

- Regra 1: Inicializar o Estado do Sistema
 - Condição: Quando o sistema é iniciado.
 - Ação: Configurar o estado inicial do drone e as propriedades do obstáculo.
 - Detalhes:
 - $\text{estado_drone} \leftarrow \text{'Descolagem'}$
 - $\text{posicao_atual} \leftarrow \text{'Base'}$
 - $\text{posicao_drone} \leftarrow \text{POSICOES['Base']}$
 - $\text{destino} \leftarrow \text{POSICOES['A']}$
 - $\text{proximo_destino} \leftarrow \text{'A'}$
 - $\text{movimento_completo} \leftarrow \text{False}$
 - $\text{obstaculo_y} \leftarrow \frac{\text{ALTURA} - \text{OBSTACULO_ALTURA}}{2}$
 - $\text{obstaculo_direcao} \leftarrow 1 \# (1 \text{ para baixo, } -1 \text{ para cima})$
- Regra 2: Transição de Estado do Drone
 - Condição: Se $\text{movimento_completo} = \text{True}$ e $\text{estado_drone} = \text{'Descolagem'}$.
 - Ação: Atualizar o estado do drone para "Em Rota".
 - Detalhes:
 - $\text{estado_drone} \leftarrow \text{'Em Rota'}$

- **Regra 3: Definir o Próximo Destino**
 - **Condição:** Se `movimento_completo = True`, `estado_drone = 'Em Rota'`, e `evitando_obstaculo = False`.
 - **Ação:** Atualizar a posição atual, definir o próximo destino e reconfigurar o estado de movimento.
 - **Detalhes:**
 - `posicao_atual` \leftarrow `proximo_destino`
 - `proximo_destino` \leftarrow `ação_obter_proxima_posicao(posicao_atual)`
 - `destino` \leftarrow `POSICOES[proximo_destino]`
 - `movimento_completo` \leftarrow `False`
 - `rota_alternativa` \leftarrow `None`
- **Regra 4: Atualizar a Posição do Obstáculo**
 - **Condição:** Sempre que a função de atualização de estado é chamada.
 - **Ação:** Mover o obstáculo verticalmente de acordo com a direção atual e inverter a direção se atingir os limites.
 - **Cálculos:**
 - `obstaculo_y` \leftarrow `obstaculo_y + OBSTACULO_VELOCIDADE` \times `obstaculo_direcao`
 - Se `obstaculo_y \leq 0` ou `obstaculo_y + OBSTACULO_ALTURA \geq ALTURA`, então `obstaculo_direcao` \leftarrow $-\text{obstaculo_direcao}$

- Regra 5: Movimento em Direção ao Destino
 - Condição: Se movimento_completo = False e não há rota alternativa ativa.
 - Ação: Calcular o novo posicionamento do drone e atualizar a sua posição.
 - Cálculos:
 - $x1, y1 \leftarrow \text{posicao_drone}$
 - $x2, y2 \leftarrow \text{destino}$
 - $dx \leftarrow x2 - x1$
 - $dy \leftarrow y2 - y1$
 - $\text{distancia} \leftarrow \sqrt{dx^2 + dy^2}$
 - Se $\text{distancia} \geq \text{velocidade}$, então:
 - $\text{fator} \leftarrow \text{velocidade} / \text{distancia}$
 - $\text{novo_x} \leftarrow x1 + dx \times \text{fator}$
 - $\text{novo_y} \leftarrow y1 + dy \times \text{fator}$
 - $\text{posicao_drone} \leftarrow (\text{novo_x}, \text{novo_y})$
- Regra 6: Verificação de Colisão
 - Condição: Se R5 está ativo (drone em movimento).
 - Ação: Verificar se a trajetória do drone cruzará o obstáculo e, se sim, calcular uma rota alternativa.
 - Cálculos:
 - Se $\min(x1, \text{novo_x}) \leq \text{LARGURA} // 2 \leq \max(x1, \text{novo_x})$, então:
 - Calcular a interseção y no ponto onde $x = \text{LARGURA} // 2$:
 - $t \leftarrow \frac{\text{LARGURA} // 2 - x1}{\text{novo_x} - x1}$
 - $y_{\text{intersecao}} \leftarrow y1 + t \times (\text{novo_y} - y1)$
 - Se $\text{obstaculo_y} \leq y_{\text{intersecao}} \leq \text{obstaculo_y} + \text{OBSTACULO_ALTURA}$:
 - Ativar evitando_obstaculo e calcular rota_alternativa

- Regra 7: Movimento na Rota Alternativa
 - Condição: Se $\text{evitando_obstaculo} = \text{True}$.
 - Ação: Calcular e seguir uma rota alternativa até que o obstáculo seja evitado.
 - Cálculos:
 - Atualizar destino \leftarrow rota_alternativa
- Regra 8: Atualização da Próxima Posição
 - Condição: Quando o drone alcança um destino.
 - Ação: Determinar o próximo destino na sequência.
 - Cálculos:
 - $\text{ordem} \leftarrow ['A', 'B', 'C', 'D', 'Base']$
 - $\text{idx} \leftarrow \text{ordem.index(posicao_atual)}$
 - $\text{proximo_destino} \leftarrow \text{ordem}[(\text{idx} + 1) \% \text{len}(\text{ordem})]$