



UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
CAMPUS PAU DOS FERROS
INTERDISCIPLINAR EM TECNOLOGIA DA INFORMAÇÃO

Algoritmos e Estrutura de Dados I
Laboratório de Algoritmos e Estrutura de Dados I

Prof. Me. Robson Locatelli Macedo

**DOCUMENTO DO PROJETO DO
SISTEMA DE APOIO AO ATENDIMENTO HOSPITALAR**

Fabício da Silva Ribeiro
Helton Wylammi Costa Pinto
Júlio César Lima Gama
Mateus Fernandes Costa

03 de Fevereiro de 2020
Pau dos Ferros - RN

Histórico de Revisões

Versão	Data	Descrição	Autor
1.0.0	31/02/2020	Elaboração da estrutura do documento e do Capítulo 1	Fabício, Helton, Julio, Mateus
1.0.0	01/02/2020	Elaboração parcial do Capítulo 2	Fabício, Helton, Julio, Mateus
1.0.0	02/02/2020	Conclusão do Capítulo 2 e elaboração do Capítulo 3	Fabício, Helton, Julio, Mateus

Sumário

1. Introdução	4
2. Implementação	5
3. Conclusão	11

1. Introdução

Neste documento é apresentado uma descrição da implementação do projeto de um sistema de apoio ao atendimento hospitalar. Este sistema é responsável por cadastrar os dados dos pacientes (nome e tempo de exame) que realizaram exame em um hospital e gerar a lista de atendimento por ordem de chamada. Essa ordem é definida com base no tempo de exame dos pacientes. Os pacientes com maior tempo de exame são sempre atendidos primeiro, pois eles são aqueles cujo problema de saúde é o mais grave.

Tecnicamente, o processo de geração da lista de atendimento implementado pelo sistema envolve três estruturas de dados: fila, lista e pilha. O primeiro passo desse processo é cadastrar em uma fila os pacientes cujo exame foi concluído. Naturalmente, esse cadastro é feito a partir do paciente com o menor tempo para o paciente com o maior tempo de exame, pois todos os pacientes começam o exame no mesmo instante de tempo. Dado que o cadastro de todos os pacientes tenha sido finalizado, uma pilha é gerada com a ordem inversa dos dados contidos na fila. Por último, o sistema gera uma lista a partir desta pilha mantendo os dados na mesma ordem em que foram armazenados na pilha. Com a lista definida, o sistema permite gerar um relatório contendo a ordem de chamada dos pacientes para o atendimento médico.

No Capítulo 2 é apresentado uma descrição das principais funcionalidades do sistema. Para apoiar essa descrição é utilizado diagramas de atividades, os quais foram construídos utilizando a Linguagem de Modelagem Unificada¹ (*Unified Modeling Language - UML*). Apesar do sistema ter sido implementado usando a linguagem de programação C e seguindo o paradigma de programação estruturado, alguns diagramas da *UML* podem ser usados para apoiar a sua documentação.

¹ <https://www.uml.org/>

2. Implementação

A implementação do sistema foi realizada usando os seguintes tipos estruturados de dados (TAD), que foram combinados para prover as funcionalidades do sistema:

- **paciente**
- **fila**
- **pilha**
- **lista**
- **interacao**
- **leitura**

No TAD **paciente** foi definido o tipo **Paciente** e as funções para manipulação de pacientes, como, por exemplo, criar e buscar pacientes. No TAD **fila** foi definido o tipo **Fila** e as funções voltadas para a manipulação de uma fila, como, por exemplo, criar e exibir uma fila. De forma análoga, os TADs **pilha** e **lista** definem, respectivamente, os tipos **Pilha** e **Lista**, bem como as funções específicas para manipular essas estruturas de dados. Por exemplo, é no TAD **lista** que está definida a função recursiva **displayPrimeiroMenorTempo**, a qual recebe a lista e um valor de tempo e imprime os dados do primeiro paciente da lista que tem o tempo de exame menor do que o valor de tempo fornecido.

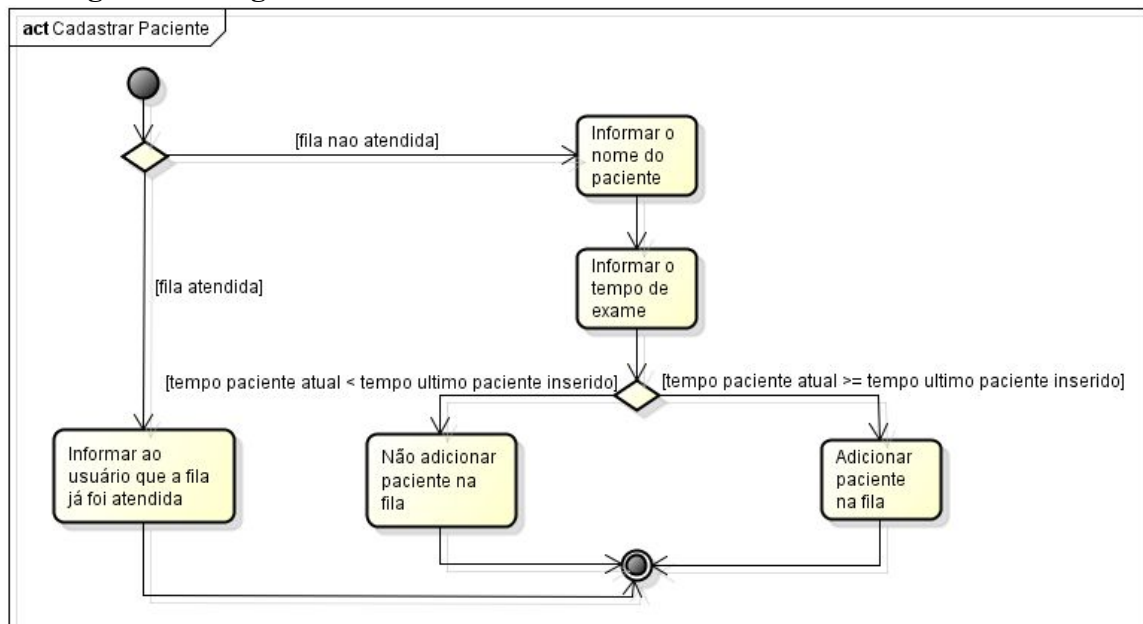
O TAD **interacao** foi definido para prover procedimentos que melhoram a interação do usuário com o sistema, tais, como, menus e limpeza de tela. Por outro lado, o TAD **leitura** disponibiliza funções importantíssimas para a leitura e a validação dos dados fornecidos pelo usuário do sistema. Algumas dessas funções são utilizadas apenas para auxiliar àquelas disponibilizadas no arquivo de cabeçalho do TAD **leitura**, as quais são evocadas no arquivo **.main**.

Uma explicação detalhada de todas as funções e procedimentos do sistema é provida no código-fonte por meio de comentários. Entretanto, a seguir é feita uma explanação sobre as principais funcionalidades do sistema, que são: cadastrar paciente, buscar paciente, buscar paciente por tempo, atender fila, exibir fila, exibir pilha, exibir lista, gerar relatório. Para apoiar essa explanação são apresentados diagramas de atividades relacionados a cada uma dessas funcionalidades.

Na Figura 1 é apresentado as atividades envolvidas no cadastro de um paciente na fila. Conforme pode ser observado, o cadastro de um paciente na fila pode ser feito enquanto a fila

não tenha sido atendida. Quando a fila for atendida a funcionalidade de cadastro é desativada. Além disso, o cadastro de um paciente está submetido a condição de que o seu tempo de exame seja maior ou igual ao tempo de exame do último paciente inserido.

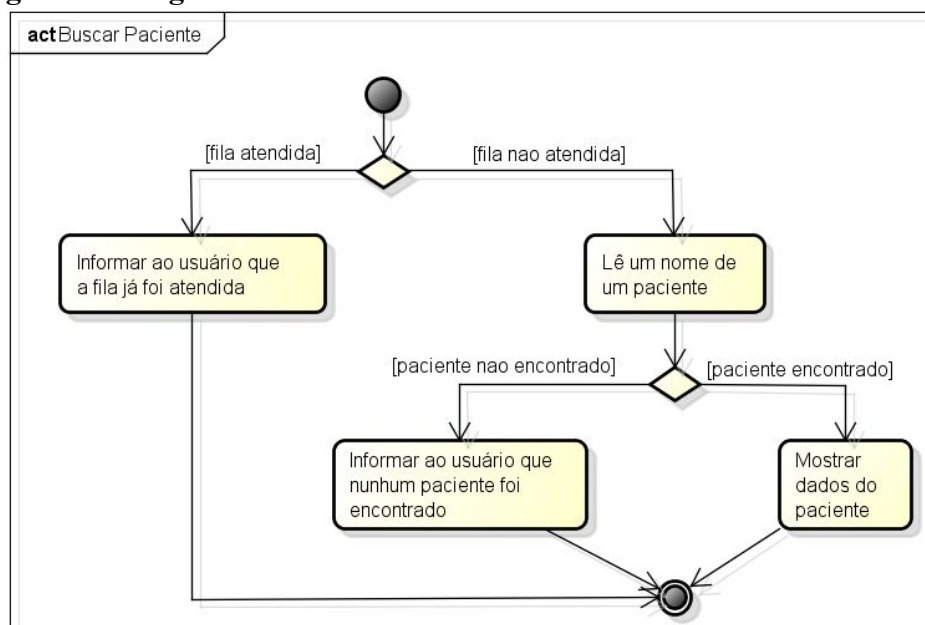
Figura 1 - Diagrama de Atividade da Funcionalidade de Cadastrar Paciente



Fonte: próprios autores.

Similar a função de cadastro, a busca de paciente fica ativada somente enquanto a fila não tenha sido atendida. Para buscar um paciente o usuário deve fornecer como entrada um nome. Caso o nome informado coincida com o nome de algum dos pacientes da fila, os dados desse paciente serão apresentados ao usuário. Todo esse processo é ilustrado na Figura 2.

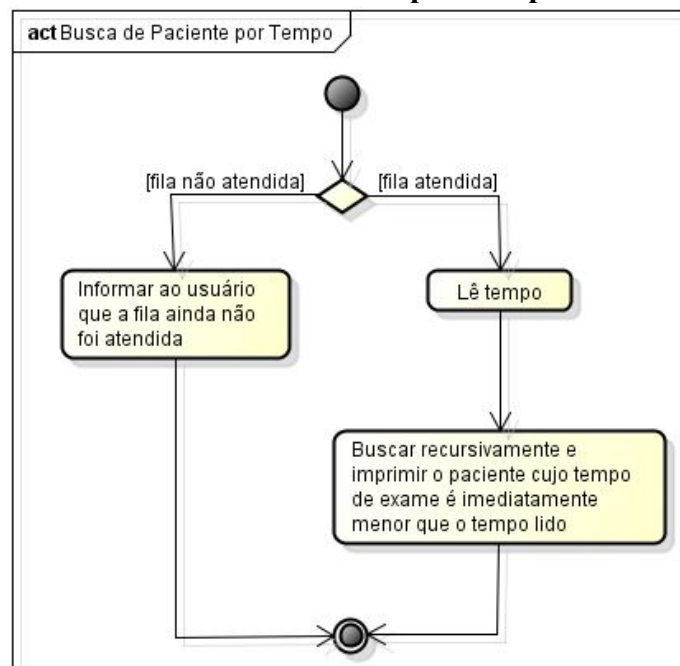
Figura 2 - Diagrama de Atividade da Funcionalidade de Buscar Paciente



Fonte: próprios autores.

Além da funcionalidade de buscar paciente por nome, o sistema provê também uma busca por tempo (vide Figura 3). Nessa busca, o usuário informa um valor de tempo em minutos e o sistema busca e imprime de forma recursiva os dados do paciente cujo tempo de exame é imediatamente menor do que o tempo lido. Diferentemente da busca por nome, a busca por tempo pode ser feita apenas depois que a fila tiver sido atendida. Tecnicamente, o controle de fila atendida ou não é feito usando uma variável inteira chamada **garanteMovi**.

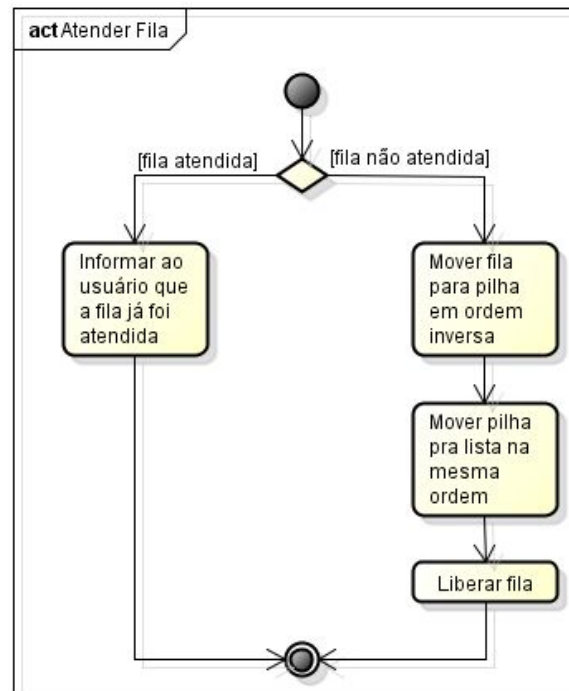
Figura 3 - Diagrama de Atividade da Funcionalidade de Buscar Paciente por Tempo



Fonte: próprios autores.

O comportamento da funcionalidade de atender fila é apresentado na Figura 4. Caso a fila ainda não tenha sido atendido, o atendimento é feito seguindo três etapas: mover a fila para a pilha em ordem inversa; mover a pilha para a lista na mesma ordem; liberar a fila. Na primeira etapa é usado a função **moveFilaPraPilha**, que em conjunto com a função **pushPilha** preenche uma pilha na ordem inversa dos pacientes da fila. Posteriormente, os pacientes armazenados na pilha são copiados para uma lista por meio das funções **movePilhaPraLista** e **pushLista**. Por último, a fila é liberada pela aplicação da função **liberaFila**. Após liberar a fila, caso o usuário tente novamente fazer a liberação, o sistema mostra uma mensagem de alerta informando sobre a impossibilidade de repetir o processo novamente.

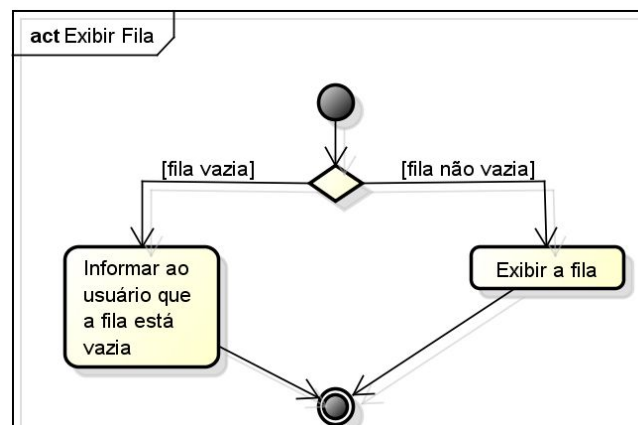
Figura 4 - Diagrama de Atividade da Funcionalidade de Atender Fila



Fonte: próprios autores.

Para permitir ao usuário visualizar os dados armazenados pelo sistema, são providas três funcionalidades de impressão. A primeira delas é apresentada na Figura 5 e é voltada para exibir os dados da fila. Trata-se de uma funcionalidade que simplesmente imprime os dados da fila somente se ela não estiver vazia. Vale destacar que, após a fila ter sido atendida, a fila é esvaziada e, portanto, a funcionalidade de exibir fila sempre informará que a fila está vazia.

Figura 5 - Diagrama de Atividade da Funcionalidade de Exibir Fila

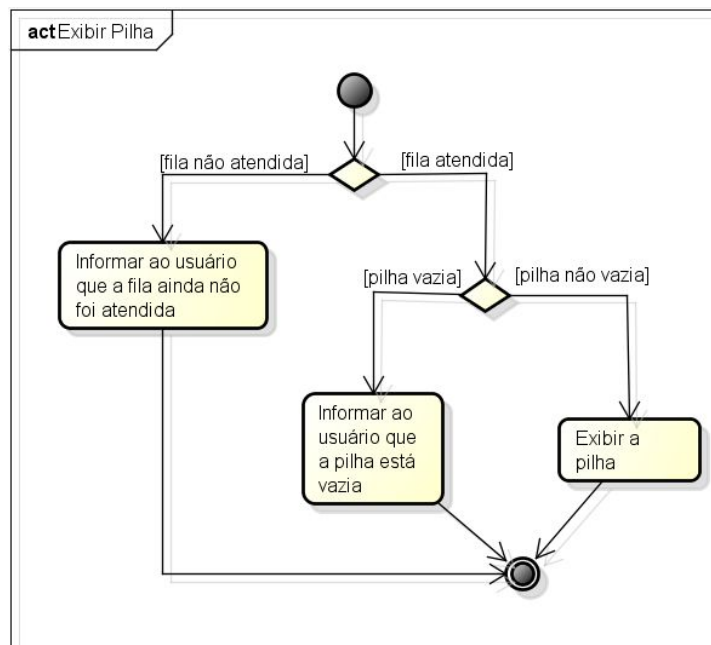


Fonte: próprios autores.

As outras duas funcionalidades de impressão dos dados são apresentadas, respectivamente, nas Figuras 6 e 7. Como é possível perceber, o comportamento delas é muito

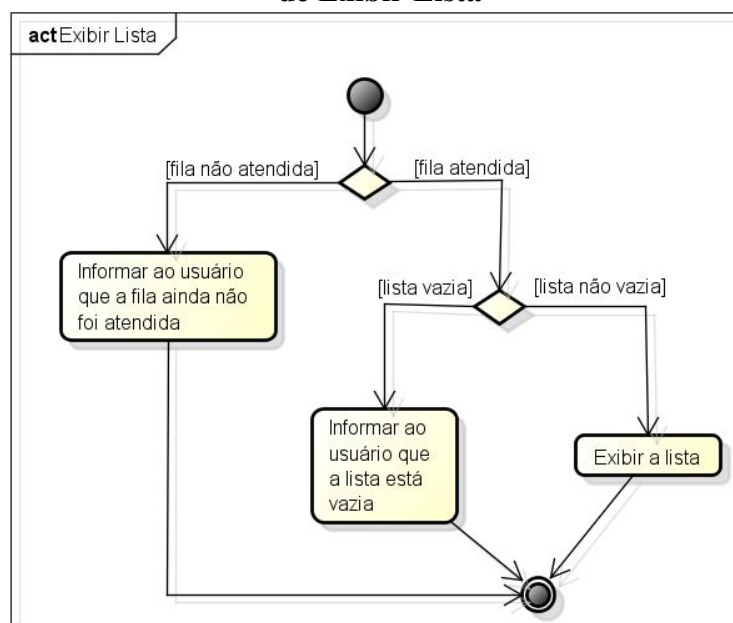
similar, apesar de que a primeira seja responsável por imprimir uma pilha e a segunda por imprimir uma lista. A similaridade entre elas está no fato de seguirem o mesmo fluxo de execução, isto é, realizar a impressão da estrutura de dados somente se a fila tiver sido atendida e a estrutura de dado que se deseja imprimir não estiver vazia.

Figura 6 - Diagrama de Atividade da Funcionalidade de Exibir Pilha



Fonte: próprios autores.

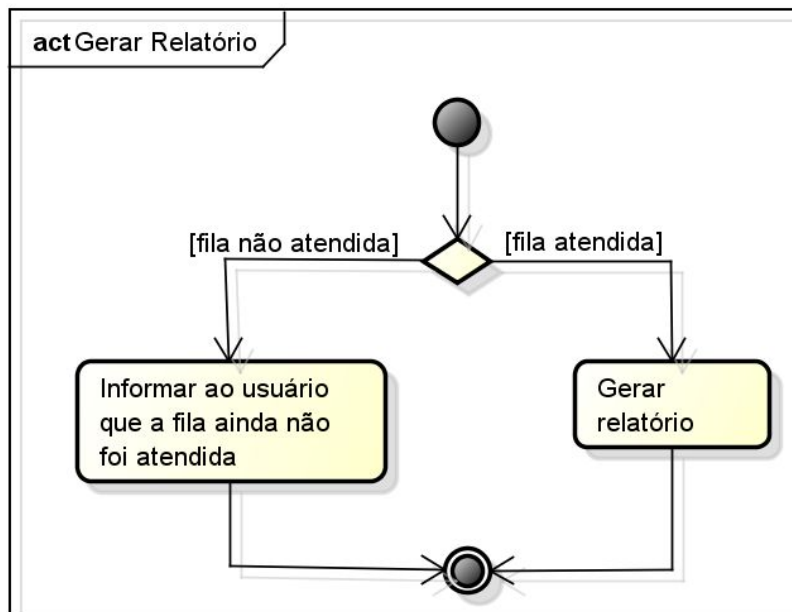
Figura 7 - Diagrama de Atividade da Funcionalidade de Exibir Lista



Fonte: próprios autores.

Por último, é apresentado na Figura 8 o diagrama de atividades da funcionalidade de gerar relatório. Essa funcionalidade salva em um arquivo de extensão .txt os dados dos pacientes armazenados na lista a partir da pilha, a qual, conforme foi mencionado anteriormente, é construída com base na fila de cadastro dos pacientes. A condição para que o relatório seja gerado é que a fila tenha sido atendida, pois, isso garante que os dados da fila de pacientes foram copiados para a pilha e depois para a lista.

Figura 8 - Diagrama de Atividade da Funcionalidade de Gerar Relatório



Fonte: próprios autores.

3. Conclusão

O desenvolvimento deste trabalho prático nos permitiu alcançar uma melhor compreensão das estruturas de dados (fila, lista, pilha) ensinadas em sala de aula, bem como também de outros recursos da linguagem C, como, por exemplo, a manipulação de arquivos por meio de ponteiros e função de escrita. Além disso, em função deste projeto, tivemos a primeira experiência com documentação de um sistema.

Apesar de todo o aprendizado, a realização deste trabalho impôs algumas dificuldades, que incluem, mas não se limitam, a: (1) compreender exatamente cada funcionalidade do sistema e como implementá-las usando a linguagem; (2) manter a coesão das funções, ou seja, manter nas funções somente o código que executa as ações específicas daquela função.