

Engenharia de Software

Engenharia de Software Experimental
Prof. Dr. Erik Aceiro Antonio

Linhas de pesquisa

Engenharia de Software

- Engenharia de Software Experimental
- Arquitetura
- Modelos
- Avaliação de Software

Engenharia de Software Não Convencional

- Engenharia de software aplicada a Bioinformática
- WebLab

Objetivos

- Conceitos de Engenharia de Software Experimental
 - Objetivos
 - Vocabulário usado na Engenharia de Software Experimental
 - Tipos de Experimentos
 - Hipóteses
 - Tipos de Erros em Experimentos
 - Pilares para se organizar um Estudo Experimental (Design)
 - Tipos de Medidas
- GQM
- Tipos de Estudos em Engenharia de Software Experimental
- Estatística Descritiva e Inferência na Engenharia de Software Experimental
- Exemplo no Laboratório (vamos para lá depois)

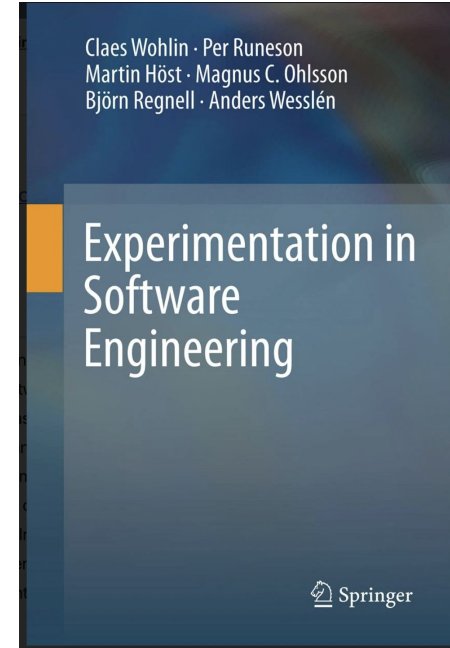
Conceitos de Engenharia de Software Experimental

Design Experimental

- Consiste na elaboração de um protocolo e metodologia para avaliação de um objeto de estudo com base em formalismo e estratégias da Engenharia de Software Experimental

Engenharia de Software Experimental

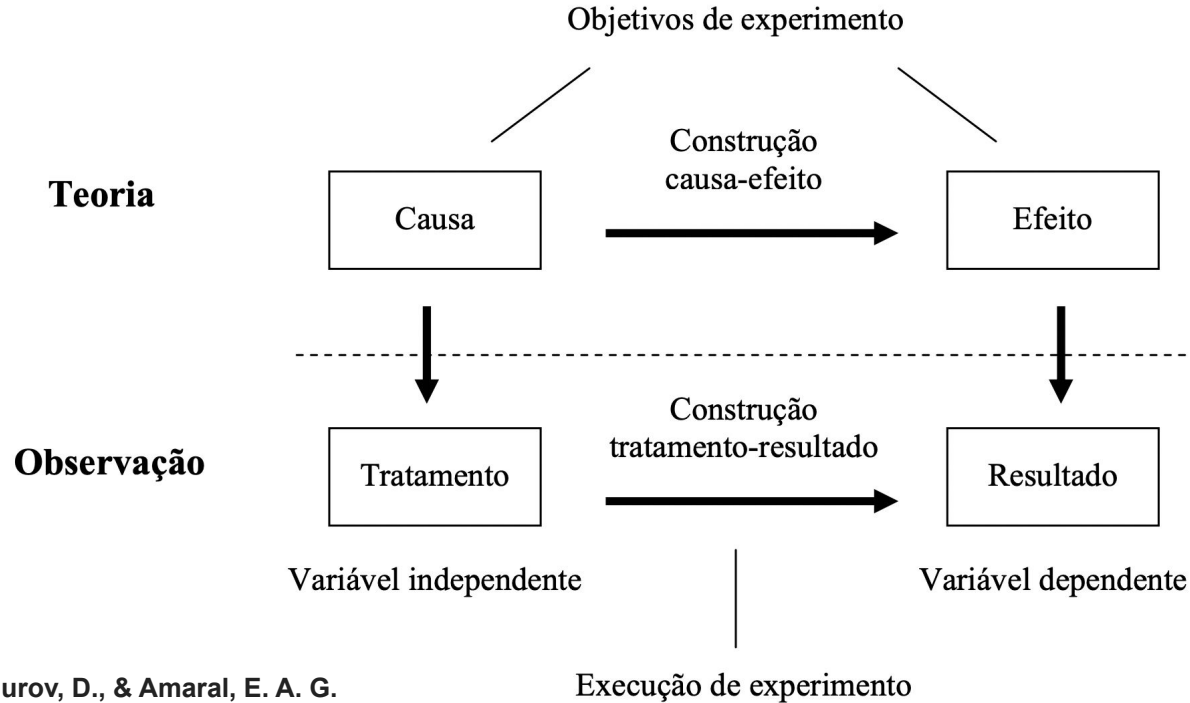
- A Engenharia de Software Experimental é um ramo da Engenharia de Software que provê um conjunto de métodos e estratégias para guiar o pesquisador e a indústria na avaliação formal de estudos e projetos.
- Segundo **Claes Wohlin (2012)** existem quatro métodos para se conduzir experimentos na Engenharia de Software
 - 1. Científico,
 - 2. Engenharia,
 - 3. Experimental, e
 - 4. Analítico
- No contexto da Engenharia de Software o método recomendado para se avaliar e propor um novo modelo é a Engenharia de Software Experimental.



Objetivos da Experimentação na Engenharia de Software

- Para compreender a natureza dos processos da informação dos pesquisadores devem observar o fenômeno, encontrar explicação, formular a teoria, e verificá-la.
 - Ajudar a construir uma base de conhecimento confiável e reduzir assim incerteza sobre quais teorias, ferramentas, e metodologias são adequadas.
 - Auxiliar a levar a novos e úteis meios da introspecção, e abrir novas áreas de investigação.
 - Encontrar novas áreas onde a engenharia age lentamente.
 - Acelerar o processo eliminando abordagens inúteis e suposições errôneas. A experimentação ajuda também a orientar a engenharia e a teoria nas direções promissoras de pesquisa.
- ~ ~ ~ ~
- Experimentos podem ser custosos, mas um experimento significativo geralmente pode se encaixar no orçamento de um pequeno laboratório
 - O crescimento do número de trabalhos científicos com uma validação empírica significativa possui a boa chance de acelerar o processo de formação da Engenharia de Software como ciência. As idéias duvidosas serão rejeitadas mais rapidamente e os pesquisadores poderão concentrar-se nas abordagens promissoras.
 - A tecnologia vem se modificando rapidamente. As mudanças sempre trazem ou eliminam as suposições. Os pesquisadores devem então antecipar as mudanças nas suposições e aplicar os experimentos para explorar as conseqüências dessas mudanças.

Vocabulário de um Experimento em Engenharia de Software



Tipos de Experimentos em Engenharia de Software

In-vitro vs. *In-vivo*.O primeiro refere-se à experimentação no laboratório sob as condições controladas. O segundo considera o estudo de um projeto real.

Alunos vs. Profissionais.....Define a equipe que vai executar o experimento.

Problema de sala de aula vs.

Problema realMostra o tamanho do problema que está sendo estudado.

Específico vs. Geral.Mostra se os resultados do experimento são válidos para um contexto particular ou para o domínio da Engenharia de Software inteiro.

Travassos, G. H., Gurov, D., & Amaral, E. A. G.
G. (2002). *Introdução à engenharia de software experimental* (Vol. 9). UFRJ.

Hipóteses Nula e Hipóteses Alternativas

“

Um experimento geralmente é formulado através de hipóteses. A hipótese principal se chama hipótese nula e declara que não há nenhum relacionamento estatisticamente significativo entre a causa e o efeito.

“

O objetivo principal do experimento é, então, rejeitar a hipótese nula a favor de uma ou algumas hipóteses alternativas. A decisão sobre rejeição da hipótese nula pode ser tomada baseado nos resultados da sua verificação utilizando um teste estatístico.

Erros estatísticos

Erro do Tipo I

- Usar um teste estatístico que indica um relação mesmo que na prática não exista
- Também conhecido como Falso-Positivo

$$P(\text{type-I-error}) = P(H_0 \text{ é rejeitada} | H_0 \text{ é verdadeira})$$

Erro do Tipo II

- Usar um teste estatístico que não indica uma relação mas que na prática existe
- Também conhecido como Falso-Negativo

$$P(\text{type-II-error}) = P(H_0 \text{ não é rejeitada} | H_0 \text{ é falsa})$$

Pilares para a Organização de Experimentos em Engenharia de software

- Aleatoriedade
 - Estratégia usada para a seleção dos participantes com menor efeito (preferível zero) de intervenção do investigador ou pesquisador
- Agrupamento
 - Estratégia usada para a definição de grupos
- Balanceamento
 - Quantidade de participantes dentro de um agrupamento indicando se existe um balanceamento (igualdade) ou não
- Repetível
 - Diz-se de um experimento que possa ser facilmente repetível
- Empacotamento
 - Diz-se do modo de distribuição do experimento para que outras pessoas possam re-executar o experimento e chegar ou não nos mesmos resultados encontrados

Tipos de Medidas

Table I: Primary measurement scales

Scale	Basic characteristics	Common examples	Marketing example	Permissible statistics	
				Descriptive	Inferential
Nominal	Numbers identify and classify objects Yes/No	Student registration numbers, Country of origin	Classification, bank types Gender	Percentages Mode	Chi-square Binomial test
Ordinal	Numbers indicate the relative position of objects but not the magnitude of difference between them	Rankings of the top four teams in the football World cup	Ranking of service quality delivered by a number of hotels /banks. Rank order of the top best 100 universities	Percentile, Median	Rank order correlation Friedman ANOVA
Interval	Difference between objects can be compared; zero point is arbitrary	Temperatures	Attitudes, opinions, index numbers, consumer behaviour	Range, Mean Standard deviation	Product moment correlations, <i>t</i> -tests, ANOVA Regression Factor analysis
Ratio	Zero point is fixed ratios of scale values can be computed	Length, weight	Age income cost sales, market share	Geometric mean (centre number) Harmonic mean	Coefficient of variation

Kataike, Joanita. (2017). What Junior Researchers Must Know Before and After Data Collection: Difference between Parametric and Nonparametric Statistics. 10.21275/ART20173571.

Dimensões e Atributos

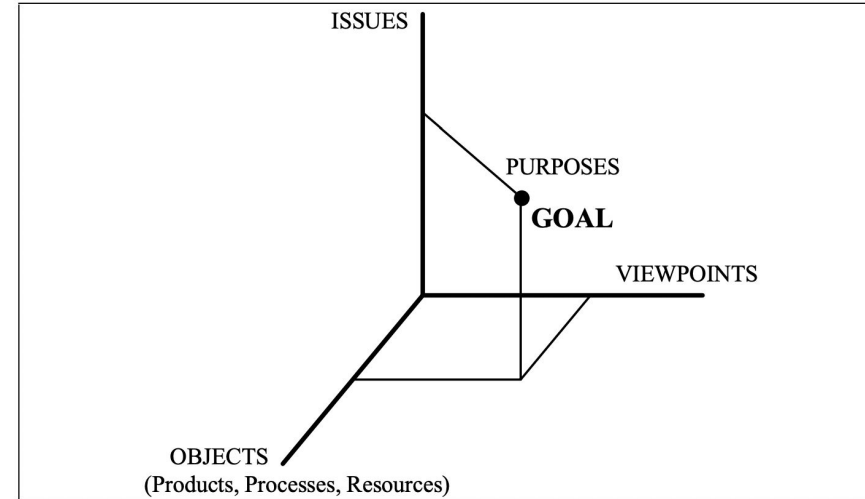
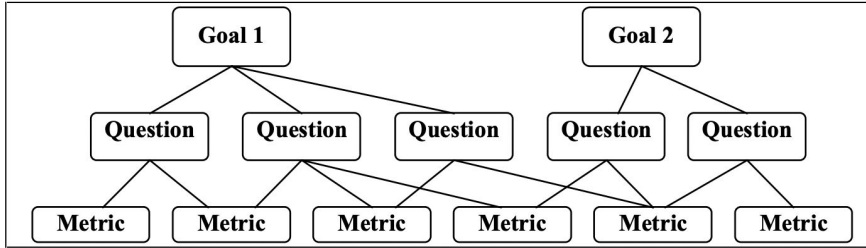
Dimensões das Medidas

- Objetiva
- Subjetiva
- Direta
- Indireta

Atributos

- Interno
- Externos

GQM - Goal Question Metric (1/2)

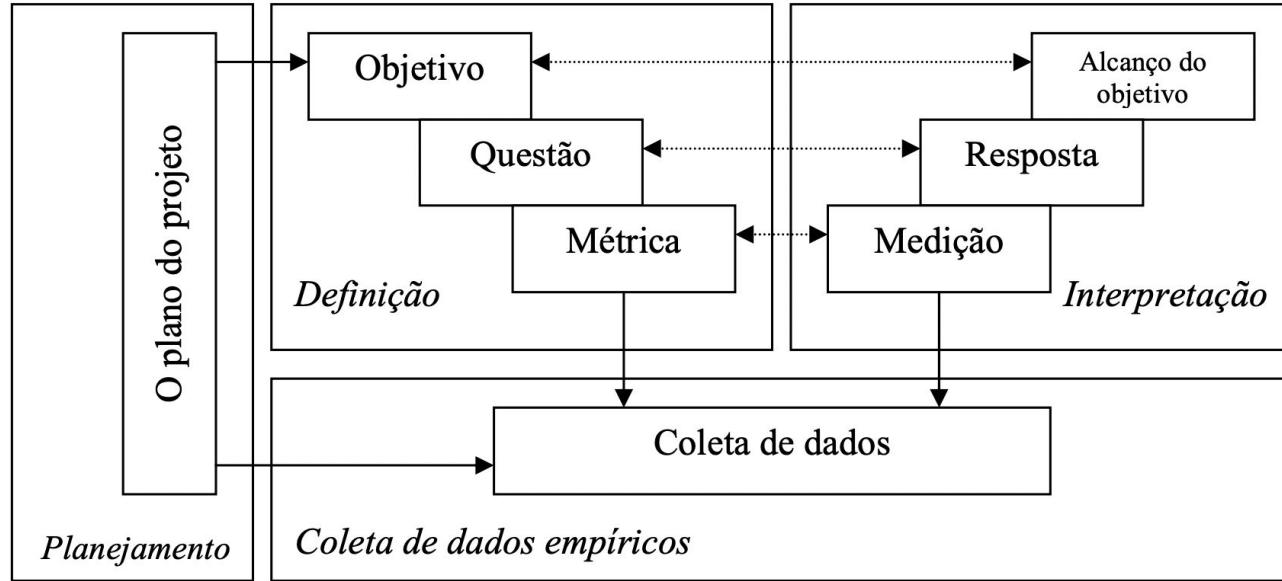


Basili, V., Van Solingen, R., Caldiera, G., & Rombach, H. D. (2002). Goal question metric (GQM) approach. *Encyclopedia of software engineering*.

GQM (2/2)

Goal	Purpose Issue Object (process) Viewpoint	Improve the timeliness of change request processing from the project manager's viewpoint
Question		What is the current change request processing speed?
Metrics		Average cycle time Standard deviation % cases outside of the upper limit
Question		Is the performance of the process improving?
Metrics		$\frac{\text{Current average cycle time}}{\text{Baseline average cycle time}} * 100$ Subjective rating of manager's satisfaction

Processo GQM



Template/Scaffolding do GQM

Analisar<Objeto do estudo>

Especifica entidades que serão estudadas ao longo do experimento.

Com o propósito de<Objetivo>

Define a intenção da realização do experimento. Geralmente é escolhida da lista enumerada: caracterizar, avaliar, prever, controlar, ou melhorar.

Com respeito a<O foco da qualidade>

Indica o principal aspecto da qualidade que está sendo estudado, por exemplo, eficiência, confiabilidade, produtividade.

Do ponto de vista<Perspectiva>

Especifica o ponto de vista que os resultados do experimento serão interpretados, por exemplo, desenvolvedor, consumidor, gerente.

No contexto de<Contexto>

Especifica o ambiente onde o experimento está sendo executado.

Sobre a validade do estudo

Validade

- a validade de conclusão,
 - Relação de validade sobre o entendimento do resultado
- a validade interna,
 - Relação de causa-efeito e teoria e prática
- a validade de construção, e
 - Relação de construção do design experimental e dos participantes
- a validade externa.
 - Relação sobre a validade de generalização

Tipos de Estudos

Fator	<i>Survey</i>	Estudo de caso	Experimento
O controle da execução	Nenhum	Nenhum	Tem
O controle da medição	Nenhum	Tem	Tem
O controle da investigação	Baixo	Médio	Alto
Facilidade da repetição	Alta	Baixa	Alta
Custo	Baixo	Médio	Alto

Travassos, G. H., Gurov, D., & Amaral, E. A. G. G. (2002). *Introdução à engenharia de software experimental* (Vol. 9). UFRJ.

Classificação dos Estudos Experimentais na Engenharia de Software

- Qualitativa
- Quantitativo
- *Benchmarking*

Tipos de Testes Estatísticos

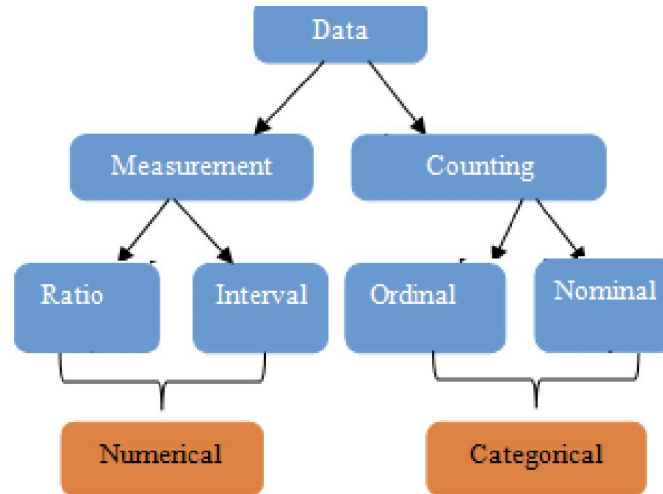


Figure I: Classifying levels of measurement scale

Source: Author's compilation based on Norman [18] and Fellows and Liu [19]

Kataike, Joanita. (2017). What Junior Researchers Must Know Before and After Data Collection: Difference between Parametric and Nonparametric Statistics. 10.21275/ART20173571.

Tipos de Testes Estatísticos

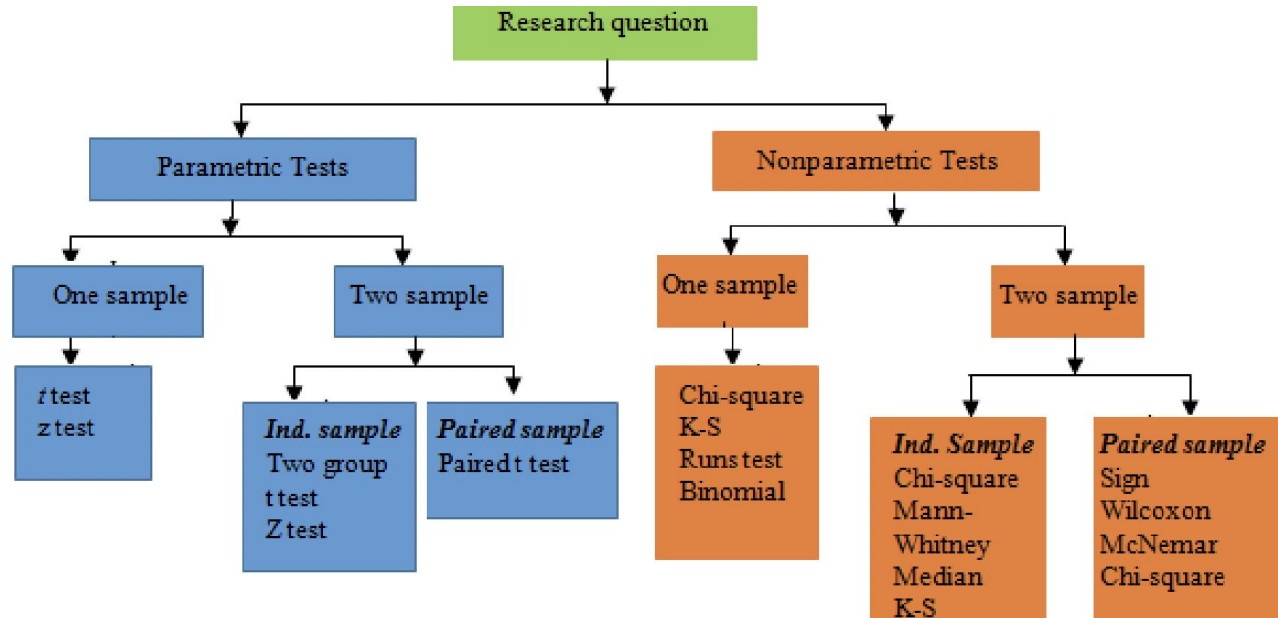


Figure II: Classification of a one and two sample tests

Sources: adapted from [8, 13]

Kataike, Joanita. (2017). What Junior Researchers Must Know Before and After Data Collection: Difference between Parametric and Nonparametric Statistics. 10.21275/ART20173571.

Referências

1. [Basic of Software Engineering Experimentation](#)
2. [Introdução à Engenharia de Software Experimental](#)
3. [The Goal Question Metric Approach](#)
4. [Experimental Software Engineering](#)
5. [Parametric x Non Parametric](#)

Prática - Laboratório DEMAC
Engenharia de Software Experimental
Exemplo de Design Experimental

Prof. Erik Aceiro Antonio

Estudo Exploratório para Avaliar a carga Carga Cognitiva

(i) Objetivo Geral

Definir se a Taxonomia de Carga Cognitiva aplica-se na avaliação de propósitos de código base.

(ii) Objetivo Específico

Tendo como visão a classificação de propósitos, avaliar:

Quais são os tipos de propósitos específicos da Taxonomia de Carga Cognitiva que se relacionam com código fonte usando Clean-Code e Design Patterns GoF ?

Objetivo do estudo usando o GQM

Goal (G)

Definir se a Taxonomia de Carga Cognitiva aplica-se na avaliação de propósitos de código base

Question (Q)

Existem diferenças nos propósitos da Taxonomia de Carga Cognitiva na leitura de código fonte COM ou SEM Clean-Code e Design Patterns GoF ?

Metric (M1)

M1: Quantidade de propósitos específicos da Taxonomia de Carga Cognitiva que se relacionam com Clean-Code ?

Metric (M2)

M2: Quantidade de propósitos específicos da Taxonomia de Carga Cognitiva que se relacionam com Design Patterns GoF ?

Metric (M3)

M3: Diferenças entre os propósitos específicos da Taxonomia de Carga Cognitiva ao usar código conte com Clean-Code e Design Patterns GoF e sem ?

Template GQM

Analisar os propósitos da Taxonomia de Carga Cognitiva

Com o propósito de caracterizar

Com respeito ao uso da Taxonomia

Do ponto de vista do desenvolvedor de software

No contexto de alunos de curso básico de Engenharia de Software

Hipóteses Nula e Alternativa

Hipótese Nula (H0): Não existem diferenças entre usar código fonte COM e SEM Clean-Code e Design-Pattern GoF baseado na Taxonomia de Carga Cognitiva

Hipótese Alternativa (H1): Existem diferenças significativas entre usar código fonte COM e SEM Clean-Code e Design-Pattern GoF baseado na Taxonomia de Carga Cognitiva

~~~~

# Seleção dos participantes

- Alunos do 4. período de Ciência da Computação
- Conhecimentos prévios em Análise de Sistema e Engenharia de Software
- Grupos de Tratamento G1 e G2
  - G1 com código fonte em Python COM Clean-Code e GoF
  - G2 com código fonte em Python SEM Clean-Code e GoF
- Total de estudantes no Grupo
  - G1 = 10
  - G2 = 10

# Artefatos utilizados - Instrumentação do Experimento

- Apresentação sobre Clean-Code e Design Patterns
  - ~ 10 min
    - Clean-Code / Design Patterns - GoF
  - ~ 5 min
    - Benefícios, vantagens e desvantagens
- Separação e Composição dos Grupos
  - ~ 5 min
    - Separação dos grupos G1 e G2
- Link por QR-CODE para os Grupos G1 e G2
  - ~ 20 min
    - Leitura do código fonte usando *Top-Down Abstraction*
- Link por QR-CODE para o questionário
  - ~ 10 min
    - Preenchimento

# Repositórios - Toy Problem em Python ~ 20 min

G1: com Clean-Code / GoF

[shorturl.at/dlnVX](https://shorturl.at/dlnVX)



G2: sem Clean-Code / GoF]

[shorturl.at/eltV3](https://shorturl.at/eltV3)



**Questionário ~ 10 min**

