Trabalho 3 – Teste de Software

Alunos: Stephanie Fay (1721631) e Pedro Paulo da Silva (1421422)

# Questão 1;

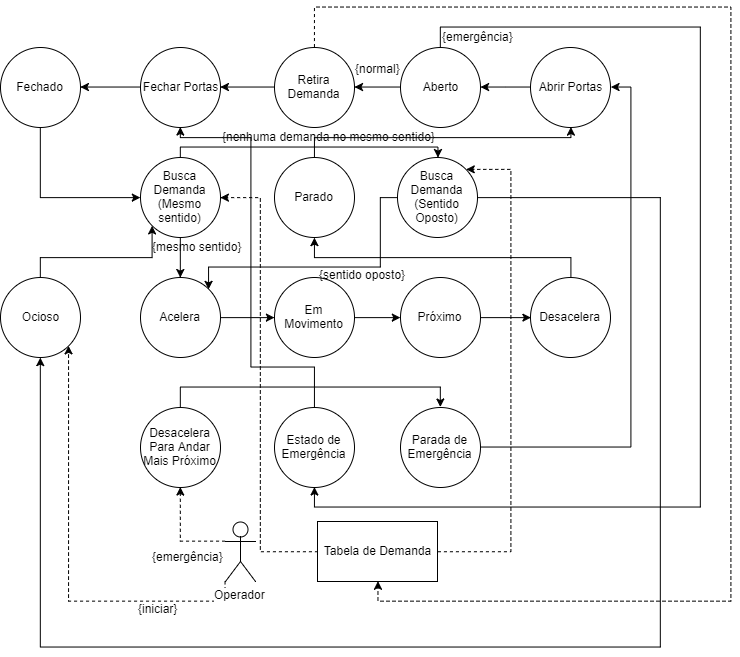
# 

# Questão 2;

Controle do Conjunto de Elevadores

# 

Controle do Motor



# Questão 3;

* Se firstCall == 0 || lastCall == 0, então calls = 0
* Se firstCall != 0 || lastCall != 0, então calls > 0
* Se calls > 0, então firstCall != nil && lastCall != nil
* Se calls == 0, então firstCall = 0 && lastCall = 0
* call.before != nil se call.before->next == call
* call.next != nil se call.next->before == call
* 0 < idFloor <= Floors

# Questão 4;

Ocioso

Elevador está ocioso, esperando atualizações na tabela de demanda ou vindas do operador

**Parâmetros Conceituais:**

* tabela\_demanda
  + tabela que contém os andares pedidos

**Resultados Conceituais:**

* estado\_elevador
  + elevador se mantém parado no estado ocioso

Busca Demanda (Mesmo Sentido)

Observa a tabela demanda esperando por pedidos na direção atual do elevador

**Parâmetros Conceituais:**

* sentido
  + contem qual o sentido que o elevador estava se movendo
* andar
  + contem o andar qual o elevador se encontra
* tabela\_demanda
  + tabela que contém os andares pedidos

**Resultados Conceituais:**

* prox\_andar
  + responde qual o próximo andar que o elevador deve parar

Busca Demanda (Sentido Oposto)

Inverte o sentido do elevador. Observa a tabela demanda esperando por pedidos na nova direção do elevador

**Parâmetros Conceituais:**

* sentido
  + contem o sentido que o elevador está se movendo
* andar
  + contem o andar qual o elevador se encontra
* tabela\_demanda
  + tabela que contem os andares pedidos

**Resultados Conceituais:**

* prox\_andar
  + responde qual o próximo andar que o elevador deve parar

Acelera

Elevador ganha aceleração no sentido atual

**Parâmetros Conceituais:**

* sentido
  + contem o sentido que o elevador está se movendo
* velocidade
  + velocidade qual o elevador deve se mover

**Resultados Conceituais:**

* estado
  + elevador em movimento

Em movimento

Enquanto se move ainda observa a tabela de demandas em busca de novas demandas no sentido atual

**Parâmetros Conceituais:**

* andar
  + contem o andar qual o elevador se encontra
* prox\_andar
  + andar para qual o elevador se destina
* sentido
  + contem o sentido que o elevador está se movendo
* velocidade
  + velocidade qual o elevador deve se mover
* tabela\_demanda
  + tabela que contem os andares pedidos

**Resultados Conceituais:**

* prox\_andar
  + pode vir alterar qual o próximo andar qual o elevador se destina

Próximo

Verifica a distância para o próximo andar

**Parâmetros Conceituais:**

* andar
  + contem o andar qual o elevador se encontra
* prox\_andar
  + andar para qual o elevador se destina
* sentido
  + contem o sentido que o elevador está se movendo
* velocidade
  + velocidade qual o elevador deve se mover

**Resultados Conceituais:**

* distância
  + diferença entre o andar atual e o próximo andar que o elevador deverá parar

Desacelera

Entra em processo de desaceleração para parada

**Parâmetros Conceituais:**

* andar
  + contem o andar qual o elevador se encontra
* prox\_andar
  + andar para qual o elevador se destina
* sentido
  + contem o sentido que o elevador está se movendo
* velocidade
  + velocidade qual o elevador deve se mover
* distancia
  + diferença entre o andar atual e o próximo andar que o elevador deverá parar

**Resultados Conceituais:**

* Velocidade
  + velocidade é diminuída conforme a aproximação do andar destino

Parado

Elevador interrompe o movimento em um dado andar – e não entre eles

**Parâmetros Conceituais:**

* andar
  + contem o andar qual o elevador se encontra
* sentido
  + contem o sentido que o elevador deverá se mover após a parada

**Resultados Conceituais:**

* estado
  + estado atualizado de “em movimento” para “parado”

Abrir Portas

Comando para abrir portas é dado

**Parâmetros Conceituais:**

* estado\_portas
  + estado que as portas se encontravam incialmente

**Resultados Conceituais:**

* estado\_portas
  + estado alterado para o que acontece no momento