


## 2022\_1 - PROGRAMAÇÃO E DESENVOLVIMENTO DE SOFTWARE II - TA\_TN - METATURMA

[PAINEL](#) > [MINHAS TURMAS](#) > [2022\\_1 - PROGRAMAÇÃO E DESENVOLVIMENTO DE SOFTWARE II - TA\\_TN - METATURMA](#) > [GERAL](#)  
> [L01E05 - FILA ATENDIMENTO \(3,0 PTS\)](#)

 Descrição

 [Visualizar envios](#)

### L01E05 - Fila atendimento (3,0 pts)

 **Data de entrega:** terça, 17 Mai 2022, 23:59

 **Arquivos requeridos:** main.cpp, Cliente.hpp, Cliente.cpp, FilaAtendimento.hpp, FilaAtendimento.cpp ( [Baixar](#))

**Tipo de trabalho:**  Trabalho individual

Neste exercício você deve fazer um programa para auxiliar no atendimento à clientes em um banco. Para isso, deve-se implementar uma Fila de Prioridades básica usando o conceito de **Listas Encadeadas**.

Você deverá implementar dois TADs: **Cliente** e **FilaAtendimento**. Cada TAD deve seguir as especificações abaixo:

- **Cliente:**

1. Atributos: `int _idade, _senha; std::string _nome;`
2. `Cliente(std::string nome, int idade, int senha):` Método construtor para a inicialização dos atributos.
3. `bool eh_prioritario():` Método que retorna TRUE caso o cliente tenha idade maior ou igual a 60, e FALSE caso contrário.
4. `int tempo_estimado_atendimento():` Método que retorna o tempo estimado de atendimento de acordo com o cliente. Se for um cliente prioritário o tempo estimado é de 12 minutos, caso contrário de 8 minutos.
5. `void imprimir_dados():` Método que faz a impressão dos atributos no seguinte formato: "**senha nome idade**", com uma quebra de linha ao final. Atenção, nesse caso utilize **tab** (\t) para separar os elementos.

- **FilaAtendimento:**

1. `adicionar_cliente(string nome, int idade):` Método que recebe os dados de um cliente e o adiciona na fila de atendimento. Clientes não prioritários são sempre adicionados ao final, já clientes prioritários são adicionados após todos os prioritários inseridos anteriormente (mas antes do primeiro não prioritário). Além disso, no momento da inserção deve ser atribuída ao cliente uma senha, que é um número inteiro sequencial (a primeira senha deve ter valor 1).
2. `Cliente* chamar_cliente():` Método que remove o primeiro cliente da fila. Você deve retornar um ponteiro para o cliente removido (não é necessário desalocar a memória). Além disso, caso se tente chamar um cliente e a fila esteja vazia deve-se imprimir a seguinte mensagem: "Fila vazia!".
3. `void estimativa_tempo_espera(int senha):` Método que faz uma estimativa do tempo de espera de acordo com a senha do cliente. Deve-se apenas fazer o somatório dos tempos estimados de atendimento de acordo com os tipos dos clientes à sua na frente da fila. Ao final deve-se imprimir uma mensagem no seguinte formato: "Tempo estimado para atender NOMECLIENTE eh X minutos".
4. `void imprimir_fila():` imprime a situação atual da fila, ou seja, percorre toda a lista (do início para o final) e chama o método 'imprimir\_dados()' de cada cliente.

Você é livre para adicionar nos TADs quaisquer outros atributos ou métodos auxiliares que julgar necessário. Além disso, você também pode escolher entre utilizar uma Lista Simplesmente ou Duplamente Encadeada.

Por fim, você deve implementar o arquivo **main.cpp** e adicionar toda a parte de entrada/saída que será responsável por manipular os seguintes comandos:

- '**a nome idade**': comando para adicionar um novo cliente na fila de acordo com os parâmetros passados. Você pode assumir que o nome sempre será uma única palavra.
- '**c**': comando para chamar/remover um cliente da fila.
- '**e senha**': comando para imprimir a estimativa de tempo de espera para o cliente com a senha informada.
- '**p**': comando para imprimir a situação fila.
- '**b**': deve chamar a função '**avaliacao\_basica()**' implementada no arquivo "**avaliacao\_basica\_atendimento.hpp**" (já incluído no main.cpp). Essa função faz uma avaliação do código (não apenas dos resultados).

Para ilustrar, abaixo é apresentado um exemplo de entrada/saída:

```
input=
a João 50
a Maria 60
a Carlos 25
a Joana 70
p
e 1
c
p
output =
2 Maria 60
4 Joana 70
1 João 50
3 Carlos 25
Tempo estimado para atender João eh 24 minutos.
4 Joana 70
1 João 50
3 Carlos 25
```

**Atenção:** Lembre-se de fazer a correta modularização utilizando os arquivos **.hpp** e **.cpp**.

#### Dica 1:

O código do arquivo de avaliação básica pode ser copiado [aqui](#), caso você queira depurar algo localmente.

#### Dica 2:

Você pode usar o código fornecido no exercício anterior (main.cpp) e o da avaliação básica como exemplos para lhe ajudar a fazer toda a parte de entrada/saída.

#### Referências:

<https://www.cplusplus.com/doc/tutorial/pointers/>  
<https://www.cplusplus.com/doc/tutorial/dynamic/>  
<https://www.cplusplus.com/doc/tutorial/structures/>  
<https://www.cplusplus.com/articles/LACRko23/>

## Arquivos requeridos

main.cpp

```
1 // NÃO ALTERE ESSA LINHA
2 #include "avaliacao_basica_atendimento.hpp"
3
4 int main() {
5
6
7     //
8     // Adicione seu código aqui e faça as demais alterações necessárias
9     //
10
11
12     return 0;
13 }
```

Cliente.hpp

Cliente.cpp

FilaAtendimento.hpp

FilaAtendimento.cpp

[VPL](#)

◀ L01E04 - Drones (3,0 pts)

Seguir para...

L01E06 - Folha Salarial (4,0 pts) ▶

