**Rede Local – Instalação**

Em informática, um servidor é um software ou um computador, com um sistema de computação centralizada que fornece serviços a uma rede de computadores, chamada de cliente.

Esses serviços podem ser de naturezas distintas, como por exemplo, ficheiros e correio eletrónico.

Esta arquitetura é chamada de modelo cliente-servidor, é utilizada em redes de médio e grande porte (com muitas máquinas) e em redes onde a questão da segurança desempenha um papel de grande importância. O termo servidor é amplamente aplicado a computadores completos, embora um servidor possa equivaler a um software ou a partes de um sistema computacional.

Os servidores podem fornecer várias funcionalidades, muitas vezes chamadas de "serviços", tais como a partilha de dados ou de recursos do sistema, entre vários clientes, ou computação desempenhada para um cliente. Um único servidor pode servir vários clientes, e um único cliente pode usar vários servidores. Um processo cliente pode ser executado no mesmo dispositivo ou pode ser conectado através de uma rede para um servidor num dispositivo diferente.

**Sistemas Operativos Servidor:**

Para que funcione uma rede cliente-servidor, é necessário que no servidor esteja instalado um sistema operativo que reconheça esse tipo de rede. Os sistemas operativos para redes cliente-servidor são:

* Unix
* Linux - Como os servidores normalmente são manipulados pela linha de comandos, a interface gráfica do usuário (GUI) não importa. Uma das melhores distribuições Linux é: Debian, pela sua alta fiabilidade, considerada a mais segura. O Debian é frequentemente utilizado para servidores por causa da sua resistência, que já foi muito testada ao longo do tempo. Possui uma gestão de pacotes muito eficiente, as ferramentas do APT, e ainda várias interfaces, tais como o GDebi.
* Solaris
* FreeBSD
* Novell Netware
* Windows NT, Windows 2000, Windows 2003, Windows Server 2008, Windows Server 2012 - Windows Server é uma família de sistemas operativos da Microsoft Windows baseado na arquitetura NT, é dirigida para o uso em servidores.

O mais recente é o Windows Server 2016, que é um sistema operativo para servidores desenvolvido pela Microsoft como parte da família de sistemas operativos Windows NT, desenvolvido simultaneamente com o Windows 10 e é o sucessor do Windows Server 2012 R2. O lançamento da primeira versão de visualização de desenvolvedor do Windows Server 2016 ocorreu no dia 01 de outubro de 2014, tendo sido a versão Technical Preview 5 a última versão beta disponibilizada para os testes públicos. A versão final do Windows Server 2016 foi lançada em 26 de setembro de 2016 na conferência Microsoft Ignite e passou a estar disponível para o público geral no dia 12 de outubro de 2016. É um sistema que utiliza a interface gráfica para uma melhor relação para o utilizador deste.

Em servidores, o sistema Unix e os sistemas baseados neste (como o Linux e o Solaris) são os sistemas mais utilizados para aplicações como Firewall e servidor web, ao passo que o sistema Windows, é mais utilizado para a Gestão de Utilizadores e serviços pela facilidade de operação e manutenção.

**Sistema ficheiros servidor (raid,etc):**

Um sistema de ficheiro é uma forma de criar uma estrutura lógica de acesso a dados numa partição. Sendo assim, também é importante referir que nunca poderá ter dois ou mais tipos de sistemas de ficheiros (formatos) numa mesma partição. Possui várias camadas:

* Programas/aplicativos
* Sistemas de arquivos lógicos
* Módulo de organização de arquivos
* Sistema básico de arquivos
* Controle de E/S
* Dispositivos

Cada camada possui características/funcionalidades específicas, e estão por ordem hierárquica.

Existem diversos tipos de sistemas de ficheiros e estão presentes nas diversas plataformas.

* Apple Macintosh (Mac OS)
* HFS
* HFS+
* APFS
* UNIX (FreeBSD, OpenBSD, Linux, Solaris, Red Hat, Android, etc.)
  + UFS
  + Ext
  + Ext2
  + Ext3
  + Ext4
  + SWAP
  + Reiser
  + HPFS
  + JFS
  + XFS
  + ZFS
* IBM (AIX, OS/2)
* JFS (AIX Version 3.1 ou superior, OS/2 Warp)
* HPFS - High Performance File System
* MS-DOS/Microsoft Windows
* FAT 12 - Microsoft BASIC Disk - MSDOS 4.0
* FAT 16 ou FAT - DOS 4.0 ou superior / Windows 1.X ou superior (1.x, 2.x, 3.x, 95, 98, ME, 2000, XP,...)
* FAT 32 - MS-DOS 7.1 e 8.0 / Windows 95 (versão OSR2!), ou superior (95 OSR2, 98, ME, NT, 2000, XP...)
* ExFAT - FAT Estendido, também conhecido como FAT64, Windows XP ou superior ( Vista, 7...)
* NTFS - Windows NT ou superior (NT, 2000, XP, 2003 Server,...)

Todos nós sabemos que a nossa informação digital, mantida nos nossos sistemas, é demasiado importante e por isso devemos ter os melhores recursos para salvaguardar a mesma. Os serviços de armazenamento na cloud vieram ajudar neste processo de salvaguarda de informação mas, em DataCenters e até para o próprio utilizador doméstico, é igualmente importante que se adotem sistemas de discos tolerantes a falhas.

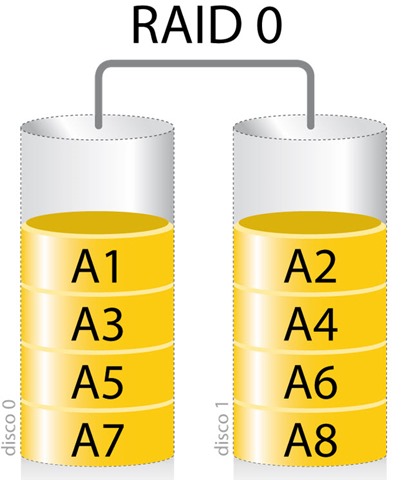
Os discos são hoje em dia um dos principais dispositivos de armazenamento de informação. No entanto, todos nós sabemos que a vida de um disco rígido não é eterna e nesse sentido é importante que exista backups da informação mas também que se usem sistemas tolerantes a falhas, para salvaguardar das falhas totais de um disco.

Uma das principais tecnologia de gestão de discos é o RAID.

O RAID (Redundant Array of Independent Disks/Conjunto Redundante de Discos Independentes), tal como o nome sugere é uma tecnologia que tem a capacidade de combinar vários discos, com o objetivo de garantir performance e essencialmente segurança contra falhas num ou vários discos. Existem vários níveis de RAID, que definem como é que vários discos podem ser utilizados em conjunto. Assim, o nível de redundância depende essencialmente da configuração RAID que foi definida.

**RAID 0 – Data Striping**

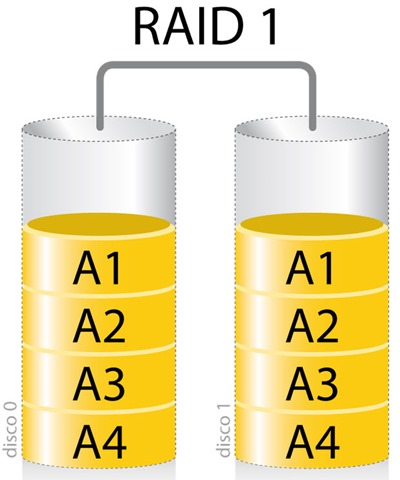
No caso de configurarmos RAID 0, a informação é segmentada e distribuída por vários discos. Neste tipo de configuração não existe qualquer tipo de redundância (se um disco falhar, os dados são perdidos) mas, como vantagem, este nível de raid garante alta performance tanto na leitura como escrita em disco.



**RAID 1 – Data Mirroring**

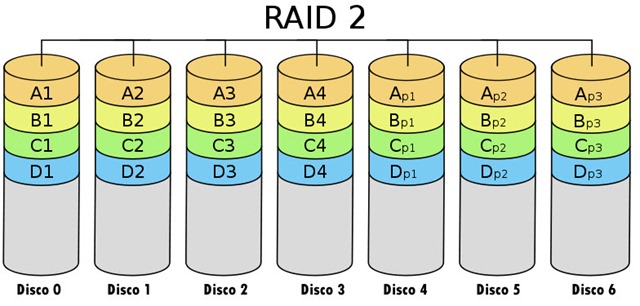
O RAID 1 é um método de armazenamento que funciona estilo um “espelho”. Considerando, por exemplo, que temos dois discos, toda a informação guardada no primeiro disco é igualmente guardada (espelhada) no segundo. Na prática é como se o sistema apenas tivesse apenas um disco, sendo que o segundo é uma cópia exata do primeiro (ou seja, quando a informação é guardada no primeiro disco é simultaneamente guardada no segundo disco), podendo este ser usado quando o primeiro falhe (por exemplo, quando o primeiro disco avaria). De referir que para criar volumes espelhados é necessário ter no mínimo 2 discos.

Ao contrário do RAID 0, o RAID 1 garante a redundância da informação. Como ponto fraco há a destacar a performance de escrita, uma vez que é necessário escrever a mesma informação em ambos os discos. De referir que o RAID 1 é provavelmente um dos modelos de RAID mais usados.



**RAID 2 – Data Striping com deteção de erros**

O RAID 2 é em parte semelhante ao RAID 0, só que distribui a informação por vários discos, sendo essa distribuição feita ao nível do bit e recorrendo a um processo de deteção e correção de erros do tipo ECC (Error Correcting Code). Este nível oferece um baixo desempenho mas um elevado nível de segurança da informação. Atualmente este modelo quase já não é usado, uma vez que os discos já vêm com o sistema de deteção e correção de erros.



**Hardware do Servidor:**

****

Os servidores dedicados, que possuem uma alta requisição de dados por parte dos clientes e que atuam em aplicações críticas utilizam hardware específico para servidores. Já os servidores que não possuam essas atuações podem utilizar o hardware de um computador comum.

Muitos servidores baseiam-se nas entradas e saídas de dados, o que implica interfaces de entrada e saída e discos rígidos de alto desempenho e confiabilidade. O tipo de disco rígido mais utilizado possui o padrão SCSI, que permite a interligação de vários periféricos, dispostos em RAID (como referido anteriormente).

Devido a operar com muitas entradas e saídas de informações, os servidores necessitam de processadores de alta velocidade, algumas vezes alguns servidores são multi-processador, ou seja, possuem mais de um processador. Eles também têm disponível uma grande quantidade de memória RAM, sendo geralmente usada para o caching de dados.

Por terem que operar por muito tempo (frequentemente de maneira ininterrupta), alguns servidores são ligados a geradores elétricos. Já os outros utilizam sistemas de alimentação (por exemplo, o UPS) que continuam a alimentar o servidor caso haja alguma queda de tensão.

E, por trabalharem durante longos intervalos de tempo, e devido à existência de um ou mais processadores de alta velocidade, os servidores precisam de um eficiente sistema de dissipação de calor, o que implica coolers mais caros, mais barulhentos, porém de maior eficiência e confiabilidade.

Existem outro hardware específico para servidor, principalmente placas, do tipo hot swapping, que permitem a troca destes enquanto o computador está ligado, o que é primordial para que a rede continue a funcionar.

Numa estrutura não crítica, um computador comum pode ser usado como servidor, já que o tamanho da rede não importa; por exemplo: uma empresa com três instrutores on-line na Internet tem 3 computadores e um deles é o servidor de acesso à Internet. Se este servidor falha o negócio da empresa está parado.

Prevendo esse tipo de necessidade, os fabricantes de componentes de computadores desenvolvem placas mais robustas, aplicam uma engenharia mais elaborada de ventilação, redundância de itens e capacidade de expansão ampliada, para que o servidor possa garantir a disponibilidade do serviço e a confiabilidade no mesmo.

Normalmente a preocupação em desenvolver servidores fica centrada nos grandes fabricantes do mercado, que possuem equipas preparadas e laboratórios com esse fim.



**Unidade de Backups:**

A unidade de backups serve para que um servidor fique conectado diretamente a esta para que não ocorra nenhum tipo de perda de informação, fazendo backups programados pela equipa de TI de uma organização, sendo assim um componente fundamental para uma empresa de médio e grande porte para uma maior segurança em relação aos dados armazenados no servidor. É parecida com um servidor.



**Protocolos e serviços de rede (modelo osi,tcp/ip):**

Podemos pensar em rede de computadores como diversas máquinas interligadas fisicamente entre si onde os seus utilizadores promovem a troca de informação do seu interesse. Entretanto, uma rede não pode ser bem estabelecida considerando apenas o hardware como preocupação principal como nas primeiras redes, atualmente o software é considerado uma das partes mais importantes na conceção de novas tecnologias de redes de computadores.

Protocolo é o conjunto de regras sobre o modo como se dará a comunicação entre as partes envolvidas.

Protocolo é a "língua" dos computadores, ou seja, uma espécie de idioma que segue normas e padrões determinados. É através dos protocolos que é possível a comunicação entre um ou mais computadores. Os protocolos de rede nasceram da necessidade de conectar equipamentos de fornecedores distintos, executando sistemas distintos, sem ter que escrever a cada caso programas específicos. Ambos os computadores devem estar configurados com os mesmos parâmetros e obedecer aos mesmos padrões para que a comunicação possa ser realizada sem erros. Existem diversos tipos de protocolos de rede, variando de acordo com o serviço a ser utilizado. De maneira geral há dois tipos de protocolos: Abertos e Proprietários ou Específicos. Os protocolos Abertos são os protocolos padrões da internet. Este podem comunicar com outros protocolos que utilizam o mesmo padrão de protocolo. Um exemplo seria o TCP/IP, pois ele pode comunicar com várias plataformas como Windows, Linux, Mac e outros. Já os protocolos Proprietários são feitos para ambiente específicos (daí o seu nome), pois ele apenas pode comunicar com uma plataforma padrão. Exemplos desse tipo de protocolo: IPX/SPX, NETBIOS e outros. São exemplos de protocolos de rede: IP (Internet Protocol), DHCP (Dynamic Host Configuration Protocol), TCP (Transmission Control Protocol), HTTP (Hypertext Transfer Protocol), FTP (File Transfer Protocol), Telnet (Telnet Remote Protocol), SSH (SSH Remote Protocol), POP3 (Post Office Protocol 3), SMTP (Simple Mail Transfer Protocol), IMAP (Internet Message Access Protocol).

Uma das funções dos protocolos é pegar nos dados que serão transmitidos pela rede, dividi-los em pequenos pedaços chamados pacotes, na qual dentro de cada pacote há informações de endereçamento que informam a origem e o destino do pacote. É através do protocolo que as fases de estabelecimento, controlo, tráfego e encerramento, componentes da troca de informações são sistematizadas.

As funções dos protocolos são as seguintes:

* Endereçamento: especificação clara do ponto de destino da mensagem.
* Numeração e sequência: individualização de cada mensagem, através do número sequencial.
* Estabelecimento da conexão: estabelecimento de um canal lógico fechado entre a fonte e o destino.
* Confirmação de receção: confirmação do destinatário, com ou sem erro, após cada segmento de mensagem.
* Controlo de erro: deteção e correção de erros.
* Retransmissão: repetição da mensagem a cada receção de mensagem.
* Conversão de código: adequação do código às características do destinatário.
* Controlo de fluxo: manutenção de fluxos compatíveis com os recursos disponíveis.

O protocolo mais usado é o TCP/IP, que é um conjunto de protocolos de comunicação entre computadores em rede. O seu nome vem de dois protocolos: o TCP (Transmission Control Protocol - Protocolo de Controlo de Transmissão) e o IP (Internet Protocol - Protocolo de Internet, ou ainda, protocolo de interconexão). O conjunto de protocolos pode ser visto como um modelo de camadas (Modelo OSI), onde cada camada é responsável por um grupo de tarefas, fornecendo um conjunto de serviços bem definidos para o protocolo da camada superior. As camadas mais altas, estão logicamente mais perto do utilizador (chamada camada de aplicação) e lidam com os dados mais abstratos, confiando em protocolos de camadas mais baixas para tarefas de menor nível de abstração.

O TCP/IP foi desenvolvido em 1969 pelo U.S. Department of Defense Advanced Research Projects Agency, como um recurso para um projeto experimental chamado de ARPANET (Advanced Research Project Agency Network) para preencher a necessidade de comunicação entre um grande número de sistemas de computadores e várias organizações militares dispersas. O objetivo do projeto era disponibilizar os links (vínculos) de comunicação a alta velocidade, utilizando as redes de comutação de pacotes. O protocolo deveria ser capaz de identificar e encontrar a melhor rota possível entre dois sites (locais), além de ser capaz de procurar rotas alternativas para chegar ao destino, caso qualquer uma das rotas tivesse sido destruída. O objetivo principal da elaboração do TCP/IP foi na época, encontrar um protocolo que pudesse tentar, de todas as formas, uma comunicação caso ocorresse uma guerra nuclear. A partir de 1972 o projeto ARPANET começou a crescer numa comunidade internacional e que se transformou no que hoje conhecemos como Internet. Em 1983 ficou definido que todos os computadores conectados ao ARPANET passariam a utilizar o TCP/IP. No final dos anos 1980 a Fundação Nacional de Ciências em Washington, D.C, começou a construir o NSFNET, um backbone para um supercomputador que serviria para interconectar diferentes comunidades de pesquisa e também os computadores da ARPANET. Em 1990 o NSFNET tornou-se no backbone das redes para a Internet, padronizando definitivamente o TCP/IP.

Os benefícios deste protocolo são:

1. Padronização: um padrão, um protocolo roteável que é o mais completo e aceite protocolo disponível atualmente. Todos os sistemas operativos modernos oferecem suporte para o TCP/IP e a maioria das grandes redes baseiam-se em TCP/IP para a maior parte do seu tráfego.
2. Interconectividade: uma tecnologia para conectar sistemas não similares. Muitos utilitários padrões de conexão estão disponíveis para acessar e transferir dados entre esses sistemas não similares, incluindo FTP (File Transfer Protocol) e Telnet (Terminal Emulation Protocol).
3. Roteamento: permite e habilita as tecnologias mais antigas e as novas a conectarem-se à Internet. Trabalha com protocolos de linha como PPP (Point to Point Protocol) permitindo uma conexão remota a partir da linha discada ou dedicada. Trabalha como os mecanismos IPCs e os interfaces mais utilizados pelos sistemas operativos, como Windows sockets e NetBIOS.
4. Protocolo Robusto: escalável, multiplataforma, com estrutura para ser utilizada em sistemas operativos cliente/servidor, permitindo a utilização de aplicações desse porte entre dois pontos distantes.
5. Internet: é através da suíte de protocolos TCP/IP que obtemos acesso a Internet. As redes locais distribuem servidores de acesso a Internet (proxy servers) e os hosts locais conectam-se a estes servidores para obter o acesso à Internet. Este acesso só pode ser conseguido se os computadores estiverem configurados para utilizar TCP/IP.

Existe a discussão de como mapear o modelo TCP/IP convencionalmente de 4 camadas dentro do modelo OSI que possui o padrão de 7 camadas. Como os modelos TCP/IP e OSI não combinam exatamente, não existe uma única resposta para esta questão.

Além do mais, o modelo OSI não é realmente rico o suficiente nas camadas mais baixas para capturar a verdadeira divisão de camadas; é necessário uma camada extra (a camada internet) entre as camadas de transporte e de rede. Protocolos específicos para um tipo de rede que funcionam em cima da estrutura de hardware básica precisam estar na camada de rede. Exemplos desse tipo de protocolo são ARP e o Spanning Tree Protocol (usado para manter pontes de rede redundantes em "espera" enquanto elas são necessárias). Entretanto, eles são protocolos locais e operam debaixo da funcionalidade internet. Reconhecidamente, colocar ambos os grupos (sem mencionar protocolos que são logicamente parte da camada internet, mas que funcionam em cima de um protocolo internet, como ICMP) na mesma camada pode ser um tanto confuso, mas o modelo OSI não é complexo o suficiente para apresentar algo melhor.

O Modelo OSI (do inglês Open System Interconnection) é um modelo de rede de computador referência da ISO dividido em camadas de funções, criado em 1971 e formalizado em 1983, com o objetivo de ser um padrão, para protocolos de comunicação entre os mais diversos sistemas numa rede local (Ethernet), garantindo a comunicação entre dois sistemas computacionais (end-to-end).

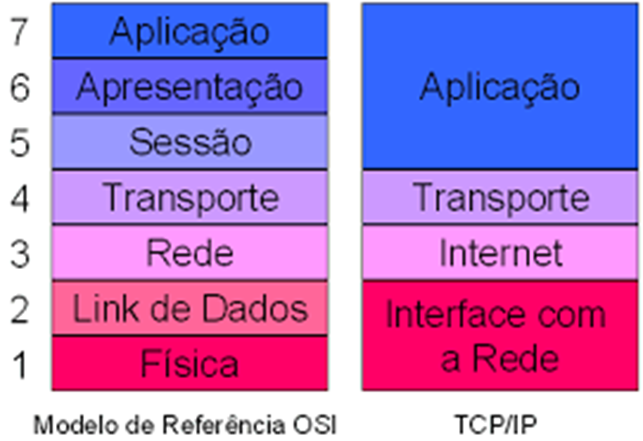
Este modelo divide as redes de computadores em 7 camadas, de forma a obterem-se camadas de abstração. Cada protocolo implementa uma funcionalidade assinalada a uma determinada camada.

Segundo Tanenbaum o Modelo OSI não é uma arquitetura de redes, pois não especifica os serviços e os protocolos exatos que devem ser usados em cada camada. Ele apenas informa o que cada camada deve fazer.

O Modelo OSI permite comunicação entre máquinas heterogéneas e define as diretivas genéricas para a construção de redes de computadores (seja de curta, média ou longa distância) independentemente da tecnologia utilizada.

As diferenças entre o TCP/IP e o OSI são as seguintes:

* O TCP/IP combina os aspetos das camadas de apresentação e de sessão dentro da sua camada de aplicação;
* O TCP/IP combina as duas camadas, física e de enlace, do OSI numa só camada;
* O TCP/IP parece ser mais simples por ter menos camadas;
* Os protocolos do TCP/IP são os padrões em torno dos quais a Internet se desenvolveu, portanto o modelo TCP/IP ganha credibilidade apenas por causa dos seus protocolos. Em contraste, nenhuma rede foi criada em torno de protocolos específicos relacionados ao OSI, embora todos usem o modelo OSI para guiar o seu raciocínio.



**Função do servidor na rede:**

Uma função do servidor é um conjunto de programas de software que, quando os programas são instalados e corretamente configurados, permite que um computador execute uma função específica para vários utilizadores ou outros computadores numa rede. Em termos gerais, as funções partilham as seguintes características.

* Descrevem a função principal, finalidade, ou utilização de um computador. Um determinado computador pode estar dedicado a executar uma única função, que seja muito utilizada por toda a empresa, ou pode executar múltiplas funções, se cada uma delas for pouco utilizada na empresa.
* Proporcionam aos utilizadores de toda uma organização acesso a recursos geridos por outros computadores, tais como Web sites, impressoras ou ficheiros armazenados noutros computadores.
* Normalmente, incluem bases de dados próprias que podem colocar em fila pedidos do utilizador ou do computador, ou gravar informação sob‎‎re utilizadores e computadores da rede que se relacionem com a função. Por exemplo, os serviços do domínio Active Directory incluem uma base de dados para armazenar os nomes e as relações hierárquicas de todos os computadores numa rede.
* Assim que são instaladas e configuradas corretamente, as funções ficam a funcionar automaticamente. Tal permite que os computadores nos quais estão instaladas executem determinadas tarefas com supervisão e comandos de utilizador limitados.

**Serviços de função:**

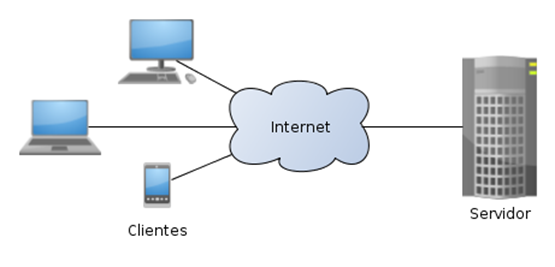
Os serviços de função são programas de software que fornecem a funcionalidade de uma função. Ao instalar uma função, é possível escolher quais os serviços de função que essa função fornece aos outros utilizadores e computadores na empresa. Algumas funções, tais como o Servidor DNS, têm apenas uma função e, por isso, não têm serviços de função disponíveis. Outras funções, tais como os Serviços de Ambiente de Trabalho Remoto, têm vários serviços de função que podem ser instalados, consoante as necessidades de computação remota da empresa.

Pode-se considerar uma função como um agrupamento de serviços de função complementares, estreitamente relacionados, relativamente aos quais, a maior parte das vezes, a instalação da função significa a instalação de um ou mais dos seus serviços.

**Funcionalidades:**

As funcionalidades são programas de software que, embora não sejam diretamente partes de funções, podem suportar ou aumentar a funcionalidade de uma ou mais funções, ou melhorar a funcionalidade do servidor, independentemente das funções instaladas. Por exemplo, a funcionalidade Clustering de Activação Pós-falha aumenta a funcionalidade de outras funções, tais como os serviços de ficheiros e o Servidor DHCP, pois permite que adiram a clusters de servidores para maior redundância e melhor desempenho. Outra funcionalidade, o Cliente Telnet, permite comunicar de forma remota com um servidor telnet através de uma ligação de rede, o que melhora as opções de comunicação do servidor.

**Contas de cliente no servidor:**

****

O modelo cliente-servidor (em inglês client/server model), em computação, é uma estrutura de aplicação distribuída que distribui as tarefas e cargas de trabalho entre os fornecedores de um recurso ou serviço, designados como servidores, e os requerentes dos serviços, designados como clientes.

Geralmente os clientes e os servidores comunicam através de uma rede de computadores em computadores distintos, mas tanto o cliente quanto o servidor podem residir no mesmo computador.

Um servidor é um host que está executando um ou mais serviços ou programas que compartilham recursos com os clientes. Um cliente não compartilha qualquer dos seus recursos, mas solicita um conteúdo ou função do servidor. Os clientes iniciam sessões de comunicação com os servidores que aguardam requisições de entrada.

O modelo cliente-servidor foi desenvolvido na Xerox PARC durante os anos 70. Este modelo é atualmente o predominante nas redes informáticas. O Email, a World Wide Web e as redes de impressão são exemplos comuns deste modelo.

A característica do modelo cliente-servidor, descreve a relação de programas numa aplicação. O componente de servidor fornece uma função ou serviço a um ou mais clientes, que iniciam os pedidos de serviço.

As funcionalidades como a troca de e-mail, acesso à internet ou o acesso a uma base de dados, são construídos com base no modelo cliente-servidor. Por exemplo, um navegador web é um programa cliente, em execução no computador do utilizador, que acede às informações armazenadas num servidor web na internet. Os utilizadores de serviços bancários, acedendo do seu computador, usam um cliente web para enviar uma solicitação para um servidor web num banco. Esse programa pode, por sua vez, encaminhar o pedido para o seu próprio programa da base de dados do cliente que envia uma solicitação para um servidor de base de dados noutro computador do banco para recuperar as informações da conta. O saldo é devolvido ao cliente da base de dados do banco, que por sua vez, serve de volta ao cliente navegador exibindo os resultados para o utilizador.

O modelo cliente-servidor tornou-se numa das ideias centrais da computação de rede. Muitas aplicações de negócios, escritas hoje, utilizam o modelo cliente-servidor. O termo também tem sido utilizado para distinguir a computação distribuída por computadores dispersos da "computação" monolítica centralizada em mainframe.

Cada instância de software do cliente pode enviar requisições a vários servidores. Por sua vez, os servidores podem aceitar esses pedidos, processá-los e retornar as informações solicitadas para o cliente. Embora este conceito possa ser aplicado por uma variedade de razões e para diversos tipos de aplicações, a arquitetura permanece fundamentalmente a mesma.

**Tipos ou Modelos de Client/Server:**

Após vários modelos estudados de cliente-servidor, caracterizou-se chamar tecnicamente de arquitetura multicamada, inspirado nas camadas no Modelo OSI, o processo de dividir a arquitetura de cliente-servidor em várias camadas lógicas facilitando o processo de programação distribuída, existe desde o modelo mais simples de duas camadas, e o mais utilizado atualmente que é o modelo de três camadas que é paralelo ao modelo de arquitetura de software denominado MVC (Model-view-controller).

**Características do Cliente:**

* Inicia pedidos para servidores;
* Espera por respostas;
* Recebe respostas;
* Conecta-se a um pequeno número de servidores de uma só vez ;
* Normalmente interage diretamente com os servidores através do seu software aplicação especifico, que lhe possibilita a comunicação com o servidor;
* Utiliza os recursos da rede.

**Características do Servidor:**

* Sempre espera por um pedido de um cliente;
* Atende os pedidos e, em seguida, responde aos clientes com os dados solicitados;
* Podem se conectar com outros servidores para atender uma solicitação específica do cliente; jamais podem se comunicar.
* Fornece recursos de rede.
* Normalmente interage diretamente com os utilizadores finais através de qualquer interface com o utilizador;
* Estrutura o sistema.

**Vantagens:**

* Na maioria dos casos, a arquitetura cliente-servidor permite que os papéis e responsabilidades de um sistema de computação possam ser distribuídos entre vários computadores independentes que são conhecidos por si só através de uma rede. Isso cria uma vantagem adicional para essa arquitetura: maior facilidade de manutenção. Por exemplo, é possível substituir, reparar, atualizar ou mesmo realocar um servidor de seus clientes, enquanto continuam a ser a consciência e não afetado por essa mudança;
* Todos os dados são armazenados nos servidores, que geralmente possuem controlos de segurança muito maiores do que a maioria dos clientes. Os servidores podem controlar melhor o acesso a recursos, para garantir que apenas os clientes com credenciais válidas possam aceder e alterar os dados;
* Como o armazenamento de dados é centralizado, as atualizações dos dados são muito mais fáceis de administrar em comparação com o paradigma P2P. Numa arquitetura P2P, as atualizações de dados podem precisar de ser distribuídas e aplicadas a cada nó na rede, o que consome tempo e é passível de erro, já que podem haver milhares ou mesmo milhões de nós;
* Muitas tecnologias avançadas de cliente-servidor estão disponíveis e foram projetadas para garantir a segurança, facilidade de interface do utilizador e a facilidade de uso;
* Funciona com vários clientes diferentes de capacidades diferentes.

**Desvantagens:**

* Os clientes podem solicitar serviços, mas não podem oferecê-los para outros clientes, sobrecarregando o servidor, pois quanto mais clientes, mais informações irão demandar mais banda.
* Um servidor poderá ficar sobrecarregado caso receba mais solicitações simultâneas dos clientes do que pode suportar;
* Este modelo não possui a robustez de uma rede baseada em P2P. Na arquitetura cliente-servidor, se um servidor crítico falha, os pedidos dos clientes não poderão ser cumpridos. Já nas redes P2P, os recursos são normalmente distribuídos entre vários nós. Mesmo se uma ou mais máquinas falharem no momento de download de um ficheiro, por exemplo, as outras ainda terão os dados necessários para completar a referida operação.

**Protocolos de transporte e aplicações de rede:**

Os protocolos do nível de transporte fornecem serviços que garantem uma transferência confiável de dados e aplicações entre computadores (ou outros equipamentos) remotos. Os programas na camada de aplicação usam os protocolos de transporte para contactar outras aplicações. Para isso, a aplicação interage com o software do protocolo antes de ser feito o contacto. A aplicação que aguarda a conexão informa ao software do protocolo local que está pronta a aceitar mensagem. A aplicação que estabelece a conexão usa os protocolos de transporte e de rede para contactar o sistema que aguarda. As mensagens entre as duas aplicações são trocadas através da conexão resultante.