

## TECNÓLOGO EM ANÁLISE E DESENVOLVIMENTO

### Disciplina de Desenvolvimento Web II

Aula 06: Relatório PDF / Gráficos Google Chart

*Gil Eduardo de Andrade*

## CONCEITOS PRELIMINARES

### GERAÇÃO DE GRÁFICOS E RELATÓRIOS PDF

**Dompdf** (<https://github.com/dompdf/dompdf>)

O Dompdf é uma ferramenta que permite converter código HTML num arquivo PDF. Em sua essência, ele trabalha com um layout HTML, compatível com CSS 2.1, e um mecanismo de renderização escrito em PHP. O Dompdf é um renderizador baseado em estilo: ele baixa e lê folhas de estilo externas, tags com estilos embutidos e atributos de estilo para elementos HTML individuais. Ele também oferece suporte à maioria dos atributos HTML de apresentação.

#### Instalação

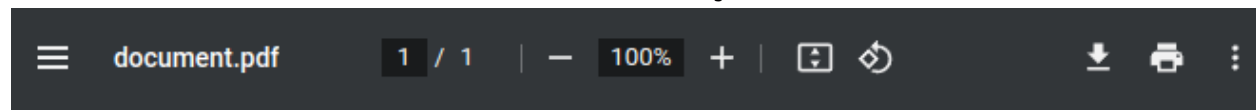
```
composer require dompdf/dompdf
```

#### Exemplo de Utilização

```
// Importa o pacote
use Dompdf\Dompdf;
...
public function test() {
    // Instancia um Objeto da Classe Dompdf
    $dompdf = new Dompdf();
    // Carrega o HTML
    $dompdf->loadHtml('hello world');
    // (Opcional) Configura o Tamanho e Orientação da Página
    $dompdf->setPaper('A4', 'landscape');
    // Converte o HTML em PDF
}
```

```
$dompdf->render();  
// Serializa o PDF para Navegador  
$dompdf->stream();  
}
```

## RESULTADO DA CODIFICAÇÃO ANTERIOR



hello world

### Google Chart ([https://developers.google.com/chart/interactive/docs/quick\\_start?hl=pt-br](https://developers.google.com/chart/interactive/docs/quick_start?hl=pt-br))

O Google Chart oferece um ferramental gráfico para visualização de dados em páginas web, desde gráficos de linhas simples até mapas complexos de árvores hierárquicas. A maneira mais comum de usar o Google Chart é através de código JavaScript incorporado à página da Web. Para tal, basta carregar a biblioteca do Google Chart, especificar os dados que devem ser transformados em gráfico, selecionar as opções de personalização disponíveis e criar um objeto do gráfico vinculado a uma **<div>** com um determinado **“id”**, responsável por exibir o gráfico em questão.

Os gráficos são renderizados usando a tecnologia HTML5/SVG para oferecer compatibilidade entre navegadores (incluindo VML para versões mais antigas do IE) e portabilidade entre plataformas para iPhones, iPads e Android. Tal abordagem garante que os usuários não precisarão manipular (se preocupar) plug-ins ou qualquer software.

### Instalação

Arquivo: /resources/views/templates/main.blade.php

```
<!-- Google Chart / Importação - Arquivo Remoto -->
<script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>

<!-- Google Chart / Importação - Arquivo Local -->
<script type="text/javascript" src="{asset('js/google-chart-loader.js')}"></script>
```

**OBS.:** é possível efetuar uma das duas opções apresentadas acima. A primeira não necessita que o arquivo JS seja baixado, pois a importação é remota. Na segunda considera-se que o arquivo foi baixado e colocado na pasta “**public/js**”, sendo nomeado como “**google-chart-loader.js**”.

### Exemplo de Utilização

#### Classe de Controller - Carrega e Repassa os Dados para View

```
public function test() {

    $data = json_encode([

        ["NOME", "TOTAL DE HORAS"],

        ["MARIA", 150],

        ["CARLOS", 90],

        ["JOÃO", 232],

        ["ANA", 197],

    ]);

    return view('grafico.exemplo', compact(['data']));

}
```

**OBS.:** os dados estáticos gerados, como exemplo, são colocados numa estrutura *Array*, onde a primeira linha possui duas strings, que indicam (rótulo) quais são os dados que serão apresentados no gráfico. Para o envio ao arquivo de visualização, os dados em formato *Array* são codificados em formato JSON.

## Arquivo de Visualização - Utiliza a Biblioteca Google Chart e Renderiza os Gráficos

```
<div class="row">

  <div class="col text-center" id="barra" style="width: 420px; height: 280px;"></div>

  <div class="col text-center" id="pizza" style="width: 420px; height: 280px;"></div>

</div>

<div class="row mt-2">

  <div class="col text-center" id="coluna" style="width: 420px; height: 280px;"></div>

  <div class="col text-center" id="linha" style="width: 420px; height: 280px;"></div>

</div>

<script type="text/javascript">

  var data_graph = <?php echo $data ?>;

  google.charts.load('current', {'packages':['corechart']})

  google.charts.setOnLoadCallback(drawChart);

  function drawChart() {

    // Dados do Gráfico

    let data = google.visualization.arrayToDataTable(data_graph);

    // GRÁFICO DE BARRAS

    // Opções de Configuração

    options = {

      title: 'TOTAL DE HORAS DOS ALUNOS',

      colors: ['#198754'],

      legend: 'none',

      hAxis: {

        title: 'Horas Validadas',
```

```
        titleTextStyle: {  
            fontSize: 12,  
            bold: true,  
        }  
    },  
    vAxis: {  
    },  
};  
  
// DESENHA GRÁFICO DE BARRAS  
  
chart = new google.visualization.BarChart(document.getElementById('barra'));  
  
chart.draw(data, options);  
  
// ===== //  
  
// GRÁFICO DE PIZZA  
  
// Opções de Configuração  
  
options = {  
    title: 'TOTAL DE HORAS DOS ALUNOS',  
    is3D: true  
};  
  
// DESENHA GRÁFICO DE PIZZA  
  
chart = new google.visualization.PieChart(document.getElementById('pizza'));  
  
chart.draw(data, options);  
  
// ===== //  
  
// GRÁFICO DE COLUNA  
  
// Opções de Configuração  
  
options = {  
    title: 'TOTAL DE HORAS DOS ALUNOS',  
    colors: ['#198754'],
```

```
    legend: 'none',

    hAxis: {

    },

    vAxis: {

        title: 'Horas Validadas',

        titleTextStyle: {

            fontSize: 12,

            bold: true,

        }

    }

};

// DESENHA GRÁFICO DE COLUNAS

chart = new google.visualization.ColumnChart(document.getElementById('coluna'));

chart.draw(data, options);

// ===== //

// GRÁFICO DE LINHA

// Opções de Configuração

options = {

    title: 'TOTAL DE HORAS DOS ALUNOS',

    colors: ['#198754'],

    curveType: 'function',

    legend: { position: 'bottom' }

};

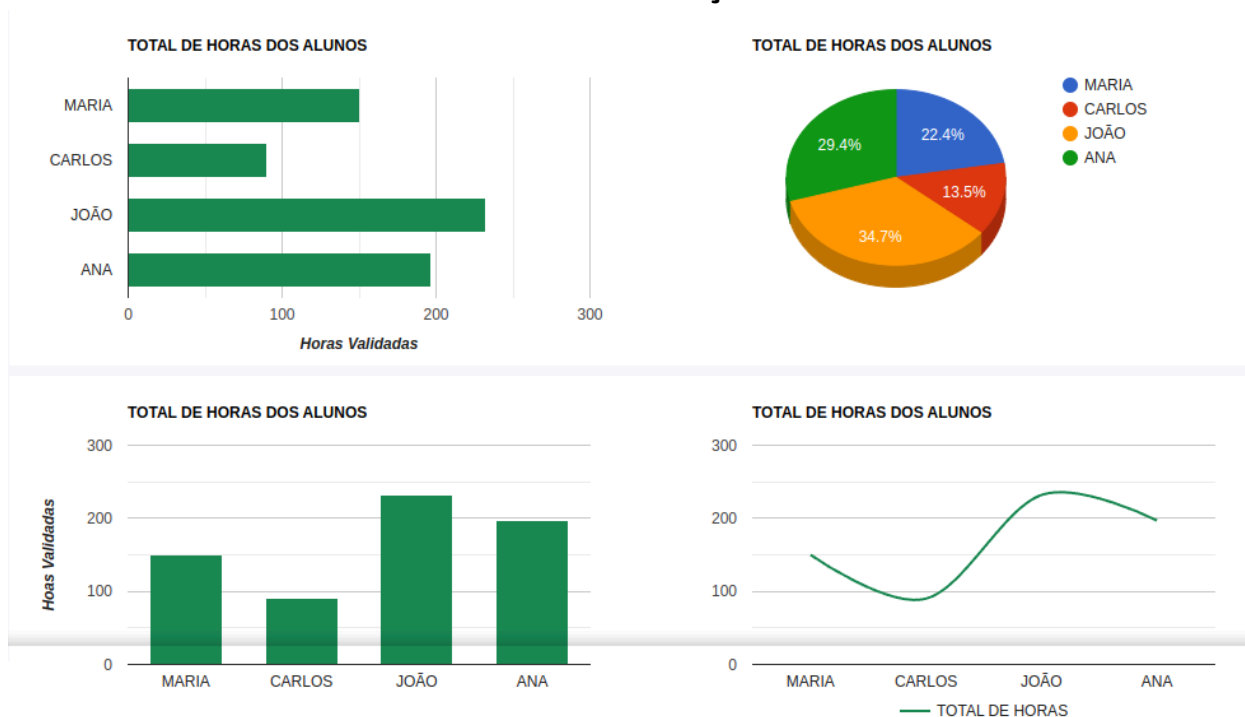
chart = new google.visualization.LineChart(document.getElementById('linha'));

chart.draw(data, options);

}

</script>
```

## RESULTADO DA CODIFICAÇÃO ANTERIOR



## IMPLEMENTAÇÃO DO SISTEMA - RF12 / RF14

Gráficos de Horas Afins

Relatório PDF de Horas Afins

### REQUISITO FUNCIONAL 12

Geração de relatórios de horas solicitadas / avaliadas / comprovadas, para um aluno específico ou para uma turma inteira.

(Frontend)

Criação do Componente “listitem”

Arquivo: `app/views/Components/listitem.php`

```
...  
  
public $data;           // array com os dados a serem apresentados (alunos)  
  
public $field;         // campo do array a ser exibido na listagem (nome)  
  
public $primaryroute;  // Rota para o relatório geral (turma)  
  
public $secondaryroute; // Rota para o relatório específico (aluno)  
  
public $id;            // Parâmetro passado para rota geral (turma_id)  
  
public function __construct($data, $field, $primaryroute, $secondaryroute, $id) {  
  
    $this->data = $data;  
  
    $this->field = $field;  
  
    $this->primaryroute = $primaryroute;  
  
    $this->secondaryroute = $secondaryroute;  
  
    $this->id = $id;  
  
    }  
  
...
```

#### Arquivo: resources/views/components/listitem.php

```
<div>  
  
    @if(count($data) > 0)  
  
        <div class="text-start mt-3 mb-3">  
  
            <a href="{{ route($primaryroute, $id) }}" target="_blank" ...>  
  
                <svg ...>  
  
                    <path .../>  
  
                    <path .../>  
  
                </svg>  
  
            </a>  
  
        </div>  
  
    @endif  
  
    <ul class="list-group list-group-flush">
```



```
@foreach ($data as $item)

    <li class="list-group-item text-success">

        <div class="row">

            <div class="col-10">

                {{ $item->$field }}

            </div>

            <div class="col-2 text-end">

                <a href="{{ route($secondaryroute, $item->id) }}" target="_blank" ...>

                    <svg ...>

                        <path ... />

                        <path .../>

                    </svg>

                </a>

            </div>

        </div>

    </li>

@endforeach

</ul>

</div>
```

## Adaptação do Componente “tablist”

Arquivo: resources/views/components/tablist.php

```
...

@if($contenttype == "datatable")

    <x-datatable

        title=""
```

```
        :header="$header"  
  
        :crud="$crud"  
  
        :data="$tab[$data] "  
  
        :fields="$fields"  
  
        :hide="$hide"  
  
        :remove="$fielddata"  
  
        :create="$create"  
  
        id=""  
  
        :modal="$cont"  
  
    />  
  
    @elseif($contenttype == "listitem")  
  
        <x-listitem  
  
            :data="$tab[$data] "  
  
            field="nome"  
  
            :primaryroute="$primaryroute"  
  
            :secondaryroute="$secondaryroute"  
  
            id="{{ $tab[$id] }}"  
  
        />  
  
    @endif  
  
    ...
```

A seguir é apresentada a interface/view criada para contemplar o Requisito Funcional 12 - *O sistema deve permitir a geração de relatórios PDF sobre as horas solicitadas / aceitas de alunos e turmas.*

**Arquivo: resources/views/relatorio/index.blade.php**

## RELATÓRIO (PDF) DE HORAS

TADS2022

TADS2023



CARLOS HENRIQUE DIAS



LARISSA MAIA



RODRIGO CARDOSO TAY



A seguir é apresentada a view/html que especifica o layout para o relatório PDF das horas solicitadas / aceitas para uma determinada turma.

**View: “resources/views/relatorio/pdf/turma.blade.php”**

```
<html>

  <!-- Não deixou carregar CSS remoto / Verificar posteriormente -->

  <style>

    .cell-header { border-style: solid; border-width: 1px; }

    .font-header { font-size: 14px; font-weight: bold; }

    .xlarge-width { width: 320px; }

    .small-width { width: 90px; }

    .font-content { font-size: 12px; }

  </style>

  <div style="width: 20%; float:left">

  </div>
```

```
<div style="width: 80%; float:right">

    <h1 style="text-align: center;">SIGAC</h1>

    <h3 style="text-align: center;">Sistema de Gerenciamento de Atividades Complementares</h3>

</div>

<br><br><br><br><br><br><br>

<div style="text-align: center;">

    <h2>Relatório Horas Solicitadas / {{ $data['turma'] }}</h2>

</div>

<div class="div-border">

    <table>

        <thead>

            <tr style="text-align: center;">

                <th class="cell-header xlarge-width font-header">ALUNO</th>

                <th class="cell-header small-width font-header">SOLICITADO</th>

                <th class="cell-header small-width font-header">VALIDADO</th>

                <th class="cell-header small-width font-header">LANÇADO</th>

                <th class="cell-header small-width font-header">TOTAL</th>

            </tr>

        </thead>

        <tbody>

            @foreach($data['aluno'] as $item)

                <tr style="text-align: center;">

                    <td class="xlarge-width font-content">{{substr($item->nome, 0, 45)}}</td>

                    <td class="small-width font-content">{{ $item->solicitado }}</td>

                    <td class="small-width font-content">{{ $item->validado }}</td>

                    <td class="small-width font-content">{{ $item->lancado }}</td>

                    <td class="small-width font-content">{{ $item->validado + $item->lancado }}</td>

                </tr>

            @endforeach

        </tbody>

    </table>

</div>
```

```
</tr>

    @endforeach

</tbody>

</table>

</div>

</html>
```

## (Backend) Repositório de Aluno

Arquivo: /app/Http/Repository/AlunoRepository.php

```
...

public function selectHoursByClass($turma_id) {

    $turma = (new TurmaRepository())->findByIdWith(['curso'], $turma_id);

    $alunos = $this->findByColumnWith('turma_id', $turma_id, ['turma', 'curso']);

    $aux = array();

    $cont = 0;

    foreach($alunos as $item) {

        // Solicitados pelo aluno

        $hours = (new DocumentoRepository())->getTotalHoursByStudent($item->user_id);

        // Lançados pelos professores

        $total_entry = (new ComprovanteRepository())->getTotalHoursByStudent($item->id);

        $aux[$cont] = (Object) [

            "nome" => $item->nome,

            "solicitado" => $this->convertNullToZero($hours->total_in),

            "validado" => $this->convertNullToZero($hours->total_out),

            "lancado" => $this->convertNullToZero($total_entry),
```

```
    ];  
  
    $cont++;  
  
    }  
  
    $data = collect();  
  
    $data["turma"] = $turma->curso->sigla . "-" . $turma->ano;  
  
    $data["aluno"] = $aux;  
  
    return $data;  
  
    }  
...  
  
public function convertNullToZero($value) {  
  
    if(is_null($value)) return 0;  
  
    return $value;  
  
    }  
...  
...
```

## Repositório do Documento

Arquivo: /app/Http/Repository/DocumentoRepository.php

```
...  
  
public function getTotalHoursByStudent($user_id) {  
  
    return Documento::Where('user_id', $user_id)  
  
        ->selectRaw("SUM(horas_in) as total_in")  
  
        ->selectRaw("SUM(horas_out) as total_out")  
  
        ->first();  
  
    }  
...  
...
```

## Repositório do Comprovante

Arquivo: /app/Http/Repository/ComprovanteRepository.php

```
...  
  
public function getTotalHoursByStudent($aluno_id) {  
  
    return Comprovante::where('aluno_id', $aluno_id)->sum('horas');  
  
}  
  
public function getHoursByStudent($aluno_id) {  
  
    return $this->findByColumnWith('aluno_id', $aluno_id, ['categoria', 'user']);  
  
}  
  
...
```

## Classe de Controle - Relatório (Criada)

Arquivo: /app/Http/Controller/RelatorioController.php

```
...  
  
private $dompdf;  
  
private $curso_id = 2; //temporário, até implementar autenticação  
  
public function __construct() {  
  
    $this->dompdf = new Dompdf();  
  
    $this->dompdf->setPaper('A4', 'portrait'); // opcional  
  
}  
  
public function index() {  
  
    $data = (new AlunoRepository())->selectAllByTurmas($this->curso_id);  
  
    return view('relatorio.index', compact(['data']));  
  
}  
  
public function reportClass($turma_id) {
```

```

$data = (new AlunoRepository())->selectHoursByClass($turma_id);

// Carrega o HTML da View

$this->dmpdf->loadHtml(view('relatorio.pdf.turma', compact('data')));

// Converte o HTML em PDF

$this->dmpdf->render();

// Serializa o PDF para Abertura em uma Nova Aba

$this->dmpdf->stream("relatorio-horas-turma.pdf", array("Attachment" => false));

}

...

```

## RESULTADO DA CODIFICAÇÃO ANTERIOR

# SIGAC

## Sistema de Gerenciamento de Atividades Complementares

### Relatório Horas Solicitadas / TADS-2022

| ALUNO                | SOLICITADO | VALIDADO | LANÇADO | TOTAL |
|----------------------|------------|----------|---------|-------|
| CARLOS HENRIQUE DIAS | 45         | 0        | 15      | 15    |
| LARISSA MAIA         | 0          | 0        | 0       | 0     |
| RODRIGO CARDOSO TAY  | 80         | 10       | 50      | 60    |

## REQUISITO FUNCIONAL 14

Geração de gráficos mostrando o total de horas cumpridas pelos alunos de uma turma.

### (Frontend)

View: "resources/views/relatorio/pdf/turma.blade.php"

```
@extends('templates/main', ['titulo'=>"GRÁFICO DE HORAS / TURMA"])
```



```
@section('conteudo')
<div class="d-flex flex-row">
  @foreach($data as $item)
    <div class="form-check form-check-inline">
      <input
        class="btn-check"
        type="radio"
        name="turma"
        id="turma_{{ $item['turma'] }}"
        value="{{ $item['turma'] }}"
        onclick="selectClass({{ $item }})"
      />
      <label for="turma_{{ $item['turma'] }}">{{ $item['turma'] }}</label>
    </div>
  @endforeach
</div>
<hr>
<div class="row overflow-auto mt-2">
  <div class="col text-center" id="barra" style="width: 480px; height: 360px;"></div>
</div>
@endsection

@section('script')
<script type="text/javascript">
  function selectClass(turma) {
    let cont = 1
    let data_graph = [
      ['Aluno', 'Solicitado', 'Lançado', 'Validado', { role: 'annotation' }]
    ]
    turma.aluno.forEach((element) => {
      data_graph[cont] = [
        element.nome,
        element.solicitado,
        element.lancado,
        element.validado + element.lancado,
        ''
      ]
      cont++
    })

    google.charts.load('current', {'packages': ['corechart']})
    google.charts.setOnLoadCallback(drawChart(data_graph));
  }

  function drawChart(data_graph) {
    let data = google.visualization.arrayToDataTable(data_graph);
    // Opções de Configuração
    let options = {
      title: 'TOTAL DE HORAS DOS ALUNOS',
      legend: { position: 'top', maxLines: 3 },
      isStacked: true,
    }
  }
</script>

```

```
        hAxis: {
            title: 'Horas Validadas',
            titleTextStyle: {
                fontSize: 12,
                bold: true,
            },
            textPosition: 'none'
        },
        vAxis: {
        },
    };
    // RÓTULOS
    let view = new google.visualization.DataView(data);
    view.setColumns([
        0,
        1,
        { calc: "stringify", sourceColumn: 1, type: "string", role: "annotation" },
        2,
        { calc: "stringify", sourceColumn: 2, type: "string", role: "annotation" },
        3,
        { calc: "stringify", sourceColumn: 3, type: "string", role: "annotation" },
    ]);
    // DESENHA GRÁFICO DE BARRAS
    chart = new google.visualization.BarChart(document.getElementById('barra'));
    chart.draw(view, options);
    }
</script>
@endsection
```

---

(Backend)

### Classe de Controle - Gráfico (Criada)

```
...
private $curso_id = 2; //temporário, até implementar autenticação

public function graphClass() {
    $turmas = (new TurmaRepository())->findByColumn('curso_id', $this->curso_id);
    $data = array();
    $cont = 0;
    foreach($turmas as $turma) {
        $data[$cont] = (new AlunoRepository())->selectHoursByClass($turma->id);
        $cont++;
    }
}
```

```
}  
// return $data;  
return view('grafico.aluno', compact('data'));  
}  
...  
}
```

## RESULTADO DA CODIFICAÇÃO ANTERIOR

