

UNIVERSIDADE PAULISTA – UNIP EaD
Projeto integrado Multidisciplinar – PIM IV

Curso Superior De Tecnologia em
Análise E Desenvolvimento De Sistemas

Victor Schiavo Pereira – RA 2465283

DESENVOLVIMENTO DE SOFTWARE EM C
PARA STARTUP DE SOLUÇÕES AMBIENTAIS
PIM IV

Barueri – São Paulo

2024

Victor Schiavo Pereira – RA 2465283

**DESENVOLVIMENTO DE SOFTWARE EM C
PARA STARTUP DE SOLUÇÕES AMBIENTAIS
PIM IV**

**Projeto Integrado Multidisciplinar em
Análise e Desenvolvimento de Projetos**

**Projeto Integrado Multidisciplinar para obtenção do
título de tecnólogo em (nome do curso), apresentado
à Universidade Paulista – UNIP EaD.**

Barueri – São Paulo

2024

RESUMO

Este trabalho tem como objetivo principal aplicar os conhecimentos adquiridos ao longo do curso de **Tecnologia em Análise e Desenvolvimento de Sistemas** e avaliar como esses conceitos se traduzem no mundo real. Para o desenvolvimento do sistema, foram utilizados conceitos de **Engenharia de Software**, adotando a metodologia **Cascata**, uma vez que os requisitos do projeto estavam claros e bem definidos desde o início. Além disso, foram empregadas ferramentas e técnicas provenientes da disciplina de **Linguagens e Técnicas de Programação**, como laços de repetição, ponteiros, funções e estruturas de dados, para alcançar os resultados desejados.

O sistema desenvolvido é focado na gestão empresarial, com funcionalidades que permitem o cadastro de empresas, o acompanhamento mensal do tratamento de resíduos sólidos e a geração de relatórios detalhados. Durante o desenvolvimento, foram explorados conceitos de manipulação de arquivos e boas práticas de programação, garantindo que o sistema fosse eficiente e atendesse aos objetivos propostos.

Os resultados demonstraram que a aplicação prática dos conceitos estudados possibilitou a criação de um sistema funcional e bem estruturado, reafirmando a importância de unir teoria e prática no contexto do desenvolvimento de sistemas.

Palavras-chave: metodologia cascata, engenharia de software, programação em C, aplicação prática, gestão empresarial.

ABSTRACT

*The primary goal of this project is to apply the knowledge acquired throughout the **Technology in Systems Analysis and Development** course and evaluate how these concepts translate into real-world scenarios. The development of the system leveraged concepts from **Software Engineering**, adopting the **Waterfall** methodology, as the project requirements were clear and well-defined from the beginning. Additionally, tools and techniques from the **Programming Languages and Techniques** course, such as loops, pointers, functions, and data structures, were utilized to achieve the desired outcomes.*

The developed system focuses on business management, providing functionalities for company registration, monthly tracking of solid waste processing, and the generation of detailed reports. During the development process, concepts of file manipulation and programming best practices were explored, ensuring that the system is both efficient and meets the proposed objectives.

The results demonstrated that the practical application of the studied concepts enabled the creation of a functional and well-structured system, reaffirming the importance of integrating theory and practice in the context of system development.

Keywords: *Waterfall methodology, software engineering, C programming, practical application, business management.*

SUMÁRIO

1. INTRODUÇÃO	6
2. ELABORAÇÃO DO PROGRAMA	7
2.1 Planejamento	7
2.2 Codificação e Telas do Sistema	8
2.3 Proteção de dados (LGPD).....	14
2.4 – Funções do Sistema (Código)	16
3. CONCLUSÃO.....	20
4. REFERÊNCIAS.....	21

1. INTRODUÇÃO

O cenário proposto para este projeto gira em torno de uma startup de soluções ambientais que busca gerenciar o cadastro e as operações de indústrias responsáveis pelo tratamento de resíduos e gases provenientes de seus processos produtivos. Com o objetivo de atender à crescente demanda, a startup necessita de um sistema computacional que permita o gerenciamento eficiente das informações, garantindo conformidade com a LGPD e oferecendo relatórios detalhados aos seus clientes e órgãos governamentais.

Diante desse contexto, o projeto foi desenvolvido com base em metodologias aprendidas ao longo do curso, com o objetivo de aplicar na prática os conceitos de análise, planejamento e implementação de sistemas. O primeiro passo foi a criação de um **Fluxograma**, utilizando a ferramenta Miro + ChatGPT, para mapear o fluxo de trabalho, os processos principais e as interações previstas no sistema. Esse diagrama foi essencial para visualizar o escopo do projeto e facilitar sua execução de forma organizada e alinhada às exigências do caso apresentado.

2. ELABORAÇÃO DO PROGRAMA

2.1 Planejamento

O uso de ferramentas de IA se mostrou muito útil para criar protótipos e esboços antes da implementação, reforçando novamente, a importância de metodologia para desenvolver *software* com qualidade e velocidade, o que imagino que implicaria também em custos reduzidos.

Inicialmente trabalhamos com uma visão geral do que o sistema deveria fazer e geramos alguns **fluxogramas** para guiar o desenvolvimento, na figura abaixo podemos ver um dos esboços que foram gerados com ajuda de IA:

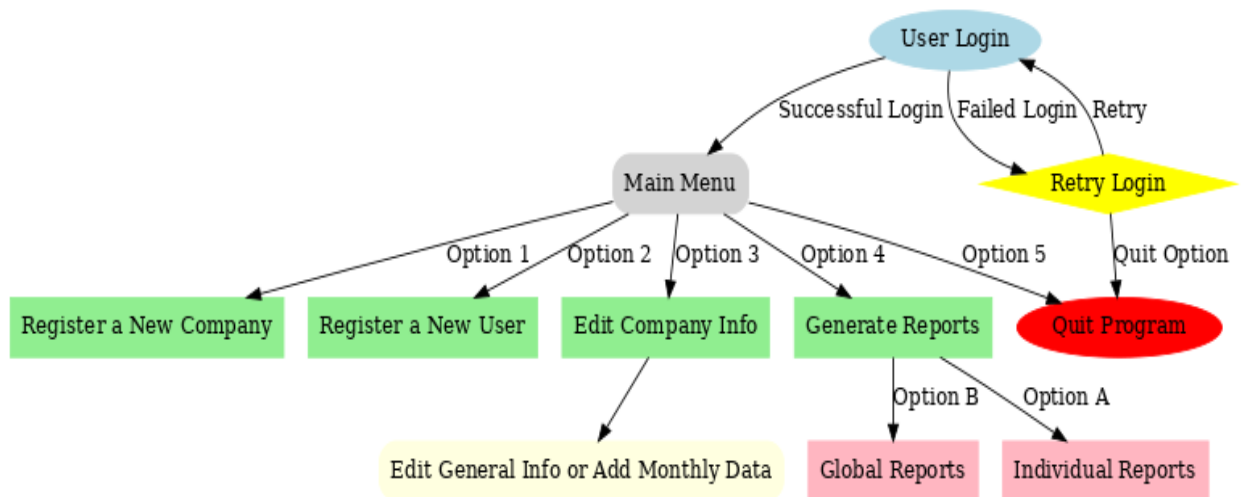


Figura 1 - Fluxograma rascunho – Autoria Própria

Uma vez definido o fluxograma, usamos a ferramenta MIRO para montar uma visão melhorada e mais profissional dos fluxos, conforme figura a abaixo:

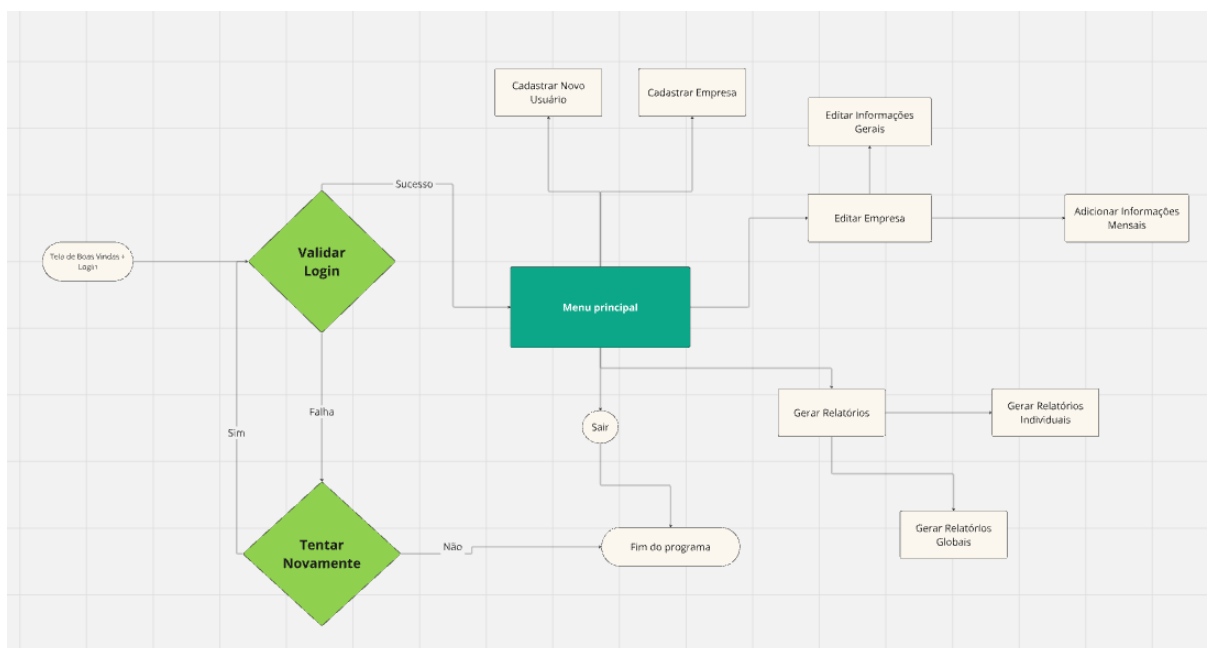


Figura 2 - Fluxograma Final - Autoria Própria

A versão completa do fluxograma pode ser exibida no seguinte endereço:

https://miro.com/app/board/uXjVLATTpTc=?share_link_id=492440734483

2.2 Codificação e Telas do Sistema

Definido os processos com ajuda do fluxograma, entramos na fase de codificação.

Começamos criando os arquivos necessários para o nosso programa, sendo estes:

- SGA.C – Arquivo Principal que contém toda lógica do programa
- Usuarios.txt – Arquivo que contém as informações de usuário
- Empresas.txt – Arquivo que contém as informações gerais das empresas, bem como as informações referente ao processamento de resíduos mensal e seus custos.
- Relatorios.txt – Arquivo que contém os relatórios exportados em uma formatação simples.
- Relatorios.csv – Arquivo que contém os relatórios exportados no formato “.CSV,” amplamente reconhecido por diversos outros programas para importação.

A codificação foi feita com a ferramenta “**VSCODE**”, em conjunto com a ferramenta “**GIT**” e o site www.github.com para facilitar a colaboração com outros desenvolvedores.

A primeira funcionalidade desenvolvida foi a tela inicial com opção de Login:

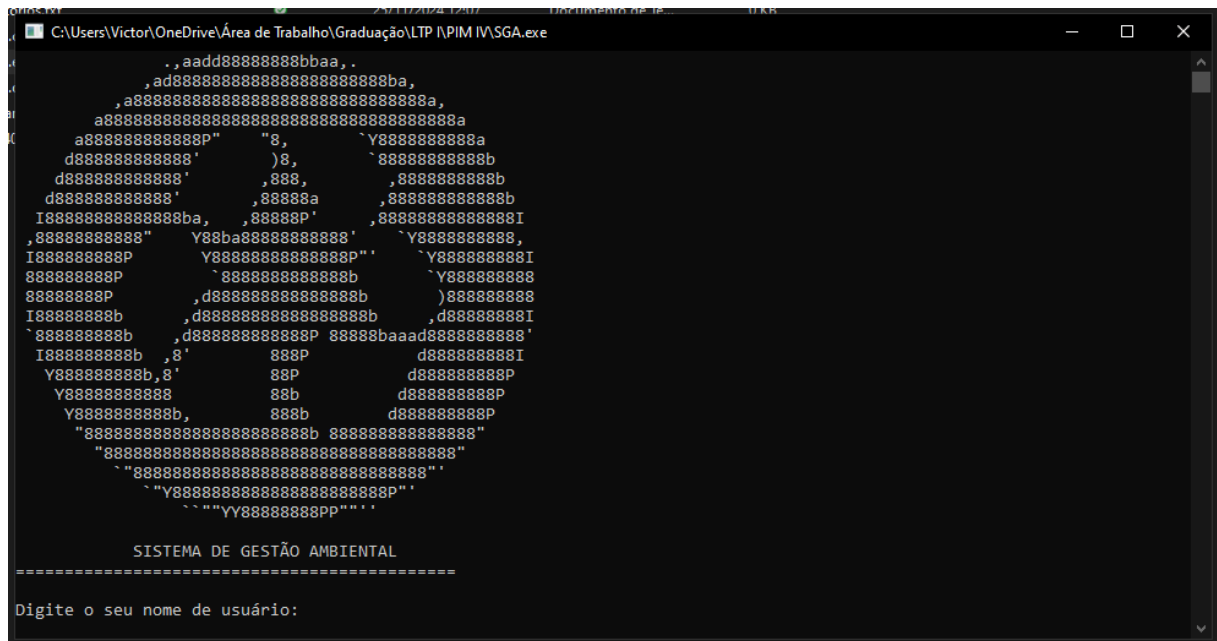


Figura 3 - Tela de Login - Autoria própria

Aqui o usuário deve inserir, sequencialmente, o nome de usuário e a senha.

Para efeitos de testes, foi criado um usuário chamado “**admin**” e senha “**snafu**”. Uma vez logado no sistema, o usuário se depara com a seguinte menu:

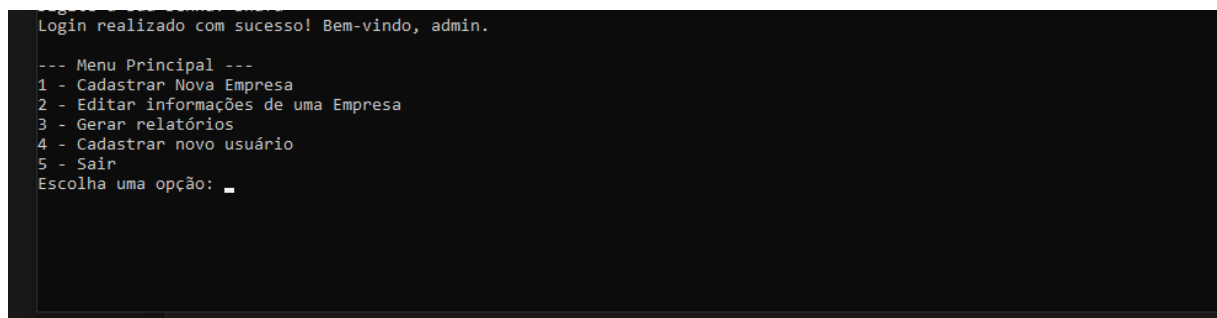


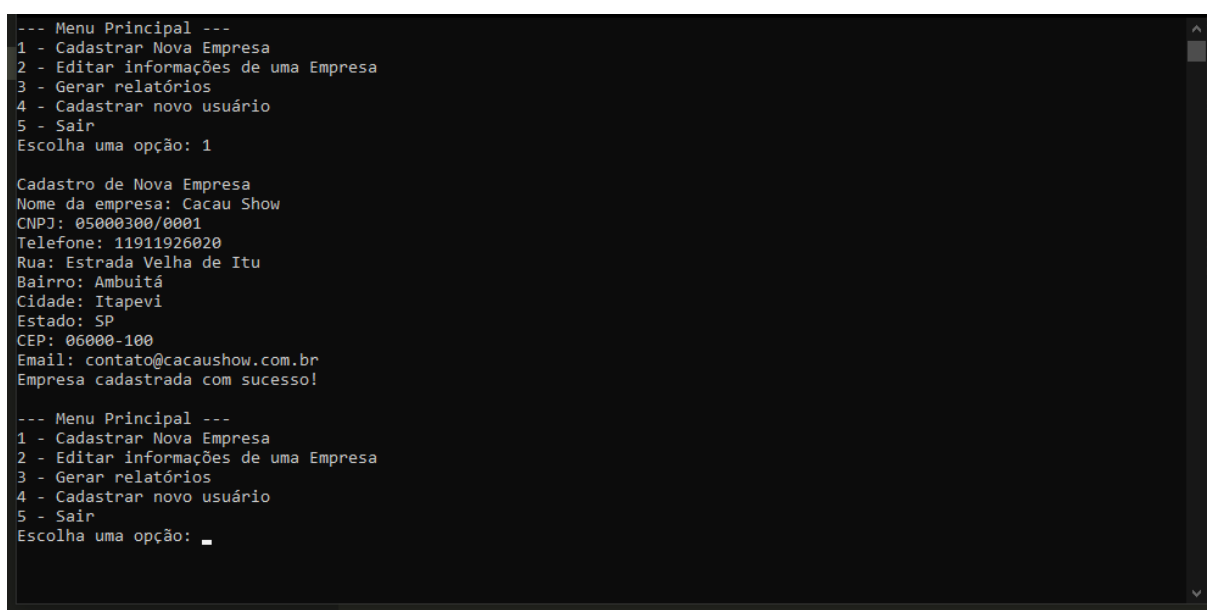
Figura 4 - Menu de Login – Autoria própria.

A opção “**1 – Cadastrar Nova Empresa**” irá solicitar que o usuário digite todas as informações relevantes de uma empresa de maneira sequencial.

As informações solicitadas são:

- Nome
- CNPJ
- Telefone
- Rua
- Bairro
- Cidade
- Estado
- CEP
- Email

Uma vez digitada, as informações são armazenadas no arquivo “empresas.txt” e um **resumo das informações** é exibido na tela, voltando para o menu principal.



```
--- Menu Principal ---
1 - Cadastrar Nova Empresa
2 - Editar informações de uma Empresa
3 - Gerar relatórios
4 - Cadastrar novo usuário
5 - Sair
Escolha uma opção: 1

Cadastro de Nova Empresa
Nome da empresa: Cacau Show
CNPJ: 05000300/0001
Telefone: 11911926020
Rua: Estrada Velha de Itu
Bairro: Ambuitá
Cidade: Itapevi
Estado: SP
CEP: 06000-100
Email: contato@cacaushow.com.br
Empresa cadastrada com sucesso!

--- Menu Principal ---
1 - Cadastrar Nova Empresa
2 - Editar informações de uma Empresa
3 - Gerar relatórios
4 - Cadastrar novo usuário
5 - Sair
Escolha uma opção: █
```

Figura 5 - Resumo do Cadastro - Autoria própria

Com uma empresa cadastrada, é possível agora lançar as informações de tratamento de resíduos e custo associado por mês, para isso, o usuário deve escolher a opção número 2.

```

--- Menu Principal ---
1 - Cadastrar Nova Empresa
2 - Editar informações de uma Empresa
3 - Gerar relatórios
4 - Cadastrar novo usuário
5 - Sair
Escolha uma opção: 2

--- Empresas registradas ---
1 - wasteful
2 - wastefulll
3 - Cacau Show

Escolha o número da empresa que deseja editar ou digite 0 para retornar ao menu principal:

```

Figura 6 - Menu para editar empresas

Uma lista é exibida com todas as empresas cadastrada e um número deve ser escolhido para prosseguir.

```

Você escolheu: Cacau Show

Empresa encontrada: Nome: Cacau Show, CNPJ: 05000300/0001, Telefone: 11911926020, Endereço: Estrada Velha de Itu, Bair
Escolha uma opção:
1 - Atualizar informações gerais
2 - Adicionar informações mensais (Tratamento de Lixo e Custo)
3 - Retornar ao menu principal
Escolha uma opção:

```

Figura 7 - Continuação do Menu para editar empresas

Aqui, o usuário pode editar as informações já inseridas previamente, ou adicionar as informações de resíduos e custo.

As informações mensais são inseridas em uma formatação especial para que seja possível resgatar o relatório posteriormente:

```

Empresa encontrada: Nome: wasteful, CNPJ: 84758347, Telefone: 11111111, Endereço: Ok dos OK, Bairro: Super OK, Cidade:
Escolha uma opção:
1 - Atualizar informações gerais
2 - Adicionar informações mensais (Tratamento de Lixo e Custo)
3 - Retornar ao menu principal
Escolha uma opção: 2
Digite o mês (ex: Novembro): Janeiro
Digite a quantidade de lixo tratado (em toneladas): 300
Digite o custo estimado (em reais): 9800
Informações mensais adicionadas com sucesso!

```

Figura 8 - Tela informando o sucesso da operação

No arquivo “empresas.txt” a informação adota a seguinte estrutura:

```

####END_COMPANY###
Nome: wastefull1, CNPJ: 12323423, Telefone: 1214325, Endereço: OK dos Oksss, Bairro: OKkkkkkkk, Cidade:
Osasco, Estado: SP, CEP: 99911119, Email: ok@ok.com
- Mês: November, Tratamento: 300.00 toneladas, Custo: 5000.00 reais
- Mês: Agosto, Tratamento: 50.00 toneladas, Custo: 3600.00 reais
- Mês: Janeiro, Tratamento: 300,00 toneladas, Custo: 9800,00 reais
####END_COMPANY###

```

Figura 9 - Estrutura do arquivo após inserção dos dados mensais

Cada mês é identificado pelo **caractere hífen**, isso adotado para que seja mais fácil gerar os relatórios na próxima etapa. A linha “**####END_COMPANY###**” é inserida para que a função gerar relatórios consiga descobrir com facilidade se está no limite das informações de uma empresa ou se está “invadindo” a próxima empresa.

No geral, pudemos observar que a estruturação correta do arquivo “.txt” é essencial para resgatar os dados com confiabilidade depois, de modo que caracteres como “**Enter**”, espaços extras, pontuações ou ausência de pontuações, causam erros na hora de “tabular” as informações em tela ou arquivo.

Essa foi uma etapa que exigiu muitos testes e a cada nova variação no modelo de digitação, percebíamos novos erros na hora de gerar o relatório ou salvar as informações em texto. Fica nítido, dessa forma, **a importância de testes e contato frequente com os clientes/stakeholders** para que erros não passem despercebidos. Esse conceito foi enfatizado em diversos momentos pela disciplina de **Engenharia de Software**.

Nesse caso simulado, os testes eram feitos com familiares ou até mesmo pelo desenvolvedor, mas é evidente que um teste por pessoas que usariam o sistema no dia-a-dia seria muito mais rico e com maior aproveitamento.

A última opção do sistema, “**3 – Gerar relatórios**” leva o usuário para a seguinte tela:

```

--- Menu Principal ---
1 - Cadastrar Nova Empresa
2 - Editar informações de uma Empresa
3 - Gerar relatórios
4 - Cadastrar novo usuário
5 - Sair
Escolha uma opção: 3

--- Menu de Relatórios ---
1 - Relatório por Empresa
2 - Relatórios Globais
3 - Retornar ao menu principal
Escolha uma opção: 1

```

Figura 10 - Tela para gerar relatórios – autoria própria

No momento da conclusão deste trabalho, somente o **Relatório por Empresa em tela foi desenvolvido**, encontramos diversos desafios técnicos na elaboração dos demais relatórios que em retrospectiva, acreditamos que ocorreu pela forma como os dados foram estruturados todos em um único arquivo de texto.

Ao escolher essa opção, o usuário terá a seguinte tela:

```

--- Menu de Relatórios ---
1 - Relatório por Empresa
2 - Relatórios Globais
3 - Retornar ao menu principal
Escolha uma opção: 1

Empresa: wasteful
CNPJ: 84758347
Telefone: 1111111

Dados Mensais:
Mês: November
Tratamento de Lixo: 300,00 toneladas
Custo: R$ 0,00
-----
Mês: Agosto
Tratamento de Lixo: 50,00 toneladas
Custo: R$ 0,00
-----

Empresa: wastefulll
CNPJ: 12323423
Telefone: 1214325

Dados Mensais:
Mês: November
Tratamento de Lixo: 300,00 toneladas
Custo: R$ 0,00
-----
Mês: Agosto
Tratamento de Lixo: 50,00 toneladas
Custo: R$ 0,00
-----
Mês: Janeiro
Tratamento de Lixo: 300,00 toneladas
Custo: R$ 9800,00
-----

Empresa: Cacau Show
CNPJ: 05000300/0001
Telefone: 11911926020

Dados Mensais:

--- Menu de Relatórios ---
1 - Relatório por Empresa
2 - Relatórios Globais
3 - Retornar ao menu principal
Escolha uma opção:

```

Figura 11 - Tela com resumo dos relatórios

Essa tela fornece uma visão geral de todas as empresas cadastradas e os seus lançamentos mensais, até 100 empresas podem ser visualizadas.

2.3 Proteção de dados (LGPD)

Como prova de conceito, aplicamos também uma função de criptografia de senha, que pode ser aplicada em qualquer informação sensível.

O modelo de criptografia escolhido, foi a Cifra de César.

A Cifra de César é um tipo simples de criptografia por substituição, na qual cada letra do texto é deslocada um número fixo de posições no alfabeto. Por exemplo, se utilizarmos um deslocamento de 3, a letra "A" se tornaria "D", a letra "B" se tornaria "E", e assim por diante. Esse tipo de cifra é chamado de "Cifra de Substituição" porque cada letra do texto original é substituída por outra, de acordo com a posição definida pelo deslocamento.

No sistema desenvolvido, a Cifra de César foi aplicada para criptografar as senhas dos usuários, aumentando a segurança das informações armazenadas. Para isso, utilizamos a seguinte lógica:

Criptografia: Quando um novo usuário é registrado no sistema, sua senha é criptografada utilizando a cifra de César com um deslocamento definido (neste caso, um valor de 3). Cada caractere da senha é deslocado para a frente no alfabeto, mantendo o padrão de letras (maiúsculas ou minúsculas). A senha criptografada é então salva no arquivo de usuários.

Tratamento de Caracteres: A cifra de César foi implementada para lidar corretamente tanto com letras maiúsculas quanto minúsculas. Além disso, caracteres não alfabéticos (como espaços) não são modificados durante o processo de criptografia e descriptografia.

A utilização da Cifra de César para criptografar as senhas dos usuários oferece uma camada básica de segurança ao sistema, dificultando o acesso não autorizado, uma vez que as senhas não são armazenadas em texto claro. **No entanto, é importante ressaltar que a cifra de César é considerada uma técnica de criptografia simples e, em contextos mais sensíveis, métodos mais avançados de criptografia devem ser utilizados para garantir maior segurança.**

Podemos observar a Cifra de César aplicada no exemplo a seguir:

```
Digite o seu nome de usuário: admin
Digite a sua senha: snafu
Login realizado com sucesso! Bem-vindo, admin.

--- Menu Principal ---
1 - Cadastrar Nova Empresa
2 - Editar informações de uma Empresa
3 - Gerar relatórios
4 - Cadastrar novo usuário
5 - Sair
Escolha uma opção: 4

Cadastro de Novo Usuário
Nome de usuário: tiburcio
Senha: supersecreta
```

Figura 12- Criação de um usuário com a senha "supersecreta"

O resultado no arquivo "usuários.txt" é uma senha embaralhada, conforme podemos ver na figura a seguir:

```
SGA.c 3, M  usuarios.txt M X  empresas.txt M  relatorios.txt U
usuarios.txt
1  admin snafu
2  tiburciodev vhfuhwd
3  sergerson rsv
4  tiburcio vxshuvhfuhwd
5  |
```

Figura 13 - Cifra de César aplicada

2.4 – Funções do Sistema (Código)

Listamos aqui as principais funções no código do sistema e sua funcionalidade esperada. Note que foi adotado um padrão de escrita em inglês para as funções e variáveis, mas os comentários estão todos em português.

```
// Protótipos das funções, para que possam ser invocadas mesmo que sua definição venha após o ponto
// chamada.
bool login(const char *username, const char *password);
void displayWelcomeMessage();
void mainMenu();
void registerCompany();
void editCompany();
void openReportsMenu();
void registerUser();
FILE *openFile(const char *filename, const char *mode);
void closeFile(FILE *file);
void getInput(const char *prompt, char *buffer, size_t size);
void listCompanies(char companyNames[][MAX_LENGTH], int *companyCount);
int selectCompany(int companyCount);
int reportByCompany();
bool processCompanyEdit(FILE *file, FILE *tempFile, const char *targetCompany);
void updateCompanyInfo(FILE *tempFile);
int loadCompanies(const char *filename, Company companies[], int maxCompanies);
void caesarEncrypt(char *input, char *output, int shift);
void caesarDecrypt(char *input, char *output, int shift);
```

Figura 14 - Protótipos de funções

As funções foram inicialmente declaradas como protótipos para que possam ser invocadas no código mesmo que tenha sido criada após a sua chamada. Essa técnica permite uma ordem um pouco menos rígida na codificação, porém causa mais trabalho para manter as funções, uma vez que agora precisamos alterar as funções em dois locais caso elas sofram alterações no nome ou na tipagem.

Função Login (const char *username, const char *password)

Cuida da lógica de login, recebe duas variáveis do tipo char que serão comparadas com o arquivo “usuarios.txt” para validar o acesso. A função usa ponteiros pois é chamada dentro da função main. Retornar verdadeiro para login com sucesso e falso para login malsucedido.

Função displayWelcomeMessage ()

Função para permitir a exibição da “imagem” de boas-vindas, a função não possui nenhum parâmetro e não há retornos.

Função mainMenu()

Gerencia o comportamento do menu principal, por essa função, fazemos chamada para as outras funções principais do programa. Foi utilizado o **switch** para cuidar das opções.

Função registerCompany()

A função registerCompany é responsável pelo processo de cadastro de uma nova empresa no sistema. Ela solicita ao usuário as informações relevantes sobre a empresa, como nome, CNPJ, telefone, endereço (rua, bairro, cidade, estado, CEP) e email, e em seguida, armazena esses dados em um arquivo chamado "empresas.txt". As informações são registradas de forma organizada, com cada campo sendo separado por vírgulas. Além disso, é adicionado um marcador "###END_COMPANY###" para separar o cadastro de uma empresa de outro. A função utiliza o modo "a" para abrir o arquivo, garantindo que novos registros sejam adicionados ao final do arquivo existente, sem sobrescrever os dados anteriores. Ao final, a função exibe uma mensagem indicando que o cadastro foi realizado com sucesso.

Função registerUser()

Responsável pelo processo de cadastro de um novo usuário no sistema. Ela solicita ao usuário as informações de nome de usuário e senha. A senha fornecida é então criptografada utilizando o método de cifra de César. Após a criptografia, o nome de usuário e a senha criptografada são armazenados no arquivo "usuarios.txt", no formato de duas strings separadas por espaço. O arquivo é aberto no modo "a+" para que os novos cadastros sejam adicionados ao final do arquivo existente sem sobrescrever os dados anteriores.

Função editCompany()

A função editCompany permite editar os dados de uma empresa cadastrada no sistema. Ela exibe uma lista de empresas e permite que o usuário selecione uma para editar. Após a seleção, os dados da empresa escolhida são lidos e processados, e as alterações são feitas em um arquivo temporário. Se a empresa for encontrada e editada com sucesso, o arquivo original é substituído pelo temporário. Caso contrário, o arquivo temporário é removido e uma mensagem de erro é exibida.

**Função listCompanies(char companyNames[][MAX_LENGTH], int
*companyCount)**

Responsável por listar todas as empresas registradas no sistema. Ela lê o arquivo de empresas (COMPANY_FILE) linha por linha, procurando por linhas que contenham o nome de uma empresa. Quando encontra um nome, ele extrai o nome da empresa e o armazena em um array de strings fornecido como parâmetro. O nome das empresas é exibido no console, e o total de empresas encontradas é atualizado

Função selectCompany(int companyCount)

Solicita ao usuário que escolha uma empresa a ser editada, fornecendo o número da empresa com base na listagem apresentada anteriormente. Caso o usuário insira um número inválido (fora do intervalo de empresas disponíveis ou um número negativo), a função exibe uma mensagem de erro e solicita a escolha novamente. A função retorna o número da opção escolhida pelo usuário.

Função addMonthlyInfo(FILE *file, FILE *tempFile, const char *targetCompany)

Adiciona informações mensais de tratamento de lixo a uma empresa específica no arquivo. A função percorre o arquivo até encontrar a empresa alvo e, ao alcançar o final de seus dados, permite ao usuário inserir o mês, quantidade de lixo tratado e custo. Esses dados são então inseridos antes da linha de término da empresa (###END_COMPANY###), e o arquivo temporário é atualizado.

**Função processCompanyEdit (FILE *file, FILE *tempFile, const char
*targetCompany)**

Processa a edição de informações de uma empresa no arquivo, permitindo ao usuário escolher entre atualizar informações gerais ou adicionar dados mensais (tratamento de lixo e custo). Se a opção for adicionar informações mensais, a função addMonthlyInfo é chamada para inserir os dados antes de finalizar o registro da empresa. A função percorre o arquivo até encontrar a empresa alvo, aplica a alteração solicitada e escreve as mudanças em um arquivo temporário.

Função updateCompanyInfo(FILE *tempFile)

Atualiza as informações gerais de uma empresa. O usuário insere novos dados (por exemplo, "Nome: Empresa A, Endereço: Rua XYZ"), que são gravados no arquivo temporário. Essas informações substituirão os dados antigos no arquivo de empresas após o processamento da edição.

Função loadCompanies (const char *filename, Company companies[], int maxCompanies)

Carrega os dados das empresas a partir de um arquivo especificado e armazena em um array de estruturas Company. O arquivo de entrada deve estar no formato esperado, contendo informações sobre as empresas e seus respectivos dados mensais.

Primeiro, a função tenta abrir o arquivo no modo leitura. Se o arquivo for aberto com sucesso, ela inicia o processamento das linhas, buscando as informações de cada empresa. Para cada empresa encontrada, a função lê e armazena os dados gerais, como nome, CNPJ, telefone, endereço, bairro, cidade, estado, CEP e e-mail, em uma estrutura Company. Em seguida, a função lê os dados mensais associados à empresa até encontrar o marcador ####END_COMPANY#### ou a linha de outra empresa. Esses dados mensais são armazenados nas variáveis correspondentes dentro da estrutura Company.

A função limita o número de meses a 12, garantindo que apenas até 12 meses de dados sejam armazenados para cada empresa. Quando o número máximo de empresas (maxCompanies) for alcançado, a função interrompe a leitura. Ao final, a função retorna o número de empresas que foram carregadas com sucesso. Caso haja algum erro ao abrir o arquivo, a função retorna 0.

Função reportByCompany()

Gera um relatório em tela com base em um array, até um máximo de 100 empresas, essa função depende integralmente da função loadCompanies() para popular o array Company e listar as empresas. Usa um loop inicial para percorrer as empresas e dentro desse loop, um segundo loop para percorrer as informações mensais, como se fosse uma tabela.

Após essa operação, uma formatação básica é exibida na tela listando todas as empresas encontradas, bem como suas informações mensais

Função caesarEncrypt(char *input, char *output, int shift)

Responsável por criptografar uma string com base em um parâmetro chamado *shift*, o número informado em shift é o deslocamento que será feito das letras, por exemplo A com shift 1, se torna B, com shift 2, se torna C e assim sucessivamente.

A função contém um IF, para verificar se o caractere deve ou não ser deslocado (caracteres especiais são desprezados) e retorna a string criptografada.

.

3. CONCLUSÃO

Durante a elaboração deste trabalho, pudemos aplicar diversos conceitos abordados durante as disciplinas de **Linguagens e técnicas de programação e Engenharia de Software**.

Mesmo em um projeto relativamente simples, com requisitos muito bem definidos, podemos perceber que a manutenção de código, erros e imprevistos e usabilidade são premissas que - caso não sejam pensadas e consideradas desde a concepção do projeto - podem facilmente sair do controle e tornar um produto de *software caro, lento e complexo*.

4. REFERÊNCIAS

COSTA, Ivanir. *Engenharia de software*. São Paulo: Editora Sol, 2014. 168 p., il. (Cadernos de Estudos e Pesquisas da UNIP, Série Didática, ano XIX, n. 2-061/14). ISBN 1517-9230.

IVO, Olavo. *Linguagens e técnicas de programação*. São Paulo, 2014. 488 p., il. (Cadernos de Estudos e Pesquisas da UNIP, Série Didática, ano XX, n. 2-082/14). ISSN 1517-9230.