Uma imagem com sentado, preto, computador

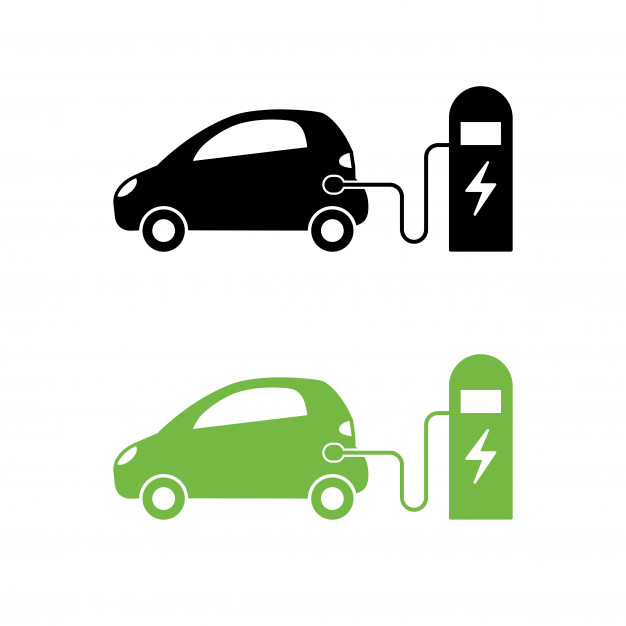
Descrição gerada automaticamente

**Faculdade de Ciências Exatas e da Engenharia**

**2019/2020**

**Arquitetura de Computadores**

Licenciatura em Engenharia Informática

**2º Projeto – Posto de Carregamento Elétrico**

**Trabalho realizado por:**

Diego Briceño (nº 2043818)

Rúben Rodrigues (nº 2046018)

**Docentes:**

Dionísio Barros, Nuno Ferreira

Sofia Inácio, Pedro Camacho

Funchal, 30 de abril de 2020

**Índice**

[1. Introdução 3](#_Toc39308299)

[2. Objetivos 3](#_Toc39308300)

[3. Desenvolvimento 3](#_Toc39308301)

[3.1. Periféricos de Entrada 3](#_Toc39308302)

[3.1.1. Ligar 3](#_Toc39308303)

[3.1.2. Botão OK 3](#_Toc39308304)

[3.1.3. ID do cliente 3](#_Toc39308305)

[3.1.4. Código de Segurança do cliente 3](#_Toc39308306)

[3.1.5. Saldo do cliente 3](#_Toc39308307)

[3.1.6. Bateria do Veículo do Cliente 4](#_Toc39308308)

[3.1.7. Opções escolhidas 4](#_Toc39308309)

[3.1.8. Tempo de Carregamento escolhido 4](#_Toc39308310)

[3.1.9. Carregamento do Posto 4](#_Toc39308311)

[3.2. Utilização do Programa 4](#_Toc39308312)

[4. Discussão de Resultados 5](#_Toc39308313)

[4.1. Criação de um Cliente 5](#_Toc39308314)

[4.2. Alteração de dados 6](#_Toc39308315)

[4.3. Carregamento do Posto 6](#_Toc39308316)

[4.4. Carregamento da bateria do veículo 6](#_Toc39308317)

[5. Conclusão 7](#_Toc39308318)

[6. Bibliografia 7](#_Toc39308319)

[7. Anexo A 8](#_Toc39308320)

[7.1. Fluxograma do Programa Geral 8](#_Toc39308321)

[7.2. Fluxograma da Alteração da Base de Dados 9](#_Toc39308322)

[7.3. Fluxograma da Verificação do Utilizador 10](#_Toc39308323)

[7.4. Fluxograma do Calcula Tempo 11](#_Toc39308324)

[7.5. Fluxograma do Descarregamento do Posto 12](#_Toc39308325)

[7.6. Fluxograma da Verificação do Saldo 13](#_Toc39308326)

[7.7. Fluxograma do Fornecimento de Energia 14](#_Toc39308327)

[7.8. Fluxograma do Débito 15](#_Toc39308328)

[7.9. Fluxograma da Atualização do Posto 16](#_Toc39308329)

[8. Anexo B 17](#_Toc39308330)

1. **Introdução**

Este relatório apresentará os objetivos relacionados ao segundo trabalho prático da unidade curricular de Arquitetura de Computadores, assim como o seu desenvolvimento, discussão de resultados e a conclusão a que os alunos chegaram.

A linguagem de programação utilizada para o desenvolvimento do trabalho foi o Assembly. Esta linguagem é específica de cada processador e permite codificar uma longa lista de valores de sinais num só número, o opcode. O assembler converte o programa assembly em números binários, permitindo que o processador consiga executar as instruções diretamente.

O programa desenvolvido foi criado em linguagem Assembly do processador PEPE e, para testá-lo utilizou-se o simulador JAVA.

1. **Objetivos**

O objetivo deste trabalho é criar um programa que simula um posto de carregamento para veículos elétricos, em linguagem Assembly, como também especificar e elaborar fluxogramas.

O posto de carregamento só está disponível aos detentores de um cartão fornecido pelos serviços da Unidade.

O processo de abastecimento é realizado pelos condutores, sendo selecionado o tipo de carregamento e o tempo de carregamento desejado e, de seguida, é efetuado o pagamento do carregamento. O posto de carregamento tem a possibilidade de inserção de novos utilizadores, como também da alteração de dados de detentores do cartão universitário.

A interface de comando do posto de carregamento foi feita através de um display de dimensões 7x16 (7 linhas de 16 carateres – bytes), que funciona como periférico de saída.

1. **Desenvolvimento**
   1. **Periféricos de Entrada**
      1. **Ligar**

Este botão é responsável por ligar e desligar a máquina do posto que permite o carregamento dos veículos. Os utilizadores só podem utilizar o posto quando o periférico ***Ligar*** estiver diferente de 0. Caso esteja a “0”, a máquina do posto é desligada. Quando a máquina está desligada o display fica em branco até o utilizador a ligar.

* + 1. **Botão OK**

Este botão é responsável pela confirmação das opções do utilizador. As opções inseridas pelo utilizador só serão válidas após o utilizador ter colocado o valor “1” no periférico ***OK***. O utilizador só poderá avançar até as suas opções serem validadas.

Caso seja inserido uma opção incorreta e o utilizador confirmar a opção, irá aparecer no display a informação de que a opção é inválida e o utilizador terá de inserir uma nova opção.

* + 1. **ID do cliente**

O periférico ***InputID*** é onde o utilizador insere o ID do cliente que pretende realizar um carregamento, alterar os seus dados ou do novo cliente que se pretende adicionar à base de dados.

* + 1. **Código de Segurança do cliente**

O periférico ***InputCodSeguranca*** é onde o utilizador insere o código de segurança do cliente que pretende realizar o carregamento, alterar os seus dados ou do novo cliente que se pretende adicionar à base de dados.

* + 1. **Saldo do cliente**

O periférico ***InputSaldo*** é onde,caso o cliente já esteja na base de dados, é inserido o novo valor do saldo do cliente escolhido, ou, caso não esteja, o valor do saldo do cliente que será criado.

Se o valor inserido pelo utilizador for inferior a 0, é mostrado no display a informação de que o saldo inserido é inválido pois não pode ser negativo.

* + 1. **Bateria do Veículo do Cliente**

O periférico ***InputBateria*** é onde,caso o cliente já esteja na base de dados, é inserido o novo valor da bateria do veículo do cliente escolhido, ou, caso não esteja, o valor da bateria do veículo do cliente que será criado.

Caso o valor inserido pelo utilizador for inferior a 0 ou superior a 100, é mostrado no display que o valor inserido é inválido e que deve ser inserido um valor entre 0 e 100.

* + 1. **Opções escolhidas**

O periférico ***InputOpcao*** é onde o utilizar irá inserir a opção escolhida para os diferentes menus.

Se o utilizador escolher uma opção diferente das opções que aparecem no display (e for confirmada), é informado de que a opção escolhida é inválida.

* + 1. **Tempo de Carregamento escolhido**

O periférico ***InputTempo*** é onde o utilizador insere a duração do carregamento que pretende realizar.

Caso o tempo seja inferior a 1, é mostrado no display que o tempo inserido tem de ser superior a 0.

E, caso o tempo escolhido seja superior ao tempo necessário para a bateria do veículo ficar completamente carregada, e não descarregue o posto, o utilizador é informado no display e, é realizado o fornecimento de energia apenas durante o tempo necessário para a bateria do veículo ficar 100% carregada, sendo feito apenas o débito do tempo que realmente foi necessário para efetuar o carregamento.

Por exemplo, se o utilizador escolher o tipo de carregamento rápido e uma duração de 3 horas, será apenas feito o débito de 3€ e não 9€, pois é apenas necessária uma hora para que a bateria do veículo fique completamente carregada.

* + 1. **Carregamento do Posto**

O periférico ***InputIncrementoBateria*** é onde o utilizador insere a quantidade de energia que pretende carregar na bateria do posto escolhida anteriormente. O carregamento da bateria do posto só é realizado se o valor inserido pelo utilizador for superior a 0. Se inserir um valor que cause overflow, o utilizador é informado e a bateria fica com o valor que tinha antes da adição.

* 1. **Utilização do Programa**

Inicialmente a máquina encontra-se desligada. Após ser ligada, são lidos os endereços referentes às baterias do posto para guardar os valores em registos, evitando inúmeros acessos à memória (R0 – Normal, R1 – Semi-rápido e R2 – Rápido), e, de seguida, é iniciado o programa.

Para facilitar os cálculos, a bateria dos carros dos clientes está em percentagem, mas a bateria dos postos não, cada percentagem que se carrega no carro é uma unidade que retiramos ao valor da bateria do posto correspondente.

* + 1. **Criar/Alterar Dados**

No início do programa é perguntado ao utilizador se pretende alterar ou criar dados na base de dados.

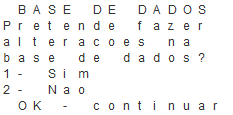
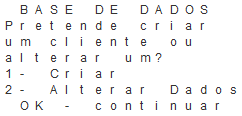
 

Figura 1 Início do programa Figura 2 Alterar ou Criar dados

Para alterar dados, é pedido ao utilizador para inserir o ID e código de segurança do cliente cujo utilizador pretende alterar os dados. Sendo depois pedido para ser inserido o novo ID, código de segurança, saldo e valor da bateria do veículo do cliente, caso este exista na base de dados.

Para criar dados, é pedido apenas para inserir os dados referentes ao novo cliente, sendo verificado se o ID inserido é igual ao ID de um cliente já existente na base de dados, se tal acontecer, o cliente não é criado e o utilizador é informado. Se o utilizador não pretender fazer nenhuma alteração, avança-se para a próxima secção.

* + 1. **Inserção de Energia**

Para inserir energia no posto, o utilizador deve escolher uma das 3 baterias existentes no posto, cada uma referente a um tipo de carregamento diferente. Após a escolha, é feita a inserção do valor de energia que se pretende carregar na bateria desejada, com as limitações indicadas anteriormente no periféricoreferente ao carregamento do posto.

Se o utilizador não pretender inserir energia no posto, é mostrado no display a informação das baterias do posto que estão funcionais. O utilizador só poderá realizar um carregamento de um tipo de carregamento que se encontre funcional. Se os 3 postos não se encontrarem funcionais, o utilizador é informado que precisa carregar um posto e volta-se ao início do programa.

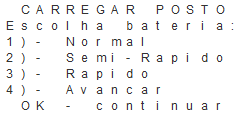
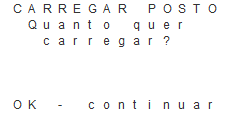
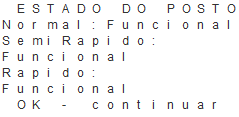
  

Figura 3 Escolha da bateria do posto Figura 4 Escolha energia a carregar Figura 5 Estado das baterias do posto

* + 1. **Carregamento do veículo**

Para realizar um carregamento, é pedido ao utilizador para inserir o ID e código de segurança do cliente que pretende realizar o carregamento. Após a verificação do cliente, caso seja autorizado, é pedido para escolher o tipo de carregamento desejado, e de seguida, a duração do mesmo. Depois, é mostrado no display o tipo de carregamento escolhido e a duração do fornecimento de energia que será feito (tendo em conta os diferentes casos possíveis mencionados anteriormente). Se o utilizador continuar, irá visualizar no display o saldo que terá no final do carregamento, tal como o custo da operação.

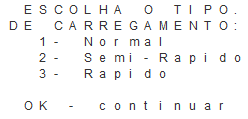
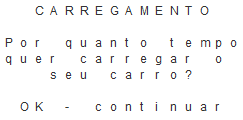
 

Figura 6 Escolha do tipo de carregamento Figura 7 Escolha do tempo de carregamento

Após ser feito o débito, é realizada a atualização da bateria do posto referente ao tipo de carregamento escolhido pelo utilizador, sendo subtraído o valor da energia fornecida ao veículo à respetiva bateria, e aparece no display uma informação de que o carregamento foi concluído, gravam-se os valores das baterias na memória e volta-se ao início do programa.

1. **Discussão de Resultados**
   1. **Criação de um Cliente**

Para exemplificar a criação de um cliente, criou-se um cliente com o ID igual a 258 (decimal, 0102 em hexadecimal), código de segurança 772 (decimal, 0304 em hexadecimal), saldo 15€ e com a bateria do veículo a 0%.

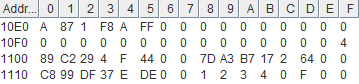
 

Figura 8 Base de Dados antes da criação Figura 9 Base de Dados depois da criação

Como se pode ver na *Figura 8*, a base de dados tem três clientes e após a criação do novo cliente, que é inserido no fim da base de dados, o tamanho da base de dados aumentou por 1 (*Figura 9*).

* 1. **Alteração de dados**

Para exemplificar esta parte do programa, decidiu-se alterar os dados do cliente criado anteriormente. Alterou-se o seu ID e código de segurança (0A0B e 0C0D, respetivamente), reduzimos o seu saldo para 9€ e “carregou-se” o seu veículo a 10%. Pode-se observar isto na *Figura 10*.

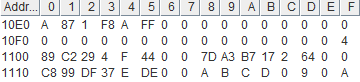


Figura 10 Base de Dados após a alteração

* 1. **Carregamento do Posto**

Para demonstrar o carregamento do posto, assumimos a situação inicial da *Figura 11*. Foi feito um carregamento da bateria do posto rápido no valor de FF, como se pode ver na *Figura 13*. Após avançarmos para a demonstração dos níveis de energia, pode-se verificar a alteração do valor na *Figura 12.*

Figura 11 Valores das baterias antes do carregamento Figura 12 Valores das baterias após o carregamento

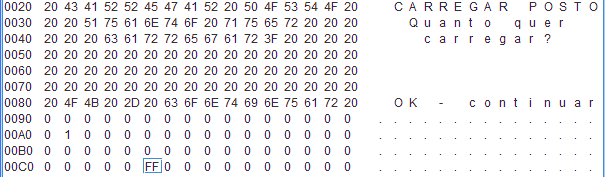


Figura 13 Escolha da energia a carregar

* 1. **Carregamento da bateria do veículo**

Para demonstrar o processo de verificação de um cliente e carregamento do seu veículo, tendo como base de dados a da *Figura 8*, na verificação introduziu-se o ID e código de segurança do primeiro cliente (*Figura 14*) e foi autorizado para carregar o seu veículo (*Figura 15*).

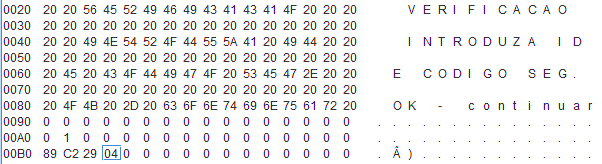
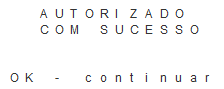
 

Figura 14 Introdução de dados do cliente Figura 15 Cliente autorizado

Para carregar o veículo, optou-se por um carregamento do tipo normal com duração de 10 horas (*Figura 16*). Como o carregamento normal carrega 20% da bateria do veículo por hora, só seriam necessárias 5 horas para carregar o veículo, por isso foi apresentado ao utilizador a mensagem da *Figura 17*. Na *Figura 18* especificou-se o tipo de carregamento e a duração deste. Por fim, para debitar o valor ao cliente, só foi debitado o custo de 5€ (no tipo normal cada hora é 1€) e foi apresentado ao cliente o custo do carregamento e o seu saldo atualizado após o débito (*Figura 19*).

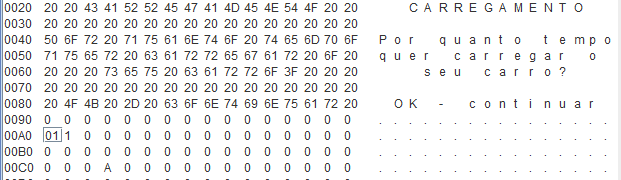
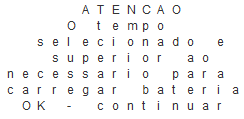
 

Figura 16 Escolha do tempo de carregamento Figura 17 Informação sobre o tempo escolhido

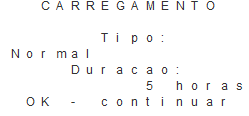
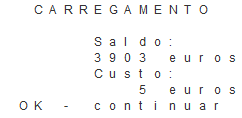
 

Figura 18 Informação sobre o carregamento Figura 19 Informação sobre o saldo e custo

Verificando o estado da base de dados antes e após o carregamento pode-se observar que o veículo do cliente está carregado a 100% e que o saldo do mesmo diminuiu por 5€. Como após um carregamento os valores das baterias dos postos em memória são atualizados também se pode observar que a bateria do posto normal diminuiu por 100 unidades, equivalente aos 100% fornecidos.

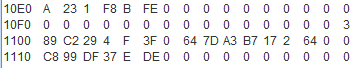
 

Figura 20 Base de dados antes do carregamento Figura 21 Base de dados após o carregamento

Se o cliente autorizado não tivesse saldo suficiente para efetuar o carregamento que desejava, seria informado como na *Figura 22* e o seu veículo não seria carregado.

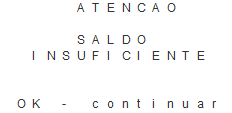


Figura 22 Informação sobre saldo insuficiente

1. **Conclusão**

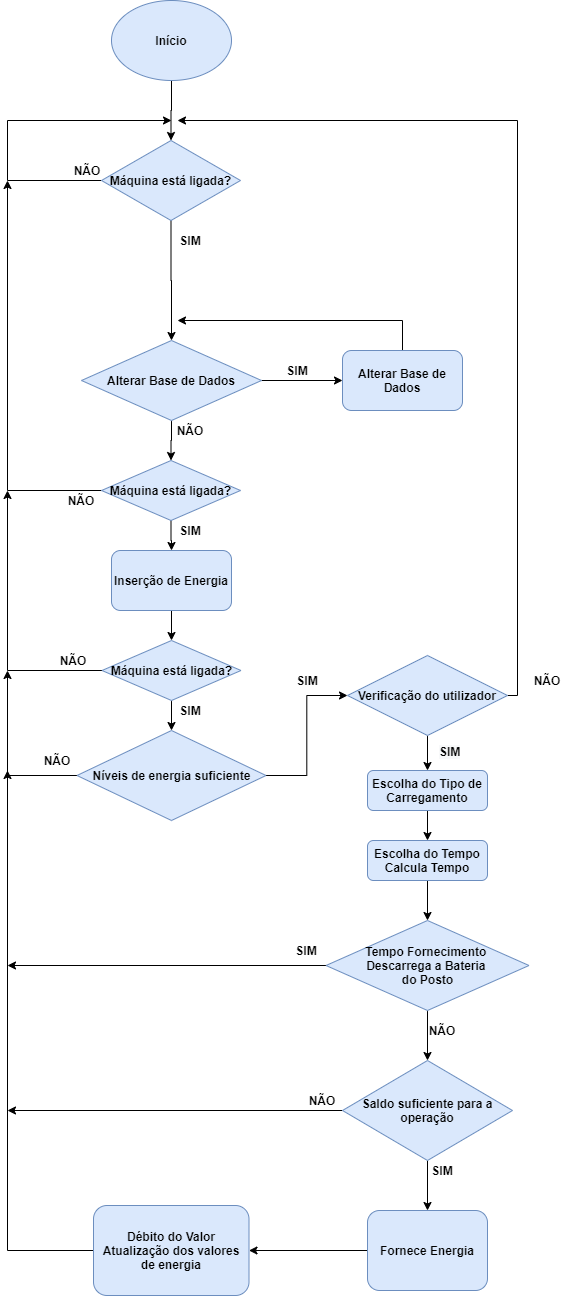
Concluindo, consideramos que os objetivos do trabalho foram atingidos e este tornou-se mais fácil quando se dividiu o problema em partes mais pequenas e mais acessíveis. A elaboração prévia dos fluxogramas, em relação ao desenvolvimento da solução, ajudou a facilitar o dito desenvolvimento.

A linguagem Assembly, embora seja mais complicada de perceber do que uma linguagem de alto nível, é mais fácil para codificar um dispositivo do que o código máquina.

Em geral, este trabalho foi uma boa ferramenta de aprendizagem para uma melhor compreensão da arquitetura de computadores e o funcionamento destes.

1. **Bibliografia**

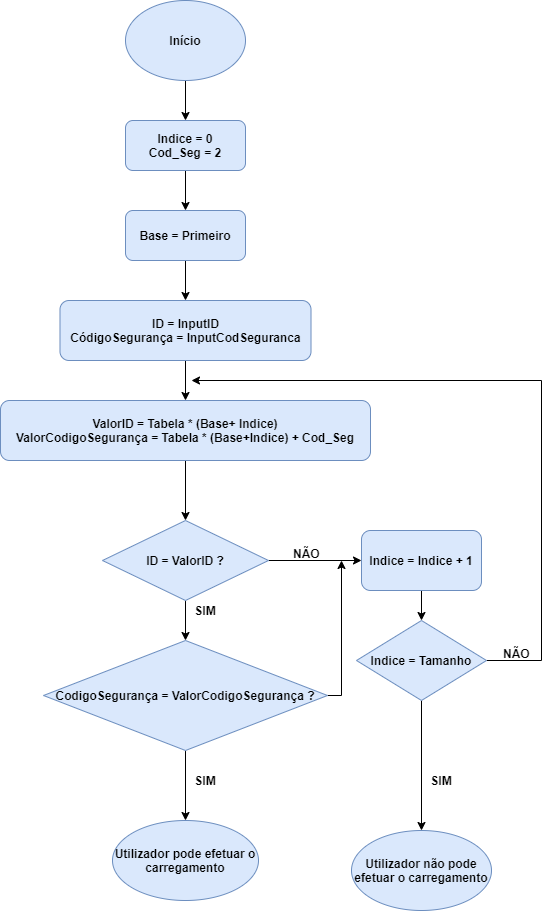
J. Delgado e C. Ribeiro, Arquitetura de Computadores, FCA - Editora de Informática, 2010.

1. **Anexo A**
   1. **Fluxograma do Programa Geral**
   2. **Fluxograma da Alteração da Base de Dados**

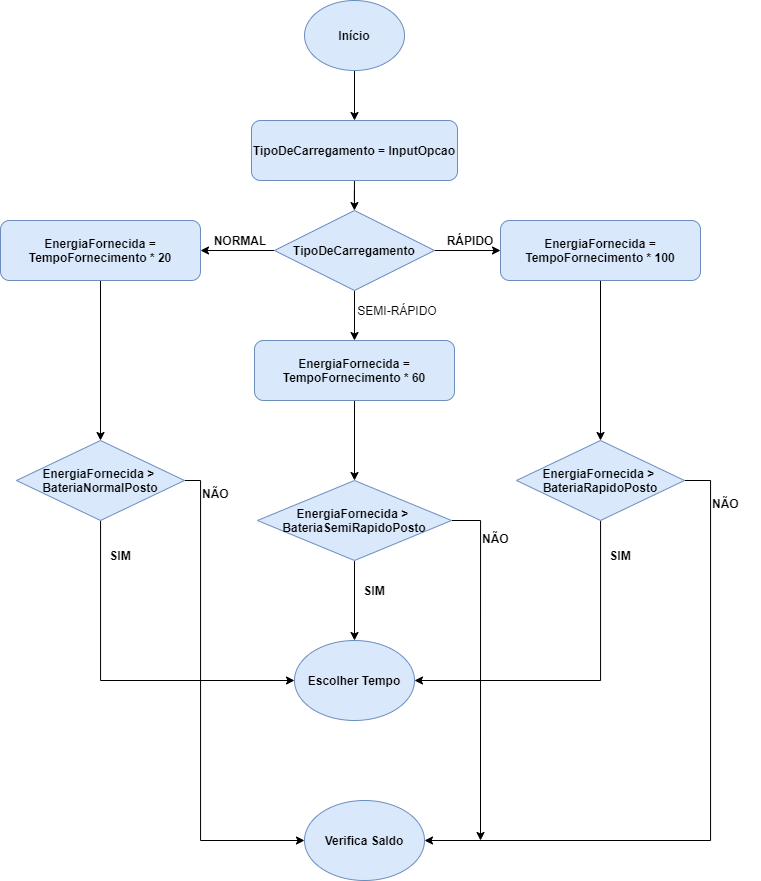
**Uma imagem com computador

Descrição gerada automaticamente**

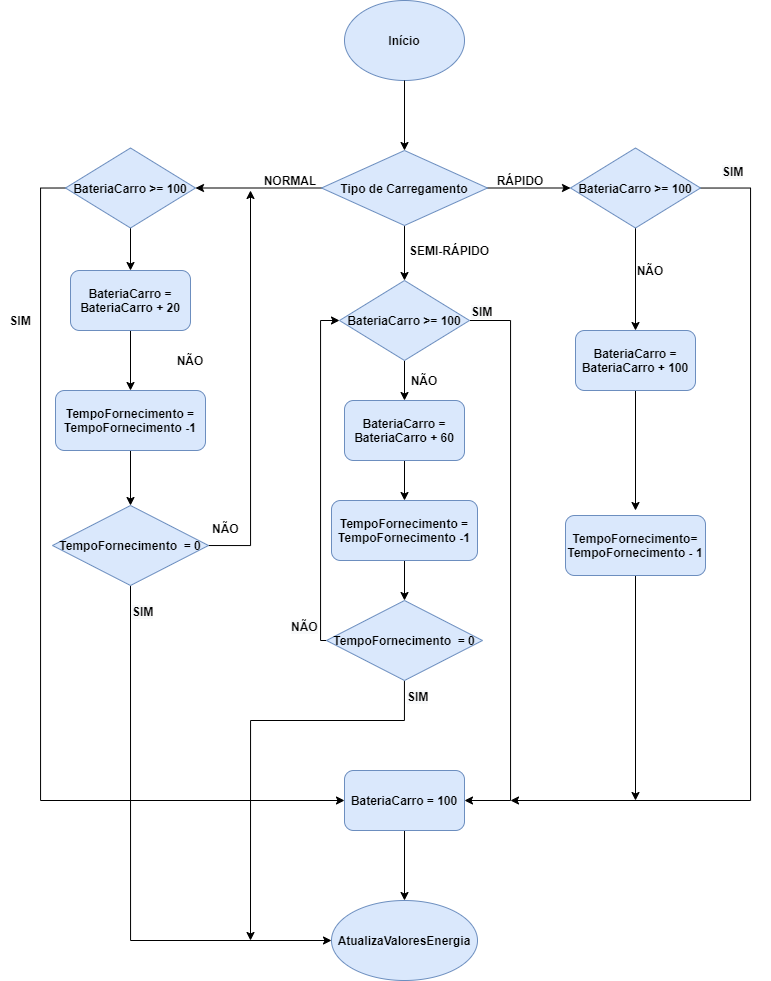
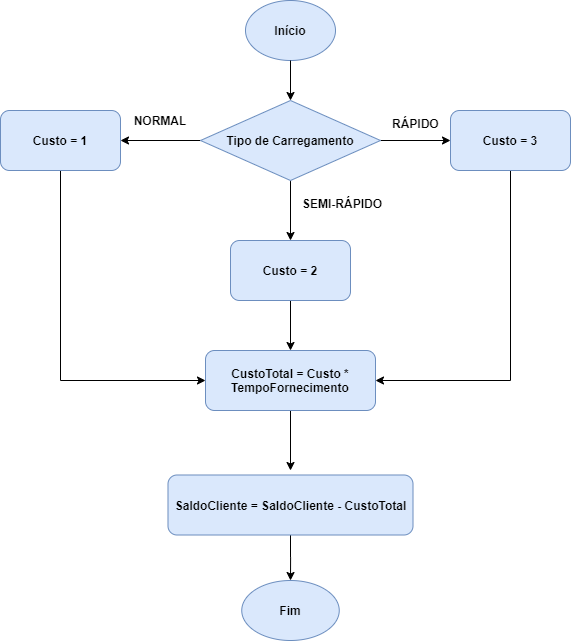
* 1. **Fluxograma da Verificação do Utilizador**

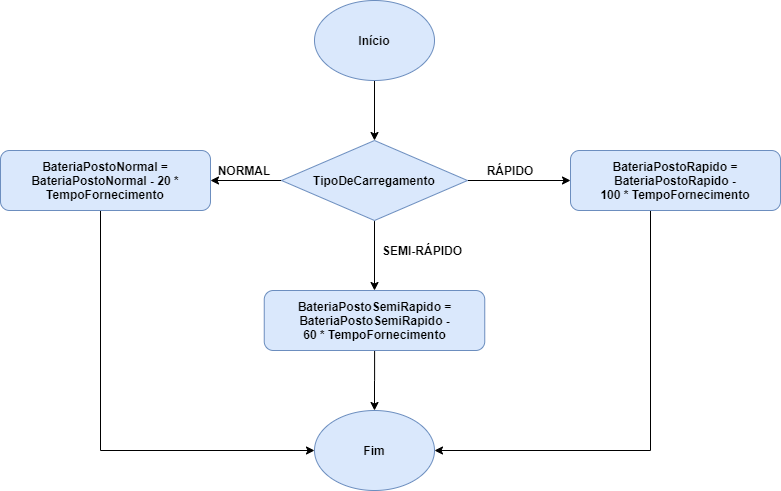
****

* 1. Uma imagem com computador

     Descrição gerada automaticamente**Fluxograma do Calcula Tempo**
  2. **Fluxograma do Descarregamento do Posto**
  3. **Fluxograma da Verificação do Saldo**

****

* 1. **Fluxograma do Fornecimento de Energia**
  2. **Fluxograma do Débito**
  3. **Fluxograma da Atualização do Posto**

****

1. **Anexo B**

