Simulação e Computação Científica

**Relatório**

**Estudo do funcionamento dum sector duma fábrica**

Francisco Bento, nª2015232998

Inês Silva, nº2015251488

Luís Bernarda, nº2015237627

ÍNDICE

[Introdução 3](#_Toc481406678)

[Arquitetura do simulador 3](#_Toc481406679)

[Simulador 3](#_Toc481406680)

[Serviço 4](#_Toc481406681)

[Evento 4](#_Toc481406682)

[Lista 4](#_Toc481406683)

[Peça 4](#_Toc481406684)

[Registrador 4](#_Toc481406685)

[Validação do Simulador 5](#_Toc481406686)

[Validação externa 5](#_Toc481406687)

[Validação interna 6](#_Toc481406688)

[Validação por análise de parâmetros 7](#_Toc481406689)

[Variação da média do Polimento A 7](#_Toc481406690)

[Variação da média da Perfuração A 9](#_Toc481406691)

[Variação da média da Perfuração B 11](#_Toc481406692)

[Variação da média da Polimento B 13](#_Toc481406693)

[Variação da média de Envernizamento 15](#_Toc481406694)

[Variação da média do tempo de chegada das peças A 17](#_Toc481406695)

[Variação da média do tempo de chegada das peças B 19](#_Toc481406696)

[Alteração do número de máquinas do polimento B 21](#_Toc481406697)

[Alteração do número de máquinas de perfuração A 23](#_Toc481406698)

[Alteração do número de máquinas de polimento A 24](#_Toc481406699)

[Alteração do número de máquinas de perfuração B 25](#_Toc481406700)

[Alteração do número de máquinas de envernizamento 26](#_Toc481406701)

[Alínea c) 28](#_Toc481406702)

[i) 28](#_Toc481406703)

[ii) 29](#_Toc481406704)

[Alínea d) 29](#_Toc481406705)

[Conclusão 30](#_Toc481406706)

[Anexo 1 – Classe Simulador 30](#_Toc481406707)

[Anexo 2 – Classe Peça 35](#_Toc481406708)

[Anexo 3 – Serviço 36](#_Toc481406709)

[Anexo 4 – Classe Lista 38](#_Toc481406710)

[Anexo 5 – Classe Evento 39](#_Toc481406711)

[Anexo 6 – Classe Aleatório 40](#_Toc481406712)

[Anexo 7 – Classe Registrador 40](#_Toc481406713)

[Anexo 8 – Simulação com os valores iniciais no GPSS 41](#_Toc481406714)

[Anexo 9 – Simulação do cenário i) 42](#_Toc481406715)

[Anexo 9 – Simulação do cenário ii) 43](#_Toc481406716)

# Introdução

Este projeto foi desenvolvido em Python e trata-se de uma simulação de um sector de uma fábrica.

Neste relatório vamos explicar como validamos o nosso simulador, como é pedido na alínea b), vamos analisar o funcionamento do simulador, detetar as falhas deste e discutir as possíveis situações de melhoramento referidas na alínea c) – “Aumentar o número de máquinas nas secções críticas” ou “substituí-las por 2 máquinas mais rápidas (média de processamento de 1.7 minutos), mas mais caras do que a solução anterior.”.

# Arquitetura do simulador

**CLASSES**

## Simulador

Esta é a classe principal do nosso trabalho. Gere os eventos que vão acontecer assim como os serviços. Reúne as estatísticas de todos os serviços e escreve-as para um ficheiro.

Os seus atributos são:

* **nome** – nome do simulador
* **numero\_pecas** - número de peças diferentes que passam pelas secções
* **media\_chegada\_pecas** – é uma lista que guarda a média das distribuições de chegada das peças
* **tempo** – inteiro que funciona como relógio do simulador, sendo sempre inicializado a 0
* **horas** – guarda as horas de produção por dia
* **dias** – guarda o número de dias da simulação
* **matriz\_serviços** – corresponde a uma lista de listas em que contém todas as peças que passaram naquele sector e a cada uma dessas peças está associada uma lista de serviços que a afetam
* **lista** – é lista de eventos que guarda todos os eventos da simulação, existe apenas uma por simulador
* **tipo\_pecas** – é uma lista com os diferentes tipos de peças
* **divida** – inteiro que guarda a dívida da fábrica (variável usada apenas na alínea e) )
* **pausa** – boolean que permite percorrer o simulador dia-a-dia caso pretendermos analisar as estatísticas ao fim de cada dia e não no fim dos dias todos de simulação
* **debug** – boolean que permite decidir se queremos imprimir a sequência de eventos que decorreram
* **registrar** – boolean que permite escolher se pretendemos guardar as estatísticas num ficheiro

No início é criada uma instância do simulador e este é inicializado, logo de seguida é chamada a função executa.

A função executa começa por criar o evento de chegada para cada peça, começa por zerar o relógio e enquanto não forem completadas as horas de trabalho o simulador retira um o primeiro evento da lista, atualiza o relógio para o instante desse evento, atualiza as estatísticas e executa evento. Este processo repete-se até que o relógio do simulador atinja as horas de trabalho totais do dia.

Seguidamente são impressas as estatísticas para aquele dia e o número de dias que foram executados aumenta. No fim é sempre verificado se ainda há dias por executar.

Assim que o número total de dias a executar é atingido, o simulador termina.

## Serviço

Esta classe representa uma secção com a sua fila de espera associada e com um número de máquinas específico para servir as peças.

É aqui que são feitos todos os cálculos estatísticos e armazenados todos os valores necessários para esses mesmos cálculos.

Os principais métodos são o: inserePeca (coloca a peça na máquina(s) do serviço ou na fila de espera caso a máquina(s) esteja ocupada) e retiraPeca (retira a peça do serviço e caso haja alguma peça na fila de espera encaminha-a para a máquina livre).

## Evento

É desta classe que são derivados todos os eventos. Nos seus atributos encontra-se o simulador, o serviço, a peça e o instante em que o evento ocorre.

As duas funções definidas nesta classe são a função Chegada() que representa a chegada de uma peça a um serviço e a função Saida() que representa a saída da peça do serviço.

## Lista

Esta classe contém os eventos do simulador guardados numa lista que está constantemente atualizada. As funções aqui incluídas são insere\_evento() que ao inserir o evento reordena a lista de modo a que no início da lista estejam sempre os eventos que vão ocorrer primeiro na simulação, a função retira\_evento() que retira sempre o primeiro evento da lista e por fim uma função que nos imprime a lista.

## Peça

Esta classe representa a peça e apresenta como atributos: o tipo (inteiro que representa cada uma das peças que entra no sector), o nome (se se trata da peça A ou peça B) e o custo da peça.

## Registrador

Esta classe permite-nos escrever as estatísticas para um ficheiro.

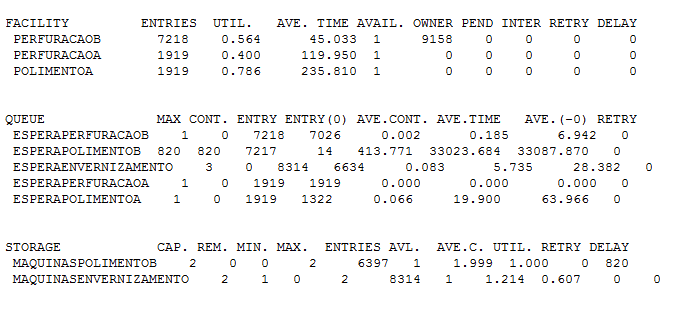
Encontra a pasta onde o programa está a correr e cria uma pasta nesse diretório. Temos a função comeca\_registo(nome) que recebe o nome que quisermos dar ao ficheiro, caso não seja dado nome ele atribui a data e hora com que foi criado o ficheiro como nome, cria-o e escreve um texto de introdução. A função regista() permite-nos adicionar informação ao ficheiro. Por fim temos uma função que serve apenas para fechar o ficheiro.

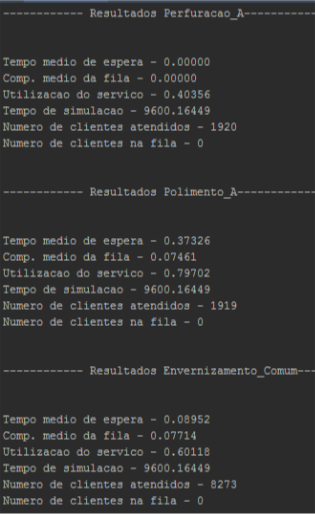
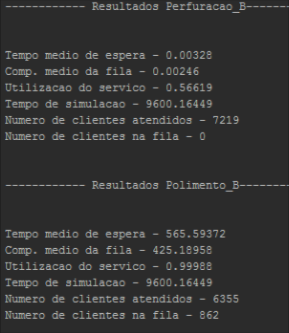
# Validação do Simulador

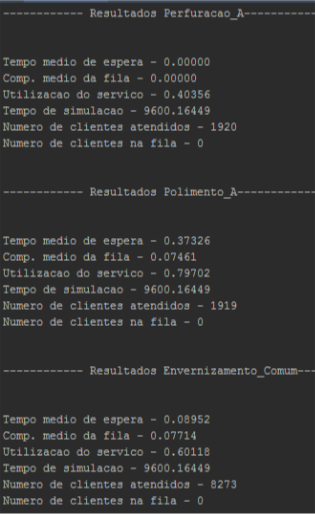
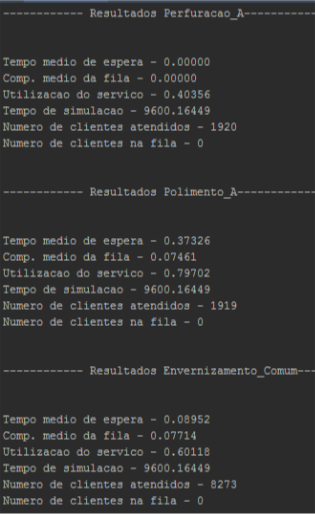
## Validação externa

Para esta validação, utilizamos o GPSS e usando os mesmos valores tanto no nosso simulador como no GPSS comparamos os resultados. Repetimos este processo várias vezes. Ao comparar temos resultados muito semelhantes de ambos os lados, o que nos permite concluir que o nosso simulador está a fazer cálculos corretos.

Abaixo vamos incluir as imagens que nos permitem fazer a comparação.









Há apenas uma ligeira diferença no número de entrada das peças no envernizamento comum, isto deve-se ao facto de mais peças ficarem retidas na fila de espera do polimento B.

## Validação interna

**Estabilidade**

Gráfico que representa a variação das taxas de utilização de cada serviço

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Perfuração A | Polimento A | Perfuração B | Polimento B | Envernizamento |  |
| 1h | 0.40327 | 0.68157 | 0.58555 | 0.98075 | 0.58606 |  |
| 2h | 0.40058 | 0.68074 | 0.5761 | 0.99037 | 0.58911 |  |
| 4h | 0.41426 | 0.75837 | 0.5588 | 0.99516 | 0.58916 |  |
| 8h | 0.40811 | 0.7762 | 0.57618 | 0.99758 | 0.58834 |  |
| 12h | 0.4137 | 0.79753 | 0.56413 | 0.99839 | 0.59545 |  |
| 16h | 0.4066 | 0.80371 | 0.56605 | 0.99879 | 0.59861 |  |
| 20h | 0.39964 | 0.81428 | 0.56805 | 0.99903 | 0.60193 |  |
| 24h | 0.40046 | 0.804 | 0.56509 | 0.99919 | 0.60539 |  |
| 28h | 0.3981 | 0.80186 | 0.56648 | 0.99931 | 0.60383 |  |
| 32h | 0.39792 | 0.79744 | 0.56892 | 0.9994 | 0.60328 |  |

Valores utilizados para o gráfico gerado acima

Através do gráfico que obtivemos, facilmente se conclui que a partir das 12h o nosso sistema atinge uma certa estabilidade. Esta informação é bastante útil quando se quer, por exemplo, correr várias vezes o simulador. Assim sabemos qual o menor tempo de simulação a usar para obtermos valores mais estáveis.

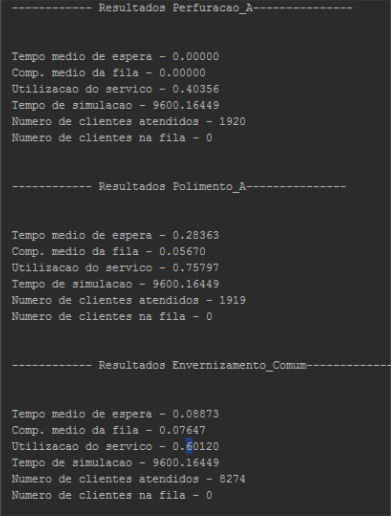
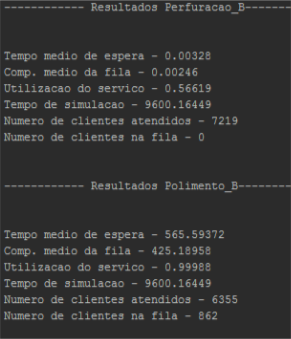
## Validação por análise de parâmetros

Vamos agora fazer variar os diferentes parâmetros (médias, tempo de chegada, tempo de paragem e número de máquinas), aumentando e diminuindo 5% em relação aos seus valores originais e vamos verificar os efeitos destas alterações no nosso sistema, para ter a certeza que coincidem com os valores teóricos esperados.

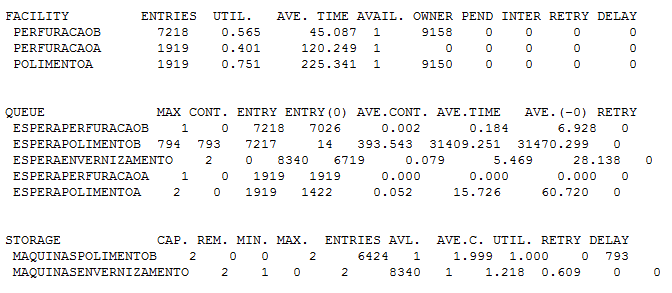
Estamos a validar também externamente cada uma das alterações feitas com o GPSS.

### Variação da média do Polimento A

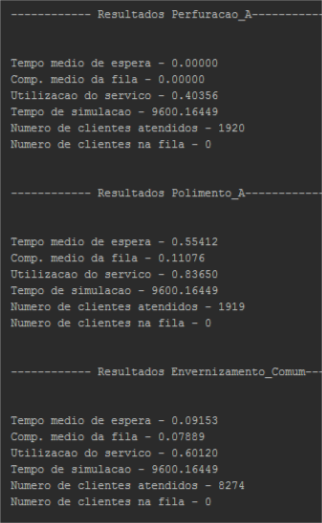
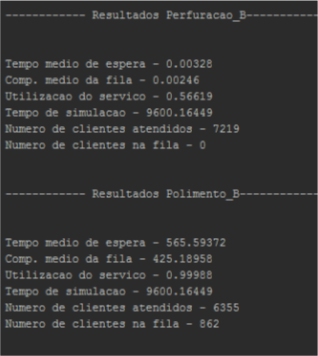
Originalmente tínhamos uma média de 4 e diminuímos para 3.8, a taxa de utilização deste serviço diminui, uma vez que cada peça leva menos tempo a ser processada.



Média do polimento A = 3.8

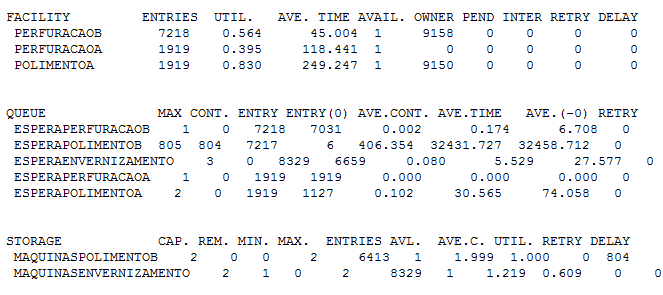


Média do polimento A = 3.8 no GPSS



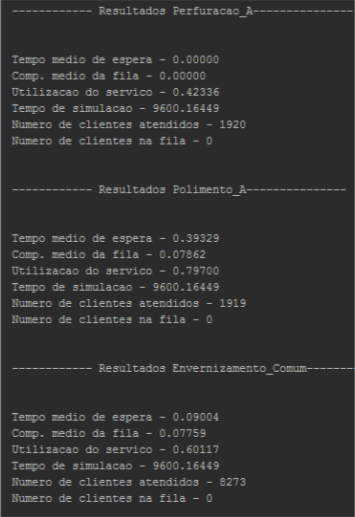
Agora como seria de esperar, a taxa de utilização que normalmente ronda os 80%, aumentou ligeiramente para 83%.

Média do polimento A = 4.2



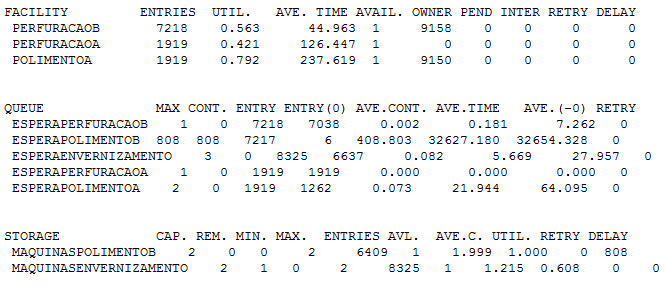
Média do polimento A = 4.2 no GPSS

### Variação da média da Perfuração A

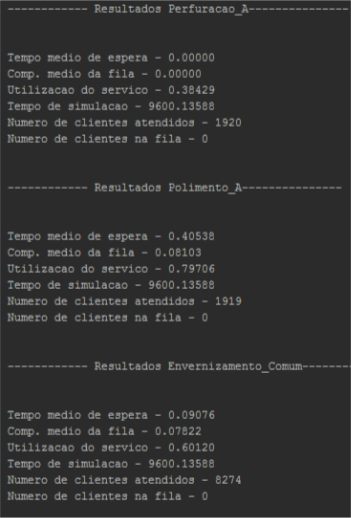


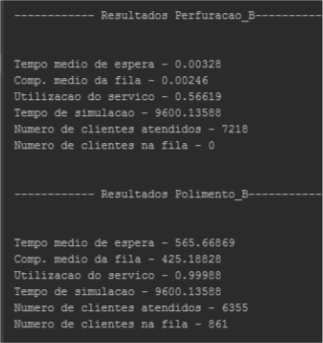
Originalmente tínhamos uma média de 2, ao aumentar para 2.1, a taxa de utilização da perfuração também aumenta de 40% para 42%.

Média da Perfuração A = 2.1



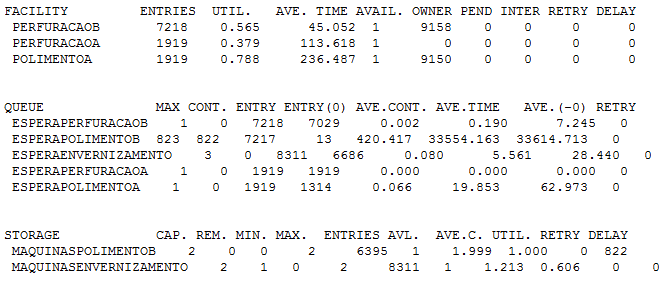
Média da Perfuração A = 2.1 no GPSS





Verificou-se que ao diminuir a média a taxa de utilização diminui de 40% para 38%.

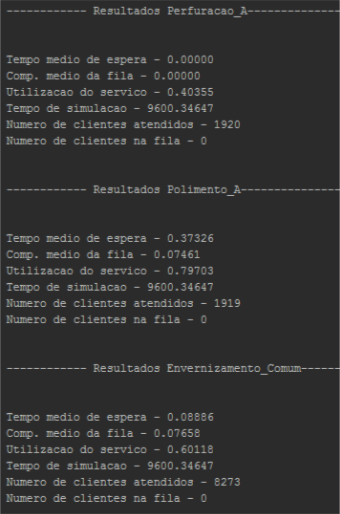
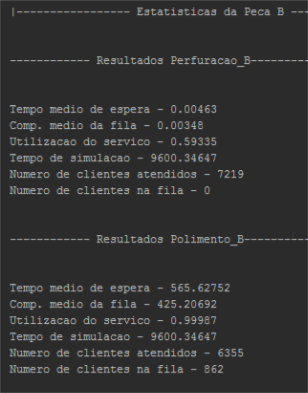
Média da Perfuração A = 1.9



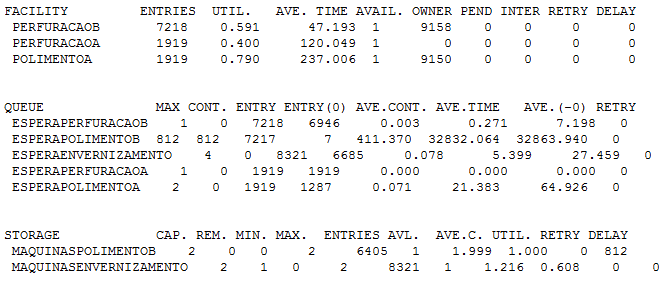
Média da Perfuração A = 1.9 no GPSS

### Variação da média da Perfuração B

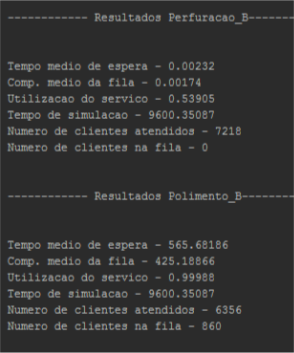
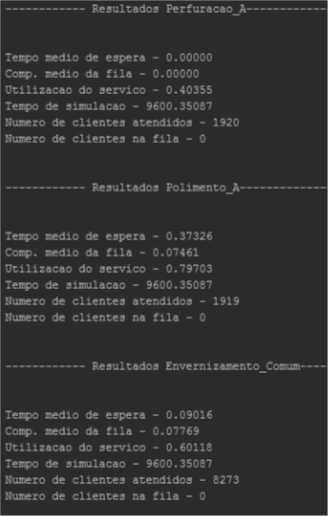
Aumentando a média do serviço de Perfuração das peças B, conseguimos notar que esta fica com uma taxa de utilização maior, pois demora um pouco mais a tratar delas.



Média da Perfuração B = 0.7875

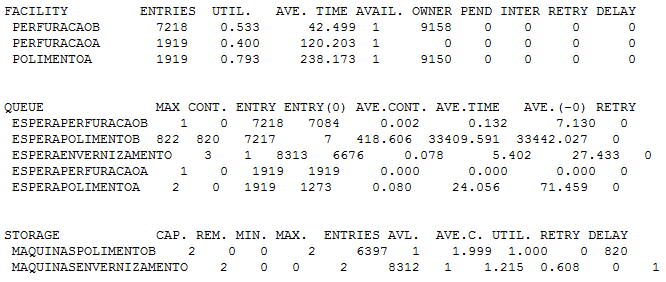


Média da Perfuração B = 0.7875 no GPSS



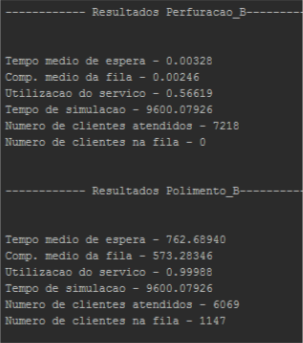
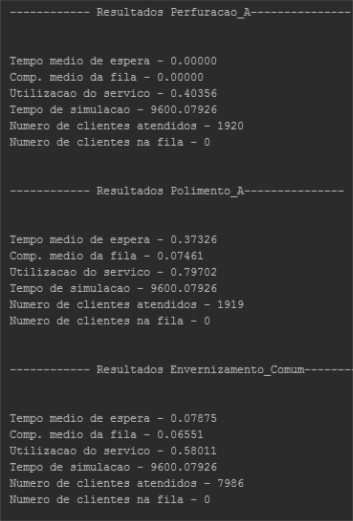
Diminuindo a média do serviço de Perfuração das peças B, como é esperado a taxa de utilização diminuiu e ronda agora 53%.

Média da Perfuração B = 0.7125



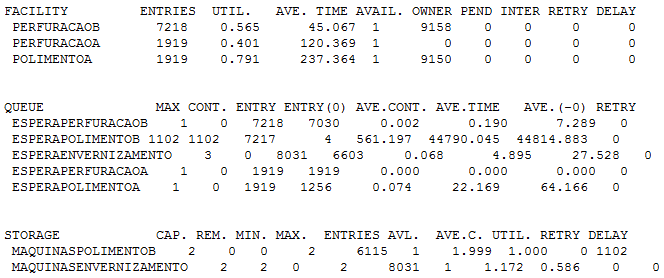
Média da Perfuração B = 0.7125 no GPSS

### Variação da média da Polimento B

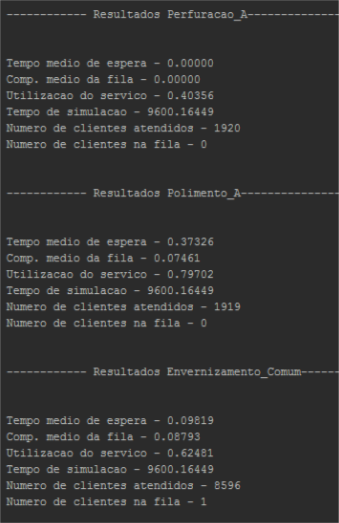
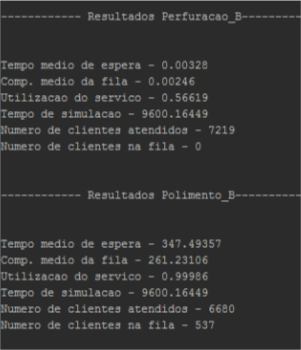


Ao variar a média do Polimento B os efeitos não são muito significativos, este serviço mantêm a sua taxa de utilização quase no limite porque tem sempre peças na fila de esperar para processar.

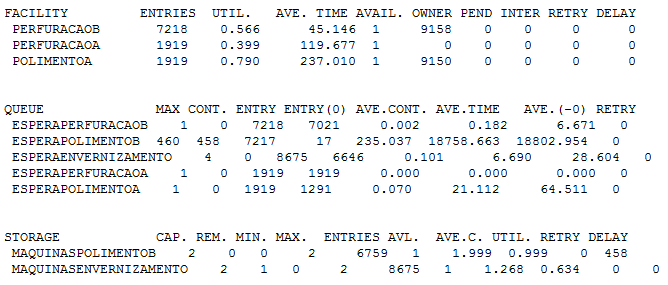
Média do Polimento B = 3.15



Média do Polimento B = 3.15 no GPSS

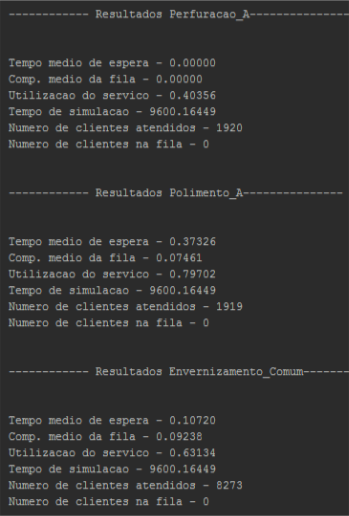
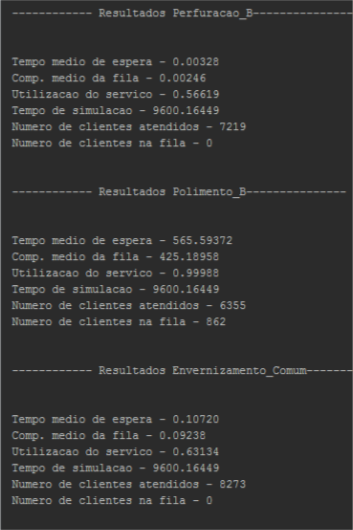


Média do Polimento B = 2.85



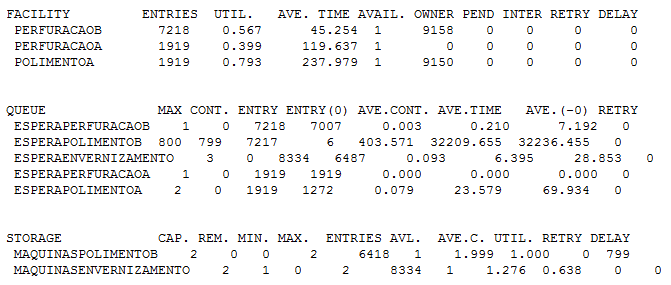
Média do Polimento B = 2.85 no GPSS

### Variação da média de Envernizamento



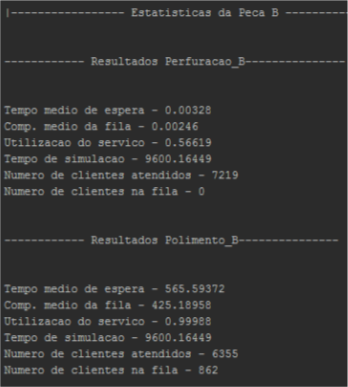
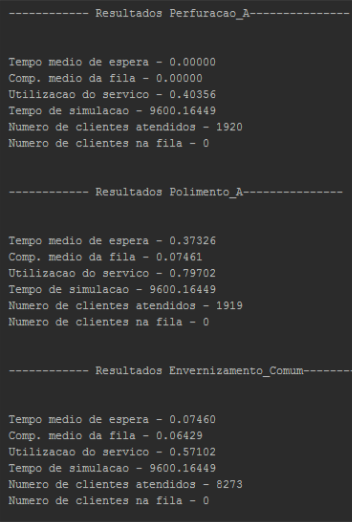
Inicialmente tínhamos uma média de 1.4 minutos no envernizamento. Ao aumentar para 1.47 a taxa de utilização aumenta ligeiramente, uma vez que a máquina leva mais tempo a processar cada peça.

Média de Envernizamento = 1.47

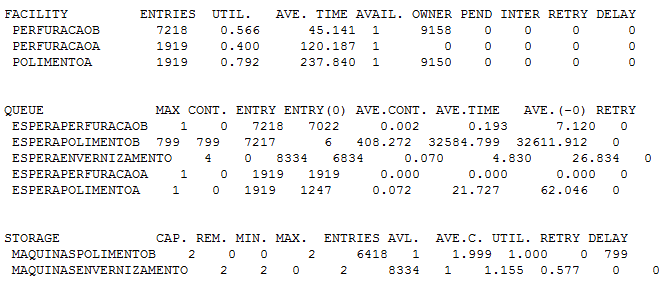


Média de Envernizamento = 1.47 no GPSS

Ao diminuir o tempo para 1.33, a taxa de utilização diminui, como seria de esperar, uma vez que cada peça demora menos tempo no envernizamento.

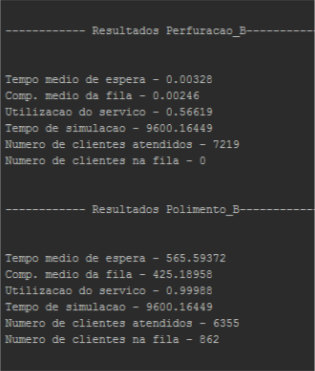
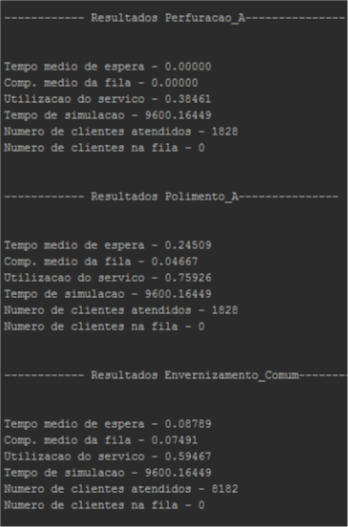


Média de Envernizamento = 1.33



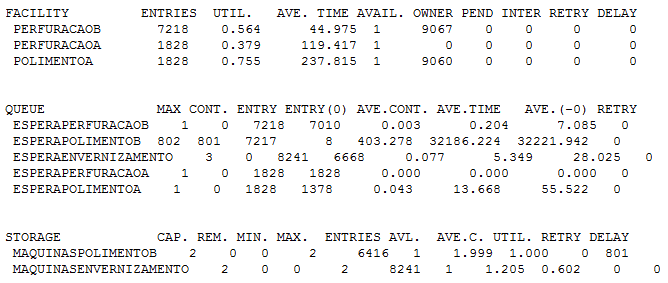
Média de Envernizamento = 1.33 no GPSS

### Variação da média do tempo de chegada das peças A

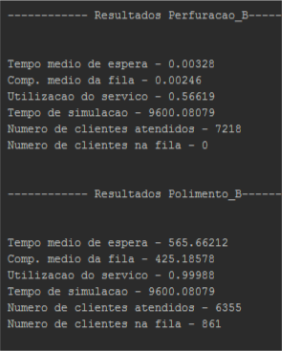
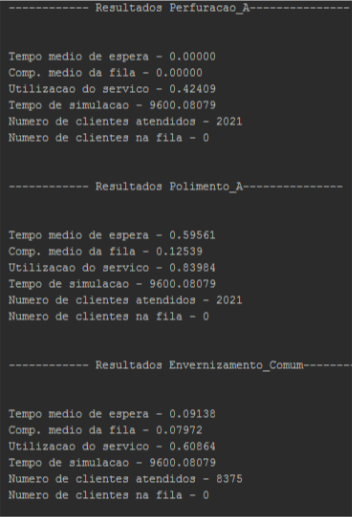


Ao aumentar a média do tempo de chegada das peças A, verificou-se uma ligeira diminuição da taxa de utilização dos serviços de Perfuração e Polimento A.

Média do tempo de chegada das peças A = 5.25

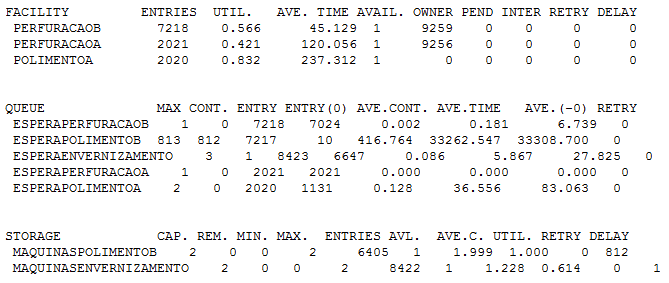


Média do tempo de chegada das peças A = 5.25 no GPSS



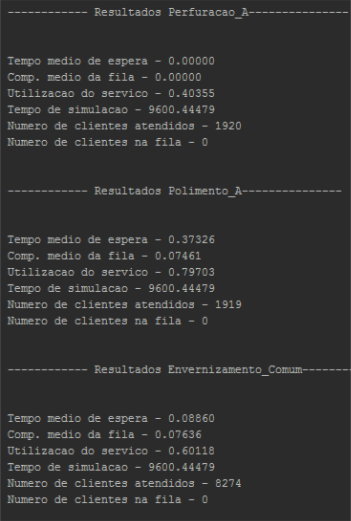
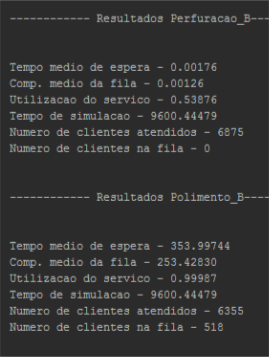
Ao diminuir a média do tempo de chegada das peças A, verificou-se um pequeno aumento da taxa de utilização dos serviços de Perfuração e Polimento A, uma vez que chegam mais peças numa taxa menor.

Média do tempo de chegada das peças A = 4.75



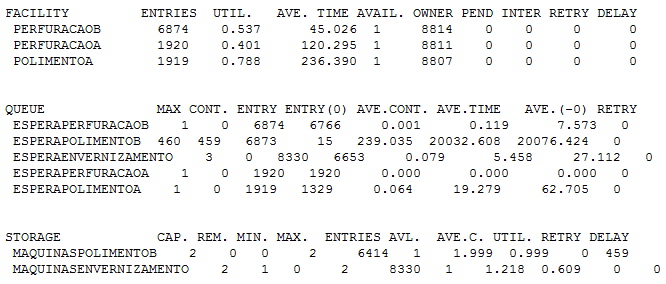
Média do tempo de chegada das peças A = 4.75 no GPSS

### Variação da média do tempo de chegada das peças B



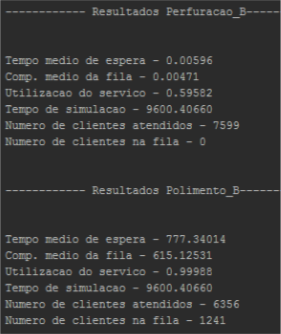
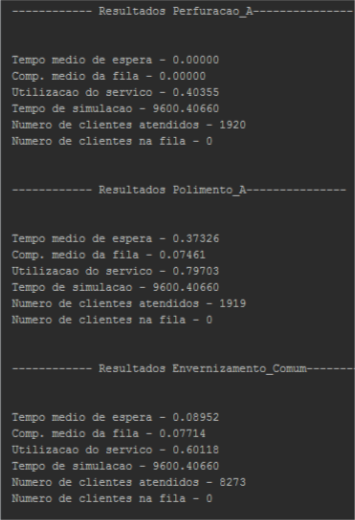
Aumentando o tempo de chegada, há uma ligeira redução na taxa de utilização do serviço da perfuração B.

Média do tempo de chegada das peças B = 1.3965

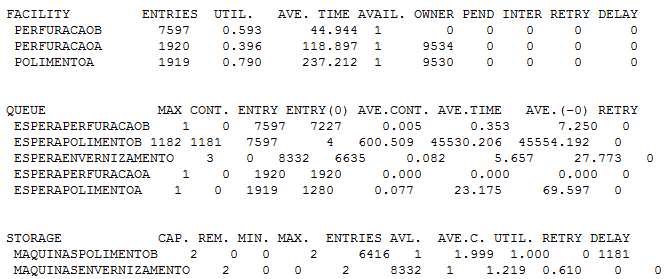


Média do tempo de chegada das peças B = 1.3965 no GPSS

Ao diminuir o tempo de chegada em 5% há um ligeiro aumento da taxa de perfuração B, o primeiro serviço que atende as peças.



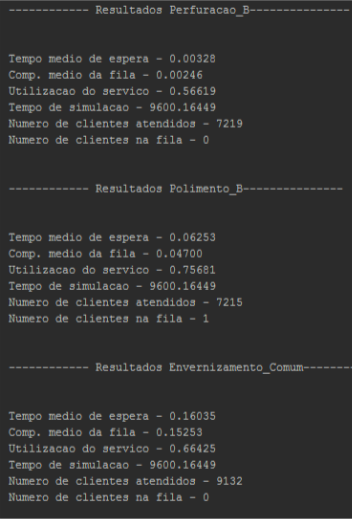
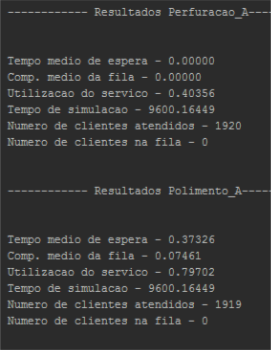
Média do tempo de chegada das peças B = 1.2635



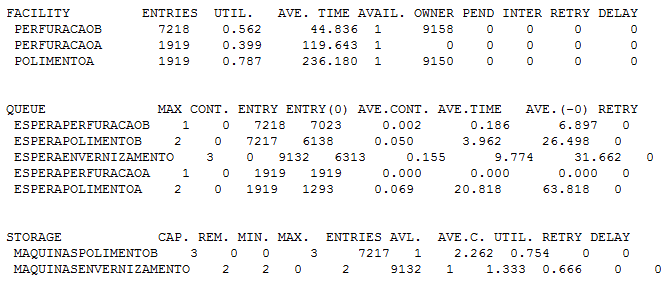
Média do tempo de chegada das peças B = 1.2635 no GPSS

### Alteração do número de máquinas do polimento B

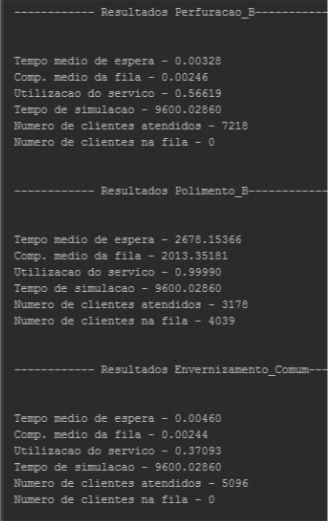
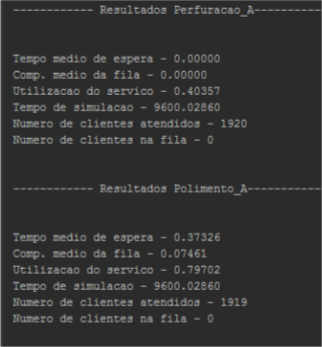
Inicialmente o número de máquinas era 2, o que fizemos aqui foi aumentar para 3. As principais diferenças foram o número de peças na fila de espera ser quase nula e a taxa de utilização que era quase de 100% passou para 75%, como seria de esperar.



Número de máquinas do polimento B = 3

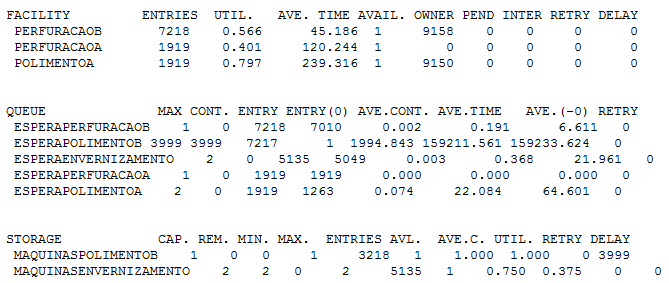


Número de máquinas do polimento B = 3 em GPSS



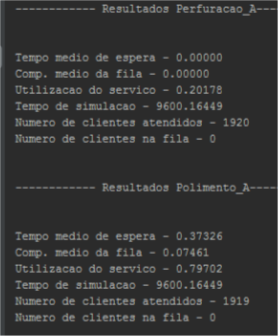
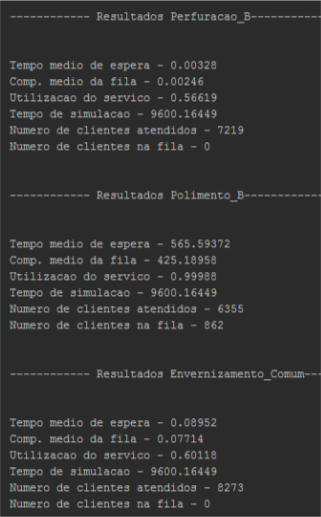
Agora diminuímos o número de máquinas de 2 para 1. A principal diferença é o grande aumento de peças na fila de espera, a taxa de utilização continuou a rondar os 100%.

Número de máquinas do polimento B = 1



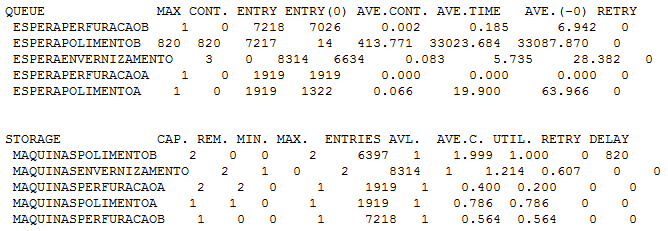
Número de máquinas do polimento B = 1 em GPSS

### Alteração do número de máquinas de perfuração A



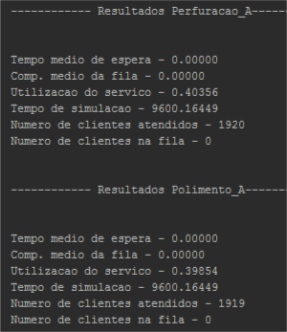
Aumentamos o número de máquinas de 1 para 2. A taxa de utilização diminui praticamente para metade e não se verificam clientes na fila.

Número de máquinas de perfuração A = 2



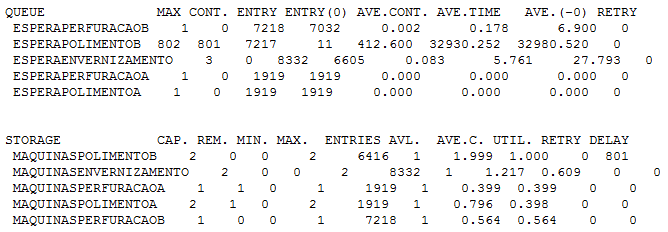
Número de máquinas de perfuração A = 2 em GPSS

### Alteração do número de máquinas de polimento A



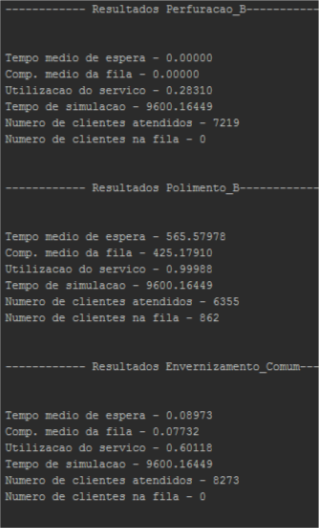
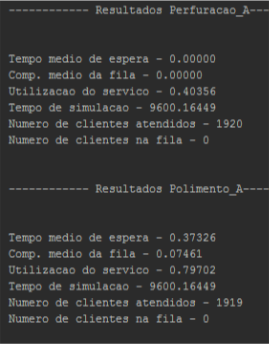
Ao aumentar o número de máquinas de 1 para 2, a taxa de utilização diminui para metade.

Número de máquinas de polimento A = 2



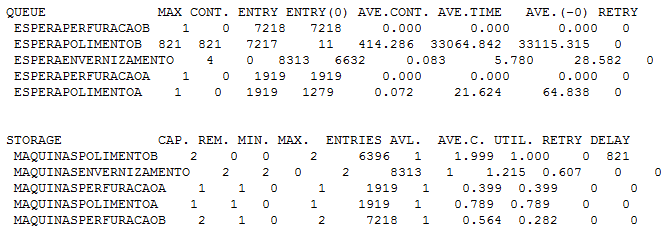
Número de máquinas de polimento A = 2 no GPSS

### Alteração do número de máquinas de perfuração B



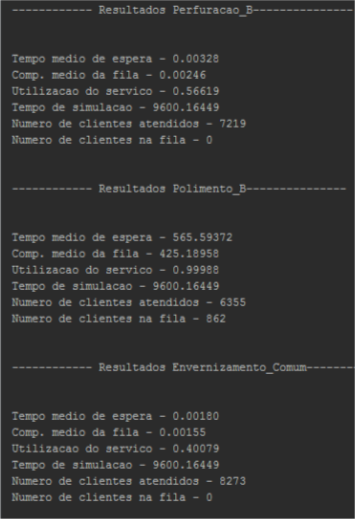
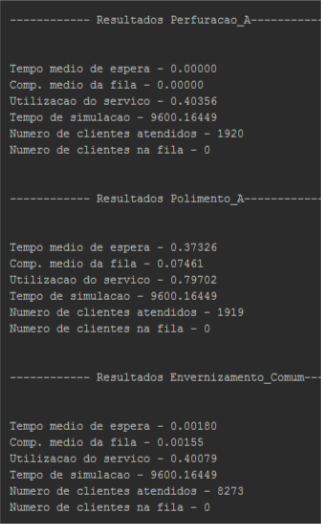
Aumentamos o número de máquinas de 1 para 2. A taxa de utilização diminui praticamente para metade e não se verificam clientes na fila.

Número de máquinas de perfuração B = 2



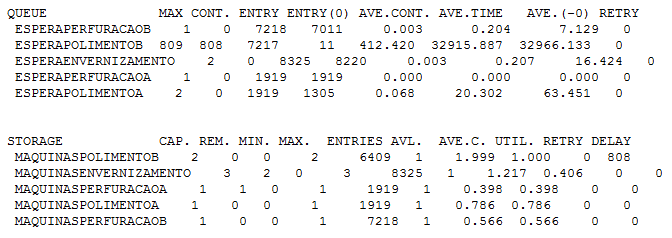
Número de máquinas de perfuração B = 2 no GPSS

### Alteração do número de máquinas de envernizamento



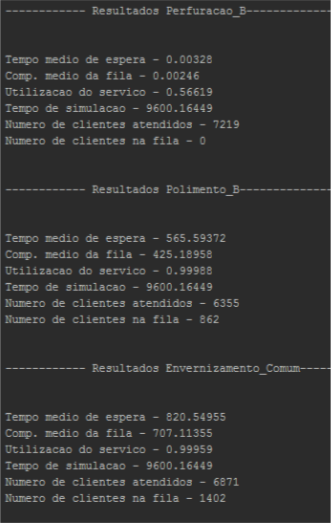
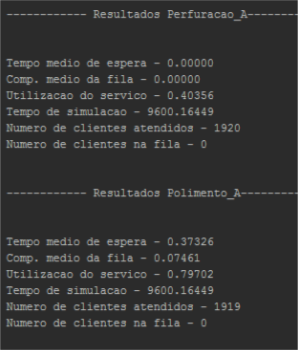
Com 3 máquinas de envernizamento em vez de 2, a taxa de utilização que normalmente rondava os 60% passa agora a ser de 40%, o tempo média de espera é quase nulo.

Número de máquinas de envernizamento = 3

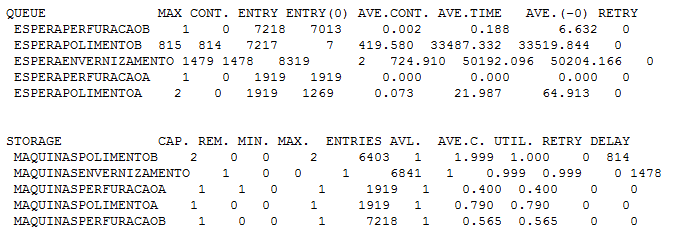


Número de máquinas de envernizamento = 3 no GPSS

Com apenas 1 máquina de envernizamento em vez de 2, a taxa de utilização que normalmente rondava os 60% passa agora a estar quase no limite e há também um enorme aumento do tempo e comprimento médio da fila de espera.



Número de máquinas de envernizamento = 1

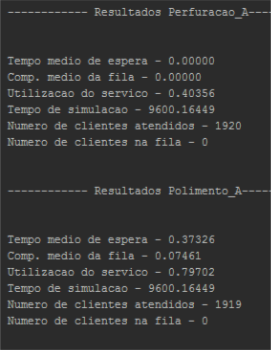
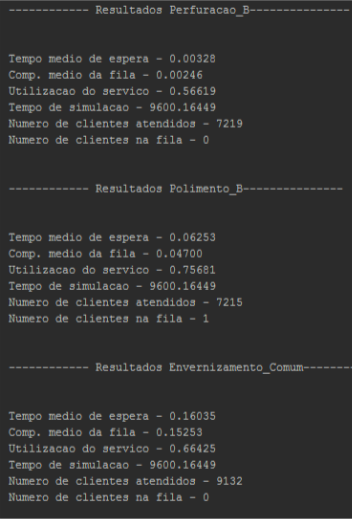


Número de máquinas de envernizamento = 1 no GPSS

# Alínea c)

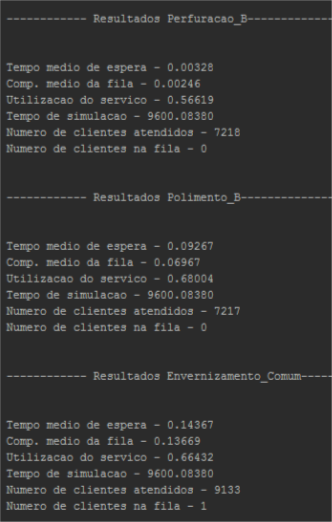
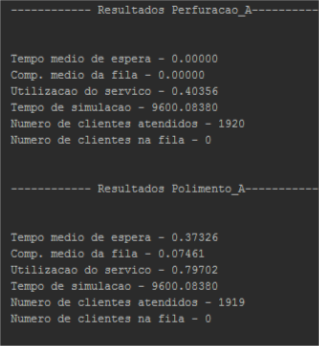
Conforme os resultados que abaixo apresentamos, verificamos que os valores das taxas de utilização e das filas de espera pouco diferem entre a alínea i) e a ii). Baseando-nos nos testes que realizámos e valores que obtivemos, sendo que a ii) seria uma solução mais cara do que a i), escolheríamos a solução mais económica, a i.

## i)



Resultados do aumento do número de máquinas do polimento B para 3

## ii)



Resultados da alteração da média do Polimento B para 1.7

# Alínea d)

Nesta alínea, começámos por testar a execução com diferentes dias para cada simulador para descobrir o dia a partir do qual eles começavam a gerar lucro.

Após termos descoberto isso testámos com o mesmo número de dias para os dois simuladores (i e ii) para descobrir o dia a partir do qual o lucro do segundo era maior que o do primeiro.

Verificou-se então, após muitos testes, que a solução i, sendo que envolve um investimento mais pequeno, apresenta lucro mais cedo. Mas a longo prazo, mais concretamente após 333 dias, a solução ii apresenta maior lucro que a primeira, sendo por isso, a mais rentável.

Os gráficos acima demonstram a relação entre as duas soluções e realçam a rentabilidade da solução ii, a longo prazo, sendo que em ambos se confirmou ser 333 dias o momento a partir do qual a solução ii rende mais que a i.

# Conclusão

Este trabalho permitiu consolidar vários conceitos importantes da cadeira, ficamos a entender como é que é possível, computacionalmente, recriar um sistema aplicado à vida real e aprendemos a importância que esta recriação tem no que toca a prever como o sistema se vai comportar a longo prazo.

# Anexo 1 – Classe Simulador

1. **import** rand\_generator
2. **import** Evento
3. **import** Lista
4. **import** Servico
5. **import** Peca
6. **import** Registrador
7. **import** Aleatorio

10. **class** Simulador:
11. """
12. Representa o simulador com os dados iniciais pre-inseridos
14. A unidade de tempo é o minuto
15. """
17. **def** \_\_init\_\_(self, nome: str = "SimuladorX", debug: bool = False, registrar: bool = True, aleatorio: bool = False,
18. gui\_stream: list = None):
19. self.nome = nome
21. self.debug = debug
22. self.registrar = registrar  *# Atrasa bastante o acesso ao programa*
23. self.seed\_aleatoria = aleatorio
25. *# Representa o sitio onde são adicionados os prints caso queiramos imprimir num GUI e não na consola*
26. self.gui\_stream = gui\_stream
28. **if** gui\_stream **is** **not** None:
29. self.debug = False
31. *# Numero de pecas diferentes*
32. self.numero\_pecas = 2
34. *# Relogio do simulador - Sempre inicializado a 0*
35. self.tempo = 0
37. *# Tempo de funcionamento da simulacao*
38. *#       Horas de producao por dia*
39. self.horas = 8
40. *#       Dias da simulacao*
41. self.dias = 20
43. *# Servicos - pode haver mais que um*
44. self.matriz\_servicos = [[] **for** i **in** range(self.numero\_pecas)]
45. *#   ---> Servicos Peca B*
46. self.matriz\_servicos[0].append(Servico.Servico(self, 2, 0.7, 10 \* 1111111, 10, nome="Perfuracao\_A"))
47. self.matriz\_servicos[0].append(Servico.Servico(self, 4, 1.2, 11 \* 1111111, 11, nome="Polimento\_A"))
48. *#   ---> Servicos Peca B*
49. self.matriz\_servicos[1].append(Servico.Servico(self, 0.75, 0.3, 12 \* 1111111, 12, nome="Perfuracao\_B"))
50. self.matriz\_servicos[1].append(Servico.Servico(self, 3, 1, 13 \* 1111111, 13, maquinas=2, nome="Polimento\_B"))
51. *#   ---> Servicos Comuns*
52. servico = Servico.Servico(self, 1.4, 0.3, 14 \* 1111111, 14, maquinas=2, nome="Envernizamento\_Comum")
53. self.matriz\_servicos[0].append(servico)
54. self.matriz\_servicos[1].append(servico)
56. self.servicos = []
57. **for** i **in** range(len(self.matriz\_servicos)):
58. **for** j **in** range(len(self.matriz\_servicos[i])):
59. **if** self.matriz\_servicos[i][j] **not** **in** self.servicos:
60. self.servicos.append(self.matriz\_servicos[i][j])
62. *# Lista de eventos - onde são mantidos todos os eventos da simulacao - Apenas existe uma por simulador*
63. self.lista = Lista.Lista(self)
65. *# Tipos de Pecas vendidas*
66. self.tipo\_pecas = [
67. Peca.Peca(0, 1 \* 1000000, 1, 5, Aleatorio.exp\_neg, "A", 0.05, seed\_aleatoria=self.seed\_aleatoria),
68. Peca.Peca(1, 2 \* 1000000, 2, 1.33, Aleatorio.exp\_neg, "B", 0.05, seed\_aleatoria=self.seed\_aleatoria)]
70. *# Numero de pecas vendidas*
71. self.pecas\_vendidas = [0 **for** i **in** range(self.numero\_pecas)]
72. self.pecas\_criadas = [0 **for** i **in** range(self.numero\_pecas)]
74. *# Custo inicial de quaisquer alteracoes*
75. self.divida = 0
77. **if** self.registrar:
78. self.registo = Registrador.comeca\_registo()
79. Registrador.regista(self.registo, str(self))
80. **else**:
81. self.registo = None
83. **def** \_\_str\_\_(self):
84. string = ""
85. string += "Simulador **\"**" + self.nome + "**\"**:**\n**"
87. **if** self.debug:
88. string += "**\t**Modo Debug**\n**"
89. **if** self.registrar:
90. string += "**\t**A registar valores em " + Registrador.diretorio\_registos + " -> " + self.registo.name + "**\n**"
91. **if** self.seed\_aleatoria:
92. string += "**\t**A usar seeds aleatorias**\n**"
94. string += "**\n\t**Horario de funcionamento: " + str(self.horas) + " "
95. **if** self.horas == 1:
96. string += "hora "
97. **else**:
98. string += "horas "
99. string += "por dia durante " + str(self.dias) + " "
100. **if** self.dias == 1:
101. string += "dia"
102. **else**:
103. string += "dias"
105. string += "**\n\t**Tipos de Pecas:**\n**"
106. **for** peca **in** self.tipo\_pecas:
107. string += altera\_string(str(peca), "**\t**", "**\t\t**") + "**\n**"
109. string += "**\n\t**Servicos:**\n**"
110. **for** i **in** range(len(self.tipo\_pecas)):
111. string += "**\t\t**Peca " + str(self.tipo\_pecas[i].nome) + ":**\n**"
112. **for** j **in** range(len(self.matriz\_servicos[i])):
113. string += altera\_string(str(self.matriz\_servicos[i][j]), "**\t**", "**\t\t\t**") + "**\n**"
115. **return** string
117. **def** insereEvento(self, evento):
118. self.lista.insere\_evento(evento)
120. **def** altera\_aleatoriedade(self, altera: bool):
121. self.seed\_aleatoria = altera
122. **for** serv **in** self.servicos:
123. serv.altera\_aleatoriedade(altera)
124. **for** peca **in** self.tipo\_pecas:
125. peca.altera\_aleatoriedade(altera)
127. **def** executa(self):
128. **for** i **in** range(self.numero\_pecas):
129. self.insereEvento(Evento.Chegada(self.tempo, self, self.matriz\_servicos[i][0], self.tipo\_pecas[i]))
130. rand\_generator.randst(self.tipo\_pecas[i].semente, self.tipo\_pecas[i].stream,
131. self.tipo\_pecas[i].seed\_aleatoria)
133. dias\_executados = 0
135. **if** self.registrar:
136. Registrador.regista(self.registo,
137. "**\n**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Inicio da Simulacao\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**\n**")
139. **while** dias\_executados < self.dias:
140. self.executa\_dia(dias\_executados)
141. **print**("Dia executado: " + str(dias\_executados + 1))
142. dias\_executados += 1
144. strings = self.relat()
145. **if** self.gui\_stream **is** None:
146. **for** linha **in** strings:
147. **print**(linha)
148. **else**:
149. self.gui\_stream.extend(strings)
151. **if** self.registrar:
152. **for** l **in** strings:
153. Registrador.regista(self.registo, l)
154. **if** self.gui\_stream **is** **not** None:
155. self.gui\_stream.append(
156. "**\n**Dados de registo salvos em " + Registrador.diretorio\_registos + " -> " + self.registo.name)
157. **print**("**\n**Dados de registo salvos em " + Registrador.diretorio\_registos + " -> " + self.registo.name)
158. Registrador.fecha\_registo(self.registo)
160. **def** executa\_dia(self, dias\_executados: int):
161. temp\_string = "**\n\n**||||||||||||||||||||**\n\n**"
162. temp\_string += "->**\t**Dia " + str(dias\_executados + 1) + ":"
163. temp\_string += "**\n\n**||||||||||||||||||||**\n**"
164. **if** self.debug:
165. **if** self.gui\_stream **is** None:
166. **print**(temp\_string)
167. **else**:
168. self.gui\_stream.append(temp\_string)
169. **if** self.registrar:
170. Registrador.regista(self.registo, temp\_string)
172. **while** self.tempo < (self.horas \* 60 \* (dias\_executados + 1)):
173. linhas = self.lista.lista\_to\_string()
174. **for** l **in** linhas:
175. **if** self.debug:
176. **if** self.gui\_stream **is** None:
177. **print**(l)
178. **else**:
179. self.gui\_stream.append(l)
180. **if** self.registrar:
181. Registrador.regista(self.registo, l)
182. evento = self.lista.retira\_evento()
183. self.tempo = evento.instante
184. self.act\_stats()
185. evento.executa()
187. **def** act\_stats(self):
188. """M�todo que actualiza os valores estat�sticos do simulador"""
189. atualizados = []
190. **for** i **in** range(self.numero\_pecas):
191. **for** j **in** range(len(self.matriz\_servicos[i])):
192. **if** self.matriz\_servicos[i][j] **not** **in** atualizados:
193. self.matriz\_servicos[i][j].act\_stats()
194. atualizados.append(self.matriz\_servicos[i][j])
196. **def** relat(self):
197. """M�todo que apresenta os resultados de simula��o finais"""
198. strings = []
199. **for** i **in** range(self.numero\_pecas):
200. strings.append("**\n\n**|----------------- Estatisticas da Peca " + str(
201. self.tipo\_pecas[i].nome) + " ------------------------|")
202. **for** j **in** range(len(self.matriz\_servicos[i])):
203. strings.append("**\n\n**------------ Resultados " + str(
204. self.matriz\_servicos[i][j].nome) + "---------------**\n\n**")
205. relat\_servico = self.matriz\_servicos[i][j].relat()
206. **for** string **in** relat\_servico:
207. strings.append(string)
209. strings.append("**\n\n**-----Resultados------------**\n\n**")
211. lucro\_total = 0
212. **for** i **in** range(self.numero\_pecas):
213. strings.append("Peca " + str(self.tipo\_pecas[i].nome) + ":")
214. strings.append("**\t**Preco por Peca: " + str(self.tipo\_pecas[i].custo) + " euros")
215. strings.append("**\t**Pecas Criadas: " + str(self.pecas\_criadas[i]))
216. strings.append("**\t**Pecas Vendidadas: " + str(self.pecas\_vendidas[i]))
217. lucro = self.pecas\_vendidas[i] \* self.tipo\_pecas[i].custo
218. lucro\_total += lucro
219. strings.append("**\t**Lucro total: " + "{:.2f}".format(lucro))
220. strings.append(
221. "**\t**Lucro apos pagar divida: " + "{:.2f}".format(lucro - self.divida))
223. strings.append("Lucro total producao: " + "{:.2f}".format(lucro\_total))
224. strings.append(
225. "Lucro producao apos pagar divida: " + "{:.2f}".format(lucro\_total - self.divida))
226. **return** strings
228. **def** regista\_servidor(self):
229. Registrador.regista(self.registo, "Simulador **\"**" + self.nome + "**\"**:")
231. string = "**\t**Horario de funcionamento: " + str(self.horas) + " "
232. **if** self.horas == 1:
233. string += "hora "
234. **else**:
235. string += "horas "
236. string += "por dia durante " + str(self.dias) + " "
237. **if** self.dias == 1:
238. string += "dia"
239. **else**:
240. string += "dias"
241. Registrador.regista(self.registo, string)
243. string = "**\t**Tipos de Pecas:**\n**"
244. **for** peca **in** self.tipo\_pecas:
245. string += "**\t\t**" + str(peca) + "**\n**"
246. Registrador.regista(self.registo, string)
248. string = "**\t**Servicos:**\n**"
249. **for** i **in** range(len(self.tipo\_pecas)):
250. string += "**\t\t**Peca " + str(self.tipo\_pecas[i].nome) + ":**\n**"
251. **for** j **in** range(len(self.matriz\_servicos[i])):
252. string += "**\t\t\t**" + str(self.matriz\_servicos[i][j]) + "**\n**"
253. Registrador.regista(self.registo, string)
255. **def** limpa\_servicos(self):
256. self.matriz\_servicos = [[] **for** i **in** range(self.numero\_pecas)]
257. self.servicos = []
259. **def** restora\_lista\_servicos(self):
260. self.servicos = []
261. **for** i **in** range(len(self.matriz\_servicos)):
262. **for** j **in** range(len(self.matriz\_servicos[i])):
263. **if** self.matriz\_servicos[i][j] **not** **in** self.servicos:
264. self.servicos.append(self.matriz\_servicos[i][j])
266. **def** remove\_servico(self, peca, indice):
267. self.matriz\_servicos[peca.tipo] = self.matriz\_servicos[peca.tipo][:indice] + self.matriz\_servicos[peca.tipo][
268. indice + 1:]
270. **def** altera\_maquinas\_servico(self, indice: int, maquinas: int):
271. self.servicos[indice].altera\_maquinas(maquinas)
273. **def** altera\_tempo\_servico(self, indice: int, media: float, desvio: float):
274. self.servicos[indice].altera\_tempo(media, desvio)
276. **def** altera\_tempo\_chegada(self, indice: int, media: float):
277. self.tipo\_pecas[indice].altera\_media(media)
279. **def** altera\_custo\_peca(self, indice: int, custo: float):
280. self.tipo\_pecas[indice].altera\_custo(custo)
282. **def** altera\_horas(self, horas: int):
283. self.horas = horas
285. **def** altera\_dias(self, dias: int):
286. self.dias = dias
288. **def** altera\_registrar(self, altera: bool):
289. **if** **not** self.registo **and** altera:
290. self.registo = Registrador.comeca\_registo()
292. **def** restora\_simulador(self):
293. self.tempo = 0
294. self.lista = Lista.Lista(self)
295. self.pecas\_vendidas = [0 **for** i **in** range(self.numero\_pecas)]
296. self.pecas\_criadas = [0 **for** i **in** range(self.numero\_pecas)]
297. **for** serv **in** self.servicos:
298. serv.restora\_servico()
300. **if** self.registrar:
301. self.registo = Registrador.comeca\_registo()
302. Registrador.regista(self.registo, str(self))
303. **else**:
304. self.registo = None

307. **def** altera\_string(string: str, prefixo: str, adiciona: str) -> str:
308. nova = ""
309. (antes, sep, depois) = string.partition(prefixo)
310. **while** sep != "":
311. nova += (adiciona + antes + sep)
312. (antes, sep, depois) = depois.partition(prefixo)
313. nova += adiciona + antes
314. **return** nova

317. **if** \_\_name\_\_ == "\_\_main\_\_":
318. sim = Simulador(registrar=True)
319. *# sim.altera\_dias(20)*
320. *# sim.altera\_horas(8)*
321. **print**(str(sim))
322. sim.executa()

# Anexo 2 – Classe Peça

1. **import** rand\_generator
3. **class** Peca:
4. """
5. Classe que representa uma peca
6. """
8. **def** \_\_init\_\_(self, tipo: int, semente: int, stream: int, media: float, metodo=None, nome: str = "",
9. custo: float = 0, seed\_aleatoria: bool = False):
10. self.tipo = tipo
11. self.nome = nome
12. self.custo = custo
14. self.semente = semente
15. self.stream = stream
17. self.media = media
18. self.metodo = metodo
19. self.seed\_aleatoria = seed\_aleatoria
21. **def** get\_chegada(self):
22. **if** self.metodo **is** None:
23. **return** self.media
24. **else**:
25. **return** self.metodo(self.stream, self.media)
27. **def** altera\_media(self, media):
28. self.media = media
30. **def** altera\_metodo(self, metodo=None):
31. self.metodo = metodo
33. **def** altera\_custo(self, custo):
34. self.custo = custo
36. **def** altera\_aleatoriedade(self, altera: bool):
37. rand\_generator.randst(self.semente, self.stream, altera)
38. self.seed\_aleatoria = altera
40. **def** \_\_str\_\_(self):
41. string = self.nome + "**\n**"
42. string += "**\t**-> Custo: " + str(self.custo) + "**\n**"
43. string += "**\t**-> Media Chegada: " + str(self.media) + "**\n**"
44. **if** self.metodo:
45. string += "**\t**-> Metodo Chegada: " + str(self.metodo.\_\_name\_\_) + "**\n**"
47. string += "**\t**-> Stream: " + str(self.stream) + "**\n**"
48. string += "**\t**-> Semente: " + str(self.semente)
49. **if** self.seed\_aleatoria:
50. string += "**\t**[Aleatoria]"
51. string += "**\n**"
52. **return** string

# Anexo 3 – Serviço

1. **import** Evento
2. **import** Aleatorio
3. **import** rand\_generator
5. **class** Servico:
6. """
7. Classe que representa um servico
8. """
9. **def** \_\_init\_\_(self, sim, media: float, desvio: float, semente: int, stream: int, maquinas: int = 1,
10. nome: str = "ServicoX"):
11. self.espera = []
12. self.simulador = sim
13. *# Numero de maquinas do Servico ocupadas*
14. self.ocupadas = 0
15. self.numero\_maquinas = maquinas
16. self.atendidos = 0
17. *# Tempo que passou desde o ultimo evento*
18. self.temp\_ultimo = sim.tempo
19. self.soma\_temp\_espera = 0
20. self.soma\_temp\_servico = 0
21. self.media = media
22. self.desvio = desvio
23. self.nome = nome
25. self.semente = semente
26. self.stream = stream
27. self.gerador = None
29. rand\_generator.randst(semente, stream, sim.seed\_aleatoria)
31. **def** \_\_str\_\_(self):
32. string = self.nome + "**\n**"
33. string += "**\t**-> Media Chegada: " + str(self.media) + " minutos**\n**"
34. string += "**\t**-> Desvio Chegada: " + str(self.desvio) + " minutos**\n**"
35. string += "**\t**-> Numero Maquinas: " + str(self.numero\_maquinas) + "**\n**"
37. string += "**\t**-> Stream: " + str(self.stream) + "**\n**"
38. string += "**\t**-> Semente: " + str(self.semente)
39. **if** self.simulador.seed\_aleatoria:
40. string += "**\t**[Aleatoria]"
41. string += "**\n**"
42. **return** string
44. **def** \_\_repr\_\_(self):
45. **return** "Servico<" + self.nome + ", " + str(self.media) + ", " + str(self.desvio) + ", " \
46. + str(self.numero\_maquinas) + ">"
48. **def** altera\_aleatoriedade(self, altera: bool):
49. rand\_generator.randst(self.semente, self.stream, altera)
51. **def** altera\_maquinas(self, maquinas: int):
52. self.numero\_maquinas = maquinas
54. **def** altera\_tempo(self, media: float, desvio: float):
55. self.media = media
56. self.desvio = desvio
58. **def** restora\_servico(self):
59. self.espera = []
60. self.ocupadas = 0
61. self.atendidos = 0
62. self.temp\_ultimo = self.simulador.tempo
63. self.soma\_temp\_espera = 0
64. self.soma\_temp\_servico = 0
65. rand\_generator.randst(self.semente, self.stream, self.simulador.seed\_aleatoria)
67. **def** get\_tempo(self):
68. *# GENERATOR POWER*
69. tempo = None
70. **while** tempo **is** None:
71. **try**:
72. tempo = next(self.gerador)
73. **except** TypeError:
74. self.gerador = Aleatorio.gerador\_dist\_normal(self.stream, self.media, self.desvio)
75. **except** StopIteration:
76. self.gerador = Aleatorio.gerador\_dist\_normal(self.stream, self.media, self.desvio)
78. **return** tempo
80. **def** inserePeca(self, peca):
81. """
82. Metodo que insere peca no servico
83. """
85. **if** self.ocupadas < self.numero\_maquinas:
86. self.ocupadas += 1
87. self.simulador.insereEvento(
88. Evento.Saida(self.simulador.tempo + self.get\_tempo(), self.simulador,
89. self, peca))
90. **else**:
91. self.espera.append(peca)
93. **def** retiraPeca(self):
94. """
95. Metodo que retira cliente do servico
96. """
97. self.atendidos += 1
98. **if** self.espera == []:  *# Se a fila esta vazia,*
99. self.ocupadas -= 1
100. **else**:
101. peca = self.espera.pop(0)
102. self.simulador.insereEvento(
103. Evento.Saida(self.simulador.tempo + self.get\_tempo(), self.simulador,
104. self, peca))
106. **def** act\_stats(self):
107. """Metodo que calcula os valores estatisticos a cada iteracao/evento do simulador"""
108. *# Calcula tempo que passou desde o ultimo evento*
109. temp\_desd\_ult = self.simulador.tempo - self.temp\_ultimo
110. self.temp\_ultimo = self.simulador.tempo
111. *# Contabiliza tempo de espera na fila*
112. *# para todos os clientes que estiveram na fila durante o intervalo*
113. self.soma\_temp\_espera += len(self.espera) \* temp\_desd\_ult
114. *# Contabiliza tempo de atendimento*
115. self.soma\_temp\_servico += self.ocupadas \* temp\_desd\_ult
117. **def** relat(self):
118. """Metodo que calcula e imprime os valores finais estatisticos"""
119. *# Tempo m�dio de espera na fila*
120. temp\_med\_fila = self.soma\_temp\_espera / (self.atendidos + len(self.espera))
121. *# Comprimento m�dio da fila de espera*
122. *# self.simulator.instant neste momento � o valor do tempo de simula��o,*
123. *# uma vez que a simula��o come�ou em 0 e este m�todo s� � chamdo no fim da simula��o*
124. comp\_med\_fila = self.soma\_temp\_espera / self.simulador.tempo
125. *# Tempo m�dio de atendimento no servi�o*
126. utilizacao\_serv = (self.soma\_temp\_servico / self.simulador.tempo) / self.numero\_maquinas
128. strings = []
129. strings.append("Tempo medio de espera - " + "{:.5f}".format(temp\_med\_fila))
130. strings.append("Comp. medio da fila - " + "{:.5f}".format(comp\_med\_fila))
131. strings.append("Utilizacao do servico - " + "{:.5f}".format(utilizacao\_serv))
132. strings.append("Tempo de simulacao - " + "{:.5f}".format(self.simulador.tempo))
133. strings.append("Numero de clientes atendidos - " + str(self.atendidos))
134. strings.append("Numero de clientes na fila - " + str(len(self.espera)))
136. **return** strings

# Anexo 4 – Classe Lista

1. **class** Lista:
2. """
3. Classe que contem os eventos do simulador a serem executados ordenados crescentemente pelo instante a serem executados
4. """
6. **def** \_\_init\_\_(self, sim):
7. self.simulador = sim  *# Simulador a que pertence a lista de eventos*
8. self.lista = []
10. **def** insere\_evento(self, evento):
11. *# Insere o evento no fim da lista e reordena-a*
12. self.lista.append(evento)
13. self.lista.sort()
15. **def** retira\_evento(self):
16. **return** self.lista.pop(0)
18. **def** imprime\_lista(self):
19. """
20. Imprime todos os eventos na lista
21. """
22. **print**("**\n\n**" + str(self.simulador.tempo) + ">>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>**\n**")
23. **for** eve **in** self.lista:
24. **print**(eve)
25. **print**("**\n\n**<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<**\n**")
27. **def** lista\_to\_string(self):
28. strings = []
29. strings.append("**\n\n**Relogio do simulador: " + "{:.5f}".format(self.simulador.tempo) + "**\n**>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>**\n**")
30. **for** eve **in** self.lista:
31. strings.append(str(eve))
32. strings.append("**\n\n**<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<**\n**")
33. **return** strings

# Anexo 5 – Classe Evento

1. **import** Peca
2. **import** Aleatorio
4. **class** Evento:
5. """
6. Classe abstrata da qual sao derivadas todos os eventos
7. """
9. **def** \_\_init\_\_(self, inst: float, sim, servico, peca):
10. self.instante = inst  *# Instante a que o evento ocorre*
11. self.simulador = sim
12. self.servico = servico
13. self.peca = peca
15. **def** \_\_lt\_\_(self, outro):
16. """
17. Determina se um evento e menor que outro usando o seu instante de ocurrencia
18. """
19. **return** self.instante < outro.instante
21. **def** \_\_str\_\_(self):
22. **return** "T: " + str(self.instante) + "**\t**" + self.\_\_class\_\_.\_\_name\_\_ + " da peca " + str(self.peca) \
23. + " do servico " + str(self.servico)

26. **class** Chegada(Evento):
27. """
28. Representa a chegada de uma Peca a um Servico
29. """
31. **def** \_\_init\_\_(self, inst: float, sim, servico, peca):
32. super().\_\_init\_\_(inst, sim, servico, peca)
34. **def** executa(self):
35. self.servico.inserePeca(self.peca)
36. indice = self.peca.tipo
37. self.simulador.pecas\_criadas[indice] += 1
38. **if** self.servico == self.simulador.matriz\_servicos[indice][0]:
39. tempo\_extra = self.peca.get\_chegada()
40. self.simulador.insereEvento(Chegada(self.simulador.tempo + tempo\_extra, self.simulador, self.servico,
41. Peca.Peca(self.peca.tipo, self.peca.semente, self.peca.stream,
42. self.peca.media, nome=self.peca.nome, custo=self.peca.custo)))

45. **class** Saida(Evento):
46. """
47. Representa a saida de uma Peca a um Servico, i.e., a libertacao do Servico
48. """
50. **def** \_\_init\_\_(self, inst: float, sim, servico, peca):
51. super().\_\_init\_\_(inst, sim, servico, peca)
53. **def** executa(self):
54. self.servico.retiraPeca()
55. indice\_peca = self.peca.tipo
56. indice\_servico = 0
57. **for** i **in** range(len(self.simulador.matriz\_servicos[indice\_peca])):
58. **if** self.simulador.matriz\_servicos[indice\_peca][i] == self.servico:
59. indice\_servico = i
60. **if** indice\_servico + 1 < len(self.simulador.matriz\_servicos[indice\_peca]):
61. self.simulador.matriz\_servicos[indice\_peca][indice\_servico + 1].inserePeca(self.peca)
62. **else**:
63. self.simulador.pecas\_vendidas[indice\_peca] += 1

# Anexo 6 – Classe Aleatório

1. **import** rand\_generator
2. **import** math

5. **def** exp\_neg(stream: int, media: float = 1):
6. """
7. Retorna um valor aleatorio seguindo uma distribuicao exponencial de media "media"
9. :param media: Media da exponencial
10. :type media: float
11. :return: Valor resultante da exponencial
12. :rtype: float
13. """
14. **return** -media \* math.log(rand\_generator.rand(stream))

17. **def** gerador\_dist\_normal(stream: int, media: float, desvio: float):
18. **while** True:
19. v1 = 2 \* rand\_generator.rand(stream) - 1
20. v2 = 2 \* rand\_generator.rand(stream) - 1
22. w = math.pow(v1, 2) + math.pow(v2, 2)
23. *# Se w for maior que 1 ou menor ou igual a 0 refaz os valores*
24. **if** w > 0 **and** **not** w > 1:
25. **break**
27. *# w esta entre 0 e 1, exclusive no 0*
28. y1 = v1 \* math.sqrt((-2 \* math.log(w) / w))
29. y2 = v2 \* math.sqrt((-2 \* math.log(w) / w))
31. X1 = media + y1 \* desvio
32. X2 = media + y2 \* desvio
34. *# Apenas fornecemos os valores se forem maiores que 0*
36. **if** X1 > 0:
37. **yield** X1
39. **if** X2 > 0:
40. **yield** X2

# Anexo 7 – Classe Registrador

1. **import** sys
2. **import** datetime
3. **import** os
5. diretorio\_registos = os.getcwd() + "**\R**egistos"

8. **def** novo\_diretorio():
9. **if** **not** os.path.isdir(diretorio\_registos):
10. os.mkdir(diretorio\_registos)

13. **def** comeca\_registo(nome: str = None):
14. data = datetime.datetime.today()
16. **if** os.getcwd() != diretorio\_registos:
17. novo\_diretorio()
18. os.chdir(diretorio\_registos)
20. nome\_ficheiro = "Registo - "
21. **if** nome **is** None:
22. nome\_ficheiro += "\_".join(
23. (str(data.year), str(data.month), str(data.day), "-".join((str(data.hour), str(data.minute)))))
24. **else**:
25. nome\_ficheiro += nome
26. nome\_ficheiro += ".txt"
28. ficheiro = open(nome\_ficheiro, "a")
29. ficheiro.write("------------------- Registo ")
30. **if** nome **is** None:
31. ficheiro.write("\_".join(
32. (str(data.year), str(data.month), str(data.day), ":".join((str(data.hour), str(data.minute))))))
33. **else**:
34. ficheiro.write("**\"**" + nome + "**\"**")
36. ficheiro.write("--------------------**\n\n**")
37. *# Fazer flush de um ficheiro esvazia o buffer criado pelo programa para ele, i.e. realiza uma*
38. *# system call para os dados serem guardados, no entanto podem existir tambem buffers no sistema operativo, ou seja,*
39. *# os dados podem não ser escritos para disco imediatamente mas para uma memoria temporaria, e.g. RAM*
40. *# Para esvaziar o buffer do SO importar o modulo "os" e usar a funcao fsync*
41. ficheiro.flush()
43. **return** ficheiro

46. **def** regista(ficheiro, linha: str):
47. ficheiro.write(linha + "**\n**")
48. ficheiro.flush()

51. **def** fecha\_registo(ficheiro):
52. ficheiro.close()

# Anexo 8 – Simulação com os valores iniciais no GPSS

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*

\* PROJETO SCC PEÇAS Simulation \*

\* \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

GENERATE 28800 ;8 horas -> 8\*60\*60segundos

TERMINATE 1

GENERATE 300 ;Peca A ;5 min -> 5\*60segundos

TRANSFER ,PerfuracaoPecaA

GENERATE 79.8 ;Peca B ;1.33min -> 1.33\*60segundos

MaquinasPolimentoB STORAGE 2

MaquinasEnvernizamento STORAGE 2

TRANSFER ,PerfuracaoPecaB

PerfuracaoPecaA QUEUE EsperaPerfuracaoA

SEIZE PerfuracaoA

DEPART EsperaPerfuracaoA

ADVANCE (ABS(Normal(3,120,42))) ;0.7 min -> e 2 min->

RELEASE PerfuracaoA

TRANSFER ,PolimentoPecaA

PolimentoPecaA QUEUE EsperaPolimentoA

SEIZE PolimentoA

DEPART EsperaPolimentoA

ADVANCE (ABS(Normal(4,240,72))) ;1.2 min -> e 4 min ->

RELEASE PolimentoA

TRANSFER ,Envernizamento

PerfuracaoPecaB QUEUE EsperaPerfuracaoB

SEIZE PerfuracaoB

DEPART EsperaPerfuracaoB

ADVANCE (ABS(Normal(5,45,18))) ;0.75 min -> e 0.3 min ->

RELEASE PerfuracaoB

TRANSFER ,PolimentoPecaB

PolimentoPecaB QUEUE EsperaPolimentoB

ENTER MaquinasPolimentoB

DEPART EsperaPolimentoB

ADVANCE (ABS(Normal(6,180,60))) ;3 min -> e 1 min ->

LEAVE MaquinasPolimentoB

TRANSFER ,Envernizamento

Envernizamento QUEUE EsperaEnvernizamento

ENTER MaquinasEnvernizamento

DEPART EsperaEnvernizamento

ADVANCE (ABS(Normal(7,84,18))) ;1.4 min-> e 0.3 min->

LEAVE MaquinasEnvernizamento

Fim TERMINATE;

# Anexo 9 – Simulação do cenário i)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*

\* PROJETO SCC PEÇAS Simulation \*

\* \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

GENERATE 28800 ;8 horas -> 8\*60\*60segundos

TERMINATE 1

GENERATE 300 ;Peca A ;5 min -> 5\*60segundos

TRANSFER ,PerfuracaoPecaA

GENERATE 63 ;Peca B ;1.33min -> 1.33\*60segundos ///mudado p 63segundos

MaquinasPolimentoB STORAGE 3

MaquinasEnvernizamento STORAGE 2

MaquinasPerfuracaoA STORAGE 1

MaquinasPolimentoA STORAGE 1

MaquinasPerfuracaoB STORAGE 1

TRANSFER ,PerfuracaoPecaB

PerfuracaoPecaA QUEUE EsperaPerfuracaoA

ENTER MaquinasPerfuracaoA

DEPART EsperaPerfuracaoA

ADVANCE (ABS(Normal(3,120,42))) ;0.7 min -> e 2 min->

LEAVE MaquinasPerfuracaoA

TRANSFER ,PolimentoPecaA

PolimentoPecaA QUEUE EsperaPolimentoA

ENTER MaquinasPolimentoA

DEPART EsperaPolimentoA

ADVANCE (ABS(Normal(4,240,72))) ;1.2 min -> e 4 min ->

LEAVE MaquinasPolimentoA

TRANSFER ,Envernizamento

PerfuracaoPecaB QUEUE EsperaPerfuracaoB

ENTER MaquinasPerfuracaoB

DEPART EsperaPerfuracaoB

ADVANCE (ABS(Normal(5,45,18))) ;0.75 min -> e 0.3 min ->

LEAVE MaquinasPerfuracaoB

TRANSFER ,PolimentoPecaB

PolimentoPecaB QUEUE EsperaPolimentoB

ENTER MaquinasPolimentoB

DEPART EsperaPolimentoB

ADVANCE (ABS(Normal(6,180,60))) ;3 min -> e 1 min

LEAVE MaquinasPolimentoB

TRANSFER ,Envernizamento

Envernizamento QUEUE EsperaEnvernizamento

ENTER MaquinasEnvernizamento

DEPART EsperaEnvernizamento

ADVANCE (ABS(Normal(7,84,18))) ;1.4 min-> e 0.3 min

LEAVE MaquinasEnvernizamento

Fim TERMINATE;

# Anexo 9 – Simulação do cenário ii)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*

\* PROJETO SCC PEÇAS Simulation \*

\* \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

GENERATE 28800 ;8 horas -> 8\*60\*60segundos

TERMINATE 1

GENERATE 300 ;Peca A ;5 min -> 5\*60segundos

TRANSFER ,PerfuracaoPecaA

GENERATE 57 ;Peca B ;1.33min -> 1.33\*60segundos ///mudado p 57segundos

MaquinasPolimentoB STORAGE 2

MaquinasEnvernizamento STORAGE 2

MaquinasPerfuracaoA STORAGE 1

MaquinasPolimentoA STORAGE 1

MaquinasPerfuracaoB STORAGE 1

TRANSFER ,PerfuracaoPecaB

PerfuracaoPecaA QUEUE EsperaPerfuracaoA

ENTER MaquinasPerfuracaoA

DEPART EsperaPerfuracaoA

ADVANCE (ABS(Normal(3,120,42))) ;0.7 min -> e 2 min->

LEAVE MaquinasPerfuracaoA

TRANSFER ,PolimentoPecaA

PolimentoPecaA QUEUE EsperaPolimentoA

ENTER MaquinasPolimentoA

DEPART EsperaPolimentoA

ADVANCE (ABS(Normal(4,240,72))) ;1.2 min -> e 4 min ->

LEAVE MaquinasPolimentoA

TRANSFER ,Envernizamento

PerfuracaoPecaB QUEUE EsperaPerfuracaoB

ENTER MaquinasPerfuracaoB

DEPART EsperaPerfuracaoB

ADVANCE (ABS(Normal(5,45,18))) ;0.75 min -> e 0.3 min ->

LEAVE MaquinasPerfuracaoB

TRANSFER ,PolimentoPecaB

PolimentoPecaB QUEUE EsperaPolimentoB

ENTER MaquinasPolimentoB

DEPART EsperaPolimentoB

ADVANCE (ABS(Normal(6,102,60))) ;3 min -> e 1 min /////////mudado p 102seg

LEAVE MaquinasPolimentoB

TRANSFER ,Envernizamento

Envernizamento QUEUE EsperaEnvernizamento

ENTER MaquinasEnvernizamento

DEPART EsperaEnvernizamento

ADVANCE (ABS(Normal(7,84,18))) ;1.4 min-> e 0.3 min

LEAVE MaquinasEnvernizamento

Fim TERMINATE;