

Universidade de Brasília – UnB  
Faculdade UnB Gama – FGA  
Engenharia de Software

# **IcePatterns: Um Site Educacional de Padrões de Projeto Orientado a uma Abordagem de Aprendizagem Gamificada**

**Autores:** Aline Helena Lermen e Davi Marinho da Silva Campos  
**Orientadora:** Profa. Dra. Milene Serrano  
**Coorientador:** Prof. Dr. Maurício Serrano

Brasília, DF  
2023



Aline Helena Lermen e Davi Marinho da Silva Campos

# **IcePatterns: Um Site Educacional de Padrões de Projeto Orientado a uma Abordagem de Aprendizagem Gamificada**

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Universidade de Brasília – UnB

Faculdade UnB Gama – FGA

Orientador: Profa. Dra. Milene Serrano

Coorientador: Prof. Dr. Maurício Serrano

Brasília, DF

2023

---

Aline Helena Lermen e Davi Marinho da Silva Campos

IcePatterns: Um Site Educacional de Padrões de Projeto Orientado a uma Abordagem de Aprendizagem Gamificada/ Aline Helena Lermen e Davi Marinho da Silva Campos. – Brasília, DF, 2023-

148 p. : il. (algumas color.) ; 30 cm.

Orientadora: Profa. Dra. Milene Serrano

Trabalho de Conclusão de Curso – Universidade de Brasília – UnB  
Faculdade UnB Gama – FGA , 2023.

1. Gamificação. 2. Padrões de Projeto. I. Profa. Dra. Milene Serrano. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. IcePatterns: Um Site Educacional de Padrões de Projeto Orientado a uma Abordagem de Aprendizagem Gamificada

CDU 02:141:005.6

---

Aline Helena Lermen e Davi Marinho da Silva Campos

# **IcePatterns: Um Site Educacional de Padrões de Projeto Orientado a uma Abordagem de Aprendizagem Gamificada**

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Trabalho aprovado. Brasília, DF, 22 de Dezembro de 2023:

---

**Profa. Dra. Milene Serrano**  
Orientadora

---

**Prof. Dr. Maurício Serrano**  
Coorientador

---

**Prof. Dr. John Lenon Cardoso  
Gardenghi**  
Convidado 1

---

**Tomáz Felipe Rodrigues Martins**  
(Lead Developer na Todo an InComm  
Payments Company)  
Convidado 2

Brasília, DF  
2023

# Agradecimentos

Agradecemos, primeiramente, a Deus, por nos guiar até onde estamos hoje, pela saúde, e por suas infinitas bênçãos durante toda a graduação.

Agradecemos às nossas famílias, por todo o apoio fornecido durante os anos, por acompanharem nossa caminhada, por ouvirem desabafos e, acima de tudo, por continuarem acreditando em nós.

Agradecemos também aos nossos orientadores, Milene e Maurício, por toda a orientação e atenção. Gratidão por todos os momentos de compreensão e apoio, e por nos guiarem no aprendizado.

Agradecemos aos nossos amigos, pelos momentos compartilhados, pelo apoio e pela motivação.

A todos esses, nossos mais sinceros agradecimentos.

*"Por vezes sentimos que aquilo que fazemos não é senão uma gota de água no mar. Mas o mar seria menor se lhe faltasse uma gota."  
(Madre Teresa de Calcutá)*

# Resumo

Os padrões de projeto, fundamentais na área de engenharia de software, fornecem soluções comuns para problemas de design na programação. Os padrões de projeto conferem uma abordagem estruturada para resolver problemas de projeto específicos, e podem ser aplicados em diferentes cenários, seja na arquitetura de sistemas; no projeto de classes, ou ainda no projeto de métodos. A aprendizagem de padrões de projeto é um processo desafiador para os estudantes, ora desinteressante, ora difícil. A gamificação, compreendendo o emprego de elementos de jogos em contextos não lúdicos, tem se apresentado como uma orientação pertinente para motivação e engajamento em diferentes contextos. Nesse trabalho, elaborou-se uma abordagem gamificada para a aprendizagem de padrões de projeto, sendo os conteúdos apresentados em um site educacional. Portanto, trata-se de uma pesquisa de abordagem qualitativa; natureza aplicada; objetivos exploratórios, e que faz uso de pesquisa bibliográfica e pesquisa-ação como procedimentos.

**Palavras-chave:** Site Educacional, Padrões de Projeto, Abordagem de Aprendizagem, Gamificação.

# Abstract

Design patterns, fundamental in the area of software engineering, provide common solutions to design problems in programming. Design patterns offer a structured approach to solving specific design problems, and can be applied in different scenarios, whether in systems architecture; in the design of classes, or even in the design of methods. Learning design patterns is a challenging process for students, sometimes uninteresting, sometimes difficult. Gamification, comprising the use of game elements in non-ludic contexts, has been presented as a relevant orientation for motivation and engagement in different contexts. In this work, we developed a gamified approach for learning design patterns, with the contents being presented on an educational website. Therefore, it is research with a qualitative approach; applied nature; exploratory objectives, and that makes use of bibliographic research and action research as procedures.

**Key-words:** Educational Website, Design Patterns, Learning Approach, Gamification.



# Lista de ilustrações

Figura 1 – Diagrama de Classes <i>Singleton</i> . . . . .	28
Figura 2 – Exemplo de <i>Singleton</i> . . . . .	28
Figura 3 – Diagrama de Classes <i>Façade</i> . . . . .	30
Figura 4 – Exemplo de <i>Façade</i> . . . . .	31
Figura 5 – Diagrama de Classes <i>Template Method</i> . . . . .	32
Figura 6 – Exemplo de <i>Template Method</i> . . . . .	33
Figura 7 – <i>Octalysis</i> da Gamificação . . . . .	36
Figura 8 – Atividades Realizadas na Internet (Educação e Trabalho) . . . . .	38
Figura 9 – Fluxograma de Atividades e Subprocessos da Primeira Etapa do TCC . . . . .	55
Figura 10 – Fluxograma de Atividades e Subprocessos da Segunda Etapa do TCC . . . . .	56
Figura 11 – Duo, o Mascote do Duolingo . . . . .	60
Figura 12 – Trilha de Níveis da Unidade . . . . .	61
Figura 13 – Cristais e Vidas . . . . .	61
Figura 14 – Tela Inicial do <i>Refactoring Guru</i> . . . . .	63
Figura 15 – Outras Telas do <i>Refactoring Guru</i> . . . . .	63
Figura 16 – <i>Brainstorming</i> . . . . .	65
Figura 17 – Persona Estudante . . . . .	66
Figura 18 – Persona Professor . . . . .	66
Figura 19 – Persona Autodidata . . . . .	67
Figura 20 – Anti-Persona Gerente . . . . .	67
Figura 21 – Anti-Persona Estudante . . . . .	68
Figura 22 – Paleta de Cores . . . . .	76
Figura 23 – Tipografia . . . . .	77
Figura 24 – Ícones . . . . .	77
Figura 25 – Página Inicial de Visitantes . . . . .	78
Figura 26 – Página Inicial para Usuário Registrado . . . . .	79
Figura 27 – Página de Nível e Subníveis . . . . .	79
Figura 28 – Página de Conteúdos . . . . .	80
Figura 29 – Página de Exercícios . . . . .	80
Figura 30 – Página de <i>Feedback</i> dos Exercícios . . . . .	81
Figura 31 – Página da Loja . . . . .	81
Figura 32 – Página das Missões . . . . .	82
Figura 33 – Página de Perfil do Usuário . . . . .	82
Figura 34 – Itens da Barra Superior . . . . .	87
Figura 35 – Cartão Explicativo dos Itens . . . . .	87
Figura 36 – Botão Maior de Nível . . . . .	88

Figura 37 – Botão Menor de Nível . . . . .	88
Figura 38 – Aviso no Centro Superior . . . . .	92
Figura 39 – Botão Maior Modificado . . . . .	92
Figura 40 – Botão Menor Modificado . . . . .	92
Figura 41 – Apresentação de Código Modificada . . . . .	93
Figura 42 – Respostas do Questionário - 1 . . . . .	94
Figura 43 – Respostas do Questionário - 2 . . . . .	95
Figura 44 – Respostas do Questionário - 3 . . . . .	95
Figura 45 – Respostas do Questionário - 4 . . . . .	96
Figura 46 – Respostas do Questionário - 5 . . . . .	96
Figura 47 – Respostas do Questionário - 6 . . . . .	97
Figura 48 – Respostas do Questionário - 7 . . . . .	98
Figura 49 – Respostas do Questionário - 8 . . . . .	99
Figura 50 – Respostas do Questionário - 9 . . . . .	100
Figura 51 – Respostas do Questionário - 10 . . . . .	100
Figura 52 – Respostas do Questionário - 11 . . . . .	101
Figura 53 – Respostas do Questionário - 12 . . . . .	102
Figura 54 – Respostas do Questionário - 13 . . . . .	103
Figura 55 – Protótipo - Página Inicial de Visitantes . . . . .	125
Figura 56 – Protótipo - <i>Login</i> e Registro de Usuário . . . . .	125
Figura 57 – Protótipo - Página Inicial para Usuário Registrado . . . . .	126
Figura 58 – Protótipo - Cartões Explicativos . . . . .	126
Figura 59 – Protótipo - Página dos Níveis . . . . .	127
Figura 60 – Protótipo - Página de Conteúdo . . . . .	127
Figura 61 – Protótipo - Página de Exercícios . . . . .	128
Figura 62 – Protótipo - Página Final de Conteúdo . . . . .	128
Figura 63 – Protótipo - Finalização de Exercícios . . . . .	129
Figura 64 – Protótipo - Página da Loja . . . . .	129
Figura 65 – Protótipo - Página de Perfil do Usuário . . . . .	130
Figura 66 – Protótipo - Página de Missões . . . . .	130
Figura 67 – Exemplo de Código - <i>Singleton</i> . . . . .	134
Figura 68 – Exemplo de Código - <i>Facade</i> . . . . .	136
Figura 69 – Exemplo de Código - <i>Template Method</i> . . . . .	139
Figura 70 – Exemplo de Código - <i>Factory Method</i> - Parte 1 . . . . .	141
Figura 71 – Exemplo de Código - <i>Factory Method</i> - Parte 2 . . . . .	142
Figura 72 – Exemplo de Código - <i>Adapter</i> . . . . .	144
Figura 73 – Código do Exercício - Desafio . . . . .	147

# Lista de tabelas

Tabela 1 – Ferramentas de Apoio ao Trabalho . . . . .	47
Tabela 2 – Cronograma da Primeira Etapa do TCC . . . . .	56
Tabela 3 – Cronograma da Segunda Etapa do TCC . . . . .	57
Tabela 4 – Aplicação do <i>Framework</i> SUS . . . . .	90
Tabela 5 – Relação entre Padrões Iniciais do Iceberg e Reconhecimento deles pelos Participantes . . . . .	99
Tabela 6 – Comparação de Outros Meios em Relação ao Uso de Sites . . . . .	104

# Lista de quadros

Quadro 1 – Principais Padrões de Projeto . . . . .	26
Quadro 2 – Classificação da Pesquisa . . . . .	48
Quadro 3 – Principais Elementos de Gamificação do Duolingo . . . . .	62
Quadro 4 – Heurísticas para Avaliações de Aplicações Gamificadas . . . . .	113

# Lista de abreviaturas e siglas

Cetic.br	Centro Regional de Estudos para o Desenvolvimento da Sociedade da Informação
GOF	<i>Gang of Four</i>
OOPSLA	<i>Object-oriented Programming, Systems, Languages, and Applications</i>
TCC	Trabalho de Conclusão de Curso
TIC	Tecnologia de Informação e Comunicação
UnB	Universidade de Brasília
XP	<i>Extreme Programming</i>

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>18</b>
1.1	Contextualização	18
1.2	Justificativa	20
1.3	Questões de Pesquisa e Desenvolvimento	20
1.4	Objetivos	21
1.4.1	Objetivos Específicos	22
1.5	Organização da Monografia	23
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	<b>24</b>
2.1	Padrões de Projeto	24
2.1.1	Padrões Criacionais	26
2.1.1.1	<i>Singleton</i>	27
2.1.2	Padrões Estruturais	27
2.1.2.1	<i>Façade</i>	29
2.1.3	Padrões Comportamentais	29
2.1.3.1	<i>Template Method</i>	30
2.2	Contexto Educacional	34
2.3	Gamificação	35
2.4	Site Educacional	37
2.5	Resumo do Capítulo	41
<b>3</b>	<b>SUPORTE TECNOLÓGICO</b>	<b>42</b>
3.1	Ferramentas de Apoio Geral	42
3.1.1	Overleaf	42
3.1.2	Zotero	42
3.1.3	Miro	42
3.1.4	Ferramentas de Comunicação	43
3.2	Ferramentas de Apoio ao Site Educacional	43
3.2.1	Figma	43
3.2.2	React 18.2	43
3.2.3	Node 18.16	44
3.2.4	Visual Studio Code 1.74.3	44
3.2.5	Git 2.39.1	44
3.2.6	GitHub	45
3.2.7	Trello	45
3.2.8	Netlify	45

3.2.9	Heroku . . . . .	45
3.2.10	LucidChart . . . . .	46
3.2.11	Google Forms . . . . .	46
<b>3.3</b>	<b>Resumo do Capítulo . . . . .</b>	<b>46</b>
<b>4</b>	<b>METODOLOGIA . . . . .</b>	<b>48</b>
<b>4.1</b>	<b>Classificação de Pesquisa . . . . .</b>	<b>48</b>
4.1.1	Abordagem da Pesquisa . . . . .	48
4.1.2	Natureza da Pesquisa . . . . .	49
4.1.3	Objetivos da Pesquisa . . . . .	49
4.1.4	Procedimentos da Pesquisa . . . . .	49
<b>4.2</b>	<b>Levantamento Bibliográfico . . . . .</b>	<b>49</b>
<b>4.3</b>	<b>Método de Desenvolvimento . . . . .</b>	<b>50</b>
4.3.1	Scrum . . . . .	50
4.3.2	XP . . . . .	51
4.3.3	Kanban . . . . .	51
4.3.4	Método Híbrido . . . . .	51
<b>4.4</b>	<b>Método de Análise de Resultados . . . . .</b>	<b>52</b>
<b>4.5</b>	<b>Fluxo de Atividades . . . . .</b>	<b>53</b>
4.5.1	Atividades e Subprocessos - Primeira Etapa do TCC . . . . .	53
4.5.2	Atividades e Subprocessos - Segunda Etapa do TCC . . . . .	55
<b>4.6</b>	<b>Cronogramas . . . . .</b>	<b>56</b>
<b>4.7</b>	<b>Resumo do Capítulo . . . . .</b>	<b>57</b>
<b>5</b>	<b>ABORDAGEM E SITE EDUCACIONAL . . . . .</b>	<b>58</b>
<b>5.1</b>	<b>Contextualização . . . . .</b>	<b>58</b>
<b>5.2</b>	<b><i>Benchmarking</i> . . . . .</b>	<b>59</b>
5.2.1	Duolingo . . . . .	59
5.2.1.1	Principais Recursos do Duolingo . . . . .	60
5.2.1.2	Gamificação . . . . .	60
5.2.2	Refactoring Guru . . . . .	62
5.2.2.1	Principais Recursos do <i>Refactoring Guru</i> . . . . .	62
<b>5.3</b>	<b><i>Brainstorming</i> . . . . .</b>	<b>64</b>
<b>5.4</b>	<b>Personas . . . . .</b>	<b>65</b>
<b>5.5</b>	<b>IcePatterns . . . . .</b>	<b>68</b>
5.5.1	Visão Geral . . . . .	68
5.5.2	Abordagem Gamificada . . . . .	69
5.5.2.1	Plano de Gamificação . . . . .	69
5.5.2.1.1	1 - Significado Épico e Chamado . . . . .	69
5.5.2.1.2	2 - Desenvolvimento e Realização . . . . .	70

5.5.2.1.3	3 - Empoderamento da Criatividade e <i>Feedback</i>	71
5.5.2.1.4	4 - Propriedade e Posse	71
5.5.2.1.5	5 - Influência Social e Pertencimento	72
5.5.2.1.6	6 - Escassez e Impaciência	72
5.5.2.1.7	7 - Imprevisibilidade e Curiosidade	73
5.5.2.1.8	8 - Perda e Rejeição	73
5.5.3	Site Educacional	74
5.5.3.1	Usabilidade	74
5.5.3.2	Design	75
5.5.3.2.1	Cores	75
5.5.3.2.2	Layout e Design	75
5.5.3.2.3	Tipografia	76
5.5.3.2.4	Iconografia	76
5.5.3.3	Protótipo do Site Educacional	76
5.5.3.4	Conteúdos e Exercícios	77
5.5.3.5	Páginas e Fluxos	77
<b>5.6</b>	<b>Resumo do Capítulo</b>	<b>83</b>
<b>6</b>	<b>ANÁLISE DE RESULTADOS</b>	<b>84</b>
<b>6.1</b>	<b>Fases da Pesquisa-ação</b>	<b>84</b>
<b>6.2</b>	<b>Primeiro Ciclo - Avaliação por Especialista</b>	<b>85</b>
<b>6.3</b>	<b>Segundo Ciclo - Avaliação por Entrevistas com Público-alvo</b>	<b>85</b>
6.3.1	Roteiro de Entrevista	86
6.3.2	Resultados e Análise das Entrevistas	87
6.3.3	Melhorias aplicadas	90
<b>6.4</b>	<b>Terceiro Ciclo - Avaliação por Questionário com Público-alvo</b>	<b>93</b>
6.4.1	Estrutura do Questionário	93
6.4.2	Resultados e Análise do Questionário	94
6.4.3	Melhorias Aplicadas	104
<b>6.5</b>	<b>Quarto Ciclo - Autoavaliação</b>	<b>105</b>
6.5.1	Autoavaliação Heurística	105
6.5.1.1	HA1	105
6.5.1.2	HA2	106
6.5.1.3	HA3	106
6.5.1.4	HA4	106
6.5.1.5	HA5	107
6.5.1.6	HA6	107
6.5.1.7	HA7	107
6.5.1.8	HA8	107
6.5.1.9	HA9	108



6.5.1.10	HA10	108
6.5.1.11	HA11	108
6.5.1.12	HA12	108
6.5.1.13	HA13	109
6.5.1.14	HA14	109
6.5.1.15	HA15	109
6.5.1.16	HA16	109
6.5.1.17	HA17	109
6.5.1.18	HA18	110
6.5.1.19	HA19	110
6.5.1.20	HA20	110
6.5.1.21	HA21	110
6.5.1.22	HA22	110
6.5.1.23	HA23	111
6.5.1.24	HA24	111
6.5.1.25	HA25	111
6.5.1.26	HA26	111
6.5.1.27	HA27	111
6.5.1.28	HA28	112
6.5.1.29	Análise	112
<b>6.6</b>	<b>Resumo do Capítulo</b>	<b>112</b>
<b>7</b>	<b>CONCLUSÃO</b>	<b>114</b>
<b>7.1</b>	<b>Status Atual do Trabalho</b>	<b>114</b>
<b>7.2</b>	<b>Considerações sobre a Abordagem e o Site Educacional</b>	<b>116</b>
7.2.1	Pontos Fortes	117
7.2.2	Fragilidades	117
7.2.3	Trabalhos Futuros	117
	<b>REFERÊNCIAS</b>	<b>119</b>
	<b>APÊNDICES</b>	<b>124</b>
	<b>APÊNDICE A – PROTÓTIPO DE ALTA FIDELIDADE</b>	<b>125</b>
	<b>APÊNDICE B – CONTEÚDO E EXERCÍCIOS NO SITE EDUCACIONAL</b>	<b>131</b>
<b>B.1</b>	<b>Conteúdo - Padrões de Projeto</b>	<b>131</b>
<b>B.2</b>	<b>Exercícios - Padrões de Projeto</b>	<b>132</b>

B.3	Conteúdo - Singleton . . . . .	133
B.4	Exercícios - Singleton . . . . .	134
B.5	Conteúdo - Facade . . . . .	135
B.6	Exercícios - Facade . . . . .	137
B.7	Conteúdo - Template Method . . . . .	138
B.8	Exercícios - Template Method . . . . .	140
B.9	Conteúdo - Factory Method . . . . .	140
B.10	Exercícios - Factory Method . . . . .	143
B.11	Conteúdo - Adapter . . . . .	143
B.12	Exercícios - Adapter . . . . .	145
B.13	Exercícios - Desafio Nível 1 . . . . .	146
B.14	Exercícios - Desafio Nível 2 . . . . .	147

# 1 Introdução

Nesse capítulo, é apresentada a [Contextualização](#), permitindo compreender o cenário no qual o presente trabalho está inserido. Nesse sentido, há foco na aprendizagem de padrões de projeto usando como base uma abordagem gamificada. Adicionalmente, visando expor de forma clara essa abordagem, foi desenvolvido um site educacional. Tornou-se necessário, portanto, compreender cada um desses conceitos e como os mesmos são combinados. Na sequência, cabe considerar uma [Justificativa](#) para a realização do trabalho; a [Questão de Pesquisa](#) que orientou o trabalho, bem como os desafios de cunho prático que o acompanham, por exemplo, questões de desenvolvimento, e os principais [Objetivos](#) desse trabalho, seja geral ou específicos. Por fim, tem-se a [Organização da Monografia](#).

## 1.1 Contextualização

Os padrões de projeto são soluções genéricas para problemas encontrados com frequência em um projeto de software ([GAMMA et al., 1995](#)). Segundo o Larman ([2012](#)), a aplicação de padrões de projeto traz diversos benefícios para um projeto, como a reutilização, o baixo acoplamento e a flexibilidade.

De acordo com estudos, criar um software com essas características é muito complexo para um estudante, pois exige um alto grau de abstração, além de certa maturidade em Engenharia de Software ([SILVEIRA; SILVA, 2006](#) apud [SILVA; MIRANDA; PEREIRA, 2016](#)).

A área da educação busca explorar diferentes abordagens pedagógicas, procurando instigar estudantes de diversas formas para impactar positivamente o aprendizado ([SANTOS, 2005](#)). Segundo Santos ([2005](#)), essa necessidade faz-se presente, principalmente, em contextos em que o objeto de aprendizado é reconhecido como complexo, ou seja, de difícil compreensão. Para ampliar essas ações pedagógicas, recentemente, os elementos das mecânicas de jogos digitais têm sido utilizados em práticas pedagógicas e educacionais, caracterizando um processo conhecido como gamificação ([FIQUEIREDO; PAZ; JUNQUEIRA, 2015](#)).

Não existe uma única e sucinta definição do que é a gamificação. Entretanto, uma forma de descrevê-la é como o processo de introduzir elementos do domínio dos jogos a diferentes contextos, visando incentivar participação e comportamentos direcionados a um objetivo específico ([MORSCHHEUSER et al., 2018](#)).

O uso da gamificação no contexto educacional tem sido amplamente estudado, contribuindo positivamente no engajamento e na motivação dos estudantes, alcançando muita popularidade nos últimos anos (DETERDING et al., 2011).

Deve-se ter atenção ao introduzi-la em algum processo, pois existe a necessidade de compreender quais elementos da mecânica de jogos são adequados para determinado grupo de pessoas ou para cumprir um propósito específico. Especialmente no campo educacional, assimilar essa necessidade pode ser uma tarefa desafiadora (RUGHINIŞ, 2013), visto que a aplicação da gamificação para fins de ensino e aprendizagem raramente segue um processo estabelecido ou uma diretriz pré-definida, tampouco há roteiro, com didática plenamente conhecida e única (DICHEVA et al., 2014).

Da mesma forma que há variabilidade nas propostas educacionais, orientadas a diferentes dinâmicas de ensino-aprendizagem (SANTOS, 2005), há vastas possibilidades de se introduzir gamificação nesses cenários de uso (ALVES, 2015). Entretanto, o viés educacional demanda cuidados inerentes à área, com destaque para a necessidade de se ter didática e adequação ao público alvo, conforme amplamente acordado por Paulo Freire<sup>1</sup>.

Há estudos, com embasamentos estatísticos muito consistentes, que indicam a pertinência da gamificação na educação (ALVES, 2015). Além disso, ainda com base na autora Flora Alves (ALVES, 2015), cabe ressaltar que a gamificação, e sua natural perspectiva sociocultural, corrobora com a sua adoção em variados contextos, sendo uma abordagem de muita aceitação em diferentes públicos, em especial, as novas gerações. O uso de elementos lúdicos confere uma dinâmica associada a entretenimento, o que leva, segundo Santos et al. (2019), a um aprendizado mais prazeroso, muitas vezes personalizado e único. Tais particularidades também permitem, de acordo com Minuzi et al. (2018), melhorias contínuas no processo de ensino-aprendizagem, uma vez que o estudante constrói o seu saber de modo interativo, sendo estimulado a tornar-se sujeito ativo do processo, conforme idealizado por grandes pensadores(as), tais como Jean Piaget<sup>2</sup>, Seymour Papert<sup>3</sup>, Maria Montessori<sup>4</sup>, e Célestin Freinet<sup>5</sup>.

Diante do exposto, os autores desse trabalho optaram por contribuir nesse contexto, propondo o uso de uma abordagem, orientada à gamificação, na aprendizagem de padrões de projeto, sendo esse processo auxiliado por um *site* educacional.

<sup>1</sup> <https://iep.utm.edu/freire>. Acesso em 19 abr, 2023

<sup>2</sup> <https://www.britannica.com/biography/Jean-Piaget>. Acesso em 19 abr, 2023

<sup>3</sup> <https://www.britannica.com/biography/Seymour-Papert>. Acesso em 19 abr, 2023

<sup>4</sup> <https://www.britannica.com/biography/Maria-Montessori>. Acesso em 19 abr, 2023

<sup>5</sup> [https://perso.telecom-paristech.fr/rodriguez/resources/PEDAGO/celestin\\_freinet.pdf](https://perso.telecom-paristech.fr/rodriguez/resources/PEDAGO/celestin_freinet.pdf). Acesso em 19 abr, 2023

## 1.2 Justificativa

O processo de ensino-aprendizagem de padrões de projeto é tido como uma área de estudo problemática e com muitas opiniões controversas em relação ao conhecimento e a maturidade dos estudantes (SILVA; MIRANDA; PEREIRA, 2016). Segundo Silva et al. (2016), uma abordagem baseada em leituras e aulas palestradas mostra-se como uma estratégia ineficiente para o ensino dos mesmos. Assim, torna-se evidente a possibilidade de buscar diferentes abordagens para a aprendizagem de padrões de projeto.

Recentemente, o uso da gamificação em diferentes áreas tem crescido muito, como, por exemplo, nas áreas de saúde, *marketing*, *fitness*, entre outras (HAMARI; KOIVISTO; SARSA, 2014). Sendo assim, empresas de todos os ramos começaram a entender esse fenômeno, incorporando em seus produtos elementos característicos da gamificação, como barras de progressão, *rankings*, pontos por ação e outros, os quais estimulam o usuário a realizar a atividade proposta (DICHEVA et al., 2014).

Nesse mesmo sentido, de acordo com alguns estudos, a gamificação pode ser uma ferramenta muito poderosa para o engajamento e a motivação de estudantes (LEE; HAMMER, 2011 apud DICHEVA et al., 2014), ainda mais em assuntos complexos e que envolvem densos conhecimentos. Conforme já colocado, a gamificação pode trazer pontos muito positivos para o aprendizado, tornando-o mais gratificante e prazeroso com os elementos de jogos, fazendo com que o estudante visualize de forma mais adequada seus avanços nos estudos e no próprio entendimento do conteúdo (SANTOS; PIMENTEL; SELLA, 2019).

Dessa forma, aplicar a gamificação na aprendizagem de padrões de projeto pode gerar os mesmos bons resultados já vistos em outras pesquisas (ALVES, 2015). Entretanto, para que essa aplicação seja realizada, torna-se necessário o desenvolvimento de uma abordagem com a ponderação sobre quais elementos de gamificação são mais apropriados para serem utilizados em um contexto educacional, e como devem ser implementados e alinhados com os objetivos de aprendizado (DICHEVA et al., 2014).

Com a elaboração desse trabalho, propõe-se uma abordagem orientada à gamificação que, baseada nas dificuldades investigadas sobre o aprendizado de padrões de projeto, possibilite auxiliar na compreensão do conteúdo por parte dos estudantes, procurando motivá-los e direcioná-los durante o processo de aprendizagem.

## 1.3 Questões de Pesquisa e Desenvolvimento

Este trabalho visou a elaboração de uma abordagem de aprendizagem orientada à gamificação, sendo essa apresentada em um site educacional de padrões de projeto. Ao final do trabalho, pretende-se responder aos seguintes questionamentos de pesquisa, mitigando as inerentes preocupações associadas ao desenvolvimento da solução:

- Questão\_de\_Pesquisa\_01: É possível, conforme ocorre em outros contextos educacionais, impactar positivamente a aprendizagem de padrões de projeto por meio de uma abordagem orientada à gamificação?
- Questão\_de\_Pesquisa\_02: Quais elementos da gamificação são mais indicados para a aprendizagem de padrões de projeto e/ou conteúdos de complexidade similar?
- Questão\_de\_Desenvolvimento\_01: Como viabilizar a apresentação da abordagem de aprendizagem de padrões de projeto, orientada à gamificação, em um site educacional?

Ao considerar as questões mencionadas, percebe-se a necessidade de levantamentos teórico e tecnológico adequados visando, respectivamente, o conhecimento sobre os domínios envolvidos (i.e. educacional e jogos), e a escolha de ferramentas e demais aparatos tecnológicos para desenvolvimento do site educacional. Complementarmente, houve necessidade de especificar métodos específicos, de viés investigativo, de desenvolvimento, e de análise de resultados, no intuito de orientar etapas importantes do trabalho.

## 1.4 Objetivos

Orientando-se pelas questões de pesquisa e desenvolvimento, tem-se que o objetivo geral do trabalho é a **elaboração de uma abordagem orientada à gamificação, cujo viés é educacional, especificamente para a aprendizagem de padrões de projeto, sendo a mesma apresentada em um site educacional.**

Como público-alvo, têm-se: aprendizes com, no mínimo, idade próxima para ingresso em uma Universidade, Faculdade ou Curso Técnico, e conhecimentos de nível intermediário sobre programação, indo um pouco além do básico, e não necessitando ser avançado.

Classificar conhecimento, ou mesmo pontuar se algo é visto como conhecimento básico, intermediário ou avançado, não é uma tarefa fácil. Sendo assim, e visando conferir uma descrição mais precisa sobre o público-alvo, esse trabalho faz uso das colocações acordadas em Souza e França (2013) para esclarecer sobre níveis de conhecimento, considerando a programação como principal insumo desse conhecimento. Nesse sentido, já em uma visão adaptada para o escopo desse trabalho, considera-se:

- um nível básico de conhecimento, quando o aprendiz tem a noção de resolver problemas muito simples na programação (ex. programar, usando pseudocódigo ou em alguma linguagem, uma operação que soma dois números). Sendo assim, compreende, por exemplo, operações aritméticas e lógicas, e estruturas de controle;

- um nível básico-intermediário de conhecimento, quando o aprendiz tem a noção de trabalhar mais amplamente os conceitos de programação (ex. programar uma estrutura de dados do tipo pilha, manipulando-a via funções). Sendo assim, compreende, por exemplo, alocação de memória em variáveis e vetores, tipos de dados abstratos, funções e outros conceitos inerentes das linguagens de programação reais (não pseudocódigo). Já tomou contato, pelo menos, com o Paradigma Imperativo Estruturado;
- um nível intermediário-avançado de conhecimento, quando o aprendiz tem a noção de solucionar problemas em domínios reais (ex. programar um MVP (*Minimum Viable Product*) de um aplicativo ou uma aplicação *web* em um domínio real, orientando-se por boas práticas). Sendo assim, compreende, por exemplo, conceitos como compilação, recursão, reutilização, além de técnicas de programação e outros. Já se aventurou em outros paradigmas de programação, em especial, a Orientação a Objetos, e
- um nível avançado de conhecimento, quando o aprendiz soluciona problemas em diferentes domínios cognitivos reais, revelando habilidade e competência em desenvolvimento, e dominando uma ou mais linguagens de programação. Tem ciência da existência de vários paradigmas de programação.

### 1.4.1 Objetivos Específicos

No intuito de atingir o objetivo geral desse trabalho, e dada a abrangência do mesmo, foram estabelecidos alguns objetivos específicos, conforme segue:

- Objetivo Específico 01: Estudo dos conceitos inerentes ao escopo do trabalho, com destaque para Padrões de Projeto, Abordagens Educacionais (Ensino-Aprendizagem), Gamificação e Site Educacional;
- Objetivo Específico 02: Aquisição de conhecimentos sobre tecnologias que viabilizem a elaboração da abordagem e projeto/desenvolvimento/validação do Site Educacional;
- Objetivo Específico 03: Especificação de métodos adequados para condução do levantamento bibliográfico, do desenvolvimento e da análise de resultados;
- Objetivo Específico 04: Detalhamento da Abordagem de Aprendizagem, baseando-se no levantamento bibliográfico e nos conceitos estudados;
- Objetivo Específico 05: Projeto e implementação do Site Educacional, apresentando a abordagem definida, e

- Objetivo Específico 06: Análise dos resultados obtidos, procurando documentá-los e expondo ideias para trabalhos futuros.

## 1.5 Organização da Monografia

Além do presente capítulo, introdutório, essa monografia está organizada em mais cinco capítulos, sendo os mesmos estruturados da seguinte forma:

- **Capítulo 2 - Referencial Teórico:** descreve os conceitos que fundamentam este trabalho. O capítulo é subdividido nas seções Padrões de Projeto, Contexto Educacional, Gamificação e Site Educacional;
- **Capítulo 3 - Suporte Tecnológico:** apresenta as ferramentas e outros apoios tecnológicos utilizados ao longo da realização desse trabalho, relacionados às Ferramentas de Apoio Geral, as quais auxiliam na elaboração da monografia; na pesquisa bibliográfica; em modelagens de cunho geral, e na comunicação entre os envolvidos, além de Ferramentas de Apoio ao Site Educacional, com tecnologias apropriadas para implementação, versionamento e hospedagem do mesmo;
- **Capítulo 4 - Metodologia:** aborda o plano metodológico adotado de forma mais detalhada, incluindo Classificação da Pesquisa; método de viés específico (Levantamento Bibliográfico, Método de Desenvolvimento e Método de Análise de Resultados); além de Fluxo de Atividades e Cronogramas, tanto para o escopo da primeira etapa do TCC, quanto para o escopo da segunda etapa do TCC;
- **Capítulo 5 - Abordagem e Site Educacional:** apresenta a abordagem gamificada e o Site Educacional resultados deste trabalho, detalhando a Contextualização; conferindo uma visão sobre o processo de ideação, com um *Benchmarking*, um *Brainstorming* e Personas. Por fim, são apresentados a abordagem e o site em si, constando detalhes sobre o Plano de Gamificação, além de tópicos como Usabilidade, Design, Protótipo de Alta Fidelidade e Telas e Fluxos do Site desenvolvido;
- **Capítulo 6 - Análise de Resultados:** apresenta os ciclos de pesquisa-ação realizados, descrevendo em cada um deles a observações e possibilidades de melhorias e
- **Capítulo 7 - Conclusão:** apresenta a conclusão do trabalho, retomando as questões de pesquisa e desenvolvimento, detalhando o cumprimentos dos objetivos, e identificando melhorias futuras e propostas de trabalhos futuros.



## 2 Referencial Teórico

Nesse capítulo, é apresentado o referencial teórico, o qual expõe os conceitos e as práticas que embasaram a elaboração desse trabalho, levando em consideração a abordagem de aprendizagem orientada à gamificação, com foco em padrões de projeto.

Esse capítulo é inicialmente centrado em [Padrões de Projeto](#), abrangendo suas categorias, bem como alguns exemplos, que compreendem o contexto de um problema e a implementação da solução. Na sequência, é apresentado um breve histórico do [Contexto Educacional](#), a partir do qual são levantadas algumas ideologias e influências de pedagogos e educadores que contribuíram de forma significativa para a evolução dos métodos de ensino-aprendizagem. Adicionalmente, é feita uma breve revisão das metodologias ativas de aprendizagem. São também apresentados os conceitos de [Gamificação](#), suas características e o *framework* Octalysis. Dado que esse trabalho apresenta a abordagem orientada à gamificação em forma de um site educacional, é relevante incluir boas práticas para orientar a implementação do mesmo. Tais aspectos são abordados no tópico [Site Educacional](#). Vale destacar que por conta de não haver traduções oficiais de alguns termos, eles serão usados em inglês para se manter a ideia inicial. Por fim, tem-se o [Resumo do Capítulo](#).

### 2.1 Padrões de Projeto

Nos anos 70, Christopher Alexander desenvolveu padrões de projeto como uma resposta a problemas recorrentes enfrentados por arquitetos. Seus livros “*A Pattern Language*” e “*A Timeless Way of Building*” estabeleceram esses padrões como boas práticas amplamente reconhecidas pela comunidade arquitetônica. Alexander propôs a criação de uma base objetiva para avaliar projetos, indo além das preferências pessoais dos arquitetos, e argumentou que essa base deveria ser fundamentada no senso comum, passível de descrição e mensuração ([ALEXANDER, 1977](#)).

A análise de diferentes construções levou Alexander a identificar similaridades entre elas. Ele documentou essas similaridades, atribuiu nomes e criou um guia composto por esses padrões. Posteriormente, nos anos 90, o conceito de padrões de projeto expandiu-se para o campo de desenvolvimento de software. Kent Beck ([BECK, 1987](#)) apresentou uma abordagem de padrões de projeto voltada para software na conferência OOPSLA em 1987, seguida por numerosos artigos, revistas e livros sobre o assunto.

Padrões de projeto são soluções para problemas comuns em um projeto de software. Os padrões de projeto são conceitos adaptáveis, em que o mesmo padrão pode aparecer em um projeto de diferentes formas ([GAMMA et al., 1995](#)). Isso é importante para não

haver uma solução rígida, ou seja, uma solução inflexível a possíveis aperfeiçoamentos durante o processo de desenvolvimento. Esses aperfeiçoamentos podem ser feitos ainda garantindo a presença dos elementos essenciais de cada padrão de projeto (APPLETON, 1997).

Os padrões de projeto são soluções gabaritadas, já utilizadas por diversos desenvolvedores, tornando-os de extrema relevância. Eles são catalogados de maneira muito bem definida. Entender seus contextos ideais torna mais simples a aplicabilidade prática dos padrões.

Para se identificar um padrão de projeto, existem alguns elementos essenciais que podem ser observados (MALDONADO et al., 2002):

**Name:** todo padrão deve ter um nome significativo, que permita uma referência rápida e fácil ao padrão;

**Problem:** apresenta a descrição do problema a ser resolvido pelo padrão, abrangendo seus propósitos e intenções em relação ao contexto e às forças particulares envolvidas;

**Context:** define as condições em que o problema e sua solução geralmente ocorrem, demonstrando a aplicabilidade do padrão. Também descreve a configuração inicial do sistema antes de aplicar-se o padrão;

**Forces:** descreve as influências, restrições e interações relevantes para o problema, incluindo um cenário concreto que motiva o padrão;

**Solution:** apresenta os relacionamentos estáticos e as regras dinâmicas que levam ao resultado desejado. Inclui instruções e descrições da estrutura, participantes e comportamento dinâmico do padrão. Também pode discutir variações e implementações específicas;

**Examples:** apresenta uma ou mais aplicações do padrão, ilustrando como ele é aplicado e transforma um contexto inicial em um contexto final;

**Resulting context:** descreve o estado ou configuração do sistema após a aplicação do padrão, incluindo suas consequências, tanto positivas quanto negativas;

**Rationale:** explica as regras e passos do padrão, mostrando como e por que ele resolve as influências contrárias (forças) e está alinhado com objetivos, princípios e filosofias propostas;

**Related Patterns:** explora os relacionamentos estáticos e dinâmicos do padrão com outros padrões na mesma linguagem ou sistema. Padrões relacionados podem ser predecessores, sucessores, alternativos ou codependentes;

**Known Uses:** descreve exemplos de ocorrências conhecidas do padrão e sua aplicação em sistemas existentes, validando-o como uma solução comprovada para um problema recorrente;

Em 1994, Erich Gamma, Richard Helm, Ralph Johnson e John Vlissides, conhecidos como “Gang of Four”, publicaram o influente livro “*Design Patterns: Elements of Reusable Object-Oriented Software*” (GAMMA et al., 1995). Essa obra tornou-se uma referência fundamental no estudo de padrões de projeto, sendo também a referência principal para o embasamento do tópico de padrões de projeto desse trabalho. No livro, foram identificados 23 padrões, divididos em três categorias, como apresentado no Quadro 1.

Quadro 1 – Principais Padrões de Projeto

<b>Padrões Criacionais</b>	<b>Padrões Estruturais</b>	<b>Padrões Comportamentais</b>
<i>Abstract Factory</i>	<i>Adapter</i>	<i>Chain of Responsibility</i>
<i>Builder</i>	<i>Bridge</i>	<i>Command</i>
<i>Factory Method</i>	<i>Composite</i>	<i>Interpreter</i>
<i>Prototype</i>	<i>Decorator</i>	<i>Iterator</i>
<i>Singleton</i>	<i>Façade</i>	<i>Mediator</i>
	<i>Flyweight</i>	<i>Memento</i>
	<i>Proxy</i>	<i>Observer</i>
		<i>State</i>
		<i>Strategy</i>
		<i>Template Method</i>
		<i>Visitor</i>

Fonte: Autores, adaptado de Gamma et al. (1995)

### 2.1.1 Padrões Criacionais

Os padrões criacionais são projetados para tornar os sistemas independentes da maneira como os objetos são criados, compostos e representados. Eles são especialmente relevantes quando os sistemas dependem mais da composição de objetos do que da herança de classes. Esses padrões encapsulam o conhecimento sobre as classes concretas usadas pelo sistema, e ocultam o processo de criação e composição das instâncias dessas classes. Eles fornecem flexibilidade na criação de objetos, e permitem a configuração estática ou dinâmica do sistema. Alguns padrões de criação podem ser usados em conjunto, enquanto outros podem se complementar mutuamente em determinadas situações. Por exemplo, o padrão *Prototype* pode usar o padrão *Singleton* em sua implementação (GAMMA et al., 1995).

### 2.1.1.1 *Singleton*

Para exemplificar os padrões criacionais, será usado o padrão *Singleton*, que, dentre os 23 padrões de projeto GOF, é um dos mais conhecidos e mais utilizados. De acordo com GAMMA et al. (1995), o *Singleton* é usado para:

Garantir que uma classe tenha somente uma instância, fornecendo um ponto global de acesso para a mesma.

Sendo assim, o *Singleton* existe essencialmente para resolver esses dois problemas. Para exemplificar o primeiro caso, imagina-se que seja preciso criar objetos essencialmente parecidos. Para não criar N objetos, utiliza-se o *Singleton*, que retornará o objeto já criado. Já no segundo caso, o *Singleton* dá acesso global a uma instância, permitindo que o objeto seja acessado de qualquer lugar, sem correr o risco dessa instância ser sobrescrita (GAMMA et al., 1995).

O diagrama de classes apresentado na Figura 1 e o código apresentado na Figura 2 são relativamente simples, compostos por apenas uma classe, um construtor privado, um método para acesso público e um atributo estático da própria classe. Cabe colocar que se optou por modelar o *Singleton*, e outros padrões doravante acordados, usando como base o idioma em inglês, uma vez que os mesmos serão implementados, na sequência, usando uma linguagem de programação em inglês. Apesar de ser uma decisão controversa na área, segundo Fabio Akita 2013, referência na comunidade de software, dentre outros autores, especificar e programar em inglês colabora com o contexto globalizado vigente.

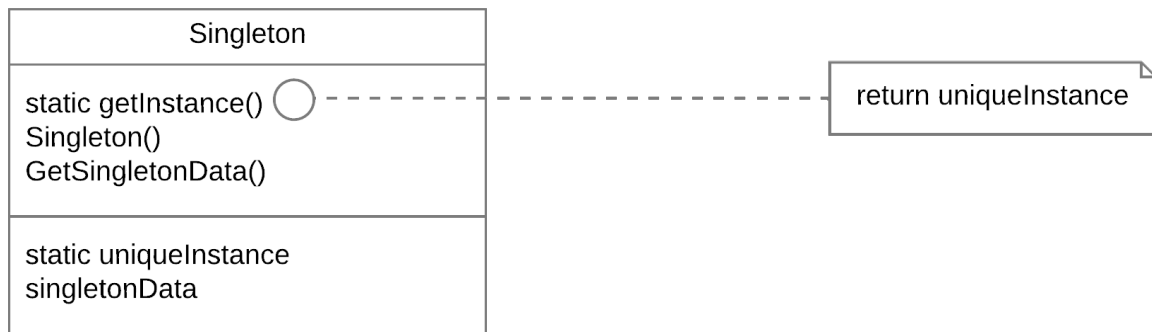
O construtor privado garante que nenhuma outra classe do sistema consiga instanciar a classe do *Singleton*, fazendo com que nenhuma outra classe crie várias instâncias dessa mesma classe.

O método de acesso público garante que exista apenas um ponto de acesso global para essa classe. É a partir desse método que a lógica do *Singleton* funciona, pois assim apenas uma instância dessa classe será retornada para o sistema.

O atributo estático garante que não seja possível criar uma classe fora da lógica do *Singleton*, fazendo com que seja preciso acessar o método estático *getInstance()*, o qual retorna esse atributo estático.

## 2.1.2 Padrões Estruturais

Os padrões estruturais abordam a composição de classes e objetos para formar estruturas maiores. Os padrões estruturais de classes utilizam a herança para combinar interfaces ou implementações, permitindo que bibliotecas de classes independentes trabalhem juntas. Por exemplo, o padrão *Adapter* cria uma interface uniforme entre duas interfaces diferentes, usando herança para adaptar uma classe existente.

Figura 1 – Diagrama de Classes *Singleton*

Fonte: Autores, adaptado de Gamma et al. (1995)

Figura 2 – Exemplo de *Singleton*

```

1 public class Singleton {
2
3     private static Singleton uniqueInstance;
4
5     private Singleton() {
6     }
7
8     public static synchronized Singleton getInstance() {
9         if (uniqueInstance == null)
10            uniqueInstance = new Singleton();
11
12        return uniqueInstance;
13    }
14 }
  
```

Fonte: Autores, adaptado de Gamma et al. (1995)

Já os padrões estruturais de objetos descrevem maneiras de compor objetos para obter novas funcionalidades, proporcionando flexibilidade ao permitir mudanças na composição em tempo de execução. O *Composite*, por exemplo, permite a criação de estruturas complexas hierárquicas, combinando objetos primitivos e compostos. O *Proxy* age como um substituto para outro objeto e pode restringir, aumentar ou alterar suas propriedades. O *Flyweight* é usado para compartilhar objetos eficientemente, enquanto o *Façade* permite que um objeto represente todo um subsistema.

Outros padrões estruturais incluem o *Bridge*, que separa a abstração da implementação, permitindo que elas variem independentemente. O *Decorator* adiciona dinamicamente responsabilidades aos objetos, permitindo uma quantidade ilimitada de responsabilidades adicionais. Esses padrões podem ser empilhados uns sobre os outros, adicionando cada vez mais funcionalidades ao objeto decorado. Os padrões estruturais estão interco-

nectados de várias maneiras, fornecendo soluções flexíveis e reutilizáveis para problemas de composição e estruturação de classes e objetos (GAMMA et al., 1995).

#### 2.1.2.1 *Façade*

Para exemplificar os padrões estruturais, será usado o padrão *Façade*, que, de acordo com GAMMA et al. (1995), compreende:

Fornecer uma interface unificada para um conjunto de interfaces em um subsistema. *Façade* define uma interface de nível mais alto que torna o subsistema mais fácil de ser usado.

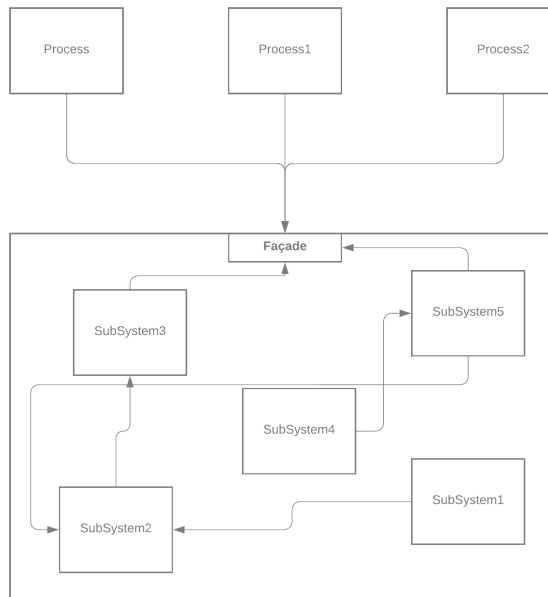
O *Façade* é utilizado para corrigir o problema de inicializar várias classes separadas. Sendo assim, cria uma única classe responsável por um aglomerado de outras classes, ajudando a reduzir a complexidade do código. Esse padrão é uma maneira de simplificar a utilização de um subsistema complexo, oferecendo uma interface simplificada para acessar as funcionalidades.

A maneira mais simples de entender sua utilização é imaginando um sistema com várias classes e métodos, em que para executar uma operação é necessário agrupar e chamar várias dessas classes e métodos, como exemplificado na Figura 3. O *Façade* nada mais é do que a interface que agrupa todas essas classes. Dessa forma, o usuário só precisaria chamar o *Façade* para realizar essa operação, tornando assim toda a utilização mais simples e diminuindo a complexidade do código, como apresentado no código da Figura 4.

### 2.1.3 Padrões Comportamentais

Os padrões comportamentais são focados em algoritmos e na atribuição de responsabilidades entre objetos, incluindo a comunicação entre eles. Esses padrões facilitam a visualização e o controle dos fluxos complexos durante a execução do sistema. Eles podem ser implementados tanto por meio de herança (padrões comportamentais de classe), quanto por composição de objetos (padrões comportamentais de objetos). Os padrões comportamentais de classe, como o *Template Method* e o *Interpreter*, distribuem o comportamento entre as classes usando herança. Por outro lado, os padrões comportamentais de objetos, como *Mediator*, *Chain of Responsibility*, *Observer*, *Strategy*, *Command*, *State*, *Visitor* e *Iterator*, utilizam a composição de objetos para definir a interação e o comportamento entre eles.

Esses padrões permitem encapsular comportamentos específicos em objetos; delegar solicitações entre objetos; definir estados de objetos, e abstrair acesso e percurso

Figura 3 – Diagrama de Classes *Façade*

Fonte: Autores, adaptado de Gamma et al. (1995)

de objetos em agregados. Eles proporcionam um acoplamento flexível entre os objetos, permitindo que a lógica seja modificada e reutilizada de maneira mais eficiente.

O padrão *Observer*, por exemplo, estabelece uma dependência entre objetos, onde as visões do modelo são notificadas sobre as mudanças de estado. O padrão *Strategy* encapsula algoritmos em objetos, facilitando a troca e a especificação do algoritmo usado por um objeto. Já o padrão *Command* encapsula solicitações em objetos, permitindo seu armazenamento e sua manipulação. O padrão *State* encapsula os diferentes estados de um objeto, alterando seu comportamento conforme o estado atual. O padrão *Iterator* abstrai acesso e percurso de objetos em um agregado.

Esses padrões comportamentais proporcionam soluções eficientes para lidar com a interação e o comportamento entre objetos, melhorando a flexibilidade, a manutenibilidade e a reutilização do código. Cada um deles aborda uma necessidade específica, mas todos compartilham o objetivo de tornar a implementação de comportamentos complexos mais gerenciável e modular (GAMMA et al., 1995).

### 2.1.3.1 *Template Method*

Para exemplificar os padrões comportamentais, será usado o padrão *Template Method*, que, de acordo com GAMMA et al. (1995), compreende:

Figura 4 – Exemplo de *Facade*

```
1 class SubSystem1 {
2     // ...
3 }
4
5 class SubSystem2 {
6     // ...
7 }
8
9 class SubSystem3 {
10    // ...
11 }
12
13 class SubSystem4 {
14    // ...
15 }
16
17 class SubSystem5 {
18    // ...
19 }
20
21 class Facade {
22     public void process() {
23         SubSystem1 subsystem1 = new SubSystem1();
24         SubSystem2 subsystem2 = new SubSystem2();
25         SubSystem3 subsystem3 = new SubSystem3();
26         SubSystem4 subsystem4 = new SubSystem4();
27         SubSystem5 subsystem5 = new SubSystem5();
28         // ...
29     }
30 }
31
32 class Application {
33     public static void main(String[] args) {
34         Facade facade = new Facade();
35         facade.process();
36     }
37 }
```

Fonte: Autores, adaptado de Gamma et al. (1995)

Definir o esqueleto de um algoritmo em uma operação, postergando alguns passos para as subclasses. *Template Method* permite que subclasses redefinem certos passos de um algoritmo sem mudar a estrutura do mesmo.

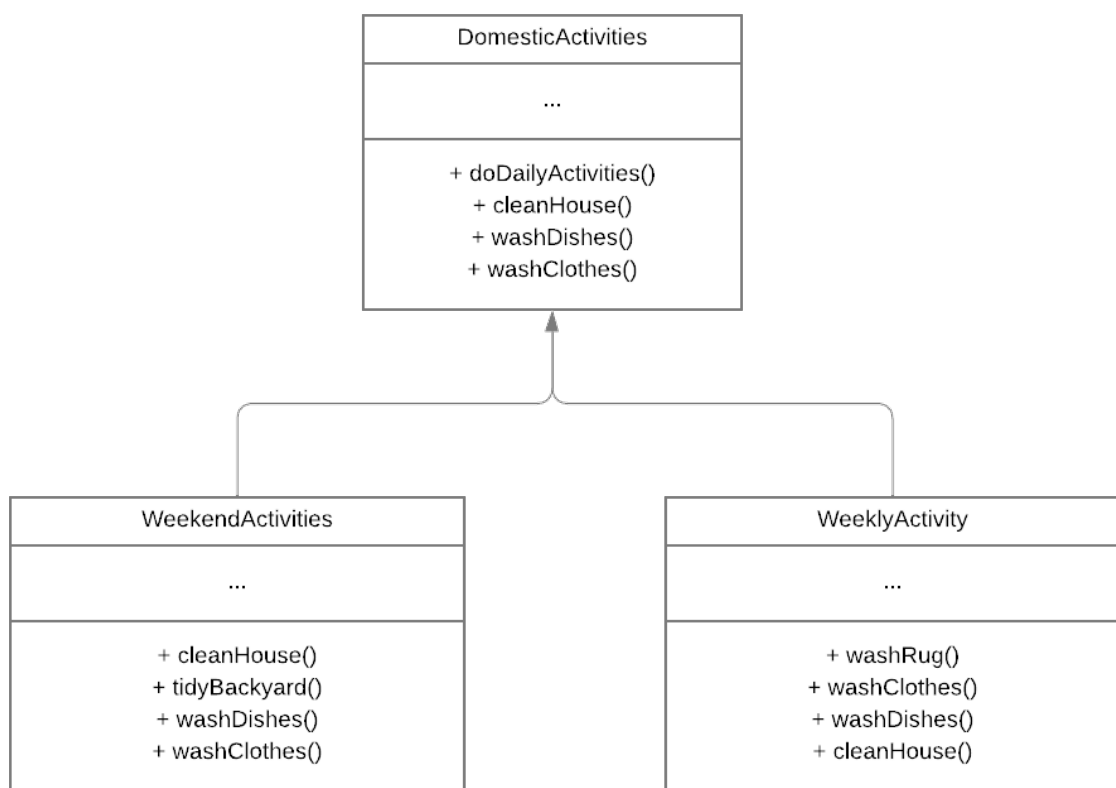
Para entender o funcionamento do *Template Method*, pode-se pensar em um exemplo de aplicação seguindo o diagrama de classes desse padrão, na Figura 5, e o código apresentado na Figura 6. As atividades diárias em uma casa precisam ser realizadas todos os dias, e todas as atividades devem seguir o que está representado no método “do-



*doDailyActivities()*” da classe “*DomesticActivities*”. As classes “*WeeklyActivity*” também implementam métodos da classe “*DomesticActivities*” e sobrescrevem o que estava na classe principal dependendo de suas necessidades. Não podem sobrescrever, entretanto, o método “*doDailyActivities()*”.

Sendo assim, toda classe pode implementar métodos novos; sobrescrever os métodos já criados na classe principal, ou simplesmente seguir o que já está definido, dependendo apenas da necessidade do usuário, o que auxilia em reutilização de código e manutenibilidade da organização do mesmo.

Figura 5 – Diagrama de Classes *Template Method*



Fonte: Autores, adaptado de Gamma et al. (1995)

Figura 6 – Exemplo de *Template Method*

```
1 public abstract class DomesticActivities {
2     final void doDailyActivities() {
3         cleanHouse();
4         washDishes();
5         washClothes();
6     }
7     void cleanHouse() {
8         System.out.println("Mop the house.");
9     }
10    void washDishes() {
11        System.out.println("Wash the dishes.");
12    }
13    abstract void washClothes();
14 }
15
16 class WeekendActivities extends DomesticActivities {
17     void cleanHouse() {
18         System.out.println("Clean the house and bathrooms.");
19     }
20     void washDishes() {
21         System.out.println("Wash the dishes and the pans");
22     }
23     void washClothes() {
24         System.out.println("Wash used towels.");
25     }
26     void tidyBackyard() {
27         System.out.println("Clean backyard and mow the lawn.");
28     }
29 }
30
31 class WeeklyActivity extends DomesticActivities {
32     void cleanHouse() {
33         System.out.println("Sweep and mop the house.");
34     }
35     void washDishes() {
36         System.out.println("Wash the dishes and tidying up the
37             kitchen");
38     }
39     void washClothes() {
40         System.out.println("Wash clothes and dirty bed linens.");
41     }
42     void washRug() {
43         System.out.println("Wash the living room rug.");
44     }
45 }
```

Fonte: Autores, adaptado de Gamma et al. (1995)

## 2.2 Contexto Educacional

Durante vários séculos, a abordagem de educação conhecida como tradicional sempre foi a adotada. Essa abordagem limitava-se basicamente à situação de um professor ensinando conteúdos para os alunos. Ademais, havia também uma hierarquia, na qual o professor era uma figura superior ao aluno. Assim, era moldado um modelo em que o estudante estava em uma sala de aula apenas para ouvir e memorizar assuntos, ou seja, não havia uma participação ativa e engajada, que permitisse a construção de um conhecimento crítico e reflexivo (LOVATO; MICHELOTTI; LORETO, 2018).

John Dewey, filósofo e pedagogo norte-americano, criticou a cultura de obediência e submissão nas escolas. Dewey defendia que o ensino deveria ocorrer pela ação (GADOTTI, 2003), e com o uso de métodos centrados em estimular o pensamento e promover a reflexão. O pedagogo trouxe para a educação a visão de busca ativa de conhecimento por parte do aluno, em que se aprende pela iniciativa de pensar e de solucionar problemas relacionados à sua própria experiência, ou seja, tratando o próprio estudante como protagonista (LOVATO; MICHELOTTI; LORETO, 2018).

Maria Montessori, médica e pedagoga italiana, teve como foco de seus estudos a aprendizagem de crianças. Montessori propunha a ideia da iniciativa individual e do direcionamento próprio, caracterizando uma “autoeducação”, em que o professor fornece os meios e a didática para o estudante, mas não atua diretamente sobre ele, apenas facilita sua autoformação. Nesse sentido, os ideais de Paulo Freire convergem com os de Montessori, pois o educador defendia que um professor deve auxiliar o processo de aprendizagem por parte do aluno, e não agir como um mestre. Freire entendia que o ato de ensinar não deve ser apenas uma forma de transferir conhecimento, mas sim possibilitar a sua construção própria, ou seja, criar uma autonomia intelectual (GADOTTI, 2003).

A área de educação e seus diferentes domínios foram amplamente estudados e influenciados por educadores, pedagogos, filósofos e psicólogos que pleiteavam mudanças nos métodos de ensino das escolas. No decorrer dos anos, essas mudanças foram incorporadas, moldando a educação como está organizada hoje. Entretanto, o contexto educacional é passível de transformações, geralmente instigadas por uma busca de abordagens mais efetivas para a aprendizagem. Na atualidade, essa busca continua, onde têm-se destacado a aplicação de metodologias ativas de aprendizagem (LOVATO; MICHELOTTI; LORETO, 2018).

Nas metodologias ativas de aprendizagem, o estudante torna-se o protagonista, enquanto o professor atua como um orientador e facilitador da aprendizagem. Essas metodologias trazem características que contribuem para que o estudante perceba-se como a origem da ação, favorecendo a motivação e a iniciativa do mesmo (BERBEL, 2011). Segundo Morán (2015), as metodologias ativas impulsionam os estudantes a reflexões mais

avanzadas, principalmente ao trazer cenários mais próximos da realidade, que instigam o estudante a pesquisar, avaliar situações, fazer escolhas e assumir riscos.

Dentre as metodologias ativas atuais, sobressaem-se algumas como *Project Based Learning* (Aprendizagem Baseada em Projetos), *Team Based Learning* (Aprendizagem Baseada em Times), *Peer Instruction* (Aprendizagem por Pares), e sala de aula invertida (MORÁN, 2015).

A Aprendizagem Baseada em Projetos é um exemplo de metodologia ativa em que as tarefas trazem aspectos mais próximos da realidade, onde a investigação de casos, a solução e a sua avaliação podem ser feitas com critérios mais sólidos e palpáveis. A aplicação dessa metodologia pode contribuir tanto para a construção do conhecimento na área de estudo, quanto para o desenvolvimento de habilidades não-técnicas, essenciais para contextos de projetos reais (SALES; SERRANO; SERRANO, 2020).

Alguns conteúdos, aqui tratando-se especificamente de Engenharia de Software, demandam maior cautela no ensino-aprendizagem. O estudo de padrões de projeto em software apresenta esse desafio, devido a aspectos como complexidade, motivação dos estudantes em experimentar soluções diferentes, e relevância do nível de conhecimento prévio do estudante (SILVA; MIRANDA; PEREIRA, 2016).

## 2.3 Gamificação

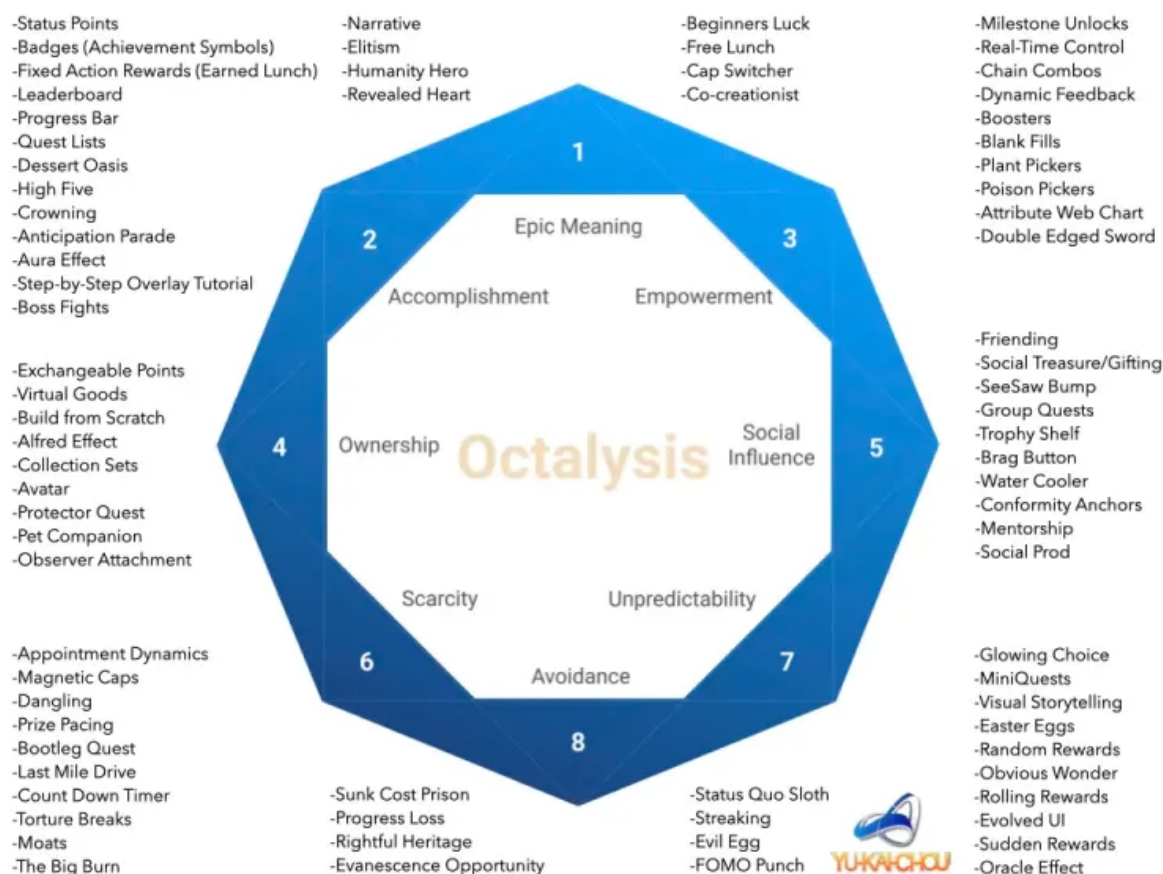
Gamificação é o uso de elementos de jogos em contextos que não são de jogos, com o objetivo de estimular engajamento e motivar os participantes de tal contexto. A gamificação pode ser utilizada em vários ambientes distintos, como educação, trabalho, tratamentos de saúde, *marketing*, dentre outros. Elementos como pontos, *rankings*, missões e recompensas são utilizados na gamificação (ZICHERMANN; CUNNINGHAM, 2011). Esses elementos são essenciais para trazer aos participantes um sentimento de dever cumprido e a vontade de sempre voltar.

De acordo com McGonigal (2011), as quatro características essenciais compartilhadas por todos os jogos são: meta, regras, *feedback* e participação voluntária.

A meta é aonde o jogador deseja chegar, é aquilo que ele vai trabalhar e se esforçar para alcançar, dando a ele o senso de propósito dentro do jogo. As regras são o que o jogador deve seguir. São impostas ao jogador, podendo ele considerá-las ao seu favor ou contra. Se boas regras são criadas, servirão também para aflorar a sua criatividade. O *feedback* utiliza alguns elementos já citados, como recompensas, *rankings* e pontuações, e serve para ajudar o jogador a visualizar o quão próximo ele está de sua meta e a continuar motivado a progredir. Além dessas características, é importante também que o jogador queira estar ali, aceitando as metas, regras e como tudo funciona (FREITAS et al., 2016).

Sendo assim, para entender as motivações básicas dos jogadores, pode-se utilizar como base o *framework Octalysis* proposto por Chou (2015), que apresenta oito *core drives*, como representado na Figura 7. Esses *core drives* apresentam técnicas de jogos relacionadas às motivações que levam os jogadores a se manterem engajados.

Figura 7 – *Octalysis* da Gamificação



Fonte: Chou (2015)

Segundo Chou (2015), as motivações são:

- 1 - Significado Épico e Chamado: quando o jogador sente que está fazendo algo para um bem maior, ou seja, está desenvolvendo uma comunidade de um jogo, ou está trabalhando em prol de um bem maior, sem ganhar algo em troca, apenas para fomentar a comunidade ou para melhorar a experiência de outras pessoas e a sua própria;
- 2 - Desenvolvimento e Realização: o sentimento de realização e sentir que está progredindo e evoluindo não existe quando não se tem desafios, que, por sua vez, são de grande importância, pois é com eles que os jogadores conseguem evoluir e aprender;
- 3 - Empoderamento da criatividade e *Feedback*: quando o jogador pode utilizar a criatividade para ultrapassar as dificuldades e receber *feedbacks* se está certo ou

errado, motiva-o a querer entender cada vez mais e melhorar o que já está sendo feito;

- 4 - Propriedade e Posse: quanto mais tempo o jogador se dedica, mais ele se sente motivado a continuar nesse processo, criando e aprendendo. Inclusive quanto mais o jogador produz, mais fácil é continuar produzindo, pois ele vê que está gerando resultados. Como, por exemplo, ao se manter no topo do *ranking* ou liderar algum tipo de grupo, entre outros;
- 5 - Influência Social e Pertencimento: ter interações sociais, comunicar-se, criar laços, habilidades, conhecer pessoas, competições são formas de motivações. Essas conexões são partes essenciais de ter pessoas com vontade de jogar, ou fazer novas ações;
- 6 - Escassez e Impaciência: ter o sentimento de algo único, que é algo somente seu ou extremamente raro. Ter um troféu por terminar um desafio, por exemplo, é um motivador para continuar buscando mais. Assim como também a plataforma forçar um usuário a ficar um tempo desconectado, para pensar nela em outras partes do dia;
- 7 - Imprevisibilidade e Curiosidade: quando algo parece não seguir o mesmo padrão, com certeza é um motivador, pois força o cérebro a criar outro pensamento sobre aquilo que já estava padronizado, fazendo com que a criatividade aflore e o jogador queira entender mais sobre o que está acontecendo, e
- 8 - Perda e Rejeição: o sentimento de perda iminente pode ser algo motivador em alguns casos, como, por exemplo, perder todo o progresso que foi construído ao longo de muito tempo; ou por causa de um erro perder algo já feito, ou até mesmo, trazendo uma oportunidade única que possivelmente não vai ocorrer novamente.

## 2.4 Site Educacional

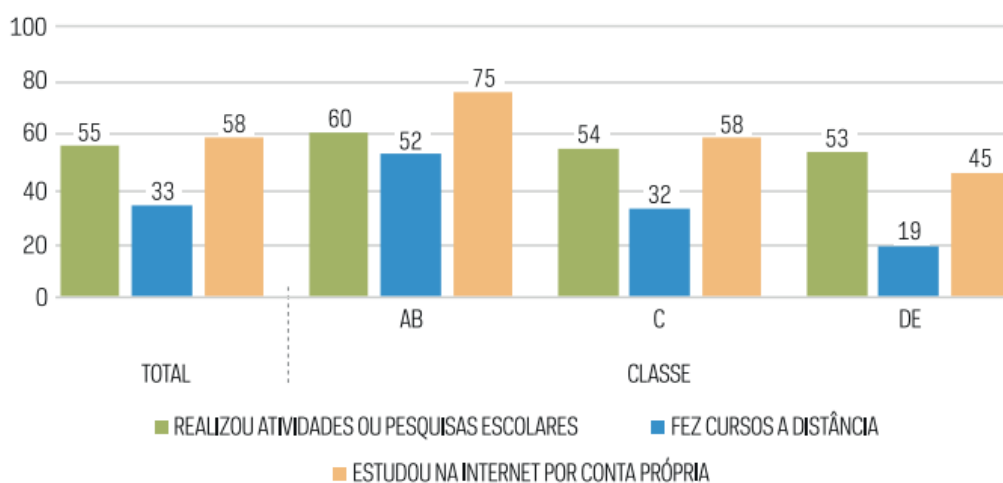
Comparando com décadas anteriores, nos dias de hoje, o acesso à internet e às Tecnologias de Informação e Comunicação (TIC) aumentou muito, e muitos usufruem diariamente de tais tecnologias para diferentes atividades do seu dia-a-dia, tais como comunicação, pesquisa de informações e serviços, educação e trabalho, comércio eletrônico e atividades culturais (ouvir música, assistir vídeos, séries e filmes, jogar jogos *online*, entre outros) ([CETIC.BR, 2022b](#)).

De acordo com pesquisa do Cetic.br<sup>1</sup>, durante a pandemia do COVID-19, houve um aumento significativo do uso da internet e das TICs, incluindo o contexto do estudo,

<sup>1</sup> Centro Regional de Estudos para o Desenvolvimento da Sociedade da Informação

tanto para pesquisas escolares, cursos a distância e estudos por conta própria (CETIC.BR, 2022a). A Figura 8 apresenta o gráfico com a porcentagem das atividades realizadas por usuários de Internet com 16 anos ou mais. As atividades que mais se destacaram foram pesquisas escolares e estudos por conta própria.

Figura 8 – Atividades Realizadas na Internet (Educação e Trabalho)



Fonte: Cetic.br (2022a)

A utilização de tecnologias em benefício da educação tem crescido muito nas últimas décadas, sendo aplicadas principalmente para transferência e aquisição de conhecimento. A tecnologia pode ser aplicada à educação com diferentes objetivos como, por exemplo, o de informar, o de motivar e estimular (GRANIĆ, 2022; ÁVILA, 2016).

Diante dessa realidade, é relevante também considerar a demanda dos cuidados inerentes à produção de diferentes materiais didáticos digitais, ambientes virtuais e ferramentas de estudo. Conforme a finalidade, o conteúdo e o público-alvo de um material, alguns aspectos a se considerar serão mais relevantes que outros para preparação e produção (SILVA, 2013; BARBA; BARCELOS, 2022; VERNEIL; BERGE, 2000).

Segundo Ávila (2016), para garantir a qualidade de um material didático digital, é necessário que as dimensões gráfica e didático-pedagógica estejam coerentes entre si (ÁVILA, 2016). Considerando isso, a seguir são apresentadas boas práticas que podem ser aplicadas à produção de um site educacional, pois incluem práticas de usabilidade, pedagógicas e até motivacionais.

Em relação à experiência de usuário, Nielsen (1994) apresenta 10 heurísticas, que abrangem características básicas de interfaces de usuário e usabilidade:

- Design simples e natural: as interfaces de usuário simplificadas ao máximo, uma vez que cada recurso adicional ou item de informação em uma tela é algo a mais para ser aprendido;

- Uso da linguagem do usuário: a terminologia nas interfaces do usuário deve ser baseada na linguagem dos usuários, e não na linguagem do sistema;
- Mínima carga na memória do usuário: o fardo de carga de memória deve ser de maior parte da máquina. Um ser humano como usuário reconhece mais facilmente algo que lhe é mostrado do que algo que precisa lembrar por sua própria memória;
- Consistência e padrões: comandos e ações iguais devem sempre resultar em ações iguais, bem como elementos iguais devem estar posicionados e organizados da mesma forma;
- *Feedback*: o sistema deve sempre estar informando o usuário de seu estado atual, apresentando *feedbacks* positivos, e não somente os negativos, como quando algum erro ocorre;
- Opções de saída bem demarcadas: o sistema deve oferecer ao usuário uma maneira fácil de sair do máximo de situações que possível, reforçando a ideia de que o usuário está no controle;
- Presença de atalhos: ao mesmo tempo que a interface deve ser simples para usuários iniciantes, deve possuir também atalhos para que usuários avançados realizem operações frequentes com maior facilidade;
- Boas mensagens de erro: mensagens de erro devem ser claras e precisas, indicando o problema encontrado ao usuário, além de ajudá-lo a resolver o problema;
- Prevenção de erros: melhor do que boas mensagens de erro, é evitar que os erros ocorram, projetando o sistema para que sejam evitadas situações comumente propensas a erros, e
- Ajuda e documentação: é preferível que o sistema sistema seja tão fácil de usar que não necessite de ajuda, porém a documentação é recomendada para complementar a interface de usuário.

Incorporando essas heurísticas juntamente com outras boas práticas apresentadas a seguir, busca-se o desenvolvimento de um site educacional de maior qualidade, tanto em interface, quanto em relação às práticas pedagógicas e à gamificação.

A seguir, são apresentadas boas práticas para o desenvolvimento de uma aplicação *web*, que complementam as heurísticas de Nielsen.

Em relação à **navegação do usuário pela interface**, o sistema deve incorporar elementos que informem ao usuário por onde ele já passou e quais atividades já realizou (VERNEIL; BERGE, 2000; BARBA; BARCELOS, 2022). É interessante também permitir



que o usuário revise atividades anteriores, ou mesmo explicações, além de deixar em evidência telas principais e mais relevantes ao usuário (BARBA; BARCELOS, 2022).

Tipos diferentes de navegação podem ser relevantes, desde que deixem explícito ao usuário em que parte do sistema ele está, e para onde pode navegar. Como dizem Verneil e Berg (2000), ferramentas pontuais auxiliam o usuário se movimentar pelo sistema; ferramentas de estrutura favorecem a visualização geral de índices, e ferramentas de histórico contribuem para que o usuário perceba por onde já passou (VERNEIL; BERGE, 2000). Essas práticas complementam as heurísticas de “mínima carga na memória do usuário” e “presença de atalhos” apresentadas por Nielsen.

O **uso de tutoriais e recursos informativos** pode ser uma ferramenta útil (BARBA; BARCELOS, 2022), mesmo que, como abordado por Nielsen (1994), seja preferível um sistema que não necessite de tais ferramentas. Ainda assim, seu uso pode ocorrer como forma de complementar a interface, ou até em situações específicas em que o usuário deseje. Os usuários têm a tendência de não ler documentações, avançando diretamente para as atividades que os fazem sentir produtivos, ou seja, começam a utilizar sistemas sem ler instruções, remetendo ao material de documentação apenas quando necessitam ou estão com problemas (NIELSEN, 1994). Isso solidifica a necessidade de um primeiro contato mais claro entre o usuário e o sistema, que justifica também uma página inicial que contenha informações relevantes ao entendimento do usuário sobre o sistema.

A página inicial deve prover visualização e sentimento de um local educacional, e deve motivar o usuário a explorar mais do sistema. Na página inicial ainda, é importante fornecer uma descrição das informações contidas na aplicação, que poderão ser exploradas por usuários registrados, caso o sistema não seja apenas para visitantes (VERNEIL; BERGE, 2000).

No que tange os **aspectos gráficos e visuais da interface**, percebe-se sobre alguns aspectos bem dependentes, principalmente, do design da interface. Elementos de tipografia, cores e ícones influenciam na usabilidade e na experiência do usuário, e devem ser selecionados levando em consideração a finalidade e o público-alvo do sistema a ser desenvolvido. É pertinente também a disposição de elementos de forma semelhante a outras soluções já existentes, buscando favorecer o reconhecimento por parte dos usuários (BARBA; BARCELOS, 2022). Quanto às cores, destaca-se a utilização de cores que permitam uma fácil leitura, bem como cores de fundo que não dificultem a visualização, nem distraiam o usuário (VERNEIL; BERGE, 2000).

No alinhamento de um texto, os blocos alinhados à esquerda e à direita oferecem indicações de alinhamento mais fortes do que os blocos de texto centralizados. Elementos posicionados mais próximos uns aos outros são percebidos como mais relacionados, assim como os elementos que são mais semelhantes entre si. As cores podem também agir como codificação para facilitar transmissão de mensagens, além de que as cores possuem signifi-

cados. O contraste entre fundos de tela e elementos são importantes. Para fundos de cores claras, utilizam-se textos de cor escura e vice-versa (LIDWELL; HOLDEN; BUTLER, 2010 apud ÁVILA, 2016). Além de todas essas características, é necessário também, em caso de utilização de imagens ou animações, que seja evitada a demora no carregamento desses elementos gráficos, para não impactar tanto a visualização, quanto possíveis ações do usuário (VERNEIL; BERGE, 2000).

Considerando o viés educacional de um sistema, é comum o uso de níveis de dificuldade, em que é preciso atentar-se à adequação de cada nível para o usuário, bem como garantir que as atividades contidas em cada nível estejam de acordo com a dificuldade do nível, e que progridam durante a evolução do usuário no aprendizado. Além disso, é interessante explicitar conquistas alcançadas e até fornecer recompensas (BARBA; BARCELOS, 2022).

## 2.5 Resumo do Capítulo

Este capítulo apresentou o referencial teórico do trabalho, que compreende quatro tópicos: padrões de projeto, contexto educacional, gamificação e site educacional. Cada um destes tópicos abordou conceitos, exemplos, dados e ideias relacionados a ele, embasados teoricamente em materiais já elaborados e publicados, principalmente livros e artigos científicos. Todas essas informações foram agregadas nesse capítulo para que o mesmo seja uma base de conhecimento que sirva de fundamento para a produção deste trabalho.

Na seção de Padrões de Projeto, foram apresentadas as categorias e os conceitos dos padrões, juntamente a exemplos para contemplar de forma mais adequada os contextos de aplicação. No Contexto Educacional, tratou-se principalmente de um breve histórico da evolução dos métodos de ensino-aprendizagem, ressaltando a discordância das ideias de educadores e pedagogos com os métodos tradicionais, e expondo também sobre metodologias ativas de aprendizagem. Para a Gamificação, o foco são seus conceitos e características, além da explicação do *framework Octalysis*. Por fim, na seção Site Educacional, é ressaltado o cuidado inerente à produção de diferentes materiais didáticos digitais, com foco em sites, abrangendo boas práticas pedagógicas, de usabilidade e de design de interface.

## 3 Suporte Tecnológico

Diferentes ferramentas foram utilizadas para apoiar a pesquisa bibliográfica (ex. Zotero); a produção da monografia (ex. Overleaf); a elaboração de insumos de cunho geral (ex. modelagem de processos na Notação BPMN usando Miro), e a comunicação entre os envolvidos (ex. Telegram, Slack e Teams). Outras ferramentas foram utilizadas para o projeto (ex. Figma para prototipação de telas e LucidChart para modelagem dos Padrões de Projeto na Notação UML); o desenvolvimento (ex. React e Node), e a análise dos resultados do site educacional (ex. Google Forms). Esse conjunto de ferramentas de apoio são apresentados nesse capítulo, sendo organizado em: [Ferramentas de Apoio Geral](#) e [Ferramentas de Apoio ao Site Educacional](#).

### 3.1 Ferramentas de Apoio Geral

Seguem as principais ferramentas utilizadas ao longo da elaboração do trabalho, e que não possuem correlação direta com o Site Educacional.

#### 3.1.1 Overleaf

Overleaf ([OVERLEAF, 2023](#)) é um editor de LaTeX *online*, que possibilita a colaboração em tempo real, com controle de versão. É frequentemente utilizado para produção de documentos técnicos e científicos. O Overleaf foi utilizado para a escrita colaborativa dessa monografia.

#### 3.1.2 Zotero

Zotero ([ZOTERO, 2023](#)) é uma ferramenta grátis, de código aberto, para armazenar, organizar e citar referências bibliográficas. O Zotero armazena diversas informações dos artigos e textos salvos, tais como: autores, títulos, data de publicação e outros. Isso permite a exportação desses campos como uma referência já formatada para ser utilizada na produção. A ferramenta possui um aplicativo *desktop* e uma versão *web*, sendo essa última a escolhida pelos autores para armazenar artigos e livros, bem como gerar e formatar citações e referências bibliográficas para essa monografia. Portanto, o Zotero foi um aliado relevante para a condução da pesquisa bibliográfica.

#### 3.1.3 Miro

Miro ([MIRO, 2023](#)) é uma plataforma que serve como lousa digital, podendo criar diagramas, *brainstorming*, estratégias, planejamentos, mapas mentais e muito mais. O

Miro tem um plano gratuito e pode ser usado por várias pessoas ao mesmo tempo. Ele possui modelos pré-prontos ou um quadro totalmente branco para criar o que quiser. A escolha dessa plataforma dá-se, principalmente, pela familiaridade dos autores com a mesma, além de sua popularidade na área.

### 3.1.4 Ferramentas de Comunicação

No intuito de facilitar a comunicação entre os autores, e entre os autores e os orientadores, foram utilizados, principalmente:

- Telegram<sup>1</sup>: para mensagens rápidas;
- Slack<sup>2</sup>: para manter os rastros das orientações conferidas durante a elaboração do trabalho, e
- Teams<sup>3</sup>: para reuniões de acompanhamento periódicos, semanais, entre autores e orientadores.

## 3.2 Ferramentas de Apoio ao Site Educacional

Seguem as principais ferramentas utilizadas para projeto, desenvolvimento, versionamento e análise do Site Educacional.

### 3.2.1 Figma

Figma ([FIGMA, 2023](#)) é uma plataforma de edição colaborativa para a construção de design de interfaces e protótipos. No desenvolvimento de um protótipo, o Figma possibilita a produção de fluxos navegáveis, com telas e elementos interativos, proporcionando uma validação mais ponderada por parte dos interessados, visto que essa navegação é apresentada de uma forma mais realista e próxima ao que será o software desenvolvido.

Nesse trabalho, o Figma foi utilizado para prototipação, colaborativamente, do Site Educacional. O detalhamento desse protótipo consta no [Capítulo 5 - Proposta](#).

### 3.2.2 React 18.2

O React ([REACT, 2023](#)) é um *framework* JavaScript para criação de interfaces de usuário. Foi criado pelo Facebook em 2011, e em 2013 teve seu código aberto pela empresa, recebendo milhares de usuários, sendo uma comunidade muito grande e ativa nas discussões e melhorias do *framework*. Atualmente, é utilizado por muitas empresas de

<sup>1</sup> <https://telegram.org>. Acesso em: Maio de 2023.

<sup>2</sup> <https://slack.com>. Acesso em: Maio de 2023.

<sup>3</sup> <https://www.microsoft.com/en-us/microsoft-teams/group-chat-software>. Acesso em: Maio de 2023.

tecnologia em suas aplicações *web* e móveis. Esse *framework* foi escolhido para realizar o desenvolvimento *frontend* do Site Educacional, devido à pertinência do mesmo ao escopo do projeto, além de ser de código aberto, e familiar aos autores.

### 3.2.3 Node 18.16

Node.js (NODE, 2023) é uma tecnologia para executar JavaScript sem depender do navegador. Com ele, é possível criar aplicações *web*, APIs, microsserviços, entre outros. O Node é *open source* e hoje é mantido pela OpenJS Foundation, cujo objetivo é crescer e promover o uso do JavaScript e de suas tecnologias relacionadas. O Node tem uma alta capacidade de escala, sendo muito flexível para diferentes usos e arquiteturas, com suporte para os maiores fornecedores de serviços *Cloud*. Ele foi escolhido para realizar o desenvolvimento *backend* do Site Educacional, novamente considerando pertinência ao escopo do trabalho, característica de código aberto, familiaridade dos autores, além de popularidade na comunidade de software.

### 3.2.4 Visual Studio Code 1.74.3

Também conhecido popularmente como VS Code, é um editor de código-fonte disponível para Windows, Linux e macOS. Trata-se de uma ferramenta de código aberto, desenvolvida e mantida pela Microsoft, em Electron, que permite a criação de produtos de software para *desktop* com HTML, CSS e JavaScript. Possui ainda diversas extensões disponíveis para, por exemplo, formatação de código, testes, linguagens, visualizações e muitas outras (Visual Studio Code, 2023).

O VS Code é uma ferramenta totalmente customizável, que auxilia muito a produtividade do usuário por ter a liberdade de utilizar as extensões, cores, atalhos da maneira que desejar. Trata-se de uma das maiores lojas de extensões entre editores de código-fonte, e qualquer um cria a sua própria extensão.

Além da pertinência ao escopo do trabalho, da familiaridade dos autores com esse ambiente de programação, evitando uma curva de aprendizado grande, esse editor foi escolhido por sua popularidade, conforme pode ser observado em uma análise comparativa com *Java Development Kit*, outra relevante plataforma, via Google Trend <sup>4</sup>.

### 3.2.5 Git 2.39.1

O Git (GIT, 2023) é uma ferramenta de controle de versão, gratuita e de código aberto. É amplamente adotada para o desenvolvimento de software, e frequentemente utilizada pelos autores para hospedagem, versionamento e compartilhamento de seus repositórios.

<sup>4</sup> <https://trends.google.com/trends/explore?date=now%201-d&geo=BR&q=Visual%20Studio%20Code,Java%20Development%20Kit&hl=en>. Acesso em maio de 2023.

tórios de projeto. Portanto, foi selecionada para uso no trabalho visando o armazenamento e versionamento do código-fonte do Site Educacional.

### 3.2.6 GitHub

O GitHub ([GITHUB, 2023](#)) é uma plataforma de hospedagem de código-fonte, que facilita o controle e o desenvolvimento colaborativo de um projeto. Por ser uma plataforma amplamente utilizada, apropriada para colaboração, e já conhecida também pelos autores, foi escolhida para a hospedagem do código-fonte do Site Educacional desenvolvido.

### 3.2.7 Trello

Trello ([TRELLO, 2023](#)) é uma ferramenta para organização de tarefas e demandas. O Trello utiliza quadros e cartões com itens e tarefas para ajudar na organização. O Trello será utilizado na estrutura do *Backlog*, e para acompanhamento das tarefas relacionadas ao desenvolvimento do Site Educacional. A escolha do Trello ocorreu, principalmente, pelo conhecimento dos autores sobre a plataforma, além de ser um recurso que permite uso gratuito e compartilhado.

### 3.2.8 Netlify

O Netlify ([NETLIFY, 2023](#)) é uma plataforma abrangente que engloba hospedagem, configuração, testagem e publicação de projetos na nuvem. Equipado com ferramentas automáticas de *deploy* e correção de problemas de implantação, o Netlify é uma escolha prática para projetos de qualquer escala. Os autores optaram por utilizar o Netlify para hospedar o *frontend* do projeto devido à familiaridade com a aplicação e à conveniência oferecida pelo *deploy* automático. A plataforma atualmente é gratuita até 100GB de upload.

### 3.2.9 Heroku

O Heroku ([HEROKU, 2023](#)) é uma plataforma de hospedagem, configuração, testagem e publicação de projetos na nuvem. Ele possui ferramentas de *deploy* automático; correção de problemas de implantação, e pode ser utilizado em projetos de qualquer escala, além de ser muito prático e facilmente automatizado. O Heroku foi escolhido como plataforma de hospedagem do *backend*, pois existe a familiaridade dos autores com a aplicação, além das facilidades do recurso em prover *deploy* automático. A plataforma tornou-se paga recentemente, mas oferece créditos de incentivo para estudantes.

### 3.2.10 LucidChart

O LucidChart ([LUCIDCHART, 2023](#)) é uma ferramenta *online* de diagramação e fluxogramas, com uma plataforma que possibilita vários usuários trabalharem simultaneamente em um projeto. A ferramenta oferece pré-construídos e uma interface intuitiva. O LucidChart é baseado na nuvem, garantindo acesso e flexibilidade, enquanto os recursos de segurança e a exportação de dados fornecem proteção e portabilidade para os diagramas criados.

O LucidChart foi escolhido para criação dos diagramas e fluxogramas utilizados nesse trabalho pela familiaridade dos autores com a plataforma e por ser simples e flexível para se utilizar e exportar, além de ser possível que ambos os autores trabalhem simultaneamente nos projetos.

### 3.2.11 Google Forms

O Google Forms ([GOOGLE, 2023](#)) é uma ferramenta de criação de formulários *online* que permite a coleta de dados de forma simples e eficiente. Com uma interface intuitiva, é possível criar questionários personalizados, enquetes e formulários de inscrição de maneira rápida. Além disso, o Google Forms oferece uma variedade de opções de resposta, como múltipla escolha, caixas de seleção e campos de texto, permitindo a adaptação dos formulários às necessidades específicas de coleta de dados.

Uma de suas maiores vantagens é a facilidade do compartilhamento dos formulários e da coleta de respostas, feito através de *links* que redirecionam aos formulários para que os participantes possam preenchê-los de forma rápida. As respostas são automaticamente computadas e organizadas em uma planilha utilizando outra plataforma do Google (Google Sheets), facilitando a análise e a visualização dos dados coletados.

O Google Forms é uma plataforma com a qual os autores já possuem familiaridade. Portanto, foi escolhida para auxiliar na coleta de dados desse trabalho.

## 3.3 Resumo do Capítulo

Esse capítulo apresentou uma breve descrição, apontando, principalmente: finalidade e motivo de escolha das ferramentas de apoio geral e ferramentas de apoio ao Site Educacional. A Tabela 1 contém um resumo das ferramentas e suas finalidades e versões.

Tabela 1 – Ferramentas de Apoio ao Trabalho

<b>Ferramenta</b>	<b>Versão</b>	<b>Finalidade</b>
Overleaf	-	Escrita colaborativa da monografia
Zotero	-	Armazenamento e organização de referências bibliográficas
Miro	-	Diagramas
Telegram, Slack, Teams	-	Comunicação entre autores e orientadores
Figma	-	Prototipação
React	18.2.0	Desenvolvimento <i>frontend</i> do site educacional
Node	18.16	Desenvolvimento <i>backend</i> do site educacional
VS Code	1.74.3	Escrita de código-fonte
Git	2.39.1	Versionamento de código
GitHub	-	Armazenamento e versionamento de código
Trello	-	Acompanhamento de tarefas e <i>backlog</i>
Netlify	-	<i>Deploy</i> do <i>frontend</i> do site educacional
Heroku	-	<i>Deploy</i> do <i>backend</i> do site educacional
LucidChart	-	Diagramação e fluxogramas do site educacional
Google Forms	-	Criação de formulários

Fonte: Autores



## 4 Metodologia

Este capítulo apresenta a metodologia aplicada para alcançar o objetivo deste trabalho. Inicialmente, é descrita a [Classificação da Pesquisa](#), compreendendo abordagem, natureza, objetivos e procedimentos. Na sequência, é explicitado o método específico de [Levantamento Bibliográfico](#), bem como o [Método de Desenvolvimento](#) e o [Método de Análise de Resultados](#). Adicionalmente, são apresentados o [Fluxo de Atividades](#) e o [Cronograma](#) da produção deste trabalho. Ao final, tem-se o [Resumo do Capítulo](#).

### 4.1 Classificação de Pesquisa

Gerhardt e Silveira (2009) definem a pesquisa científica como “o resultado de um inquérito ou exame minucioso, realizado com o objetivo de resolver um problema, recorrendo a procedimentos científicos”. Além disso, classificam os tipos de pesquisa quanto à abordagem, à natureza, aos objetivos e aos procedimentos. Essa definição foi utilizada para realizar a classificação da pesquisa deste trabalho, representada em síntese no [Quadro 2](#).

Quadro 2 – Classificação da Pesquisa

<b>Abordagem</b>	qualitativa
<b>Natureza</b>	aplicada
<b>Objetivos</b>	exploratória
<b>Procedimentos</b>	pesquisa bibliográfica e pesquisa-ação

Fonte: Autores

#### 4.1.1 Abordagem da Pesquisa

A abordagem da pesquisa deste trabalho é classificada como **qualitativa** de acordo com os conceitos tratados por Gerhardt e Silveira (2009), que afirmam que uma pesquisa qualitativa faz uso de uma metodologia própria, e que preocupa-se em explicar aspectos da realidade que não podem ser quantificados. Sendo assim, confirma-se com o fato de que os resultados da presente pesquisa não têm foco na representatividade numérica, nem nos dados quantitativos.

### 4.1.2 Natureza da Pesquisa

A natureza dessa pesquisa é classificada como **aplicada**, a qual é caracterizada, segundo Gerhardt e Silveira (2009), pela busca de gerar conhecimento para uma aplicação prática, dirigido à solução de problemas específicos.

### 4.1.3 Objetivos da Pesquisa

Gil (2007) identifica três classificações para a pesquisa quanto aos objetivos: exploratória, descritiva e explicativa. A pesquisa deste trabalho pode ser identificada como **exploratória**, visto que atende às principais definições dessa categoria propostas por Gil (2007): possuir como objetivo proporcionar uma maior familiaridade com o problema para torná-lo mais explícito ou construir hipóteses em cima dele; realizar-se pelo levantamento bibliográfico e documental, além de normalmente não envolver técnicas quantitativas para a coleta de dados (GIL, 2007).

### 4.1.4 Procedimentos da Pesquisa

Para este trabalho, enquadram-se tanto a **pesquisa bibliográfica** quanto a **pesquisa-ação**. No primeiro momento, utilizou-se a pesquisa bibliográfica, em que foi feito levantamento e estudo de materiais já elaborados e publicados, majoritariamente livros e artigos científicos (GIL, 2007), conforme consta na seção [Levantamento Bibliográfico](#). Já a pesquisa-ação foi utilizada para coleta e análise de dados, no formato descrito na seção de [Método de Análise de Resultados](#).

## 4.2 Levantamento Bibliográfico

Para construir uma base de conhecimento e garantir embasamento ao trabalho, foi realizado um levantamento de diferentes materiais, entre artigos e livros, que tratem sobre o tema de pesquisa.

As principais plataformas utilizadas para a busca de artigos e livros foram Scopus e Scielo. A partir da leitura de alguns artigos encontrados inicialmente, foram definidas **strings** de busca: “*gamification AND education*”, “*design patterns*” e “*gamification AND design patterns*”. Adicionalmente, a análise da bibliografia de artigos já selecionados, bem como a busca por textos e livros de autores conceituados na área, contribuíram para o enriquecimento da base bibliográfica.

Os critérios de seleção dos artigos foram:

- Possuir um resumo que compreenda o assunto em estudo;
- Estar redigido em língua portuguesa ou inglesa, e

- Possuir tema compatível com o assunto em estudo, seja gamificação, educação, padrões de projeto ou práticas para desenvolvimento de site educacional.

Todos os artigos e livros selecionados foram reunidos no [Zotero](#) para uma melhor organização e rastreabilidade das referências que embasam esse trabalho. A seguir são listados os principais materiais que orientam o trabalho, sendo relevantes para a escrita, principalmente, do Capítulo - [Referencial Teórico](#).

- *Design patterns: elements of reusable object-oriented software*, de Gamma et al (1995);
- *A course gamification platform supporting student motivation and engagement*, de Dicheva et al (2014);
- *Actionable Gamification - Beyond Points, Badges, and Leaderboards*, de Chou (2015), e
- *Usability Engineering*, de Nielsen (1994).

## 4.3 Método de Desenvolvimento

Para o desenvolvimento, foi adotado um método híbrido, ou seja, composto pela combinação de outros métodos existentes, cujos princípios e características são apresentados a seguir.

### 4.3.1 Scrum

Como definido por Schwaber e Sutherland (2020) no Guia do Scrum 2020, o Scrum é uma metodologia simples, que busca auxiliar pessoas, times e organizações a gerar valor através de soluções adaptativas para problemas complexos. É uma metodologia iterativa e incremental, o que reflete na busca por aumentar a previsibilidade de situações e aspectos. Caracterizando o Scrum, têm-se o time, os eventos e os artefatos.

O time Scrum é constituído por um *Scrum Master*, um *Product Owner* e pelos desenvolvedores. Esse time caracteriza-se por ser pequeno, auto gerenciável e multifuncional, além de não possuir hierarquias internas (SCHWABER; SUTHERLAND, 2020).

O time Scrum participa de eventos, sendo o principal deles a *sprint*, na qual todos os outros eventos estão inseridos. No início de uma *sprint*, ocorre o evento *sprint planning*, em que é gerado o artefato *sprint backlog*, que representa o trabalho que será realizado durante essa *sprint*. Esse planejamento baseia-se no artefato *product backlog*, que representa o trabalho planejado para o desenvolvimento do produto como um todo. Durante a *sprint*, são feitas reuniões diárias de 15 minutos, a *daily*, em que se concentra no plano de

ação para o próximo dia de trabalho, além de identificar possíveis impedimentos e promover uma rápida tomada de decisões. Ao final da *sprint*, o trabalho realizado durante a mesma é revisado na reunião de *sprint review*. Para encerrar a *sprint*, ocorre a *sprint retrospective*, em que o time verifica como foi a *sprint* quanto a indivíduos, interações e processos, e identifica possíveis mudanças para melhorar a eficácia do trabalho do time (SCHWABER; SUTHERLAND, 2020).

### 4.3.2 XP

*Extreme Programming* (XP) é uma metodologia ágil de desenvolvimento de software. Os princípios e valores do XP incluem tópicos como comunicação, simplicidade, qualidade do trabalho, mudanças incrementais, dando ênfase à aplicação de práticas específicas para o desenvolvimento do software. Práticas como a produção de histórias de usuário (*user stories*), testes de aceitação, programação em pares, contínuas entregas e outros são amplamente aplicadas em projetos de software (BECK, 2000; WELLS, 2013).

### 4.3.3 Kanban

O método Kanban aplicado ao desenvolvimento de software possibilita uma visibilidade do processo de software, pois, com o quadro Kanban, mostra o trabalho atribuído a cada desenvolvedor e evidência as prioridades. A adoção do Kanban juntamente a metodologias ágeis pode impulsionar seus benefícios, como, por exemplo: protegendo a equipe de tarefas externas; criando uma cultura de colaboração na resolução de tarefas e problemas; fornecendo uma visão mais transparente para toda a equipe, dentre outros benefícios (AHMAD; MARKKULA; OIVO, 2013).

### 4.3.4 Método Híbrido

No método híbrido adotado para o desenvolvimento deste trabalho, foram utilizados conceitos e práticas de diferentes métodos. Provenientes do Scrum, foram utilizados os artefatos *product backlog* e *sprint backlog*, bem como o conceito de *sprints* e eventos de *sprint planning* e *sprint review*, com algumas adaptações. Em relação ao XP, foram aplicadas as práticas de programação em pares, padronização e refatoração. Também foi utilizado o quadro Kanban, para organização e visualização das tarefas.

Após a consolidação dos requisitos, foi definido o *product backlog*, que compreende todas as tarefas de desenvolvimento do projeto. Com esse artefato pronto, iniciaram-se os ciclos de *sprint*, com duração de duas semanas. No início de cada *sprint*, foi feita a *sprint planning* para definir o *sprint backlog*. Ao final dela, era realizada a *sprint review*. Os *backlogs* foram visualizados e gerenciados no quadro Kanban, dividido nas colunas *Product backlog*, *Sprint backlog*, Em andamento, Em revisão e Concluído. Para montar e

gerenciar o quadro, foi utilizada a ferramenta [Trello](#). Adicionalmente, no desenvolvimento das tarefas durante a *sprint*, foi ocasionalmente utilizada a programação em pares.

## 4.4 Método de Análise de Resultados

Para análise dos resultados obtidos, foi utilizada a metodologia de pesquisa-ação. Segundo Gil (2007), para a coleta de dados na pesquisa-ação, pode-se utilizar diferentes técnicas, como entrevistas coletivas ou individuais, questionários, análise de conteúdo, entre outras. Com base nas fases descritas por Engel (2000), a seguir, são apresentadas as fases adaptadas, para utilização neste trabalho, que abrangem a coleta de dados e a análise dos mesmos:

- Definir o problema: compreende a identificação de um problema, e a sua definição, avaliando também o grau de relevância prática ou mesmo a viabilidade. O termo “problema”, no contexto desse trabalho, pode comportar vários níveis de granularidade e escopo. Sendo assim, pode-se pensar que o problema é saber se um dado padrão de projeto está bem coberto no Site Educacional (problema de menor nível de granularidade); ou ainda se o Site Educacional é ou não informativo (problema de nível intermediário de granularidade); ou até mesmo se o Site Educacional atende às expectativas gerais de seu público-alvo (problema de maior nível de granularidade). O importante é que, seja qual for a necessidade de análise, essa começa por “Definir o Problema”;
- Pesquisar: inclui o levantamento bibliográfico, que além de verificar pesquisas semelhantes, cria uma base de conhecimento para resolução do problema identificado na fase anterior. Pretende-se buscar embasamento na literatura, sempre que necessário, para que a resolução do problema seja orientada pelo que tem de mais adequado na visão da área;
- Desenvolver o plano de ação: para reverter a situação problemática definida, são consideradas alternativas e opções, introduzidas no plano de ação. O plano de ação, portanto, é estabelecido com base nas orientações oriundas da fase anterior, que podem conferir alternativas, sendo escolhida a alternativa que mais agregar valor no dado momento da elaboração do plano;
- Implementar o plano de ação: é a fase em que o plano de ação esboçado no item anterior é posto em prática. Aqui, dependendo do problema e sua granularidade, pode ser uma fase que demanda ação pontual, executada por único membro, até mesmo um conjunto de ações, envolvendo múltiplos membros; pode ainda ser algo que demande esforço interdisciplinar ou multidisciplinar; por fim, pode ser algo que implique um nível de abstração alto (bem conceitual), até um nível de abstração

mais baixo (implementação em código e configuração de ferramentas e ambiente de trabalho). Seja qual for o plano implantado, há, necessariamente, a atuação de ambos os autores desse trabalho;

- Coletar dados para avaliação: a fim de ter subsídios para realização da análise e da avaliação do plano, dados são coletados a partir de diferentes opções de métodos (ex. questionários que permitem coletar dados junto ao público-alvo), e
- Avaliar plano: de posse dos dados levantados na fase anterior, resta analisá-los e interpretá-los, para se tirar as conclusões cabíveis. De acordo com a classificação dessa pesquisa, essa análise é feita via uma abordagem qualitativa, ou seja, baseada em aspectos não quantificáveis. Para tanto, pretende-se apoiar em escalas como Likert (ALBAUM, 1997). Trata-se de uma escala de qualificação que pergunta ao envolvido sobre seu grau de satisfação ou concordância em relação a um dado tópico em análise. Permite compreender sobre reações, atitudes e comportamentos desse envolvido. Como exemplo, se há uso da escala Likert com valores de 1 a 3, tem-se que: 1 pode ser “desaprovo totalmente”; 2 pode ser “sou indiferente”, e 3 pode ser “aprovo totalmente”. Se atribuídos mais valores, torna-se possível compreender melhor sobre as percepções dos envolvidos.

## 4.5 Fluxo de Atividades

O fluxo de atividades foi elaborado para consolidar visualmente as etapas para a produção do trabalho. A Figura 9 e a Figura 10 apresentam, respectivamente, o fluxograma de atividades do TCC 1 e TCC 2, também explicitadas nos subtópicos [Atividades TCC 1](#) e [Atividades TCC 2](#).

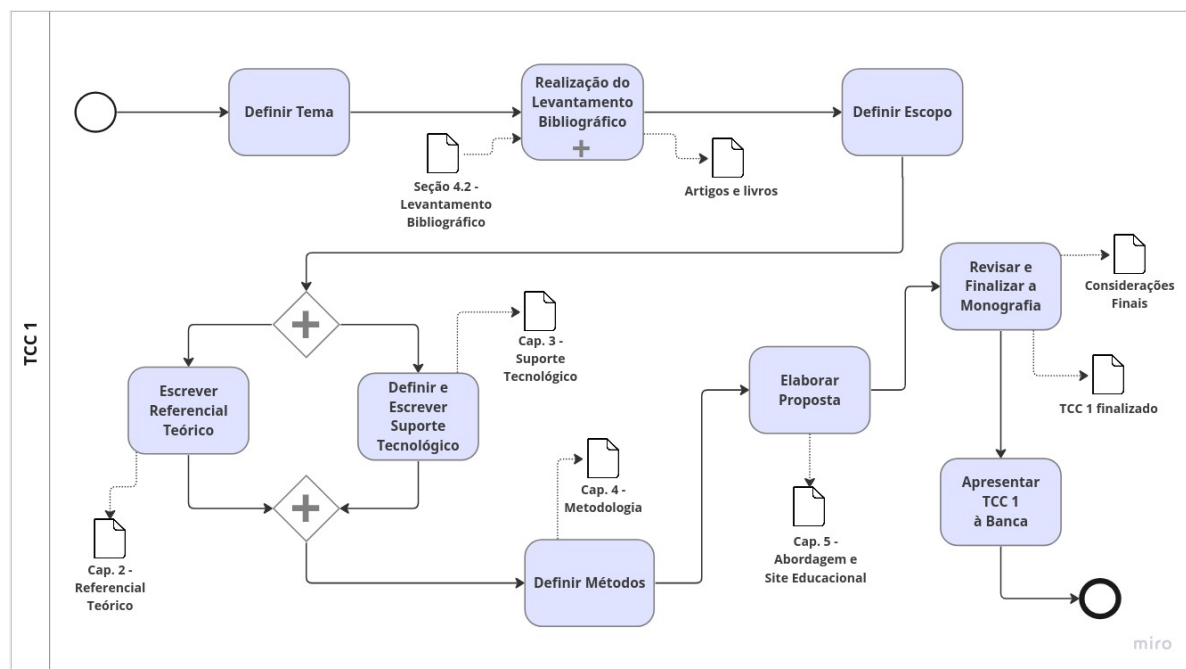
### 4.5.1 Atividades e Subprocessos - Primeira Etapa do TCC

- **Definir Tema:** realizada inicialmente para escolher o tema a ser abordado no trabalho. A escolha foi feita para propor uma abordagem que contribuísse para a aprendizagem de padrões de projeto, sendo esse um conteúdo complexo, que demanda recursos para maior engajamento e interesse por parte dos aprendizes;
- **Realização do Levantamento Bibliográfico:** foi realizado para coletar material e formar uma base de conhecimento necessário sobre o tema, visando maior familiaridade com os tópicos de interesse, e facilidade na escrita de tais aspectos nessa monografia. Nesse subprocesso, orientou-se pelo [Levantamento Bibliográfico](#);
- **Definir Escopo:** com base na definição do tema e no levantamento bibliográfico, foi estabelecido um limite em termos de escopo; bem como acordou-se sobre os objetivos, dentre outros focos do trabalho, resultando no Capítulo 1 - [Introdução](#);

- **Definir e Escrever Suporte Tecnológico:** considerando o escopo do trabalho e seus objetivos, definiu-se um suporte tecnológico inicial, com as ferramentas de apoio à pesquisa, ao desenvolvimento e à comunicação, descritas no Capítulo 3 - [Suporte Tecnológico](#). Esse capítulo foi atualizado em outros momentos durante a escrita da monografia;
- **Escrever Referencial Teórico:** essa atividade resultou no Capítulo 2 - [Referencial Teórico](#), que levanta os conceitos teóricos que fundamentam a elaboração da abordagem e o desenvolvimento do Site Educacional proposto. Ressalta-se que essa atividade foi realizada, na verdade, em paralelo à atividade anterior, mais foi concluída de forma efetiva posteriormente, uma vez que reflete todo o embasamento da proposta. A escrita do capítulo teve como base, principalmente, os materiais coletados durante a atividade de [Levantamento Bibliográfico](#), mas também outros levantados sob demanda durante o desenvolvimento do capítulo;
- **Definir Métodos:** atividade em que se definiram as práticas metodológicas para realização do [Levantamento Bibliográfico](#); os métodos e as práticas que serão aplicados para o [Desenvolvimento](#) do Site Educacional; e os detalhes do método pesquisa-ação escolhido para orientar a [Análise de Resultados](#) obtidos neste trabalho. Os resultados dessa atividade são apresentados no presente Capítulo 4 - Metodologias;
- **Elaborar Proposta:** a elaboração da proposta, no intuito de conferir uma ideia mais concreta sobre o produto final desse trabalho, inclui uma versão preliminar da abordagem orientada à gamificação, bem como o protótipo do Site Educacional, ambos fundamentados nos conceitos apresentados no Capítulo 2 - [Referencial Teórico](#), e detalhados no Capítulo 5 - [Abordagem e Site Educacional](#);
- **Revisar e Finalizar Monografia:** é finalizada a escrita da monografia, apontando as Considerações Finais acordadas na primeira versão dessa monografia, e cumprindo o processo de revisão para corrigir erros e efetuar possíveis melhorias. Ressalta-se que o processo de revisão ocorreu desde o início, sendo realizado um acompanhamento semanal, e recorrentemente incorrendo em revisões por capítulo dessa monografia, e
- **Apresentar TCC 1 à Banca:** o trabalho foi enviado à banca e apresentado.

Ressalta-se que, com exceção da atividade “Apresentar TCC 1 à Banca”, todas as atividades, bem como o subprocesso de Realização de Levantamento Bibliográfico, foram concluídas ao longo da primeira etapa do TCC.

Figura 9 – Fluxograma de Atividades e Subprocessos da Primeira Etapa do TCC



Fonte: Autores

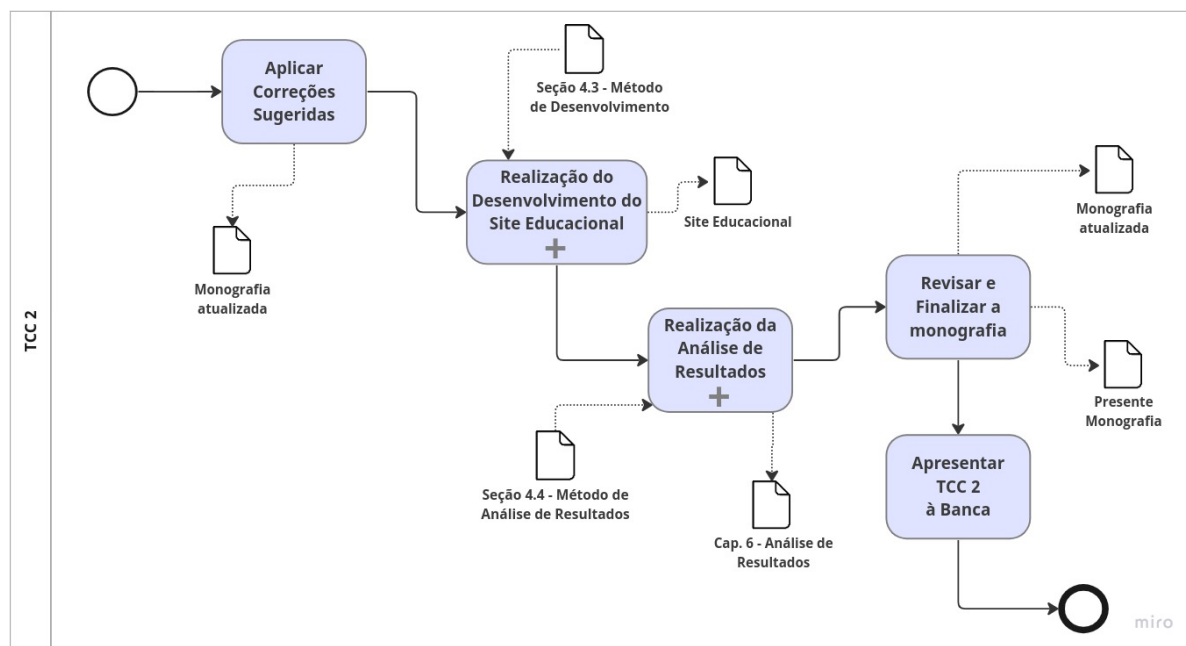
#### 4.5.2 Atividades e Subprocessos - Segunda Etapa do TCC

- **Aplicar Correções:** foram aplicadas correções e ajustes apresentados e sugeridos pela banca;
- **Realização do Desenvolvimento do Site Educacional:** foi realizado todo o processo de desenvolvimento do Site Educacional, com os ciclos de *sprints*, como definido na seção 4.3 - [Método de Desenvolvimento](#);
- **Realização da Análise de Resultados:** foi realizado todo o processo relacionado à avaliação da proposta de abordagem e do Site Educacional, com encaminhamento de questionário para os participantes. Posteriormente, os resultados obtidos com a avaliação foram analisados, com base no método definido na seção 4.4. - [Método de Análise de Resultados](#);
- **Revisar e Finalizar Monografia:** foi finalizada a escrita na monografia, fazendo necessário um processo de revisão para corrigir erros e efetuar possíveis melhorias, resultando do Capítulo 7 - [Conclusão](#);
- **Apresentar TCC 2 à Banca:** o trabalho final é enviado à banca e apresentado juntamente ao Site Educacional.

Ressalta-se que essas atividade e esses subprocessos foram realizados ao longo da segunda etapa do TCC.



Figura 10 – Fluxograma de Atividades e Subprocessos da Segunda Etapa do TCC



Fonte: Autores

## 4.6 Cronogramas

Com o objetivo de planejar e organizar de forma mais adequada os prazos de entrega deste trabalho, foram elaborados os cronogramas das atividades definidas no [Fluxo de Atividades](#). Os cronogramas para a primeira e a segunda etapas do TCC são apresentados, respectivamente, na Tabela 2 e na Tabela 3.

Tabela 2 – Cronograma da Primeira Etapa do TCC

Atividade	Abril	Mai	Junho	Julho
Definir tema	X			
Realização do Levantamento Bibliográfico	X	X		
Definir escopo	X			
Definir Suporte Tecnológico		X		
Escrever Referencial Teórico		X		
Definir Métodos			X	
Elaborar Proposta			X	
Revisar e Finalizar Monografia				X
Apresentar TCC 1 à Banca				X

Fonte: Autores

Tabela 3 – Cronograma da Segunda Etapa do TCC

<b>Atividade</b>	<b>Agosto</b>	<b>Setembro</b>	<b>Outubro</b>	<b>Novembro</b>	<b>Dezembro</b>
Aplicar Correções	X	X			
Realização do Desenvolvimento do Site Educacional	X	X	X	X	
Realização da Análise de Resultados				X	X
Revisar e Finalizar Monografia					X
Apresentar TCC 2 à Banca					X

Fonte: Autores

## 4.7 Resumo do Capítulo

Este capítulo apresentou a visão geral da metodologia deste trabalho, o que compreende a classificação da pesquisa quanto à sua abordagem, natureza, objetivos e procedimentos, bem como os métodos específicos adotados para levantamento bibliográfico, desenvolvimento e análise de resultados. Por fim, foram evidenciados os fluxos de atividades referentes às primeira e segunda etapas do TCC, assim como os respectivos cronogramas das mesmas. Resumidamente, a pesquisa é classificada como qualitativa na abordagem; aplicada na natureza; exploratória nos objetivos, e pesquisa bibliográfica e pesquisa-ação nos procedimentos. Orienta-se por levantamento bibliográfico consultando bases científicas para realização da pesquisa bibliográfica; uma abordagem combinada de Scrum, XP e Kanban para desenvolvimento, e protocolo de pesquisa-ação para análise de resultados.

## 5 Abordagem e Site Educacional

O principal foco desse capítulo é conferir detalhes sobre a abordagem e o site educacional desse trabalho. Dentre esses detalhes, consta uma versão da abordagem apoiando-se no plano de gamificação definido, o protótipo de alta fidelidade do Site Educacional, bem como imagem das telas do Site desenvolvido.

Inicialmente, é feita uma [Contextualização](#) quanto à escolha do tema do trabalho. Na sequência, são apresentadas etapas do processo de ideação, com análises de soluções similares já existentes, ou seja, um [Benchmarking](#), bem como um [Brainstorming](#) e a descrição de [Personas](#). Esses elementos explicitados contribuíram para abranger de forma geral os requisitos da abordagem gamificada, assim como servir de guia para decisões de design e usabilidade do Site Educacional que será desenvolvido. Na sequência, são apresentadas a [Abordagem gamificada](#) e o [Site Educacional](#). Por fim, tem-se o [Resumo do Capítulo](#).

### 5.1 Contextualização

A escolha do tema foi iniciada em uma visão mais ampla, em uma área de interesse de ambos os autores: a Gamificação. Dado o próprio contexto educacional em que se encontram os autores, durante o fechamento de escopo desse trabalho, refinou-se o tema. O tema refinado combina, portanto: Gamificação e Ensino-Aprendizagem. Sendo assim, pretende-se utilizar a gamificação para ensino-aprendizagem na área de software, tomando como base um tópico de estudo de difícil assimilação por parte dos alunos, uma vez que a ideia é usar elementos de jogos no processo de ensino-aprendizagem desses alunos, e perceber se as inerentes dificuldades podem ser mitigadas.

No contexto educacional, é possível a criação de abordagens não tradicionais para o aprendizado de algum conteúdo, seja ele complexo ou não. Tais abordagens podem ter o intuito de serem utilizadas como fonte principal de conhecimento sobre o assunto, ou então podem ser apenas auxiliares nos estudos.

Quando um tópico frequentemente causa dúvidas, ou maior dificuldade para ser compreendido, ou ainda gera desinteresse por parte dos alunos, o uso de metodologias alternativas torna-se algo desejado. Padrões de projeto em software podem ser vistos como exemplos desse tipo de tópico. Sendo normalmente abordados nos conteúdos de disciplinas como Arquitetura de Software e afins, representam verdadeiros desafios para os professores, em termos de ensino; e para os alunos, em termos de aprendizado.

Nesse cenário desafiador, o uso de abordagens diferenciadas pode trazer benefícios aos envolvidos, em especial, aos que se encontram na posição de aprendizes. Sendo a gamificação muito utilizada atualmente em diferentes contextos, inclusive no educacional, desenvolver uma abordagem orientada à gamificação pode ser uma proposta pertinente para conferir apoio ao ensino e à aprendizagem de padrões de projeto.

Além disso, considerando o uso das tecnologias nos dias atuais, bem como a realidade dos estudos *online* e autônomos, o desenvolvimento do Site Educacional pode conferir vantagens adicionais, disponibilizando os conteúdos de forma facilitada. Além disso, o site foi elaborado orientando-se pelas heurísticas de Usabilidade de Nielsen e demais boas práticas acordadas em detalhes na seção 2.4, do Capítulo 2 - [Referencial Teórico](#). Diante do exposto, acredita-se que haverá maior engajamento por parte dos interessados e, conseqüentemente, espera-se um maior aprendizado.

Como última consideração, em termos de contexto, busca-se, complementarmente, oferecer insumos para que a essência desse trabalho seja útil em trabalhos de similar demanda, ou seja: aqueles que desejarem replicar a ideia, para atendimento ao ensino / à aprendizagem de outros tópicos de estudo que se aproximam das características descritas anteriormente, podem usar os referenciais teórico-tecnológicos desse trabalho, bem como as metodologias. Isso possibilitará obterem soluções que mitiguem mais rapidamente demandas próximas. Lembrando que essas soluções já estarão orientadas às boas práticas da literatura especializada, cabendo apenas atualizações pontuais, e permitindo esforço concentrado nas demandas de alto nível de especificidade.

## 5.2 *Benchmarking*

Visando conhecer propostas relacionadas, ocorreu a necessidade de realizar um levantamento de soluções similares, mesmo considerando que essa similaridade ocorre em diferentes aspectos, seja no conteúdo de padrões de projeto; seja na própria implementação de gamificação. Para cada solução analisada, são apresentadas as principais características e os principais recursos, conteúdos abordados e, caso aplicável, até mesmo os elementos de gamificação presentes. Nesse levantamento, duas soluções que compreendem similaridades com a presente proposta merecem menção, sendo: Duolingo, e *Refactoring Guru*. Na seqüência, são conferidos detalhes sobre cada uma dessas soluções.

### 5.2.1 Duolingo

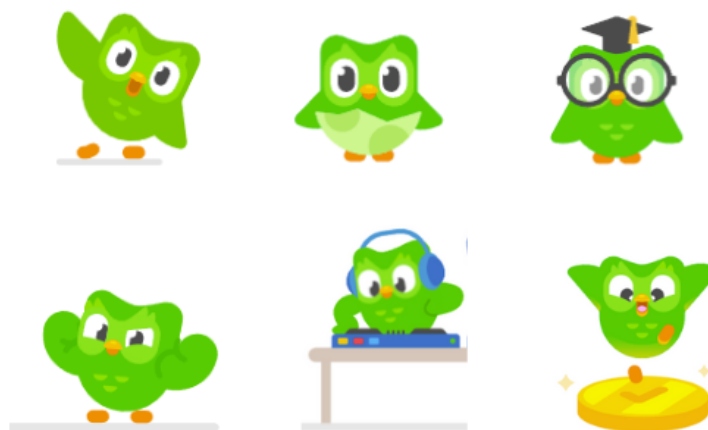
O Duolingo<sup>1</sup> é uma plataforma gamificada para a aprendizagem de idiomas. A plataforma apresenta uma abordagem competitiva, com o apoio de diferentes recursos de gamificação, como pontuação e *rankings*. O Duolingo possui uma interface gráfica

<sup>1</sup> <https://pt.duolingo.com/>. Acesso em 07 jul, 2023

amigável, com desenhos e mensagens motivadoras. Essa plataforma também é bastante reconhecida por seu mascote, uma coruja verde nomeada Duo (vide Figura 11).

Esta análise é direcionada para a versão *web* do Duolingo.

Figura 11 – Duo, o Mascote do Duolingo



Fonte: Duolingo, 2023

#### 5.2.1.1 Principais Recursos do Duolingo

A caminhada de aprendizagem no Duolingo é apresentada em etapas. Há diversas Unidades numeradas, em que cada uma possui uma trilha de níveis, desbloqueados à medida que o usuário completa os anteriores, como mostrado na Figura 12. Ao longo das trilhas, é possível também praticar exercícios a qualquer momento.

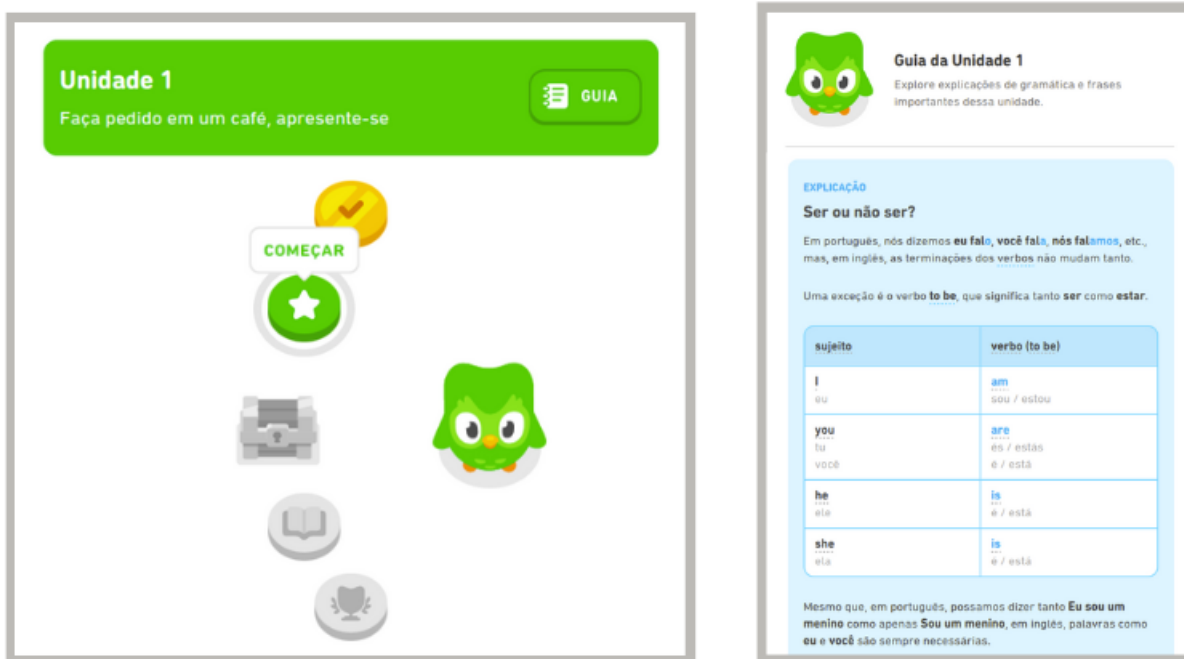
Ao completar lições da trilha e praticar exercícios, o usuário adquire pontos de experiência. Esses pontos de experiência servem tanto para completar algumas missões, quanto para competir em *rankings* com outros usuários.

Para começar novas lições, é necessário possuir vidas. Essas vidas são perdidas ao cometer erros durante uma lição, e podem ser adquiridas ao praticar exercícios; ao comprá-las utilizando cristais, ou ao esperar um determinado período. Ao completar missões e metas diárias, o usuário recebe cristais, que são utilizados para comprar vidas ou “superpoderes”. As vidas e os cristais são mostrados na Figura 13.

#### 5.2.1.2 Gamificação

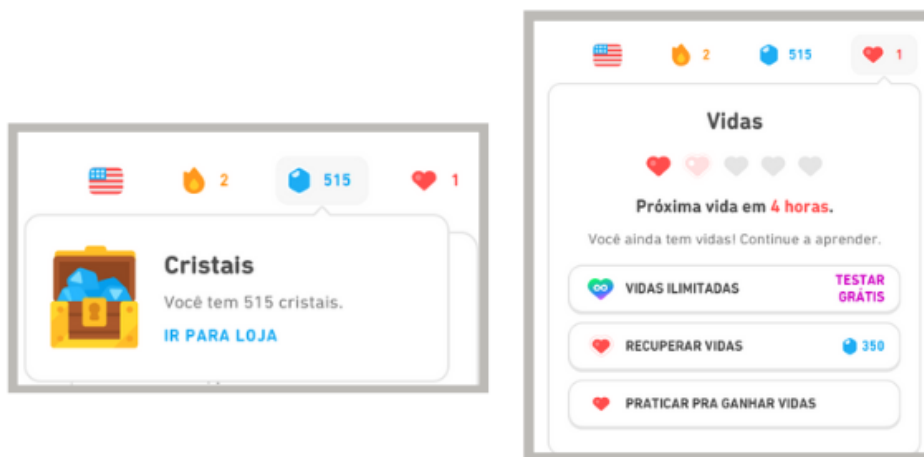
Com base na análise realizada na subseção anterior, é possível explicitar elementos de jogos utilizados para a gamificação no Duolingo, tendo como referência também o *framework Octalysis*.

Figura 12 – Trilha de Níveis da Unidade



Fonte: Duolingo, 2023

Figura 13 – Cristais e Vidas



Fonte: Duolingo, 2023

O Duolingo utiliza alguns recursos para impulsionar o usuário a progredir, como pontos e experiência; conquistas e barras de progresso nas lições e práticas, além de fornecer algumas recompensas aleatórias. Ademais, à medida que o usuário avança nas lições, são entregues *feedbacks* constantes e instantâneos a ele, utilizando-se também de ressaltar quando o usuário acerta várias questões em uma sequência. O representante que fornece esses *feedbacks* é o próprio mascote Duo.

Além disso, o Duolingo apresenta outras técnicas e outros recursos de jogos, como o acúmulo de bens virtuais, chamados “cristais”, e promove também uma competição entre os usuários, com as “ligas”.

O Quadro 3 representa os aspectos de gamificação identificados no Duolingo, com base no *framework Octalysis*.

Quadro 3 – Principais Elementos de Gamificação do Duolingo

<i>Core Drive</i>	<i>Aplicação</i>
1 - Significado Épico e Chamado	Participar de uma comunidade de aprendizado;
2 - Desenvolvimento e Realização	Pontos de experiência; Conquistas; Barras de progresso nas lições;
3 - Empoderamento da Criatividade e Feedback	Feedbacks constantes nas lições;
4 - Propriedade e Posse	Bens virtuais (Cristais) para trocar por outro item na loja;
5 - Influência Social e Pertencimento	Seguidores e amizades; Competição de usuários (Ligas);
6 - Escassez e Impaciência	"Vidas" para poder realizar lições; Esperar determinado tempo para recuperar "vidas" perdidas;
7 - Imprevisibilidade e Curiosidade	Recompensas aleatórias;
8 - Perda e Rejeição	Missões diárias; Manter suas "vidas"

Fonte: Autores

## 5.2.2 Refactoring Guru

O *Refactoring Guru*<sup>2</sup> é uma plataforma direcionada para consulta e aprendizagem de Refatoração e Padrões de Projeto. Ele está disponível em diversos idiomas, e possui conteúdo tanto gratuito quanto pago. O intuito do site é simplificar o aprendizado desses tópicos para os usuários. A página de entrada sobre os Padrões de Projeto é ilustrada na Figura 14. Como pode ser visto, há paleta de cores bem definida, bem como imagens que chamam a atenção de quem acessa o site.

### 5.2.2.1 Principais Recursos do *Refactoring Guru*

O *Refactoring Guru* é uma referência no ensino de padrões de projeto. Tendo como base os Padrões GOF's definidos por Gamma et al (1995), ele traz, de uma maneira simples, a explicação do que são os padrões de projeto apresentados. O site ilustra muito bem sobre problemas, soluções, estruturas e códigos de todos os padrões de projeto, expli-

<sup>2</sup> <https://refactoring.guru/>. Acesso em 05 jul, 2023

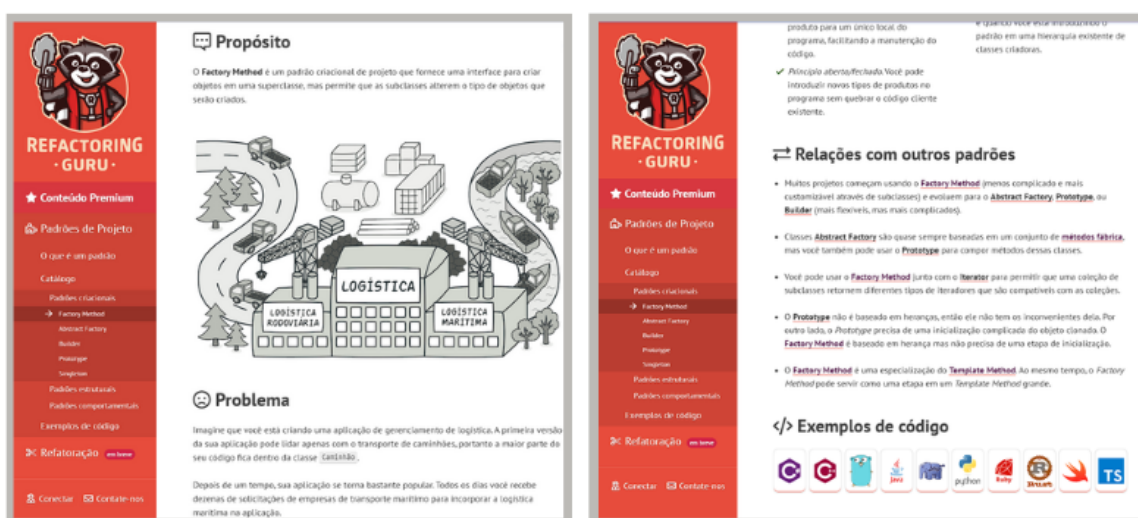
Figura 14 – Tela Inicial do Refactoring Guru



Fonte: Refactoring Guru, 2023

cando com um vocabulário mais simples, e tentando trazer uma solução que exemplifique a aplicação desse padrão.

Figura 15 – Outras Telas do Refactoring Guru



Fonte: Refactoring Guru, 2023



É relevante destacar que o *Refactoring Guru* apresenta códigos funcionais para cada padrão de projeto, em diversas linguagens de programação, além de fornecer pseudocódigos e diagramas para ilustrar os padrões, como na Figura 15. Nesse caso, há menção ao padrão de projeto *Factory Method*, acordando sobre o propósito e o problema de forma textual e ilustrada, além de conteúdos mais aprofundados, tais como as principais relações com outros padrões de projeto e exemplos de código.

O site não apresenta os padrões de projeto em uma ordem específica que oriente o usuário, ou seja, o usuário fica livre para navegar. Dessa forma, assume que o usuário, que deseja aprender o conteúdo via esse site, já tem maturidade o suficiente em Engenharia de Software para entender o que está procurando e aprendendo. Sendo assim, seria desejado que o site tivesse uma forma de apresentar os padrões por níveis de dificuldade, conforme explorado pelo Duolingo. Essas e outras estratégias mais didáticas são objetos de estudo do presente trabalho, que pretende desenvolver uma solução combinando boas práticas dessas plataformas, além de outras soluções encontradas com apoio da literatura especializada.

### 5.3 *Brainstorming*

Os autores fizeram um *brainstorming* para reunir diversas ideias iniciais sobre diferentes aspectos do Site Educacional e da abordagem, tais como: funcionalidades; elementos visuais, e elementos de gamificação, conforme constam apresentados na Figura 16. Foram utilizados *post-its* virtuais, visando anotar cada *feedback* e ponto de reflexão conferido pelos autores em tempo de *brainstorming*. Tais anotações permitiram dar continuidade ao processo de elaboração da abordagem, bem como do Site Educacional, com a definição de [personas](#) e, posteriormente, o detalhamento da [abordagem gamificada](#) e o desenvolvimento do [Site Educacional](#) baseado no [protótipo de alta fidelidade](#).

Figura 16 – *Brainstorming*

Fonte: Autores

## 5.4 Personas

Foram mapeadas algumas personas que representam o público-alvo da abordagem e do Site Educacional, bem como anti-personas para as quais a abordagem não é direcionada. As personas mapeadas se tratam de uma estudante e um professor, ambos da área de tecnologia e programação, bem como um desenvolvedor de software autodidata. As anti-personas se tratam de pessoas que não possuem interesse na área de padrões de projeto, ou então, que não possuem conhecimentos necessários para começar o estudo de padrões de projeto. As especificações constam nas Figuras de 17 a 21, com Persona, Nome, Idade, Profissão, e Descrição, além de uma imagem. As imagens foram retiradas da fonte “*This Person Does Not Exist*”.

Figura 17 – Persona Estudante

**Persona:** Estudante

**Nome:** Ana Silva

**Idade:** 21 anos

**Profissão:** Estudante de Engenharia de Software



Fonte: This Person Does Not Exist, 2023

**Descrição:** Ana é uma estudante dedicada e curiosa. Ela está sempre buscando maneiras de aprimorar seus conhecimentos em programação e desenvolvimento de *software*. Ana gosta de desafios e está disposta a experimentar novas abordagens e tecnologias. Ela procura padrões de projeto que possam ajudá-la a escrever código limpo e de fácil manutenção. Ana está aberta a aprender com os erros e valoriza o trabalho em equipe.

Figura 18 – Persona Professor

**Persona:** Professor

**Nome:** Carlos Santos

**Idade:** 51 anos

**Profissão:** Professor de Ciência da Computação



Fonte: This Person Does Not Exist, 2023

**Descrição:** Professor Carlos é um profissional experiente e dedicado. Ele possui amplo conhecimento em padrões de projeto e acredita na importância de ensiná-los aos seus alunos. Ele busca maneiras de transmitir os conceitos de forma clara e prática, utilizando exemplos do mundo real. Professor Carlos incentiva seus alunos a utilizar padrões de projeto como ferramentas para resolver problemas de software complexos. Ele valoriza a colaboração e a troca de conhecimento entre os estudantes.

Figura 19 – Persona Autodidata

**Persona:** Autodidata

**Nome:** Pedro Oliveira

**Idade:** 29 anos

**Profissão:** Desenvolvedor de *Software* Autodidata



Fonte: This Person Does Not Exist, 2023

**Descrição:** Pedro é um autodidata apaixonado por programação. Ele aprendeu a desenvolver software por conta própria, utilizando diversos recursos *online*. Pedro gosta de explorar diferentes padrões de projeto para aprimorar suas habilidades e resolver problemas de maneira eficiente. Ele valoriza a liberdade de experimentação e aprendizado independente. Pedro está sempre em busca de novos desafios e oportunidades de crescimento profissional.

Figura 20 – Anti-Persona Gerente

**Anti-Persona**

**Nome:** Mariana Torres

**Idade:** 38 anos

**Profissão:** Gerente de Projetos



Fonte: This Person Does Not Exist, 2023

**Descrição:** Mariana é uma gerente de projetos com pouca familiaridade com conceitos de desenvolvimento de software. Ela costuma tomar decisões e mudar constantemente de ideia, o que dificulta o progresso dos projetos. Mariana tem dificuldade em entender a importância dos padrões de projeto, e tende a optar por soluções rápidas e improvisadas. Ela não está aberta a investir tempo e recursos em aprender e implementar padrões de projeto adequados.

Figura 21 – Anti-Persona Estudante

**Anti-Persona****Nome:** Eric Gomes**Idade:** 16 anos**Profissão:** Estudante do Ensino Médio

Fonte: This Person Does Not Exist, 2023

**Descrição:** Eric, um estudante do 2º ano do Ensino Médio, demonstra interesse em cursar uma faculdade na área de software e, por isso, começou recentemente a estudar lógica de programação. Com conhecimento limitado em lógica de programação e pouca experiência prática, Eric não conhece orientação a objetos nem padrões de projeto, não tendo ainda maturidade e experiência recomendadas para iniciar o aprendizado de padrões de projeto. Eric gosta de aplicar seus conhecimentos resolvendo exercícios de programação *online*.

## 5.5 IcePatterns

Nessa seção, é apresentada uma [Visão Geral](#) sobre as características e os recursos principais da abordagem e do site. Em seguida, é apresentada a [Abordagem Gamificada](#), com o detalhamento do [Plano de Gamificação](#), e o [Site Educacional](#), com as decisões de design e usabilidade.

### 5.5.1 Visão Geral

No intuito de já ambientar o leitor no sentido da gamificação, o conceito geral seguirá uma identidade de gelo e frio, com elementos como iceberg e cubos de gelo.

A essência da trilha de aprendizagem de padrões de projeto está contida em um **iceberg**, que será dividido em **níveis**. Há uma ordem de aprendizagem, ou seja, apenas após completar o nível 1, será possível estudar os padrões de projeto contidos no nível 2. Em cada nível, haverá explicações acerca dos padrões e, para cada um, terão exercícios para fixar e testar os conhecimentos aprendidos. Ao final de cada nível, é possível realizar o “desafio”, que consistirá basicamente em exercícios que abordam todos os padrões anteriores. Ao passar pelo desafio, será possível alcançar o próximo nível.

Os **pontos de experiência** também serão considerados para que o usuário possa avançar para o próximo nível. Só será possível realizar o “desafio”, após completar todos os pontos de experiência necessários no nível em que se encontra atualmente.

Será utilizado o conceito de **energia** como uma forma de limitar a quantidade de atividades que o usuário conseguirá realizar de uma vez. Aqui, já ocorre o uso de boas práticas adotadas por educadores, que adotam estratégias mais atuais em termos educacionais. Nessas estratégias, estabelecer limites, e procurar dosar o tempo e o esforço gastos em uma dada atividade, tendem a beneficiar o aluno, uma vez que não incorrem em esgotamentos ou jornadas exaustivas de aprendizagem. O aprendizado precisa ser prazeroso, visando engajar com naturalidade o aprendiz. As energias serão recarregadas automaticamente ao longo do tempo, mas poderão também ser recarregadas através da troca por cubos de gelo na loja.

Os **cubos de gelo** serão a moeda de troca no site. Serão entregues como recompensas para missões e exercícios, e o usuário poderá receber uma quantidade máxima de cubos de gelo diariamente. Dessa forma, precisará fazer trocas de maneira estratégica, por exemplo, ao trocar por energias ou impulsionadores, explicados na seção do [Plano de Gamificação](#).

As **missões** serão tarefas que o usuário poderá realizar dentro do site. Ao concluí-las, receberá recompensas, que poderão ser pontos de experiência; cubos de gelo, ou outros itens. As dificuldades das missões serão conforme o nível que o usuário estará.

As **conquistas** serão uma forma de reconhecer alguma realização ou marco na jornada de aprendizagem do usuário, podendo cada uma delas ser obtida apenas uma vez.

## 5.5.2 Abordagem Gamificada

Nessa seção, é apresentada a abordagem orientada à gamificação, com o detalhamento do plano de gamificação.

### 5.5.2.1 Plano de Gamificação

Com base no *framework Octalysis* proposto por Chou (2015), os autores definiram os elementos de gamificação que foram aplicados na abordagem e no Site Educacional. Portanto, as seções apresentadas a seguir orientam-se pelas motivações descritas na seção 2.3, do Capítulo 2 - Referencial Teórico.

#### 5.5.2.1.1 1 - Significado Épico e Chamado

As pessoas sentem-se motivadas quando pensam estar em uma causa, trabalhando por um bem maior. Na abordagem proposta, não há muito destaque para essa motivação

com aplicação de técnicas específicas. Entretanto, há a intenção de estudar e praticar visando desbloquear os níveis mais profundos do iceberg, avançando no aprendizado.

#### 5.5.2.1.2 2 - Desenvolvimento e Realização

Para motivar o usuário a continuar evoluindo no aprendizado, tem-se foco no seu senso de progresso e na sua necessidade de alcançar um objetivo definido. Considerando isso, buscou-se aplicar os seguintes elementos de gamificação:

- Missões: são objetivos bem definidos para que o usuário alcance-os aplicando determinado esforço. As missões devem ser alcançáveis, porém devem representar desafios para que o usuário se sinta atraído a concluir a missão e receber a recompensa. Esse equilíbrio entre algo ser desafiador o suficiente para não desestimular o aprendiz, ao mesmo tempo que passível de cumprimento por parte do aprendiz, é uma boa prática educacional. As missões são desbloqueadas ao longo do aprendizado e dos níveis, buscando manter o interesse e curiosidade do aprendiz. As missões definidas nessa primeira versão da abordagem foram:
  - “Complete sua primeira bateria de exercícios com mais de 60% de acerto”
  - “Ative e utilize um impulsionador”
  - “Complete um desafio com sucesso”
- Conquistas: devem ser significativas, ou seja, devem representar algo de valor único para o usuário. As conquistas existentes não são necessariamente alcançadas por todos os usuários, fazendo com que o usuário sinta-se valorizado por receber tal conquista pelo seu sucesso. Novamente, trata-se de uma boa prática educacional conferir algo mais significativo ao aprendiz, quando um feito maior é realizado. Isso tende a estimulá-lo a realizar novos feitos maiores, em busca desse reconhecimento. As conquistas definidas nessa primeira versão da abordagem foram:
  - “Você completou seu primeiro nível no Iceberg!”
  - “Você realizou sua primeira compra na loja!”
  - “Você acumulou mais de 1.000 cubos!”
  - “Você subiu de nível estando impulsionado!”
  - “Você completou a trilha do Iceberg!”
- Pontos de experiência: o usuário receberá pontos de experiência por práticas de exercícios. Esses pontos permitem que o usuário avance de nível quando atingir a pontuação máxima de experiência do nível em que se encontra, visualizando essa

pontuação por meio de uma barra de progresso. Com base na literatura especializada, conferir um retorno que permite ao aprendiz quantificar seus feitos e seu progresso ajuda esse aprendiz a se sentir realizado;

- Barras de progresso: além de ser utilizada para representar os pontos de experiência, estará presente nas missões e nos módulos de exercícios. Dessa forma, elas precisam ser significativas, trazendo a noção de progresso, assim, mantendo o usuário ciente de seu avanço em determinada atividade. Conferir *feedback* não é apenas uma boa prática educacional, mas também uma boa prática de design. A noção de que há um acompanhamento constante, bem como a percepção de como está evoluindo seus aprendizados, são elementos valiosos, deixando claro ao aprendiz que ele está sendo assistido, e
- Subir de níveis: eles serão o ponto focal para o usuário entender o seu aprendizado no conteúdo. Os níveis estão diretamente relacionados aos pontos de experiência do usuário, e mostrarão o trajeto por onde ele passou; o que já aprendeu, e o conteúdo que conquistou até o momento. Quanto maior o nível, mais fundo no iceberg do conhecimento o usuário estará. Novamente, esses retornos conferem ao aprendiz a noção clara de que está sendo assistido constantemente.

#### 5.5.2.1.3 3 - Empoderamento da Criatividade e *Feedback*

Busca-se engajar o usuário com sentimentos de poder e de liberdade de criatividade, também apresentando *feedback* a ele.

- Impulsionadores: são influências limitadas entregues ao usuário (ou compradas por ele) como incentivo para completar mais missões e receber mais recompensas, como, por exemplo: “Recebe o dobro de recompensas ao completar missões ou exercícios pelos próximos 15 minutos”. Estimular a realização de avanços em trilhas de aprendizagem é um recurso comumente utilizado por educadores para garantir que o aprendiz tenha interesse e incentivo em conferir um aprendizado contínuo. Nesse contexto, típicos impulsionadores são pontos extras entregues aos aprendizes com desempenhos diferenciados. Ao receber esses pontos extras, o aprendiz tende a se sentir empoderado e encorajado para ir além.

#### 5.5.2.1.4 4 - Propriedade e Posse

A motivação pode ser conduzida pelo sentimento de possuir algo e, como consequência, desejar mantê-lo e obter mais dele.

- Pontos permutáveis: seriam pontos de trocas. Na abordagem, esses pontos (cubos de gelo) podem ser obtidos ao concluir missões e exercícios; e serão utilizados para



recarregar energias e comprar impulsionadores. Novamente, a ideia é conferir motivadores que passam a pertencer ao aprendiz, podendo por ele serem trocados por algo de interesse. Isso já ajuda a criar um sentimento de pertencimento, além de estimular que o aprendiz aprenda a se desprender e administrar esse recurso em função de novos anseios e novas estratégias de aprendizagem. O aprendiz passa a fazer parte de forma mais efetiva na construção do seu próprio conhecimento.

#### 5.5.2.1.5 5 - Influência Social e Pertencimento

Esse *core drive* trabalha com atividades que envolvem o que outras pessoas pensam e fazem. Baseia-se, portanto, no desejo das pessoas se conectarem e se compararem entre si. Tanto na abordagem, quanto no Site Educacional, não há foco na parte social, pelo menos não nessa primeira versão. Entretanto, ao receber conquistas, por exemplo, o usuário pode, de forma externa ao site, mostrar e falar sobre elas, conferindo valor aos seus feitos, além de claro compartilhamento. Esse compartilhamento de feitos e avanços nos aprendizados podem, e frequentemente o fazem em diferentes abordagens de ensino-aprendizagem, proporcionar o interesse de outros aprendizes. Sendo assim, há um efeito que se assemelha aos conceitos e propósitos de uma rede social, com aprendizes influenciando outros aprendizes.

#### 5.5.2.1.6 6 - Escassez e Impaciência

Consiste em motivar o usuário, simplesmente, pelo fato de não ser possível possuir algo imediatamente, ou então por haver certa dificuldade em se obter algo.

- Intervalos de tortura (*Torture Breaks*): não permitir que o usuário faça algo imediatamente. Na abordagem e no site, haverá o item de energia, que será finito e consumido toda vez que o aprendiz realizar uma atividade. Quando esgotar todas as suas energias, o usuário deverá esperar um determinado tempo para que elas sejam recarregadas automaticamente ou então compradas por ele, para poder voltar a realizar atividades. Essa estratégia provoca ansiedade, que tende a incentivar aprendizes, os quais aguardam desejosos pelo recurso “escasso” para continuar investindo nas trilhas de aprendizagem;
- Pontos permutáveis: a aplicação desse elemento foi citada no *core drive 4* (Propriedade e Posse). No atual cenário, a ideia é tratar sua escassez, ou seja, o usuário precisa utilizar esses pontos de forma estratégica. É importante que a abordagem tenha uma adequada relação entre o esforço para obter os pontos e o valor dos itens e recursos que podem ser comprados/trocados por esses pontos. Ao encontrar

coerência nesse balanceamento, há possibilidade de provocar nos aprendizes, novamente, ansiedade na medida certa e, portanto, o desejo de continuarem nas trilhas de aprendizagem, evoluindo seus conhecimentos.

#### 5.5.2.1.7 7 - Imprevisibilidade e Curiosidade

O usuário pode ser motivado por experiências que são incertas e envolvem possibilidade e imprevisibilidade.

- *Glowing Choice*: atrair o usuário para a direção correta utilizando a sua curiosidade. No site, haverá pequenos mecanismos que ajudarão o usuário a se situar e explorar. Por exemplo, quando for a primeira vez que ele entrar no site, ou quando desbloquear uma nova funcionalidade, será ajudado para entender o que ele pode fazer. A ideia é guiar esse aprendiz, conferindo graus de liberdade, mas de forma coerente e didática, para que ele vá avançando no aprendizado. Sendo assim, *pop-ups*, e outras provocações usando elementos de interface gráfica, podem ser recursos úteis, uma vez combinados e apresentados de forma estratégica ao aprendiz.

#### 5.5.2.1.8 8 - Perda e Rejeição

Esse *core drive* procura motivar o usuário através do medo de perder algo ou de algum evento indesejado ocorrer, gerando um senso de urgência e obsessão. Relaciona-se diretamente com outros *core drives*, como apresentado nos elementos a seguir:

- Perda de energia: ocorrerá naturalmente quando o usuário realizar atividades. Porém, ao apresentar um desempenho não satisfatório nos exercícios, ou seja, errar muitas questões em uma atividade, será perdido o dobro de energia. Isso ocorrerá com o intuito de motivar o esforço do usuário para não perder mais do que o necessário. Entretanto, essa estratégia será utilizada em conjunto com outras estratégias que conferem alternativas ao aprendiz. A intenção é evitar desestímulo ou abandono por parte do aprendiz pelo simples fato dele não ter cumprido com algo de maior dificuldade, incompatível com seu nível de conhecimento.
- Impulsionador: como apresentado no *core drive* 3 (Empoderamento da Criatividade e *Feedback*), haverá impulsionadores que possuem tempo limitado para serem usufruídos pelo usuário. Assim, haverá uma contagem regressiva, mostrando qual é o tempo restante do efeito do impulsionador, para ser mais bem reconhecido esse limite de tempo, despertando a necessidade do usuário de utilizar ao máximo seu impulsionador durante esse tempo.

### 5.5.3 Site Educacional

Nessa seção, são detalhados os aspectos do Site Educacional desenvolvido, baseado na abordagem gamificada. São apresentadas as decisões de usabilidade e de design, bem como o protótipo de alta-fidelidade que orientou o desenvolvimento da interface do site. Por fim, são detalhados os conteúdos e exercícios de padrões de projeto de software abordados nessa primeira versão do Site Educacional, bem como as telas e fluxos que o compõem.

O site está hospedado no link <https://icepatterns.netlify.app/>.

#### 5.5.3.1 Usabilidade

Para o desenvolvimento do Site Educacional, a usabilidade foi levada em consideração, com foco principalmente nas heurísticas de Nielsen, além de alguns outros aspectos detalhados na seção [Site Educacional](#) do Capítulo 2 - [Referencial Teórico](#). Com essas considerações, buscou-se uma melhor experiência do usuário ao interagir com o site.

Para a melhor usabilidade do site, foi dado foco às seguintes heurísticas de Nielsen:

- Design simples e natural: utiliza uma interface que não possui recursos e itens adicionais sem necessidade, apenas apresente ao usuário o necessário, no momento necessário. Por exemplo, também, o uso de cores é feito sem exageros, e garantindo que a interface não dependa das cores, ou seja, que seja possível compreendê-la mesmo sem as cores, sendo as mesmas utilizadas mais para categorizar e destacar;
- Uso da linguagem do usuário: foi utilizada a linguagem do usuário, especificamente em português, sem-termos de sistema;
- Consistência e padrões: foi feita uma padronização das ações no site, bem como consistência de elementos gráficos, detalhados na seção [Design](#) deste capítulo;
- *Feedback*: buscou-se constantemente informar ao usuário o que está ocorrendo, e como as escolhas dele estão sendo interpretadas pelo sistema, aplicando *feedbacks* de baixa, média e alta duração, dependendo da necessidade de apresentação do mesmo;
- Boas mensagens de erro: são utilizadas mensagens claras, relatando o problema ao usuário de forma direta, e
- Ajuda e documentação: são apresentadas mensagens para alguns elementos gráficos do site, para ajudar um usuário iniciante a entender alguns itens, principalmente os mais comuns à gamificação, não utilizados em site mais tradicionais.

Outras práticas também foram aplicadas:

- Visualização e histórico: com foco na navegação do usuário pela interface, são apresentadas ao usuário quais atividades já foram realizadas, mantendo um rastro, e ainda assim permitindo que ele revise tais atividades, e
- Página inicial com informações relevantes: foi desenvolvida uma página inicial do site, que pode ser acessada por usuários visitantes, ou seja, que não possuem cadastro. Nessa página, estão contidas informações principais sobre o site, como uma breve explicação sobre ele e sobre o que está contido nele, bem como as opções de se cadastrar ou entrar como usuário já cadastrado;

### 5.5.3.2 Design

A seguir, são apresentados os principais detalhes inerentes ao design do Site Educacional, com destaque para cores, layout, tipografia e iconografia.

#### 5.5.3.2.1 Cores

As cores **principais** do site são no tom de azul, que evoca uma sensação de frio e tranquilidade. Tons mais escuros foram utilizados juntamente com tons claros, para criar uma atmosfera gelada e suave, e conferir maior contraste entre elementos.

Para cores **secundárias**, foi usado principalmente o branco como cor de fundo, para realçar a sensação de gelo e aumentar a legibilidade do conteúdo. O branco também foi utilizado para casos em que os elementos textuais estiverem sobre um fundo de tom escuro de azul. A cor preta também é utilizada ocasionalmente em alguns ícones e elementos textuais. A Figura 22 resume sobre a paleta de cores.

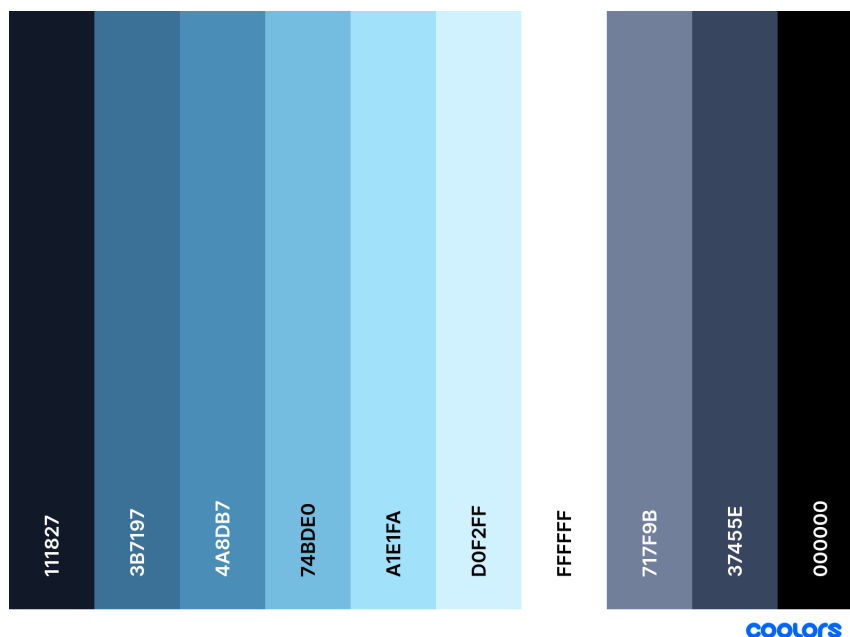
#### 5.5.3.2.2 Layout e Design

O design do site é limpo e transmite a sensação de tranquilidade e clareza, evitando sobrecarregar o site com muitos elementos visuais ou informações excessivas, e deixando espaço em branco suficiente para criar uma experiência visual agradável e equilibrada.

Imagens e ilustrações utilizadas são de alta qualidade e de acordo com o tema do site, complementando o design geral e adicionando um toque de autenticidade e imersão.

O espaçamento entre os elementos é adequado para tornar o site fácil de ler e navegar. Elementos como caixas de texto e sombras são aplicados de forma sutil. Os conteúdos relacionados são agrupados para criar uma hierarquia visual clara. Isso ajuda os visitantes a entenderem a organização e a estrutura do conteúdo de forma intuitiva.

Figura 22 – Paleta de Cores



Fonte: Autores

### 5.5.3.2.3 Tipografia

A tipografia é a fonte utilizada no site, e ela está atrelada à legibilidade, à simplicidade, à mensagem e à facilidade da leitura do usuário. A fonte escolhida foi a *Roboto*, como apresentada na Figura 23.

### 5.5.3.2.4 Iconografia

Para iconografia, são utilizados ícones da biblioteca *Font Awesome*<sup>3</sup>, que compreende uma grande variedade de opções. Os ícones são customizáveis para melhor atendimento ao site. Além disso, alguns ícones foram criados pelos autores, mostrados na Figura 24.

### 5.5.3.3 Protótipo do Site Educacional

Um protótipo de alta fidelidade foi inicialmente elaborado para conferir um maior detalhamento e uma visualização mais concreta da proposta do site educacional. Os fluxos e telas detalhados no protótipo estão disponíveis no Apêndice A - Protótipo de Alta Fidelidade e é possível interagir com o protótipo em <https://www.figma.com/proto/1bXXXNsC VJBBIu59fdIKWq/Alta-Fidelidade-TCC-1?type=design&node-id=1-3&t=DkYT1AIFrjs5feRP-1&scaling=scale-down&page-id=0%3A1&starting-point-node-id=1%3A3&mode=design>.

<sup>3</sup> <https://fontawesome.com/>. Acesso em 14 jul, 2023

Figura 23 – Tipografia



Fonte: Google Fonts, 2023

Figura 24 – Ícones



Fonte: Autores

#### 5.5.3.4 Conteúdos e Exercícios

Os conteúdos e exercícios definidos para o site foram elaborados utilizando como referência, principalmente, o livro *Design patterns: elements of reusable object-oriented software* de (GAMMA et al., 1995) e o site *Refactoring Guru*. Em sua atual e primeira versão, o Site Educacional não aborda todos os padrões de projeto. Os conteúdos e exercícios estão disponíveis no Apêndice B - [Conteúdo e Exercícios no Site Educacional](#).

#### 5.5.3.5 Páginas e Fluxos

A seguir são apresentadas as páginas do Site Educacional desenvolvido. A base para o desenvolvimento delas foi o protótipo de alta-fidelidade elaborado anteriormente,

porém com algumas modificações, tanto por decisão dos desenvolvedores, quanto por melhorias com base nos *feedbacks* recebidos durante os ciclos de análise de resultados.

A Figura 25 representa a página inicial ao acessar o site, em que o usuário encontrará breves informações sobre o conteúdo do site, assim como as opções de registrar-se como usuário ou entrar como usuário já cadastrado.

Figura 25 – Página Inicial de Visitantes



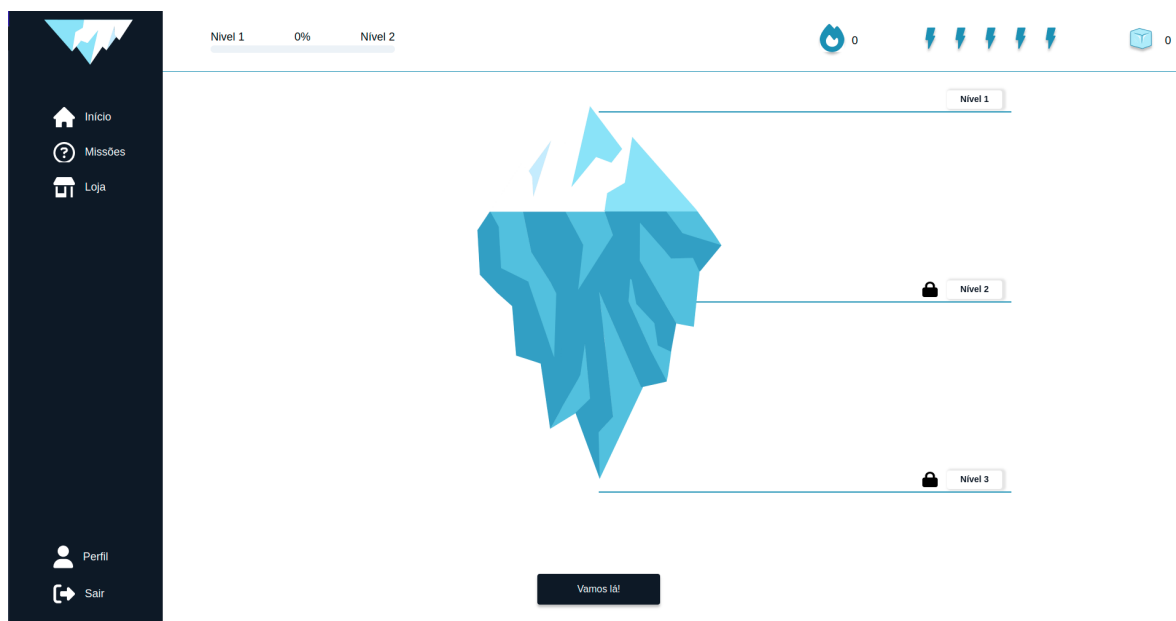
Fonte: Autores

Ao acessar o site realizando o login com sua conta já registrada, o usuário será recebido pela página inicial mostrada na Figura 26. Essa página contém alguns elementos que podem ser inicialmente desconhecidos. Portanto, para que o usuário não se sinta desamparado, além de motivar a sua curiosidade, são conferidos alguns “cartões” explicativos acerca desses elementos.

Na página inicial, bem como em todas as outras, há o menu lateral, que confere uma visão geral das páginas que o usuário pode acessar, e o redireciona diretamente para elas.

Com uma trilha de aprendizagem dividida em níveis, o site apresenta um iceberg dividido em camadas, cada qual representando um nível diferente. Ao selecionar um dos níveis desbloqueados, o usuário será redirecionado para a página mostrada na Figura 27.

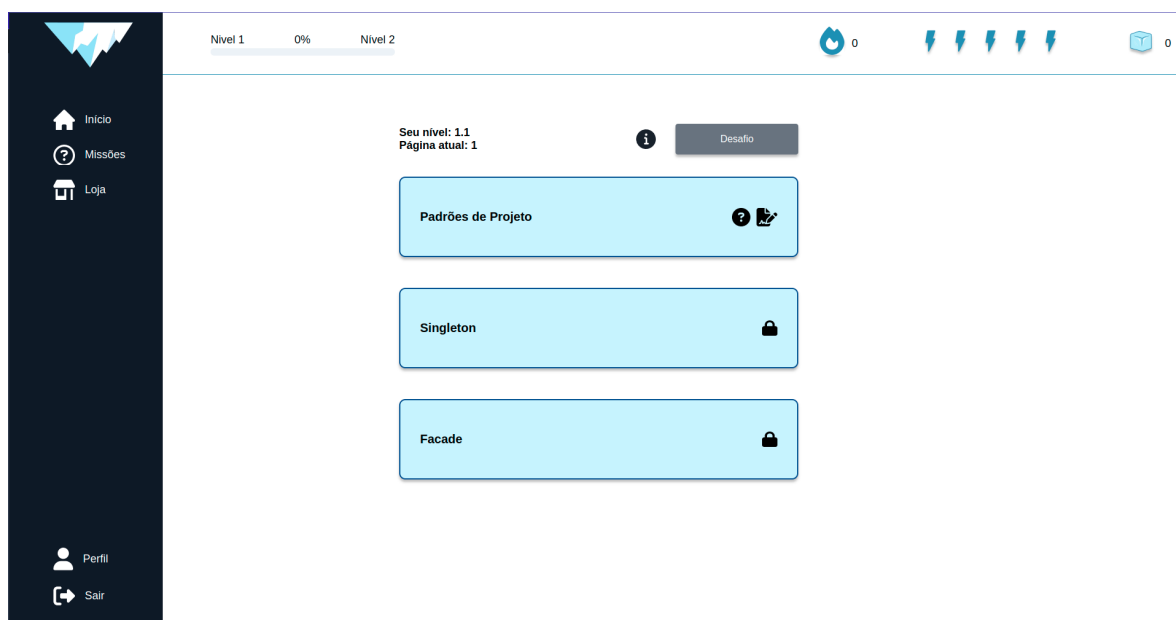
Figura 26 – Página Inicial para Usuário Registrado



Fonte: Autores

Nessa página, é possível visualizar os subníveis de aprendizado que serão abordados nesse nível, bem como as opções de ir para a página de conteúdo explicativo (Figura 28); ou para a página de exercícios (Figura 29) de um subnível.

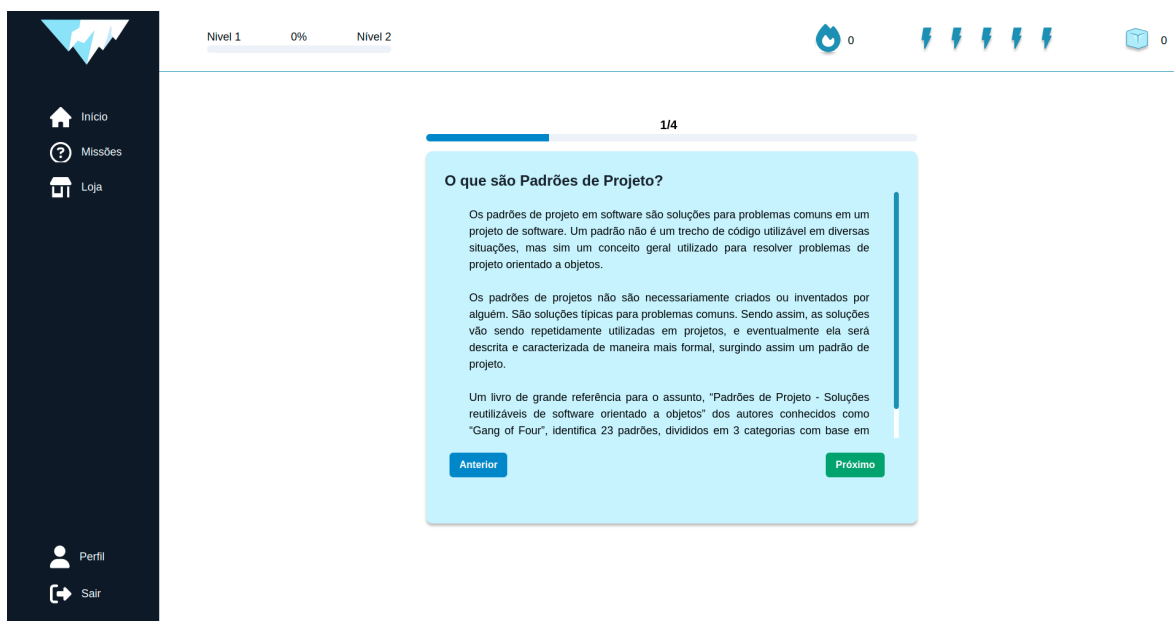
Figura 27 – Página de Nível e Subníveis



Fonte: Autores

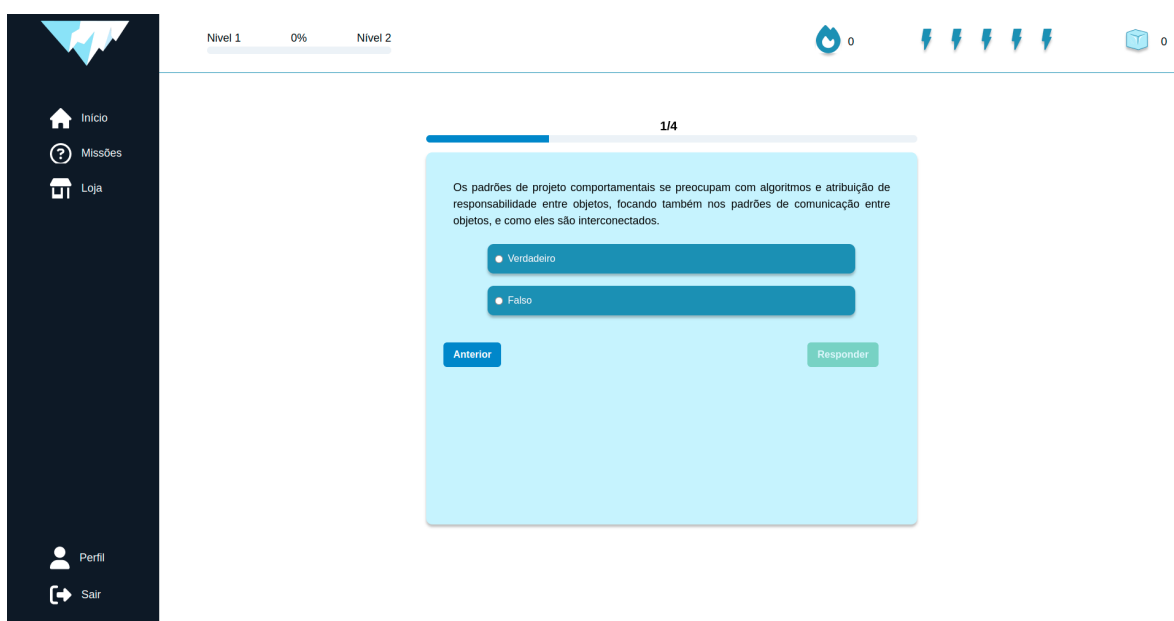


Figura 28 – Página de Conteúdos



Fonte: Autores

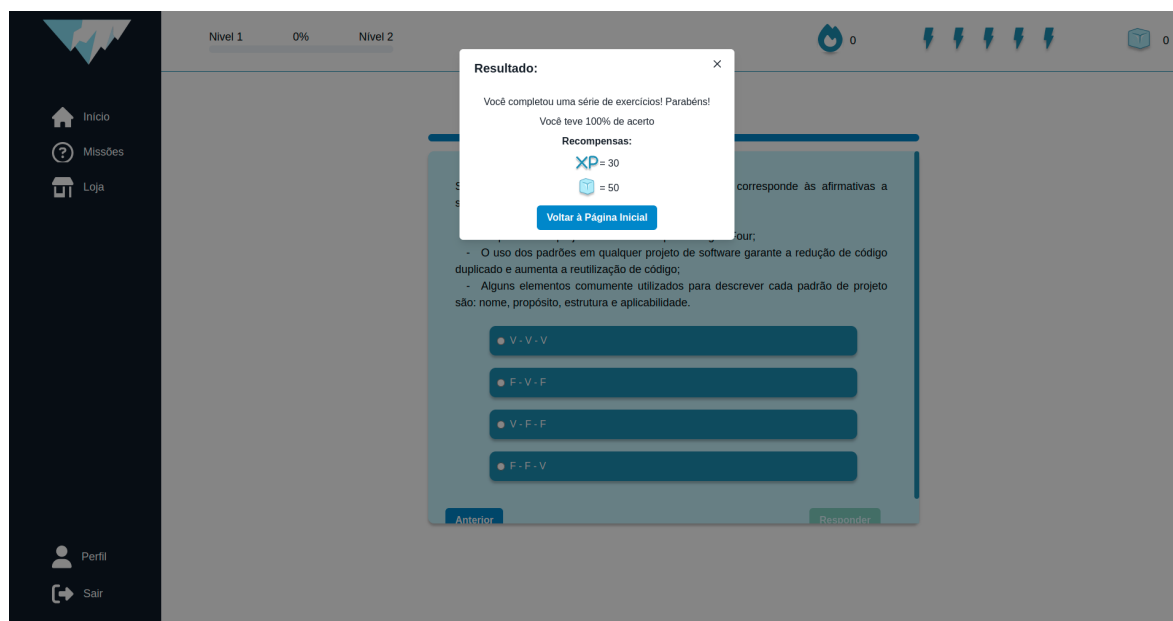
Figura 29 – Página de Exercícios



Fonte: Autores

Após o último exercício da sequência, o usuário recebe feedback sobre seu desempenho, bem como detalhamento sobre as recompensas que recebe ao concluir esses exercícios, como exemplificado na Figura 30.

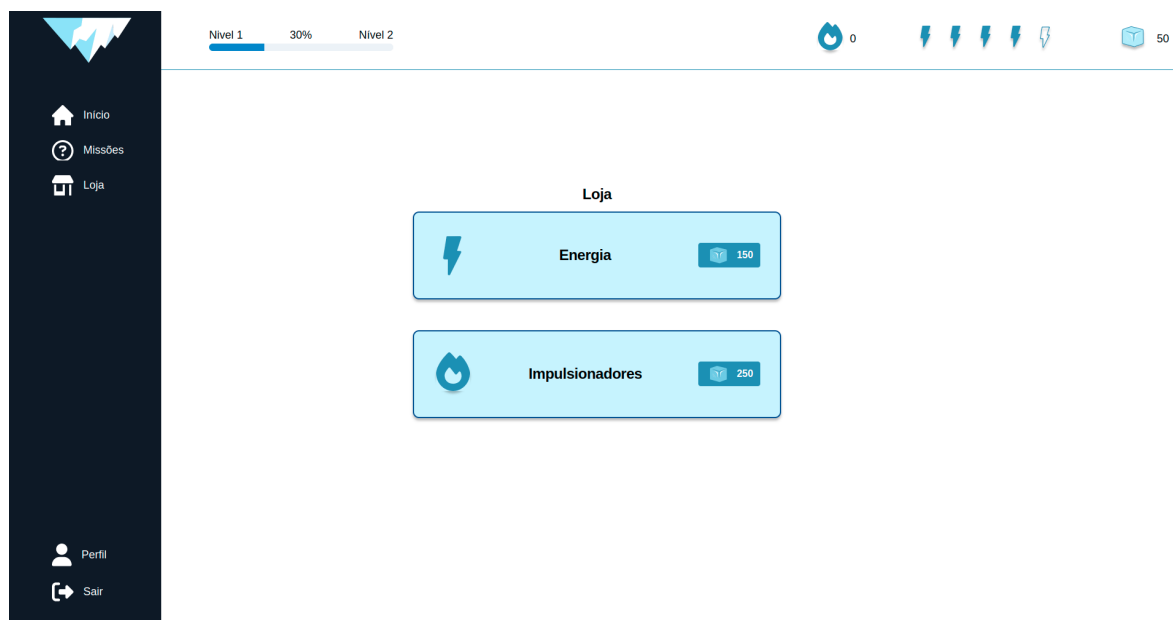
Figura 30 – Página de *Feedback* dos Exercícios



Fonte: Autores

A Figura 31 apresenta a página da loja do site, em que o usuário pode utilizar seus cubos de gelo para trocar por energias ou impulsionadores.

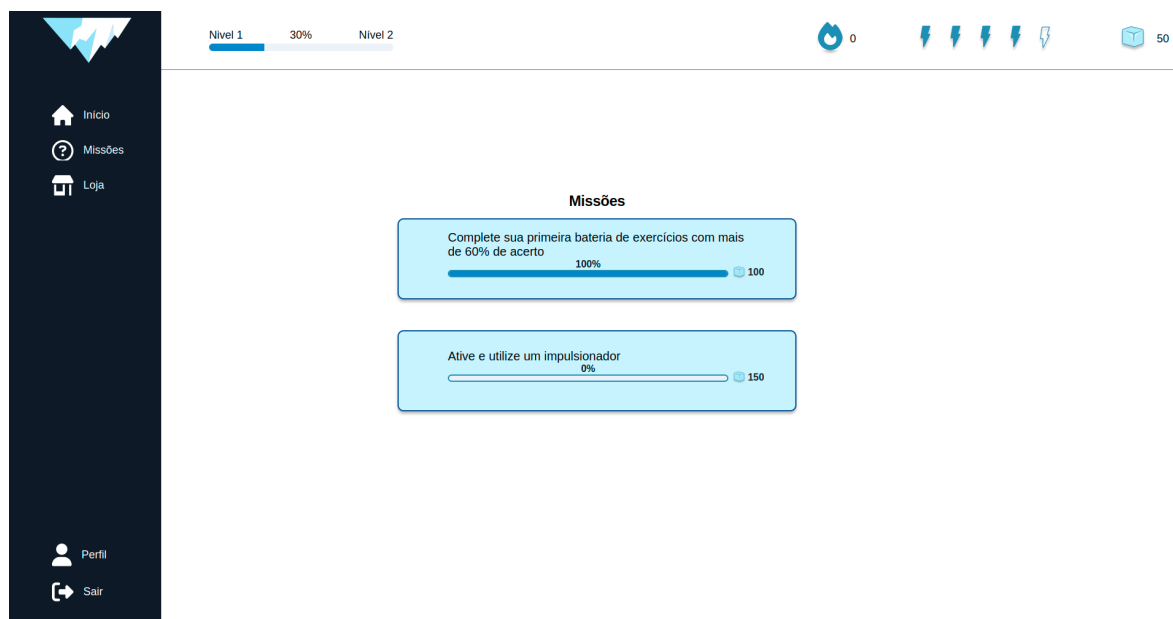
Figura 31 – Página da Loja



Fonte: Autores

Há também uma página de missões, como apresentado na Figura 32, em que o usuário pode visualizar seu progresso em cada missão.

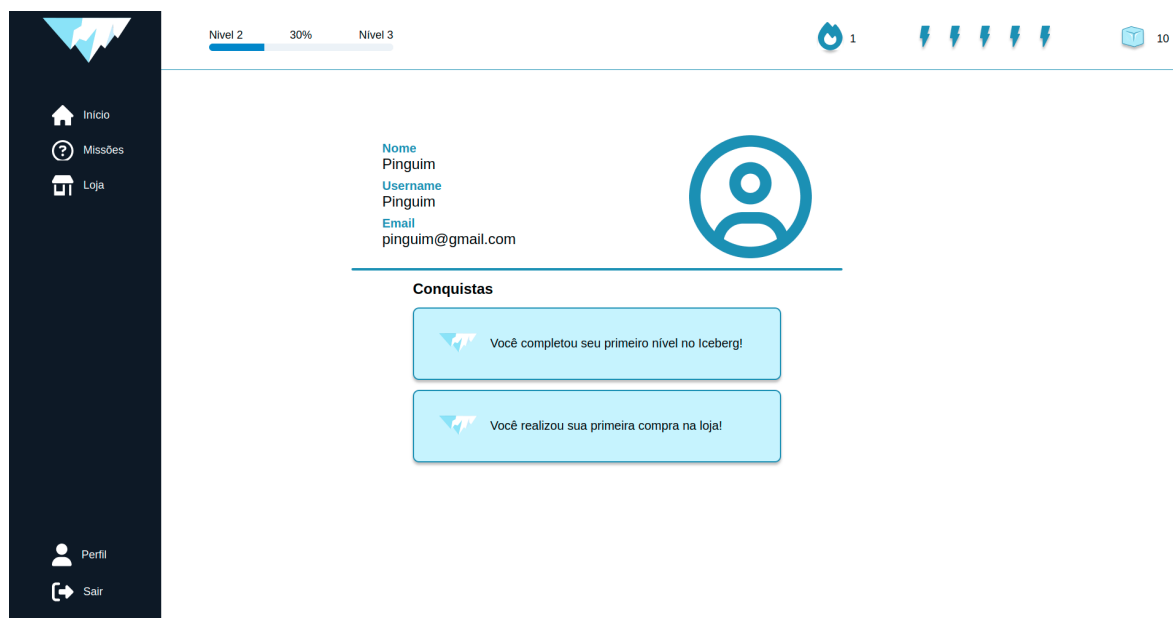
Figura 32 – Página das Missões



Fonte: Autores

A Figura 33 apresenta a página do perfil do usuário, que contém suas informações e conquistas recebidas, e pode ser acessada através do menu lateral.

Figura 33 – Página de Perfil do Usuário



Fonte: Autores

## 5.6 Resumo do Capítulo

Nesse capítulo, foram apresentados o Site educacional e a Abordagem gamificada que o orienta. Inicialmente foi conferida uma contextualização acerca da motivação e da escolha do tema, seguido de *Benchmarking*, *Brainstorming* e Personas, que fazem parte do processo de ideação da abordagem e do Site Educacional. No *benchmarking*, foram analisadas duas soluções existentes, sendo: uma com foco na gamificação durante o aprendizado, o Duolingo, e outra com foco no conteúdo de padrões de projeto, o *Refactoring Guru*. O *brainstorming* e a criação de personas também foram apresentados, explicitando melhor as ideias iniciais e o público-alvo da abordagem e do site.

Em seguida, foram detalhadas a abordagem, com o plano de gamificação que define os elementos que foram aplicados neste trabalho, com base no *framework Octalysis*. Também foi apresentado o Site Educacional, explicitando o foco dado a algumas heurísticas de Nielsen e outras práticas relacionadas à usabilidade, bem como a definição de questões de design da interface, além do protótipo de alta fidelidade que orientou o desenvolvimento da interface do site. Por fim, foram apresentadas os conteúdos e exercícios de padrão de projeto, além das telas e fluxos do Site Educacional desenvolvido, ao qual foi dado o nome de IcePatterns.

## 6 Análise de Resultados

Neste capítulo é apresentada a análise dos resultados obtidos ao longo dos ciclos de pesquisa-ação. Também são detalhadas as adaptações realizadas com base nessa análise feita, visando melhorar a abordagem e o Site Educacional em seus aspectos de conteúdo, gamificação e usabilidade, além do atendimento às expectativas do público-alvo. Portanto, o capítulo orienta-se pelas [Fases de Pesquisa-Ação](#), cumprindo com quatro ciclos de pesquisa-ação, sendo: [Primeiro Ciclo - Avaliação Por Especialista](#); [Segundo Ciclo - Avaliação por Entrevistas com Público-Alvo](#); [Terceiro Ciclo - Avaliação por Questionário com Público-Alvo](#) e [Quarto Ciclo - Autoavaliação](#). Ao final, tem-se o [Resumo do Capítulo](#). Ao final, tem-se o [Resumo do Capítulo](#).

### 6.1 Fases da Pesquisa-ação

Assim como explicado na seção [Método de Análise de Resultados](#) do Capítulo 4 - Metodologia, as fases dessa pesquisa-ação abrangem: definir o problema; pesquisar; desenvolver o plano de ação; implementar o plano de ação; coletar dados para avaliação, e avaliar o plano.

O primeiro problema a ser definido é o que remete às questões de pesquisa e desenvolvimento desse trabalho, explicitadas na seção [Questões de Pesquisa e Desenvolvimento](#). Tais problemas, de maior granularidade, são o foco da pesquisa, logo, devem ser analisados os resultados do plano implementado para resolvê-los, sendo o tal plano a abordagem e o Site Educacional em si.

Dessa forma, pode considerar os seguintes problemas: saber se é possível impactar positivamente a aprendizagem de padrões de projeto por meio de uma abordagem orientada à gamificação; saber quais são os elementos mais indicados para a aprendizagem de padrões de projeto e/ou conteúdos de complexidade similar; e como viabilizar a apresentação da abordagem de aprendizagem de padrões de projeto, orientada à gamificação, em um site educacional.

Nos ciclos apresentados nas seções a seguir, o problema definido anteriormente está intrinsecamente presente. Porém, além dele, são definidos para cada ciclo alguns outros problemas de menor granularidade. A resolução desses problemas contribui para a resolução do problema mais geral; por isso, são relevantes para este trabalho.

## 6.2 Primeiro Ciclo - Avaliação por Especialista

O primeiro ciclo teve como foco a parte de conteúdo da abordagem, ou seja, os conteúdos referentes a padrões de projeto de software, que seriam tratados no Site Educacional. Dessa forma, iniciando pela primeira fase, a definição do problema, têm-se um problema de menor granularidade: saber se os conteúdos e exercícios estão corretos e bem planejados.

Para esse primeiro ciclo, contou-se com um especialista na área, que pôde ser enquadrado no público-alvo mapeado nas personas na seção [Personas](#) do Capítulo 5, mais especificamente, na persona que representa um professor na área de software.

O plano de ação consistiu em planejar e detalhar um documento com os conteúdos e exercícios de padrões de projeto de software, estruturá-lo da melhor forma para ser lido, enviá-lo ao especialista para receber um *feedback* e uma avaliação sobre o conteúdo e exercícios propostos. Após isso, seriam realizadas melhorias no documento considerando o *feedback* recebido.

O plano foi então implementado: foi criado um documento com conteúdo e exercícios de padrões de projeto de software, utilizando como base o referencial teórico, especificamente a seção [Padrões de Projeto](#), além de pesquisas extras, com foco no site *Refactoring Guru*. Após a criação e revisão do documento pelos autores, o mesmo foi enviado ao especialista para avaliação. O documento foi enviado no formato “pdf”, via Slack, e o especialista adicionou comentários ao longo do documento com suas observações.

Após o recebimento da avaliação do documento, os autores realizaram correções e melhorias, desde melhorias ortográficas até correções de código e conteúdo. O documento já corrigido consta no [Apêndice B - Conteúdo e Exercícios no Site Educacional](#).

## 6.3 Segundo Ciclo - Avaliação por Entrevistas com Público-alvo

No segundo ciclo, a definição do problema foi tratada de forma um pouco mais abrangente, incluindo os tópicos tanto de conteúdo de padrões de projeto, quanto aspectos de gamificação e de usabilidade. Alguns problemas definidos para serem tratados nesse ciclo foram: “o conteúdo está compreensível e os exercícios estão balanceados em nível de dificuldade?”; “o site está intuitivo para uso, de acordo com as expectativas de um usuário?”; e “o usuário entende os elementos e mecânicas de gamificação incluídos na abordagem?”.

Inicialmente, foi feito um roteiro para as entrevistas, para manter o foco nos objetivos da entrevista, e para realizar os mesmos testes em cada uma, para que fosse cabível compará-las caso necessário. De posse dos resultados das entrevistas, os dados foram

agrupados, categorizados e analisados. Por fim, fizeram-se melhorias no site com base na análise feita.

As seções a seguir detalham o plano de entrevistas, seus resultados e análise dos mesmos. Por fim, são explicitadas as modificações e melhorias aplicadas.

### 6.3.1 Roteiro de Entrevista

A entrevista foi essencialmente dividida em três etapas: iniciando pela introdução e contextualização; seguindo para a etapa de execução de tarefas no site, e por fim a etapa de perguntas gerais e observações. Durante todo o processo, os entrevistadores fizeram anotações.

Na primeira etapa, os entrevistadores, que nesse caso são os próprios autores deste trabalho, contextualizaram os entrevistados sobre o tema e o estudo desse trabalho. Logo em seguida, foi explicado como funcionaria a entrevista, e quais seriam as etapas dela. Também foram passadas algumas instruções como, por exemplo, que seria interessante o entrevistado “pensar em voz alta”, para que os entrevistadores entendessem o raciocínio do usuário ao utilizar o site. Além disso, foi reforçado que, de maneira nenhuma, o entrevistado está sendo avaliado, e que a avaliação é voltada somente para a abordagem e para o site desenvolvidos.

Na segunda etapa, os entrevistadores passaram para o entrevistado tarefas a serem executadas. As tarefas foram passadas uma a uma, apenas quando a anterior já foi executada. As tarefas foram: cadastrar-se e realizar o *login*; verificar suas missões atuais; verificar suas informações de usuário; iniciar o aprendizado; completar alguma missão; repor suas energias; ler alguns conteúdos e fazer alguns exercícios.

A terceira etapa foi uma etapa de perguntas, em que os entrevistadores questionavam sobre alguns tópicos, podendo então surgir uma pequena conversa sobre esse tópico, dando sempre ênfase na opinião, sentimento e observações da pessoa entrevistada.

Ao final da entrevista, após os agradecimentos, foi passado um breve questionário para os entrevistados. Esse questionário refere-se ao *framework* SUS (*System Usability Scale*).

O *framework* SUS, criado por John Brooke, é um modelo simples de dez itens para conseguir uma visão geral de aspectos subjetivos de usabilidade (BROOKE, 1995). Os itens são afirmações que seguem uma escala Likert, em que as respostas variam de acordo com o nível de concordância da pessoa que está julgando as afirmações. Para a aplicação do SUS na presente análise de resultados, foi utilizada uma escala de 1 a 5, sendo 1 correspondente a “discordo totalmente” e 5 correspondente a “concordo totalmente”.

### 6.3.2 Resultados e Análise das Entrevistas

Foram entrevistadas três pessoas, sendo duas estudantes e uma profissional de desenvolvimento de software. Para melhor visualização dos resultados, pode-se dividir em entrevistado A, entrevistado B e entrevistado C.

Os entrevistados A e B informaram que já utilizaram aplicativos gamificados, como exemplo citaram o Duolingo, o aplicativo Habitica e um aplicativo que ajuda o usuário a lembrar de beber água. O entrevistado C informou que não lembra ter utilizado aplicativos gamificados anteriormente.

Na primeira etapa da entrevista, a etapa de execução de tarefas, todos os entrevistados finalizaram sem problemas a primeira tarefa, referente a registra-se no site e realizar o *login*. Porém, quando o entrevistado B executou essa primeira tarefa, percebeu-se que o *pop-up* que indica o sucesso de uma ação estava passando despercebido pelo usuário, no centro inferior da tela. Dessa forma, os entrevistadores anotaram que seria interessante pensar em modificar o local em que o *pop-up* aparece, de forma que fique em uma área que a visão do usuário repouse naturalmente e com mais frequência.

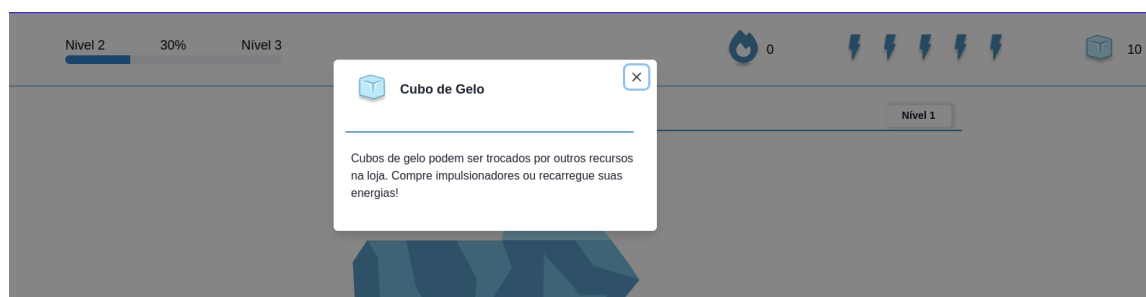
Ao acessar a página inicial, os três entrevistados reconheceram de início alguns elementos: a barra de energias, barra de progresso e a escala de níveis. Outros elementos como os cubos de gelo e o impulsionador não foram rapidamente identificados, apresentando então uma justificativa da existência de cartões com uma breve explicação sobre esses itens. Os entrevistados então clicaram nos itens para ler esses cartões explicativos, vide Figura 34 e Figura 35. De início, os entrevistados A e C não perceberam que o elemento era passível de ser clicado, então viu-se uma oportunidade de melhoria nesse aspecto.

Figura 34 – Itens da Barra Superior



Fonte: Autores

Figura 35 – Cartão Explicativo dos Itens



Fonte: Autores



Os entrevistados também realizaram com tranquilidade as tarefas 2 e 3, referentes a verificar as missões disponíveis e verificar seus dados de usuário.

Na tarefa 4, “iniciar o aprendizado”, os entrevistados A e B optaram por iniciar utilizando o botão maior, mostrado na Figura 36, e o entrevistado C iniciou pelo botão menor, mostrado na Figura 37. O entrevistado B ficou um pouco receoso ao clicar no botão maior, que possuía o texto “Nível” escrito, por isso os entrevistadores anotaram um ponto de melhoria para que o texto do botão fosse mais explicativo, trazendo a ideia de uma ação, como “Começar” ou “Vamos lá!”.

Figura 36 – Botão Maior de Nível



Fonte: Autores

Figura 37 – Botão Menor de Nível



Fonte: Autores

A tarefa 5, de “completar uma missão”, e a tarefa 6, de “repor suas energias”, foram tranquilamente realizadas por todos os entrevistados. Na tarefa 6, o entrevistado A foi diretamente para a Loja através do menu lateral, enquanto os entrevistados B e C primeiro clicaram no item de energia para abrir o cartão explicativo e ler, e então foram para a Loja através do menu lateral.

Por fim, a tarefa 7, referente a continuar lendo alguns conteúdos e realizando alguns exercícios, foi realizada com facilidade, visto que o entrevistado já havia acessado a parte de conteúdos e exercícios anteriormente. Passados alguns minutos com o entrevistado explorando essa parte, os entrevistadores encerraram a etapa da entrevista, pois todas as tarefas haviam sido executadas.

Na próxima etapa da entrevista, foram feitas perguntas aos entrevistados, que pontuaram alguns aspectos positivos e negativos, bem como fizeram observações sobre elementos do site. A partir disso, os entrevistadores fizeram anotações para possíveis melhorias da abordagem e do site, sendo algumas delas implementadas logo após o fim desse ciclo de avaliação, detalhadas melhor na próxima seção.

Quanto aos conteúdos e exercícios do site, o entrevistado A apontou que poderia haver algum outro recurso além do texto, como imagens, ou algum gif. Além disso, comentou que seria interessante receber algum *feedback* após responder os exercícios; caso não seja viável ser um comentário sobre cada exercício, poderia ser ao final do conjunto de exercícios, sugerindo que o usuário retorne aos conteúdos para estudar mais um determinado tópico indicado. Quanto ao código de exemplo apresentado nos conteúdos, o entrevistado gostaria de ter uma melhor visibilidade da indentação do código, para facilitar a leitura e chamar mais a atenção do usuário. O entrevistado B apontou que os exercícios estão bem de acordo com o que foi apresentado nos conteúdos, e que sentiu que os exercícios estão balanceados, permitindo que o usuário acerte pelo menos um para não desanimar. Porém, sentiu que seria interessante existir “pré-exercícios”, que seriam exercícios apenas de prática, em que não se perde nem ganha itens (energia, cubos, pontos de experiência). Por fim, o entrevistado C deu um *feedback* curto, apenas ressaltando que para ele o exemplo de código apresentado na parte de conteúdo foi fundamental para que ele lembrasse de como é aplicado o padrão de projeto. Também apontou que muitas das questões são de respostas mais “decoradas”, mas que achou bem interessante um exercício que tratava com mais contexto sobre o padrão de projeto, fazendo com que o usuário precisasse pensar mais antes de responder. Por fim, os entrevistados B e C identificaram alguns erros de ortografia a serem corrigidos.

Quanto ao recebimento de *feedbacks* do site ao realizar alguma atividade, o entrevistado A sugeriu que aparecesse um aviso quando o usuário completasse uma missão. Isso já ocorria no site, então concluiu-se que o entrevistado não percebeu o aviso que aparece no centro inferior da tela. Isso já foi comentado no início dessa seção como um ponto de melhoria, pois o mesmo aconteceu em outro momento com o entrevistado B. O entrevistado B apontou que sentiu falta de um aviso quando o desafio do nível for liberado para ser feito, além também de um aviso quando novas missões são desbloqueadas e aparecem na página de “Missões”, ao invés de apenas surgirem novas. O entrevistado C comentou que o que mais incomodou ele no sentido de receber *feedbacks* do site foi o carregamento das páginas, que estava “cru”, e não indicava se algo estava sendo carregado, ou se a página era somente o que estava sendo mostrado ali no momento.

Quanto à navegação, percebeu-se que no geral os entrevistados não utilizavam os botões menores de nível ao lado do iceberg (Figura 37) para voltar aos conteúdos do nível anterior, pois o botão maior (Figura 36) direciona o usuário apenas para o nível que ele

está atualmente. Assim, concluiu-se que a visibilidade e design dos botões não deixavam claros que era um elemento clicável. Considerando que é desejável que o usuário possa ler os conteúdos e praticar exercícios de níveis anteriores, é necessário haver uma maneira clara de acessar esses elementos, o que torna esse aspecto uma oportunidade de melhoria no site. Além disso, o entrevistado C apontou que seria melhor, ao finalizar os exercícios, retornar à página do nível, e não à página inicial, para ser possível prosseguir diretamente para o próximo subnível.

Essas observações concluíram a última etapa da entrevista. Ao final, o entrevistado B expressou o seu desejo de continuar utilizando o site, pois queria experimentar mais o impulsionador, queria ver a barra de progresso “saindo do 0%”, e porque via o ícone de cadeado ao lado dos níveis, o que despertava nele o desejo de desbloqueá-los.

De forma geral, os entrevistados executaram tarefas e entenderam o site e seus elementos com facilidade. Ao ver os entrevistados utilizando o site ao vivo, os próprios entrevistadores viram oportunidades de melhorias, como, por exemplo, a mensagem de explicação da barra de progresso estar equivocada e os botões do iceberg que devem ser mais visíveis.

Ao final das entrevistas, foi aplicado o *Framework* SUS através de um formulário do Google Forms. Cada entrevistado respondeu às afirmativas com um número de uma escala de 1 a 5, sendo 1 “discordo totalmente” e 5 “concordo totalmente”. A Tabela 4 resume os resultados obtidos.

Tabela 4 – Aplicação do *Framework* SUS

Afirmativas	Entrevistados		
	A	B	C
1. Eu acho que gostaria de usar esse sistema com frequência.	4	5	3
2. Eu acho o sistema desnecessariamente complexo.	1	1	1
3. Eu achei o sistema fácil de usar.	5	3	4
4. Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o sistema.	1	4	1
5. Eu acho que as várias funções do sistema estão muito bem integradas.	4	3	4
6. Eu acho que o sistema apresenta muita inconsistência.	1	1	3
7. Eu imagino que as pessoas aprenderão como usar esse sistema rapidamente.	5	2	5
8. Eu achei o sistema atrapalhado de usar.	2	3	1
9. Eu me senti confiante ao usar o sistema.	5	3	4
10. Eu precisei aprender várias coisas novas antes de conseguir usar o sistema.	3	2	1

Fonte: Autores

### 6.3.3 Melhorias aplicadas

De acordo com a análise apresentada na seção anterior, as seguintes melhorias foram aplicadas:

- Localização dos avisos na tela: os avisos de sucesso e erros apresentados ao usuário foram trocados de local na tela; anteriormente eram, apresentados no centro inferior,

e então passaram a aparecer no centro superior (vide Figura 38), no qual a visão do usuário repousa mais naturalmente e com maior frequência;

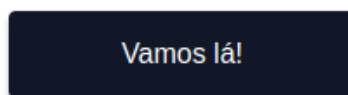
- Área clicável de elementos: os elementos da barra superior do site não eram facilmente percebidos como itens nos quais poderia-se clicar, para ler uma explicação sobre o mesmo, então foram feitas modificações para adicionar uma dica quando o cursor do mouse passa por cima do elemento; além disso, foi aumentada a área clicável dos elementos, principalmente da barra de progresso, para que fosse mais fácil clicá-los;
- Avisos ao liberar desafios e missões: agora são dados avisos ao usuário quando uma nova missão é desbloqueada para ser executada, e também quando o usuário completa o último subnível de um nível, assim desbloqueando o Desafio para fazer e subir de nível;
- Botões da página inicial: no botão maior da página inicial, o texto foi alterado para “Vamos lá!”, para ser um termo que chamasse o usuário em uma forma de ação (Figura 39); os botões menores de níveis, ao lado direito do iceberg, também foram modificados, para serem mais percebíveis como elementos clicáveis (Figura 40);
- Redirecionamento: ao finalizar a leitura de um conteúdo, o usuário é redirecionado para a página do nível, ao invés da página inicial; o mesmo acontece ao fechar um exercício sem completá-lo;
- Carregamento de página e imagem: foi adicionado um carregamento apropriado para as páginas e elementos do site, para que o mesmo fique mais polido, e para que o usuário fique ciente do que está acontecendo enquanto aguarda algum carregamento;
- Correções ortográficas: foram feitas algumas simples correções ortográficas.
- Modificação na apresentação de códigos: os códigos de exemplo foram alterados para terem uma melhor visualização, com foco em uma indentação que facilite a leitura (vide Figura 41);

Figura 38 – Aviso no Centro Superior



Fonte: Autores

Figura 39 – Botão Maior Modificado



Fonte: Autores

Figura 40 – Botão Menor Modificado



Fonte: Autores

Figura 41 – Apresentação de Código Modificada



Fonte: Autores

## 6.4 Terceiro Ciclo - Avaliação por Questionário com Público-alvo

No terceiro ciclo, a definição do problema girou entorno das expectativas do público-alvo e do entendimento do público-alvo sobre padrões de projeto e a apresentação deles no site.

O questionário foi criado utilizando o Google Forms, e foi estruturado em seções, como apresentado na seção [Estrutura do Questionário](#). Após sua criação, o formulário foi divulgado de forma *online*, contextualizando brevemente sobre a pesquisa e convidando participantes a responderem o questionário.

A seção [Resultados e Análise do Questionário](#) apresenta os dados coletados com o questionário e uma análise feita sobre esses dados.

### 6.4.1 Estrutura do Questionário

As perguntas iniciais do questionário são direcionadas para entender melhor a relação da pessoa com a área de software. Passando por perguntas básicas, como idade e escolaridade, chegou-se em perguntas mais específicas, começando com o esclarecimento se a pessoa está de alguma forma inserida na área de desenvolvimento de software, o que é relevante para que ela siga respondendo o questionário.

Em seguida, foi elaborada uma seção do questionário apenas para entender os conhecimentos específicos sobre programação dos participantes. Já na seção seguinte, buscou-se entender o nível de conhecimento e a relação da pessoa com padrões de projeto de software, com perguntas como, por exemplo: “Como você classifica a abrangência do

seu conhecimento em padrões de projeto de software?"; "Onde você ouviu falar/aprendeu sobre padrões de projeto de software?"; "Quais padrões a seguir você lembra do nome, conceito e aplicação?"; "Você costuma conscientemente aplicar padrões de projeto nos projetos que desenvolve?".

Por fim, na última seção, pediu-se que as pessoas visitassem o site educacional e navegassem por ele. Depois disso, ao retornar para o formulário, a pessoa responderia questões relacionadas ao site, com suas opiniões, percepções e observações. As primeiras perguntas estavam no formato de escala Likert, e as outras perguntas eram abertas, ou seja, receberiam respostas por escrito.

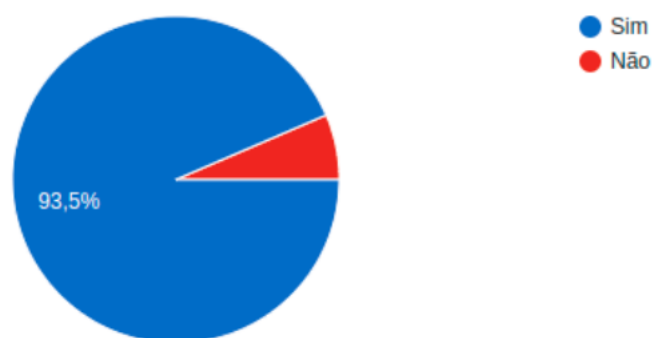
### 6.4.2 Resultados e Análise do Questionário

Foram obtidas 31 respostas no formulário. Desses 31 participantes, 29 estão inseridos na área de software e se consideram estudantes na área de desenvolvimento de software, como mostra a Figura 42 e a Figura 43. Desses 29 estudantes, 19 já ouviram falar ou conhecem padrões de projeto de software, como mostra o gráfico da Figura 44. Esse critério foi um filtro para seguir respondendo o questionário, então o restante das perguntas englobam as respostas desses 19 participantes.

Figura 42 – Respostas do Questionário - 1

**Você está inserido ou possui algum conhecimento na área de desenvolvimento de software?**

31 respostas

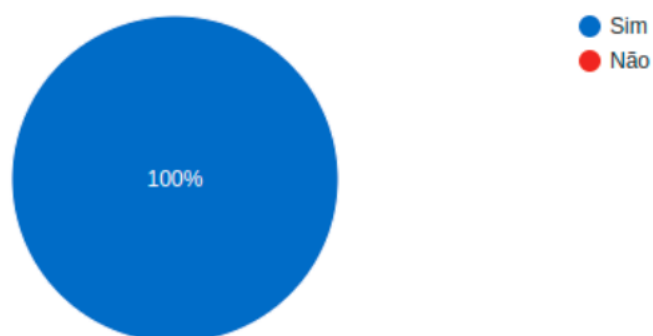


Fonte: Autores

Figura 43 – Respostas do Questionário - 2

**Você se considera um estudante na área de desenvolvimento de software?**

29 respostas

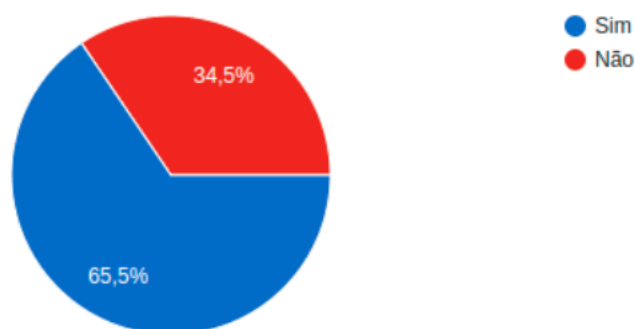


Fonte: Autores

Figura 44 – Respostas do Questionário - 3

**Você já ouviu falar ou conhece padrões de projeto de software?**

29 respostas



Fonte: Autores

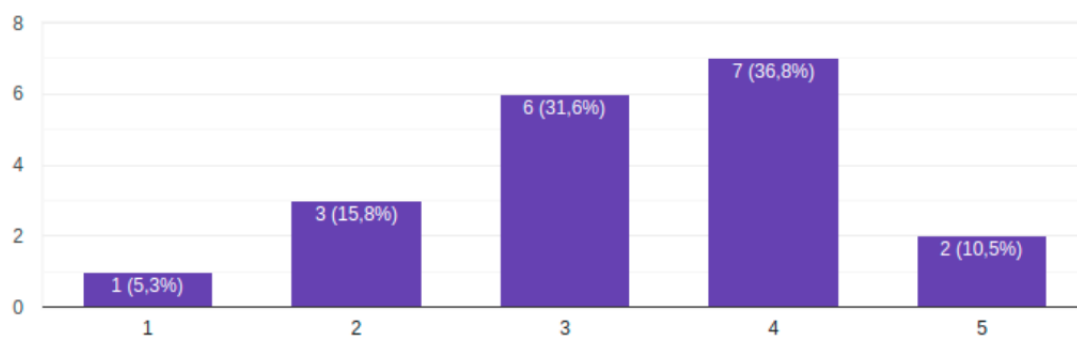
Na seção seguinte, buscou-se entender melhor a relação do participante com padrões de projeto, perguntando sobre a abrangência do conhecimento dele em padrões de projeto (Figura 45); por onde ele aprendeu sobre esse conteúdo (Figura 46); quais padrões de projeto ele conhece e lembra (Figura 47); e se ele aplica padrões nos projetos que desenvolve (Figura 48). Os dados apresentados nessas figuras serão analisados ao longo dessa seção.



Figura 45 – Respostas do Questionário - 4

Como você classifica a abrangência do seu conhecimento em padrões de projeto de software?

19 respostas

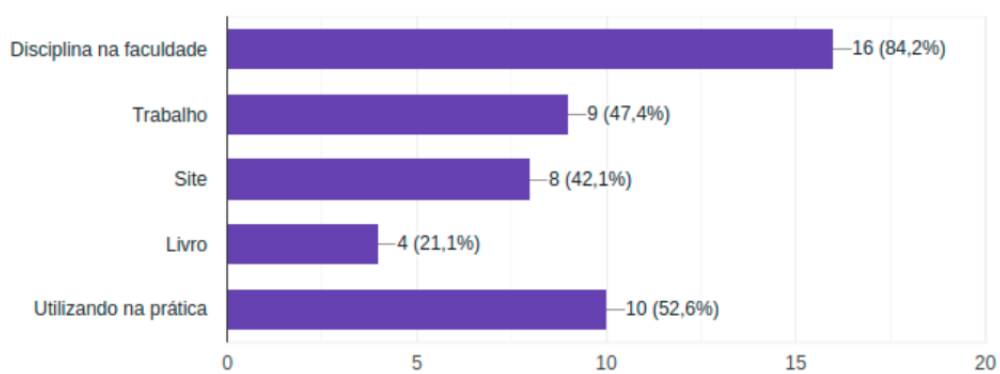


Fonte: Autores

Figura 46 – Respostas do Questionário - 5

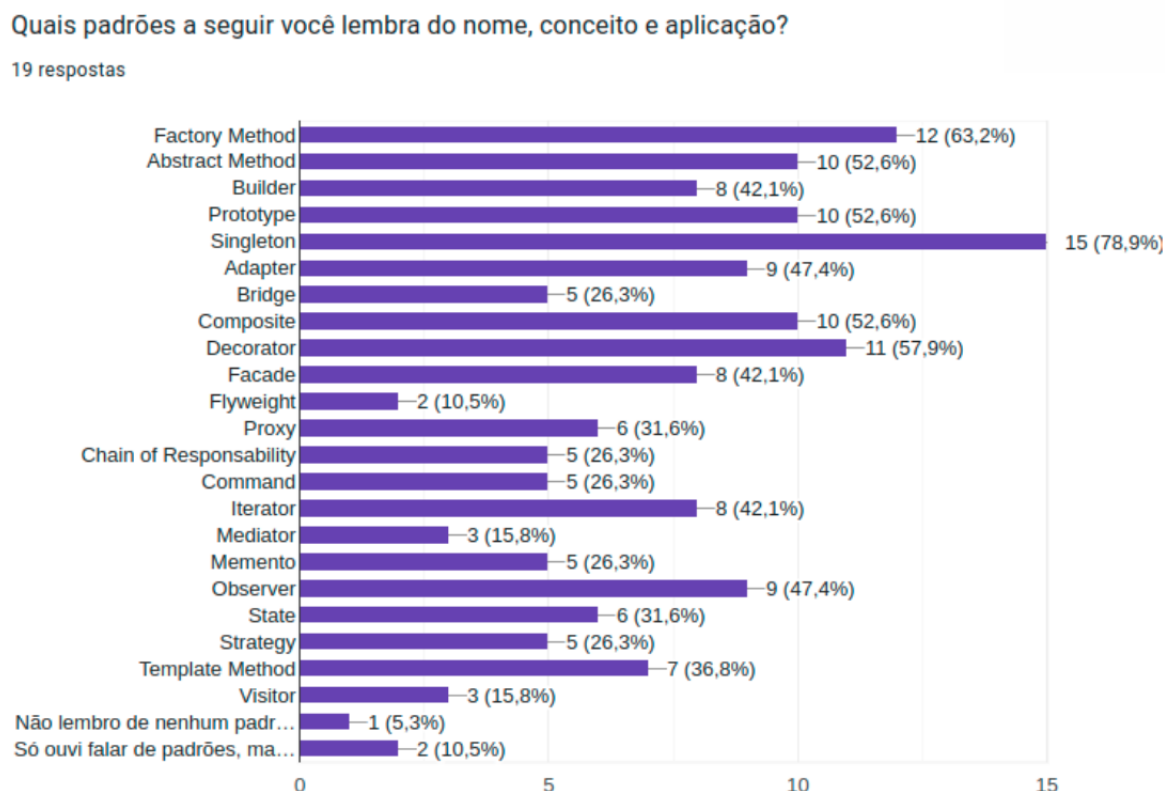
Onde você ouviu falar/aprendeu sobre padrões de projeto de software?

19 respostas



Fonte: Autores

Figura 47 – Respostas do Questionário - 6



Fonte: Autores

Na última seção, o objetivo foi receber *feedbacks* sobre o site e a abordagem em si, para realizar análises que validem o que foi desenvolvido e que apresentem oportunidades de melhoria. Foi indicado, então, que os participantes acessassem o link disponibilizado e navegassem pelo site utilizando um computador. Então, foram feitas algumas perguntas a serem respondidas com uma escala Likert, de 1 a 5, como apresentado na Figura 49, Figura 50 e Figura 51. Essas respostas são analisadas ao longo desta seção.

Por fim, foram feitas perguntas de resposta aberta aos participantes, buscando entender quais eram as expectativas deles quanto ao site, e quais recursos e elementos eles acreditam ser interessante estar presentes no site e na abordagem. Esses resultados estão apresentados na Figura 52, Figura 53 e Figura 54. As respostas recebidas e apresentadas nessas figuras serão destacadas em análises a seguir.

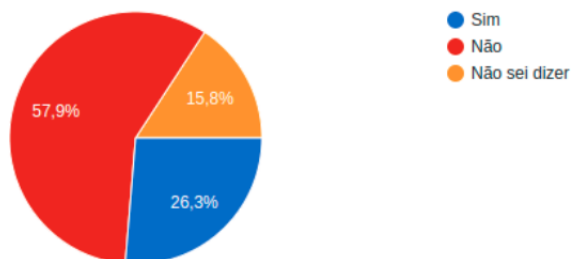
Com base nos resultados apresentados acima, os autores fizeram algumas análises. Essas análises não são verdades absolutas, pois se baseiam nessa pesquisa específica feita com esses participantes. Entretanto, as análises podem trazer uma visão real, mesmo que

não seja exata e precisa, sobre a abordagem e o site, e como os usuários os recebem e utilizam.

Figura 48 – Respostas do Questionário - 7

Você costuma conscientemente aplicar padrões de projeto nos projetos que desenvolve?

19 respostas



Caso tenha aplicado, cite algum padrão (ou alguns padrões) que utilizou

4 respostas

Adapter e Factory Method

Os que mais aplico no dia-a-dia são Strategy, Iterator, Mediator, Chain of Responsibility, Facades, Builders, Observers e Decorators

SOLID

Principalmente Singleton, Observer, Strategy

Fonte: Autores

A primeira análise foi a percepção de que os 2 padrões de projeto que os participantes mais lembram, como mostrado anteriormente na Figura 47, são padrões que no site são apresentados logo nos primeiros níveis do iceberg. O Singleton é reconhecido por 78,9% dos participantes e o Factory Method por 63,2% deles. Os outros padrões dispostos nos níveis iniciais do iceberg no site, o Facade, Template Method e o Adapter, também são relativamente bem reconhecidos e lembrados, com respectivamente 42,1%, 36,8% e 47,4%. A Tabela 5 resume esses dados. Dessa forma, entende-se que a apresentação de tais padrões nos níveis iniciais do site pode ser um ponto positivo, pois, visto que muitos já lembram desses padrões, seriam “fáceis” e apenas uma forma de relembrá-los. Então, ao avançar na trilha de aprendizado do iceberg, se deparariam com os padrões mais “difíceis”, que são os que não são lembrados. Uma adaptação futura, porém, pode ser personalizar essa trilha de aprendizado do usuário, desde que o mesmo informe previamente se já possui algum conhecimento sobre os conteúdos que serão abordados.

Tabela 5 – Relação entre Padrões Iniciais do Iceberg e Reconhecimento deles pelos Participantes

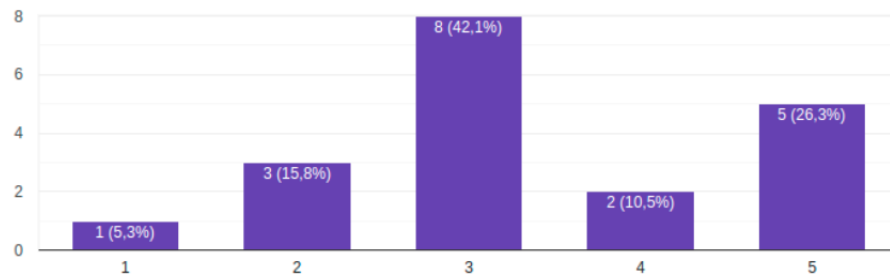
<b>Padrões iniciais do iceberg</b>	<b>% dos participantes</b>
Singleton	78.9%
Facade	42.1%
Template Method	36.8%
Factory Method	63.2%
Adapter	47.4%

Fonte: Autores

Figura 49 – Respostas do Questionário - 8

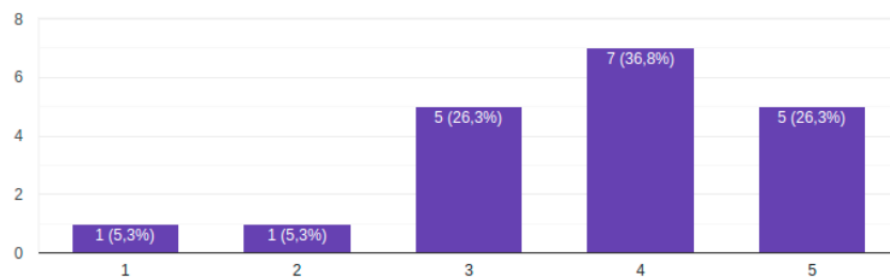
Como você avalia a profundidade dos conteúdos apresentados no site?

19 respostas



Como você avalia o balanceamento de questões mais simples e mais elaboradas e com contexto?

19 respostas



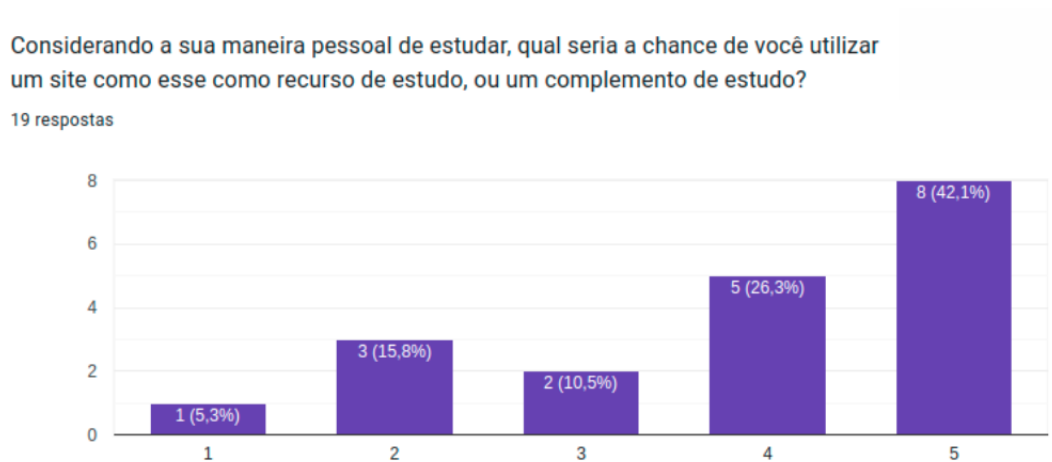
Fonte: Autores

Figura 50 – Respostas do Questionário - 9



Fonte: Autores

Figura 51 – Respostas do Questionário - 10



Fonte: Autores

Figura 52 – Respostas do Questionário - 11

Você encontrou recursos semelhantes a outros sites ou aplicativos que tenha utilizado, desde aspectos visuais até mecânicas do site? Se sim, cite alguns.

19 respostas

Não
Não
No momento não, pois eu nunca tinha visto um site do qual está ali para ensinar os conteúdos de padrões de projeto e implementando também a gamificação.
Sim, barras de xp, energia, "dinheiro" e loja
Me lembrou um pouco do Duolingo
Sim, mas não me recordo
A forma como os conteúdos são apresentados me lembrou a utilizada no TOP (The Odin Project)
refactoring.guru para o conteúdo, para a gamificação do conteúdo de aprendizado exercism.io para o tipo de questionário de pergunta e resposta talvez um pluralsight ou udemy
Alura
<pre>{"https://www.tutorialspoint.com/design_pattern/design_pattern_online_quiz.htm", "https://www.w3schools.blog/design-patterns-online-quiz-test", "https://www.vincehuston.org/dp/patterns_quiz.html"}</pre>
Energias - Parece com as vidas no Duolingo
Sim. ideia de vidas/energia semelhante a jogos, e ideia de progresso associado a conhecido como aplicativos educacionais como o Duolingo
Sim, barra de energia, loja, missões e impulsionador

Fonte: Autores

Figura 53 – Respostas do Questionário - 12

Você acredita que possa haver algum outro recurso para estudo no site? Se sim, quais?

19 respostas

Não
Não
Muito mais padrões de projeto e exemplos de implementação na próprio texto onde explica sobre o padrão
Assistente "virtual", vídeo "shorts", animações, apresentação de implementações reais
Poderia não precisar gastar energia ao ler sobre um conteúdo, para não limitar o seu uso.
O site apresenta bem o conteúdo. Em questão de apresentação, mostrar diagramas da implementação seria interessante.
Conceitos básicos de programação
Sim, indicações de bibliografias relacionadas ao conteúdo apresentado.
Diagramas UML dos padrões.
Explicar o que foi errado para orientar o aprendizado.
Acesso a bibliografias referentes ao assunto
Poder praticar antes de fazer os exercícios
Não sei
Acredito que não

Fonte: Autores

Figura 54 – Respostas do Questionário - 13

Além de recursos, há alguma outra melhoria que você acredita ser interessante para melhorar a experiência e estudo do usuário?

19 respostas

Não
acho que no momento, eu acharia bacana quando você se registrar, o site já me redirecionar para a tela inicial sem precisar me logar manualmente. Além disso, é por ter alguns passo a passo para apresentar melhor a plataforma para o usuário que caiu ali e não sabe como que mexe. Boa sorte e desejo sucesso!
Mais itens na loja
Não descontar energia vendo conteúdos teóricos, já que o usuário pode tentar rever o conteúdo posteriormente para estudo
Atualização constante
Acredito que apresentar mais exemplos de como utilizar, com os diagramas mostrando o padrão de projeto seria interessante. Um conteúdo interessante é mostrar como se pode juntar certos padrões de projetos. O conteúdo é apresentado em uma caixa um pouco pequena, poderia fazer melhor o uso do espaço em branco da tela.
Estudo mais objetivos. "Crie seu primeiro site" ou "Crie seu primeiro APP" ou "C"
Não.
Mostrar as respostas corretas/erradas no final do quiz e Ao finalizar uma sessão de quiz, retorne para a página anterior, no nível, em vez da home.
Está muito bom.
Feedback dos acertos e erros, diz a porcentagem mas não mostra o feedback dos erros
Acredito que não
No momento não
Loadings e polimento

Fonte: Autores

Outra observação feita pelos autores é em relação aos meios utilizados para o aprendizado de padrões de projeto, em que o participante pode selecionar 1 ou mais opções. A opção “site” foi selecionada 8 vezes, representando 42,1%, como mostrado na Figura 46. Dessas 8 vezes, em 7 delas ocorreu a seleção de “site” juntamente com os outros meios, como apresentado na Tabela 6. Assim, percebe-se que, de fato, um site para estudos pode ser utilizado em conjunto com outros recursos.



Tabela 6 – Comparação de Outros Meios em Relação ao Uso de Sites

Meios usados junto a sites	Vezes utilizado junto a sites	%
Disciplina na faculdade	7	43.7%
Trabalho	3	33.3%
Livro	1	25%
Utilizando na prática	5	50%

Fonte: Autores

Isso também é tratado no questionário quando é perguntado ao participante “Considerando a sua maneira pessoal de estudar, qual seria a chance de você utilizar um site como esse como recurso de estudo, ou um complemento de estudo?”, ao qual ele responde com números de uma escala de 1 a 5, com 1 indicando uma chance muito baixa e 5 indicando uma chance muito alta. Os resultados a essa pergunta estão retratados na Figura 51. De forma geral, as respostas obtidas foram positivas quanto a isso, com 42,1% dos participantes respondendo chance 5 e 26,3% respondendo chance 4, o que já resulta em 68,4% de respostas com alta chance de uso do site.

As análises feitas sobre as perguntas de resposta aberta foram bem objetivas, com base no que os participantes escreveram. As respostas apresentadas na Figura 52 tratam das semelhanças da abordagem e do Site Educacional desenvolvido comparados a outros sites e aplicativos existentes. Percebe-se que alguns participantes assimilaram os elementos e mecânicas, como barras de energia, barra de progresso, “moeda” do jogo, presença de uma loja e de missões. Outros participantes responderam que não reconheceram recursos semelhantes. Utilizando a ferramenta de questionário, não foi possível averiguar se esses participantes conseguiram, mesmo sem reconhecer elementos, entendê-los no próprio site desenvolvido.

A segunda e a terceira perguntas abertas abordaram essencialmente *feedback* e sugestões dos participantes, quanto a recursos de estudo e a melhorias gerais para melhorar a experiência do usuário (Figura 53 e Figura 54). Essas sugestões foram avaliadas pelos autores, que optaram por concordar em atender algumas e anotar outras para possíveis trabalhos futuros.

### 6.4.3 Melhorias Aplicadas

De acordo com a análise apresentada na seção anterior, as seguintes melhorias foram aplicadas:

- Ao se registrar na aplicação, agora o usuário é automaticamente logado, sem necessidade de fazer o login logo após o cadastro, e

- Foram adicionados alguns diagramas UML junto ao conteúdo de cada padrão de projeto;

Algumas melhorias que podem ser feitas em possíveis trabalhos futuros:

- Adicionar mais padrões de projeto e mais exercícios;
- Encontrar uma forma de orientar o aprendiz ao errar exercícios, porém sem afetar a mecânica da gamificação e de poder realizar mais tentativas em um mesmo exercício;
- Adicionar mais recursos para estudos, direcionando a diferentes bibliografias também, e
- Implementar um passo a passo para quando o usuário acessar o site pela primeira vez;

## 6.5 Quarto Ciclo - Autoavaliação

No quarto ciclo, a definição do problema teve como base a gamificação. Para isso, foi feita uma pesquisa sobre de qual forma seria possível avaliar o site levando em consideração seu aspecto de gamificação. Em estudos, entendeu-se que a maioria das heurísticas de avaliação de jogos podem ser aplicadas para avaliação de aplicativos gamificados (SOUTO; R., 2015). Para a autoavaliação, foi selecionado o método de avaliação utilizado por Souto e Souza (2015), em que definiram 28 heurísticas, utilizadas pelos autores para avaliar a abordagem e site educacional. O Quadro 4 apresenta as heurísticas utilizadas para realizar a autoavaliação.

### 6.5.1 Autoavaliação Heurística

São apresentados os resultados da autoavaliação e, ao final, são detalhadas algumas análises e percepções sobre os resultados. A definição de cada heurística seguiu o método de Souto e Souza (2015). Cada heurística foi classificada como “satisfeita”, “parcialmente satisfeita”, “não satisfeita” ou “não se aplica”.

#### 6.5.1.1 HA1

Os jogadores não precisam ter um manual, embora ele deva existir. Toda a informação necessária deve estar inclusa no próprio jogo. A ajuda deve ser dada durante o jogo, de maneira que o usuário não fique preso ou tenha que utilizar um manual. Para tanto, pequenos itens de ajuda devem ser oferecidos no decorrer do jogo como, por exemplo,

personagens que aparecem a medida que o jogo evolui, com dicas, informações e próximos passos.

Heurística satisfeita: Parcialmente

Comentário: o site possui instruções sobre seus elementos e sobre o que pode ser feito. Entretanto, alguns usuários não precisarão de manual, pois reconhecem os diversos elementos de gamificação e mecânicas de jogos implementados; entretanto, outros usuários podem ficar em dúvida e precisar usar o manual;

#### 6.5.1.2 HA2

O jogador deve poder facilmente desligar ou ligar o jogo, visualizar opções, obter ajuda, salvar e pausar em diferentes estágios. As informações vitais para o funcionamento do jogo devem ser fáceis de ser encontradas.

Heurística satisfeita: Parcialmente

Comentário: no geral, há grande facilidade nisso, entretanto, ao precisar sair do site no meio de um conteúdo ou de um exercícios, esse progresso não será salvo, ou seja, quando entrar novamente, o usuário precisará passar por páginas anteriores de conteúdo, e iniciar os exercícios desde o início.

#### 6.5.1.3 HA3

O jogo deve ser projetado de modo a prevenir erros antes de eles acontecerem. A prevenção de erros pode incluir mensagens de aviso e alerta como “Tem certeza que deseja sair?” ou “Você quer salvar o jogo antes de sair?”.

Heurística satisfeita: Não

Comentário: o site não possui muitos fluxos em que o usuário possa cometer erros; entretanto, ao sair de um conteúdo ou exercícios, ele precisa recomeçar a jornada, e isso não é alertado antes ao usuário.

#### 6.5.1.4 HA4

O jogador deve poder ver o progresso do jogo e comparar os resultados.

Heurística satisfeita: Sim

Comentário: é possível visualizar o progresso de diferentes maneiras. O progresso na trilha de aprendizado pode ser visualizado pela barra de progresso, pelos níveis do iceberg que estão desbloqueados na página inicial, e pelos subníveis que estão desbloqueados na página de cada nível. Além disso, há o progresso de leitura de conteúdos e de exercícios, que indica em que estágio o usuário está.

#### 6.5.1.5 HA5

O jogo deve assegurar que o jogador não tenha que recomeçar a partir de cada erro. Deve ser possível gravá-lo em diferentes estágios e carregá-lo novamente quando solicitado ou permitir que o usuário retorne à última ação correta.

Heurística satisfeita: Não

Comentário: Quando o usuário quer sair de um exercício que está pela metade, ao voltar, perde-se todo o progresso.

#### 6.5.1.6 HA6

A interface do jogo deve ser consistente quanto à navegação, design e diálogos. Os padrões da indústria, para controlar as funcionalidades, devem ser utilizados a fim de permitir o fácil acesso. A interface deve ser o menos intrusiva possível ao jogador. O menu deve ser experimentado como parte do jogo - menu com “look & feel” semelhante.

Heurística satisfeita: Sim

Comentário: os ícones são padronizados, a tipografia e cores seguem um design consistente.

#### 6.5.1.7 HA7

O jogo deve fornecer múltiplas maneiras de se realizar uma ação, assegurando que o jogador escolha a que mais gostar. Sua interface deve ser suficientemente simples para que o iniciante aprenda a usar os controles básicos rapidamente e, ao mesmo tempo, deve ser expansível para que um usuário expert possa usar atalhos a fim de melhorar seu desempenho.

Heurística satisfeita: Sim

Comentário: um exemplo disso são as opções de botões que levam o usuário ao aprendizado.

#### 6.5.1.8 HA8

O jogo não deve apresentar tarefas repetitivas ou entediadas. Deve oferecer várias opções de caminhos, garantindo o sentido de liberdade, tornando a experiência única e permitindo que cada um faça a sua história.

Heurística satisfeita: Parcialmente

Comentário: os conteúdos e exercícios do site não são repetitivos, porém há necessidade de missões e conquistas mais variadas para que o usuário não se sinta entediado

em algum momento. Além disso, é necessário a adição de mais conquistas únicas, ou seja, que não serão necessariamente recebidas por todos os jogadores.

#### 6.5.1.9 HA9

O jogo deve fornecer *feedback* imediato para as ações realizadas. A cada comando deve existir uma resposta do sistema. É preciso existir *feedback* constante a respeito do progresso do jogador para que ele possa identificar sua pontuação e seu status.

Heurística satisfeita: Sim

Comentário: o usuário é constantemente informado quando recebe conquistas, quando recebe recompensas, quando desbloqueia ou completa missões. Também é informado ao completar exercícios e ao subir de níveis.

#### 6.5.1.10 HA10

Seguir as tendências da comunidade de jogos para diminuir a curva de aprendizado. Isso inclui controles, interface e a linguagem utilizada no jogo.

Heurística satisfeita: Sim

Comentário: o site segue padrões de aplicações gamificadas, com barras de progresso, de energia, lojas, missões, reconhecidas por diversos usuários.

#### 6.5.1.11 HA11

O jogo deve ter objetivos claros (ou suportar objetivos criados pelos jogadores) cuja realização é incerta.

Heurística satisfeita: Sim

Comentário: o objetivo do site é claro: completar os níveis e aprender padrões de projeto.

#### 6.5.1.12 HA12

O jogador deve obter resultados de maneira justa.

Heurística satisfeita: Sim

Comentário: o site não possui mecanismos que permitam que o usuário incorra em não obediência às regras.

#### 6.5.1.13 HA13

O jogador deve ser recompensado e suas recompensas devem ser significativas. Durante o período de aprendizado é importante que o jogador deva ser recompensado por qualquer tipo de realização, para que a primeira experiência com o jogo seja encorajadora.

Heurística satisfeita: Sim

Comentário: o usuário recebe recompensas ao realizar exercícios e completar níveis, e essas recompensas permitem que ele prossiga estudando e que compre itens da loja.

#### 6.5.1.14 HA14

O jogo deve ter desafios identificáveis.

Heurística satisfeita: Sim

Comentário: o usuário deve escalar os níveis e subníveis de aprendizado, realizar desafios de exercícios e completar missões para alcançar seus objetivos de aprendizado.

#### 6.5.1.15 HA15

O jogo deve oferecer diferentes níveis de dificuldade. A dificuldade pode ser determinada automaticamente de acordo com o desempenho do jogador, escolhida pelo jogador e ainda determinada pela habilidade do oponente.

Heurística satisfeita: Parcialmente

Comentário: o site possui níveis que o usuário deve completar. Entretanto, a variação de dificuldade entre eles pode ser potencializada com uma trilha de aprendizado que seja personalizada ao usuário.

#### 6.5.1.16 HA16

O jogo deve ter múltiplas maneiras de se ganhar.

Heurística satisfeita: Não se aplica

Comentário: O ganho é intrínseco, pois é a aprendizagem de padrões de projeto.

#### 6.5.1.17 HA17

O jogo deve ter uma fantasia, ou seja, evocar imagens e objetos de situações fictícias. A fantasia deve ser envolvente e consistente para eliminar a descrença. O jogo deve ser original e surpreendente, mas não completamente incompreensível. O design deve ser passível de ser reconhecido pelo jogador e se relacionar com a sua função.

Heurística satisfeita: Não se aplica

Comentário: O site não possui fantasia nem narrativa.

#### 6.5.1.18 HA18

Os jogos que envolvam histórias e personagens devem suscitar o interesse pelo que representam.

Heurística satisfeita: Não se aplica

Comentário: O site não possui história nem personagens.

#### 6.5.1.19 HA19

O jogo deve oferecer efeitos sonoros interessantes e um visual atraente para envolver o jogador no ambiente.

Heurística satisfeita: Parcialmente

Comentário: o site não possui efeitos sonoros, mas o visual é consistente e não é poluído.

#### 6.5.1.20 HA20

O jogo deve ter novidades, surpresas e violação das expectativas de maneira a estimular ações e reações do jogador.

Heurística satisfeita: Sim

Comentário: o site envia conquistas aleatórias ao usuário, desbloqueia novas missões ao subir de subníveis.

#### 6.5.1.21 HA21

O jogo deve suportar comunicação. Os jogadores devem ter conhecimento de outros jogadores e serem capazes de interagir entre si.

Heurística satisfeita: Não

Comentário: o site não abrange comunicação e interações entre usuários.

#### 6.5.1.22 HA22

O jogo deve oferecer razões para comunicação entre os jogadores, seja para discutir tática ou obter colaboração.

Heurística satisfeita: Não se aplica

Comentário: o site não abrange comunicação e interações entre usuários.

#### 6.5.1.23 HA23

O jogo deve suportar grupos ou comunidades. Jogadores que sentem que fazem parte de uma comunidade têm maior chance de continuar participando do jogo.

Heurística satisfeita: Não

Comentário: o site não suporta grupos e comunidades.

#### 6.5.1.24 HA24

O jogo deve incluir suporte para que jogadores possam encontrar outros. Os jogadores devem sentir que existem outros e ter o desejo de encontrá-los e conhecê-los. O provimento de uma ferramenta de busca é aconselhável.

Heurística satisfeita: Não

Comentário: o site não abrange interações entre usuários.

#### 6.5.1.25 HA25

O jogo deve minimizar as implicações que envolvem a conexão de rede. A latência e a desconexão podem interromper o jogo e causar atrasos na interação, podendo ocasionar a perda de uma partida.

Heurística satisfeita: Não

Comentário: O site é completamente dependente de conexão com a internet. Não pode ser utilizado sem que o usuário esteja conectado a internet.

#### 6.5.1.26 HA26

As sessões do jogo devem ser iniciadas e fechadas rapidamente, e devem manter a atenção do jogador em intervalos pequenos de tempo.

Heurística satisfeita: Sim

Comentário: O site fornece uma navegação rápida e direta para que um usuário possa iniciar o aprendizado de um padrão de projeto, trocar de subnível, realizar exercícios, sair do site, visualizar os níveis, entre outros.

#### 6.5.1.27 HA27

Ensinar habilidades e controles antes que o jogador precise utilizá-los.

Heurística satisfeita: Não

Comentário: o site não possui nenhum tutorial que guie um novo usuário pelos elementos e fluxos do site.



#### 6.5.1.28 HA28

O ritmo do jogo deve pressionar o jogador, mas sem frustrá-lo.

Heurística satisfeita: Parcialmente

Comentário: o usuário pode utilizar o site à vontade, exceto quando acabar suas energias, o que gera uma pressão de recuperá-las na loja, ou aguardar o tempo necessário para que elas sejam repostas automaticamente. Para obtê-las na loja, o usuário precisa praticar mais exercícios.

#### 6.5.1.29 Análise

Percebe-se que o aspecto de interações entre usuário é inexistente na abordagem gamificada desenvolvida, sendo, porém, algo muito comum tanto em jogos quanto em outras aplicações gamificadas. Essas interações podem ser trabalhadas futuramente na abordagem e site educacional deste trabalho. Além disso, como já visto nos ciclos 2 e 3 da pesquisa ação realizada, assim como apresentado nesta avaliação (HA27), é interessante a existência de um tutorial opcional para novos usuários, mostrando-os brevemente como explorar e utilizar o site.

## 6.6 Resumo do Capítulo

Nesse capítulo, foram apresentados os resultados dos ciclos de pesquisa-ação realizados para analisar a abordagem e o Site Educacional. Inicialmente, foi retomada uma breve definição das fases de cada ciclo e, em seguida, detalhou-se cada um dos 4 ciclos realizados.

Iniciou-se pelo ciclo de Avaliação por Especialista, em que o foco esteve nos conteúdos e exercícios referentes a padrões de projeto, em que um documento foi enviado ao especialista para *feedback*, e, em seguida, os autores realizaram as devidas correções e melhorias no documento. A seguir, foi realizado o segundo ciclo, de Avaliação por Entrevistas com o Público-alvo, em que buscou-se tratar de diferentes aspectos, desde conteúdos até gamificação e usabilidade. As entrevistas seguiram um roteiro e, com elas, foi possível obter visões sobre os pontos fortes e sobre possíveis melhorias, algumas das quais foram prontamente já realizadas. O terceiro ciclo foi de Avaliação por Questionário com Público-alvo, em que o foco foi em relação às expectativas do público-alvo. Com base nas respostas obtidas, foram realizadas análises e algumas melhorias no site educacional. Por fim, realizou-se o quarto ciclo, de Autoavaliação, na qual foi feita uma autoavaliação com base em 28 heurísticas para avaliação de jogos.

Quadro 4 – Heurísticas para Avaliações de Aplicações Gamificadas

Heurística	Descrição
HA1	Os jogadores não precisam ter um manual.
HA2	O jogador deve poder facilmente desligar ou ligar o jogo, visualizar opções, obter ajuda, salvar e pausar em diferentes estágios.
HA3	O jogo deve ser projetado de modo a prevenir erros antes de eles acontecerem.
HA4	O jogador deve poder ver o progresso do jogo e comparar os resultados.
HA5	O jogo deve assegurar que o jogador não tenha que recomeçar a partir de cada erro.
HA6	A interface do jogo deve ser consistente quanto a navegação, design e diálogos.
HA7	O jogo deve fornecer múltiplas maneiras de se realizar uma ação.
HA8	O jogo não deve apresentar tarefas repetitivas ou entediantes.
HA9	O jogo deve fornecer <i>feedback</i> imediato para as ações realizadas.
HA10	Seguir as tendências da comunidade de jogos para diminuir a curva de aprendizado.
HA11	O jogo deve ter objetivos claros.
HA12	O jogador deve obter resultados justos.
HA13	O jogador deve ser recompensado e suas recompensas devem ser significativas.
HA14	O jogo deve ter desafios identificáveis.
HA15	O jogo deve oferecer diferentes níveis de dificuldade.
HA16	O jogo deve ter múltiplas maneiras de se ganhar.
HA17	O jogo deve ter uma fantasia, ou seja, evocar imagens e objetos de situações fictícias.
HA18	Os jogos que envolvam histórias e personagens devem suscitar o interesse pelo que representam.
HA19	O jogo deve oferecer efeitos sonoros interessantes e um visual atraente.
HA20	O jogo deve ter novidades, surpresas e violação das expectativas.
HA21	O jogo deve suportar comunicação.
HA22	O jogo deve oferecer razões para comunicação entre os jogadores.
HA23	O jogo deve suportar grupos ou comunidades.
HA24	O jogo deve incluir suporte para que jogadores possam encontrar outros.
HA25	As sessões do jogo devem ser iniciadas rapidamente, fechadas rapidamente.
HA26	As sessões do jogo devem ser iniciadas rapidamente, fechadas rapidamente.
HA27	Ensinar habilidades e controles antes que o jogador precise usar.
HA28	O ritmo do jogo deve pressionar o jogador, mas sem frustrá-lo.

Fonte: (SOUTO; R., 2015)

## 7 Conclusão

Neste capítulo, busca-se apresentar as considerações finais do trabalho, bem como os principais resultados obtidos com a sua elaboração. Assim, há uma seção *Status Atual do Trabalho*, em que são retomados os objetivos apresentados no capítulo da Introdução, para analisar o cumprimento dos mesmos, e são retomadas também as questões de pesquisa e desenvolvimento. Em seguida, são reportadas algumas *Considerações sobre a Abordagem e o Site Educacional*, com *Pontos Fortes* e as possíveis *Fragilidades*. Por fim, constam as oportunidades de *Trabalhos Futuros*.

A gamificação tem sido amplamente utilizada em diferentes contextos, com diferentes objetivos, e apresentando resultados conforme o contexto e a forma que é aplicada. Aplicar elementos da gamificação, sem possuir uma visão clara do porquê se está aplicando, pode tornar a experiência falsa e não trazer os resultados esperados (DICHEVA et al., 2014). Sendo assim, é importante aplicar elementos significativos, com base na visão do que se deseja alcançar com eles. Por isso, é relevante considerar e entender o lado humano, os sentimentos, as motivações e as ambições. Além disso, há necessidade de conhecimentos embasados na literatura especializada.

Na educação, por exemplo, é possível utilizar a gamificação para engajar e manter o interesse de estudantes em uma aula ou disciplina durante o processo de aprendizagem (DETERDING et al., 2011). Neste trabalho, o objeto da aprendizagem são padrões de projeto. Por serem amplamente aceitos na comunidade de software, os padrões de projeto são de grande relevância no aprendizado dos estudantes da área.

Em vista dessas considerações, buscou-se a elaboração de uma abordagem coerente para auxiliar a aprendizagem de padrões de projeto, optando por elaborar uma abordagem orientada à gamificação e apresentá-la em um site.

### 7.1 *Status Atual do Trabalho*

As atividades e subprocessos referentes à primeira e à segunda etapa do TCC, definidos na seção *Fluxo de Atividades* do Capítulo 4 - Metodologia, foram concluídos.

Em relação aos *Objetivos Específicos* estabelecidos, relacionados a estudos e aquisição de conhecimentos, bem como os referentes à especificação de metodologia e detalhamento, desenvolvimento e análise, tanto da abordagem, quanto do Site, foram atendidos.

No intuito de conferir maior clareza quantos aos *Objetivos Específicos* cumpridos, seguem as colocações:

- Objetivo Específico 01: Estudo dos conceitos inerentes ao escopo do trabalho, com destaque para Padrões de Projeto, Abordagens Educacionais (Ensino-Aprendizagem), Gamificação e Site Educacional. *Status*: Cumprido. Insumos Gerados e Comprobatórios: principalmente, o Capítulo 1 - [Introdução](#) e o Capítulo 2 - [Referencial Teórico](#);
- Objetivo Específico 02: Aquisição de conhecimentos sobre tecnologias que viabilizem a elaboração da abordagem e projeto/desenvolvimento/validação do Site Educacional. *Status*: Cumprido. Insumos Gerados e Comprobatórios: principalmente, o Capítulo 3 - [Suporte Tecnológico](#);
- Objetivo Específico 03: Especificação de metodologias adequadas para condução do levantamento bibliográfico, do desenvolvimento e da análise de resultados. *Status*: Cumprido. Insumos Gerados e Comprobatórios: principalmente, o Capítulo 4 - [Metodologia](#). Aqui, constam métodos de cunho específico ([Levantamento Bibliográfico](#), [Método de Desenvolvimento](#) e [Método de Análise de Resultados](#)), além da classificação da pesquisa, e detalhamentos de fluxo de atividades e cronogramas;
- Objetivo Específico 04: Detalhamento da Abordagem de Aprendizagem, baseando-se no levantamento bibliográfico e nos conceitos estudados. *Status*: Cumprido. Insumos Gerados e Comprobatórios: principalmente, o Capítulo 5 - [Abordagem e Site Educacional](#). Esse capítulo contém a especificação da abordagem orientada ao plano de gamificação estabelecido;
- Objetivo Específico 05: Projeto e implementação do Site Educacional, apresentando a abordagem definida. *Status*: Cumprido. Insumos Gerados e Comprobatórios: principalmente, o Capítulo 5 - [Abordagem e Site Educacional](#). Esse capítulo contém o detalhamento da elaboração do Site Educacional, sendo esse orientado às boas práticas de Usabilidade (um recorte centrado nas heurísticas de Nielsen) e às demais práticas de design detalhadas no capítulo. Constam, dentre outros esforços, a identidade visual e o protótipo de alta fidelidade do site, bem como explicação sobre suas telas e fluxos, e
- Objetivo Específico 06: Análise dos resultados obtidos, procurando documentá-los e expondo ideias para trabalhos futuros. *Status*: Cumprido. Insumos Gerados e Comprobatórios: principalmente, o Capítulo 6 - [Análise de Resultados](#). Nele, constam os ciclos de pesquisa-ação realizados, com suas respectivas fases e resultados obtidos. Também o presente capítulo, Capítulo 7 - [Conclusão](#), onde são detalhadas mais claramente as oportunidades de trabalhos futuros.

Diante do exposto, pode-se retomar às questões de pesquisa e desenvolvimento.

“É possível, conforme ocorre em outros contextos educacionais, impactar positivamente a aprendizagem de padrões de projeto por meio de uma abordagem orientada à gamificação?”

É possível, sim, impactar positivamente a aprendizagem de padrões de projeto. Essa é uma afirmativa generalizada, pois não são todos os estudantes que se sentirão motivados através de uma abordagem gamificada. Entretanto, os resultados desse trabalho apontam que, em uma certa parcela, os estudantes têm interesse de utilizar um site gamificado para aprender padrões de projeto. Esse site, entretanto, seria provavelmente utilizado em sua maioria das vezes como um complemento de estudo, e não apenas um recurso principal.

“Quais elementos da gamificação são mais indicados para a aprendizagem de padrões de projeto e/ou conteúdos de complexidade similar?”

Os elementos mais indicados são os que buscam o engajamento do estudante. Esse é o foco principal. Entretanto, há diferentes formas de engajá-los e, neste trabalho, algumas delas foram abordadas com foco: a motivação através do senso de progresso e de alcançar objetivos, com a aplicação de missões, conquistas, pontos de experiência e subir de níveis; a motivação através de influências limitadas de empoderamento, com a aplicação de “impulsionadores”; a motivação através do sentimento de posse e de desejar manter algo para si, com a aplicação de pontos permutáveis (cubos de gelo); e a motivação através da urgência e do medo de perder algo, com a aplicação de perda de energias ao realizar atividades e duplamente ao errar as atividades, e a aplicação de um tempo limitado aos impulsionadores.

“Como viabilizar a apresentação da abordagem de aprendizagem de padrões de projeto, orientada à gamificação, em um site educacional?”

O detalhamento desse aspecto foi tratado no Capítulo 5 - [Abordagem e Site Educacional](#). O site é desenvolvido como uma “instância” da abordagem, voltado ao aprendizado de padrões de projeto. Os elementos de gamificação da abordagem não estão diretamente relacionados a padrões de projeto, mas sim ao engajamento do estudante no site. Sendo assim, a abordagem no site faz-se presente através de missões, conquistas e elementos do usuário.

## 7.2 Considerações sobre a Abordagem e o Site Educacional

Nas seções a seguir, são apresentadas considerações sobre a abordagem e o site educacional desenvolvidos, bem como oportunidades para trabalhos futuros.

### 7.2.1 Pontos Fortes

A abordagem elaborada especifica um plano de gamificação com detalhamento dos elementos a serem aplicados, bem como o objetivo de cada elemento, deixando claro o motivo do mesmo estar sendo utilizado. O Site Educacional, alinhado a princípios de usabilidade e a boas práticas de design, bem como uma identidade visual bem definida, demonstra consistência e busca a melhor experiência para o usuário. Assim, com base nos resultados apresentados nesse trabalho através de entrevistas e de um questionário, percebeu-se que a aderência ao uso do site demonstrou-se positiva.

### 7.2.2 Fragilidades

A quantidade de padrões de projeto de software apresentados no site é apenas uma amostra, então é bem pequena. Sendo assim, é relevante ressaltar que, para a uma melhor aplicação do site em um contexto educacional, é importante que todos os padrões estejam inseridos no site.

Adicionalmente, a análise da aderência e da efetividade da abordagem e do site junto a estudantes foi baseada, principalmente, nas expectativas dos estudantes e usuários. A análise foi realizada em um período reduzido, então o engajamento dos usuários ocorreu apenas nesse curto período. Dessa forma, abrem-se oportunidades para que sejam feitas pesquisas mais profundas, tratando da motivação a médio ou longo prazo, além da verificar efetividade do aprendizado em si.

Outra fragilidade, atrelada diretamente ao Site Educacional desenvolvido, é que o mesmo não está preparado para um ambiente responsivo, o que pode ser mais bem tratado em trabalhos futuros.

### 7.2.3 Trabalhos Futuros

Com base nas fragilidades apresentadas, e em observações feitas durante a [Análise de Resultados](#), foram identificadas as seguintes melhorias e possibilidades para trabalhos futuros:

- Personalização da trilha de aprendizado, estruturando a ordem de apresentação dos padrões de projeto com base no conhecimento prévio do estudante;
- Inserir conteúdos e exercícios de mais padrões de projeto, bem como diversificar os exercícios, e elaborar alguns de maior complexidade e contexto;
- Tornar o site responsivo, para uma melhor experiência do usuário em diferentes ambientes;

- 
- Adicionar mais elementos de gamificação à abordagem, como, por exemplo, relacionamentos, comunicação e *rankings* para interação entre usuários;
  - Encontrar uma forma de orientar o aprendizado com base nos erros do estudante em exercícios, porém sem afetar a mecânica de várias tentativas do usuário em um mesmo exercício;
  - Apresentar um “passo a passo” ou tutorial opcional ao usuário quando utilizar o site pela primeira vez;
  - Entender a possibilidade de adicionar recursos variados de estudo, como vídeos disponibilizados no site e direcionamento para bibliografias externas; e
  - Realizar uma pesquisa mais profunda acerca do engajamento de estudantes a médio/longo prazo, e da efetividade do site no aprendizado obtido.

## Referências

- AHMAD, M. O.; MARKKULA, J.; OIVO, M. Kanban in software development: A systematic literature review. In: *2013 39th Euromicro Conference on Software Engineering and Advanced Applications*. [S.l.: s.n.], 2013. p. 9–16. ISSN: 2376-9505. Citado na página 51.
- AKITA, F. *Quais são algumas das piores práticas para aplicações Ruby on Rails?* 2013. Disponível em: <<https://www.akitaonrails.com/2013/03/24/quais-sao-algumas-das-piores-praticas-para-aplicacoes-ruby-on-rails--2>>. Acesso em: 28/06/2023. Citado na página 27.
- ALBAUM, G. The likert scale revisited. *Market Research Society. Journal.*, SAGE Publications Sage UK: London, England, v. 39, n. 2, p. 1–21, 1997. Citado na página 53.
- ALEXANDER, C. *A pattern language: towns, buildings, construction*. [S.l.]: Oxford university press, 1977. Citado na página 24.
- ALVES, F. *Gamification: como criar experiências de aprendizagem engajadoras*. [S.l.]: DVS editora, 2015. Citado 2 vezes nas páginas 19 e 20.
- APPLETON, B. Patterns and software: Essential concepts and terminology. *Object Magazine Online*, v. 3, n. 5, p. 20–25, 1997. Citado na página 25.
- BARBA, D.; BARCELOS, O. Design de interface de um aplicativo voltado à alfabetização infantil de crianças de 2 a 8 anos. 2022. Disponível em: <<https://repositorio.ufsc.br/handle/123456789/242872>>. Citado 4 vezes nas páginas 38, 39, 40 e 41.
- BECK, K. Using pattern languages for object-oriented programs. In: *OOPSLA-87 workshop on the Specification and Design for Object-Oriented Programming*. [S.l.: s.n.], 1987. Citado na página 24.
- BECK, K. *Extreme Programming Explained: Embrace Change*. [S.l.]: Addison-Wesley Professional, 2000. ISBN 9780201616415. Citado na página 51.
- BERBEL, N. A. N. As metodologias ativas e a promoção da autonomia de estudantes. v. 32, n. 1, p. 25–40, 2011. ISSN 1679-0383. Disponível em: <<https://ojs.uel.br/revistas/uel/index.php/seminasoc/article/view/10326>>. Citado na página 34.
- BROOKE, J. SUS: A quick and dirty usability scale. v. 189, 1995. Citado na página 86.
- CETIC.BR. *Painel TIC COVID-19: Pesquisa on-line com usuários de Internet no Brasil - 4ª edição: Cultura, Comércio Eletrônico, Serviços Públicos On-line, Telessaúde, Ensino Remoto e Teletrabalho*. 2022. Disponível em: <<https://www.cetic.br/pt/publicacao/painel-tic-covid-19-pesquisa-online-com-usuarios-de-internet-no-brasil-4edicao/>>. Citado na página 38.



- CETIC.BR. *Pesquisa sobre o uso das Tecnologias de Informação e Comunicação nos domicílios brasileiros - TIC Domicílios 2021*. 2022. Disponível em: <[https://cetic.br/media/docs/publicacoes/2/20221121125504/tic\\_domicilios\\_2021\\_livro\\_eletronico.pdf](https://cetic.br/media/docs/publicacoes/2/20221121125504/tic_domicilios_2021_livro_eletronico.pdf)>. Citado na página 37.
- CHOU, Y. Actionable gamification. *Beyond points, badges, and leaderboards*, 2015. Citado 3 vezes nas páginas 36, 50 e 69.
- DETERDING, S. et al. From game design elements to gamefulness: defining "gamification". In: *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments*. [S.l.: s.n.], 2011. p. 9–15. Citado 2 vezes nas páginas 19 e 114.
- DICHEVA, D. et al. A course gamification platform supporting student motivation and engagement. In: IEEE. *2014 International Conference on Web and Open Access to Learning (ICWOAL)*. [S.l.], 2014. p. 1–4. Citado 4 vezes nas páginas 19, 20, 50 e 114.
- ENGEL, G. I. Pesquisa-ação. p. 181–191, 2000. ISSN 0104-4060, 1984-0411. Disponível em: <<https://www.scielo.br/j/er/a/dDzfLYyDpPZ3kM9xNSqG3cw/?lang=pt>>. Citado na página 52.
- FIGMA. *Prototyping*. 2023. Online, acessado em 30 de abril de 2023. Disponível em: <<https://www.figma.com/prototyping/>>. Citado na página 43.
- FIQUEIREDO, M.; PAZ, T.; JUNQUEIRA, E. Gamificação e educação: um estado da arte das pesquisas realizadas no brasil. v. 4, n. 1, p. 1154, 2015. ISSN 2316-8889. Disponível em: <<http://ojs.sector3.com.br/index.php/wcbie/article/view/6248>>. Citado na página 18.
- FREITAS, S. de et al. Gamificação e avaliação do engajamento dos estudantes em uma disciplina técnica de curso de graduação. In: *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*. [S.l.: s.n.], 2016. v. 27, n. 1, p. 370. Citado na página 35.
- GADOTTI, M. História das ideias pedagógicas. 2003. Disponível em: <<http://acervo.paulofreire.org:8080/xmlui/handle/7891/2787>>. Citado na página 34.
- GAMMA, E. et al. *Design patterns: elements of reusable object-oriented software*. [S.l.]: Pearson Deutschland GmbH, 1995. Citado 17 vezes nas páginas 18, 24, 26, 27, 28, 29, 30, 31, 32, 33, 50, 62, 77, 134, 136, 139 e 147.
- GERHARDT, T. E.; SILVEIRA, D. T. *Métodos de Pesquisa*. [S.l.]: PLAGEDER, 2009. ISBN 9788538600718. Citado 2 vezes nas páginas 48 e 49.
- GIL, A. C. *Como elaborar projetos de pesquisa*. 4. ed. [S.l.]: Atlas, 2007. ISBN 9788522431694. Citado 2 vezes nas páginas 49 e 52.
- GIT. *Git*. 2023. Online, acessado em 30 de abril de 2023. Disponível em: <<https://git-scm.com/>>. Citado na página 44.
- GITHUB. *Quickstart, GitHub*. 2023. Online, acessado em 02 de maio de 2023. Disponível em: <<https://docs.github.com/en/get-started/quickstart>>. Citado na página 45.

- GOOGLE. *googleforms*. 2023. *Online*, acessado em 09 de maio de 2023. Disponível em: <<https://www.google.com/intl/pt-BR/forms/about/>>. Citado na página 46.
- GRANIĆ, A. Educational technology adoption: A systematic review. v. 27, n. 7, p. 9725–9744, 2022. ISSN 1573-7608. Disponível em: <<https://doi.org/10.1007/s10639-022-10951-7>>. Citado na página 38.
- HAMARI, J.; KOIVISTO, J.; SARSA, H. Does gamification work?—a literature review of empirical studies on gamification. In: IEEE. *2014 47th Hawaii international conference on system sciences*. [S.l.], 2014. p. 3025–3034. Citado na página 20.
- HEROKU. *Heroku*. 2023. *Online*, acessado em 02 de maio de 2023. Disponível em: <<https://devcenter.heroku.com/articles/how-heroku-works>>. Citado na página 45.
- LARMAN, C. *Applying UML and patterns: an introduction to object oriented analysis and design and interative development*. [S.l.]: Pearson Education India, 2012. Citado na página 18.
- LEE, J. J.; HAMMER, J. Gamification in education: What, how, why bother? *Academic exchange quarterly*, Chattanooga State Technical Community College, v. 15, n. 2, p. 146, 2011. Citado na página 20.
- LIDWELL, W.; HOLDEN, K.; BUTLER, J. *Princípios Universais do Design*. [S.l.]: Bookman, 2010. Citado na página 41.
- LOVATO, F. L.; MICHELOTTI, A.; LORETO, E. L. d. S. Metodologias ativas de aprendizagem: Uma breve revisão. v. 20, n. 2, 2018. ISSN 2178-7727. Disponível em: <<http://www.periodicos.ulbra.br/index.php/acta/article/view/3690>>. Citado na página 34.
- LUCIDCHART. *lucidchart*. 2023. *Online*, acessado em 09 de maio de 2023. Disponível em: <<https://www.lucidchart.com/blog/pt/como-usar-os-recursos-basicos-do-lucidchart>>. Citado na página 46.
- MALDONADO, J. C. et al. Padrões e frameworks de software. *Notas Didáticas, Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo, ICMC/USP, São Paulo, SP, Brasil*, p. 28, 2002. Citado na página 25.
- MCGONIGAL, J. *Reality is broken: Why games make us better and how they can change the world*. [S.l.]: Penguin, 2011. Citado na página 35.
- MINUZI, N. A. et al. GAMIFICAÇÃO NA EDUCAÇÃO PROFISSIONAL e TECNOLÓGICA. v. 7, n. 1, 2018. ISSN 2594-4576. Disponível em: <<http://seer.faccat.br/index.php/redin/article/view/1061>>. Citado na página 19.
- MIRO. *Miro*. 2023. *Online*, acessado em 15 de junho de 2023. Disponível em: <<https://miro.com/pt/>>. Citado na página 42.
- MORSCHHEUSER, B. et al. How to design gamification? a method for engineering gamified software. *Information and Software Technology*, v. 95, p. 219–237, 2018. ISSN 0950-5849. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S095058491730349X>>. Citado na página 18.

- MORÁN, J. Mudando a educação com metodologias ativas. v. 2, n. 1, p. 15–33, 2015. Citado 2 vezes nas páginas 34 e 35.
- NETLIFY. *Netlify*. 2023. *Online*, acessado em 08 de dezembro de 2023. Disponível em: <<https://www.netlify.com/>>. Citado na página 45.
- NIELSEN, J. *Usability Engineering*. [S.l.]: Morgan Kaufmann, 1994. Google-Books-ID: 95As2OF67f0C. ISBN 9780125184069. Citado 3 vezes nas páginas 38, 40 e 50.
- NODE. *About Node*. 2023. *Online*, acessado em 15 de junho de 2023. Disponível em: <<https://nodejs.org/en>>. Citado na página 44.
- OVERLEAF. *Documentation*. 2023. *Online*, acessado em 24 de abril de 2023. Disponível em: <<https://pt.overleaf.com/learn>>. Citado na página 42.
- REACT. *React Quick Start*. 2023. *Online*, acessado em 15 de junho de 2023. Disponível em: <<https://react.dev/learn>>. Citado na página 43.
- RUGHINIŞ, R. Flexible gamification in a social learning situation. insights from a collaborative review exercise. International Society of the Learning Sciences, 2013. Citado na página 19.
- SALES, A. Barros de; SERRANO, M.; SERRANO, M. Aprendizagem baseada em projetos na disciplina de interação humano-computador. n. 37, p. 49–64, 2020. ISSN 1646-9895. Disponível em: <<https://dialnet.unirioja.es/servlet/articulo?codigo=7871592>>. Citado na página 35.
- SANTOS, I.; PIMENTEL, F.; SELLA, A. Gamificação no contexto educacional e o processo de ensino-aprendizagem. p. 1–9, 2019. Citado 2 vezes nas páginas 19 e 20.
- SANTOS, R. V. d. Abordagens do processo de ensino e aprendizagem. *Rev Integr*, v. 11, n. 40, p. 19–31, 2005. Citado 2 vezes nas páginas 18 e 19.
- SCHWABER, K.; SUTHERLAND, J. *Scrum Guide | Scrum Guides*. 2020. Disponível em: <<https://scrumguides.org/scrum-guide.html>>. Citado 2 vezes nas páginas 50 e 51.
- SILVA, M. R. d. Contribuições do design para a evolução do hiperlivro do AVEA-LIBRAS: o processo de desenvolvimento de interfaces para objetos de aprendizagem. 2013. Citado na página 38.
- SILVA, R.; MIRANDA, E. M. de; PEREIRA, T. F. Uma experiência no ensino de padrões de projeto utilizando jogo no processo de ensino-aprendizagem. In: *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*. [S.l.: s.n.], 2016. v. 5, n. 1, p. 1218. Citado 3 vezes nas páginas 18, 20 e 35.
- SILVEIRA, I.; SILVA, L. Aprendizagem de padrões de projeto em ciência da computação através de jogos digitais in: Anais do xxvi congresso da sbc. *São Paulo*, 2006. Citado na página 18.
- SOUTO, E.; R., S. Utilizando heurísticas de jogos para avaliação de um aplicativo gamificado. p. 666–673, 2015. Citado 2 vezes nas páginas 105 e 113.
- SOUZA, M.; FRANÇA, C. Ferramentas de auxílio ao aprendizado de programação : Um estudo comparativo. In: . [S.l.: s.n.], 2013. Citado na página 21.

TRELLO. *Trello*. 2023. *Online*, acessado em 15 de junho de 2023. Disponível em: <<https://trello.com/>>. Citado na página 45.

VERNEIL, M. d.; BERGE, Z. L. Going online: Guidelines for faculty in higher education. v. 1, n. 13, p. 13–32, 2000. ISSN 1065-6901. Disponível em: <<https://www.learntechlib.org/primary/p/8057/>>. Citado 4 vezes nas páginas 38, 39, 40 e 41.

Visual Studio Code. *Getting Started*. 2023. *Online*, acessado em 15 de abril de 2023. Disponível em: <<https://code.visualstudio.com/docs>>. Citado na página 44.

WELLS, D. *Extreme Programming: A Gentle Introduction*. 2013. Disponível em: <<http://www.extremeprogramming.org/>>. Citado na página 51.

ZICHERMANN, G.; CUNNINGHAM, C. *Gamification by design: Implementing game mechanics in web and mobile apps*. [S.l.]: "O'Reilly Media, Inc.", 2011. Citado na página 35.

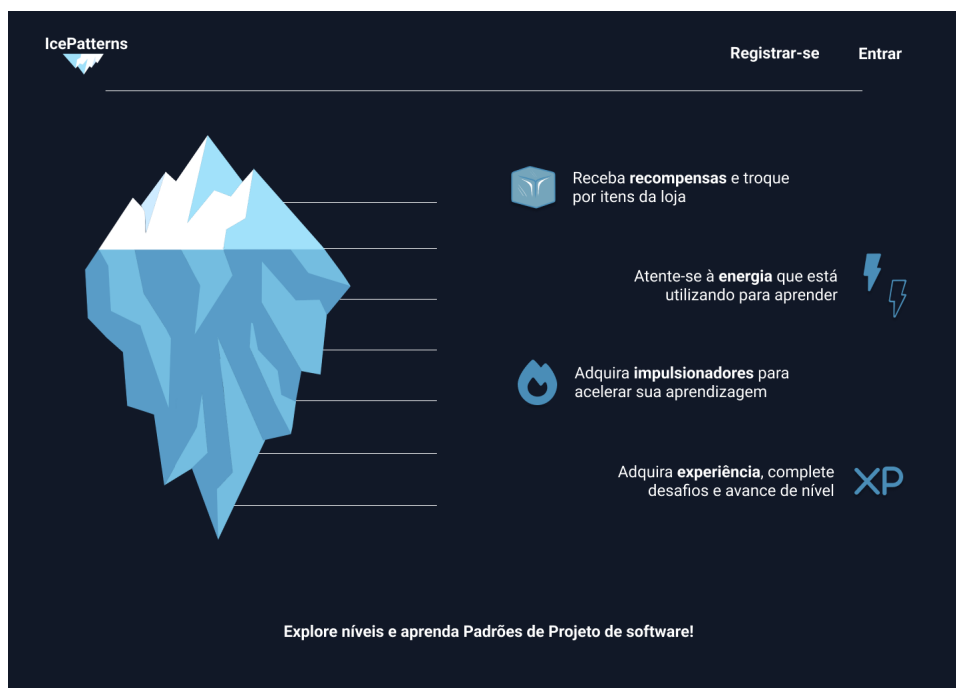
ZOTERO. *Quick Start Guide*. 2023. *Online*, acessado em 24 de abril de 2023. Disponível em: <[https://www.zotero.org/support/quick\\_start\\_guide](https://www.zotero.org/support/quick_start_guide)>. Citado na página 42.

ÁVILA, J. d. Guia de boas práticas para o desenvolvimento de materiais didáticos digitais para educação a distância: o caso do núcleo multiprojetos de tecnologia educacional (NUTE-UFSC). 2016. Disponível em: <<https://repositorio.ufsc.br/handle/123456789/165072>>. Citado 2 vezes nas páginas 38 e 41.

# Apêndices

# APÊNDICE A – Protótipo de Alta Fidelidade

Figura 55 – Protótipo - Página Inicial de Visitantes



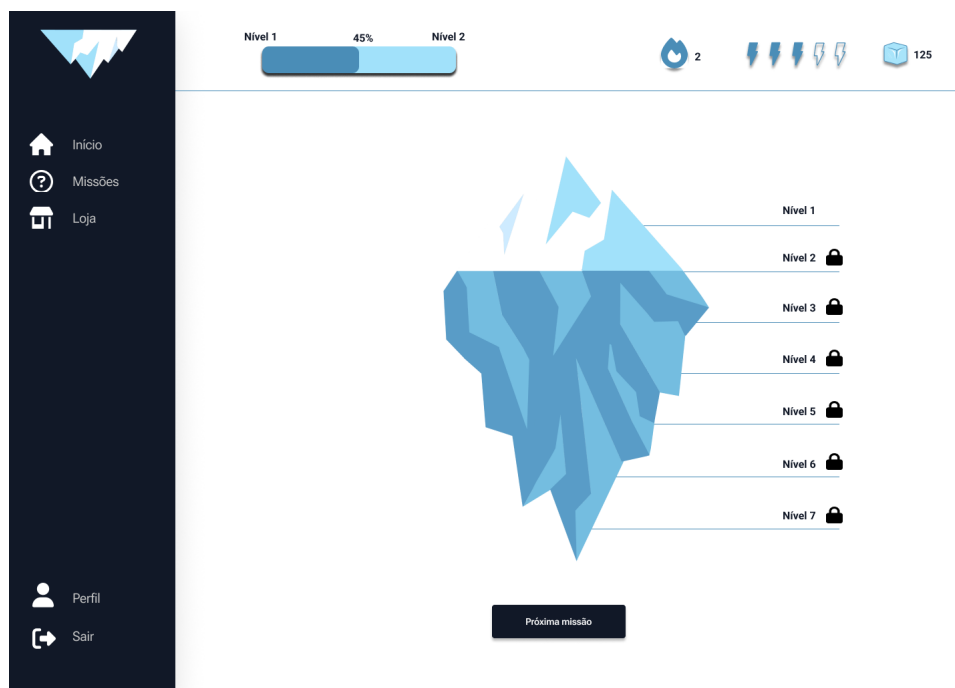
Fonte: Autores

Figura 56 – Protótipo - *Login* e Registro de Usuário

Dois protótipos de formulários de usuário são exibidos lado a lado. O formulário de 'Login' possui campos para 'Usuário' e 'Senha', com o botão 'Entrar' no rodapé. O formulário de 'Registro' possui campos para 'Usuário', 'Email', 'Senha' e 'Confirmar senha', com o botão 'Registrar' no rodapé. Ambos os formulários têm um ícone de fechamento 'x' no canto superior direito.

Fonte: Autores

Figura 57 – Protótipo - Página Inicial para Usuário Registrado



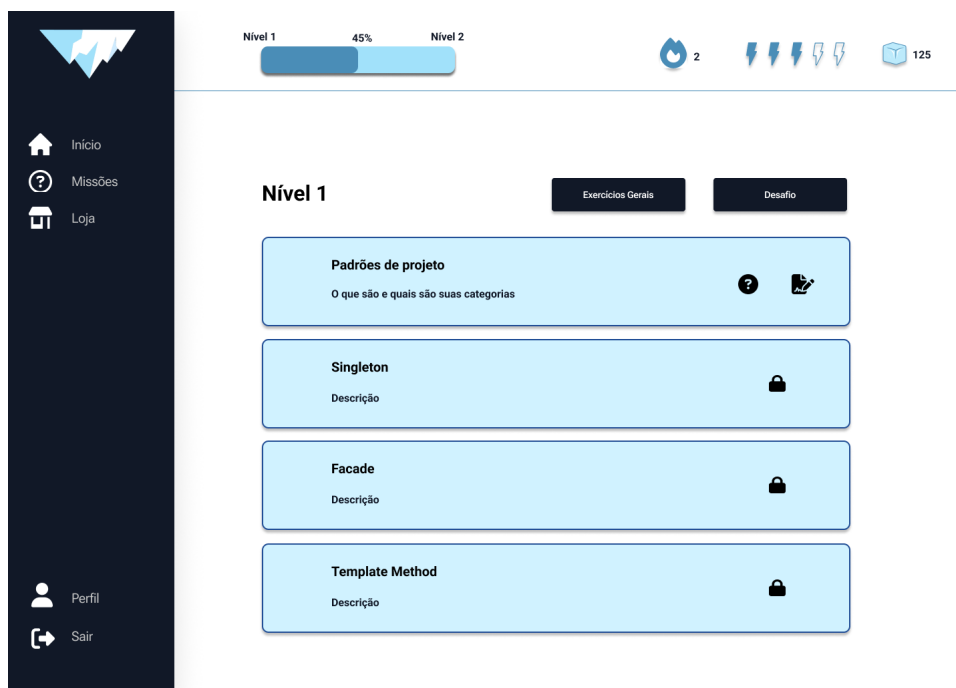
Fonte: Autores

Figura 58 – Protótipo - Cartões Explicativos



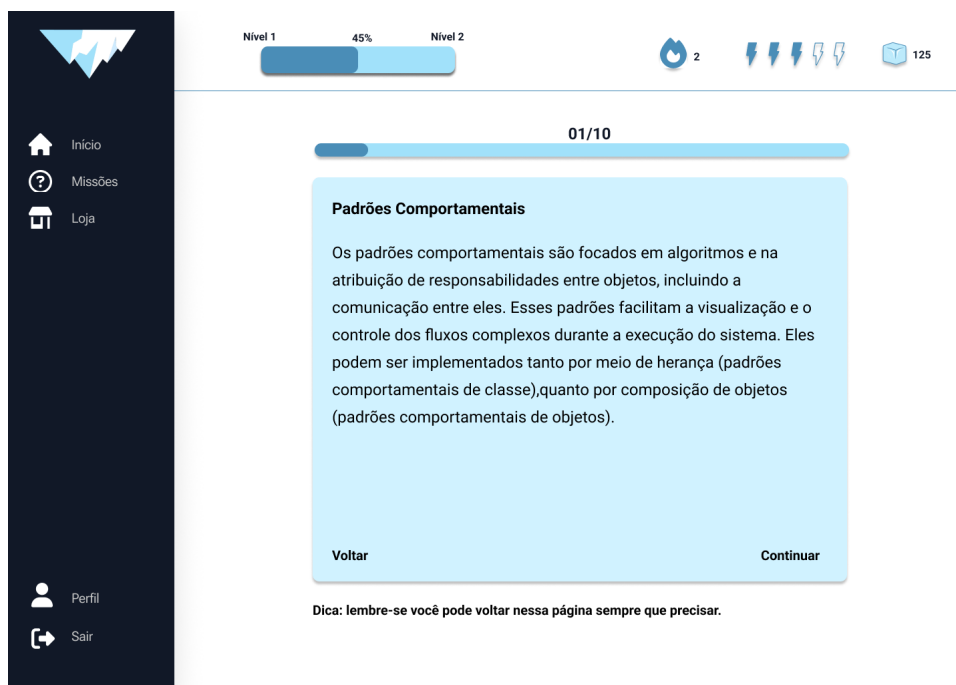
Fonte: Autores

Figura 59 – Protótipo - Página dos Níveis



Fonte: Autores

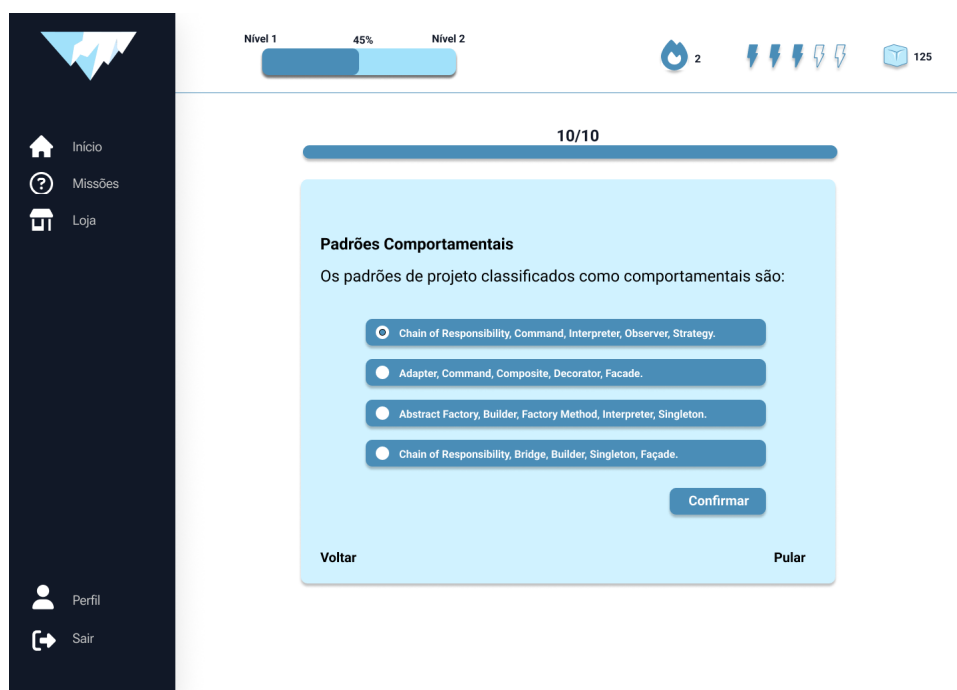
Figura 60 – Protótipo - Página de Conteúdo



Fonte: Autores



Figura 61 – Protótipo - Página de Exercícios



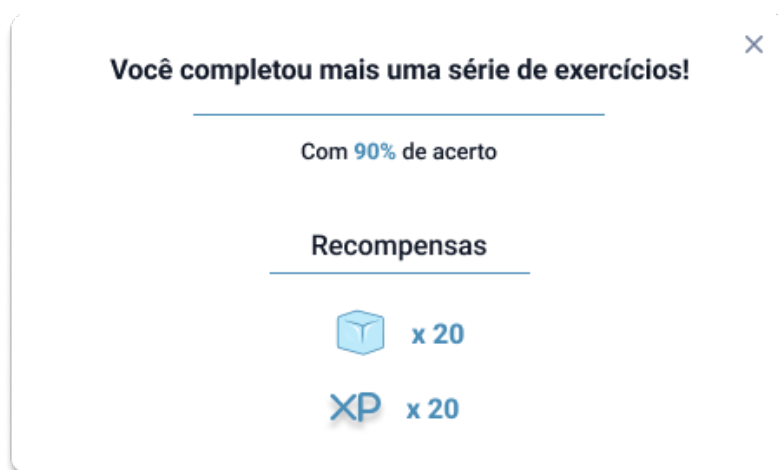
Fonte: Autores

Figura 62 – Protótipo - Página Final de Conteúdo



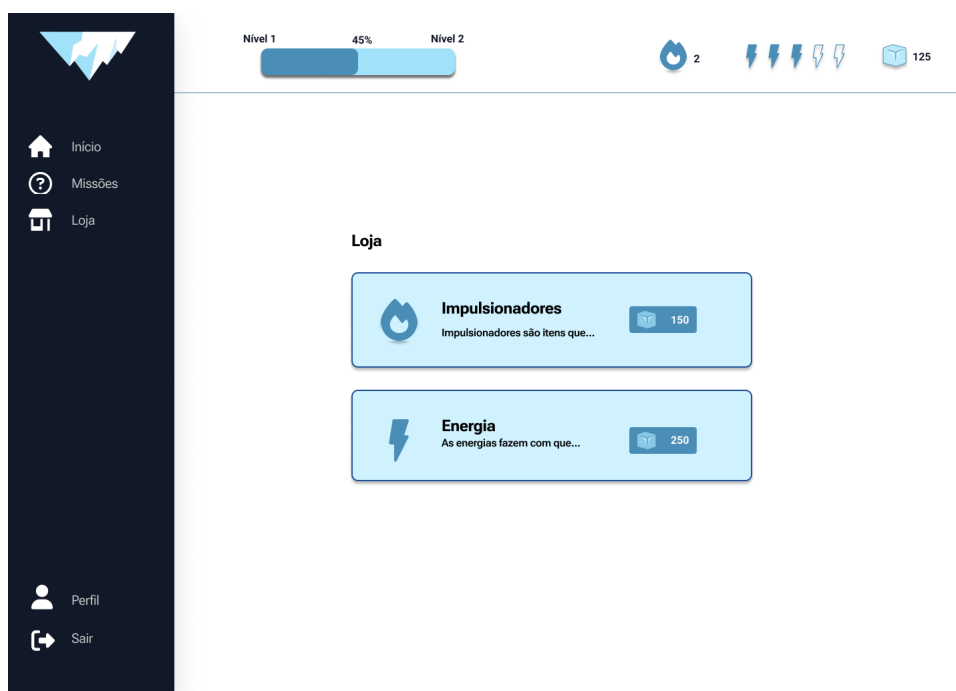
Fonte: Autores

Figura 63 – Protótipo - Finalização de Exercícios



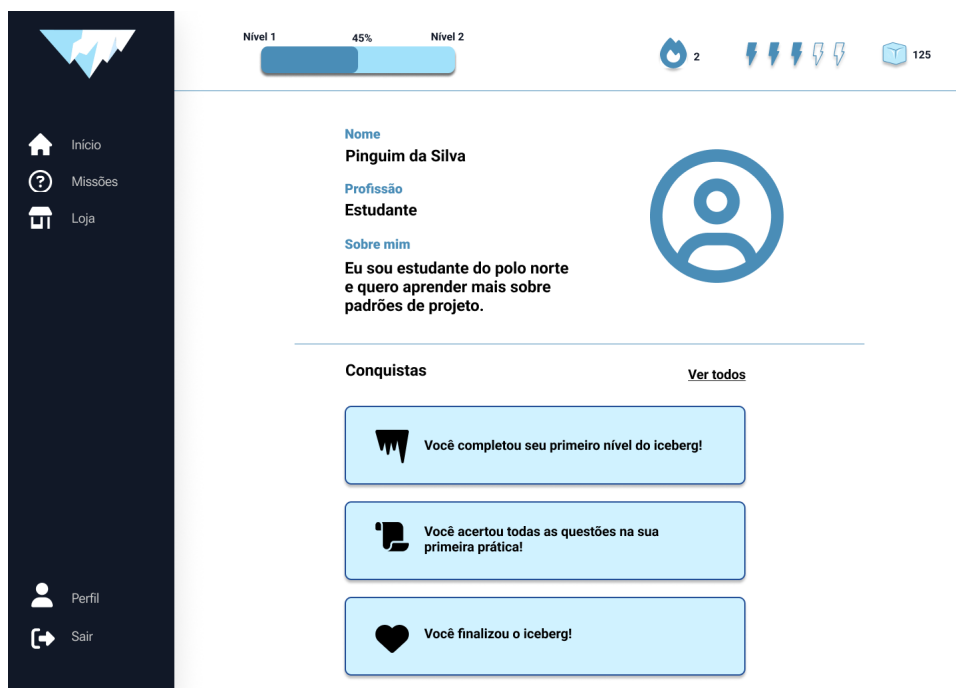
Fonte: Autores

Figura 64 – Protótipo - Página da Loja



Fonte: Autores

Figura 65 – Protótipo - Página de Perfil do Usuário



Fonte: Autores

Figura 66 – Protótipo - Página de Missões



Fonte: Autores

# APÊNDICE B – Conteúdo e Exercícios no Site Educacional

## B.1 Conteúdo - Padrões de Projeto

Os padrões de projeto em software são soluções para problemas comuns em um projeto de software. Um padrão não é um trecho de código utilizável em diversas situações, mas sim um conceito geral utilizado para resolver problemas de projeto orientado a objetos.

Os padrões de projetos não são necessariamente criados ou inventados por alguém. São soluções típicas para problemas comuns. Sendo assim, as soluções vão sendo repetidamente utilizadas em projetos, e eventualmente ela será descrita e caracterizada de maneira mais formal, surgindo assim um padrão de projeto.

Um livro de grande referência para o assunto, “Padrões de Projeto - Soluções reutilizáveis de software orientado a objetos” dos autores conhecidos como “Gang of Four”, identifica 23 padrões, divididos em 3 categorias com base em seu propósito: criacionais, estruturais e comportamentais.

- Padrões Criacionais: são utilizados na criação de objetos para flexibilizar, tornando os sistemas independentes da maneira como os objetos são criados, além de aumentar a reutilização de código;
- Padrões Estruturais: tratam da composição de classes e objetos para formar estruturas maiores, ainda mantendo-as flexíveis e eficientes, e
- Padrões Comportamentais: focam na atribuição de responsabilidades entre objetos e a comunicação entre eles, facilitando a visualização e o controle de fluxos complexos.

### **Alguns elementos dos padrões de projeto**

Para descrever os padrões de projeto, alguns elementos são comumente utilizados, sendo alguns deles:

- Nome e classificação;
- Motivação: explica o problema e a solução que o padrão torna possível;
- Aplicabilidade: explicita situações em que o padrão pode ser aplicado, e como reconhecer essas situações;

- Estruturas: mostram as partes do padrão e como se relacionam;
- Exemplo de código: trechos de código que mostram como o padrão pode ser implementado, e
- Padrões relacionados: apresentam os padrões que estão intimamente relacionados a ele, explicitando as semelhanças, diferenças e como podem ser usados em conjunto.

### Contras de alguns padrões

Os padrões de projeto não são soluções prontas fixas. Logo, não devem ser usadas em todos os contextos sem justificativas. O uso ineficiente de padrões de projeto pode complicar o projeto de software e gerar mais dificuldade do que solucionar problemas. Martin Fowler trata um pouco disso em seu livro “*Refactoring*”.

Sendo assim, é importante sempre entender as situações em que os padrões podem ser úteis, bem como estar ciente dos seus possíveis malefícios em caso de alterações futuras no projeto, além de compreender como utilizar os padrões adaptados a cada contexto.

## B.2 Exercícios - Padrões de Projeto

1. Os padrões de projeto comportamentais se preocupam com algoritmos e atribuição de responsabilidade entre objetos, focando também nos padrões de comunicação entre objetos, e como eles são interconectados.
  - Verdadeiro
  - Falso
2. Os padrões de projeto são soluções fixas que podem ser aplicadas nos problemas de projeto de software. Mesmo que não seja a solução mais eficiente, aplicar algum padrão de projeto com certeza trará algum benefício à sua implementação.
  - Verdadeiro
  - Falso
3. Assinale a alternativa INCORRETA:
  - Os padrões de projeto são soluções comuns para problemas de software
  - No livro do Gang of Four os padrões de projeto são classificados em criacionais, estruturais e comportamentais
  - **É sempre importante utilizar diferentes padrões em um único projeto de software para garantir a flexibilização e a reutilização do código**

4. Sobre padrões de projeto, assinale a sequência que corresponde às afirmativas a seguir, sendo V para verdadeira e F para falsa:
- Os padrões de projeto foram criados pelo Gang of Four;
  - O uso dos padrões em qualquer projeto de software garante a redução de código duplicado e aumenta a reutilização de código;
  - Alguns elementos comumente utilizados para descrever cada padrão de projeto são: nome, propósito, estrutura e aplicabilidade.
- V - V - V
  - F - V - F
  - V - F - F
  - **F - F - V**

### B.3 Conteúdo - Singleton

O *Singleton* é um padrão criacional que, de acordo com Gamma et al., é usado para “garantir que uma classe tenha somente uma instância, fornecendo um ponto global de acesso para a mesma”.

O *Singleton* resolve essencialmente dois problemas:

- Garantir que uma classe tenha somente uma instância: cria-se um objeto, e depois decide-se criar um novo. Porém, ao invés de receber um novo objeto, receberá o objeto que já foi criado anteriormente.
- Fornecer um ponto global de acesso à instância: assim como uma variável global, o padrão *Singleton* permite que você acesse algum objeto de qualquer lugar no programa. Contudo, ele também protege aquela instância de ser sobrescrita por outro código.

Figura 67 – Exemplo de Código - *Singleton*

```
1 public class Singleton {
2     private static Singleton uniqueInstance ;
3
4     private Singleton () {
5     }
6
7     public static synchronized Singleton getInstance () {
8         if ( uniqueInstance == null )
9             uniqueInstance = new Singleton () ;
10
11         return uniqueInstance ;
12     }
13 }
```

Fonte: Autores, adaptado de Gamma et al. (1995)

O construtor privado garante que nenhuma outra classe do sistema consiga instanciar a classe do *Singleton*, fazendo com que nenhuma outra classe crie várias instâncias dessa mesma classe.

O método de acesso público garante que exista apenas um ponto de acesso global para essa classe. É a partir desse método que a lógica do *Singleton* funciona, pois assim apenas uma instância dessa classe será retornada para o sistema.

O atributo estático garante que não seja possível criar uma classe fora da lógica do *Singleton*, fazendo com que seja preciso acessar o método estático *getInstance()*, o qual retorna esse atributo estático.

### Aplicabilidade

- Quando o programa deve ter apenas uma instância disponível para todos os clientes, e
- Quando é necessário um controle mais estrito sobre as variáveis globais.

## B.4 Exercícios - Singleton

1. O padrão de projeto *Singleton*, que garante que uma classe tenha somente uma instância, fornecendo um ponto global de acesso para essa instância, é classificado como um padrão de projeto:
  - **Criacional**, que fornece um mecanismo de criação de objetos mais flexível, ajudando a tornar o sistema independente da forma como os objetos são criados, compostos e representados;

- Estrutural, que define como montar objetos e classes em estruturas maiores;
  - Comportamental, que cuida da comunicação eficiente e da atribuição de responsabilidade entre objetos.
2. O padrão *Singleton* funciona criando um novo construtor da classe, que verifica se o objeto a ser criado já existe, e retorna sempre essa mesma instância, sem criar novas instâncias toda vez que for chamado:
- **Verdadeiro**
  - Falso
3. Maria deparou-se com um problema no seu projeto de software, e para resolvê-lo, pensou em utilizar o padrão *Singleton*. Porém, está na dúvida se a utilização do padrão realmente é necessária e será benéfica. Qual das situações a seguir representa uma oportunidade de aplicar o padrão *Singleton*:
- Quando não se sabe exatamente os tipos e as dependências exatas dos objetos com o qual o código funcionará.
  - Quando a interface de uma classe existente não for compatível com o restante do código.
  - Quando se deseja estruturar um subsistema em camadas.
  - **Quando possui uma classe que deve ter apenas uma única instância disponível para todos os clientes.**

## B.5 Conteúdo - Facade

O *Facade* é um padrão estrutural que fornece uma interface simplificada para uma biblioteca, um framework ou qualquer conjunto complexo de classes.

De acordo com Gamma et al., o *Facade* “fornece uma interface unificada para um conjunto de interfaces em um subsistema. Facade define uma interface de nível mais alto que torna o subsistema mais fácil de ser usado.”

Ter uma fachada (*Facade*) é útil quando é necessário integrar a aplicação com uma biblioteca mais complexa, com muitas funcionalidades. Dessa forma, a fachada poderia fornecer convenientemente o acesso a uma funcionalidade específica do subsistema (biblioteca). Ao final, ao invés da aplicação se comunicar diretamente com os objetos do subsistema, ela é utilizada através da fachada.



Figura 68 – Exemplo de Código - *Facade*

```
1 class SubSystem1 {
2     // ...
3 }
4
5 class SubSystem2 {
6     // ...
7 }
8
9 class SubSystem3 {
10    // ...
11 }
12
13 class SubSystem4 {
14    // ...
15 }
16
17 class Facade {
18     public void process () {
19         SubSystem1 subsystem1 = new SubSystem1 () ;
20         SubSystem2 subsystem2 = new SubSystem2 () ;
21         SubSystem3 subsystem3 = new SubSystem3 () ;
22         SubSystem4 subsystem4 = new SubSystem4 () ;
23         // ...
24     }
25 }
26
27 class Application {
28     public static void main ( String [] args ) {
29         Facade facade = new Facade () ;
30         facade.process () ;
31     }
32 }
```

Fonte: Autores, adaptado de Gamma et al. (1995)

Os "subsystems" representam algumas classes, por exemplo, de um framework complexo.

A classe *Facade* é a classe fachada que esconde a complexidade do framework e de seus subsistemas, fornecendo uma interface mais simples. Essa classe possuirá, por exemplo, métodos (como o `process` do exemplo) que farão o tratamento e a organização necessários para as funcionalidades do framework.

Por fim, a classe *Application* representa a aplicação em si, que não dependerá diretamente de diversas classes fornecidas pelo framework complexo. Ao invés disso, a classe se comunicará com a fachada.

### Aplicabilidade

- Uma vantagem é que, se decidir trocar de subsistema (por exemplo, trocar de um framework para outro de mesmo propósito), será necessário basicamente reescrever a classe fachada, minimizando o esforço de alterações na aplicação;
- Quando se precisa ter uma interface limitada e simples para um subsistema complexo, e
- Quando quer estruturar um subsistema em camadas.

### Relação com outros padrões

*Singleton*: uma classe fachada pode frequentemente ser transformada em uma *singleton*, já que um único objeto fachada é suficiente na maioria dos casos.

## B.6 Exercícios - Facade

1. Sobre o padrão de projeto *Facade*, assinale a alternativa CORRETA:
  - É usado especificamente na programação estruturada de sistemas
  - O padrão não pode ser usado quando há interesse em dividir seus subsistemas em camadas
  - **Define uma interface comum para um conjunto de interfaces em um subsistema**
  - Promove forte acoplamento entre os subsistemas e seus clientes
  - Evita que aplicações acessem as subclasses diretamente
2. Um dos principais objetivos do padrão *Facade* é isolar a complexidade de um subsistema, seja ele uma biblioteca, framework ou outro conjunto de classes:
  - **Verdadeiro**
  - Falso
3. O padrão de projeto *Facade*, que fornece uma interface unificada para um conjunto de interfaces em um subsistema, é classificado como um padrão de projeto:
  - Criacional, que fornece um mecanismo de criação de objetos mais flexível, ajudando a tornar o sistema independente da forma como os objetos são criados, compostos e representados;
  - **Estrutural, que define como montar objetos e classes em estruturas maiores, ou**
  - Comportamental, que cuida da comunicação eficiente e da atribuição de responsabilidade entre objetos.

## B.7 Conteúdo - Template Method

O *Template Method* é um padrão comportamental que define o esqueleto de um algoritmo na superclasse, mas permite que as subclasses sobrescrevam etapas específicas do algoritmo sem modificar sua estrutura.

O padrão sugere que se divida um algoritmo em etapas e transforme essas etapas em métodos, colocando uma série de chamadas para esses métodos dentro de um único método padrão. As etapas podem ser tanto abstratas quanto ter alguma implementação padrão. Para usar o algoritmo, o cliente deve fornecer sua própria subclasse, implementar todas as etapas abstratas e sobrescrever algumas das opcionais se necessário (mas não o próprio método padrão).

No exemplo, na classe base abstrata *DomesticActivities*, temos o método padrão (*doDailyActivities*), que chama as etapas do algoritmo; método abstrato (*washClothes*), que deve ser implementado por cada subclasse; e métodos que já possuem alguma implementação padrão (*cleanHouse*, *washDishes*), mas ainda podem ser sobrescritos se necessário. Além de sobrescrever alguns métodos, as subclasses ainda implementam métodos próprios, como *tidyBackyard* e *washRug*.

Figura 69 – Exemplo de Código - *Template Method*

```
1 public abstract class DomesticActivities {
2     final void doDailyActivities () {
3         cleanHouse () ;
4         washDishes () ;
5         washClothes () ;
6     }
7     void cleanHouse () {
8         System.out.println ( " Mop the house. " ) ;
9     }
10    void washDishes () {
11        System.out.println ( " Wash the dishes. " ) ;
12    }
13    abstract void washClothes () ;
14 }
15
16 class WeekendActivities extends DomesticActivities {
17     void cleanHouse () {
18         System.out.println ( " Clean the house and bathrooms. " )
19         ;
19     }
20     void washDishes () {
21         System.out.println ( " Wash the dishes and the pans " ) ;
22     }
23     void washClothes () {
24         System.out.println ( " Wash used towels. " ) ;
25     }
26     void tidyBackyard () {
27         System.out.println ( " Clean backyard and mow the lawn. "
28         ) ;
28     }
29 }
30
31 class WeeklyActivity extends DomesticActivities {
32     void cleanHouse () {
33         System.out.println ( " Sweep and mop the house. " ) ;
34     }
35     void washDishes () {
36         System.out.println ( " Wash the dishes and tidying up the
37         kitchen " ) ;
37     }
38     void washClothes () {
39         System.out.println ( " Wash clothes and dirty bed linens.
40         " ) ;
40     }
41     void washRug () {
42         System.out.println ( " Wash the living room rug. " ) ;
43     }
44 }
```

## Aplicabilidade

- Quando se deseja deixar os clientes estender apenas etapas particulares de um algoritmo, mas não todo o algoritmo e sua estrutura, e
- Quando se tem várias classes que contêm algoritmos quase idênticos com algumas diferenças menores. Como resultado, pode-se querer modificar todas as classes quando o algoritmo muda.

## B.8 Exercícios - Template Method

1. O *Template Method* é utilizado para definir um esqueleto de um algoritmo em uma classe base, deixando que as subclasses implementem os detalhes específicos:
  - **Verdadeiro**
  - Falso
2. O *Template Method* é um padrão baseado em heranças:
  - **Verdadeiro**
  - Falso
3. O padrão de projeto *Template Method*, que define um esqueleto de algoritmo em uma classe para que as subclasses implementem etapas específicas, é classificado como um padrão de projeto:
  - Criacional, que fornece um mecanismo de criação de objetos mais flexível, ajudando a tornar o sistema independente da forma como os objetos são criados, compostos e representados;
  - Estrutural, que define como montar objetos e classes em estruturas maiores;
  - **Comportamental, que cuida da comunicação eficiente e da atribuição de responsabilidade entre objetos, tornando a implementação de comportamentos complexos mais gerenciável e modular.**

## B.9 Conteúdo - Factory Method

O *Factory Method* é um padrão criacional que, de acordo com Gamma et al, define uma interface para criar um objeto, mas deixa as subclasses decidirem qual classe instanciar. Permite adiar a instanciação para a subclasse. O padrão *Factory Method* traz a ideia de substituir chamadas diretas de construção de objetos por chamadas para um método “*factory*” especial.

Figura 70 – Exemplo de Código - *Factory Method* - Parte 1

```
1 public interface Transport {
2     void deliver();
3 }
4
5 public class Truck implements Transport {
6     public void deliver() {
7         System.out.println ( "Entrega por rodovias em
8             caixas" ) ;
9     }
10 }
11 public class Ship implements Transport {
12     public void deliver() {
13         System.out.println ( "Entrega por mar em
14             container" ) ;
15     }
16 }
17 public abstract class Logistics {
18     public void planDelivery() {
19         Transport transp = createTransport();
20         transp.deliver();
21     }
22     public abstract Transport createTransport();
23 }
24
25
26 public class RoadLogistics extends Logistics {
27     public Transport createTransport() {
28         return new Truck();
29     }
30 }
31
32 public class SeaLogistics extends Logistics {
33     public Transport createTransport() {
34         return new Ship();
35     }
36 }
```

Fonte: Autores, adaptado do site *Refactoring Guru*

Figura 71 – Exemplo de Código - *Factory Method* - Parte 2

```
1
2 * Exemplo de código cliente *
3
4 public class Application {
5     private static Logistics logistics;
6
7     public static void main(String[] args) {
8         prepare();
9         runLogistics();
10    }
11
12    static void prepare() {
13        if (args[0] == "road" | args[0] == "" ) {
14            logistic = new RoadLogistics();
15        } else {
16            logistic = new SeaLogistic();
17        }
18
19    static void runLogistics() {
20        logistic.planDelivery();
21    }
22 }
23 }
```

Fonte: Autores, adaptado do site *Refactoring Guru*

O *Transport* declara a interface, que é comum a todos os objetos que podem ser criados pelo *Logistics* e suas subclasses. *Truck* e *Ship* são implementações diferentes da interface *Transport*.

A classe *Logistics* declara o método fábrica que retorna novos objetos *Transport*. É importante que o tipo de retorno desse método corresponda à interface. O método fábrica pode ser declarado como abstrato para forçar todas as subclasses a implementar suas próprias versões do método.

*RoadLogistics* e *SeaLogistics* sobrescrevem o método fábrica base para retornar um tipo diferente de transporte.

### Aplicabilidade

- Quando não souber de antemão os tipos e dependências exatas dos objetos com os quais seu código deve funcionar;
- Quando desejar fornecer aos usuários da biblioteca ou *framework* uma maneira de estender seus componentes internos, e

- Quando desejar economizar recursos do sistema reutilizando objetos existentes em vez de recriá-los sempre.

## B.10 Exercícios - Factory Method

1. O *Factory Method* é definido corretamente como um padrão de projeto que:
  - Garante que apenas uma instância de uma classe seja criada e fornece um ponto global de acesso a essa instância
  - Facilita a comunicação entre objetos de diferentes classes
  - Permite a criação de múltiplas instâncias de uma classe.
  - **Define uma interface para criar um objeto, mas deixa as subclasses decidirem qual classe instanciar**
2. No *Factory Method*, o método “*factory*” (fábrica) fica dentro da classe criadora, e deve obrigatoriamente ser abstrato:
  - Verdadeiro
  - **Falso**
3. Uma das principais vantagens do padrão *Factory Method* é evitar o acoplamento entre o método criador abstrato e os objetos concretos, além de o código de criação ficar em um único local do programa, facilitando a manutenção do código:
  - **Verdadeiro**
  - Falso

## B.11 Conteúdo - Adapter

O *Adapter* é um padrão estrutural que converte a interface de uma classe em outra interface, permitindo que objetos com interfaces incompatíveis colaborem entre si. O *adapter* é um objeto que converte a interface de um objeto para que outro objeto possa entendê-lo, encobrendo a complexidade da conversão.

Suponha a comunicação entre dois objetos. Com esse padrão de projeto *Adapter*, um objeto existente chama os métodos do *adapter* de acordo com a interface. Ao receber a chamada, o *adapter* repassa a chamada para o segundo objeto em um formato e uma ordem que o segundo objeto espera. Também é possível desenvolver um *adapter* de duas vias, que pode adaptar as chamadas nos dois sentidos, do objeto 1 para o objeto 2 e vice-versa.



## Adaptador de objeto

Figura 72 – Exemplo de Código - *Adapter*

```
1 public class Client {
2 }
3
4 public class Target {
5     public void method(data) {
6     }
7 }
8
9 public class Adaptee {
10     public void utilMethod(specialData) {
11     }
12 }
13
14 public class Adapter extends Target {
15     private Adaptee adaptee
16
17     public void method(data) {
18         specialData = convertToAdapteeFormat(data);
19         return adaptee.utilMethod(specialData);
20     }
21 }
```

Fonte: Autores, adaptado do site *Refactoring Guru*

*Client* é uma classe que contém a lógica de negócio do programa existente. A classe *Target* é a interface do cliente, que descreve um protocolo que outras classes devem seguir para ser capaz de colaborar com o código cliente.

A classe *Adaptee* é alguma classe útil, a qual o cliente não pode usar diretamente porque ela tem uma interface incompatível.

O *Adapter* é uma classe que é capaz de trabalhar tanto com o cliente quanto a classe *Adaptee*: ela implementa a interface do cliente enquanto encobre o objeto da classe *Adaptee*. O *adapter* recebe chamadas do cliente através da interface do cliente (*Target*) e as traduz em chamadas para o objeto encoberto do *Adaptee* em um formato que ele possa entender.

O código cliente não é acoplado à classe concreta do *adapter* desde que ele trabalhe com o *adapter* através da interface do cliente. Graças a isso, pode-se introduzir novos tipos de *adapter* no programa sem quebrar o código cliente existente. Isso pode ser útil quando a interface de uma classe de serviço é mudada ou substituída: você pode apenas criar uma nova classe *adapter* sem mudar o código cliente.

## Adaptador de Classe

A implementação de um adaptador de classe utiliza herança, onde o adaptador herda interfaces de ambos os objetos ao mesmo tempo. A classe *Adapter* não precisa encobrir quaisquer objetos porque ela herda os comportamentos tanto do cliente quanto da outra classe incompatível. A adaptação acontece dentro dos métodos sobrescritos. O *adapter* resultante pode ser usado em lugar de uma classe cliente existente.

### Aplicabilidade

- Quando quiser utilizar uma classe existente, mas sua interface não for compatível com o resto do código, e
- Quando quer reutilizar diversas subclasses existentes que não possuam alguma funcionalidade comum, a qual, portanto, não pode ser adicionada à superclasse.

## B.12 Exercícios - Adapter

1. Assinale a alternativa INCORRETA sobre o padrão de projeto *Adapter*:
  - É um padrão estrutural que converte a interface de uma classe em outra interface, permitindo objetos com interfaces incompatíveis colaborarem entre si.
  - Pode ser implementado em duas vias, adaptando a classe X para a classe Y e vice-versa.
  - **Só pode ser utilizado entre uma classe do programa e uma classe externa, ou seja, de uma biblioteca ou *framework*.**
  - Pode ser implementado tanto um adaptador de objeto quanto um adaptador de classe.
2. João está trabalhando em um projeto de software que trata dados. Uma classe de seu programa recebe dados em formato XML, mas João possui uma outra classe de análise de dados que trabalha apenas com o formato JSON. Dessa forma, o padrão de projeto *Adapter* poderia ser utilizado como um adaptador do formato de dados no programa. Qual dos seguintes critérios NÃO É relevante para a decisão de utilizar ou não o padrão *adapter* nesse contexto?
  - Verificar se o programa possui classes de interfaces incompatíveis, mas que precisam colaborar entre si.
  - **Verificar se é possível subdividir o algoritmo da classe de análise de dados em etapas.**
  - Verificar se uma ou mais classes do programa seriam beneficiadas pelo uso da classe de análise de dados.

- Verificar se João possui acesso ao código da classe de análise de dados, e se a mesma possui muitas dependências existentes.

## B.13 Exercícios - Desafio Nível 1

1. Suponha que deseja-se criar uma classe com apenas uma instância disponível para todos os clientes, como, por exemplo, um objeto único de base de dados, compartilhado por diferentes partes do programa. Para esse contexto, qual seria a melhor opção de padrão de projeto para ser utilizado, para que não sejam criadas várias instâncias dessa mesma classe?
  - Façade
  - **Singleton**
  - Template Method
  - Decorator
2. Os padrões de projeto são separados em categorias. A categoria de padrões estruturais é responsável por:
  - fornecer maneiras eficientes de criar objetos
  - **descrever como os objetos são colocados juntos**
  - distribuir responsabilidades entre os objetos
3. Assinale a alternativa que apresenta um padrão de projeto classificado como estrutural:
  - Template Method
  - Singleton
  - Factory Method
  - **Facade**
4. O Singleton é corretamente definido como um padrão de projeto que:
  - **garante que apenas uma instância de uma classe seja criada e fornece um ponto global de acesso a essa instância**
  - permite a criação de múltiplas instâncias de uma classe
  - facilita a comunicação entre objetos de diferentes classes
  - facilita a criação de herança múltipla em linguagens de programação que não a suportam nativamente

## B.14 Exercícios - Desafio Nível 2

1. Considerando a implementação de código a seguir, qual padrão de projeto melhor a representa?

Figura 73 – Código do Exercício - Desafio

```
1 public class Servico {
2     private static Servico servico;
3
4     private Servico() {
5     }
6
7     public static synchronized Servico getInstance () {
8         if ( servico== null )
9             servico = new Servico() ;
10
11         return servico;
12     }
13 }
```

Fonte: Autores, adaptado de (GAMMA et al., 1995)

- **Singleton**
  - Facade
  - Template Method
  - Factory Method
2. Qual alternativa apresenta a sequência correta sobre as afirmativas a seguir, sendo V para verdadeira e F para falsa?
    - Template Method tem por objetivo definir o esqueleto de um algoritmo em uma operação, postergando a definição de alguns passos para subclasses.
    - O padrão de projeto Facade garante que determinada classe só seja instanciada uma única vez, e estabelece um ponto de acesso global a ela.
    - O padrão de projeto Singleton fornece um ponto por meio do qual um objeto controla o acesso a outro e as mudanças ao objeto instanciado acontecem no momento de execução ou interrupção.
    - O padrão Adapter converte a interface de uma classe em outra interface, permitindo objetos com interfaces incompatíveis colaborarem entre si.
- F - F - V - F
  - **V - F - F - V**

- V - V - F - V
  - V - F - V - F
3. Considerando os padrões Adapter e Facade, assinale a alternativa INCORRETA:
- Ambos são padrões estruturais
  - O Adapter busca fazer uma interface existente ser utilizável por outra interface incompatível a ela
  - **O Adapter trabalha com subsistemas de objetos, enquanto o Facade envolve apenas um objeto**
4. Nos padrões de projeto, são exemplos de padrões criacionais, de padrões estruturais e de padrões comportamentais, respectivamente:
- Adapter, Factory Method, Template Method
  - **Singleton, Adapter, Template Method**
  - Factory Method, Singleton, Facade
  - Facade, Factory Method, Adapter