A black and white logo

Description automatically generatedA black text on a white background

Description automatically generated

Relatório do Trabalho de Projeto Integrado 1

Relatório elaborado por: Carlos Silva Nº21583

Afonso Gomes nº25076

João Louro Nº25072

2022/23



# **Agradecimentos**

Gostaríamos de estender um momento para distribuir os nossos sinceros agradecimentos a todos envolvidos na assistência da elaboração do projeto, sejam eles professores ou colegas. Qualquer ajuda para a deliberação de dúvidas presentes é, foi e sempre será apreciada.

Adicionalmente, gostaríamos de agradecer ao Professor António Manso, por se encontrar confortável o suficiente para confiar os alunos com ficheiros pertinentes às informações pessoais dos utilizadores do programa a que os logs corresponde.

Ao professor Paulo Alexandre Santos, por se disponibilizar para o esclarecimento de dúvidas relativas a problemas encontrados no desenvolvimento ao longo da elaboração do projeto.

E à Professora Cristina Costa, na assistência da estruturação e redação do relatório, fornecendo sabedoria para a otimização das nossas apresentações e PowerPoint que as acompanham.

# Resumo

**Índice**

[**1-Introdução** 8](#_Toc139918396)

[**2-Algoritmi- Explicação Breve** 9](#_Toc139918397)

[**3-Desenvolvimento do Projeto** 10](#_Toc139918398)

[3.1-Filtragem de dados 10](#_Toc139918399)

[3.2- Processamento dos Dados 12](#_Toc139918400)

[3.3- Front-End/Demonstração gráfica dos dados. 15](#_Toc139918401)

[**4. Conclusão** 23](#_Toc139918402)

[Ferramentas Utilizadas 24](#_Toc139918403)

**Índice de Imagens**

[Imagem 1- Código de Filtragem universal. 10](#_Toc139918287)

[Imagem 2- Código de Extração de Horas 10](#_Toc139918288)

[Imagem 3- filtrohour.csv 11](#_Toc139918289)

[Imagem 4-- Contagem dos registos separados por dia. 12](#_Toc139918290)

[Imagem 5- Código para a separação de semanas. 12](#_Toc139918291)

[Imagem 6-- Código para a inserção dos dados processados. 13](#_Toc139918292)

[Imagem 7- Código para a transmissão dos dados para reprodução gráfica. 13](#_Toc139918293)

[Imagem 8- Código para Horas por Dias 14](#_Toc139918294)

[Imagem 9-- Código para Horas por Estudante 14](#_Toc139918295)

[Imagem 10- Código permitindo a demonstração de datas sem registos 15](#_Toc139918296)

# **1-Introdução**

No âmbito da disciplina de Projeto Integrado 1 foi-nos proposto o seguinte trabalho:

- Recolher dados do ficheiro “logIPRP.csv”;

- Filtrar informação através do comando “SEND\_ALGORITHM” presente no ficheiro, que permite saber quando um aluno submeteu um trabalho para avaliação na aplicação ALGORITHMI;

- Elaborar gráficos, com os dados recolhidos das avaliações diárias, semanais e mensais.

Importa clarificar o que é a aplicação Algorithmi. A aplicação Algorithmi ajuda o aluno a programar e facilita a aprendizagem das várias linguagens. O software permite a programação por fluxogramas e em modo texto, é uma ferramenta intuitiva que se foca na aprendizagem de algoritmos para resolver os problemas propostos.

Por outro lado, é uma ferramenta que permite avaliar o aluno pelo trabalho feito na aplicação. O nosso trabalho foca-se precisamente nos registos feitos quando um aluno ao usar a aplicação, envia o seu trabalho (“SEND\_ALGORITHM”) para avaliação.

Começámos por recolher a informação e filtrámo-la com os ficheiros “readdataX.php”. Estes ficheiros armazenam a informação em documentos “csv”. Através de vários ficheiros (“getdayssend.php”, “getweekssend.php”, “getmonthssend.php”, etc.), convertemos os dados em json e com recurso do ficheiro “final” “sendchart.html” construímos a página pretendida, página essa que apresenta os gráficos diários, semanais e mensais.

# **2-Algoritmi- Explicação Breve**

Este projeto foi estabelecido com ideia de manipular ficheiros logs. Para alcançar este propósito, foi-nos entregue logs relacionados à utilização do programa Algoritmi.  
  
 Algoritmi, programa desenvolvido pelo Professor António Manso com o objetivo de servir como suplemento curricular para a disciplina Programação, disponibiliza vários exercícios que utilizadores, após conclusão, necessitam de ser entregues e podem preliminarmente enviar para confirmar o seu progresso na descoberta da solução correta. Está presente outras funcionalidades, mas foi se destacadas as funcionalidades relevantes para este projeto.

É este envio preliminar em que este projeto se foca na sua direção. Apurando-os de entre os registos de outros comandos e processos que presidem no ficheiro log e extraindo deles as informações requeridas.

# **3-Desenvolvimento do Projeto**

Capítulo dedicado à descrição extensiva da solução descoberta para a elaboração do projeto. Divido por ficheiros e processos para a divulgação visual como secções.

## 3.1-Filtragem de dados

Conforme o enunciado ditou, era necessário a filtração de dados para a eliminação de dados redundantes. Após suficiente deliberação, foi se determinado um conjunto de código simples e adaptável para o que era requerido para a demonstração apropriada dos dados num gráfico.

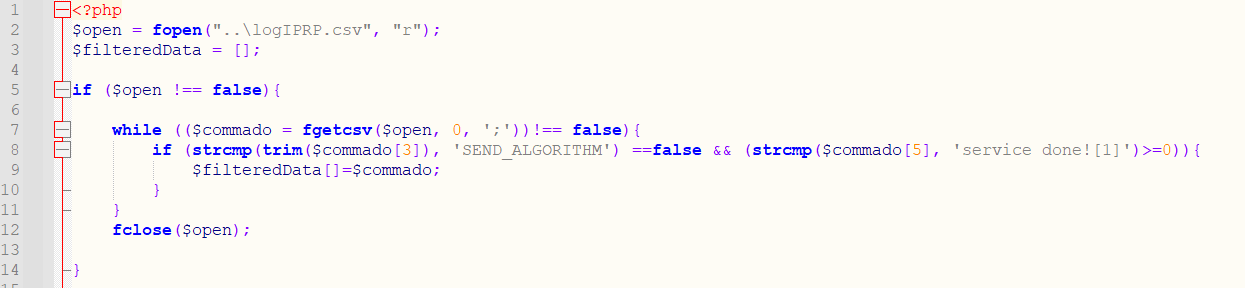


Imagem 1- Código de Filtragem universal.

Encontra-se dentro deste código o processo para a submeter sobre o ficheiro CSV “logIPRP” um filtro para excluir todas as linhas de dados que não se encaixam sobre o critério posto dentro do while, sendo este que o registo deve ser do comando “SEND\_ALGORITHM” e que este registo origine da confirmação da conclusão do serviço, escrevendo todos os registos para um array “$filteredData”. Em seguida, o array é passado por um ciclo for que seleciona exclusivamente os dados pertinentes à data e ao email to utilizador.

A screen shot of a computer code

Description automatically generated

Imagem 2- Código de Extração de Horas

Evidencia-se que estes pedaços de código são incrivelmente adaptáveis, levando à criação de múltiplos ficheiros em que a única diferença, fora do nome, é que dado este se dedica à extração. Dentro de si próprio, existe a criação e/ou abertura do ficheiro que será utilizado na construção gráfica, depois dentro do ciclo for, através da função substr que extrai partes de uma string para a formulação de uma nova, é se extraído a informação necessária da coluna que é chamada, no caso da Imagem 2 a coluna é a coluna sobre qual os registos das datas se reside e é escrita após o ficheiro CSV “filtrohour”.

A screenshot of a computer

Description automatically generated

Imagem 3- filtrohour.csv

*Tarefa Realizada por Carlos Silva nº21583*

## 3.2- Processamento dos Dados

Após a filtração e extração dos dados, é necessitado o processamento dos registos. Para alcançar esse objetivo foi se determinado que o seguinte código alcançava o resultado que era necessitado. A screenshot of a computer code

Description automatically generated

Imagem 4-- Contagem dos registos separados por dia.

Na imagem acima divulga o código elaborado para a contagem de registos pertinentes a cada data, iterando sobre uma lista que possui as todas. Se a data não se encontra dentro da lista cria-se uma contagem e se existe continua a contar para o dia relevante. Graças à simplicidade do código evidenciou-se que adaptar o código para trabalhar sobre outros critérios, como semana ou meses, seria possível e fácil.

*Código Elaborado por: Afonso Gomes nº25076*

No entanto, encontrou-se algumas dificuldades com a elaboração do código para o processamento dos dados perante as semanas do ano. Mas, com tempo, um dos membros do grupo chegou à seguinte solução. A screen shot of a computer code

Description automatically generated

Imagem 5- Código para a separação de semanas.

Neste código, o campo extraído é convertido para uma variável do tipo datetime, de seguida, através das propriedades da função date() é extraído o numero da semana perante o ano relevante. Após a obter o número de semana, este é armazenado dentro da variável $key para a eventual demonstração gráfica, e é iterado a frequência de registos que correspondem à semana utilizando o código que se evidencia na Imagem 4.

*Código Elaborado por: João Louro nº25072*

A seguir do processamento inicial, dentro duma nova classe criada antes da execução do código é inicializado duas variáveis que correspondem ao título do gráfico e à as etiquetas e os valores correspondentes inseridos através de um array\_push().

A screen shot of a computer code

Description automatically generated

Imagem 6-- Código para a inserção dos dados processados.

Com os dados processados sendo depois inseridos dentro do array() para a sua visualização nos grafos correspondentes, através do array\_push() e de um ciclo for, no final transmitidos através de um comando para a conversão dos dados para formato json.

A screenshot of a computer code

Description automatically generated

Imagem 7- Código para a transmissão dos dados para reprodução gráfica.

Não só nos limitando aos resultados requeridos pelos Professores, decidimos alcançar a capacidade de demonstrar estes dados filtrando-os entre sim próprios e até especificando o utilizador. Por exemplo: dentro de um dia divulgar a frequência de registos por hora. Em total encontrou-se os seguintes dois pares elaborado (código anteriormente divulgado não demonstrado):

- Horas por Dias

A screenshot of a computer code

Description automatically generated

Imagem 8- Código para Horas por Dias

-Horas por Estudante

A screenshot of a computer code

Description automatically generated

Imagem 9-- Código para Horas por Estudante

*Código Elaborado por: Afonso Gomes nº25076*

## 3.3- Front-End/Demonstração gráfica dos dados.

Com a filtragem e processamento dos dados concluída, resta o front-end para a divulgação gráfica cuja elaboração foi resolvida através do uso do API Google Charts, disponível para o publico. Embora, a solução foi descoberta facilmente, não significa que obstáculos não se foram identificados, visto o que o nosso grupo desejava alcançar mais do que uma simples exibição dos dados processados. Com o problema mais prominente sendo a falta de dias em que não se evidenciam registos, porém com tempo foi-se realizada uma resolução.

A screenshot of a computer code

Description automatically generated

Imagem 10- Código permitindo a demonstração de datas sem registos

O código permite a demonstração de datas sem registo através da criação de um intervalo formato datetime() para após declarar que os dias adicionados possuem nenhum registo associado.

*Código Elaborado por: Afonso Gomes nº25076*

Com o obstáculo resolvido, a construção gráfica dos dados processados pode ser corretamente demonstrada com a utilização do código que segue na conclusão deste paragrafo.

|  |
| --- |
| <html>  <head>  <style>  h1{font: bold 30px Verdana; color: #030374}  input{font: 14px Arial;}  p{text-align: center; font: 12px;}  .data{border: none; display: inline-block; font-size: 19px; border-radius: 4px; background-color: #8093e0; color:#ffffff;}  .filter{margin: 3px;}  #scrollBottom{position: fixed; right: 20px;}  #change-label1, #change-label2{font: 16px Arial;}  #change-chart5, #change-Chart6{ font: 14px Arial;}  </style>  <!--Load the AJAX API-->  <script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>  <script type="text/javascript">  google.charts.load('current', {packages: ['corechart', 'bar']});  var chartData;  xmlhttp = new XMLHttpRequest();  xmlhttp.onreadystatechange = function() {  if (this.readyState == 4 && this.status == 200) {  chartData = JSON.parse(this.responseText);  // Set a callback to run when the Google Visualization API is loaded.  google.charts.setOnLoadCallback(function(){ drawBarColors(chartData) });;  }  };  xmlhttp.open("GET", "getDaysSend.php", true);  xmlhttp.send();  function drawBarColors(chartData) {  var data = google.visualization.arrayToDataTable(chartData.data);  // variaveis referentes aos elementos do html  var button1 = document.getElementById('change-chart1');  var button2 = document.getElementById('change-chart2');  var button3 = document.getElementById('change-chart3');  var button4 = document.getElementById('change-chart4');  var label1 = document.getElementById('change-label1');  var dateInput = document.getElementById('date-input');  var button5 = document.getElementById('change-chart5');  var label2 = document.getElementById('change-label2');  var userInput = document.getElementById('user-input');  var button6 = document.getElementById('change-chart6');  var chartDiv = document.getElementById('chart\_div');  // caracteristicas do grafico de envios por dia  var daysOptions = {  title: chartData.title,  chartArea: {width: '50%', height: '90%'},  colors: ['#8093e0'],  hAxis: {  title: 'Sendings',  minValue: 0  },  vAxis: {  title: 'Days'  }  };  // caracteristicas do grafico de envios por semana  var weeksOptions = {  title: chartData.title,  chartArea: {width: '50%', height: '90%'},  colors: ['#8093e0'],  hAxis: {  title: 'Sendings',  minValue: 0  },  vAxis: {  title: 'Weeks'  }  };  // caracteristicas do grafico de envios por mês  var monthsOptions = {  title: chartData.title,  chartArea: {width: '50%'},  colors: ['#8093e0'],  hAxis: {  title: 'Sendings',  minValue: 0  },  vAxis: {  title: 'Months'  }  };  // caracteristicas do grafico da moda de envios por hora  var hoursOptions = {  title: chartData.title,  chartArea: {width: '50%', height: '90%'},  colors: ['#8093e0'],  hAxis: {  title: 'Sendings',  minValue: 0  },  vAxis: {  title: 'Hours'  }  };  // caracteristicas do grafico dos envios por hora de um dia selecionado  var hoursDayOptions = {  title: chartData.title,  chartArea: {width: '50%', height: '90%'},  colors: ['#8093e0'],  hAxis: {  title: 'Sendings',  minValue: 0  },  vAxis: {  title: 'Hours'  }  };  // caracteristicas do grafico de envios por dia de um aluno selecionado  var daysStudentOptions = {  title: chartData.title,  chartArea: {width: '50%', height: '90%'},  colors: ['#8093e0'],  hAxis: {  title: 'Sendings',  minValue: 0  },  vAxis: {  title: 'Hours'  }  };    function loadDataAndDrawChart(url, options, button) {  var xmlhttp = new XMLHttpRequest();  xmlhttp.onreadystatechange = function() {  if (this.readyState == 4 && this.status == 200) {  var newData = JSON.parse(this.responseText);  data = google.visualization.arrayToDataTable(newData.data);  // Atualiza o título do gráfico com o novo título dos dados  options.title = newData.title;  var chart = new google.visualization.BarChart(chartDiv);  chart.draw(data, google.charts.Bar.convertOptions(options));  button.disabled = false;  }  };  xmlhttp.open("GET", url, true);  xmlhttp.send();  }  // ao clica neste butão, ele irá mostrar o grafico de envios por dia  function drawDaysChart() {  button1.disabled = true;  loadDataAndDrawChart("getDaysSend.php", daysOptions, button1);  button2.onclick = drawWeeksChart;  button3.onclick = drawMonthsChart;  button4.onclick = drawHoursChart;  button5.onclick = drawHDChart;  button6.onclick = drawDSChart;  chartDiv.style.height = '3500px';    button1.style.display = "none";  button2.style.display = "inline-block";  button3.style.display = "inline-block";  button4.style.display = "inline-block";    label1.style.display = "inline-block";  dateInput.style.display = "inline-block";  button5.style.display = "inline-block";  label2.style.display = "inline-block";  userInput.style.display = "inline-block";  button6.style.display = "inline-block";  }  // ao clica neste butão, ele irá mostrar o grafico de envios por semana  function drawWeeksChart() {  button2.disabled = true;  loadDataAndDrawChart("getWeeksSend.php", weeksOptions, button2);  button1.onclick = drawDaysChart;  button3.onclick = drawMonthsChart;  button4.onclick = drawHoursChart;  chartDiv.style.height = '1000px';    button1.style.display = "inline-block";  button2.style.display = "none";  button3.style.display = "inline-block";  button4.style.display = "inline-block";  label1.style.display = "none";  dateInput.style.display = "none";  button5.style.display = "none";  label2.style.display = "none";  userInput.style.display = "none";  button6.style.display = "none";  }  // ao clica neste butão, ele irá mostrar o grafico de envios por mês  function drawMonthsChart() {  button3.disabled = true;  loadDataAndDrawChart("getMonthsSend.php", monthsOptions, button3);  button1.onclick = drawDaysChart;  button2.onclick = drawWeeksChart;  button4.onclick = drawHoursChart;  chartDiv.style.height = '500px';    button1.style.display = "inline-block";  button2.style.display = "inline-block";  button3.style.display = "none";  button4.style.display = "inline-block";  label1.style.display = "none";  dateInput.style.display = "none";  button5.style.display = "none";  label2.style.display = "none";  userInput.style.display = "none";  button6.style.display = "none";  }  // ao clica neste butão, ele irá mostrar o grafico da moda de envios por hora  function drawHoursChart() {  button4.disabled = true;  loadDataAndDrawChart("getHoursSend.php", hoursOptions, button4);  button1.onclick = drawDaysChart;  button2.onclick = drawWeeksChart;  button3.onclick = drawMonthsChart;  chartDiv.style.height = '1000px';  button1.style.display = "inline-block";  button2.style.display = "inline-block";  button3.style.display = "inline-block";  button4.style.display = "none";  label1.style.display = "none";  dateInput.style.display = "none";  button5.style.display = "none";  label2.style.display = "none";  userInput.style.display = "none";  button6.style.display = "none";  }  // inicializa o grafico de envios por hora de um dia selecionado  function drawHDChart() {  // obtem o valor do 'date-input'  var selectedDate = dateInput.value;  // cria a url dos dados do respectivo dia  var urlDay = "getHoursDay.php?day=" + selectedDate;    loadDataAndDrawChart(urlDay, hoursDayOptions, button5);  chartDiv.style.height = '1000px';  button1.onclick = drawDaysChart;  button1.style.display = "inline-block";  label2.style.display = "none";  userInput.style.display = "none";  button6.style.display = "none";  }  // inicializa o grafico de envios por dia de um estudante selecionado  function drawDSChart() {  // obtem o valor do 'user-input'  var selectedUser = userInput.value;  // cria a url dos dados do respectivo dia  var urlStudent = "getDaysStudent.php?student=" + selectedUser;    loadDataAndDrawChart(urlStudent, daysStudentOptions, button6);  chartDiv.style.height = '3500px';  button1.onclick = drawDaysChart;  button1.style.display = "inline-block";  label1.style.display = "none";  dateInput.style.display = "none";  button5.style.display = "none";  }  // inicializa o grafico de envios por dia  drawDaysChart();  }  </script>  </head>  <body>  <div id="top"></div>  <h1 >SEND ALGORITHM</h1>  <button class="data" id="change-chart1">Change to Days</button>  <button class="data" id="change-chart2">Change to Weeks</button>  <button class="data" id="change-chart3">Change to Months</button>  <button class="data" id="change-chart4">Change to Hours</button>  <button id="scrollBottom"><a href="#bottom" id="btn-scroll">Go to the Bottom</a></button>  <br>  <label class="filter" id="change-label1" for="date">Select a Day:</label>  <input class="filter" type="date" id="date-input">  <button class="filter" id="change-chart5">Create Chart</button>  <br>  <label class="filter" id="change-label2" for="date">Select a Student:</label>  <input class="filter" type="text" size="12" maxlength="10" id="user-input" value="aluno">  <button class="filter" id="change-chart6">Create Chart</button>  <br><br>  <!--Div that will hold the pie chart-->  <div id="chart\_div"></div>  <button><a href="#top" id="btn-scroll">Go to the Top</a></button>  <hr>  <!-- Marcador para o final da página -->  <div id="bottom">  <p>Desenvenvoldedor do Algorithmi: Antonio Manso</p>  <p>Contactos: manso@ipt.pt</p>  <p>Instituto: Instituto Politecnico de Tomar</p>  <p>Site do Algorithmi: algorithmi.ipt.pt</p>  </div>  </body>  </html> |

Declarando como variável os dados pertinentes aos gráficos, este código faz uso de funções para:

* Desenhar e colorar as barras dos gráficos [Função drawBarColors]
* Carregar dados correspondentes ao gráfico, estes sendo o titulo, etiquetas e frequência de registos.
* Desenhar o gráfico correspondente e disponibilizar os botões para trocar de gráfico [drawDaysChart, drawWeeksChart, drawMonthsChart, drawDSChart, drawHoursChart, drawHDChart]

*Código reapropriado por: Afonso Gomes nº25076*

# **4. Conclusão**

No contexto deste projeto descobrimos que embora a resolução do trabalho se evidenciou relativamente fácil de concluir, a nossa coordenação encontrou-se como um pouco abaixo de medíocre, identificando-se problemas a distribuir trabalho de uma maneira balançada para que todos os membros tenham executado uma parte igual.

No entanto, também evidenciou, que embora tenha existido dificuldades em termos de organização, todos os membros do grupo se encontram como programadores capazes, adquirindo novos conhecimentos e reforçando um sentido de proficiência nos conhecimentos adquiridos antes da elaboração deste trabalho.

# Ferramentas Utilizadas

No contexto deste projeto encontra-se que as seguintes ferramentas foram se utilizadas:

* Notepad++ e Visual Studio Code, editores de código escolhidos pela sua acessibilidade e compatibilidade com a linguagem PHP e HTML. Funcionalidades que adicionalmente influenciaram a escolha foram a deteção preliminar de erros (Visual Studio Code) e direto acesso através do XAMPP(Notepad++) como exemplos mais prominentes.
* XAMPP, pacote de software que disponibiliza a criação e manutenção de um servidor local com suporte para múltiplos serviços para a execução e testagem do código elaborado.
* ChatGPT e Google, ferramentas para esclarecimento de dúvidas e assistência para a resolução de obstáculos.
* Google Chart API, interface para a programação de aplicações gráficas baseadas em grafos.