

Relatório

# Gestão Propostas de Crédito

Engenharia Informática

Programação 1

Instituto Politécnico de Viana do Castelo



*Submetido por:*

**André Brandão 26244**

**Diogo Campos 24888**

*Sob a orientação de*

**Prof. Pedro Coutinho**

p.coutinho@estg.ipvc.pt

**Prof. Domingos Gomes**

domingosg@estg.ipvc.pt

**2020 - 2021**

Escola Superior de Tecnologia e Gestão

Instituto Politécnico de Viana do Castelo

Viana do Castelo - 4900-348

# [Gestão Propostas de Crédito]

André Brandão, Diogo Campos

4 de novembro de 2021

## **Resumo**

O presente programa pretende apoiar os seus utilizadores no processo de gestão de propostas de crédito de uma instituição financeira.

Este projecto foi desenvolvido no âmbito da unidade curricular de Programação 1 e contempla um conjunto de objetivos que são descritos nesta secção. Um dos objetivos deste projecto permitiu aos alunos aplicar os conhecimentos adquiridos ao longo das aulas da unidade curricular. Desta forma, os alunos conseguiram ter uma perceção prática, isto é, uma aplicação do mundo real onde foi possível através dos conhecimentos adquiridos desenvolver uma aplicação com utilidade real.

Tem-se também como objetivo deste projecto o desenvolvimento de capacidades de trabalho em grupo. Os alunos tiveram de partilhar soluções e debater problemas, entre si, de forma a chegar à melhor solução e ao algoritmo mais eficiente.

Em termos de objetivos do trabalho em si, pretende-se que a aplicação permita operações básicas de inserir, alterar, remover, pesquisar/listas (aka CRUD) sobre os dados das propostas de crédito.

Em termos de perfis de utilizadores, o sistema de informação pode possuir vários utilizadores, sendo estes do administrador ou do tipo analista. Toda a informação gerada através da aplicação será organizadamente guardada em ficheiro do tipo binário e texto

O presente documento apresenta abordagem implementada pelo grupo de trabalho e os resultados obtidos com essa implementação.

# Conteúdo

	<b>1</b>	<b>Revisão Teórica</b>	<b>3</b>
	1.1	Ficheiros de texto e ficheiros binários . . . . .	3
	1.2	Algoritmos de ordenação e pesquisa . . . . .	3
5	1.3	Alocação dinâmica de memória . . . . .	3
	1.4	Listas ligadas simples . . . . .	4
	1.5	Listas duplamente ligadas . . . . .	4
	1.6	Queues . . . . .	4
	<b>2</b>	<b>Desenvolvimento</b>	<b>5</b>
10	2.1	Compreensão do enunciado . . . . .	6
	2.2	Fluxograma . . . . .	7
	2.3	Código em linguagem C . . . . .	8
	2.3.1	Organização . . . . .	8
	2.3.2	Estrutura de dados utilizada . . . . .	9
15	2.3.3	Algoritmos utilizados . . . . .	9
	2.3.4	Problemas encontrados . . . . .	10
	<b>3</b>	<b>Conclusão</b>	<b>11</b>
	<b>A</b>	<b>Imagens de Estruturas</b>	<b>13</b>
	<b>B</b>	<b>Manual do Utilizador</b>	<b>16</b>
20	B.1	Administrador . . . . .	17
	B.2	Analista . . . . .	29

# 1 Revisão Teórica

## 1.1 Ficheiros de texto e ficheiros binários

Na linguagem C existem dois tipos de ficheiros:

25 1. Ficheiros de texto

2. Ficheiros binários

Por um lado, um ficheiro é considerado de texto quando é constituído por caracteres percetíveis para o ser humano, ou seja, quando o ficheiro é composto por caracteres da tabela ASCII, sendo estes, algarismos, letras do alfabeto, caracteres de acentuação, pontuação e outros, como por  
30 exemplo o carater “New Line” responsável pela mudança de linha.

Por outro lado, os dados dos ficheiros binários, são constituídos por qualquer carater existente na tabela ASCII e são armazenados como bytes. De modo geral os dados binários não são muito portáveis, visto que em alguns sistemas os inteiros são representados um número diferente de bits, além disso, os ficheiros binários não estão organizados por linhas, são escritos em blocos de  
35 memória para o disco e lidos em blocos do disco para a memória.

## 1.2 Algoritmos de ordenação e pesquisa

Um Algoritmo de ordenação é um algoritmo que coloca os elementos de uma dada sequência numa ordem específica, assim, os algoritmos de ordenação devem ser eficientes quer em termos de velocidade de execução como em termos de espaço ocupado.

40 Para a realização do trabalho prático apenas foi usado o algoritmo de ordenação Bubble sort, este é um algoritmo de ordenação simples, que consiste, em sucessivas trocas de valores, isto é, dependendo do modo como queremos ordenar a informação, isto é, de forma crescente ou decrescente, o algoritmo comprara os dois valores e caso este seja maior ou menor, executa a troca de posição, deste modo, ao fim de  $n-1$  trocas, onde  $n$  é o número de elementos, a informação esta  
45 ordenada.

## 1.3 Alocação dinâmica de memória

A alocação dinâmica de memória surgiu pois em muitos programas não se sabia qual era a quantidade necessária de memória para alocar e deste modo, foram administradas funções malloc, calloc, realloc, estas são responsáveis por reservar apenas a memória estritamente necessária para  
50 a aplicação funcionar corretamente, contudo também foi implementada a função free, que tem como objetivo libertar a memória quando a aplicação é encerrada.

## 1.4 Listas ligadas simples

Uma lista ligada é uma estrutura de dados linear e dinâmica, esta é composta por vários nós que estão interligadas através de apontadores, ou seja, cada nó possui um apontador que aponta para o endereço de memória do próximo nó.

Desse modo, os nós da estrutura não precisam estar em posições contíguas da memória. Isso faz com que a estrutura se torne dinâmica, pois a qualquer instante, é possível alocar um novo nó e mudar os apontadores dos nós já existentes, de modo a que o novo nó seja inserido na estrutura com êxito, na posição desejada pelo programador.

## 1.5 Listas duplamente ligadas

As listas duplamente ligadas são um método que visa a procura de endereços de memória com espaço necessário para guardar toda a informação pretendida pelo o utilizador. Estas listas conhecem a localização uma das outras através do endereço de memória de cada.

Através das listas duplamente ligadas o utilizador consegue armazenar informações na memória em locais onde esta esteja disponível memória suficiente para armazenar as mesmas.

Estes locais de memória não têm obrigatoriamente que se encontrar de forma sucessiva na memória uma vez que cada estrutura de dados contém o endereço de memória do nó anterior e do nó seguinte.

## 1.6 Queues

As queues são um tipo de estrutura de dados dinâmica onde obedece ao princípio de inserção/remoção: FIFO (First-in First-out), ou seja, a primeira informação a entrar é a primeira informação a sair.

Deste modo, a inserção e a remoção de elementos de uma queue faz-se utilizando as extremidades. Visto que numa queue, a única forma de inserir um elemento é no fim da fila, apenas se consegue obter o elemento que está no início da fila, assim, não é permitido o acesso de algum elemento em particular que se situe no meio da queue.

## 2 Desenvolvimento

Para o desenvolvimento deste trabalho, e para uma melhor organização do desenvolvimento do projeto, o grupo reuniu-se para definir uma estratégia que permitisse entrar o projeto cumprindo  
80 tudo o que era pedido dentro do prazo definido pela equipa docente. Essa estratégia foi dividida em etapas que são apresentadas nesta secção.

### 1. Compreensão do enunciado

Nesta etapa, o grupo reuniu-se e leu o enunciado, em conjunto, para compreender tudo aquilo que era pedido.

### 85 2. Planeamento

Nesta etapa, o grupo dividiu o projeto em tarefas e definiu em que altura essas tarefas deviam estar concluídas, nomeadamente:

- **Fluxograma** - prazo para terminar (21/05/2021)
- **Código** - prazo para terminar (03/06/2021)
- **Relatório** - prazo para terminar (06/06/2021)

### 90 3. Verificação e optimização

Nesta etapa, o grupo verificou se o programa cumpria todos os parâmetros do enunciado e se não existia qualquer tipo de *bug* no algoritmo. Após a revisão da sua criação, o grupo decidiu optimizar o seu algoritmo implementando libertação de memória, `system("cls")`, `system("pause")`, `#include < windows.h >...`  
95

### 4. Implementação do manual de utilizador e de um breve vídeo tutorial

Nesta etapa, o grupo decidiu implementar um manual de utilizador [B] onde explica passo-a-passo as funcionalidades do programa. Irá também em anexo um breve vídeo a explicar as funcionalidades do programa.

100 É importante notar que após o desenvolvimento deste programa, o grupo utilizou certos comandos que apenas são funcionais em sistemas operativos *Windows*, portanto caso o utilize em outro sistema operativo é muito provável que haja alguma incompatibilidade.

Como o programa foi criado a partir da linguagem C, é nativamente *case-sensitive*, e o grupo não teve oportunidade para conseguir contornar essa situação.

## 105 2.1 Compreensão do enunciado

O grupo antes de começar a resolver o projeto em si, teve de ler várias vezes, atenciosamente, o enunciado apresentado, para evitar qualquer tipo de falha ou de lapso. Após várias leituras, o grupo acabou por perceber o que era pretendido fazer e começou por organizar dados, objetivos por secções[2.3.1].

110 O programa iria possuir dois tipos de utilizadores.

- Administrador
- Analista

Os utilizadores do tipo **Administrador** teriam acesso a várias opções de listagens, de CRUD de propostas de crédito, e como boa prática, decidiu-se também fazer um CRUD extra para todos os utilizadores do programa.

Os utilizadores do tipo **Analista** apenas teriam acesso à análise de propostas de crédito que se encontram em filas de processamento próprias de cada prioridade. Após a análise de uma proposta, o analista irá receber pontos que depois iram gerar um ranking de analistas.

Cada proposta de crédito teria um número sequencial (número gerado automaticamente pelo programa), um identificador, um nome do cliente que solicita a tal proposta, o IBAN, o montante do crédito solicitado e um conjunto de até cinco garantias e uma prioridade determinada pelo próprio programa de acordo com o montante.

Existem quatro tipos de garantias(**Imóvel, Fiador, Depósitos, Produtos**), uma descrição e um valor associado.

125 O analista, ao analisar uma proposta de crédito, terá que dar um parecer do tipo (**positivo/negativo**) e ainda inserir uma justificação para tal, que ficará registada numa determinada data do tipo (DD/MM/YYYY).

A aplicação irá ser parametrizável ao nível das prioridades e dos respetivos limites (montante inferior e superior). Essa parametrização irá ser efetuada após a leitura de um ficheiro do tipo CSV (**prioridade;montanteInferior;montanteSuperior**).

Por fim, o programa deverá garantir que o seu estado de execução seja guardado para mais tarde ser executado sem o ficheiro CSV.

## 2.2 Fluxograma

O grupo começou a trabalhar a solução definindo um algoritmo que iria ser utilizado para realizar o projeto. Os elementos do grupo apresentaram diferentes soluções e debateram entre a solução final, aproveitando partes propostas por ambos os elementos do grupo. O fluxograma global é apresentado na figura 1.

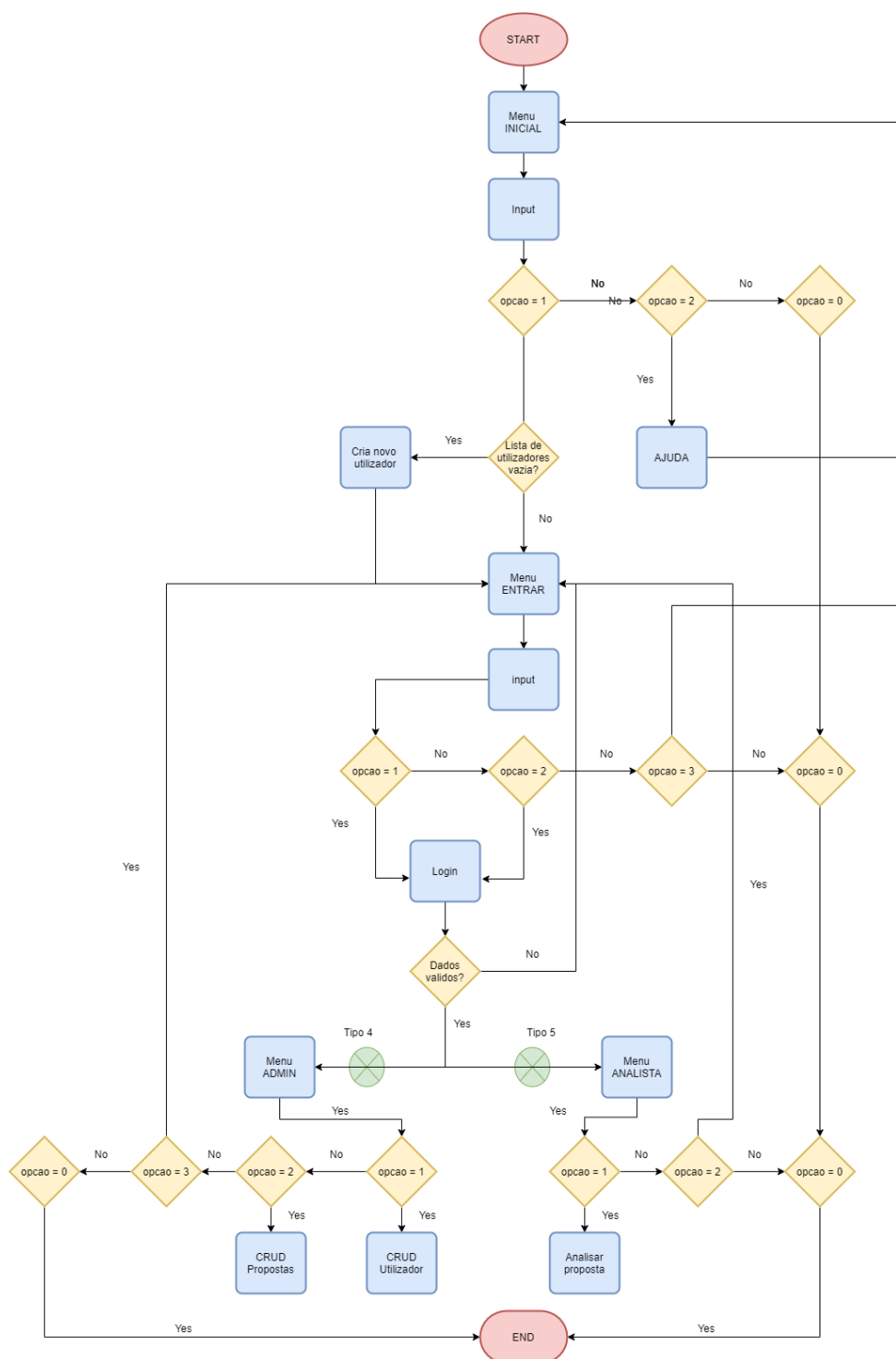


Figura 1: Fluxograma global do projeto



## 2.3 Código em linguagem C

### 2.3.1 Organização

140 Na figura [2], encontra-se todos os ficheiros implementados na realização deste trabalho prático. A aplicação utilizada para desenvolver este trabalho foi o Microsoft Visual Studio Code.

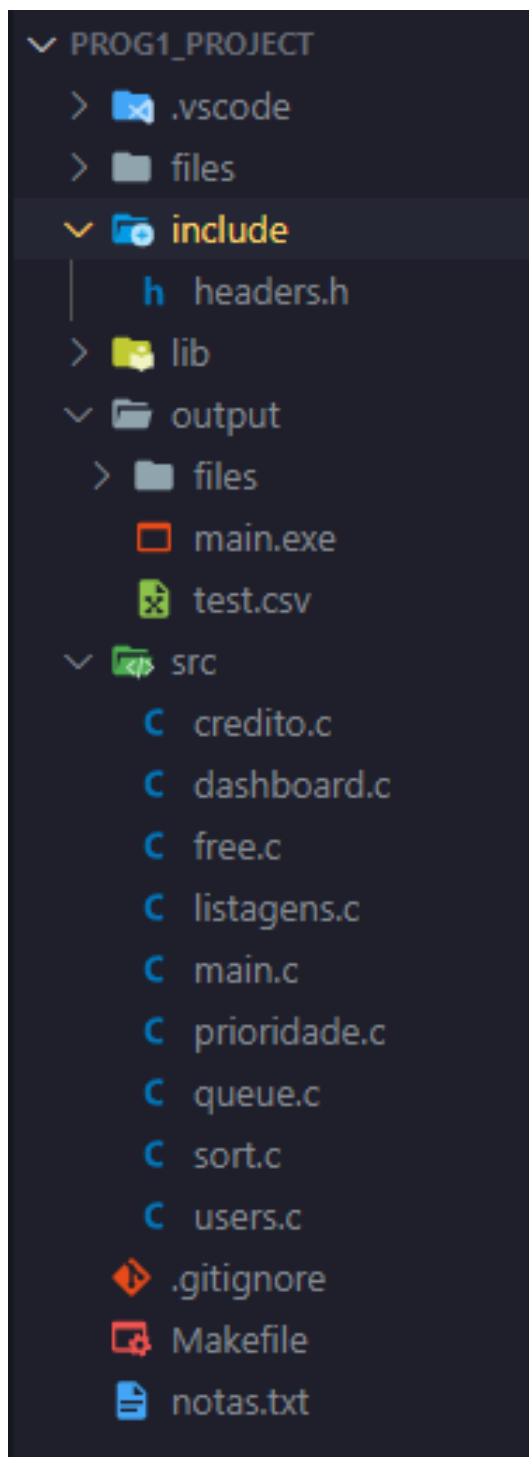


Figura 2: Organização dos vários ficheiros do programa

### 2.3.2 Estrutura de dados utilizada

Durante a organização do trabalho os membros decidiram que seria mais vantajoso a utilização de listas ligadas e listas duplamente ligadas, porem também foram implementados arrays.

145 Deste modo, foram utilizadas quatro listas duplamente ligadas, para a estrutura do utilizador, para a estrutura das propostas de credito analisadas, para a estrutura das prioridades e por fim, para as propostas de credito não analisadas. Por outro lado, foi necessário recorrer a implementação de uma lista simples de listas duplamente ligadas de propostas de credito com uma respetiva prioridade.

150 Por fim, recorreu-se a um array de estruturas para implementar as garantias na estrutura do credito. No apêndice[A], encontra-se imagens relativamente a cada estrutura de dados utilizadas

### 2.3.3 Algoritmos utilizados

Durante a realização do trabalho prático recorreu-se a diferentes algoritmos. Assim, para cumprir todos os objetivos propostos recorreu-se ao algoritmo Bubblesort para ordenar os dados de forma crescente/decrescente segundo um determinado tipo de dados. Escolheu-se o algoritmo 155 Bubblesort, pois tem um algoritmo de implementação mais simples e adequa-se melhor ao objetivo pretendido.

Também foi necessário alocar memória, e para isso o grupo tinha ao seu dispor duas funções: a função malloc e a função calloc, deste modo, o grupo optou por utilizar a função calloc pois além 160 de alocar a memória, inicializa a mesma.

Por fim, recorreu-se a um algoritmo de pesquisa que tem como parâmetro o nome da proposta de credito, assim, o grupo utilizou a ciclo "for" com o intuito de percorrer a lista duplamente ligada, e dentro do mesmo recorreu-se a função "string compare" de modo a comparar o nome recebido com os nomes existentes na lista duplamente ligada, assim se os nomes forem iguais será imprimida a 165 proposta de credito, caso contrário, o utilizador será informado que não existe nenhuma proposta de credito registada com aquele determinado nome.

### 2.3.4 Problemas encontrados

Ao longo da realização do trabalho prático o grupo foi-se deparando com vários problemas tais como: nas operações com listas de listas duplamente ligadas relacionadas com as filas de processamento, ordenação de listas, leitura do ficheiro csv, e por fim, na implementação do "switch" da função main.

Por um lado, para combater os problemas referidos anteriormente o grupo reuniu-se diversas vezes para tentar obter uma solução para os problemas detetados, porém nem sempre o diálogo entre elementos do grupo foi o suficiente para alcançar uma solução, deste modo, o grupo tentou, através da discussão entre outros colegas, alcançar uma forma de solucionar os contratempos achados.

Por outro lado, o grupo por vezes não conseguiu solucionar os problemas, e foi necessário encontrar novas soluções, deste modo, o grupo recorreu aos livros de [1][2][3][4], contudo, o grupo teve de recorrer à ajuda dos professores Domingos Gomes e Pedro Coutinho, que por numerosas vezes se mostraram disponíveis para apoiar o grupo no esclarecimento de dúvidas em relação ao enunciado do trabalho prático, bem como, em relação ao algoritmo desenvolvido.

### 3 Conclusão

Durante o desenvolvimento do projecto, o grupo foi-se deparando com algumas dificuldades que conseguiu superar recorrendo a vários recursos. Ao longo do trabalho, foram surgindo erros quer de compilação quer de lógica que obrigaram o grupo a procurar soluções através de vários recursos. Um dos recursos utilizados foram os apontamentos e as aulas da unidade curricular Programação 1 e Algoritmos e Estrutura de Dados. Outro recurso utilizado para resolver este problema foi a discussão entre elementos do grupo e entre o grupo e outros colegas. Outro recurso utilizados foram os [1][2][3][4]. De um modo geral, as opiniões dos elementos do grupo podem traduzir-se numa opinião unânime. Por um lado, a realização deste projecto permitiu desenvolver competências de trabalho em equipa que são muito enriquecedoras, pois ao longo da carreira profissional certamente será necessário realizar trabalhos em equipa. Por outro lado, poder aplicar os conceitos teóricos de programação numa aplicação de contexto real, permitiu melhorar a perceção da utilidade da programação nos dias de hoje. Por fim, as dificuldades encontradas ao longo do trabalho obrigaram os alunos a procurar soluções e a desenvolver as capacidades de algoritmia, tornando o desenvolvimento deste projecto um desafio que permitiu adquirir competências e assimilar melhor os conteúdos de programação em C. A apreciação global deste trabalho é positiva.

## Referências

- <sup>200</sup> [1] S. Herbert, *C: The Complete Reference*. McGrawHill, 2000.
- [2] A. M. A. Rocha, *Programação Avançada Usando C*. FCA, ISBN: 978-978-722-546-0, 2006.
- [3] J. V. Vasconcelos, José Braga de Vasconcelos e Carvalho, *Algoritmia e Estruturas de Dados*. FCA, ISBN: 989-615-012-5, 2005.
- [4] L. Damas, *Linguagem C*. Grupo Gen-LTC, 2016.

## A Imagens de Estruturas

```
// Estrutura para o utilizador
typedef struct utilizador
{
    int ID;
    char username[20];
    char nome[50];
    char password[20];
    char tipo[20];
    int tipoID; /* 4 - Administrador 5 - Analista
    int rank;    // Número de propostas de crédito analisadas
} UTILIZADOR;
```

Figura 3: Estrutura de dados utilizador

```
// Estruturas de uma propostas de crédito
typedef struct credito
{
    int numeroSequencial; // Proposta 1 ... Proposta 2 ...
    int ID;                // Posição na fila
    char nome[100];
    char IBAN[50];
    int garantiaNumero;
    GARANTIA garantia[MAX_GARANTIAS]; // Número máximo de garantias 5
    float montante;                // Montante pedido
    char prioridade[20];           // Carregada do ficheiro csv
} CREDITO;
```

Figura 4: Estrutura de dados credito

```
// Estrutura para as prioridades
typedef struct prioridade
{
    char nome[50];
    float montanteInferior, montanteSuperior;
} PRIORIDADE;
```

Figura 5: Estrutura de dados das prioridades

```
// Estrutura para até 5 garantias
typedef struct garantia
{
    char tipo[20]; // Imóvel Fiador Depósitos Produtos
    char descricao[200];
    float valor;
} GARANTIA;
```

Figura 6: Estrutura de dados garantias

```
// Estrutura para a análise de uma dada proposta
typedef struct analise
{
    char utilizador[20]; // = ao username do utilizador que a analisou
    char situacao[20]; // * NEGATIVA OU POSITIVA
    char justificacao[200];
    DATA data;
} ANALISE;
```

Figura 7: Estrutura de dados da análise

```
// Estrutura para as datas das propostas de crédito
typedef struct data /* DD/MM/AAAA
{
    int dia, mes, ano;
} DATA;
```

Figura 8: Estrutura de dados da data

```
// Estrutura da lista dos UTILIZADORES
typedef struct elem_UTILIZADOR
{
    UTILIZADOR info;
    struct elem_UTILIZADOR *anterior; // Aponta para o nó (UTILIZADOR) anterior
    struct elem_UTILIZADOR *seguinte; // Aponta para o nó (UTILIZADOR) seguinte
} ELEM_UTILIZADOR;
```

Figura 9: Estrutura lista duplamente ligada do utilizador

```
// Estrutura da lista dos CREDITOS /* Analisadas
typedef struct elem_CREDITO
{
    CREDITO info;
    ANALISE analise; // informação da análise
    struct elem_CREDITO *anterior; // Aponta para o nó (CREDITO) anterior
    struct elem_CREDITO *seguinte; // Aponta para o nó (CREDITO) seguinte
} ELEM_CREDITO;
```

Figura 10: Estrutura lista duplamente ligada do credito

```
// Estrutura da lista das PRIORIDADES
typedef struct elem_PRIORIDADE
{
    PRIORIDADE info;
    struct elem_PRIORIDADE *anterior; // Aponta para o nó (PRIORIDADE) anterior
    struct elem_PRIORIDADE *seguinte; // Aponta para o nó (PRIORIDADE) seguinte
} ELEM_PRIORIDADE;
```

Figura 11: Estrutura lista duplamente ligada de prioridades

```
// Estrutura para a fila de processamento /* Falta Análise
typedef struct queue_CREDITO
{
    CREDITO info;
    struct elem_CREDITO *seguinte; // Aponta para o nó (CREDITO) seguinte
    struct elem_CREDITO *anterior; // Aponta para o nó (CREDITO) anterior
} QUEUE_CREDITO;
```

Figura 12: Estrutura lista duplamente ligada das propostas de credito não analisadas

```
// Estrutura para lista simples de várias filas de processamento (duplamente ligadas)
typedef struct queues
{
    QUEUE_CREDITO *inilista;
    QUEUE_CREDITO *fimLista;
    PRIORIDADE prioridade;
    struct queues *seguinte; // Aponta para o nó (QUEUE_CREDITO) seguinte
} QUEUES;
```

Figura 13: Estrutura de listas duplamente ligada da queue



## B Manual do Utilizador

No momento em que o programa é iniciado deparamo-nos com o seguinte menu que está presente na figura abaixo. Deste modo, temos 3 opções de escolha: Opção [0] - Sair Opção [1] - Entrar e Opção [2] - Ajuda.

```
★-----MENU INICAL-----★
1 - Entrar
2 - Ajuda

0 - Sair do programa
★-----★

Insire opcao: □
```

Figura 14: Menu inicial

210 Caso o utilizador se encontre dentro do menu inicial e escolha a segunda opção é lhe mostrada a seguinte mensagem presente na figura abaixo.

```
★-----AJUDA-----★
# Caso seja a primeira vez que usa este programa, por favor insira um ficheiro do tipo .csv
# O programa nao aceita "Enters" para mudar de linha.
# O primeiro administrador sera o primeiro utilizador registado no programa.
# Quaso se esqueceu da sua palavra-passe ou username, por favor contacte um administrador.
# Para mais informacao nao hesite em contactar!
# Em anexo encontra-se um manual de utilizador.
★-----★

Copyright: Andre Brandao e Diogo Campos IPVC - ESTG
```

Figura 15: Ajuda

Caso o utilizador se encontre dentro do menu inicial e escolha a primeira opção é nos mostrado mais um menu, o menu entrar onde o utilizador pode escolher 4 opções sendo estas: Opção [0] - Sair, Opção [1] - Entrar como Administrador, Opção [2] - Entrar como Analista e Opção [3] -  
215 Voltar ao menu anterior.

```
★-----LOGIN-----★
1 - Administrador
2 - Analista

3 - Voltar para o menu anterior
0 - Sair do programa
★-----★

Insire opcao: □
```

Figura 16: Menu de entrada

Caso o utilizador se encontre dentro do menu entrar e escolha a primeira opção vai ser direcionado para o menu administrador onde estão presentes todas as funções que o mesmo tem acesso.

## B.1 Administrador

```
*-----MENU ADMIN-----*
1 - CRUD utilizador
2 - CRUD proposta de credito
3 - Relatorio das propostas de credito
4 - Menu de listagens

5 - Voltar ao menu anterior
0 - Sair do programa
*-----*

Insire opcao: 
```

Figura 17: Menu administrador

220 No menu administrador estão presentes as seguintes opções: opção [1] - CRUD utilizador, opção [2] - CRUD propostas de credito, opção [3] - Relatório das propostas de credito, opção [4] - Menu listagens, opção [5] - Voltar ao menu anterior, opção [0] - Sair do programa.

Assim, se o utilizador selecionar a primeira opção terá acesso às funções do gerenciamento do utilizador, sendo estas inserir um utilizador, remover um utilizador e por fim alterar um utilizador  
225 como é possível verificar na figura abaixo.

```
*-----CRUD UTILIZADOR-----*
1 - Inserir utilizador
2 - Remover utilizador
3 - Menu alterar utilizador

4 - Voltar para o menu anterior
0 - Sair do programa
*-----*

Insire opcao: 
```

Figura 18: CRUD utilizador

Caso o utilizador se encontre dentro do menu administrador e escolha a segunda opção será direcionado para o CRUD proposta de credito, onde o utilizador tem acesso as funções de inserir, remover, pesquisar e alterar propostas de credito.

```

*-----CRUD PROPOSTA DE CREDITO-----*
1 - Inserir proposta de credito
2 - Menu alterar proposta de credito
3 - Remover proposta de credito
4 - Pesquisar proposta de credito

5 - Voltar para o menu anterior
0 - Sair do programa
*-----*

Insire opcao: 

```

Figura 19: CRUD proposta de credito

230 Caso o utilizador se encontre dentro do menu administrador e selecione a terceira opção será gerado um ficheiro de texto onde estão presentes as propostas de crédito analisadas, ordenadas por montante;

```

files > relatorio.txt
1
2  *-----PROPOSTA-----*
3
4  Proposta #0
5      ID: 10
6      Nome: teste
7      IBAN: 123
8      Montante: 100.00
9      Prioridade: minimo
10     Numero de garantias: 1
11     *-----*
12
13     *-----GARANTIAS-----*
14
15     Garantia #1
16         Tipo:Produtos
17         Descricao:teste
18         Valor:100.00
19     *-----*
20
21     *-----ANALISE-----*
22     Analise da proposta #0
23
24         Analisada por: analista
25         Situacao: Positiva
26         Justificacao: teste
27         Data: 12/12/2021
28     *-----*
29

```

Figura 20: Relatório propostas de credito

Caso o utilizador se encontre dentro do menu administrador e selecione a quarta opção pode listar por uma proposta de credito ou os utilizadores;

```

*-----MENU LISTAR-----*
1 - Listar as propostas de credito por analisar
2 - Listar as propostas de credito analisadas
3 - Listar as propostas de credito por uma prioridade
4 - Listar as propostas de credito acima de um determinado montante
5 - Listar todas as propostas de credito analisadas por um determinado utilizador
6 - Listar ranking de analistas
7 - Listar utilizador

8 - Voltar para o menu anterior
0 - Sair do programa
*-----*

Insire opcao: 

```

Figura 21: Menu listagens

No menu listar estão presentes as seguintes opções: opção [1] - Listar propostas de crédito não analisadas, opção [2] - Listar propostas de credito analisadas, opção [3] - Listar as propostas de credito por prioridade, opção [4] - Listar as propostas de credito acima de um determinado montante, opção [5] - Listar todas as propostas de crédito analisadas por um determinado analista

Caso o utilizador se encontre dentro do menu administrador e escolha a quinta opção será levado para o menu anterior e por fim, se o mesmo escolher a última opção sairá do programa.

No menu CRUD utilizador estão presentes as seguintes opções: opção [1] - Inserir utilizador, opção [2] - Remover utilizador, opção [3] - Menu alterar utilizador, opção [4] - Voltar ao menu anterior, opção [0] - Sair do programa.

Assim, se o utilizador selecionar a primeira opção conseguirá introduzir um novo utilizador.

```

*-----CRUD UTILIZADOR-----*
1 - Inserir utilizador
2 - Remover utilizador
3 - Menu alterar utilizador

4 - Voltar para o menu anterior
0 - Sair do programa
*-----*

Insire opcao: 1
Introduza um username:
analista
Introduza um nome:
analista
Introduza uma password:
analista
Introduza o tipo de utilizador
[4] - Administrador
[5] - Analista
5

```

Figura 22: Inserir utilizador

Caso o utilizador se encontre dentro do menu CRUD utilizador e selecione a segunda opção conseguirá remover um utilizador.

```
*-----CRUD UTILIZADOR-----*
1 - Inserir utilizador
2 - Remover utilizador
3 - Menu alterar utilizador

4 - Voltar para o menu anterior
0 - Sair do programa
*-----*

Insire opcao: 2
Insira o username a ser removido: diogo
```

Figura 23: Remover utilizador

Caso o utilizador se encontre dentro do menu CRUD utilizador e selecione a terceira opção terá acesso ao menu de alterações do utilizador, onde pode realizar operações como: alterar o nome, o username, a password e por fim, o tipo, isto é, se é administrador ou analista.

250 Caso o utilizador se encontre dentro do menu CRUD utilizador e escolha a quarta opção será levado para o menu anterior e por fim, se o mesmo escolher a última opção sairá do programa.

```
*-----MENU ALTERA UTILIZADOR-----*
1 - Alterar o nome do utilizador
2 - Alterar o username do utilizador
3 - Alterar a password do utilizador
4 - Alterar o tipo do utilizador

5 - Voltar para o menu anterior
0 - Sair do programa
*-----*

Insire opcao: 
```

Figura 24: Menu altera utilizador

No menu altera utilizador estão presentes as seguintes opções: opção [1] - Alterar nome do utilizador, opção [2] - Alterar username do utilizador, opção [3] - Alterar password do utilizador, opção [4] - Alterar o tipo do utilizador, opção [5] - Voltar ao menu anterior, opção [0] - Sair do programa.

255 Assim, se o utilizador seleccionar a primeira opção tem acesso função que permite alterar o nome do utilizador

```
Insira o ID: 1

*-----MENU ALTERA UTILIZADOR-----*
1 - Alterar o nome do utilizador
2 - Alterar o username do utilizador
3 - Alterar a password do utilizador
4 - Alterar o tipo do utilizador

5 - Voltar para o menu anterior
0 - Sair do programa
*-----*

Insire opcao: 1
Utilizador #1
Nome atual: admin
Corresponde ao inserido
**0-NAO**
**1-SIM**
1
Insira o novo nome:
admin1
```

Figura 25: Alterar nome do utilizador

Caso o utilizador se encontre dentro do menu altera utilizador e escolha a segunda opção terá acesso a função que altera o username do utilizador.

```
*-----MENU ALTERA UTILIZADOR-----*
1 - Alterar o nome do utilizador
2 - Alterar o username do utilizador
3 - Alterar a password do utilizador
4 - Alterar o tipo do utilizador

5 - Voltar para o menu anterior
0 - Sair do programa
*-----*

Insire opcao: 2
Utilizador #1
Username atual: admin
Corresponde ao inserido
**0-NAO**
**1-SIM**
1
Insira o novo username:
admin1
```

Figura 26: Alterar username do utilizador

Caso o utilizador se encontre dentro do menu altera utilizador e escolha a terceira opção terá acesso a função que altera a password do utilizador.

```
*-----MENU ALTERA UTILIZADOR-----*
1 - Alterar o nome do utilizador
2 - Alterar o username do utilizador
3 - Alterar a password do utilizador
4 - Alterar o tipo do utilizador

5 - Voltar para o menu anterior
0 - Sair do programa
*-----*

Insire opcao: 3
Utilizador #1
Password atual: admin
Corresponde ao inserido
**0-NAO**
**1-SIM**
1
Insira a nova password:
admin1
```

Figura 27: Alterar password do utilizador

Caso o utilizador se encontre dentro do menu altera utilizador e escolha a quarta opção terá acesso a função que altera o tipo do utilizador.

```
*-----MENU ALTERA UTILIZADOR-----*
1 - Alterar o nome do utilizador
2 - Alterar o username do utilizador
3 - Alterar a password do utilizador
4 - Alterar o tipo do utilizador

5 - Voltar para o menu anterior
0 - Sair do programa
*-----*

Insire opcao: 4
Utilizador #2
Tipo atual: Analista
Corresponde ao inserido
**0-NAO**
**1-SIM**
1
Introduza o tipo de utilizador
[4] - Administrador
[5] - Analista
4
```

Figura 28: Alterar o tipo do utilizador

Caso o utilizador se encontre dentro do menu altera utilizador e escolha a quinta opção será levado para o menu anterior e por fim, se o mesmo escolher a última opção sairá do programa.

265 No menu CRUD utilizador estão presentes as seguintes opções: opção [1] - Inserir proposta de credito, opção [2] - Alterar propostas de credito, opção [3] - Remover proposta de credito, opção [4] - Pesquisar proposta de credito, opção [5] - Voltar ao menu anterior, opção [0] - Sair do programa. Assim, se o utilizador selecionar a primeira opção conseguirá introduzir uma nova proposta.

```
*-----CRUD PROPOSTA DE CREDITO-----*
1 - Inserir proposta de credito
2 - Menu alterar proposta de credito
3 - Remover proposta de credito
4 - Pesquisar proposta de credito

5 - Voltar para o menu anterior
0 - Sair do programa
*-----*

Insire opcao: 1
Introduza identificador:
10
Introduza nome:
teste
Introduza IBAN:
123
Quantas garantias?
1
Tipo de garantia?

[0] - Imovel
[1] - Fiador
[2] - Depositos
[3] - Produtos
0
Descricao sobre a proposta:
teste
Valor:
100
Montante:
100
```

Figura 29: Inserir proposta de credito

270 Caso o utilizador se encontre dentro do menu CRUD proposta de credito e selecione a segunda opção terá acesso ao menu de alterações das propostas de credito, onde pode realizar operações como por exemplo, opção [1] - Alterar o nome, opção [2] - Alterar o IBAN, opção [3] - Alterar as garantias, opção [4] - Alterar o montante, opção [5] - Alterar erro analise, opção [6] - Voltar ao menu anterior, e por fim, opção [0] - Sair do programa.

```
*-----MENU ALTERA PROPOSTAS CREDITO-----*
1 - Alterar o nome
2 - Alterar IBAN
3 - Alterar as garantias
4 - Alterar o montante
5 - Alterar erro na analise

6 - Voltar ao menu anterior
0 - Sair do programa
*-----*

Insira opcao: □
```

Figura 30: Menu altera propostas de credito



Assim, caso o utilizador se encontre dentro do menu altera propostas, e selecione a primeira  
275 opção poderá alterar o nome da proposta de credito.

```
*-----MENU ALTERA PROPOSTAS CREDITO-----*
1 - Alterar o nome
2 - Alterar IBAN
3 - Alterar as garantias
4 - Alterar o montante
5 - Alterar erro na analise

6 - Voltar ao menu anterior
0 - Sair do programa
*-----*

Insira opcao: 1
Proposta #1
Nome atual: teste1
Corresponde ao inserido
**0-NAO**
**1-SIM**
1
Insira o novo nome:
teste2
```

Figura 31: Alterar nome da propostas de credito

Caso o utilizador se encontre dentro do menu altera proposta de credito e escolha a segunda opção terá acesso a função que altera o IBAN.

```
*-----MENU ALTERA PROPOSTAS CREDITO-----*
1 - Alterar o nome
2 - Alterar IBAN
3 - Alterar as garantias
4 - Alterar o montante
5 - Alterar erro na analise

6 - Voltar ao menu anterior
0 - Sair do programa
*-----*

Insira opcao: 2
Proposta #1
IBAN atual: 1234
Corresponde ao inserido
**0-NAO**
**1-SIM**
1
Insira o novo IBAN:
12345
```

Figura 32: Alterar IBAN da proposta de credito

Caso o utilizador se encontre dentro do menu altera proposta de credito e escolha a terceira opção terá acesso a função que altera as garantias da proposta de credito.

```
*-----MENU ALTERA PROPOSTAS CREDITO-----*
1 - Alterar o nome
2 - Alterar IBAN
3 - Alterar as garantias
4 - Alterar o montante
5 - Alterar erro na analise

6 - Voltar ao menu anterior
0 - Sair do programa
*-----*

Insira opcao: 3
Proposta #1
Numero de garantias atual: 1
Corresponde ao inserido
**0-NAO**
**1-SIM**
1

*-----ALTERAR GARANTIAS-----*
1 - Alterar #1 garantia
2 - Alterar #2 garantia
3 - Alterar #3 garantia
4 - Alterar #4 garantia
5 - Alterar #5 garantia
6 - Listar garantias

7 - Voltar para o menu anterior
0 - Sair do programa
*-----*

Insire opcao: 1
Tipo de garantia?

[0] - Imovel
[1] - Fiador
[2] - Depositos
[3] - Produtos
2
Descricao sobre a proposta:
teste2
Valor:
100
```

Figura 33: Alterar as garantias da proposta de credito

280

Caso o utilizador se encontre dentro do menu altera proposta de credito e escolha a quarta opção terá acesso a função que altera o montante.

```
*-----MENU ALTERA PROPOSTAS CREDITO-----*
1 - Alterar o nome
2 - Alterar IBAN
3 - Alterar as garantias
4 - Alterar o montante
5 - Alterar erro na analise

6 - Voltar ao menu anterior
0 - Sair do programa
*-----*

Insira opcao: 4
Proposta #0
Montante atual: 100.00
Corresponde ao inserido
**0-NAO**
**1-SIM**
1
Insira o novo montante:
500
```

Figura 34: Alterar o montante da proposta de credito

Caso o utilizador se encontre dentro do menu altera proposta de credito e escolha a quinta opção terá acesso a função que altera um erro na analise de uma proposta de credito.

```
*-----MENU ALTERA PROPOSTAS CREDITO-----*
1 - Alterar o nome
2 - Alterar IBAN
3 - Alterar as garantias
4 - Alterar o montante
5 - Alterar erro na analise

6 - Voltar ao menu anterior
0 - Sair do programa
*-----*

Insira opcao: 5
Situacao atual: Positiva
Corresponde ao inserido
**0-NAO**
**1-SIM**
1
Insira a nova situacao:
**0-NEGATIVO**
**1-POSITIVO**
0
Insira uma nova justificacao:
teste 2
Insira um dia:
13
Insira um mes:
13
Insira um ano:
2021
```

Figura 35: Alterar erro na analise de uma proposta de credito

285 Caso o utilizador se encontre dentro do menu CRUD propostas de credito e escolha a sexta opção será levado para o menu anterior e por fim, se o mesmo escolher a última opção sairá do programa.

Caso o utilizador se encontre dentro do menu CRUD proposta de credito e selecione a terceira opção poderá remover uma proposta de credito.

```
*-----CRUD PROPOSTA DE CREDITO-----*
1 - Inserir proposta de credito
2 - Menu alterar proposta de credito
3 - Remover proposta de credito
4 - Pesquisar proposta de credito

5 - Voltar para o menu anterior
0 - Sair do programa
*-----*

Insire opcao: 3
Insira o ID da proposta de credito a ser removida: 1
```

Figura 36: Remover proposta de credito

290

Caso o utilizador se encontre dentro do menu CRUD proposta de credito e selecione a quarta opção poderá pesquisar as propostas de credito inseridas através do nome em que está registada a proposta de credito.

```
*-----CRUD PROPOSTA DE CREDITO-----*
1 - Inserir proposta de credito
2 - Menu alterar proposta de credito
3 - Remover proposta de credito
4 - Pesquisar proposta de credito

5 - Voltar para o menu anterior
0 - Sair do programa
*-----*

Insire opcao: 4
Insira o nome:
teste

*-----PROPOSTAS-----*

Proposta #0
  ID: 10
  Nome: teste
  IBAN: 123
  Montante: 100.00
  Prioridade: minimo
Numero de garantias: 1
*-----*

*-----GARANTIAS-----*

Garantia #1
  Tipo:Imovel
  Descricao:teste
  Valor:100.00
*-----*

*-----ANALISE-----*
Analise da proposta #0

  Analisada por: analista
  Situacao: Positiva
  Justificacao: teste
  Data: 12/12/2021
*-----*
```

Figura 37: Pesquisar proposta de credito

Caso o utilizador se encontre dentro do menu CRUD propostas de credito e escolha a quinta opção será levado para o menu anterior e por fim, se o mesmo escolher a última opção sairá do programa.

295

## B.2 Analista

Caso o utilizador se encontre dentro do menu entrar e escolha a segunda opção será direcionado para o menu analista onde estão presentes todas as funções que o mesmo tem acesso.

```
★-----MENU ANALISTA-----★  
1 - Analisar proposta de credito  
  
2 - Voltar para o menu anterior  
0 - Sair do programa  
★-----★  
  
Insire opcao: 
```

Figura 38: Menu analista

Caso o utilizador se encontre dentro do menu entrar e escolha a terceira opção será levado para o menu anterior e por fim, se o mesmo escolher a ultima opção sairá do programa.