Uma imagem com sentado, preto, computador

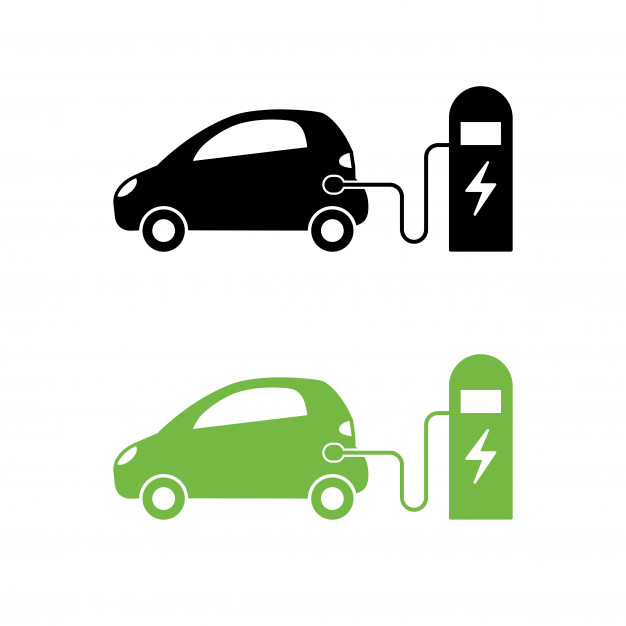
Descrição gerada automaticamente

**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, 25 de abril de 2020

**Índice**

[1. Introdução 3](#_Toc38879993)

[2. Objetivos 3](#_Toc38879994)

[3. Desenvolvimento 3](#_Toc38879995)

[3.1. Periféricos de Entrada 3](#_Toc38879996)

[3.1.1. Ligar 3](#_Toc38879997)

[3.1.2. Botão OK 4](#_Toc38879998)

[3.1.3. ID do cliente 4](#_Toc38879999)

[3.1.4. Código de Segurança do cliente 4](#_Toc38880000)

[3.1.5. Saldo do cliente 4](#_Toc38880001)

[3.1.6. Bateria do Veículo do Cliente 4](#_Toc38880002)

[3.1.7. Opções escolhidas 5](#_Toc38880003)

[3.1.8. Tempo de Carregamento escolhido 5](#_Toc38880004)

[3.1.9. Carregamento do Posto 5](#_Toc38880005)

[3.2. Utilização do Programa 5](#_Toc38880006)

[4. Discussão de Resultados 6](#_Toc38880007)

[5. Conclusão 6](#_Toc38880008)

[6. Bibliografia 6](#_Toc38880009)

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 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 a 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.

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

Inicialmente, o posto encontra-se desligado e, após ligar a máquina, é mostrado no display a opção do utilizador alterar a base de dados.

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.

Por exemplo, quando o utilizador escolhe o tipo de carregamento pretendido, só poderá escolher o tempo de carregamento após a confirmação da sua opção, ou seja, quando for inserido o valor “1” no periférico ***OK***. Se o utilizador escolher uma opção diferente de “1”, “2” ou “3”, e confirmar a sua escolha, aparecerá no display a informação mencionada anteriormente.

* + 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.

Na realização de um carregamento ou de uma alteração de dados, se o ID inserido não existir, aparece no display que o cliente não foi autorizado, voltando assim, ao menu inicial do programa.

Na criação de um novo cliente na base de dados, se o ID inserido já pertencer a algum cliente, o novo cliente não é criado e o utilizador é informado de que outro cliente já tem esse ID.

* + 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.

Na realização de um carregamento ou de uma alteração de dados, se o código de segurança do cliente com o ID inserido for incorreto, aparece no display que o cliente não foi autorizado, voltando assim, ao menu inicial do programa.

* + 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, como a escolha do tipo de carregamento, se pretende criar ou alterar a base de dados e, a escolha da bateria do posto que pretende carregar.

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, como já foi referido anteriormente.

* + 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.

Caso o tempo pretendido faça com que, durante o carregamento, a bateria do posto do tipo de carregamento escolhido fique sem energia, é mostrado no display que não há energia suficiente para o carregamento pretendido.

E, caso o tempo escolhido seja superior ao tempo necessário para a bateria do veículo ficar completamente carregada, 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 foi apenas necessário 1 hora para que a bateria do veículo ficar completamente carregada.

* + 1. **Carregamento do Posto**

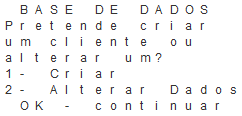
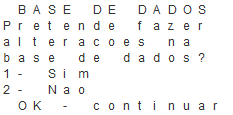
O periférico ***InputIncrementoBateria*** é onde o utilizador insere a quantidade de energia que pretende carregar na bateria do posto escolhida anteriormente. O carregamento da bateria do posto só é realizada se o valor inserido pelo utilizador for superior a 0.

* 1. **Utilização do Programa**

Inicialmente a máquina encontra-se desligada. Após ser ligada, são lidos os endereços referentes às baterias do posto e, de seguida, é iniciado o programa.

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

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



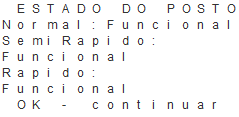
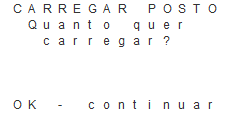
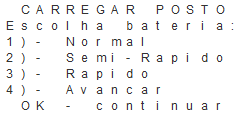
Para alterar dados, é pedido ao utilizador para inserir o ID e código de segurança do cliente cujo utilizador pretende alterar os dados. Sendo depois pedido para ser inserido o novo ID, código de segurança, saldo e valor da bateria do veículo do cliente, caso este exista na base de dados.

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 na base de dados, se tal acontecer, o cliente não é criado. Se o utilizador não pretender fazer nenhuma alteração, é perguntado se pretende inserir energia no posto.

* + 1. **Inserção de Energia**

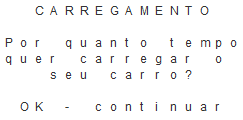
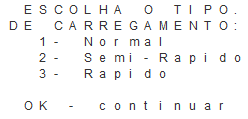
Para inserir energia, o utilizador deve escolher uma das 3 baterias existentes no posto, cada uma referente a um tipo de carregamento diferente. Após a escolha, é feita a inserção do valor de energia que se pretende carregar na bateria desejada, com as limitações indicadas anteriormente no periféricoreferente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



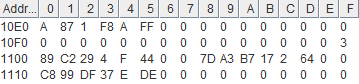
* + 1. **Carregamento do veículo**

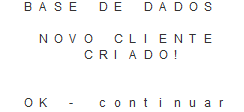
Para realizar um carregamento, é pedido ao utilizador para inserir o ID e código de segurança do cliente que pretende realizar o carregamento. Após a verificação do cliente, caso seja autorizado, é pedido para escolher o tipo de carregamento desejado, e de seguida, a duração do mesmo. Depois, é mostrado 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.

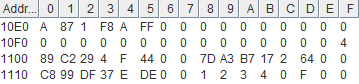


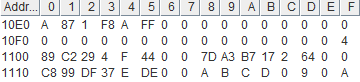
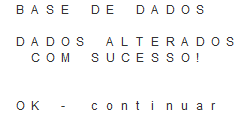
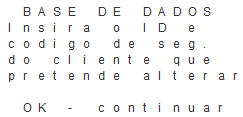
Após ser feito o débito, é realizada a atualização da bateria do posto referente ao tipo de carregamento escolhido pelo utilizador, sendo subtraído o valor da energia fornecida ao veículo à respetiva bateria, e aparece no display uma informação de que o carregamento foi concluído, gravam-se os valores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

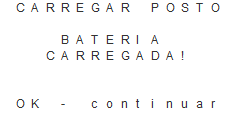
1. **Discussão de Resultados**

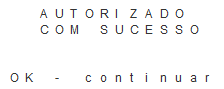
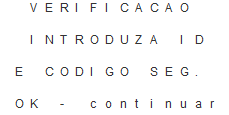


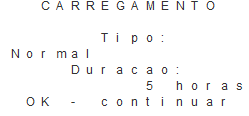
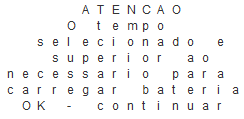


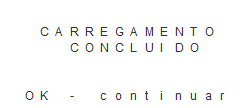
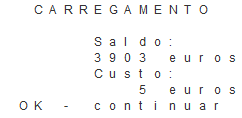


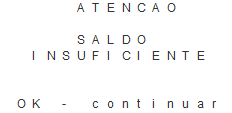












1. **Conclusão**

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

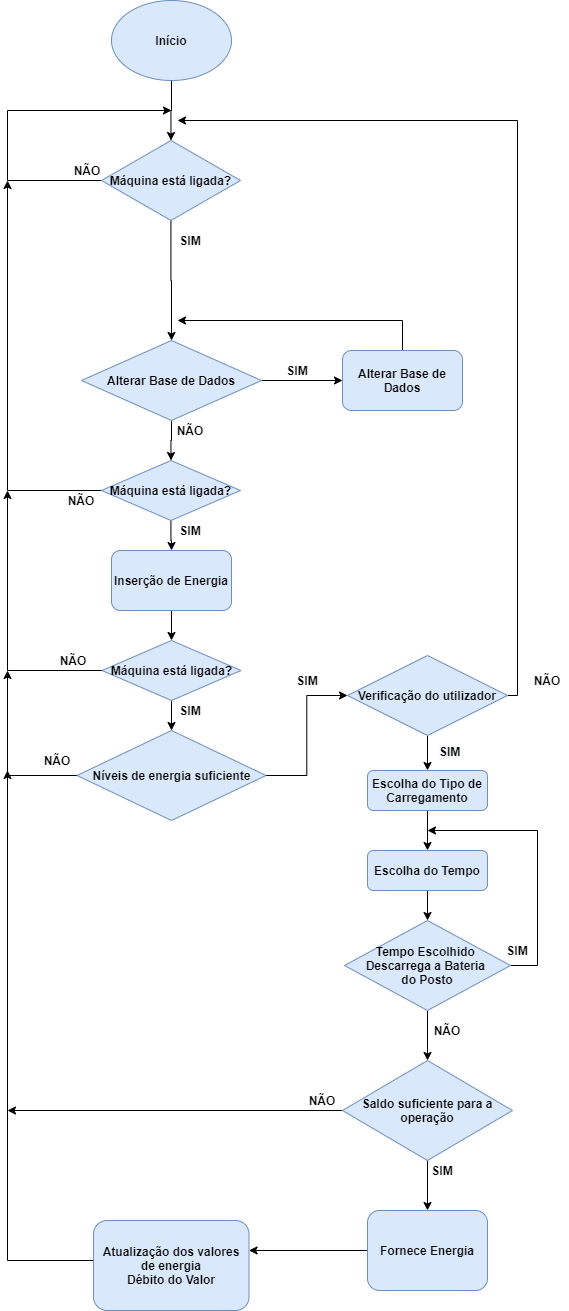
A linguagem Assembly embora seja mais complicada de perceber do que uma linguagem de alto nível, é mais fácil para codificar um dispositivo do que o código máquina.

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

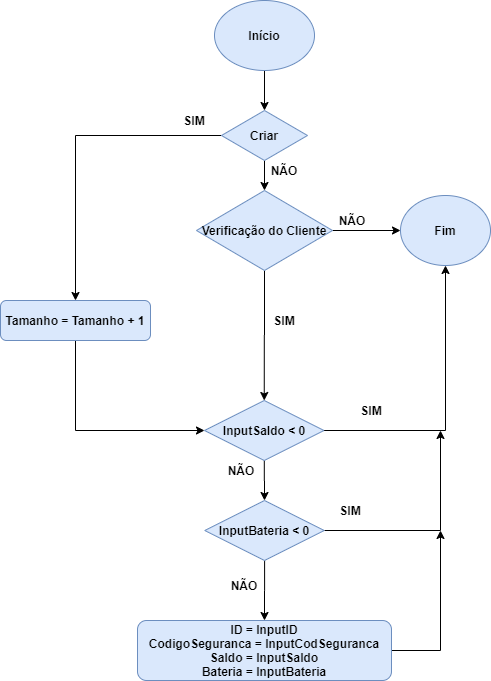
1. **Bibliografia**

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

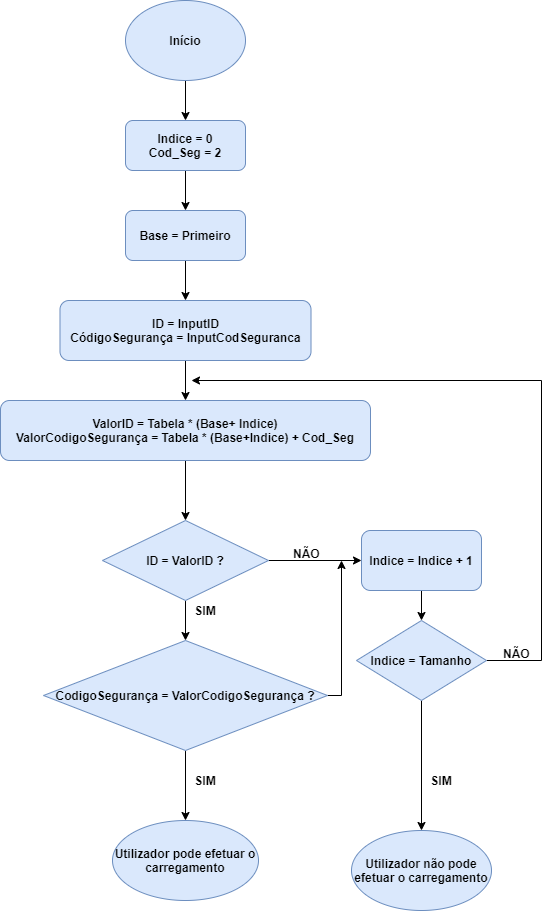
1. **Anexo A**
   1. **Fluxograma do Programa Geral**

****

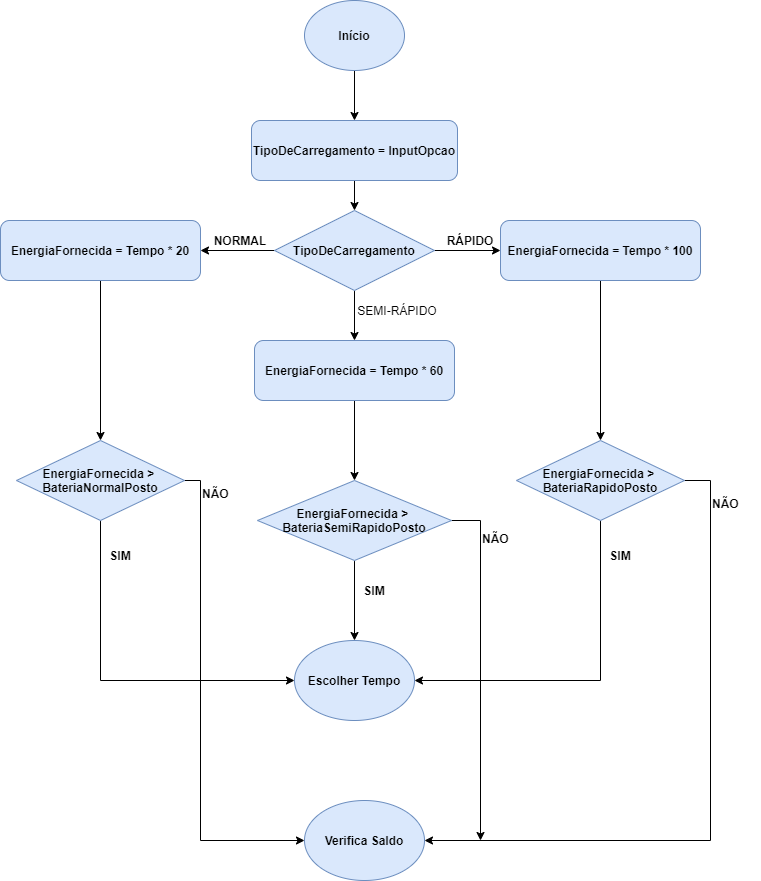
* 1. **Fluxograma da Alteração da Base de Dados**

****

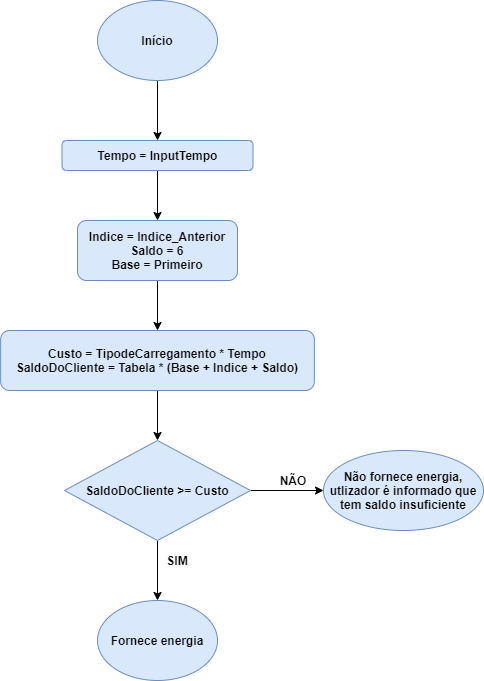
* 1. **Fluxograma da Verificação do Utilizador**

****

* 1. **Fluxograma do Descarregamento do Posto**

****

* 1. **Fluxograma da Verificação do Saldo**

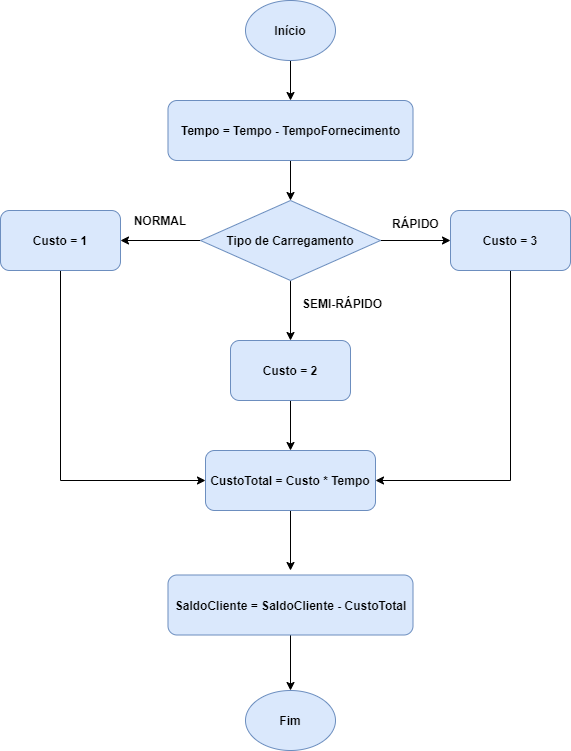


* 1. **Fluxograma do Fornecimento de Energia**

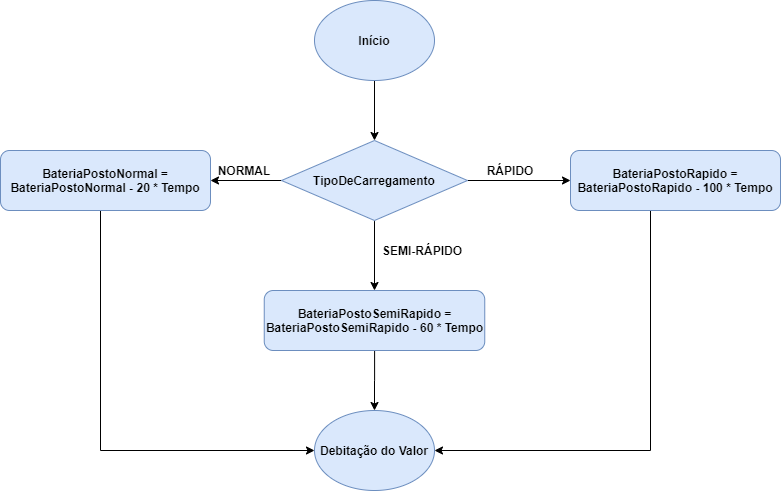
Uma imagem com computador

Descrição gerada automaticamente

* 1. **Fluxograma do Débito**

****

* 1. **Fluxograma da Atualização do Posto**

****

1. **Anexo B**