

ENGENHARIA DE COMPUTAÇÃO - ED1

Feito por:

Nicolas Kalebe

Vitor Alexandre



APRESENTAÇÃO DO PROBLEMA BASE

FILA DE BANCO



Nossa ideia inicial era da criação de um programa em C que facilitasse e permitisse que a espera nas filas de bancos acontecesse de forma mais ágil. Diversas notícias correm por todo país por conta da alta demanda e correria diária enfrentadas pelos brasileiros.

Alguns exemplos de notícias:



TEMPO PERDIDO

Tempo na fila do banco acima do limite fixado em lei, por si só, não gera dano moral

Danilo Vital

26 de abril de 2024, 18h52

Consumidor

O simples descumprimento do limite de tempo previsto em lei municipal ou estadual para a prestação de serviços bancários não gera, por si só, dano moral presumido.

Essa é a conclusão final sobre o tema da 2ª Seção do Superior



FILA DE BANCO



Dentro do programa, definimos os Clientes do banco em 4 aspectos:

ID: número de identificação do Banco Idade: idade do cliente Nome: nome do cliențe

Preferência: aspecto físico ou psicológico que disponibilize ao cliente acesso prioritário (gestantes, idosos e PCD - Pessoas com Deficiência)



ETAPA 1 DO PROJETO

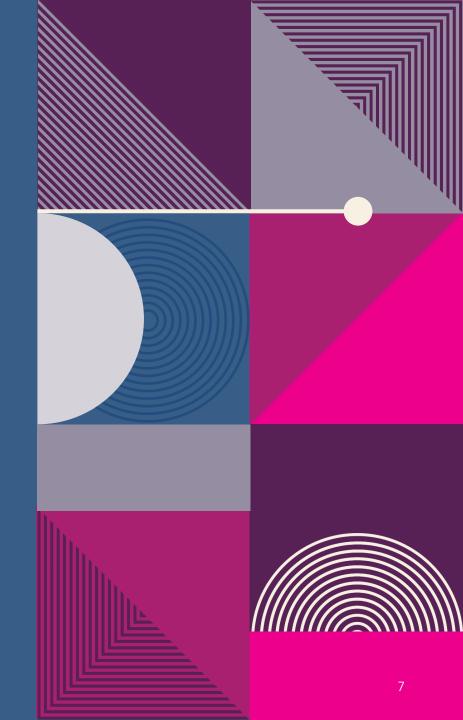
Estrutura de dados

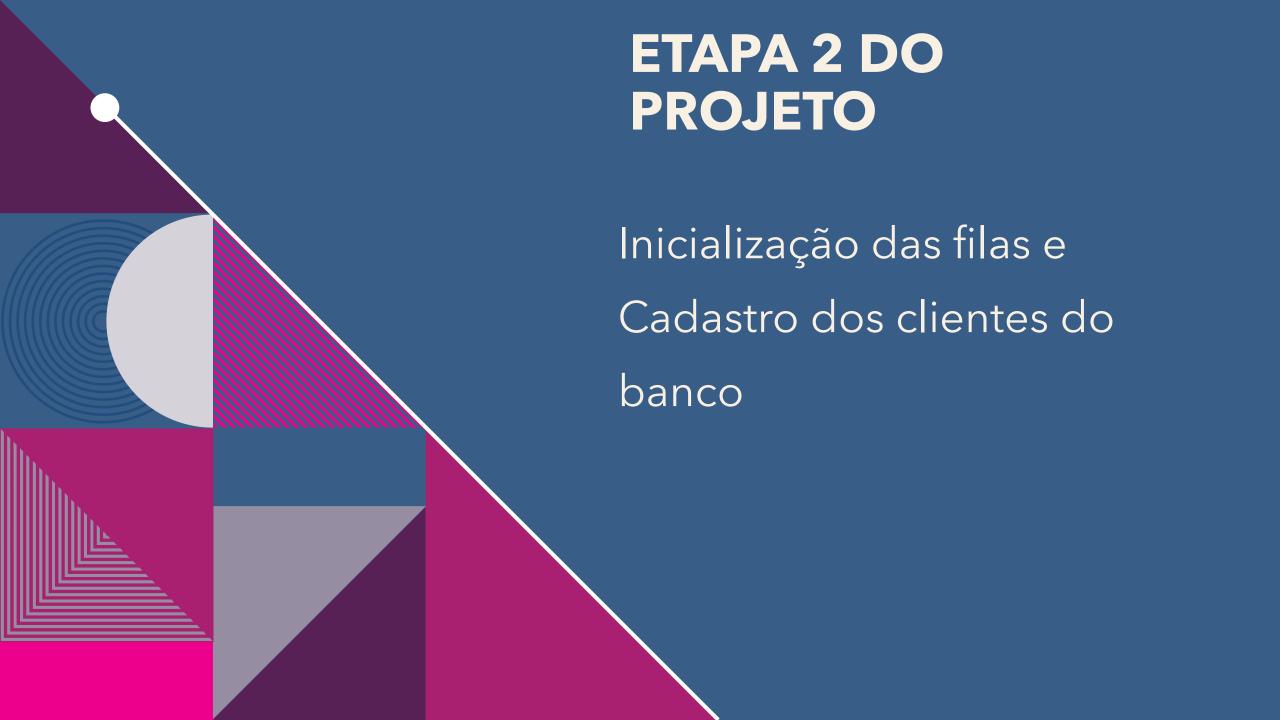
ESTRUTURA DE DADOS

Dentro do programa, definimos a estrutura dos "clientes" do banco, gerando as informações na fila: ID, nome, idade, preferência e o ponteiro para o próximo cliente.

Estrutura da fila do tipo FIFO (First In, First Out), apresentada em sala de aula. Nesse tipo de fila, o primeiro elemento a ser inserido é o primeiro a ser removido, seguindo uma ordem dinâmica.

```
// Estrutura do cliente do banco
     typedef struct Cliente {
         int id;
         char nome[50];
         int idade;
         bool preferencia;
         struct Cliente *proximo;
12
13
      Cliente;
15
     // Estrutura da fila
     typedef struct Fila {
         Cliente *frente:
17
         Cliente *tras;
18
       Fila;
19
```





INICIALIZAÇÃO DAS FILAS

O código apresentado inicia as filas. Iniciando uma fila, ela estará sem clientes tanto na frente quanto atrás (início e fim da fila vazios)

```
int contadorID = 0; // Contador para IDs

// Função para inicializar a fila

void iniciaFila(Fila *fila) {

fila->frente = fila->tras = NULL;

}
```

CADASTRO DOS CLIENTES

```
// Função para adicionar cliente à fila
      void adicionaCliente(Fila *filaPreferencial, Fila *filaNormal) {
          char nome[50];
          int idade;
         printf("\nDigite o nome do cliente: ");
          scanf(" %[^\n]", nome);
          printf("Digite a idade do cliente: ");
          scanf("%d", &idade);
         Cliente *novo = (Cliente *)malloc(sizeof(Cliente));
          if (!novo) -
             printf("Erro ao alocar memória.\n");
              return;
          novo->id = ++contadorID;
          strcpv(novo->nome, nome);
          novo->idade = idade;
          novo->preferencia = (idade >= 60);
          novo->proximo = NULL;
         Fila *fila = (novo->preferencia) ? filaPreferencial : filaNormal;
          if (fila->tras == NULL) {
              fila->frente = fila->tras = novo;
              fila->tras->proximo = novo;
              fila->tras = novo;
          printf("\nCliente %s cadastrado com sucesso! ID: %d\n", nome, novo->id);
114
          // Mensagem dada quando o processo é finalizado corretamente
```

No código ao lado, os clientes são adicionados e cadastrados no Banco. Nesse processo, dados como a **idade** e a preferência irão indicar para qual fila o cliente deverá ir. O ID será usado para alocação de memória e mais fácil posterior remoção da fila que foi atendida.

CADASTRO DOS CLIENTES

No exemplo do código fornecido, é possível ver que, caso o cliente possua idade igual ou superior a 60 anos, ele se encaixa nos clientes com preferência, recebendo prioridade e indo até a fila preferêncial

```
97 }
98 novo->id = ++contadorID;
99 strcpy(novo->nome, nome);
100 novo->idade = idade;
101 novo->preferencia = (idade >= 60);
102 novo->proximo = NULL;
103
104 Fila *fila = (novo->preferencia) ? filaPreferencial : filaNormal;
```

ETAPA 3 DO PROJETO



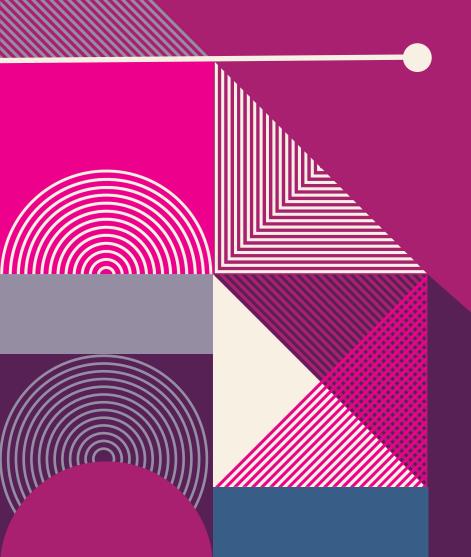
Exibição da Fila

EXIBIÇÃO DA FILA

O processo de exibição das filas é muito importante não só para transparência do sistema, mas também para analisar se os clientes estão sendo atendidos corretamente.

```
116
      // Função para exibir a fila
117
      void exibirFila(Fila *fila, const char *descricao) {
118
          printf("\nFila %s:\n", descricao);
          if (fila->frente == NULL) {
                         Nenhum cliente na fila.\n");
              printf("
              return:
124
          Cliente *atual = fila->frente;
          while (atual != NULL) {
126
              printf(" ID: %d | Nome: %s | Idade: %d | Preferência: %s\n",
127
128
                     atual->id, atual->nome, atual->idade, atual->preferencia ? "Sim" : "Não");
              atual = atual->proximo;
129
130
131
```

ETAPA 4 DO PROJETO



Atendimento ao Cliente

e Remoção do Cliente por ID

ATENDIMENTO AO CLIENTE

O atendimento dos clientes é necessário para que o andamento da fila ocorra não só de forma ágil, mas também de forma correta e dando prioridade sempre àqueles que possuem preferência no sistema.

```
// Função para remover cliente pelo ID
41 void removerClientePorID(Fila *fila, int id) {
         if (fila->frente == NULL) {
             printf("\nA fila está vazia.\n");
             return:
         Cliente *anterior = NULL;
         Cliente *atual = fila->frente;
         while (atual != NULL) {
             if (atual->id == id)
                 if (anterior == NULL)
                     fila->frente = atual->proximo;
                     if (fila->frente == NULL) {
                         fila->tras = NULL;
                     anterior->proximo = atual->proximo;
                     if (atual == fila->tras) -
                         fila->tras = anterior;
                 printf("\nCliente com ID %d (%s) foi removido da fila.\n", atual->id, atual->nome);
                 free(atual);
                 return;
             anterior = atual;
             atual = atual->proximo;
         printf("\nCliente com ID %d não encontrado.\n", id);
```

REMOÇÃO DO CLIENTE POR ID

Nesse caso, o cliente com preferência será sempre priorizado antes no atendimento, atendendo os clientes localizados na fila normal apenas se a prioritária estiver vazia, garantindo assim dinamicidade e facilidade de atendimento as pessoas mais necessitadas.



CRIAÇÃO DO MENU

O menu foi criado para facilitar e permitir a visualização dos comandos do código de modo eficiente para um administrador que esteja trabalhando no Banco. A partir dele, é possível **Cadastrar Cliente**, **Exibir Fila**, **Atender Próximo** e **Remover por ID.** O comando **Sair** finaliza o programa e fecha o terminal.

```
// Função do menu principal
133
      void menu(Fila *filaPreferencial, Fila *filaNormal) {
          int opcao:
135
136
          do {
137
              printf("\n- - - Bem vindo ao Banco UFG - - -\n");
              printf("1. Cadastrar cliente no banco\n");
138
              printf("2. Exibir filas\n");
139
              printf("3. Atender próximo cliente\n");
              printf("4. Remover cliente por ID\n");
141
142
              printf("0. Sair\n");
143
              printf("Escolha uma opção: ");
              scanf("%d", &opcao);
144
```

CRIAÇÃO DO MENU

```
switch (opcao)
       case 1:
          adicionaCliente(filaPreferencial, filaNormal);
          break;
      case 2:
          exibirFila(filaPreferencial, "Preferencial");
           exibirFila(filaNormal, "Normal");
          break;
       case 3:
          if (filaPreferencial->frente != NULL) {
               removerClientePorID(filaPreferencial, filaPreferencial->frente->id);
           } else if (filaNormal->frente != NULL) {
               removerClientePorID(filaNormal, filaNormal->frente->id);
           } else {
               printf("\nNenhum cliente na fila para atender.\n");
          break;
       case 4: {
           int id;
          printf("\nDigite o ID do cliente a ser removido: ");
           scanf("%d", &id);
          removerPorID(filaPreferencial, filaNormal, id);
          break;
      case 0:
           printf("\nEncerrando o sistema...\n");
          break;
      default:
          printf("\nOpção inválida. Tente novamente.\n");
while (opcao != 0);
```

Dentro do código, a função do **Menu** foi criada e desenvolvida por meio de um *switch* case, facilitando a alteração do código e visualização das ações.

CONCLUSÕES FINAIS

A partir do nosso programa em C, o funcionamento das filas de bancos poderá se tornar muito mais simplificado e facilitado. Com sua implementação, a agilidade em atender os clientes com prioridade permitirá maior aceitação do público e, consequentemente, irá melhorar a visão da empresa.

O código do programa apresenta poucas funções e comandos, aos quais são intuitivos e facilmente manipuláveis por um administrador de banco que trabalha nesse setor