

EMENTA

SYSTEM DESIGN



TREINAMENTO EM TURMA

SYSTEM DESIGN



CLOUD

8 MESES

INTERMEDIÁRIO

mergulhe nos fundamentos de system design com foco em escalabilidade, tolerância a falhas, alta disponibilidade e sistemas distribuídos

vamos te ajudar a entender e propor soluções para cenários complexos, se destacar em entrevistas técnicas e, acima de tudo, aplicar arquitetura de forma estratégica no seu dia a dia. o impacto será real: no seu time, no produto e na sua carreira. o que antes parecia complicado, agora vai fazer total sentido.

ACESSO POR 36 MESES



MATERIAL DE APOIO



AULAS PRÁTICAS



SUORTE



PRA QUEM É ESSE TREINAMENTO

SRES focados em alta disponibilidade e confiabilidade.

ENGENHEIROS DEVOPS evoluir na construção de sistemas escaláveis e resilientes.

ARQUITETOS DE SOFTWARE que desejam aplicar princípios modernos de arquitetura em ambientes distribuídos.

ESTUDANTES que querem fortalecer a base técnica e aprender a projetar sistemas robustos.

PRÉ-REQUISITOS

- ☑ noções básicas de sistemas operacionais e linguagens de programação.
- ☑ familiaridade com alguns padrões de **projeto de software e arquitetura de software**.
- ☑ familiaridade com fundamentos de redes e protocolos (**HTTP, TCP/IP**).

GRADE DE APRENDIZADO





AULA 1

└ system design: protocolos de rede

fundamentos de protocolos de rede (tcp/ip, http, dns) e como a comunicação entre sistemas distribuídos acontece na prática.

AULA 2

└ system design: storage, raid e i/o

conceitos de armazenamento, raid e padrões de i/o, explorando latência, throughput e impacto em sistemas distribuídos.

AULA 3

└ teorema cap, databases acid e base

entendimento do teorema cap e dos modelos acid vs base para decisões de consistência em bancos de dados e considerando o teorema pascal.

AULA 4

└ databases, modelos de dados e indexação

modelagem de dados, tipos de bancos e estratégias de indexação para otimizar leitura e escrita.

AULA 5

└ estratégias de cache

padrões e estratégias avançadas de cache para reduzir latência, aliviar carga e melhorar performance sistêmica.



AULA 6

└ microsserviços, monolitos e domínios

comparação entre monólitos e microsserviços com foco em domínio, acoplamento e evolução arquitetural.

AULA 7

└ load balancers e proxies reversos

funcionamento de load balancers e proxies reversos na distribuição eficiente de tráfego utilizando os principais algoritmos.

AULA 8

└ api gateways

papel dos api gateways na centralização de entrada, governança, segurança e controle de tráfego.

AULA 9

└ backend for frontends (bff's)

uso de bffs para adaptação de apis por canal do cliente e otimização da experiência do cliente.

AULA 10

└ service mesh pattern

conceitos de service mesh para controle de comunicação, observabilidade e segurança entre serviços.

AULA 11

└ princípios de concorrência e paralelismo

princípios de concorrência e paralelismo para maximizar uso de recursos e throughput de aplicações.

AULA 12

└ comunicação síncrona - http, grpc, rpc, rest

comunicação síncrona via http, rest, rpc e grpc, com trade-offs de latência e acoplamento.

AULA 13

└ comunicação assíncrona - mensageria, eventos e streaming

comunicação assíncrona com filas, eventos e streaming para desacoplamento e escalabilidade.

AULA 14

└ performance, capacidade e escalabilidade

fundamentos de performance, capacidade e escalabilidade em sistemas de alta demanda.





AULA 15

└ **scale cube**

modelo scale cube para evolução arquitetural em dimensões x, y e z.

AULA 16

└ **principios de sharding e particionamento**

estratégias de sharding e particionamento para distribuição de dados em larga escala.

AULA 17

└ **principios de replicação de dados**

modelos de replicação de dados e seus impactos em consistência, disponibilidade e latência.

AULA 18

└ **day 18 - cqrs (command-query responsibility segregation)**

padrão CQRS para separação de leitura e escrita visando escalabilidade e performance.

AULA 19

└ saga pattern

saga pattern para coordenação de transações distribuídas com consistência eventual.

AULA 20

└ event sourcing pattern

event sourcing como abordagem para persistência baseada em eventos e reconstrução de estado.

AULA 22

└ modelos de deployment - blue/green, canary, feature toggles e etc

estratégias de deployment como blue/green, canary e feature toggles para releases seguras, combinadas com padrões mais avançados.



AULA 21

└ patterns de resiliência: idempotencia, circuit breakers, retries, fallbacks, timeouts e etc

patterns de resiliência como circuit breaker, retry e fallback para aumentar a robustez do sistema.

AULA 23

└ capacity planning

capacity planning para previsão de carga e dimensionamento eficiente de recursos.

AULA 24

└ testes de carga e estresse

testes de carga e estresse para validar limites e comportamento sob alta demanda.

AULA 25

└ bulkhead patterns

bulkhead pattern para isolamento de falhas e controle de blast radius.

AULA 26

└ cell-based pattern

cell-based architecture para segmentação sistêmica e aumento de resiliência em escala.

AULA 27

└ single point of failure (spof) e disaster recovery

identificação de single points of failure e estratégias de disaster recovery para continuidade do negócio.

AULA 28

└ observabilidade e monitoramento

observabilidade e monitoramento como pilares para entender, medir e evoluir sistemas distribuídos.



QUEM SERÁ
SEU GUIA:



INSTRUTOR

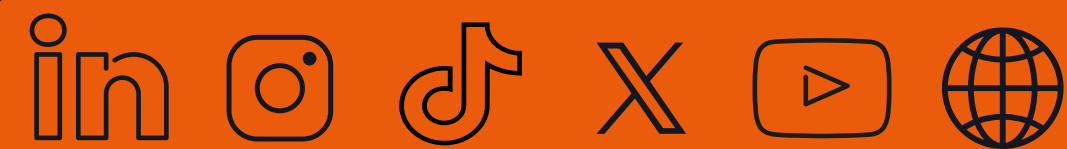
MATHEUS FIDELIS

PRINCIPAL ENGINEER
NO ITAÚ UNIBANCO



SOBRE O MATHEUS

COMO SITE RELIABILITY ENGINEER, SOFTWARE ENGINEER e PLATFORM ENGINEER, atuou em algumas das principais empresas do mercado brasileiro, adquirindo uma vasta experiência no gerenciamento de sistemas críticos em cenários complexos e de grande escala. sua especialização em disponibilidade e escalabilidade de sistemas permite monitorar, analisar e otimizar o desempenho de aplicações, infraestrutura e pipelines de CI/CD. está constantemente em busca de novos desafios para aplicar suas habilidades e contribuir para o sucesso de empresas visionárias.



SOBRE A

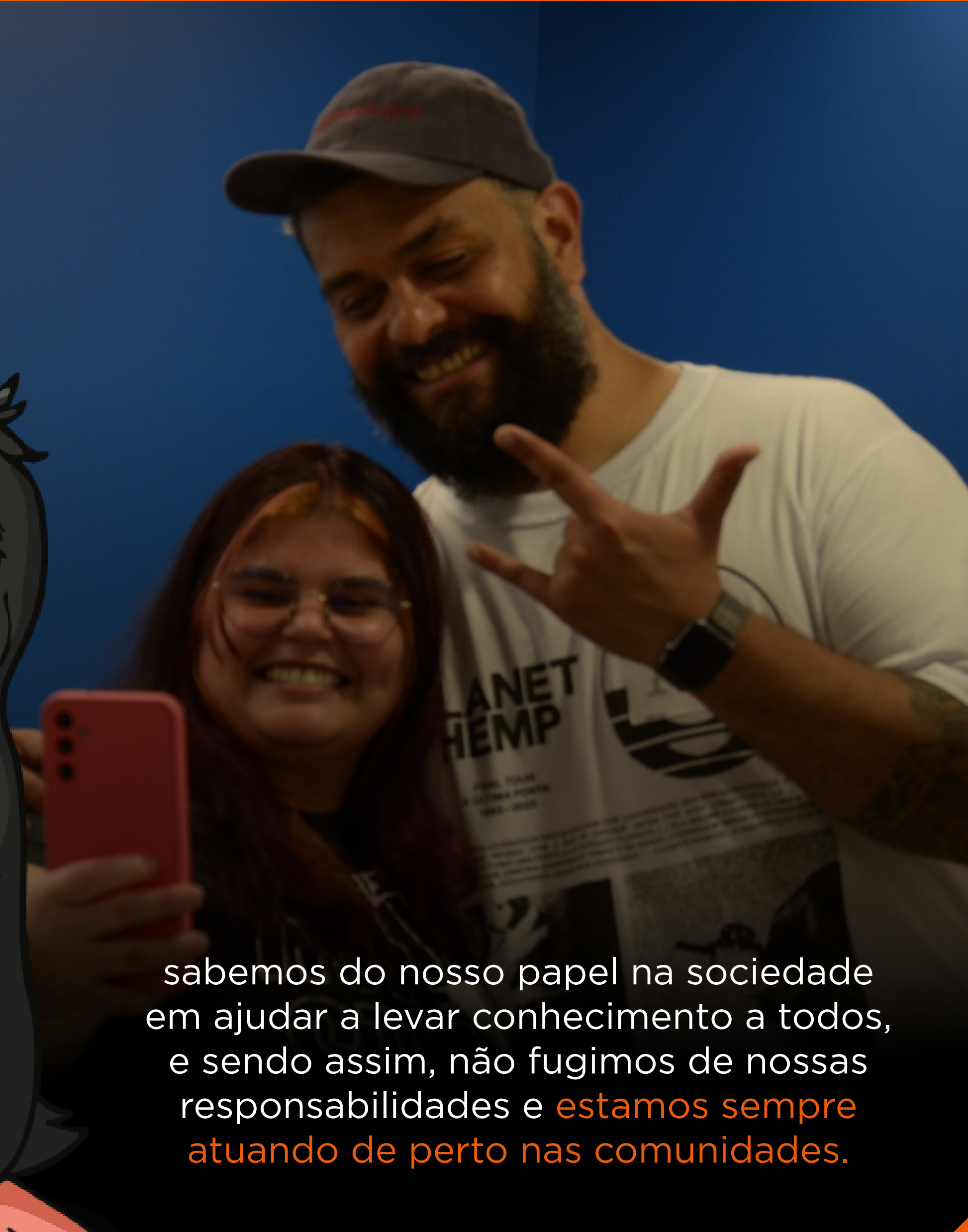
LINUXTIPS

desde 2014 compartilhando conhecimento e sendo a re-evolução na forma de aprender sobre TI.

hoje, a LINUXTIPS mantém mais de 20 treinamentos atualizados: todos focados na prática. acreditamos que a melhor forma de aprender algo em tecnologia é reproduzindo, testando, errando, corrigindo e evoluindo rápido.

FALE CONOSCO.

contato@linuxtips.io



sabemos do nosso papel na sociedade em ajudar a levar conhecimento a todos, e sendo assim, não fugimos de nossas responsabilidades e **estamos sempre atuando de perto nas comunidades.**