

## CURSO TÉCNICO DE GESTÃO E PROGRAMAÇÃO DE SISTEMAS INFORMÁTICOS

### Programação



Aluno: Gabriel Asafe De Jesus Martins Nº 13

Cantanhede

2020/2021

Escola Técnico Profissional de Cantanhede

CURSO TÉCNICO DE GESTÃO E PROGRAMAÇÃO DE SISTEMAS INFORMÁTICOS  
2019/22

Programação

Gabriel Asafe De Jesus Martins Nº 12

11º Ano TGPSI

Conceitos avançados de programação

Módulo 9

Formador: Michael Teixeira

Cantanhede

24/09/2020

# Índice

|   |           |
|---|-----------|
| <b>Índice</b>                                       | <b>3</b>  |
| <b>Introdução</b>                                   | <b>3</b>  |
| <b>Sistema operativo</b>                            | <b>5</b>  |
| <b>Janela</b>                                       | <b>7</b>  |
| <b>Interface gráfica com o utilizador</b>           | <b>7</b>  |
| <b>Memória virtual</b>                              | <b>8</b>  |
| <b>Sistema operativo multitarefa e monotarefa</b>   | <b>9</b>  |
| <b>Programação por eventos</b>                      | <b>11</b> |
| <b>API- Interface para programação de aplicação</b> | <b>12</b> |
| <b>Conclusão</b>                                    | <b>14</b> |
| <b>Webgrafia</b>                                    | <b>15</b> |

## Introdução

Este trabalho foi proposto pelo professor Michael Teixeira no âmbito da disciplina de Programação como matéria programática do módulo 9. Teve por objetivo que os alunos salientarem de forma mais aprofundada conceitos estudados no 10º ano, como: sistemas monotarefas e multitarefa; Memória virtual. Além de nos introduzir a novas ferramentas e conceitos relacionados a programação com interface gráfica. No decorrer do trabalho os temas são abordados e exemplificados da forma mais esclarecedora possível, visando o aprendizado do aluno e a explicação esclarecedora dos temas.

O trabalho está organizado em tópicos referentes ao tema proposto pelo professor e está organizado da seguinte forma: Sistema operativo; Janela; Interface gráfica com o utilizador; Memória virtual; Sistema operativo multitarefa e monotarefa; Programação por eventos; API- Interface para programação de aplicação.

O trabalho teve uma grande importância pois foi a base introdutória para os estudos em sala e posteriormente a experimentação prática do tema.

## Sistema operativo

Um computador é controlado por um SO (Sistema Operativo), sigla utilizada para designar esta classe de software. Na prática um SO é a interface entre o utilizador e a máquina cujas principais funções são:

- Gestão de Processos;
- Gestão de Memória;
- Sistema de Ficheiros;
- Entrada e Saída de Dados;

### Componentes

Um sistema operativo é composto por diversos componentes que permitem o funcionamento conjunto do computador.

As componentes mais importantes são:

- Kernel;
- Rede;
- Segurança;
- Interface;

O Kernel (em português, núcleo), em conjunto com os drivers e o firmware do dispositivo, fornece o nível mais básico de controle sobre todos os dispositivos de hardware do computador. Este gerencia os acessos à memória RAM e determina quais os programas com maior prioridade e quais os recursos do hardware a que as aplicações acedem.

A Rede é fundamental na maioria dos Sistemas Operativos pois permite-nos a ligação com outras máquinas, incluindo máquinas que estão na Internet.

Assim, computadores com Sistemas Operativos diferentes, podem participar na mesma rede e partilhar recursos entre eles, como impressoras, scanners, etc.

A segurança é outra das características fundamentais num sistema operativo pois deverá ter mecanismos para proteger a informação, para que os processos acedem às zonas de memória permitidas, etc-

A interface com o utilizador pode ser gráfica ou linha de comandos (terminal). Atualmente, quase todos os Sistemas Operativos contam com uma interface gráfica, mas ainda é possível encontrar o terminal em muitos locais.

Depois do arranque do computador pela BIOS, parte do sistema operativo é carregado para a memória RAM. Os programas de computador funcionam dentro do Sistema Operativo e passam a chamar-se processos quando executados.



## Janela

Em informática, uma janela é uma área visual contendo algum tipo de interface do utilizador, permitindo a saída do sistema ou permitindo a entrada de dados. Uma interface gráfica do usuário que use janelas como uma de suas principais metáforas é chamada sistema de janelas, como um gerenciador de janelas.

As janelas são geralmente apresentadas como objetos bidimensionais e retangulares, organizados em uma área de trabalho. Normalmente um programa de computador assume a forma de uma janela para facilitar a assimilação pelo utilizador. Entretanto, o programa pode ser apresentado em mais de uma janela, ou até mesmo sem uma respectiva janela.

## Interface gráfica com o utilizador

Na informática, a sigla GUI refere-se à denominação "Graphical User Interface" (Interface Gráfica do Usuário, em português), que consiste em um modelo de interface do utilizador que permite a interação com os dispositivos digitais através de elementos gráficos.

A Xerox foi quem utilizou a interface gráfica pela primeira vez, porém, como produto, a GUI foi utilizada de maneira inédita pela Apple. A interação com os dispositivos digitais é geralmente realizada por meio de um mouse, teclado ou através do simples toque em telas sensíveis que equipam smartphones, tablets e computadores. Por meio dessa interação, os usuários podem selecionar e manipular ícones e símbolos a fim de conseguir realizar tarefas com algum dispositivo. Esses símbolos são denominados de widgets, na maioria dos casos, e são agrupados em kits.

O ambiente gráfico é um programa criado para tornar mais fácil a utilização do computador por meio de representações visuais do sistema operacional. O Windows possui apenas o ambiente gráfico padrão, com a chamada Windows Aero nas versões Windows Vista e Windows 7. Já para a GNU/Linux possui vários ambientes gráficos, como o KDE e o Gnome.

A interface gráfica é utilizada amplamente como solução em diversos ramos da indústria. Ela foi adotada pela primeira vez em um sistema de reservas no começo dos anos 2000,

com a primeira versão do Amadeus Selling Platform. O objetivo da interface gráfica do usuário no projeto da plataforma Amadeus foi oferecer maior facilidade aos agentes de viagem para as reservas de voos e assentos, bem como hotéis, carros, seguros e demais serviços.

A GUI utiliza uma combinação de tecnologias e dispositivos para fornecer uma plataforma prática na qual o usuário possa interagir. Nos computadores pessoais, a combinação formada por janelas, ícones, menus e ponteiros é conhecida como WIMP. Para o funcionamento desse sistema, usa-se um mouse como dispositivo de ponteiro para que seja controlada a execução de aplicações e para apresentar informações organizadas dentro de janelas e simbolizadas por meio de ícones. Os comandos disponíveis no sistema são acionados por meio do mouse. Para a interação entre as janelas, aplicações e o sistema de janelas, existe um gerenciador.

## Memória virtual

Também conhecida como arquivos de paginação, a memória virtual consiste em um espaço que é reservado no momento em que o sistema operacional é instalado em um computador. Essa ação independe do fato do SO ser Linux ou Windows.

A memória virtual é importante porque, assim que o SO percebe que a memória RAM está sem espaço, ele passa a executar os programas nela. Ou seja, a memória virtual pode ser considerada uma reserva da memória RAM.

Dessa forma, ela estende a quantidade de memória disponível para dados temporários. Estes costumam ser usados pelos programas em execução. Com isso, o usuário não precisa realizar upgrades de hardware para usar vários aplicativos simultaneamente.

Sempre que um programa usa toda a capacidade da memória RAM, o SO passa a usar a memória virtual. Ou seja, ela funciona como um escape para que o usuário possa continuar realizando suas atividades no computador. Entretanto, ao ser utilizada, o sistema fica consideravelmente mais lento.



Para funcionar, a memória virtual utiliza um pedaço do HD. Com isso, ela transfere partes da memória RAM para o disco rígido, viabilizando o acesso a mais memória RAM para a execução de programas. Contudo, esse recurso só é disponibilizado quando o computador está no seu limite ou quando o usuário está usando programas mais pesados.

## Sistema operativo multitarefa e monotarefa

### Sistemas Monotarefa

Os primeiros Sistemas Operacionais eram tipicamente voltados para a execução de um único programa. Qualquer outra aplicação, para ser executada, deveria aguardar o término do programa atualmente em execução. Este tipo de sistema se caracteriza por permitir que o processador, a memória e os periféricos dediquem-se, exclusivamente, a um único programa.

Estes sistemas são chamados de monotarefa, enquanto um programa aguarda por um evento, como a digitação de um dado pelo usuário.

O processador permanece ocioso, sem realizar qualquer tipo de processamento.

A memória é subutilizada caso o programa não a preencha totalmente, e os periféricos, como discos e impressoras, estão sempre à espera de um único usuário, nem sempre utilizados de forma integral - o que torna o sistema, também, monousuário.

Um exemplo típico de sistema monotarefa é o MS-DOS. Apenas uma janela era aberta e apenas um usuário utilizava os seus recursos. Não havia condições de compartilhamento em rede ou qualquer coisa relacionada.

Mesmo as primeiras versões do Windows, que já foram chamadas de multitarefa, não possuíam esta característica de fato, pois eram executadas sobre o MS-DOS. Para procurar contornar essa limitação, emulava-se a multitarefa, mas o núcleo do programa fazia a sua própria gestão dos processos. Se um processo bloqueasse o Windows, todas as aplicações teriam que ser terminadas pois eram todas dependentes.

```
C:\>MEM  
  
655360 bytes total conventional memory  
655360 bytes available to MS-DOS  
578352 largest executable program size  
  
4194304 bytes total EMS memory  
4194304 bytes free EMS memory  
  
19822944 bytes total contiguous extended memory  
0 bytes available contiguous extended memory  
15586160 bytes available XMS memory  
MS-DOS resident in High Memory Area  
  
C:\>
```

## Sistemas Multitarefa

Num sistema operacional multitarefa, várias aplicações podem utilizar a memória ao mesmo tempo. O SO define quais são os processos entre os quais o usuário poderá alternar-se. É como se o tempo do processador fosse dividido em "fatias": num momento, ele atende um programa; depois, atende outro programa.

Aquilo que se designa por multitarefa é a possibilidade oferecida pelo SO de gerenciar o funcionamento do processador e da memória, de forma que vários programas possam, aparentemente, funcionar simultaneamente. São exemplos de SO multitarefa: Unix, Novell, VM, Windows NT, Windows XP e Windows 2000.

As principais características de um SO moderno são:

- interface extremamente amigável com usuário final
- recurso Plug and Play
- suporte a rede e a Internet
- aproveitamento máximo dos recursos de hardware disponíveis
- compatibilidade com programas feitos para versões anteriores do mesmo SO e com programas destinados para outros SOs
- disponha de segurança aos dados e privacidade ao usuário
- multitarefa preemptiva
- multithreading
- não vulnerável a crash, travamentos ou deadlock

## Programação por eventos

Um evento é uma resposta de um objeto, a partir de um código escrito previamente.

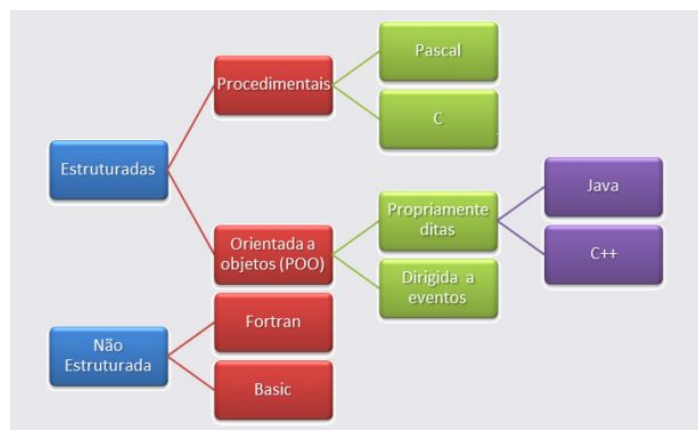
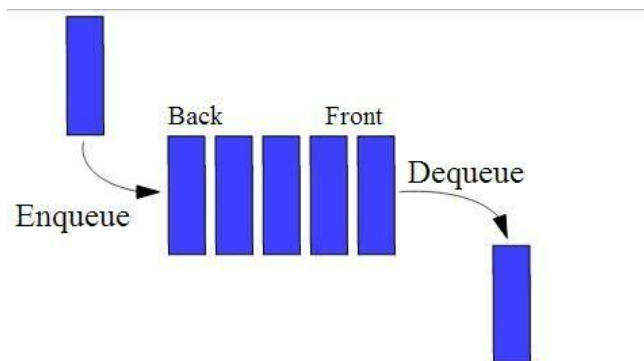
Na programação orientada a eventos, uma rotina especializada em monitorar os eventos avisa o código especializado em responder a um determinado evento que esperava ocorrer; e então o código recém avisado responde ao evento.

Resumindo:

A programação por eventos orienta os programas através de indicações externas. Este tipo de programação é normalmente utilizada no desenvolvimento de sistemas de interface de utilizador.

Programação por Queues- Orienta os programas por uma ordem sequencial de execução de rotinas, funções ou processos.

Uma Queue é um tipo abstrato de dados em que o primeiro elemento adicionado à fila será o primeiro a ser removido. Uma fila é um exemplo de uma estrutura de dados linear. Filas são comuns na programação onde são implementadas as estruturas de dados juntamente com rotinas de acesso.



## API- Interface para programação de aplicação

A sigla API significa Application Programming Interface, em Português, Interface de Programação de Aplicações.

Esta permite a integração de sistemas, apresentando vantagens como a proteção e segurança dos dados, otimização da performance da plataforma, automatização dos processos, facilidade na troca de informação entre sistemas com diferentes linguagens de programação e a própria monetização da API (cobrando, por exemplo, para que os utilizadores possam aceder a um determinado tipo de informação).

Esta interface agrega um conjunto de padrões, de rotinas e de funções de programação definidos por intermédio de um software que possibilita o uso das funcionalidades das aplicações para uso dos serviços, sem necessidade do seu envolvimento em processos demasiado complexos de implementação.

Resumidamente, uma API é um conector/interface que faz a interligação entre diferentes aplicações/sistemas com linguagens de programação distintas, de forma célere e com toda a segurança necessária. Estas podem ser utilizadas nos mais diversificados tipos de negócio e nos mais diferenciados contextos.

### Vantagens de usar API's

- Segurança: Proporcionam uma troca de informações muito fiável e segura. No seu desenvolvimento, os programadores implementam todos os mecanismos necessários para assegurar a segurança a informações sensíveis. Para isso, entre outras tecnologias, recorrem a [SSL/TLS](#), OAuth, AppToken, JSON, XML Threat Protection, etc. para que a proteção dos dados cedidos não seja comprometida.
- Websites/Ecommerce: No desenvolvimento de um website ou de uma loja online irá necessitar de implementar determinadas funcionalidades para as quais precisará de recorrer a APIs.
  - [Google Recaptcha](#);
  - [Google Maps](#);
  - Google Shopping ;
  - Integração dos Métodos de Pagamento;
- Social Login: Provavelmente, ao aceder a determinados websites ou aplicações já se deparou com a possibilidade de efetuar o login via Facebook (e outras redes

sociais) ou Gmail. Essa possibilidade é conseguida através de uma API implementada para o efeito, permitindo a otimização e a personalização da experiência do utilizador nos diferentes ambientes digitais.

## Conclusão

Este trabalho visava a introdução do aluno a conceitos avançadas de programação, dando destaque a alguns elementos que foram abordados no corpo do trabalho. acredito que o objetivo foi alcançado, tendo em vista que em uma breve leitura o aluno pode ser introduzido a diversos conceitos que podem ser aprofundados, dependendo do interesse e a necessidade.

O tema é importante porque prepara o estudante para um ambiente completamente novo dentro da programação. Antes era uma programação estruturada e agora estamos entrando em uma linguagem orientada a objetos, classes, segurança, API's e outros.

## Webgrafia

<https://kids.pplware.sapo.pt/o-meu-computador/o-que-e-um-sistema-operativo/>

<https://sites.google.com/site/programacaobasicaafware/conceito-de-janela>

<https://canaltech.com.br/produtos/O-que-e-GUI/>

<https://www.infonova.com.br/artigo/o-que-e-a-memoria-virtual/>

<http://alexandresmc campos.adm.br/fiec/so01.htm>

<https://sites.google.com/site/programacaobasicaafware/programacao-por-eventos-e-queues>

<https://pplware.sapo.pt/high-tech/sabe-o-que-e-uma-api-application-programming-interface/>