

# Plano de Especificação Formal

## 1. Introdução

Este documento apresenta o plano de especificação formal para a aplicação **Gestor Pomodoro**, um software desktop que auxilia no gerenciamento do tempo de estudo utilizando a técnica Pomodoro. A especificação formal visa garantir a correção, consistência e ausência de ambiguidade na definição do sistema.

## 2. Objetivos

- Modelar formalmente os requisitos funcionais do sistema.
- Utilizar a notação Z para descrever o comportamento do software.

## 3. Metodologia

- Levantamento dos requisitos funcionais.
- Modelagem dos estados e operações usando a notação Z.

## 4. Linguagem

- **Linguagem:** Notação Z.

## 5. Especificação Formal em Z

### 5.1 Declarações Básicas

Definimos alguns tipos abstratos e constantes para representar os usuários, tempos (em unidades naturais – por exemplo, minutos ou segundos) e os estados do cronômetro.

Unset

```
[USER]           % Tipo abstrato para Usuário
TEMPO == ℕ       % Tempo representado como número natural
```

```
EstadoDoCronometro ::= ocioso | estudando | pausando | pausado
```

```
% ocioso: estado em que o sistema não está realizando nenhuma operação.
```

```
% estudando: estado ativo, onde o usuário está envolvido em uma atividade
```

```
% pausando: estado de 'transição' em que o sistema está
migrando de estudando para pausado
% pausado: estado em que o sistema é interrompido
brevemente, mas mantendo as informações salvas
```

## 5.2 Estado do Sistema Pomodoro

Definimos um esquema que representa o estado global do sistema, contendo os limites configurados, o estado atual do cronômetro e, para efeitos de relatório, um histórico de sessões concluídas.

```
Unset
PomodoroSystem
    tempoDeEstudo : TEMPO
    % Tempo configurado para estudo
    tempoDePausa : TEMPO
    % Tempo configurado para pausa
    estado : EstadoDoCronometro
    % Estado atual do cronômetro
    tempoDeEstudoRestante : TEMPO
    % Tempo restante na fase de estudo
    tempoDePausaRestante: TEMPO
    % Tempo restante na fase de pausa
    historico : seq (Registro)

% Invariante: se o sistema estiver ocioso, o tempo
restante deve estar igual aos tempos configurados.
∴ estado = ocioso ⇒ (tempoDeEstudoRestante =
tempoDeEstudo ∧ tempoDePausaRestante = tempoDePausa)
```

## 5.3 Operações do Sistema

### 5.3.1 Cadastro dos Limites de Tempo

Esta operação permite que o usuário cadastre os tempos de estudo e pausa, mas somente se não houver um cronômetro ativo.

Unset

RegistrarLimitesDeTempo

$\Delta$ SistemaPomodoro

novoSudo? : TEMPO

novoSusa? : TEMPO

-----

pre: estado = ocioso

post: tempoSudo' = novoSudo?  $\wedge$

tempoSusa' = novoSusa?  $\wedge$

tempoSdeSudoRestante' = novoSudo?  $\wedge$

tempoSdeSusaRestante' = novoSusa?  $\wedge$

estado' = estado

### 5.3.2 Início da Sessão Pomodoro

Para iniciar uma sessão, os tempos devem ter sido configurados (valores positivos) e o sistema deve estar inativo.

Unset

IniciarPomodoro

$\Delta$ SistemaPomodoro

-----

pre: estado = ocioso  $\wedge$  tempoSudo > 0  $\wedge$  tempoSusa > 0

post: estado' = estudando  $\wedge$

tempoSdeSudoRestante' = tempoSdeSudo  $\wedge$

tempoSdeSusaRestante' = tempoSdeSusa

### 5.3.3 Contagem Regressiva

Para simular a passagem do tempo, definimos operações de contagem que decrementam o tempo restante da fase atual. Se o tempo chegar a zero, ocorre a transição de estado e a emissão de notificação.

**Durante a fase de estudo:**

Unset

ContagemEstudando

$\Delta$ SistemaPomodoro

```

-----
pre: estado = estudando  $\wedge$  tempoEstudoRestante > 0
post: tempoEstudoRestante' = - 1  $\wedge$ 
      estado' = (se tempoEstudoRestante - 1 = 0 então
emPausa senão estudando)  $\wedge$ 
      tempoPausaRestante' = tempoPausaRestante

```

#### Durante a fase de pausa:

```

Unset
ContagemPausando
 $\Delta$ SistemaPomodoro
-----
pre: estado = emPausa  $\wedge$  tempoPausaRestante > 0
post: tempoPausaRestante' = tempoPausaRestante - 1  $\wedge$ 
      estado' = (se tempoPausaRestante - 1 = 0 então
ocioso senão emPausa)  $\wedge$ 
      tempoEstudoRestante' = tempoEstudoRestante

```

### 5.3.4 Operações de Controle Manual (Pausa, Retomada e Reinício)

#### Parar o cronômetro:

```

Unset
PararTemporizador
 $\Delta$ SistemaPomodoro
-----
pre: estado  $\in$  {estudando, emPausa}
post: estado' = parado  $\wedge$  (tempoEstudoRestante =
tempoEstudoRestante  $\wedge$  tempoPausaRestante' =
tempoPausaRestante)

```

#### Retomar o cronômetro (após uma pausa):

Unset

RetomarCronometro

$\Delta$ SistemaPomodoro

-----

pré: estado = pausado

pós: estado' = (se tempoEstudoRestante > 0 então emEstudo  
                  senão se tempoPausaRestante > 0 então

emPausa senão inativo)

$\wedge$  (tempoEstudoRestante' = tempoEstudoRestante  $\wedge$   
tempoPausaRestante' = tempoPausaRestante)

### Reiniciar o cronômetro para a fase atual:

Unset

ResetarCronometro

$\Delta$ SistemaPomodoro

-----

pré: estado  $\neq$  inativo

pós: estado' = inativo  $\wedge$   
      tempoEstudoRestante' = tempoEstudo  $\wedge$   
      tempoPausaRestante' = tempoPausa

### 5.3.5 Notificações Automáticas

Embora as notificações possam ser consideradas eventos externos (outputs), podemos formalizá-las como parte das transições.

### Notificação de término da fase de estudo (para iniciar a pausa):

Unset

NotificarPausa

$\Delta$ SistemaPomodoro

-----

pre: estado = estudando  $\wedge$  tempoEstudoRestante = 0

post: estado' = emPausa  $\wedge$  tempoPausaRestante' =  
tempoPausa

### Notificação de término da fase de pausa (para retomar o estudo):

```
Unset
NotificarEstudo
ΔSistemaPomodoro
-----
pré: estado = emPausa ∧ tempoPausaRestante = 0
pós: estado' = inativo ∧
    % Registra a sessão concluída no histórico
    histórico' = histórico ~ <registro> ∧
    tempoEstudoRestante' = tempoEstudo ∧
    tempoPausaRestante' = tempoPausa
```

### 5.3.6 Geração de Relatório Gráfico

Para fins de relatório, o sistema extrai dados do histórico de sessões concluídas. O tipo do relatório pode ser abstrato.

```
Unset
RELATORIO ::= reatorio
GerarRelatorio
ΞSistemaPomodoro
relatorio! : RELATORIO
-----
pre: #(historio) > 0
post: relatorio! = < "Gráfico com: " , #(historico) , "
    sessões, estudo total: " , Σ { p: historico .
    p.tempoEstudo } >
```

## 6. Conclusão

Esta especificação formal em Z transforma os requisitos originais em modelos e operações que definem, de forma precisa, o comportamento esperado do sistema Pomodoro.