



# Desenvolvimento Web II

**Framework Laravel 5**

**Gerando Gráficos – Lavacharts**

<http://lavacharts.com/>

**Gil Eduardo de Andrade**





# LAVACHARTS

## Introdução:

- O desenvolvimento de aplicações web acarreta, muitas vezes, na necessidade de utilização de elementos visuais que permitam, ao usuário, analisar os dados manipulados pela aplicação;
- Nesse contexto, podemos utilizar, agregado ao Laravel, o pacote *Lavacharts*;





# LAVACHARTS

## Introdução:

- O pacote *Lavacharts* disponibiliza uma grande quantidade de tipos de gráficos: *barra*, *linhas*, *pizza*, *desempenho*, *etc*;
- Ele é baseado na *API Javascript Chart* do Google, e permite gerar de gráficos sem a necessidade da utilização do *javascript* puro;



# LAVACHARTS

## Instalação (via composer):

(Arquivo: *composer.json*)

```
"require-dev": {  
    "filp/whoops": "^2.0",  
    "fzaninotto/faker": "^1.4",  
    "mockery/mockery": "^1.0",  
    "nunomaduro/collision": "^2.0",  
    "phpunit/phpunit": "^7.0",  
    "khill/lavacharts" : "3.0.*"  
},
```

1º - adicione a dependência ao pacote lavacharts (*khill/lavacharts* : "3.0") no arquivo *composer.json*.

2º - via terminal, dentro do diretório da aplicação, utilize o composer (*composer update*) para atualizar e baixar o lavacharts e suas dependências.

```
[g1l3du4rd0@localhost pdf]$ composer update  
Loading composer repositories with package information  
Updating dependencies (including require-dev)  
Package operations: 2 installs, 13 updates, 0 removals  
- Updating symfony/css-selector (v4.0.9 => v4.0.10): Downloading (100%)  
- Updating symfony/var-dumper (v4.0.9 => v4.0.10): Loading from cache  
- Updating symfony/routing (v4.0.9 => v4.0.10): Downloading (100%)  
- Updating symfony/process (v4.0.9 => v4.0.10): Downloading (100%)  
- Updating symfony/http-foundation (v4.0.9 => v4.0.10): Downloading (100%)  
- Updating symfony/event-dispatcher (v4.0.9 => v4.0.10): Loading from cache  
- Updating symfony/debug (v4.0.9 => v4.0.10): Downloading (100%)  
- Installing symfony/polyfill-ctype (v1.8.0): Downloading (100%)  
- Updating symfony/http-kernel (v4.0.9 => v4.0.10): Downloading (100%)  
- Updating symfony/finder (v4.0.9 => v4.0.10): Downloading (100%)  
- Updating symfony/console (v4.0.9 => v4.0.10): Downloading (100%)  
- Updating symfony/translation (v4.0.9 => v4.0.10): Downloading (100%)  
- Updating laravel/framework (v5.6.22 => v5.6.23): Downloading (100%)  
- Updating psy/psysh (v0.9.3 => v0.9.4): Downloading (100%)  
- Installing khill/lavacharts (3.0.8): Downloading (100%)  
khill/lavacharts suggests installing khill/datatableplus (Adds extra features to  
such as csv and laravel collection parsing.)  
Writing lock file  
Generating optimized autoload files  
> Illuminate\Foundation\ComposerScripts::postAutoloadDump  
> @php artisan package:discover  
Discovered Package: fideloper/proxy  
Discovered Package: laravel/tinker  
Discovered Package: nunomaduro/collision  
Discovered Package: barryvdh/laravel-dumpdf  
Package manifest generated successfully.  
[g1l3du4rd0@localhost pdf]$
```



# LAVACHARTS

## Configuração – Provider / Alias:

*(Arquivo: config/app.php – Considerando Laravel 5.5 ou Superior)*

```
'providers' => [  
    Khill\Lavacharts\Laravel\LavachartsServiceProvider::class,  
],  
  
'aliases' => [  
    'Lava' => Khill\Lavacharts\Laravel\LavachartsFacade::class,  
],
```





# LAVACHARTS

## Aplicação Exemplo

[http://www.gileduardo.com.br/ifpr/dwii/downloads/dwii\\_exdoc14.zip](http://www.gileduardo.com.br/ifpr/dwii/downloads/dwii_exdoc14.zip)



# LAVACHARTS

Aplicação Adaptada

## SETA – Menu e Rotas Principais:

(Adaptação: Exemplo da Aula Anterior com DOMPDF)

SETA - Sistema de Entrega de Trabalhos e Atividades [Home](#)

 Menu Principal

 Curso	 Turma	 Aluno	 Disciplina
 Conceito	 Relatório	 Importar	 Exportar

# LAVACHARTS

Aplicação Adaptada

## SETA – View Aluno:

(Arquivos: *aluno.blade.php*)



### Alunos Cadastrados

Foram adicionados 8 botões que permitem gerar e visualizar 8 tipos diferentes de relatórios. Os dados utilizados na geração são o número de alunos cadastrados no sistema por turma.

Cadastrar Novo Aluno

buscar

Relatório dos Alunos (PDF)

Certificado dos Alunos (PDF)

Gráfico de Barra

Gráfico de Área

Gráfico de Coluna





Gráfico de Círculo

Gráfico de Ponteiro

Gráfico de Mapa Mundi

Gráfico de Pizza

Gráfico de Linha

ID	ALUNO	CURSO	TURMA	EVENTOS
3	GIL EDUARDO DE ANDRADE	EMI - TÉCNICO EM INFORMÁTICA	INFO 17	   



# LAVACHARTS

Aplicação Adaptada

## SETA – Rotas e Controle:

(Arquivos: *routes/web.php* e *AlunoController.php*)

```
Route::get('/aluno/plotar/{id}', 'AlunoController@plotar');
```

```
public function plotar($id) {  
  
    switch ($id) {  
        // Barra  
        case 1:  
            return self::Barra();  
            break;  
        // Área  
        case 2:  
            return self::Area();  
            break;  
    }  
}
```

No arquivo de rotas foi adicionada uma nova rota que invoca o método “plotar” da classe de controle do aluno. Esse método recebe como parâmetro um número inteiro que indica qual tipo de gráfico deve ser plotado.

Dentro do método “plotar” o valor \$id, recebido como parâmetro, é testado, e de acordo com o seu valor um método específico (criado) para cada gráfico é invocado.

# LAVACHARTS

Aplicação Adaptada

## SETA – Configurando os Gráficos:

(Arquivo: *AlunoController.php*)

```
$info15 = AlunoModel::where('turma', '=', 'INFO 15')->count();
$info16 = AlunoModel::where('turma', '=', 'INFO 16')->count();
$info17 = AlunoModel::where('turma', '=', 'INFO 17')->count();
$info18 = AlunoModel::where('turma', '=', 'INFO 18')->count();
$tads16 = AlunoModel::where('turma', '=', 'TADS 16')->count();
$tads17 = AlunoModel::where('turma', '=', 'TADS 17')->count();
$tads18 = AlunoModel::where('turma', '=', 'TADS 18')->count();
```



Todos os dados utilizados para geração dos gráficos, exceto o gráfico de linhas onde são utilizados dados estáticos, são selecionados da base de dados, mais especificamente da tabela de alunos. Para tal é feita uma contagem de alunos por turmas cadastradas, como mostrado na codificação acima, ela se repete para cada método construído para gerar cada um dos 8 tipos de gráficos exemplificados.

# LAVACHARTS

Aplicação Adaptada

## SETA – Configurando os Gráficos (Barras):

(Arquivo: *AlunoController.php*)

```
// Gráfico de Barras
$lava = new Lavacharts;
$alunos = $lava->DataTable();
$alunos->addStringColumn('Turma')
->addNumberColumn('Nr. Alunos')
->addRow(['INFO 15', $info15])
->addRow(['INFO 16', $info16])
->addRow(['INFO 17', $info17])
->addRow(['INFO 18', $info18])
->addRow(['TADS 16', $tads16])
->addRow(['TADS 17', $tads17])
->addRow(['TADS 18', $tads18]);

$lava->BarChart('Dados', $alunos);

// Invoca a View
return view('alunoGrafico')->with(compact('lava'))
->with('tipo', 'BarChart');
```

Após a criação de uma instância “Lavacharts” para o novo gráfico que será gerado, uma tabela de dados é definida para receber as informações que serão exibidas.

Duas colunas (eixos) são definidos (adicionados) para o gráfico, um no formato String (nome das turmas) e outro no formato Number (número de alunos). Após a definição do tipo de dados e Rótulo de cada Eixo, são adicionado os dados, respeitando o formato especificado, ou seja, String, Number.

Gráfico do tipo Barras é definido, os dados \$alunos recebem o rótulo ‘Dados’.

# LAVACHARTS

Aplicação Adaptada

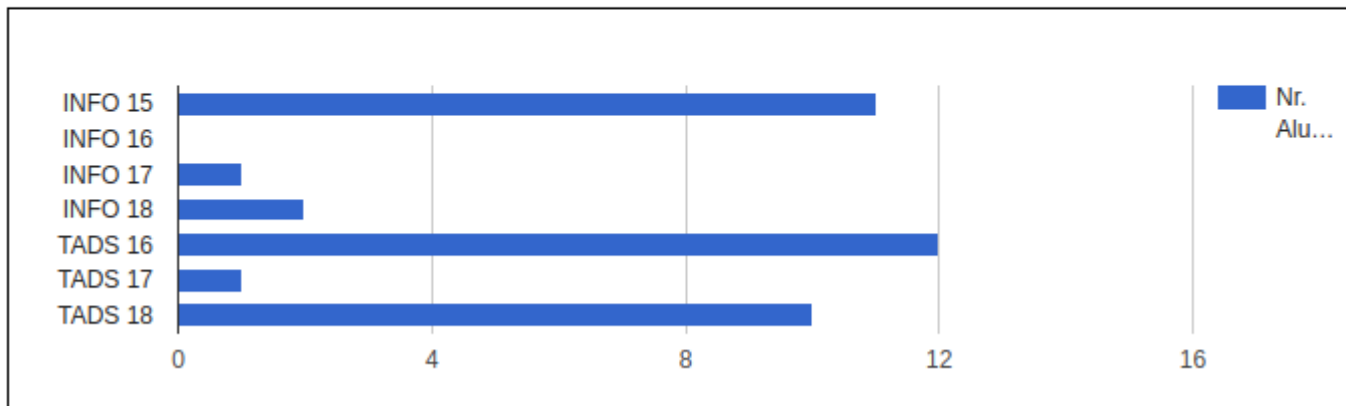
## SETA – Renderizando os Gráficos (Barras):

(Arquivo: *alunoGrafico.blade.php*)

```
<div id='lava_div' style='border:1px solid black'>
  <?php echo $lava->render($tipo, 'Dados', 'lava_div'); ?>
</div>
```



### Exemplos de Gráficos - Lavacharts



Uma `<div>` foi definida para renderizar o gráfico, o método “render()” foi invocado, recebendo o tipo do gráfico, os dados e o id da div onde será apresentado.



# LAVACHARTS

Aplicação Adaptada

## SETA – Configurando os Gráficos (Colunas):

(Arquivo: AlunoController.php)

```
$alunos->addStringColumn('Turma')
->addNumberColumn('Nr. Alunos')
->addRow(['INFO 15', $info15])
->addRow(['INFO 16', $info16])
->addRow(['INFO 17', $info17])
->addRow(['INFO 18', $info18])
->addRow(['TADS 16', $tads16])
->addRow(['TADS 17', $tads17])
->addRow(['TADS 18', $tads18]);

$lava->ColumnChart('Dados', $alunos, [
    'title' => 'Total de Alunos / Turma',
    'titleTextStyle' => [
        'color' => '#eb6b2c',
        'fontSize' => 14
    ]
]);
```

Novamente foram utilizadas duas colunas (eixos) para o gráfico, um no formato String (nome das turmas) e outro no formato Number (número de alunos). Após essa definição foram adicionado os dados, respeitando o formato de especificado, ou seja, String, Number.

Observe que para o gráfico de colunas foi definido, no momento da especificação do modelo do gráfico (Coluna) e dados a serem utilizados, um título e o estilo a ser usado para esse texto. Foram definidos a cor e o tamanho de fonte.

# LAVACHARTS

Aplicação Adaptada

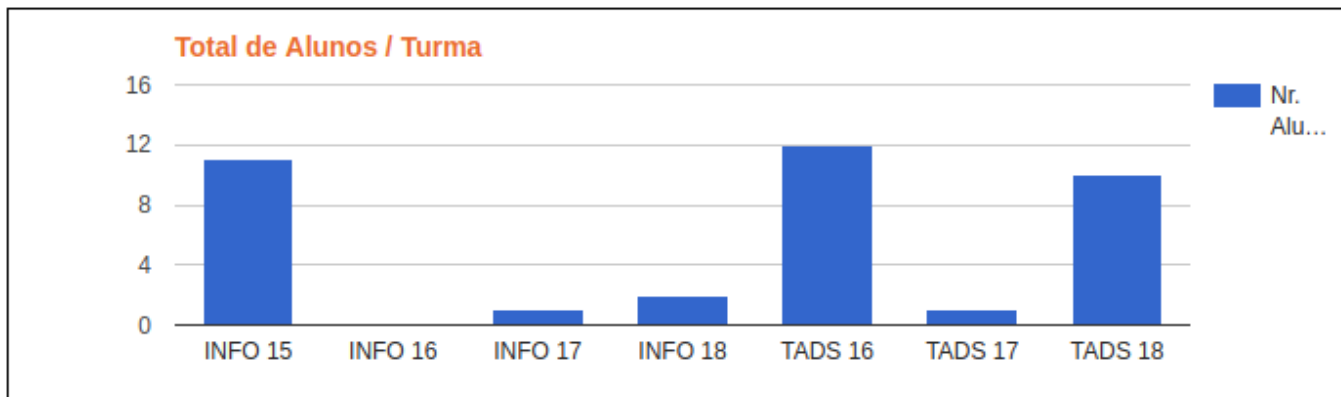
## SETA – Renderizando os Gráficos (Colunas):

(Arquivo: *alunoGrafico.blade.php*)

```
<div id='lava_div' style='border:1px solid black'>
  <?php echo $lava->render($tipo, 'Dados', 'lava_div'); ?>
</div>
```



## Exemplos de Gráficos - Lavacharts



Uma `<div>` foi definida para renderizar o gráfico, o método “render()” foi invocado, recebendo o tipo do gráfico, os dados e o id da div onde será apresentado.



# LAVACHARTS

## Outros Modelos de Gráficos:

- Os outros modelos de gráficos não foram abordados em detalhes porque seguem a mesma lógica dos gráficos de Barra e Coluna, possuindo algumas configurações peculiares que podem ser compreendidas pela análise do código-fonte disponibilizado;





# LAVACHARTS

## Exemplos Utilizados no Documento

[http://www.gileduardo.com.br/ifpr/dwii/downloads/dwii\\_exdoc14zip](http://www.gileduardo.com.br/ifpr/dwii/downloads/dwii_exdoc14zip)

## Exercício sobre o Conteúdo

[http://www.gileduardo.com.br/ifpr/dwii/downloads/dwii\\_pratica14.pdf](http://www.gileduardo.com.br/ifpr/dwii/downloads/dwii_pratica14.pdf)

