

Plano de Especificação Formal em Redes de Petri – Jogo ODS 14

1) Objetivo e Escopo

Formalizar o comportamento do jogo educativo ODS 14 (navio, coleta de lixo, resgate de fauna, ondas e cards) por meio de **Redes de Petri** (RdP), permitindo **verificação de propriedades** (segurança, vivacidade, ausência de deadlock, limitação de recursos, metas de onda) e **análise de desempenho** (vazão de capturas, taxa de resgates, penalidades).

2) Escolhas de Modelagem

- **Tipo de RdP:**
- **RdP Colorida (CPN)** para distinguir tipos de lixo (garrafa, sacola, pneu) e animais (peixe, tartaruga) sem multiplicar sub-redes idênticas.
- **RdP Temporizada** (Time Petri Nets, TPN) para modelar spawn (inter-arrival), cooldown do disparo, duração de ondas e expiração de projéteis (redes).
- **Hierárquica/Modular:** sub-redes independentes para *Spawn*, *Disparo/Projétil*, *Captura*, *Resgate*, *Onda/Meta*, *Card/Pausing*.
- **Granularidade:** nível de evento (spawn, disparo, colisão, resgate, penalidade, avanço de onda). A movimentação contínua é **abstraída** por habilitação de transições (e.g., "projétil e lixo se encontram" \approx transição Capturar habilitada sob condições lógicas/temporais).

3) Vocabulário Formal

3.1 Definição (simplificada)

Uma RdP temporizada e colorida $\mathcal{N} = (P, T, F, W, C, M_0, I)$, onde: - P é o conjunto de lugares; T é o conjunto de transições; $F \subseteq (P \times T) \cup (T \times P)$ é o fluxo; W as multiplicidades (pesos) nos arcos. - C é o conjunto de cores (tipos): $LIXO = \{\text{garrafa, sacola, pneu}\}$, $ANIMAL = \{\text{peixe, tartaruga}\}$, $REDE$, $ONDA$ etc. - M_0 é a marcação inicial; $I : T \rightarrow [a, b]$ atribui **janela temporal** (mín-máx de firing) às transições temporizadas.

3.2 Tokens e Cores

- Tokens em P_Lixos carregam cor $LIXO$ e atributos (pontos, reciclagem).
- Tokens em $P_Animais$ carregam cor $ANIMAL$.
- Tokens em P_Redes representam projéteis ativos.
- Tokens em P_Meta carregam progresso da reciclagem e meta da onda.
- Tokens em P_Onda identificam a onda atual.

4) Decomposição em Sub-Redes

4.1 Sub-rede de Onda/Meta

Intuição: controla duração e objetivo (meta de reciclagem) da onda. - Lugares: $P_OndaAtual$, P_Meta (progresso), $P_TempoOnda$, $P_OndaConcluida$, $P_PausaCard$. - Transições:

T_IniciaOnda, T_AvancaMeta, T_TempoEsgota, T_FinalizaOnda, T_MostraCard, T_ContinuaJogo. - Temporalidade: T_TempoEsgota com intervalo $[D, D]$ (duração da onda). T_ContinuaJogo habilitada via evento do jogador. - Invariantes de lugar (ex.): sempre **no máx. 1 token** em P_OndaAtual e P_PausaCard (controle de modo).

4.2 Sub-rede de Spawn (Lixo e Animais)

- Lugares: P_SpawnLixo, P_Lixos, P_SpawnAnimal, P_Animais.
- Transições temporizadas: T_GeraLixo (taxa λ_{lixo} variando por onda), T_GeraAnimal (taxa λ_{animal} com jitter).
- Guardas/cores: seleção probabilística da cor do token (mix da onda). Em CPN, usar case/if para garrafa/sacola/pneu.

4.3 Sub-rede de Disparo/Projétil (Rede)

- Lugares: P_RedePronta (cooldown), P_Redex (ativas).
- Transições: T_Dispara (consome de P_RedePronta, produz em P_Redex), T_Recarga (temporizada; repõe token em P_RedePronta após cooldown), T_ExpiraRede (temporizada; remove de P_Redex).
- Propriedade: **limitação** de projéteis simultâneos via capacity de P_Redex ou bound em P_RedePronta.

4.4 Sub-rede de Captura (Rede × Lixo)

- Lugares: P_Redex, P_Lixos, P_Capturas, P_Pontuacao, P_Meta.
- Transição: T_Capturar (sincroniza presença de uma REDE e um LIXO habilitando coleta).
Efeitos:
- Remove 1 token de P_Redex e 1 de P_Lixos (da mesma "área lógica").
- Produz 1 token em P_Capturas (log de evento), incrementa P_Pontuacao, avança P_Meta (via T_AvancaMeta).
- Propriedades: **vivacidade** (enquanto há lixo e redes, captura permanece possível) e **limitação** (não criar tokens fantasma).

4.5 Sub-rede de Resgate/Proteção (Navio × Animal × Rede)

- Lugares: P_Animais, P_Resgates, P_Penalidades, P_Pontuacao.
- Transições: T_Resgatar (Navio encontra Animal), T_FeriuAnimal (Rede atinge Animal).
- Efeitos:
- T_Resgatar: remove 1 ANIMAL de P_Animais, adiciona em P_Resgates e incrementa P_Pontuacao.
- T_FeriuAnimal: remove 1 ANIMAL de P_Animais e 1 REDE de P_Redex, adiciona 1 penalidade.
- Propriedade: **segurança** (sempre que T_FeriuAnimal dispara, há decremento coerente em pontuação/reciclagem).

4.6 Sub-rede de Card/Pausa

- Lugares: P_PausaCard, P_InputContinuar.
- Transições: T_MostraCard (entra em pausa), T_ContinuaJogo (sai da pausa).
- Invariantes: Em modo pausa, transições de jogo (spawn, captura, etc.) ficam **inibidas** (modelar por inibição ou partição de modos com lugares de controle).

5) Mapeamento Eventos ↔ RdP

Evento do jogo	Transição (RdP)	Pré-condições (lugares)	Pós-efeitos
Iniciar Onda	T_IniciaOnda	P_OndaAtual vazio, P_PausaCard vazio	token em P_OndaAtual, P_TempoOnda
Spawn Lixo	T_GeraLixo	P_SpawnLixo, mix da onda	token colorido em P_Lixos
Spawn Animal	T_GeraAnimal	P_SpawnAnimal	token em P_Animais
Disparar Rede	T_Dispara	P_RedePronta	token em P_Redes, inicia T_ExpiraRede
Capturar Lixo	T_Capturar	P_Redes & P_Lixos	P_Capturas, P_Pontuacao↑, P_Meta↑
Resgatar Animal	T_Resgatar	P_Animais	P_Resgates, P_Pontuacao↑
Ferir Animal	T_FeriuAnimal	P_Redes, P_Animais	P_Penalidades↑, P_Pontuacao↓, P_Meta↓
Meta atingida	T_FinalizaOnda	P_Meta ≥ objetivo	P_OndaConcluida, P_PausaCard
Tempo esgotado	T_TempoEsgota	relógio da onda	P_OndaConcluida, P_PausaCard
Avançar para próxima	T_ContinuaJogo	P_PausaCard, P_InputContinuar	limpa P_PausaCard, novo T_IniciaOnda

6) Marcação Inicial (M_0) – Exemplo

- P_OndaAtual : 1 token (onda 1)
- P_Meta : 0 tokens (progresso = 0, meta = 15 via parâmetro)
- P_TempoOnda : 1 token com temporizador D_1
- P_RedePronta : 1 token (cooldown ok)
- P_Lixos, P_Animais, P_Redes : vazios
- P_Pontuacao, P_Resgates, P_Penalidades : 0
- P_PausaCard : vazio

7) Propriedades a Verificar

7.1 Segurança (Safety)

- **P1 – Coerência de captura:** Ao disparar T_Capturar, sempre existe 1 REDE e 1 LIXO consumidos; não há criação de tokens sem causa.

- **P2 – Proteção à fauna:** $T_FeriuAnimal$ implica decrementos em $P_Pontuacao$ e/ou P_Meta conforme regras.
- **P3 – Pausa consistente:** Com token em $P_PausaCard$, transições de jogo (spawn/captura/resgate) ficam bloqueadas.

7.2 Vivacidade/Liveness

- **L1 – Sem deadlock em onda ativa:** Enquanto $P_OndaAtual$ tiver token e jogo não estiver pausado, existe sequência infinita de habilitações (spawn e interações) até $T_FinalizaOnda / T_TempoEsgota$.

7.3 Limitação/Boundedness

- **B1 – Redes/projéteis limitados:** $|P_Redes| \leq K$ (ex.: bound pelo pool).
- **B2 – Lixos/Animais:** limites superiores dependem das taxas e limpeza; parametrizar para manter estados finitos.

7.4 Invariantes (esboço)

- **P-invariante de modo:** $m(P_OndaAtual) + m(P_OndaConcluida) = 1$ por onda.
- **P-invariante de pausa:** $m(P_PausaCard) \in \{0,1\}$ e $m(P_PausaCard)=1 \Rightarrow$ spawn/captura/resgate inibidos.
- **T-invariantes (ciclos):** ciclos de disparo $Disparar-Expirar/Colidir$ para redes.

8) Métricas e Desempenho

- **Vazão de** $T_Capturar$ (capturas por segundo) e de $T_Resgatar$ (resgates por segundo).
- **Taxa de penalidades** (disparos indevidos em animais).
- **Tempo médio para atingir meta** por onda; **probabilidade** de finalizar por meta vs. por tempo.

9) Plano de Validação/Verificação

1. **Ferramentas** (sugestões):
2. **CPN Tools** (CPN/temporal), **TINA** (Time Petri Nets), **PIPE** (Petri net), **TimeNet**.
3. **Experimentos:**
4. Cenários básicos: (i) baixa taxa de lixo, (ii) alta taxa, (iii) cooldown maior/menor, (iv) mais animais.
5. *Stress test*: taxas máximas para checar boundedness e ausência de deadlock.
6. **Propriedades formais:**
7. Model checking para reachability (atingir meta), invariantes (modo/pausa), ausência de deadlock.
8. **Calibração:**
9. Ajuste de $I(T)$ (intervalos temporais) para refletir parâmetros do jogo (spawnRate, cooldown, duração da onda, expiração da rede).

10) Especificações FormaIS (Incidência – exemplo mínimo)

Considere o submodelo $Captura$ com lugares $P = \{P_R, P_L, P_C\}$ e transição $T = \{T_C\}$ (R=redes, L=lixos, C=capturas). Matriz de incidência $C = C^+ - C^-$:

$$C^- = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \quad C^+ = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} -1 \\ -1 \\ +1 \end{bmatrix}.$$

Para marcação $M = (r, l, c)^T$, T_C habilita se $r \geq 1 \wedge l \geq 1$. Disparo: $M' = M + C$ (consome 1 de P_R , 1 de P_L , produz 1 em P_C). Em CPN, restrinja por cor (pareamento válido Rede \times Lixo).

11) Traços de Execução (Cenários)

- **Cenário A (normal):** $T_IniciaOnda \rightarrow \text{spawns } T_GeraLixo / T_GeraAnimal \rightarrow \text{disparos } T_Dispara \rightarrow T_Capturar / T_Resgatar \rightarrow T_FinalizaOnda \text{ (meta)} \rightarrow \text{card } T_MostraCard \rightarrow T_ContinuaJogo$.
- **Cenário B (penalidade):** $T_Dispara + T_FeriuAnimal \rightarrow P_Penalidades \uparrow$ e $P_Pontuacao \downarrow \rightarrow \text{ainda assim onda conclui por } T_TempoEsgota$.

12) Rastreabilidade Jogo \leftrightarrow RdP

- Disparo/rede \leftrightarrow Sub-rede Disparo/Projétil.
- Coleta/reciclagem \leftrightarrow Sub-rede Captura + Onda/Meta.
- Resgate/penalidade \leftrightarrow Sub-rede Resgate/Proteção.
- Card/pausa \leftrightarrow Sub-rede Card/Pausa.

13) Entregáveis

- **Modelo CPN/TPN** (.cpn ou .net) com sub-redes e parâmetros por onda.
- **Especificações formais** (este plano + matrizes/incidências dos principais submodelos).
- **Relatório de verificação** (propriedades, logs de firing, gráficos de vazão/tempo).

14) Extensões Futuras

- Boss/evento (macro-lixo): nova sub-rede com estados de fase e vida.
- Power-ups (rede maior, ímã): lugares de buffer e transições de ativação/expiração.
- Mobile (toques): entrada de eventos como lugares de input.