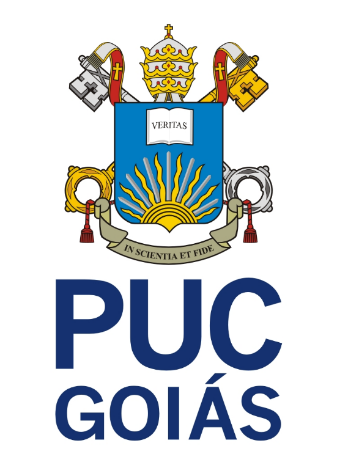
**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS**

**Escola de Ciências Exatas e da Computação**

**Ciência da Computação**



Leonardo Emilly Dias Santos Bidó

Yuri Pereira Fogaça

Walefe Pereira De Jesus

**Pseudocódigo produtor-consumidor utilizando a API do Windows**

Goiânia

*2018/2*

Sumário

[Objetivo 3](#_Toc532830399)

[Descrição 3](#_Toc532830400)

[Etapas 3](#_Toc532830401)

[Pseudocódigos 3](#_Toc532830402)

[Referências 6](#_Toc532830403)

# Objetivo

O pseudocódigo visa estabelecer uma relação entre as chamadas ao sistema para manipulação de processos, sincronização de processos e threads e memória compartilhada para permitir o desenvolvimento de aplicação de produtores e consumidores com memória compartilhada.

# Descrição

Utilizou-se toda a documentação de programação no Windows, tais como as bibliotecas padrão das linguagens C/C++ e Python para desenvolver programas que utilizem chamadas à API (*application programming interface*) do sistema para controlar o acesso a seções críticas, utilizando controle de concorrência, criação de processos e threads, memória compartilhada e outros aspectos de sistemas operacionais.

# Etapas

O projeto consiste das seguintes etapas:

1. Especifica os pseudocódigos para criação de processos e threads, sendo que há pelo menos 1 (um) processo produtor e 1 (um) consumidor.
2. Desenvolve algoritmo com pseudocódigo para permitir que threads no mesmo processo possam compartilhar buffer com bloqueio de produtor e do consumidor nas condições clássicas do algoritmo.
3. Propor um mecanismo de compartilhamento de buffer utilizando os mecanismos de memória compartilhada do Windows, em conjunto com as soluções obtidas nos itens anteriores.

# Pseudocódigos

Abaixo seguem os pseudocódigos que realizam as chamadas ao sistema Windows utilizando a API *pywin32* na linguagem Python. Tais chamadas representam a criação de *processos* produtores/consumidores associados à criação de *threads* e possuem gerenciamento de *memória compartilhada.*



Figura 1: Criação do Processo Produtor



Figura 2: Criação do Processo Consumidor

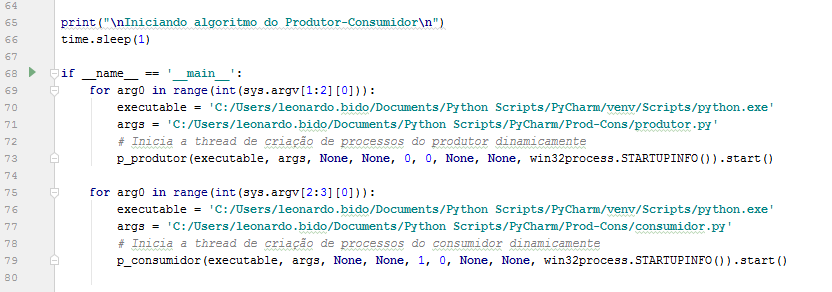


Figura 3: Menu para inicialização de threads produto/consumidor

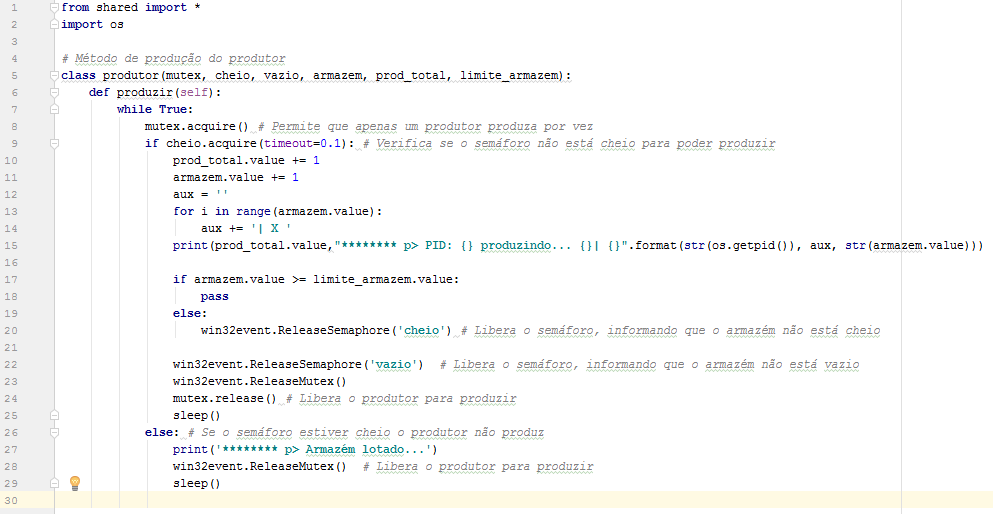


Figura 4: Classe Produtor

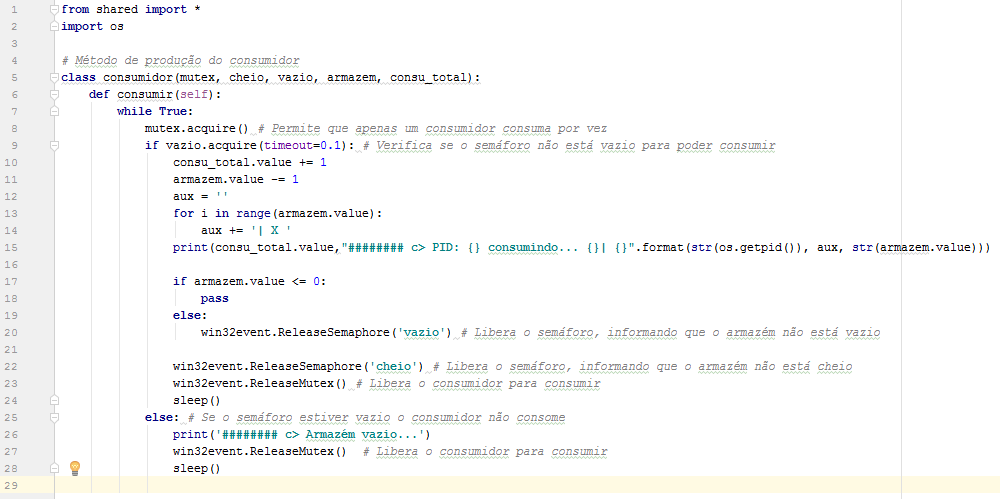


Figura 5: Classe Consumidor

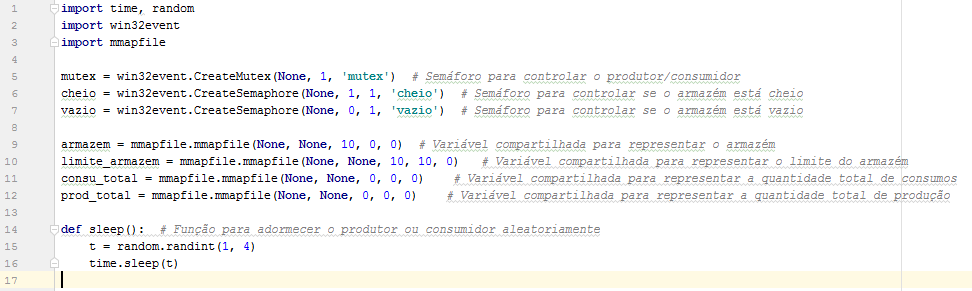


Figura 6: Declaração e compartilhamento de semáforos e espaço de memória

# Referências

RICHTER, J.; NASARRE, C. Windows via C/C++, Fifth Edition, Microsoft Press, 2008.

RUSSINOVICH, Mark E.; SALOMON, David A.; IONESCU, Alex. **Windows internals – parte 1**. 6. ed. USA: Microsoft Press, 2012.

RUSSINOVICH, Mark E.; SALOMON, David A.; IONESCU, Alex. **Windows Internals – parte 2**. 6. ed. USA: Microsoft Press, 2012.

GOLDEN, TIM. Python for Win32 Extensions Help. Disponível em: <<http://timgolden.me.uk/pywin32-docs/contents.html>>. Acesso em: 17/12/2018.

HAMMOND, MARK. Python for Windows (pywin32) Extensions. Disponível em: <https://github.com/mhammond/pywin32>. Acesso em: 17/12/2018.