

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE CAMPINAS
ESCOLA POLITÉCNICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

RELATÓRIO SOBRE PROJETO DE SISTEMA DE COMPANHIA
AÉREAS

Felipe Cassante Ganzarolli — RA: 24003492

Felipe Grolla Freitas — RA: 24004846

Gabriel Henrique Pozeti de Faria — RA: 24011960

João Victor Vasconcelos Junqueira Criscuolo — RA: 22024547

Lucas Espica Rezende — RA: 24006575

Campinas — Sp

2024

1. INTRODUÇÃO

O presente relatório visa demonstrar o processo de criação de um software criado em C para a gestão dos voos de uma companhia aérea (no caso, a companhia “Voe Sempre, Voe Feliz”), bem como os resultados obtidos.

O sistema proposto visa fornecer funcionalidades essenciais para o gerenciamento eficiente dos voos, como a adição, alteração e remoção de voos, bem como filtros de checagem com base nos voos cadastrados.

Através da implementação dessas funcionalidades, busca-se otimizar o processo de administração dos itinerários, assim garantindo a precisão das informações disponibilizadas aos usuários e a melhora da eficiência operacional da companhia aérea.

2. PROPOSTA DE PROJETO E DESENVOLVIMENTO

2.1. Menu De Início

O sistema desenvolvido em questão deve expor em sua tela inicial um menu (função main) por meio do qual o usuário possa decidir e, então, selecionar uma de oito funções disponíveis para serem realizadas, sendo elas incluir voo, alterar informações de um voo, apagar um voo, determinar quais voos saem de uma cidade de origem escolhida, determinar qual voo dentre todos os que saem e chegam em cidades iguais possui o menor número de escalas, determinar quais voos chegam em uma cidade de destino escolhida, exibir todos os voos e suas informações e, por fim, sair do sistema, sendo que a cada função selecionada e finalizada (com exceção da última), é ofertada novamente ao usuário a escolha de opções, por fim, a qualquer momento nas funções, é informado que digitar -1 encerrará a função prematuramente.

2.2. Função 1, Incluir voo

A função 1 é utilizada para adicionar novos voos ao sistema, atualizando o array de voos (*v*) e o contador de voos (*contador_voo*). Seu funcionamento se dá a partir de 4 etapas, dentre elas estão: verificação de voo duplicado (a partir do código do voo), entrada de dados de cadastro do voo (preenchimento dos campos como: código do

voo, cidade origem, destino, escala e número de escalas, ocorrendo também uma conversão na entrada de dados, onde todos os caracteres inseridos pelo usuário são convertidos em maiúsculo com a função `toupper`), posterior confirmação e, por fim, laços de repetição (Onde após o usuário preencher os dados do voo, há um contador de voos (`contador_voo`) que é incrementado).

A cada adição a função verifica se o número máximo de voos foi atingido e, se sim, informa o usuário e encerra o loop; Além disso, o usuário é questionado se deseja adicionar mais um voo. Se a resposta for negativa, a função limpa a tela e finaliza o processo de adição de voos).

Assim sendo, o processo interativo em questão assegura que os dados sejam inseridos corretamente e oferece ao usuário a opção de continuar adicionando voos até atingir o limite máximo.

2.3. Função 2, Alterar Informações de um Voo

A função 2 permite ao usuário modificar as informações de um voo previamente cadastrado.

A função começa verificando se há voos cadastrados e, caso existam, exibe todos os voos e suas informações em forma de lista. Após este processo o usuário é então solicitado a escolher o voo que deseja alterar.

Se o número do voo informado for válido, o usuário pode escolher alterar qualquer informação do mesmo. Caso a escolha seja alterar as escalas, o usuário pode optar por aumentar ou diminuir a quantidade de escalas, ou alterar a cidade de uma escala específica.

Se a opção for diminuir a quantidade de escalas, o usuário informa o número da cidade de escala a ser excluída, e a função remove a cidade de escala correspondente, ajustando o número total de escalas.

Se a opção for alterar uma cidade de escala, o usuário informa o número da escala e o novo nome da cidade, sendo então atualizado no sistema.

A função inclui verificações de entrada e mensagens de erro caso o voo ou a opção escolhida sejam inválidos, garantindo que o usuário seja notificado de qualquer entrada inválida e possa tentar novamente. Se não houver voos cadastrados, a função exibe uma mensagem de erro e pausa a execução, informando que é necessário adicionar voos antes de poder alterá-los.

2.4. Função 3, Apagar voo

A função 3 é usada para excluir voos previamente cadastrados no sistema, atualizando o array de voos `v` e o contador de voos `contador_voo`.

A função começa verificando se há voos cadastrados, e caso não haja, exibe uma mensagem de erro informando não haver voos para serem excluídos, pausando a execução para que o usuário possa ler a mensagem. Se há voos cadastrados, a função exibe uma lista de todos os voos, mostrando seus respectivos dados.

Em seguida, a função solicita ao usuário que insira o número do voo que deseja excluir, validando o número inserido para garantir que está entre os voos cadastrados. Se o número do voo a ser excluído é válido, o índice do voo é ajustado e a função remove o voo, deslocando todos os voos subsequentes uma posição para trás no array, sobrescrevendo efetivamente o voo a ser excluído.

Caso o número do voo inserido pelo usuário seja inválido, a função exibe uma mensagem de erro informando que o voo não existe e retorna ao menu principal.

Portanto, o principal objetivo da função é gerenciar a exclusão de voos cadastrados no sistema, garantindo que o usuário possa visualizar todos os voos antes de selecionar qual deseja excluir, validando a entrada para evitar erros e atualizando o array de voos e o contador de voos eficientemente.

2.5. Função 4, Determinar Quais Voos Saem De Uma Cidade De Origem Escolhida

A função 4 visa exibir todos os voos cadastrados e permitir ao usuário filtrar os voos por cidade de origem. A função começa listando todos os voos cadastrados, mostrando todas suas respectivas informações como o número do voo, cidade de origem, cidade de destino e a quantidade de escalas. Se um voo possui escalas, a função também exibe as cidades de escala. Se existem voos cadastrados (ou seja, `contador_voo` é diferente de zero), a função pede ao usuário para inserir uma cidade de destino para filtrar os voos. Após a entrada do usuário, a função converte o nome da cidade para letras maiúsculas (`strupr(filtro_cid_dest)`) para assegurar que a comparação seja correta.

Em seguida, a função percorre a lista de voos novamente e mostra apenas os voos cujo destino corresponde à cidade inserida pelo usuário. Se encontrar voos com o destino especificado, ela exibe os detalhes desses voos. Caso não encontre nenhum voo com o destino especificado, a função avisa ao usuário que não há voos cadastrados para essa cidade de destino.

OBS: Se não houver voos cadastrados, a função mostra uma mensagem de erro informando não haver voos adicionados e solicita ao usuário que adicione voos antes de tentar visualizar ou filtrar a lista novamente.

Em ambos os casos, a função pausa a execução do programa no final para o usuário poder ler as mensagens exibidas.

2.6. Função 5, Determinar Qual Voo Dentre Todos Os Que Saem E Chegam Em Cidades Iguais Possui O Menor Número De Escalas

A função 5 é utilizada para listar todos os voos cadastrados e encontrar o voo com o menor número de escalas entre uma cidade de origem e uma cidade de destino especificadas pelo usuário.

A função começa inicializando as variáveis de comparação para as escalas e o número do voo, logo em seguida, verifica se há voos cadastrados e, se houver, exibe uma lista detalhada de todos os voos com suas informações. Caso não haja voos cadastrados, a função limpa a tela e exibe uma mensagem de erro, informando não haver voos disponíveis e pausa a execução.

Após a listagem dos voos, a função solicita ao usuário que insira a cidade de origem e destino para começar a filtrar os voos. As entradas são convertidas para letras maiúsculas para padronização e a função então itera sobre os voos cadastrados, comparando a cidade de origem e destino de cada voo com os filtros fornecidos pelo usuário. Se encontrar um voo que corresponda aos filtros e tenha menos escalas do que os previamente encontrados, a função atualiza as variáveis de comparação.

O objetivo principal da função é fornecer ao usuário uma ferramenta para listar todos os voos cadastrados e identificar o voo com o menor número de escalas entre uma origem e um destino específicos, facilitando a visualização e a seleção de voos mais eficientes.

2.7. Função 6, Determinar Quais Voos Chegam Em Uma Cidade De Destino Escolhida

A função 6 foi feita para exibir todos os voos cadastrados e permitir ao usuário filtrar os voos por cidade de destino.

Inicialmente, a função imprime uma lista de todos os voos, mostrando detalhes como o número do voo, cidade de origem, cidade de destino e o número de escalas. Se um voo tiver escalas, a função também lista as cidades de escala.

Após a entrada do usuário, a função converte a cidade para letras maiúsculas (``strupr(filtro_cid_dest)``) para garantir a comparação correta. Em seguida, a função percorre novamente a lista de voos, desta vez exibindo apenas os voos cujo destino corresponde à cidade filtrada pelo usuário.

Se encontrar voos com o destino especificado, a função imprime os detalhes desses voos. Caso contrário, se nenhum voo corresponder ao destino filtrado, a função informa ao usuário que não há voos cadastrados com essa cidade de destino.

OBS: Se não há voos cadastrados, a função exibe uma mensagem de erro indicando não haver voos adicionados e solicita ao usuário que adicione voos antes de tentar visualizar ou filtrar a lista novamente.

Em ambos os casos, a função pausa a execução do programa no final para o usuário poder ler as mensagens exibidas.

2.8. Função 7, Exibir Todos Os Voos E Suas Informações

A função 7 é responsável por exibir a lista dos voos cadastrados no sistema.

Quando chamada, a função imprime uma mensagem inicial indicando que a lista de voos será mostrada e então verifica se há voos cadastrados, verificando se a variável ``contador_voo`` é diferente de zero.

Caso haja voos cadastrados, a função percorre o vetor ``v`` (que contém os dados dos voos) e imprime detalhes de cada voo, incluindo o número do voo, código do voo, cidade de origem, cidade de destino, e o número de escalas.

Se um voo possui escalas, a função também imprime as cidades de escala correspondentes.

Ao final da listagem, a função pausa a execução do programa para o usuário poder visualizar a lista e, caso não haja voos cadastrados, a função limpa a tela e exibe

uma mensagem de erro informando não haver voos adicionados, solicitando ao usuário que adicione voos antes de tentar acessar a lista novamente, pausando então a execução para que o usuário possa ler a mensagem.

2.9. Função 8, Sair Do Programa

A função 8, que não é uma função de fato, se trata da forma que o usuário possui de sair do programa, tendo sido desenvolvida dentro do programa principal (main), utilizando o comando `exit(0)` para que o programa seja fechado instantaneamente.

3. DIFICULDADES, SOLUÇÕES E DESENVOLVIMENTO DO PROJETO

Algumas das dificuldades com as quais o grupo se deparou foram por exemplo: entendimento e utilização de ponteiros, a impressão de todas as escalas e suas respectivas cidades, utilização de switch case dentro de outro switch case, nomenclatura de variáveis (comunicação do grupo), necessidade de utilização de comandos como “`fflush`” para limpeza de buffer, entre outras.

Para solucionar tais problemas, nos aprofundamos ainda mais nos estudos sobre conceitos (ponteiros, void, int main e etc), para assim melhorarmos nossa lógica, conseguindo então realizar funções mais complexas.

Em problemas os quais o grupo não conseguiu uma solução por conta própria, foi então solicitada a ajuda da professora da disciplina, que então direcionou o grupo para o caminho certo.

4. CONCLUSÃO

Concluimos então que o desenvolvimento do sistema de gestão para a companhia aérea alcançou seus objetivos principais, proporcionando uma plataforma robusta e eficiente para a administração de voos.

As funcionalidades implementadas, como a adição, alteração e remoção de voos, bem como filtros para ver voos que saem e chegam em uma determinada cidade e uma lista na qual apresenta todos os voos cadastrados, demonstraram ser formas eficazes na otimização dos processos internos bem como otimizadoras da experiência de uso dos usuários finais do programa.

O uso da linguagem C permitiu a criação de um sistema escalável e adaptável às necessidades futuras da companhia, tendo o projeto não apenas validado a viabilidade técnica para soluções práticas, mas também destacado a importância de um gerenciamento eficiente no setor aéreo.

5. REFERÊNCIAS

GUIMARÃES, Lúcia Filomena de Almeida. **SEGUNDO_PROJETO-1.docx**. 2024. Disponível em <https://puc-campinas.instructure.com/courses/51258/files/2423775?wrap=1>. Acesso em 21 de maio de 2024.