

**Sistemas Operativos I**

**Job Scheduler**

****

**Elaborado por:**

João Calhau nº 31621

José Pimenta nº 31677

Indice

[Introdução 3](#_Toc421636949)

[Processo de implementação 4](#_Toc421636950)

[Explicação resumida 4](#_Toc421636951)

[Código comentado 5](#_Toc421636952)

[Escalonador 5](#_Toc421636953)

[PCB 9](#_Toc421636954)

[Process 10](#_Toc421636955)

[Process\_Table 11](#_Toc421636956)

[Teste 12](#_Toc421636957)

# Introdução

O trabalho de Sistemas Operativos I tem como objectivo simular um job scheduler.

Iremos assim implementar o dito job scheduler de modo a podermos realizar operações , incluindo a criação de processos (estado *NEW*), a preparação destes (estado *READY*), a execução (estado *RUN*), bloquear processos (estado *BLOCKED*), e a finalização dos mesmos (estado *EXIT*).

Para isso iremos implementar, através dos conhecimentos adquiridos na disiciplina de SO1, o dito job scheduler em que os processos, constituidos por instruções tais como *CPU, DISK, NOP, GOTOBEGIN, FORK, EXIT ,* irão através do job scheduler, alternar entre si e no final acabarem a sua execução, ou através de a instrução GOTOBEGIN, continuar indefinidamente.

# Processo de implementação

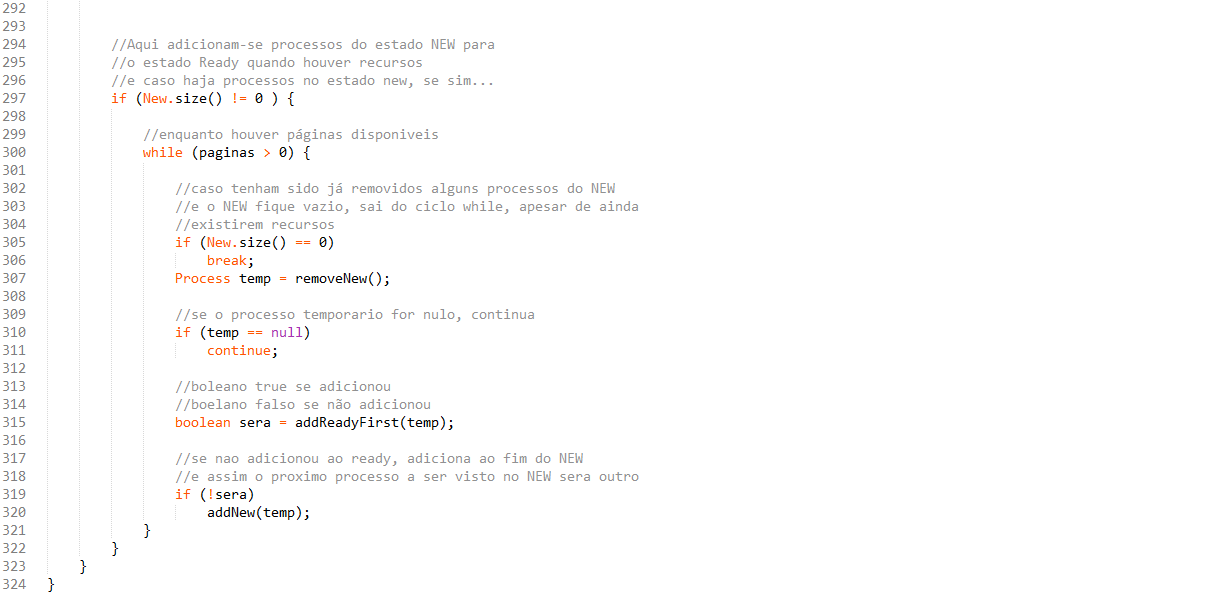
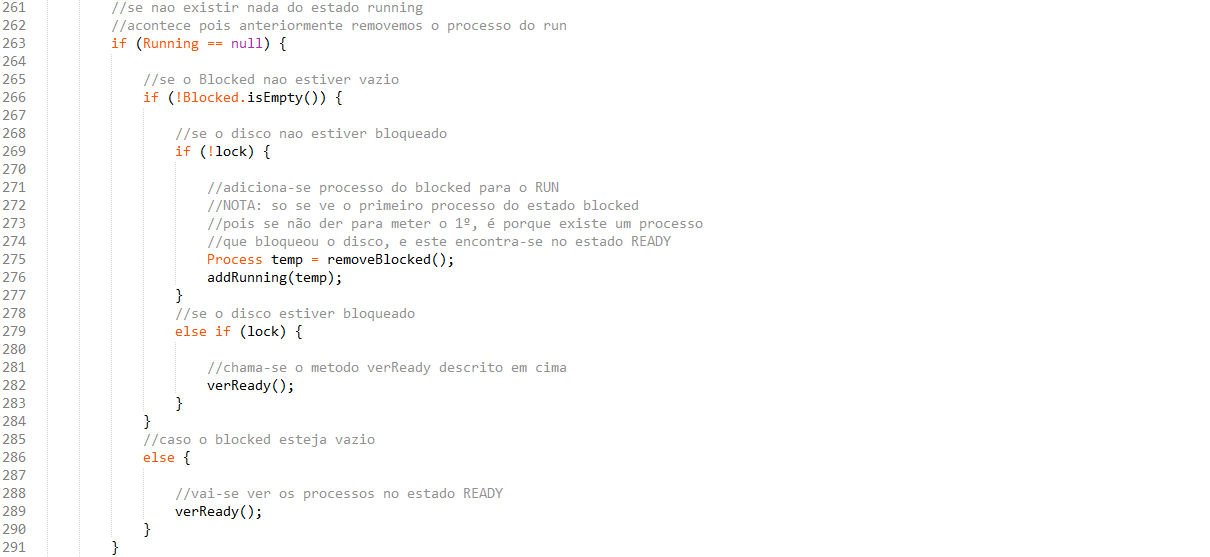
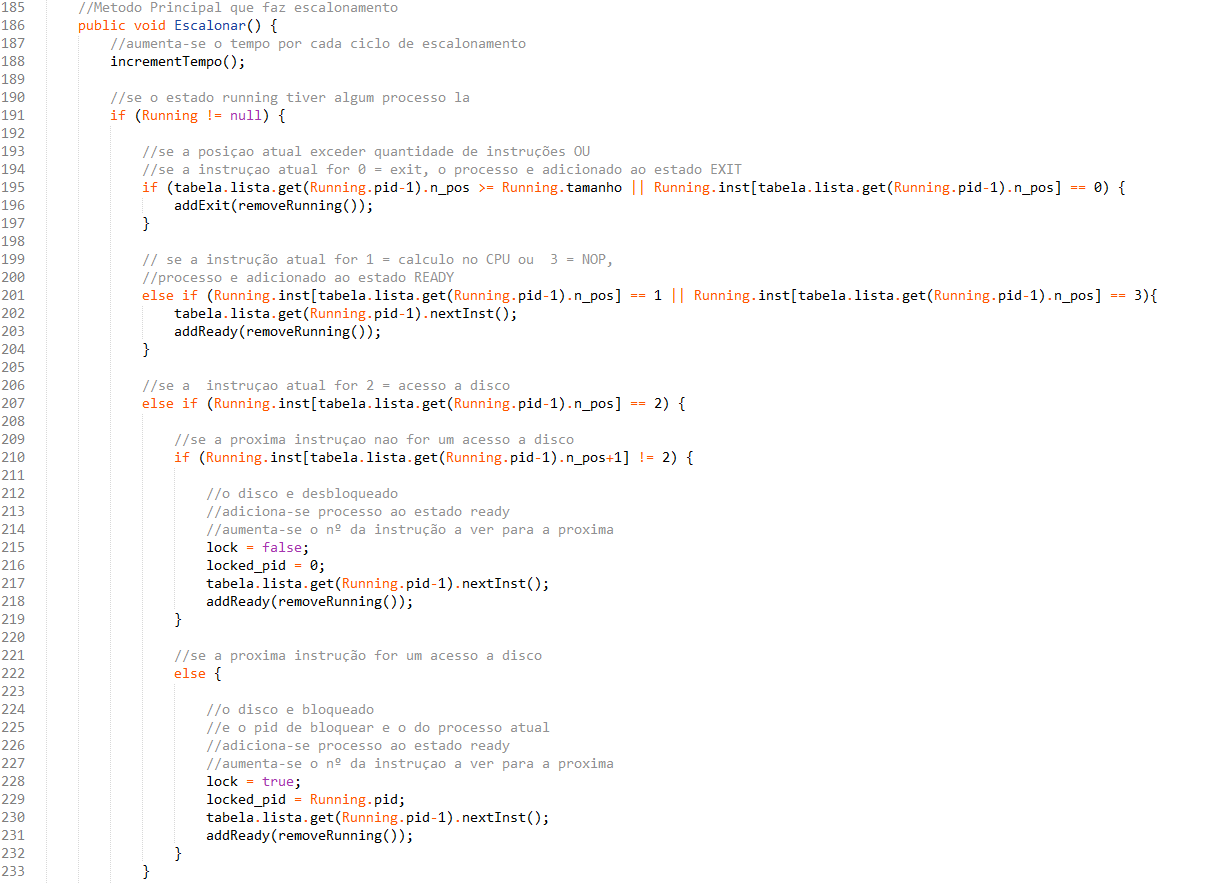
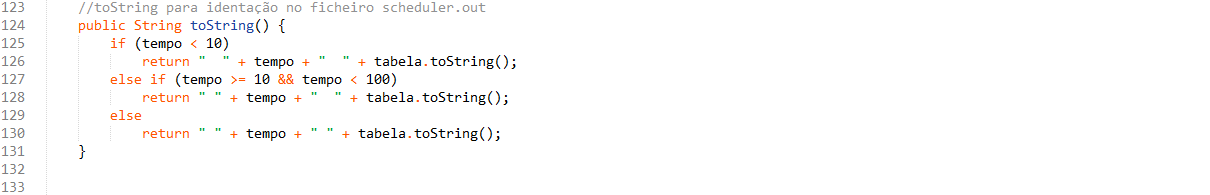
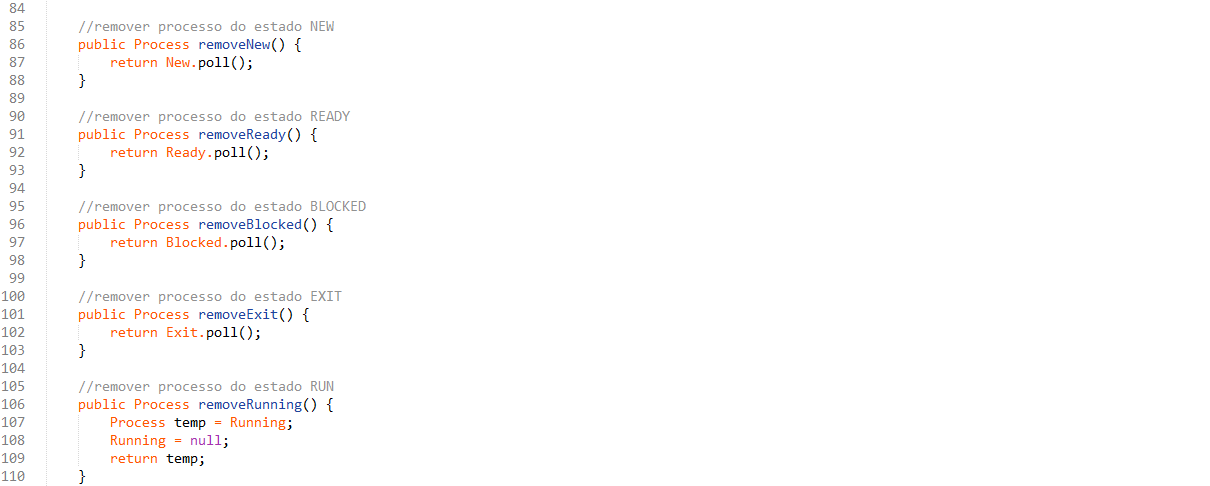
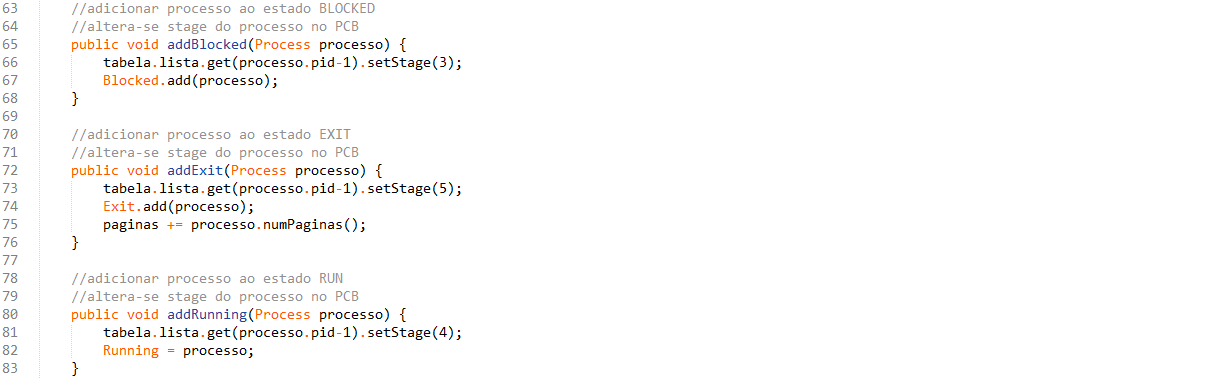
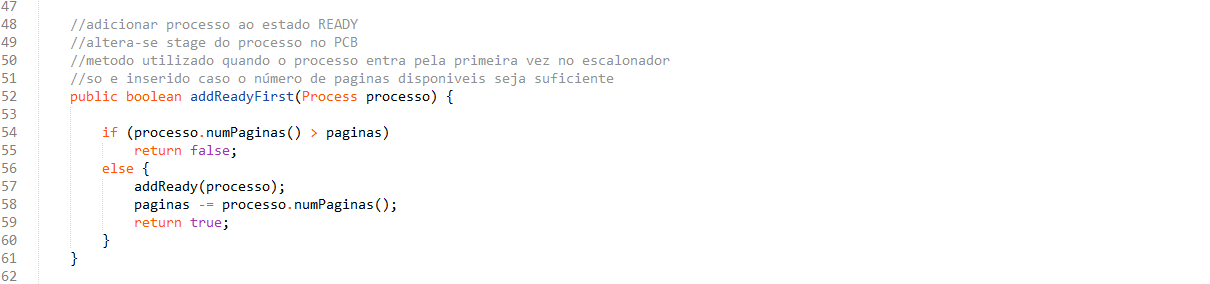
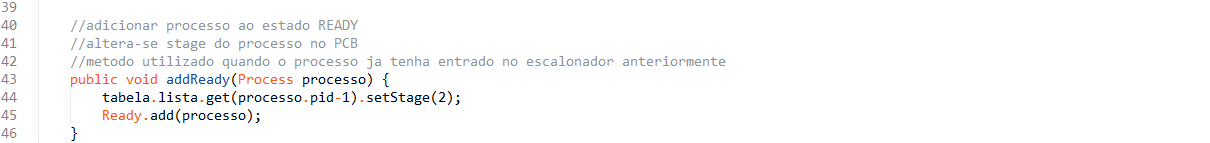
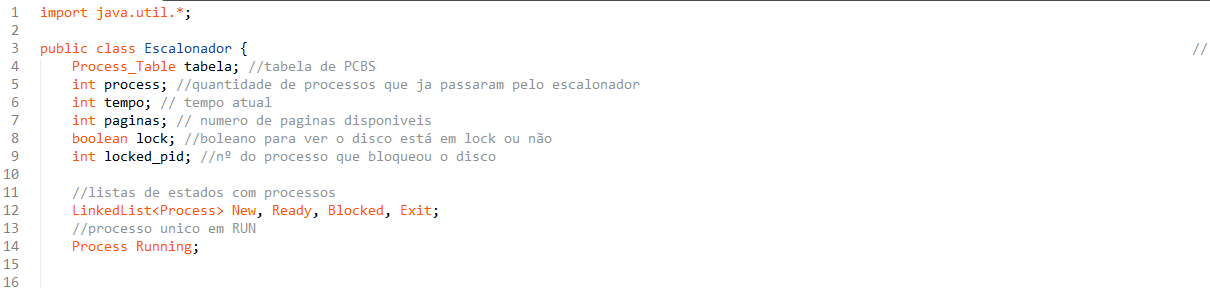
## Explicação resumida

O programa inicia no main (no nosso caso a main está na classe Teste). Neste ficheiro criamos 2 threads, scheduler e addProcess. A thread scheduler faz exactamente o que o nome diz, vai organizando os processos a cada iteração e vai guardando no ficheiro os estados em que cada processo se encontra. A thread addProcess fica à espera de input do utilizador e adiciona para a lista new do escalonador.

Na classe Escalonador, primeiramente vamos verificar se existe algum processo em estado running, caso afirmativo, vamos ver qual é a instrução que este executa e é retirado desse estado. Em caso negativo, vamos à lista dos processos bloqueados e ver se o disco ainda se encontra bloqueado, se se encontrar bloqueado então vamos ver ao ready se podemos meter algum desses processos para running, se não se encontrar bloqueado adicionamos para o estado running um do estado blocked. Se não houver nenhum processo em estado blocked, vamos directamente ao estado ready. Depois disto vamos ainda ver ao estado new se existem processos que podem ser transferidos para o estado ready (só podem ser transferidos e estado se o numero de páginas disponível for superior ao requerido pelo processo).

## Código comentado

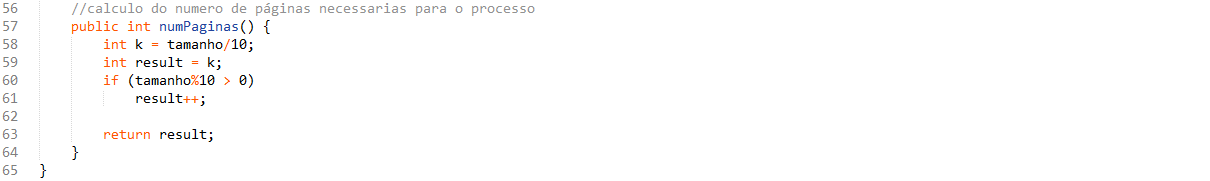
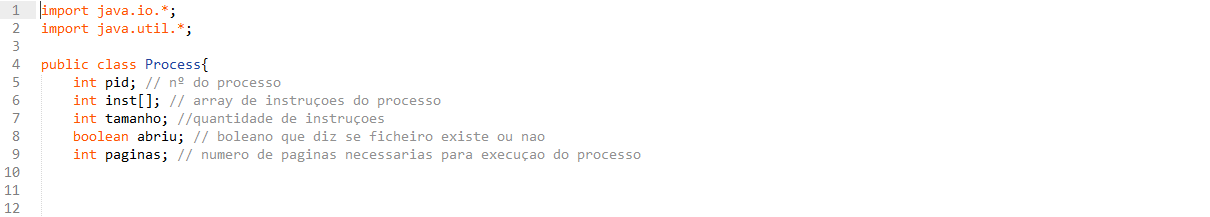
### Escalonador



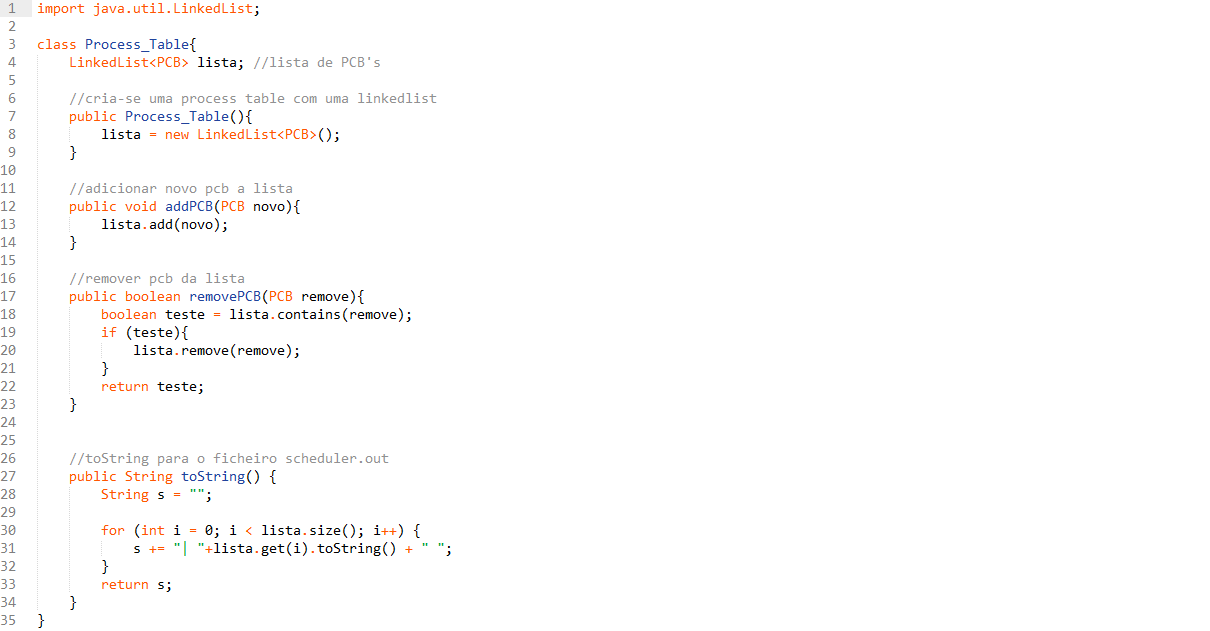
### PCB



### Process



### Process\_Table



### Teste

