Engenharia de Software 19/11/2020

**Trabalho em COBOL**

Novembro 2020

Curso Profissional Programador de Informática

|  |  |
| --- | --- |
| InformAÇÕES | |
| NomeS | Arianna Cicero, Diogo Marques. |
| N.º | 3, 7. |
| Turma | 12º TPI. |
| Morada | Braga |
| E-mail | [**33479@alunos.aecarlosamarante.pt**](mailto:33479@alunos.aecarlosamarante.pt)**,** 27691@alunos.aecarlosamarante.pt. |
| Telemóvel | 914495006(Arianna), 961777998(Diogo) |
| Nome do/a professor/a | Ana Maria Monteiro. |
| Disciplina | Engenharia de Software. |

Conteúdo

[Introdução 1](#_Toc57594619)

[Explicação do programa 2](#_Toc57594620)

[Variáveis 2](#_Toc57594621)

[Introduzir 4](#_Toc57594622)

[Consultar 5](#_Toc57594623)

[Alterar 5](#_Toc57594624)

[Eliminar 5](#_Toc57594625)

[Consultar eliminados 5](#_Toc57594626)

[Salvar dados nos ficheiros 6](#_Toc57594627)

[Carregar dados nos ficheiros 6](#_Toc57594628)

[Sair 6](#_Toc57594629)

[Código do programa 7](#_Toc57594630)

[Início do programa 7](#_Toc57594631)

[Menu 7](#_Toc57594632)

[Introduzir 8](#_Toc57594633)

[Consultar 14](#_Toc57594634)

[Alterar 17](#_Toc57594635)

[Eliminar 22](#_Toc57594636)

[Consultar apagados 25](#_Toc57594637)

[Salvar dados nos ficheiros 28](#_Toc57594638)

[Carregar dados nos ficheiros 31](#_Toc57594639)

[Conclusão 35](#_Toc57594640)

[Bibliografia 36](#_Toc57594641)

[Anexos 37](#_Toc57594642)

Índice de imagens

[Figura 1. Inicio do programa 7](#_Toc57594564)

[Figura 2. Menu de opções 7](#_Toc57594565)

[Figura 3. Evaluate true do menu principal 8](#_Toc57594566)

[Figura 4. Menu introduzir 8](#_Toc57594567)

[Figura 5. Opção 1 e 2 do evaluate true do menu introduzir 9](#_Toc57594568)

[Figura 6. Opção 3 e 0 do evaluate true do menu introduzir 9](#_Toc57594569)

[Figura 7. Introduzir clientes 10](#_Toc57594570)

[Figura 8. Introduzir Produtos (1/2) 11](#_Toc57594571)

[Figura 9. Introduzir Produtos (2/2) 11](#_Toc57594572)

[Figura 10. Introduzir Faturas (1/3) 12](#_Toc57594573)

[Figura 11. Introduzir Faturas (2/3) 12](#_Toc57594574)

[Figura 12. Introduzir Faturas (3/3) 13](#_Toc57594575)

[Figura 13. Menu consultar (1/3) 14](#_Toc57594576)

[Figura 14. Menu Consultar (2/3) 14](#_Toc57594577)

[Figura 15. Menu Consultar (3/3) 15](#_Toc57594578)

[Figura 16. Consultar Clientes 15](#_Toc57594579)

[Figura 17. Consultar Produtos 16](#_Toc57594580)

[Figura 18. Consultar Faturas 16](#_Toc57594581)

[Figura 19. Menu Alterar 17](#_Toc57594582)

[Figura 20. Alterar clientes 18](#_Toc57594583)

[Figura 21. Alterar produtos 19](#_Toc57594584)

[Figura 22. Alterar faturas (1/3) 20](#_Toc57594585)

[Figura 23. Alterar faturas (2/3) 21](#_Toc57594586)

[Figura 24. Alterar faturas (3/3) 21](#_Toc57594587)

[Figura 25. Menu Eliminar 22](#_Toc57594588)

[Figura 26. Eliminar Clientes 23](#_Toc57594589)

[Figura 27. Eliminar Produtos 23](#_Toc57594590)

[Figura 28. Eliminar Faturas 24](#_Toc57594591)

[Figura 29. Menu Consultar Apagados (1/3) 25](#_Toc57594592)

[Figura 30. Menu Consultar Apagados (2/3) 25](#_Toc57594593)

[Figura 31. Menu Consultar Apagados (3/3) 26](#_Toc57594594)

[Figura 32. Consultar Clientes Apagados 26](#_Toc57594595)

[Figura 33. Consultar Produtos Apagados 27](#_Toc57594596)

[Figura 34. Consultar Faturas Apagadas 27](#_Toc57594597)

[Figura 35. Salvar dados 28](#_Toc57594598)

[Figura 36. Salvar Clientes 29](#_Toc57594599)

[Figura 37. Salvar Produto 29](#_Toc57594600)

[Figura 38. Salvar Faturas 30](#_Toc57594601)

[Figura 39. Carregar dados 31](#_Toc57594602)

[Figura 40. Carregar Clientes 31](#_Toc57594603)

[Figura 41. Ler próximo cliente 32](#_Toc57594604)

[Figura 42. Exibir cliente 32](#_Toc57594605)

[Figura 43. Guardar cliente 32](#_Toc57594606)

[Figura 44. Carregar produtos 32](#_Toc57594607)

[Figura 45. Ler próximo produto 32](#_Toc57594608)

[Figura 46. Exibir produto 32](#_Toc57594609)

[Figura 47. Guardar produto 32](#_Toc57594610)

[Figura 48. Carregar faturas 33](#_Toc57594611)

[Figura 49. Ler próxima fatura 33](#_Toc57594612)

[Figura 50. Exibir fatura 33](#_Toc57594613)

[Figura 51. Guardar fatura 33](#_Toc57594614)

[Figura 52. Ler produtos-faturas 33](#_Toc57594615)

[Figura 53. Ler próximo produto-fatura 33](#_Toc57594616)

[Figura 54. Exibir produto-fatura 34](#_Toc57594617)

[Figura 55. Guardar produto-fatura 34](#_Toc57594618)

# Introdução

Neste trabalho sobre programação em COBOL, UFCD 0807, ESW, foi completada a primeira e segunda parte de um programa de gestão para uma loja de doces. Este programa permite à loja ou utilizador do programa, escolher funções como: introduzir, consultar, alterar, eliminar, consultar eliminados e sair (fechar o programa). A criação deste programa começou no dia 05/11/2020 e foi acabado no dia 26/11/2020, no entanto, a sua data de entrega foi estabelecida no dia 30/11/2020. Neste relatório foram anexadas imagens do código do programa e a sua respetiva explicação.

# Explicação do programa

Este programa foi criado com a intenção de gerir informações de uma loja de doces. As funções que o programa permite fazer são básicas, mas necessárias para qualquer loja; estas funções são: introduzir, consultar, alterar, eliminar, consultar eliminados, carregar dados, salvar dados e sair.

## Variáveis

As variáveis utilizadas neste programa foram:

\*Variável de decisão

01 opcao\_continuar pic x value space.

88 sim value "s" "S".

88 nao value "n" "N".

**\*Estrutura da tabela**

01 tabela.

**\*Começo da tabela clientes**

02 clientes occurs 10 times.

03 id-cliente pic 99.

03 nome-cliente pic x(50).

03 morada-cliente pic x(50).

03 telefone-cliente pic x(12).

03 nif-cliente pic 9(9).

03 descricao-cliente pic x(50).

**\*Fim da tabela clientes**

**\*Começo da tabela faturas**

02 faturas occurs 10 times.

03 id-fatura pic 99.

03 dia-fatura pic 99.

03 mes-fatura pic 99.

03 ano-fatura pic 9999.

03 id-cliente-fatura pic 99.

03 n-produtos-fatura pic 9.

03 produtos occurs 5 times indexed by x.

04 id-produtos-fatura pic 99.

04 quantidade-produto-fatura pic 99.

03 descricao-fatura pic x(50).

**\*Fim da tabela faturas**

**\*Começo da tabela produtos**

02 produtos occurs 10 times.

03 id-produto pic 99.

03 nome-produto pic x(50).

03 tipo-produto pic x(50).

88 compota value "c" "C".

88 marmelada value "m" "M".

88 licor value "l" "L".

03 stock-produto pic 9(3).

03 descricao-produto pic x(50).

**\*Fim da tabela produtos**

**\*Variavel para guardar o id dos produtos que temos de alterar/apagar**

77 procurar pic 99 value zero.

**\*Variavel para o indice de procurar**

77 indice pic 99 value zero.

**\*Variavel da opÃ§Ã£o que o usuario escolhe**

77 opcao pic 99 value zero.

**\*variavel para terminar o programa**

77 termina pic 99 value zero.

**\*variavel de quantos produtos a fatura vai ter**

77 quant\_produtos pic 99 value zero.

**\*Variavel do indice dos clientes**

77 indice\_cliente pic 99 value zero.

**\*Variavel do indice dos produtos**

77 indice\_produtos pic 99 value zero.

**\*Variavel do indice de compras/faturas**

77 indice\_faturas pic 99 value zero.

**\*variavel para ver se existe algum campo apagado nas tabelas**

77 apagados pic 99 value zero.

**\*variável de indices**

77 y pic 99 value zero.

**\*mais uma variavel de indices.**

77 z pic 99 value zero.

**\*variavel para ver se ja lemos todos os dados dos ficheiros**

77 final-arquivo pic x.

O total de variáveis utilizadas são 14.

Foram criadas três tabelas: Clientes, Produtos e Faturas. A tabela dos clientes tem 6 variáveis, a tabela produtos tem 5 variáveis, a tabela faturas tem 9 variáveis.

## Introduzir

A opção introduzir permite ao utilizador, como o mesmo título indica, introduzir dados, estes dados que podem ser introduzidos são: dados do cliente, dados do produto e dados da fatura.

* **Dados dos clientes**: os dados do cliente a introduzir são: nome do cliente, morada do cliente, telefone do cliente e NIF do cliente. No entanto, as variáveis são id-cliente, nome-cliente, morada-cliente, telefone-cliente, nif-cliente, descricao-cliente; apesar de ter seis variáveis, o utilizador só pode introduzir quatro (nome do cliente, morada do cliente, telefone do cliente e NIF do cliente), esto se deve a que o ID do cliente é automático, o utilizador não precisa de por o ID de cada cliente nem dos produtos e faturas, a descrição não se pode introduzir porque esta é utilizada como uma variável de verificação, ou seja, se o cliente é apagado, a descrição vai ser “apagado”.
* **Dados dos produtos**: os dados dos produtos a introduzir são: nome do produto, tipo do produto (compota ["c" "C"], marmelada ["m" "M"] e licor ["l" "L"]) e stock do produto. No entanto, as variáveis id-produto, nome-produto, tipo-produto (compota ["c" "C"], marmelada ["m" "M"] e licor ["l" "L"]) stock-produto e descricao-produto; a razão pela qual não se pode introduzir o id do produto e a descrição do produto é porque o id é automático e a descrição do produto não se pode introduzir porque esta é utilizada como uma variável de verificação, ou seja, se o cliente é apagado, a descrição vai ser “apagado”.
* **Dados das faturas**: os dados que podem ser introduzidos na fatura são: dia da fatura, mês da fatura, ano da fatura, id do cliente na fatura, número de produtos na fatura, id de produto na fatura e a quantidade do produto na fatura. A variáveis utilizadas foram: id-fatura, dia-fatura, mes-fatura, ano-fatura, id-cliente-fatura, n-produtos-fatura, id-produtos-fatura, quantidade-produto-fatura, descricao-fatura. Igualmente que nos produtos e os clientes, não é possível introduzir o id das faturas porque este é automático, e também não é possível inserir descrição porque a variável funciona como uma variável de verificação, ou seja, se o cliente é apagado, a descrição vai ser “apagado”.

Se o limite a inserir de clientes/produtos/faturas for atingido, programa vai mostrar uma mensagem a dizer “Limite atingido pela tabela”.

## Consultar

A opção consultar é simples, esta só lista as informações que foram introduzidas (se não foram apagados), como: Clientes, Produtos e Faturas. No entanto, esta opção não lista as informações todas juntas, o programa pergunta ao utilizador qual das três opções (clientes, produtos ou faturas) quer listar e, dependendo da resposta, vai ser listado.

Se a tabela da opção escolhida (clientes, produtos ou faturas) estiver vazia, o programa vai mostra uma mensagem a dizer “A tabela clientes/produtos/faturas esta vazia”.

## Alterar

Esta opção permite ao utilizador, como o nome indica, alterar as informações introduzidas. Se pode alterar os clientes, as faturas e os produtos, mas não todos os campos.

Campos com possibilidade de alterar:

* **Clientes**: nome do cliente, morada do cliente, telefone do cliente e NIF do cliente.
* **Produtos**: nome do produto, tipo do produto (compota ["c" "C"], marmelada ["m" "M"] e licor ["l" "L"]) e stock do produto.
* **Faturas**: dia da fatura, mês da fatura, ano da fatura, id do cliente na fatura, número de produtos na fatura, id de produto na fatura e a quantidade do produto na fatura.

Se a tabela da opção escolhida (clientes, produtos ou faturas) estiver vazia, o programa vai mostra uma mensagem a dizer “A tabela clientes/produtos/faturas esta vazia”.

## Eliminar

A opção eliminar permite ao utilizador eliminar as informações inseridas. O programa pergunta a o utilizador o que quer eliminar e dá-lhe três opções: clientes, produtos ou faturas. Independentemente da opção escolhida, o programa vai pergunta o id do cliente/produto/fatura (dependendo da opção escolhida) e apaga em seguida. A descrição do produto/cliente/fatura vai ficar como “apagado”.

Se a tabela da opção escolhida (clientes, produtos ou faturas) estiver vazia, o programa vai mostra uma mensagem a dizer “A tabela clientes/produtos/faturas esta vazia”.

## Consultar eliminados

Esta opção tem a mesma função que o consultar, só que neste caso só vai listar os dados eliminados. De igual maneira que o primeiro consultar, esta opção não lista as informações todas juntas, o programa pergunta ao utilizador qual das três opções (clientes, produtos ou faturas) quer listar e, dependendo da resposta, vai ser listado.

Se a tabela da opção escolhida (clientes, produtos ou faturas) estiver vazia, o programa vai mostra uma mensagem a dizer “A tabela clientes/produtos/faturas esta vazia”.

## Salvar dados nos ficheiros

Esta opção permite ao utilizador salvar os dados introduzidos num ficheiro “.dat”. Desta vez, o programa não pergunta qual das três opções (clientes, produtos ou faturas quer salvar, simplesmente salva todas as informações automaticamente.

No entanto, antes de salvar, o programa pergunta ao utilizador: “Ao realizar esta opção os dados contidos nos ficheiros serão apagados. Queres mesmo realizar esta opção?”, se a resposta é “sim” (s/S), o programa vai apagar os dados já antes salvados (se é que existem), se a resposta é “não” (n/N) ele anula o salvamento e sai da opção.

## Carregar dados nos ficheiros

Esta opção carrega automaticamente os dados salvados anteriormente nos ficheiros “.dat” ao programa.

## Sair

A opção sair é para terminar o programa.

# Código do programa

Código do programa, imagens e a explicação:

## Início do programa

No início do programa é simples, um “perform” para o menu de opções que vai terminar quando a variável “termina” tenha o valor de 1.

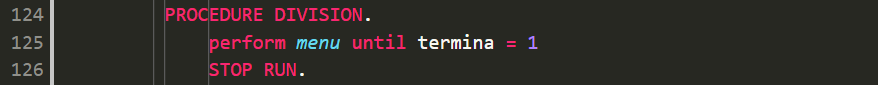


Figura . Inicio do programa

## Menu

O menu de opções apresenta as diferentes opções que o programa oferece ao utilizador, tais como: introduzir, consultar, alterar, eliminar, consultar eliminados, salvar dados nos ficheiros, carregar dados nos ficheiros, sair.

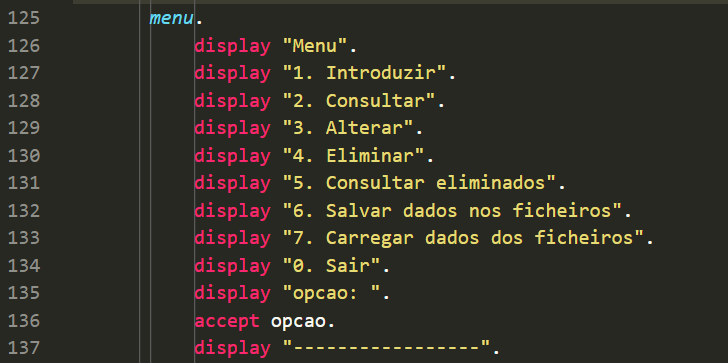


Figura . Menu de opções

A seguir, é apresentado um “evaluate true”, esto vai permitir ao programa escolher a opção inserida pelo utilizador (1, 2, 3, 4, 5, 6, 7, 0), ou seja, se a variável “opcao” tem como valor 1, o programa vai apresentar a opção 1, nesse caso seria a opção Introduzir.

Se o utilizador escrever um valor diferente dos que são apresentados, o programa vai mostrar uma mensagem a dizer “Opção invalida.”, a seguir, pede ao utilizador voltar a introduzir uma opção.

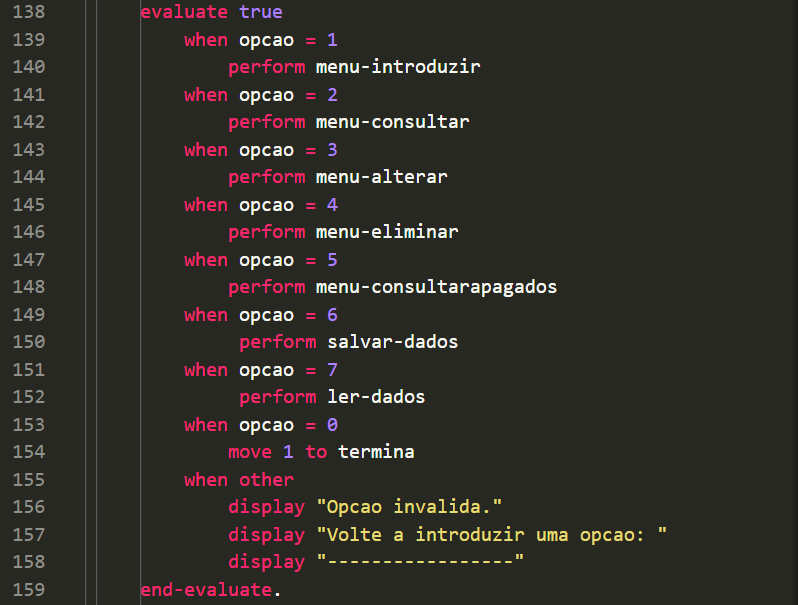


Figura . Evaluate true do menu principal

## Introduzir

Introduzir é uma opção do menu principal que permite ao utilizador introduzir dados de clientes, produtos e faturas.

Ao utilizador escolher esta opção, é apresentado um menu:

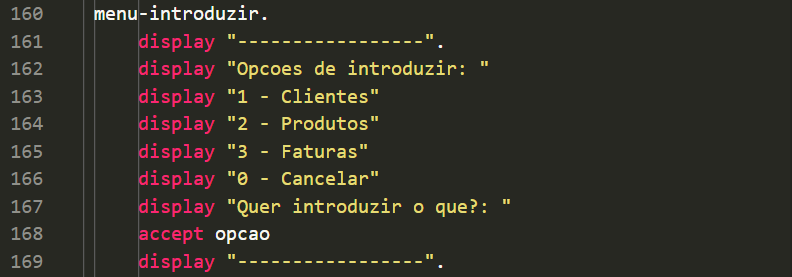


Figura . Menu introduzir

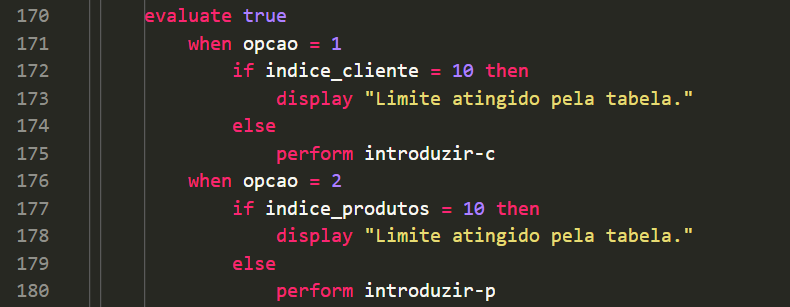


Figura . Opção 1 e 2 do evaluate true do menu introduzir

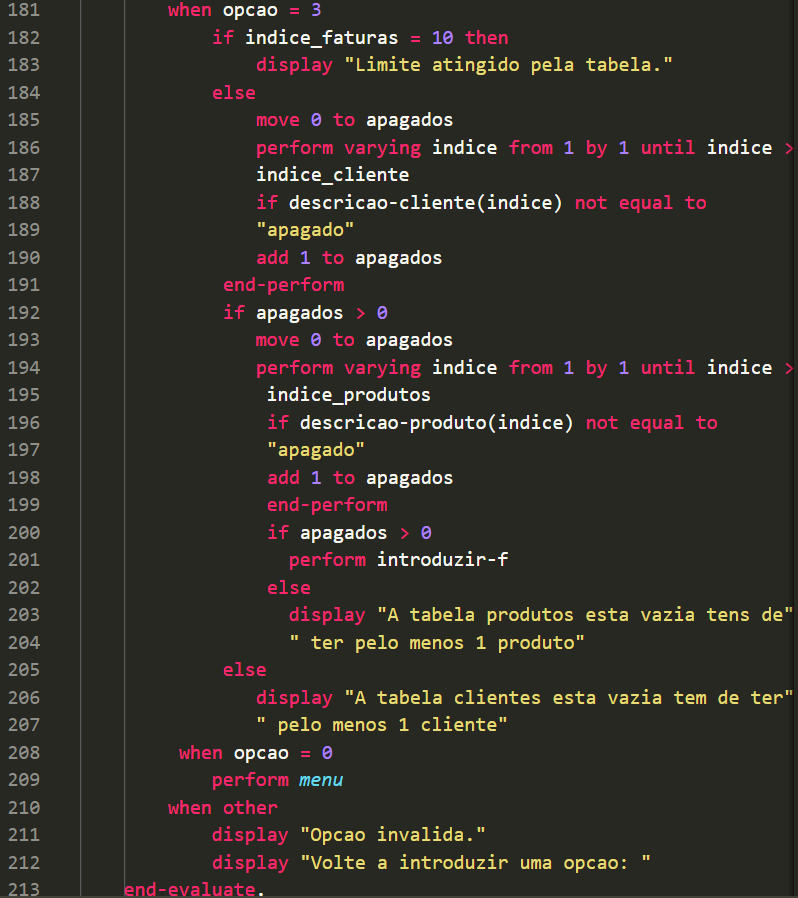


Figura . Opção 3 e 0 do evaluate true do menu introduzir

Se o utilizador escolher a opção 1 (Introduzir clientes) no menu de introduzir, o programa vai a apresentar uma serie de perguntas (como nome do cliente, morada, etc.); por último, pergunta ao utilizador se quer introduzir mais clientes, dependendo da resposta ele vai sair do introduzir ou continuar (também o faz com as outras opções):



Figura . Introduzir clientes

Se o utilizador escolher a opção 2 (introduzir produtos), o programa vai apresentar uma serie de perguntas em relação aos produtos (como nome, quantidade, etc.):

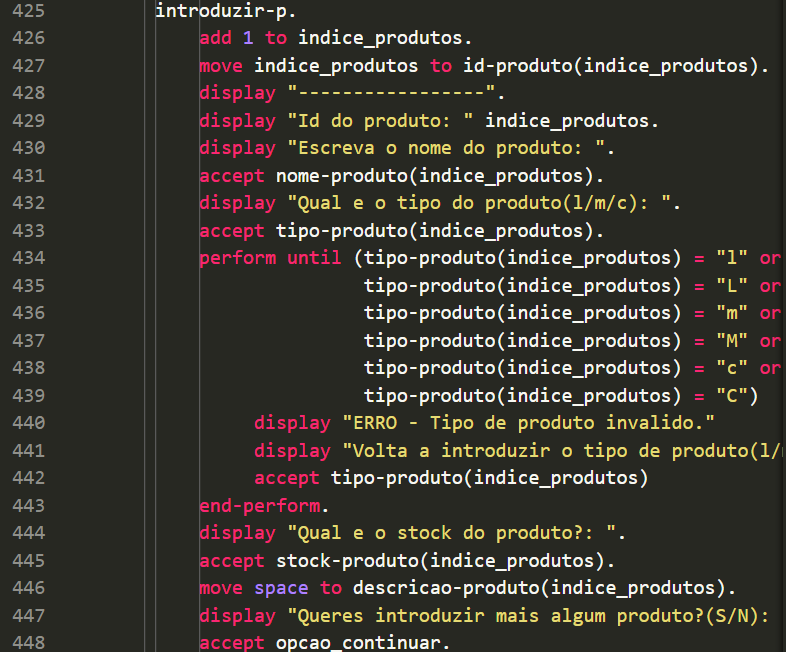


Figura . Introduzir Produtos (1/2)

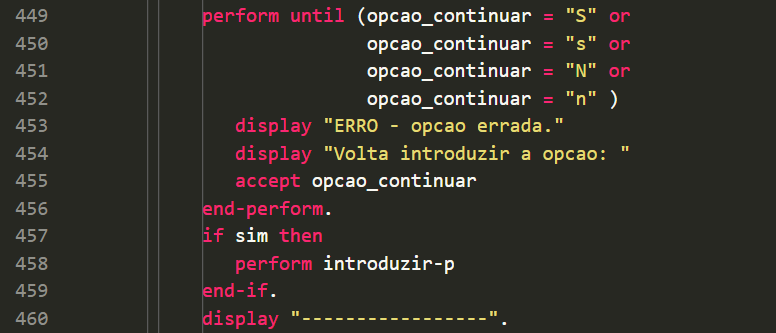


Figura . Introduzir Produtos (2/2)

Logo, a opção 3 é introduzir faturas, o utilizador pode criar uma fatura onde tenho produtos e o cliente que efetuou a compra.

Ao executar esta opção, o programa apresenta uma serie de perguntas relacionadas com a criação da fatura.

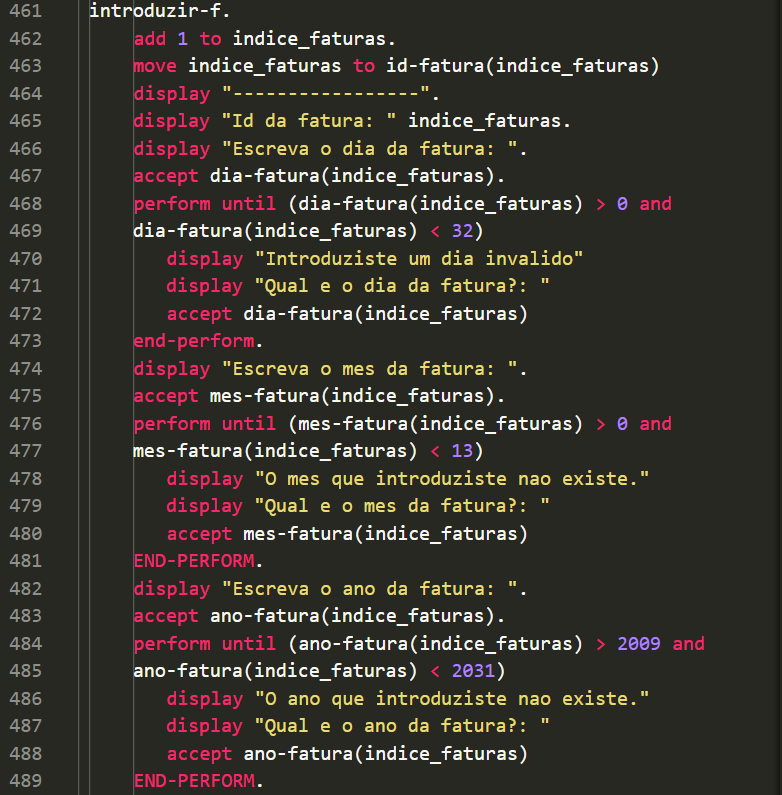


Figura . Introduzir Faturas (1/3)

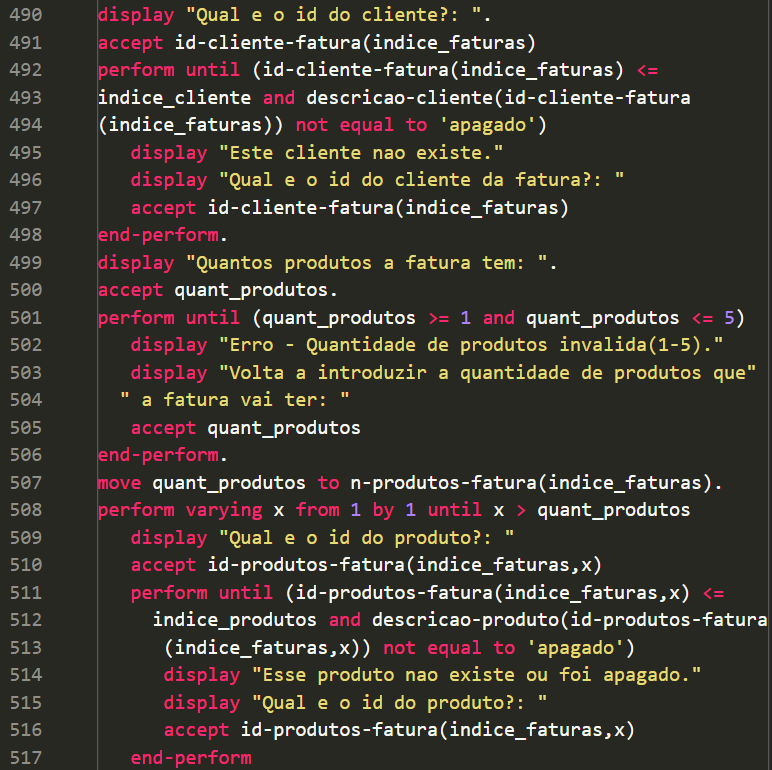


Figura . Introduzir Faturas (2/3)

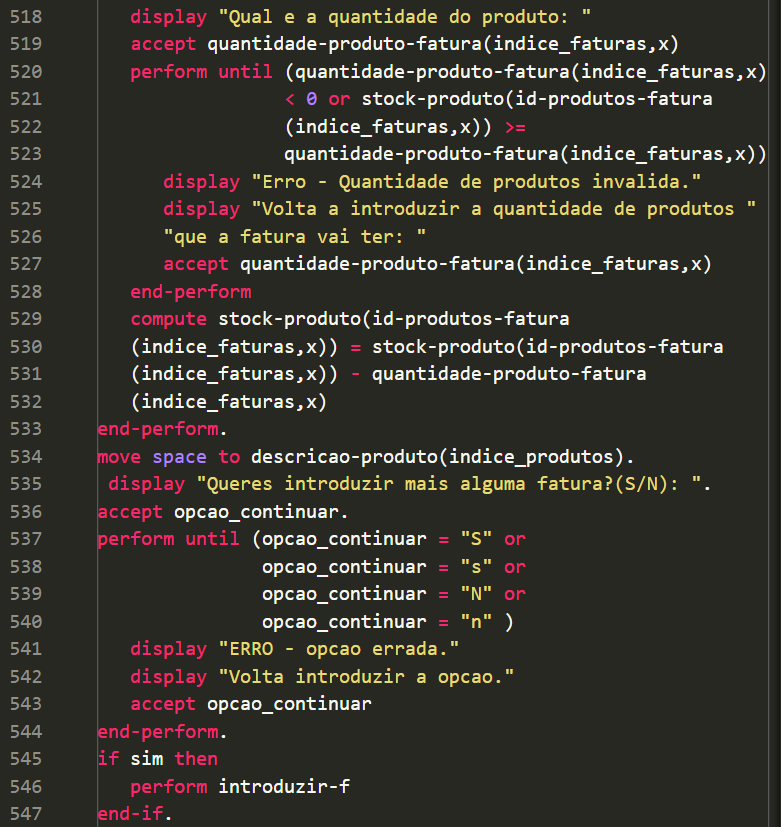


Figura . Introduzir Faturas (3/3)

## Consultar

A opção consultar, permite ao utilizador listar todo o que foi inserido no programa, como: clientes, produtos ou faturas.

A seguir, é apresentado o menu consultar, este presenta as opções disponíveis para executar.

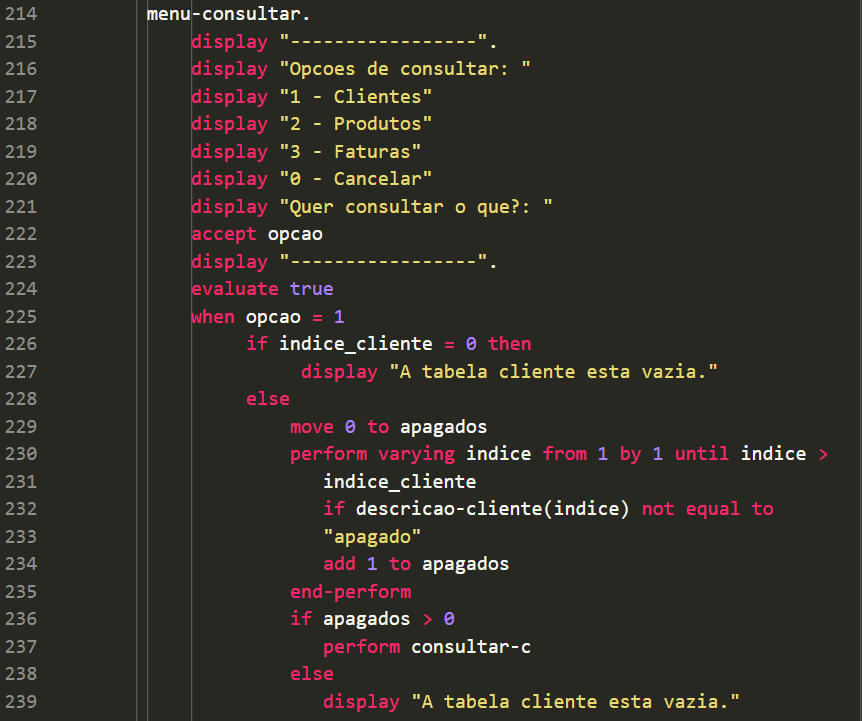


Figura . Menu consultar (1/3)

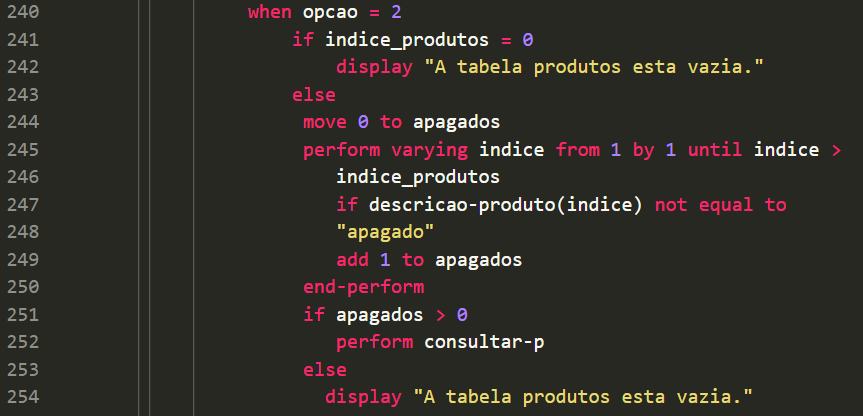


Figura . Menu Consultar (2/3)

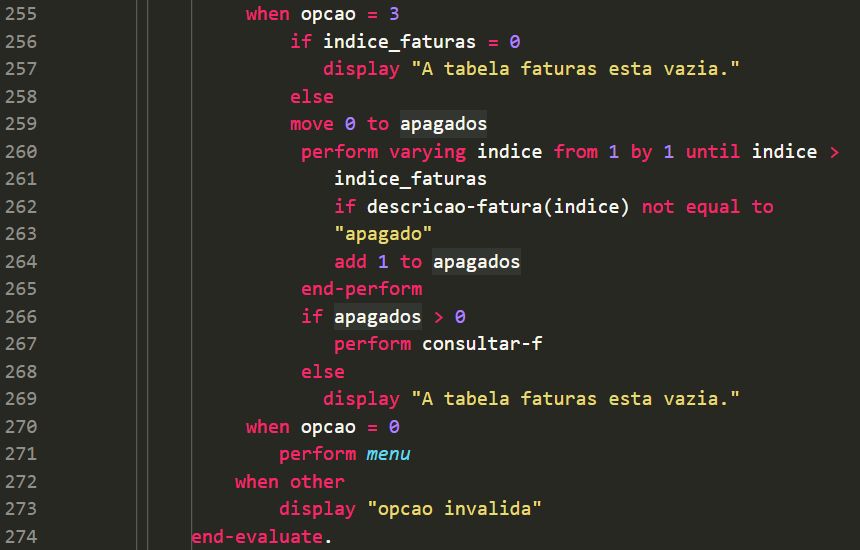


Figura . Menu Consultar (3/3)

A primeira opção do menu consultar, consultar clientes, é a opção que vai listar todos os clientes (com a sua respetiva informação) que foram introduzidos.

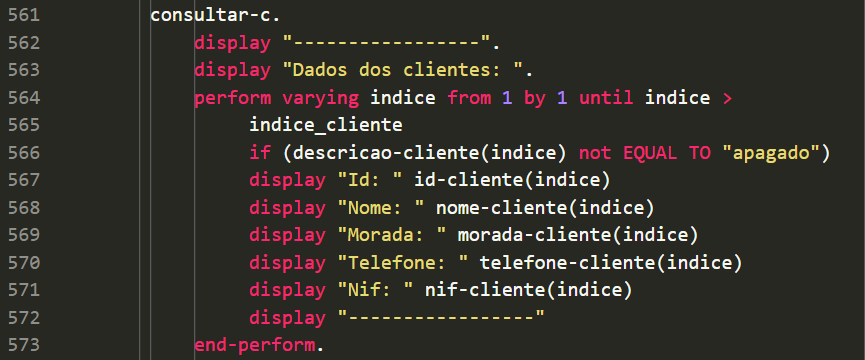


Figura . Consultar Clientes

A segunda opção, consultar produto, vai listar todos os produtos introduzidos (como a sua respetiva informação).



Figura . Consultar Produtos

A terceira opção, consultar faturas, vai listar todas as faturas que foram introduzidas (com as suas respetivas informações.

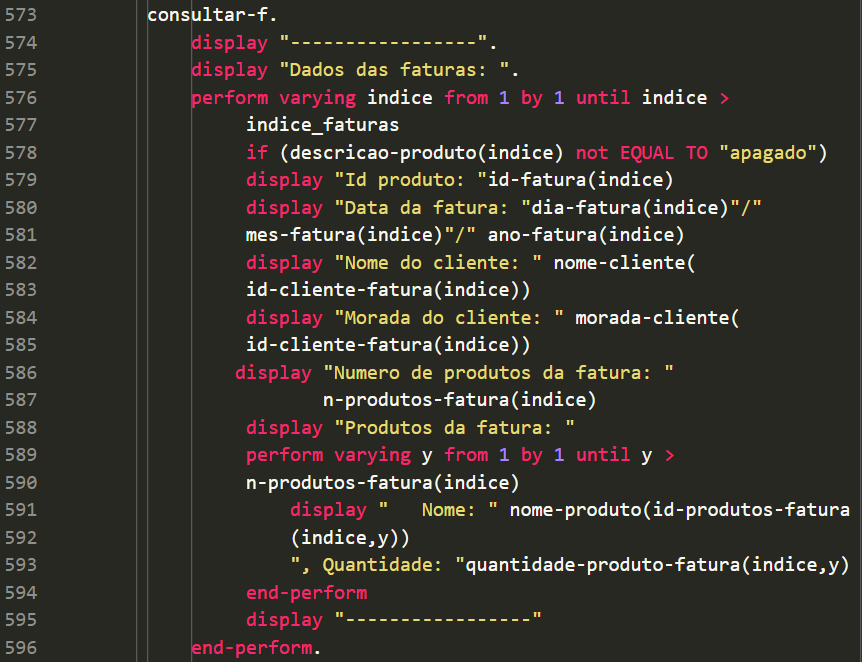


Figura . Consultar Faturas

## Alterar

A opção alterar do menu, permite ao utilizador fazer modificações nas informações de: Clientes, Produtos e Faturas. Dependendo da opção escolhida pelo utilizador dentro do menu de alterar, o programa vai permitir modificar a informação do/a cliente/produto/fatura.

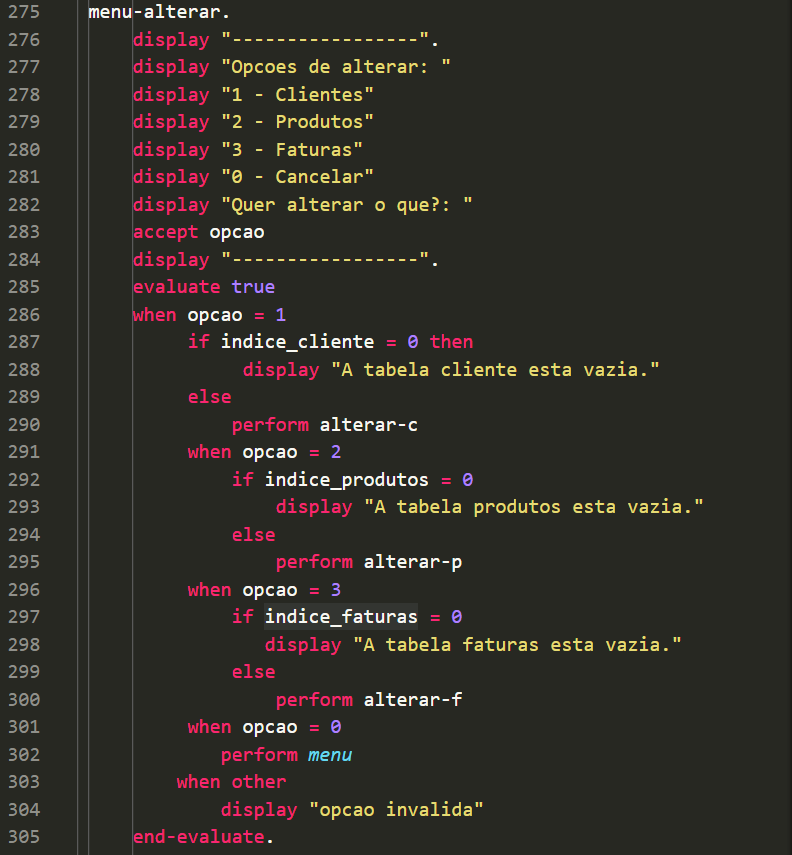


Figura . Menu Alterar

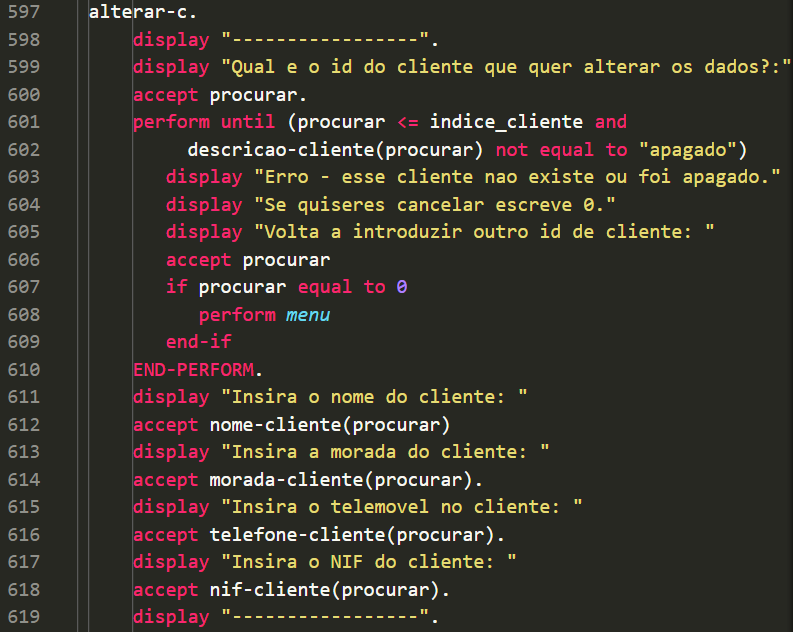


Figura . Alterar clientes

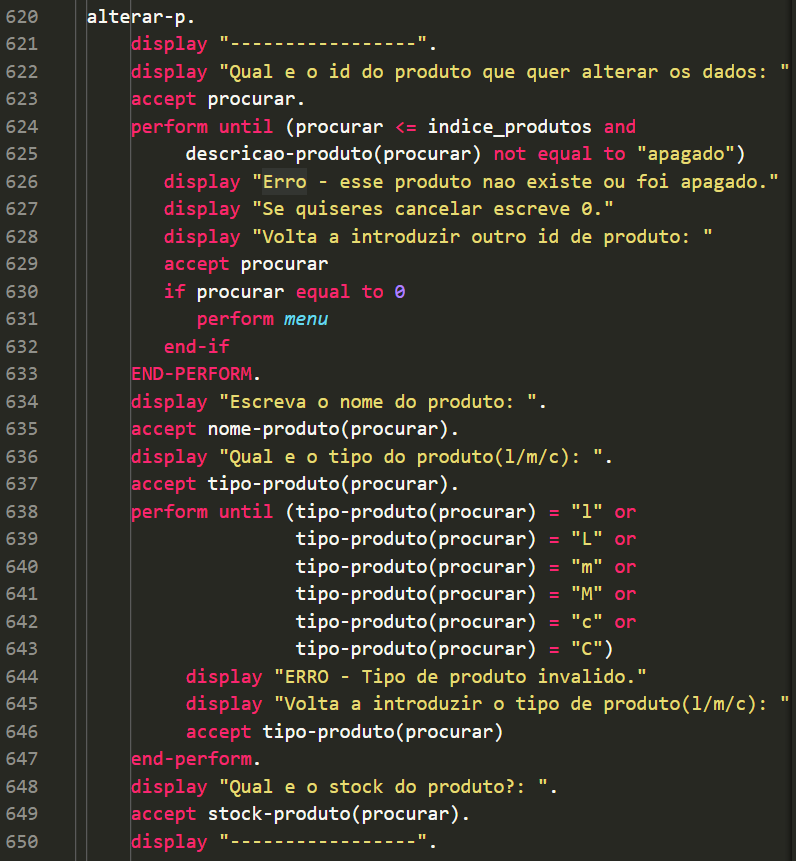


Figura . Alterar produtos

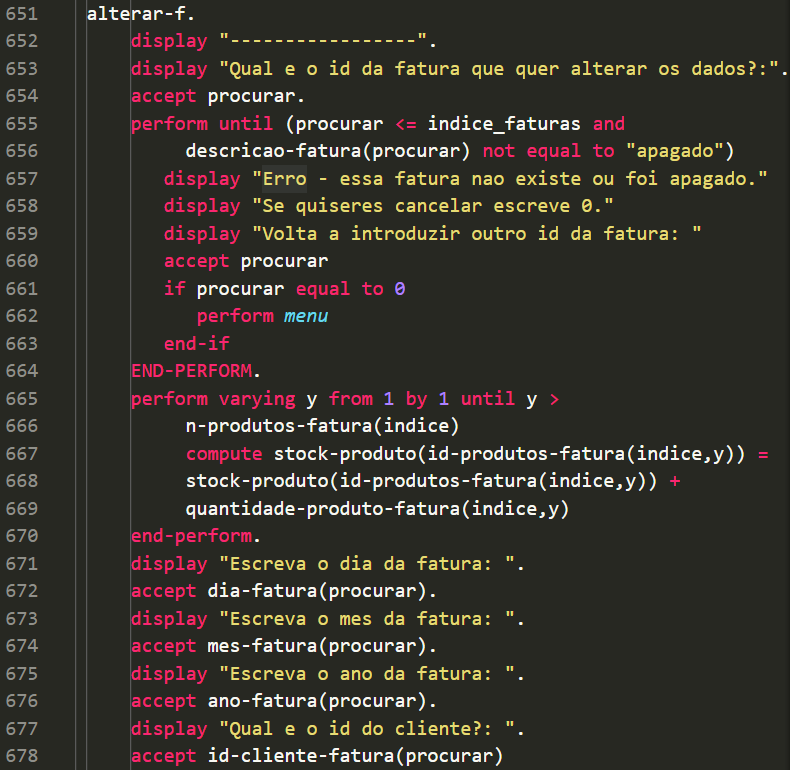


Figura . Alterar faturas (1/3)

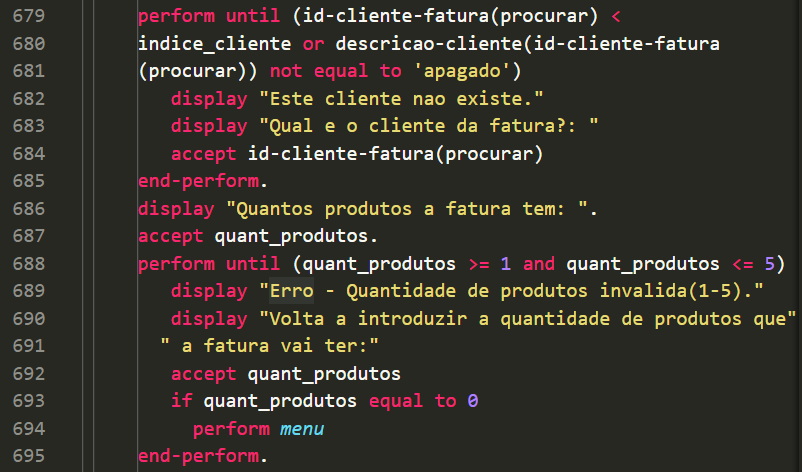


Figura . Alterar faturas (2/3)

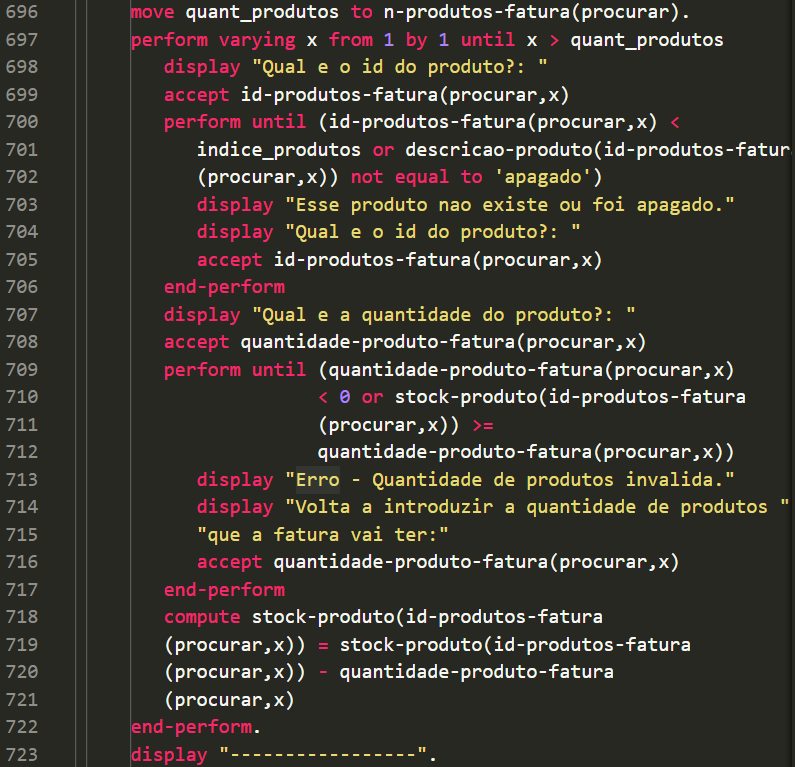


Figura . Alterar faturas (3/3)

## Eliminar

A opção eliminar do menu principal, permite ao utilizador apagar clientes, produtos ou faturas.

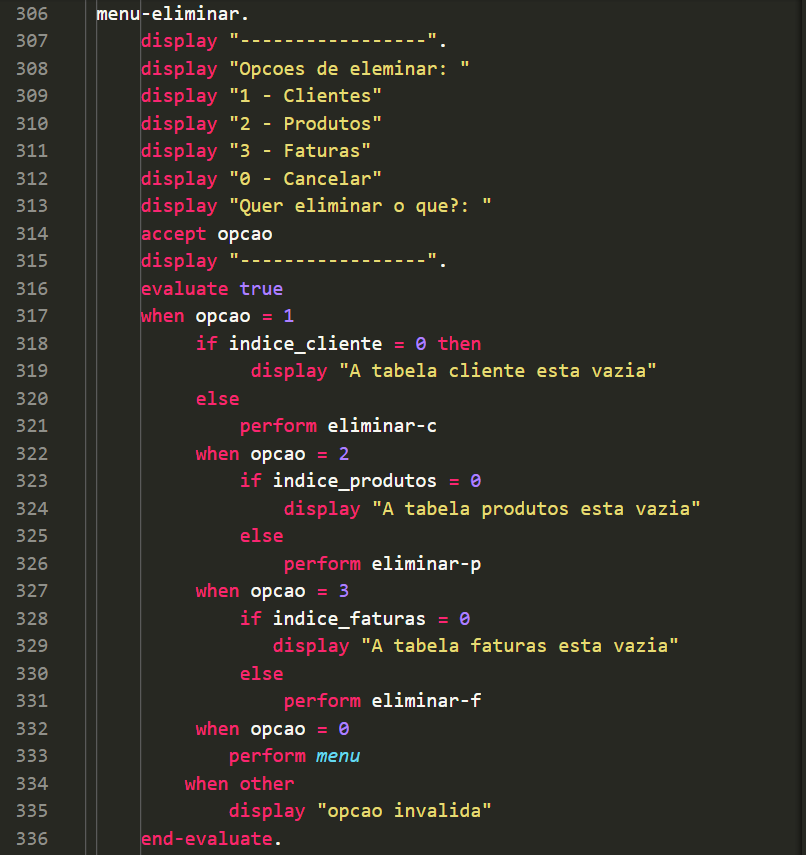


Figura . Menu Eliminar

Este “apagamento” de dados não é mais do que fazer que esse cliente, produtos ou fatura, não esteja disponível, ou seja, é colocado na sua descrição como “apagado”. Como consequência desta descrição, o cliente, produto ou fatura que tenha a essa descrição, não vai poder ser utilizado na criação de faturas, nem vai ser listado com as outras informações inseridas, só vai ser listado na opção “consultar apagados”.

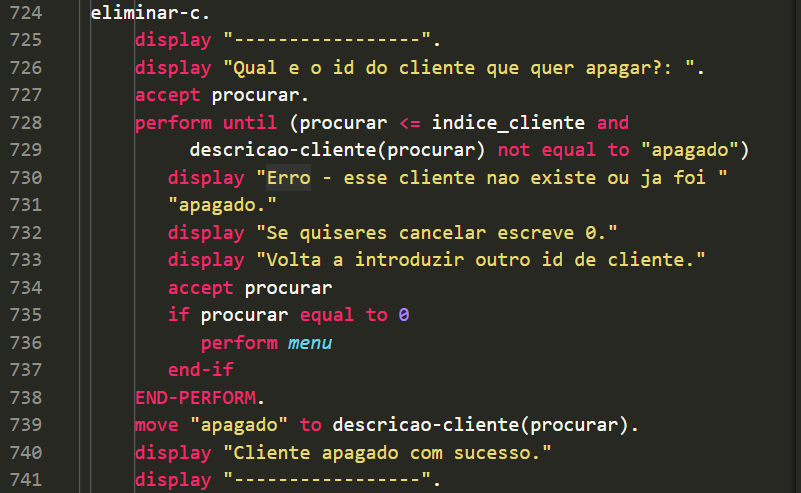


Figura . Eliminar Clientes

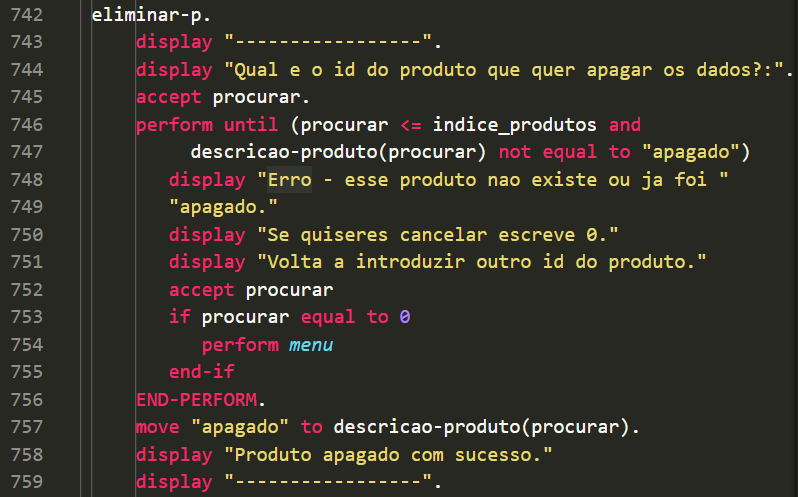


Figura . Eliminar Produtos

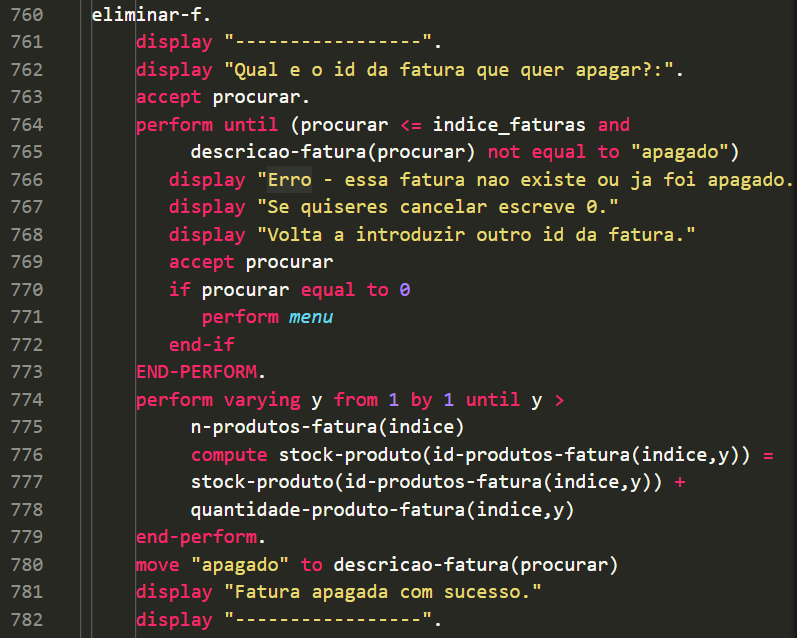


Figura . Eliminar Faturas

## Consultar apagados

Esta opção lista unicamente os elementos apagados, sejam clientes, produtos ou faturas.

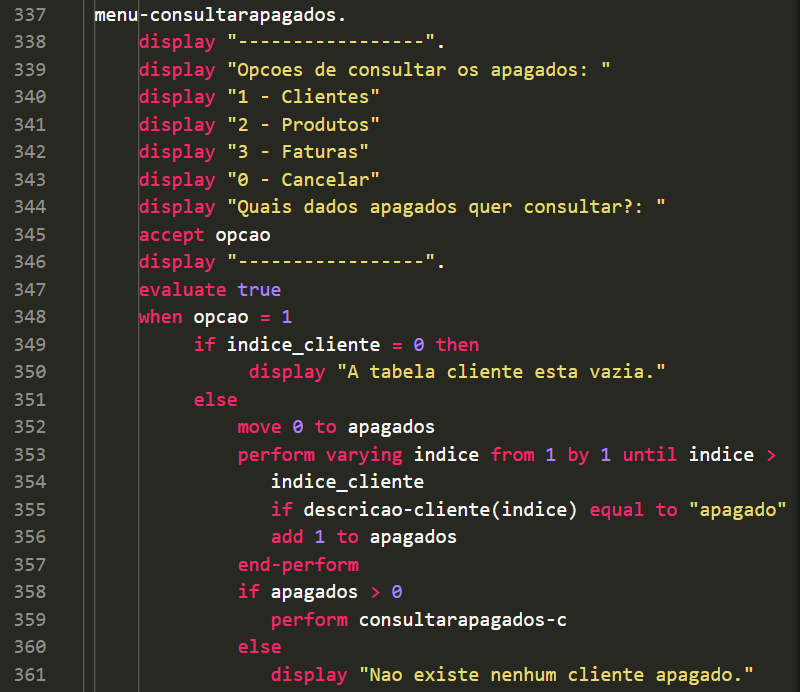


Figura . Menu Consultar Apagados (1/3)

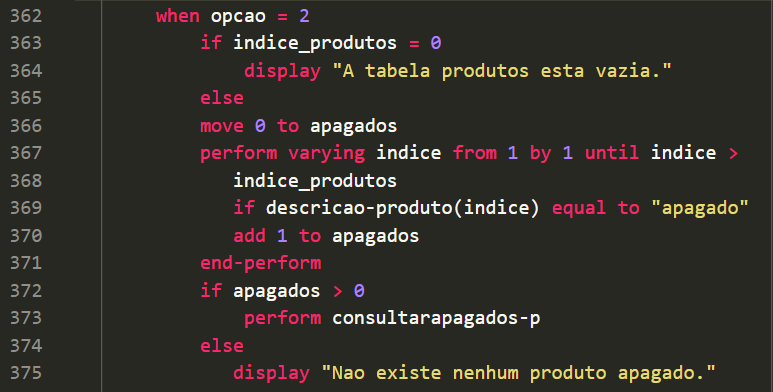


Figura . Menu Consultar Apagados (2/3)

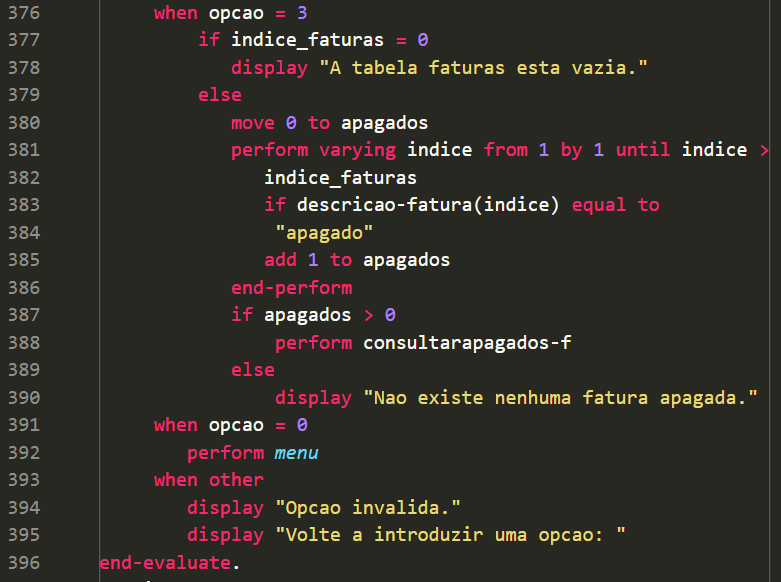


Figura . Menu Consultar Apagados (3/3)

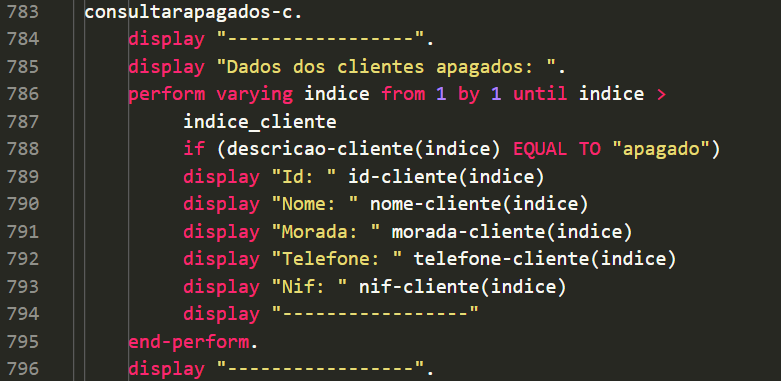


Figura . Consultar Clientes Apagados

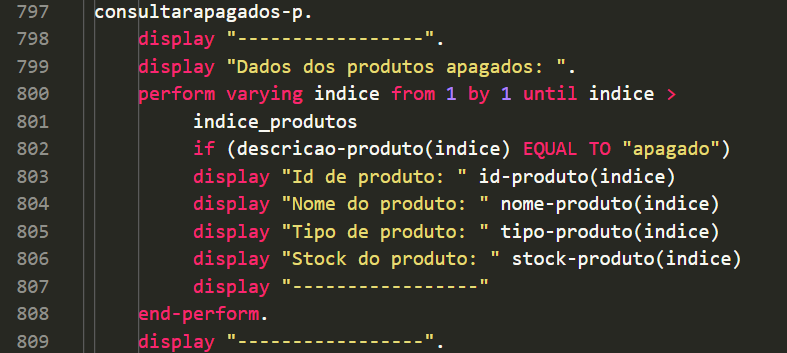


Figura . Consultar Produtos Apagados



Figura . Consultar Faturas Apagadas

## Salvar dados nos ficheiros

A opção de salvar dados faz que o programa guarde as informações introduzidas em ficheiros “.dat”.



Figura . Salvar dados

Desta maneira, o utilizador não terá que inserir as informações previas sempre que volte a utilizar o programa (não se perderam as informações introduzidas).

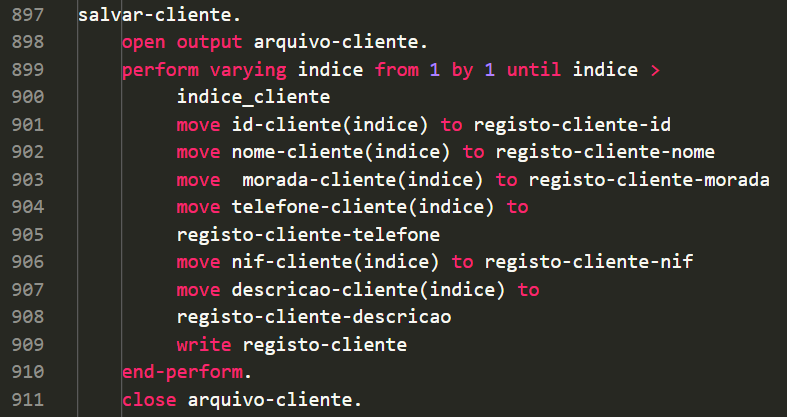


Figura . Salvar Clientes

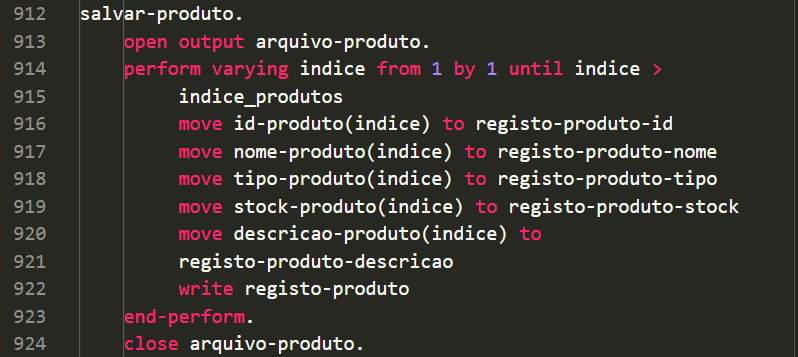


Figura . Salvar Produto

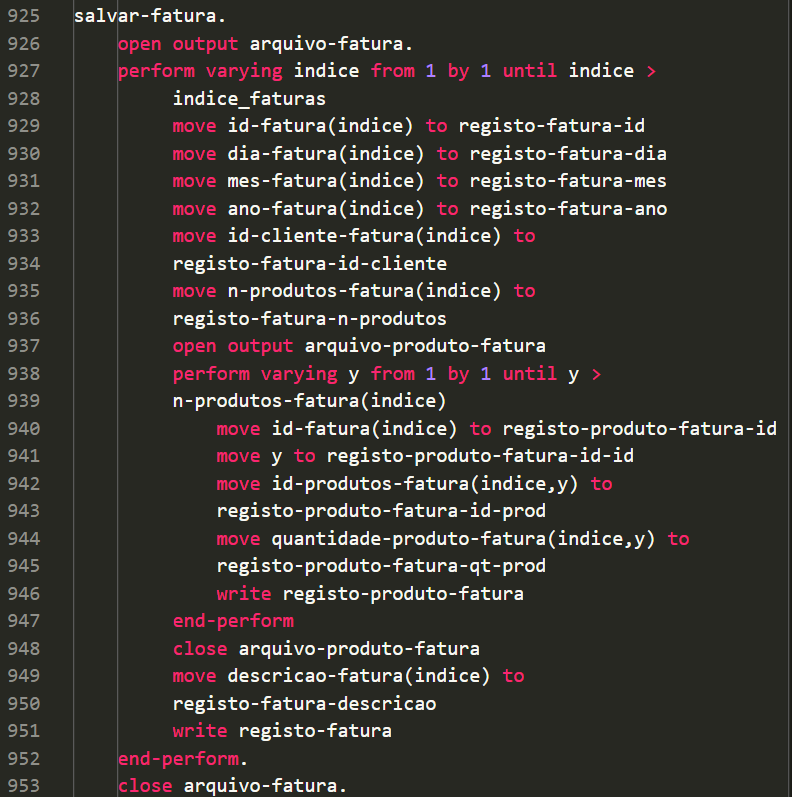


Figura . Salvar Faturas

## Carregar dados nos ficheiros

Esta última opção, permite carregar os dados previamente guardados nos ficheiros .dat ao programa.

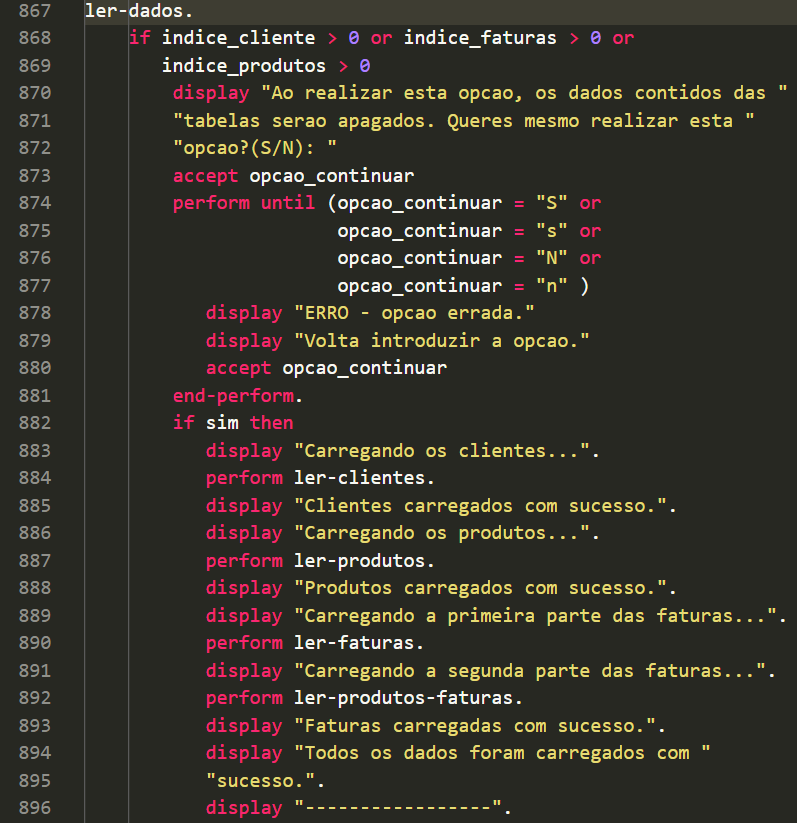


Figura . Carregar dados

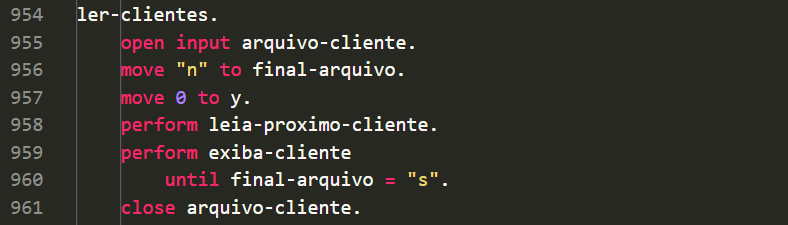


Figura . Carregar Clientes

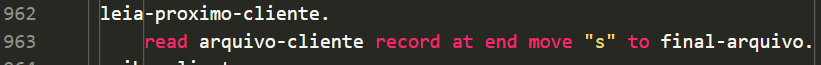


Figura . Ler próximo cliente

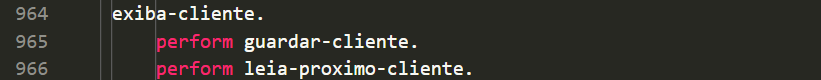


Figura . Exibir cliente

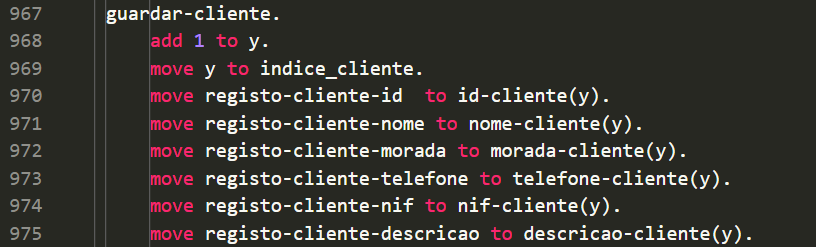


Figura . Guardar cliente

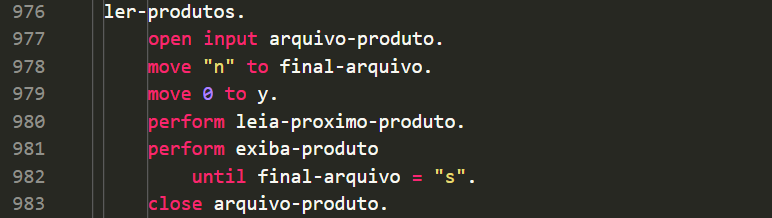


Figura . Carregar produtos



Figura . Ler próximo produto

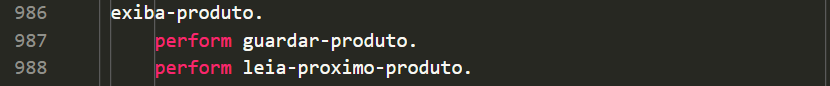


Figura . Exibir produto

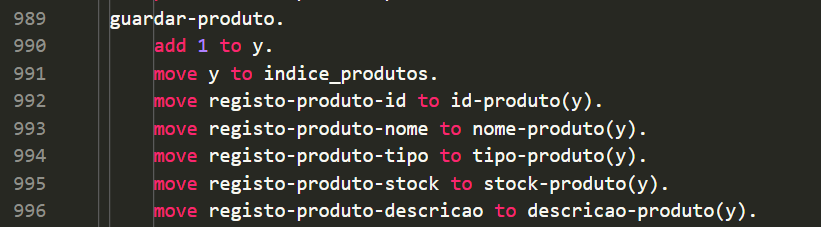


Figura . Guardar produto

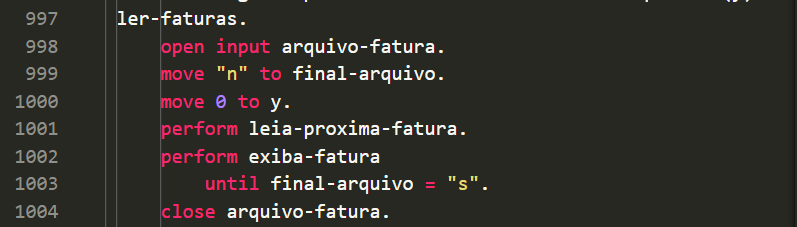


Figura . Carregar faturas



Figura . Ler próxima fatura

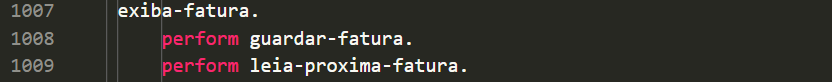


Figura . Exibir fatura

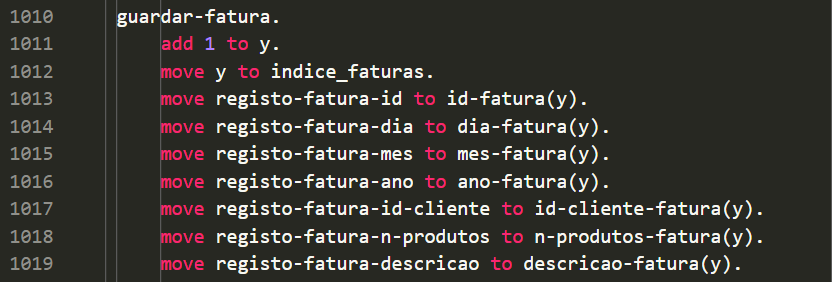


Figura . Guardar fatura

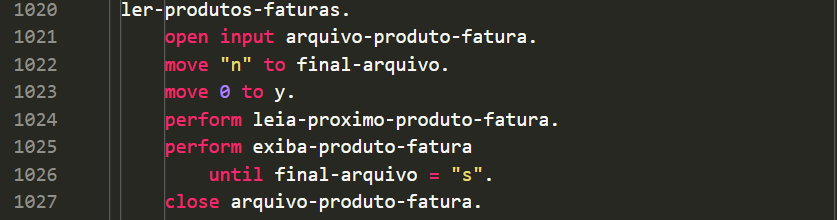


Figura . Ler produtos-faturas

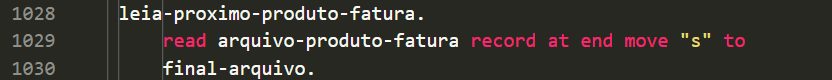


Figura . Ler próximo produto-fatura

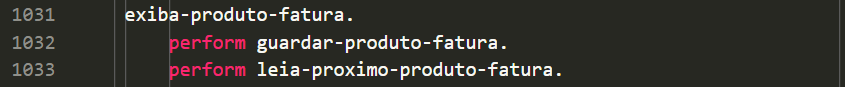


Figura . Exibir produto-fatura

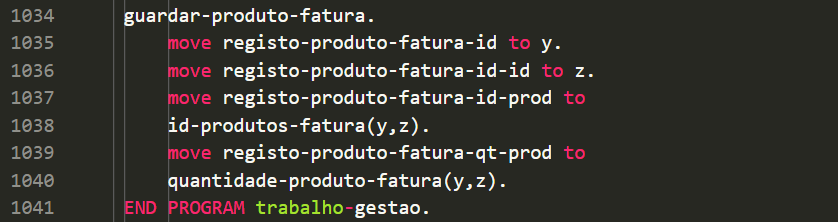


Figura . Guardar produto-fatura

# Conclusão

O nosso objetivo de criar um programa de gestão foi concluído com êxito. Resumindo o projeto, foi criado um programa com a habilidade de introduzir, consultar, alterar, eliminar, consultar eliminados, carregar dados, salvar dados e sair; este programa não foi muito complicado de fazer, mas em algumas partes tivemos de modificar o código porque não conseguimos continuar da maneira que nós queríamos. No entanto, as modificações foram para melhor.

# Bibliografia

* Apontamentos fornecidos nas aulas de ESW;
* Trabalhos feitos previamente.

# Anexos

**Código completo:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Author: Arianna Cicero Nº3, Diogo Marques Nº7

\* Date: Inicio:05/11/2020 Entraga: 19/11/2020

\* Purpose:

\* Tectonics: cobc

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

IDENTIFICATION DIVISION.

PROGRAM-ID. trabalho-gestao.

ENVIRONMENT DIVISION.

INPUT-OUTPUT SECTION.

FILE-CONTROL.

\* Arquivo clientes

SELECT OPTIONAL arquivo-cliente

ASSIGN to "clientes.dat"

ORGANIZATION IS SEQUENTIAL.

\* Arquivo produtos

SELECT OPTIONAL arquivo-produto

ASSIGN TO "produtos.dat"

ORGANIZATION IS SEQUENTIAL.

\* Arquivo faturas

SELECT OPTIONAL arquivo-fatura

ASSIGN TO "faturas.dat"

ORGANIZATION IS SEQUENTIAL.

\* Arquivo produtos da fatura

select OPTIONAL arquivo-produto-fatura

assign to "produto-fatura.dat"

organization is SEQUENTIAL.

DATA DIVISION.

FILE SECTION.

fd arquivo-cliente.

01 registo-cliente.

05 registo-cliente-id pic 99.

05 registo-cliente-nome pic x(50).

05 registo-cliente-morada pic x(50).

05 registo-cliente-telefone pic x(12).

05 registo-cliente-nif pic 9(9).

05 registo-cliente-descricao pic x(50).

fd arquivo-produto.

01 registo-produto.

05 registo-produto-id pic 99.

05 registo-produto-nome pic x(50).

05 registo-produto-tipo pic x(50).

88 compota value "c" "C".

88 marmelada value "m" "M".

88 licor value "l" "L".

05 registo-produto-stock pic 9(3).

05 registo-produto-descricao pic x(50).

fd arquivo-fatura.

01 registo-fatura.

05 registo-fatura-id pic 99.

05 registo-fatura-dia pic 99.

05 registo-fatura-mes pic 99.

05 registo-fatura-ano pic 9999.

05 registo-fatura-id-cliente pic 99.

05 registo-fatura-n-produtos pic 9.

05 registo-fatura-descricao pic x(50).

fd arquivo-produto-fatura.

01 registo-produto-fatura.

05 registo-produto-fatura-id pic 99.

05 registo-produto-fatura-id-id pic 99.

05 registo-produto-fatura-id-prod pic 99.

05 registo-produto-fatura-qt-prod pic 99.

WORKING-STORAGE SECTION.

\*Variavel para ele dizer se quer introduzir mais dados ou voltar para o menu principal

01 opcao\_continuar pic x value space.

88 sim value "s" "S".

88 nao value "n" "N".

\*Estrutura da tabela

01 tabela.

02 clientes occurs 10 times.

03 id-cliente pic 99.

03 nome-cliente pic x(50).

03 morada-cliente pic x(50).

03 telefone-cliente pic x(12).

03 nif-cliente pic 9(9).

03 descricao-cliente pic x(50).

02 faturas occurs 10 times.

03 id-fatura pic 99.

03 dia-fatura pic 99.

03 mes-fatura pic 99.

03 ano-fatura pic 9999.

03 id-cliente-fatura pic 99.

03 n-produtos-fatura pic 9.

03 produtos occurs 5 times indexed by x.

04 id-produtos-fatura pic 99.

04 quantidade-produto-fatura pic 99.

03 descricao-fatura pic x(50).

02 produtos occurs 10 times.

03 id-produto pic 99.

03 nome-produto pic x(50).

03 tipo-produto pic x(50).

88 compota value "c" "C".

88 marmelada value "m" "M".

88 licor value "l" "L".

03 stock-produto pic 9(3).

03 descricao-produto pic x(50).

\*Variavel para guardar o id dos produtos que temos de alterar/apagar

77 procurar pic 99 value zero.

\*Variavel para o indice de procurar

77 indice pic 99 value zero.

\*Variavel da opÃ§Ã£o que o usuario escolhe

77 opcao pic 99 value zero.

\*variavel para terminar o programa

77 termina pic 99 value zero.

\*variavel de quantos produtos a fatura vai ter

77 quant\_produtos pic 99 value zero.

\*Variavel do indice dos clientes

77 indice\_cliente pic 99 value zero.

\*Variavel do indice dos produtos

77 indice\_produtos pic 99 value zero.

\*Variavel do indice de compras/faturas

77 indice\_faturas pic 99 value zero.

\*variavel para ver se existe algum campo apagado nas tabelas

77 apagados pic 99 value zero.

\*variavel de indices

77 y pic 99 value zero.

\*mais uma variavel de indices.

77 z pic 99 value zero.

\*variavel para ver se ja lemos todos os dados dos ficheiros

77 final-arquivo pic x.

PROCEDURE DIVISION.

perform menu until termina = 1

STOP RUN.

menu.

display "Menu".

display "1. Introduzir".

display "2. Consultar".

display "3. Alterar".

display "4. Eliminar".

display "5. Consultar eliminados".

display "6. Salvar dados nos ficheiros".

display "7. Carregar dados dos ficheiros".

display "0. Sair".

display "opcao: ".

accept opcao.

display "-----------------".

evaluate true

when opcao = 1

perform menu-introduzir

when opcao = 2

perform menu-consultar

when opcao = 3

perform menu-alterar

when opcao = 4

perform menu-eliminar

when opcao = 5

perform menu-consultarapagados

when opcao = 6

perform salvar-dados

when opcao = 7

perform ler-dados

when opcao = 0

move 1 to termina

when other

display "Opcao invalida"

display "-----------------"

end-evaluate.

menu-introduzir.

display "-----------------".

display "Opcoes de introduzir: "

display "1 - Clientes"

display "2 - Produtos"

display "3 - Faturas"

display "0 - Cancelar"

display "Quer introduzir o que: "

accept opcao

display "-----------------".

evaluate true

when opcao = 1

if indice\_cliente = 10 then

display "Limite atingido pela tabela"

else

perform introduzir-c

when opcao = 2

if indice\_produtos = 10 then

display "Limite atingido pela tabela"

else

perform introduzir-p

when opcao = 3

if indice\_faturas = 10 then

display "Limite atingido pela tabela"

else

perform introduzir-f

when opcao = 0

perform menu

when other

display "opcao invalida"

end-evaluate.

menu-consultar.

display "-----------------".

display "Opcoes de consultar: "

display "1 - Clientes"

display "2 - Produtos"

display "3 - Faturas"

display "0 - Cancelar"

display "Quer consultar o que: "

accept opcao

display "-----------------".

evaluate true

when opcao = 1

if indice\_cliente = 0 then

display "A tabela cliente esta vazia"

else

move 0 to apagados

perform varying indice from 1 by 1 until indice >

indice\_cliente

if descricao-cliente(indice) not equal to

"apagado"

add 1 to apagados

end-perform

if apagados > 0

perform consultar-c

else

display "A tabela cliente esta vazia"

when opcao = 2

if indice\_produtos = 0

display "A tabela produtos esta vazia"

else

move 0 to apagados

perform varying indice from 1 by 1 until indice >

indice\_produtos

if descricao-produto(indice) not equal to

"apagado"

add 1 to apagados

end-perform

if apagados > 0

perform consultar-p

else

display "A tabela produtos esta vazia"

when opcao = 3

if indice\_faturas = 0

display "A tabela faturas esta vazia"

else

move 0 to apagados

perform varying indice from 1 by 1 until indice >

indice\_faturas

if descricao-fatura(indice) not equal to

"apagado"

add 1 to apagados

end-perform

if apagados > 0

perform consultar-f

else

display "A tabela faturas esta vazia"

when opcao = 0

perform menu

when other

display "opcao invalida"

end-evaluate.

menu-alterar.

display "-----------------".

display "Opcoes de alterar: "

display "1 - Clientes"

display "2 - Produtos"

display "3 - Faturas"

display "0 - Cancelar"

display "Quer alterar o que: "

accept opcao

display "-----------------".

evaluate true

when opcao = 1

if indice\_cliente = 0 then

display "A tabela cliente esta vazia"

else

perform alterar-c

when opcao = 2

if indice\_produtos = 0

display "A tabela produtos esta vazia"

else

perform alterar-p

when opcao = 3

if indice\_faturas = 0

display "A tabela faturas esta vazia"

else

perform alterar-f

when opcao = 0

perform menu

when other

display "opcao invalida"

end-evaluate.

menu-eliminar.

display "-----------------".

display "Opcoes de eleminar: "

display "1 - Clientes"

display "2 - Produtos"

display "3 - Faturas"

display "0 - Cancelar"

display "Quer eliminar o que: "

accept opcao

display "-----------------".

evaluate true

when opcao = 1

if indice\_cliente = 0 then

display "A tabela cliente esta vazia"

else

perform eliminar-c

when opcao = 2

if indice\_produtos = 0

display "A tabela produtos esta vazia"

else

perform eliminar-p

when opcao = 3

if indice\_faturas = 0

display "A tabela faturas esta vazia"

else

perform eliminar-f

when opcao = 0

perform menu

when other

display "opcao invalida"

end-evaluate.

menu-consultarapagados.

display "-----------------".

display "Opcoes de consultar os apagados: "

display "1 - Clientes"

display "2 - Produtos"

display "3 - Faturas"

display "0 - Cancelar"

display "Quer consultar que apagados: "

accept opcao

display "-----------------".

evaluate true

when opcao = 1

if indice\_cliente = 0 then

display "A tabela cliente esta vazia"

else

move 0 to apagados

perform varying indice from 1 by 1 until indice >

indice\_cliente

if descricao-cliente(indice) equal to "apagado"

add 1 to apagados

end-perform

if apagados > 0

perform consultarapagados-c

else

display "Nao existe nenhum cliente apagado"

when opcao = 2

if indice\_produtos = 0

display "A tabela produtos esta vazia"

else

move 0 to apagados

perform varying indice from 1 by 1 until indice >

indice\_produtos

if descricao-produto(indice) equal to "apagado"

add 1 to apagados

end-perform

if apagados > 0

perform consultarapagados-p

else

display "Nao existe nenhum produto apagado"

when opcao = 3

if indice\_faturas = 0

display "A tabela faturas esta vazia"

else

move 0 to apagados

perform varying indice from 1 by 1 until indice >

indice\_faturas

if descricao-fatura(indice) equal to

"apagado"

add 1 to apagados

end-perform

if apagados > 0

perform consultarapagados-f

else

display "Nao existe nenhuma fatura apgada"

when opcao = 0

perform menu

when other

display "opcao invalida"

end-evaluate.

introduzir-c.

add 1 to indice\_cliente.

move indice\_cliente to id-cliente(indice\_cliente).

display "-----------------".

display "Id do cliente: " indice\_cliente.

display "Escreva o nome do cliente: "

accept nome-cliente(indice\_cliente).

display "Escreva a morada do cliente: ".

accept morada-cliente(indice\_cliente).

display "Escreva o telemovel do cliente: ".

accept telefone-cliente(indice\_cliente).

display "Escreva o NIF do cliente: ".

accept nif-cliente(indice\_cliente).

move space to descricao-cliente(indice\_cliente).

display "Queres introduzir mais algum cliente?(S/N)".

accept opcao\_continuar.

perform until (opcao\_continuar = "S" or

opcao\_continuar = "s" or

opcao\_continuar = "N" or

opcao\_continuar = "n" )

display "ERRO - opcao errada"

display "Volta introduzir a opcao"

accept opcao\_continuar

end-perform.

if sim then

perform introduzir-c

end-if.

display "-----------------".

introduzir-p.

add 1 to indice\_produtos.

move indice\_produtos to id-produto(indice\_produtos).

display "-----------------".

display "Id do produto: " indice\_produtos.

display "Escreva o nome do produto: ".

accept nome-produto(indice\_produtos).

display "Qual e o tipo do produto(l/m/c): ".

accept tipo-produto(indice\_produtos).

perform until (tipo-produto(indice\_produtos) = "l" or

tipo-produto(indice\_produtos) = "L" or

tipo-produto(indice\_produtos) = "m" or

tipo-produto(indice\_produtos) = "M" or

tipo-produto(indice\_produtos) = "c" or

tipo-produto(indice\_produtos) = "C")

display "ERRO - Tipo de produto invalido."

display "Volta a introduzir o tipo de produto(l/m/c): "

accept tipo-produto(indice\_produtos)

end-perform.

display "Qual e o stock do produto?: ".

accept stock-produto(indice\_produtos).

move space to descricao-produto(indice\_produtos).

display "Queres introduzir mais algum produto?(S/N)".

accept opcao\_continuar.

perform until (opcao\_continuar = "S" or

opcao\_continuar = "s" or

opcao\_continuar = "N" or

opcao\_continuar = "n" )

display "ERRO - opcao errada"

display "Volta introduzir a opcao"

accept opcao\_continuar

end-perform.

if sim then

perform introduzir-p

end-if.

display "-----------------".

introduzir-f.

add 1 to indice\_faturas.

move indice\_faturas to id-fatura(indice\_faturas)

display "-----------------".

display "Id da fatura: " indice\_faturas.

display "Escreva o dia da fatura: ".

accept dia-fatura(indice\_faturas).

perform until (dia-fatura(indice\_faturas) > 0 and

dia-fatura(indice\_faturas) < 32)

display "Introduziste um dia invalido"

display "envia 0 para cancelar"

display "Qual e o dia da fatura?: "

accept dia-fatura(indice\_faturas)

if dia-fatura(indice\_faturas) equal to 0

compute indice\_faturas = indice\_faturas - 1

perform menu

end-perform.

display "Escreva o mes da fatura: ".

accept mes-fatura(indice\_faturas).

perform until (mes-fatura(indice\_faturas) > 0 and

mes-fatura(indice\_faturas) < 13)

display "O mes que introduziste nao existe"

display "envia 0 para cancelar"

display "Qual e o mes da fatura?: "

accept mes-fatura(indice\_faturas)

if mes-fatura(indice\_faturas) equal to 0

compute indice\_faturas = indice\_faturas - 1

perform menu

END-PERFORM.

display "Escreva o ano da fatura: ".

accept ano-fatura(indice\_faturas).

perform until (ano-fatura(indice\_faturas) > 2009 and

ano-fatura(indice\_faturas) < 2031)

display "O ano que introduziste nao existe"

display "envia 0 para cancelar"

display "Qual e o ano da fatura?: "

accept ano-fatura(indice\_faturas)

if ano-fatura(indice\_faturas) equal to 0

compute indice\_faturas = indice\_faturas - 1

perform menu

END-PERFORM.

display "Qual e o id do cliente?: ".

accept id-cliente-fatura(indice\_faturas)

perform until (id-cliente-fatura(indice\_faturas) <=

indice\_cliente and descricao-cliente(id-cliente-fatura

(indice\_faturas)) not equal to 'apagado')

display "Este cliente nao existe"

display "envia 0 para cancelar"

display "Qual e o id do cliente da fatura?: "

accept id-cliente-fatura(indice\_faturas)

if id-cliente-fatura(indice\_faturas) equal to 0

compute indice\_faturas = indice\_faturas - 1

perform menu

end-perform.

display "Quantos produtos a fatura tem: ".

accept quant\_produtos.

perform until (quant\_produtos >= 1 and quant\_produtos <= 5)

display "Erro - Quantidade de produtos invalida(1-5)."

display "Envia 0 para cancelar a introducao da fatura"

display "Volta a introduzir a quantidade de produtos que"

- " fatura vai ter:"

accept quant\_produtos

if quant\_produtos equal to 0

compute indice\_faturas = indice\_faturas - 1

perform menu

end-perform.

move quant\_produtos to n-produtos-fatura(indice\_faturas).

perform varying x from 1 by 1 until x > quant\_produtos

display "Qual e o id do produto?: "

accept id-produtos-fatura(indice\_faturas,x)

perform until (id-produtos-fatura(indice\_faturas,x) <=

indice\_produtos and descricao-produto(id-produtos-fatura

(indice\_faturas,x)) not equal to 'apagado')

display "Esse produto nao existe ou foi apagado"

display "Para cancelar envia 0"

display "Qual e o id do produto?:"

accept id-produtos-fatura(indice\_faturas,x)

if id-produtos-fatura(indice\_faturas,x) equal to 0

compute indice\_faturas = indice\_faturas - 1

perform menu

end-perform

display "Qual e a quantidade do produto): "

accept quantidade-produto-fatura(indice\_faturas,x)

perform until (quantidade-produto-fatura(indice\_faturas,x)

< 0 or stock-produto(id-produtos-fatura

(indice\_faturas,x)) >=

quantidade-produto-fatura(indice\_faturas,x))

display "Erro - Quantidade de produtos invalida."

display "Envia 0 para cancelar a introducao da fatura"

display "Volta a introduzir a quantidade de produtos "

"que fatura vai ter:"

accept quantidade-produto-fatura(indice\_faturas,x)

if quantidade-produto-fatura(indice\_faturas,x)

equal to 0

compute indice\_faturas = indice\_faturas - 1

perform menu

end-perform

compute stock-produto(id-produtos-fatura

(indice\_faturas,x)) = stock-produto(id-produtos-fatura

(indice\_faturas,x)) - quantidade-produto-fatura

(indice\_faturas,x)

end-perform.

move space to descricao-produto(indice\_produtos).

display "Queres introduzir mais alguma fatura?(S/N)".

accept opcao\_continuar.

perform until (opcao\_continuar = "S" or

opcao\_continuar = "s" or

opcao\_continuar = "N" or

opcao\_continuar = "n" )

display "ERRO - opcao errada"

display "Volta introduzir a opcao"

accept opcao\_continuar

end-perform.

if sim then

perform introduzir-f

end-if.

consultar-c.

display "-----------------".

display "Dados dos clientes: ".

perform varying indice from 1 by 1 until indice >

indice\_cliente

if (descricao-cliente(indice) not EQUAL TO "apagado")

display "Id: " id-cliente(indice)

display "Nome: " nome-cliente(indice)

display "Morada: " morada-cliente(indice)

display "Telefone: " telefone-cliente(indice)

display "Nif: " nif-cliente(indice)

display "-----------------"

end-perform.

consultar-p.

display "-----------------".

display "Dados dos produtos: ".

perform varying indice from 1 by 1 until indice >

indice\_produtos

if (descricao-produto(indice) not EQUAL TO "apagado")

display "Id de produto: " id-produto(indice)

display "Nome do produto: " nome-produto(indice)

display "Tipo de produto: " tipo-produto(indice)

display "Stock do produto: " stock-produto(indice)

display "-----------------"

end-perform.

consultar-f.

display "-----------------".

display "Dados das faturas: ".

perform varying indice from 1 by 1 until indice >

indice\_faturas

if (descricao-produto(indice) not EQUAL TO "apagado")

display "Id produto: "id-fatura(indice)

display "Data da fatura: "dia-fatura(indice)"/"

mes-fatura(indice)"/" ano-fatura(indice)

display "Nome do cliente: " nome-cliente(

id-cliente-fatura(indice))

display "Morada do cliente: " morada-cliente(

id-cliente-fatura(indice))

display "Numero de produtos da fatura: "

n-produtos-fatura(indice)

display "Produtos da fatura: "

perform varying y from 1 by 1 until y >

n-produtos-fatura(indice)

display " Nome: " nome-produto(id-produtos-fatura

(indice,y))

", Quantidade: "quantidade-produto-fatura(indice,y)

end-perform

display "-----------------"

end-perform.

alterar-c.

display "-----------------".

display "Qual e o id do cliente que quer alterar os dados".

accept procurar.

perform until (procurar <= indice\_cliente and

descricao-cliente(procurar) not equal to "apagado")

display "Erro - esse cliente nao existe ou foi apagado"

display "Se quiseres cancelar escreve 0"

display "volta a introduzir outro id de cliente: "

accept procurar

if procurar equal to 0

perform menu

end-if

END-PERFORM.

display "Insira o nome do cliente: "

accept nome-cliente(procurar)

display "Insira a morada do cliente: "

accept morada-cliente(procurar).

display "Insira o telemovel no cliente: "

accept telefone-cliente(procurar).

display "Insira o NIF do cliente: "

accept nif-cliente(procurar).

display "-----------------".

alterar-p.

display "-----------------".

display "Qual e o id do produto que quer alterar os dados".

accept procurar.

perform until (procurar <= indice\_produtos and

descricao-produto(procurar) not equal to "apagado")

display "Erro - esse produto nao existe ou foi apagado"

display "Se quiseres cancelar escreve 0"

display "volta a introduzir outro id de produto"

accept procurar

if procurar equal to 0

perform menu

end-if

END-PERFORM.

display "Escreva o nome do produto: ".

accept nome-produto(procurar).

display "Qual e o tipo do produto(l/m/c): ".

accept tipo-produto(procurar).

perform until (tipo-produto(procurar) = "l" or

tipo-produto(procurar) = "L" or

tipo-produto(procurar) = "m" or

tipo-produto(procurar) = "M" or

tipo-produto(procurar) = "c" or

tipo-produto(procurar) = "C")

display "ERRO - Tipo de produto invalido."

display "Volta a introduzir o tipo de produto(l/m/c): "

accept tipo-produto(procurar)

end-perform.

display "Qual e o stock do produto?: ".

accept stock-produto(procurar).

display "-----------------".

alterar-f.

display "-----------------".

display "Qual e o id da fatura que quer alterar os dados".

accept procurar.

perform until (procurar <= indice\_faturas and

descricao-fatura(procurar) not equal to "apagado")

display "Erro - essa fatura nao existe ou foi apagado"

display "Se quiseres cancelar escreve 0"

display "volta a introduzir outro id da fatura"

accept procurar

if procurar equal to 0

perform menu

end-if

END-PERFORM.

perform varying y from 1 by 1 until y >

n-produtos-fatura(indice)

compute stock-produto(id-produtos-fatura(indice,y)) =

stock-produto(id-produtos-fatura(indice,y)) +

quantidade-produto-fatura(indice,y)

end-perform.

display "Escreva o dia da fatura: ".

accept dia-fatura(procurar).

display "Escreva o mes da fatura: ".

accept mes-fatura(procurar).

display "Escreva o ano da fatura: ".

accept ano-fatura(procurar).

display "Qual e o id do cliente?: ".

accept id-cliente-fatura(procurar)

perform until (id-cliente-fatura(procurar) <

indice\_cliente or descricao-cliente(id-cliente-fatura

(procurar)) not equal to 'apagado')

display "Este cliente nao existe"

display "envia 0 para cancelar"

display "Qual e o cliente da fatura? "

accept id-cliente-fatura(procurar)

if id-cliente-fatura(procurar) equal to 0

perform menu

end-perform.

display "Quantos produtos a fatura tem: ".

accept quant\_produtos.

perform until (quant\_produtos >= 1 and quant\_produtos <= 5)

display "Erro - Quantidade de produtos invalida(1-5)."

display "Envia 0 para cancelar a introducao da fatura"

display "Volta a introduzir a quantidade de produtos que"

- " fatura vai ter:"

accept quant\_produtos

if quant\_produtos equal to 0

perform menu

end-perform.

move quant\_produtos to n-produtos-fatura(procurar).

perform varying x from 1 by 1 until x > quant\_produtos

display "Qual e o id do produto?: "

accept id-produtos-fatura(procurar,x)

perform until (id-produtos-fatura(procurar,x) <

indice\_produtos or descricao-produto(id-produtos-fatura

(procurar,x)) not equal to 'apagado')

display "Esse produto nÃ£o existe ou foi apagado"

display "Para cancelar envia 0"

display "Qual e o id do produto?:"

accept id-produtos-fatura(procurar,x)

if id-produtos-fatura(procurar,x) equal to 0

perform menu

end-perform

display "Qual e a quantidade do produto): "

accept quantidade-produto-fatura(procurar,x)

perform until (quantidade-produto-fatura(procurar,x)

< 0 or stock-produto(id-produtos-fatura

(procurar,x)) >=

quantidade-produto-fatura(procurar,x))

display "Erro - Quantidade de produtos invalida."

display "Envia 0 para cancelar a introducao da fatura"

display "Volta a introduzir a quantidade de produtos "

"que fatura vai ter:"

accept quantidade-produto-fatura(procurar,x)

if quantidade-produto-fatura(procurar,x)

equal to 0

perform menu

end-perform

compute stock-produto(id-produtos-fatura

(procurar,x)) = stock-produto(id-produtos-fatura

(procurar,x)) - quantidade-produto-fatura

(procurar,x)

end-perform.

display "-----------------".

eliminar-c.

display "-----------------".

display "Qual e o id do cliente que quer apagar".

accept procurar.

perform until (procurar <= indice\_cliente and

descricao-cliente(procurar) not equal to "apagado")

display "Erro - esse cliente nao existe ou ja foi "

"apagado"

display "Se quiseres cancelar escreve 0"

display "volta a introduzir outro id de cliente"

accept procurar

if procurar equal to 0

perform menu

end-if

END-PERFORM.

move "apagado" to descricao-cliente(procurar).

display "Cliente apagado com sucesso"

display "-----------------".

eliminar-p.

display "-----------------".

display "Qual e o id do produto que quer apagar os dados".

accept procurar.

perform until (procurar <= indice\_produtos and

descricao-produto(procurar) not equal to "apagado")

display "Erro - esse produto nao existe ou ja foi "

"apagado"

display "Se quiseres cancelar escreve 0"

display "volta a introduzir outro id do produto"

accept procurar

if procurar equal to 0

perform menu

end-if

END-PERFORM.

move "apagado" to descricao-produto(procurar).

display "Produto apagado com sucesso"

display "-----------------".

eliminar-f.

display "-----------------".

display "Qual e o id da fatura que quer apagar".

accept procurar.

perform until (procurar <= indice\_faturas and

descricao-fatura(procurar) not equal to "apagado")

display "Erro - essa fatura nao existe ou ja foi apagado"

display "Se quiseres cancelar escreve 0"

display "volta a introduzir outro id da fatura"

accept procurar

if procurar equal to 0

perform menu

end-if

END-PERFORM.

perform varying y from 1 by 1 until y >

n-produtos-fatura(indice)

compute stock-produto(id-produtos-fatura(indice,y)) =

stock-produto(id-produtos-fatura(indice,y)) +

quantidade-produto-fatura(indice,y)

end-perform.

move "apagado" to descricao-fatura(procurar)

display "Fatura apagada com sucesso"

display "-----------------".

consultarapagados-c.

display "-----------------".

display "Dados dos clientes apagados: ".

perform varying indice from 1 by 1 until indice >

indice\_cliente

if (descricao-cliente(indice) EQUAL TO "apagado")

display "Id: " id-cliente(indice)

display "Nome: " nome-cliente(indice)

display "Morada: " morada-cliente(indice)

display "Telefone: " telefone-cliente(indice)

display "Nif: " nif-cliente(indice)

display "-----------------"

end-perform.

display "-----------------".

consultarapagados-p.

display "-----------------".

display "Dados dos produtos apagados: ".

perform varying indice from 1 by 1 until indice >

indice\_produtos

if (descricao-produto(indice) EQUAL TO "apagado")

display "Id de produto: " id-produto(indice)

display "Nome do produto: " nome-produto(indice)

display "Tipo de produto: " tipo-produto(indice)

display "Stock do produto: " stock-produto(indice)

display "-----------------"

end-perform.

display "-----------------".

consultarapagados-f.

display "-----------------".

display "Dados das faturas apagadas: ".

perform varying indice from 1 by 1 until indice >

indice\_faturas

if (descricao-produto(indice) not EQUAL TO "apagado")

display "Id produto: "id-fatura(indice)

display "Data da fatura: "dia-fatura(indice)"/"

mes-fatura(indice)"/" ano-fatura(indice)

display "Nome do cliente: " nome-cliente(

id-cliente-fatura(indice))

display "Morada do cliente: " morada-cliente(

id-cliente-fatura(indice))

display "Numero de produtos da fatura: "

n-produtos-fatura(indice)

display "Produtos da fatura: "

perform varying y from 1 by 1 until y >

n-produtos-fatura(indice)

display " Nome: " nome-produto(id-produtos-fatura

(indice,y))

", Quantidade: "quantidade-produto-fatura(indice,y)

end-perform

display "-----------------"

end-perform.

display "-----------------".

salvar-dados.

display "Ao realizar esta opcao os dados contidos nos "

"ficheiros serao apagados. Queres mesmo realizar esta opcao?"

"(S/N)"

accept opcao\_continuar

perform until (opcao\_continuar = "S" or

opcao\_continuar = "s" or

opcao\_continuar = "N" or

opcao\_continuar = "n" )

display "ERRO - opcao errada"

display "Volta introduzir a opcao"

accept opcao\_continuar

end-perform.

if sim then

display "Salvando clientes...".

perform salvar-cliente.

display "Clientes salvos com sucesso".

display "Salvando produtos...".

perform salvar-produto.

display "Produtos salvos com sucesso".

display "salvando faturas...".

perform salvar-fatura.

display "Faturas salvas com sucesso".

display "Todos os dados foram salvos.".

display "-----------------".

ler-dados.

display "Carregando os clientes...".

perform ler-clientes.

display "Clientes carregados com sucesso".

display "Carregando os produtos...".

perform ler-produtos.

display "Produtos carregados com sucesso".

display "Carregando a primeira parte das faturas...".

perform ler-faturas.

display "Carregando a segunda parte das faturas...".

perform ler-produtos-faturas.

display "Faturas carregadas com sucesso".

display "Todos os dados foram carregados com sucesso".

display "-----------------".

salvar-cliente.

open extend arquivo-cliente.

delete arquivo-cliente.

perform varying indice from 1 by 1 until indice >

indice\_cliente

move id-cliente(indice) to registo-cliente-id

move nome-cliente(indice) to registo-cliente-nome

move morada-cliente(indice) to registo-cliente-morada

move telefone-cliente(indice) to

registo-cliente-telefone

move nif-cliente(indice) to registo-cliente-nif

move descricao-cliente(indice) to

registo-cliente-descricao

write registo-cliente

end-perform.

close arquivo-cliente.

salvar-produto.

open extend arquivo-produto.

delete arquivo-produto.

perform varying indice from 1 by 1 until indice >

indice\_produtos

move id-produto(indice) to registo-produto-id

move nome-produto(indice) to registo-produto-nome

move tipo-produto(indice) to registo-produto-tipo

move stock-produto(indice) to registo-produto-stock

move descricao-produto(indice) to

registo-produto-descricao

write registo-produto

end-perform.

close arquivo-produto.

salvar-fatura.

open extend arquivo-fatura.

open extend arquivo-produto-fatura

delete arquivo-fatura.

delete arquivo-produto-fatura.

perform varying indice from 1 by 1 until indice >

indice\_faturas

move id-fatura(indice) to registo-fatura-id

move dia-fatura(indice) to registo-fatura-dia

move mes-fatura(indice) to registo-fatura-mes

move ano-fatura(indice) to registo-fatura-ano

move id-cliente-fatura(indice) to

registo-fatura-id-cliente

move n-produtos-fatura(indice) to

registo-fatura-n-produtos

perform varying y from 1 by 1 until y >

n-produtos-fatura(indice)

move id-fatura(indice) to registo-produto-fatura-id

move y to registo-produto-fatura-id-id

move id-produtos-fatura(indice,y) to

registo-produto-fatura-id-prod

move quantidade-produto-fatura(indice,y) to

registo-produto-fatura-qt-prod

write registo-produto-fatura

end-perform

move descricao-fatura(indice) to

registo-fatura-descricao

write registo-fatura

end-perform.

close arquivo-produto-fatura

close arquivo-fatura.

ler-clientes.

open input arquivo-cliente.

move "n" to final-arquivo.

move 0 to y.

perform leia-proximo-cliente.

perform exiba-cliente

until final-arquivo = "s".

close arquivo-cliente.

leia-proximo-cliente.

read arquivo-cliente record at end move "s" to final-arquivo.

exiba-cliente.

perform guardar-cliente.

perform leia-proximo-cliente.

guardar-cliente.

add 1 to y.

move y to indice\_cliente.

move registo-cliente-id to id-cliente(y).

move registo-cliente-nome to nome-cliente(y).

move registo-cliente-morada to morada-cliente(y).

move registo-cliente-telefone to telefone-cliente(y).

move registo-cliente-nif to nif-cliente(y).

move registo-cliente-descricao to descricao-cliente(y).

ler-produtos.

open input arquivo-produto.

move "n" to final-arquivo.

move 0 to y.

perform leia-proximo-produto.

perform exiba-produto

until final-arquivo = "s".

close arquivo-produto.

leia-proximo-produto.

read arquivo-produto record at end move "s" to final-arquivo.

exiba-produto.

perform guardar-produto.

perform leia-proximo-produto.

guardar-produto.

add 1 to y.

move y to indice\_produtos.

move registo-produto-id to id-produto(y).

move registo-produto-nome to nome-produto(y).

move registo-produto-tipo to tipo-produto(y).

move registo-produto-stock to stock-produto(y).

move registo-produto-descricao to descricao-produto(y).

ler-faturas.

open input arquivo-fatura.

move "n" to final-arquivo.

move 0 to y.

perform leia-proxima-fatura.

perform exiba-fatura

until final-arquivo = "s".

close arquivo-fatura.

leia-proxima-fatura.

read arquivo-fatura record at end move "s" to final-arquivo.

exiba-fatura.

perform guardar-fatura.

perform leia-proxima-fatura.

guardar-fatura.

add 1 to y.

move y to indice\_faturas.

move registo-fatura-id to id-fatura(y).

move registo-fatura-dia to dia-fatura(y).

move registo-fatura-mes to mes-fatura(y).

move registo-fatura-ano to ano-fatura(y).

move registo-fatura-id-cliente to id-cliente-fatura(y).

move registo-fatura-n-produtos to n-produtos-fatura(y).

move registo-fatura-descricao to descricao-fatura(y).

ler-produtos-faturas.

open input arquivo-produto-fatura.

move "n" to final-arquivo.

move 0 to y.

perform leia-proximo-produto-fatura.

perform exiba-produto-fatura

until final-arquivo = "s".

close arquivo-produto-fatura.

leia-proximo-produto-fatura.

read arquivo-produto-fatura record at end move "s" to

final-arquivo.

exiba-produto-fatura.

perform guardar-produto-fatura.

perform leia-proximo-produto-fatura.

guardar-produto-fatura.

move registo-produto-fatura-id to y.

move registo-produto-fatura-id-id to z.

move registo-produto-fatura-id-prod to

id-produtos-fatura(y,z).

move registo-produto-fatura-qt-prod to

quantidade-produto-fatura(y,z).

END PROGRAM trabalho-gestao.