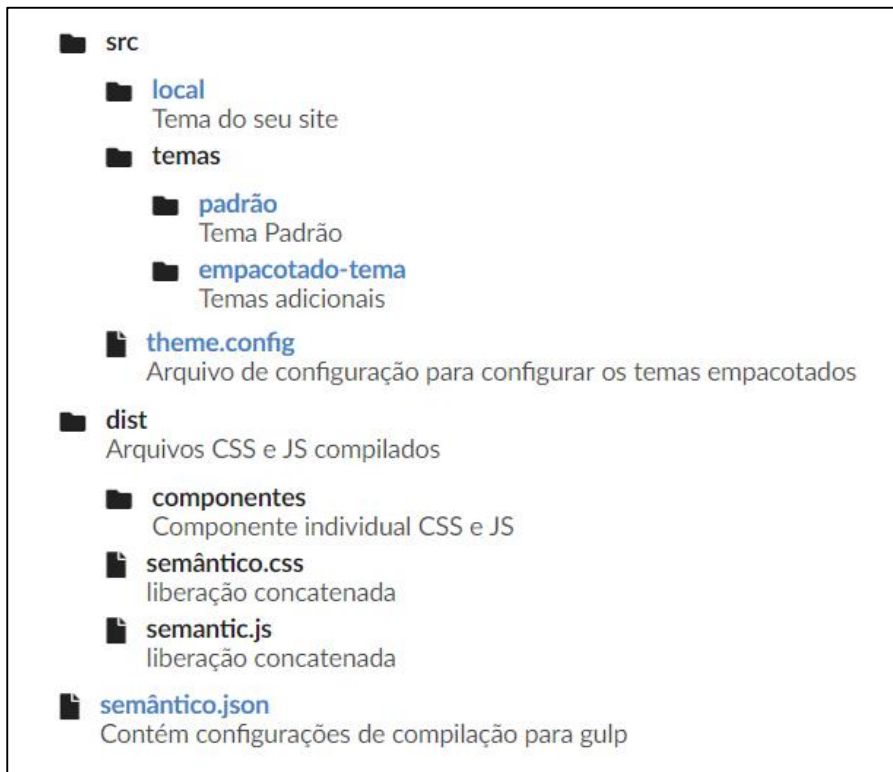
## Por que construir ferramentas?

A interface semântica usa o Gulp por vários motivos:

- Processar arquivos LESS com variáveis temáticas
- Adicionar prefixos de fornecedores para navegadores suportados com autoprefixer
- Assista a alterações de arquivos e reconstrua os arquivos necessários ao ajustar os temas
- Processar caminhos de ativos automaticamente, para que versões minificadas e concatenadas possam existir em diretórios diferentes
- Permitir construções personalizadas com apenas os componentes necessários para seu site
- Permitir uso avançado, como idiomas da direita para a esquerda (RTL) usando RTLCS

## Estrutura de pastas

Um projeto de interface do usuário semântica inclui a seguinte estrutura de pastas



## Comandos Gulp

Abaixo está uma lista de todos os comandos gulp disponíveis

Comando	Descrição
instalar	Pede instalar perguntas para configurar o semantic.json
watch (padrão)	Observa as alterações nos arquivos de origem e cria o arquivo na alteração
construir	Construa todos os arquivos
limpar \ limpo	dist/Pasta Cleans
versão	Saídas da versão atual
construir-javascript	Construa arquivos JS
build-css	Construa arquivos CSS

Comando	Descrição
construir-ativos	Construa arquivos de ativos
servir-docs	Serve arquivos para a instância de documentos locais
build-docs	Cria arquivos para a pasta docs

## Arquivos de Configuração

semântico.json

As configurações da ferramenta Build são armazenadas em um arquivo especial chamado semantic.json. It pode ser incluído em qualquer pasta que esteja em uma pasta pai da pasta de instalação semântica.

Se você usou o npm ou o meteor, um semantic.json arquivo é gerado automaticamente para você na raiz do seu projeto. Se você usou um gerenciador de pacotes diferente, pode executar gulp install para executar o instalador interativo.

```
{
  // base path added to all other paths specified in "paths"
  "base": "",
  // current version of Semantic UI
  "version": "2.4.2",
  "paths": {
    "source": {
      // source theme.config
      "config" : "src/theme.config",
      // source definition folder
      "definitions" : "src/definitions/",
      // source site theme
      "site" : "src/site/",
      // source themes folder
    }
  }
}
```



```

    "themes"      : "src/themes/"
  },
  "output": {
    // packaged source (both compressed/uncompressed)
    "packaged"    : "dist/",
    // uncompressed source
    "uncompressed" : "dist/components/",
    // compressed source
    "compressed"   : "dist/components/",
    // output directory for theme assets
    "themes"       : "dist/themes/"
  },
  // directory for gulp clean task
  "clean"         : "dist/"
},
// when set to an integer permission, will set dist files with this file
permission
"permission" : false,
// whether gulp watch/build should run RTLCS
"rtl": false,
// will only include components with these names
"components": [
  "reset",
  "site",
  "button",
  "container",
  "divider",
  "flag",
  "header",
  "icon",
  "image",
  "input",
  "label",
  "list",
  "loader",
  "rail",
  "reveal",
  "segment",
  "step",
  "breadcrumb",
  "form",
  "grid",
  "menu",
  "message",
  "table",
  "ad",
  "card",
  "comment",
  "feed",
  "item",
  "statistic",
  "accordion",
  "checkbox",
  "dimmer",
  "dropdown",

```

```
"embed",
"modal",
"nag",
"popup",
"progress",
"rating",
"search",
"shape",
"sidebar",
"sticky",
"tab",
"transition",
"api",
"state",
"visibility"
],
// whether to include special project maintainer tasks
"admin": false
}
```

### **theme.config**

O LESS usa um arquivo de configuração especial theme.config que permite especificar qual tema cada componente deve usar. Se você ainda não gerou este arquivo, receberá uma mensagem de erro ao executar ferramentas de compilação

Você pode aprender mais theme.config consultando o guia de temas.

### **Importando Tarefas de Gulp**

Veja nossa seção de receitas para exemplos de como importar tarefas de gulp individuais para o seu Gulpfile personalizado.

## **Framework de programação para back-end**

### **Laravel**

O Laravel é um framework de desenvolvimento rápido para PHP, livre e de código aberto. Cujas o principal objetivo é permitir que você trabalhe de forma estruturada e rápida.

#### **As principais características**

Linguagem	<ul style="list-style-type: none"><li>• PHP</li></ul>
Banco de dados	<ul style="list-style-type: none"><li>• MySQL</li><li>• Postgress</li><li>• Redis</li><li>• MongoDB</li><li>• Cassandra</li><li>• SQL Server</li></ul>
Modelos de DB	<ul style="list-style-type: none"><li>• Relacional</li><li>• Orientado a objeto</li></ul>
Features	<ul style="list-style-type: none"><li>• Scaffold</li><li>• Internacionalização (i18n)</li></ul>
Princípios de desenvolvimento	<ul style="list-style-type: none"><li>• TDD (Test-driven development)</li><li>• Don't repeat yourself</li></ul>

Os bancos de dados listados nas características são altamente compatíveis com o Laravel. Sendo assim, basta trocar as configurações e apontar qual banco de dados deseja e o sistema já funciona se a estrutura estiver pronta.

Se não estiver, basta rodar as migrations que a estrutura do banco é criada. Tudo funcionará perfeitamente desde que o banco esteja modelado de forma relacional ou orientado a objeto, que são os dois modelos que o Framework trabalha.

## Seus principais pontos fortes e fracos

### Prós

- Implementação nativa do Composer
- Fácil escalabilidade do sistema
- Ferramentas e recursos formam um workflow funcional e completo
- Facilidade de manutenção do projeto
- Linha de comando
- Documentação ótima
- Fácil de encontrar soluções na internet

### Contras

- Complexidade da estrutura de trabalho dificulta a instalação em geral
- Rodar Laravel em hospedagens comuns não é produtivo
- Quantidade de recursos reflete em uma quantidade de conteúdo maior para se aprender

## Quem desenvolveu e quem mantém o software

Laravel é um framework PHP livre e open-source criado por Taylor B. Otwell para o desenvolvimento de sistemas web que utilizam o padrão MVC (model, view, controller).

De acordo com uma pesquisa feita em Março de 2015 com desenvolvedores, o Laravel foi listado como o framework PHP mais popular de 2015, seguido pelo Symfony2, Nette, CodeIgniter, Yii2 e outros. Em Agosto de 2015, o Laravel já era o principal framework de projetos PHP no GitHub.

Laravel foi desenvolvido sob o MIT License, tendo seu código-fonte hospedado no GitHub. Atualmente encontra-se na versão 5.7

## Quando o software foi lançado

A primeira versão beta do Laravel foi disponibilizada em 9 de Junho de 2011, seguido pelo lançamento do Laravel 1 no final do mesmo mês.

## Históricos de duas versões

Laravel 1 incluiu suporte embutido para autenticação, localização, modelos, visualizações, sessões, encaminhamento e outros mecanismos, mas não tinha suporte para controladores.

Em fevereiro de 2015 o Laravel chegou a sua quinta versão. Novas funcionalidades na versão Laravel 5 incluem suporte para o agendamento de tarefas periódicas realizadas através de um pacote chamado Scheduler, uma camada de abstração chamada Flysystem que permite o armazenamento remoto para ser usado da mesma forma como sistemas de arquivos locais, melhorias no processamento de bens de pacotes através de Elixir , e simplificado externamente no quesito autenticação (através do pacote Socialite). Laravel 5 também introduziu uma nova estrutura de árvore de diretório interno para aplicações desenvolvidas.

## **Endereço do site**

<https://laravel.com/>

## **Link para Download**

<https://github.com/laravel/laravel>

## **Instalação**

Requisitos do servidor

A estrutura do Laravel possui alguns requisitos de sistema. É claro que todos esses requisitos são atendidos pela máquina virtual Laravel Homestead , portanto, é altamente recomendável que você use Homestead como seu ambiente de desenvolvimento local do Laravel.

No entanto, se você não estiver usando o Homestead, será necessário garantir que seu servidor atenda aos seguintes requisitos:

- PHP >= 7.1.3
- Extensão PHP OpenSSL
- Extensão PHP PDO
- Extensão PHP Mbstring
- Extensão PHP Tokenizer
- Extensão PHP XML
- Extensão PHP Ctype
- Extensão PHP JSON
- Extensão PHP BCMath

Instalando o Laravel

O Laravel utiliza o Composer para gerenciar suas dependências. Então, antes de usar o Laravel, certifique-se de ter o Composer instalado em sua máquina

Instalador Via Laravel

Primeiro, baixe o instalador do Laravel usando o Composer:

```
composer global require laravel/installer
```

Certifique-se de colocar o diretório bin do fornecedor em todo o sistema do compositor \$PATH para que o executável do laravel possa ser localizado pelo seu sistema. Este diretório existe em diferentes localizações com base no seu sistema operacional; no entanto, alguns locais comuns incluem:

Mac OS: \$HOME/.composer/vendor/bin

Distribuições GNU / Linux: \$HOME/.config/composer/vendor/bin

Uma vez instalado, o comando criará uma nova instalação do Laravel no diretório que você especificar. Por exemplo, criará um diretório chamado contendo uma nova instalação do Laravel com todas as dependências do Laravel já instaladas:laravel newlaravel new blogblog

```
laravel new blog
```

Via Composer Create-Project

Alternativamente, você também pode instalar o Laravel emitindo o comando Composer no seu terminal:create-project

```
composer create-project --prefer-dist laravel/laravel blog
```

Servidor de Desenvolvimento Local

Se você tem o PHP instalado localmente e gostaria de usar o servidor de desenvolvimento integrado do PHP para servir seu aplicativo, você pode usar o servecomando Artisan. Este comando irá iniciar um servidor de desenvolvimento em :http://localhost:8000

```
php artisan serve
```

Naturalmente, opções de desenvolvimento local mais robustas estão disponíveis via Homestead e Valet .

## Configuração

**Diretório público**

Depois de instalar o Laravel, você deve configurar o documento / raiz da Web do servidor da Web para ser o publicdiretório. O neste diretório serve como o front controller para todas as solicitações HTTP que entram no seu aplicativo.index.php

## **Arquivos de Configuração**

Todos os arquivos de configuração do framework Laravel são armazenados no configdiretório. Cada opção é documentada, portanto sinta-se à vontade para examinar os arquivos e se familiarizar com as opções disponíveis para você.

## **Permissões de diretório**

Depois de instalar o Laravel, você pode precisar configurar algumas permissões. Os diretórios no storagee nos diretórios devem ser graváveis pelo seu servidor da Web ou o Laravel não será executado. Se você estiver usando a máquina virtual Homestead , essas permissões já devem estar definidas.bootstrap/cache

## **Chave de aplicativo**

A próxima coisa que você deve fazer depois de instalar o Laravel é definir sua chave de aplicação para uma string aleatória. Se você instalou o Laravel através do Composer ou do instalador do Laravel, esta chave já foi configurada para você pelo comando.php artisan key:generate

Normalmente, essa sequência deve ter 32 caracteres. A chave pode ser definida no .arquivo de ambiente. Se você não renomeou o arquivo para , você deve fazer isso agora. Se a chave do aplicativo não estiver definida, suas sessões de usuário e outros dados criptografados não serão seguros!.env.example.env

## **Configuração Adicional**

O Laravel não precisa de quase nenhuma outra configuração fora da caixa. Você está livre para começar a desenvolver! Entretanto, você pode querer revisar o arquivo e sua documentação. Ele contém várias opções, tais como e que você pode querer mudar de acordo com sua aplicação.config/app.phptimezonelocale

Você também pode querer configurar alguns componentes adicionais do Laravel, como:

- Cache
- Base de dados
- Sessão

## **Configuração do servidor da web**

URLs bonitas

Apache

O Laravel inclui um arquivo que é usado para fornecer URLs sem o front controller no caminho. Antes de servir o Laravel com o Apache, certifique-se de ativar o módulo para que o arquivo seja respeitado pelo servidor.  
`public/.htaccessindex.phpmod_rewrite.htaccess`

Se o `.htaccess` arquivo que acompanha o Laravel não funcionar com a instalação do Apache, tente esta alternativa:

```
Options +FollowSymLinks -Indexes
RewriteEngine On
RewriteCond %{HTTP:Authorization} .
RewriteRule .* - [E=HTTP_AUTHORIZATION:%{HTTP:Authorization}]

RewriteCond %{REQUEST_FILENAME} !-d
RewriteCond %{REQUEST_FILENAME} !-f
RewriteRule ^ index.php [L]
```

## Nginx

Se você estiver usando o Nginx, a seguinte diretiva na configuração do seu site direcionará todas as solicitações para o front controller: `index.php`

```
location / {
    try_files $uri $uri/ /index.php?$query_string;
}
```

Claro, ao usar Homestead ou Valet, URLs bonitas serão configuradas automaticamente.



## **Fontes de Referências:**

<https://richellyitalo.com/pt/semantic-ui-tente-algo-diferente/>

[https://www.slant.co/versus/504/520/~bootstrap\\_vs\\_semantic-ui](https://www.slant.co/versus/504/520/~bootstrap_vs_semantic-ui)

<https://www.linkedin.com/in/jacklukic>

<https://github.com/Semantic-Org/Semantic-UI/releases>

<https://semantic-ui.com/introduction/getting-started.html>

<https://semantic-ui.com/introduction/build-tools.html>

<http://blog.locaweb.com.br/artigos/frameworks-php-conheca-o-laravel/>

<https://medium.com/joaorobertopb/o-que-%C3%A9-laravel-porque-us%C3%A1-lo-955c95d2453d>

<http://dev.rbtech.info/codeigniter-vs-laravel-melhor/>

<https://pt.wikipedia.org/wiki/Laravel>

<https://www.portalgsti.com.br/laravel/sobre/>

<https://laravel.com/docs/5.7>