

**Aluno: André Luiz N. Carneiro RA: 92854**



---

FUNDAÇÃO HERMÍNIO OMETTO

**Investigação - Arquiteturas de Sistema**

**Klausner**

**ARARAS/SP**

**04/2021**

## Arquiteturas de sistemas (centralizadas, descentralizadas e híbridas)

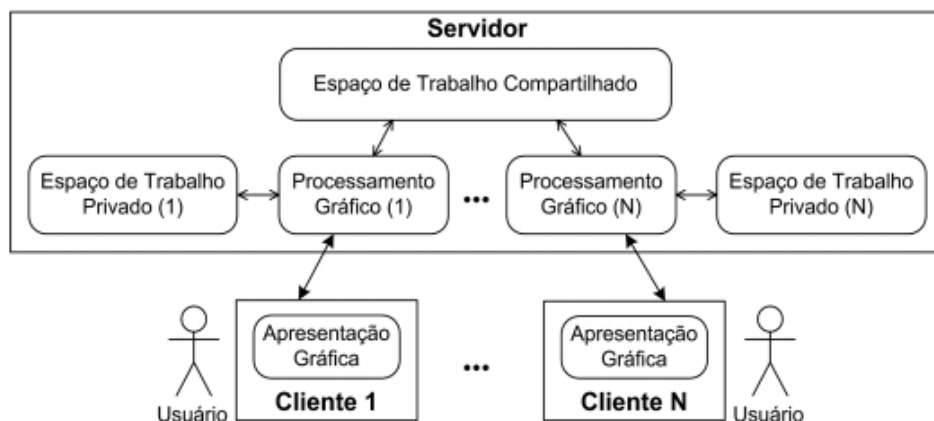
### 1) Arquitetura Centralizadas.

O foco dela e o exemplo principal é o cliente/servidor, do qual pode ser implementado por meio de um protocolo simples, caso a rede seja confiável. Ela é centralizada porque temos um servidor e os clientes solicitam o conteúdo desse servidor. Tendo o processo dividido em 2 grupos, servidor e cliente.

- Lado do servidor: processo que implementa um serviço específico.
- Lado cliente: processo que solicita um serviço de um servidor.

Temos a forma de **Forma de interação: Requisição-Resposta**

Esse sistema tem uma limitação, da qual se o servidor travar o sistema não funcionará corretamente, e os usuários não poderão acessar os seus dados. Como um sistema centralizado precisa de um proprietário central para conectar todos os outros usuários e dispositivos, a disponibilidade da rede depende desse proprietário. Também existem preocupações em relação a segurança que acabam surgindo quando um proprietário armazena (e pode acessar) dados do usuário, e é fácil entender por que os sistemas centralizados não são mais a primeira escolha para muitas organizações.



**Figura 1 - Arquitetura Centralizada Processamento Utilizado Em Um Servidor**

**Exemplos** de arquitetura que a utilizam: Aplicações de **acesso remoto**, como **Telnet**, e alguns tipos de aplicações **de compartilhamento de desktop**.

**Prós:**

- Implantação simples
- Pode ser desenvolvido rapidamente
- Acessível para manter
- Prático quando os dados precisam ser controlados centralmente

**Contras:**

- Propenso a falhas
- Maiores riscos de segurança e privacidade para os usuários
- Tempos de acesso mais longos aos dados para usuários que estão longe do servidor

## 2) Arquitetura descentralizada.

Como o próprio nome indica, os sistemas descentralizados não têm um proprietário central. Em vez disso, eles usam vários proprietários centrais, cada um dos quais geralmente armazena uma cópia dos recursos que os usuários podem acessar; um sistema descentralizado pode ser tão vulnerável a travamentos quanto um centralizado. No entanto, é por design mais tolerante a falhas. Isso porque, quando um ou mais proprietários centrais ou servidores falham, os outros podem continuar a fornecer acesso aos dados aos usuários.

Pode possuir uma distribuição vertical; com componentes logicamente diferentes em máquinas diferentes, tabelas BD subdivididas em colunas e distribuídas, divisão lógica e física. Também podem possuir uma distribuição horizontal, tendo o servidor/cliente divididos em partes logicamente equivalentes, tendo cada parte operando sobre o seu próprio conjunto de dados e tendo distribuição de carga.

**Exemplos: distribuição de arquivos (bitTorrent), streaming (KanKan), VoIP (Skype).**

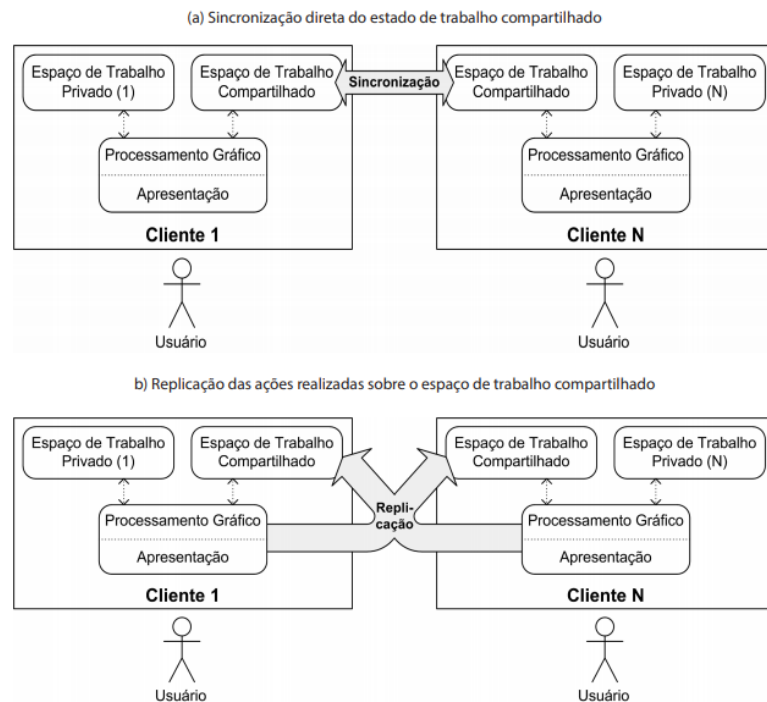


Figura 20.3 Arquiteturas descentralizadas

Os recursos permanecem ativos se pelo menos um dos servidores centrais continuar a operar. Normalmente, isso significa que os proprietários do sistema podem reparar servidores com defeito e resolver quaisquer outros problemas enquanto o próprio sistema continua a funcionar normalmente.

As falhas de servidor em um sistema descentralizado podem afetar o desempenho e limitar o acesso a alguns dados. Mas em termos de tempo de atividade geral do sistema, este sistema oferece uma grande melhoria em relação a um sistema centralizado.

**Prós:**

- Menos probabilidade de falhar do que um sistema centralizado
- Melhor performance
- Permite um sistema mais diversificado e flexível

*Outra vantagem desse design é que o tempo de acesso aos dados costuma ser mais rápido. Isso ocorre porque os proprietários podem criar nós em diferentes regiões ou áreas onde a atividade do usuário é alta.*

**Contras:**

- Riscos de segurança e privacidade para os usuários
- Custos de manutenção mais altos
- Desempenho inconsistente quando não otimizado adequadamente

*No entanto, os sistemas descentralizados ainda estão sujeitos aos mesmos riscos de segurança e privacidade para os usuários que os sistemas centralizados. E embora sua tolerância a falhas seja maior, isso tem um preço. Manter um sistema descentralizado costuma ser mais caro.*

### 3) Arquitetura híbrida.

Combina mais de um tipo de arquitetura, podendo ter características de arquiteturas centralizadas e descentralizadas em um único sistema, por ex: cliente-servidor e P2P;

Um modelo de arquitetura híbrida, bastante adotado, utiliza um servidor responsável pelo gerenciamento do espaço de trabalho compartilhado, enquanto o cliente gerencia o espaço de trabalho privado e realiza todo o processamento gráfico. Este modelo promove uma maior escalabilidade já que uma parte do processamento é transferida para os clientes. Esse modelo conserva a simplicidade de uma arquitetura centralizada, pois mantém uma única instância do espaço de trabalho compartilhado.

Dependendo do tipo de sistema colaborativo, o tempo de feedback é crítico, sobretudo se for um sistema síncrono. Uma alternativa é dividir o espaço de trabalho compartilhado: uma parte é gerenciada pelo servidor e outra parte é gerenciada pelos clientes. Esta arquitetura pode ser utilizada para implementar diferentes mecanismos de controle de concorrência. O servidor aplica um mecanismo pessimista para controlar o acesso a recursos compartilhados cuja consistência seja crítica. Ao mesmo tempo, com base em um mecanismo otimista, os clientes gerenciam os recursos compartilhados cujo acesso deve apresentar pequenos tempos de resposta.

Suponha que um grupo de professores ministrou uma disciplina e esteja agora editando o relatório de desempenho dos alunos. A primeira parte do relatório contém comentários sobre os alunos, enquanto a segunda parte contém as tabelas com as notas obtidas pelos alunos em cada atividade. Enquanto na primeira parte do relatório alguma inconsistência no texto é tolerável, podendo até ser vista como erro de digitação, na segunda parte do relatório é importante garantir a consistência das tabelas. Desta forma, a primeira parte pode ser gerenciada diretamente pelos sistemas clientes com o uso de um mecanismo otimista, o que possibilita que usuários digitem simultaneamente trechos do texto. Já a segunda parte é gerenciada pelo servidor que garante acessos exclusivos a cada tabela.

Uma arquitetura híbrida também é adequada para acomodar em um mesmo ambiente diferentes tipos de sistemas computacionais. Neste caso, para os sistemas mais restritos, por exemplo, que apresentem capacidades de processamento e memória limitadas, a

aplicação cliente realiza apenas o processamento gráfico e o gerenciamento do espaço de trabalho privado. Já o acesso ao espaço compartilhado é gerenciado pelo servidor. Para sistemas com mais recursos computacionais, a aplicação cliente também gerencia o espaço compartilhado de forma distribuída juntamente com o servidor, seguindo um dos modelos de sincronização.

Também é comum em sistemas distribuídos colaborativos. E também o Cliente-servidor é utilizado para nós conectarmos ao sistema, depois utilizam esquema descentralizado.

**Exemplos: BitTorrent** – sistema de download de arquivo.

**Globule** - rede de distribuição de conteúdo colaborativa na qual usuários ou organizações voluntariamente disponibilizam servidores “web” melhorados que são capazes de colaborar para a replicação de páginas “web”.

Links de Pesquisa Utilizados:

<https://berty.tech/blog/decentralized-distributed-centralized>

<https://sistemascolaborativos.uniriotec.br/wp-content/uploads/sites/18/2019/06/SC-cap20-arquiteturas.pdf>

[http://www.facom.ufu.br/~faina/BCC\\_Crs/GSI028-2014-1S/DL/DS-Ch02.pdf](http://www.facom.ufu.br/~faina/BCC_Crs/GSI028-2014-1S/DL/DS-Ch02.pdf)

[https://www.ic.unicamp.br/~bit/ensino/mc714\\_2s13/aulas/aula10.pdf](https://www.ic.unicamp.br/~bit/ensino/mc714_2s13/aulas/aula10.pdf)