

Instituto Santos Dumont
Programa de Pós-Graduação em Neuroengenharia
Atividade 01 - Fundamentos de Programação
Discente: Tainá dos Santos Rêgo (2021020002)

1. As tecnologias de reabilitação tratadas no texto facilitam o amparo do paciente para a caminhada e retomada progressiva de sua capacidade motora.

No caso do **Lokomat®**, há uma estrutura mais complexa de suporte do corpo:

- Explicação a respeito da técnica ao paciente
- Coleta de medidas do paciente necessárias para a adequação do instrumento
- Utilização de fivelas no corpo do paciente como sistema de segurança
- Utilização de um suporte mecânico para manter o corpo na postura necessária
- Superfície de contato da máquina com o corpo do paciente em um formato acoplável e regulável
- Interface de comunicação entre o movimento e o software
 - a. Há um treinamento inicial de passada, como forma de calibração do software. A movimentação, a cadência e a velocidade são adequadas à necessidade do paciente. Os dados de movimento são enviados para o software do equipamento.
 - b. Como tomada de decisão, o início do treinamento se dá apenas com a estrutura devidamente acoplada no paciente. Assim como o treinamento deve ser parado caso o paciente reporte algum desconforto.
 - c. Como estruturas de repetição o software é configurado para a permanência da deambulação conforme o tempo definido previamente ou conforme uma estrutura de parada.
 - d. Estruturas recebidas de um usuário são referentes aos graus de atividade motora do paciente.
 - e. Informações a respeito do tempo, técnica, velocidade e metragem de marcha são apresentadas ao usuário para acompanhamento do treinamento.

No caso do **ZeroG®** a estrutura é destinada ao amparo de diferentes necessidades, fornecendo um suporte parcial:

- Explicação a respeito da técnica ao paciente
- Coleta de medidas do paciente necessárias para a adequação do instrumento
- Cadastramento dos dados do paciente
- Adequação de um suporte do tronco do paciente conforme os dados coletados e a necessidade
- A interface é calibrada com o treinamento e os dados, e realiza a medição enquanto ele está em andamento
- O software recebe os dados da caminhada e o paciente possui autonomia para definir o ritmo ou necessidades de parada.

2. A **memória** trata-se da resposta à execução de comandos cadastrados no software. Variáveis utilizadas para o feedback de ações, por exemplo, após o tempo definido de treinamento, o software retornar “Treinamento Finalizado”. Dessa forma, exemplos de variáveis dependentes de memória são: cadastro_paciente, tipo literal; treinamento_paciente, tipo literal; equipamento_em_posicao, tipo booleano; emergencia, tipo booleano; tempo_treinamento, tipo real; velocidade_treinamento, tipo real; tempo_medido, tipo real; velocidade_medida, tipo real.

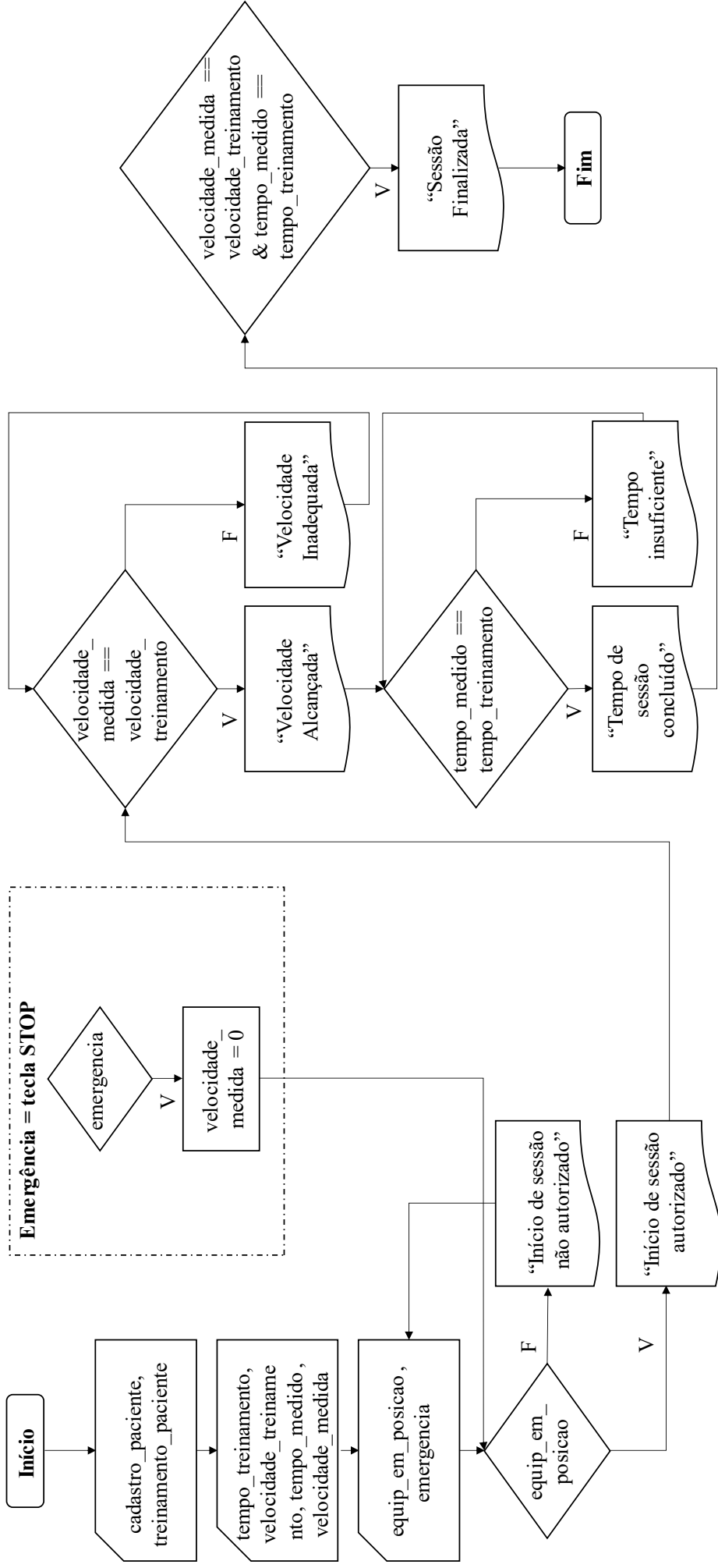
Já o **processamento** se trata da execução do comando cadastrado. Por exemplo: ligar a esteira na velocidade cadastrada (ligar_esteira); contabilizar a quantidade de passos (marcha_paciente01); contabilização do tempo de marcha (tempomarcha_paciente01); contabilização de metragem de caminhada (distancia_paciente01).

Por fim, **entrada e saída** dão os dispositivos acoplados ao sistema que contribuem para a execução de todo o treinamento. Por exemplo: display equipamento.

3. Estruturas de repetição são utilizadas para a definição do treinamento. Para a execução terapêutica e consequente melhora motora do paciente, é necessário que sejam feitas “N” sessões com o tempo estimado em “M” minutos. Para que o progresso do paciente seja como o esperado, é necessário que as sessões possuam um protocolo. Logo, uma estrutura de repetição é necessária para facilitar esse acompanhamento. Por exemplo, o treinamento é considerado “Em Andamento” enquanto o número de passos do dia < número de passos estabelecido.

No caso de uma estrutura de decisão, existem várias etapas do processo que podem precisar delas. Exemplos são: se os dados principais do paciente não estiverem cadastrados, não será iniciado o treinamento; caso o paciente relatar algum desconforto, parar o treinamento; caso o equipamento esteja mal instalado o software não inicia.

4. Diagrama de blocos



5. Pseudocódigo:

Algoritmo terapia_motora

Var cadastro_paciente, treinamento_paciente; **literal**

Var tempo_treinamento, velocidade_treinamento, tempo_medido, velocidade_medida;
real

Var equip_em_posicao, emergencia; **booleano**

Início

Leia cadastro_paciente, treinamento_paciente, tempo_treinamento,
velocidade_treinamento, tempo_medido, velocidade_medida

Se equip_em_posicao == true **então**

Escreva “Início de sessão autorizado”

Senão

Escreva “Início de sessão não autorizado”

Fim_se

Se emergencia == true **então**

 velocidade_medida = 0

Escreva “Parada de emergência”

Senão

Fim_se

Se velocidade_medida == velocidade_treinamento **então**

Escreva “Velocidade Alcançada”

Senão

Escreva “Velocidade inadequada”

Fim_se

Se tempo_medido == tempo_treinamento **então**

Escreva “Tempo do experimento cumprido”

Senão

Escreva “Tempo insuficiente”

Fim_se

Se velocidade_medida == velocidade_treinamento & tempo_medido ==
tempo_treinamento **então**

Escreva “Sessão finalizada”

Senão

Fim_se

Fim