

**UNIVERSIDADE FEDERAL DO RIO GRANDE
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E
DESENVOLVIMENTO DE SISTEMAS**

SimpleReport

Framework em PHP para criação de relatórios
modelados no iReport

Orientador: Prof. Marcio Torres
Graduando: Anderson da Silva de Souza

Introdução

- Importância dos relatórios nos softwares
- Aplicações Web
- Criação de relatórios no PHP
- PDF
 - Fidelidade do layout
 - Portabilidade

Problema

- Gerar relatórios em PDF de forma simples e visual para aplicações PHP.

Alternativas

- FPDF
- PDFLib

Mas não são visuais!

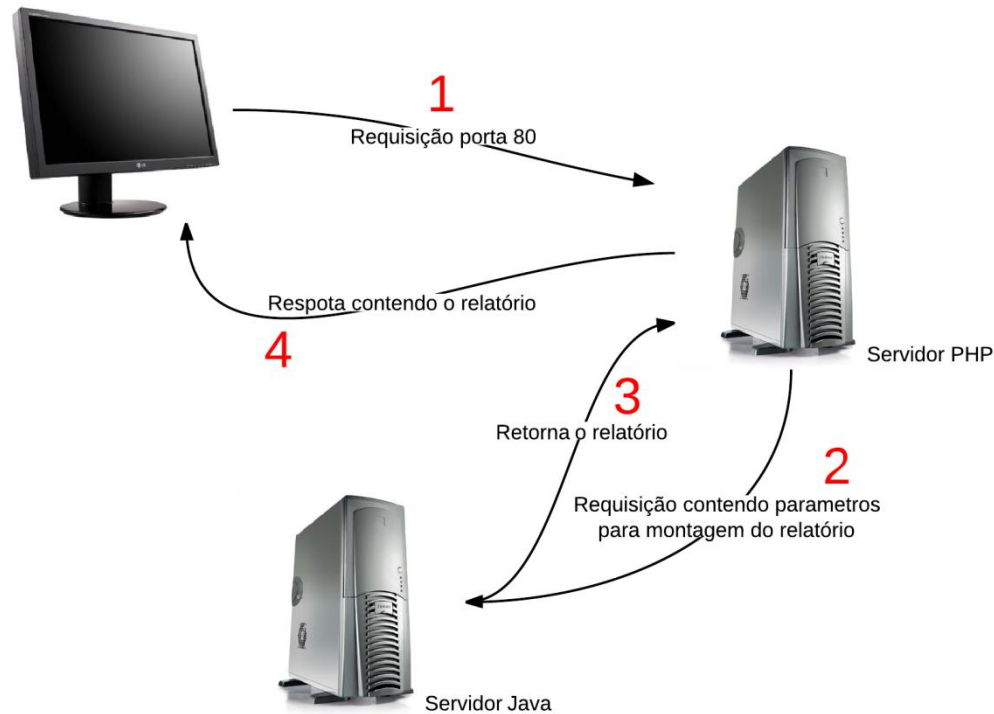
```
require('fpdf.php');
class PDF extends FPDF{
    function Header(){
        $this->Image('logo.png',10,6,30);
        $this->SetFont('Arial','B',15);
        $this->Cell(80);
        $this->Cell(30,10,'Title',1,0,'C');
        $this->Ln(20);
    }
    function Footer(){
        $this->SetY(-15);
        $this->SetFont('Arial','I',8);
        $this->Cell(0,10,'Page '.$this->PageNo().'/{nb}',0,0,'C');
    }
}
$pdf = new PDF();
$pdf->AliasNbPages();
$pdf->AddPage();
$pdf->SetFont('Times','',12);
for($i=1;$i<=40;$i++){
    $pdf->Cell(0,10,'Printing line number '.$i,0,1);
}

$pdf->Output();
```

Alternativas visuais

- PHP JasperXML
 - Semelhante ao SimpleReport
 - Projeto descontinuado
 - Não é extensível
- JasperReport
 - Necessita de um servidor Java
 - Conhecimento de duas linguagens
 - Demanda mais recursos

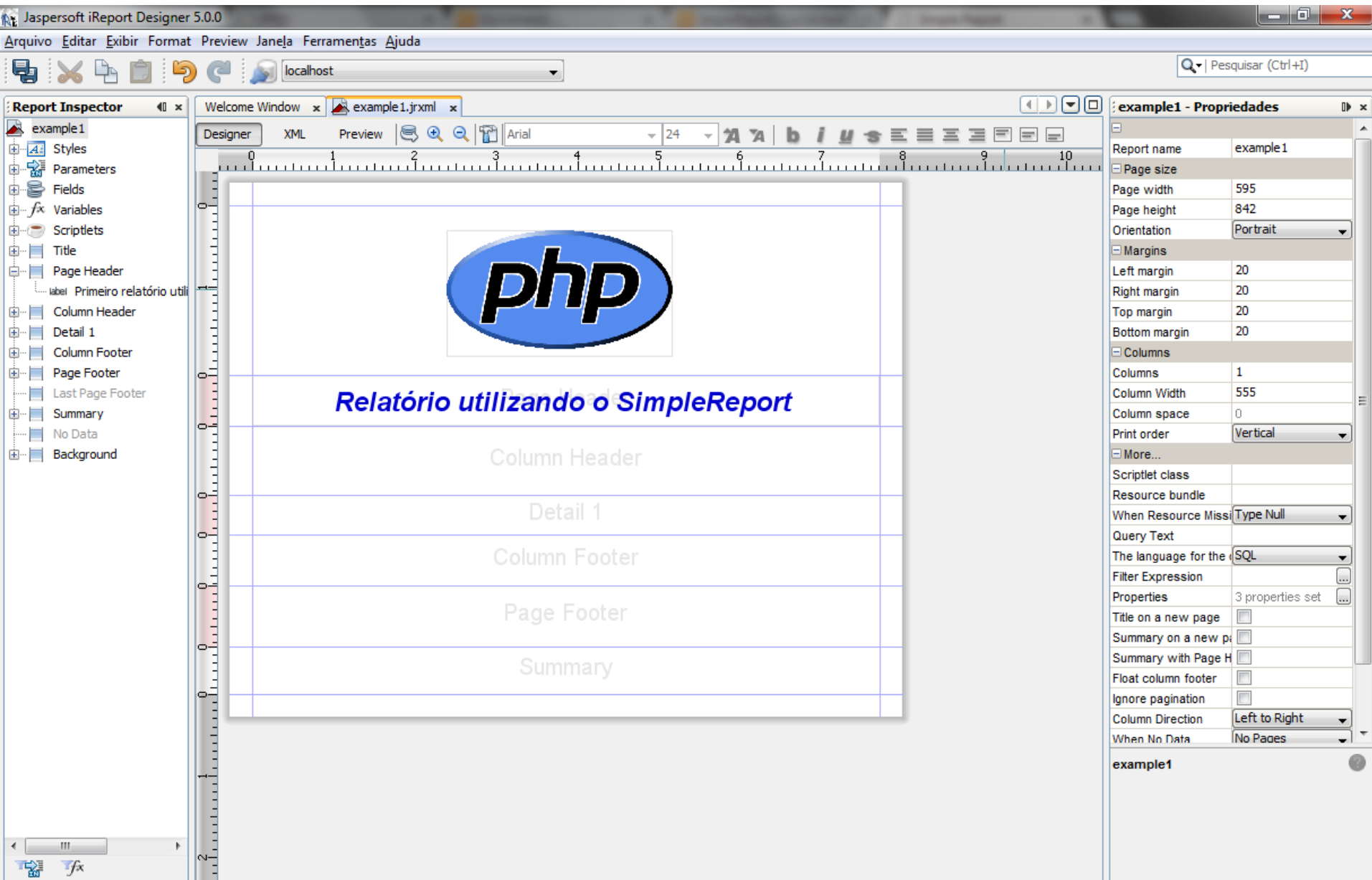
Como usar JasperReport em PHP



Solução proposta

- Criar relatórios em PDF com base nos layouts modelados no iReport.

iReport



.jrxml

```
<?xml version="1.0" encoding="UTF-8"?>
<jasperReport xmlns="http://jasperreports.sourceforge.net/jasperreports" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  <property name="ireport.zoom" value="1.0"/>
  <property name="ireport.x" value="0"/>
  <property name="ireport.y" value="0"/>
  <background>
    <band splitType="Stretch"/>
  </background>
  <title>
    <band height="150" splitType="Stretch">
      <image>
        <reportElement uuid="063c6808-f773-4c5c-827f-04e34418dd45" x="172" y="22" width="200" height="112"/>
        <imageExpression><![CDATA["C:\\www\\SimpleReport\\docs\\imagens\\php-logo.png"]]></imageExpression>
      </image>
    </band>
  </title>
  <pageHeader>
    <band height="45" splitType="Stretch">
      <staticText>
        <reportElement uuid="473efa40-cd75-4426-8e89-122c77c98a68" mode="Transparent" x="0" y="0" width="555" height="45"/>
        <textElement textAlignment="Center" verticalAlignment="Middle">
          <font fontName="Arial" size="24" isBold="true" isItalic="true"/>
        </textElement>
        <text><![CDATA[Relatório utilizando o SimpleReport]]></text>
      </staticText>
    </band>
  </pageHeader>
  <columnHeader>
    <band height="61" splitType="Stretch"/>
  </columnHeader>
  <detail>
    <band height="35" splitType="Stretch"/>
  </detail>
```

SimpleReport

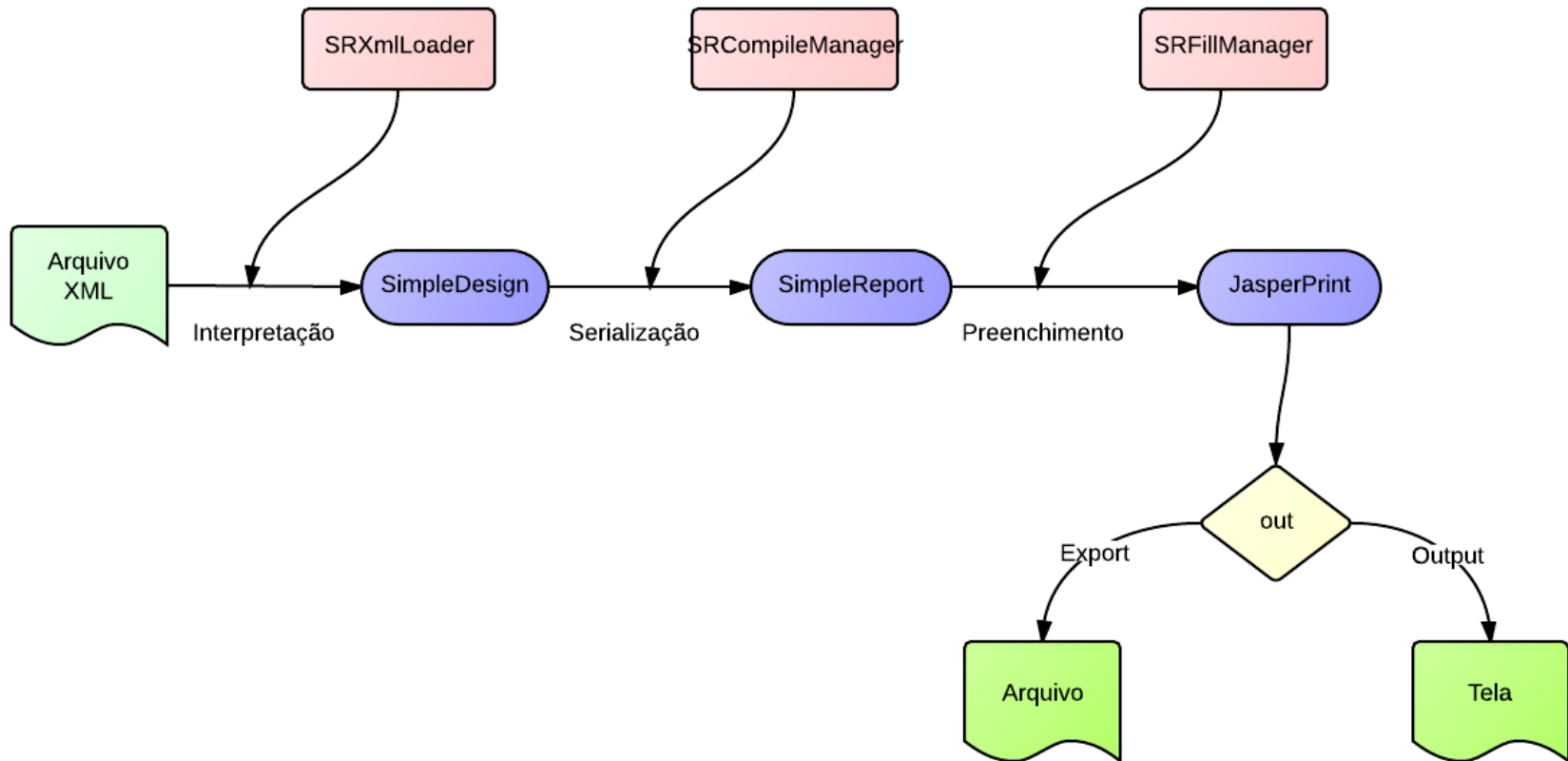
- Código aberto
 - LGPL
- Princípios e padrões
 - Convention Over Configuration
 - Iterator
 - Static Method Factory
- Dependência da biblioteca FPDF
- Não depende do XML (.jrxml)
- Pré compilação (cache)

SimpleReport

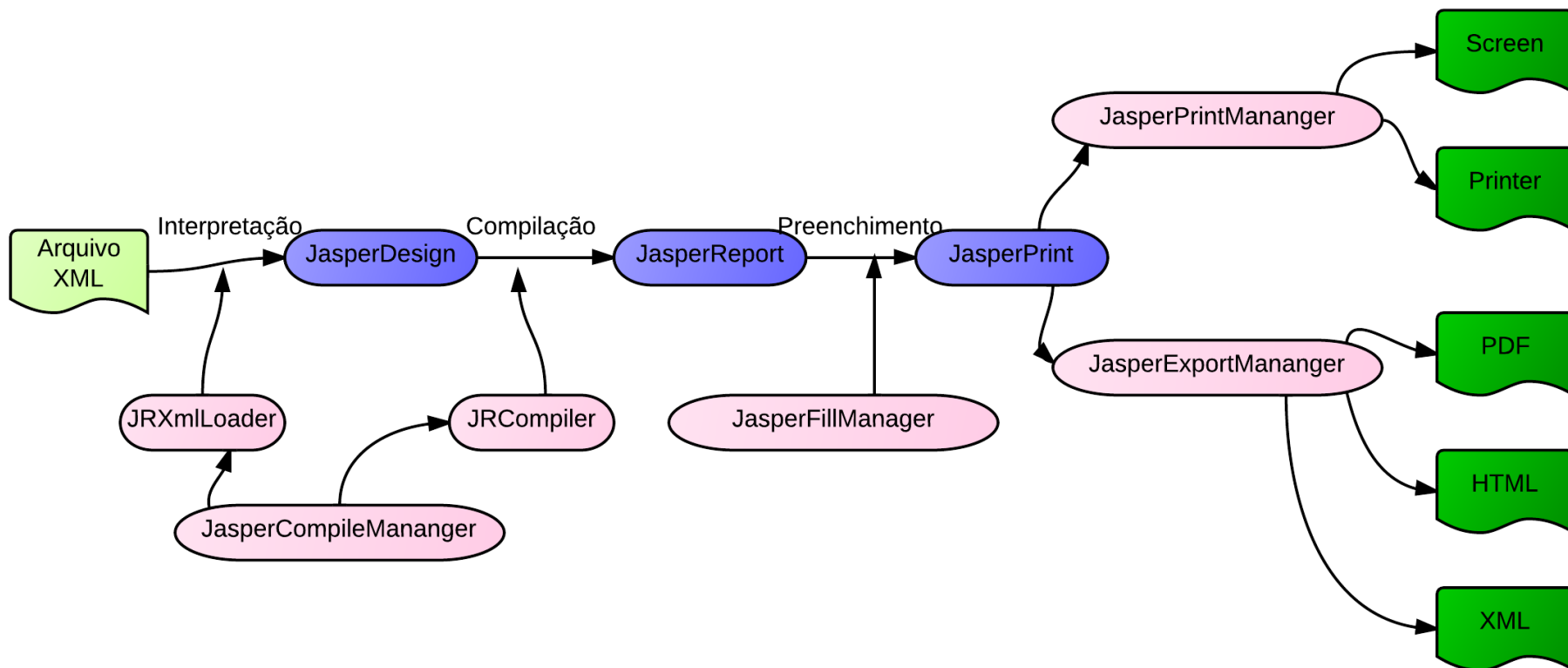
O SimpleReport é composto por três classes principais, são elas;

- **SimpleDesing:** Essa classe é a representação do modelo do relatório, nela serão definidas todas as propriedades, bandas, elementos e etc.
- **SimpleReport:** Essa classe representa o SimpleDesign serializado, apenas com as propriedades definidas e já interpretadas.
- **JasperPrint:** Essa classe é responsável por gerar o relatório, ela precisa de uma instância da SimpleReport e/ou uma fonte de dados. Ela possui dois métodos para gerar o relatório: o *outPut* que mostra o relatório na tela e o *export* que faz um download do arquivo.

Estrutura do SimpleReport



Estrutura JasperReport



Utilizando jrxml

```
if(file_exists('file.sr')){  
    $report =  
        SRInstanceManager::getInstance(file_get_contents('file.sr'));  
}else{  
    $load = new SRXmlLoader();  
    $design = $load->load('file.jrxml');  
    $report = SRCompileManager::compile($design, 'file');  
}
```

```
$link_identifier = mysql_connect('localhost', 'root', 'root');  
$db_selected = mysql_select_db('tads', $link_identifier);  
$result = mysql_query('select * from alunos', $link_identifier);
```

```
$dados = SRDataSource::getInstance('MySQL', $result);
```

```
$fill = new SRFillManager();  
$print = $fill->fillReport($report, $dados);
```

```
$print->outPut();
```

Mais simples utilizando Fábricas

```
//Sem fonte de dados
```

```
Report::from('file.jrxml')->outPut();
```

```
//Com fonte de dados
```

```
$link_identifier = mysql_connect('localhost', 'root', 'root');
```

```
$db_selected = mysql_select_db('tads', $link_identifier);
```

```
$result = mysql_query('select * from alunos', $link_identifier);
```

```
Report::from('file.jrxml',
```

```
    SRDataSource::getInstance('MySQL', $result))->outPut();
```


Exemplo sem jrxml

```
$sd = new SimpleDesign();
$sd->name = 'Report';

$sd->width = 595;
$sd->height = 842;
$sd->topMargin = 20;
$sd->rightMargin = 20;
$sd->leftMargin = 20;
$sd->bottomMargin = 20;

$bandTitle = new SRBand();
$bandTitle->height = 84;

$staticText1 = new StaticText();
$staticText1->text = "Texto";
$staticText1->x = 0;
$staticText1->y = 31;
$staticText1->width = 555;
$staticText1->height = 38;
$bandTitle->addElement($staticText1);

$sd->bandTitle = $bandTitle;

$report = SRCompileManager::compile($sd,
    'Report');

$fill = new SRFillManager();
$print = $fill->fillReport($report);
$print->output();
```

Demonstração

Implementações pendentes

Problemas estruturais encontrado durante a implementação do projeto.

- Colocar um elemento dentro do outro (Frame, Sub-relatórios)
 - *Possível solução: utilizar o padrão de projeto Decorator*
- Obter a própria query interna do iReport
 - *Possível solução: utilizar o framework PDO (PHP Data Object)*
- Utilizar fontes de dados como XML e CSV
 - *Esse problema ainda carece de solução*

Implementações futuras

- Desenvolver uma ferramenta que possibilite modelar o relatório.

Vantagens e desvantagens

- **Vantagens**

- Desenvolver relatório em PDF com iReport.
- Rodar nativamente em PHP.
- Código aberto.
- Enviar parâmetros para o relatório facilmente.

- **Desvantagens**

- Não suporta todas as funcionalidades do JasperReport
- Não interpreta código Java incorporado dentro do iReport

Conclusão

Percebemos que o SimpleReport é uma ideia muito proeminente e funciona razoavelmente bem, pois existe essa brecha no desenvolvimento visual de relatórios PDF para a linguagem PHP.

Serviu como aprendizado principal a importância de criar um bom projeto antes de começar a implementação, pois durante o processo de implementação foram encontrados alguns “erros” que poderiam ser evitados.

Perguntas?