

# ACID × BASE, Teorema CAP





# ACID × BASE



# Propriedades ACID

---

- Os bancos de dados **relacionais** costumam apresentar as chamadas **propriedades ACID**:
  - **Atomacidade**: uma transação será totalmente executada ou não será executada de forma alguma
  - **Consistência**: garante que um banco de dados passará sempre de um estado consistente a outro estado também consistente
  - **Isolamento**: garante que nenhuma transação sofrerá interferência de uma transação concorrente
  - **Durabilidade**: garante que os dados salvos não serão perdidos (persistência)





# Propriedades BASE

- Os bancos de dados **não relacionais**, por sua vez, apresentam as chamadas **propriedades BASE**:
  - **Basically Available** (disponível de forma básica)
  - **Soft-*state*** (estado não-duradouro)
  - **Eventually *C*onsistent** (consistente em algum momento)
- Uma aplicação NoSQL funciona, ainda que parcialmente, o tempo todo (*basically available*), não sendo consistente o tempo todo (*soft-state*), mas tornando-se consistente no momento devido (*eventually consistent*)

# Comparação

ACID	BASE
Consistência forte	Consistência fraca
Isolamento	Foco em disponibilidade
Concentra-se na execução das transações	Melhor esforço
Transações aninhadas	Respostas aproximadas
Disponibilidade	Mais simples e mais rápido
Conservador (pessimista)	Agressivo (otimista)
Evolução dificultada (p. ex., esquema)	Evolução mais fácil

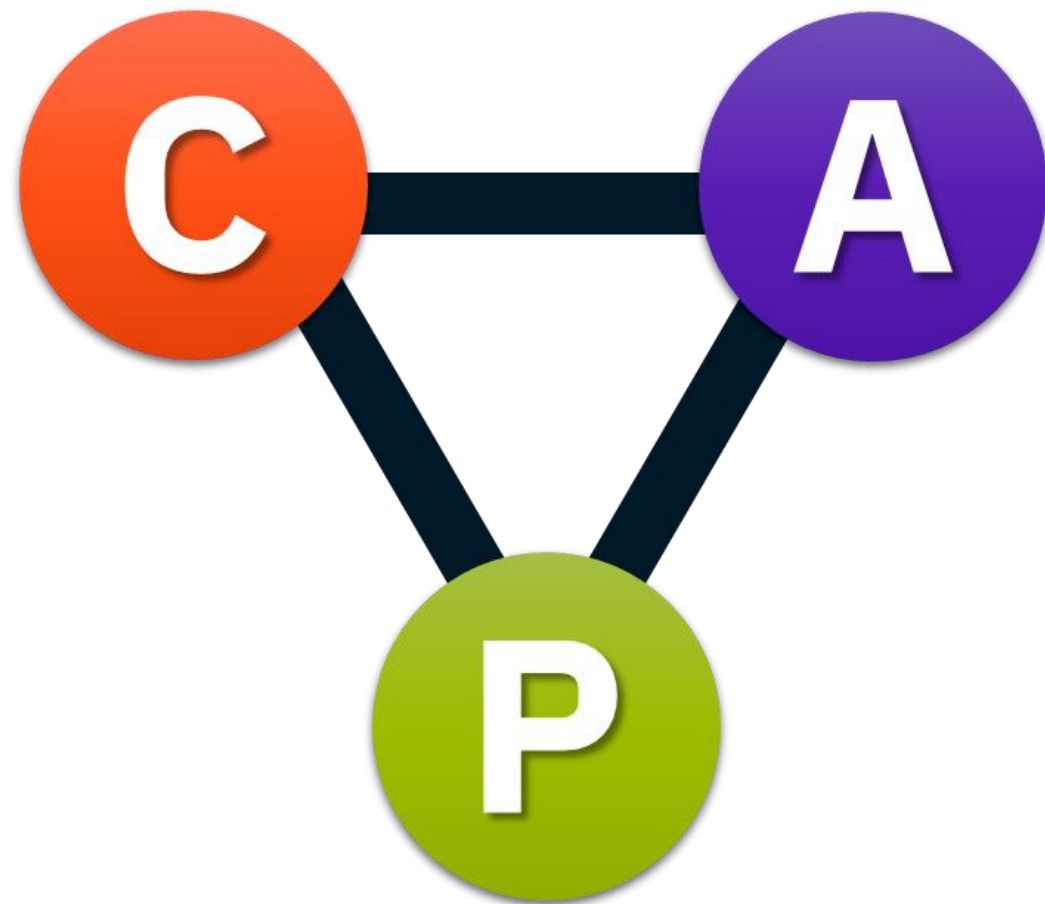


# TEOREMA CAP

# Teorema CAP (ou Teorema de Brewer)

---

- Introduzido por Eric A. Brewer em 2000
- Postula que sistemas distribuídos são capazes de entregar simultaneamente apenas duas de três características desejadas:
  - **Consistência**
  - **Disponibilidade** (**Aavailability**)
  - **Tolerância a Partição**



# Consistência

---

- Significa que todos os clientes veem os mesmos dados ao mesmo tempo, não importa em qual nó eles se conectem
- Para que isso aconteça, sempre que os dados forem gravados em um nó, ele deve ser instantaneamente encaminhado ou replicado para todos os outros nós do sistema antes que a gravação seja considerada “bem-sucedida”







# Disponibilidade (*availability*)

- Significa que qualquer cliente que fizer uma solicitação de dados obterá uma resposta, mesmo que um ou mais nós estejam desativados
- Ou seja, todos os nós em funcionamento no sistema distribuído retornam uma resposta válida para qualquer solicitação, sem exceção

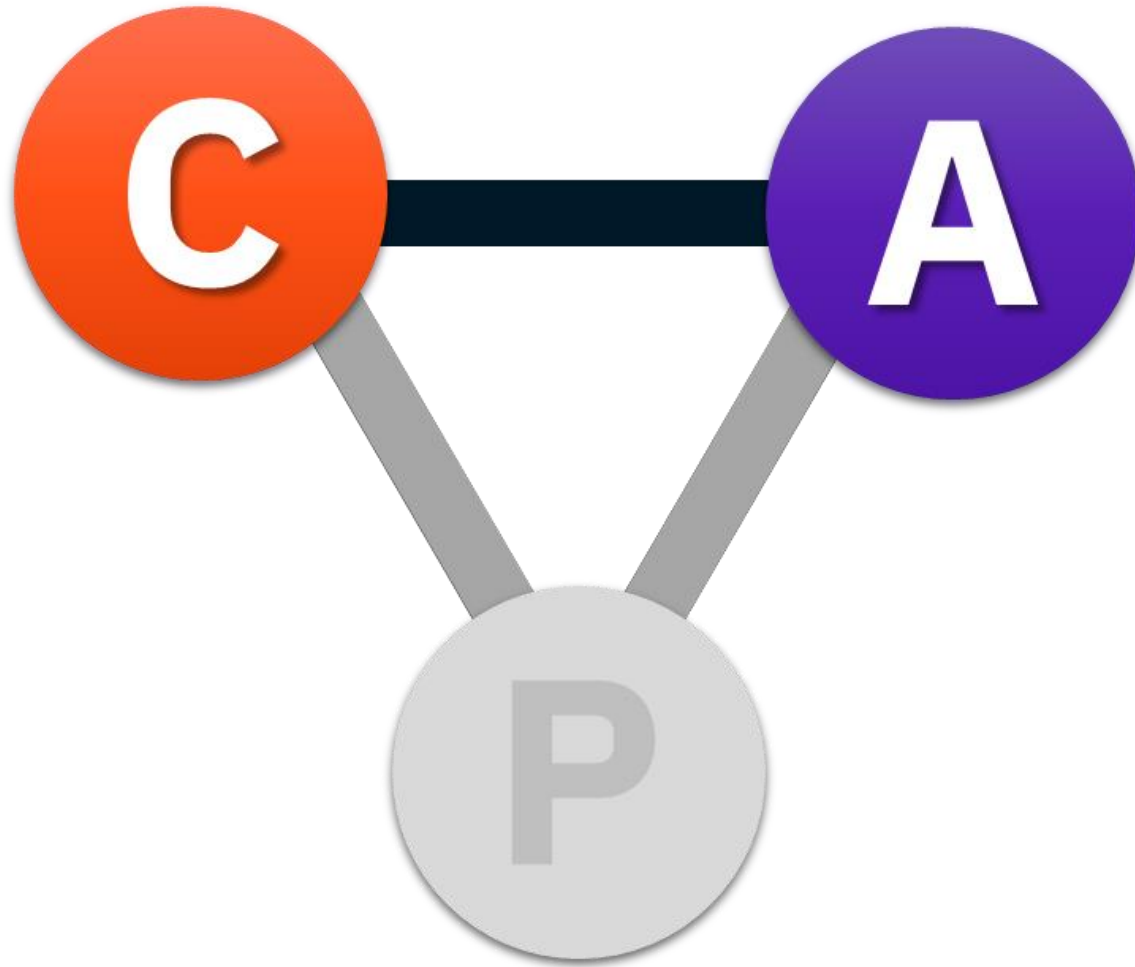
# Tolerância a partição

- A partição é uma quebra de comunicações dentro de um sistema distribuído, uma conexão perdida ou temporariamente lenta entre dois nós
- Tolerância de partição significa que o *cluster* deve continuar a funcionar mesmo de ocorrer uma ou mais falhas de comunicação entre os nós no sistema



# Diagrama





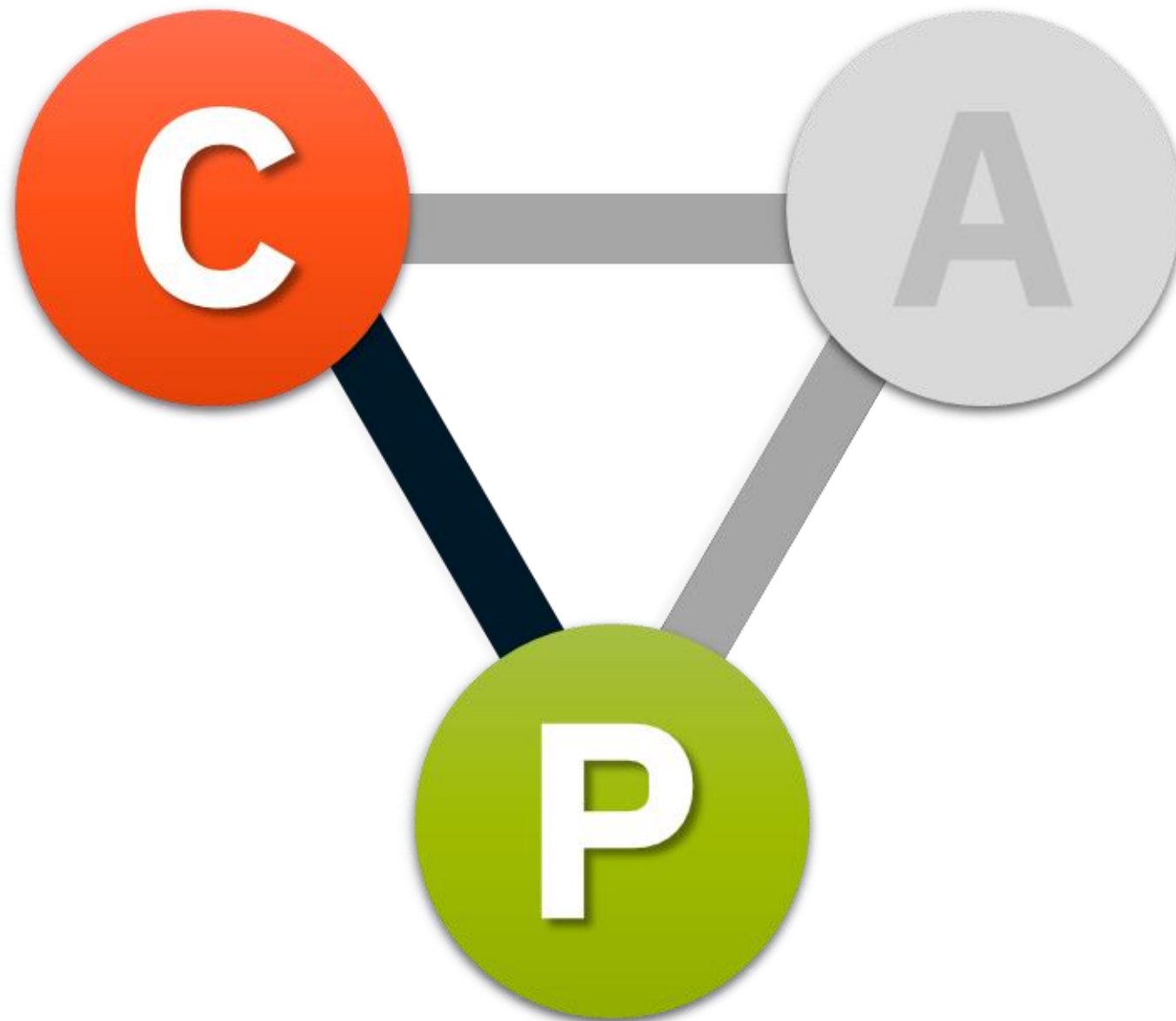
## Sistemas CA

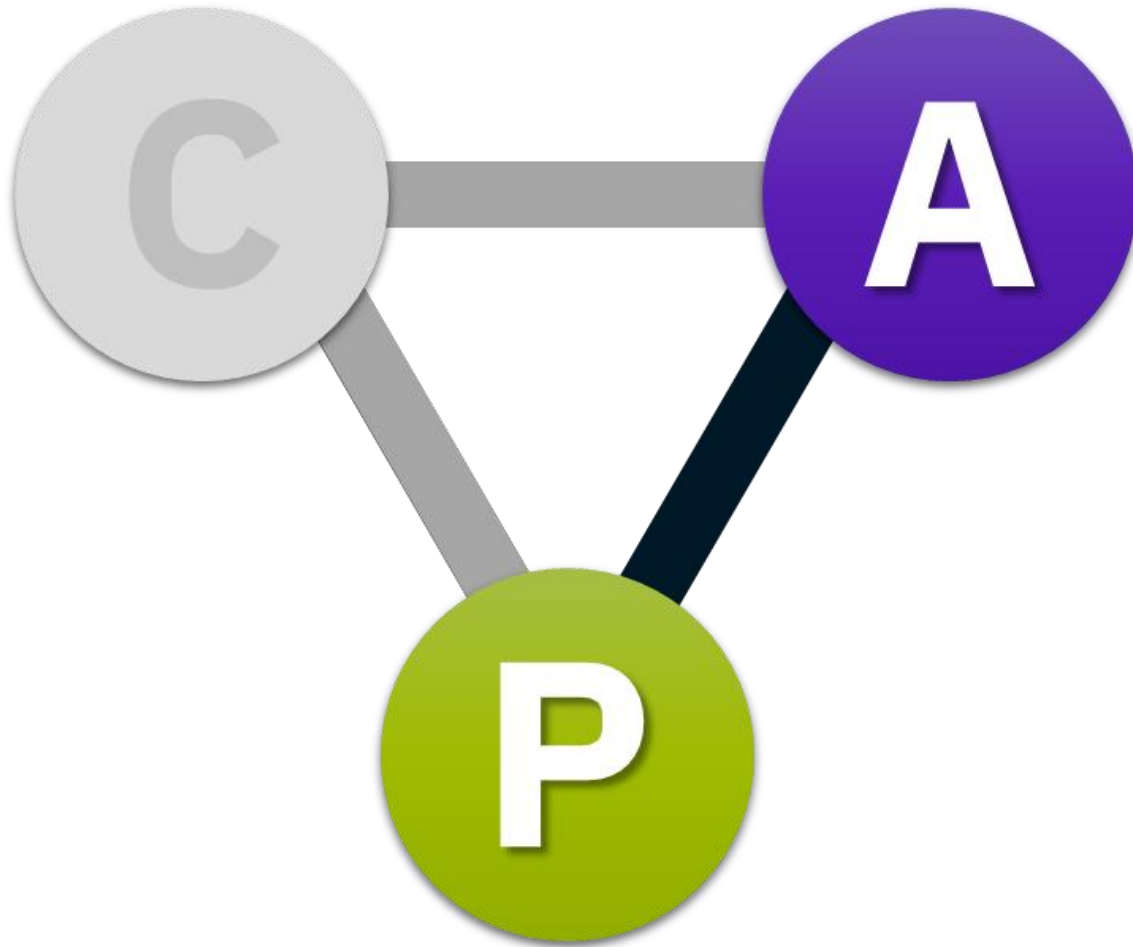
- Sistemas com consistência forte e alta disponibilidade não sabem lidar com uma possível falha de partição
- Caso ocorra, o sistema inteiro pode ficar indisponível até o membro do *cluster* ser restaurado
- É o caso da maioria dos SGBD relacionais



# Sistemas CP

- Sistemas que requerem consistência forte e tolerância a partição acabam precisando abrir mão de um pouco de disponibilidade
- Exemplos:
  - BigTable
  - HBase
  - MongoDB
  - Redis





## Sistemas AP

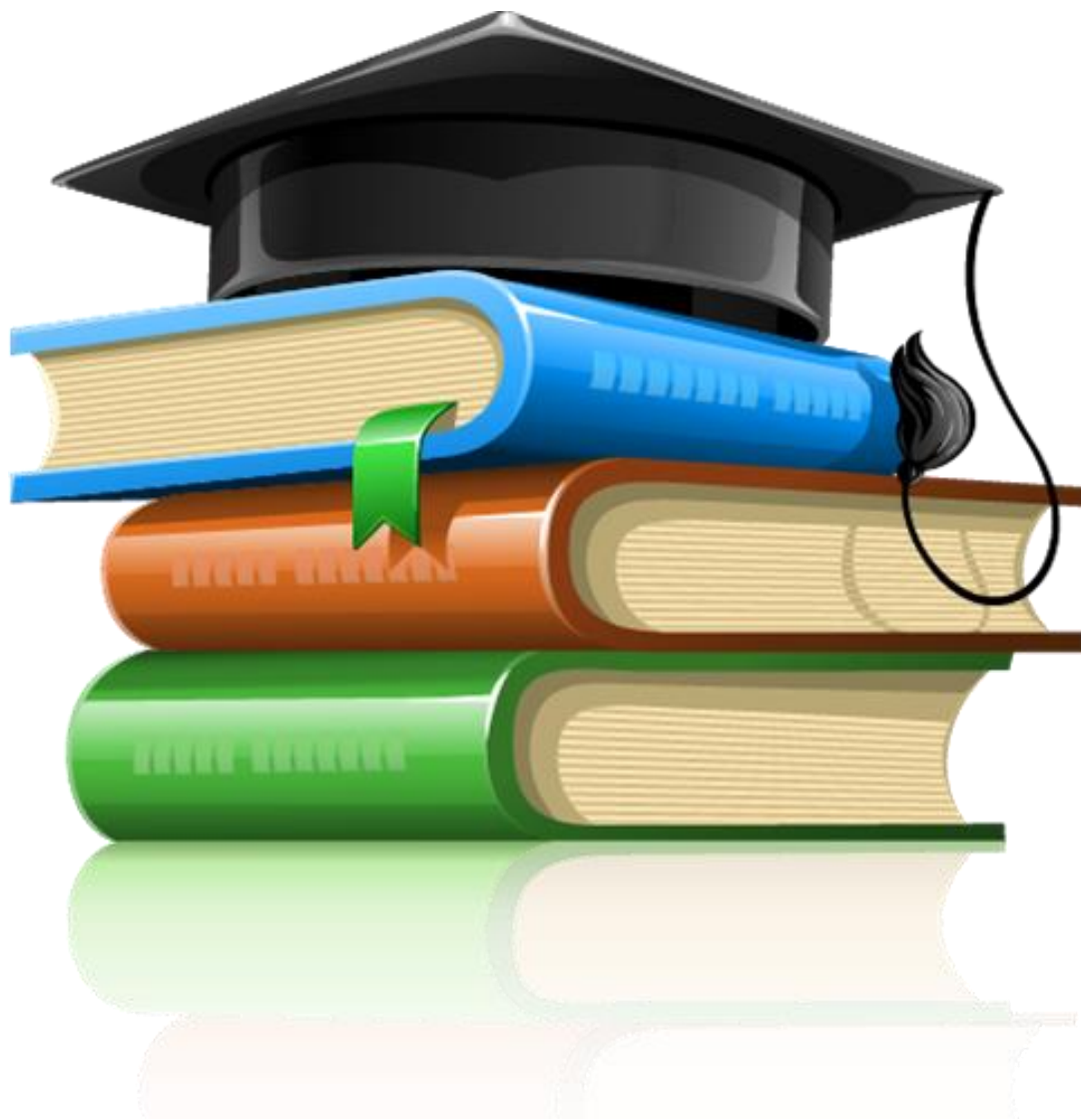
- Há sistemas que jamais podem ficar *offline* e, portanto, não podem sacrificar a disponibilidade
- Para manter uma alta disponibilidade mesmo com tolerância a partição, é necessário prejudicar a consistência
- Exemplos:
  - Apache Cassandra
  - CouchDB
  - DynamoDB
  - Riak



# Conclusão

- O Teorema CAP e as propriedades ACID × BASE são importantes para a compreensão do funcionamento do bancos de dados relacionais e não relacionais
- No entanto, com a evolução da tecnologia, a diferença entre as duas categorias de SGBD tem ficado cada vez mais tênue. Exemplos:
  - Desde a versão 4, o MongoDB suporta transações ACID envolvendo múltiplos documentos
  - Bancos de dados relacionais, como PostgreSQL e MySQL, passaram a permitir instalações em *clusters* para obter melhor tolerância a partição





## Para saber mais

- **Princípios de funcionamento ACID vs BASE nos bancos de dados.** Disponível em: <https://blog.compass.uol/tech/principios-de-funcionamento-acid-vs-base-nos-bancos-de-dados/>. Acesso em 11 mar. 2022
- **Teorema CAP.** Disponível em: <https://www.ibm.com/br-pt/cloud/learn/cap-theorem>. Acesso em: 11 mar. 2022
- **SGBDs distribuídos e o Teorema CAP.** Disponível em: [https://edisciplinas.usp.br/pluginfile.php/5214218/mod\\_resource/content/1/mac439\\_aula\\_6\\_teorema\\_cap.pdf](https://edisciplinas.usp.br/pluginfile.php/5214218/mod_resource/content/1/mac439_aula_6_teorema_cap.pdf). Acesso em: 11 mar. 2022