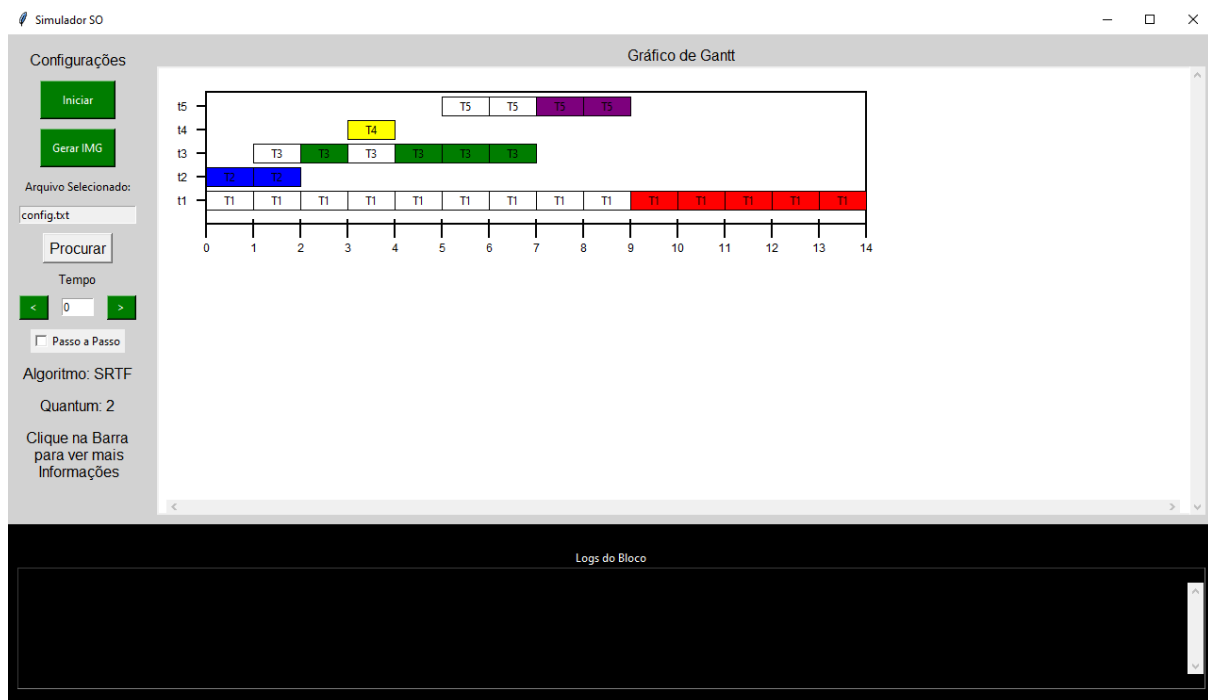


Descrição da interface do PROJETO A

Visão Geral:

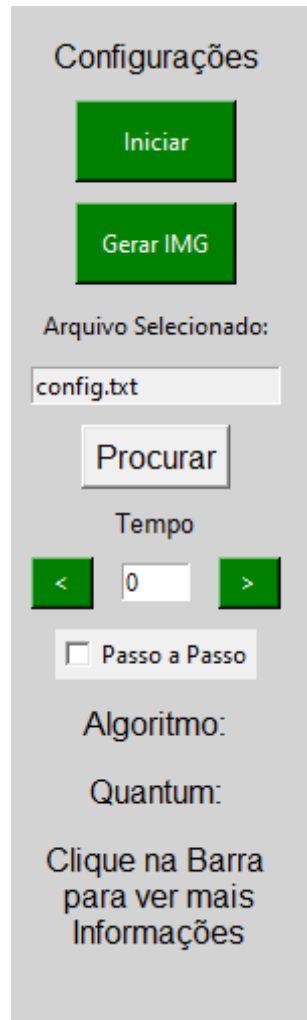
A interface do projeto possui:

- Botão para iniciar a simulação e gerar o gráfico de Gantt (caso não seja selecionado um arquivo contendo os parâmetros, ele irá utilizar valores armazenados internamente)
- Botão para gerar um arquivo .png do gráfico de Gantt
- Botão de “Procurar” para selecionar um arquivo de texto contendo parâmetros para simulação
- Botões de avançar e voltar caso seja selecionado o checkbox de Passo a Passo
- Uma tela/canvas principal onde é gerado o gráfico interativo.
- Um bloco de texto para exibir informações extras ao clicar em um dos blocos do Gráfico;

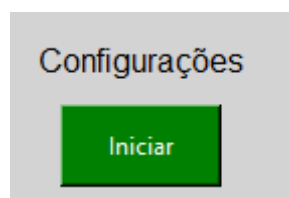


Painel de Configurações

Contém todos os botões necessários para a utilização do simulador, sendo ele criado e posicionado em [Janelas.py](#) na linha **CriarTelaConfig(self,root)** chamada por outra função também no mesmo arquivo **CriarInterface(self)**. Onde o seu painel ou frame é criado na linha 33 e travado à esquerda da janela principal na linha 34.

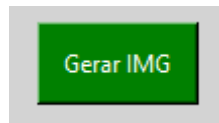


Botão Iniciar:



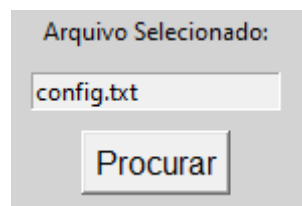
Gerado a partir de [Janelas.py](#) na função **CriarTelaConfig(self,root)** especificamente na linha 41 e posicionado no painel na linha 70 sendo a sua funcionalidade atrelada a função contida em **IniciarSimulacao(self)** em [SimulEscal.py](#) na linha 35

Botão Gerar IMG



Gerado a partir de [Janelas.py](#) na função **CriarTelaConfig(self,root)** especificamente na linha 42 e posicionado no painel na linha 71 sendo a sua funcionalidade atrelada a função contida em **SalvarDiagrama(self)** em [SimulEscal.py](#) na linha 169

Conjunto seleccionar arquivo



O texto: "Arquivo Seleccionado":

Vem da linha 43 em [Janelas.py](#) e posicionado no painel na linha 72, serve apenas para indicar para que serve os campos abaixo.

Arquivo Seleccionado:

O campo contendo "config.txt":

É um tipo tk.Entry, criado na linha 45 em [Janelas.py](#) e posicionada no painel na linha 73, serve apenas para facilitar a alteração e exibição de nomes do atual arquivo selecionado para simulação.

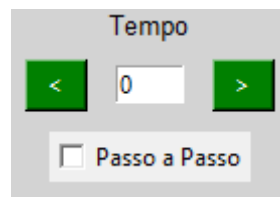
config.txt

O botão "Procurar":

É criado na linha 48 em [Janelas.py](#) e posicionado no painel na linha 74, ele está atrelado a função **PegarCaminho(self)** em [SimulEscal.py](#) na linha 77 apenas para facilitar a sua chamada que está na função **SelArquivo(self.arqAtualEntry)** em [Funcoes.py](#), que abre uma tela de busca para seleccionar o caminho do arquivo de texto.

Procurar

Conjunto Passo a Passo



O texto: “Tempo”:

Vem da linha 65 em [Janelas.py](#) e posicionada no painel linha 76, apenas serve para indicar que o campo abaixo onde está um '0' representa o tempo atual da simulação passo a passo.

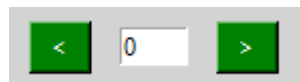


O campo contendo '0':

É um tipo tk.Entry, criado na linha 54 e posicionado no painel na linha 79 em [Janelas.py](#), da mesma forma que o campo anterior “config.txt” o seu tipo serve apenas para facilitar a troca de texto contido nele sempre que o tempo atual é alterado.



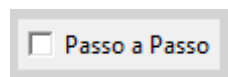
Botões voltar (<) e avançar (>)



Ambos os botões foram criado em [Janelas.py](#) nas linhas 51 e 52 e posicionados no painel nas linhas 77 e 78 e servem para alterar o tempo atual da simulação passo a passo e atualizar o gráfico de acordo com o tempo no campo ao centro.

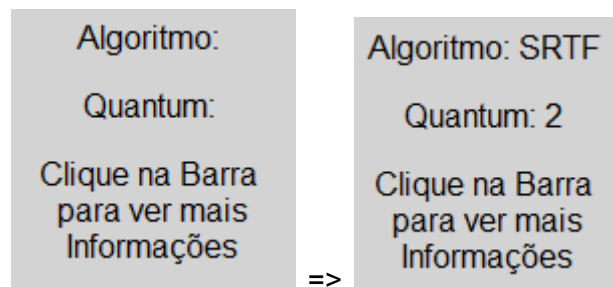
- Botão Avançar: está atrelado com a função **AvançarPasso(self)** na linha 107 em [SimulEscal.py](#) responsável criar os blocos do próximo tempo e acrescentar em 1 no tempo.
- Botão Voltar: está atrelado com a função **VoltarPasso(self)** na linha 116 em [SimulEscal.py](#) responsável por apagar os blocos do tempo atual e voltar em 1 o tempo.

Checkbox passo a passo



Criado na linha 50 em [Janelas.py](#) e posicionado no painel na linha 80, ele está atrelado a função **CheckPasso(self)** na linha 81 em [SimulEscal.py](#) alternando a variável usarPasso entre verdadeiro e falso ao mesmo tempo que ‘reinicia’ a simulação apagando os conteúdos da tela onde é desenhado o gráfico e resetando o tempo atual do passo a passo a 0.

Conjunto de informações extras



Os textos contendo “**Algoritmo:**”, “**Quantum:**” e “**Clique...**” estão contido em um tk.Label, sendo eles criados nas linhas 63, 64, 66 em [Janelas.py](#) e posicionados no painel nas linhas 82, 83 e 84. Sendo os dois primeiros atrelados uma variável **infoAlgo** e **infoQuantum** para edições simples de texto feitas na função **AtualizarInfos()** em [SimulEscal.py](#) na linha 63.

Painel Log do Bloco

Composto apenas por um título (tk.Label), bloco de texto (tk.Text) e um scroll (tk.Scrollbar), tem a função de exibir informações detalhadas de um bloco selecionado no gráfico Gantt acima dele.

Ele é criado pela função **CriarTelaInfos(root)** na linha 115 em [Janelas.py](#) e chamada dentro de **CriarInterface(self)** na linha 6 do mesmo arquivo.

O painel em si é criado nas linhas 118 e travados na parte de baixo da janela principal em 119 e ‘desbloqueado’ a configuração do seu tamanho na linha 120

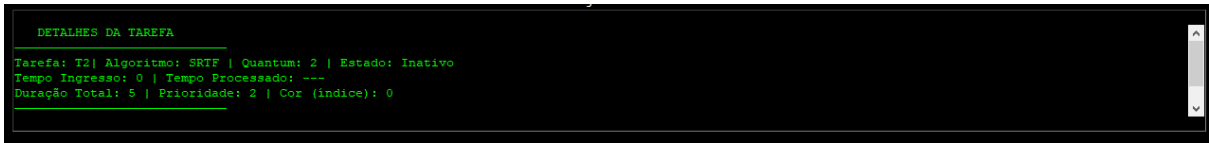


Texto: Log de Blocos

Logs do Bloco

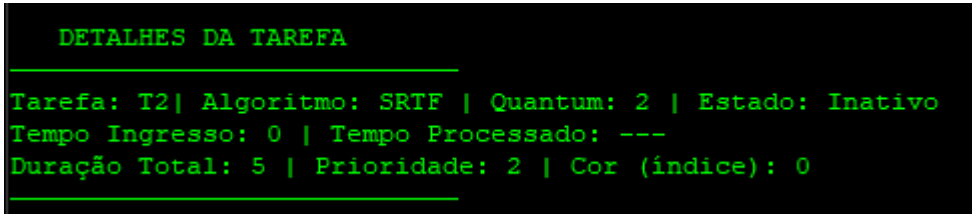
É um tipo tk.Label, contendo apenas um título para indicar para que serve a interface abaixo dele criado na linha 121 em [Janelas.py](#)

Bloco de Texto e Scrollbar



Criados em [Janelas.py](#) nas linhas 124 e 128 e travados em relação ao painel nas linhas 125 e 129, tem a função de armazenar e exibir as informações extras contidas em cada bloco do gráfico gantt.

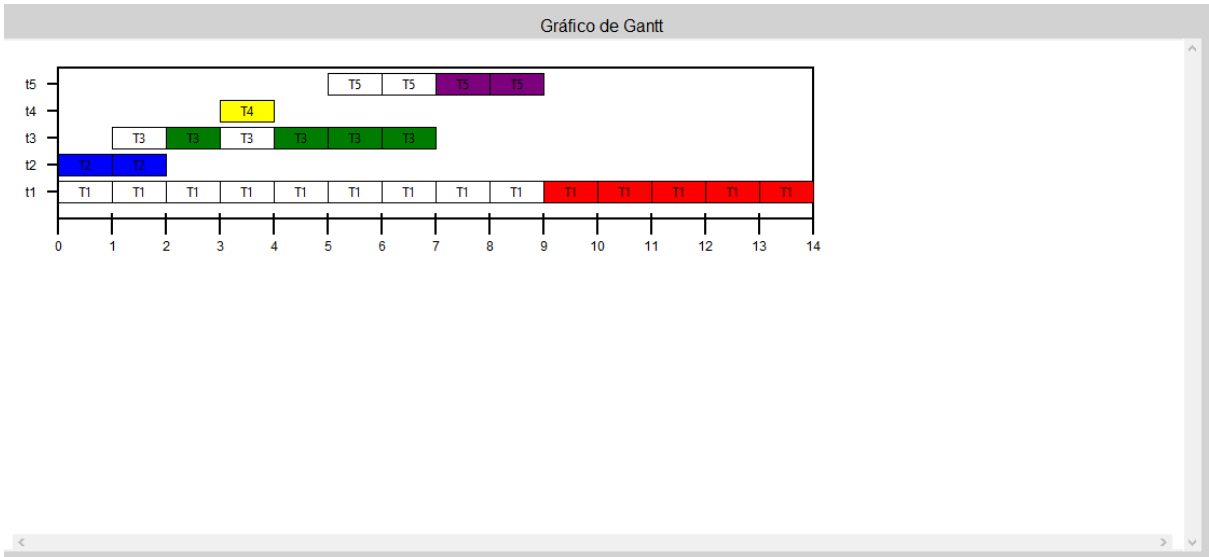
Texto Gerado



É gerado na função **PegarInfoBarra(self, instrucao)** na linha 87 em [SimulEscal.py](#) a partir da lista de instruções tratadas e geradas após a iniciar a simulação e o gráfico ser desenhado na tela.

Painel Gráfico de Gantt

Parte principal do simulador para exibir o algoritmos de escalonamento funcionando visualmente o painel é criado em [Janelas.py](#) na chamada de função **CriarTelaGantt(root)** chamada também pela função **CriarInterface(self)** na linha 6, a criação do painel está na linha 92 e travada na direta na linha 93.

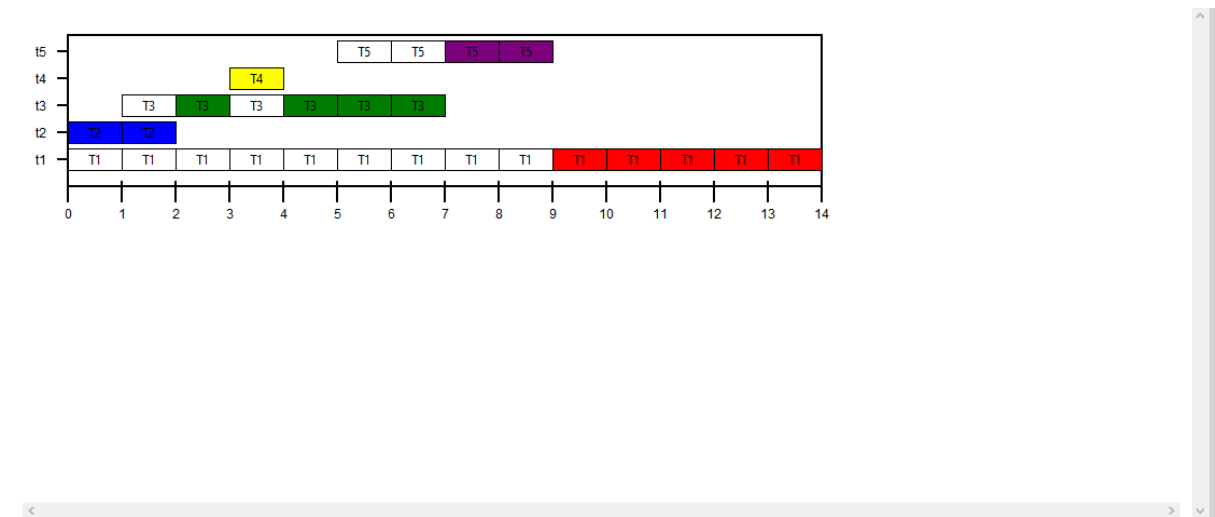


Texto: “Grafico de Gantt”

Gráfico de Gantt

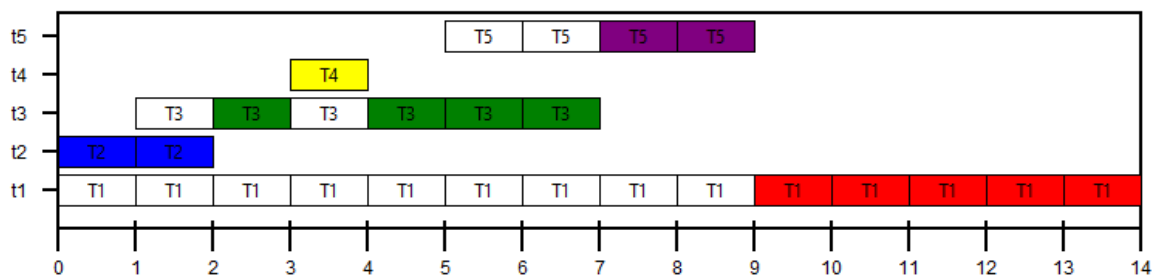
Criado a posicionado na linha 96 em [Janelas.py](#) , serve apenas para indicar que o campo abaixo em braco é reservado para exibição do gráfico de gantt.

Canvas e Scrollbar



O canvas onde é desenhado o gráfico está sendo criados na linha 99 em [Janelas.py](#) e preenchida dentro do próprio painel o máximo possível, enquanto os scrolls vertical e horizontal são criados em 103 e 107 e travados nos extremos do painel nas linhas 104 e 108 no mesmo script, servindo para mover o gráfico para caso a janela esteja muito pequena ou que o gráfico esteja muito grande. O retorno do objeto deste canvas pela função é importante pois com a referência dele é possível começar todo o desenho do gráfico.

Gráfico de Gantt



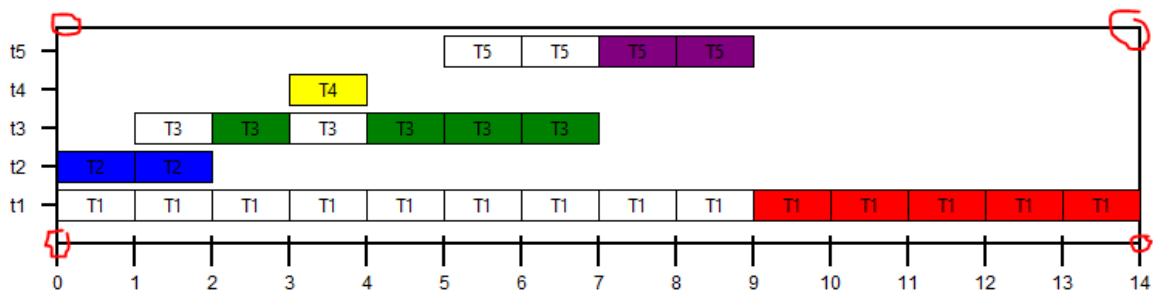
O fluxo da lógica por trás da criação do gráfico se da seguinte forma:

1. Inicia na chamada da função **IniciarSimulacao(self)** em [SimulEscal.py](#) nele é criado uma lista de 'instruções' gerada por um dos algoritmos selecionados anteriormente na interface do "Painel de Configurações" a partir de um arquivo de texto selecionado
2. Chamada da função **ProcessarDados(texto)** na linha 44 em [SimulEscal.py](#) que pertence a [Funcao.py](#) na linha 100 responsável por fazer o tratamento do texto dentro do arquivo selecionado ou é utilizado a função **UsarConfigPadrão()**

pertencente ao [SimulEscal.py](#) na linha 47 caso não tenha sido selecionado nenhum arquivo.

- Envia os parâmetros extraídos do arquivo de texto para a função **GerarInstrucoes(algoritmo, tarefas, quantum)** na linha 21 pertencentes a [Funcoes.py](#) onde é gerada uma lista de instruções dependendo do algoritmo contido no texto extraído.
- As instruções são geradas de acordo com o algoritmo selecionado que são chamados na linha 34 em [Funcoes.py](#) com suas funções presentes em [Algoritmos.py](#). Após geradas as instruções “Brutas”, é feito um tratamento para adicionar em uma nova lista de instruções detalhes extras para cada ‘bloco’ que será desenhado no gráfico.
- Ao final de **GerarInstrucoes(algoritmo, tarefas, quantum)** é retornado a lista de instruções finalizadas que é entregue para a função **GerarGrafico(canvasGantt, config, instrucoes, instrucoesInativas, maxTempo, maxTid, usarPasso, PegarInfoBarra=None)** presentes em [Grafico.py](#) na linha 7, onde é criado todo o desenho do gráfico de Gantt a partir das seguintes funções

- CriarBordas(canvasGantt, config, maxTempo, maxTid)** linha 10: cria um retângulo para englobar todos os elementos do gráfico:



- CriarEixos(canvasGantt, config, maxTempo, maxTid)** linha 11: cria as faixas contendo o tempo e a tarefa



- CriarBarra(canvasGantt, config, instrucao, maxTid, PegarInfoBarra)** linhas 15 e 18: Cria as barras contidas no gráfico, tanto para as barras para representar as inativas (em branco) quanto as ativas com cores distintos.

