

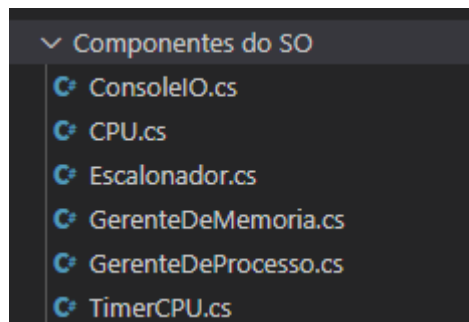
Link do GitHub: <https://github.com/fmbordignonn/MaquinaVirtual>

Integrantes do grupo: Felipe Bordignon, Fernando Elger, Dimitri Finger, Carlos Henrique e Frederico Thofehr

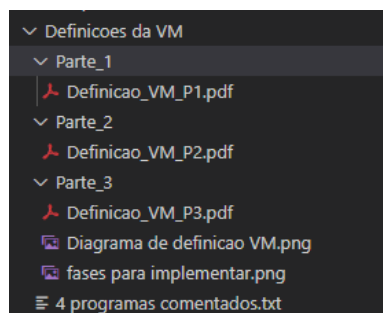
Nessa terceira parte do projeto da máquina virtual, foi implementado:

- conceitos de *threads* no Shell, Escalonador, CPU e no Console de IO;
- fila de bloqueados por IO, fila de pedidos para o Console, a fila de prontos foi reaproveitada e criamos uma fila de programas finalizados para melhor visualização do sistema;
- rotinas de tratamento de acesso indevido, de divisão por zero, finalização (quando o programa atinge o STOP) e tratamento do timer, que no nosso caso foi utilizado linhas de código como *time-slice*;
- rotina de tratamento de IO e tratamento de retorno de IO;
- *exception* de acesso indevido e divisão por zero;
- comando para dar *shutdown* na máquina virtual desligando todas as *threads*;

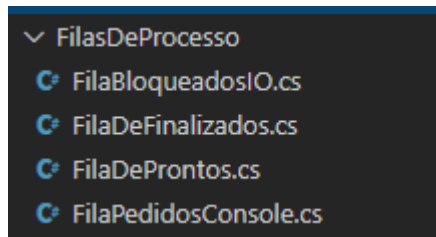
Explicação da estrutura da implementação:



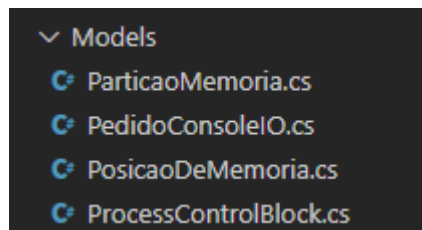
Componentes principais que constroem o sistema operacional



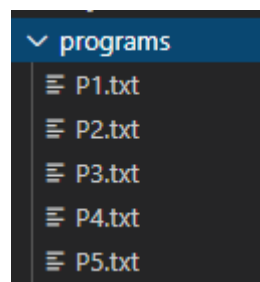
Organização dos artefatos solicitados para cada entrega da máquina virtual



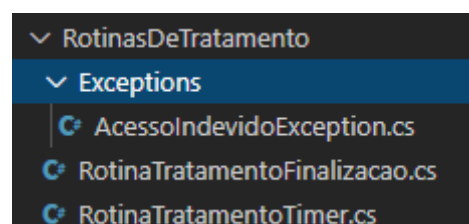
Filas que armazenam processos em andamento, bloqueados, finalizados e pedidos ao console de IO



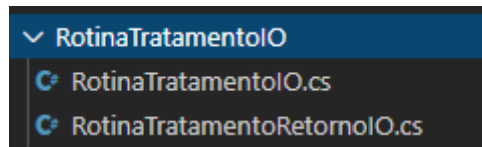
Contém os componentes básicos da máquina virtual



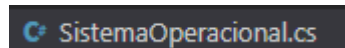
Contém os 4 programas solicitados na entrega 1 e um quinto programa extra para testar acesso indevido de memória



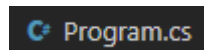
Contém as rotinas de finalização de processos, e uma exceção customizada para acesso indevido de memória



Contém as rotinas para tratamento de processos que envolvem IO



Contém o "coração" do sistema operacional, inclui o Shell que requisita os comandos do usuário e inicializa as *threads* básicas do sistema (Shell, escalonador e ConsoleIO - que são loops eternos)



Contém o método "*Main*", que solicita o número de partições na memória desejadas pelo usuário para então dar início a execução do sistema operacional

As limitações da nossa aplicação se resumem em:

- A máquina virtual é executada apenas utilizando um console, o que torna alguns *logs* da aplicação assíncronos. Foi feito no menu do Shell um acesso para o Console de IO, para que o usuário possa inserir e ler dados, o que faz com que o mesmo deva necessariamente acessar o Console para concluir o IO e assim desbloquear o processo.

- O escalonamento de processos não foi feito por tempo, mas sim por quantidade de linhas de comando executados, pois priorizamos melhorar o funcionamento da máquina virtual em detrimento de um conceito que já estava funcionando.

Passo a passo para a execução:

- instalar o .NET Core 2.1 ---> <https://dotnet.microsoft.com/download/dotnet-core/2.1>;
- instalar a extensão 'C#' no VSCode;
- use o comando 'dotnet run' para rodar.