

# Explicação da solução

Marco Antônio M. Roos

Após começar a ler os requisitos, a primeira solução que me veio a mente foi usar react, uma vez que seria simples fazer uma apresentação visual em jsx / html e acertar as proporções em css. Entretanto, ao me deparar com o requisito de finalizar o programa, percebi que não seria eficaz, uma vez que javascript não pode finalizar nem o navegador e nem fechar abas que não tenham sido abertas pelo software.

Os requisitos poderiam ser compridos facilmente com java, o qual já possuía algum conhecimento sobre, mas java por si só não possui interface gráfica, oque eu achei que seria interessante para apresentação dos resultados, então fui atrás de uma.

Entre Swing e JavaFX, acabei escolhendo JavaFX, o tempo era curto e eu precisava definir rapidamente, por isso, fiz rápida leitura em alguns fóruns na internet e comecei a estudar JavaFX para compreender seu funcionamento.

No primeiro dia (08/09) criei a base em JavaFX, com janelas e navegação, defini o layout e criei um cronograma baseado no que tinha aprendido.

08/09/2021	Estudo de JavaFX
09/09/2021	Implementação CSV
10/09/2021	Apresentação de dados
11/09/2021	Telas de Erros e Documentação
12/09/2021	Folga para possíveis problemas

Entretanto, no segundo dia (09/09), percebi que seria mais fácil analisar os dados resultantes, uma vez que houvesse uma interface gráfica, pois através do console do intelliJ há perda de dados para um volume tão grande quanto o arquivo csv, que possui mais de 50 mil linhas com diversas colunas, assim, decidi que seria melhor trabalhar por telas e usar o conteúdo de uma para dar base da construção da próxima. Nesse dia completei as telas de média de idade e de interações por ano :

## Tela Inicial



Para criar a tela inicial ( *home* no código ), foi necessária pouca lógica, utilizei para todas as telas a ferramenta *SceneBuilder*, que facilitou muito o trabalho, evitando ter que definir o layout em FXML. Assim, foi preciso apenas criar as telas, definir os controladores de cada uma, programar a navegação e testar.

Uma boa forma de entender o funcionamento da navegação é conferir o arquivo *HomeController.java* e *Main.java* onde é definido a criação e navegação entre telas. ( que basicamente consiste em criar uma nova tela e deletar a tela em que o usuário se encontrava ).

Assim que comecei a trabalhar com o arquivo CSV, importantando ele para a pasta *src/Document* do projeto, percebi que ele usava “;” ao invés de “,” para separar as colunas, assim, utilizei *BufferedStream* para ler o arquivo linha por linha, e separei os dados usando *.split*.

Eu percebi que uma vez que o arquivo CSV estaria sempre presente, a necessidade de salvar os dados em memória era um tanto quanto secundário entre as funcionalidades e deixei para implementar por último, uma vez que a apresentação dos dados seria o core do que seria apresentado ao “cliente” e assim, foquei em tornar o sistema um pouco mais rápido, carregando somente as colunas necessárias para cada tela, por exemplo, se uma tela exige a média de idade dos pacientes, a partir do input da cidade de residência, então eu não preciso de dados como hospital executante e solicitante, datas de entrada, saída e etc.

Assim, fiz um overload do construtor da classe para facilitar, caso fosse necessário buscar todas as colunas futuramente. ( um construtor não exige parametros, o outro exige uma lista de números que dizem respeito as colunas do CSV ).

Creio que seria mais seguro buscar as colunas iniciadas por determinados termos, uma vez que o usuário poderia alterar a ordem das colunas ou mesmo excluir colunas, por outro lado, dessa forma foi mais rápido e é suficientemente eficiente na tabela csv fixa apresentada.

A partir da próxima tela, temos um padrão para todas elas, que é filtrar as linhas que possuem o dado pesquisado na coluna relevante, por exemplo, se a linha 0, 1 e 2 possuem o dado “PORTO ALEGRE” na coluna residencia e as outras linhas não, eu crio uma lista com os números 0, 1 e 2. Salvando os endereços ao invés de ocupar um espaço maior em memória para filtrar os dados, assim eu defino quais as linhas relevantes, com pouca variação, em todas as telas.

Mas por que não apagar os dados irrelevantes para a pesquisa ao invés de salvar os endereços das linhas ? Simples, o usuário pode fazer mais de uma pesquisa, e as linhas relevantes podem mudar, nessa situação, os endereços antigos já foram descartados e novos são filtrados, oque não seria possível sem carregar novamente o arquivo csv e refazer o processo de filtragem. Assim, posso carregar as colunas relevantes apenas uma vez, e usa-las até que o usuário saia da janela atual.

## Tela de consulta de média de idade dos internados

DELL - Internações : Consultar média de idade dos pacientes

Consultar média de idade dos pacientes

Município

porto alegre

Buscar

Total de pacientes do município

20947

Média de idade dos pacientes do sexo masculino

48

Média de idade dos pacientes do sexo feminino

51


Média de idade de todos os pacientes

50

DELL

Exercício Técnico [ Marco Antônio Marcos Roos ]

Voltar



Após selecionar as linhas relevantes ( no total 20947 ), foi só uma questão de tirar a média, pra isso criei uma lista de idades para pessoas do sexo masculino, filtrando pela coluna do sexo, e a partir daí tirando a média. Criei uma nova função adaptando para o sexo feminino e também uma nova função para a média global.


## Tela de consulta de internações por ano


DELL - Internações : Consultar internações por ano

Consultar internações por ano

Município

2018	11936
2019	5798
2020	1502
2021	1671

 Exercício Técnico [ Marco Antônio Marcos Roos ]



Após selecionar as linhas relevantes, precisei apenas filtrar as linhas que possuíam o valor do ano solicitado. Como eram apenas 4 anos , usei um for, filtrando os primeiros 4 caracteres da data em formato string fornecido pelo csv, obtendo assim o ano manualmente, e seguido de um switch para somar os casos em seus respectivos anos.

A próxima tela foi completada no terceiro dia, pois enfrentei alguns erros relacionados ao sistema de “espelhamento”, foi quando percebi que telas diferentes não podem ter nomes de componentes visuais iguais. Isso ocasionou uma série de problemas que levaram algumas horas pra solucionar, por isso não consegui completar ela no segundo dia.

## Tela de consulta de dados de um hospital executor

DELL - Internações : Consultar hospitais

Consultar hospitais

Executante hospital sao lucas da pucrs

Buscar

ID Usuário	Idade	Município Paciente	Nome MunicípioSolicitante	Data hora autorização	Data internação	Data alta	Descrição CNES Ex...
3121e0969...	57	PORTO ALEGRE	PORTO ALEGRE	2018-05-02 00:00:00.0...	2018-05-02 00:00:00...	2018-05-03 00...	HOSPITAL SAO LUCA..
2719be008...	62	ALVORADA	PORTO ALEGRE	2019-01-24 00:00:00.0...	2019-01-24 00:00:00...	2019-02-01 00...	HOSPITAL SAO LUCA..
ec956bd84...	54	CAPAO DA CANOA	PORTO ALEGRE	2018-04-02 00:00:00.0...	2018-04-02 00:00:00...	2018-04-03 00...	HOSPITAL SAO LUCA..
ca1102c22...	67	PORTO ALEGRE	PORTO ALEGRE	2018-03-13 00:00:00.0...	2018-03-13 00:00:00...	2018-03-14 00...	HOSPITAL SAO LUCA..
6afb24cd6d...	77	ROSARIO DO SUL	PORTO ALEGRE	2018-05-01 00:00:00.0...	2018-05-01 00:00:00...	2018-05-15 00...	HOSPITAL SAO LUCA..
a7f0e97679...	51	TRIUNFO	PORTO ALEGRE	2018-04-02 00:00:00.0...	2018-04-02 00:00:00...	2018-04-03 00...	HOSPITAL SAO LUCA..
f4dc2ff71d9...	69	CACHOEIRINHA	PORTO ALEGRE	2018-04-02 00:00:00.0...	2018-04-02 00:00:00...	2018-04-03 00...	HOSPITAL SAO LUCA..
518d7d9d9...	36	PORTO ALEGRE	PORTO ALEGRE	2019-01-24 00:00:00.0...	2019-01-24 00:00:00...	2019-01-25 00...	HOSPITAL SAO LUCA..
17a7b0803...	47	PORTO ALEGRE	PORTO ALEGRE	2018-04-03 00:00:00.0...	2018-04-03 00:00:00...	2018-04-03 00...	HOSPITAL SAO LUCA..
73a1af3331...	58	PORTO ALEGRE	PORTO ALEGRE	2019-01-24 00:00:00.0...	2019-01-24 00:00:00...	2019-01-27 00...	HOSPITAL SAO LUCA..
5b940f2209...	17	CAMPOS DOS GOYTA...	PORTO ALEGRE	2018-03-12 00:00:00.0...	2018-03-12 00:00:00...	2018-04-12 00...	HOSPITAL SAO LUCA..
966539f254...	51	NOVO HAMBURGO	PORTO ALEGRE	2018-03-12 00:00:00.0...	2018-03-12 00:00:00...	2018-03-13 00...	HOSPITAL SAO LUCA..
65934aee7...	11	CACHOEIRINHA	PORTO ALEGRE	2018-03-12 00:00:00.0...	2018-03-12 00:00:00...	2018-03-13 00...	HOSPITAL SAO LUCA..
4e8625ffe5f...	53	PORTO ALEGRE	PORTO ALEGRE	2018-05-02 00:00:00.0...	2018-05-02 00:00:00...	2018-05-03 00...	HOSPITAL SAO LUCA..
ca1102c22...	67	PORTO ALEGRE	PORTO ALEGRE	2018-05-03 00:00:00.0...	2018-05-03 00:00:00...	2018-05-04 00...	HOSPITAL SAO LUCA..
7406bc4cff...	12	TAQUARA	PORTO ALEGRE	2018-04-02 00:00:00.0...	2018-04-02 00:00:00...	2018-04-02 00...	HOSPITAL SAO LUCA..
6ea2d2ca6...	47	NOVO HAMBURGO	NOVO HAMBURGO	2021-01-07 00:00:00.0...	2021-01-07 00:00:00...	2021-03-25 00...	HOSPITAL SAO LUCA..

DELL

Exercício Técnico [ Marco Antônio Marcos Roos ]

Voltar

Após selecionar as linhas relevantes, neste caso bastou apenas relacionar corretamente as colunas dos csv a *TableView* do *JavaFX*. Apesar de alguns problemas para chegar nisso, logo compreendi melhor o padrão de inicialização e espelhamento da ferramenta gráfica e foi possível sanar os erros.



## Tela de consulta de dados de um hospital solicitante

DELL - Internações : Calcular tempo de internação

Calcular tempo de internação

Solicitante

ID Usuário	Descrição CNESExecutante	Tempo que passou internado em dias
3121e09691f382cd3fdefa799139abc1	HOSPITAL SAO LUCAS DA PUCRS	1
2719be008ca1c1ebca9cdf5e8987f1cb	HOSPITAL SAO LUCAS DA PUCRS	8
ec956bd842027df0298a83acad58674f	HOSPITAL SAO LUCAS DA PUCRS	1
ca1102c22ed5240452efa3d4df01114e	HOSPITAL SAO LUCAS DA PUCRS	1
6afb24cd6d3c0237a8f1d887635267bb	HOSPITAL SAO LUCAS DA PUCRS	14
a7f0e97679888a2ff6aacb86fdc51625	HOSPITAL SAO LUCAS DA PUCRS	1
f4dc2ff71d9f79d2913e176c146990ed	HOSPITAL SAO LUCAS DA PUCRS	1
518d7d9d9edd132334d5e97d1f2f08be	HOSPITAL SAO LUCAS DA PUCRS	1
17a7b0803ece684b94cee71c8ae2dc92	HOSPITAL SAO LUCAS DA PUCRS	0
73a1af3331d9ccabae5c9bac6d35ac39	HOSPITAL SAO LUCAS DA PUCRS	3
5b940f2209bcbd71cbac8c55ade23baf	HOSPITAL SAO LUCAS DA PUCRS	31
966539f254dc9e82fac20e9167a918e	HOSPITAL SAO LUCAS DA PUCRS	1
65934aee714d0c341486a0089dd16d2f	HOSPITAL SAO LUCAS DA PUCRS	1
4e8625ffe5f7f15415409c3654827964	HOSPITAL SAO LUCAS DA PUCRS	1
ca1102c22ed5240452efa3d4df01114e	HOSPITAL SAO LUCAS DA PUCRS	1
7406bc4cff587ec2139873ed3d061147	HOSPITAL SAO LUCAS DA PUCRS	0
55af3906015689254f4ae6e85ab5a082	HOSPITAL SAO LUCAS DA PUCRS	0

 Exercício Técnico [ Marco Antônio Marcos Roos ]  

Após selecionar as linhas relevantes, além de relacionar os dados simples, foi necessário calcular o número de dias, baseado em datas, para isso, recebi os dados em formato string do csv, peguei as partes relevantes e transformei em data, depois disso usei uma função de calculo não inclusivo de dias entre datas. Uma vez que a função estava funcionando, bastou converter para string e relacionar no *TableView*.

Parando para analisar os dados obtidos através do programa, me deparei com uma situação estranha, o número de internações parecia maior no ano de 2018 do que 2020 e 2021, mas como isso é possível ?

Então fui até o LibreOffice Calc, e comecei a pesquisar, acontece que na tabela apresentada, dos mais de 50 mil casos, cerca de 29 mil são de 2018, mais da metade das internações. Oque por um lado é estranho, por outro foi um alívio, ao perceber

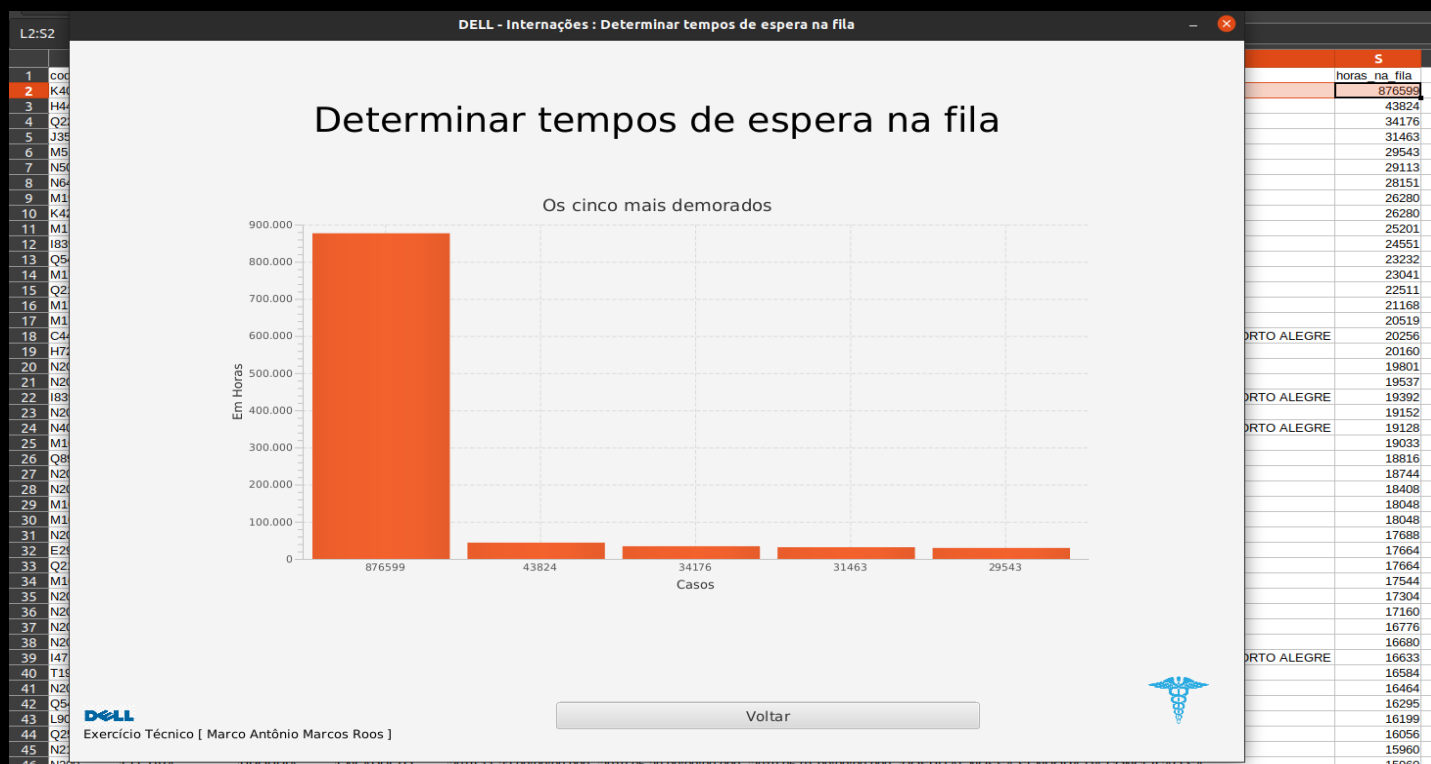
que os dados apresentados no programa são aparentemente proporcionais aos dados contidos na tabela.

Nesse ponto percebi que o csv possui erros de digitação, alguns dos tempos de espera são negativos, existem datas de alta que ocorrem antes da data de solicitação ou autorização.

Eu geralmente procuraria uma solução junto a um superior, mas nesse caso não temos nada além de instruções, por tanto decidi excluir os dados que estão necessariamente incorretos durante o carregamento do csv, quando eu for criar o sistema de save e load de dados atrelado ao iniciar do software, assim modificarei os dados em um único lugar.

Por hora foquei em completar a tela de tempo de espera, sem filtros, já que os dados serão filtrados uma única vez no processamento do csv para salvar os dados no futuro.

## Tela de tempo de espera na fila



A única tela que não exige um filtro primário, seja uma cidade ou data, apenas organiza em ordem de tempo de espera, para tal, foi apenas necessário passar os dados por uma função que transforma os dados em valores numéricos, os compara e executa um selection sort para organizar somente os 5 primeiros



valores do array em ordem decrescente, assim basta buscar os primeiros 5 valores para o demonstrativo, sem a necessidade de organizar todo o array.

A tabela organizada de forma decrescente no fundo, demonstra que o gráfico está correto, porém percebi que um dos dados mostra uma espera de mais de 100 anos, então decidi colocar mais um filtro de dados, em que esperas de mais de 100 mil horas ( pouco mais de 11 anos ) serão descartadas, a fim de preservar a confiabilidade dos dados.

Para finalizar o terceiro dia, criei uma tela para situações em que o programa não encontra nenhum resultado na pesquisa, visando ajudar o usuário a encontrar o que deseja.

### Tela de erro na consulta

Calcular tempo de internação

Solicitante 4fadf Buscar

ID Usuário

tempo que passou internado em dias

**OPS, Não pude encontrar o que você deseja**

Mas não desanime, para encontrar o que deseja você deve escrever o nome da referência exatamente como está na tabela csv, não se preocupe em colocar em caixa alta, pois o software faz isso por você, mas não use acentuação.

Fechar

No content in table

DELL Exercício Técnico [ Marco Antônio Marcos Roos ] Voltar

É uma tela simples, feita no SceneBuilder e apresentada quando nenhum resultado compatível é encontrado.

Terceiro dia ( 10/09 ), ainda tendo uma hora sobrando e tendo me programado para parte final ( save / load com filtros ) sábado de tarde ( 11/09 ), resolvi melhorar o layout das telas de média de idade e internações por ano.



No quarto dia ( 11/09 ) terminei o programa, criando o sistema de criação, escrita e leitura de um arquivo “.txt” com os dados do arquivo csv. O qual é salvo após a remoção de linhas que possuam alguma dessas características :

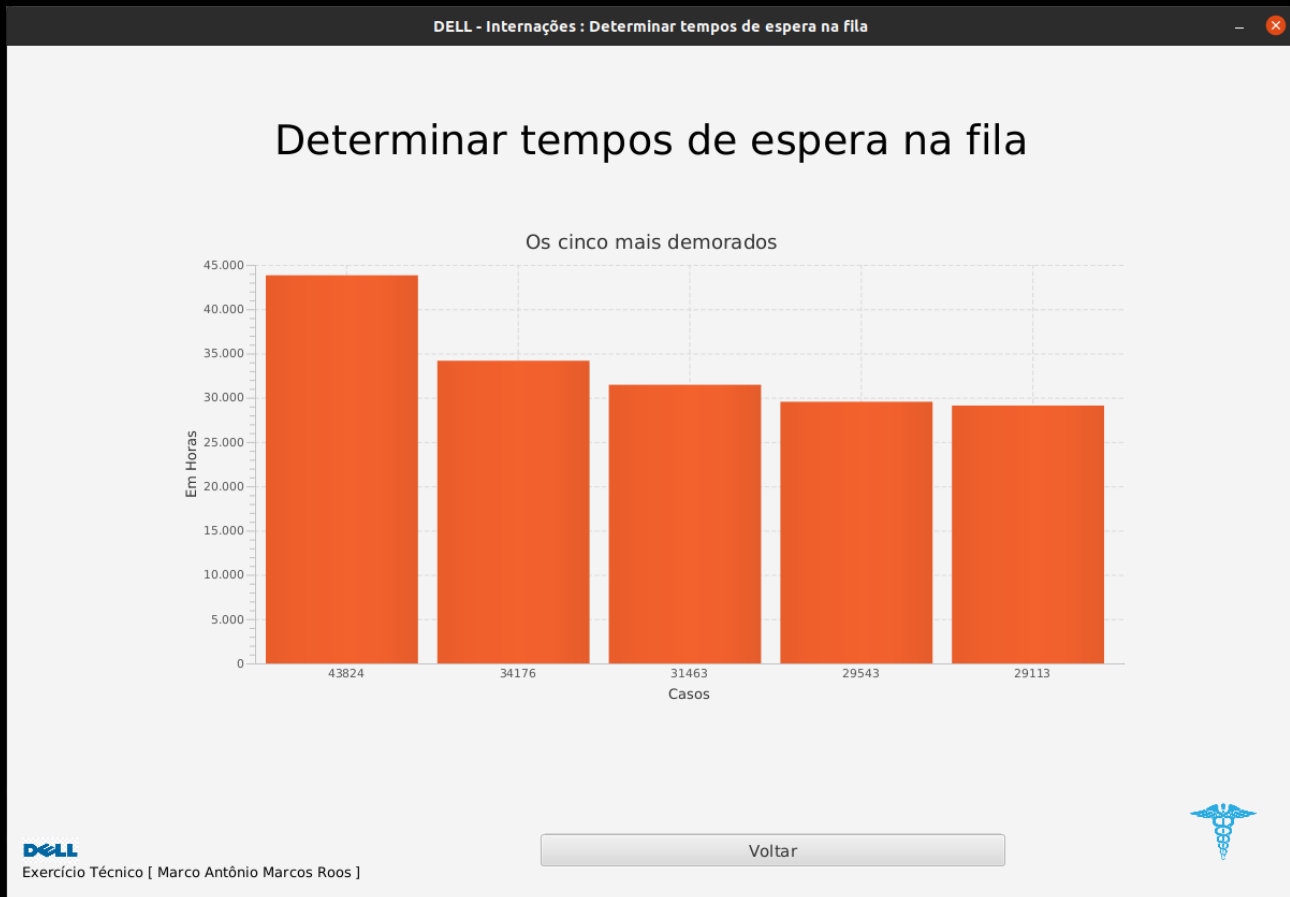
- Horas de espera negativas
- Horas de espera superior a 100'000 Horas ( Pouco maias de 11 anos )
- Autorizações que ocorreram antes da solicitação
- Internações que correram antes da autorização
- Altas que ocorreram antes da internação

Essas regras foram colocadas por conta própria para os fins do exame, em um cenário real, eu teria entrado em contato com um superior ( ou cliente ) para compreender melhor as necessidades e assim garantir que não estou apagando dados cruciais. A ideia do filtro é manter a confiabilidade dos dados, e não o contrário.

Utilizei a classe *PrintStream* para escrever as linhas e *BufferedReader* para ler as linhas do arquivo e trata-las.

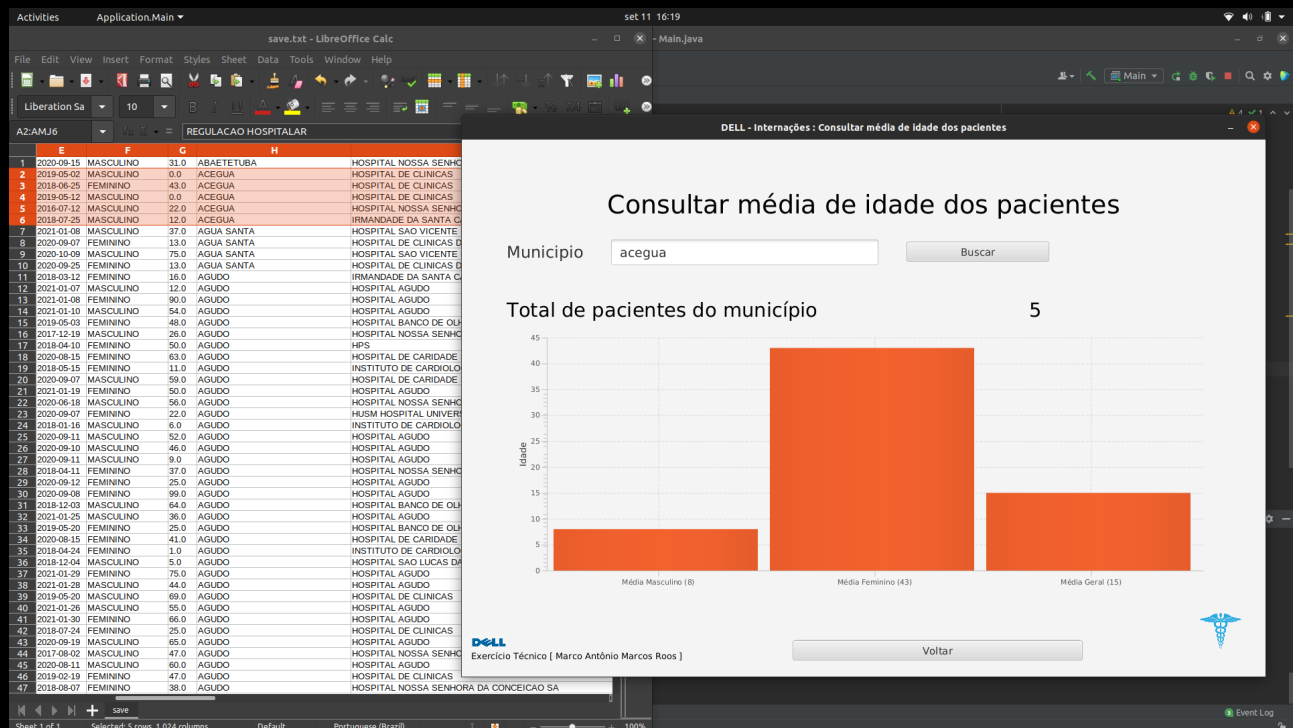
Foram necessários diversos loops para ler o csv, filtrar os dados conforme as requisições acima e escrever o arquivo. Para depois ser lido e dividido por outra classe, que busca apenas as colunas necessárias por cada tela.

A imagem a seguir demonstra o resultado da filtragem de linhas ao gravar os dados no arquivo “.txt” na tela de tempo de espera.



Não foram criados testes automatizados para este caso, pois a comparação das informações resultantes do software com resultados dos filtros do calc ( Alternativo do ubuntu ao excell ) demonstra perfeitamente se as informações estão corretas, além disso me pareceu mais confiavel do que escrever testes unitários específicos, já que estes, são pontuais, entretanto reconheço a necessidade de escreve-los durante a prdoução em ambiente real de desenvolvimento, já que eles vão muitas vezes guiar o desenvolvimento.

A imagem abaixo demonstra a situação descrita no parágrafo acima sobre testes.



Temos uma paciente do sexo feminino com 43 anos, frente a média de 43 anos apresentado no programa, temos também 4 pacientes do sexo masculino, com as idades de 0, 0, 22 e 12 anos. Presumo que os pacientes com 0 anos são recém nascidos, assim  $(22+12)/4 = 34/4 = 8.5$  anos, o qual o programa arredondou para 8.

Através desse método, pude conferir que para vários casos o programa encontra corretamente as médias, encontrei erros apenas em situações em que por exemplo, não há dados suficientes para criar uma média, como é o caso de abaetetuba.

Nada grave, mas o programa não mostrará o comparativo, não atualizando os dados da tela.

Assim decidi modificar o código, para que, a cada pesquisa, os elementos na tela sejam limpos antes de receberem dados novamente. De forma que quando não houver dados suficientes ou houverem erros na pesquisa, o programa não mais apresentará dados da pesquisa anterior e sim uma tela limpa e pronta para uma nova pesquisa.