

CIÊNCIA DA COMPUTAÇÃO – 2º PERÍODO

ALGORITMOS II

DR. MANOEL PEREIRA JÚNIOR

MANUAL TÉCNICO DO SISTEMA DE HOTEL

VINÍCIUS RODRIGUES FERREIRA

SAMUEL RODRIGUES VIANA DE FARIA

FORMIGA – MG

ABRIL DE 2021

SUMÁRIO

1) Main.c.....	3
2) GestaoDeDados.c.....	3
3) Reservas.c.....	3
4) Transacoes.c.....	4
5) Arquivos.c.....	4
6) Feedback.c.....	4
7) Importacao_exportacao.c.....	5

1. main.c

Este módulo contém as principais ações do sistema. Ele é organizado no menu principal e submenus, com opções que são selecionadas através de estruturas de controle com os comandos if, else e “switch. Cada opção selecionada dentro de um submenu leva a chamada de sub-rotinas de outros módulos. Abaixo estão suas sub-rotinas:

- Função “void CriarAcomodacao()”

A opção 4 do sub-menu de cadastro armazena os valores lidos pelas funções fgets e scanf nos dados referentes a struct Acomodacoes. A sub-rotina “void CriarAcomodacao ()” compara a variável categoria da struct Acomodacoes com a descrição de uma categoria pré-cadastrada. Se achar um resultado compatível, os campos da struct de categorias de acomodação (valor da diária e a quantidade de crianças e adultos que comporta) serão associadas às variáveis da struct de acomodações.

- Função “void CriarAcomodacaoArquivos()”

Tem a mesma função da sub-rotina explicada anteriormente, mas está é chamada quando um arquivo, texto ou binário, é lido.

- Função “int main()”

Esta função contém os menus e a chamada das funções de outros módulos.

2. GestaoDeDados.c

O módulo de gestão de dados é o responsável pela leitura dos dados de hóspedes, categorias de acomodação, acomodações, produtos para os hóspedes, fornecedores e operadores. Os dados estão contidos em structs, e são armazenados em variáveis através das funções scanf, para o caso de dados do tipo numérico (int, float, long int, etc) e fgets, para dados do tipo string. Este módulo também é responsável por modificar, excluir e mostrar em tela todos os dados cadastrados. Ao modificar um cadastro, como por exemplo, o do hotel, um menu feito através da estrutura de controle switch seleciona um dos case para modificar apenas uma variável, mudando permanentemente seu valor.

A exclusão de dados é feita ao digitar o nome ou descrição do cadastro selecionado, que compara este nome com o de um cadastro existente através da função strcmp, responsável por comparar duas strings. Após esta operação, o índice do cadastro selecionado com strcmp é passado para a última posição e, em seguida, a struct é decrementada um valor.

Já as sub-rotinas que mostram os dados das structs em tela tem a opção tanto de mostrar todos os dados, como também mostrar somente um cadastro selecionado pelo operador.

3. Reservas.c

Este módulo tem a função de agendar ou cancelar uma reserva realizada. O módulo contém as seguintes sub-rotinas:

- Função “int *AgendarReserva()”

Essa função recebe como parâmetros as structs referentes às acomodações e reservas e seus respectivos tamanhos. Na sub-rotina são informados pelo usuário a data inicial e final, código da acomodação que deseja ser agendada, categoria, facilidades, quantidade de adultos e crianças, tais variáveis são lidas através de fgets e scanf. É feita uma manipulação, através da função sscanf, nas datas retidas pelo fgets, a fim de transformar a string data armazenada em segundos por meio de uma função da biblioteca time.h. Após toda coleta de informação são utilizados laços para verificar se o agendamento da reserva está disponível e assim a função retorna um vetor que indica o agendamento ou o motivo da reserva de estar indisponível.

- Função “void CancelarReserva()”

Essa função é responsável pelo cancelamento de uma reserva existente, a mesma recebe como parâmetros a struct referente às reservas e seu tamanho. Na subrotina é coletado o nome do responsável pela reserva e é feito um laço para verificar se o mesmo está dentre as reservas existentes, se o strcmp for satisfeito a reserva é excluída através de um while que passa a determinada reserva para a última posição e diminui o tamanho em 1 da struct reserva.

4. Transacoes.c

O módulo de transações é responsável por todo processo referente ao hotel que envolva entrada e saída de dinheiro. O mesmo trabalha com sub-rotinas que englobam pagamentos de hóspedes em check-in, check-out e venda de produtos e também possui funções que realizam pagamentos à fornecedores, entrada de produtos industrializados, geração de nota fiscal e amostragem/leitura de contas referentes à hóspedes ou fornecedores.

5. Arquivos.c

Este módulo é responsável por todas as sub-rotinas que envolvem gravação e leitura de dados locais, sejam em formato de texto ou binário. As funções de gravação de dados recebem como parâmetro suas respectivas structs de acordo com seu nome e uma variável que define o formato de gravação. Já as funções responsáveis pela leitura de arquivos são separadas em tipo txt e binário, pelo fato de possuírem um processo diferente em sua execução. Em sub-rotinas de gravação de arquivo do tipo txt é utilizado a função fscanf e em bin é utilizado o fwrite. Já em sub-rotinas de leitura são utilizadas as funções fscanf e fread para arquivos txt e bin.

6. Feedback.c

O módulo de feedback é responsável por imprimir relatórios correspondentes à determinadas structs selecionadas pelo usuário, tais relatórios podem ser imprimidos em texto ou em tela. Cada relatório é gerado a partir de um determinado filtro que indica o que será selecionado da struct geral. Todas as sub-rotinas desse módulo são responsáveis por realizar a gravação de um determinado relatório em texto, as mesmas utilizam laços, funções if e strcmp para fazer a “filtração” e gerar o relatório que será gravado.

Obs: Não conseguimos implementar os filtros referentes à Vendas, Contas a Pagar e Movimentação do Caixa.

7. Importacao_exportacao.c

Este módulo é responsável por importar ou exportar dados específicos de um hotel para um arquivo XML. O módulo apenas as duas respectivas sub-rotinas:

- Função “void ExportarDados()”

Essa função é responsável por gravar as tabelas no arquivo e recebe como parâmetro as opções de tabelas escolhidas pelo usuário na função main e as demais structs que podem ser utilizadas e seus respectivos tamanhos. As opções referentes ao parâmetro estão armazenadas inicialmente em uma string que é posteriormente “quebrada” através da função strtok e cada string é armazenada em uma posição de um vetor. A gravação das tabelas em um arquivo XML é feita por meio da função fprintf que é implementada diversas vezes em um laço e grava as tabelas selecionadas.

- Função “int ImportarDados()”

Essa função é responsável por ler uma tabela específica de um arquivo XML existente e recebe como parâmetro a tabela escolhida pelo usuário no main e as demais structs que podem ser utilizadas e seus respectivos tamanhos. O primeiro switch utilizado na função verifica com um while a primeira identificação de uma tabela específica, como por exemplo na tag <tabela=hospede>, para isso é utilizado um fscanf que armazena em uma variável a string na posição do documento determinada pela função fseek que percorre uma posição por vez no mesmo. Posteriormente, utilizando uma função fseek são percorridas 58 posições do documento, por causa da diferença entre o fim da posição da tag que identifica a tabela e os registros, tal fato ocorre para todos os registros das tabelas. No segundo switch são armazenados todos os dados dos registros com a função fseek percorrendo as posições do documento. O while do segundo switch ocorre enquanto a função fscanf armazena a string registro, se a função fscanf armazenar uma string diferente, significa que os registros para determinada tabela terminaram.