

Automated Tests

Testes automatizados são qualquer tipo de verificação do comportamento de um sistema executados automaticamente.

Por que utilizar testes automatizados?

- Documentação de código
- Safety Net
- Aumento de produtividade

Documentação de código

```
it(  
  "should return empty list when teacher has no associated curricular component",  
  async () => { ...  
}  
)  
  
it(  
  "should return one curricular component when teacher is associated",  
  async () => { ...  
}  
)  
  
it(  
  "should return many curricular components when teacher is associated",  
  async () => { ...  
}  
)
```

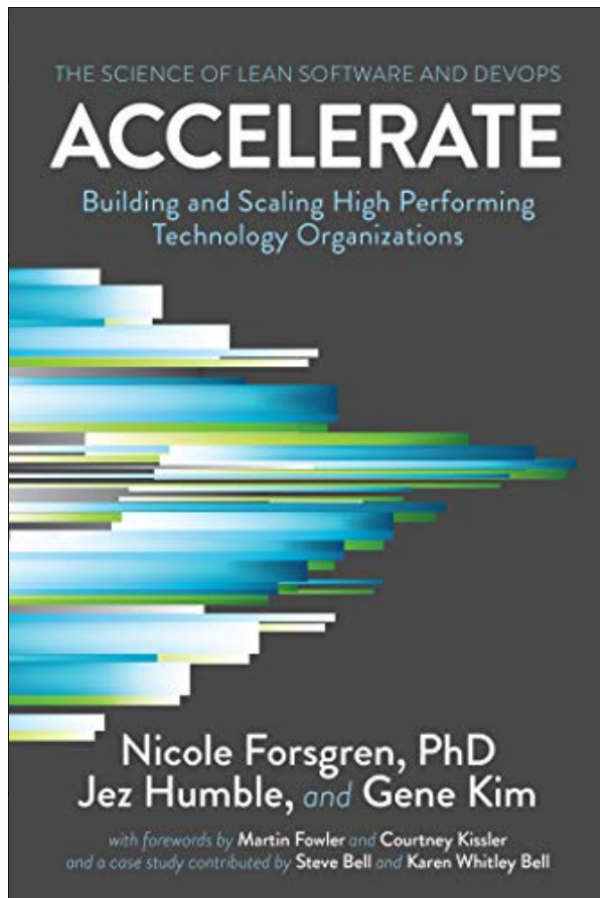
Safety Net

```
bbdac@DESKTOP-RAPR71V C:\gh_branding\edusfera-frontend develop
# npm run test:unit

> edusfera-front@0.1.0 test:unit C:\gh_branding\edusfera-frontend
> vue-cli-service test:unit

PASS src/helpers/tests/ListComparer.spec.js
PASS src/helpers/tests/ObjectOperations.spec.js
PASS src/store/user/tests/UserCacheManager.spec.js
PASS src/services/tests/ShiroTaxonomyFilter.spec.js
PASS src/views/Teachers/Theme/tests/ThemeLoadingTest.spec.js
PASS src/helpers/tests/ListOperations.spec.js
PASS src/views/Teachers/Theme/tests/ThemeTeacherVisionButtonTest.spec.js
PASS src/services/tests/ShiroTeacherService.spec.js
PASS src/views/InAction/tests/ActionGroupsCacheService.spec.js
PASS src/services/tests/ShiroServiceTest.spec.js
```

Aumento de produtividade



Times que empregam técnicas como TDD, Continuous integration and delivery gastam 44 por cento mais tempo em trabalho útil (novas funcionalidades) do que corrigindo bugs

Tipos

- Integration Tests
- Security Tests
- Performance Tests
- Acceptance Tests
- Quality Assurance Tests

Fases

- Unit
- API (Application Programming Interface)
- UI (User interface)

Testes: O que são? Como vivem? O que fazem?

- Arrange (Preparação), Act (Ação), and Assert (Verificação)

```
test('should add two numbers and return sum value', () => {  
  // arrange  
  const calculator = new Calculator()  
  
  // act  
  const result = calculator.sum(1, 2)  
  
  // assert  
  expect(result).toBe(3);  
})
```


Tipos mais comuns de verificações

<code>expect(result).toBe(number)</code>	<code>// Espera um valor igual</code>
<code>expect(result).not.toBe(number)</code>	<code>// Espera um valor diferente</code>
<code>expect(result).toBeGreaterThan(number bigint)</code>	<code>// Espera um valor maior</code>
<code>expect(result).toBeLessThan(number bigint)</code>	<code>// Espera um valor menor</code>
<code>expect(method()).toThrow(error?)</code>	<code>// Espera que method lance uma exceção</code>

Code Coverage

Apresenta uma visão gráfica da quantidade de código submetido a testes

- **Observação:** Não deve ser utilizada isoladamente como métrica de qualidade de código
- Utilizar Mutation e outros tipos de técnicas para garantir a qualidade

Exemplo de um relatório de cobertura de código

Coverage									
Collapse all Expand all		By assembly		Compare with: Date		Filter:			
		Grouping:							
▼ Name	▼ Covered	▼ Uncovered	▼ Coverable	▼ Total	▼ Line coverage	▼ Covered	▼ Total	▼ Branch coverage	
+ Assembly-CSharp	5	431	436	1027	1.1% <div></div>	0	0		
+ Assembly-CSharp-Editor	0	100	100	160	0% <div></div>	0	0		
+ Items	63	3	66	168	95.4% <div></div>	0	0		
+ Player	61	356	417	168	14.6% <div></div>	0	0		
+ UnityFoundation.BuildingPlacementSystem	0	224	224	245	0% <div></div>	0	0		
+ UnityFoundation.CameraScripts	0	445	445	297	0% <div></div>	0	0		
+ UnityFoundation.CarSystem	0	439	439	604	0% <div></div>	0	0		
+ UnityFoundation.Character2D	0	428	428	469	0% <div></div>	0	0		
+ UnityFoundation.Character3D	0	768	768	1182	0% <div></div>	0	0		
+ UnityFoundation.Code	418	1631	2049	2089	20.4% <div></div>	0	0		
+ UnityFoundation.DialogueSystem	0	438	438	551	0% <div></div>	0	0		
+ UnityFoundation.DialogueSystem.Editor	0	1178	1178	1514	0% <div></div>	0	0		
+ UnityFoundation.Editor	0	485	485	1416	0% <div></div>	0	0		
+ UnityFoundation.EditorInspector	0	257	257	1477	0% <div></div>	0	0		
+ UnityFoundation.HealthSystem	78	526	604	662	12.9% <div></div>	0	0		
+ UnityFoundation.HealthSystem.Editor	0	142	142	153	0% <div></div>	0	0		
+ UnityFoundation.LeaderBoardSystem	0	114	114	134	0% <div></div>	0	0		
+ UnityFoundation.ObjectPooling	0	214	214	259	0% <div></div>	0	0		
+ UnityFoundation.PathFinder	0	216	216	216	0% <div></div>	0	0		
+ UnityFoundation.ProceduralGeneration	0	134	134	117	0% <div></div>	0	0		
+ UnityFoundation.SavingSystem	0	24	24	32	0% <div></div>	0	0		
+ UnityFoundation.SettingsSystem	0	180	180	188	0% <div></div>	0	0		
+ UnityFoundation.UI	0	749	749	899	0% <div></div>	0	0		

Exemplo de um arquivo específico no relatório de cobertura de código

File(s)			Methods/Properties
C:/Users/bbdac/Documents/projects/zombieluca/Assets/UnityFoundation/Code/Common/LinearInterpolation/LerpByTime.cs			
#	Line	Line coverage	
	1	using UnityEngine;	StartValue()
	2		EndValue()
	3	namespace UnityFoundation.Code	CurrentValue()
	4	{	CurrentValue(System.Single)
	5	public class LerpByTime	CurrentTime()
	6	{	CurrentTime(System.Single)
	7	public float StartValue { get; }	Time()
4	8	public float EndValue { get; }	LerpByTime(System.Single, System.Sing...
4	9	public float CurrentValue { get; private set; }	Eval(System.Single)
12	10	public float CurrentTime { get; private set; }	
12	11	public float Time { get; }	
4	12		
	13	public LerpByTime(float startValue, float endValue, float time)	
3	14	{	
3	15	StartValue = startValue;	
3	16	CurrentValue = startValue;	
3	17	EndValue = endValue;	
3	18	Time = time;	
3	19	}	
	20		
	21	public void Eval(float timeAmount)	
4	22	{	
4	23	CurrentTime += timeAmount;	
4	24	CurrentValue = Mathf.Lerp(
	25	StartValue,	
	26	EndValue,	
	27	CurrentTime / Time	
	28);	
4	29	}	
	30	}	
	31	}	

Exemplos

- ex_1: Testes unitários 101
- ex_2: GUI testes unitário utilizando Veu 3 + Vue-Jest

Dever de casa

- Implementar testes unitários para 50% de cobertura de linhas do código.
- Apresentar exemplos de testes implementados
 - Qual tipo de teste foi implementado?
 - Qual tecnologia foi utilizada para o desenvolvimento dos testes?
 - Quais foram as principais dificuldades?
- **(Opcional)** Mentorados podem implementar outros testes automatizados
 - Testes de performance
 - Testes de segurança
 - Testes de qualidade de código

Bibliografia

- [Documentação do Jest](#)
- [Clean Code Class 4](#)
- [Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations](#)