



Universidade Federal do Piauí  
Aluna: Jhoisnáyra Vitória Rodrigues de Almeida  
Curso: Ciência da Computação.

Relatório de Estrutura de Dados:  
Fila com prioridade.

Teresina, 11 de setembro de 2019.

Na classe **Fila**, foi usada a mesma lógica da lista duplamente encadeada, com algumas alterações:

```
package FilaComPrioridade;

public class Fila<Item>{
    public Node<Item> inicio;
    public Node<Item> fim;
    private int quant;
    private int atendidas;

    public class Node<Item>{
        public Item elem;
        protected Node<Item> prox;
        protected Node<Item> ant;
        public Node(Item el) {
            elem = el;
            prox = null;
            ant = null;
        }
    }
}
```

Nesse trecho foi criada a classe Node dentro da fila e seu construtor, e as variáveis de contagem de pessoas na fila e de pessoas atendidas.

```
    public int getAtendidas() {
        return atendidas;
    }

    public Fila() {
        inicio = null;
        fim = null;
        quant = 0;
    }

    boolean isEmpty() {
        return quant == 0;
    }

    int size() {
        return quant;
    }
}
```

```
}
```

Acima foram criados os métodos de verificar tamanho da fila, se está vazia, o construtor da fila e o método get que retorna a quantidade de pessoas atendidas para o cliente.

```
boolean inserir(Item el) {
    Node<Item> aux = inicio;

    if(isEmpty()) {
        Node<Item> a = new Node<Item>(el);
        inicio = a;
        fim = a;
        quant++;
        return true;
    }

    while(aux != fim){
        aux = aux.prox;
    }

    Node<Item> a = new Node<Item>(el);

    aux.prox = a;
    a.ant = fim;
    fim = a;
    quant++;
    return true;
}
```

Nessa parte foi criado o método de inserção, nele cria-se um nó auxiliar para percorrer a fila, e, no primeiro caso (fila vazia), insere-se o elemento no início, no segundo case, percorre até achar um espaço nulo para então inserir.

```
public String toString() {
    String str = "";

    Node<Item> aux = inicio;

    while(aux!=null) {
        str += (aux.elem + "\n");
        aux = aux.prox;
    }
    return str;
}
```

```
}
```

Acima o método para mostrar os elementos da fila.

```
boolean remover() {  
    inicio = inicio.prox;  
    quant--;  
    atendidas ++;  
    return true;  
}
```

E por fim, o método para remover o item, que sempre é feito no início.

```
}
```

No **ClienteFila**, temos:

```
package FilaComPrioridade;  
  
import javax.swing.JOptionPane;  
  
public class ClienteFila {  
  
    public static void main(String[] args) {  
        Fila<String> FilaComPrioridade = new Fila<String>();  
        Fila<String> FilaNormal = new Fila<String>();  

```

Criou-se duas filas, uma para ser prioritária e uma normal.

```
while(true) {  
    String item;  
    int opcao = Integer.parseInt(  
        JOptionPane.showInputDialog("Escolha uma  
das opcoes:\n" +  
        "1 - inserir pessoa comum\n" +  
        "2 - inserir pessoa com prioridade\n"+  
        "3 - atender pessoa\n"+  
        "4 - listar pessoas\n"+  
        "0 - Sair"));
```

No menu, é permitida a opção de adicionar 2 tipos de pessoas, com prioridade e normal.

```

switch (opcao) {
case 1:
    item = JOptionPane.showInputDialog("Pessoa a ser inserida:");
    FilaNormal.inserir(item);
    break;
case 2:
    item = JOptionPane.showInputDialog("Pessoa a ser inserida:");
    FilaComPrioridade.inserir(item);
    break;

```

Acima, a opções de adicionar as pessoas como prioridade ou não.

```

case 3:
    if(FilaComPrioridade.size() == 0 && FilaNormal.size() == 0) {
        JOptionPane.showMessageDialog(null, "Fila vazia! ");
        break;
    }
    if(FilaComPrioridade.size() == 0) {
        FilaNormal.remover();
    }else {
        FilaComPrioridade.remover();
    }
    break;

```

No momento de atender, ou seja, remover o elemento, verificam-se 3 possibilidades: 1 - ambas filas estão vazias, se sim, avisa ao cliente; 2 - se fila com prioridade está vazia, então atende uma pessoa sem prioridade; 3 - se há prioridade, então atende a prioridade.

```

case 4:
    JOptionPane.showMessageDialog(null, FilaComPrioridade.toString()
    +FilaNormal.toString());
    break;

```

Opção de listar elementos da fila, implementada com toString.

```

default:
    if(FilaComPrioridade.size() == 0 && FilaNormal.size() == 0) {
        String total =
Integer.toString(FilaComPrioridade.getAtendidas()+FilaNormal.getAtendidas());

        JOptionPane.showMessageDialog(null, "Pessoas atendidas: "+total);
return;
    }else {

```

```
        JOptionPane.showMessageDialog(null, "Ainda ha pessoas a  
serem atendidas! ");  
        break;
```

```
    }  
}
```

E, por fim, só é permitido encerrar o programa ao verificar se ambas as filas estão vazias.

```
    }  
}  
}
```