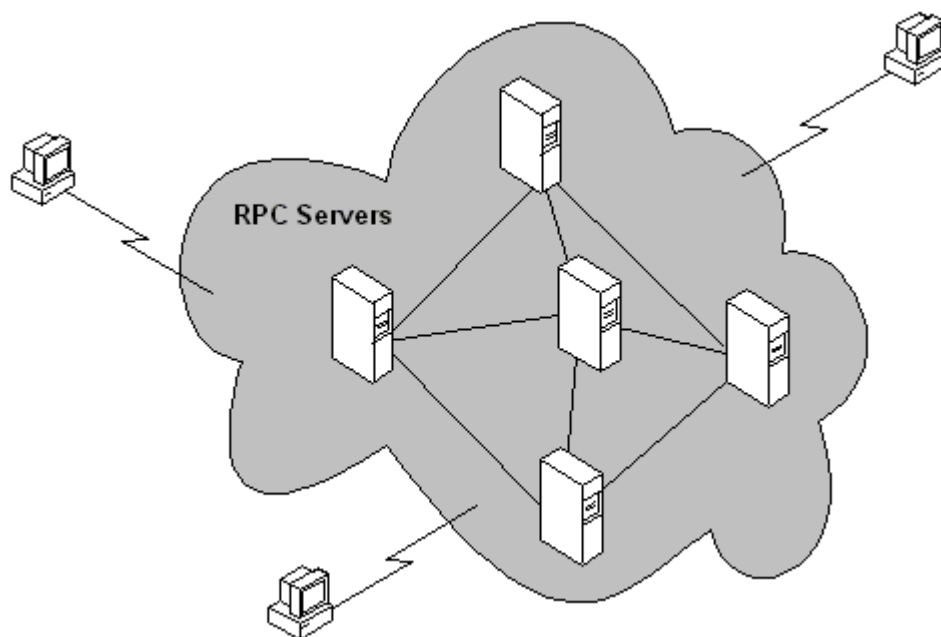


**CURSO DE SISTEMAS DE INFORMAÇÃO**  
**DISCIPLINA DE SISTEMAS DISTRIBUÍDOS - 7º PERÍODO**  
**ALUNO: JOÃO VICTOR PÓVOA FRANÇA**

### 1) O que é um Sistema Distribuído?

Um sistema distribuído é um conjunto de computadores independentes, interligados por rede, que para o usuário aparentam um único sistema coerente. Em outras palavras: componentes de hardware e software, em máquinas diferentes, se comunicam e coordenam ações trocando mensagens. Normalmente podemos ver exemplos em serviços de streaming, redes sociais, internet banking e aplicativos de mensagens: por trás da tela do celular, há dezenas ou centenas de servidores coordenados para entregar uma experiência contínua.

Em termos simples, várias máquinas trabalham juntas como se fossem uma só, sincronizando dados e respondendo a pedidos via mensagens. Isso permite escalar a capacidade e aumentar a disponibilidade do serviço.



### 2) Quais as principais metas de um Sistema Distribuído?

- **Transparência:** esconder a complexidade da distribuição (acesso a recursos, em que máquina eles estão, se foram movidos ou replicados, como lidar com acessos simultâneos e até com falhas).
- **Compartilhamento de recursos:** usar em conjunto hardware, dados e serviços para reduzir custos e melhorar o aproveitamento da infraestrutura.
- **Abertura:** adotar padrões e interfaces bem definidas para facilitar integração e portabilidade.
- **Escalabilidade:** crescer em número de usuários, volume de dados e alcance geográfico mantendo desempenho aceitável. Essas metas guiam decisões de arquitetura e operação no dia a dia.

**CURSO DE SISTEMAS DE INFORMAÇÃO**  
**DISCIPLINA DE SISTEMAS DISTRIBUÍDOS - 7º PERÍODO**  
**ALUNO: JOÃO VICTOR PÓVOA FRANÇA**

**3) Quais as vantagens e desvantagens do compartilhamento de recursos?**

**Vantagens**

- Redução de custo; **reuso** de infraestrutura, código, banco de dados e serviços.
- **Colaboração** e troca de informações facilitadas.

**Desvantagens (típicas)**

- **Concorrência** (acessos simultâneos), **consistência** de dados e **sincronização**.
- **Segurança** e **privacidade**.
- **Dependência de rede** e possíveis **gargalos**.

**4) O que significa dizer que um sistema distribuído é transparente?**

Dizer que o sistema é transparente significa que a distribuição fica “invisível” para quem usa e, muitas vezes, para quem desenvolve. O objetivo é que operações pareçam locais e simples, mesmo que, por baixo, envolvam múltiplos servidores, replicação de dados e mecanismos de tolerância a falhas. A transparência reduz o acoplamento entre a aplicação e a topologia da infraestrutura, permitindo evoluir e operar o sistema com menos atrito.

**5) Quais os tipos de transparência existem para sistemas distribuídos?**

De acordo com o material, destacam-se: **acesso, localização, migração, replicação, concorrência e falhas**.

- **Acesso** (usar um recurso por uma interface uniforme);
- **Localização** (não precisar saber onde o recurso está);
- **Migração** (o recurso pode mudar de máquina sem interromper o uso);
- **Replicação** (existirem várias cópias sem o cliente perceber);
- **Concorrência** (vários acessos simultâneos sem conflito aparente);
- **Falhas** (o sistema lida com erros de forma a minimizar impacto percebido). Esses aspectos se complementam para manter a experiência consistente.

**6) O que caracteriza a Transparência de Localização?**

**CURSO DE SISTEMAS DE INFORMAÇÃO**  
**DISCIPLINA DE SISTEMAS DISTRIBUÍDOS - 7º PERÍODO**  
**ALUNO: JOÃO VICTOR PÓVOA FRANÇA**

Na transparência de localização, o usuário ou a aplicação acessa um recurso sem precisar saber em qual servidor ou datacenter ele está. Por exemplo, um arquivo na nuvem pode ser servido por um servidor no Brasil ou na Europa, e o cliente continua usando a mesma URL. Provedores frequentemente usam CDNs para entregar conteúdo do ponto mais próximo, reduzindo a latência e melhorando desempenho, enquanto o cliente segue usando o mesmo endereço.

### 7) O que caracteriza a Transparência de Replicação?

Replicação é manter múltiplas cópias de dados ou serviços para melhorar **disponibilidade** (se uma cópia falhar, outra atende) e **desempenho** (cópias mais próximas do usuário). A transparência entra quando o cliente não precisa escolher qual réplica usar: o sistema roteia automaticamente e mantém as cópias coerentes. Isso exige políticas de **consistência** (forte, eventual, etc.) e mecanismos de sincronização para resolver atualizações concorrentes.

### 8) O que caracteriza a Transparência de Concorrência?

Concorrência acontece quando vários clientes acessam o mesmo recurso simultaneamente. A transparência de concorrência significa que o sistema previne conflitos e preserva propriedades como integridade dos dados sem exigir que o usuário coordene manualmente esses acessos. Na prática, usa-se bloqueio pessimista/otimista, versões de registros e, em cenários de banco de dados, transações com isolamento adequado. O importante é que o efeito final pareça “sequencial e correto” para quem está consumindo o serviço.

### 9) O que caracteriza a Transparência de Falha?

O sistema **detecta e contorna falhas** (reativa ou proativamente), mantendo o serviço sempre que possível (ex.: **failover**). Idealmente, o usuário **não percebe** a interrupção.

### 10) Quais os principais tipos de sistemas distribuídos? Diferencie cada um.

#### a) Sistemas de Computação Distribuídos (foco em alto desempenho)

No grupo dos **sistemas de computação distribuídos** (foco em desempenho), temos três exemplos clássicos.

- **Cluster** Um cluster é um conjunto coeso de máquinas (geralmente semelhantes) conectadas

**CURSO DE SISTEMAS DE INFORMAÇÃO**  
**DISCIPLINA DE SISTEMAS DISTRIBUÍDOS - 7º PERÍODO**  
**ALUNO: JOÃO VICTOR PÓVOA FRANÇA**

por rede local e administradas como um único recurso. A ideia é somar poder de processamento e memória para executar cargas paralelas (por exemplo, processamento científico, big data, renderização, treinamento de modelos).

- **Grid:** No grid, a capacidade vem de **várias organizações** colaborando: universidades, centros de pesquisa e empresas “federam” recursos (CPU, GPU, armazenamento, datasets). **Federação** de recursos de **múltiplas organizações**; compartilha servidores, armazenamento, bases de dados e até GPUs/NPUs.
- **Cloud:** provedor oferece **recursos virtualizados** (IaaS/PaaS/SaaS), em nuvem **privada, pública ou híbrida**.

**b) Sistemas de Informação Distribuídos (foco em aplicações e dados)**

- **Processamento de Transações (ACID):** São sistemas que precisam garantir que operações críticas em dados sejam **Atômicas, Consistentes, Isoladas e Duráveis**. Em ambiente distribuído, uma mesma transação pode envolver múltiplos serviços/bancos.

**Integração de Aplicações Empresariais (EAI):** É o “tecido conjuntivo” entre sistemas diferentes. Três estilos aparecem o tempo todo:

1. **Chamadas remotas síncronas — RPC/RMI, REST/gRPC:** ideais quando o cliente precisa da resposta **agora** (consulta de saldo, cálculo de frete).
  - Simples de programar, mas **acoplam** mais e propagam falhas/latência em cascata.
2. **Mensageria assíncrona — filas (ponto-a-ponto) e publish/subscribe (tópicos):** produtor envia mensagem/evento, consumidor processa quando puder.
  - **Desacopla** produtor de consumidor, absorve picos, melhora resiliência; porém exige **processamento eventual** e desenho para **reexecução/idempotência**.
3. **Microserviços —** organização da aplicação como um conjunto de serviços pequenos, independentes, com dados e deploy próprios.
  - Facilitam escalar partes específicas, atualizações frequentes e autonomia de times; pedem **observabilidade** (logs, métricas, tracing), **contratos bem definidos** e governança de **versões**.

**c) Sistemas Pervasivos / IoT:** Sistemas pervasivos integram dispositivos no mundo físico (sensores, atuadores, celulares, veículos) que coletam dados e interagem com serviços na borda (edge) e na

**CURSO DE SISTEMAS DE INFORMAÇÃO**  
**DISCIPLINA DE SISTEMAS DISTRIBUÍDOS - 7º PERÍODO**  
**ALUNO: JOÃO VICTOR PÓVOA FRANÇA**

nuvem. **Ubíquos, móveis e redes de sensores**, hoje agrupados no termo **IoT**.

### 11) Como são classificados os Sistemas de Computação Distribuídos?

O material aborda a evolução das **classificações dos sistemas computacionais** que embasam o cenário distribuído:

Tudo (processamento e dados principais) está concentrado em um **único computador** de grande porte; os terminais apenas acessam. É simples de administrar, mas vira **ponto único de falha e gargalo**.

#### **Paralelos fortemente acoplados**

Vários processadores **compartilham a mesma memória física** (SMP/NUMA). A comunicação é via barramento interno ou interconexão de altíssima velocidade.

- **Prós:** latência muito baixa, adequado para workloads que compartilham muitos dados.
- **Contras:** escalabilidade limitada pelo compartilhamento de memória; custo de hardware.

#### **Paralelos fracamente acoplados**

Cada nó tem **sua própria memória**, e a comunicação acontece por **rede**. É a base de **clusters** e dos sistemas distribuídos modernos (cliente-servidor, P2P, arquiteturas descentralizadas).

- **Prós:** escala horizontal, custo incremental, flexibilidade geográfica.
- **Contras:** latência de rede, **falhas parciais** (nem tudo cai ao mesmo tempo), necessidade de lidar com **consistência e partições**.

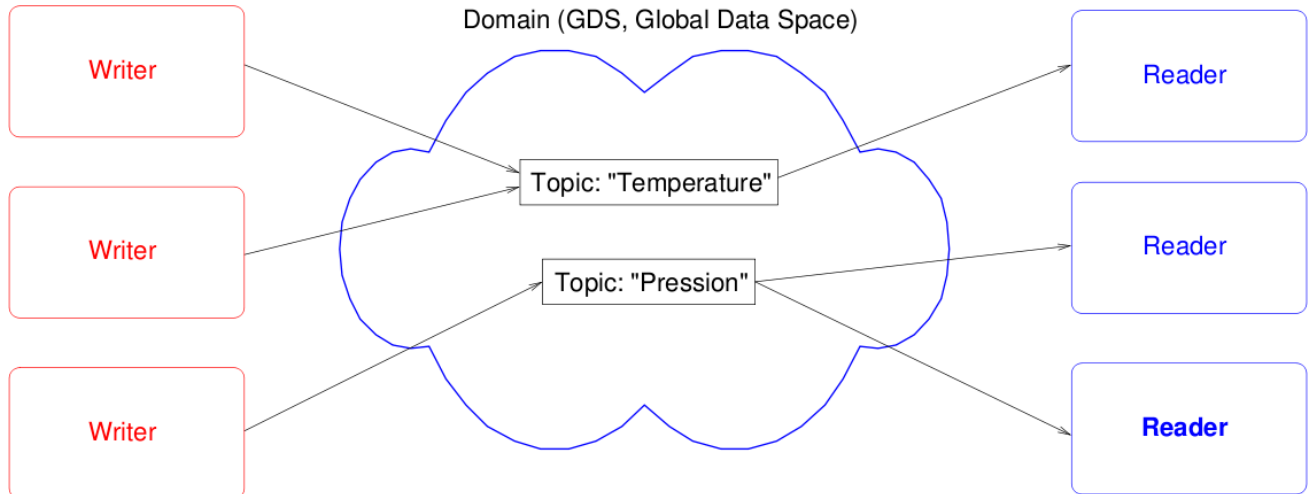
**Ligando com o tópico 10:** clusters e nuvem se apoiam no modelo **fracamente acoplado**; grids adicionam a dimensão **multi-organização**; sistemas centralizados e fortemente acoplados servem como contraste histórico/técnico.

### 12) Quais os tipos de Sistemas de Informação Distribuídos?

- **Processamento de Transações Distribuídas:** executam “tudo ou nada” (**ACID**), com **subtransações** em servidores diferentes.
- **Integração de Aplicações Empresariais:** **aplicações remotas** compartilham funcionalidades (**RPC, RMI, pub-sub, microserviços**).

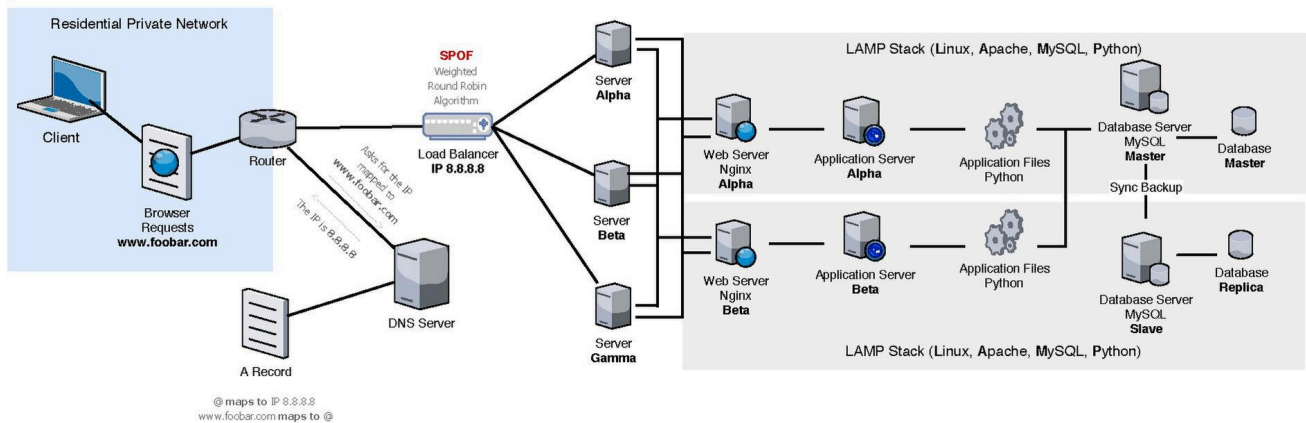
Ilustração (pub-sub — exemplo conceitual):

**CURSO DE SISTEMAS DE INFORMAÇÃO**  
**DISCIPLINA DE SISTEMAS DISTRIBUÍDOS - 7º PERÍODO**  
**ALUNO: JOÃO VICTOR PÓVOA FRANÇA**  
Domain (GDS, Global Data Space)



### 13) Como são formulados os Estilos Arquitetônicos?

## 1. Distributed Web Infrastructure



- **Dividido em componentes** A divisão em componentes você define fronteiras (o que entra/sai de cada parte), responsabilidades (o que cada parte resolve bem) e tamanho (componentes coesos, com baixo acoplamento).

Diretrizes práticas:

- **Coesão alta, acoplamento baixo:** cada componente deve ter um propósito claro (ex.: "Serviço de Pagamentos");

**CURSO DE SISTEMAS DE INFORMAÇÃO**  
**DISCIPLINA DE SISTEMAS DISTRIBUÍDOS - 7º PERÍODO**  
**ALUNO: JOÃO VICTOR PÓVOA FRANÇA**

- **Corte por domínio de negócio:** em microserviços, o recorte costuma seguir **capacidades de negócio** (ex.: catálogo, carrinho, checkout).
- **Camadas lógicas bem separadas:** em “camadas”, você tipicamente tem **UI → serviço → domínio → dados**.
- **Dados perto de quem decide:** sempre que possível, o componente **dono** dos dados relevantes decide sua própria regra (evita “banco compartilhado” como ponto de acoplamento duro).

Esses princípios se aplicam aos estilos dos slides: **camadas, orientado a objetos/serviços, baseado em recursos (REST) e publish–subscribe**.

## 2) Como os componentes se conectam

A partir de protocolos conhecidos em redes de computadores:

Pontos-chave:

- **Síncrono vs. assíncrono:**
  - **Síncrono** (HTTP/REST, gRPC, RPC/RMI) é simples e direto, bom para consultas/respostas rápidas.
  - **Assíncrono (filas e pub-sub)** desacopla produtor e consumidor, absorve picos e melhora resiliência, ao custo de **respostas eventuais**.
- **Contratos de interface:** descreva rotas/operções (ex.: OpenAPI/Swagger, protobuf), **timeouts**, **retries com backoff**, **circuit breaker** e **idempotência** (para reprocessar sem efeitos duplicados).
- **Descoberta e balanceamento:** como um componente encontra o outro? Via **service registry**, **DNS**, **gateway** ou **broker** (no caso de mensagens).
- **Qualidade de serviço:** latência, throughput, ordenação de mensagens, garantias de entrega (pelo menos uma vez, no máximo uma vez, etc.).

## 3) Quais dados trocam



**CURSO DE SISTEMAS DE INFORMAÇÃO**  
**DISCIPLINA DE SISTEMAS DISTRIBUÍDOS - 7º PERÍODO**  
**ALUNO: JOÃO VICTOR PÓVOA FRANÇA**

**Contratos de dados**, versões e semântica.

Boas práticas:

- **Contratos estáveis e versionados:** documente **esquemas** (JSON, protobuf/Avro), evitando que mudanças quebrem consumidores; use versionamento (v1, v2) e **compatibilidade retroativa** quando possível.
- **Semântica clara:** diferencie **comandos** (“faça X”) de **eventos** (“X aconteceu”). Em pub-sub, prefira **eventos imutáveis** com informação suficiente para que consumidores processem de forma independente.
- **Privacidade e segurança:** **criptografia** em trânsito (TLS) e em repouso; **mascaramento** de PII, **autorização** por escopo.
- **Consistência e ownership:** cada dado deve ter um **dono**; quando replicado, explicita o **modelo de consistência** (forte vs. eventual) e como resolver conflitos.

#### 4) Como são configurados em conjunto

Por fim, defina **onde e com que parâmetros rodam**: topologia de implantação, escalabilidade, disponibilidade e operação.

Elementos essenciais:

- **Topologia e orquestração:** monorepo vs. multirepo, **containers** (Kubernetes), **pipelines de deploy**, **configuração por ambiente** (variáveis/secrets).
- **Escala e resiliência:** **horizontal vs. vertical**, **auto-scaling**, **zonas/regiões**, **replicação e failover**; políticas de **retentativa** e **fila/retentores** para picos.
- **Observabilidade:** **logs estruturados**, **métricas** (SLIs/SLOs) e **tracing distribuído** para entender fluxos ponta a ponta.
- **Governança de APIs e dados:** **catálogo** de serviços, versões ativas/obsoletas, **gestão de schemas**; no pub-sub, **tópicos**, **partições**, **retenção**, **dead-letter** e **reprocessamento**.