

**UNIDADE CURRICULAR:**

Introdução á Programação

**DOCENTE:**

Fábio Mendoça

**DATA:**

24 de Janeiro de 2023

**RELATÓRIO DE MÁQUINA PARA CONTROLO AUTOMÁTICO DE ACESSOS EM PARQUE DE ESTACIONAMENTO**

**Jackeline Câmara Nº2058122**

**Miguel Peñaranda Nº2019122**

**Curso Técnico Superior Profissional em Tecnologias e Programação de Sistemas de Informação**

**ÍNDICE**

Introdução…………………………………………………………………..…………...1

Desenvolvimento do Programa........................................................................................2

Anexos.............................................................................................................................11

Parte I – Fluxograma.......................................................................................... 11

Parte II – Fluxograma........................................................................................ 12

Parte III – Fluxograma....................................................................................... 13

Parte IV – Fluxograma....................................................................................... 14

Parte V – Fluxograma........................................................................................ 15

Parte VI – Fluxograma....................................................................................... 16

Parte VII– Fluxograma....................................................................................... 17

Parte VIII – Fluxograma..................................................................................... 18

Parte IX – Fluxograma........................................................................................ 19

Parte X – Fluxograma......................................................................................... 20

Parte XI – Fluxograma........................................................................................ 21

Parte XII – Fluxograma....................................................................................... 22

Parte XIII – Fluxograma..................................................................................... 23

Parte XIV – Fluxograma..................................................................................... 24

Parte XV– Fluxograma....................................................................................... 25

Parte XVI– Fluxograma...................................................................................... 26

Parte XVII– Fluxograma..................................................................................... 27

Parte XVIII – Fluxograma................................................................................... 28

Conclusão....................................................................................................................... 29

**AGRADECIMENTOS**

Para iniciar, queremos agradecer ao docente Fábio Mendoça, que fez um grande esforço em nos ensinar o conteúdo necessário para nos adentrar no mundo da programação. Alem disso, sempre estive disposto para esclarecer a nossas dúvidas e motivar-nos a desenvolver os conhecimentos adquiridos durante as aulas.

1. **INTRODUÇÃO**

No seguinte relatório, será apresentado o projeto atribuído pelo docente Fábio Mendoça na unidade curricular Introdução á Programação.

O projeto consiste na criação de um programa que visa simular o funcionamento de uma máquina de controlo de acessos e gestão de ocupação de lugares de um parque de estacionamento. Este programa devera ter as seguintes características:

* Dois andares, cada um com um número específico de lugares e um custo por minuto diferente
* Implementação de uma opção "Estacionar" no menu principal, permitindo ao utilizador escolher um andar e um lugar para estacionar o seu veículo
* Implementação de uma opção "Sair" no menu principal, permitindo ao utilizador retirar o seu veículo do parque de estacionamento e fazer o pagamento pelo tempo estacionado
* Implementação de uma opção “Administrador” no menu principal permitindo ao administrador aceder à informação privilegiada, tal como ver andares, valor acumulado durante o dia e uma opção que define o fecho ou abertura do parking

Este trabalho tem como objetivos consolidar os conhecimentos adquiridos na análise de problemas, escolher estruturas de dados e codificar algoritmos adequados para resolver problemas específicos.

1. **DESENVOLMIENTO DO PROGRAMA**

Uma imagem com texto

Descrição gerada automaticamente

**Fig 1. Apresentação do Menu Principal**

Para começar, criamos um tuplo que contem todos os menus que vão ser usados ao longo de programa, os tuplo principais são: “Menu Principal, Menu Sair, Menu Pagar e Menu Administrador”. Existem dois tipos de menus, os menus que apenas tem caracteres para mostrar as funções dos mesmos e os menus que contem caracteres mutáveis.

Uma imagem com texto

Descrição gerada automaticamente

**Fig 2. Condição para não imprimir os menus que contem caracteres mutáveis.**

Para dar início ao programa, temos uma variável “menu\_a\_mostrar= 0”, que vai ser usada para imprimir os menus. Deste modo, iniciamos o mesmo com um ciclo while e criamos uma condição para não imprimir os menus que contem caracteres mutáveis.

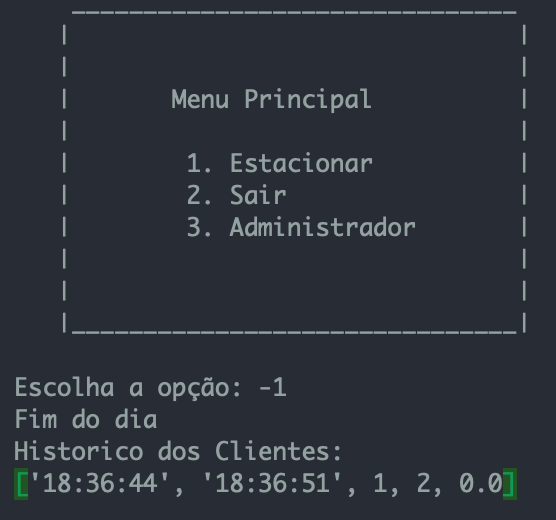
Uma imagem com texto

Descrição gerada automaticamente

**Fig 3. Apresentação do funcionamento das opções dos menus**

Ao longo do programa, o menu a mostrar é sempre atualizado através da variável “opção\_escolhida”, que se trata de um input a pedir ao usuário para escolher uma opção do menu. Caso o usuário escolha uma opção que não existe será apresentada uma mensagem de erro.

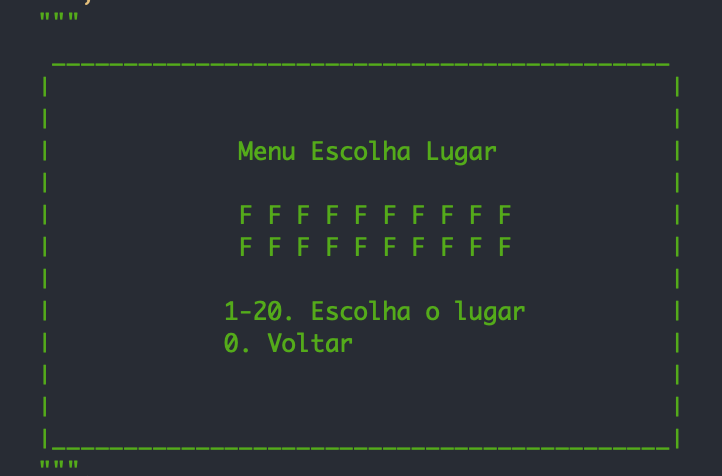
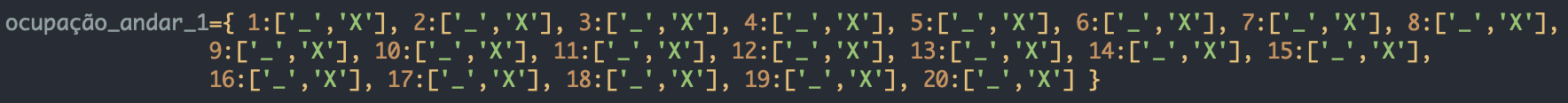
A forma como o usuário pode-se movimentar pelos menus aplica da mesma forma para todos os menus, mas alguns destes têm uma forma personalizada de pedir ao usuário que introduza a informação em função do que é pedido.

Uma imagem com texto

Descrição gerada automaticamente

**Fig 4. Histórico de Clientes**

No menu principal, existe uma função oculta que aparece quando o usuário introduze “-1” e esta consiste em apresentar um histórico e dar fim ao dia. Isto é, temos uma função para ordenar a informação da seguinte maneira: “Hora de Entrada, Hora de Saída, Andar, Lugar e Valor Pagado”. Mas também, um ciclo while para imprimir cada histórico que tem sido a adicionado ao “dicionário\_historicos”.

 Uma imagem com texto

Descrição gerada automaticamente

**Fig 5. Menu Escolha Lugar**

Uma imagem com texto

Descrição gerada automaticamente

**Fig 6. Funcionamento do Menu Escolha lugar**

O menu escolha lugar 1 e 2, tem caracteres mutáveis que são designados pela letra “F” e têm o mesmo método para mostrar o parking ao usuário.

Deste modo, quando o usuário acede ao “Menu Escolha Lugar”, os “F” são substituídos através de um ciclo for, que percorre o tuplo e verifica as posições em que se encontram ubicados os nossos caracteres mutáveis, sendo assim substituídos, neste caso, pela primeira posição de cada chave (corresponde aos números de cada lugar) do dicionário “ocupação\_andar\_1”.

Uma imagem com texto

Descrição gerada automaticamente

**Fig 7. Funcionamento do Menu Escolha lugar**

Quando o usuário escolhe o lugar, é verificado através de uma condição se o lugar se encontra realmente disponível e se for o caso é atualizada a informação que contem o respetivo lugar dentro do dicionário. Esta informação consiste em: “ocupação do lugar (X), início da cronometragem e número de cliente”.

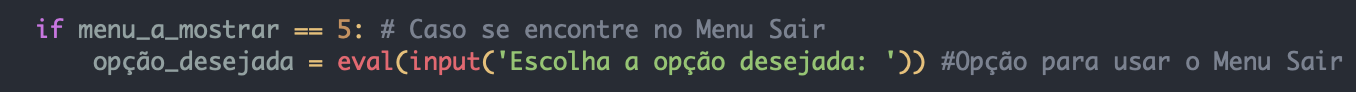
O número do cliente é criado com o objetivo de levar o controlo dos clientes que têm estacionado no parking durante o dia. Por tanto, este é usado como chave do “dicionário\_historicos”, para guardar informação que vai ser adicionada após cada estacionamento, por exemplo: “hora de entrada, andar selecionado e lugar selecionado”.

Alem disso, temos variáveis que levam o controle da quantidade de veículos que se encontram ocupando os lugares de um respetivo andar. De tal maneira, se um andar se encontrar cheio, então monstra uma mensagem ao usuário proporcionando-lhe está informação.

Uma imagem com texto

Descrição gerada automaticamente

**Fig 8. Apresentação do Menu Sair**



**Fig 9. Menu Sair**

Uma imagem com texto

Descrição gerada automaticamente

**Fig 10. Opções do Menu Sair**

No caso de o menu sair, temos um novo input “opção\_desejada” criado apenas para este menu. Se a opção desejada for 1, então é pedido ao usuário que forneça o andar e o lugar em que estacionou, mas se a informação indicada for incorreta (não existir o andar ou lugar indicados, ou também a informação corresponder a um lugar que não se encontra ocupado), então é apresentada uma mensagem de erro.

Uma imagem com texto

Descrição gerada automaticamente

**Fig 11. Verificação da informação forneçida pelo utilizador**

No caso de a informação fornecida estiver correta e corresponder a um lugar ocupado, o usuário é direcionado ao menu pagar.

Uma imagem com texto

Descrição gerada automaticamente

**Fig 11. Funcionamento do Menu Pagar**

O nosso menu pagar começa por calcular o valor a pagar do respetivo andar e lugar selecionado, sendo assim que o andar 1 tem um custo de 0.02€ por segundo e o andar 2 tem um custo de 0.01€ por segundo.

Uma imagem com texto

Descrição gerada automaticamente

**Fig 12. Ciclo para o Menu Pagar**

Após calcular valor a pagar, é criada duas variáveis fundamentais para o funcionamento do menu pagar, estas variáveis são “valor\_introduzido” e “total\_a\_pagar”.

A variável “valor\_introduzido” tem como objetivo mostrar ao usuário o valor que tenha introduzido e subtrair esse valor à variável “total\_a\_pagar”, assim atualizando o valor que aparece no menu, procurando mostrar o resto que falta por pagar ao usuário.

A variável “total\_a\_pagar” tem como objetivo ser usada para substituir o caracter mutável do menu pagar pelo valor a pagar, proporcionando-lhe assim informação necessária ao usuário.

Uma imagem com texto

Descrição gerada automaticamente

**Fig 13. Apresentação da execução do Menu Pagar**



**Fig 14. Valor a pagar**

Para o menu pagar, temos um input diferente com o objetivo dar as instruções necessárias ao usuário. Isto é, o input permite ao utilizador introduzir moedas em cêntimos ou voltar ao menu sair.

Alem disso, temos uma condição que define que só podem ser inseridas moedas que existam, tal como 0.01€, 0.10€ ou 0.50€ e mais. Por conseguinte, cada vez que é inserida uma moeda, o valor da mesma é somado à variável “valor\_introduzido”; também é atualizado o valor do total a pagar que se monstra no menu pagar e mostra-se ao usuário uma mensagem que indica o valor que foi inserido.

Uma imagem com texto

Descrição gerada automaticamente

**Fig 15. Calcular o Troco**

Uma vez que o valor introduzido é maior do que o valor a pagar, calcula-se o troco, soma-se o valor pagado uma variável que contem o dinheiro acumulado durante o dia e o menu a mostrar atualiza-se ao talão.

Uma imagem com texto

Descrição gerada automaticamente

**Fig 16. Actualização do estado do lugar ocupado a disponível**

Após finalizar o pagamento, também temos duas condições que identificam o andar em que foi estacionado, para assim atualizar o estado do lugar ocupado a disponível; e adicionando informação á chave que corresponde ao cliente que retirou o seu veículo. Esta informação trata-se da hora de saída e o valor pagado pelo cliente.

Também temos que, as variáveis que controlam a quantidade de veículos que existem estacionados no respetivo andar é atualizada.

Uma imagem com texto

Descrição gerada automaticamente

**Fig 17. Apresentação do Talão**

Uma imagem com texto

Descrição gerada automaticamente

**Fig 18. Informação do Talão**

No talão usamos o mesmo método para percorre os tuplos e substituir os caracteres mutáveis, sendo estes trocados pela informação correspondente ao seu enunciado. Por exemplo: o andar, o lugar, hora de entrada, hora de saída e preço.

Uma imagem com texto

Descrição gerada automaticamente

**Fig 19. Funcionamento no Menu Administrador**

No caso do menu administrador encontramos funções privilegiadas, que consistem em ver os dois andares ao mesmo tempo, ver o valor acumulado e a capacidade de encerrar/abrir o parking.

Dentro deste menu, temos que se a opção escolhida for igual 0 e o parking estiver aberto, consegues voltar às funcionalidades do parking, mas se for o caso contrário monstra uma mensagem indicando que o parking se encontra fechado.

Uma imagem com texto

Descrição gerada automaticamente

**Fig 20. Apresentação do Menu Andares do Menu Administrador**

Alem disso, se a opção escolhida for igual a 1 o administrador consegue ver os dois andares ao mesmo tempo.

Uma imagem com texto

Descrição gerada automaticamente

Uma imagem com texto

Descrição gerada automaticamente

**Fig. 21 Funcionamento do Valor Acumulado do Menu Administrador**

Também temos, que se a opção escolhida for igual a 2 é pedido ao usuário a senha para aceder a esta função, por conseguinte mostra-se o valor acumulado.

Uma imagem com texto

Descrição gerada automaticamente

**Uma imagem com texto

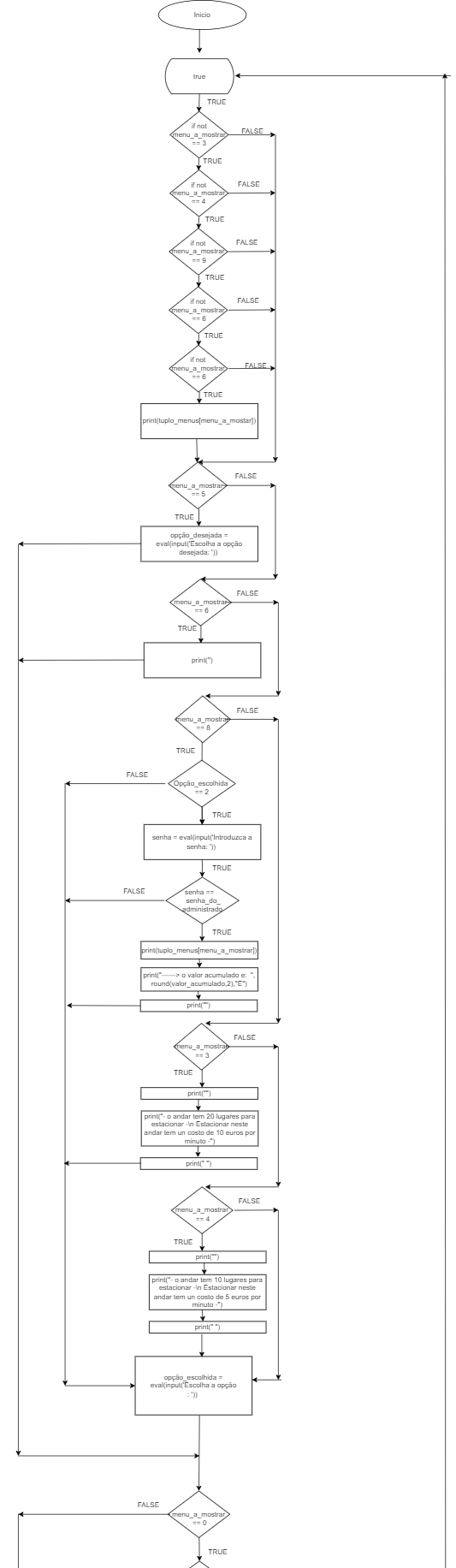
Descrição gerada automaticamente**

**Fig. 22 Abrir ou Fechar o Parking**

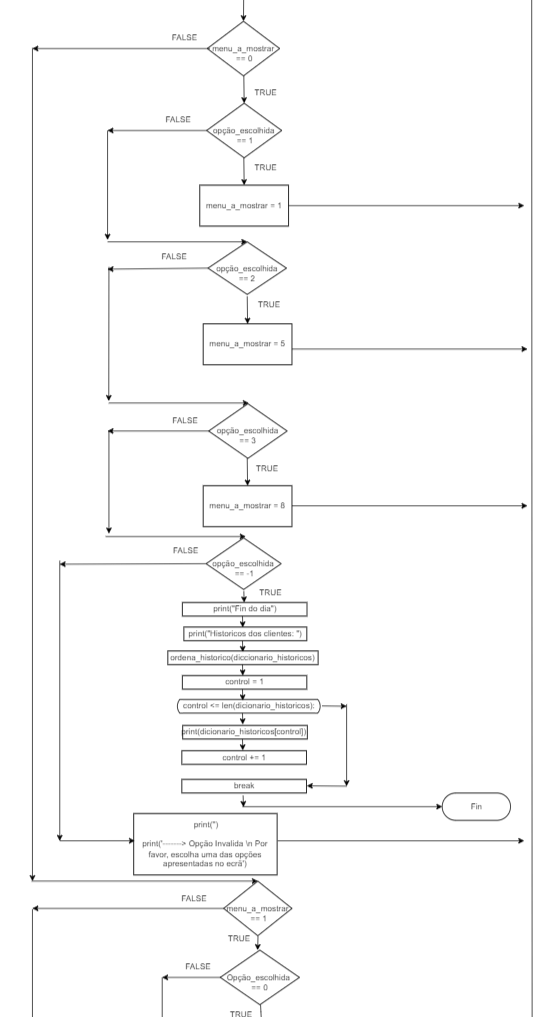
Finalizando com o menu administrador, temos que se a opção escolhida for igual a 3 é pedido ao usuário a senha para aceder a esta função, por conseguinte através da variável “iniciar\_ou\_parar” conseguimos definir se o parque se encontra aberto ou encerrado, alternando assim entre “True e False” cada vez que a função seja usada. Por tanto, caso a variável seja igual a True o parking encontra-se aberto, caso contrário encontra-se fechado

**ANEXOS**

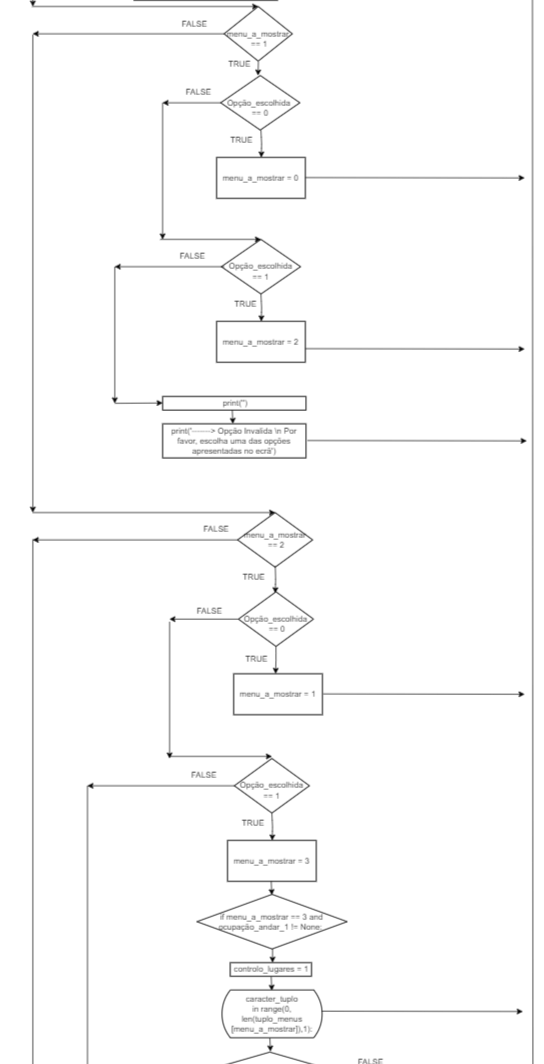
* 1. **PARTE I - FLUXOGRAMA**



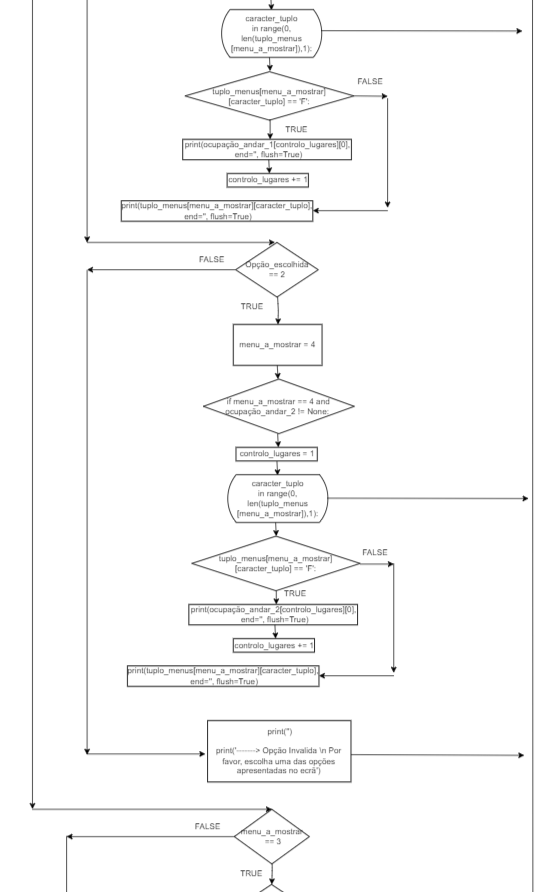
* 1. **PARTE II - FLUXOGRAMA**



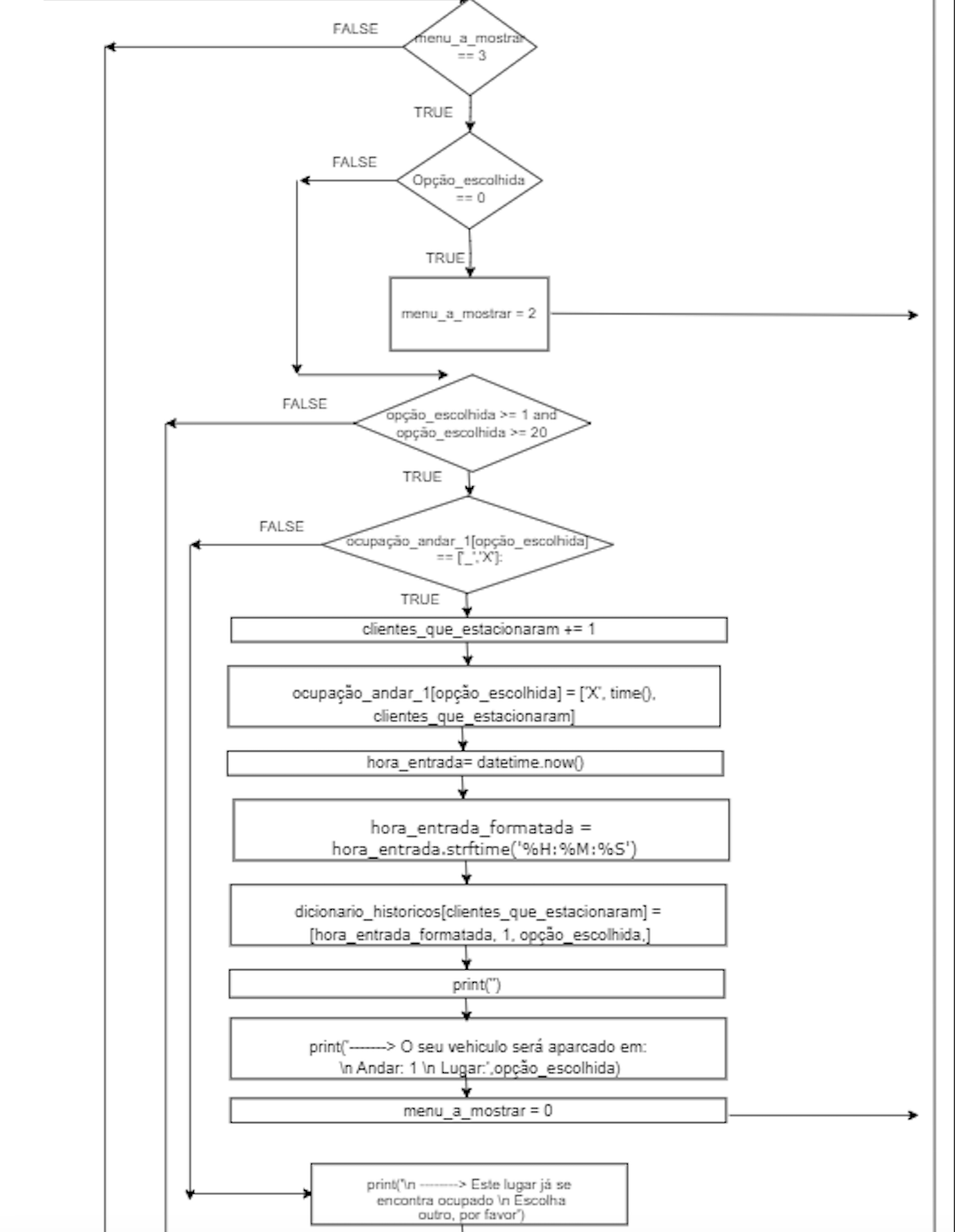
* 1. **PARTE III - FLUXOGRAMA**



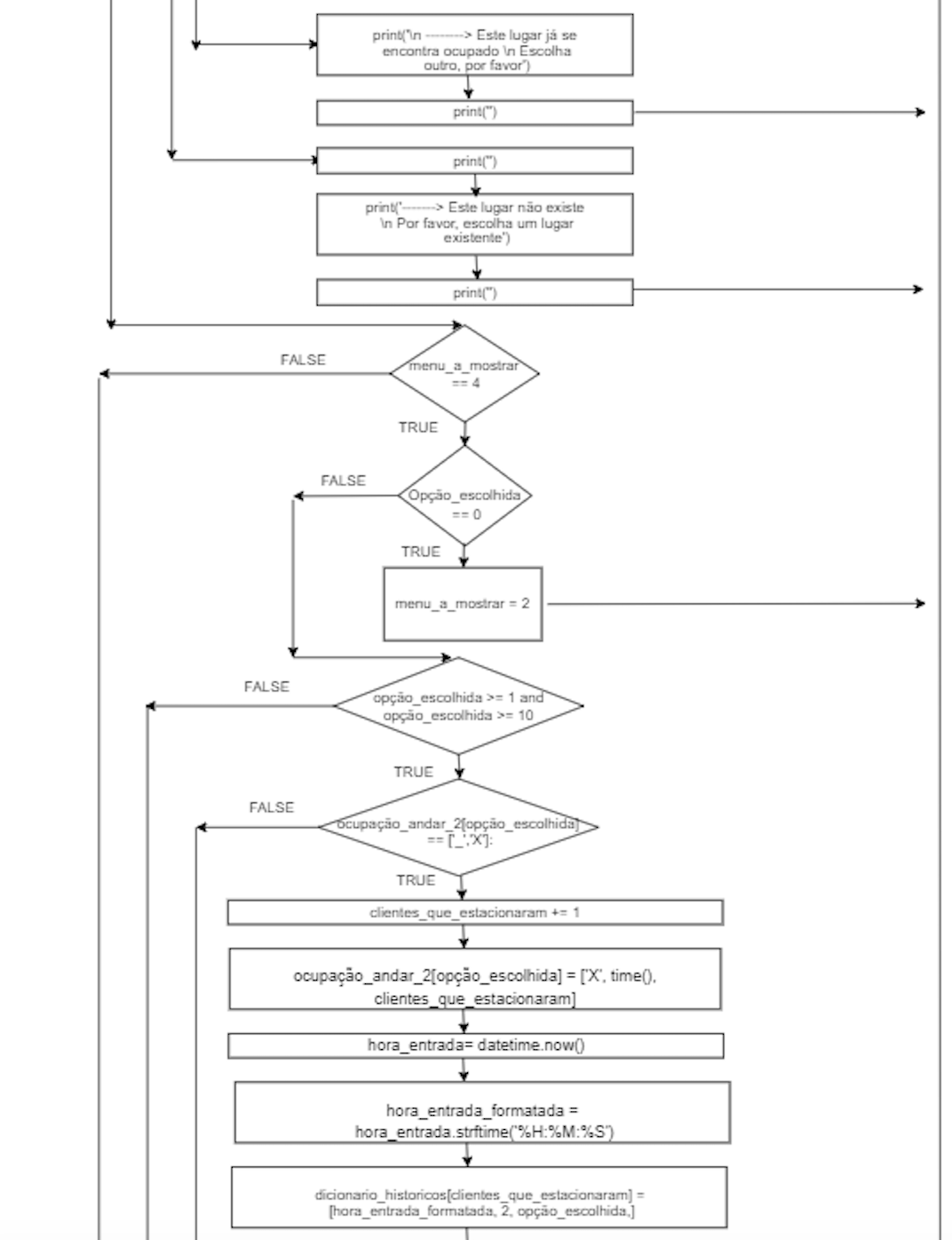
* 1. **PARTE IV - FLUXOGRAMA**

****

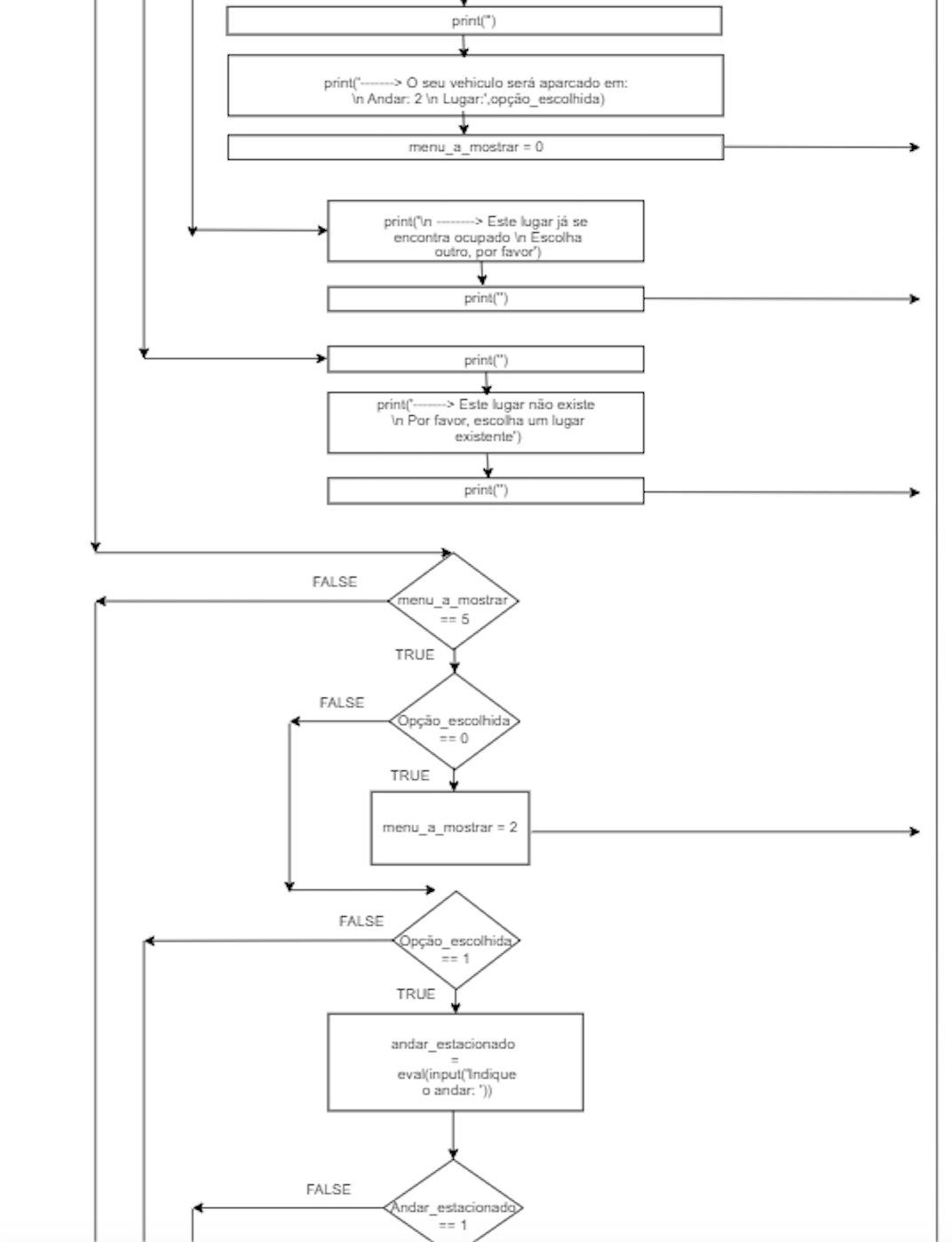
* 1. **PARTE V - FLUXOGRAMA**

****

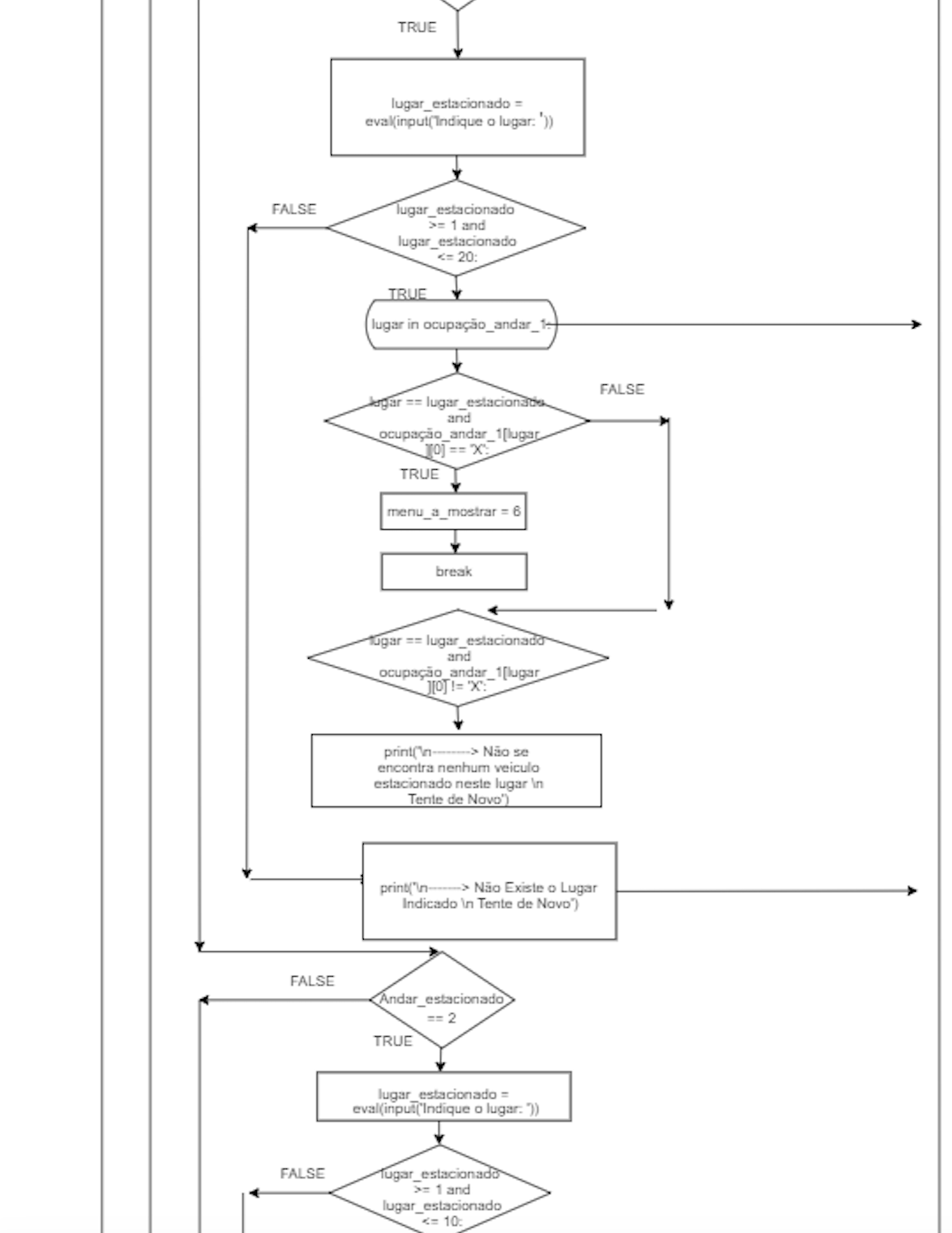
* 1. **PARTE VI - FLUXOGRAMA**



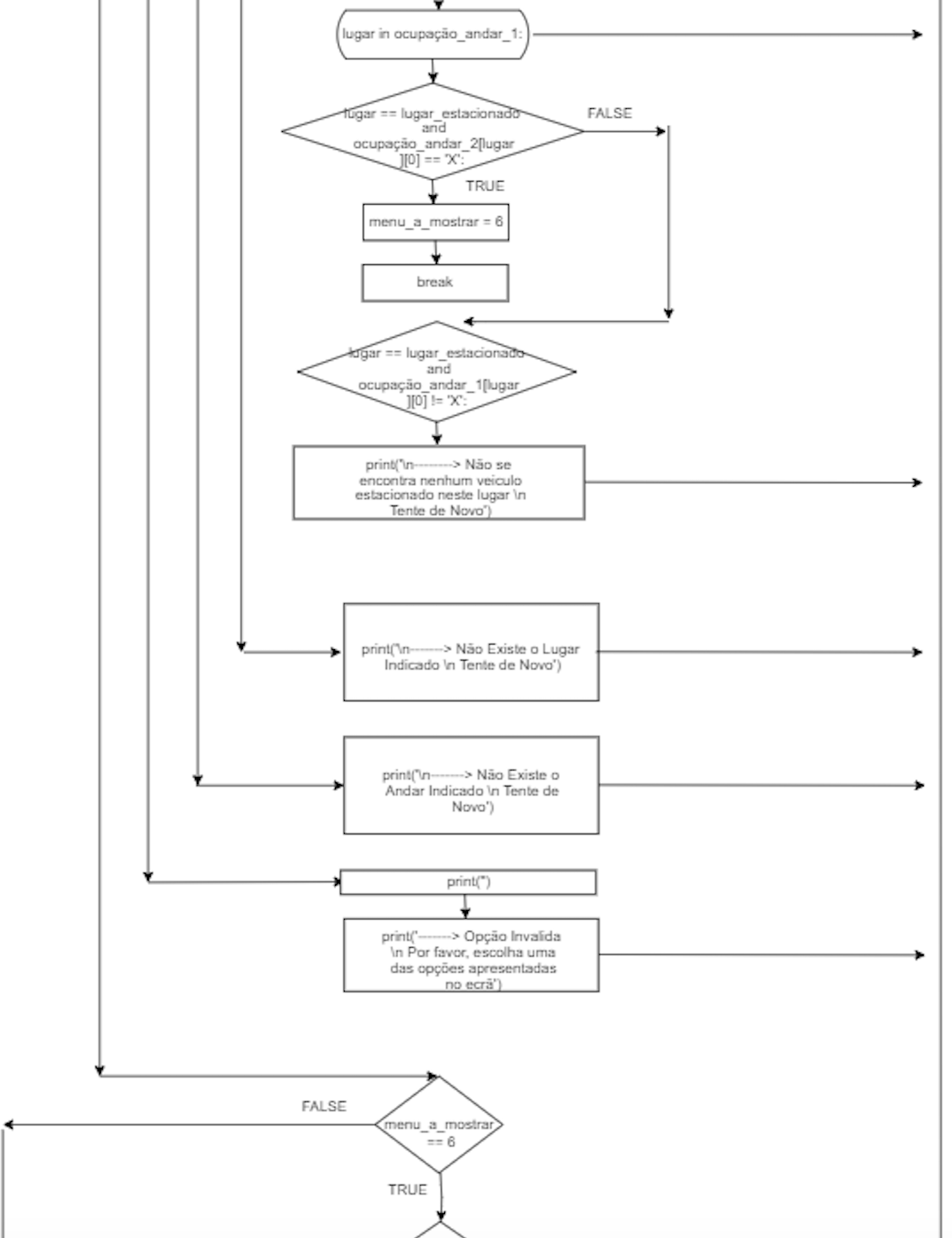
* 1. **PARTE VII - FLUXOGRAMA**



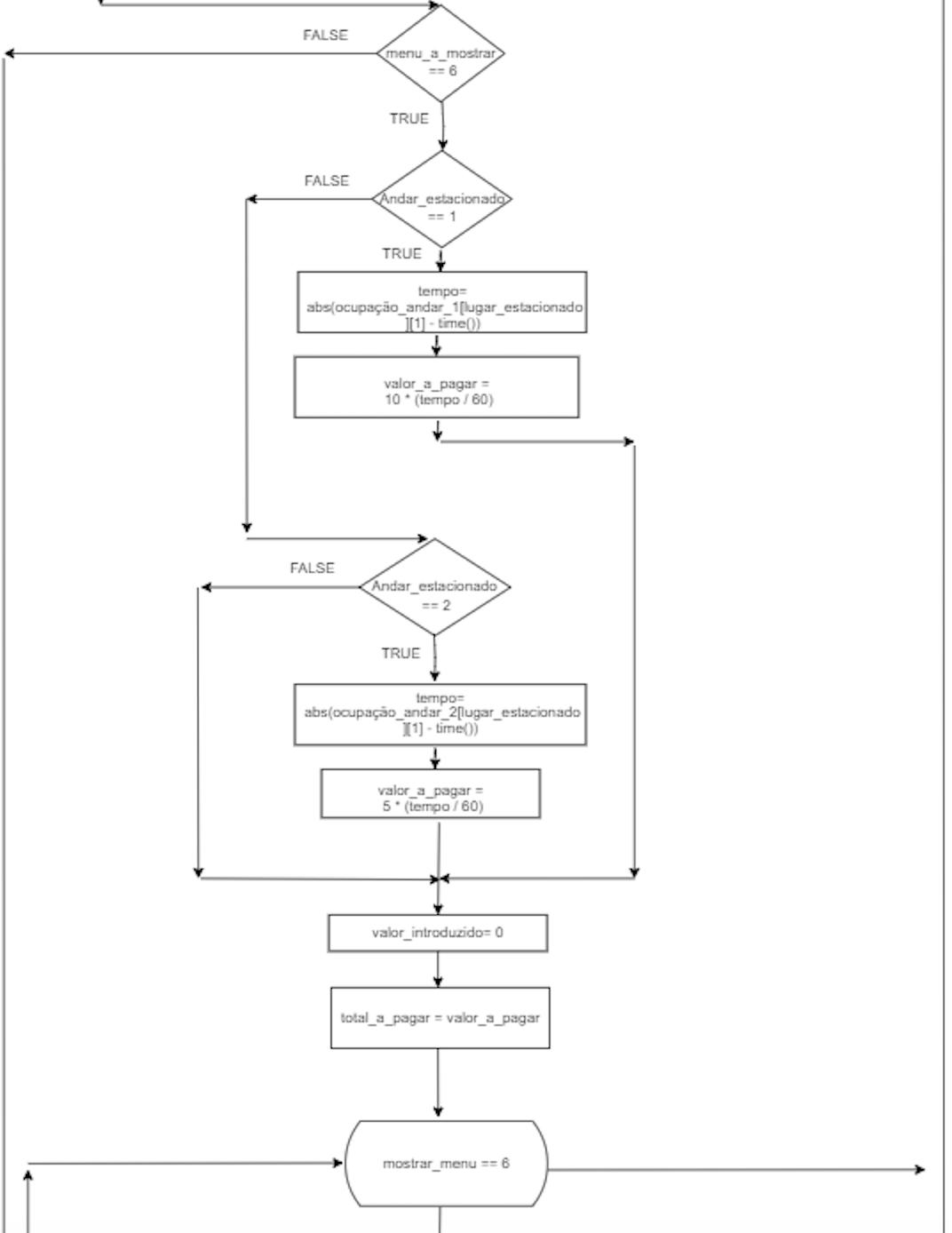
* 1. **PARTE VIII - FLUXOGRAMA**



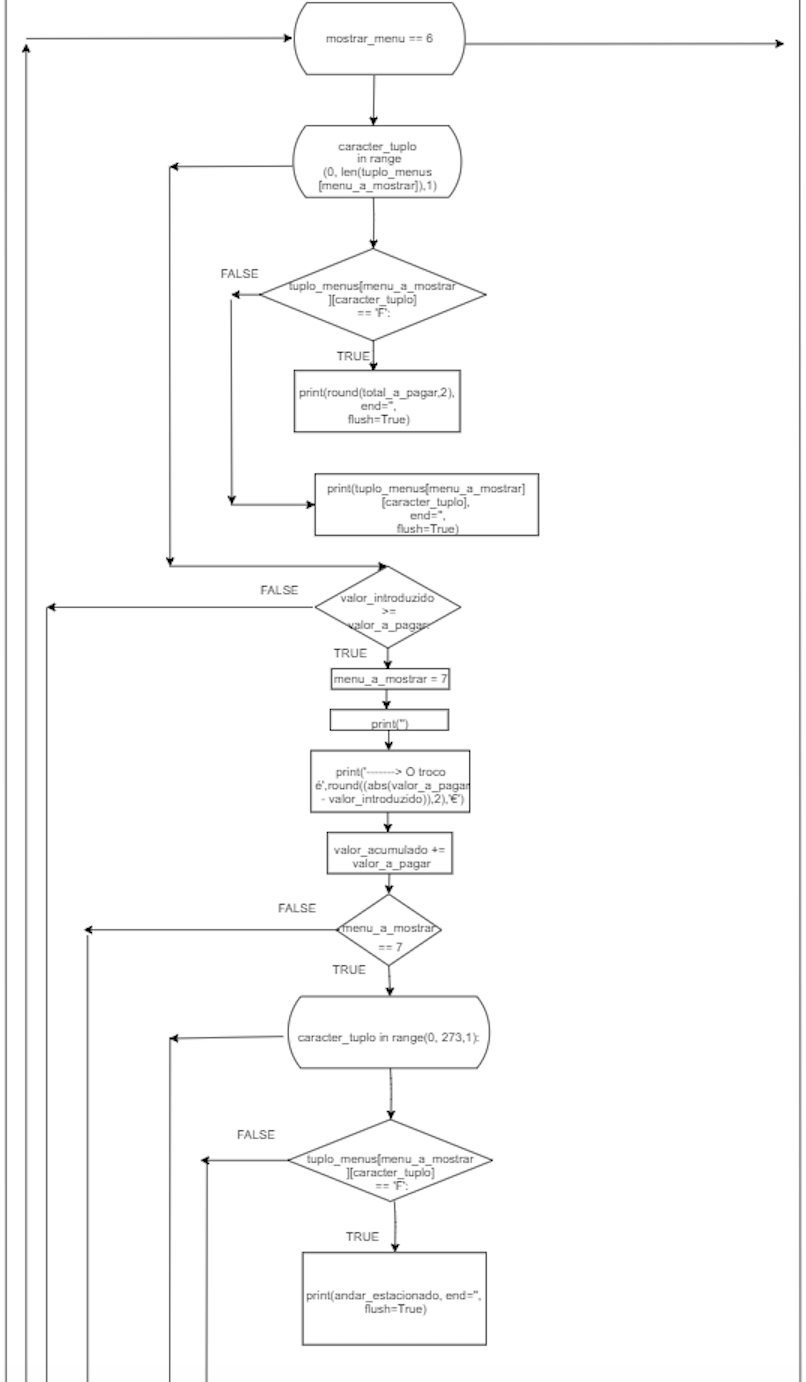
* 1. **PARTE IX - FLUXOGRAMA**



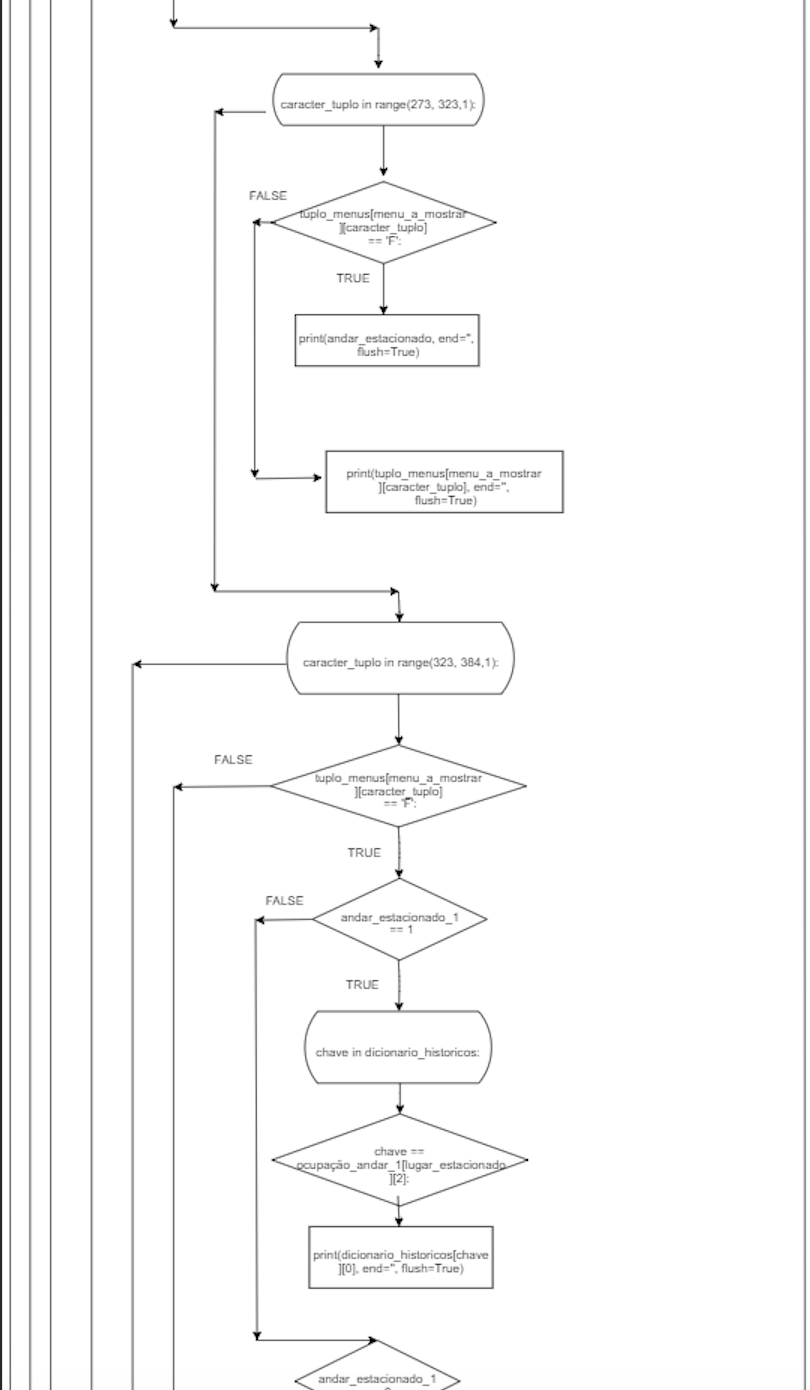
* 1. **PARTE X - FLUXOGRAMA**



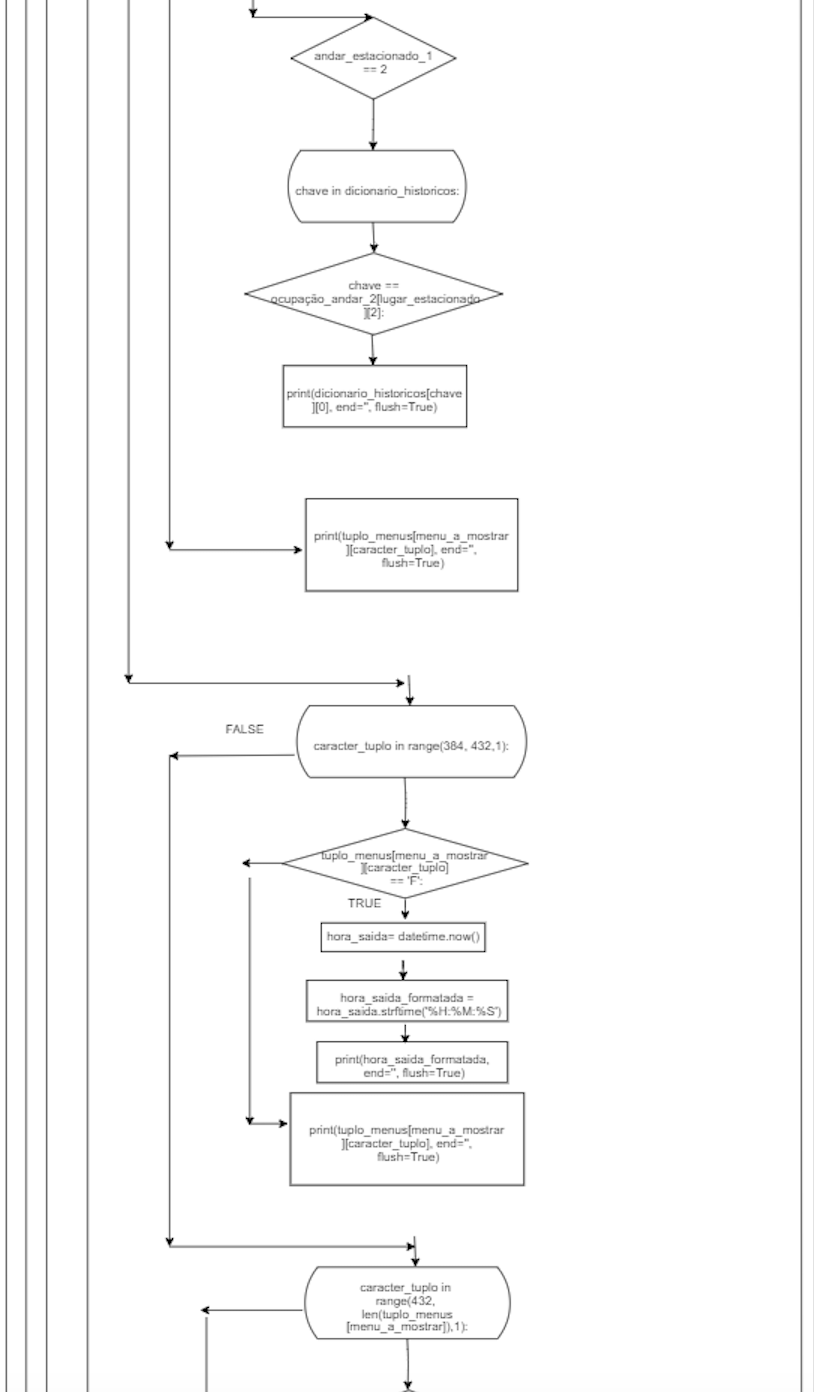
* 1. **PARTE XI - FLUXOGRAMA**



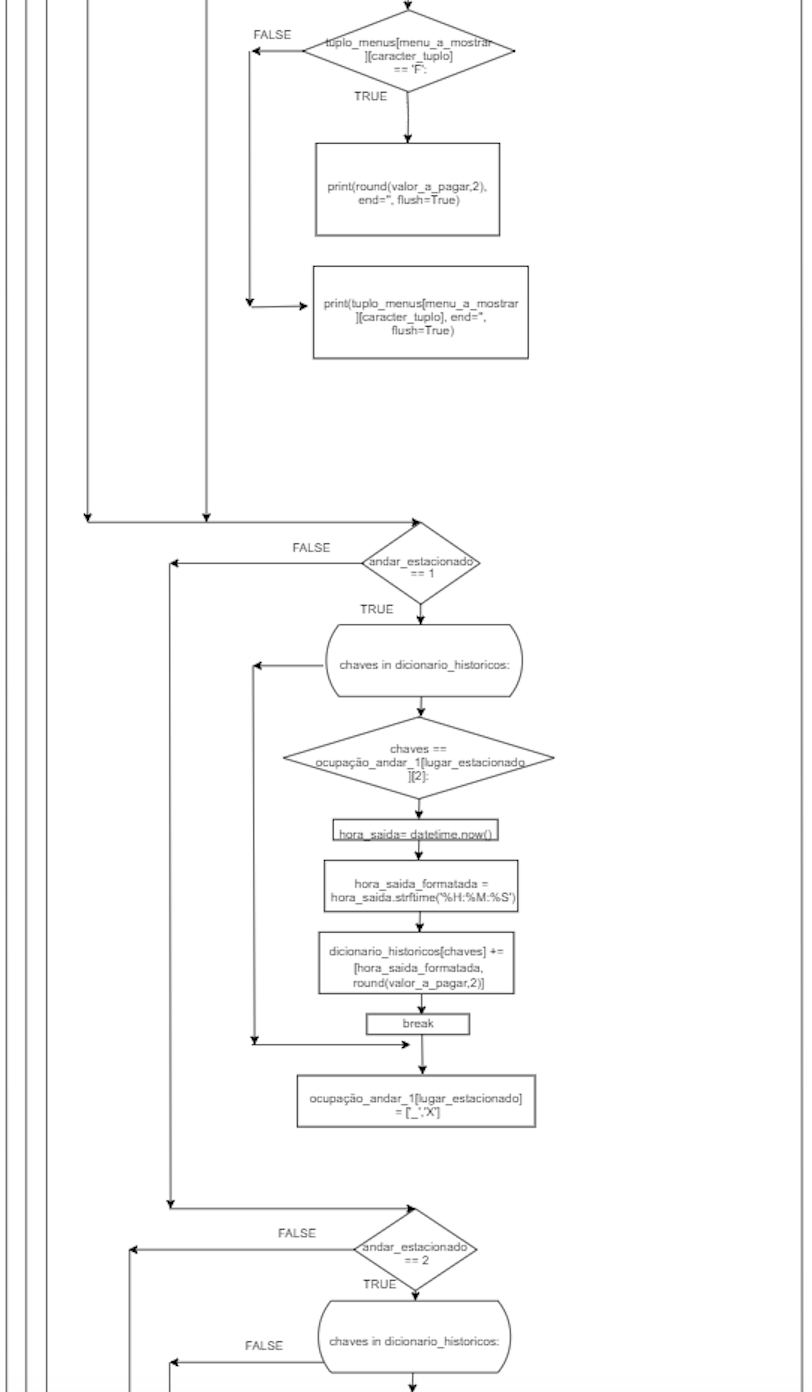
* 1. **PARTE XII - FLUXOGRAMA**



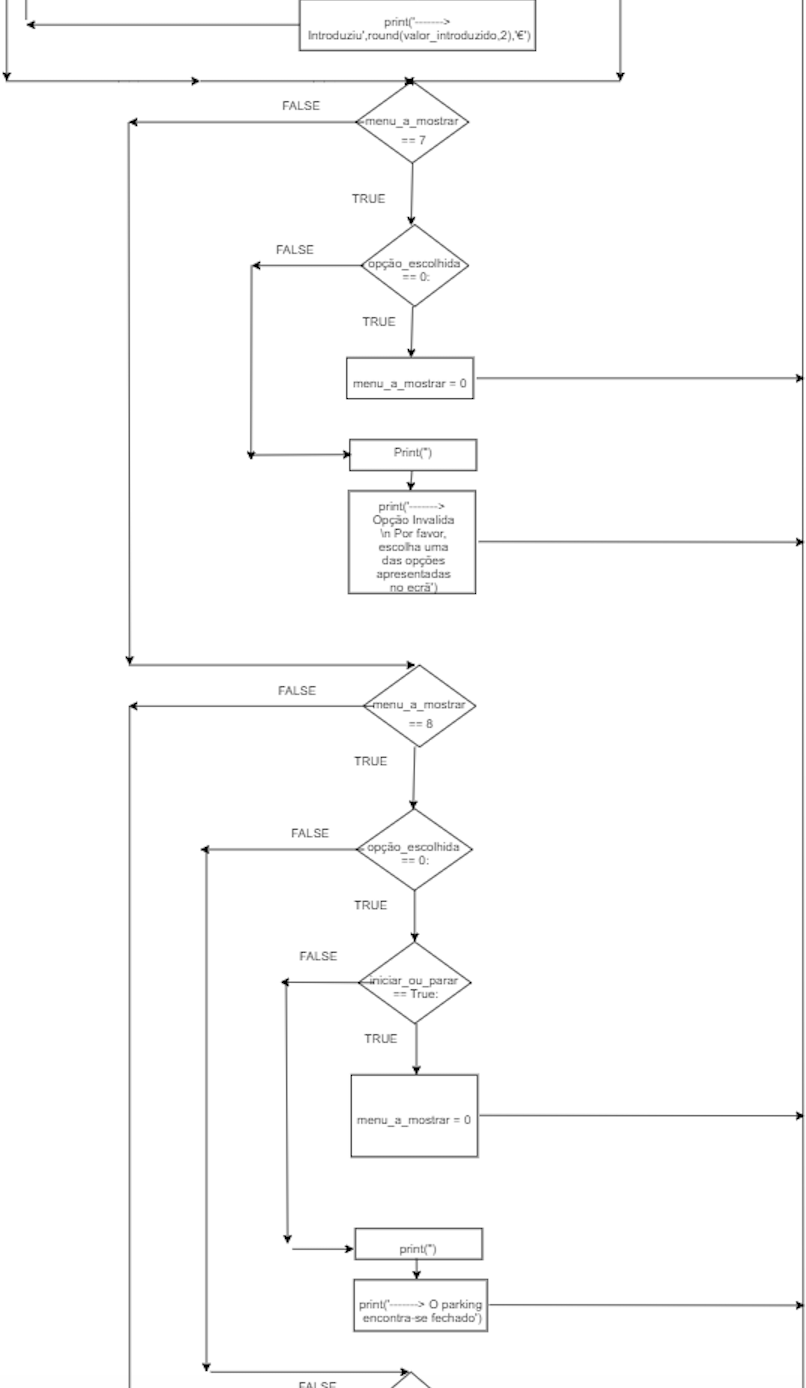
* 1. **PARTE XIII - FLUXOGRAMA**



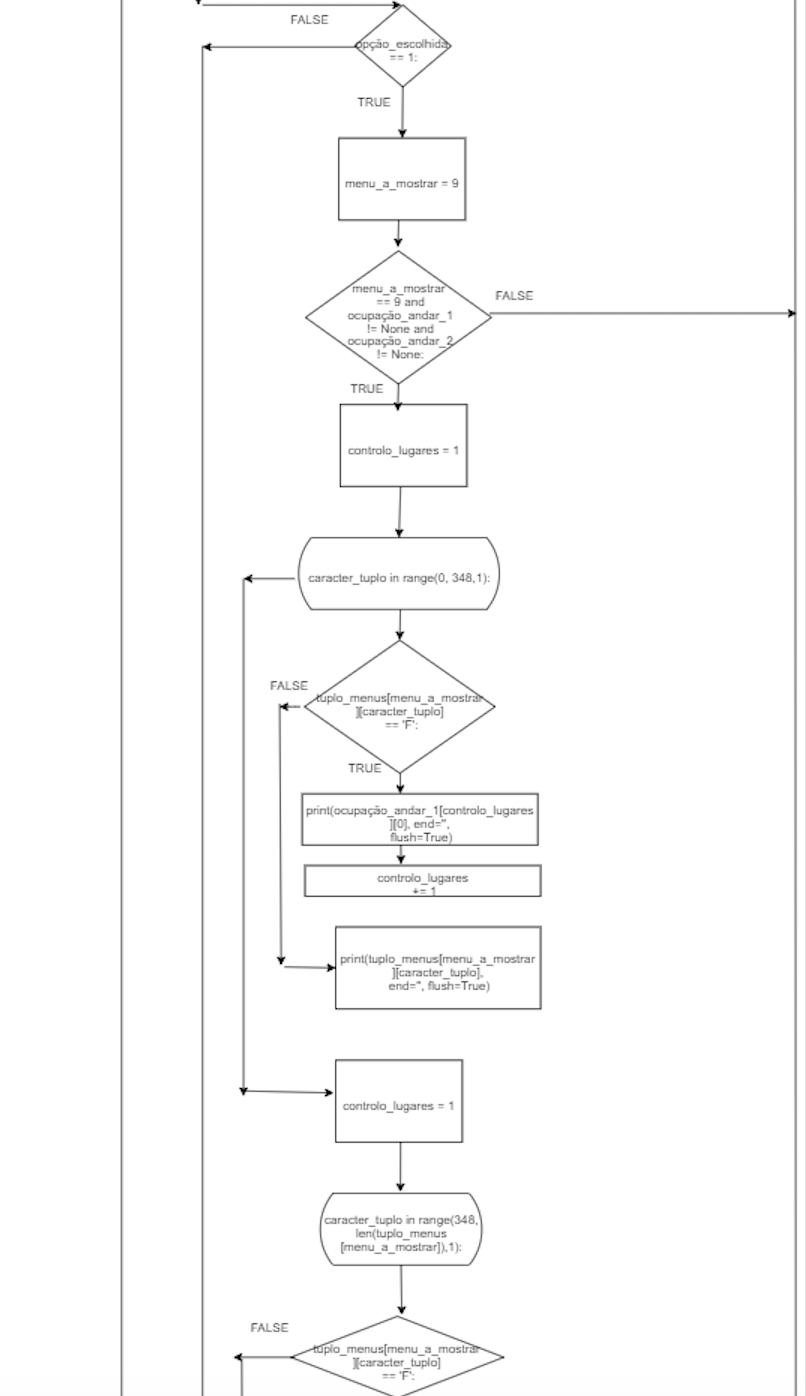
* 1. **PARTE XIV - FLUXOGRAMA**



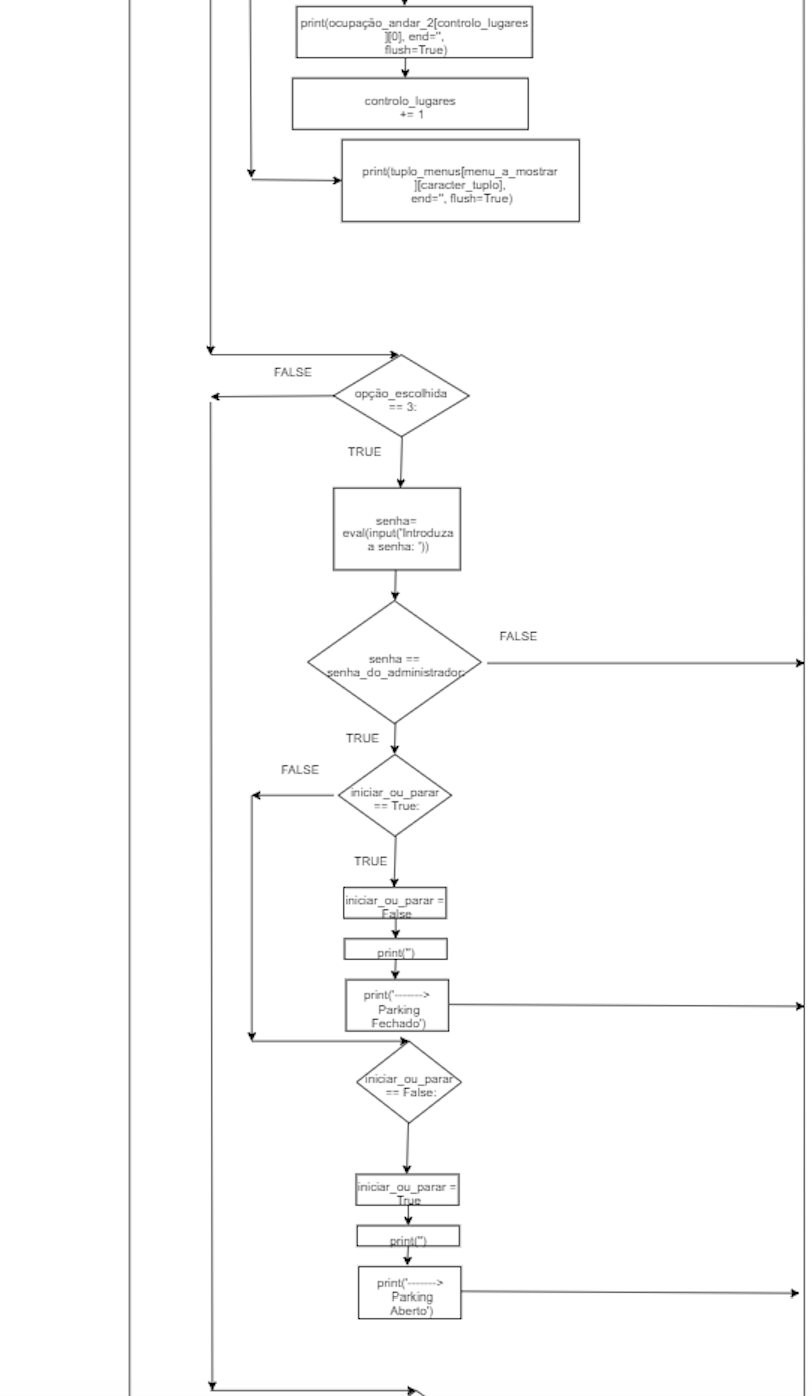
* 1. **PARTE XV - FLUXOGRAMA**



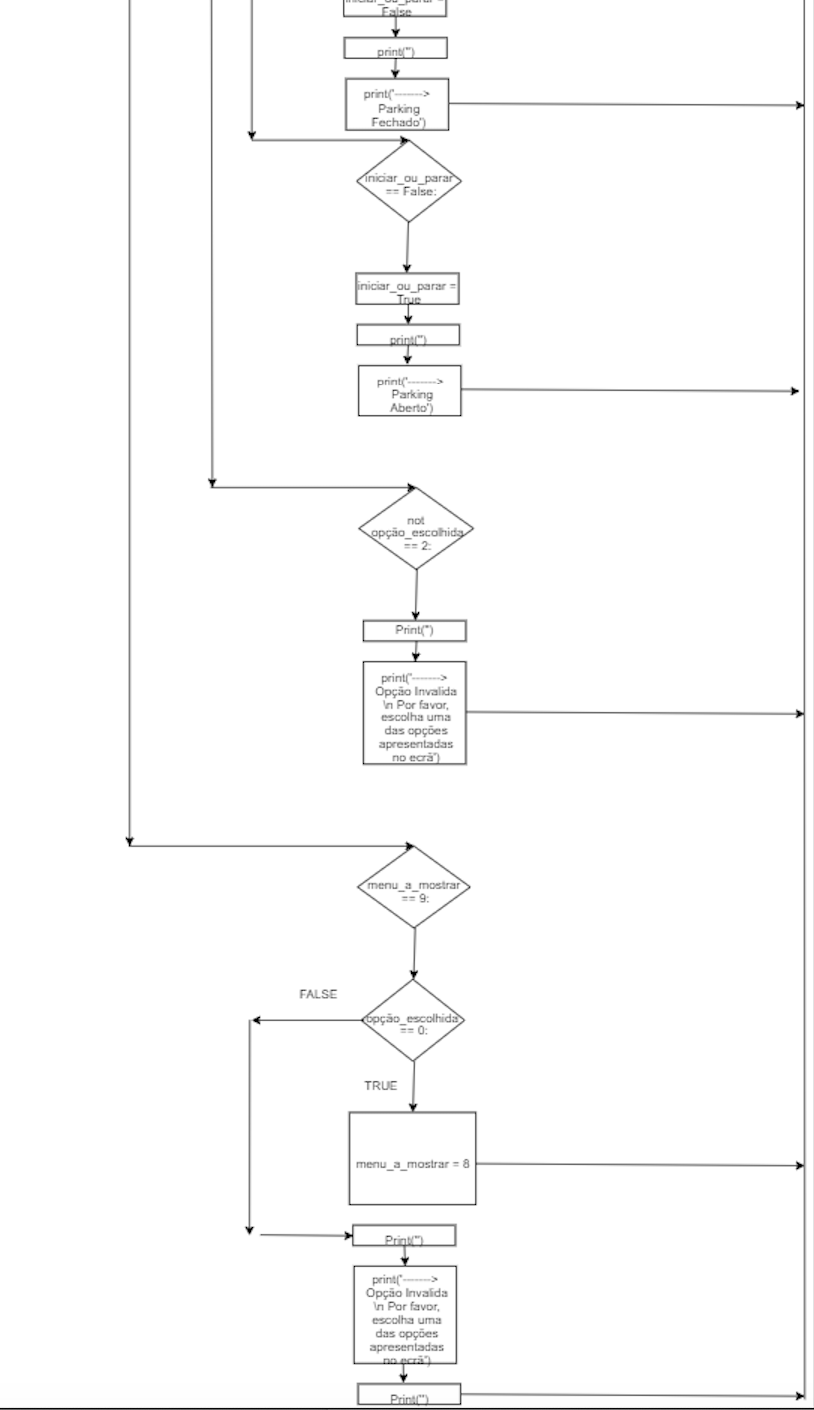
* 1. **PARTE XVI - FLUXOGRAMA**



* 1. **PARTE XVII - FLUXOGRAMA**



* 1. **PARTE XVIII - FLUXOGRAMA**



1. **CONCLUSÃO**

Em conclusão, o programa desenvolvido neste trabalho, além de demonstrar como as habilidades e conhecimentos adquiridos na Unidade Curricular Introdução à Programação foram aplicados para simular o funcionamento de uma máquina de controle de acesso e gestão de ocupação em um estacionamento, tambén contribuiu para reforçar o aprendido nas aulas e proporcionou uma oportunidade para por em prática as habilidades de programação.

Por tanto, a realização do programa foi um grande desafio devido à sua complexidade. No entanto, foi muito valiosa, pois permitiu aprender a encontrar soluções para os problemas apresentados e procurar metodos eficazes para cumplir todas as funcões que o programa devia ter.

Enfim, para isto foram utilizadas estruturas de dados e variáveis e foram implementados menus e opções para que o usuário pudesse interagir com a máquina. De modo geral, este projeto é um exemplo eficaz de como os conceitos aprendidos podem ser aplicados para resolver problemas práticos e reais. O projeto resultou numa experiência enriquecedora que ajudou a desenvolver habilidades importantes para obter um melhor desenvolvimento na programação.