

- fase 2
  - FOCO DINAMICA
    - Particao de Equivalencia
    - POISED
      - api 1 - book
      - api 2 - ServeRest - paulo
  - mandar links
    - Tabela de Decisao
    - Tabela Verdade
    - Analise de Risco
  - HELPS

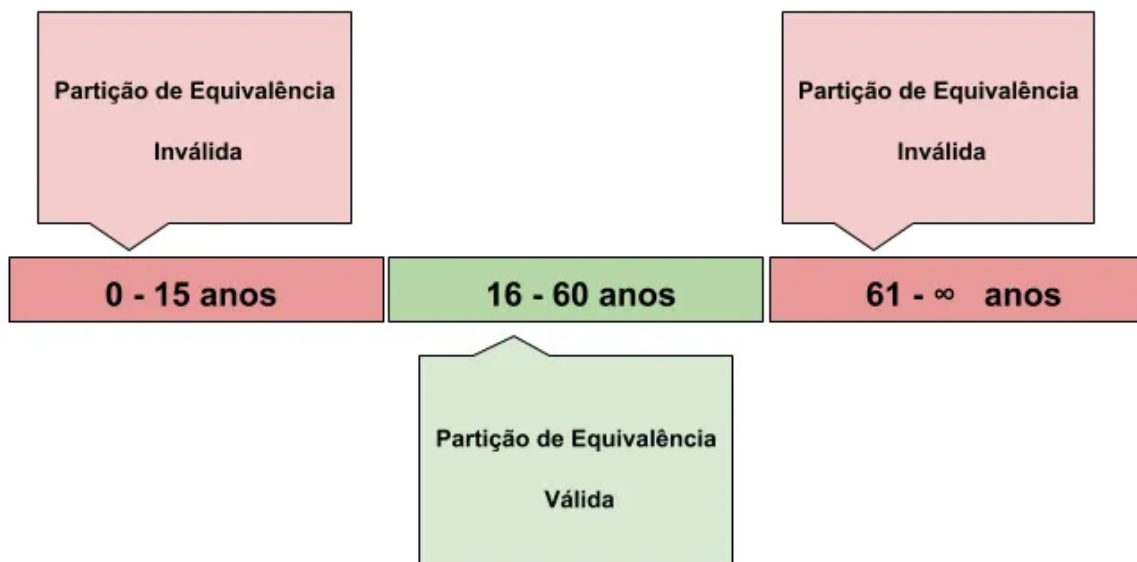
## fase 2

### FOCO DINAMICA

#### Particao de Equivalencia

**nao pode**    **necessita permissao pais**    **pode**

0 a 15                      16 a 18                      19 ou mais



1. testes que fariamos nesse caso...

idade	mensagem	status
14	nao pode	
15	nao pode	
16	necessita permissao pais	

idade	mensagem	status
17	necessita permissao pais	
18	pode	
19	pode	
80	pode	

## POISED

Tipo	detalhe	exemplo **
Parameters	campos enviados na requisicao / corpo	campos body envio
Output	validar retorno	campos / json retorno, statusCode, mensagens
Interop	comunicacao do sistema	preenche valor valido, sistema retorna sucesso / como esperado (exemplo 10x)
Security	login / token	token valido, invalido, sem informar, outra pessoa, scripts...
Errors	erros / excessao	excessoes, erros 500, exception, deletar registo 2x
Data		validar dados criados / retornados (***)

**Interop** = Interoperabilidade

### 1. Usar técnica de POISED:

Aqui vamos ver quais os cenarios mais criticos de uma api com base em tecnicas, que nos ajudam a extrair os cenarios mais criticos, que mais dao problema em uma api:

**Parametros** (campos enviados na requisicao / corpo): **Saidas** (validar retorno):

**Interoperabilidade** (comunicacao do sistema) **Seguranca:** **Erros:** **Data** (validar dados criados / retornados):

- Criar um teste POST, criar recurso
  - validar dados enviados versus retornados no response
  - Fazer chamada GET por id, validar dados retornados

<https://github.com/cursodsousa/library-api>

**api 1 - book**

<https://github.com/mwinteringham/restful-booker>

mongo

- Run `docker run -d -p 27017:27017 --name example-mongo mongo:latest`

<https://www.howtogeek.com/devops/how-to-run-mongodb-in-a-docker-container/>

Or you can run this via Docker:

1. Clone the repo
2. Navigate into the restful-booker root folder
3. Run `docker-compose build`
4. Run `docker-compose up`
5. APIs are exposed on <http://localhost:3001/apidoc/index.html>

## api 2 - ServeRest - paulo

<https://github.com/ServeRest/ServeRest>

<http://serverest.dev>

## mandar links

---

### Tabela de Decisao

Ela nos ajuda a extrair testes de documentação / requisitos

TIPO	VARIÁVEIS	PARTIÇÕES	T1	T2	T3	T4
Entrada	Variável 1	Sim	•	•		
		Não			•	•
	Variável 2	>100	•		•	
		<=100		•		•
Saída	Ação de Saída	Sim	✓			
		Não		✓	✓	✓

Imagem extraída do artigo: <https://medium.com/revista-tspi/a-técnica-de-tabela-de-decisão-no-mundo-dos-testes-29d3a332473c>, acessado em 14/10/2021 as 19:00

Quintas na Prática: Criando Tabelas de Decisão com x|decision -> <https://youtu.be/3WtavNon34A> Ferramenta xdecision: [juliodelima.com.br/xdecision](http://juliodelima.com.br/xdecision)

### Tabela Verdade

opcao1	opcao2	resultado
V	V	✓
V	F	✓
F	V	✓
F	F	✓

quero saber mais sobre: <https://www.significados.com.br/tabela-verdade/>

## **Analise de Risco**

<https://medium.com/revista-tspi/testes-baseados-em-riscos-risk-based-testing-b7dfa751ec17>

## **HELPS**

---