

## 1.1 Abstração de Recursos

A abstração de recursos é um dos **principais objetivos** dos sistemas operacionais, além da gerência desses recursos.

Para compreender como é **importante para os desenvolvedores** de aplicações, vale ressaltar que a abstração de recursos, visa **prover interfaces de acesso aos dispositivos** específicos de cada hardware no qual o sistema operacional está instalado.

A abstração de recursos **garante que o acesso aos dispositivos do hardware** por meio do Sistema Operacional, se faça de forma **independente desse hardware**, derivando interfaces homogêneas para dispositivos com tecnologias diferentes e diversas.

A abstração pode ser melhor compreendida se utilizarmos o modelo de camadas de um Sistema Operacional em sistema computacionais.

Um sistema computacional pode ser composto pelos seguintes recursos:

- um ou mais processadores
- memória principal
- discos, impressoras, monitor de vídeo, teclado, interfaces de redes, dentre outros dispositivos de E/S.

### **Quem abstrai esses recursos físicos para serem manipulados?**

É o Sistema Operacional, provendo suporte para as aplicações, o usuário e o hardware.

Sendo assim, há a necessidade de um tradutor genérico que consiga facilitar o interfaceamento desses recursos.

Por apresentar uma **alta complexidade** no acesso aos recursos de hardware, (físicos) para a manipulação de um programa, uma das funções dos sistemas operacionais (SO) é a **abstração de recursos fornecendo um interfaceamento mais amigável e homogêneo ao usuário**. Permitindo também o acesso independente aos dispositivos do hardware.

Para ilustrar seguimos com a operação de abertura de um arquivo (open/read) em um disco rígido“

1. verificar se os parâmetros informados estão corretos (nome do arquivo, identificador do disco, buffer de leitura etc.);
2. verificar se o disco está disponível;
3. ligar o motor do disco e aguardar atingir a velocidade de rotação correta;
4. posicionar a cabeça de leitura sobre a trilha onde está a tabela de diretório;
5. ler a tabela de diretório e localizar o arquivo ou subdiretório desejado;
6. mover a cabeça de leitura para a posição do bloco inicial do arquivo;
7. ler o bloco inicial do arquivo e depositá-lo em um buffer de memória.” (MAZIERO, 2019. p18).

## a) Prática

Vamos verificar por meio do WPS as informações do sistema operacional, versão e fabricante.

Utilize o comando **systeminfo**

```
PS C:\Users\Public\Documents> systeminfo

Nome do host: DESKTOP-00I5LD3
Nome do sistema operacional: Microsoft Windows 10 Education
Versão do sistema operacional: 10.0.17763 N/A compilado 17763
Fabricante do sistema operacional: Microsoft Corporation
Configuração do SO: Estação de trabalho autônoma
Tipo de compilado do sistema operacional: Multiprocessor Free
Proprietário registrado: Marise Miranda
Organização registrada:
Identificação do produto: 00328-00096-81061-AA700
Data da instalação original: 10/01/2020, 09:43:02
Tempo de inicialização do sistema: 17/01/2020, 16:11:38
Fabricante do sistema: Hewlett-Packard
Modelo do sistema: HP ProBook 4430s
Tipo de sistema: x64-based PC
Processador(es): 1 processador(es) instalado(s).
[01]: Intel64 Family 6 Model 42 Stepping 7
GenuineIntel ~2501 Mhz
Versão do BIOS: Hewlett-Packard 68SRR Ver. F.23, 09/03/2012
Pasta do Windows: C:\Windows
Pasta do sistema: C:\Windows\system32
Inicializar dispositivo: \Device\HarddiskVolume1
Localidade do sistema: pt-br;Português (Brasil)
Localidade de entrada: pt-br;Português (Brasil)
Fuso horário: (UTC-03:00) Brasília
Memória física total: 8.126 MB
Memória física disponível: 2.347 MB
Memória Virtual: Tamanho Máximo: 13.307 MB
Memória Virtual: Disponível: 2.730 MB
Memória Virtual: Em Uso: 10.577 MB
Local(is) de arquivo de paginação: C:\pagefile.sys
Domínio: WORKGROUP
Servidor de Logon: \DESKTOP-00I5LD3
Hotfix(es): 8 hotfix(es) instalado(s).
[01]: KB4532937
[02]: KB4462930
[03]: KB4465065
[04]: KB4486153
[05]: KB4486172
[06]: KB4516115
[07]: KB4523204
[08]: KB4534273
Placa(s) de Rede: 2 NIC(s) instalado(s).
[01]: Realtek PCIe GBE Family Controller
Nome da conexão: Ethernet
Status: Mídia desconectada
[02]: Qualcomm Atheros AR9285 802.11b/g/n
WiFi Adapter
Nome da conexão: Wi-Fi
DHCP ativado: Sim
Servidor DHCP: 10.1.2.2
Endereço(es) IP
[01]: 10.1.2.93
[02]: fe80::c949:e2ca:4f8a:9cd8
Requisitos do Hyper-V:
Extensão de Modo de Monitor VM: Sim
Virtualização Habilitada no Firmware: Sim
Conversão de Endereços de Segundo Nível:
Sim
Prevenção de Execução de Dados Disponível:
Sim
```

Você pode dispor de **informação** rápida sobre a versão do Sistema Operacional por meio do comando Get-WmiObject

O commandlet Get-WmiObject obtém instâncias das classes WMI (Instrumentação de Gerenciamento do Windows) ou informações sobre as classes disponíveis.

Aplique o cmdlet a seguir

```
Get-WmiObject -Class Win32_OperatingSystem
```

```
SystemDirectory : C:\windows\system32
Organization    :
BuildNumber     : 17763
RegisteredUser  : Marise Miranda
SerialNumber    : 00328-00096-81061-AA700
Version         : 10.0.17763
```

O retorno desse comando mostra a organização do sistema de diretórios, o número da compilação do kernel relativo aquela versão o registro do usuário da máquina, o número serial do produto SO e a versão.

Podemos afirmar que este simples comando pode auxiliar em uma **auditoria** sobre **sistemas operacionais não oficiais e oficial**.

Orientando a organização a regularizar suas versões de ambiente físico da empresa.

O commandlet Get-WmiObject obtém instâncias das classes **WMI (Instrumentação de Gerenciamento do Windows)** ou informações sobre as classes disponíveis.

```
Get-WmiObject
```

Só que é necessário informar qual o sistema de diretórios se é de 32 bits ou 64 bits.

```
PS C:\Users\Public\Documents> Get-WmiObject
cmdlet Get-WmiObject na posição de comando 1 do pipeline
Forneça valores para os seguintes parâmetros:
Class:
```

Ao completar a classe do sistema de diretórios é necessário informar Win32\_OperatingSystem.

```
Class: Win32_OperatingSystem
```

A resposta do WPS a esse comando

```
SystemDirectory : C:\windows\system32
Organization    :
BuildNumber     : 17763
RegisteredUser  : Marise Miranda
SerialNumber    : 00328-00096-81061-AA700
Version         : 10.0.17763
```

Observe que o **System32** significa que o sistema de diretório do Windows é de **32 bits**.

Todas as DLL (que são os drivers e bibliotecas do sistema associados a algum hardware) são "bitados" em **32 bit** no seu executável.

Mas e se o meu computador for de **64 bits**?

Tem certeza que é?

Pois bem, o **system 32** é um padrão do Windows.

Caso você tenha um computador de **64bits**, por projeto você terá o **Windows on Windows**, por isso o Sistema de diretórios será o **SysWoW64**.

Vamos encontrar esse diretório

**SysWoW64**

Busque o diretório raiz Windows pelo WPS, dê um dir e encontre o **SysWoW64**

```
PS C:\> cd windows
PS C:\windows> dir

Diretório: C:\windows

Mode                LastWriteTime         Length Name
----                -
d-----          15/09/2018         04:33      addins
d-----          13/01/2020         10:35      appcompat
d-----          14/01/2020         14:59      apppatch
d-----          29/01/2020         18:57      AppReadiness
d-r---          13/01/2020         15:28      assembly
d-----          16/01/2020         10:28      bcastdvr
d-----          15/09/2018         04:33      Boot
d-----          15/09/2018         04:33      Branding
d-----          16/01/2020         12:39      CbsTemp
d-----          15/09/2018         14:06      Containers
d-----          10/01/2020          09:44      CSC
d-----          15/09/2018         04:33      Cursors
d-----          13/01/2020         11:11      debug
```

Entre nesse diretório e examine as DLLs ali incluídas, são várias.

Mas afinal o sistema de diretório do SO da minha máquina é de **32** ou **64 bits**? **O seu System é de 32 bits. Porém nesse sistema de diretório está o SysWoW64, que conterà todas as DLLs de 64 bits.**

Vamos examinar o Processador do computador via WPSISE

get-WMIObject -Class win32\_Processor

```
PS C:\windows\SysWOW64> Get-WMIObject -Class Win32_Processor
```

Retorna

```
Caption           : Intel64 Family 6 Model 42 Stepping 7
DeviceID          : CPU0
Manufacturer      : GenuineIntel
MaxClockSpeed     : 2501
Name              : Intel(R) Core(TM) i5-2450M CPU @ 2.50GHz
SocketDesignation : CPU_1
```

```
MaxClockSpeed     : 2501
```

Isto quer dizer um clock máximo de 2.5 GHz

Vá no portal da [CPU Benchmarks](#) e encontre as características do processador da sua máquina.

CPU cores	CPU brand	CPU name	CPU stepping	CPU speed (MHz)	Threads	Total time (sec.) <a href="#">sort</a>	Rating time (*)	Rating calc (*)
4	GenuineIntel	Intel(R) Core(TM) i5-2450M CPU @ 2.50GHz	Intel64 Family 6 Model 42 Stepping 7	2,494	4	7.893	16.523x	7.005x

Portanto, este processador é de 64 bits, porém para este Sistema Operacional algumas recomendações sobre o hardware são necessárias. O SO Windows de 64 bits é recomendado para processadores de 64 bits e pelo menos 4 GB de memória Ram.



## SO Windows 64 bits ou 32 bits by Microsoft

Computadores que executam as versões de **64 bits** do Windows geralmente possuem **mais recursos**, como a capacidade de **processamento** e **memória**, que seus **antecessores de 32 bits**.

Além disso, os aplicativos de 64 bits podem acessar **mais memória** do que aplicativos de 32 bits (até 18,4 milhões de Petabytes).

Portanto, se seus cenários incluem arquivos grandes e/ou você vai trabalhar com grandes conjuntos de dados e o seu computador está executando a versão de **64 bits** do Windows, 64 bits é a escolha certa quando:

- **Trabalhar com tipos ou conjuntos de dados muito grandes**
- **Trabalhar com imagens extremamente grandes, vídeos ou animações**

(<https://support.office.com/pt-br/article/escolha-entre-uma-vers%C3%A3o-de-64-bits-ou-de-32-bits-do-office-2dee7807-8f95-4d0c-b5fe-6c6f49b8d261?ui=pt-BR&rs=pt-BR&ad=BR>)

Vamos observar agora as características do SO instalado LEGALMENTE de outra maneira bem simplificada.

**Winows**  + **R**, digite **winver** na caixa **Abrir** e selecione **OK**




Vamos pegar mais algumas informações do sistema com o **Get-ComputerInfo**

```
PS C:\Users\Public\Documents> Get-ComputerInfo

WindowsBuildLabEx           : 17763.1.amd64fre.rs5_release.180914-1434
WindowsCurrentVersion       : 6.3
WindowsEditionId            : Education
WindowsInstallationType     : Client
WindowsInstallDateFromRegistry : 10/01/2020 12:43:02
WindowsProductId            : 00328-00096-81061-AA700
WindowsProductName          : Windows 10 Education
WindowsRegisteredOrganization : Marise Miranda
WindowsRegisteredOwner      : C:\windows
WindowsSystemRoot            : 1809
BiosCharacteristics          :
BiosBIOSVersion              :
BiosBuildNumber              :
BiosCaption                  :
BiosCodeSet                  :
BiosCurrentLanguage          :
BiosDescription              :
BiosEmbeddedControllerMajorVersion :
BiosEmbeddedControllerMinorVersion :
BiosFirmwareType             :
BiosIdentificationCode       :
BiosInstallableLanguages     :
BiosInstallDate              :
BiosLanguageEdition          :
BiosListOfLanguages          :
BiosManufacturer             :
BiosName                     :
BiosOtherTargetOS            :
BiosPrimaryBIOS              :
BiosReleaseDate              :
```

```
S C:\Users\Public\Documents> Get-ComputerInfo -Property windows*

WindowsBuildLabEx           : 17763.1.amd64fre.rs5_release.180914-1434
WindowsCurrentVersion       : 6.3
WindowsEditionId            : Education
WindowsInstallationType     : Client
WindowsInstallDateFromRegistry : 10/01/2020 12:43:02
WindowsProductId            : 00328-00096-81061-AA700
WindowsProductName          : Windows 10 Education
WindowsRegisteredOrganization : Marise Miranda
WindowsRegisteredOwner      : C:\windows
WindowsSystemRoot            : 1809
WindowsVersion              :
```

<p>Refresh on Memory</p> 	<p>Continuando o desenvolvimento do tema “Instrumentação de Gerenciamento do Windows (WMI)” por conta a sua “abstração sobre os recursos” de hw e sw, agora vamos verificar como alguns objetos em instâncias das classes podem auxiliar na obtenção de informação a respeito do Sistema Operacional via WPS.</p>
--	---

Utilizando a classe WMI para obter informações do sistema.

Vamos listar as classes WMI disponíveis no computador local por meio da CimClass.

O Cim contém os cmdlets que interagem com os objetos do Modelo Comum de Informação - **Common Information Model** (CIM), como o serviço do Windows Management Instrumentation (WMI).

Para listar todos os nomes de serviços de informação usaremos o

```
Get-CimClass -Namespace root/CIMV2 |
Where-Object CimClassName -like win32* |
Select-Object CimClassName
```

Retorna:

```
PS C:\windows\SysWOW64> Get-CimClass -Namespace root/CIMV2 |
Where-Object CimClassName -like Win32* |
Select-Object CimClassName

CimClassName
-----
Win32_PrivilegesStatus
Win32_JobObjectStatus
Win32_Trustee
Win32_ACE
Win32_SecurityDescriptor
Win32_ComputerSystemEvent
Win32_ComputerShutdownEvent
Win32_IP4RouteTableEvent
Win32_SystemTrace
Win32_ProcessTrace
Win32_ProcessStartTrace
Win32_ProcessStopTrace
Win32_ModuleTrace
Win32_ModuleLoadTrace
Win32_ThreadTrace
Win32_ThreadStartTrace
Win32_ThreadStopTrace
Win32_PowerManagementEvent
Win32_DeviceChangeEvent
Win32_SystemConfigurationChangeEvent
Win32_VolumeChangeEvent
Win32_CollectionStatistics
Win32_NamedJobObjectStatistics
Win32_NTLogEvent
Win32_ActiveRoute
Win32_OfflineFilesUserConfiguration
Win32_AccountSID
```



Pergunta: A Win32 é uma api com os objetos da classe Cim e não tem relação com o sistema de arquivos de 32 bits. Esta afirmação é verdadeira ou falsa?

**Resposta: A afirmação é verdadeira, o system32 ou system64 tem relação com o projeto do processador se 32 bits ou 64 bits, daí o sistema de diretórios precisa ser compatível com o projeto de hardware no qual o processador funciona. O Win32 define as classes usadas para descrever o hardware ou software disponível nos sistemas operacionais Windows e os relacionamentos entre eles.**

**O Win32 API é uma referência de programação relativa a tecnologia, hw e sw, e portanto está referenciada desde a interface do usuário, área de trabalho, gráficos e jogos, áudio e vídeo, dentre tantos outros e até segurança e identidade. Para saber mais vá no endereço <https://docs.microsoft.com/en-us/windows/win32/api/>.**



### Descrição das aplicações relacionadas à api Win32, hw e sw

Então, vamos usar alguns dessas classes para obter informações de hw e sw. Execute:

```
Get-CimInstance -ClassName Win32_Desktop
```

Retorna

SettingID	Name	ScreenSaverActive	ScreenSaverSecure	ScreenSaverTimeout
AUTORIDADE	NT\SISTEMA	False		
MARCIO SANTANA	Marcio Santana	False	False	60
.DEFAULT		False		

Por que é false no usuário????



Porque o recurso modo de proteção de tela não está ativo para esse usuário.

Se você digitar no source do Windows **"screen saver"** e ativar o modo de proteção da tela desktop que não seja nenhum, verá depois de aplicar o mesmo cmdlet que o usuário desktop estará com o recurso "true".

Agora para que possamos concluir o tema abstração do sistema operacional, digite o cmdlet abaixo:

```
(Get-ItemProperty -Path c:\windows\system32\hal.dll)
```

Mas, entender cada parte do comando:

**Get-ItemProperty** - é um cmdlet que captura as propriedades específicas do item  
**-Path** c:\windows\system32\hal.dll - é o caminho até o diretório do system32

Mas tem um detalhe com relação a **hal.dll**

O **hal (hardware abstraction layer)**, ou seja, **é a camada de abstração de hardware, fica entre o hardware físico de um computador e o software que corre nesse computador.**

A sua função é **ocultar diferenças** em hardware e, conseqüentemente, disponibilizar uma plataforma consistente para executar as aplicações.

Dica: este arquivo é o mais susceptível a ataques de vírus e em forense computacional geralmente está dll não é a oficial. Quando você vai corrigir o problema percebe que a **hal** não está registrada.

#### **Possíveis mensagens de erro em relação a esse arquivo:**

- HAL.DLL está faltando
- erro ao carregar HAL.DLL
- HAL.DLL parou de funcionar
- HAL.DLL não foi encontrado
- o ponto de entrada do procedimento HAL.DLL
- HAL.DLL não pôde ser localizado
- Violação de acesso HAL.DLL
- Não foi possível encontrar HAL.DLL
- Não foi possível registrar HAL.DLL

REFERÊNCIA BIBLIOGRÁFICA:

TANENBAUM, A. Sistema Operacionais Modernos. Tradução Jorge Ritter. 2ª Edição, São Paulo. Pearson Education do Brasil, 2009.

MACHADO, F. B. Arquitetura de Sistemas Operacionais, 4ª Ed, Rio de Janeiro. LTC, 2007.

SILBERSCHATZ, A. Sistemas Operacionais: Conceitos. 5ª Ed. São Paulo. Prentice Hall, 2000.

- REDHAT. Disponível em [www.redhat.com/topics/middleware](http://www.redhat.com/topics/middleware). Acessado em 19/12/2019.

- FERRARI, F. O Shell. Disponível em <http://www.ferrari.pro.br/home/documents/FFerrari-O-Shell-Unix.pdf>. Acessado em 19/12/2019.

<http://www.agasus.com.br/4-grandes-motivos-para-atualizar-hardware-e-sistemas-operacionais-da-empresa/>

DONDA, D. Windows Power Shell 3.0. Um Guia de Windows PowerShell desenvolvido especificamente para profissionais de infraestrutura. Todo o conteúdo está sob licença da Creative Commons Attribution 3.0 Unported License <http://bit.ly/ZnVDOD>.

Disponível em

<http://professorramos.com/Materiais/Documentos/PowerShell%20para%20IT%20Pro-%20Book.pdf>. Acessado em 19/12/2019.

LICENÇA MICROSOFT EDUCATION: Instituições de ensino credenciadas, como escolas de ensino fundamental e médio, universidades, faculdades públicas e privadas e faculdades comunitárias estaduais, poderão efetuar o download e reproduzir os Documentos para serem distribuídos em sala de aula. A distribuição fora de sala de aula exigirá permissão por escrito.