

**As leis da**

**arquitetura**

**funcional**

**Elton Minetto**

# Elton Minetto

- Escrevo códigos, sou professor, palestrante, e escritor

# Elton Minetto

- Escrevo códigos, sou professor, palestrante, e escritor
- Escrevo no <https://eltonminetto.dev>

# Elton Minetto

- Escrevo códigos, sou professor, palestrante, e escritor
- Escrevo no <https://eltonminetto.dev>
- Principal Software Engineer

# Elton Minetto

- Escrevo códigos, sou professor, palestrante, e escritor
- Escrevo no <https://eltonminetto.dev>
- Principal Software Engineer
- Google Developer Expert em Go

# Elton Minetto

- Escrevo códigos, sou professor, palestrante, e escritor
- Escrevo no <https://eltonminetto.dev>
- Principal Software Engineer
- Google Developer Expert em Go
- Pai da Alice (ela me obrigou a colocar essa parte)



**DR. WERNER  
VOGELS, CTO,  
AMAZON.COM**



# O desenvolvedor do futuro



# O desenvolvedor do futuro

- Pensamento sistêmico e design de arquiteturas



# O desenvolvedor do futuro

- Pensamento sistêmico e design de arquiteturas
- Comunicação clara com o negócio



# O desenvolvedor do futuro

- Pensamento sistêmico e design de arquiteturas
- Comunicação clara com o negócio
- Resolução estratégica de problemas e conhecimento especializado

# O que é frugal?

Frugal é um adjetivo que descreve alguém ou **algo moderado, econômico e simples**, especialmente **no uso de recursos, dinheiro ou alimentação**. Pessoa que **evita gastos desnecessários, sendo prudente com recursos financeiros e tempo**. Sinônimos: poupador, econômico, parco, moderado, sóbrio, simples, contido, regrado.

# **Os 3 Pilares da Arquitetura Frugal**

# Os 3 Pilares da Arquitetura Frugal

1. **Design (Projetar)**: Tomar decisões conscientes antes de escrever o código.

# Os 3 Pilares da Arquitetura Frugal

1. **Design (Projetar)**: Tomar decisões conscientes antes de escrever o código.
2. **Measure (Medir)**: Entender o comportamento e os custos reais na prática.

# Os 3 Pilares da Arquitetura Frugal

1. **Design (Projetar)**: Tomar decisões conscientes antes de escrever o código.
2. **Measure (Medir)**: Entender o comportamento e os custos reais na prática.
3. **Optimize (Otimizar)**: Melhoria contínua e questionamento do status quo.

**Design**

**Arquitetar é uma série de escolhas (“trade-offs”)**

# Arquitetar é uma série de escolhas (“trade-offs”)

- Ex: Aumentar a resiliência aumenta o custo

# Arquitetar é uma série de escolhas (“trade-offs”)

- Ex: Aumentar a resiliência aumenta o custo
- É preciso encontrar o balanço entre requisitos técnicos e de negócio, alinhando tolerância a riscos e orçamento.

**Faça do custo um requisito  
não-funcional**

# Faça do custo um requisito não-funcional

- Se os custos são maiores do que sua receita, seu negócio está em risco

# Faça do custo um requisito não-funcional

- Se os custos são maiores do que sua receita, seu negócio está em risco
- Custos deve ser um requisito a ser considerado desde o início do projeto e não somente depois que alguma crise financeira

**Sistemas que duram tem  
alinhamento de custos com o  
negócio**

# Sistemas que duram tem alinhamento de custos com o negócio

- Quando fazemos o design de um sistema é importante entendermos de onde vem o dinheiro do negócio e garantir que a arquitetura “segue o dinheiro”.

**Measure**

**Arquiteturas sensíveis ao  
custo devem implementar  
controles de custo**

# Arquiteturas sensíveis ao custo devem implementar controles de custo

- É essencial monitorar os custos

# Arquiteturas sensíveis ao custo devem implementar controles de custo

- É essencial monitorar os custos
- Componentes mais críticos são essenciais e devem ser otimizados independente dos custos

# Arquiteturas sensíveis ao custo devem implementar controles de custo

- É essencial monitorar os custos
- Componentes mais críticos são essenciais e devem ser otimizados independente dos custos
- Componentes menos críticos podem ter um custo menor e serem menos escaláveis ou robustos

**Sistemas não observados  
geram custos desconhecidos**

# Sistemas não observados geram custos desconhecidos

- Quanto mais visível e de fácil entendimento maior é o impacto na redução e controle dos custos

**Optimize**

**Otimização de custos é algo  
incremental**

# Otimização de custos é algo incremental

- A busca por eficiência de custos é uma jornada constante

# Otimização de custos é algo incremental

- A busca por eficiência de custos é uma jornada constante
- Pequenas otimizações vão se acumulando com o tempo e reduzindo custos, especialmente em ambientes de escala

**O sucesso incontestável leva  
a suposições**

# O sucesso incontestável leva a suposições

- Times que tiveram sucesso lançando produtos no passado podem se tornar confiantes em suas escolhas e pararem de se questionar se estão usando as melhores ferramentas

# O sucesso incontestável leva a suposições

- Times que tiveram sucesso lançando produtos no passado podem se tornar confiantes em suas escolhas e pararem de se questionar se estão usando as melhores ferramentas
- Uma stack que pode ter sido um sucesso em um projeto pode se tornar o gargalo em outro

# O sucesso incontestável leva a suposições

- Times que tiveram sucesso lançando produtos no passado podem se tornar confiantes em suas escolhas e pararem de se questionar se estão usando as melhores ferramentas
- Uma stack que pode ter sido um sucesso em um projeto pode se tornar o gargalo em outro
- É importante manter a mente aberta em relação a diferentes linguagens, bancos de dados e ferramentas para analisar qual é a melhor alternativa para o projeto em questão

# Lembre-se

[...] a frugalidade consiste em maximizar o valor, não apenas em minimizar os gastos. E para isso, você precisa determinar pelo que está disposto a pagar.



**P.S.**

**Eu não sou  
“pão-duro”,  
sou  
“frugal” :)**

# Links

# Links

- [Sobre a talk Building Systems that Last do Werner Vogels](#)

# Links

- [Sobre a talk Building Systems that Last do Werner Vogels](#)
- [As leis da arquitetura frugal](#)

# Links

- [Sobre a talk Building Systems that Last do Werner Vogels](#)
- [As leis da arquitetura frugal](#)
- [All Things Distributed](#)

# Links

- [Sobre a talk Building Systems that Last do Werner Vogels](#)
- [As leis da arquitetura frugal](#)
- [All Things Distributed](#)
- [Building Systems that Last](#)

# Contato

`elton@minetto.dev`

`https://eltonminetto.dev`