UNIVERSIDADE PAULISTA – UNIP Ead

Projeto Integrado Multidisciplinar

Curso Superior de Tecnologia em

Análise e Desenvolvimento de sistemas

Arthur Ferreira Aguiar dos Santos – 2464871

Relatórios de Soluções Ambientais

São Paulo – SP

2024

**Resumo**

Software desenvolvido em C que permite empresas cadastrem seus clientes, criem relatórios sobre os serviços prestados e consultem essas informações de maneira eficiente. Os dados são armazenados em arquivos de texto ou CSV, permitindo fácil acesso e persistência das informações. O sistema oferece criptografia de dados dos colaboradores e clientes.

**Palavras-chave**: segurança. dados. eficiente.

**Abstract**

Software developed in C that allows companies to register their clients, generate reports on the services provided, and efficiently query this information. Data is stored in text or CSV files, enabling easy access and persistence. The system also offers encryption for employee and client data.

**Keywords:** security, data, efficiency.

**SUMÁRIO**

1. **Introdução..............................................................................5**
2. **Desenvolvimento...................................................................6**
3. **Criptografia.............................................................................6**
4. **Struct.......................................................................................7**
5. **Função....................................................................................8**
6. **Receber e armazenar dados do Usuário.............................9**
7. **Manipulação de dados em Arquivo....................................10**
8. **Acessando dados do vetor da Struct.................................12**
9. **Comparar input com string de um arquivo.......................14**
10. **Conclusão............................................................................16**
11. **Referencia............................................................................17**

**Introdução**

A busca por soluções que minimizem os impactos ambientais e promovam práticas mais responsáveis é um dos maiores desafios enfrentados pelo mercado. Diante desse cenário, este projeto propõe o desenvolvimento de um sistema que permita às empresas gerenciar e monitorar suas ações ambientais de maneira mais eficiente.

O software, permite a empresa cadastrar seus clientes, criar relatórios sobre resíduos tratados, não trados, preço da mão de obra e relatório global.

O objetivo é proporcionar uma ferramenta que não apenas facilite a organização desses dados, mas também permita análises precisas sobre o impacto ambiental das atividades realizadas. Além disso, a solução oferece a possibilidade de armazenar as informações de forma segura, com ênfase na transparência e na rastreabilidade dos processos.

**Desenvolvimento**

Passaremos pelos pontos mais importantes do projeto informando com exemplo pratico e ilustrativo sobre os tópicos solicitados na pesquisa.

**Criptografia**

 printf("Digite sua senha:");

                    while (1)// MASCARAR SENHA DO USUARIO

                    {

                        mascaraSenha = getch();

                        if (mascaraSenha == 13)

                        {

                            break;

                        }

                        else if(mascaraSenha == 8)

                        {

                            if (i > 0)

                            {

                                i--;

                                printf("\b \b");

                            }

                        }

                        else

                        {

                            senha[i] = mascaraSenha;

                            i++;

                            printf("\*"); // imprimi \* no lugar no input do usuario

                        }

                    }

Usamos switch casa para obter a escolha do usuário e definimos oque ira ser passado na opção escolhido, para a senha, utilizamos getch para retornar \* durante o input do usuário.

**Struct**

“A ideia de usar uma **struct** é permitir que, ao armazenar os dados de uma mesma entidade, isto possa ser feito com uma única variável. Por exemplo, se for preciso armazenar a *altura*, o *peso* e a idade de uma pessoa, pode-se criar uma **struct** chamada **Pessoa** e agrupar os dados em um único tipo de dado, conforme o exemplo a seguir.” - PUCRS

Utilizamos struct para criar os dados da Empresa e os dados do relatórios, foi de suma importância o uso de struct para resgatar a informação do usuário e utiliza-la posteriormente

typedef struct

{

    char nome[50];

    char nomeResponsavel[50];

    char nomedadosEmpresa[50];

    char razaoSocial[50];

    char nomeFantasia[50];

    char endereco[100];

    char email[20];

    char dadosRelevantes[20];

    char cnpj [14];

    char telefone [9];

    char dataAbertura[10];

}cadastrodadosEmpresa;

typedef struct

{

    char nomeResiduos[50];

    int quantidadesResiduosTratados;

    int quantidadesResiduosNaoTratados;

    float precoSugeridoDeCusto;

}ListaRelatorio; //listagem para o dados da empresa

typedef struct

{

    int Id;

    ListaRelatorio quantidadeRelatorios[99];

    float totalRelatorio;

}Relatorio; //relatorio dos dados da empresa

**Funções**

Uma função geralmente é criada para executar uma tarefa específica e seu nome geralmente reflete essa tarefa. Uma função contém declarações e instruções. Esta seção descreve como declarar, definir e chamar funções do C.

Uma das funções que que criamos foi especifica para limpar buffer de uma determinada ação de tipagem:

void limparBuffer() {

    while(getchar() != '\n');  // Remove todos os caracteres até o próximo '\n'

}

Alem de forneções ações especificas, as funções são muito utilizadas para facilitar a leitura e organização do código

**Receber e armazenar dados do Usuário**

Aprendemos que para ler um input do usuário usamos um comando do C chamado scanf, porem não é alocado um espaço especifico na memoria ao realizar dessa forma, por tanto, utilizaremos fgets para ler o input do usuário, e para podermos usar esse comando devemos adicionar em nosso código a biblioteca conio.h

printf("Digite o nome do Responsavel: ");

    fgets(dadosDaEmepresa.nomeResponsavel, sizeof(dadosDaEmepresa.nomeResponsavel), stdin);

    dadosDaEmepresa.nomeResponsavel[strcspn(dadosDaEmepresa.nomeResponsavel, "\n")] = '\0'; // recebe o input do usuario e aloca em um espaço

#include <conio.h> // biblioteca para utilizar get

**Manipulação de dados em Arquivo**

Em nosso projeto utilizamos formas para manipular dados para armazenar, ler e adicionar dados em arquivo txt e csv.

Para fazermos a manipulação devemos uma variável do tipo FILE e utilizar ponteiro \*, em seguida abrir o arquivo com fopen e dar um nome e sua propriedade, após isso devemos especificar oque gostaríamos de fazer:

R Este código permite apenas abrir um arquivo texto para leitura de seus dados. É necessário que o arquivo esteja presente no disco.

W Este código permite apenas abrir um arquivo de texto para escrita (gravação). Este código cria o arquivo para ser trabalhado. Caso o arquivo exista, este código recria o arquivo, ou seja, você perde o arquivo criado anteriormente. Deve ser usado com muito cuidado.

A Este código permite apenas abrir um arquivo de texto para escrita (gravação), permitindo acrescentar novos dados ao final dele. Caso o arquivo não exista, ele será então criado.

FILE \*salvarTxt = fopen("relatorio.txt", "a"); // abri o arquivo txt e insere os dados do vetor

                                fprintf(salvarTxt,"%s \n",relatorioVetor->quantidadeRelatorios->nomeResiduos);

                                fprintf(salvarTxt,"%d \n",relatorioVetor->quantidadeRelatorios->quantidadesResiduosTratados);

                                fprintf(salvarTxt,"%d \n",relatorioVetor->quantidadeRelatorios->quantidadesResiduosNaoTratados);

                                fprintf(salvarTxt,"%f \n",relatorioVetor->quantidadeRelatorios->precoSugeridoDeCusto);

                                fclose(salvarTxt);

FILE \*salvarCsv = fopen("relatorioCsv.csv", "a"); // abri um arquivo csv e insere os dados do vetor

                            fprintf(salvarCsv, "%s ", relatorioVetor->quantidadeRelatorios->nomeResiduos);

                            fprintf(salvarCsv, "%d ", relatorioVetor->quantidadeRelatorios->quantidadesResiduosTratados);

                            fprintf(salvarCsv, "%d ", relatorioVetor->quantidadeRelatorios->quantidadesResiduosNaoTratados);

                            fprintf(salvarCsv, "%f ", relatorioVetor->quantidadeRelatorios->precoSugeridoDeCusto);

                            fclose(salvarCsv);

FILE \*ponteiro\_arquivo;

    int i = 0;

    char linha[500];

    char cnpjLido[20];

    int encontrado = 0;

  ponteiro\_arquivo = fopen("cadastro de Cliente.txt", "r"); // abri arquivo txt

Para inserirmos string no arquivo de texto devemos utilizar o fprintf:

 fprintf(ponteiro\_arquivo, "Nome do Responsável: %s\n", dadosDaEmepresa.nomeResponsavel); // insere no arquivo txt os dados do usuario

    fprintf(ponteiro\_arquivo, "Nome da Empresa: %s\n", dadosDaEmepresa.nomedadosEmpresa);

    fprintf(ponteiro\_arquivo, "CNPJ: %s\n", dadosDaEmepresa.cnpj);

    fprintf(ponteiro\_arquivo, "Razão Social: %s\n", dadosDaEmepresa.razaoSocial);

    fprintf(ponteiro\_arquivo, "Nome Fantasia: %s\n", dadosDaEmepresa.nomeFantasia);

    fprintf(ponteiro\_arquivo, "Telefone: %s\n", dadosDaEmepresa.telefone);

    fprintf(ponteiro\_arquivo, "Endereço: %s\n", dadosDaEmepresa.endereco);

    fprintf(ponteiro\_arquivo, "E-mail: %s\n", dadosDaEmepresa.email);

    fprintf(ponteiro\_arquivo, "Data de Abertura: %s\n", dadosDaEmepresa.dataAbertura);

    fprintf(ponteiro\_arquivo, "Outros Dados: %s\n", dadosDaEmepresa.dadosRelevantes);

**Acessando dados do vetor da Struct**

Um dos desafios dessa pesquisa foi como podemos acessar os dados que temos de um vetor de um relatório que o usuário poderia passaria seus dados sem interromper a experiencia do usuário.

Para isso fizemos um loop dentro de outro loop percorrendo 2 variáveis e imprimindo na tela o input que foi passado pelo usuário e armazenado com fget.

for (int i = 0; i < id; i++)

                {

                    printf("Relatorio: %d", i+1);

                    for (int j = 0; j< 99 ; j ++)

                    {   //Imprime os input do usuario

                        printf("\nNome do residuo: %s\n", relatorioVetor[i].quantidadeRelatorios[j].nomeResiduos);

                        printf("\nQuantidade que foram tratados: %d\n", relatorioVetor[i].quantidadeRelatorios[j].quantidadesResiduosTratados);

                        printf("\nQuantidade sem tratamento: %d\n", relatorioVetor[i].quantidadeRelatorios[j].quantidadesResiduosNaoTratados);

                        printf("\nGasto  do tratamento: %f\n", relatorioVetor[i].quantidadeRelatorios[j].precoSugeridoDeCusto);

                        //condição para achar o maior residuo tratado

                        if(relatorioVetor[i].quantidadeRelatorios[j].quantidadesResiduosTratados > maiorResiduoTratado)

                        {

                            maiorResiduoTratado = relatorioVetor[i].quantidadeRelatorios[j].quantidadesResiduosTratados;

                        }

                        // condição para achar o menor residuo tratado

                        if(relatorioVetor[i].quantidadeRelatorios[j].quantidadesResiduosNaoTratados > menorResiduoTratado)

                        {

                            menorResiduoTratado = relatorioVetor[i].quantidadeRelatorios[j].quantidadesResiduosNaoTratados;

                        }

                        // condição para achar o maior gasto para tratamento para cada item tratado

                        if(relatorioVetor[i].quantidadeRelatorios[j].precoSugeridoDeCusto > maiorGastoTratamento)

                        {

                            maiorGastoTratamento = relatorioVetor[i].quantidadeRelatorios[j].precoSugeridoDeCusto;

                        }

                        //condição de parada

                        if(relatorioVetor[i].quantidadeRelatorios[j + 1].quantidadesResiduosTratados == 0 && relatorioVetor[i].quantidadeRelatorios[j + 1].quantidadesResiduosNaoTratados == 00)

                        {

                            break;

                        }

                    }

                }

**Comparar input com string de um arquivo**

Criamos uma função onde é preciso passar o input de CNPJ como Char, abrimos o arquivo de texto como “r” para apenas ler os dados que já estavam armazenados no txt. Em seguida um loop recebendo um char linha para receber os dados e fazendo referencia a variável FILE, temos a condição de verificação e em caso positivo incrementa em uma nova variável e terminamos a função

void consultarCliente(char cnpjInformado[])// recebe uma variavel de string

{

    FILE \*ponteiro\_arquivo;

    int i = 0;

    char linha[500];

    char cnpjLido[20];

    int encontrado = 0;

    ponteiro\_arquivo = fopen("cadastro de Cliente.txt", "r"); // abri arquivo txt

    //fgets(dadosDaEmpresa.cnpj,sizeof(dadosDaEmepresa.cnpj),stdin);

    //dadosDaEmepresa.cnpj[strcspn(dadosDaEmepresa.cnpj,"\n")] ='\0';

    while (fgets(linha, sizeof(linha), ponteiro\_arquivo)) // loop para verificar cnpj

    {

        if(strstr(linha, "CNPJ:") != NULL)

        {

            sscanf(linha, "CNPJ: %s", cnpjLido);

            if(strcmp(cnpjLido, cnpjInformado) == 0)

            {

            encontrado = 1;

            break;

            }

        }

    }

    if(encontrado)

    {

        printf("Cliente já cadastrado");

    }

    else

    {

        printf("Cliente não possui cadastro");

    }

    fclose(ponteiro\_arquivo);

}

Conclusão

Em um cenário onde a sustentabilidade se torna cada vez mais uma prioridade para as empresas, o desenvolvimento de soluções tecnológicas que facilitem a gestão e monitoramento de ações ambientais é essencial. O sistema proposto neste projeto visa atender a essa necessidade, oferecendo uma plataforma eficiente para o cadastro de clientes e geração de relatórios detalhados sobre resíduos tratados, custos e impactos ambientais. Ao proporcionar uma organização mais eficiente dos dados,. Além disso, com o armazenamento seguro das informações e a garantia de transparência e rastreabilidade, a solução fortalece a responsabilidade ambiental das empresas, permitindo que elas se alinhem cada vez mais às demandas de um mercado que valoriza práticas sustentáveis.

Referencias

Documentação da linguagem C – Microsoft

Structs - PUCRS