

**Universidade Federal do Ceará - Campus Crateús**  
**Disciplina: Estrutura de Dados**  
**Ciência da Computação**

**Autores:**

**Antonio Emerson Gomes Camelo - 555535**  
**Wenderson Gabriel Sousa Mourão - 537734**  
**Eduardo Levi Portela Miranda - 537030**

**Relatório de atividades - Trabalho 2**

**Crateús - CE**  
**2023**

## **METODOLOGIA:**

A resolução do TAD foi feita por etapas. Cada Função foi separada entre os membros do grupo que resolveram suas respectivas demandas. Emerson foi responsável pelo desenvolvimento do TAD para a fila (cabeçalho e implementação), TAD para a pilha, implementação das funções relacionadas à fila, como adicionar elementos, verificar se está vazia, verificar se está cheia, entre outras operações necessárias para o funcionamento das filas e Implementação das funções relacionadas à pilha, como empilhar, desempilhar, verificar se está vazia, entre outras operações necessárias para o funcionamento da pilha. Eduardo e Wenderson ficaram responsáveis pelos testes e depuração das funções implementadas, bem como o desenvolvimento do programa principal em C e implementação dos menus de controle, interação com o usuário e integração com os TADs de fila e pilha.

## **DIFICULDADES ENFRENTADAS:**

Tivemos dificuldade para desenvolver uma lógica de geração de senhas que tenha uma quantidade finita e um intervalo determinado para cada tipo de fila. Por exemplo, para a fila normal, tivemos que determinar a geração de no intervalo entre 1 e 10. Já para pessoas prioritárias, de 11 a 20. Com isso, se tiver um cliente com a senha 10, o próximo cliente que chegar deverá receber a senha 1, isto acontecerá somente se não tiver 10 pessoas na fila ainda. Ou seja, foi uma forma que achamos para que não gerasse senhas que são muito grandes.

Além disso, outra dificuldade foi em relação a como guardar pessoas que estão em filas diferentes e serem inseridas na pilha que guarda as últimas chamadas. Para resolver isso, fizemos basicamente uma cópia da senha e o tipo de fila que a pessoa estava e localizamos na pilha .

Também tivemos certa dificuldade em desenvolver uma lógica que chamar uma pessoa prioritária a cada duas da fila normal. Para isso, tivemos que analisar na pilha quais foram as últimas duas pessoas que foram atendidas e assim chamar a próxima seguindo a regra de prioridade.

Ademais, algumas dúvidas sobre alocação dinâmica e TAD surgiram, mas foram superadas e o código está sendo executado da forma correta.

## **CONTRIBUIÇÕES PARA O APRENDIZADO:**

- Trabalhar em um projeto com diferentes responsabilidades entre os membros da equipe, comunicando-se e coordenando esforços para alcançar um objetivo comum. Visto que uma parte do grupo ficou responsável por realizar testes e depuração enquanto o outro escreveu o código principal foi necessário bastante comunicação.
- A importância de realizar testes frequentes para verificar a integridade e correção do código, assim como a depuração para encontrar e corrigir erros.
- Capacidade de decompor um problema complexo em subproblemas menores, desenvolver um plano de ação, atribuir responsabilidades e combinar as partes em um todo funcional.
- Capacidade de lidar com eventos inesperados, como filas vazias ou entradas de usuário incorretas, e desenvolver estratégias para resolvê-los sem interromper o programa
- Agora na parte de código se fez necessários alguns conhecimentos sólidos como:

Fila e pilha: Aprendemos a implementar essas estruturas de dados em C, usando ponteiros para criar filas e pilhas, e implementando funções para inserir, remover e manipular elementos.

Memória dinâmica: Para lidar com a capacidade variável de filas e pilhas, entendendo como alocar e liberar memória.

Funções, Estruturas e Tipos de Dados: Compreender profundamente o uso de funções, estruturas (struct), tipos de dados e manipulação de memória em C.

Manipulação de Entrada e Saída de Terminal: conseguir interagir com o usuário por meio de menus, exibição de informações e leitura de entrada pelo terminal.