

Projeto do Elevador

Experiência Criativa – BCC – PUCPR

Prof. Alcides Calsavara

Prof. Edson Justino

HARDWARE

INDICADOR DE POSIÇÃO DO ELEVADOR

**INDICADORES
DE ESTADO**

CONTROLADOR

**BOTÕES PARA
SUBIR E DESCER
EM CADA ANDAR**

BOTÕES PARA IR AOS ANDARES

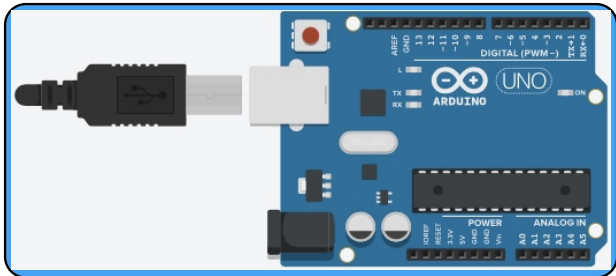
**BOTÃO
PARA
FECHAR
PORTA**

**BOTÕES
ON
OFF**

**BOTÃO
DE
EMERGÊNCIA**

INDICADOR DE POSIÇÃO DO ELEVADOR

**INDICADORES
DE ESTADO**



**BOTÕES PARA
SUBIR E DESCER
EM CADA ANDAR**

BOTÕES PARA IR AOS ANDARES

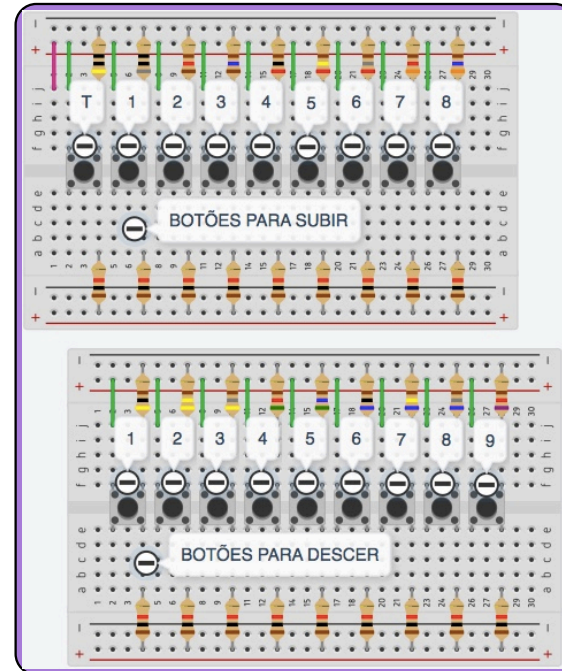
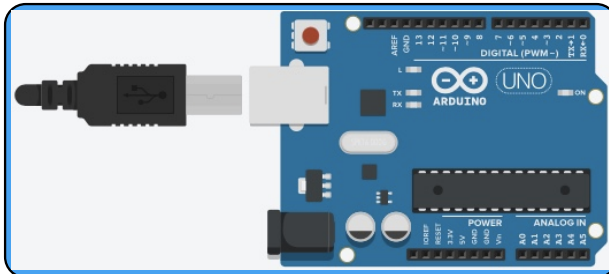
**BOTÃO
PARA
FECHAR
PORTA**

**BOTÕES
ON
OFF**

**BOTÃO
DE
EMERGÊNCIA**

INDICADOR DE POSIÇÃO DO ELEVADOR

INDICADORES DE ESTADO



BOTÕES PARA IR AOS ANDARES

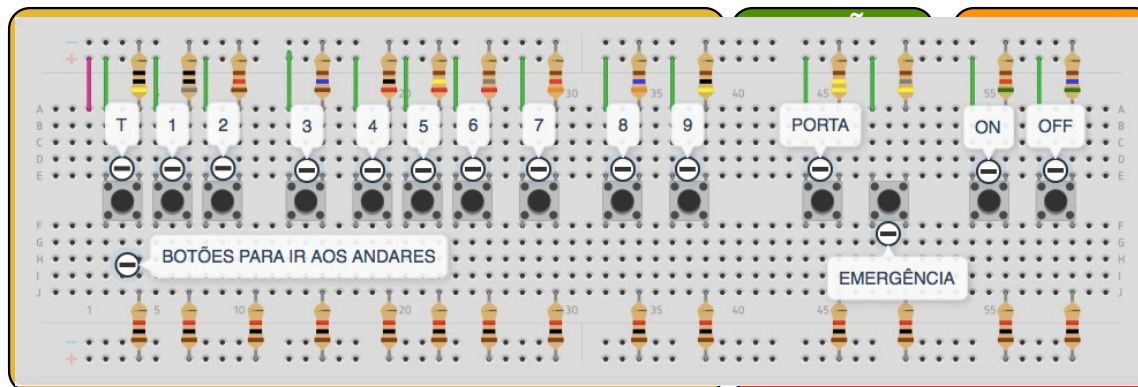
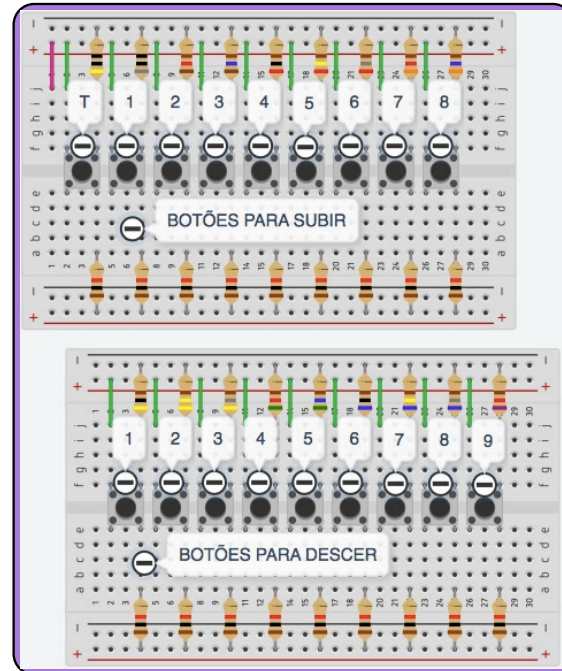
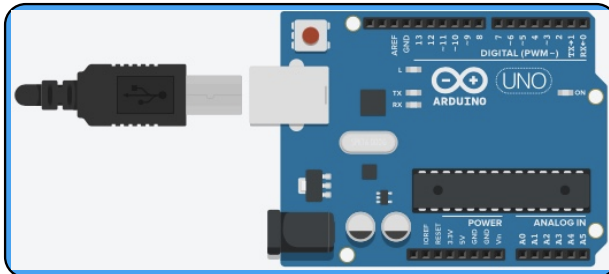
**BOTÃO
PARA
FECHAR
PORTA**

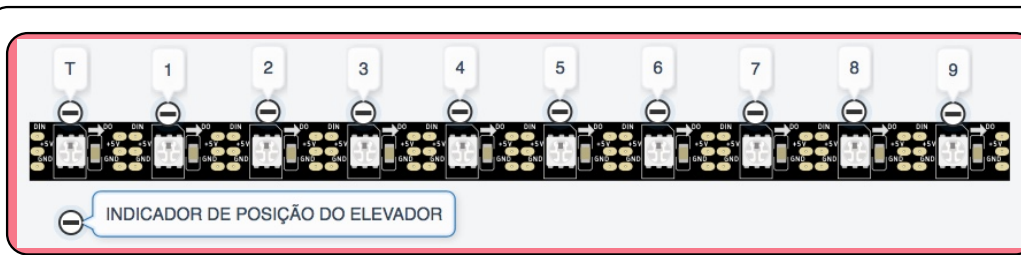
**BOTÕES
ON
OFF**

**BOTÃO
DE
EMERGÊNCIA**

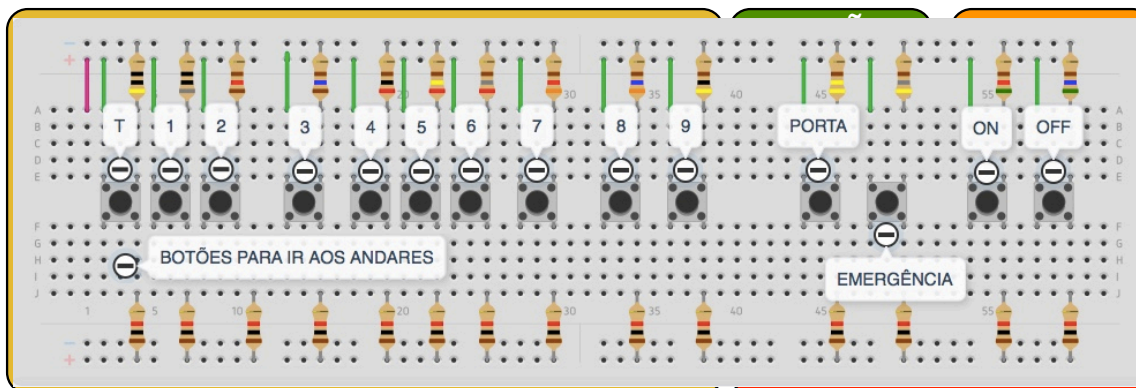
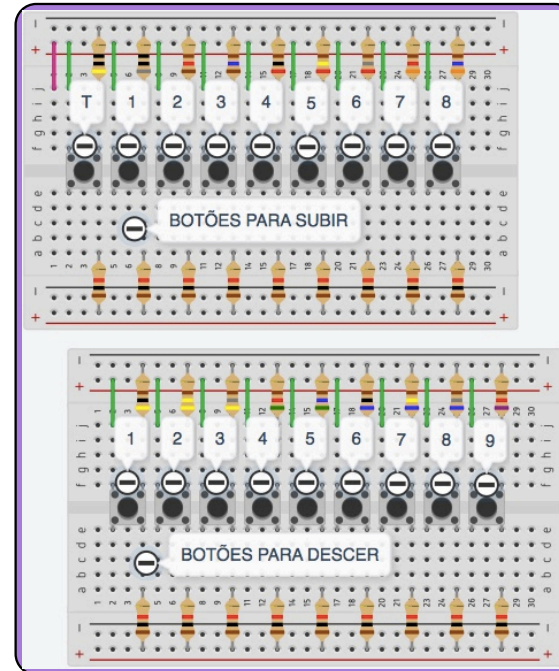
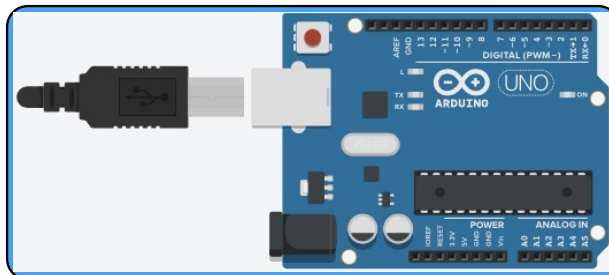
INDICADOR DE POSIÇÃO DO ELEVADOR

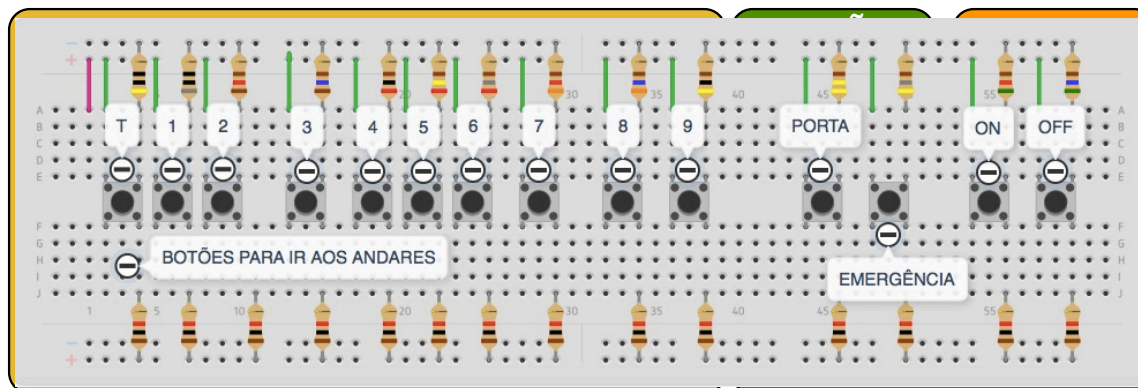
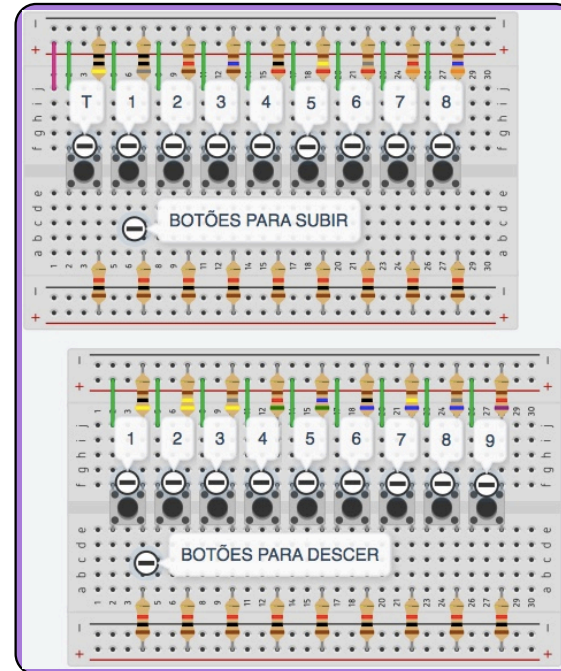
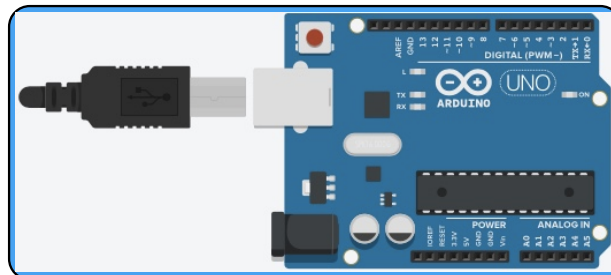
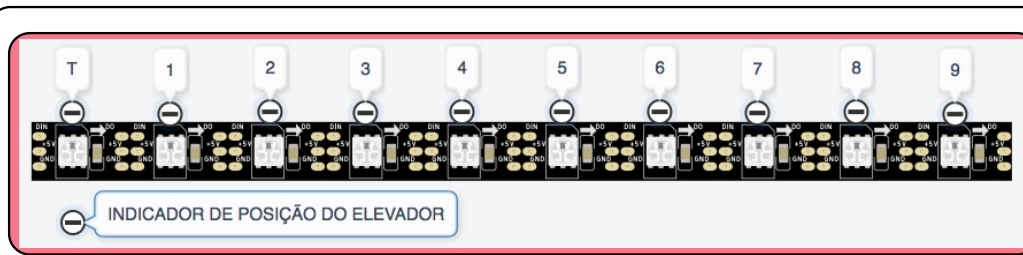
INDICADORES DE ESTADO





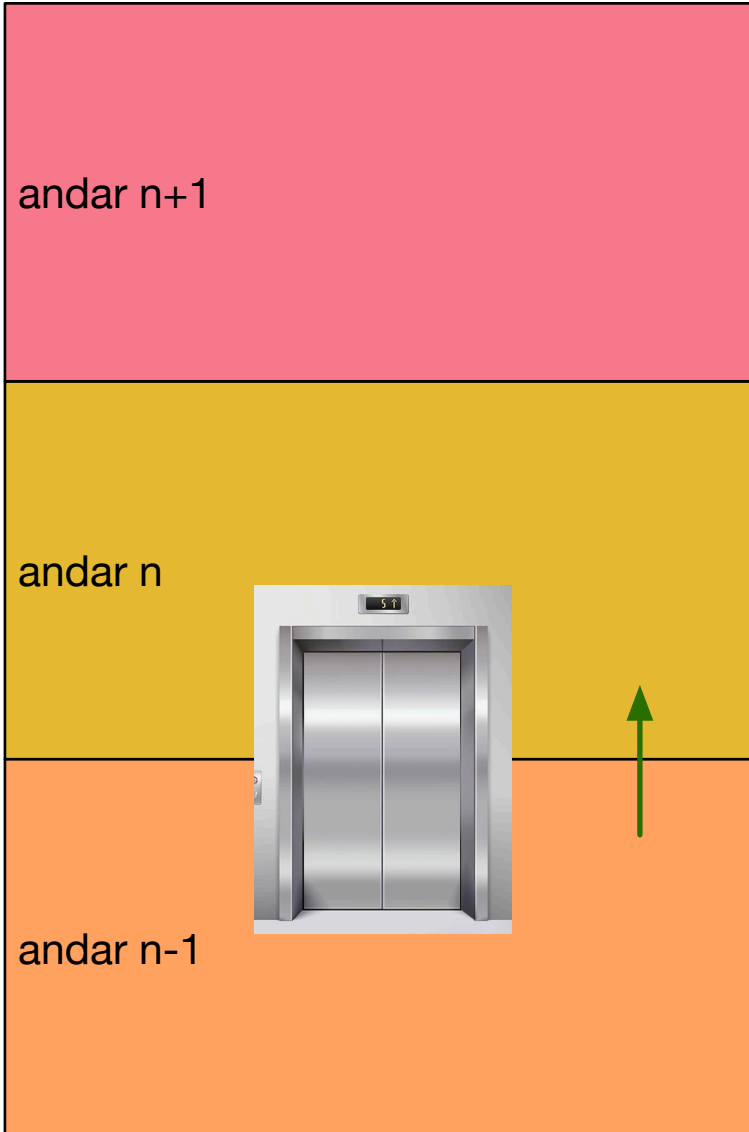
INDICADORES DE ESTADO





SOFTWARE

ESTADOS DO ELEVADOR



**MOVENDO
SUBINDO**



**ALINHADO
SUBINDO**

andar n+1

andar n



andar n-1

**ESTACIONADO
SUBINDO**

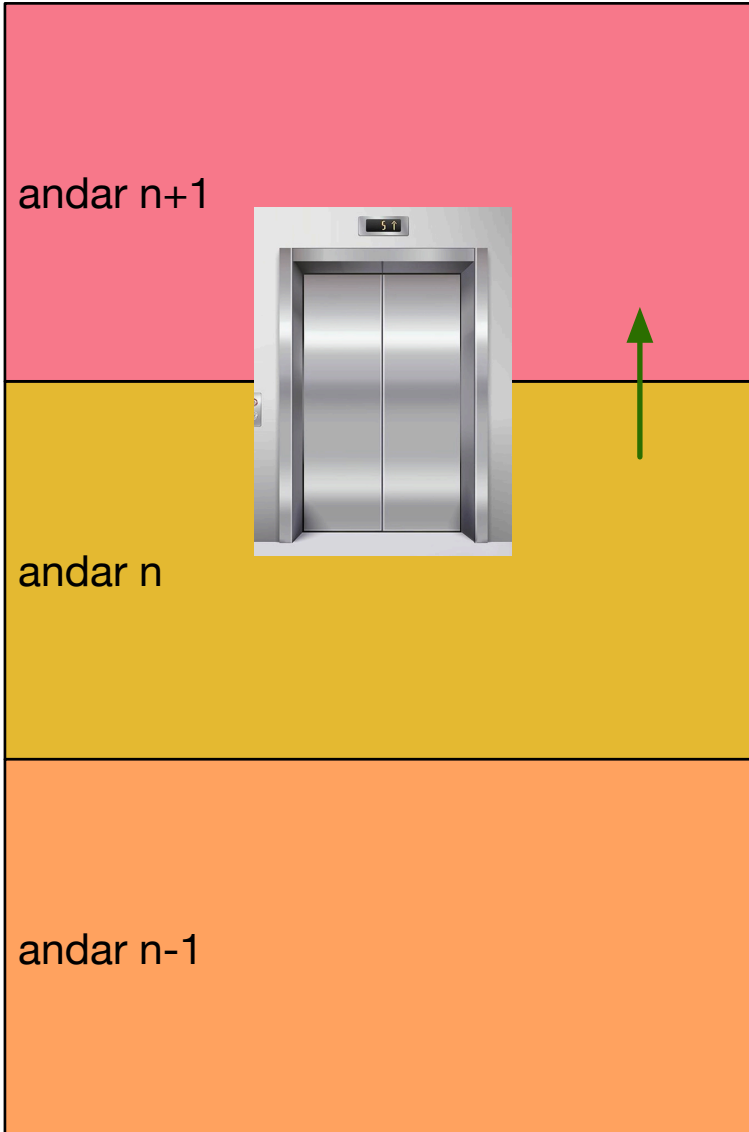
andar n+1

andar n

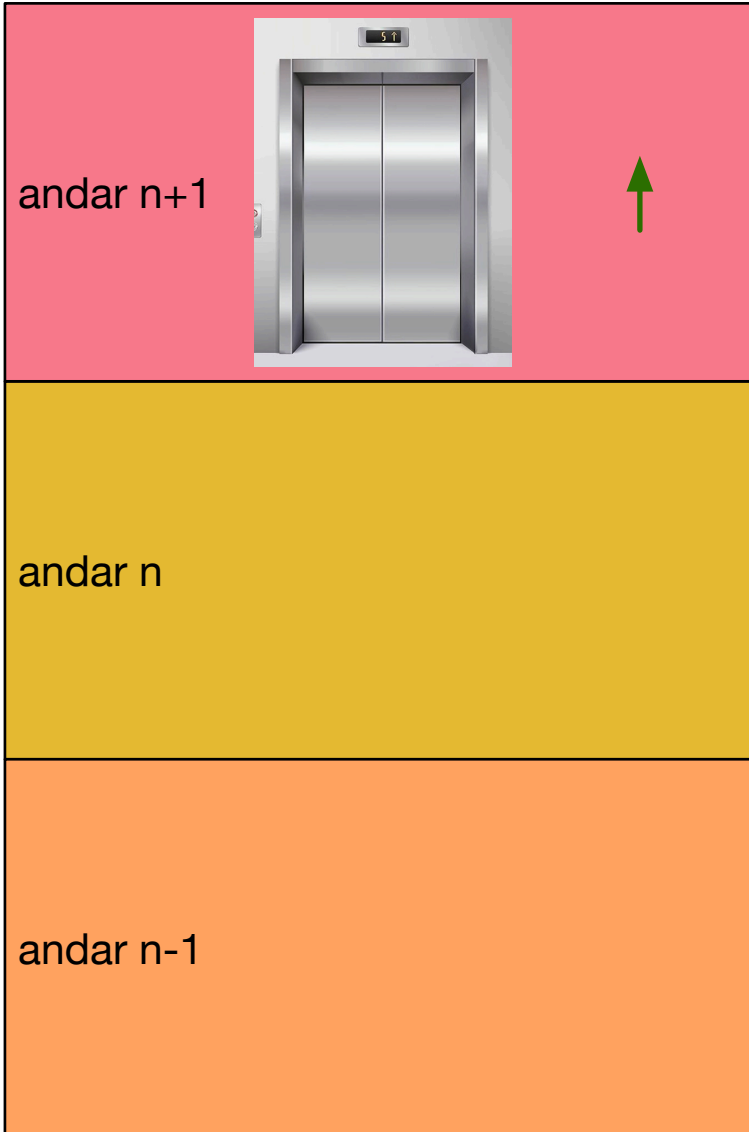
andar n-1



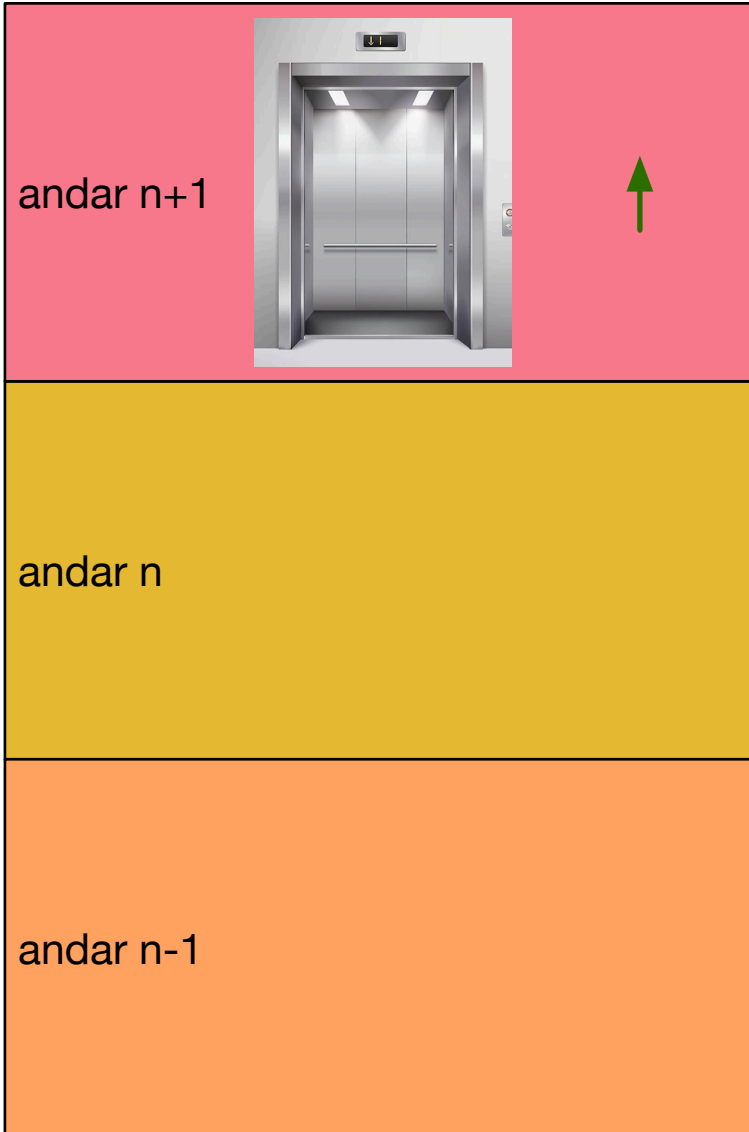
**ALINHADO
SUBINDO**



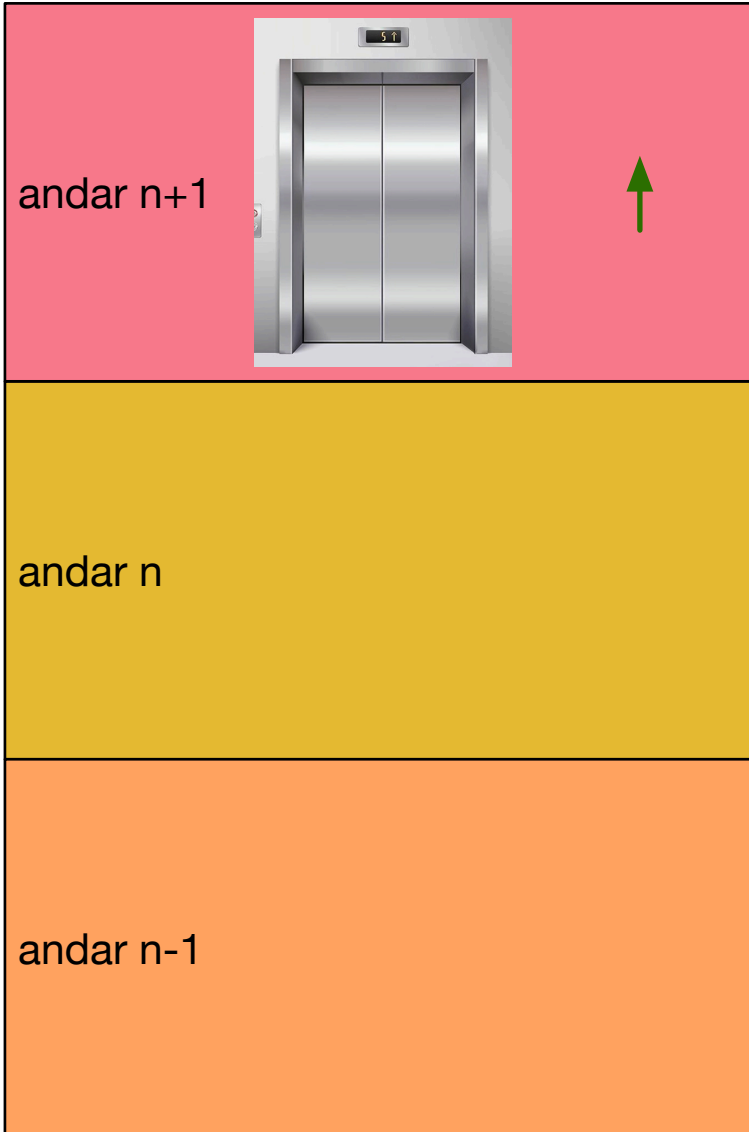
**MOVENDO
SUBINDO**



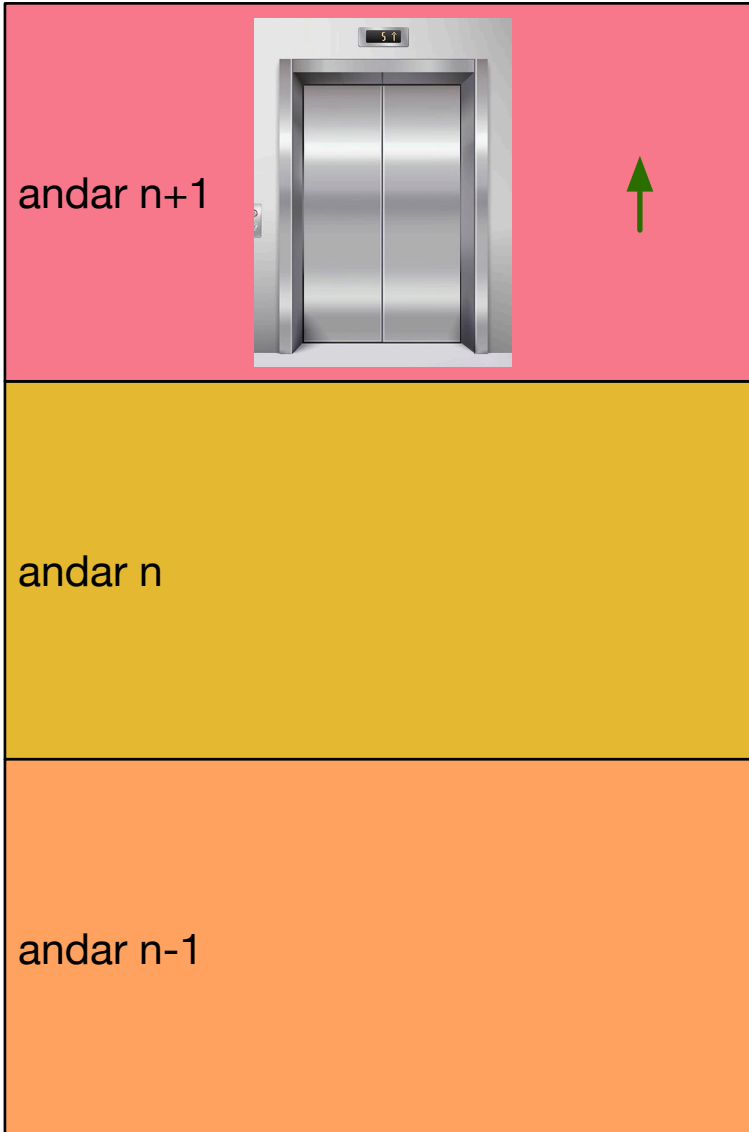
**ALINHADO
SUBINDO**



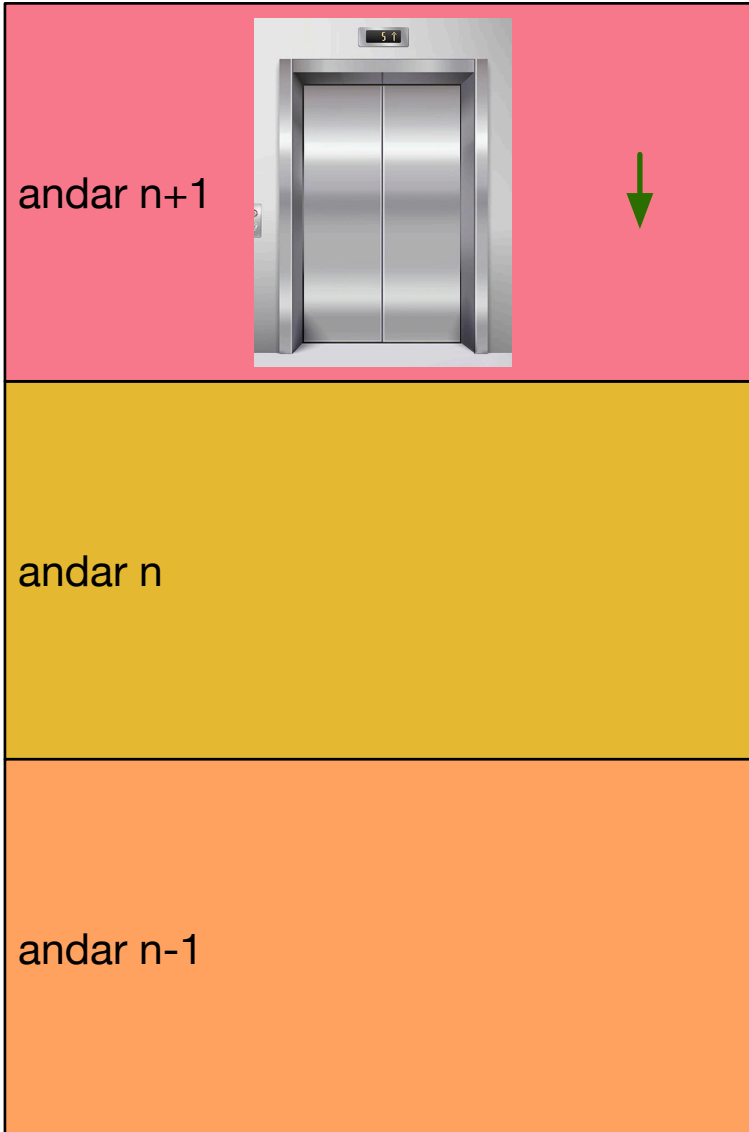
**ESTACIONADO
SUBINDO**



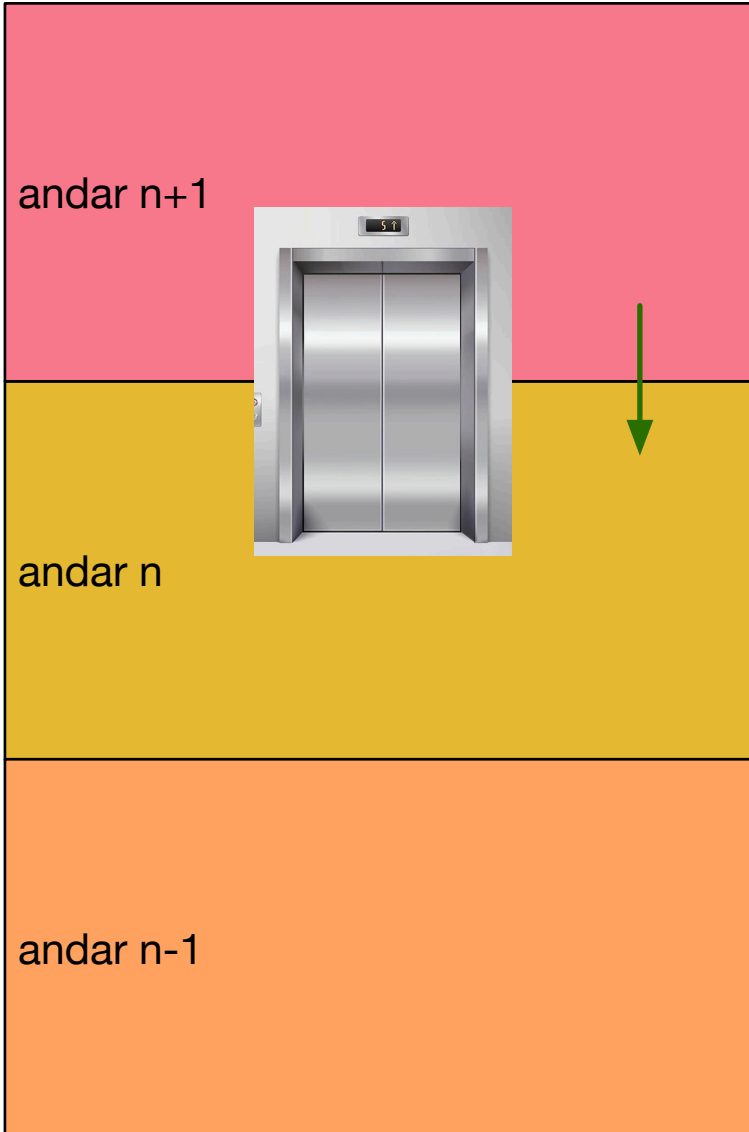
**ALINHADO
SUBINDO**



**FIM DA
SUBIDA**



**ALINHADO
DESCENDO**



**MOVENDO
DESCENDO**

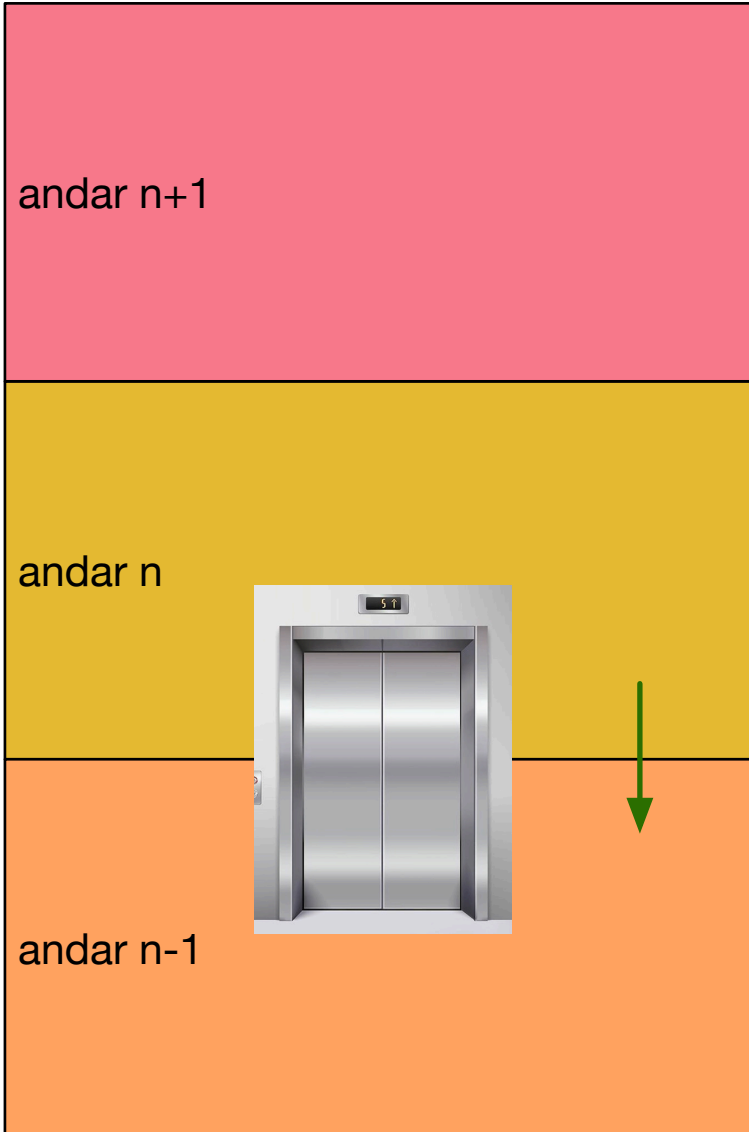
andar n+1

andar n



**ALINHADO
DESCENDO**

andar n-1



**MOVENDO
DESCENDO**



**ALINHADO
DESCENDO**



**ESTACIONADO
DESCENDO**



**ALINHADO
DESCENDO**

andar n+1

andar n

andar n-1



**FIM DA
DESCIDA**

andar n+1

andar n

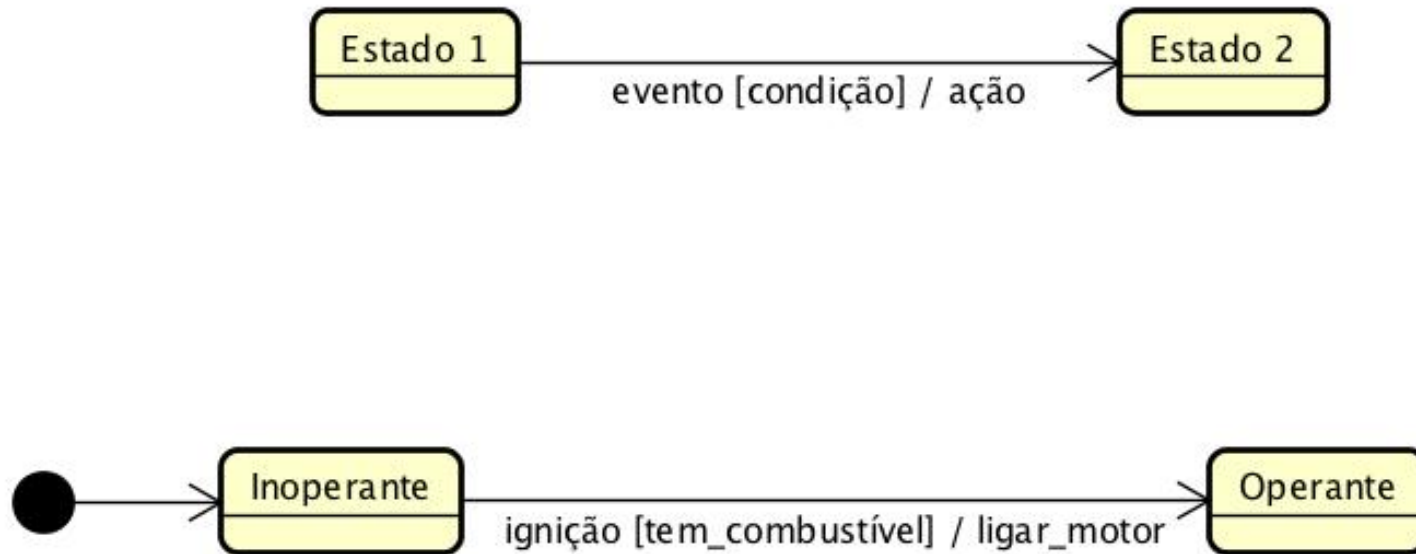
andar n-1



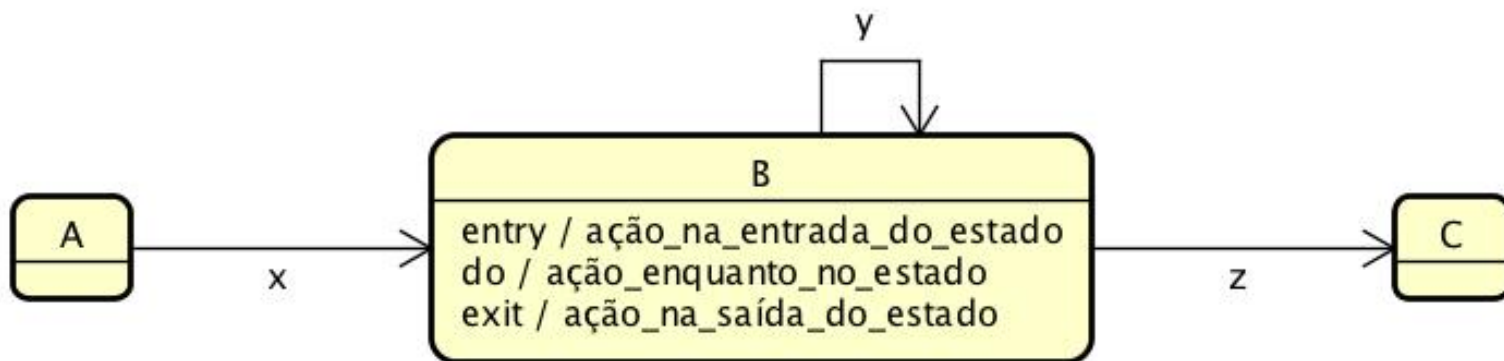
OCIOSO

NOTAÇÃO PARA DIAGRAMA DE ESTADOS

stm Estados e Transições



stmAções Internas



stm Estados Aninhados

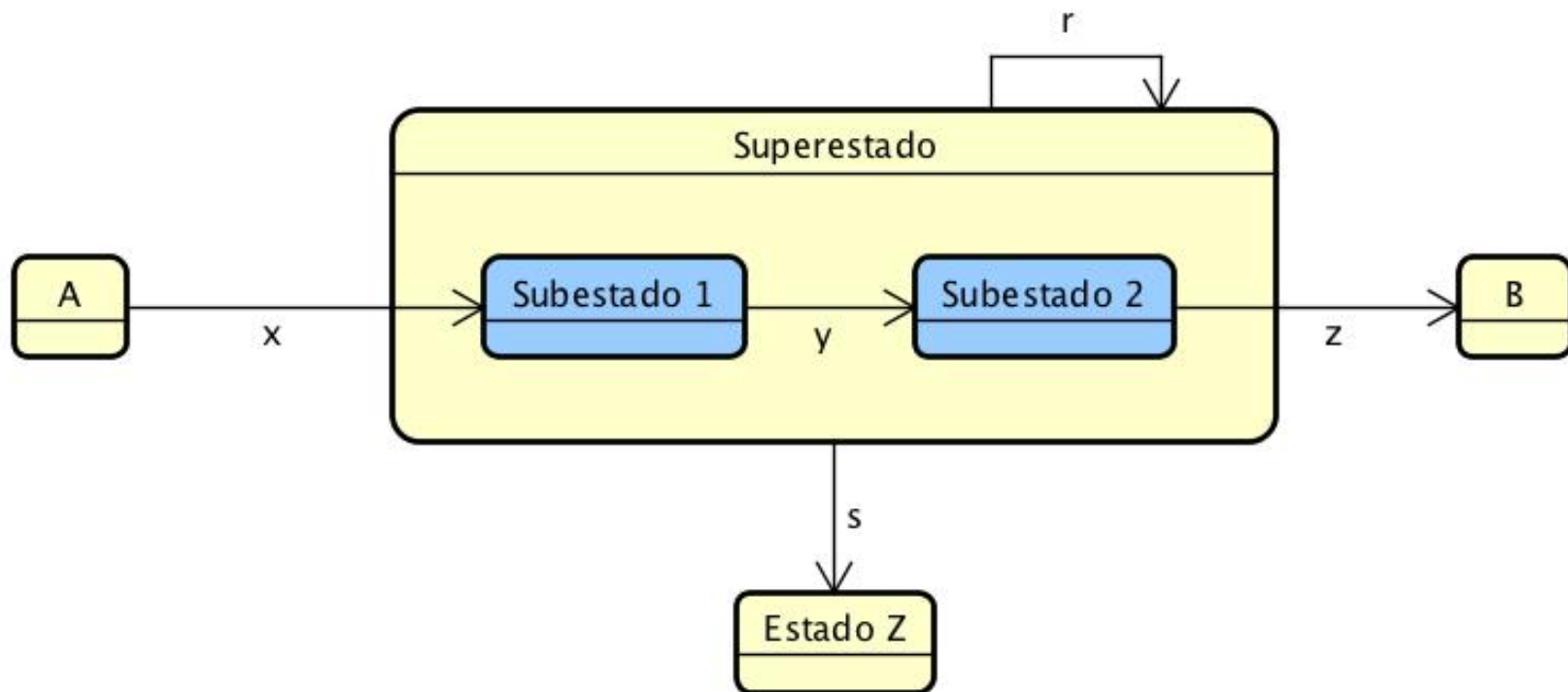
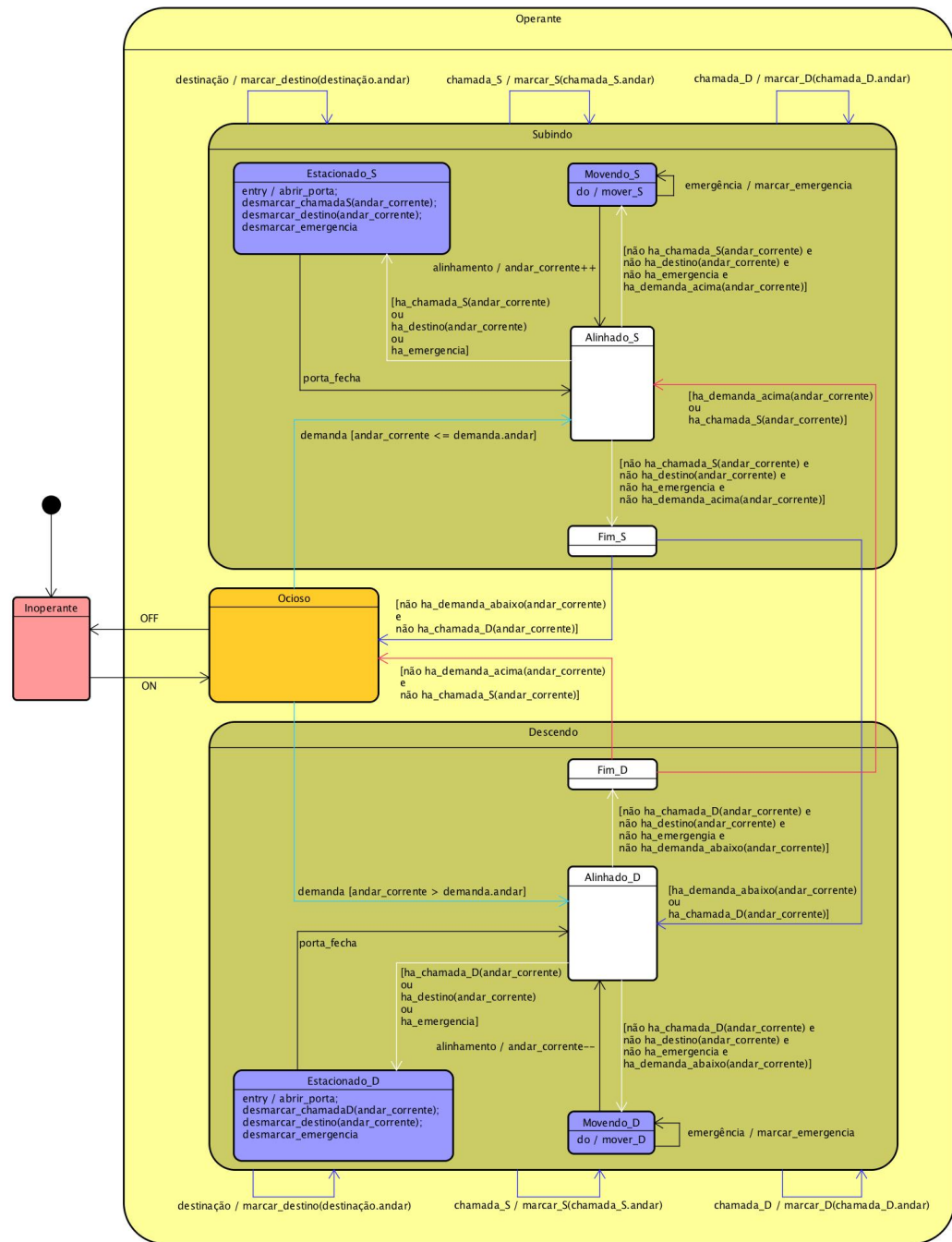


DIAGRAMA DE ESTADOS DO ELEVADOR



Eventos

1. **ON:** botão para iniciar a operação do elevador é apertado
2. **OFF:** botão para terminar a operação do elevador é apertado
3. **porta_fechada:** botão para fechar a porta do elevador é apertado
4. **alinhamento:** elevador alinha-se com um andar
5. **emergência:** botão de emergência é apertado
6. **destinação:** seleção de um andar para ir
7. **chamada_S:** chamada do elevador em algum andar para subir
8. **chamada_D:** chamada do elevador em algum andar para descer
9. **demanda:** pode ser um evento de **destinação**, **chamada_S** ou **chamada D**

Ações

1. marcar_destino(x): marca o andar x como destino para o elevador estacionar
2. marcar_S(x): marca o andar x para estacionar na subida
3. marcar_D(x): marca o andar x para estacionar na descida
4. marcar_emergência: marca que o botão de emergência foi acionado

Ações

5. desmarcar_destino(x): desmarca o andar x como destino para o elevador estacionar
6. desmarcar_S(x): desmarca o andar x para estacionar na subida
7. desmarcar_D(x): desmarca o andar x para estacionar na descida
8. desmarcar_emergência: desmarca que o botão de emergência foi acionado

Condições

1. $ha_emergencia$: verifica se está marcado que o botão de emergência foi acionado
2. $ha_destino(x)$: verifica se o andar x está marcado como destino para o elevador
3. $ha_chamada_S(x)$: verifica se o andar x está marcado para estacionar na subida
4. $ha_chamada_D(x)$: verifica se o andar x está marcado para estacionar na descida

Condições

5. `ha_demanda_acima(x)`: verifica se há algum andar acima do andar `x` que esteja marcado para estacionar (na subida ou na descida) ou como destino para o elevador
6. `ha_demanda_abaixo(x)`: verifica se há algum andar abaixo do andar `x` que esteja marcado para estacionar (na subida ou na descida ou como destino para o elevador)

Programação dos Estados

É importante notar que, na programação de cada estado, pode ser necessário escrever um **loop** que aguarde até ocorrer algo que provoque a mudança de estado.

Por exemplo, a função que trata o estado INOPERANTE pode ser programada como mostrado a seguir. O comando **while** garante que só ocorre mudança de estado (para OCIOSO) depois que a tecla ON for apertada.

Função para o Estado INOPERANTE

```
void estado_INOPERANTE()
{
    Serial.println("==> INOPERANTE");

    mostrar_elevador(andar_corrente, R_inoperante, G_inoperante, B_inoperante);
    desmarcar_demandas();
    tecla_ON_apertada = false;

    while (!tecla_ON_apertada)
    {
        delay(500); // não é necessário, mas evita testar demais
    }

    estado = OCIOSO;
}
```