



UNIVERSIDADE FEDERAL DE SERGIPE
PRÓ-REITORIA DE PÓS-GRADUAÇÃO E PESQUISA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Introdução a Programação, Aprendizagem Móvel e Colaborativa: Uma Abordagem Construcionista no Ensino Fundamental e Superior

Dissertação de Mestrado

Elisrenan Barbosa da Silva



São Cristóvão – Sergipe

2021

UNIVERSIDADE FEDERAL DE SERGIPE
PRÓ-REITORIA DE PÓS-GRADUAÇÃO E PESQUISA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Elisrenan Barbosa da Silva

**Introdução a Programação, Aprendizagem Móvel e
Colaborativa: Uma Abordagem Construcionista no Ensino
Fundamental e Superior**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Sergipe como requisito parcial para a obtenção do título de mestre em Ciência da Computação.

Orientador(a): Kalil Araujo Bispo
Coorientador(a): Daniel Oliveira Dantas

São Cristóvão – Sergipe

2021

Elisrenan Barbosa da Silva

Introdução a Programação, Aprendizagem Móvel e Colaborativa: Uma Abordagem Construcionista no Ensino Fundamental e Superior/ Elisrenan Barbosa da Silva. – São Cristóvão – Sergipe, 2021-

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

Orientador: Kalil Araujo Bispo

Dissertação de Mestrado – UNIVERSIDADE FEDERAL DE SERGIPE
PRÓ-REITORIA DE PÓS-GRADUAÇÃO E PESQUISA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO, 2021.

1. Palavra-chave1: Construcionismo. 2. Palavra-chave2: Mobile Learning. I. Orientador: Kalil Bispo Araujo. II. Universidade Federal de Sergipe. III. Título: Introdução a Programação, Aprendizagem Móvel e Colaborativa: Uma Abordagem Construcionista no Ensino Fundamental e Superior

CDU 02:141:005.7

Elisrenan Barbosa da Silva

Introdução a Programação, Aprendizagem Móvel e Colaborativa: Uma Abordagem Construcionista no Ensino Fundamental e Superior

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Sergipe como requisito parcial para a obtenção do título de mestre em Ciência da Computação.

Trabalho aprovado. São Cristóvão – Sergipe, 30 de Agosto de 2021:

Kalil Araujo Bispo
Orientador

Daniel Oliveira Dantas
Coorientador

Bruno Otavio Piedade Prado
Convidado Interno

Alberto Costa Neto
Convidado Externo

São Cristóvão – Sergipe
2021



UNIVERSIDADE FEDERAL DE SERGIPE
PRÓ-REITORIA DE PÓS-GRADUAÇÃO E PESQUISA
COORDENAÇÃO DE PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Ata da Sessão Solene de Defesa da Dissertação do
Curso de Mestrado em Ciência da Computação-UFS.
Candidato: **ELISRENAN BARBOSA DA SILVA**

Em 30 dias do mês de agosto do ano de dois mil e vinte um, com início às 09h00min, realizou-se na Sala virtual <https://meet.google.com/uhy-kfwg-qwf>. A Sessão Pública de Defesa de Dissertação de Mestrado do candidato **ELISRENAN BARBOSA DA SILVA**, que desenvolveu o trabalho intitulado: “**Introdução a Programação, Aprendizagem Móvel e Colaborativa: Uma Abordagem Construcionista no Ensino Fundamental e Superior**”, sob a orientação do Prof. Dr. **Kalil Araujo Bispo**. A Sessão foi presidida pelo Prof. Dr. **Kalil Araujo Bispo** (PROCC/UFS), que após a apresentação da dissertação passou a palavra aos outros membros da Banca Examinadora, Prof. Dr. **Alberto Costa Neto** (Dcomp/UFS), logo depois passou a palavra para o Prof. Dr. **Bruno Otávio Piedade Prado** (PROCC/UFS) e em seguida, ao Prof. Dr. **Daniel Oliveira Dantas** (PROCC/UFS). Após as discussões, a Banca Examinadora reuniu-se e considerou o(a) mestrando(a) aprovado “(aprovado/reprovado)”. Atendidas as exigências da Instrução Normativa 01/2017/PROCC, do Regimento Interno do PROCC (Resolução 67/2014/CONEPE), Resolução nº 25/2014/CONEPE e da Portaria nº 413 de 27 de maio de 2020 (Banca por videoconferência) que regulamentam a Apresentação e Defesa de Dissertação, e nada mais havendo a tratar, a Banca Examinadora elaborou esta Ata que será assinada pelos seus membros e pelo mestrando.

Cidade Universitária “Prof. José Aloísio de Campos”, 30 de agosto de 2021.

gov.br
Documento assinado digitalmente
Kalil Araujo Bispo
Data: 09/09/2021 17:54:26-0300
Verifique em <https://verificador.iti.br>

Prof. Dr. Kalil Araujo Bispo
(PROCC/UFS)
Presidente

Prof. Dr. Daniel Oliveira Dantas
(PROCC/UFS)
Co-orientador

Prof. Dr. Bruno Otávio Piedade Prado
(PROCC/UFS)
Examinador Interno

gov.br
Documento assinado digitalmente
Bruno Otavio Piedade Prado
Data: 30/08/2021 15:43:21-0300
Verifique em <https://verificador.iti.br>

Prof. Dr. Alberto Costa Neto
(DCOMP/UFS)
Examinador Externo

gov.br
Documento assinado digitalmente
Alberto Costa Neto
Data: 30/08/2021 14:35:54-0300
Verifique em <https://verificador.iti.br>

Elisrenan Barbosa da Silva
Candidato

gov.br
Documento assinado digitalmente
Elisrenan Barbosa da Silva
Data: 15/09/2021 23:37:27-0300
Verifique em <https://verificador.iti.br>

Dedico essa dissertação a minha mãe, mulher, família, amigos e meus orientadores que me deram o apoio necessário para chegar até aqui.

Agradecimentos

Nenhuma dessas linhas escritas nesta dissertação seriam possíveis sem as figuras de duas pessoas, meu orientador Kalil Araujo e meu coorientador Daniel Oliveira que me auxiliaram com muita dedicação e paciência, meus eternos agradecimentos.

A minha mãe Eliane Silva, por ter se esforçado e se dedicado sempre a mostrar o caminho que eu deveria seguir, me ensinando a lutar com toda alma e coração pelos meus objetivos e a nunca aceitar a derrota, o meu muito obrigado, sempre vou te amar.

O meu muito obrigado também a minha doce e eterna paixão, Edilene Carvalho, que não deixou jamais eu desistir, sempre me apoiando e ficando ao meu lado quando eu estava fraco, cansado e desmotivado, eu te amo.

Queria deixar aqui registrado alguns nomes que me ajudaram nessa caminhada, deveras cansativa e difícil. Assim, agradeço as professoras Jane Consuelo, Ana Beatriz e Helen Kaline por todo o apoio durante as aulas com as crianças. E não poderia deixar de fora a minha atual mentora, a professora Conceição Vieira, uma mulher com quem eu aprendo todos os dias.

Por fim, e não menos importante, meu agradecimento eterno a cada aluno que passou por mim durante esses dois anos e meio, meu muito obrigado a minha família, amigos e a cada pessoa que contribuiu, mesmo que um pouco, para o meu crescimento.

*Você não pode ensinar as pessoas tudo o que elas precisam saber.
O melhor que você pode fazer é posicioná-las onde elas podem encontrar
o que elas precisam saber quando precisam saber.
A programação é o meio mais poderoso do desenvolvimento
do pensamento sofisticado e rigoroso necessário
para a matemática, para a gramática, para a física, por estatísticas,
para todos os assuntos.
(Seymour Papert)*

Resumo

O *m-learning* ou aprendizagem móvel cresceu nos últimos anos de forma significativa, sendo abordado em escolas, na cultura, no meio ambiente, nas empresas e até em áreas rurais e isso em todo o mundo. O *m-learning* se caracteriza pela utilização de aparelhos portáteis, em especial *smartphones* ou *tablets*, possibilitando ao professor expandir o ensino para além da sala de aula, e também dando para os alunos uma maior acessibilidade de recursos educacionais.

Em um cenário onde a computação está em todos os lugares, pautado pela mobilidade global das pessoas, conectividade de internet ubíqua e uma maior independência de dispositivos móveis com relação ao ambiente computacional tradicional, os usuários conseguem acessar qualquer recurso em qualquer lugar e em qualquer momento. Isto mostra o quanto o *m-learning* pode ser utilizado tanto no ensino a distância como no contexto presencial, pois o processo de ensino aprendizagem não ocorre apenas enquanto o aluno está em sala de aula.

O grande problema dos pesquisadores brasileiros no que diz respeito a como inserir o ensino de computação no Brasil é de como colocar mais uma matéria em uma grade curricular tão apertada, e fazer isso utilizando poucos recursos financeiros para uma aplicação tecnológica como a que há em países desenvolvidos. Porém, pouco se fala sobre a utilização do *m-learning* no ensino de introdução a programação.

Esta dissertação tem como objetivo mostrar como utilizar dispositivos móveis (m-learning) como *smartphones*, *tablets* e *notebooks* no ensino de introdução a programação, usando como metodologia de ensino a abordagem construcionista de Seymour Papert, um método de ensino onde o professor tem como responsabilidade acompanhar cada aluno, identificar cada dificuldade individual e criar situações nas quais os alunos consigam se desenvolver tanto sozinhos como de forma coletiva.

Assim, durante dois anos abordamos o m-learning na educação básica antes da pandemia com 19 alunos e no ensino superior com 103 alunos e ambos da rede pública de ensino. Com as crianças utilizamos a linguagem ScratchJr para tablets e com os jovens a linguagem Python que é multiplataforma.

Palavras-chave: Aprendizagem móvel, m-learning, mobile learning, ensino de programação, algoritmos, construcionismo.

Abstract

M-learning or mobile learning has grown significantly in recent years, being addressed in schools, in culture, in the environment, in companies and even in rural areas, and this all over the world. M-learning is characterized by the use of portable devices, especially smartphones or tablets, enabling the teacher to expand teaching beyond the classroom, and also giving students greater accessibility to educational resources.

In a scenario where computing is everywhere, driven by the global mobility of people, ubiquitous internet connectivity and greater independence from mobile devices compared to the traditional computing environment, users can access any resource anywhere and anytime. This shows how much m-learning can be used both in distance learning and in the face-to-face context, as the teaching-learning process does not only occur while the student is in the classroom.

The big problem for Brazilian researchers with regard to how to introduce computer education in Brazil is how to put one more subject in such a tight curriculum, and do this using few financial resources for a technological application like that which exists in countries developed. However, little is said about the use of m-learning in teaching introductory programming.

This dissertation aims to show how to use mobile devices (m-learning) such as smartphones, tablets and notebooks in teaching introduction to programming, using as teaching methodology Seymour Papert's constructionist approach, a teaching method where the teacher is responsible for accompanying each student, identifying each individual difficulty and creating situations in which students can develop both alone and collectively.

Thus, for two years, we addressed m-learning in basic education before the pandemic with 19 students and in higher education with 103 students, both from the public school system. With children we use the ScratchJr language for tablets and with young people the Python language, which is cross-platform.

Keywords: Mobile learning, m-learning, mobile learning, programming teaching, algorithms, constructionism.

Lista de ilustrações

Figura 1 – Funções do ScratchJr	37
Figura 2 – Labirinto 1 e 2	39
Figura 3 – Labirinto 3 e 4	39
Figura 4 – Tetris	41
Figura 5 – Torre de Hanói	42
Figura 6 – Tangram	43
Figura 7 – Jogo da velha	43
Figura 8 – Quebra-cabeça do alfabeto	44
Figura 9 – Jogo do alfabeto baseado no ScratchJr	46
Figura 10 – Exercícios	46
Figura 11 – Avaliação 1 e 2	47
Figura 12 – Avaliação 3 e 4	48
Figura 13 – Avaliação 5 e 6	49
Figura 14 – Crianças realizando as atividades do labirinto	51
Figura 15 – Jogo centopeia robô	52
Figura 16 – Desenho da atividade descrição narrativa	53
Figura 17 – Primeiro contato dos alunos com o ScratchJr	54
Figura 18 – Atividades conectando as letras	55
Figura 19 – Inserindo Imagem no <i>tablet</i>	56
Figura 20 – Alunos fazendo atividades	58
Figura 21 – <i>Scripts</i> com bloco de início, movimento, aparência e finalização	59
Figura 22 – <i>Scripts</i> com bloco de início e movimento	60
Figura 23 – Um dos jogos da velha realizado pelos alunos	61
Figura 24 – Projetos	62
Figura 25 – Torre de Hanói sendo resolvida por um dos alunos	63
Figura 26 – Desafio 1 contra 1 torre de Hanói	64
Figura 27 – Projeto na praia com transição de cenários	65
Figura 28 – Tangram resolvido pelos alunos	65
Figura 29 – Quebra-cabeças resolvido pelos alunos	66
Figura 30 – Projeto na praia	67
Figura 31 – Tetris resolvidos pelos alunos	68
Figura 32 – <i>Scripts</i> que usaram a criatividade	69
Figura 33 – Provas didáticas aplicadas	70
Figura 34 – <i>Screenshots</i> de algumas telas do QPython	78

Figura 35 – <i>Screenshot</i> da uma das telas do <i>Replit</i>	83
Figura 36 – <i>Screenshot</i> da uma das telas do <i>Replit</i>	84
Figura 37 – <i>Screenshot</i> da organização dos materiais de aula no Classroom	85
Figura 38 – <i>Screenshot</i> de um grupo de alunos apresentando um código no Google Meet	86
Figura 39 – Resultados obtidos para o conteúdo de estrutura sequencial	109
Figura 40 – Resultados obtidos para o conteúdo de estrutura de decisão	110
Figura 41 – Resultados obtidos para o conteúdo de <i>string</i>	111
Figura 42 – Resultados obtidos para o conteúdo de repetição	112
Figura 43 – Resultados obtidos nas atividades de Lista	113
Figura 44 – Análise do resultado das turmas no mesmo período letivo	115
Figura 45 – Alunos programando em equipe no <i>QPython</i>	117
Figura 46 – Resultados obtidos para os primeiros programas	119
Figura 47 – Resultados obtidos para expressões	119
Figura 48 – Resultados obtidos para condicionais simples	120
Figura 49 – Resultados obtidos para condicionais compostas	121
Figura 50 – Resultados obtidos para <i>strings</i>	122
Figura 51 – Resultados obtidos para estrutura de repetição com <i>for</i>	123
Figura 52 – Resultados obtidos para estrutura de repetição com <i>while</i>	124
Figura 53 – Resultados obtidos para tuplas, listas e matrizes	125
Figura 54 – Resultados obtidos para dicionários, funções e tratamento de exceções	126
Figura 55 – Resultados obtidos com os projetos colaborativos	127
Figura 56 – Resultados de todas as turmas do semestre	131
Figura 57 – Código de um dos alunos no <i>QPython</i>	164
Figura 58 – Apresentação projeto crossfit	172
Figura 59 – Apresentação projeto disciplinas	173
Figura 60 – Apresentação projeto geticktes	174
Figura 61 – Apresentação projeto aulas particulares	176
Figura 62 – Apresentação projeto aulas particulares	177
Figura 63 – Apresentação projeto moto boy	177
Figura 64 – Grupo 3: apresentando o projeto estoque produtos empresa SenteQui	180
Figura 65 – Grupo 4: apresentando o projeto cadastro de viagem	181
Figura 66 – Grupo 5: apresentando o projeto concessionária	187
Figura 67 – Grupo 6 apresentado o código fonte de seu projeto	190
Figura 68 – Grupo 7 apresentado o código fonte de seu projeto	195
Figura 69 – Grupo 8: apresentado o projeto campeonato de varzea	197

Lista de tabelas

Tabela 1 – Ementa do projeto	33
Tabela 2 – Organização das atividades	74
Tabela 3 – Estratégia de ensino utilizada	88
Tabela 4 – Estratégia de ensino utilizada	93
Tabela 5 – Atividades dos ScratchJr de acordo com o nível de alfabetização	101
Tabela 6 – Atividades desplugadas	102
Tabela 7 – Assuntos ensinados no ScratchJr e na atividade desplugada	102
Tabela 8 – Atividades e a correlação com matéria	103
Tabela 9 – Saída formatada	164
Tabela 10 – Saída formatada com calculo do total da compra	165
Tabela 11 – O produto mais caro é:	165
Tabela 12 – O produto pesquisado é:	165
Tabela 13 – Times cadastrados	166
Tabela 14 – Tabela de classificação atualizada	166
Tabela 15 – O campeão é:	166
Tabela 16 – Formato de saída	167
Tabela 17 – Atualizar imc	167
Tabela 18 – Use a seguinte tabela como base para calcular o resultado	167
Tabela 19 – Maior imc	167
Tabela 20 – Formato de saída	168
Tabela 21 – Formato de saída	169
Tabela 22 – Formato de saída	169
Tabela 23 – Formato de saída	170
Tabela 24 – Formato de saída	170
Tabela 25 – Formato de saída	170
Tabela 26 – Exercício do dia	172
Tabela 27 – Entrada:	173
Tabela 28 – Entrada:	174
Tabela 29 – Saída	174
Tabela 30 – Saída:	179
Tabela 31 – Saída:	179

Lista de abreviaturas e siglas

AVA	Ambiente Virtual de Aprendizagem
EC	Estudo de Caso
EAD	Educação a Distância
IDEs	Integraged Development Environments
PC	Pensamento Computacional
QP	Questões de Pesquisa
TA	Turma
TIC	Tecnologia de Informação e Comunicação

Sumário

1	Introdução	19
2	Trabalhos relacionados	23
2.1	Aprendizagem móvel no ensino de Introdução a Programação	23
2.2	Ensino de programação na educação básica	28
2.3	Construcionismo de Papert	29
3	Metodologia no ensino fundamental	31
3.1	ScratchJr	36
3.2	Atividades baseadas em labirinto	38
3.2.1	Labirinto: leve o gatinho até a casinha	39
3.2.2	Centopeia robô	40
3.2.3	Descrição narrativa	40
3.3	Tetris	40
3.4	Torre de Hanói com três discos	41
3.5	Tangram	42
3.6	Jogo da velha	43
3.7	Quebra-cabeça do alfabeto e Conectando as letras	43
3.7.1	Quebra-cabeça do alfabeto	44
3.7.2	Conectando as letras	44
3.8	Avaliações	47
4	Aplicação no ensino fundamental	50
4.1	Aula 1 (Pré teste)	50
4.2	Aula 2 - 4 (Jogo centopeia robô)	52
4.3	Aula 5 (Jogo centopeia robô e descrição narrativa)	53
4.4	Aula 6 - 8 (Primeiro contato com o ScratchJr)	54
4.5	Aula 9 - 11 (Jogo conectando as letras)	55
4.6	Aula 12 - 14 (Pintando, desenhando e escrevendo no ScratchJr)	55
4.7	Aula 15 - 17 (ScratchJr)	56
4.8	Aula 18 - 20 (ScratchJr)	58
4.9	Aula 21 (Avaliação)	58
4.10	Aula 22 - 24 (Jogo da velha)	60
4.11	Aula 25 - 27 (ScratchJr - Projetos: gatinho feliz, João e Maria e livre)	61
4.12	Aula 28 (ScratchJr - Aula livre)	62
4.13	Aula 29 - 31 (Torre de Hanói)	62

4.13.1	Primeiro dia	63
4.13.2	Segundo dia	64
4.13.3	Terceiro dia	64
4.14	Aula 32 - 34 (ScratchJr - Projetos)	64
4.15	Aula 35 - 37 (Tangram)	65
4.16	Aula 38 (ScratchJr - Aula livre e quebra-cabeças)	66
4.17	Aula 39 - 42 (ScratchJr - Projetos)	67
4.18	Aula 43 - 45 (Tetris)	67
4.19	Aula 46 (ScratchJr - Aula livre)	68
4.20	Aula 47 - 50 (Avaliação)	68
5	Metodologia no ensino superior	72
5.1	Ensino presencial	72
5.1.1	QPython	77
5.1.2	WhatsApp	78
5.2	Ensino a distância	79
5.2.1	Replit	83
5.2.2	Google Classroom	84
6	Aplicação no ensino superior	87
6.1	Aplicação no ensino presencial	87
6.1.1	Projetos colaborativos	93
6.2	Aplicação EAD	93
6.2.1	Projetos colaborativos EAD	98
7	Resultados e discussão	100
7.1	Resultados no ensino fundamental	100
7.1.1	Categorizar as atividades de acordo com a adequação ao nível de alfabetização e a idade do aluno	100
7.1.2	Caracterizar quais os assuntos referentes ao ensino de programação os alunos conseguiram assimilar e executar	102
7.1.3	Foi possível integrar as atividades relacionadas a ciência da computação com o que era ensinado pela professora em sala de aula?	103
7.1.4	Quais as dificuldades encontradas pelos alunos ao serem ensinados a programar? E a linguagem escolhida facilitou o ensino de programação?	107
7.1.5	Ameaças à validade	108
7.2	Resultados no ensino superior presencial	108
7.2.1	QPI. Os alunos conseguiram aprender o conteúdo de estrutura sequencial? Quais foram as dificuldades encontradas?	109

7.2.2	QP2. Os alunos conseguiram aprender o conteúdo de estrutura de decisão? Quais foram as dificuldades encontradas?	110
7.2.3	QP3. Os alunos conseguiram aprender o conteúdo de <i>string</i> ? Quais foram as dificuldades encontradas?	111
7.2.4	QP4. Os alunos conseguiram aprender o assunto de estrutura de Repetição? Quais foram as dificuldades encontradas?	112
7.2.5	QP5. Os alunos conseguiram aprender o conteúdo de lista? Quais foram as dificuldades encontradas?	113
7.2.6	QP6. Os assuntos que os alunos mais tiveram dificuldade?	114
7.2.7	<i>Feedbacks</i> dos alunos	116
7.2.8	Ameaças à validade	117
7.3	Resultados no ensino superior EAD	118
7.3.1	QP1. Os alunos conseguiram aprender o conteúdo de estrutura sequencial? Quais foram as dificuldades encontradas?	118
7.3.2	QP2. Os alunos conseguiram aprender o conteúdo de estrutura de Decisão? Quais foram as dificuldades encontradas?	120
7.3.3	QP3. Os alunos conseguiram aprender o conteúdo de <i>string</i> ? Quais foram as dificuldades encontradas?	121
7.3.4	QP4. Os alunos conseguiram aprender o assunto de estrutura de repetição? Quais foram as dificuldades encontradas?	123
7.3.5	QP5. Os alunos conseguiram aprender o conteúdo de tuplas, listas e matrizes? Quais foram as dificuldades encontradas?	125
7.3.6	QP6. Os alunos conseguiram aprender o conteúdo de dicionários com listas compostas, uso de funções e tratamento de exceções? Quais foram as dificuldades encontradas?	126
7.3.7	QP7. O uso de projetos colaborativos facilitaram o desenvolvimento dos alunos no EAD?	127
7.3.8	QP8. A abordagem construcionista no EAD, facilitou o desenvolvimento dos alunos? E quais os principais pontos a serem destacados sobre a abordagem?	128
7.3.9	Ameaças à validade	132
7.4	Discussão	132
8	Conclusões	134
	Referências	137

Apêndices	144
APÊNDICE A Formulário parte 1 sobre o ensino de Python presencial	145
APÊNDICE B Formulário parte 2 sobre o ensino de Python presencial	148
APÊNDICE C Formulário parte 3 sobre o ensino de Python presencial	150
APÊNDICE D Formulário Final sobre o ensino de Python presencial	151
APÊNDICE E Formulário parte 1 sobre o ensino de Python EAD	153
APÊNDICE F Formulário parte 2 sobre o ensino de Python EAD	156
APÊNDICE G Formulário parte 3 sobre o ensino de Python EAD	158
APÊNDICE H Formulário Final sobre o ensino de Python EAD	160
APÊNDICE I Projetos colaborativos no ensino presencial	163
I.1 Grupo A: projeto rota transporte	163
I.2 Grupo B: projeto mercearia Itabaiana	164
I.3 Grupo C: projeto campeonato sergipano	165
I.4 Grupo D: projeto academia bem estar	166
I.5 Grupo E: projeto pizzaria paulistinha	167
I.6 Grupo F: projeto Formula 1	169
I.7 Grupo H: projeto aluguel de apartamentos	170
APÊNDICE J Projetos colaborativos no EAD	171
J.1 Grupo 1: Projetos colaborativos modelo 1	171
J.1.1 Treino crossFit	171
J.1.2 Disciplinas	172
J.1.3 GetTickets	173
J.2 Grupo 2: Projetos colaborativos modelo 1	174
J.2.1 Aulas particulares	175
J.2.2 Delta construtora	176
J.2.3 Moto boy	177
J.3 Grupo 3: Projetos colaborativos modelo 2	178
J.3.1 Estoque de produtos - Empresa SinteQui	178
J.4 Grupo 4: Projetos colaborativos modelo 2	180
J.4.1 Cadastro de viagem	180
J.5 Grupo 5: Projetos colaborativos modelo 2	184
J.5.1 Estoque de concessionária	184
J.6 Grupo 6: Projetos colaborativos modelo 2	189

J.6.1	Meu Delivery	189
J.7	Grupo 7: Projetos colaborativos modelo 2	191
J.7.1	Hortifruti delivery	191
J.8	Grupo 8: Projetos colaborativos modelo 2	195
J.8.1	Campeonato de várzea	195

1

Introdução

A atual ubiquidade dos dispositivos móveis, em especial dos smartphones, viabiliza cada vez mais o acesso das classes sociais C e D ¹ a recursos como a Internet (NIELSEN, 2015). Uma pesquisa realizada pela Telebrasil (2019) identificou que 70% da população brasileira tem acesso à Internet e que 97% dela a utiliza pelo *smartphone*, sendo que a maior taxa de crescimento desse uso está nas classes C, D e E. De acordo com a UNESCO (2014), muitos educadores em todo o mundo têm utilizado esses dispositivos móveis como ferramenta de ensino, buscando facilitar a aprendizagem com recursos inovadores.

Em uma sociedade em que o conhecimento, a mobilidade e a aprendizagem se tornam cada vez mais relevantes com pessoas cada vez mais conectadas principalmente por meio de dispositivos móveis, mesmo quando em movimento, de casa para o trabalho ou do trabalho para a faculdade (MOHAMMAD, 2015). Permite que pesquisas sejam realizadas a fim de utilizar esses recursos para melhorar a mobilidade urbana, a forma de trabalho e a educação.

Dentro dessa perspectiva, um novo modelo de ensino e aprendizagem vem se destacando, o *mobile learning* ou *m-learning*, que é apresentada na literatura como o uso da tecnologia móvel para ampliar as possibilidades de colaboração e comunicação das pessoas dentro do ambiente educacional. Existem três modalidades do *m-learning* que são: o uso de dispositivos móveis para acesso a informações, para realizar uma aprendizagem colaborativa e para o trabalho coletivo (HERRERA et al., 2015).

Dentro da computação essas modalidades do *m-learning* podem ser utilizadas em conjuntos com aplicativos que emulam *integrated development environments (IDEs)*, o que possibilita o ensino de programação principalmente para os alunos do primeiro período da graduação ou que ainda estejam no ensino médio (SHEARD et al., 2009) e (ORTIZ et al., 2015). Esses podem ser instalados em *smartphones* por possuírem ferramentas e capacidade de processamento similar à

¹ A divisão por classes é de acordo com a renda familiar, ou seja, divisão socioeconômica. Assim, cada grupo (alta, média e baixa) é caracterizado por letras: classe A, B, C, D e E.

de computadores pessoais.

Apesar da onipresença dos *smartphones*, seu uso é pouco explorado pelas instituições de ensino, fato que se apresenta como uma oportunidade de tentar introduzir uma prática educacional inovadora. Isso se torna possível, de acordo com [Borba e Lacerda Hannah \(2015\)](#), pois os *smartphones* já estão nas mãos dos alunos, o que possibilita que o ensino de programação seja acessível a pessoas e lugares sem recursos como computadores.

Isso mostra que a sociedade tem mudado e tem se tornado cada vez mais digital, sendo esta influenciada pelo desenvolvimento tecnológico mundial que traz consigo mudanças na forma como a todos devem interagir com as novas informações, seja em casa, no trabalho ou na escola estando assim imersas no uso da tecnologia tendo suas interações apenas voltadas para o consumo ([RESNICK, 2012](#)). Por esse motivo diversos países têm buscado meios de evoluir o nível educacional para transformar seus estudantes em pessoas fluentes em tecnologias e não apenas consumidoras ([Heintz; Mannila; Färnqvist, 2016](#)).

Com isso geram-se questionamentos de como os alunos podem se desenvolver em todas as disciplinas da sua grade curricular e ainda assim serem alfabetizados tecnologicamente desde o ensino fundamental até o ensino superior, bem como preparar os professores desde a graduação até a atuação em sala de aula ([ENGLISH, 2016](#)). De acordo com [Valente \(2016\)](#), os alunos serão os mais beneficiados com essa intervenção, pois eles irão fazer uso dos processos da computação como a decomposição e abstração para solucionar problemas e serem criativos independentes da idade ou da sua área de estudo, impulsionando o desenvolvimento tecnológico à medida que evoluem durante o seu processo educacional.

De acordo com [Brackmann \(2017\)](#), a computação tem influenciado diversas áreas e solucionado problemas complexos. O autor ainda reforça que os conhecimentos obtidos na computação são tão importantes quanto os adquiridos em Matemática, Português e Física, por exemplo. É, então, fundamental para o presente e para o futuro que cada indivíduo possua a habilidade de usar a tecnologia independente da sua área de formação ou de trabalho.

O desafio no Brasil consiste em como implementar o ensino de computação de forma que as pessoas aprendam não somente a teoria, mas que saibam utilizar também na prática do cotidiano em qualquer situação, e mais, com um custo baixo com relação a aparelhos tecnológicos e uma didática simples no que diz respeito ao ensino de programação e acessível a qualquer professor que não possua conhecimentos relacionados a computação, uma vez que 65% das escolas não possuem laboratório de informática e 44% delas não possuem acesso à Internet até antes da pandemia ([INEP, 2019](#)).

Uma outra pesquisa realizada pelo ([INEP, 2021](#)) mostra que durante a pandemia houve um aumento significativo de novas abordagens de ensino e capacitação por parte do professor para utilizar recursos de forma virtual, o estudo mostra ainda que as escolas aumentaram seus equipamentos tecnológicos como computador, notebooks, tablets e smartphones, aos docentes.

Outro ponto importante é de que todos os recursos de comunicação utilizados foram por e-mail, telefone, redes sociais e aplicativo de mensagem, o que demonstra em maior parte o uso de dispositivos moveis, visto que facilita acessibilidade.

Para obter a união desses conceitos mencionados anteriormente, precisamos entender primeiro como usar a computação no ambiente educacional, por meio do pensamento computacional (PC), que é responsável por abordar estratégias para se ensinar computação em conjunto com a grade curricular. Uma dessas estratégias usa o ensino de introdução à programação, envolvendo habilidades de realizar cálculos matemáticos, de ler e de escrever (WING, 2006).

Com o uso do PC é possível utilizar técnicas da Ciência da Computação para solucionar problemas do cotidiano de forma abstrata, dividindo-os em partes menores para melhor gerenciar a solução. Essa é uma habilidade necessária para o século XXI devido à grande quantidade de informações disponíveis. Assim, torna-se necessário utilizar alguma metodologia de ensino, em conjunto, que fortaleça o aprendizado de forma a potencializar o desenvolvimento de todos os alunos, uma delas é o construcionismo de Papert (1980), que recomenda a apresentação de problemas familiares aos alunos para estimular seu interesse, como também define que o professor deve realizar um acompanhamento individual para identificar as dificuldades específicas de cada um e fazer de tudo para saná-las.

Nesse aspecto, objetivo desse trabalho é mostrar como ensinar programação com o menor recurso tecnológico possível com uma abordagem de ensino totalmente baseada no uso de *smartphones* e *tablets*, para facilitar a compreensão dos alunos durante as aulas é utilizado algumas atividades desplugadas, ou seja, sem o uso de computadores.

No ensino fundamental, utilizamos como ferramenta para ensinar as crianças a programarem o ScratchJr para introduzir os assuntos aos alunos entre cinco e dez anos de idade por meio de tablets disponíveis na escola. Ademais, foram utilizadas atividades desplugadas de labirinto, descrição narrativa, montagem de figuras por meio do tetris, torre de Hanói, tangram, jogo da velha e atividades baseadas em quebra-cabeças.

No ensino superior foi utilizado a linguagem Python² como linguagem introdutória para aprender a programar, uma vez que para Beltran Sotomayor (2019) esta é um das mais utilizados no mundo. Aplicamos ela de duas formas, a primeira em um ambiente presencial antes da pandemia por meio do aplicativo para *smartphone* QPython e durante a pandemia no EAD por meio da plataforma Replit.

Em ambos os casos presencial ou EAD, utilizamos o aplicativo de comunicação WhatsApp para acompanhamento e interação durante e após as aulas, tendo em vista que, de acordo com Romero Moreno (2019), essa é uma das opções ideais para realizar um debate, tirar dúvidas, receber e enviar documentos.

No decorrer do projeto no ensino fundamental, foram aplicadas mais de 40 atividades

² <https://www.computer.org/publications/tech-news/trends/programming-languages-you-should-learn-in-2020>

e seis provas que envolviam os assuntos ensinados em cada módulo do ScratchJr e que foram resolvidos por 19 alunos. Com relação ao ensino superior antes da pandemia foram dados 135 problemas para serem resolvidos individualmente e 20 para serem resolvidos em equipe e os alunos desenvolveram projetos colaborativos em grupos criados e respondidos por eles dentro do que havia sido ensinado, no total de participantes foram de 103 alunos.

Além disso, apresentamos um material descritivo com informações sobre como aplicamos as atividades no dia a dia de aula no ensino fundamental. Apresentamos também como separamos os conteúdos dentro de cada aula no ensino superior e coletamos informações dos alunos por meio de questionários online, como também analisamos como os alunos se desenvolveram durante cada assunto dentro de sala de aula por meio dos exercícios e atividades respondidos por eles até o fim do período.

Essa dissertação é separada da seguinte forma: o capítulo 2 traz os trabalhos relacionados que embasaram toda a pesquisa, bem como as lacunas que poderiam ser mitigadas com aplicação desse estudo. No capítulo 3 é mostrado a metodologia do ensino fundamental e no capítulo 4 a sua respectiva aplicação. Já no capítulo 5 é mostrado a metodologia do ensino superior e no capítulo 6 a sua respectiva aplicação. Por fim, apresentamos no capítulo 7 os resultados e discussão desse trabalho e sua conclusão no capítulo 8.

2

Trabalhos relacionados

A grande quantidade de *smartphones*, que já ultrapassa mais da metade do número de habitantes no Brasil (FGV, 2021), permite a acessibilidade de aprendizagem para todos. Visto que quanto maior for o número de aparelhos, maior será a competitividade em relação ao preço, reduzindo os custos e tornando o produto mais acessível para todos. Os dispositivos móveis possuem uma grande capacidade de auxiliar o ensino de linguagens de programação no que diz respeito aos conteúdos introdutórios.

Nosso foco foi pesquisar a utilização do *mobile learning* no ensino de introdução à programação com uso dos dispositivos móveis tanto em sala de aula, quanto fora dela. Os alunos precisam realizar atividades utilizando uma *IDE* de programação e um aplicativo de mensagens os ajudarão na aprendizagem.

De acordo com a UNESCO (2014), a aprendizagem móvel pode ser definida como:

“A aprendizagem móvel como o uso de tecnologias móveis, isoladamente ou em combinação com outras tecnologias de informação e comunicação (TIC), a fim de permitir a aprendizagem a qualquer hora e em qualquer lugar. A aprendizagem pode ocorrer de várias formas: as pessoas podem usar aparelhos móveis para aceder recursos educacionais, conectar-se a outras pessoas ou criar conteúdos, dentro ou fora da sala de aula. A aprendizagem móvel também abrange esforços em apoio a metas educacionais amplas, como a administração eficaz de sistemas escolares e a melhor comunicação entre escolas e famílias” (p.8).

2.1 Aprendizagem móvel no ensino de Introdução a Programação

Esta subseção aborda o tema do *mobile learning* especificamente na área da programação de computadores, objeto desta dissertação. Os estudos foram selecionados por meio de um mapeamento sistemático, usando como critério de seleção trabalhos que apresentassem em seu

conteúdo alguma abordagem teórica ou prática do ensino de programação utilizando *smartphones* e/ou *tablets* que tenham sido publicados entre os anos de 2015 até o ano de 2021.

De acordo com [Silva et al. \(2015b\)](#), a aprendizagem móvel possibilita ao professor e ao aluno o acesso a recursos educacionais em qualquer lugar sem nenhuma restrição. Segundo [Nascimento e Filho \(2016\)](#), isso se dá devido à tecnologia estar cada vez mais presente na vida das pessoas, tornando o seu uso até mesmo indispensável.

Segundo [Carneiro e Marinho \(2018\)](#), o uso da *Internet* em dispositivos móveis cresceu exponencialmente no Brasil, alcançando pessoas de diversas classes sociais. No ano de 2016, 116 milhões de brasileiros conectados à Internet tinham idade acima de dez anos. Esse fato gera questionamentos de como o grande alcance ao dispositivo móvel pode ser inserido na escola, sem a perda de autonomia do professor em sala de aula. Dessa forma, foram investigados trabalhos que utilizaram o ensino semipresencial na escola ou que tenham utilizado aplicativos de mensagens no ambiente escolar para promover o ensino.

Já para [Marcolino et al. \(2018\)](#), os *smartphone* possuem uma grande capacidade de hardware, reduz o custo de suporte e por ser simples de ser manuseado facilita a sua adoção em todo o mundo. Porém, o quão eficaz será o seu uso em uma determinada matéria, irá depender do objeto de estudo em que será aplicado, pois, a depender disso, pode motivar ou desmotivar os alunos no processo de aprendizagem.

Procuramos trabalhos que disponibilizem informações sobre quais eram os problemas encontrados nas matérias de programação. Encontramos dois trabalhos: um é o de [Barbosa \(2016\)](#) e o outro refere-se ao de [Marcolino e Barbosa \(2017\)](#).

[Barbosa \(2016\)](#), classificou os problemas de programação em seis categorias por meio de um mapeamento sistemático. O primeiro, aprendizagem de programação, o segundo, aplicação de conceitos de programação, o terceiro, compreensão de programas. Essas três, segundo o autor, podem ser resolvidas por meio de soluções de visualização que utilizem *softwares*, ou vídeos para simular uma sequência de passos de um algoritmo até a sua solução. O quarto, programas de refatoração e fatoraçoão. O quinto, motivação dos alunos que segundo eles podem ser amenizados utilizando *Serious Games* (o uso de jogos para auxiliar na aprendizagem de programação) e Ambientes de Desenvolvimento Pedagógicos (o local que provê recursos para a criação e execução de programas). E o sexto, dificuldade do professor.

[Marcolino e Barbosa \(2017\)](#), buscaram investigar se existia algum consenso nos problemas relatados no Brasil quanto aos que foram relatados internacionalmente no trabalho de [Barbosa \(2016\)](#). Para tanto, eles realizaram uma pesquisa com professores e alunos brasileiros com o objetivo de identificar a relevância dos problemas internacionais no ensino superior brasileiro, identificar as principais soluções que têm sido adotadas nas disciplinas de programação, bem como suas limitações, e isso inclui soluções baseadas em *software*, como também coletar informações desses problemas na perspectiva dos alunos.

O resultado obtido por [Marcolino e Barbosa \(2017\)](#) foi um consenso entre os problemas para a maior parte das universidades, outros resultados obtidos foram sobre qual linguagem é a menos complexa, com menos estrutura de códigos e menos palavras reservadas. A linguagem mais simples foi o *Python*. Dentro dessa pesquisa, os alunos foram perguntados e responderam que, se pudessem, eles adotariam dispositivos móveis para aprender a programar. Nesse sentido, dentre os 14 que responderam ao questionário, 13 afirmaram de forma positiva o seu uso que, de acordo com eles, a justificativa gira em torno de que eles podem ser acessados e usados em qualquer lugar e a qualquer hora, já o outro participante que respondeu não, justificou que um computador é mais potente para desenvolvimento.

Ainda sobre o trabalho dos autores citados de [Marcolino e Barbosa \(2017\)](#), estes conseguiram verificar a partir da visão dos alunos quais foram as maiores dificuldades na visão deles, e 50% responderam que foram os assuntos de vetores e funções. Já os demais se dividiram em estruturas de repetição, estrutura condicionada e nenhuma dificuldade. Quando perguntados sobre o porquê de terem esses problemas, eles justificaram que isso se deu na aplicação da teoria à prática.

Com base nas informações coletadas, sobre como reduzir os problemas relacionados à programação no que diz respeito à motivação dos alunos, buscou-se identificar o cenário da adoção de aplicações de *mobile learning* e constatou-se que muitos estudos focam em dispositivos móveis, mas nenhum deles no ensino de programação como relatado no mapeamento sistemático de [Barbosa \(2015\)](#), em que o autor buscou identificar *softwares* educacionais para o ensino de programação.

Nesse sentido, é necessário investigar trabalhos que evidenciem de forma empírica os resultados relativos ao ensino de programação por meio de aplicativos de *m-learning*, ao realizar um mapeamento com a *string* de busca *teaching programming or m-learning or mobile learning and (mobile devices)* na base científica internacional *IEEE Xplore*, que contém diversas revistas, workshops e conferências entre o ano de 2015 a 2021. Foram encontrados 10 artigos, dos quais apenas três possuem informações sobre o tema que são: [Ortiz et al. \(2015\)](#), [Amro e Romli \(2019\)](#) e [Utesch et al. \(2020\)](#).

[Ortiz et al. \(2015\)](#) apresentam uma experiência baseada em *m-learning* no ensino superior, que utiliza uma aplicação web para a linguagem C e Java que pode ser programada em qualquer dispositivo via navegador. O objetivo do trabalho foi verificar o grau de aceitação que os 163 alunos dos cursos de Engenharia e Telecomunicações teriam ao utilizar essa versão. A aplicação foi testada durante o período letivo de 2012-2013 na Universidade Politécnica de Cartagena na Espanha e, por se tratar de uma aplicação desenvolvida há mais de dez anos e sem relatos encontrados de que foi atualizada, a sua usabilidade é baixa e pouco intuitiva para os alunos poderem programar. Contudo o foco foi em avaliar se a ferramenta possibilitou um melhor desempenho dos alunos, bem como analisar quais foram os dispositivos mais utilizados pelos estudantes e em quais lugares. Foi observado que mais de 70% dos alunos utilizaram

algum dispositivo móvel. Os alunos que participaram da experiência afirmaram que o uso dos dispositivos móveis no seu dia a dia educacional foi bastante positivo.

Amro e Romli (2019), buscaram identificar, dentre os aplicativos disponíveis no *Google Play e Apple Store*, alguns que pudessem ser utilizados para apoiar técnicas de programação. Eles realizaram um experimento controlado com quarenta e cinco alunos e três aplicativos: o Sololearn¹, Hub² e Mimo³ que possuem recursos para serem utilizados em aulas de programação. Na visão dos alunos, o Mimo foi a escolha ideal, por ter a melhor usabilidade e possibilita a utilização das melhores técnicas de programação de acordo com eles, o que é uma ótima alternativa e pode servir para ser usado de forma alternativa. No entanto, os três aplicativos funcionam como plataformas nos quais podem ser aprendidas uma ou mais linguagens de programação por meio de perguntas simples e respostas diretas de códigos. Deve-se ressaltar que, nesse estudo, não foi abordado nenhuma *IDE* completa que possibilite a criação de algoritmos desde o simples ao mais complexo. Como ponto positivo, as plataformas permitem que todos os conteúdos sejam centralizados em um só lugar, fáceis de ser acessados e os alunos podem praticar.

Utesch et al. (2020) desenvolveram um jogo sério para *Android Pic2Program*, que pode ser utilizado na fase pré-escolar. Foi feito um estudo piloto com 20 alunos, em sua grande maioria do 9º ano do ensino fundamental. Com o jogo é possível que os usuários conduzam seus personagens dentro de labirintos cada vez mais complexos. Além disso, o aplicativo permite que os jogadores criem novos labirintos através de desenhos permitindo assim a criação de programas básicos.

A fim de investigar pesquisas no cenário nacional, a mesma *string* de busca não obteve resultados. A *string* foi modificada para: “*m-learning*” OR “*mobile learning*” OR “*aprendizagem móvel*” entre os anos de 2015 a 2021. A busca na base dos Anais do Simpósio Brasileiro de Informática na Educação resultou em 16 artigos. Já nos Anais dos Workshops do Congresso Brasileiro de Informática na Educação foram oito artigos encontradas e na Revista Brasileira de Informática na Educação - RBIE obtivemos quatro artigos, dos quais não foram encontrados trabalhos próximos ou relativos ao ensino de programação via *smartphone* ou *tablets*.

Por fim, utilizamos a seguinte *string* de busca “*ensino de programação*” AND (*m-learning* OR “*mobile learning*” OR “*aprendizagem móvel*”) no Google Acadêmico entre os anos de 2015 a 2021, com o intuito de identificar trabalhos que não haviam sido encontrados nas bases mencionadas anteriormente. Obtivemos 194 resultados e destes se destacaram 4 trabalhos que utilizaram um ambiente móvel para o ensino de programação: André et al. (2017), Ribeiro e Brasileiro (2016), Carvalho, Aguiar e Dantas (2017) e (ANDRADE, 2019), os demais trabalhos encontrados tiveram seus focos em aplicações *web* como App Inventor e Scratch, sendo esses

¹ Link para visualização do app: <https://www.sololearn.com/home>

² Link para visualização do app: <https://programming-hub.br.uptodown.com/android>

³ Link para visualização do app: <https://mimo.br.uptodown.com/android>

artigos aplicados apenas em contextos com uso de computadores.

André et al. (2017) tiveram como foco o ensino do aplicativo ScratchJr, uma ferramenta de programação em blocos voltadas para crianças desde os quatro anos até os dez anos de idade. Eles relataram uma experiência de introdução à programação no ambiente escolar, em que participaram os pais e os alunos em um *Scratch Day* que é um dia para aprender brincado durante 130 minutos. Todos puderam criar histórias, animações interativas todas programadas em blocos. Os benefícios da aplicação é o desenvolvimento da criatividade e inovação. Nesse estudo de caso, o foco foi em introduzir os conceitos e não de analisar como o uso do aplicativo pode influenciar os resultados educacionais da turma.

Carvalho, Aguiar e Dantas (2017) abordaram, na disciplina de introdução à programação para 27 alunos, o uso de uma aplicação chamada *ARDisk* para celular. Esta aplicação utiliza realidade aumentada, com o objetivo de ensinar um único conteúdo que foi o de laços de repetição com *for* e o uso da função *range* em *Python*. Verificou-se que o uso da aplicação como apoio no processo de aprendizado foi eficaz e que é viável adotar essa aplicação como ferramenta de aula. Um ponto a ser observado é que o trabalho não utiliza uma *IDE* de programação e sim um aplicativo para explicar de forma lúdica o assunto em questão.

Ribeiro e Brasileiro (2016) propõe um ambiente híbrido onde os alunos de computação possam utilizar o mesmo servidor da universidade para programar em um computador, *notebook* ou *smartphones*. A proposta é de que não só os assuntos estejam em um ambiente AVA, como também os estudantes consigam manter seus códigos na nuvem e, conseqüentemente, consigam acessar de qualquer dispositivo. O artigo descreve apenas a proposta e não a implementação, portanto a análise se baseia apenas nisso.

Andrade (2019) utilizou o *m-learning* em um projeto na Universidade de Cabo Verde, na África Ocidental. Foi aplicado em duas turmas do 1º ano do curso de engenharia eletrotécnica, na disciplina de introdução à programação, uma de controle e outra experimental, divididos em três módulos. O primeiro módulo abordava conceitos teóricos sobre algoritmos com o objetivo de criação de fluxogramas e decomposição de problemas. No segundo módulo, foi utilizado no *moodle* para que os alunos pudessem programar na linguagem C em uma interface acessível tanto a partir do *smartphone* quanto do computador. No final, foi ensinado aos alunos como utilizar o *App Inventor* para a criação de aplicativos para *smartphones* por uma interface de programação baseada em blocos. Vale destacar que o projeto foi utilizado como complemento da matéria, sendo que não ficou clara a forma de avaliar se os estudantes absorveram ou não os assuntos. Porém, o foco do trabalho foi em analisar de forma estatística a coleta de respostas advindas de questionários e com os resultados finais de aprovação, reprovação e desistência em cada turma.

Face aos estudos analisados, identificamos que a literatura dentro do que foi pesquisado carece de trabalhos que abordem o ensino de programação por *smartphones* e/ou *tablets*, seja no nível fundamental, médio ou superior. Na subseção 2.2 é apresentado o estado da arte referente

ao ensino de programação na educação básica. Na seção 2.3 é apresentado o embasamento teórico da metodologia de ensino aplicada.

2.2 Ensino de programação na educação básica

O ensino de computação, desde os anos iniciais da educação básica, tem sido tema de pesquisas há diversos anos no Brasil. Estudos encontrados na literatura demonstram os problemas e as dificuldades encontradas pelos pesquisadores e exemplos de sucesso. Quanto ao ensino-aprendizagem da computação alinhada à grade curricular escolar, mesmo que muitas das vezes por projetos fora do horário escolar, é possível destacar que, mesmo que o projeto não seja realizado dentro das aulas, os resultados obtidos são satisfatórios no que diz respeito ao PC.

[Silva et al. \(2015a\)](#) apresentam os resultados sobre o ensino-aprendizagem no Brasil, quais ferramentas, metodologias e abordagens pedagógicas e níveis de ensino foram aplicadas. Os dados apresentados por eles mostram que o ensino interdisciplinar entre a Computação e as demais matérias da grade curricular escolar ainda é pouco utilizado e que apenas 11% dos trabalhos encontrados são referentes ao ensino fundamental.

[Souza, Batista e Barbosa \(2016\)](#) apresentam o seguinte problema de pesquisa: quais as maiores dificuldades dos alunos no processo de ensino-aprendizagem de programação? Para respondê-la, fizeram um mapeamento de 70 estudos em que foram encontradas as principais dificuldades para aprender os conceitos de programação, para aplicar esses conceitos durante a construção de programas e causas para a falta de motivação entre os alunos na realização da atividade de programação. Porém, não é especificado no artigo em qual nível escolar esses problemas são encontrados.

Dentro do contexto interdisciplinar, o ensino de programação é uma das tendências mundiais por meio da disciplina de pensamento computacional, que tem como base quatro pilares: decomposição, reconhecimento de padrões, abstração e algoritmos. Todas as técnicas juntas potencializam o conhecimento dos alunos para resolver problemas do cotidiano, além de fomentar o desenvolvimento do raciocínio lógico, como relatado por [Santos, Araujo e Bittencourt \(2018\)](#), [Brackmann et al. \(2016\)](#) e [Heintz, Mannila e Färnqvist \(2016\)](#).

[Wangenheim, Rodrigues e Santos \(2014\)](#) tiveram como objetivo ensinar a programar a partir do uso do Scratch de forma interdisciplinar com as disciplinas de literatura infantil e artes. Eles desenvolveram uma unidade instrucional para a criação de uma versão computacional da história infantil da Chapeuzinho Vermelho, durante 6 encontros de 1h. Os resultados encontrados demonstram que as crianças conseguiram assimilar todo o conteúdo e o Scratch estimulou a criatividade dos alunos. Quando comparado os horários utilizados ao presente estudo, verifica-se a similaridade no uso do Scratch e no que diz respeito a implementar o estudo de caso com uma das matérias da grade curricular.

Nunes et al. (2018) buscaram ensinar programação a alunos entre 7 e 8 anos de idade e tiveram como foco potencializar o pensamento criativo com uso do Scratch e de atividades desplugadas. Os resultados foram apresentados de forma dissertativa qualitativa por meio da análise feita no dia a dia e por um questionário realizado ao final do estudo de caso. Validou-se que é possível ensinar a programar e manter as crianças criativas, o que foi reiterado também no trabalho feito por Alves et al. (2016).

Brackmann et al. (2016) destacam alguns projetos privados não governamentais que trabalham a inserção da Ciência da Computação no cotidiano de alunos do ensino básico ao fundamental, por ensinar programação plugada ou desplugada, pensamento algorítmico e resolução de problemas. Os projetos são: [code.org](http://www.code.org/)¹ (EUA), [Programaê](http://www.programae.org.br/)² (Brasil), [Supergeeks](https://www.supergeeks.com.br/)³ (Brasil) e *Developing the Computational Thinking* (Chile). Todos esses trabalham em sua essência o desenvolvimento social, a inclusão digital e uma forma simples de aprender sobre computação.

Santana et al. (2017) realizaram seu projeto com 19 crianças de 5 anos de idade durante 130 minutos. Os resultados encontrados mesmo que em um curto espaço de tempo demonstram que é possível inserir o ensino de programação e manter o pensamento criativo das crianças. O outro é o de Farias et al. (2019), trata-se de um relato sobre como pais e filhos puderam aprender conceitos introdutórios de computação para estimular o raciocínio lógico. Vale ressaltar que os trabalhos vistos exploram o ensino interdisciplinar no máximo em duas matérias. Verificou-se também que a forma de análise para identificar se um aluno aprendeu a programar é observacional, ou seja, o pesquisador analisa as informações de forma qualitativa.

Foi possível identificar que a abordagem do pensamento computacional tem sido utilizada com foco em analisar o desenvolvimento dos alunos no aprendizado computacional, e que há uma carência muito grande no ensino de conceitos da Ciência da Computação aplicada interdisciplinarmente com as matérias da grade curricular estudantil. Também, nos trabalhos encontrados que objetivaram o ensino interdisciplinar, não foram achados registros sobre se a computação tem influenciado de forma direta ou indireta, positiva ou negativa, o processo de aprendizagem dos alunos nas disciplinas da sua grade curricular até o momento. Isso se confirma ao se analisar todos os projetos, relatos, pesquisas e experiências de diversos trabalhos publicados ou em andamento no Brasil feitas nas últimas décadas descritos no livro *Computação na Educação Básica* de Raabe, Zorzo e Blikstein (2020).

2.3 Construcionismo de Papert

O método de ensino que foi utilizado é o construcionismo de Seymour Papert, que define o aluno como protagonista do seu processo de aprendizagem. Em seu livro *A Máquina das Crianças*, é abordado como o sujeito gera conhecimento ao partir da prática do abstrato para a

¹ <http://www.code.org/>

² <http://www.programae.org.br/>

³ <https://www.supergeeks.com.br/>

construção do concreto, através do uso do computador (PAPERT, 2008). Ou seja, o professor é responsável por provocar situações em que o aluno aprende a aprender. As circunstâncias criadas podem abordar problemas relacionados a abstração, formulação de hipóteses, reconhecimento de padrões, decomposição dentre outras. Esses conteúdos devem valorizar a vivência diária de cada um (BEN-ARI, 1998).

De acordo com Papert (1980), o construcionismo faz com que os alunos aproveitem ao máximo as suas capacidades criativas, pois com esse método os alunos se sentem e se tornam parte do projeto, seja esse um jogo, um problema ou, até mesmo, uma brincadeira.

Para Papert, ao utilizar a abordagem construcionista em sala de aula, o professor se torna o mediador responsável por fomentar a aprendizagem dos seus alunos, trazendo atividades e conteúdo de acordo com a realidade ou experiência de cada um.

Ainda segundo Papert, o aluno é um ser pensante capaz de construir suas próprias estruturas cognitivas, ou seja, ele consegue aprender mesmo sem ser ensinado (PAPERT, 1986). Um exemplo da aplicação do construcionismo é no trabalho de Santos G. e Lima S. (2018), que aplicaram as cinco dimensões de aprendizagem proposta por Papert. Esse estudo foi realizado com alunos do 1º ano de um curso técnico de informática do ensino médio, sendo o recurso principal abordado o estudo de robótica. Para os autores, a metodologia baseada no construcionismo foi de grande relevância para o aprendizado, visto que os estudantes conseguiram adquirir conhecimento de forma autônoma.

3

Metodologia no ensino fundamental

Para suprir a carência de material sobre PC destinado ao ensino fundamental, propusemos uma cartilha de atividades com o uso do pensamento computacional para o desenvolvimento de habilidades como base, a saber: pensamento lógico, decomposição de problemas e criatividade que auxiliem as crianças no seu processo educacional. Buscamos criar uma metodologia de ensino que possa ser replicada por professores mesmo sem o conhecimento de conceitos de computação. A cartilha foi desenvolvida para utilizar o mínimo possível de recursos tecnológicos. Para tanto, apresentamos também uma métrica de seis avaliações para analisar o desenvolvimento acadêmico dentro do ensino de programação.

Estudos demonstram que o ensino de programação ajuda os alunos a lidarem com as diversas disciplinas de sua grade curricular. Programar exige pensar em diversas possibilidades de como resolver um determinado problema e ajuda a desenvolver a habilidade de lidar com as dificuldades do cotidiano (IOANNOU; MAKRIDOU, 2018). Todas as atividades que integram essa cartilha seguem um plano de aula detalhado sobre como, quando e em que aula aplicar, de forma a buscar sempre integrar o PC à realidade dos alunos.

Assim, o ambiente no qual o estudo de caso foi inserido é na educação básica, em uma escola pública do ensino fundamental I do município de Aracaju/SE, que já conta com o Projeto Aula Digital. Este projeto beneficia as escolas com kits tecnológicos (um notebook, um roteador, um mini projetor, uma tela de proteção e 34 *tablets*) como material de apoio. Estes equipamentos facilitam o acesso à tecnologia e a informação, desde o ano de 2018, já beneficiou 129 mil estudantes e preparou 4.690 educadores, que receberam formação especial para ensinar de forma inovadora em diferentes contextos utilizando esses recursos (VIVO, 2019).

O objetivo é apresentar uma cartilha de atividades com foco no ensino de programação e com uma abordagem interdisciplinar alinhada ao plano pedagógico de uma turma do 1º ano do ensino fundamental. Quanto à linguagem de programação, foi usada a ScratchJr para atividades com utilização do *tablet* e de atividades desplugadas para a implementação dos exercícios.

Para alcançar o objetivo deste projeto foi definido um problema de pesquisa: de que forma é possível ensinar crianças a programarem a partir do uso de um *tablet* em conjunto com atividades relacionadas à programação que podem auxiliar os alunos durante o processo educacional?

Desse modo, pretende-se com essa pesquisa alcançar os seguintes objetivos específicos:

1. Categorizar as atividades de acordo com a adequação ao nível de alfabetização e quanto à idade do aluno.
2. Caracterizar quais os assuntos referentes ao ensino de programação os alunos conseguiram assimilar e executar.
3. Verificar se é possível integrar o que é ensinado em uma linguagem de programação ao que é ensinado pela professora em sala de aula.
4. Identificar as dificuldades encontradas pelos alunos ao serem ensinados a programar.
5. Verificar se a linguagem escolhida facilitou o ensino de programação.

Foram desenvolvidos três jogos, o “labirinto: leve o gatinho até a casinha”, a “centopeia robô” e “conectando as letras”. Esses jogos abordam os conceitos de programação de forma implícita, sem focar em suas regras formais. Como complemento para trabalhar o desenvolvimento cognitivo, vamos utilizar o “Tetris”, a “Torre de Hanói”, o “Tangram”, o “jogo da velha” e “quebra-cabeça do alfabeto”. De acordo com Santos, Araujo e Bittencourt (2018), os jogos possuem a capacidade de fomentar o engajamento dos alunos nas disciplinas correlacionadas quando utilizados em sala de aula de forma interdisciplinar. Migliori (2013) declara em seu livro, Neurociências e educação, que o uso de uma abordagem implícita cria para o cérebro um ambiente propício para o desenvolvimento da memória de curto prazo, ou seja, isso gera experiências que favorecem o aprendizado do estudante.

Esta pesquisa foi feita principalmente de forma qualitativa. Foram feitas entrevistas com o(a) professor(a) responsável, analisados os programas e desenhos feitos pelos alunos no ScratchJr, relatos e comentários dos próprios alunos durante as aulas. Além disso, contou com alguns dados quantitativos referentes às provas que estamos propondo como métrica de avaliar o pensamento computacional. As avaliações foram realizadas pelos alunos e o tema foi a linguagem de programação escolhida, vide a subseção 3.8 sobre as atividades/avaliações.

Os benefícios para o aluno estão diretamente ligados ao processo de aprendizagem de cada um. É possível utilizar o PC como forma de estimular esse processo, isto é, trabalhando a criatividade por meio de ferramentas, linguagens de programação e atividades sem o uso da tecnologia (BRENNAN; RESNICK, 2012). Segundo Blikstein (2008), o uso do PC estimula o poder cognitivo e operacional dos indivíduos, pois o campo de situações que podem ser

trabalhados com esse instrumento é amplo, variando desde trabalhar a capacidade de análise, abstração e decomposição de um problema até a usar a computação na aprendizagem, que não é apenas aprender a programar, é também pensar, planejar e saber tomar decisões de forma clara e objetiva.

Na cartilha proposta, cada atividade é descrita e classificada de acordo com a grade curricular, o nível de alfabetização e os pilares do pensamento computacional. O planejamento das aulas segue abaixo na Tabela 1. Nela, é apresentado o que houve em cada encontro, bem como sua duração, o que foi ensinado sobre programação e qual a relação dessa aula com as matérias da grade curricular dos alunos. As aulas são totalmente construcionista e divididas em pequenos projetos para que o conhecimento seja adquirido em sua totalidade (WING, 2006), (RESNICK et al., 2009) e (MARTINS, 2012).

Tabela 1 – Ementa do projeto

Aula	Tempo	Atividades	Qual objetivo	Matéria correlacionada
1	45 min	Pré-Teste	Programação desplugada: leve o gatinho até a casinha usando setas: para baixo, para cima, para a direita ou para a esquerda	Matemática, Português e Computação
2 - 4	2h e 15 min	Jogo centopeia robô	Programação desplugada: abstração, decomposição, reconhecimento de padrões, estruturas de condição e repetição	Geografia, Matemática e Computação
5	45 min	Centopeia robô e desenhando seu trajeto: pedir aos alunos para desenharem o caminho que eles fazem de casa até a escola	Programação desplugada: descrição narrativa	Geografia, Artes, Matemática e Computação
6 - 8	90 min	Centopeia Robô e o primeiro contato com o ScratchJr:	Ser criativo: colocando o desenho que foi realizado na aula quatro que trata da trajetória dos alunos no aplicativo	Português, Matemática e Computação

9 - 11	2h 15 min	Centopeia robô e o jogo: conectando as letras	Programação desplugada: abstração, decomposição, reconhecimento de padrões, estruturas de condição e repetição	Português, Matemática e Computação
12 - 14	2h 15 min	Centopeia robô, pintando, desenhando e escrevendo no ScratchJr	Usar aleatoriamente: Faça um exemplo no aplicativo e deixe os alunos mexerem livremente	Artes, Português e Computação
15 - 17	2h e 15 min	Centopeia robô e ScratchJr: blocos de evento, movimento, criação, edição e remoção de personagem e cenário.	Abstração, decomposição, reconhecimento de padrões, estruturas de condição e repetição e interação com os personagens	Apenas Computação
18 - 20	2h e 15 min	Centopeia robô e ScratchJr: blocos de evento, de movimento, criação, edição e remoção de personagem e cenário, blocos de aparência e finalização	Abstração, decomposição, reconhecimento de padrões, estruturas de condição e repetição e interação com os personagens	Apenas Computação
21	45 min	Teste sobre o conhecimento de programação adquirido até o momento	Teste sobre os blocos de: inicialização, movimentação e finalização	Avaliação: apenas Computação
22 - 24	2h e 15 min	Jogo da velha	Abstração, decomposição, reconhecimento de padrões e estrutura de decisão	Matemática, Ed. Física e Computação
25 - 27	2h e 15 min	ScratchJr: Bloco de evento, de som, de movimento, de aparência e de finalização. Projetos feitos: gatinho feliz e João e Maria	Abstração, decomposição, reconhecimento de padrões, estruturas de condição e repetição e interação com os personagens	Apenas Computação

28	45 min	ScratchJr: Aula livre	Criatividade	Apenas Computação
29 - 31	2h e 15 min	Torre de Hanói com 3 peças	Algoritmos, abstração, decomposição, sequência, decisão e repetição	Matemática e Computação
32 - 34	2h e 15min	ScratchJr: Bloco de evento, de som, de movimento, de aparência e de finalização. Projeto feito: contando até 10 sem uso de cenários e de formas complexas	Abstração, decomposição, reconhecimento de padrões, estruturas de condição e repetição e interação com os personagens	Matemática e Computação
35 - 37	2h e 15 min	Tangram com 7 peças e 3 formas geométricas: quadrado, paralelogramo, 1 triângulo grande, 1 médio e 2 menores	Raciocínio lógico, reconhecimento dos tipos das figuras, rotação para encaixe e formar as figuras. Isso trabalhou abstração, decomposição, reconhecimento de padrões, estruturas de condição e repetição	Matemática e Computação
38	45 min	ScratchJr: Aula livre	Criatividade	Apenas Computação
39 - 42	2h e 15min	ScratchJr: bloco de evento, de movimento, de controle, de aparência, de som e de finalização.. Projeto feito: contando até 10 utilizando todos os recursos	Abstração, decomposição, reconhecimento de padrões, estruturas de condição e repetição e interação com os personagens	Matemática e Computação
43 - 45	2h e 15 min	Tetris desplugado com 4 linhas e 7 colunas	Raciocínio lógico, interpretação e resolução de problemas em relação a Computação trabalha a abstração e algoritmo	Matemática e Computação
46	45 min	ScratchJr: Aula livre	Criatividade	Apenas Computação

47 - 50	4h	Teste sobre o conhecimento de programação adquirido até o momento	Teste sobre os blocos do ScratchJr em uma avaliação desplugada	Avaliação: Ape- nas Computação
---------	----	---	--	--------------------------------

Fonte: Acervo pessoal

Nesse aspecto, a compreensão de uma linguagem de programação ajudou os alunos no desenvolvimento da criatividade. Testar possibilidades e hipóteses auxiliou também no desenvolvimento do pensamento científico. Utilizar o ScratchJr em conjunto com outras atividades como “Torre de Hanói”, “Tangram”, “Tetris”, “quebra-cabeça do alfabeto”, “conectando as letras” e “labirinto: leve o gatinho até a casinha”, permite que os estudantes apliquem todas as habilidades cognitivas listadas nessa pesquisa. Ademais, permitiu também um desenvolvimento mais eficaz, visto que cada uma dessas atividades foi trabalhada em paralelo às disciplinas da grade curricular. A seguir são descritas cada uma dessas atividades e seu respectivo embasamento teórico.

3.1 ScratchJr

O ScratchJr é disponibilizado gratuitamente pelo DevTech da Universidade Tufts e pelo MIT Media Lab. Esse aplicativo possibilita a criação de histórias, jogos e animações interativas, bem como o compartilhamento dessas criações. O ScratchJr instiga crianças e jovens a pensar de forma criativa, a raciocinar sistematicamente e a trabalhar colaborativamente. Ademais, trabalha ainda com os quatro pilares do pensamento computacional: decomposição, reconhecimento de padrões, abstração e algoritmos. Pode ser utilizado online ou instalado em um computador. Vale salientar que, além de ser um ambiente fácil e amigável, é possível realizar códigos dos mais simples aos mais complexos, permitindo assim que habilidades de colaboração, criatividade e raciocínio lógico sejam desenvolvidos (AMRO; ROMLI, 2019).

Essa linguagem de programação já foi utilizada no mundo inteiro e possui mais de 9,5 milhões de downloads. Seus criadores identificaram que a ferramenta tem sido bastante usada para a alfabetização, uma vez que ela foi projetada para que os alunos criem seus programas da esquerda para a direita, como a escrita e leitura das línguas ocidentais, como o português. Programas são um conjunto de instruções para que uma função seja executada. As crianças podem interagir inserindo texto, áudio, imagens e criando narrativas que expressem aquilo que estão pensando, de forma até mesmo criativa. Em muitos países ela é utilizada de forma integrada com algumas unidades de ensino como Teatro, Matemática e Estudos Sociais (Bers, 2018).

Segundo Flannery et al. (2013) existe uma infinidade de ferramentas para o ensino de programação que são voltadas para as crianças, assim como vários *softwares* e tecnologias destinadas aos jovens. O ScratchJr foi projetado para ser utilizado com crianças de idade menor

3. Expandir o cenário.
4. Ativar e desativar o modo de cenário e em linhas com coordenadas x-y.
5. Criar ou alterar cenário.
6. Adicionar texto ao cenário.
7. Reiniciar o *script*.
8. Iniciar o *script* ao clicar na bandeira verde.
9. Clicar para mudar de cenário. Clicar e segurar para excluir o cenário. Para adicionar um novo cenário, clicar no sinal de mais.
10. Alterar o nome do projeto e compartilhar por *e-mail*.
11. Desfazer ou refazer alguma ação feita na programação do *script*.
12. Encaixar blocos aqui para formar um novo *script*. Para excluir algum bloco, clicar, segurar e arrastar para fora da raia.
13. Nesta raia branca é onde ficam as instruções de cada personagem.
14. Aqui fica o menu de cada bloco de programação.
15. Aqui é possível selecionar uma categoria de blocos de programação: blocos de eventos (amarelo), movimento (azul), aparência (roxo), sons (verde), controle (laranja), fim (vermelho).
16. Aqui ficam todos os atores criados para o *script*, caso queira colocar esse personagem em outro cenário, clicar, segurar e arrastar. Para excluir um personagem, precisa selecioná-lo e segurar que irá aparecer o x e ai é só clicar para remover.

3.2 Atividades baseadas em labirinto

As atividades que são descritas nessa seção são do tipo desplugadas, realizadas sem o uso de tecnologia. Ao resolver os exercícios “labirinto: leve o gatinho até a casinha”, “centopeia robô” e “descrição narrativa”, é possível aplicar o raciocínio para identificar fatos do cotidiano do aluno, utilizando duas habilidades cognitivas específicas: a memória de longo e curto prazo. Dentro de cada uma dessas três atividades, são abordados os quatro pilares do pensamento computacional: decomposição, reconhecimento de padrões, abstração e algoritmos — este último, por sua vez, é usado com mais ênfase na resolução desses problemas, visto que a capacidade de pensar de forma algorítmica faz com que o aluno analise todo o conjunto de procedimentos que ele fez, uma ou mais vezes até aquele determinado momento (CHRISTIAN; GRIFFITHS, 2017). As subseções a seguir trazem uma descrição detalhada de cada atividade.

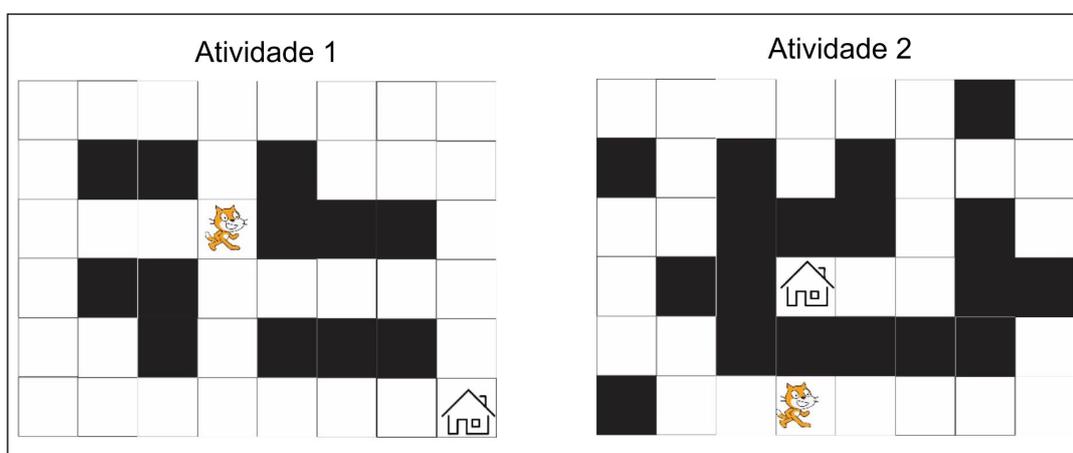
3.2.1 Labirinto: leve o gatinho até a casinha

Essa atividade foi desenvolvida para familiarizar o aluno com a ferramenta ScratchJr, sendo utilizado o personagem gatinho e vários quadrados conectados em forma de labirinto. Em suma, o aluno precisa traçar uma linha do gato até a casinha. Para cada exercício, há uma ou mais soluções possíveis.

O objetivo desta atividade é analisar se os alunos resolvem as atividades utilizando todos os caminhos possíveis, o maior ou o menor caminho ou se não vão conseguir realizar a atividade.

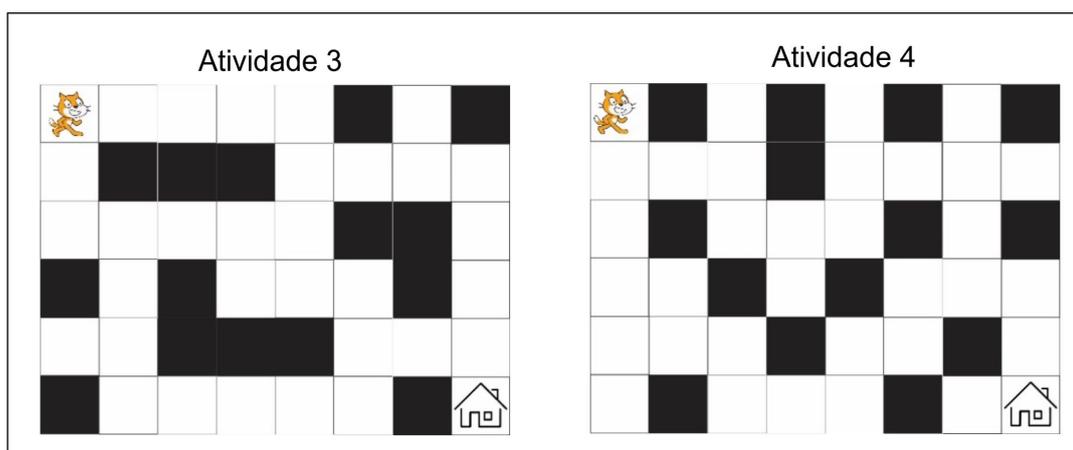
As Figuras 2 e 3 mostram exemplos de atividades.

Figura 2 – Labirinto 1 e 2



Fonte: Acervo pessoal

Figura 3 – Labirinto 3 e 4



Fonte: Acervo pessoal

3.2.2 Centopeia robô

O professor deve organizar a sala em forma de labirinto. Depois, deve chamar algum aluno para ser a cabeça da centopeia e, caso seja a primeira vez da realização da atividade, o professor deve ficar em fila indiana com as mãos nos ombros do aluno para começar a formar o corpo da centopeia. Em seguida, deve explicar como funciona o jogo por meio das regras:

1. Os alunos que ficarem de fora da centopeia ajudarão a cabeça da centopeia a ganhar corpo, à medida que ela for passando por seus colegas de classe, como também a chegar ao fim do labirinto.
2. Os alunos dirão para a centopeia os seguintes comandos: para frente, para a direita, para a esquerda.
3. Quando existir aluno à direita ou à esquerda, a turma deve dizer: direita pega ou esquerda pega para adicionar o aluno à centopeia. Caso seja uma cadeira ou uma mesa quando for dito o comando, a centopeia deve ir para o início do labirinto, se ainda estiver no primeiro terço do trajeto.
4. Quando todos os alunos fizerem parte da centopeia, o professor deve dizer os comandos finais para que a centopeia saia do labirinto.

É possível fazer uma disputa de grupos e as regras são as mesmas. Se o Grupo A for a centopeia quem dará o comando serão os integrantes do Grupo B e vice-versa. Dito isto, vence o jogo quem errar menos os comandos.

3.2.3 Descrição narrativa

É pedido, nesta atividade, que os alunos desenhem suas respectivas casas e, com isso, façam o seu trajeto até a escola. Para facilitar o entendimento é preciso mostrar, no quadro negro, o trajeto do professor como exemplo. De forma similar ao jogo “centopeia robô”, todo o trajeto deve ser demonstrado em formato de setas de cima para baixo, para a direita e para a esquerda.

3.3 Tetris

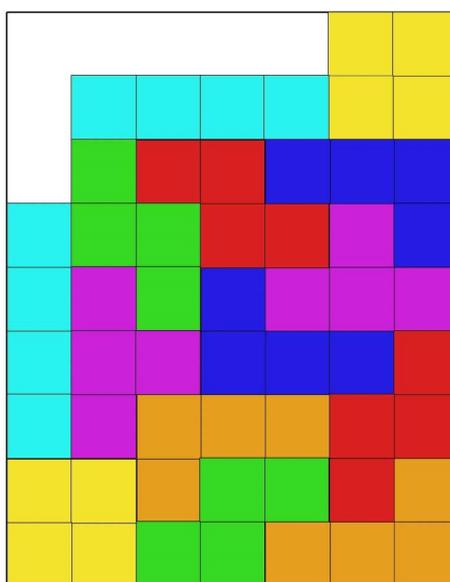
É uma atividade desplugada que desenvolve a habilidade de formação e solução de problemas através do uso de instruções condicionais, que são conceitos fundamentais da Computação.

Segundo Hoff e Wechsler (2004), o Tetris funciona da seguinte forma: uma peça é liberada por vez e, automaticamente, vai descendo. Usando o teclado, pode-se movê-la à direita ou à esquerda, para baixo, rápida ou lentamente, fazer rotações e pausar o jogo. A meta é formar

uma linha completa ou, simultaneamente, duas, três ou quatro, o que gera pontuação diferente (1, 5, 10 e 20, respectivamente). Cada linha completa desaparece e libera mais área. O jogo acaba quando sua área ficar preenchida por linhas incompletas e não há tempo limite.

Porém, para a aplicação desta atividade foi feita uma adaptação para o papel, conforme pode ser visto na Figura 4. As crianças precisam completar todas as linhas e colunas com as peças disponibilizadas, sem sobrepor nenhuma delas. Isso faz com que essa atividade trabalhe com os quatro pilares do pensamento computacional: decomposição, reconhecimento de padrões, abstração e algoritmos. Ou seja, é possível exercitar o raciocínio lógico, interpretação e resolução de problemas.

Figura 4 – Tetris



Fonte: Acervo pessoal

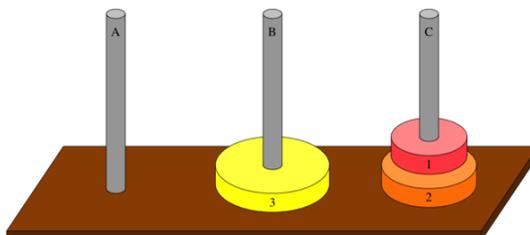
3.4 Torre de Hanói com três discos

A Torre de Hanói, de acordo com Scopim (2014) e Manoel (2019), consiste em um jogo criado pelo matemático francês Edouard Lucas, em 1883, sendo inspirado em uma lenda Hindu. O nome se refere a uma torre na cidade de Hanói, no Vietnã. Este é um jogo de estratégia com uma base de madeira, três hastes fixas e dois a sete discos com diâmetros crescentes. O objetivo é deslocar todos os discos de haste onde se encontram para uma haste diferente, respeitando as seguintes regras: deslocar um disco de cada vez; deslocar o disco do topo de uma haste para o topo de outra; e cada disco nunca poderá ser colocado sobre outro de diâmetro menor.

No âmbito educacional brasileiro, a torre de Hanói tem sido utilizada apenas como uma forma de explorar a coordenação motora das crianças. Com essa atividade, visamos trabalhar o

pensamento computacional por meio de seus quatro pilares. A Figura 5 mostra uma ilustração da torre.

Figura 5 – Torre de Hanói



Fonte: Acervo pessoal

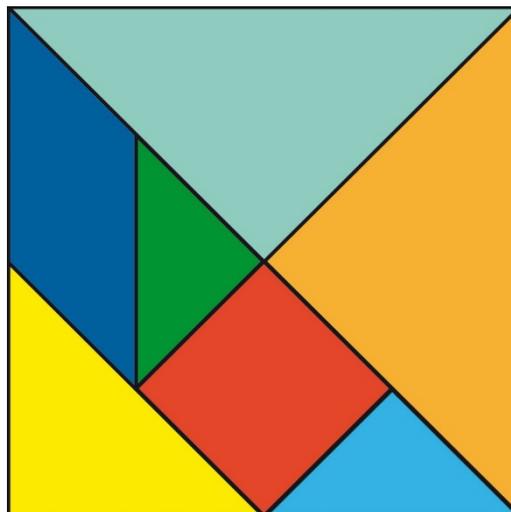
3.5 Tangram

O Tangram é um quebra-cabeça chinês que consiste na reprodução de figuras geométricas usando sete peças: cinco triângulos, um quadrado e um paralelogramo. Não exigindo grandes habilidades dos jogadores, basta ter criatividade, paciência e tempo. Todas as peças devem ser utilizadas e não é permitido sobrepor nenhuma peça.

O uso de materiais manipuláveis como as peças do Tangram possibilita efetuar diferentes rotações, composições e decomposições (NACARATO, 2004). Em outras palavras, as crianças poderão exercitar além do raciocínio lógico, também a criatividade. Serão exercitadas ainda as noções de espaço a fim de que todas as peças se encaixem.

A Figura 6 mostra a adaptação para o papel do Tangram que foi utilizado em sala de aula. Foi dada a cada aluno uma figura ilustrativa para que eles possam replicar com o tangram embaralhado, aplicando a habilidade do raciocínio lógico, interpretação e resolução de problemas. Trabalha com os quatro pilares do pensamento computacional: decomposição, reconhecimento de padrões, abstração e algoritmos.

Figura 6 – Tangram

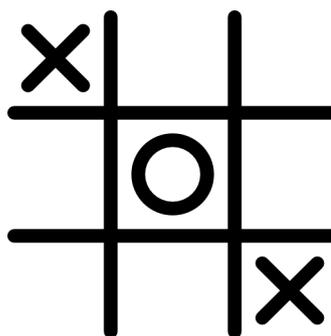


Fonte: Acervo pessoal

3.6 Jogo da velha

O Jogo da velha trabalha o raciocínio lógico, matemático e estratégico. Quando trabalhada em grupo, ela favorece princípios como cooperação, liderança e competição. Nesta pesquisa, vamos utilizar a versão tradicional do jogo, mas existem trabalhos que abordaram uma utilização diferente ao aplicá-lo com figuras geométricas, como é o caso de [Silva \(2017\)](#) e [Silva e Teles \(2017\)](#). Trabalha com os quatro pilares do pensamento computacional: decomposição, reconhecimento de padrões, abstração e algoritmos. Um exemplo segue na Figura 7.

Figura 7 – Jogo da velha



Fonte: Acervo pessoal

3.7 Quebra-cabeça do alfabeto e Conectando as letras

Para [Johnson e Mayer \(2009\)](#), o uso de imagens que fazem referências a um texto, à letra ou à palavra, melhora o desenvolvimento da memória de longo prazo. Dessa forma, isso quer dizer que a aprendizagem baseada em multimídia ajuda o sistema cognitivo do aluno, fixando melhor

o conteúdo. Isto é, quanto mais a criança pratica uma determinada atividade, mais ela relaciona as novas informações aprendidas com as velhas, gerando uma rede de informações associativa, assim, o estudante tende a ter um desenvolvimento escolar mais eficiente (AUSUBEL; NOVAK; HANESIAN, 1980).

O “quebra-cabeça do alfabeto” possui uma imagem para cada letra do alfabeto. No jogo “conectando as letras” é dada uma figura e algumas letras embaralhadas para o aluno conectar as letras e formar uma palavra relacionada à figura. Essa atividade pode ser trabalhada tanto individualmente quanto em grupo. Os detalhes sobre ambas são descritos nas subseções a seguir.

3.7.1 Quebra-cabeça do alfabeto

Com o objetivo de auxiliar no processo de alfabetização, durante a execução do projeto, foi utilizado um quebra-cabeça ilustrativo com figuras das 26 letras do alfabeto. Os alunos aprendem a ordem alfabética e a reconhecer as letras enquanto se divertem. Quanto ao desenvolvimento do pensamento computacional, trabalha com os quatro pilares: decomposição, reconhecimento de padrões, abstração e algoritmos. Na Figura 8 é possível ver um exemplo.

Figura 8 – Quebra-cabeça do alfabeto



Fonte: Acervo pessoal

3.7.2 Conectando as letras

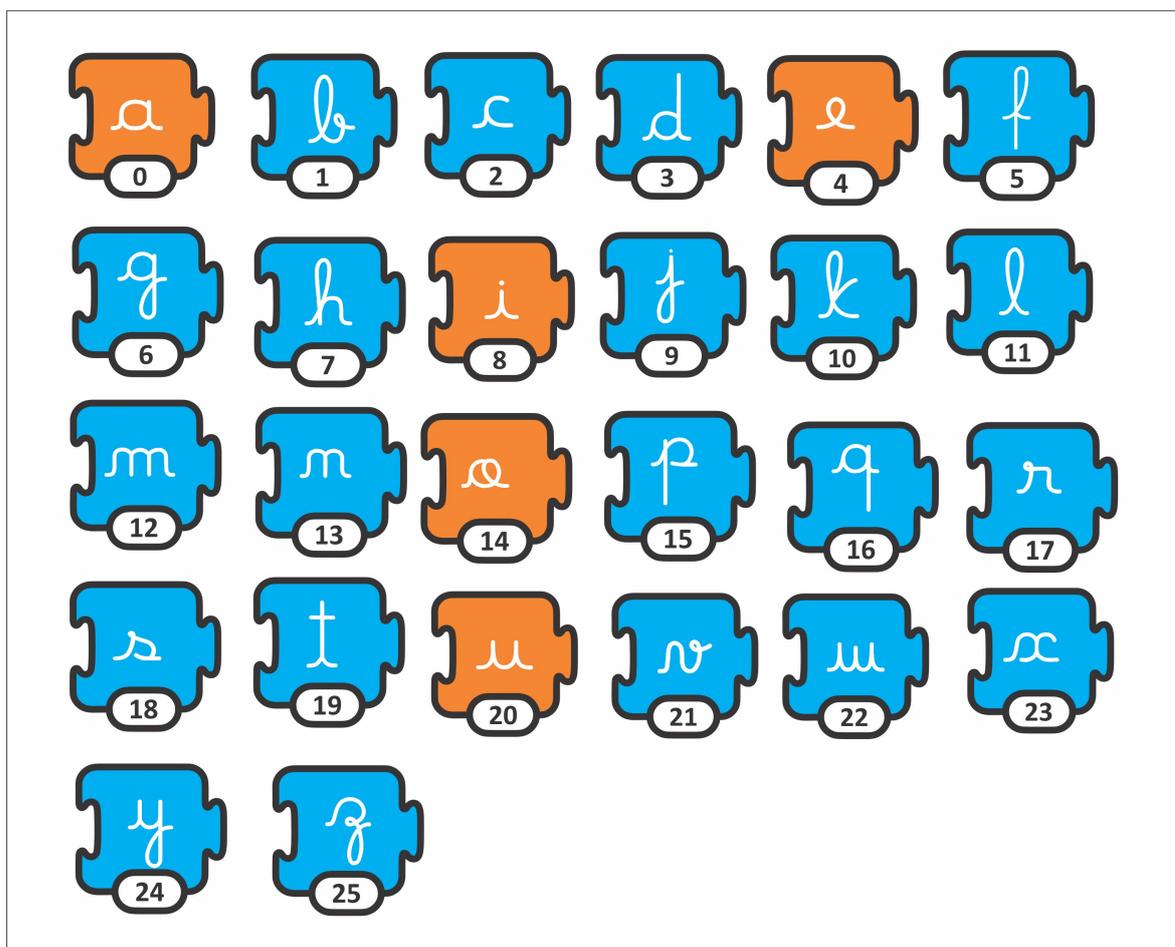
Este é um jogo (Figuras 9 e 10) desplugado, com aparência baseada no ScratchJr. É possível conectar as letras da mesma forma que é possível também conectar os blocos na ferramenta. Essa atividade consiste em formar palavras com as peças que se conectam ao estar na posição correta. As regras da atividade são as seguintes:

1. Os alunos devem ser divididos em grupos de até seis crianças. Depois disso, é explicado que eles devem trabalhar como um time, ou seja, de forma colaborativa.
2. O professor deve mostrar uma figura, perguntar aos alunos o que eles acham que é. Se todos falarem corretamente, pede para começarem a montar a palavra.

3. Quando um dos grupos termina, pede para um dos membros do grupo soletrar a palavra para a turma. Se soletrar corretamente, o grupo ganha um ponto. Vence o grupo com mais pontos no final.
4. Ao final da primeira rodada, deve ser possível identificar quais alunos sabem mais o alfabeto e quais têm dificuldade. O professor deve sempre acompanhar e oferecer ajuda sem dar a resposta.
5. Sempre reforce aos grupos que eles devem trabalhar como um time, que precisam ser companheiros e ajudar uns aos outros a aprender.

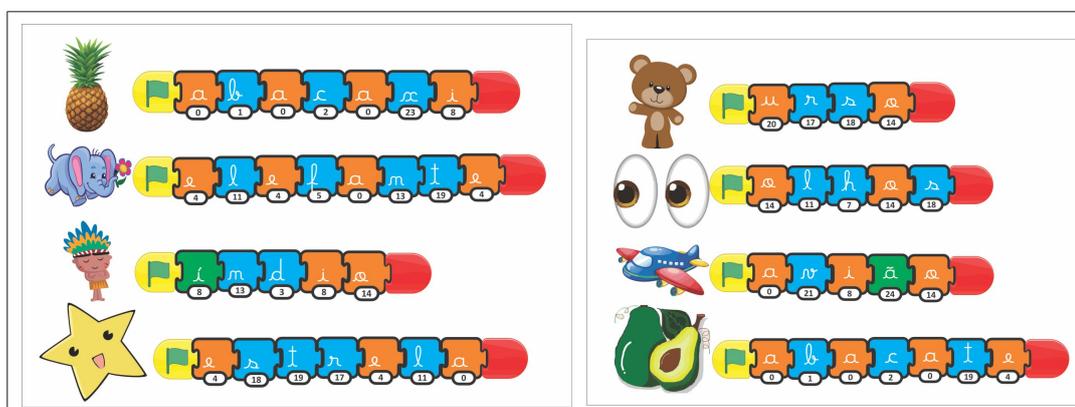
Quanto ao desenvolvimento do pensamento computacional, este jogo trabalha com os quatro pilares: decomposição, reconhecimento de padrões, abstração e algoritmos. A figura 9 mostra como as peças são usadas no jogo. Foi criado com o intuito dos alunos aprenderem de forma desplugada como usar os blocos usados para programar no ScratchJr. A figura 10 apresenta alguns exemplos de exercícios. Existe um padrão de cores, as vogais são de cor laranja, as consoantes são de cor azul e a fonte utilizada é a cursiva, que é normalmente ensinada nas escolas.

Figura 9 – Jogo do alfabeto baseado no ScratchJr



Fonte: Acervo pessoal

Figura 10 – Exercícios



Fonte: Acervo pessoal

3.8 Avaliações

As avaliações foram desenvolvidas para serem aplicadas de duas formas:

1. **Objetiva:** nessa modalidade de avaliação, é dada uma folha com os desenhos ilustrados da ferramenta ScratchJr. Os alunos responderão com um X o que uma figura se relaciona a qual tipo de funcionalidade aprendida durante as aulas;
2. **Prática:** é pedido aos alunos que criem e programem uma história interativa no ScratchJr usando o conteúdo aprendido.

Nas Figuras 11, 12 e 13 são ilustradas as avaliações criadas para verificar o que os alunos aprenderam sobre programação, mais especificamente sobre o uso do ScratchJr.

Figura 11 – Avaliação 1 e 2

Avaliação 1

MEU ROBÔ



1) CIRCULE O QUE VOCÊ USARIA PARA **INICIAR** O ROBÔ





2) CIRCULE O QUE VOCÊ USARIA PARA **MOVIMENTAR** O ROBÔ





2) CIRCULE O QUE VOCÊ USARIA PARA **PARAR** O ROBÔ





Avaliação 2

JOÃO E MARIA



1) CIRCULE O QUE VOCÊ USARIA PARA **FAZER JOÃO E MARIA FALAR**






2) CIRCULE O QUE VOCÊ USARIA PARA **FAZER JOÃO DESAPARECER**






3) MARQUE UM X ONDE VOCÊ CLICARIA COM OS DEDOS PARA **AUMENTAR A VELOCIDADE DOS MOVIMENTOS DO SEU PERSONAGEM**

PRIMEIRO AQUI 

DEPOIS AQUI 

PRIMEIRO AQUI 

DEPOIS AQUI 

Fonte: Acervo pessoal

Figura 12 – Avaliação 3 e 4

Avaliação 3



1) CIRCULE O QUE VOCÊ USARIA PARA FAZER O PERSONAGEM SE MEXER QUANDO FOSSE TOCADO POR OUTRO PERSONAGEM?






1) CIRCULE O QUE VOCÊ USARIA PARA INICIAR O PROGRAMA AO TOCAR NO PERSONAGEM?






1) CIRCULE O QUE VOCÊ USARIA PARA GRAVAR UM AUDIO DE UM PERSONAGEM?






Avaliação 4



1) CIRCULE O QUE VOCÊ USARIA PARA REPETIR OS MOVIMENTOS DE UM PERSONAGEM?






1) CIRCULE O QUE VOCÊ USARIA PARA PARAR UM MOVIMENTO SO POR UM SEGUNDO?






1) CIRCULE O QUE VOCÊ USARIA PARA PARAR UM MOVIMENTO DE UM PERSONAGEM?






Fonte: Acervo pessoal

Figura 13 – Avaliação 5 e 6

Avaliação 5



1) CIRCULE O QUE VOCÊ USARIA PARA FAZER UM PERSONAGEM VOLTAR AO INÍCIO?



1) CIRCULE O QUE VOCÊ USARIA PARA PARAR GIRAR UM PERSONAGEM?



1) CIRCULE O QUE VOCÊ USARIA PARA PARAR IR PARA OUTRA PÁGINA?



Avaliação 6



1) CIRCULE O QUE VOCÊ USARIA PARA AUMENTAR UM PERSONAGEM?



1) CIRCULE O QUE VOCÊ USARIA PARA PARAR DIMINUIR?



1) MARQUE UM X ONDE VOCÊ USARIA PARA PARAR AUMENTAR E DIMINUIR VARIAS VEZES ?



Fonte: Acervo pessoal

4

Aplicação no ensino fundamental

As atividades descritas na metodologia foram aplicadas, observadas e feitas anotações no dia a dia de todos, com os detalhes, para que a análise qualitativa fosse confiável no que diz respeito às questões de pesquisas propostas nesta aplicação. As seções a seguir irão destacar as informações coletadas.

4.1 Aula 1 (Pré teste)

O pré-teste foi realizado usando uma atividade chamada: leve o gatinho até a casinha, que é um exercício de tabuleiro baseado no ScratchJr. A atividade teve o propósito de identificar quais crianças possuíam mais dificuldades com questões relacionadas à lógica. Assim, desenvolvemos quatro tipos diferentes de testes com diferentes níveis de complexidade de acordo com a quantidade de obstruções no labirinto.

O objetivo das crianças era encontrar qual o menor caminho para o gatinho chegar em casa. Foram entregues quatro tipos de atividades com caminhos com diferentes dificuldades. Como não havia auxílio por parte do professor os alunos realizaram os exercícios de forma livre, o que permitiu que muitos deles escolhessem apenas um caminho, já outros alunos escolheram vários caminhos e outros nenhum dos caminhos possíveis. Com essa intervenção foi possível identificar aqueles alunos com raciocínio lógico, abstração e decomposição de problemas mais avançados, como também aqueles que teriam mais dificuldades nas atividades relacionadas ao ensino de programação.

É apresentado a seguir, como os alunos responderam a cada uma das atividades em ordem crescente de suas respectivas dificuldades.

Atividade 1: possuía seis caminhos possíveis onde dois deles eram os menores e apenas um era o maior, os demais eram tamanhos intermediários. O nível de complexidade é considerado baixo para esta atividade, visto que o labirinto possuía poucas obstruções. Foram encontrados

os seguintes resultados: quatorze alunos realizaram esta atividade, dos quais dois utilizaram os seis caminhos possíveis, outros dois utilizaram três caminhos, sendo que dois deles eram os menores, os outros dez alunos utilizaram apenas um caminho e, dentre estes, sete utilizaram o menor caminho possível.

Atividade 2: possuía apenas um único caminho possível, entretanto, o nível de complexidade passou a aumentar devido à existência de diversas obstruções no labirinto. Foram encontrados os seguintes resultados: onze alunos realizaram essa atividade, dos quais três deles não conseguiram construir um **script** de forma correta; já os demais conseguiram concluir com sucesso.

Atividade 3: possuía cinco caminhos possíveis dos quais três eram os menores e apenas um era o maior. O nível de dificuldade, nessa atividade, passou a ser mais alto, uma vez que havia muito mais obstruções, o que obrigava o aluno a pensar um pouco mais. Sendo assim, os seguintes resultados foram encontrados: dos quatorze alunos que realizaram essa atividade, um utilizou todos os caminhos possíveis e os demais utilizaram apenas um único caminho, sendo que nove foram pelo menor e quatro pelo maior caminho.

Atividade 4: possuía três caminhos possíveis dos quais um era o menor e dois eram os maiores. O nível nessa atividade passou a ser a mais complexa, tendo em vista que havia muito mais obstruções, o que obrigava o aluno a pensar um pouco mais. Com isso, os seguintes resultados foram encontrados: dos quatorze alunos que realizaram esta atividade, três não conseguiram realizar a atividade, deixando assim seu **script** sem uma solução correta; quanto aos demais, estes utilizaram apenas um caminho, sendo que seis foram pelo menor e cinco pelo maior caminho.

Na Figura 14 é possível visualizar as crianças realizando o teste aplicado.

Figura 14 – Crianças realizando as atividades do labirinto



Fonte: Acervo pessoal

4.2 Aula 2 - 4 (Jogo centopeia robô)

O professor deve organizar a sala em forma de labirinto, depois deve chamar algum aluno para ser a cabeça da centopeia (se for a primeira vez que os alunos forem realizar essa atividade, o professor deve ficar em fila indiana com as mãos nos ombros do aluno para começar a formar o corpo), deve explicar como funciona o jogo e suas regras, conforme descrito na seção 3.2.2.

Execução da atividade

Durante três aulas foi ensinado o jogo desplugado centopeia robô para que os alunos pudessem aprender como funcionava e, com isso, trabalhar as habilidades cognitivas, tempo de resposta, atenção focada e memória de trabalho. Além disso, este exercício serviu para correlacionar o que seria ensinado nos assuntos de programação. Assim, foi criado um labirinto dentro da sala, como pode ser visto na Figura 15, com as próprias carteiras e depois foram dadas as instruções que foram observadas no parágrafo anterior.

Figura 15 – Jogo centopeia robô



Fonte: Acervo pessoal

O que foi observado?

Foi observado que, no início, as crianças que ficaram dando os comandos para a centopeia tiveram um pouco de dificuldade para abstrair sua localização quando estavam de frente, como por exemplo: eles davam um comando para a esquerda, mas deveria ter sido direita. Em resumo, isso indica que as crianças ainda possuíam dificuldade nas matérias de Geografia e Matemática, no ambiente em que vivem, espaço e forma, ou seja, não conseguiam identificar pontos de referências, como confirmado pela professora responsável da turma

Para auxiliá-los na primeira partida, o professor precisou participar do jogo. Ele levantava a mão para a direita e para a esquerda e, em seguida, perguntava para os que estavam de fora para qual lado deveria ir ou se deveria tocar em um participante para que fizesse parte da centopeia, fazendo-a crescer. Quando não havia nenhum estudante do lado, ele ficava calado, o que ocasionava um erro de comando, já que, ao invés de dizer para frente, diziam para a direita,

de modo a fazer com que a centopeia acabasse se chocando com a cadeira. Logo, o jogo iniciava novamente.

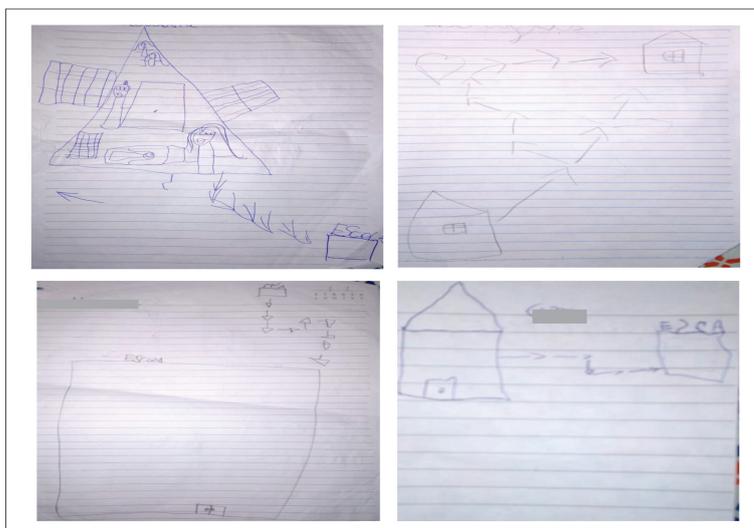
Assim, o jogo foi recommençado três vezes até que eles entendessem o que precisavam fazer. Ao finalizar a brincadeira, o professor mostrou no quadro tudo o que aconteceu, desenhando um labirinto, uma centopeia, criando setas e indagando diante do desenho o que aquela centopeia deveria fazer, se deveria ir para frente, para direita ou esquerda, fixando de forma visual o que foi aprendido naquele dia.

Durante as aulas três e quatro, o jogo já foi realizado de forma mais dinâmica e interativa entre os próprios alunos, de maneira que, nesses dois dias que se seguiram, apenas eles participaram. Ademais, buscando fortalecer o trabalho em equipe e deixar o jogo mais divertido, foi realizado um duelo entre meninos contra meninas. Foi observado que os alunos passaram a não ter dificuldades em relação a que lado deveriam dar os comandos para a centopeia.

4.3 Aula 5 (Jogo centopeia robô e descrição narrativa)

Durante esta aula foi perguntado aos alunos o que eles lembravam das aulas anteriores. Foi pedido a eles que desenhassem suas respectivas casas e fizessem o seu trajeto até a escola. Para facilitar o entendimento, mostrei a eles o meu trajeto no quadro como exemplo e, para contextualizar o que tinha sido estudado anteriormente, foram utilizadas setas para frente, para a direita e para a esquerda conforme pode ser visto na Figura 16.

Figura 16 – Desenho da atividade descrição narrativa



Fonte: Acervo pessoal

Foi observado que alguns dos alunos replicaram seus trajetos utilizando o exemplo dado, já outros foram mais criativos e desenharam além de suas casas, seus pais, mães, etc. Nesta atividade chamada “descrição narrativa”, é preciso observar que parte dos alunos não

conseguiram utilizar a função cognitiva quanto à percepção, ou seja, não conseguiram abstrair que deveriam criar seus trajetos diferentes do exemplo dado.

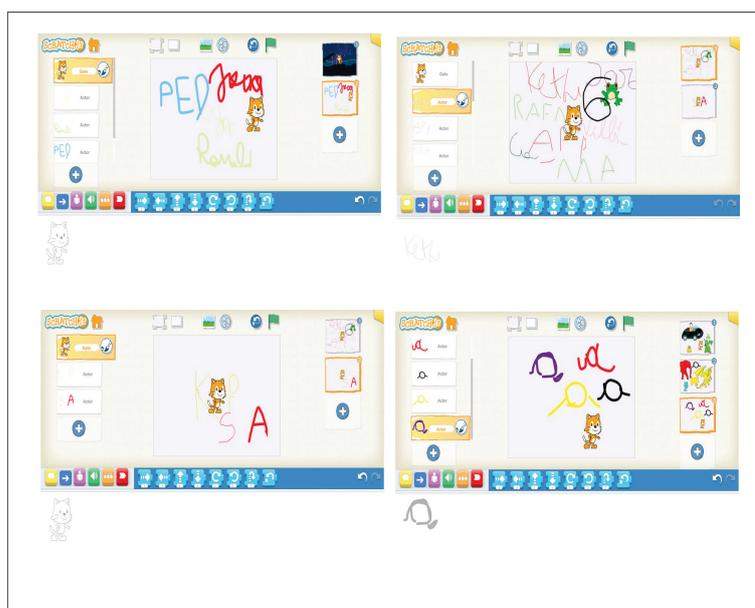
4.4 Aula 6 - 8 (Primeiro contato com o ScratchJr)

Didática seguida nesta aula e nas subsequentes:

Como não havia a possibilidade de utilizar um projetor no quadro, tudo o que era ensinado era mostrado primeiro no *tablet*: onde se deveria clicar, como fazer para arrastar os blocos e assim sucessivamente. Após mostrar as funcionalidades, o professor entregava os *tablets* aos alunos e apenas observava de longe, acompanhando os que tinham mais dificuldades em como puxar o bloco, visto que como o sensor de tela do *tablet* não era dos melhores, foi preciso treinar os alunos, por exemplo, como dar toques para que de fato fosse possível puxar os blocos ou até mesmo como mudar paletas de cores e de tipos de personagens.

Devido ao fato de que nem todos os alunos estavam presentes nessa aula inicial com o ScratchJr, ela foi refeita duas vezes da seguinte forma: a turma foi dividida em quatro grupos para que fosse mostrado o aplicativo e criado o primeiro projeto. Assim, para aqueles que tinham faltado, foi mostrado como fazer para desenhar e como criar seus próprios personagens. Para os que já tinham tido essa aula, foi pedido que tentassem criar um novo personagem que seria seus respectivos nomes ou as suas iniciais. A Figura 17 mostra como os alunos utilizaram a ferramenta após serem ensinados sobre as funções básicas.

Figura 17 – Primeiro contato dos alunos com o ScratchJr



Fonte: Acervo pessoal

4.5 Aula 9 - 11 (Jogo conectando as letras)

Foi utilizado um jogo, desenvolvido durante o planejamento e desenvolvimento do projeto, que consiste em formar palavras que estão em formas de blocos que se conectam ao estar na posição correta e seguem as regras da atividade, conforme visto na subseção 3.7.2. Os alunos foram divididos em grupos de cinco e durante as três aulas as equipes foram sendo modificadas. Como o objetivo do jogo é desenvolver o lado da alfabetização e do trabalho em equipe dos estudantes, porém durante a atividade percebeu-se que eles estavam muito individualistas. Então foi preciso criar um campeonato com prêmios para motivar o desenvolvimento em grupo, devido as regras do jogo.

Foi possível observar, com essa competição, que os alunos estavam tímidos e que ao serem motivados conseguiram trabalhar em grupo. O ponto mais importante aqui é de que um ajudou o outro, pois, quando um aluno não sabia uma determinada letra, o grupo dizia e ensinava para ele, de maneira que isso fez com que eles aprendessem ensinando e se divertindo. Na Figura 18 há algumas palavras construídas pelos alunos.

Figura 18 – Atividades conectando as letras



Fonte: Acervo pessoal

4.6 Aula 12 - 14 (Pintando, desenhando e escrevendo no ScratchJr)

Nesta aula, foram utilizados os desenhos dos alunos. Nesta aula, foram utilizados os desenhos que os alunos fizeram durante a aula cinco, que mostravam seus trajetos de casa para a escola, para criar o cenário. Foi ensinado como utilizar a câmera pelo aplicativo e como adicionar

a imagem como cenário, de modo que eles puderam pintar e animar no *tablet* um desenho feito à mão.

Foi observado um pouco de dificuldade por parte de alguns dos alunos nas funcionalidades avançadas como inserção de imagem externa, edição, escrita e salvar o projeto. Em decorrência disso, foi preciso dedicar um pouco mais de atenção ao assunto e quando todos estavam familiarizados com as funções, passamos para o ensino de programação. Abaixo segue a Figura 19 de um dos projetos que utilizou a câmera para inserir um cenário com um desenho externo.

Figura 19 – Inserindo Imagem no *tablet*



Fonte: Acervo pessoal

4.7 Aula 15 - 17 (ScratchJr)

Nestas três aulas foi possível contar com a presença de 18 alunos. No primeiro dia de aula foram entregues 9 *tablets* para que os alunos trabalhassem em duplas. Uma vez que os pares eram formados por alunos com mais e com menos familiaridade na ferramenta, era pedido que criassem um programa cada um. Isso serviu para equilibrar o desenvolvimento da turma dentro do assunto, já que aquele que sabia um pouco, acabava ajudando quem sabia um pouco menos e, com isso, ampliavam seus conhecimentos.

Do segundo dia em diante, foi entregue um *tablet* para cada aluno e ensinados novos conteúdos. Após o ensino, era pedido para que eles criassem *scripts* como desejassem e que sempre testassem ao implementar qualquer coisa. Depois disso, em um terceiro momento, foi apontado o bloco de movimento e feitas analogias com o jogo centopeia robô, como por exemplo o bloco de movimento para virar a esquerda ou a direita, lembrava como era no jogo desplugado, isso facilitou o entendimento da ferramenta. Quando o professor finalizava a explicação, era

perguntado aos alunos qual era a cor que fazia o personagem se mexer, qual a cor que fazia o *script* funcionar e, em ambas as perguntas, a maioria da turma respondia instantaneamente de forma correta. Contudo, quando havida dúvidas nas respostas, o professor explicava novamente.

O que foi observado?

Observou-se que, ao demonstrar em forma de exemplos o que acontecia quando se colocava o bloco de inicialização em conjunto com o bloco de movimento, os alunos replicavam com seus próprios *scripts* sem auxílio e já testavam, apagavam tudo e faziam novamente. Além disso, os alunos realizavam coisas que não tinham sido pedidas nesta aula, como criar um novo personagem, editar suas cores, colocar novos cenários e criar os respectivos códigos. Isso aconteceu com quase toda a turma.

Uma das alunas apenas queria ficar usando o *tablet* para criar personagem, pintá-los, colocar cenário e editar por ainda ter 6 anos, 1 a menos que a maioria da turma. Diante disso, foi preciso acompanhá-la com mais atenção para que se desenvolvesse como os outros. Já o aluno especial, de 10 anos, criou códigos utilizando quase todos os blocos. Vale destacar que ele criou uma fazendinha, colocou uma vaca e, utilizando o bloco de som, gravou o mugido da vaca. Dentre os alunos de 7 anos de idade, somente um se destacou por criar o código com todos os blocos e dizia: “tio, isso aqui faz desaparecer, isso aqui deixa o boneco grande e depois ele volta ao normal e isso daqui faz ele andar”. Por fim, os demais alunos utilizavam apenas os blocos mostrados, porém, testaram todas as funcionalidades possíveis do bloco ensinado, visto que eram vários tipos diferentes. A Figura 20 apresenta os alunos realizando as atividades.

Figura 20 – Alunos fazendo atividades



Fonte: Acervo pessoal

4.8 Aula 18 - 20 (ScratchJr)

Nas aulas 18 a 20, foi dada continuidade às atividades com o ScratchJr. Nesses três dias foram revisados os assuntos já dados e, por fim, foram realizados novos *scripts* pelos alunos para que continuassem a praticar mais ainda a função cognitiva da memória, coordenação e concentração. Observou-se um pouco de dificuldade em alguns alunos, em consequência das suas excessivas faltas durante as aulas e consecutivamente no projeto. Assim, eles necessitaram de acompanhamento com mais atenção, bem como precisaram realizar atividades anteriores durante esses dias.

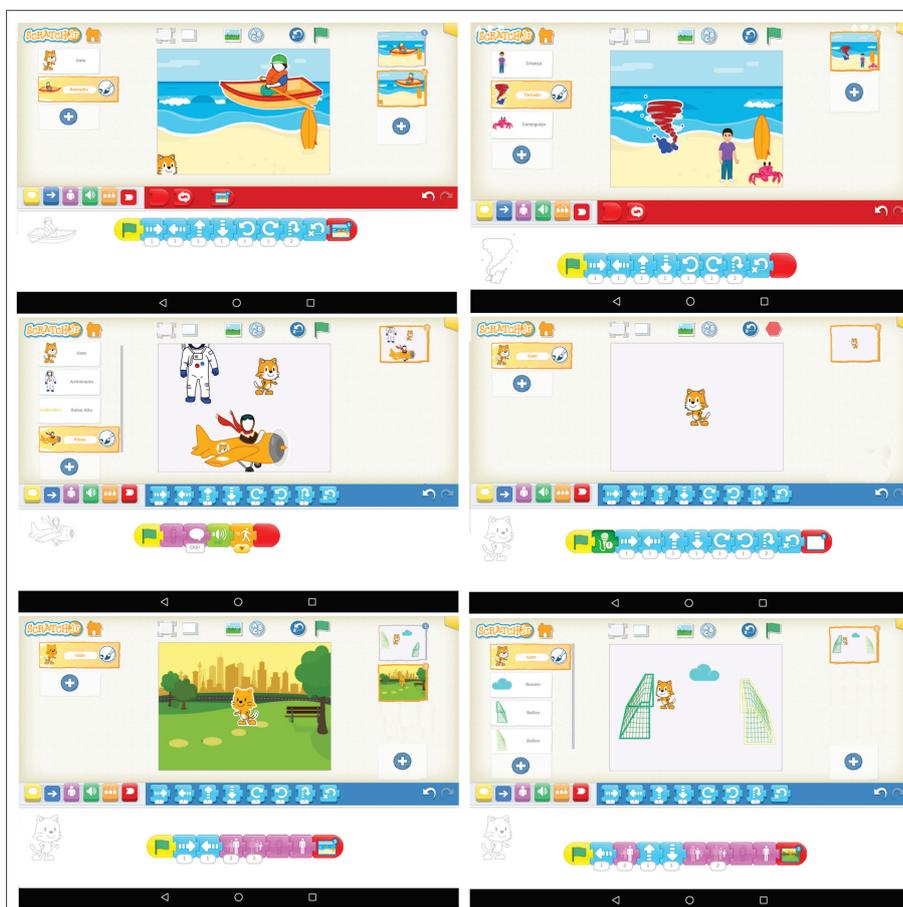
4.9 Aula 21 (Avaliação)

A avaliação foi aplicada para que fosse possível identificar quais os assuntos os alunos tinham absorvido. Pedimos que eles criassem projetos com tudo o que já haviam aprendido no ScratchJr. Vale lembrar que, antes de serem aplicados os testes, foi feita uma revisão com o auxílio do *tablet*, em que era mostrado bloco por bloco da ferramenta e era questionado aos alunos o que eles sabiam sobre e para que servia, no entanto, não foi sanado as dúvidas nesse momento.

Com efeito, foram encontrados os seguintes resultados: dos dezesseis alunos presentes

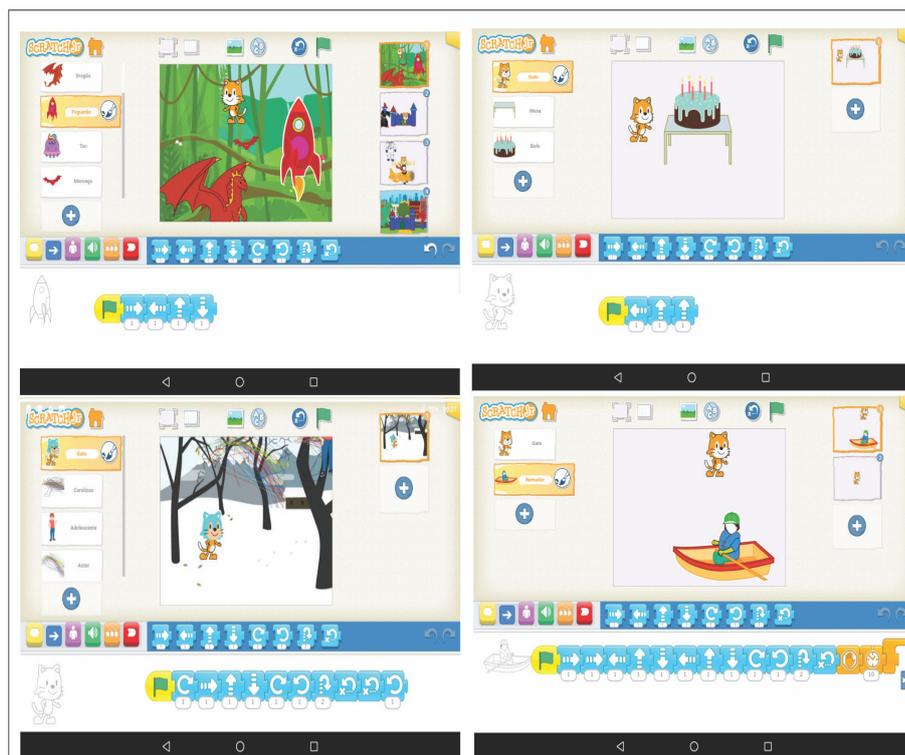
neste dia, seis deles conseguiram criar *scripts* com início, meio e fim, ou seja, bloco de iniciar, de movimento e de finalização; e os outros dois fizeram *scripts* com os mesmos blocos sem finalizar. Segue o que foi feito por estes oito na Figura 21.

Figura 21 – *Scripts* com bloco de início, movimento, aparência e finalização



Fonte: Acervo pessoal

Na Figura 22, é possível visualizar os *scripts* de cinco alunos que apenas inseriram blocos de início e de movimento.

Figura 22 – *Scripts* com bloco de início e movimento

Fonte: Acervo pessoal

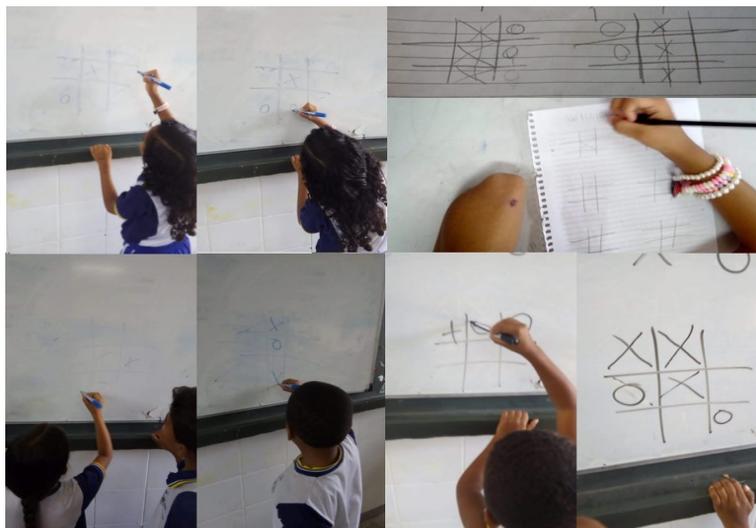
Finalmente, notou-se que cinco alunos não criaram *scripts*, eles somente direcionaram seus esforços para personalizar os personagens e os cenários.

4.10 Aula 22 - 24 (Jogo da velha)

Durantes essas três aulas, o quadro foi utilizado para jogar com as crianças o jogo da velha. O objetivo era que, ao final do terceiro dia, alguns pudessem vencer o professor. Além de jogar contra o professor, eles jogavam entre si, meninos contra meninas, o que deixou a atividade dinâmica e divertida.

Da turma 3 dos alunos não sabiam jogar e então foram ensinadas as regras. No primeiro dia ninguém conseguiu ganhar do professor, porém, foi surpreendente como os alunos se desenvolveram no decorrer das intervenções. No final das aulas, havia cinco alunos que conseguiram ganhar apenas uma vez, cumprindo o objetivo. Vale ressaltar que esses cinco foram os que mais se desenvolveram em sala de aula, tanto com os conteúdos da grade curricular quanto com a programação, de acordo com o que pode ser constatado na Figura 23

Figura 23 – Um dos jogos da velha realizado pelos alunos

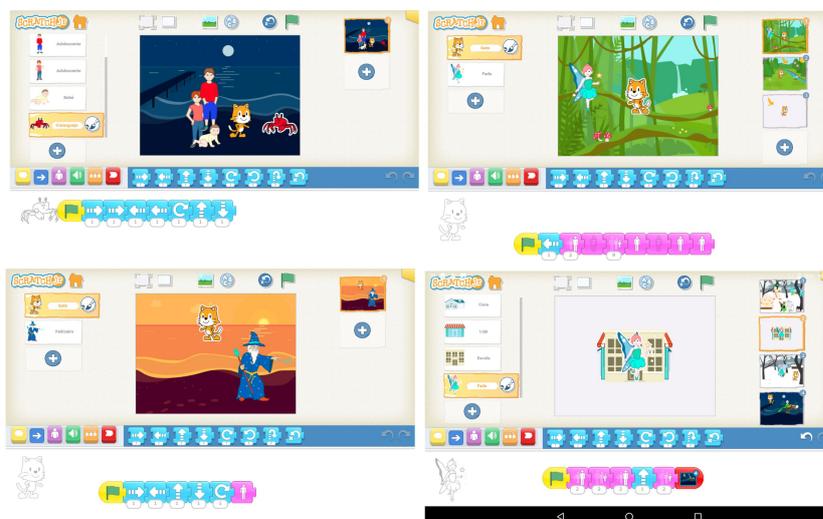


Fonte: Acervo pessoal

4.11 Aula 25 - 27 (ScratchJr - Projetos: gatinho feliz, João e Maria e livre)

Nessas três aulas, os assuntos do ScratchJr foram recapitulados. Os alunos foram ensinados de forma tradicional, ou seja, foram pedidos para replicar o que o professor fazia, sem serem deixados livres para criarem seus programas. Assim fizeram os estudantes em seus *tablets* para que fixassem todos os blocos da linguagem. Além disso, eram feitas perguntas sempre que necessário para saber se todos estavam aplicando de forma correta o que estava sendo solicitado. Nessas atividades, foram identificados os alunos que ainda estava com dificuldade na programação de qualquer personagem e, a estes, foi dada atenção individual após cada explicação. Na Figura 24 é possível visualizar alguns dos projetos de João e Maria realizado pelos alunos nessa aula.

Figura 24 – Projetos



Fonte: Acervo pessoal

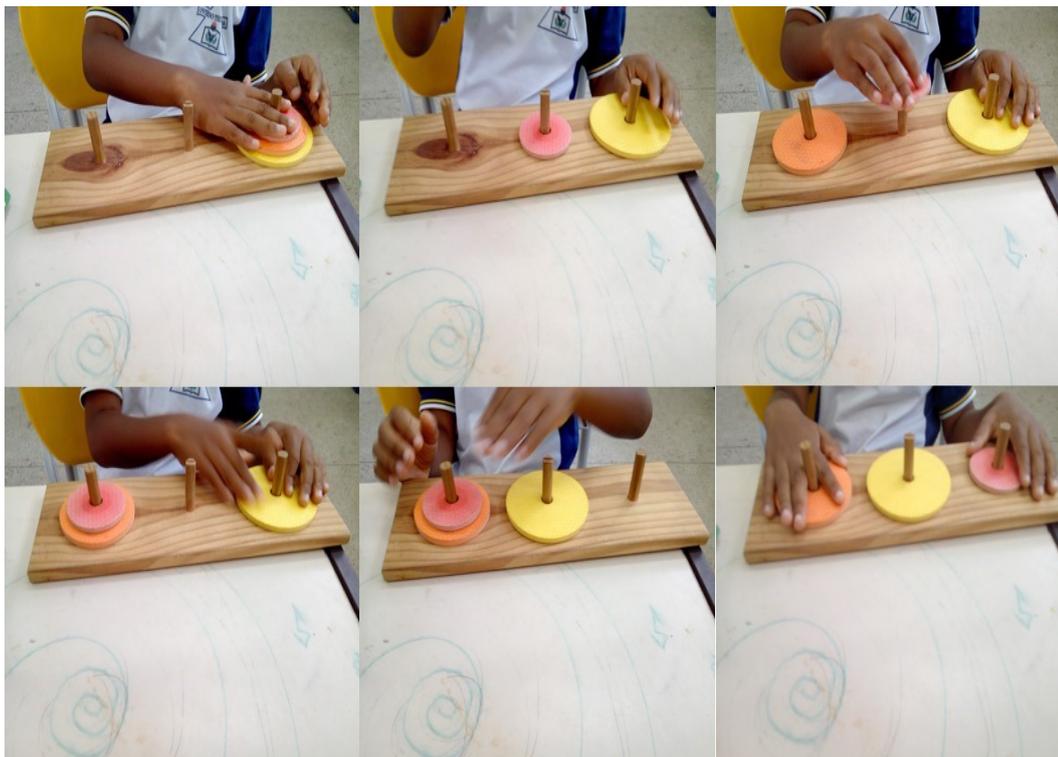
4.12 Aula 28 (ScratchJr - Aula livre)

As aulas livres que se seguirão a partir desta servirão para que os alunos usassem a ferramenta sem auxílio, com os conhecimentos já obtidos sobre como manusear e utilizar a linguagem ScratchJr. Frisa-se que apenas o responsável da turma deveria ficar observando quem ainda estava com dificuldade. Neste dia, foi possível contar com 14 alunos, dos quais somente 2 ainda tinham dificuldade em utilizar o aplicativo.

4.13 Aula 29 - 31 (Torre de Hanói)

Durante os três dias de aula tivemos quantidades de alunos diferentes: 10 no primeiro dia, 15 no segundo dia e 19 no terceiro dia. Na Figura 25, pode ser visualizado um dos alunos resolvendo passo a passo o problema da Torre de Hanói.

Figura 25 – Torre de Hanói sendo resolvida por um dos alunos



Fonte: Acervo pessoal

Dividimos a atividade em três dias que são vistos nas subseções [4.13.1](#), [4.13.2](#) e [4.13.3](#).

4.13.1 Primeiro dia

Nesta aula foram mostradas as regras aos alunos de forma visual, sendo estas: o maior disco sempre tem que estar embaixo, eles só podem pegar uma peça de cada vez e tem que levar a torre completa para qualquer pino de sua escolha que não seja o que foi iniciado. Depois foi dada uma torre para cada aluno. Seis das torres eram totalmente de madeira e cor uniforme, já as outras quatro eram com discos coloridos emborrachados.

Percebeu-se uma dificuldade nos 10 primeiros minutos e foi possível observar que alguns dos alunos só conseguiram realizar a atividade com os discos coloridos e outros só com os de madeira, tendo em vista que foi realizada uma troca das torres. Outro ponto observado foi que, dentre os alunos que estavam com dificuldade, eles tinham também dificuldades em observar visualmente qual era o maior disco para poder ficar embaixo. Desse modo, estes precisavam pegar nos dois discos para fazer a troca, comparavam qual era o maior, colocavam de volta na posição e depois realizavam a troca. Contudo, com um deles não houve sucesso, mesmo ele pegando em ambos os discos. Com isso, troquei a torre por uma colorida, já que ele estava com uma de madeira e, de imediato, ele conseguiu resolver o problema.

Todos os 10 alunos deste dia concluíram com sucesso a atividade.

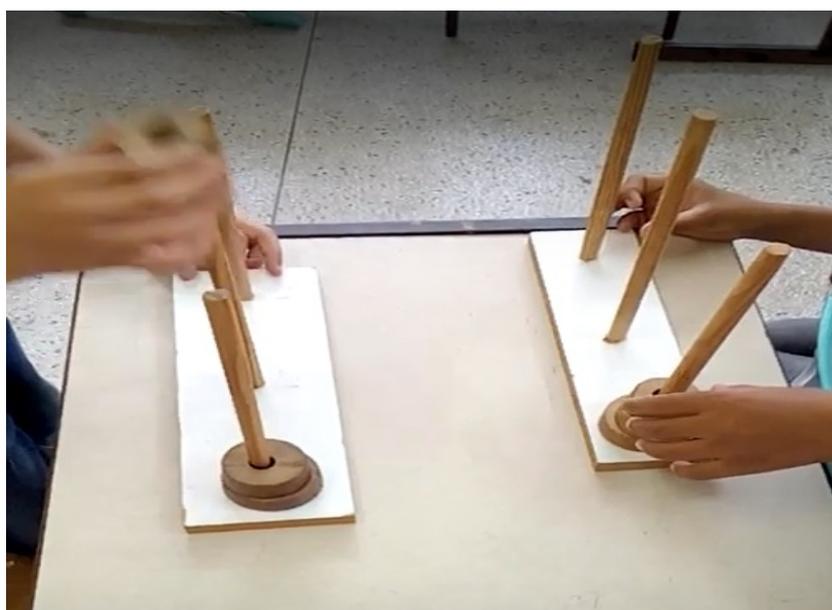
4.13.2 Segundo dia

Os cinco alunos que não estavam na aula anterior tiveram que ser auxiliados. Os demais continuaram a resolver novamente a torre de Hanói. Deve-se salientar, inclusive, que os meninos, por iniciativa própria, começaram a disputar um contra o outro para ver quem resolvia o problema de forma mais rápida.

4.13.3 Terceiro dia

Então, enquanto eu auxiliava os que tinham faltado nos dias anteriores, os demais que já tinham aprendido pediram para criar um campeonato para ver quem era o melhor. Nesse sentido, deixei disputarem e fiquei verificando se estavam fazendo corretamente como pode ser visto na Figura 26. Depois que todos os alunos pegaram o jeito da coisa, estes não pararam de jogar e ficaram a aula toda resolvendo várias vezes a torre, inclusive de formas diferentes.

Figura 26 – Desafio 1 contra 1 torre de Hanói



Fonte: Acervo pessoal

4.14 Aula 32 - 34 (ScratchJr - Projetos)

Após trabalhar a torre de Hanói, foram utilizadas mais três aulas de programação, porém, agora trabalhando nos projetos aspectos mais avançados do ScratchJr. Dessa maneira, foi identificado que o raciocínio lógico dos alunos que ainda estavam com dificuldades melhorou bastante. Em situações anteriores muitos alunos não conseguiam criar programas com blocos simples e após os projetos muitos conseguiram se desenvolverem melhor e criarem sozinhos novas histórias interativas no ScratchJr. Podemos verificar isso, por exemplo, nas transições de cenários. Um dos projetos realizados pode ser observado na Figura 27.

Figura 27 – Projeto na praia com transição de cenários



Fonte: Fonte: Acervo pessoal

4.15 Aula 35 - 37 (Tangram)

Durante três dias de aulas foi realizada a intervenção, com a utilização do Tangram com sete peças. O objetivo dessas aulas foi melhorar o raciocínio lógico das crianças para que elas pudessem se desenvolver melhor tanto na programação como principalmente em sala de aula. A Figura 28 mostra um dos alunos resolvendo uma das figuras passadas como problemas de atividades em sala de aula.

Figura 28 – Tangram resolvido pelos alunos



Fonte: Acervo pessoal

Foi observado que, dos 16 alunos que estavam presentes no primeiro dia, apenas dois alunos tiveram dificuldade em entender como funcionava para resolver o Tangram. Já no segundo dia, foi possível contar com os 19 alunos da turma e, dentre eles, só os três que faltaram no dia

anterior tiveram dificuldade. O terceiro dia serviu para fixar os conhecimentos e para que os alunos continuassem a tentar resolver os Tangram que ainda não tinham sido resolvidos.

Foi dada, a cada aluno, uma folha que possuía 6 figuras (casa, barco, cachorro, camelo, cavalo e cisne) geométricas de animais em preto e branco, cada uma delas tem um nível de dificuldade diferente. Com efeito, os resultados encontrados foram:

Na figura da casa, 100% dos alunos conseguiram resolver durante os três dias mais de uma vez. Na figura do barco, no primeiro dia de aula apenas 12 dos alunos conseguiram resolver; no segundo dia 15 alunos e no terceiro dia todos resolveram. Na figura do cachorro 17 alunos conseguiram resolver no primeiro dia e no segundo dia todos resolveram.

Quanto às figuras do cavalo, do camelo e do cisne somente um aluno conseguiu resolver no primeiro dia. No segundo dia, 10 alunos resolveram o cavalo e o camelo. No terceiro dia, 19 alunos conseguiram resolver a imagem do cavalo e 12 alunos resolveram o camelo. Por último, durante os três dias apenas 4 dos alunos resolveram o Tangram do cisne.

4.16 Aula 38 (ScratchJr - Aula livre e quebra-cabeças)

Nessa aula livre, diferente da primeira, não tínhamos mais alunos com dificuldade em usar a ferramenta, os problemas encontrados foram com relação a alguns blocos. Assim, esses alunos foram auxiliados e os demais continuaram editando os seus projetos. Após um período de tempo, todos os alunos participaram de uma dinâmica de quebra-cabeças do alfabeto para que pudessem praticar conceitos de programação e, ao mesmo tempo, conseguissem praticar sua leitura como pode ser visto na Figura 29.

Figura 29 – Quebra-cabeças resolvido pelos alunos

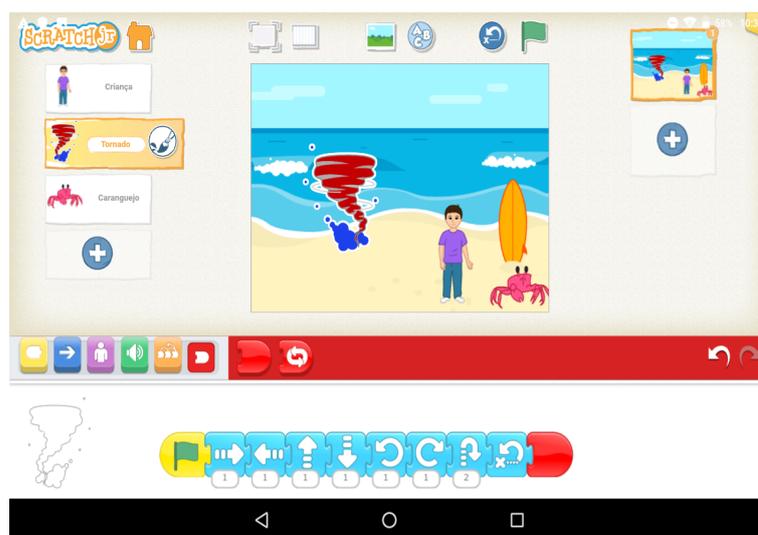


Fonte: Acervo pessoal

4.17 Aula 39 - 42 (ScratchJr - Projetos)

Durante os quatro dias, contamos com uma média de 90% da turma durante as aulas, em que passamos realizando os projetos pré-definidos. Foi identificado que os alunos progrediram bastante realizando diversas atividades diferentes e que, após replicarem os **scripts** ensinados, puderam fixar o conhecimento e abstrair, de forma mais simples, os problemas dados em sala de aula. A Figura 30 demonstra um dos projetos realizados nessa aula.

Figura 30 – Projeto na praia

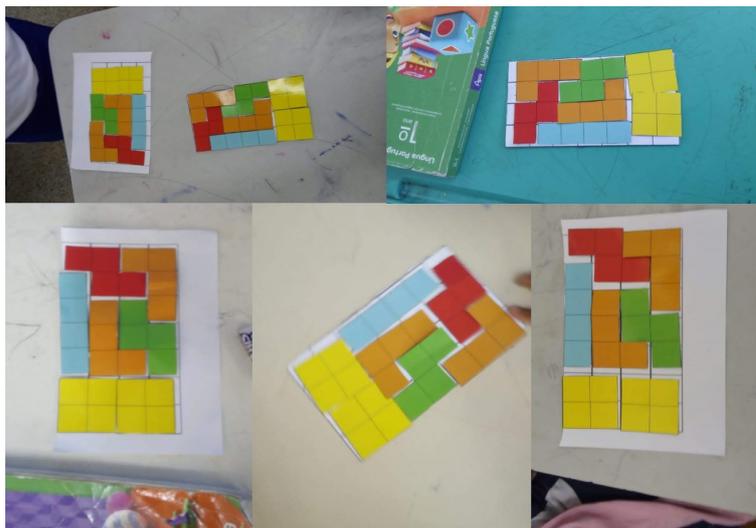


Fonte: Acervo pessoal

4.18 Aula 43 - 45 (Tetris)

Durante três dias tivemos a presença de todos os alunos da turma e trabalhamos o uso do Tetris conforme demonstrado na Figura 31. No primeiro dia, usamos um com quatro linhas e sete colunas, o que recebeu o nome de nível um; no segundo dia um com cinco linhas e sete colunas, o que recebeu o nome de nível dois; e no último dia apenas a de nível dois.

Figura 31 – Tetris resolvidos pelos alunos



Fonte: Acervo pessoal

Dos 19 alunos no primeiro dia, 14 conseguiram resolver o problema. No segundo dia, nove conseguiram resolver. No último dia, utilizamos mais uma vez os testes e os resultados, em relação ao nível um, subiram para 16 e, o nível dois, para 12. Foi identificado que o Tetris fez com que os alunos utilizassem mais o raciocínio lógico, alguns depois de um tempo não quiseram resolver mais o problema e os outros continuaram tentando, o que explica os resultados obtidos.

4.19 Aula 46 (ScratchJr - Aula livre)

Essa última aula livre do projeto foi utilizada para revisar todo o conteúdo, bem como praticar, editar e tirar dúvidas. Deve-se ressaltar que, mesmo que durante todo o projeto não tenhamos tido uma presença de 100% dos alunos, as metas de ensino traçadas foram alcançadas e isso se deu por analisarmos de forma interpretativa tudo o que os alunos fizeram no dia a dia.

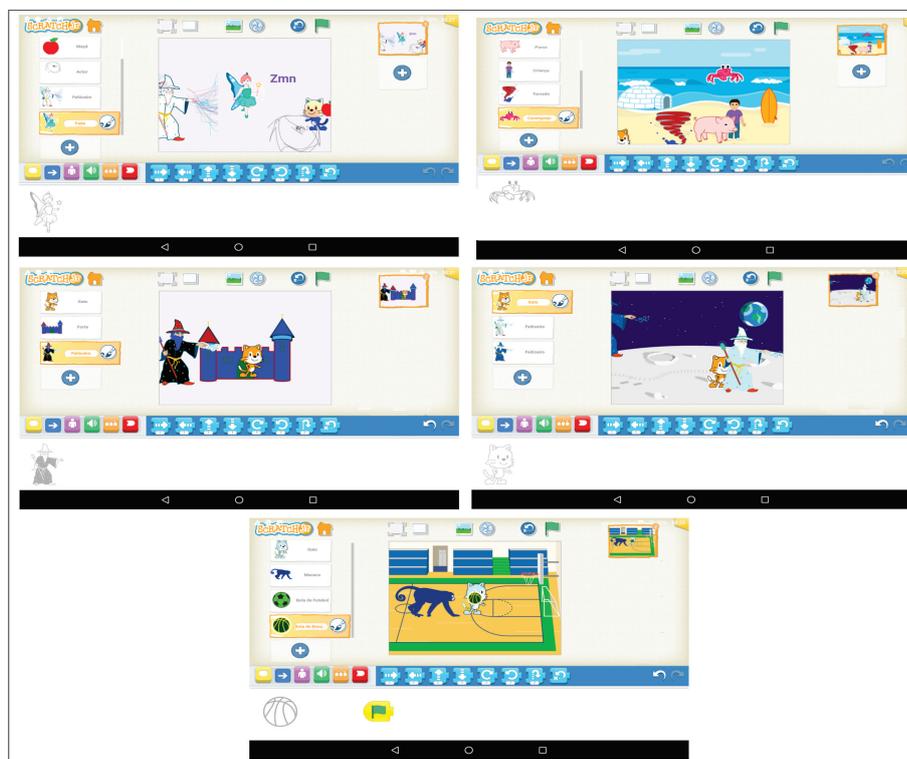
4.20 Aula 47 - 50 (Avaliação)

Para verificar qual aproveitamento obtivemos de forma objetiva, foram desenvolvidas seis atividades diferentes, com níveis de dificuldade diferentes sobre os blocos de programação utilizados no Scratch. Com ajuda da professora da turma conseguimos aplicar os testes e tirar dúvidas, uma vez que nem todos alunos sabiam ler corretamente, de forma que foi preciso ler e explicar. Além disso, foi pedido aos alunos após cada prova teórica que eles criassem **scripts** com o que já tinham aprendido até o momento.

Destacamos que cinco alunos na última avaliação, onde eles em tese deveriam criar toda

uma história interativa com o que tinham aprendido até o momento, não o fizeram como os demais alunos, mas focaram apenas no uso da criatividade ao ficarem desenhado e pintando seus respectivos cenários e personagens. A Figura 32 mostra o que cada um dos seis realizaram no ScratchJr.

Figura 32 – *Scripts* que usaram a criatividade



Fonte: Acervo pessoal

Como na avaliação prática os alunos ficavam livres para montar seus programas da maneira como achassem melhor sem que fosse dito a eles o que deveriam fazer e sim apenas auxiliando tirando dúvidas. Ao serem perguntados porque não utilizaram blocos de movimentação, transição de telas e etc, responderam que:

1. O aluno que criou um personagem do tipo mago, onde ele editou criando raios para ele e colocou saindo das mãos dele. A ideia segundo ele era: *“uma guerra entre magos e fadas e um gato com poderes”*
2. *“professor eu tentei criar um furacão onde todo mundo era teletransportado para o mar”*
3. Os demais alunos apenas criaram personagens com o intuito de editá-los no modo de pintura e desenho, quando eles eram indagados do porque não criavam *scripts* a resposta era: *“ah eu quero pintar; eu gosto de pintar; eu gosto de desenhar”*

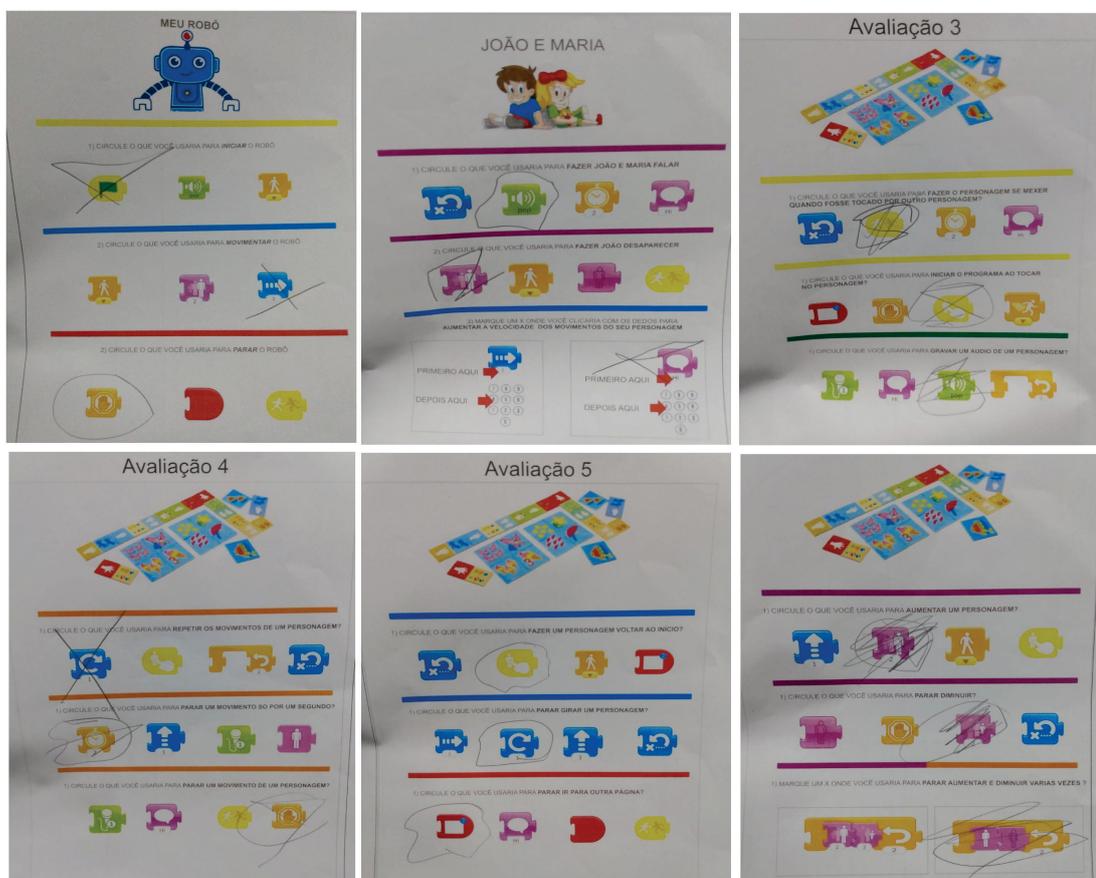
Por fim, os resultados obtidos em relação à atividade um foi de 89,47% de aproveitamento, na atividade dois foi de 63,15%, na atividade três de 73,68%, na atividade quatro de 78,94%,

na atividade 5 de 84,21% e na atividade seis de 68,42%, na Figura 33 mostra alguns dos testes aplicados. Concluímos que os resultados quanto ao ensino de programação foram alcançados de forma satisfatória dentro do planejamento e das aulas traçadas. Com relação ao desenvolvimento acadêmico deles nas outras matérias, a professora responsável pela turma aprovou todo o projeto e disse:

Meus alunos conseguiram se desenvolver melhor nos assuntos ensinados, eles conseguiram pensar um pouco mais do que de costume. Além disso, os alunos que tinham bastante dificuldade com o alfabeto, hoje já conseguem identificar com mais facilidade as letras. Acho bastante válida a iniciativa, pois não envolveu apenas tecnologia e todo o projeto utilizou de todos os recursos que nossa escola oferece. Espero que possamos realizar novamente esse projeto só que agora de forma diluída dentro de um período letivo (Pedagoga responsável ao ser entrevistada no fim do projeto).

Na figura 33 é possível visualizar foto de algum dos testes aplicados.

Figura 33 – Provas didáticas aplicadas



Fonte: Acervo pessoal

A grande quantidade de *smartphones*, que já ultrapassa mais da metade do número de habitantes do mundo, permite a acessibilidade de aprendizagem para todos. Visto que quanto maior o número de aparelhos maior será o preço competitivo, reduzindo os custos e tornando o produto mais acessível para todos. Os dispositivos móveis possuem uma grande capacidade de auxiliar o ensino de linguagens de

5

Metodologia no ensino superior

Para suprir as carências encontradas na literatura no que diz respeito ao ensino de programação por meio de dispositivos móveis como um *smartphone* no ensino superior da graduação especificamente no primeiro período, foi proposta uma abordagem de ensino baseada no construcionismo de Seymour Papert, que fosse capaz de auxiliar os alunos nesse processo de aprendizagem, buscando extrair deles habilidades como decomposição de problemas, raciocínio lógico, criatividade e um ótimo desenvolvimento. Assim, buscamos aplicar uma estratégia de ensino baseado em pequenos problemas relacionados aos conteúdos e que se aplicasse ao dia a dia do aluno. Acreditamos que a melhor forma de poder acompanhar vários alunos em uma turma sem computadores é dando o suporte dentro e fora da universidade. Assim utilizamos do aplicativos de comunicação *WhatsApp* para manter o diálogo com os alunos.

Sendo assim, podemos dividir a metodologia em dois momentos, um é antes da pandemia com o ensino voltado para o ambiente presencial e o outro durante a pandemia voltado para o EAD. A seção 5.1 mostra toda a estratégia de ensino utilizada no ambiente presencial. A seção 5.2 traz a estratégia utilizada no EAD, para o ensino de programação baseado no construcionismo de Papert.

5.1 Ensino presencial

O ambiente no qual o estudo de caso foi aplicado se dá em uma turma mista da disciplina de Introdução à Programação com carga horária de 60 horas cuja ementa possui os assuntos: entrada e saída de dados, condicionais, strings, laços de repetição, listas, matrizes, dicionários e bibliotecas de cálculos matemáticos. A turma contou com a matrícula de 53 alunos, dos quais 26 eram mulheres e 27 homens entre 18 e 22 anos de idade dos cursos de Engenharia, Matemática e Física da Universidade Federal de Sergipe (UFS), durante o semestre de 2019.2 (outubro de 2019 a março de 2020). Vale frisar que esta disciplina é pré-requisito de Cálculo

Numérico. No início de cada período é definido um laboratório e/ou uma sala sem computadores. Esta turma ficou em uma sala com um projetor portátil, um quadro branco e outro negro, sem computadores disponíveis. Coube, então, indicar algum ambiente de programação para que os discentes pudessem praticar em sala de aula e em casa. Cada docente pode escolher a linguagem de programação a ser usada. Coube também ao docente criar alguma metodologia de ensino capaz de suprir a falta de recursos. No nosso caso, foi escolhida a linguagem de programação Python que pode ser utilizada através de IDE's em *desktop, web e mobile*.

O objetivo desse estudo de caso é descrever uma estratégia de ensinar a linguagem Python utilizando um *smartphone*, além de elencar as maiores dificuldades encontradas pelos alunos tanto durante os conteúdos como também no uso dos dispositivos móveis e mostrar os benefícios e pontos fracos dessa abordagem.

Para alcançar os objetivos de ensino foi utilizado como base parte do modelo *Goal Questions Metric (GQM)*, proposto por [Basili Victor e Weiss David \(1984\)](#) e também por [Solingen D. e Berghout E. \(1999\)](#), em que foram definidas sete questões de pesquisa (QP) que fazem referência aos conteúdos necessários para que um aluno possa aprender a programar. As questões são:

- QP1. Os alunos conseguiram aprender o conteúdo de estrutura sequencial? Quais foram as dificuldades encontradas?
- QP2. Os alunos conseguiram aprender o conteúdo de estrutura de decisão? Quais foram as dificuldades encontradas?
- QP3. Os alunos conseguiram aprender o conteúdo de *string*? Quais foram as dificuldades encontradas?
- QP4. Os alunos conseguiram aprender o conteúdo de estrutura de repetição? Quais foram as dificuldades encontradas?
- QP5. Os alunos conseguiram aprender o conteúdo de tuplas, lista e matrizes? Quais foram as dificuldades encontradas?
- QP6. Os alunos conseguiram aprender o conteúdo de dicionários, tratamento de exceções e listas compostas com dicionários? Quais foram as dificuldades encontradas?
- QP7. Em quais assuntos, dentre os explorados no semestre, os alunos apresentaram maior dificuldade?

Essas questões de pesquisa foram utilizadas para definir a metodologia de aula, o planejamento e criação de atividades, servindo como métrica o percentual de acerto da resolução dos problemas feitos pelos alunos na ferramenta QPython, que foram corrigidas manualmente pelo professor responsável. Ademais, estas questões de pesquisa serão respondidas por meio da

análise desses dados, em que será utilizado conceitos de A a E para definir o nível de aprendizagem da turma. Cada questão aplicada durante o período letivo possuía os níveis de dificuldade, fácil, intermediário e difícil, o que possibilitou responder as questões de pesquisas definidas anteriormente.

Foi explicado aos alunos no primeiro dia de aula como funcionaria a metodologia de ensino e que poderiam praticar programação pelo *smartphone*. Depois, foi pedido para, que todos aqueles que possuíssem um *smartphone* com sistema operacional Android baixassem o aplicativo QPython. Para aqueles que possuíam iPhone, foi indicado o aplicativo *Python3IDE*¹ disponível na *App Store* que realiza a mesma função do QPython.

Após a instalação dos aplicativos em sala de aula, foi disponibilizado um manual contendo as instruções de como configurar a ferramenta no celular e no computador, criar e testar código no console e no editor, além de conteúdo de 135 exercícios individuais que estão disponíveis na página *web* do Python Brasil², que foram resolvidos em sala de aula. As atividades possuíam três níveis de dificuldade de acordo com a complexidade de cada questão e foram organizadas como segue na Tabela 2. Ademais, foram trabalhados 6 projetos colaborativos, que foram de autoria dos próprios alunos, que foram resolvidos em equipe dentro e fora da universidade, aplicando a teoria construcionista, segundo a qual o aluno se desenvolve melhor ao ser apresentado aos problemas relacionados ao seu cotidiano. Por fim, foi informado aos alunos que seria criado um grupo no WhatsApp para suporte acadêmico, o qual não tinha restrição de horário. Esse grupo serviria como um fórum de discussão, cujos estudantes poderiam compartilhar seus conhecimentos, tirar dúvidas, compartilhar código e receber *feedback*.

Tabela 2 – Organização das atividades

ASSUNTO	QUESTÕES	DIFICULDADE
Estrutura sequencial	Da primeira à sétima	Fácil
Estrutura sequencial	Da oitava à décima terceira	Intermediário
Estrutura sequencial	Da décima quarta à décima oitava	Difícil
Estrutura de decisão	Da primeira à décima	Fácil
Estrutura de decisão	Da décima primeira à décima oitava	Intermediário
Estrutura de decisão	Da décima nona à vigésima oitava	Difícil
String	Da primeira à quinta	Fácil
String	Da sexta à décima	Intermediário
String	Da décima primeira à décima quarta	Difícil
Estrutura de repetição	Da primeira à décima quarta	Fácil
Estrutura de repetição	Da décima quinta à trigésima segunda	Intermediário
Estrutura de repetição	Da trigésima terceira à quinquagésima primeira	Difícil

¹ Disponível para download: <https://apps.apple.com/br/app/python3ide/id1357215444>

² Disponível em: <https://wiki.python.org.br/listaDeExercicios>

Listas	Da primeira à oitava	Fácil
Listas	Da nona à décima sexta	Intermediário
Listas	Da décima sétima à vigésima quarta	Difícil
Funções	Da primeira à quinta	Fácil
Funções	Da sexta até a nona	Médio
Funções	Da décima à décima quarta	Difícil

Fonte: Acervo pessoal

Foi realizado um piloto, com quatro encontros de duas horas cada, para averiguar se a ferramenta atenderia à necessidade das aulas, bem como testar sua usabilidade na criação, edição e teste dos programas em Python. Durante este período, foram resolvidos, no *smartphone*, problemas sobre o primeiro conteúdo de estrutura sequencial.

A evolução dos alunos foi perceptível. Conseguiram resolver os problemas propostos e compartilhar os programas em Python, tanto com os colegas do grupo quanto com o professor, por meio do WhatsApp.

Depois de identificar dois alunos sem *smartphone* nem computador pessoal, foi permitido que eles utilizassem papel e lápis para fazer os exercícios fora da Universidade. Para poderem praticar em sala de aula, o docente levou dois aparelhos, um tablet e um *smartphone*, para que não fossem prejudicados e desistissem da disciplina.

Segundo Papert (1980), é preciso acompanhar cada aluno após a explicação do conteúdo e a cada exercício. Com isso, através dessa abordagem, foi possível perceber que cada aluno tem o seu tempo e método de aprendizagem, o que torna necessária a criação de novos exemplos e atividades, a depender do perfil do aluno.

Assim, na primeira unidade foram abordados os assuntos de estrutura sequencial e estrutura de decisão, expressões booleanas, alguns métodos de string como IN e NOT IN que retornam valores booleanos e cálculos matemáticos utilizando a biblioteca math. Os exemplos de expressões booleanas foram aproveitados ao ensinar sobre operadores lógicos if, elif e else.

Quanto à segunda unidade, foi aprofundado o assunto de manipulação de *string* com o ensino de estrutura de repetição. Por ser uma cadeia de caracteres que pode ser percorrida e alterada com algumas funções da própria *string*, foi importante ensinar como se utilizam os laços com for e while. Além desses, foi ensinado tuplas e listas, assim como sua relação com *strings*, que são listas de caracteres, como indexar, acessar, modificar e excluir itens.

Na terceira unidade foi ensinado como criar uma função e como reutilizá-la dentro do código, tratamento de exceções, listas compostas com dicionários e matrizes. Por fim, foi solicitado aos alunos que desenvolvessem projetos em equipe que abordassem todo o conteúdo passado em sala de aula, eles ficaram livres para criar um problema e para solucioná-lo. Assim, a

última etapa da disciplina consistia em um projeto colaborativo, especificado e implementado no QPython para aqueles que utilizavam *smartphones* ou no *IDLE* ferramenta que acompanha o Python quando configurado no computador pelos próprios discentes.

De acordo com Brehm et al. (2019), projetos colaborativos motivam os alunos e melhoram seu desempenho em conteúdos relacionados à programação. Desse modo, a turma foi dividida em grupos de cinco ou seis pessoas, para que pudessem resolver os projetos elaborados por eles mesmos. Segundo Brehm et al. (2019), ao trabalhar com projetos colaborativos, os discentes conseguem desenvolver melhor algumas habilidades como pensar de forma algorítmica, ao ter que explicar algum código e a capacidade de comunicação interpessoal. Os exercícios colaborativos, passados em sala de aula, serviram como base para a elaboração dos projetos finais da disciplina, cujo peso era 100% da nota final. O projeto deveria ser totalmente implementado pelos alunos, contemplando todos os assuntos vistos em sala de aula.

Durante as aulas o WhatsApp, foi bastante importante para que uma turma sem laboratório tivesse uma dinâmica e interação relevante, já que durante o período letivo os alunos afirmaram ter utilizado dentro e fora da sala de aula o *smartphone* em mais de 80% dos exercícios, 15% utilizaram o notebook pessoal ou ambos os recursos e os outros 5% utilizaram um dispositivo móvel e papel para programar. Assim, quando os alunos precisavam tirar dúvidas referentes a algum código, eles compartilhavam no grupo e o docente transmitia de forma instantânea no projetor. Deve-se salientar que isto acontecia quando apresentavam os projetos colaborativos. Já fora da sala de aula, as dúvidas que surgiam eram compartilhadas enviando o código direto da ferramenta QPython e o professor conseguia dar um *feedback* no mesmo momento.

Após o término de cada unidade, era passado aos alunos um questionário online, feito no *Google Forms*, para que eles pudessem expor de forma anônima as suas dificuldades ou não durante o aprendizado de um determinado assunto.

Foram aplicado três formulários. O primeiro encontra-se no apêndice A e é referente aos assuntos de entrada e saída de dados, formatação de saída, concatenação, blocos de condições simples e aninhados, variáveis explícitas e implícitas, operadores lógicos e manipulação de string, no total foram respondidos por 45 alunos. O segundo questionário que encontra-se no apêndice B foi aplicado referente aos assuntos de: laços de repetição simples e compostos com *for* e *while*, tuplas, listas e matrizes, respondido por 38 alunos. Antes da terceira avaliação foi aplicado um terceiro questionário que encontra-se no apêndice C que buscou identificar a opinião dos alunos no que diz respeito aos conteúdos de dicionários, funções, tratamento de exceções e listas compostas com dicionários, que foram respondidas por 33 alunos. Ao final do período foi aplicado um último questionário disponível no apêndice D respondido por 31 alunos, que tinha como objetivo coletar informações sobre a metodologia, a forma que foi aplicada, sugestões, saber quais recursos os alunos mais utilizaram, o que acharam do acompanhamento no WhatsApp e a opinião deles com relação ao QPython.

Os resultados obtidos serviram como base para comparar as respostas dadas com o

que foi produzido em sala de aula. Por fim, comparamos também as informações das turmas anteriores, no que diz respeito à assiduidade, quanto ao total de aprovados, reprovados e a quantos desistiram da disciplina.

Nas seções 5.1.1 e 5.1.2 apresentamos as ferramentas de aprendizagem utilizadas nesta metodologia de ensino, que possibilitaram o ensino de programação utilizando um *smartphone*.

5.1.1 QPython

O QPython³ é uma IDE Python para o sistema operacional Android. Nela, o usuário pode criar, editar e executar programas em um console disponível no próprio aplicativo. Isso é possível através de um interpretador Python versão 3.2.1 (versão utilizada na época) e de uma Biblioteca SL4A, responsável por acessar recursos do Android, como câmera, sensores, APIs de mídia, SMS, dentre outras qpython.

O QPython foi desenvolvido pela comunidade QPython e possui código aberto. Para fazer o download da IDE, a partir de dispositivos Android, basta acessar a loja de aplicativos do Google Play. É possível contribuir e consultar o código fonte da ferramenta através da página do GitHub⁴.

A ferramenta contém recursos como: intérprete offline do Python 3, suporta a execução de vários tipos de projetos como *console*, *SL4A*⁵, *webapp*, leitor de *QRcode*, camada de *script* integrada e estendida, ambiente de tempo de execução e editor.

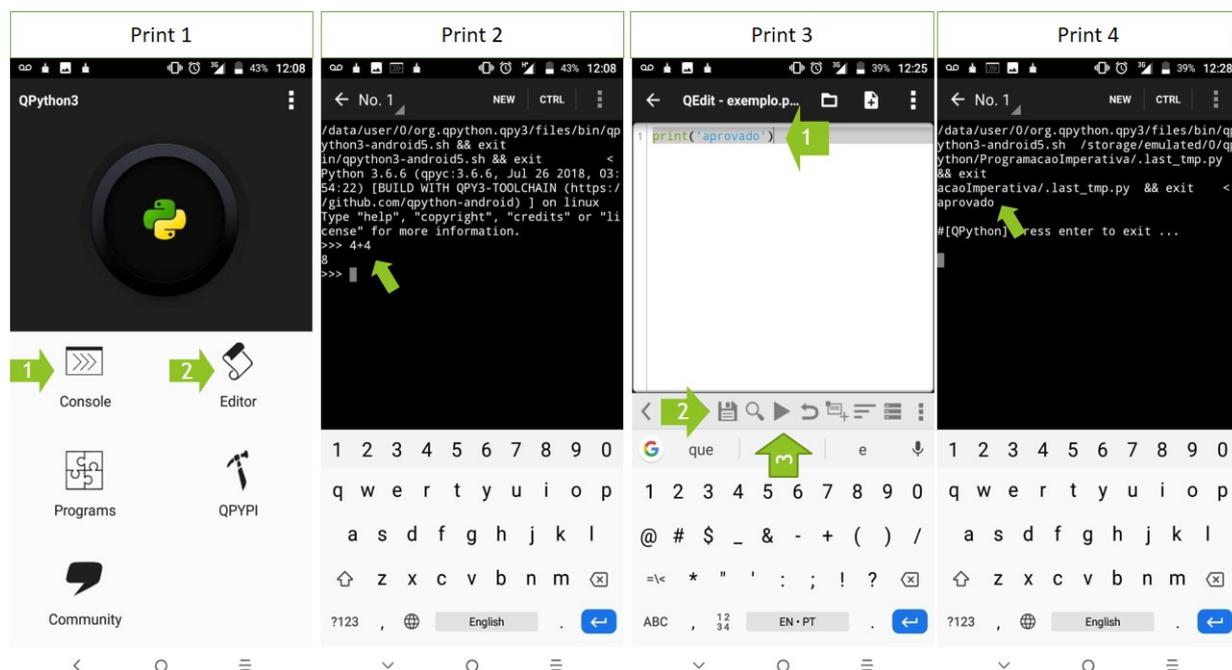
A Figura 34 mostra quatro capturas de tela do QPython. A primeira apresenta a tela inicial onde é possível escolher entre programar diretamente via console ou editar um código fonte e depois testá-lo. A segunda mostra um código sendo executado diretamente via console. Já a terceira se refere a um programa simples que pode ser salvo em uma pasta definida pelo usuário. A quarta mostra a execução via console do código feito na terceira captura de tela.

³ Link para download da versão 3L: <https://play.google.com/store/apps/details?id=org.qpython.qpy3>

⁴ Link de acesso ao projeto no GitHub: <https://github.com/qpython-android/qpython>

⁵ *Script Layer for Android*, uma biblioteca que permite ao usuário criar e executar em dispositivos *Android* programas escritos em várias linguagens populares, incluindo Python.

Figura 34 – Screenshots de algumas telas do QPython



Fonte: Acervo pessoal

5.1.2 WhatsApp

O WhatsApp⁶ é um aplicativo de troca de mensagens instantâneas multiplataforma, que possui uma versão *desktop* e *web*. Com esse aplicativo, é possível enviar mensagens de texto, áudio e voz, imagens, vídeos e anexar arquivos. Além disso, também é possível criar grupos de conversas com temas específicos.

Alguns trabalhos utilizaram o WhatsApp como ferramenta de suporte ao ensino em vários níveis escolares, no ensino fundamental Lima et al. (2016), no ensino médio, Lopes Cristiano e Vaz (2016), como também no ensino superior, mais especificamente no ensino de programação Martins e Gouveia (2019).

No caso de Martins e Gouveia (2019) eles apresentaram uma experiência de utilização do aplicativo WhatsApp para a realização de atividades fora da universidade com foco no ensino de programação. A fim de verificar o potencial do aplicativo, eles criaram um grupo que serviu como um fórum de dúvidas e respostas em horários pré estabelecidos e ao final do semestre foi verificada sua eficácia no ponto de vista de 32 alunos que responderam a um questionário online. Constatou-se que o WhatsApp pode ser utilizado de forma efetiva como uma extensão da sala de aula.

⁶ Link para download do WhatsApp: <https://www.WhatsApp.com/>

5.2 Ensino a distância

O estudo de caso foi aplicado em uma turma de Introdução à Programação com alunos de cursos diferentes, com carga horária de 60 horas. Eram obrigatórios no mínimo 50% das aulas serem síncronas nas quais o professor interage diretamente com o aluno online, e os outros 50% assíncronas sem interação direta. Os assuntos abordados foram: entrada e saída de dados, condicionais, strings, laços de repetição, listas, matrizes, dicionários, listas compostas, bibliotecas de cálculos matemáticos e funções, e contou com a matrícula de 50 alunos, dos quais 26 eram mulheres e 24 homens entre 18 e 30 anos de idade dos quais 60% (30 alunos) fazem parte do curso de Química Industrial, 10% (5 alunos) do curso de Engenharia Química, outros 10% do curso de Física, mais 8% do curso de Engenharia de Petróleo que equivale a 4 alunos, por fim 6% de Engenharia Civil, 4% de Engenharia de Alimentos e 2% de Ciências Atuariais (3, 2 e 1 alunos respectivamente) da Universidade Federal de Sergipe (UFS). Esse estudo de caso foi realizado no segundo semestre do ano de 2020 iniciando-se no dia 30 de março e sendo finalizado no dia 3 de agosto de 2021.

Antes do período começar foi traçado todo o planejamento e estratégia que seria utilizada durante as aulas. Desta forma, para que fosse possível inserir uma nova metodologia de ensino durante as aulas EAD, fez-se necessário primeiro realizar um estudo piloto na turma anterior. Assim, durante primeiro semestre do ano letivo de 2020 que foi realizado entre outubro do mesmo ano até janeiro de 2021 foi aplicado à primeira turma de ensino de introdução a programação totalmente EAD devido a pandemia. O período mencionado serviu de aprendizagem e de adaptação para todos que estavam acostumados ao ambiente presencial e foi dentro dessa nova perspectiva que fui indagado sobre como manter um ensino EAD no mesmo nível do presencial, além de ter uma abordagem construcionista, quais ferramentas utilizar e como deveria ser a melhor abordagem de ensino para que o rendimento dos alunos continuasse adequado e todos desfrutassem de um ensino amplo.

Desta forma, optou-se por continuar com o mesmo material didático referente a como programar utilizando um dispositivo móvel, como também deu-se um maior foco a ferramentas colaborativas para o ensino de programação. Assim, foi utilizado a linguagem Python em conjunto com uma IDE online de programação que é o Replit⁷, onde é possível criar um projeto e dentro dele termos um ou mais arquivos de código fonte. A vantagem observada durante as aulas assíncronas é do aluno poder acessar o mesmo código fonte em diversos aparelhos diferentes desde um *smartphone* até um computador *desktop*.

A abordagem utilizada no estudo piloto e os resultados obtidos com essa turma do primeiro semestre de 2020 possibilitaram que fosse desenvolvida uma estratégia de ensino capaz de manter o nível de aprendizagem dos alunos equiparados ao ensino presencial. Como a mesma aplicação foi realizada na turma seguinte, descrevemos aqui a metodologia e os resultados

⁷ Disponível em: <https://Replit.com/>

obtidos com a turma do segundo semestre de 2020.

Assim, o objetivo desse estudo de caso é descrever uma estratégia de ensinar a linguagem Python utilizando um *smartphone*, *tablet*, e/ou computador, além de elencar as maiores dificuldades encontradas pelos alunos tanto durante os conteúdos como também no uso dos dispositivos móveis e mostrar os benefícios e pontos fracos dessa abordagem.

Para alcançar os objetivos de ensino foram criadas oito questões de pesquisas (QP) que fazem referência aos conteúdos necessários para que um aluno possa aprender a programar. As questões são:

- QP1. Os alunos conseguiram aprender o conteúdo de estrutura sequencial? Quais foram as dificuldades encontradas?
- QP2. Os alunos conseguiram aprender o conteúdo de estrutura de decisão? Quais foram as dificuldades encontradas?
- QP3. Os alunos conseguiram aprender o conteúdo de *string*? Quais foram as dificuldades encontradas?
- QP4. Os alunos conseguiram aprender o conteúdo de estrutura de repetição? Quais foram as dificuldades encontradas?
- QP5. Os alunos conseguiram aprender o conteúdo de tuplas, lista? Quais foram as dificuldades encontradas?
- QP6. Os alunos conseguiram aprender o conteúdo de matrizes, dicionários, tratamento de exceções e listas compostas com dicionários? Quais foram as dificuldades encontradas?
- QP7. O uso de projetos colaborativos facilitaram o desenvolvimento dos alunos no ensino EAD?
- QP8. A abordagem construcionista no EAD, facilitou o desenvolvimento dos alunos? E quais os principais pontos a serem destacados sobre a abordagem? Quais foram as maiores dificuldades encontradas pelos alunos?

Essas questões de pesquisa foram utilizadas para definir a metodologia de aula, o planejamento e criação de atividades, servindo como métrica o percentual de acerto da resolução dos problemas feitos pelos alunos no Replit, corrigidos de forma manual. Ademais, estas questões de pesquisa serão respondidas por meio da análise desses dados, em que serão utilizados cinco conceitos de notas insuficientes a notas ótimas para definir o nível de aprendizagem de cada aluno e da turma de forma geral.

Assim, foi explicado aos alunos, no primeiro dia de aula, como funcionaria a metodologia de ensino e que poderiam praticar programação pelo *smartphone*, *tablet* ou computador. Depois,

foi pedido para que todos os alunos criassem uma conta no Replit e enviassem os seus respectivos usuários para o professor.

Após os alunos terem criado suas contas no Replit, foi disponibilizado um manual sobre como aprender a programar do zero ao intermediário, além de *links* com videoaulas disponíveis no YouTube para servir como material de apoio durante o período letivo. Por fim, foi explicado aos alunos que a cada assunto ensinado haveria uma bateria de exercícios a serem resolvidos para fixação do conteúdo programado uns como parte da nota e outros a serem desenvolvidos nas aulas assíncronas totalizando 134 exercícios individuais dos quais 100 ficaram para os alunos desenvolverem como parte da nota e os outros 34 foram trabalhados de forma colaborativa, esses últimos estão disponíveis na página *web* do Python Brasil⁸.

Foi informado ainda aos alunos que seria criado um grupo no WhatsApp para suporte acadêmico, o qual não tinha restrição de horário. Esse grupo serviria como um fórum de discussão, cujos estudantes poderiam compartilhar seus conhecimentos, tirar dúvidas, compartilhar código e receber *feedback*. Para um melhor aproveitamento das aulas assíncronas por meio da metodologia construcionista foi solicitado aos alunos que se dividissem em grupos de até cinco pessoas, com o objetivo de realizar pequenos projetos colaborativos e facilitar o acompanhamento do professor com os alunos envolvidos. Ademais, foram trabalhados 8 projetos colaborativos que envolviam todos os conteúdos abordados durante o período, estes foram de autoria dos próprios alunos, e foram resolvidos em equipe aplicando a teoria construcionista, segundo a qual o aluno se desenvolve melhor ao ser apresentado aos problemas relacionados ao seu cotidiano.

Assim, o período foi dividido de duas formas a primeira unidade contaria com o ensino de 70% o conteúdo programado: os assuntos de estrutura sequencial e estrutura de decisão, expressões booleanas, cálculos matemáticos utilizando a biblioteca `math`, blocos de condicionais simples e aninhados lógicos, com `if`, `elif` e `else`, isso na primeira parte do conteúdo. Na segunda parte foi aprofundado o assunto de manipulação de string com o ensino de estrutura de repetição `For`. Em seguida foi abordado o uso de `while` dentro dos laços de repetição, para aplicar o conteúdo abordado até o momento foi ensinado tuplas e listas, assim como sua relação com *strings*, que são listas de caracteres, como indexar, acessar, modificar e excluir itens. A primeira prova abordou todos os conteúdos citados anteriormente.

Na segunda unidade, foram ensinados os outros 30% do conteúdo que consistem no uso de matrizes, de dicionários e de listas compostas com dicionários, além do tratamento de erros e o uso de funções que o próprio aluno pode criar para encapsular seus códigos. Como são assuntos mais complexos estes ficaram para a segunda prova que seria da seguinte forma: com os grupos já criados no início do período foram mostrados dois modelos de projetos que eles deveriam criar e resolver em equipe, desde o processo de reuniões, relatórios, definição de atividades de cada integrante, como a criação dos problemas relacionado ao cotidiano de cada um e sua respectiva solução tecnológica em formato de algoritmo na linguagem Python. Todos deveriam criar o

⁸ Disponível em: <https://wiki.python.org.br/listaDeExercicios>

documento textual no Google Docs e a solução no Replit e ambas foram compartilhadas com o professor para auxílio diário e durante as aulas síncronas e assíncronas. Assim, a última etapa da disciplina consistia em um projeto colaborativo, especificado e implementado pelos próprios discentes.

Durante as aulas o WhatsApp foi bastante utilizado visto que os alunos tinham acesso direto a mim. Eles conseguiam tirar dúvidas mais rápido e a qualquer hora do dia, quer fosse no grupo geral, quer fosse no subgrupo com as equipes, e quer fosse diretamente com o professor. Além disso, utilizar o Replit, o Google Meet e o WhatsApp em conjunto facilitou a aprendizagem colaborativa como pode ser visto no capítulo 7 na seção de resultados da aplicação da abordagem construcionista no ambiente EAD.

Após o término de cada unidade, era passado aos alunos um questionário online, feito no *Google Forms*, para que eles pudessem expor de forma anônima as suas dificuldades ou não durante o aprendizado de um determinado assunto.

Após o término de cada unidade, era passado aos alunos um questionário online, feito no *Google Forms*, para que eles pudessem expor de forma anônima as suas dificuldades ou não durante o aprendizado de um determinado assunto. Com essas informações foram traçadas novas formas de explicar assuntos e exercícios nas aulas síncronas e assíncronas.

Foram aplicados três formulários. O primeiro encontra-se no apêndice E que é referente aos assuntos de entrada e saída de dados, formatação de saída, concatenação, blocos de condições simples e aninhados, variáveis explícitas e implícitas, operadores lógicos e manipulação de string, no total foram respondidos por 27 alunos. O segundo questionário que encontra-se no apêndice F foi aplicado referente aos assuntos de: laços de repetição simples e compostos com *for e while*, tuplas e listas, respondido por 19 alunos. Antes da segunda prova foi aplicado um terceiro questionário que encontra-se no apêndice G que buscou identificar a opinião dos alunos no que diz respeito aos conteúdos de matrizes, dicionários, funções, tratamento de exceções e listas compostas com dicionários, que foram respondidas por 28 alunos. Ao final do período foi aplicado um último questionário disponível no apêndice H respondido por 32 alunos, que tinha como objetivo coletar informações sobre a metodologia, a forma que foi aplicada, sugestões, saber quais recursos os alunos mais utilizaram, o que acharam do acompanhamento no WhatsApp e a opinião deles com relação ao Replit, como também a opinião no que diz respeito aos projetos colaborativos, como também com relação a abordagem de ensino utilizada.

Os resultados obtidos serviram como base para comparar as respostas dadas com o que foi produzido durante as aulas online. Por fim, comparamos também as informações das turmas do mesmo período letivo, no que diz respeito a alunos trancados, reprovados e o total de aprovados.

Nas seções 5.2.1 é apresentada a ferramenta de programação utilizada, na seção 5.2.2 é apresentada a ferramenta colaborativa para o ensino de programação online e disponibilização

de materiais de aula, códigos fontes, aplicações de formulários e aplicação de prova online. Vale salientar que o WhatsApp já foi apresentado na seção 5.1.

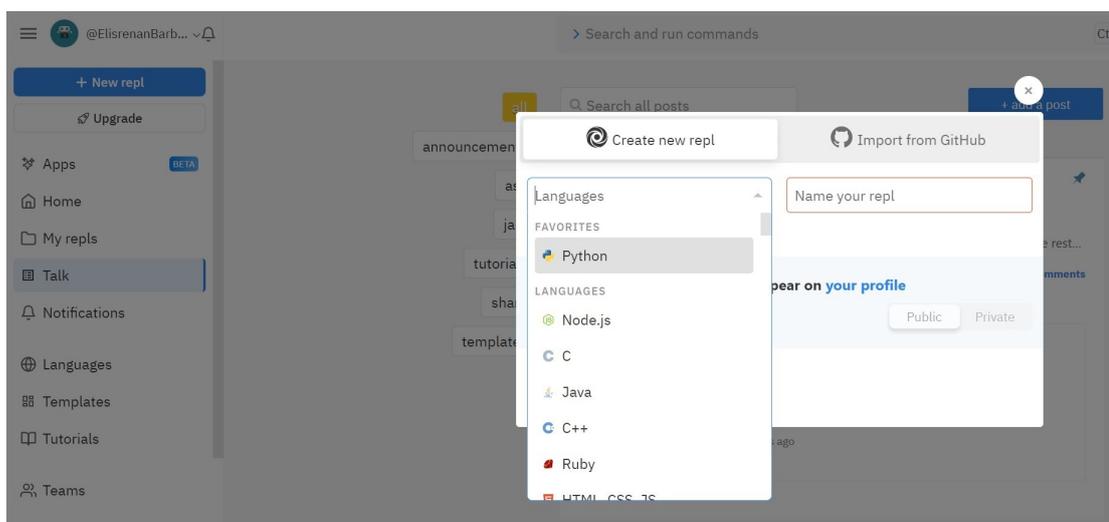
5.2.1 Replit

Na era de novas tecnologias inteligentes, aplicativos cada vez mais robustos e vários serviços de sistemas online que proporcionam um ensino cada vez mais interativo, fazem-se necessárias novas abordagens além do método tradicional de ensino de introdução a programação que possa evitar a evasão acadêmica, principalmente se forem alunos do primeiro período.

Assim, mostramos a experiência obtida no curso introdutório de Ciência da Computação, juntando aprendizagem colaborativa com a metodologia construcionista e um ambiente de diálogo como o WhatsApp, para que os alunos possam aprender usando plataformas interativas, ferramentas, tecnologias, sistemas e serviços disponíveis para eles e colaboração dentro e entre grupos. Para possibilitar um melhor desenvolvimento dos alunos foi utilizado no ensino EAD o Replit, bem como uma apostila online. Foi projetada uma abordagem baseada na construção de vários exercícios em sala de aula, atribuições de atividades, pequenos projetos baseados em laboratório com códigos de exemplo e resultados esperados, com pequenos grupos de alunos de uma mesma turma. Também, no meio do semestre, introduzimos a aprendizagem colaborativa por meio do trabalho em equipe em projetos construcionistas.

Assim no Replit os alunos podem: criar projetos, editar e testar o código em qualquer lugar e de qualquer dispositivo, como também trabalhar de forma colaborativa. Na Figura 35 é possível ver a criação de um projeto, lembrando que com o Replit é possível programar em diversas linguagens de programação.

Figura 35 – Screenshot da uma das telas do Replit

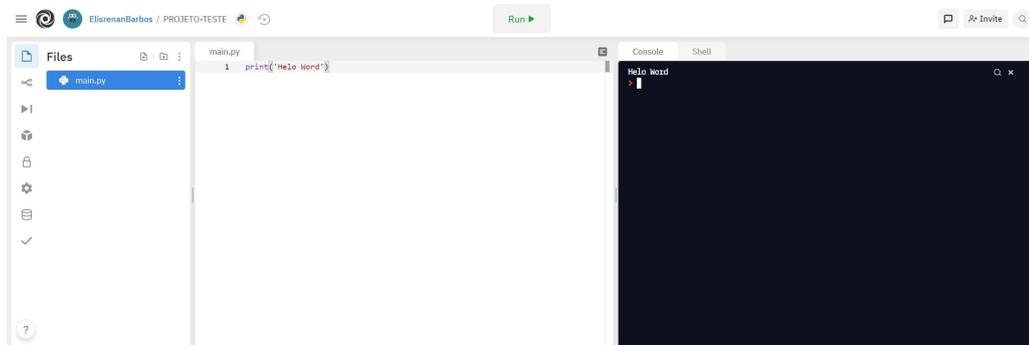


Fonte: Acervo pessoal

A Figura 36 mostra a tela onde os alunos desenvolvem seus projetos, podendo programar

desde coisas simples até coisas complexas. Vale salientar que, os alunos podem inserir outros integrantes, conseguem criticar, complementar, melhorar o trabalho dos colegas e aprender.

Figura 36 – *Screenshot* da uma das telas do *Replit*



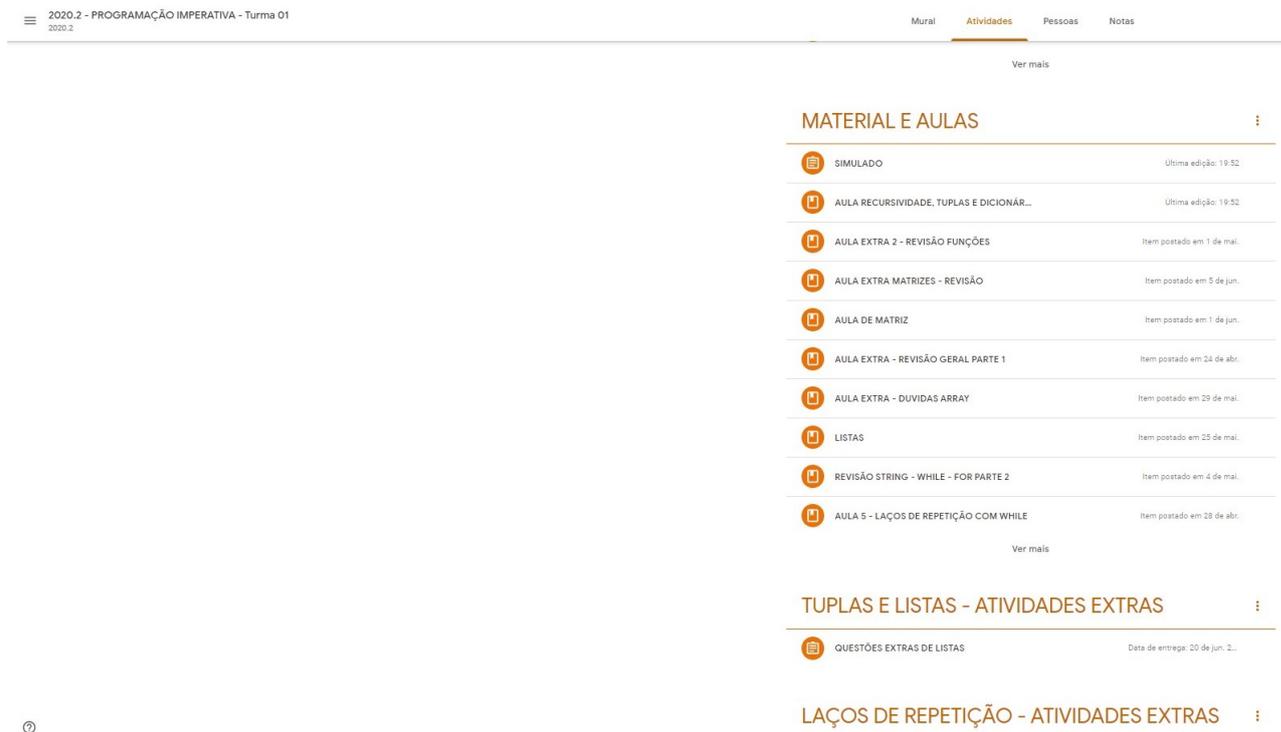
Fonte: Acervo pessoal

Por fim, foi possível analisar os efeitos do uso conjunto das ferramentas descritas anteriormente. As pesquisas realizadas em três semestres diferentes, possibilitaram uma avaliação inicial dos resultados da pesquisa e desempenho (projeto final e notas finais dos alunos), estas mostram evidências de que o uso de smartphones pode ser um instrumento auxiliar no processo de ensino aprendizagem e acompanhamento dos alunos. Mas, destacamos que é necessário uma pesquisa experimental com uma turma de controle para realizar qualquer afirmativa do ponto de vista científico.

5.2.2 Google Classroom

Durante a pandemia a tecnologia teve um crescimento sobre sua utilização no dia a dia das pessoas de forma muito importante. O Google Classroom foi bastante utilizado em todos os estágios da educação desde o ensino básico até o superior. Nesse estudo de caso não foi diferente, seguimos a ferramenta disponibilizada pela a universidade para ministrar as aulas, disponibilizar material de aula, passar atividades e aplicar provas como pode ser visto na Figura 37. Essa aplicação se torna muito útil a partir do momento que ela é acessível através de vários tipos de dispositivos, desde o *smartphone* até o *desktop*.

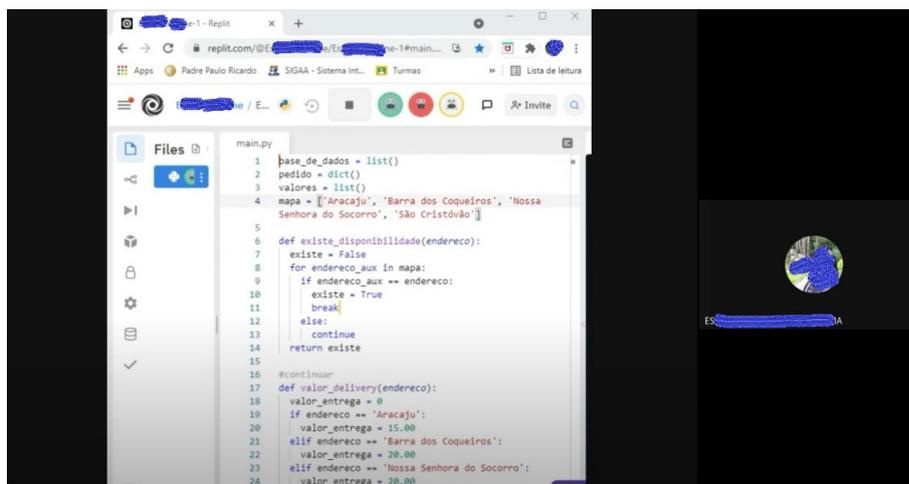
Figura 37 – Screenshot da organização dos materiais de aula no Classroom



Fonte: Acervo pessoal

Utilizamos a ferramenta para executar a sala invertida com o método construcionista. Como foi dito anteriormente na seção 6.2 nós buscamos implementar a estratégia onde o aluno é o principal responsável pelo seu desenvolvimento e o professor é o responsável por direcioná-lo nesse caminho, criando situações que desenvolvam seus potenciais de aprendizagem. Então a ferramenta que mais utilizamos do Classroom foi o Google Meet, pois com ela conseguimos fazer a sala invertida uma vez que o aluno consegue apresentar sua tela do celular e/ou computador, como pode ser visto na Figura 38 onde temos um grupo de alunos realizando uma apresentação. Durante as aulas síncronas usamos a ferramenta citada para lecionar as aulas, trabalhar exemplos de exercícios, já nas aulas assíncronas que eram divididas por horários de quinze minutos para cada grupo responder a uma determinada atividade passada, como eram 8 grupos totalizou 2h de aula, vale lembrar que nessa aula não havia obrigatoriedade do aluno estar presente, porém como eram aulas rápidas facilitava a participação de todos os envolvidos.

Figura 38 – Screenshot de um grupo de alunos apresentando um código no Google Meet



Fonte: Acervo pessoal

Foi possível observar que ao realizar um agrupamento produtivo defendido por [PIAGET \(1970\)](#) ao dividir os alunos em pequenos grupos e realizar uma heterogeneidade de conhecimentos de estudantes com conhecimentos distintos, foi uma estratégia que facilitou a aprendizagem de todos, já que assim foi possível ter uma interação direta com os educandos e acompanhamento do desenvolvimento e da dificuldade de todos possibilitou ainda perceber que aprenderam uns com os outros. Outro ponto a destacar é a disponibilidade de gravação das aulas fazendo com que cada aula se tornasse um material para revisão dos conteúdos lecionados.

Foi obtido êxito na aplicação do construcionismo no modo EAD, visto que, as correções de exercícios eram feitas de forma simultânea a entrega quando em aulas assíncronas e quando síncronas os exercícios eram analisados durante a aula. Vale destacar que isso só foi possível ser feito dessa forma e maneira pelo fato do professor ter apenas uma única turma sendo lecionada no referido período.

Diante disso, foi possível acompanhar todos os alunos, identificar dificuldade e saná-las, criar situações que potencializassem o desenvolvimento dos alunos por meio da sala invertida, como também desenvolvemos nos alunos uma das coisas principais do método que é aplicar o conhecimento em algo relacionado ao cotidiano do aluno por meio dos projetos colaborativos.

6

Aplicação no ensino superior

Nesse capítulo é apresentada a estratégia de ensino aplicada no ensino presencial como pode ser vista na seção 6.1, como também no ensino EAD que pode ser visto na seção 6.2.

6.1 Aplicação no ensino presencial

A Tabela 3 expõe de forma objetiva como a metodologia de ensino de programação foi aplicada durante o semestre de 2019.2 (outubro de 2019 a março de 2020). A estratégia descrita abaixo foi montada de acordo com a carga horária, visto que, duas horas de aula eram dadas às terças-feiras e duas horas dadas às quintas-feiras. Nas primeiras duas horas da semana era priorizado passar o conteúdo. Nas últimas duas horas eram realizadas atividades com acompanhamentos. Caso as atividades não fossem solucionadas na quinta feira, os alunos teriam mais quatro dias até a próxima aula para resolver os problemas propostos, sempre que precisassem poderiam tirar dúvidas no grupo ou por mensagem privada ao professor.

Tabela 3 – Estratégia de ensino utilizada

AULA	CH	ASSUNTO	ESTRATÉGIA UTILIZADA
1 e 2	2h	Composição de Programas: instruções de entrada e saída, operadores, expressões identificadores, constantes, variáveis e atribuição. Entrada e saída de dados.	Foi explicado o assunto e propostas atividades simples para os alunos fazerem. Ao final, foi passada uma lista com uma lista de exercícios focados em estrutura sequencial.
3 e 4	2h	Exercícios sobre composição de programas	Essa aula foi apenas para resolução de problemas e acompanhamento do professor aos alunos, a fim de identificar dificuldades. Os alunos ficaram para responder em casa as atividades que não foram concluídas durante a aula, com liberdade para tirar dúvidas pelo WhatsApp.
5	2h	Estruturas de decisão if	Por ser um assunto mais complexo, foi explicado e passados exemplos e feitas atividades simples sem o acompanhamento do professor. Ao final da aula foi passada uma lista exercícios do mais simples ao mais complexo para ser respondido em sala de aula.
6, 7, 8 e 9	4h	Exercícios de estruturas de decisão if	Durante quatro aulas o professor realizou um acompanhamento aluno por aluno, identificando os que possuíam mais dificuldades e buscando sanar na própria aula. Lembrando que nessas aulas é importante anotar as maiores dificuldades encontradas, pois caso não haja tempo hábil para ajudar os alunos, faz-se necessário resolver na aula seguinte.
10	1h	Estruturas de decisão compostas if	Foi passado em um período de no máximo um hora, o conteúdo de estruturas compostas de decisão, realizado atividades simples para fixar o conhecimento e nas aulas seguinte foi utilizada para resolver uma lista de problemas.

11, 12 e 13	3h	Exercícios de estruturas de decisão compostas <i>if</i>	Como já citado anteriormente essa aulas de atividades o foco é total em acompanhar os alunos, para que seja possível extrair o máximo de informações possíveis, bem como resolver as dificuldades dos alunos mostrando outros exemplos de códigos, ou ainda buscando criar situações exemplos que faça parte do cotidiano deles. Ainda nessa aula, deve ser aplicado o primeiro questionário de acompanhamento o mesmo deve ser enviado aos alunos pelo grupo do WhatsApp.
10 e 11	2h	Revisão para a prova	Na aula de revisão deve-se levar em consideração todas as dificuldades expostas pelos alunos e tentar saná-la da melhor forma possível, sempre buscando criar situações que o aluno consiga identificar e abstrair aquilo que não está conseguindo absorver.
12 e 13	2h	Primeira avaliação	A prova deve ser aplicada como de costume com problemas contextuais para que os alunos interpretem e deem soluções, buscando sempre variar as questões. Deve-se ter questões que avalie o aluno desde o conteúdo mais simples ao mais complexo.
14	1h	Manipulação de <i>strings</i>	Como o assunto seguinte é laços de repetição, optou-se por aprofundar os conhecimentos sobre manipulação de string no início da segunda unidade, visto que, esse conteúdo possui semelhança com listas e em alguns casos faz-se necessário o uso de comandos de repetição.
15 e 16	2h	Exercícios de manipulação de <i>strings</i>	Foi aplicado durante duas aulas uma lista de exercícios apenas com foco em <i>strings</i> . Como alguns exercícios eram mais complexos esses foram solicitados para serem feitos em casa com apoio do professor pelo WhatsApp sempre que necessário.
17	1h	Estrutura de repetição <i>for</i>	Devido os comandos de repetição <i>for</i> serem mais simples de serem aprendidos, visto que para realizar um laço é necessário apenas entender sobre numeração. Desta forma, foi explicado o assunto, quando se deve aplicá-lo e passado exemplos e atividades simples em sala de aula.

18 e 19	2h	Exercícios de estrutura de repetição for	Nesta aula, todos os exercícios abordados, buscaram fazer com que os alunos praticassem esse conteúdos. Lembrando que o professor deve acompanhar todos os alunos durante essas duas horas, como de costume os alunos não conseguiam realizar todas as atividades dentro da sala de aula, assim, os mesmos ficavam responsáveis por responder até o próximo encontro.
20	1h	Estrutura de repetição com while	Da mesma forma que com o for, o while foi explicado.
21 e 22	2h	Exercícios de estrutura de repetição com while	Nessa aula foi entregue mais uma lista de exercícios para serem resolvidos em sala de aula, como aconteceu com o for, algumas atividades ficaram para serem resolvidas em casa.
23	1h	Tuplas	O assunto de tuplas foi escolhido para iniciar os conteúdos de estruturas de dados do Python por ser o mais simples dos assuntos e o que iria permitir ao meu ver que todos os alunos conseguissem compreender com mais facilidade os demais conteúdos. Nessa aula foi aplicado atividades, exemplos que os alunos pudessem resolver dentro da sala de aula.
24 e 25	2h	Listas	Após o ensino de Tuplas foi aplicado o conteúdo de listas e dado inicio a resolução de 24 exercícios que proporciona-se aos alunos entendimento do assunto. Nas duas aulas seguintes os alunos utilizaram para responder a maior parte das questões.
26 e 27	2h	Exercícios sobre listas	Nesta aula os alunos tentaram resolver as demais questões, sempre com o acompanhamento do professor. Nessa aula, foi passado o segundo questionário aos alunos para que eles expusessem suas dificuldades e o professor pudesse saná-las nas aulas de revisão.
28 e 29	2h	Revisão para a segunda prova	A aula de revisão antes da prova, foi utilizada como dito anteriormente para sanar as dúvidas com exercícios, explicações que demonstrassem aos alunos como poderiam resolver os problemas.
30 e 31	2h	Prova segunda unidade	Como na primeira unidade foi aplicado uma prova com questões diversas e que abordasse uma questão de cada assunto e uma com todos os assuntos, com níveis de dificuldades da mais simples a mais complexas.

32	1h	Matrizes	Um dos assuntos mais complexos de introdução a programação é o de matrizes, esse ficou para a terceira unidade e foi abordado com exemplos que mostrassem quando e como se aplica o seu conteúdo. Devido à sua alta complexidade, foi solicitado aos alunos que se dividissem em grupos de até cinco pessoas para que as atividades pudessem ser respondidas.
33 e 34	2h	Exercícios sobre matrizes	Foram passadas cinco atividades que foram desenvolvidas em grupo pelos alunos, buscando introduzir a metodologia de projetos colaborativos.
35	1h	Listas compostas com dicionários	Foi ensinado primeiro o conteúdo de dicionários e após o uso de listas para armazenar dicionários
36	1h	Exercícios sobre dicionários	Nessa aula assim como na aula 29 e 30, os mesmos grupos resolveram cada um cinco questões de forma colaborativa.
37	1h	Funções do Python, criação de funções e tratamento de erros	O ultimo conteúdo ensinado foi o de funções, para que os alunos conseguissem automatizar todos os seus códigos e como tratar os erros de exceções do Python.
38	1h	Exercícios sobre funções do Python, criação de funções e tratamento de erros	Nessa ultima aula prática foi abordado mais cinco questões resolvidas em grupos pelos alunos.
39	1h	Projetos colaborativos	Foi passado aos alunos um modelo de projeto colaborativo que eles deveriam implementar o mesmo fez parte da nota de todos os alunos. Eles deveriam formular seus próprios problemas e que este estivesse relacionado ao seu cotidiano, ficando livres para serem criativos, desde que houvesse ao menos um conteúdo de cada unidade nas soluções propostas.

40 a 46	7h	Acompanhamento dos projetos colaborativos	Durante sete aulas presenciais os alunos utilizaram para desenvolver seus respectivos projetos, o professor acompanhou e tirou dúvidas de todos durante duas semanas e sempre que os alunos tinham dificuldades fora da universidade foi dada toda a liberdade para que eles interagissem, assim foi solicitado a criação de sub-grupos no WhatsApp com apenas os membros das equipes para que facilitasse a comunicação do professor com cada grupo.
47 a 50	4h	Apresentação dos projetos	Os projetos colaborativos foram analisados no dia a dia de aula o que permitia identificar quem estava ou não desenvolvendo, passando assim desde uma avaliação parcial até a avaliação final na apresentação dos trabalhos. Foi então solicitado aos alunos que apresentassem os projetos como se estivessem vendendo algo, ou seja, que falassem com propriedade suas ideias e suas soluções. Buscamos assim, uma abordagem baseada no mercado de trabalho, com projetos colaborativos e apresentações voltadas para a regra de negócio.

Fonte: Acervo pessoal

Os formulários aplicados seguem a seguinte estratégia: uma ou duas questões perguntando de forma objetiva as dificuldades de um determinado assunto e uma questão perguntando de forma subjetiva com o objetivo de saber a opinião do aluno e evitando, assim, algum viés nos resultados objetivos, pois dessa forma é possível analisar o que foi respondido com o que foi descrito por eles.

Para se obter informações que pudessem ser analisadas de forma estatística, em conjunto com uma análise qualitativa do dia a dia de aula, foram passadas para os 50 alunos um total de 135 questões que foram respondidas, de forma a gerar 6.486 programas individuais que foram analisadas uma a uma sem o uso de qualquer software de automação, principalmente por causa do contexto apresentado onde os alunos estavam programando em sua maioria pelo celular e alguns outros no notebook inviabilizava o uso de qualquer ferramenta de análise de código. Assim, como forma de acompanhar o desenvolvimento e identificar a dificuldade dos alunos nos conteúdos optou-se por uma análise linha a linha de código.

A última avaliação, feita com os alunos, foi em cima dos projetos realizados em grupos. Vale mencionar que todos os projetos foram avaliados por completo, bem como pela entrega que cada aluno tinha que fazer dentro de um cronograma especificado anteriormente a eles.

Assim, todos os projetos eram divididos em entregas de códigos, em que cada membro do grupo era escolhido de forma aleatória para explicar o que fez e, ao final de cada projeto, o grupo apresentava toda a lógica desenvolvida em dias de apresentações pré-definidas.

Na subseção 6.1.1 são apresentados os projetos colaborativos propostos e as respectivas soluções apresentadas pelos alunos.

6.1.1 Projetos colaborativos

A terceira unidade foi completamente abordada através da metodologia construcionista, em que o aluno é o principal responsável por sua aprendizagem. Desta forma, foram desenvolvidos seis projetos colaborativos pelos alunos, utilizando papel, lápis, o QPython e o WhatsApp para compartilhar o código, realizar reuniões e para debater sobre o projeto. Com isso, a primeira aula consistiu em os alunos criarem os problemas relacionados ao seu cotidiano. A primeira entrega foi a parte teórica do projeto e, durante as aulas seguintes, precisavam mostrar o andamento dos códigos e, por fim, na última aula do período, os alunos apresentaram suas ideias e soluções.

Os projetos entregues estão descritos no apêndice I.

6.2 Aplicação EAD

A Tabela 4 expõe de forma objetiva como a metodologia de ensino de programação foi aplicada durante o semestre de 2020.2 (de 30 março a 2 de agosto de 2021), o que totalizou 47 aulas responsáveis pela carga horária, mais 36 aulas extras. Vale ressaltar que a estratégia descrita abaixo foi montada de acordo com as aulas, visto que duas horas de aula eram dadas às terças-feiras, aulas síncronas, e duas horas dadas às quintas-feiras, assíncronas, no período da manhã das 07:00h até às 9:00, assim como aulas extras aos sábados seguindo o mesmo padrão das aulas das quintas, porém, das 9:00h até às 12:00h. Nas aulas das terças era prioridade passar o conteúdo e, nas quintas e sábados, eram trabalhados a sala invertida, em que os alunos ficavam como responsáveis por responder questões extras pelo Google Meet e o Replit. Sempre que os alunos precisassem tirar alguma dúvida fora dos dias e horários mencionados, eles tinham total liberdade para interagir no grupo do WhatsApp ou por mensagem privada ao professor.

Tabela 4 – Estratégia de ensino utilizada

AULA	CH	ASSUNTO	ESTRATÉGIA UTILIZADA
------	----	---------	----------------------

1 e 2	2h	Composição de programas: instruções de entrada e saída, operadores, expressões identificadores, constantes, variáveis e atribuição. Entrada e saída de dados.	Primeiro é explicado o assunto com exemplos práticos e com questões simples para facilitar o entendimento dos alunos.
3 e 4	2h	Exercícios sobre composição de programas	Sala invertida onde o aluno é que era o responsável por resolver o problema e tentar explicar ao professor. Essa é uma das vantagens em se utilizar o Google Meet e o Replit sincronizados. O tamanho pequeno dos grupos facilitou o acompanhamento.
EXTRA 1	4h	Resolução de atividades	Sala invertida com apoio pedagógico a fim de resolver mais questões e tirar dúvidas dos assuntos da semana.
5 e 6	2h	Estruturas de Decisão if	Aplicação do conteúdo da matéria, com exemplos práticos.
7 e 8	2h	Exercícios de Estruturas de Decisão if	Sala invertida.
EXTRA 2	4h	Resolução de atividades	Sala invertida com apoio pedagógico a fim de resolver mais questões, tirar dúvidas dos assuntos da semana.
10 e 11	2h	Estruturas de decisão compostas if	Foi passado em um período de no máximo um hora o conteúdo de estruturas compostas de decisão, realizado atividades simples para fixar o conhecimento e nas aulas seguinte foi utilizada para resolver uma lista de problemas.
12 e 13	2h	Exercícios de estruturas de decisão compostas if	Sala invertida.
EXTRA 3	4h	Resolução de atividades	Sala invertida com apoio pedagógico a fim de resolver mais questões, tirar dúvidas dos assuntos da semana.

14 e 15	1h	Manipulação de <i>strings</i>	Como o assunto seguinte é laços de repetição, optou-se por aprofundar os conhecimentos sobre manipulação de <i>string</i> na metade da primeira unidade, visto que, esse conteúdo possui semelhança com listas e em alguns casos faz-se necessário o uso de comandos de repetição.
16 e 17	2h	Exercícios de Manipulação de <i>strings</i>	Foi aplicado durante duas aulas uma lista de exercícios apenas com foco em <i>strings</i> . Ainda nessa aula, deve ser aplicado o primeiro questionário de acompanhamento o mesmo deve ser enviado aos alunos pelo grupo do WhatsApp.
EXTRA 4	4h	Resolução de atividades	Sala invertida com apoio pedagógico a fim de resolver mais questões, tirar dúvidas dos assuntos da semana.
18 e 19	1h	Estrutura de repetição <i>for</i>	Foi explicado o assunto, quando se deve aplicá-lo e passado exemplos e atividades simples em sala de aula.
20 e 21	2h	Exercícios de estrutura de repetição <i>for</i>	Sala invertida.
EXTRA 5	4h	Resolução de atividades	Sala invertida com apoio pedagógico a fim de resolver mais questões, tirar dúvidas dos assuntos da semana.
20 e 21	1h	Estrutura de repetição com <i>while</i>	O <i>while</i> foi explicado da mesma forma como o <i>for</i> . Depois exercícios para compreensão.
22 e 23	2h	Exercícios de estrutura de repetição com <i>while</i>	Sala invertida.
EXTRA 6	4h	Resolução de atividades	Sala invertida com apoio pedagógico a fim de resolver mais questões, tirar dúvidas dos assuntos da semana.
23 e 24	2h	Tuplas e listas	O assunto de tuplas foi escolhido para iniciar os conteúdos de estruturas de dados do Python por ser o mais simples.
25 e 26	2h		Nessa aula deve-se iniciar a resolução de exercícios e nas duas aulas seguintes os alunos devem responder as demais questões da lista.

EXTRA 7	4h	Exercícios sobre listas	Durante as quatro horas foram trabalhados exercícios sobre tuplas e listas com os subgrupos, cada um deles em intervalos de tempos separados. Nessa aula, foi passado o segundo questionário aos alunos para que eles expusessem suas dificuldades e o professor pudesse saná-las nas aulas de revisão.
27 e 28	2h	Revisão Primeira prova	Foram abordados exemplos práticos e passadas outras atividades para ser trabalhado como sala invertida.
29 e 30	2h	Primeira Prova	Foi aplicada uma prova com problemas contextuais e com questões diversas e que abordasse uma questão de cada assunto e uma com todos os assuntos com quatro modelos diferentes, ou seja, questões distintas em cada uma delas, além disso com níveis de dificuldades da mais simples a mais complexas. Cada grupo escolheu um horário para realizar a prova, onde eles foram avaliados de forma individual com os exercícios aplicados.
31 e 32	2h	Matrizes	Um dos assuntos mais complexos de introdução a programação é o de matrizes, esse ficou para a segunda unidade e foi abordado com exemplos que mostrassem quando e como se aplica o seu conteúdo.
32 e 33	2h	Exercícios sobre matrizes	Foram passadas atividades que foram desenvolvidas em grupo pelos alunos, na sala invertida.
34 e 35	1h	Listas compostas com dicionários	Foi ensinado primeiro o conteúdo de dicionários e após o uso de listas para armazenar dicionários
36 e 37	2h	Exercícios sobre dicionários	Foram dadas cinco questões diferentes para serem trabalhadas de forma colaborativa, no modelo sala invertida.
38 e 39	2h	Funções do Python, criação de funções e tratamento de erros	O último conteúdo ensinado foi o de funções, para que os alunos conseguissem automatizar todos os seus códigos e como tratar os erros de exceções do Python. Nessa aula, foi passado o terceiro questionário aos alunos para que eles expusessem suas dificuldades.

EXTRA 8	4h	Exercícios sobre matrizes, dicionários e funções	Foram trabalhadas atividades no modelo de sala invertida, com foco não mais em projetos separados e sim em projetos apenas colaborativos
40 e 41	2h	Projetos colaborativos	Foi passado aos alunos um modelo de projeto colaborativo que eles deveriam implementar o mesmo fez parte da nota de todos os alunos. Eles deveriam formular seus próprios problemas e que este estivesse relacionado ao seu cotidiano, ficando livres para serem criativos desde que, houvesse ao menos um conteúdo de cada unidade nas soluções propostas.
42 e 43	2h	Projetos colaborativos	Essa aula os alunos utilizaram para dar continuidade ao projeto, porém eles tinham que fazer a primeira entrega que era um relatório detalhado, informando quem ficou responsável pelo que, tanto na parte teórica como na prática.
EXTRA 9	4h	Dúvidas	Essa aula extra foi separada para os alunos tirarem dúvidas no código e ver se tudo estava alinhado com a regra de negócio definida por eles mesmos.
44 a 47	4h	Apresentação dos projetos	Foi solicitado aos alunos que apresentasse os projetos como se estivessem vendendo algo, ou seja, que falassem com propriedade suas ideias e suas respectivas soluções. Buscamos assim uma abordagem baseada no mercado de trabalho, com projetos colaborativos e apresentações voltadas para a regra de negócio.

Fonte: Acervo pessoal

A subseção 6.2.1 apresenta os projetos colaborativos desenvolvidos pelos grupos. Vale ressaltar que a transcrição abaixo está de acordo com o que foi entregue pelas equipes, sendo assim temos que um grupo tenha apresentado um relatório mais detalhado, tanto sobre sua implementação da ideia, como também explicação do código e ainda um relatório final. Isso porque a nota final do projeto consistia em 3 pontos pela parte teórica, mais 2 pontos pela apresentação e mais 5 pontos pelo código fonte. Porém, essas notas eram dependentes uma das outras, ou seja, não adiantava entregar tudo e não apresentar e explicar cada parte do código desenvolvido. Isso permitiu com que os alunos tivessem um foco muito grande na regra de negócio de cada projeto, potencializando assim habilidades de comunicação, trabalho em equipe e desenvolvimento pessoal.

6.2.1 Projetos colaborativos EAD

Durante todo o período foi abordado a metodologia construcionista, cujo o aluno é o principal responsável por sua aprendizagem. Na segunda unidade foram desenvolvidos oito projetos colaborativos pelos alunos, utilizando Google Meet, WhatsApp e Replit para compartilhar o código, realizar reuniões e para debater sobre o projeto. A primeira aula consistiu nos alunos criarem os problemas relacionados ao seu cotidiano. A primeira entrega foi a parte teórica do projeto e, durante as aulas seguintes precisam mostrar o andamento dos códigos e por fim na última aula do período os alunos apresentaram suas ideias e soluções.

Então, foram apresentados aos alunos dois modelos de projetos:

MODELO 1

Cada equipe deve propor um problema relacionado ao seu cotidiano, explicando o que deve ser feito e a solução de código implementado do que foi proposto. Assim, se uma equipe possui quatro integrantes, deve-se entregar quatro problemas, ou seja, cada um desses terá um autor e esse ficará responsável por apresentar sua questão e sua solução no dia da apresentação.

Desta forma, os problemas devem conter os seguinte requisitos básicos em sua solução:

- 1- Deve utilizar listas compostas para salvar várias informações e dicionários para salvar uma única informação
- 2- Deve ser utilizado as listas para salvar os dicionários
- 3- Deve utilizar blocos de condições e laços de repetição
- 4- Deve utilizar as seguinte funções próprias:
 - a. Função cadastrar
 - b. Função pesquisar
 - c. Função imprimir
 - d. Função remover
 - e. Função atualizar

MODELO 2

Será aceito que uma equipe crie projetos que se complementem, nesse caso específico os alunos poderão criar soluções de códigos que auxiliam a um projeto principal, por exemplo:

1. Se o projeto principal é construir uma tabela de somar, multiplicar, subtrair e dividir, os integrantes devem propor subproblemas que automatizem outros cálculos como: porcentagem, equações, raiz quadrada, etc.
2. Assim, deve ser apresentado um único projeto com alguns subprojetos (soluções extras).

3. Esse projeto único deve conter os requisitos básicos do modelo um, com o acréscimo de duas funções criadas por cada integrante que resolva algum problema extra do projeto. A questão só será válida se apresentada por quem criou.

Os projetos colaborativos entregues estão descritos no apêndice J.

7

Resultados e discussão

Este Capítulo está dividido em três seções. A Seção 7.1 aborda os resultados obtidos na aplicação realizada no ensino fundamental. A Seção 7.2 traz os resultados obtidos no ensino superior no modelo presencial. A Seção 7.3 traz os resultados obtidos no ensino superior a distância. A Seção 7.4 traz uma discussão sobre os resultados obtidos.

7.1 Resultados no ensino fundamental

O estudo de caso gerou uma cartilha de atividades para 50 aulas com 45 minutos. As subseções 7.1.1, 7.1.2, 7.1.3, 7.1.4 mostram os resultados obtidos com relação aos objetivos de pesquisa que deveriam ser alcançados. A subseção 7.1.1 mostra como as atividades ficaram divididas de acordo com o nível de alfabetização. Na subseção 7.1.2 apresentamos como os assuntos ensinados de programação foram absorvidos pelos alunos. Na subseção 7.1.3 detalhamos como ficou a integração das atividades com plano pedagógico. Por fim, a subseção 7.1.4 mostra quais foram as dificuldades encontradas pelos alunos e se a linguagem escolhida facilitou o ensino de programação.

7.1.1 Categorizar as atividades de acordo com a adequação ao nível de alfabetização e a idade do aluno

Cada atividade nessa dissertação foi analisada no dia a dia de aula, a fim de evidenciar quais assuntos eram equivalentes ao nível de alfabetização e a idade de cada aluno. A Tabela 5 demonstra uma descrição de todos os assuntos do ScratchJr e a Tabela 6 apresenta a descrição das atividades desplugadas.

Tabela 5 – Atividades dos ScratchJr de acordo com o nível de alfabetização

Atividade ScratchJr	Nível de Alfabetização	Idade
Criação, edição e pintura de personagem ou cenário	Não é necessário ler e escrever	De cinco anos em diante
Bloco de Eventos do tipo: inciar com a bandeira verde, inciar com Choque e inciar com Toque	Não é necessário ler e escrever	De cinco anos em diante
Bloco de Eventos inciar com mensagem e enviar mensagem	É necessário ler e escrever	De sete anos em diante
Todos os Blocos de Movimento	Não é necessário ler e escrever	De cinco anos em diante
Edição da quantidade de passos de um bloco de Movimento	É necessário saber os números naturais de zero até nove no mínimo	De seis anos em diante
Blocos de aparência do tipo: Aumentar, diminuir, repõe o tamanho, esconder e aparecer	Não é necessário ler e escrever	De cinco anos em diante
Blocos de aparência do tipo falar	É necessário ler e escrever	De sete anos em diante
Blocos de som	Não é necessário ler e escrever	De cinco anos em diante
Todos os Blocos de controle	É necessário ler e escrever, pois requer um entendimento um pouco mais avançado de interpretação e do uso dos números naturais	De sete anos em diante
Blocos de Finalização	Não é necessário ler e escrever	De cinco anos em diante

Fonte: Acervo pessoal

Tabela 6 – Atividades desplugadas

Assunto	Nível de alfabetização adequado	Idade Adequada
Centopeia Robô	Não é necessário ler e escrever	De cinco anos em diante
Conectando as Letras	Não é necessário ler e escrever	De cinco anos em diante

Fonte: Acervo pessoal

7.1.2 Caracterizar quais os assuntos referentes ao ensino de programação os alunos conseguiram assimilar e executar

O ensino de programação no ScratchJr é dividido em blocos, onde cada um é responsável por uma determinada ação dentro da linguagem. Assim, o projeto foi ensinado de acordo com seus respectivos objetivos, após cada aula era analisado de forma observacional como os alunos iriam se desenvolvendo e no final do projeto foi realizado testes relacionado a cada bloco para identificar o nível de conhecimento alcançado pela turma. As notas foram classificadas em cinco grupos: insuficiente é uma nota entre 0 e 2, ruim é uma nota entre 3 e 4, razoável é uma nota entre 5 e 6, bom é uma nota entre 7 e 8 e ótimo é equivalente a uma nota entre 9 e 10, os detalhes de como ficou o resultado final da turma está na Tabela 7.

Tabela 7 – Assuntos ensinados no ScratchJr e na atividade desplugada

Assunto	Objetivo	Nível alcançado pela turma
Bloco de Eventos	Criação, Inicialização e Envio de Mensagens	Excelente
Bloco de Movimento	Execução de ações dos personagens	Excelente
Blocos de aparência	Efeitos e Criação de diálogos entre os personagens da aplicação	Satisfatório
Blocos de som	Criação de sons já existentes ou para criação de sons próprios para os personagens	Excelente
Blocos de controle	Estrutura de decisão e estrutura de repetição são utilizados para criar determinadas ações em momentos específicos da aplicação	Bom
Blocos de Finalização	Transição de tela, repetição infinita do mesmo programa e finalização da aplicação	Razoável
Centopeia Robô	Estrutura de decisão, estrutura de repetição, movimentos de personagem, decomposição, abstração e Reconhecimento de padrão	Excelente
Conectando as Letras	Abstração, Decomposição, Reconhecimento de padrão, estrutura de repetição e estrutura de decisão	Razoável

Fonte: Acervo pessoal

7.1.3 Foi possível integrar as atividades relacionadas a ciência da computação com o que era ensinado pela professora em sala de aula?

Pode-se afirmar que, sim, é possível integrar o que é ensinado pela professora com o ensino de programação. A Tabela 8 lista as atividades, sua descrição, disciplina relacionada, eixo e conteúdo que é abordado em cada uma delas. Essa divisão, foi montada em conjunto com a pedagoga responsável pela turma onde aplicamos o estudo de caso. Cada atividade foi analisada de acordo como a forma como ela seria aplicada e como melhor ela deveria ser realizada, para que houvesse um aproveitamento estudantil satisfatório e que o projeto não comprometesse o desenvolvimento dos estudantes, uma vez que as atividades realizadas eram no primeiro horário das 7h até as 7:45 todos os dias.

Tabela 8 – Atividades e a correlação com matéria

Atividade	Disciplina	Eixo e Capacidade	Conteúdo
Conectando as letras	Português	Apropriação do sistema de escrita: compreender a natureza alfabética do sistema de escrita	Quantidade, variação e posição das letras nas determinadas palavras.
Conectando as letras	Português	Leitura: saber decodificar palavras e textos escritos	Relação entre grafemas (letra) e fonemas (som). Identificação de unidades fonológicas.
Conectando as letras	Português	Oralidade: participar das interações cotidianas em sala de aula, escutar com atenção e compreensão, responder as questões propostas pelo professor e expor opinião nos debates com os colegas e com o professor	Formas de participação adequadas para os espaços sociais públicos. Participação em sala de aula. Regras de convivência. Respeito mútuo.

Centopeia robô	Geografia	O ambiente que se vive (casa, rua, escola e bairro), suas representações para a vida em sociedade, como reconhecer ponto de referência, saber situar-se e tudo ao seu entorno: sistematizar a noção espacial, percebendo a proporção, distância e direção dos objetos, por meio da observação, representação e localização desses espaços, tendo como referência a sua moradia	Organização do espaço.
Centopeia robô	Ciências	Ser humano e saúde: reconhecer os tipos de movimentos que as partes do corpo podem realizar, conforme as articulações	Articulações e tipos de movimentos (movimentação do próprio corpo e do corpo dos colegas).
Descrição narrativa	História	Autoconhecimento: trabalhar o procedimento da construção da linha do tempo	A linha do tempo. Minha história ao longo do tempo.
Descrição narrativa	Geografia	O ambiente que se vive (casa, rua, escola e bairro) suas representações para a vida em sociedade, como reconhecer ponto de referência, saber situar-se e tudo ao seu entorno: Observar a organização dos espaços vividos, identificando as semelhanças e diferenças entre objetos a serem representados através do reconhecimento da função de cada um destes.	A paisagem local: minha casa, minha rua e a vizinhança. Organização do espaço. Mudanças da paisagem e seus elementos culturais e naturais ao longo do tempo.
Descrição narrativa	Matemática	Espaço e forma: identificar pontos de referência para situar e deslocar pessoas/objetos no espaço	Dimensionamento de espaços: relação de tamanho e forma.

ScratchJr: edição do quantitativo de movimentos de um personagem	Matemática	Números e operações: Interpretar e resolver situações-problemas, compreendendo diferentes significados das operações envolvendo números naturais	Noções de adição: juntar e acrescentar. Noções de subtração: tirar, comparar e completar.
ScratchJr: criação, edição e pintura de um personagem ou cenário	Artes	Artes visuais: experimentar, selecionar e utilizar diversos suportes, materiais e técnicas artísticas a fim de se expressar e se comunicar em artes visuais.	Seleção, manipulação e utilização de referências visuais e audiovisuais.
Labirinto: leve o gatinho até a casinha	Matemática	Espaço e forma: representar o espaço por meio de maquetes, croquis e outras representações gráficas.	Representação de deslocamento por meio de desenhos, mapas e plantas (para o reconhecimento do espaço e localização nele).
Torre de Hanói	Matemática	Espaço e forma: identificar, descrever e comparar padrões (por exemplo: blocos lógicos) usando uma grande variedade de atributos como tamanho, forma, espessura e cor.	Dimensionamento de espaços: relação de tamanho e forma. As formas geométricas presentes no cotidiano (escola, objetos, natureza, etc.). Construção e representação de formas geométricas.

Tetris	Matemática	Espaço e forma: 1- identificar, descrever e comparar padrões (por exemplo: blocos lógicos) usando uma grande variedade de atributos como tamanho, forma, espessura e cor; 2- identificar triângulos e quadriláteros (quadrado, retângulo, trapézio, paralelogramo, losango) observando as posições relativas entre seus lados.	1- Dimensionamento de espaços: relação de tamanho e forma. As formas geométricas presentes no cotidiano (escola, objetos, natureza, etc.). Construção e representação de formas geométricas. 2- Figuras Planas: quadrado, triângulo e retângulo. Triângulos e quadriláteros no Tangram. Semelhanças e diferenças entre as formas geométricas espaciais e planas.
Tangram	Matemática	Espaço e forma: 1- identificar, descrever e comparar padrões (por exemplo: blocos lógicos) usando uma grande variedade de atributos como tamanho, forma, espessura e cor; 2- identificar triângulos e quadriláteros (quadrado, retângulo, trapézio, paralelogramo, losango) observando as posições relativas entre seus lados.	1- Dimensionamento de espaços: relação de tamanho e forma. As formas geométricas presentes no cotidiano (escola, objetos, natureza, etc.). Construção e representação de formas geométricas. dois - Figuras Planas: quadrado, triângulo e retângulo. Triângulos e quadriláteros no Tangram. Semelhanças e diferenças entre as formas geométricas espaciais e planas.
Quebra-cabeça do alfabeto	Português	Leitura: compreender a natureza alfabética do sistema de escrita e saber ler reconhecendo globalmente as palavras.	Reconhecimento de palavras sem análise de fonemas e sílabas.
Jogo da velha	Matemática	Espaço e forma: Identificar, descrever e comparar padrões (por exemplo: blocos lógicos) usando uma grande variedade de atributos como tamanho, forma, espessura e cor.	Dimensionamento de espaços: relação de tamanho e forma. As formas geométricas presentes no cotidiano (escola, objetos, natureza, etc.). Construção e representação de formas geométricas.

Fonte: Acervo pessoal

7.1.4 Quais as dificuldades encontradas pelos alunos ao serem ensinados a programar? E a linguagem escolhida facilitou o ensino de programação?

Foi observado que os alunos interagiram de forma dinâmica com o ScratchJr, todos eles gostaram e se divertiram durante as aulas. Outro ponto muito importante a ser destacado é que durante boa parte das atividades em média de quatro a seis alunos usavam o ScratchJr apenas com o objetivo de editar seus personagens ou o cenário deixando assim muitas das vezes a programação de lado, já o restante fazia tudo, desde utilizar a criatividade na edição dos personagens, como a programação das suas histórias interativas, ou seja, deixavam além de bonito e criativo, também funcional.

O que foi dito no parágrafo anterior reforça os resultados encontrados na pesquisa de [Kazakoff \(2014b\)](#) que utilizou o ScratchJr com 38 estudantes. Ela observou que eles estavam mais preocupadas em utilizar o editor de pintura ou a edição dos personagens de suas histórias do que passar a maior parte do tempo programando, demonstrando que as crianças nessa idade gostam muito de usar a criatividade. Isso se dá pelo fato do ScratchJr possuir blocos de comandos com uma usabilidade bem intuitiva, facilitando no processo de ensino e aprendizagem quando relacionado a programação.

Sendo assim, devido à facilidade no uso da linguagem eles acabam descobrindo coisas novas e fixam suas vontades naquilo que gostam de fazer sem o uso da tecnologia, como desenhar e pintar, e isso demonstra a falta da habilidade da atenção focada. Aqueles que conseguem manter o foco durante as atividades acabam construindo histórias interativas completas.

Observou-se ainda que a linguagem escolhida facilitou sim o ensino de programação para crianças de diversos níveis de alfabetização e que a maior dificuldade encontrada durante as aulas foi o assunto de laços de repetição que é abordado quando as crianças precisam modificar quantas vezes um determinado bloco irá se repetir, isso se dá na opinião da pedagoga responsável pelo fato dos alunos não terem um conhecimento sólido de numerais. Por fim, todos estão aptos a utilizar dentro do ScratchJr as seguintes atividades:

1. Desenho no *Scratch*
2. Criação e edição de personagem
3. Criação e edição de cenário
4. Inserir mais de um personagem, colocar mais de um cenário e fazer transições de telas
5. Eventos

6. Movimentação
7. Sequenciamento
8. Ação e reação
9. Controle e Aparência de personagens
10. Interação com os personagens

7.1.5 Ameaças à validade

As ameaças quanto à validade da pesquisa realizada são:

1. Validade de construção: a escolha do ScratchJr se deu pelo professor já ter utilizado a plataforma Scratch, bem como por ter um ambiente no qual a única ferramenta que poderia ser utilizada de forma offline e em um *tablet* era o ScratchJr. Não foi possível testar outras ferramentas para atenuar esta ameaça, então procurou-se analisar se a mesma poderia ser utilizada durante as aulas e se ela facilitaria a aprendizagem dos alunos.
2. Validade de seleção: a metodologia de ensino construcionista e o uso de atividades desplugadas foi também uma escolha do professor responsável. Para que não houvesse algum viés de seleção foi analisado em outros trabalhos acadêmicos a eficácia de se mesclar o ensino de programação com atividades sem o uso de tecnologia, e constatou-se que era uma boa opção para ser aplicada no ensino fundamental.
3. Validade interna: os resultados obtidos e analisados não levaram em consideração apenas uma análise quantitativa, mas principalmente qualitativa. Desta forma, não é possível realizar afirmações concretas, já que nem todas as variáveis foram analisadas. Isso pode gerar um viés no resultado, que não pôde ser mitigado devido à pandemia e o projeto não poder ser replicado.
4. Validade externa: apesar de analisar os resultados obtidos durante os 50 encontros realizados com 19 alunos no período informado não é possível afirmar que a metodologia, a linguagem e as atividades criadas e escolhidas são mais ou menos eficientes do que outras pesquisas da literatura, mas sim possui um potencial bastante relevante para ser replicado devido a simplicidade dos exercícios, das avaliações. Não é possível realizar qualquer afirmação, além de dizer que a abordagem obteve êxito em sua aplicação.

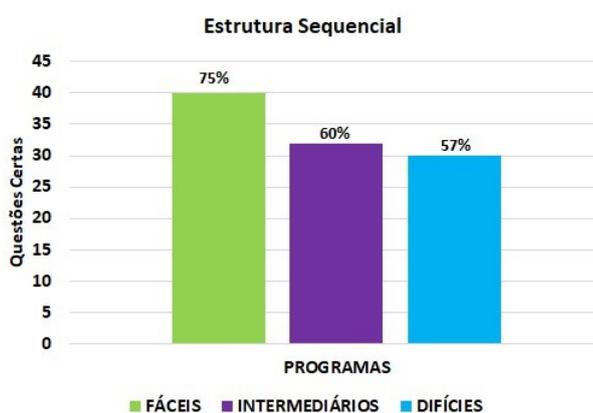
7.2 Resultados no ensino superior presencial

Usamos os programas passados em sala de aula que foram desenvolvidos pelos discentes e as respostas dos questionários aplicados aos alunos a cada assunto e ao final do período, para responder as questões de pesquisa nas subseções a seguir.

7.2.1 QP1. Os alunos conseguiram aprender o conteúdo de estrutura sequencial? Quais foram as dificuldades encontradas?

Sobre o conteúdo de estrutura sequencial, esperava-se que os alunos aprendessem a criar variáveis implícitas e explícitas, a resolver problemas lógicos sem o uso de estrutura de decisão e a formatar saída de dados. Assim, foram passados um total de 18 exercícios que variavam em seu nível de dificuldade de fácil a difícil, os resultados encontrados estão na Figura 39.

Figura 39 – Resultados obtidos para o conteúdo de estrutura sequencial



Fonte: Acervo pessoal

Os resultados obtidos foram que 75% dos alunos obtiveram êxito em resolver questões fáceis, 60% questões intermediárias e 57% questões difíceis. Diante dos resultados pode-se analisar que é possível que entre 50% a 70% dos alunos podem ter absorvido o conteúdo de estrutura sequencial. Ainda assim foi possível observar que os alunos que não conseguiram resolver questões mais complexas tiveram dificuldade com a interpretação dos enunciados das questões. Em linhas gerais, pode-se afirmar isso já que, ao observar todos os códigos, foi possível identificar que todos tinham o domínio da sintaxe da linguagem de programação até o momento. Porém, nem todos conseguiram executar um programa totalmente funcional.

Como forma de analisar o ponto de vista dos alunos, foram obtidas, por meio de um questionário online, 45 respostas sobre as dificuldades encontradas por eles durante o primeiro conteúdo. Dos alunos respondentes, 70% afirmaram ter tido dificuldade em interpretar enunciados mais elaborados, e os outros 30% afirmaram não ter.

Dessa forma, em resposta à questão de pesquisa QP1, é possível afirmar que os resultados obtidos demonstram que sim, mais de 60% dos alunos conseguiram aprender todo o conteúdo passado sobre estrutura sequencial.

7.2.2 QP2. Os alunos conseguiram aprender o conteúdo de estrutura de decisão? Quais foram as dificuldades encontradas?

Quanto ao conteúdo de estrutura de decisão, esperava-se que os alunos aprendessem a criar programas mais elaborados, utilizando os operadores lógicos AND, OR e NOT em conjunto com if, ou if e else ou ainda if, elif e else, como também blocos de condições aninhados em conjunto com o que foi apresentado na subseção 7.2.1.

No total, foram 28 problemas resolvidos pelos estudantes. A Figura 40 mostra que houve uma diferença pequena entre os índices de acertos dos três níveis de questões.

Figura 40 – Resultados obtidos para o conteúdo de estrutura de decisão



Fonte: Acervo pessoal

Conforme é possível observar na Figura 40, houve um aumento de aprendizagem dos alunos com o assunto de estrutura de decisão no que diz respeito ao conteúdo anterior, em que 89% concluíram com êxito estruturas lógicas fáceis e 75% concluíram questões difíceis, o que permitiu obter notas entre 5 e 10 na primeira avaliação, sendo a maioria destas bem acima da média ponderada desejada que é de 5. Deve-se frisar que apenas três alunos ficaram abaixo da média e acabaram truncando a disciplina. Estes não responderam nenhum dos exercícios passados, não entregaram suas atividades nos projetos em que estavam participando e não informaram suas dificuldades ao serem perguntados.

No final do conteúdo, foi feito outro questionário pedindo para o aluno identificar onde teve mais dificuldade e na opinião de 38 dos que responderam temos: 32% informaram ter mais dificuldade em problemas que envolviam comparações lógicas com AND, OR e NOT, ou ainda como aplicar if's aninhados, já os demais 68% informaram que não tiveram dificuldade. Essas informações foram confrontadas ao analisar as questões desenvolvidas por eles dentro e fora de sala de aula.

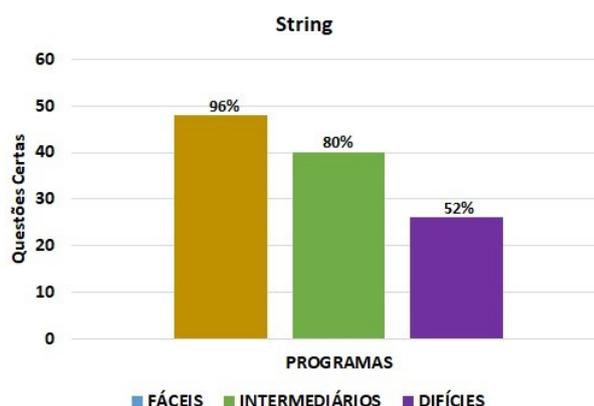
Em resposta à questão de pesquisa QP2, pode-se afirmar que os 81,66% dos alunos que conseguiram obter bons resultados nas atividades, demonstram que é possível que quase todos

os alunos absorveram o conteúdo. É importante ressaltar que, esse número é obtido através das questões realizadas em sala de aula e das notas de cada um. Entretanto, é importante ressaltar que além dos três alunos que trancaram a matéria, tivemos outros quatro alunos que conseguiram realizar equivalência de outra matéria e que deixaram de participar das aulas. Como o sistema não retira os alunos durante o período, somente depois, essas informações foram dadas pelo coordenador do curso.

7.2.3 QP3. Os alunos conseguiram aprender o conteúdo de *string*? Quais foram as dificuldades encontradas?

O conteúdo de *string* foi visto em paralelo com uma introdução aos conceitos de estruturas de repetição. Sobre o conteúdo de *string*, esperava-se que os estudantes aprendessem como manipulá-las e como alterar texto usando, por exemplo, operadores de pertencimento, como IN e NOT IN. A Figura 41 mostra os resultados obtidos com as resoluções dos exercícios.

Figura 41 – Resultados obtidos para o conteúdo de *string*



Fonte: Acervo pessoal

Os resultados obtidos são referentes a 14 exercícios sobre o conteúdo de *string*. Assim, temos 96% dos alunos que acertaram os problemas fáceis e 80% que acertaram os intermediários, demonstrando boa capacidade de interpretação e de colocar em prática o que foi abordado. Todavia, eles demonstram dificuldade em desenvolver questões que envolviam construções mais complexas como, por exemplo, o problema *leet speak generator*, que tinha como objetivo decodificar palavras na linguagem *leet speak*. Os alunos não conseguiram entregar programas totalmente funcionais, muitos pela metade e outros entre 80% a 90% da resolução correta. Isto é, trazendo assim o resultado de 52% dos exercícios corretos por completo.

A análise que foi feita é de que os alunos, ao mudarem de questões que envolviam mais cálculos matemáticos, como foi bastante abordado na primeira unidade e na segunda unidade, passaram a trabalhar com manipulação de *string*, de modo que isto pode ter influenciado nos resultados desse conteúdo. Com isso, outro questionário foi passado aos alunos para que eles

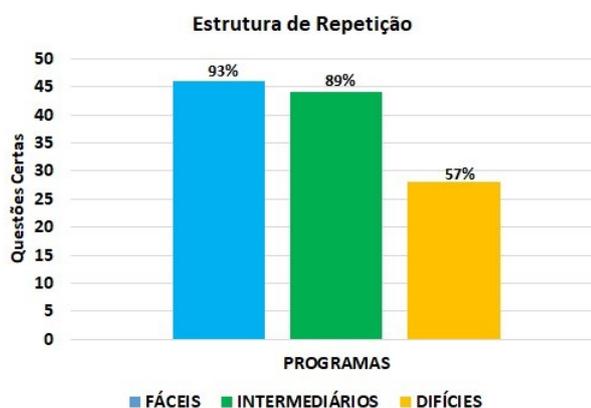
informassem o que tiveram mais dificuldades de entender, aprender ou até mesmo executar. Em suma, dos 42 alunos que responderam, 38% informaram não saber quando deveriam ou não usar operadores ou funções pré-definidas da *string*, outros 52% informaram ter tido dificuldade em questões mais complexas, já para 10% informaram não ter tido dificuldade.

Em resposta à questão de pesquisa QP3, é possível afirmar que os alunos aprenderam a manipular *strings* de forma parcial, visto que conseguiram implementar por completo apenas programas simples e intermediários.

7.2.4 QP4. Os alunos conseguiram aprender o assunto de estrutura de Repetição? Quais foram as dificuldades encontradas?

Para finalizar o conteúdo da segunda unidade foi dada continuidade com a estrutura de repetição, ministrada a partir da metade do período letivo. Nesse momento, esperava-se que os alunos fossem capazes de resolver diversos tipos de problemas, como cálculos matemáticos, manipulação de *strings* e problemas lógicos mais complexos, utilizando uma das duas estruturas de repetições, *while* e/ou *for*. A Figura 42 mostra os resultados obtidos nos 51 programas passados em sala de aulas para os alunos responderem.

Figura 42 – Resultados obtidos para o conteúdo de repetição



Fonte: Acervo pessoal

Os índices de acerto de questões fáceis e intermediárias foram próximos entre si, mostrando uma boa aplicação das estruturas de decisões quando não envolviam questões complexas com *for* e/ou *while* aninhados, ou seja, isso fica perceptível ao observar que o rendimento dos alunos caiu quase 35% com relação às outras questões.

Quando os alunos foram perguntados, por meio do formulário online, 72% informaram ter tido dificuldade com o laço de repetição *while* e com a aplicação de um *loop* dentro de outro. Assim, o que ajudou a sanar essas dificuldades foi a grande quantidade de questões aplicadas em sala de aula e o auxílio do professor.

Ao analisar os programas realizados em cada aula e as entregas das atividades dos projetos, foi possível perceber que, além de erros simples na implementação de um laço de repetição com `while`, os alunos tiveram dificuldades em abstrair problemas complexos. Como forma de mitigar tal dificuldade, foi realizado uma gincana valendo dois pontos, no qual era passado um problema e cada grupo tinha de 10 a 20 minutos para implementar a solução. Essa dinâmica competitiva potencializou, em alguns alunos com mais dificuldades, a vontade de praticar mais ainda, bem como a de buscar mais acompanhamento dentro e fora da universidade.

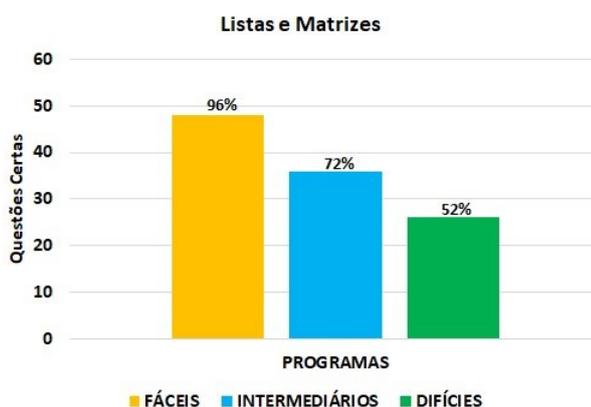
Os resultados das notas correspondem a 70% dos alunos que mantiveram suas notas entre 7 e 10, outros 25% que melhoraram suas notas de 5 para 6 e 7, já os outros 5% que não conseguiram responder as questões das provas que valiam mais pontos, ocasionando assim a ficarem abaixo da média.

Quanto à pesquisa QP4, pode-se afirmar que 95% dos alunos conseguiram assimilar e colocar em prática todo o conteúdo de laços de repetição.

7.2.5 QP5. Os alunos conseguiram aprender o conteúdo de lista? Quais foram as dificuldades encontradas?

Sobre o último conteúdo abordado, de listas, matrizes e tuplas esperava-se que os alunos soubessem criar, inserir, excluir e alterar elementos de uma lista e de uma matriz aplicando em casos práticos reais. Esperava-se também que tivessem compreendido e soubessem utilizar todos os conteúdos anteriores. A Figura 43 mostra os resultados dos 24 programas sobre os conteúdos citados.

Figura 43 – Resultados obtidos nas atividades de Lista



Fonte: Acervo pessoal

Foi atingido um ótimo índice de acertos nas questões fáceis que envolviam apenas as aplicações de listas e tuplas, o que significa que os alunos aprenderam a ideia principal desse conteúdo. Tiveram, porém, dificuldades nas questões difíceis na aplicação de matrizes, que abrangem um nível de complexidade bastante alta.

Um último questionário foi aplicado com os alunos para saber o que eles tiveram mais dificuldades e 30% responderam que foi na aplicação dos métodos `reverse()`, `copy()`, `replace()` e `extend()`. Outros 21% informaram ter tido dificuldade em entender a parte do conteúdo classificada como fácil, que inclui inserir, remover e atualizar um determinado elemento em uma posição. Já para 28% deles, foi difícil entender e aplicar o conteúdo de matrizes e, por fim, 21% dos estudantes afirmam não ter tido nenhuma dificuldade.

Os resultados obtidos nas provas da terceira e última unidade tiveram uma queda de rendimentos das notas. Os resultados finais obtidos, mostraram que, os alunos que tinham tirado entre sete e nove reduziram suas notas em pelo menos dois pontos, ficando mais de 25% com notas entre cinco e 6, outros 30% que tinham notas mais baixas conseguiram melhorar ao menos um ponto com relação à unidade anterior. Já para 35% dos alunos, estes conseguiram manter suas médias e tivemos 10% dos alunos que ficaram abaixo da média de tal forma que dois alunos acabaram reprovando e outros três ficaram com média igual a 5.

Alunos que estavam acima da média antes da segunda unidade reduziram a nota na terceira unidade, provavelmente pelo o fato de já estarem aprovados. Já aqueles que estavam abaixo da média aumentaram a nota. Além disso, é preciso destacar que os estudantes tiveram bastante dificuldade na aplicação de matrizes.

Em resposta à questão de pesquisa QP5, pode-se afirmar que 75% dos alunos conseguiram aprender tudo o que foi ensinado na matéria e que 20% aprenderam de forma parcial, conseguindo aplicar, mas sem resultados por completo em questões mais complexas. Também, temos 5% que não se desenvolveram da forma esperada.

7.2.6 QP6. Os assuntos que os alunos mais tiveram dificuldade?

Quanto à questão de pesquisa QP6, os assuntos que os alunos mais tiveram dificuldade foram: laço de repetição com `while`, manipulação de *string* e aplicação de matrizes. Acredita-se que, por serem conteúdos mais abstratos de serem absorvidos, tenha-se gerado um pouco mais de complexidade.

Sobre o uso de dispositivos móveis, os discentes informaram que 80% deles utilizaram um *smartphone* ou *tablet* e 20% utilizaram *notebooks*. De acordo com eles, ainda, o uso dos *smartphones* possibilitou que pudessem praticar questões de programação com maior frequência, quer fosse no ônibus, no pátio da universidade ou durante as aulas. Além disso, a facilidade de poder compartilhar o código com o professor pelo WhatsApp e tirar dúvidas em qualquer horário, possibilitou que tivessem um melhor desenvolvimento.

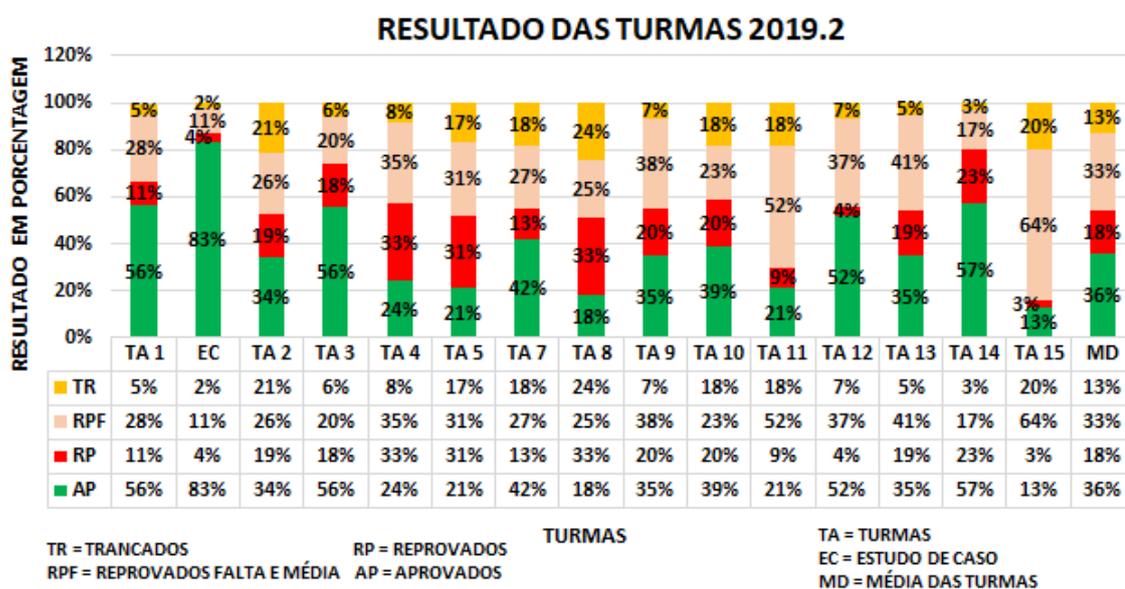
Diante disso, a análise que foi feita durante o acompanhamento no dia a dia de aula é de que muitos alunos tinham facilidade na digitação pelo celular, de maneira que eles conseguiam ser tão rápidos quanto os alunos que utilizavam um *notebook*. No entanto, alguns alunos tiveram bastante dificuldade em utilizar os *smartphones*, o que só foi possível resolver quando consegui-

mos uma sala com mesas de estudo e 25 computadores. Assim, foi definido em consenso que aqueles que não conseguiram se desenvolver com os dispositivos móveis poderiam utilizar os computadores disponíveis.

Do ponto de vista construcionista, a capacidade do professor em saber mitigar as dificuldades por meio de projetos, explicações e gincanas que possibilitassem o aluno a entender o conteúdo, como também conseguir uma sala, mesmo que com poucos computadores para que ninguém fosse prejudicado, possa ter influenciado diretamente nos resultados alcançados.

Esses resultados obtidos com a aplicação da metodologia construcionista são animadores, mesmo precisando de ajustes em turmas futuras. Com efeito, encerrou-se a matéria com uma taxa de aprovação de 83%, 15% foram reprovados por falta ou média e apenas 2% de evasão. Essas informações são expressas na segunda coluna da esquerda para direita na Figura 44. Ademais, também é possível verificar os dados das outras turmas no mesmo período.

Figura 44 – Análise do resultado das turmas no mesmo período letivo



Fonte: Acervo pessoal

Ao analisarmos os dados obtidos com os de outras turmas no mesmo período, temos que, o estudo de caso teve 47% aprovados, 11% de reprovados por falta e média, 4% de alunos e 2% evasão acadêmica e se esses dados forem comparados com os de outras turmas os números são bem promissores. Porém, é preciso destacar que existem professores com metodologias e abordagens de ensino diferentes, como também podem lecionar com linguagens de programação distintas, isso nos mostra que a educação não é exata e que depende de diversas variáveis e situações que influenciam de forma positiva ou negativa nos resultados finais.

É importante ressaltar que cada professor lecionava em uma turma. Nesse período apenas um dos professores ficou com duas, a 13 e a 15, nas quais utilizou a linguagem Python, porém, em um laboratório. Outro ponto é de que devido à alta demanda de turmas de introdução à

programação, os laboratórios são divididos por ordem de prioridade de curso. Os de ciência da computação e sistemas de informação ficam com laboratórios, turmas que tenham apenas um curso de exatas revezam ficando um dia em sala normal e um dia em laboratório e, por fim, as turmas mistas que possuem mais de um curso que ficam em salas normais, podendo conseguir um laboratório no meio do período ou não.

Nessa análise final não é feito um comparativo com os resultados do período anterior, por não haver informações concretas sobre qual metodologia foi utilizada, ou qual linguagem. Assim, concluímos que a metodologia construcionista necessita por parte do professor de um empenho e tempo dobrado no que diz respeito ao ensino de programação que, na maioria das turmas, conta com mais de 50 alunos. Os resultados podem ser positivos se houver engajamento de ambas as partes, já que o construcionismo possibilita essa interação dinâmica entre docente e discente.

7.2.7 *Feedbacks dos alunos*

Ao final do curso, os alunos responderam um questionário final online através do Google Forms, onde relataram suas dificuldades encontradas e deram sua opinião sobre a metodologia. Para preservar a privacidade, não era necessário incluir nenhuma informação pessoal. Alguns relatos são transcritos abaixo:

- **Aluno 1:** *“Muito boa a estratégia ao deixar com que os alunos fossem os principais responsáveis por sua aprendizagem.”*
- **Aluno 2:** *“Instigante. O professor oferecia todo suporte necessário ao aprendizado dos alunos. Estando disponível para esclarecer dúvidas e ajudar quem estava interessado, incluindo em horário extraclasse. Sempre pronto pra ajudar.”*
- **Aluno 3:** *“Muito diversificada, na questão de ter mais de uma forma de entender os assuntos, e bastante dinâmica e prática com exercícios que aplicamos diariamente”*
- **Aluno 4:** *“Um método novo, criativo e de fácil entendimento que resultou em aulas descontraídas e leves.”*
- **Aluno 5:** *“Definitivamente uma metodologia maravilhosa, extremamente pertinente e didática, conseguindo através de jogos e listas facilitar o aprendizado e fixar o assunto com mais facilidade.”*
- **Aluno 6:** *“A opção de utilizar o Python pelo celular foi importante pela facilidade. Pelo celular, com um clique era possível enviar o código para qualquer pessoa e a interação é maior e mais rápida. A metodologia adotada foi muito boa, desde o primeiro dia de aula, programando o básico na folha de papel.”*

Na Figura 45 é possível ver os alunos programando em equipes e usando os smartphones. Além disso, ainda temos um dos discentes utilizando um *notebook*.

Figura 45 – Alunos programando em equipe no *QPython*



Fonte: Acervo pessoal

Acreditamos que a metodologia de ensino construcionista foi bem-sucedida no que diz respeito à sua aplicação. Em outras palavras, buscou-se de forma incessante usar aulas interativas, gincanas com questões de programação, trabalhos em grupos e desafios que eram lançados durante as aulas sobre algum conteúdo, que os alunos tivessem mais dificuldade, sempre buscando estimular o aprendizado de forma autônoma.

7.2.8 Ameaças à validade

Toda e qualquer aplicação de uma metodologia é passível de erros e ajustes. Nessa seção abordamos as ameaças quanto à validade da pesquisa realizada que são:

1. Validade de construção: a escolha da linguagem Python para o ensino de programação foi ocorreu pela familiaridade do professor com ela, gerando um viés de construção. Para atenuar esta ameaça, procurou-se analisar o contexto das aulas para saber qual linguagem poderia ser utilizada em qualquer dispositivo móvel ou *desktop*, e constatou-se que o Python era a melhor opção.
2. Validade de seleção: a metodologia de ensino construcionista e o uso do *smartphone* em sala de aula foi uma escolha do professor devido ao cenário em que a turma se encontrava, que era a falta de recursos tecnológicos. Para constatar se as escolhas eram as melhores opções, o ideal seria ter realizado uma abordagem prévia antes da turma, entretanto, esta foi testada com os próprios alunos nas duas primeiras semanas.
3. Validade interna: os resultados obtidos e analisados não levaram em consideração outras metodologias, outras ferramentas ou até mesmo outras abordagens de ensino. Desta forma, não é possível realizar afirmações concretas, já que nem todas as variáveis foram analisadas. Isso pode gerar um viés no resultado, que não pôde ser mitigado devido à pandemia.

4. Validade externa: apesar de analisar os resultados das 16 turmas no período informado não é possível analisar a quantidade de exercícios, avaliação e ensino, visto que são professores diferentes com autonomia para aplicar da forma como acharem melhor. Mesmo que seis turmas tenham utilizado a mesma linguagem de programação, não foi possível ter um padrão de ensino. No entanto, esse trabalho apresentou evidências sobre sua aplicação da maneira mais detalhada possível para que a metodologia possa ser replicada por outros professores.
5. Delineamento experimental: por não se tratar de uma pesquisa científica, mas, sim, de um relato de experiência, não foi utilizado nenhum um grupo de controle, nem homogeneidade entre grupos, nem distribuição aleatória. Essa dissertação não tem um viés experimental e sim um viés qualitativo, que busca passar informações de um relato de estudo de caso, mesmo que traga os resultados por meio de gráficos estatísticos, que fazem relação com o próprio trabalho e não com grupos de pesquisas.

7.3 Resultados no ensino superior EAD

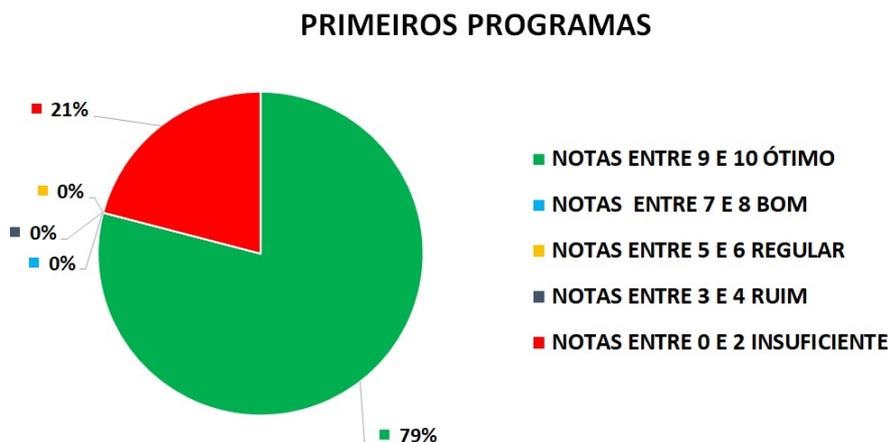
Usamos os programas feitos nas aulas síncronas e assíncronas, desenvolvidos pelos discentes e as respostas do questionário realizado com os alunos ao fim de cada assunto e no final do período, para responder as questões de pesquisa nas subseções a seguir.

7.3.1 QP1. Os alunos conseguiram aprender o conteúdo de estrutura sequencial? Quais foram as dificuldades encontradas?

Sobre o conteúdo de estrutura sequencial, esperava-se que os alunos aprendessem a criar variáveis implícitas e explícitas, a resolver problemas lógicos sem o uso de estrutura de decisão e a formatar saída de dados. Assim, foram passados um total de 35 exercícios que variavam em seu nível de dificuldade de fácil a difícil, dos quais 19 foram aplicados para serem respondidos pela plataforma The Huxley e os outros 8 foram trabalhados nos subgrupos com os alunos de forma colaborativa, os resultados obtidos estão nas Figuras 46 e 47.

Dessa forma, em resposta à questão de pesquisa QP1, é possível afirmar que sim, os alunos conseguiram aprender todo o conteúdo passado sobre estrutura sequencial.

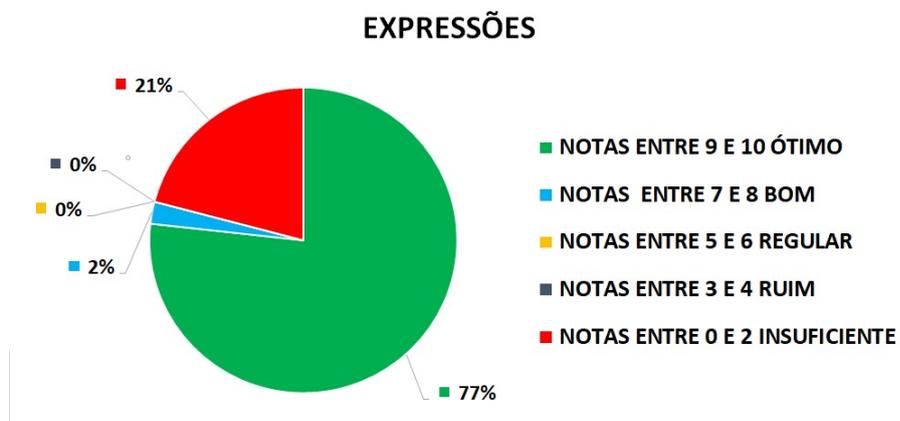
Figura 46 – Resultados obtidos para os primeiros programas



Fonte: Acervo pessoal

Os resultados obtidos com os primeiros programas fazem referência a 4 exercícios respondidos no The Huxley, nesses obtivemos que 79% dos alunos tiveram notas entre 9 e 10.

Figura 47 – Resultados obtidos para expressões



Fonte: Acervo pessoal

Os resultados obtidos com programas de expressões fazem referência a quatro exercícios respondidos no The Huxley, nesses obtivemos que 79% dos alunos tiveram notas entre 9 e 10.

Os 21% de alunos com notas insuficientes nos primeiros programas e nos de expressões se explica pelo fato de que nove alunos não se pronunciaram, nem por e-mail, nem por WhatsApp, e inclusive durante as aulas esses alunos não compareceram, ficando assim com notas 0 no que diz respeito as atividades práticas.

Em linhas gerais, pode-se afirmar que, ao observar todos os códigos desenvolvidos nos projetos colaborativos os alunos tinham o domínio da sintaxe da linguagem de programação até o momento. Porém, nem todos conseguiam executar uma estrutura lógica completa no início e que isso foi mitigado durante as aulas totalizando ao final do conteúdo 100% de alunos com domínio dos assuntos.

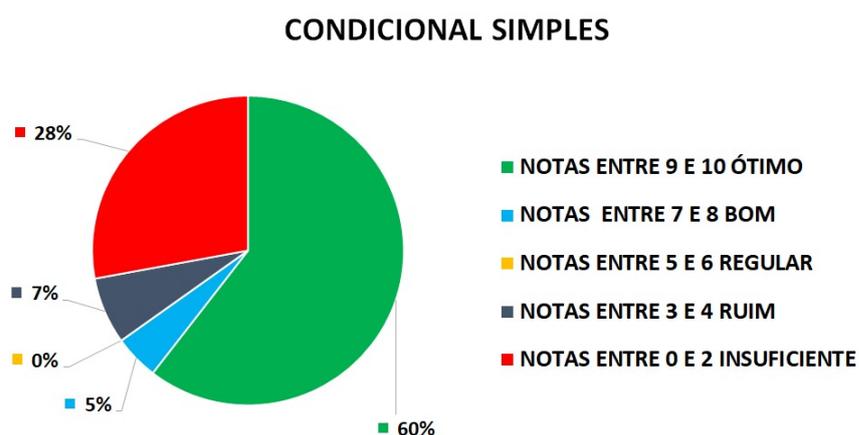
Como forma de analisar o ponto de vista dos alunos, foram obtidas, por meio de um questionário online, 27 respostas sobre as dificuldades encontradas por eles durante o primeiro conteúdo, em que 55,6% dos alunos afirmaram ter tido dificuldade em variáveis do tipo booleana e 59,3% com concatenação de *strings*.

7.3.2 QP2. Os alunos conseguiram aprender o conteúdo de estrutura de Decisão? Quais foram as dificuldades encontradas?

Quanto ao conteúdo de estrutura de decisão, esperava-se que os alunos aprendessem a criar programas mais elaborados, utilizando os operadores lógicos AND, OR e NOT em conjunto com if, ou if e else ou ainda if, elif e else, como também blocos de condições alinhados em conjunto com o que foi apresentado na subseção 7.3.1.

No total, foram 18 problemas resolvidos pelos estudantes por meio do The Huxley e 24 problemas resolvidos de forma colaborativa, esses últimos foram divididos em três questões por equipes, os resultados obtidos podem ser vistos nas Figuras 48 e 49.

Figura 48 – Resultados obtidos para condicionais simples

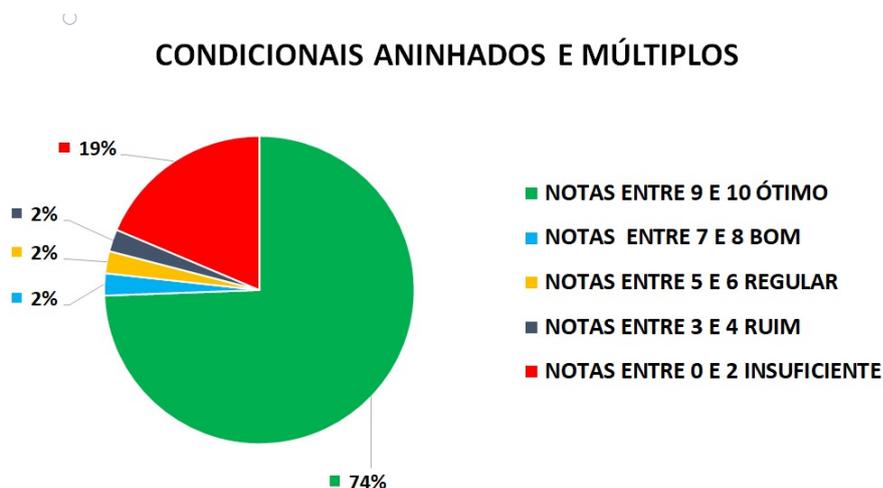


Fonte: Acervo pessoal

Os resultados obtidos com o assuntos de condicionais simples fazem referência a 10 exercícios respondidos no The Huxley, nesses obtivemos que 60% dos alunos tiveram notas entre 9 e 10, 5% notas boas entre 7 e 8, 7% notas ruins entre 3 e 4 e tivemos um aumento nas notas insuficientes de 21% para 28% o que significa que três alunos da turma tiveram mais dificuldades

do que os demais, o que foi mitigado com o decorrer dos exercícios como pode ser visto nos resultados disponíveis na Figura 49.

Figura 49 – Resultados obtidos para condicionais compostas



Fonte: Acervo pessoal

Conforme é possível observar na Figura 49, houve uma redução de alunos com notas insuficientes isso se dá pelo fato de que um dos nove alunos que não estavam participando das aulas começou a participar fazendo com que essa taxa negativa reduzisse. Além disso, obtivemos um aumento de 14% entre um assunto com condicionais simples e condicionais aninhados e múltiplos, mostrando assim que houve um aumento na aprendizagem dos discentes.

No questionário feito ao final do conteúdo, foi perguntado ainda para os alunos onde tiveram mais dificuldades com relação a blocos de condições, simples ou compostos e na opinião de 27 dos que responderam temos: 29,6% informaram ter mais dificuldade em problemas que envolviam o uso de `if`, `elif` e `else` juntos, 37% tiveram dificuldades com comparações textuais e 37% em usar a sintaxe lógica `NOT`.

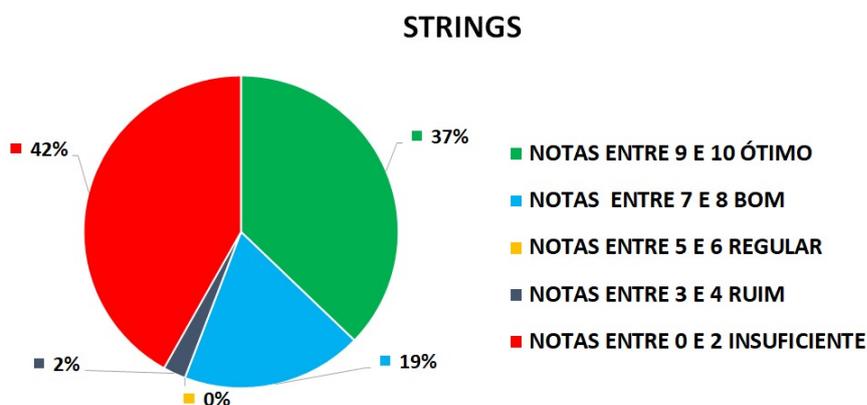
Em resposta à questão de pesquisa QP2, é possível afirmar que 70% dos alunos conseguiram absorver todo o conteúdo ensinado e 30% um domínio parcial, significa dizer que os alunos estavam aptos a passarem adiante sem nenhuma dificuldade aparente. Nesse aspecto, esse número é obtido através das questões realizadas em sala de aula e das notas de cada um.

7.3.3 QP3. Os alunos conseguiram aprender o conteúdo de *string*? Quais foram as dificuldades encontradas?

O conteúdo de *string* foi visto em paralelo com uma introdução aos conceitos de estruturas de repetição `for`. Sobre o conteúdo de *string*, esperava-se que os estudantes aprendessem como manipulá-las. Foi passado aos alunos dez questões para serem respondidas no The Huxley e

oito questões trabalhadas em grupos na Figura 50 é possível ver os resultados obtidos com as resoluções desses exercícios.

Figura 50 – Resultados obtidos para *strings*



Fonte: Acervo pessoal

O alto índice de notas insuficientes se dá principalmente pelo fato da manipulação de *strings* não ser algo simples, mas também pelo fato de que o assunto devido ao cronograma não ter tido uma atenção como os demais, ou seja, não foi possível trabalhar a mesma quantidade de tempo.

Um ponto a destacar é de que o uso de varias plataformas para os alunos utilizarem, para responder questionários, enviar atividades e responder questões praticas pode ter impactado na forma que os alunos estavam aprendendo, um exemplo é de que os estudantes conseguiam realizar códigos simples ao programarem em ma IDE e quando iam executar seus códigos na plataforma The Huxley utilizada para automatizar as correções eles tinham bastante dificuldades em corrigir saídas idênticas ao esperado da ferramenta. Visto isso, optou-se por permitir aos alunos que possuíam bastante dificuldade no que foi mencionado a enviarem seus códigos diretamente pelo Google Classroom, para que assim fosse avaliado pelo professor.

Os resultados obtidos são que 37% dos alunos tiveram notas ótimas, 19% notas boas e 42% notas insuficientes. Ao analisar os alunos nos projetos colaborativos observou-se que, eles demonstram dificuldade em desenvolver questões que envolviam construções mais complexas como, por exemplo, o problema *leet speak generator*, que tinha como objetivo decodificar palavras na linguagem *leet speak*, em que não conseguiram entregar programas totalmente funcionais, muitos pela metade e outros entre 70% a 85% da resolução correta. Isto é, trazendo assim o resultado de 60% dos exercícios corretos por completo. Isso, também explica o fato dos alunos terem tido notas ruins visto que o The Huxley só aceita o resultado correto.

Com isso, outro questionário foi passado aos alunos para que eles informassem o que

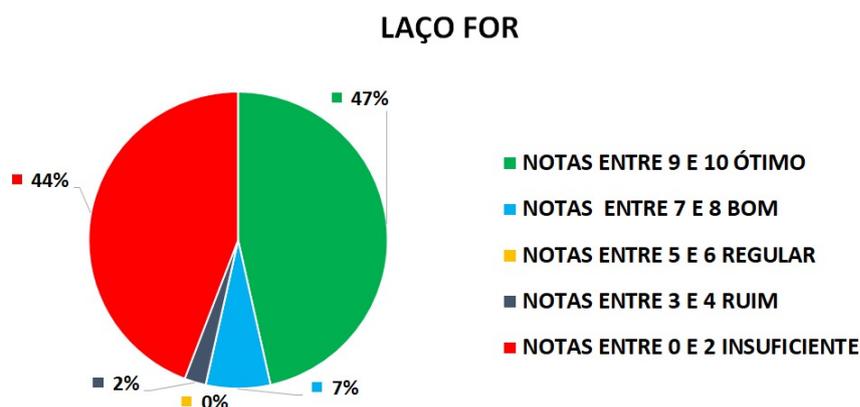
tiveram mais dificuldades de entender, aprender ou até mesmo executar. Em suma, dos 19 alunos que responderam, 57,9% informaram não saber quando deveriam ou não usar operadores ou funções pré-definidas da *string*, e informaram ainda ter bastante de dificuldade em interpretar questões com esse assunto.

Em resposta à questão de pesquisa QP3, é possível afirmar que os alunos aprenderam a manipular *strings* de forma parcial, visto que conseguiram implementar por completo apenas programas simples e intermediários.

7.3.4 QP4. Os alunos conseguiram aprender o assunto de estrutura de repetição? Quais foram as dificuldades encontradas?

Para dar continuidade ao conteúdo ainda da primeira unidade foi passado primeiro o assunto de estrutura de repetição com *for*. Foram aplicados oito questões pelo The Huxley e 16 projetos para serem trabalhados em grupo. Em seguida foi ministrado o assunto de estrutura de repetição com *while*, no qual foram aplicadas nove atividades no The Huxley e mais 16 exercícios em equipes. Nesse momento, esperava-se que os alunos fossem capazes de resolver diversos tipos de problemas, como cálculos matemáticos, manipulação de *strings* e problemas lógicos mais complexos, utilizando uma das duas estruturas de repetições, *while* e/ou *for*. As Figuras 51 e 52 mostram os resultados obtidos nos 19 programas passados como atividades avaliativas para os alunos responderem.

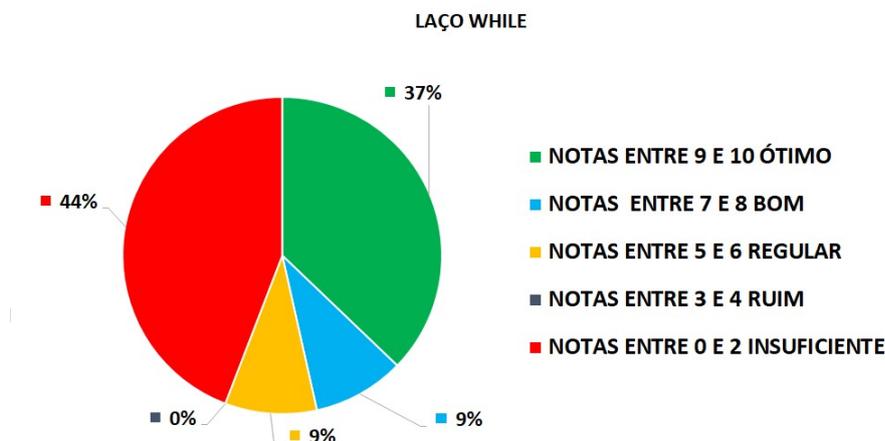
Figura 51 – Resultados obtidos para estrutura de repetição com *for*



Fonte: Acervo pessoal

Os resultados com laços *for* foram que 47% obtiveram notas ótimas, 7% notas boas, 2% notas ruins e 44% notas insuficientes.

Figura 52 – Resultados obtidos para estrutura de repetição com while



Fonte: Acervo pessoal

Já os resultados com laços `while` foram que 37% obtiveram notas ótimas, 9% notas boas, 9% notas regulares e 44% notas insuficientes.

Foi observado que os alunos continuaram a ter dificuldades em usarem múltiplas plataformas e que ainda possuíam problemas no uso do The Huxley. Vale ressaltar que manteve-se a modalidade dos alunos poderem enviar as questões pelo Google Classroom o que nem todos faziam e mesmo assim tentavam responder por lá.

Quando os alunos foram perguntados, por meio do formulário online, 31% informaram ter tido dificuldade com o laço de repetição `while`, 47,4% com a aplicação de um *loop* dentro de outro. Assim, o que ajudou a sanar essas dificuldades foi a grande quantidade de questões aplicadas em sala de aula e o auxílio do professor.

Ao analisar os programas desenvolvidos nas aulas com os grupos, foi possível perceber que, além de erros simples na implementação de um laço de repetição com `while`, os alunos tiveram dificuldades em abstrair problemas complexos. Como forma de mitigar tal dificuldade, foi realizado uma gincana em uma das aulas síncronas entre os grupos valendo um ponto extra, o grupo que respondesse mais questões certas ganhava o ponto, simulando assim um campeonato de programação como as Olimpíadas Brasileiras de Informática, no qual era passado um problema e cada grupo tinha de 10 a 20 minutos para implementar a solução.

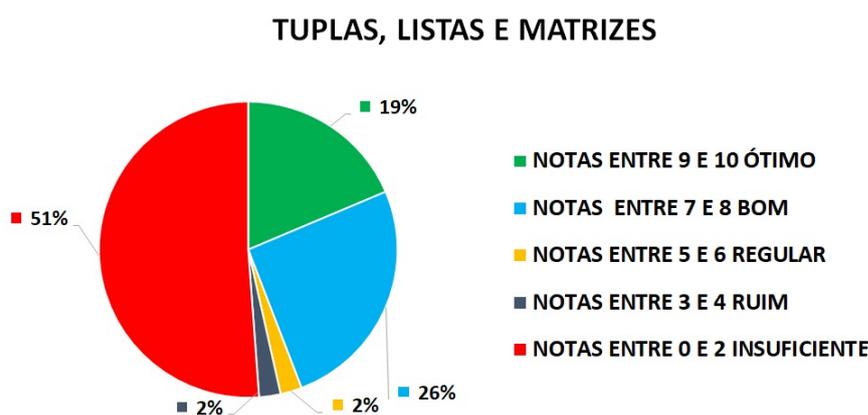
Quanto à pesquisa QP4, pode-se afirmar que 56% dos alunos conseguiram assimilar e colocar em prática todo o conteúdo de laços de repetição e que os outros 44% tiveram bastante dificuldades, desses apenas 22% eram alunos ativos durante as aulas, ou seja, que compareciam e realizavam atividades, por ser constatado tal dificuldade fez-se necessário diminuir o máximo de plataformas diferentes que os alunos precisavam acompanhar para está alinhado a matéria. Assim, optou-se por permanecer somente com o Google Classroom para os alunos receberem materiais e

assistirem as aulas, bem como para o envio de atividades praticas e avaliativas, permitindo assim que os alunos tivessem seus focos em único lugar e assim entregassem soluções otimizadas e até mesmo criativas.

7.3.5 QP5. Os alunos conseguiram aprender o conteúdo de tuplas, listas e matrizes? Quais foram as dificuldades encontradas?

Sobre os penúltimos conteúdos abordados, de listas, matrizes e tuplas esperava-se que os alunos soubessem criar, inserir, excluir e alterar elementos de uma lista e de uma matriz aplicando em casos práticos reais. Esperava-se também que tivessem compreendido e soubessem utilizar todos os conteúdos anteriores. A Figura 53 mostra os resultados dos 25 programas sobre os conteúdos citados, dos quais, cinco sobre tuplas, dez sobre listas e dez sobre matrizes.

Figura 53 – Resultados obtidos para tuplas, listas e matrizes



Fonte: Acervo pessoal

Foi atingido um ótimo índice de acertos nas questões de tuplas e listas, o que significa que os alunos aprenderam a ideia principal desse conteúdo. Tiveram, porém, dificuldades nas questões difíceis na aplicação de matrizes, que abrangem um nível de complexidade bastante alta. Assim, obtivemos um percentual 51% de notas ruins elevadas mas que diz respeito a grande parte das atividades de matrizes.

A mudança de uma abordagem parcial construcionista para uma abordagem total, potencializou os alunos a buscarem solucionar o problema ao invés de apenas acertar, o que permitiu um melhor desenvolvimento de assuntos que antes muitos estavam com dificuldades como é exemplo dos laços de repetição e mesmo com um deficit na aplicação de matrizes, acreditamos que obteve-se êxito na absorção dos conteúdos por parte dos alunos.

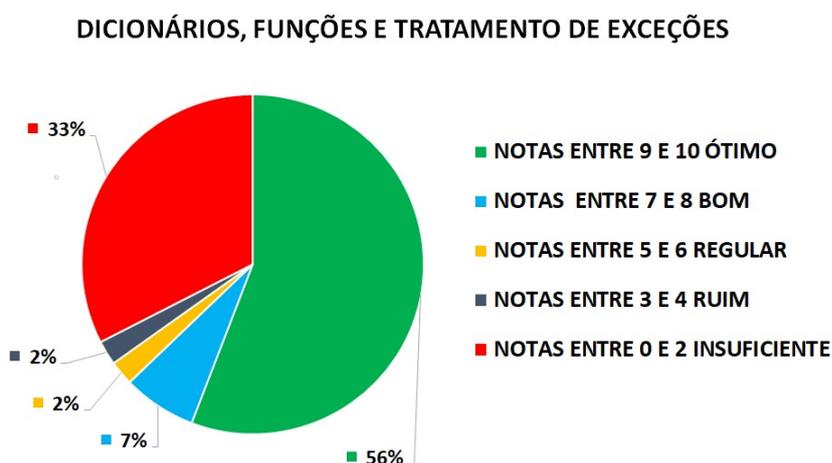
Os alunos ao serem perguntados quais suas maiores dificuldades, 31,6% responderam ser com o uso de matrizes, e 15,8% informaram ter dificuldades com listas.

Em resposta à questão de pesquisa QP5, pode-se afirmar que 100% dos alunos conseguiram aprender tudo o que foi ensinado sobre tuplas e listas, 69% tiveram conhecimentos parciais sobre o uso de matrizes e matéria e que 31% dos estudantes tiveram bastante dificuldade.

7.3.6 QP6. Os alunos conseguiram aprender o conteúdo de dicionários com listas compostas, uso de funções e tratamento de exceções? Quais foram as dificuldades encontradas?

Os últimos conteúdos ensinados foram os de dicionários e seu uso dentro de listas compostas, criação de funções e tratamento de erros. Os resultados obtidos nos exercícios nas aulas síncronas e assíncronas podem ser vistos na Figura 54, que equivalem a dez exercícios individuais e oito exercícios colaborativos.

Figura 54 – Resultados obtidos para dicionários, funções e tratamento de exceções



Fonte: Acervo pessoal

Houve um aumento significativo de 37% nas notas ótimas quando comparado aos assuntos aplicados anteriormente, mesmo aplicando listas compostas. Foi possível perceber durante as atividades que 70% dos alunos tinham um domínio completo da linguagem de programação, e que 7% ainda possuíam poucas dificuldades. Assim, para que fosse possível nivelar todos os conhecimentos obtidos foram aplicados dois modelos de projetos colaborativos descritos na subseção 6.1.1, com o intuito de potencializar os alunos de forma educativa e profissional.

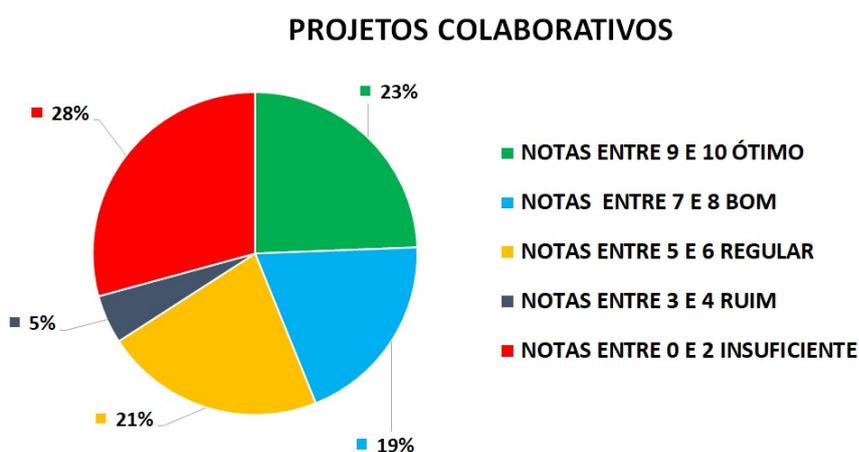
Um último questionário foi aplicado com os alunos para saber o que eles tiveram mais dificuldades e 10,5% responderam que foi na criação e manipulação de funções, 28,1% declararam ter tido dificuldades com dicionários, e 9,4% informaram ainda possuírem dificuldades com listas compostas.

A resposta para a QP6 é sim, todos os alunos conseguiram aprender os conteúdos de dicionários, funções e tratamento de exceções.

7.3.7 QP7. O uso de projetos colaborativos facilitaram o desenvolvimento dos alunos no EAD?

A resposta para a QP7 é que sim, os projetos colaborativos facilitaram o desenvolvimento dos alunos, sendo que dos 43 alunos inscritos como ativos, apenas 33 foram até o final da matéria de forma participativa nas aulas, atividades e projetos. O que permitiu os resultados que são vistos na Figura 55.

Figura 55 – Resultados obtidos com os projetos colaborativos



Fonte: Acervo pessoal

Os resultados foram que 23% dos alunos tiveram notas ótimas, 19% notas boas, 21% notas regulares, 5% notas ruins e 28% notas insuficientes. Com isso, tivemos que dos 33 alunos que concluíram a matéria apenas um reprovou, tendo um aproveitamento geral de 74% da turma de aprovados e de 2% de reprovados e de alunos que reprovaram por não participar das aulas de 23%.

Quando os alunos foram perguntados sobre os projetos colaborativos, 93,8% informaram que a abordagem utilizada facilitou seu desenvolvimento na matéria, e aqui vale ressaltar que uma das alunas que vinha sendo acompanhada e que até a primeira prova tinha bastante dificuldade, ela conseguiu dar a volta por cima e se recuperar na matéria de forma a ser ela quem ajudava os colegas durante as aulas de projetos.

7.3.8 QP8. A abordagem construcionista no EAD, facilitou o desenvolvimento dos alunos? E quais os principais pontos a serem destacados sobre a abordagem?

Como forma de fazer uma análise qualitativa e não quantitativa sobre as ferramentas utilizadas, sobre a metodologia de ensino e abordagem de atividades usadas durante o período separamos as opiniões dos alunos em quatro sub-tópicos, o primeiro fala sobre a metodologia, o segundo sobre os projetos colaborativos no Replit, o terceiro sobre o WhatsApp e o quarto sobre o uso do The Huxley para as atividades. A seguir fazemos uma análise geral.

1. No que diz respeito a metodologia os alunos responderam:

- “É uma metodologia ótima, principalmente a segunda unidade, me fez ‘dominar’ ativamente os conceitos aprendidos na primeira unidade.”
- “Ótima metodologia. Finalmente tive interesse com relação a matéria e entendi o verdadeiro sentido de ter ela no meu curso e na vida também.”
- “Perfeita, ensinou de maneira bem didática com bastante exemplo”
- “Achei boa. Só não gostei um pouco porque ao longo do período tivemos muitas atividades pra entregar e acaba que fica impossível entregar todas.”
- “Bacana, porém achei muito automatizado no que tange as atividades passadas para serem respondidas no The Huxley. Poderia ser um pouco menos robótico igual eram nas reuniões das atividades em grupos.”

2. Com relação aos projetos colaborativos e ao Replit:

- “Achei criativo, e ainda com a presença do professor, poderia ter ajuda nas dificuldades, bem legal.”
- “Foi desafiador, pois estávamos livres para desenvolver qualquer código. Mas, podemos mostrar nossa criatividade sem limitações.”
- “Ter que criar um problema e sua solução, assim como montar um código baseado nele do zero nos obrigou a investir mais em estudar e compreender os assuntos.”
- “É interessante pois a gente cria o problema com base nos critérios requeridos, acaba que a gente ‘cria’ aplicações cotidianas aos conceitos de programação.”
- “Foi essencial pra esclarecer algumas coisas que ainda estavam no ar.”
- “Achei legal, pois faz com que se aplique nossos conhecimentos num problema que nós mesmos fomos responsáveis por criar, melhorando assim a visão de como tudo funciona.”
- “Ajudou muito na compreensão do assunto, pois tivemos que preencher as lacunas em relação ao conteúdo.”

- “Muito boa podemos trabalhar em grupo, e desenvolver ainda mais o conhecimento aprendido durante o período, expondo nossas ideias e forma de programar fazendo o nosso próprio código.”
- “Achei a forma mais interessante de aprender pi. Pois os componentes do grupo trocaram os seus conhecimentos.”

3. No que diz respeito aos grupos do WhatsApp:

- “É interessante pois você tem uma sensação de maior proximidade pelo WhatsApp do que pelo e-mail, o que facilita a comunicação e a resolução de problemas, dúvidas, entre outros.”
- “Quebrava bastante o gelo na comunicação aluno/professor o que incentivava o aluno a continuar na matéria e a tirar dúvidas.”
- “Ele me ajudava bastante com os códigos do The Huxley, foi fundamental para eu conseguir finalizar os meus códigos, sempre estava disposto e respondia o mais rápido possível, gostei muito.”
- “Acessibilidade. Qualquer hora que você estivesse fazendo seu programa e não conseguisse algo, mandando para o professor, resposta rápida.”
- “Foi excelente. Ajuda a sanar nossas dúvidas rápido e o professor nos ajudou a não desistir, e sim tentar de novo a cada desanimo.”
- “A comunicação facilitou muito, muitas vezes fica difícil por e-mail e por whatsapp foi mais prático.”
- “É interessante pois você tem uma sensação de maior proximidade pelo WhatsApp do que pelo e-mail, o que facilita a comunicação e a resolução de problemas, dúvidas, entre outros.”
- “Foi importante pois a matéria exige muito cuidado na hora de resolver os códigos. Então ter um professor disponível no whatsapp contribuiu para entendermos os nossos erros e ter ajuda mais rápida e de forma que compreendemos.”

4. Com relação ao The Huxley:

- “O professor explica muito bem e fez o possível para que pudéssemos absorver o conteúdo dado em aula, sua didática foi clara e objetiva, mas o uso do The Huxley deixou um pouco a desejar, pois me limitou a programar de um único jeito correto.”
- “Gostei da matéria, pois me ajudou a entender bastante sobre a importância da programação na sociedade e no mercado de trabalho, me fez despertar interesse em continuar. O professor, foi compreensivo e ajudou muito os alunos, tanto no ensino como para tirar dúvidas sobre as dificuldades, gostei. Não gostei apenas do número de questões em cada tópico do The Huxley, poderia ser menos, ficou exaustivo.”

- “Muitas questões passadas no The Huxley e com um único modelo certo para o sistema. Como programar leva um certo tempo (entre erros e acertos), ‘perdíamos’ muito tempo na matéria que acabava tirando o foco de outras tão quão importantes.”
- “A minha maior sugestão é em relação a demanda gigante de atividades semanais. Acredito que no início do conteúdo é válido questionários grandes mas quando o assunto complica, poderiam ser listas menores. A quantidade de exercícios muitas vezes condiciona o aluno a entregar e não aprender, é o fazer porque preciso passar.”
- “A única coisa que me complicou na matéria foi o The Huxley, principalmente o envio de respostas. Muitas das vezes não nos fornece o caso teste pra entender como acontece o erro e fica algo muito abstrato de corrigir, acaba frustrando um pouco com o prosseguimento, principalmente quando os itens vão ficando mais complexos(não difíceis em si, mas complicados de imprimir os resultados conforme o The Huxley deseja).”

A análise feita com o que foi observado em sala de aula com o que foi dito pelos alunos é de que, faz-se necessário um ajuste na quantidade de atividades a serem passadas para os alunos responderem de forma individual que não comprometa o desenvolvimento de cada uma. O outro ponto é de que, o uso do The Huxley em conjunto com a metodologia construcionista não se mostrou eficaz no que diz respeito a abordagem de ensino utilizada, sendo assim, é necessário utilizar apenas uma plataforma de programação, que nesse caso a mais eficaz foi o Replit por permitir o desenvolvimento conjunto da turma junto com pequenos e grandes projetos colaborativos. Diante disso, vale ressaltar que por mais que o WhatsApp seja uma ferramenta que se mostrou extremamente importante para a metodologia, faz-se necessário para aqueles que desejam replicar utilizá-la de forma controlada, pois precisa-se de tempo para responder a todas as dúvidas. Caso o local de ensino possua tutores, ou alunos que tenham um bom desenvolvimento durante a matéria, os grupos do WhatsApp se mostrou extremamente eficaz, visto que, mesmo se o professor demorasse a tirar uma dúvida outros alunos com mais experiência respondiam, gerando um engajamento educacional ótimo.

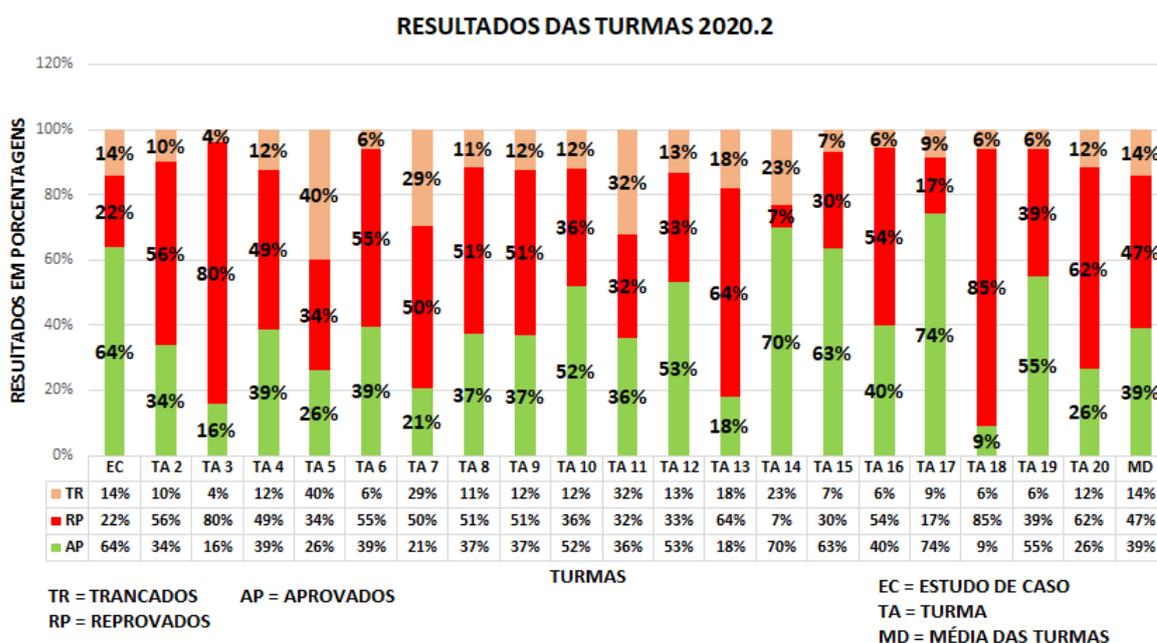
Pode-se destacar que a abordagem construcionista melhor se encaixa para professores com poucas turmas, ou com turmas de até 20 alunos, a experiência vivida demonstrou que é uma ótima opção analisar e está ao lado dos alunos e que sim, isso tem um impacto bastante positivo nos resultados de aprendizagem dos alunos.

É preciso destacar que o Replit foi uma ferramenta útil para o ensino de programação, seja, de forma individual, ou colaborativa, pois permitiu ao professor sanar dúvidas de qualquer aparelho, como os *smartphones* e computadores. Além disso, a ferramenta do Google, o Classroom foi essencial para o EAD, pois permitiu ao professor o envio de arquivos como *script*, materiais de aula, além de poder realizar reuniões online sem limite de tempo com toda a turma ou com os subgrupos a qualquer hora e lugar e de qualquer dispositivo. Ainda, é preciso falar da

importância que foi o uso do WhatsApp durante o estudo, pois ela potencializou o alcance para tirar dúvidas em grupo ou individualmente.

Os resultados de aprovação, reprovação e evasão acadêmica de todas as turmas do semestre e esses dados podem ser vistos na Figura 56.

Figura 56 – Resultados de todas as turmas do semestre



Fonte: Acervo pessoal

Temos que, a média alunos por turma foi de 39 e que o estudo de caso possuiu 50 alunos. Dentro da teoria construcionista é preciso trabalhar com uma média de 20 a 30 estudantes, por isso os alunos foram divididos em subgrupos de até cinco pessoas, para facilitar o acompanhamento.

O outro ponto a ser destacado é que ao verificarmos a média de aprovação das outras turmas que utilizaram do método tradicional, constatou-se que foi de 39% e que a do estudo de caso foi de 64% um bom resultado final, e mesmo analisando algumas turmas com médias aproximadas ou maiores percebemos que a abordagem de ensino ainda sim se mostra eficaz.

Já no que diz respeito a média de reprovados foram de 47% e que o estudo de caso possuiu metade desse número sendo umas das três turmas com os menores índices ficando com 22%. Os números de reprovados mesmo abaixo da média são considerados altos, mas ao analisarmos os alunos que realmente realizaram atividades, fizeram provas o resultado final seria de apenas 2% o que faz referência a apenas um aluno dos 33 ativos no período.

Já com relação a média de evasão acadêmica e os resultados do estudo de caso os números se mantiveram iguais em 14%. Vale destacar que, os alunos que trancaram a matéria na turma fizeram isso no início do período, e também não chegaram a participar de nenhuma atividade desde o início do período até o momento de trancar.

7.3.9 Ameaças à validade

Nessa seção abordamos as ameaças quanto à validade da pesquisa realizada que são:

1. Validade de construção: a escolha do The Huxley para analisar os códigos dos alunos de forma automatizada se deu por um consenso entre os professores, foi ajustado dentro da metodologia construcionista no início do período, que inviabilizou um pouco a metodologia e o desenvolvimento dos alunos. Para atenuar esta ameaça, procurou-se analisar o contexto das aulas e de forma estratégica retirá-la, ficando apenas com uma única plataforma de programação que foi o Replit, que poderia ser utilizada em qualquer dispositivo móvel ou *desktop*, em conjunto com o Python constatou-se que era a melhor opção.
2. Validade de seleção: a metodologia de ensino construcionista e o uso do Replit e projetos colaborativos em um ambiente EAD foi uma escolha do professor responsável pela turma. Para que não houvesse algum viés de seleção a mesma foi testada de forma parcial no período anterior para constatar se as escolhas eram as melhores opções.
3. Validade interna: os resultados obtidos e analisados não levaram em consideração outras metodologias, outras ferramentas ou até mesmo outras abordagens de ensino. Desta forma, não é possível realizar afirmações concretas, já que nem todas as variáveis foram analisadas. Isso pode gerar um viés no resultado, que não pôde ser mitigado devido à pandemia.
4. Validade externa: apesar de analisar os resultados das 20 turmas no período informado não é possível afirmar que a metodologia escolhida é mais eficiente do que a metodologia tradicional, mesmo que a quantidade de exercícios, avaliação e ensino tenham sido iguais até um certo período, e que não tenha sido aplicada a mesma linguagem em quase todas as turmas, não é possível realizar qualquer afirmativa, além de realizar as análises dos resultados obtidos. Porém, esse trabalho apresentou evidências sobre sua aplicação da maneira mais detalhada possível para que a metodologia possa ser replicada por outros professores.
5. Delineamento experimental: por não se tratar de uma pesquisa científica, mas, sim, de um relato de experiência, não foi utilizado nenhum um grupo de controle, nem homogeneidade entre grupos, nem distribuição aleatória. Essa dissertação não tem um viés experimental e sim um viés qualitativo, que busca passar informações de um relato de estudo de caso, mesmo que traga os resultados por meio de gráficos estatísticos, que fazem relação com o próprio trabalho e não com grupos de pesquisas.

7.4 Discussão

As aplicações realizadas buscaram mostrar um método de ensino baseado em pequenos projetos, de forma colaborativa e com acompanhamento educacional. Percebeu-se que no ensino

fundamental o construcionismo foi melhor aplicado por se tratar de uma turma com poucos alunos e mesmo com só 45 minutos de aula era possível atender os alunos de forma individual.

No contexto do ensino superior, aplicação do construcionismo foi um grande desafio, pois por ter quase que o triplo de alunos desejáveis para uma boa aplicação era preciso uma dedicação muito grande por parte do professor muito extenso. Isso já não foi tão sentido na segunda vez que a metodologia foi aplicada, mesmo que em um ambiente EAD, foi possível acompanhar todos os alunos, já que a estratégia utilizada foi de dividir os estudantes em pequenos grupos de até cinco pessoas o que facilitou e muito o trabalho do professor responsável.

Os objetivos alcançados com ambas as intervenções foi a utilização de dispositivos móveis na educação e essa foi um sucesso, visto que todos os alunos que participaram de forma ativa nas aulas conseguiram se desenvolver a ponto de estarem aptos a realizarem qualquer tipo de programa e solucionar problemas desde formas simples até as mais criativas.

Tendo em vista que a estratégia de utilizar o construcionismo nasceu com os alunos do ensino fundamental, pode-se afirmar que o seu uso do ensino superior foi essencial nos resultados obtidos, mesmo essa sendo aplicada em contextos com poucos ou muitos recursos educacionais, ela permite que o professor consiga identificar os alunos com mais problemas, e que consiga também traçar estratégias para sanar todas as dúvidas. Além disso, o *mobile learning* se mostrou eficiente em ambas as intervenções, pois permitiu a troca de informações, e o acesso a qualquer recursos em qualquer hora e lugar.

8

Conclusões

A fim de minimizar as dificuldades encontradas durante os estudos de casos, a metodologia usada no ensino superior foi sendo aperfeiçoada em sua aplicação período após período, mesmo com a pandemia. O que não foi possível no ensino fundamental onde as aulas não retornaram nem a distância até o presente momento que essa dissertação foi escrita.

Propomos o uso de dispositivos móveis e computadores tanto no ensino fundamental quanto no ensino superior. Esses dispositivos potencializam as habilidades dos alunos e permitem a acessibilidade de todos.

A utilização dos dispositivos móveis no ensino de programação seja com o ScratchJr por meio de *tablets*, ou o Python de forma acessível por meio do Replit, oferece aos alunos além da mobilidade e da aprendizagem móvel, diversos outros recursos:

1. A decomposição de problemas algoritmos envolvendo informação audiovisual, por meio de videoaulas e reuniões online.
2. Construção de programas usando recursos visuais. O ScratchJr pode ser programado por meio de blocos. O Replit pode ser aplicado de forma visual e colaborativa. O WhatsApp permite o envio de códigos fontes de forma rápida para dúvidas fora do horário das aulas, tanto para os colegas, como para tutores e para o professor, permitindo socialização e discussão de soluções ampliando as oportunidades de interação.
3. O acesso aos recursos educacionais a qualquer lugar dos alunos seja por um dispositivo móvel ou *desktop*, a materiais de aulas, exercícios, exemplos de fluxogramas, animações e vídeos gravados por meio do Google Classroom.

A utilização de *mobile learning* no ensino de introdução a programação para alunos da rede pública a nível fundamental ou superior tem potencial para melhorar o desempenho

dos estudante e traz bons resultados quando apoiado por uma metodologia construcionista com projetos colaborativos, com o uso de apoio multimídia tal como vídeos, gráficos e exemplos de acordo com o cotidiano dos alunos.

No ensino fundamental, foi possível vivenciar como o aluno compreende o ensino de programação por meio de um *tablet* ou atividades desplugadas, e seu uso de forma interdisciplinar, permitindo explorar as habilidades dos estudantes dentro de cada atividade realizada. Isso possibilitou a criação de uma cartilha de atividades com conceitos teóricos atrelados aos práticos.

O ensino de computação durante as aulas, permitiu uma transversalidade de conceitos, que foram capazes de melhorar significativamente as habilidades de resolução de problemas. Foi visto ainda que os alunos, ao interagirem com o *ScratchJr*, puderam criar programas funcionais e criativos.

Acreditamos dentro do que foi proposto e do perfil de cada estudante que os objetivos para estimular as habilidades de concentração, coordenação e raciocínio lógico foram alcançados. Isso foi confirmado pela pedagoga responsável, que relatou que os alunos passaram a ter um desempenho melhor durante as aulas no processo de alfabetização.

No ensino superior foi explorado uma metodologia de ensino baseado no construcionismo de Seymour Papert em conjunto com Google Classroom, WhatsApp, Replit e QPython, realizando um acompanhamento individual de cada aluno, com aulas interativas, exercitando a criatividade, comunicação, trabalho em equipe e a capacidade de solucionar problemas.

Na abordagem realizada no ensino presencial, com foco no uso do QPython e do WhatsApp, foi possível observar que, como os alunos tinham habilidades com os *smartphones* quase não era perceptível a diferença de tempo para quem utilizava um computador para programar. Uma observação quanto ao QPython foi de que alguns alunos sentiram dificuldade ao utilizá-lo por ser todo em inglês.

Quanto à abordagem realizada no ensino a distância, foi utilizado o Google Meet e o Replit com todos os alunos e com pequenos grupos por meio da sala invertida, o que facilitou o acompanhamento, deu dinamismo e gerou resultados positivos para a turma.

O WhatsApp em ambos os estudos de caso foi um ferramenta essencial, porque possibilitou uma comunicação interpessoal e rápida para sanar dúvidas durante e após as aulas, servindo até como um grande fórum de debate. É preciso destacar que não houve problemas e que, se utilizado de forma correta, pode servir como suporte para a educação.

A construção dos resultados apresentados só foi possível graças ao engajamento dos alunos na disciplina. Acredita-se, pelos comentários dos estudantes, que os projetos colaborativos, os desafios, as gincanas e o suporte via WhatsApp dentro e fora da universidade, tenham gerado os resultados alcançados. Pode-se afirmar que o construcionismo por abordar exercícios e projetos relacionados ao cotidiano do aluno, pode gerar motivação e evitar a evasão acadêmica, permitindo ainda que continuem praticando e desenvolvam seus conhecimentos de forma conjunta e nivelada.

A experiência vivida e os índices apresentados mostram que é possível ensinar e aprender utilizando apenas um *smartphone*, um *tablet*, um *notebook* ou ainda um computador. Contudo, para que a metodologia construcionista possa ser executada de forma eficiente no acompanhamento individual dos alunos, nas dificuldades encontradas por eles, é necessário empenho por parte do professor, principalmente nas turmas do ensino superior, que geralmente contam com, no mínimo, 50 alunos. O professor deve analisar e preparar estratégias durante as aulas e plantões.

Com efeito, buscamos não somente trazer uma nova perspectiva de ensino, mas também mostrar sua importância ao praticar inclusão social e digital, permitindo que todos os envolvidos tenham acesso à aprendizagem com os recursos disponíveis. Como trabalhos futuros, pretendemos ampliar o projeto e continuar a replicar a metodologia de ensino no nível superior, como também desejamos abordar no nível médio, a fim de aprimorá-la, ampliando o alcance e número de beneficiados.

Concluimos que a metodologia obteve êxito, pois os resultados encontrados evidenciam que a metodologia pode ser utilizada no ensino introdutório de programação no nível fundamental e superior, deixando os alunos aptos para trabalharem com foco em soluções, de forma colaborativa dentro do ambiente educacional ou no mercado de trabalho.

Referências

ALVES, N. et al. Ensino de computação de forma multidisciplinar em disciplinas de história no ensino fundamental – um estudo de caso. *Revista Brasileira de Informática na Educação - RBIE*, v. 24, n. 3, p. 31, 2016. ISSN 2317-6121. Acesso em 1 de dezembro de 2019. Citado na página 29.

AMRO, J. S.; ROMLI, R. Investigation on the learning programming techniques via mobile learning application. In: *2019 4th International Conference and Workshops on Recent Advances and Innovations in Engineering (ICRAIE)*. [S.l.: s.n.], 2019. p. 1–7. Citado 3 vezes nas páginas 25, 26 e 36.

ANDRADE, E. A. Mobile learning e programação: o desenvolvimento de apps como metodologia para a aprendizagem da programação em contexto universitário. 2019. Citado 2 vezes nas páginas 26 e 27.

ANDRÉ, S. et al. Scratch jr - brincando em família: um relato de experiências sobre introdução a programação no ambiente escolar. *Anais do Workshop de Informática na Escola*, v. 23, n. 1, p. 432–441, 2017. ISSN 2316-6541. Disponível em: <<http://www.br-ie.org/pub/index.php/wie/article/view/7261>>. Citado 2 vezes nas páginas 26 e 27.

AUSUBEL, D.; NOVAK, J.; HANESIAN, H. *Psicologia Educacional*. Rio de Janeiro: Interamericana, 1980. Citado na página 44.

BARBOSA, A. M. e E. Softwares educacionais para o ensino de programação: Um mapeamento sistemático. *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)*, v. 26, n. 1, p. 190, 2015. ISSN 2316-6533. Disponível em: <<http://www.br-ie.org/pub/index.php/sbie/article/view/5150>>. Citado na página 25.

BARBOSA, D. S. e Marisa Batista e E. Problemas e dificuldades no ensino de programação: Um mapeamento sistemático. *Revista Brasileira de Informática na Educação*, v. 24, n. 1, 2016. Disponível em: <<http://www.br-ie.org/pub/index.php/rbie/article/view/3317>>. Citado na página 24.

BASIL VICTOR, R.; WEISS DAVID, M. A Methodology for Collecting Valid Software Engineering Data. *IEEE Transactions on Software Engineering*, v. 6, n. 6, 1984. ISSN 1939-3520. [GS Search](#). Citado na página 73.

BELTRAN SOTOMAYOR, C. First year undergraduate electronic engineering students from a Peruvian university learn to program with Python. In: IEEE (Ed.). *2019 IEEE World Conference on Engineering Education (EDUNINE)*. Lima, Peru: IEEE Xplore, 2019. p. 1–4. [GS Search](#). Citado na página 21.

BEN-ARI, M. Constructivism in computer science education. *SIGCSE Bull.*, Association for Computing Machinery, New York, NY, USA, v. 30, n. 1, p. 257–261, mar. 1998. ISSN 0097-8418. Acesso em 17 de janeiro de 2020. Citado na página 30.

Bers, M. U. Coding, playgrounds and literacy in early childhood education: The development of kibo robotics and scratchjr. In: *2018 IEEE Global Engineering Education Conference*

(EDUCON). [S.l.: s.n.], 2018. p. 2094–2102. Acesso em 29 de setembro de 2019. Citado na página 36.

BLIKSTEIN, P. O pensamento computacional e a reinvenção do computador na educação. *Education and Courses*, 2008. Acesso em 17 de janeiro de 2020. Disponível em: <https://www.blikstein.com/paulo/documents/online/ol_pensamento_computacional.html>. Citado na página 32.

BORBA, M.; LACERDA HANNAH, D. Políticas públicas e tecnologias digitais: Um celular por aluno public policies and digital technologies: A smartphone per student. *Educação Matemática Pesquisa : Revista do Programa de Estudos Pós-Graduados em Educação Matemática*, v. 1, n. 3, 2015. ISSN 1983-3156. [GS Search](#). Citado na página 20.

Brackmann, C. et al. Computational thinking: Panorama of the americas. In: *2016 International Symposium on Computers in Education (SIIE)*. [S.l.: s.n.], 2016. p. 1–6. ISSN null. Acesso em 14 de janeiro de 2020. Acesso em: Acesso em 14 de janeiro de 2020. Citado 2 vezes nas páginas 28 e 29.

BRACKMANN, C. P. *Desenvolvimento do pensamento computacional através de atividades desplugadas na educação básica*. 226 p. Tese (Doutorado) — Universidade Federal do Rio Grande do Sul - UFRGS, Porto Alegre, RS. [Acesso em 3 de julho de 2019], 2017. Acesso em 3 de julho de 2019. Disponível em: <<http://hdl.handle.net/10183/172208>>. Acesso em: Acesso em 3 de julho de 2019. Citado na página 20.

BREHM, L. et al. Collaborative Learning with COZMO to Teach Programming in SCRATCH and Python. In: IEEE (Ed.). *2019 IEEE Global Engineering Education Conference (EDUCON)*. Dubai, Emirados Árabes Unidos: IEEE Xplore, 2019. p. 448–452. [GS Search](#). Citado na página 76.

BRENNAN, K.; RESNICK, M. New frameworks for studying and assessing the development of computational thinking. In: *In AERA 2012*. [S.l.: s.n.], 2012. Citado na página 32.

CARNEIRO, F.; MARINHO, S. Dispositivos móveis e a formação para a cidadania digital: o desafio de família e escola. *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, v. 7, n. 1, 2018. Disponível em: <<https://www.br-ie.org/pub/index.php/wcbie/article/view/8327>>. Citado na página 24.

CARVALHO, M.; AGUIAR, P.; DANTAS, V. Ensino da estrutura de repetição For em Python com realidade aumentada através do Aurasma. In: CTRL+E (Ed.). *Anais do II Congresso sobre Tecnologias na Educação - Ctrl+E*. Paraíba, Brasil: Ctrl+E, 2017. p. 443–454. [GS Search](#). Citado 2 vezes nas páginas 26 e 27.

CHRISTIAN, B.; GRIFFITHS, T. *Algoritmos para viver: a ciência exata das decisões humanas*. São Paulo: Companhia das Letras, 2017. Citado na página 38.

ENGLISH, L. D. Stem education k-12: perspectives on integration. *International Journal of STEM Education*, v. 3, n. 1, p. 3, Mar 2016. ISSN 2196-7822. Acesso em 08 de fevereiro de 2020. Citado na página 20.

FARIAS, C. et al. Estimulando o pensamento computacional: uma experiência com scratchjr. *Anais do Workshop de Informática na Escola*, v. 25, n. 1, p. 197, 2019. ISSN 2316-6541. Disponível em: <<https://br-ie.org/pub/index.php/wie/article/view/8506>>. Citado na página 29.

FGV. *Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira: Censo escolar*. 2021. Disponível em: <<https://portal.fgv.br/noticias/brasil-tem-424-milhoes-dispositivos-digitais-uso-revela-31a-pesquisa-anual-fgvicia>>. Citado na página 23.

FLANNERY, L. P. et al. Designing scratchjr: Support for early childhood learning through computer programming. In: *Proceedings of the 12th International Conference on Interaction Design and Children*. New York, NY, USA: Association for Computing Machinery, 2013. (IDC '13), p. 1–10. ISBN 9781450319188. Acesso em 17 de janeiro de 2020. Citado 2 vezes nas páginas 36 e 37.

Heintz, F.; Mannila, L.; Färnqvist, T. A review of models for introducing computational thinking, computer science and computing in k-12 education. In: *2016 IEEE Frontiers in Education Conference (FIE)*. [S.l.: s.n.], 2016. p. 1–9. Acesso em 14 de janeiro de 2020. Citado 2 vezes nas páginas 20 e 28.

HERRERA, S. I. et al. Mobile technologies in engineering education. In: *2015 International Conference on Interactive Collaborative Learning (ICL)*. [S.l.: s.n.], 2015. p. 1157–1164. Citado na página 19.

HILL, C. et al. Floors and flexibility: Designing a programming environment for 4th-6th grade classrooms. In: *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*. New York, NY, USA: Association for Computing Machinery, 2015. (SIGCSE '15), p. 546–551. ISBN 9781450329668. Acesso em 17 de janeiro de 2020. Disponível em: <<https://doi.org/10.1145/2676723.2677275>>. Citado na página 37.

HOFF, M. S.; WECHSLER, S. M. Processo resolutivo do jogo computadorizado tetris: análise microgenética. *Psicologia: Reflexão e Crítica*, scielo, v. 17, p. 129 – 141, 2004. ISSN 0102-7972. Acesso em 10 de janeiro de 2020. Disponível em: <<http://dx.doi.org/10.1590/S0102-79722004000100016>>. Citado na página 40.

INEP. *Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira: Censo escolar*. 2019. Disponível em: <<http://portal.inep.gov.br/resultados-e-resumos>>. Citado na página 20.

INEP. *Fundação Getúlio Vargas: Portal FGV*. 2021. Disponível em: <<https://www.gov.br/inep/pt-br/assuntos/noticias/centso-escolar/divulgados-dados-sobre-impacto-da-pandemia-na-educacao>>. Citado na página 20.

IOANNOU, A.; MAKRIDOU, E. Exploring the potentials of educational robotics in the development of computational thinking: A summary of current research and practical proposal for future work. *Education and Information Technologies*, v. 23, n. 6, p. 2531–2544, 2018. ISSN 1573-7608. Acesso em 20 de maio de 2019. Acesso em: Acesso em 20 de maio de 2019. Citado na página 31.

JOHNSON, C. I.; MAYER, R. E. A testing effect with multimedia learning. *Journal of Educational Psychology*, v. 101, n. 3, p. 621–629, 2009. Disponível em: <<https://doi.org/10.1037/a0015183>>. Citado na página 43.

KAZAKOFF, E. R. *Cats in Space, Pigs that Race: Does self-regulation play a role when kindergartners learn to code?* 180 p. Tese (Doutorado) — Tufts University, Medford/Somerville, Massachusetts, EUA. [Acesso em 17 de janeiro de 2020], 2014. Acesso em 17 de janeiro de

2020. Disponível em: <https://ase.tufts.edu/DevTech/resources/Theses/EKazakoff_2014.pdf>. Acesso em: Acesso em 17 de janeiro de 2020. Citado na página 37.

KAZAKOFF, E. R. *Cats in Space, Pigs that Race: Does self-regulation play a role when kindergartners learn to code?* 180 p. Tese (Doutorado) — Tufts University, Medford/Somerville, Massachusetts, EUA. [Acesso em 17 de janeiro de 2020], 2014. Acesso em 17 de janeiro de 2020. Disponível em: <https://ase.tufts.edu/DevTech/resources/Theses/EKazakoff_2014.pdf>. Acesso em: Acesso em 17 de janeiro de 2020. Citado na página 107.

LIMA, M. et al. Experiência de aprendizagem móvel: o uso do WhatsApp no Ensino Fundamental. *Anais dos Workshops do Congresso Brasileiro de Informática na Educação - WCBIE*, v. 5, n. 1, 2016. ISSN 2316-8889. [GS Search](#). Citado na página 78.

LOPES CRISTIANO, G.; VAZ, B. B. O Uso Pedagógico dos Grupos do Whatsapp no Ensino de História. In: *V Congresso Internacional de História - Novas Epistemes e Narrativas Contemporâneas*. Goiás, Brasil: [s.n.], 2016. [GS Search](#). Citado na página 78.

MANOEL, L. R. S. *Torre de Hanói*. [s.n.], 2019. Acesso em 12 de dezembro de 2019. Disponível em: <https://www.ibilce.unesp.br/Home/Departamentos/Matematica/labmat/torre_de_hanoi.pdf>. Citado na página 41.

MARCOLINO, A. S.; BARBOSA, E. A survey on problems related to the teaching of programming in brazilian educational institutions. In: *2017 IEEE Frontiers in Education Conference (FIE)*. [S.l.: s.n.], 2017. p. 1–9. Citado 2 vezes nas páginas 24 e 25.

MARCOLINO, A. S. et al. Towards a catalog of gestures for m-learning applications for the teaching of programming. In: *2018 IEEE Frontiers in Education Conference (FIE)*. [S.l.: s.n.], 2018. p. 1–9. Citado na página 24.

MARTINS, A. R. Q. *Usando o Scratch para potencializar o pensamento criativo em crianças do Ensino Fundamental*. 114 p. Tese (Doutorado) — Universidade de Passos Fundos - UPF, Porto Alegre, RS. [Acesso em 18 de agosto de 2019], 2012. Acesso em 18 de agosto de 2019. Disponível em: <<http://tede.upf.br/jspui/bitstream/tede/750/1/2012AmiltonRodrigodeQuadrosMartins.pdf>>. Citado na página 33.

MARTINS, E.; GOUVEIA, L. Uso do WhatsApp em Atividades Educativas Extraclasse On-line no Ensino de Programação. In: *Anais do XXVII Workshop sobre Educação em Computação - WEI*. Porto Alegre, Brasil: SBC, 2019. p. 141–150. ISSN 2595-6175. [GS Search](#). Citado na página 78.

MIGLIORI, R. *Neurociências e educação*. São Paulo: editora Brasil, 2013. Citado na página 32.

MOHAMMAD, S. Effectiveness of M-Learning in Blended Learning-Design of Prototype Framework for AOU Bahrain. In: IEEE (Ed.). *2015 Fifth International Conference on e-Learning (econf)*. Manama, Bahrain: IEEE Xplore, 2015. p. 201–206. [GS Search](#). Citado na página 19.

NACARATO, A. M. Eu trabalho primeiro no concreto. revista de educação matemática. *Revista de Educação Matemática*, 2004. ISSN 1676-8868. Acesso em 10 de abril de 2020. Disponível em: <https://edisciplinas.usp.br/pluginfile.php/4291874/mod_resource/content/1/Nacarato_eu20trabalho20primeiro20no20concreto.pdf>. Citado na página 42.

NASCIMENTO, K.; FILHO, J. Dispositivos móveis na educação: ensinando e aprendendo em diferentes contextos. *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)*, v. 27, n. 1, p. 1225, 2016. ISSN 2316-6533. Disponível em: <<https://www.br-ie.org/pub/index.php/sbie/article/view/6812>>. Citado na página 24.

Nauman, M.; Uzair, M. Se and cs collaboration: Training students for engineering large, complex systems. In: *20th Conference on Software Engineering Education Training (CSEET'07)*. [S.l.: s.n.], 2007. p. 167–174. Citado na página 37.

NIELSEN. Um panorama sobre o uso de smartphone e internet no brasil. (*Último acesso em 27 de set de 2019*), 2015. Disponível em: <<https://www.nielsen.com/br/pt/press-releases/2015/68-milhoes-usam-a-internet-pelo-smartphone-no-brasil/>>. Citado na página 19.

NUNES, M. et al. Uso da lógica de programação para potencializar o pensamento criativo em crianças do ensino básico. *Anais dos Workshops do Congresso Brasileiro de Informática na Educação - WCBIE*, v. 7, n. 1, p. 325, 2018. ISSN 2316-8889. Acesso em 1 de janeiro de 2020. Citado na página 29.

ORTIZ, O. et al. M-learning tools: The development of programming skills in engineering degrees. *IEEE Revista Iberoamericana de Tecnologías del Aprendizaje*, v. 10, n. 3, p. 86–91, 2015. Citado 2 vezes nas páginas 19 e 25.

PAPERT, S. *Mindstorms: Children, Computers, and Powerful Ideas*. USA: Basic Books, Inc., 1980. [GS Search](#). Citado 3 vezes nas páginas 21, 30 e 75.

PAPERT, S. *Logo: computadores e educação*. Brasil: São Paulo, Brasiliense, 1986. [GS Search](#). Citado na página 30.

PAPERT, S. *A máquina das crianças : repensando a escola na era da informática*. Porto Alegre: Artmed, 2008. 220 p. ((Biblioteca de ciências da educação)). Tradução de Sandra Costa. ISBN 9788536310589. Citado na página 30.

PIAGET, J. *Psicologia e pedagogia*. [S.l.]: Rio de Janeiro: Florense, 1970. Citado na página 86.

RAABE, A.; ZORZO, A. F.; BLIKSTEIN, P. *Computação na Educação Básica Fundamentos e Experiências. Série: Tecnologia e Inovação na Educação Brasileira*. Porto Alegre: Grupo A, 2020. 316 p. ISBN 9786581334048. Citado na página 29.

RESNICK, M. Mother's day, warrior cats, and digital fluency: Stories from the scratch online community. In: *In Proceedings of the constructionism 2012 conference: Theory, practice and impact*. [s.n.], 2012. Acesso em 14 de janeiro de 2020. Disponível em: <<http://web.media.mit.edu/~mres/papers/mothers-day-warrior-cats.pdf>>. Acesso em: Acesso em 14 de janeiro de 2020. Citado na página 20.

RESNICK, M. et al. Scratch: Programming for all. *Commun. ACM*, Association for Computing Machinery, New York, NY, USA, v. 52, n. 11, p. 60–67, nov. 2009. ISSN 0001-0782. Acesso em 17 de janeiro de 2020. Citado na página 33.

RIBEIRO, M. M. B.; BRASILEIRO, T. S. A. Ambientes virtuais híbridos. uma possível solução para ensino de programação de computadores em uma universidade amazônica. In: *XVIII Simposio Internacional de Informática Educativa, SIIIE 2016*. [S.l.: s.n.], 2016. v. 15, n. 1, p. 173. Citado 2 vezes nas páginas 26 e 27.

ROMERO MORENO, L. M. Methodology with Python Technology and Social Network Analysis Tools to Analyze the Work of Students Collaborating in Facebook Groups. In: IEEE (Ed.). *2019 14th Iberian Conference on Information Systems and Technologies (CISTI)*. Coimbra, Portugal: IEEE Xplore, 2019. p. 1–6. [GS Search](#). Citado na página 21.

SANTANA, A. et al. Scratch jr - brincando em família: um relato de experiências sobre introdução a programação no ambiente escolar. *Anais do Workshop de Informática na Escola*, v. 23, n. 1, p. 432, 2017. ISSN 2316-6541. Acesso em 26 de Dezembro de 2019. Disponível em: <https://www.br-ie.org/pub/index.php/wie/article/view/7261>. Citado na página 29.

SANTOS G., J.; LIMA S., J. Robótica Educacional e Construcionismo como proposta metodológica para o desenvolvimento de ambientes de aprendizagem significativa. *Revista Novas Tecnologias na Educação - RENOTE*, v. 2, n. 16, 2018. ISSN 1679-1916. [GS Search](#). Citado na página 30.

Santos, P. S. C.; Araujo, L. G. J.; Bittencourt, R. A. A mapping study of computational thinking and programming in brazilian k-12 education. In: *2018 IEEE Frontiers in Education Conference (FIE)*. [S.l.: s.n.], 2018. p. 1–8. ISSN 1539-4565. Acesso em 14 de janeiro de 2020. Citado 2 vezes nas páginas 28 e 32.

SCOPIM, V. *A Torre de Hanói e o jogo de xadrez - uma proposta de ensino de matemática aliada ao lúdico*. [s.n.], 2014. Acesso em 10 de dezembro de 2019. Disponível em: <https://repositorio.roca.utfpr.edu.br/jspui/handle/1/4245>. Citado na página 41.

SHEARD, J. et al. Analysis of research into the teaching and learning of programming. In: *Proceedings of the Fifth International Workshop on Computing Education Research Workshop*. New York, NY, USA: Association for Computing Machinery, 2009. (ICER '09), p. 93–104. ISBN 9781605586151. Disponível em: <https://doi.org/10.1145/1584322.1584334>. Citado na página 19.

SILVA et al. Ensino-aprendizagem de programação: uma revisão sistemática da literatura. *Revista Brasileira de Informática na Educação*, v. 23, n. 01, p. 182, 2015. ISSN 2317-6121. Disponível em: <https://br-ie.org/pub/index.php/rbie/article/view/2838>. Citado na página 28.

SILVA, F. et al. Requisitos para integração entre ambientes de aprendizado e m-learning: uma revisão sistemática da literatura. *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)*, v. 26, n. 1, p. 269, 2015. ISSN 2316-6533. Disponível em: <https://www.br-ie.org/pub/index.php/sbie/article/view/5166>. Citado na página 24.

SILVA, R. *Conhecimentos matemáticos de professores dos anos iniciais do ensino fundamental: um estudo sobre o jogo da velha com figuras geométricas como recurso didático*. Dissertação (Dissertação de mestrado) — Universidade Federal de Pernambuco, 2017. Acesso em 15 de dezembro de 2019. Disponível em: <https://attena.ufpe.br/handle/123456789/29853>. Citado na página 43.

SILVA, R.; TELES, R. Conhecimentos geométricos mobilizados por professores na confecção do jogo da velha com figuras geométricas. *Anais do VII Encontro Pernambucano de Educação Matemática*, 2017. Acesso em 15 de dezembro de 2019. Disponível em: https://epem.sbempe.com.br/anais/2017/PDFs/CCO4332624432_112905.pdf. Citado na página 43.

SOLINGEN D., M.; BERGHOUT E., W. *The goal/question/metric method: a practical guide for quality improvement of software development*. INGLATERRA: McGraw-Hill, 1999. [GS Search](#). Citado na página 73.

SOUZA, D.; BATISTA, M.; BARBOSA, E. Problemas e dificuldades no ensino de programação: Um mapeamento sistemático. *Revista Brasileira de Informática na Educação*, v. 24, n. 1, p. 39, 2016. ISSN 2317-6121. Disponível em: <https://www.br-ie.org/pub/index.php/rbie/article/view/3317>. Citado na página 28.

STRAWHACKER, A. et al. Scratchjr demo: A coding language for kindergarten. In: *Proceedings of the 14th International Conference on Interaction Design and Children*. New York, NY, USA: Association for Computing Machinery, 2015. (IDC '15), p. 414–417. ISBN 9781450335904. Disponível em: <https://doi.org/10.1145/2771839.2771867>. Citado na página 37.

TELEBRASIL. Sete em cada 10 brasileiros usam a internet. *Associação Brasileira de Telecomunicações*, 2019. Último acesso em 30 de set de 2019. Disponível em: <http://www.agenciatelebrasil.org.br/Noticias/Sete-em-cada-10-brasileiros-usam-a-internet-395.html?UserActiveTemplate=site>. Citado na página 19.

UNESCO. *Diretrizes de políticas da UNESCO para a aprendizagem móvel*. [S.l.]: Organização das Nações Unidas para a Educação, a Ciência e a Cultura (UNESCO), 2014. ISBN 978-85-7652-190-7. Citado 2 vezes nas páginas 19 e 23.

UTESCH, M. C. et al. Pic2program - an educational android application teaching computational thinking. In: *2020 IEEE Global Engineering Education Conference (EDUCON)*. [S.l.: s.n.], 2020. p. 1493–1502. Citado 2 vezes nas páginas 25 e 26.

VALENTE, J. A. Integração do pensamento computacional no currículo da educação básica: diferentes estratégias usadas e questões de formação de professores e avaliação do aluno. *Revista E-curriculum*, v. 14, n. 3, p. 864–897, 2016. Acesso em 14 de outubro de 2019. Disponível em: <https://www.redalyc.org/pdf/766/76647706006.pdf>. Citado na página 20.

VIVO. *Fundação Telefônica VIVO*. 2019. Disponível em: <https://fundacaotelefonicavivo.org.br/acervo/projeto-aula-digital-chega-ao-estado-de-sergipe/>. Citado na página 31.

WANGENHEIM, C.; RODRIGUES, V.; SANTOS, G. Ensinar computação com o scratch nas escolas primárias - um estudo de caso. *Revista Brasileira de Informática na Educação - RBIE*, v. 22, n. 03, 2014. ISSN 2317-6121. Acesso em 1 de dezembro de 2019. Citado na página 28.

WING, J. M. Computational thinking. *Commun. ACM*, Association for Computing Machinery, New York, NY, USA, v. 49, n. 3, p. 33–35, mar. 2006. ISSN 0001-0782. Acesso em 17 de janeiro de 2020. Citado 2 vezes nas páginas 21 e 33.

Apêndices

APÊNDICE A – Formulário parte 1

sobre o ensino de Python presencial

1. Qual(is) tipo(s) de variável(is) você teve mais dificuldade em aprender?
 - int
 - string
 - float
 - boolean
 - Não tive dificuldade em aprender
2. Você teve mais dificuldade em aprender variável explícita ou implícita?
 - Variável explícita. Ex: idade = 25
 - Variável implícita. Ex: idade = int(input('Informe a sua idade'))
 - Não tive dificuldade em aprender
3. Relacionado ao comando print, você teve mais dificuldade em aprender?
 - Format. Ex: print('O cálculo dos números e é:'.format(74, 12))
 - Concatenação . Ex: print('O cálculo dos números'+74+' e '+12+' é:')
 - Não tive dificuldade em aprender
4. Relacionado aos comparadores, você teve mais dificuldade em aprender?
 - == igual
 - != não igual
 - > maior que
 - >= maior ou igual
 - < menor que
 - <= menor ou igual que
 - Não tive dificuldade em aprender
5. Relacionado ao assunto de estrutura de decisão, você teve mais dificuldade em aprender?
 - Marque todas que se aplicam.
 - If

- else
 - elif
 - If e else
 - If | elif | else
 - Não tive dificuldade em aprender
6. Relacionado ao assunto de estrutura de decisão com comparações, você teve mais dificuldade em aprender?
- comparações numéricas
 - comparações textuais
 - Não tive dificuldade em aprender
7. Relacionado ao assunto de estrutura de decisão com mais de uma comparação utilizando operador lógico, você teve mais dificuldade em aprender?
- AND
 - OR
 - NOT
 - Não tive dificuldade em aprender
8. Relacionado ao assunto de manipulação de strings e seus comandos, você teve mais dificuldade em aprender?
- IN. Ex: matéria = 'programação' print(('m' in matéria))
 - NOT IN. Ex: Ex: matéria = 'programação' print(('t' not in matéria))
 - startswith. Ex: estadio = 'maracanã' verdade = estadio.startswith('mara') resultado é igual a True
 - endswith. Ex: estadio = 'maracanã recife' verdade = estadio.endswith('recife') resultado é igual a True
 - Não tive dificuldade em aprender
9. Relacionado ao assunto de manipulação de strings e como acessar, você teve mais dificuldade em aprender?
- Acessar uma letra específica. Ex: matéria = 'programação' letra = matéria[2]
 - Acessar o prefixo de uma palavra. Ex: matéria = 'programação' letra = matéria[:2]
 - Acessar o sufixo de uma palavra. Ex: matéria = 'programação' letra = matéria[2:]

- Acessar parte de uma palavra. Ex: matéria = 'programação' letra = matéria[1:4]
 - Não tive dificuldade em aprender
10. Relacionado ao assunto de manipulação de strings e como alterar, você teve mais dificuldade em aprender?
- Concatenação de string. Ex: linguagem = 'python' texto = 'A linguagem é '+linguagem+' versão 3'
 - Capitalize. Ex: print('programação ',texto_capitalize.capitalize()) o resultado é Programação
 - Separação de String com Split. Ex: futebol = 'Copa do Mundo 2014' futebol = futebol.split(' ') resultado é [0] possui 'copa', na [1] possui 'do', na [2] possui 'Mundo' e na [3] possui '2014'
 - Não tive dificuldade em aprender
 - Outro:
11. Com relação aos problemas mais complexos sobre manipulação de String, fale com suas palavras as dificuldades obtidas?
12. Durante o período letivo você está utilizando para programar? (Um ou mais)
- Smartphone
 - Computador
 - Papel, caneta e lápis
13. Você pode afirmar que conseguiu aprender sobre os assuntos até o momento?
- Entre 0 e 25%
 - Entre 25% e 50%
 - Entre 50% e 75%
 - Aprendi todo o assunto
14. Relate aqui quais as dificuldades você está sentindo, quais exercícios está tendo mais dificuldade?
15. Sobre a metodologia de ensino abordada pelo o professor como está sendo para você, isso inclui todo o material de apoio, grupo no WhatsApp e etc?
- Ruim
 - Regular
 - Boa
 - Ótima

APÊNDICE B – Formulário parte 2 sobre o ensino de Python presencial

1. Qual laço de repetição você está tendo dificuldade?
 - for elemento in lista:
 - for i in range(qtd_elementos)
 - while true:
 - Não tive dificuldade em aprender
2. Você teve ou tem dificuldade com um laço de repetição dentro de outro?
 - consegui executar parcialmente
 - Consegui executar mais ainda tenho dificuldade
 - Não tive ou não tenho dificuldade
3. Informe aqui suas dificuldade com laços de repetição.
4. Sobre o conteúdo de tuplas, você consegue afirmar que:
 - Consigo manipular tuplas sem dificuldade.
 - Consigo manipular tuplas mais ainda tenho dificuldade.
 - Não consigo manipular tuplas
5. Informe aqui suas dificuldade com as questões de tuplas.
6. Você conseguiu entender o conteúdo de lista e pode afirmar que:
 - Consigo aplicar o que aprendi em qualquer questão.
 - Consegui entender o assunto, mas não consigo colocar em pratica.
 - Não entendi o assunto.
7. Informe aqui quais suas dificuldades na hora de aplicar o conteúdo de listas:
8. Sobre o conteúdo de Matrizes você afirmar que
 - Entendeu o conceito, mas não consegue aplicar.
 - Entendeu o conceito e consegue aplicar em qualquer conteúdo.

- Não entendeu o assunto, está com bastante dificuldade.
9. Descreva aqui quais suas dificuldades com o assunto de matrizes.
10. Você consegue aplicar todo o conteúdo ensinado até o momento?
- Sim, completamente em qualquer questão
- Sim, parcialmente ainda tenho dificuldades de conectar todos os assuntos
- Não consigo utilizar todos os assuntos juntos.
11. Se você pudesse modificar alguma coisa no modo que as aulas estão sendo dadas o que você sugere?

APÊNDICE C – Formulário parte 3

sobre o ensino de Python presencial

1. Com relação ao ensino de dicionários?
 - aprendi parcialmente
 - aprendi parcialmente e tive muitas dificuldades
 - tive dificuldades em aprender, consigo criar um código com dicionário mas o conteúdo ainda é abstrato
 - Não tive dificuldade em aprender
2. Informe aqui suas dificuldades com relação ao ensino de dicionários:
3. Com relação ao uso de dicionários dentro de listas você pode afirmar que:
 - Consigo manipular sem dificuldade.
 - Consigo manipular mais ainda tenho dificuldade.
 - Consigo manipular parcialmente
 - Não consegui aprender a manipular sem olhar algum código anterior.
4. Fale aqui suas dificuldade com relação ao uso de dicionários dentro de listas:
5. Com relação ao uso de funções que são criadas por você, é possível afirmar que:
 - Consigo criar qualquer função e usar em minhas logicas.
 - Consigo criar qualquer função, mas tenho dificuldade de entender quando usar
 - Não consegui aprender o conteúdo de funções
6. Descreva de forma sucinta suas dificuldades em aplicar funções em seus códigos:
7. Sobre o modelo de prova baseado em pequenos projetos aplicado ao final da segunda unidade, você pode afirmar que:
 - Gostou
 - Gostou parcialmente
 - Não gostou
8. Com relação a pergunta seis, se você não gostou ou gostou parcialmente, informe aqui os motivos para essa abordagem possa ser melhorada:

APÊNDICE D – Formulário Final sobre o ensino de Python presencial

1. A abordagem utilizada na terceira unidade como sala invertida facilitou seu desenvolvimento na matéria?
 - Sim, completamente
 - Sim, parcialmente
 - Não ajudou continuei sentindo dificuldades
 - Apenas não ajudou

2. Fale com suas palavras o que achou da abordagem da terceira unidade onde você foi o próprio responsável por criar seu problema e solucioná-lo:

3. Fazer seu projeto em grupo facilitou você aprender a programar?
 - Sim, fez toda a diferença
 - Sim, parcialmente
 - Não, porque eu não tinha dificuldades
 - Não, continuei sentindo dificuldades
 - Não ajudou de forma alguma

4. Descreva quais foram as maiores dificuldades do trabalho em grupo:

5. Ter um acompanhamento do professor pelo o WhatsApp facilitou o seu desenvolvimento na matéria?
 - Sim, completamente
 - Sim, parcialmente
 - Não ajudou em nada

6. Sobre o Ensino de Programação você acha que ter um acompanhamento do professor pelo o WhatsApp e realizar a sala invertida seria o ideal para facilitar o processo de aprendizagem?
 - Sim, completamente acho o ideal
 - Sim, parcialmente acho o ideal
 - Não, prefiro realizar atividades e provas normais

- Não, prefiro ter apenas o acompanhamento pelo o WhatsApp e realizar uma prova normal
 - Apenas não acho o ideal
7. Descreva com suas palavras se possível o quão importante foi ter um acompanhamento de perto do professor pelo o WhatsApp:
8. O que você achou da metodologia de ensino utilizada pelo o professor?
9. Com relação aos problemas mais complexos sobre manipulação de String, fale com suas palavras as dificuldades obtidas?
10. Durante o período letivo você utilizou para programar? (Um ou mais)
- Smartphone
 - Computador
 - Papel, caneta e lápis
11. Com relação ao QPython, se você o utilizou em algum momento ou durante todo o período letivo para programar, você pode afirmar que:
- Fácil de usar e ao mesmo tempo intuitivo
 - Fácil de usar, mas pouco intuitivo
 - Difícil de usar
12. Com relação ao QPython, fale com suas palavras as maiores dificuldades obtidas:
13. Você pode afirmar que conseguiu aprender sobre tudo o que viu na matéria?
- Entre 0 e 25%
 - Entre 25% e 50%
 - Entre 50% e 75%
 - Aprendi todo o assunto
14. Deixe aqui sua opinião sobre o curso, sobre o professor e sobre tudo o que você gostou, não gostou etc?

APÊNDICE E – Formulário parte 1 sobre o ensino de Python EAD

1. Qual(is) tipo(s) de variável(is) você teve mais dificuldade em aprender?
 - int
 - string
 - float
 - boolean
 - Não tive dificuldade em aprender
2. Você teve mais dificuldade em aprender variável explícita ou implícita?
 - Variável explícita. Ex: idade = 25
 - Variável implícita. Ex: idade = int(input('Informe a sua idade'))
 - Não tive dificuldade em aprender
3. Relacionado ao comando print, você teve mais dificuldade em aprender?
 - Format. Ex: print('O cálculo dos números e é:'.format(74, 12))
 - Concatenação . Ex: print('O cálculo dos números'+74+' e '+12+' é:')
 - Não tive dificuldade em aprender
4. Relacionado aos comparadores, você teve mais dificuldade em aprender?
 - == igual
 - != não igual
 - > maior que
 - >= maior ou igual
 - < menor que
 - <= menor ou igual que
 - Não tive dificuldade em aprender
5. Relacionado ao assunto de estrutura de decisão, você teve mais dificuldade em aprender?
 - Marque todas que se aplicam.
 - If

- else
 - elif
 - If e else
 - If | elif | else
 - Não tive dificuldade em aprender
6. Relacionado ao assunto de estrutura de decisão com comparações, você teve mais dificuldade em aprender?
- comparações numéricas
 - comparações textuais
 - Não tive dificuldade em aprender
7. Relacionado ao assunto de estrutura de decisão com mais de uma comparação utilizando operador lógico, você teve mais dificuldade em aprender?
- AND
 - OR
 - NOT
 - Não tive dificuldade em aprender
8. Relacionado ao assunto de manipulação de strings e seus comandos, você teve mais dificuldade em aprender?
- IN. Ex: matéria = 'programação' print(('m' in matéria))
 - NOT IN. Ex: Ex: matéria = 'programação' print(('t' not in matéria))
 - startswith. Ex: estadio = 'maracanã' verdade = estadio.startswith('mara') resultado é igual a True
 - endswith. Ex: estadio = 'maracanã recife' verdade = estadio.endswith('recife') resultado é igual a True
 - Não tive dificuldade em aprender
9. Relacionado ao assunto de manipulação de strings e como acessar, você teve mais dificuldade em aprender?
- Acessar uma letra específica. Ex: matéria = 'programação' letra = matéria[2]
 - Acessar o prefixo de uma palavra. Ex: matéria = 'programação' letra = matéria[:2]
 - Acessar o sufixo de uma palavra. Ex: matéria = 'programação' letra = matéria[2:]

- Acessar parte de uma palavra. Ex: matéria = 'programação' letra = matéria[1:4]
 - Não tive dificuldade em aprender
10. Relacionado ao assunto de manipulação de strings e como alterar, você teve mais dificuldade em aprender?
- Concatenação de string. Ex: linguagem = 'python' texto = 'A linguagem é '+linguagem+' versão 3'
 - Capitalize. Ex: print('programação ',texto_capitalize.capitalize()) o resultado é Programação
 - Separação de String com Split. Ex: futebol = 'Copa do Mundo 2014' futebol = futebol.split(' ') resultado é [0] possui 'copa', na [1] possui 'do', na [2] possui 'Mundo' e na [3] possui '2014'
 - Não tive dificuldade em aprender
 - Outro:
11. Com relação aos problemas mais complexos sobre manipulação de String, fale com suas palavras as dificuldades obtidas?
12. Durante o período letivo você está utilizando para programar? (Um ou mais)
- Smartphone
 - Computador
 - Papel, caneta e lápis
13. Você pode afirmar que conseguiu aprender sobre os assuntos até o momento?
- Entre 0 e 25%
 - Entre 25% e 50%
 - Entre 50% e 75%
 - Aprendi todo o assunto
14. Relate aqui quais as dificuldades você está sentindo, quais exercícios está tendo mais dificuldade?
15. Sobre a metodologia de ensino abordada pelo o professor como está sendo para você, isso inclui todo o material de apoio, grupo no WhatsApp e etc?
- Ruim
 - Regular
 - Boa
 - Ótima

APÊNDICE F – Formulário parte 2 sobre o ensino de Python EAD

1. Qual laço de repetição você está tendo dificuldade?
 - for elemento in lista:
 - for i in range(qtd_elementos)
 - while true:
 - Não tive dificuldade em aprender
2. Você teve ou tem dificuldade com um laço de repetição dentro de outro?
 - consegui executar parcialmente
 - Consegui executar mais ainda tenho dificuldade
 - Não tive ou não tenho dificuldade
3. Informe aqui suas dificuldade com laços de repetição.
4. Sobre o conteúdo de tuplas, você consegue afirmar que:
 - Consigo manipular tuplas sem dificuldade.
 - Consigo manipular tuplas mais ainda tenho dificuldade.
 - Não consigo manipular tuplas
5. Informe aqui suas dificuldade com as questões de tuplas.
6. Você conseguiu entender o conteúdo de lista e pode afirmar que:
 - Consigo aplicar o que aprendi em qualquer questão.
 - Consegui entender o assunto, mas não consigo colocar em pratica.
 - Não entendi o assunto.
7. Informe aqui quais suas dificuldades na hora de aplicar o conteúdo de listas:
8. Sobre o conteúdo de string, você consegue afirmar que:
 - Consigo manipular string sem dificuldade.
 - Consigo manipular string mais ainda tenho dificuldade

- Não entendi o assunto, ainda estou com bastante dificuldade.
9. Você conseguiu ver a relação que uma string tem com uma lista? *
- Sim
- Não
10. Informe aqui suas dificuldade com as questões de string.
11. Você consegue aplicar todo o conteúdo ensinado até o momento?
- Sim, completamente em qualquer questão
- Sim, parcialmente ainda tenho dificuldades de conectar todos os assuntos
- Não consigo utilizar todos os assuntos juntos.
12. Se você pudesse modificar alguma coisa no modo que as aulas estão sendo dadas o que você sugere?

APÊNDICE G – Formulário parte 3

sobre o ensino de Python EAD

1. Com relação ao ensino de dicionários?
 - aprendi parcialmente
 - aprendi parcialmente e tive muitas dificuldades
 - tive dificuldades em aprender, consigo criar um código com dicionário mas o conteúdo ainda é abstrato
 - Não tive dificuldade em aprender
2. Informe aqui suas dificuldades com relação ao ensino de dicionários:
3. Com relação ao uso de dicionários dentro de listas você pode afirmar que:
 - Consigo manipular sem dificuldade.
 - Consigo manipular mais ainda tenho dificuldade.
 - Consigo manipular parcialmente
 - Não consegui aprender a manipular sem olhar algum código anterior.
4. Fale aqui suas dificuldade com relação ao uso de dicionários dentro de listas:
5. Sobre o modelo de prova baseado em pequenos projetos aplicado ao final da primeira unidade, você pode afirmar que:
 - Gostou
 - Gostou parcialmente
 - Não gostou
6. Com relação a pergunta cinco, se você não gostou ou gostou parcialmente, informe aqui os motivos para essa abordagem possa ser melhorada:
7. Com relação ao uso de funções que são criadas por você, é possível afirmar que:
 - Consigo criar qualquer função e usar em minhas logicas.
 - Consigo criar qualquer função, mas tenho dificuldade de entender quando usar
 - Não consegui aprender o conteúdo de funções
8. Descreva de forma sucinta suas dificuldades em aplicar funções em seus códigos:

9. Sobre o conteúdo de Matrizes você afirmar que:
- Entendeu o conceito, mas não consegue aplicar.
 - Entendeu o conceito e consegue aplicar em qualquer conteúdo.
 - Não entendeu o assunto, está com bastante dificuldade.
10. Descreva aqui quais suas dificuldades com o assunto de matrizes:
11. Você consegue aplicar todo o conteúdo ensinado até o momento?
- Sim, completamente em qualquer questão
 - Sim, parcialmente ainda tenho dificuldades de conectar todos os assuntos
 - Não consigo utilizar todos os assuntos juntos.
12. Se você pudesse modificar alguma coisa no modo que as aulas estão sendo dadas o que você sugere?

APÊNDICE H – Formulário Final sobre o ensino de Python EAD

1. Sobre o modelo de prova baseado em pequenos projetos aplicado ao final da primeira unidade, você pode afirmar que:
 - Gostou
 - Gostou parcialmente
 - Não gostou
2. Com relação a pergunta cinco, se você não gostou ou gostou parcialmente, informe aqui os motivos para essa abordagem possa ser melhorada:
3. A abordagem utilizada na segunda unidade como sala invertida facilitou seu desenvolvimento na matéria?
 - Sim, completamente
 - Sim, parcialmente
 - Não ajudou continuei sentindo dificuldades
 - Apenas não ajudou
4. Fale com suas palavras o que achou da abordagem da segunda unidade onde você foi o próprio responsável por criar seu problema e solucioná-lo:
5. Fazer seu projeto em grupo facilitou você aprender a programar?
 - Sim, fez toda a diferença
 - Sim, parcialmente
 - Não, porque eu não tinha dificuldades
 - Não, continuei sentindo dificuldades
 - Não ajudou de forma alguma
6. Descreva quais foram as maiores dificuldades do trabalho em grupo:
7. Ter um acompanhamento do professor pelo o whatsapp facilitou o seu desenvolvimento na matéria?
 - Sim, completamente

- Sim, parcialmente
 - Não ajudou em nada
8. Sobre o EAD você acha que ter um acompanhamento do professor pelo o whatsapp e realizar a sala invertida seria o ideal para aprender a programar sem está em sala de aula?
- Sim, completamente acho o ideal
 - Sim, parcialmente acho o ideal
 - Não, prefiro realizar atividades e provas normais
 - Não, prefiro ter apenas o acompanhamento pelo o whatsapp e realizar uma prova normal
 - Apenas não acho o ideal
9. Descreva com suas palavras se possível o quão importante foi ter um acompanhamento de perto do professor pelo o whatsapp:
10. O que você achou da metodologia de ensino utilizada pelo o professor?
11. Durante o período letivo você utilizou para programar? (Um ou mais)
- Smartphone
 - Computador
 - Papel, caneta e lápis
12. Você pode afirmar que conseguiu aprender sobre tudo o que viu na matéria?
- Entre 0 e 25%
 - Entre 25% e 50%
 - Entre 50% e 75%
 - Aprendi todo o assunto
13. Com relação a programar com e sem o The Huxley?
- O The Huxley dificultou eu me desenvolver na matéria
 - O The Huxley dificultou eu me desenvolver na matéria e inibiu minha criatividade
 - O The Huxley facilitou eu me desenvolver na matéria
 - O The Huxley facilitou eu me desenvolver na matéria e potencializou minha criatividade
 - Prefiro não utilizar o The Huxley pois inibi minha capacidade de desenvolver um código

14. Com relação a programar usando o Replit?

- Gostei, simples e intuitivo
- Gostei parcialmente, a pouco intuitiva
- Não gostei, a plataforma não simples nem intuitiva

15. Com relação a programar usando o Replit, relate aqui suas maiores dificuldades ao usar a plataforma:

16. Deixe aqui sua opinião sobre o curso, sobre o professor e sobre tudo o que você gostou, não gostou etc?

APÊNDICE I – Projetos colaborativos no ensino presencial

I.1 Grupo A: projeto rota transporte

O projeto Rota Transporte, foi o primeiro trabalhado em equipe para que os alunos pudessem entender como deveriam desenvolver seus projetos colaborativos. Assim, foi passado para eles um enunciado que foi trabalhado durante duas aulas todo o escopo, desde a decomposição do problema até o desenvolvimento da solução. Como já era de costume a troca de informações pelo WhatsApp, durante esses dois dias os alunos foram instigados à troca de figurinhas dentro dos grupos para chegar na melhor solução possível.

A seguir é apresentado o enunciado da questão e uma foto de uma das soluções propostas por um dos grupos utilizando o QPython 3.

Enunciado do Problema:

A empresa Rota Transportes contratou a sua equipe para desenvolver um programa capaz de vender passagens interurbanas de Aracaju a Salvador, Salvador a Aracaju, ambas ida e volta. Os valores das passagens custam R\$ 78,90 a ida e R\$ 58,90 a volta, para ambas as cidades. Cada pessoa pode comprar mais de uma passagem desde que esteja com a documentação de cada passageiro. O sistema para de pedir os dados quando o usuário pressiona a tecla *Esc*. Ao encerrar a inserção de dados, informa o valor total a ser pago. Se o pagamento for em dinheiro, o sistema deve perguntar quanto será entregue e informar o valor do troco. Se o pagamento for em cartão de débito, apenas realizar a transação. Se for em cartão de crédito, informe em até quantas parcelas o valor pode ser parcelado. Após a escolha da quantidade de parcelas, deve-se incrementar 5% de juros ao mês e mostrar ao usuário o valor total a ser pago. Após finalizar a transação, imprima o recibo contendo todas as informações da viagem de cada passagem.

Na figura 57, é apresentado prints de um dos códigos feitos pelo o grupo A no aplicativo QPython usando o smartphones, referente ao projeto Rota Transporte.



Figura 57 – Código de um dos alunos no QPython

I.2 Grupo B: projeto mercearia Itabaiana

A mercearia Itabaiana necessita automatizar o processo de cadastro e vendas de produtos. Assim, faz-se necessário criar um programa que receba uma lista de produtos e seus respectivos códigos e valores. Cada produto deve ser salvo em um dicionário e depois ser armazenado em uma lista. Lembre-se que se houver algum erro de digitação em alguma das informações é necessário repetir a pergunta até que a pessoa tenha digitado corretamente. Por exemplo, se no lugar de código a pessoa digitar um nome, esse produto é inválido e deve ser preenchido novamente.

Parâmetros de entrada n que identifica quantos produtos vão ser cadastrados e as informações dos produtos: cod_produto, nome_produto, valor como pode ser visto na Tabela 9 que mostra como deve ser a saída formatada.

Tabela 9 – Saída formatada

CÓDIGO	NOME PRODUTO	VALOR
1	ARROZ	R\$ 4.8
2	FEIJÃO	R\$ 8.75
3	GOIABA	R\$ 2.25
	TOTAL	R\$ 0.00

Acervo pessoal

É preciso ter uma função responsável por calcular o valor total da compra toda a impressão deve sair formatada da seguinte forma, como é visto na Tabela 10.

Tabela 10 – Saída formatada com calculo do total da compra

CÓDIGO	NOME PRODUTO	VALOR
1	ARROZ	R\$ 4.8
2	FEIJÃO	R\$ 8.75
3	GOIABA	R\$ 2.25
	TOTAL	R\$ 15.80

Acervo pessoal

É preciso ter também uma função responsável por informar o produto mais caro, e imprima da seguinte forma, como é visto na Tabela 11.

Tabela 11 – O produto mais caro é:

CÓDIGO	NOME PRODUTO	VALOR
2	FEIJÃO	R\$ 8.75

Acervo pessoal

Por fim, uma função responsável por pesquisar um determinado produto, e imprima da seguinte forma, como é visto na Tabela 12.

Tabela 12 – O produto pesquisado é:

CÓDIGO	NOME PRODUTO	VALOR
3	GOIABA	R\$ 2.25

Acervo pessoal

I.3 Grupo C: projeto campeonato sergipano

Você foi contratado para trabalhar no sistema do campeonato brasileiro de futebol, assim você deve cadastrar 5 equipes e algumas informações iniciais. Assim, faz-se necessário que você crie um programa que receba os 5 times e suas informações, cada um deve ser salvo em um dicionário e, após, ser armazenado em uma MATRIZ 5x5. Lembre-se que, se houver algum erro de digitação em alguma das informações, é necessário repetir a pergunta até que a pessoa tenha digitado corretamente, por exemplo, se no lugar de pontos a pessoa digitar um nome, essa pontuação é inválida e deve ser preenchida novamente.

Parâmetros de Entrada: nome_time, vitorias, empates, derrotas, pontuacao e ter uma função de impressão que retorne os valores da seguinte forma, conforme é visto na Tabela 13.

Tabela 13 – Times cadastrados

TIME	VITORIAS	EMPATES	DERROTAS	PONTUACAO
SERGIPE	0	0	0	0
CONFIANÇA	0	0	0	0
BOCA JUNIORS	0	0	0	0
RIVER PLATER	0	0	0	0
SOCORRENSE	0	0	0	0

Acervo pessoal

Criar uma função responsável por atualizar as pontuações de cada time de uma só vez, vitórias, empates e derrotas e que calcule as pontuações, depois chame a função de impressão e mostre as informações atualizadas, como é visto na Tabela 14.

Tabela 14 – Tabela de classificação atualizada

TIME	VITORIAS	EMPATES	DERROTAS	PONTUACAO
SERGIPE	4	2	1	14
CONFIANÇA	3	2	2	11
BOCA JUNIORS	2	2	3	8
RIVER PLATER	1	4	2	7
SOCORRENSE	0	3	4	3

Acervo pessoal

Observação: cada vitória vale 3 pontos, o empate 1 e a derrota 0

criar uma função responsável por informar qual o time foi o campeão, como é visto na Tabela 15.

Tabela 15 – O campeão é:

Saída formatada
O maior de Sergipe é o SERGIPE e se sagrou campeão com 4 vitórias, 2 empates e 1 derrota, conquistando 14 pontos. Congratulations, Sergipe

Acervo pessoal

I.4 Grupo D: projeto academia bem estar

Você foi contratado para trabalhar no sistema de uma academia. Assim, faz-se necessário que você crie um programa que receba “n” alunos e suas informações pessoais, cada um deve ser salvo em um dicionário e, posteriormente, ser armazenado em uma lista. O sistema para de cadastrar quando, no nome, for informado a palavra finalizar.

Parâmetros de Entrada: nome_aluno, cpf, altura, peso, imc, resultado. Crie uma função de impressão que retorne aos valores da seguinte forma, como pode ser visto na Tabela 16.

Tabela 16 – Formato de saída

NOME	CPF	ALTURA	PESO	IMC	RESULTADO
JOANA	181	1.8	94.2	29.1	INDISPONÍVEL
THIAGO	224	1.9	79.9	22.1	INDISPONÍVEL

Acervo pessoal

Crie uma função responsável por atualizar os resultados dos IMC de cada aluno, a seguir chame a função de impressão e mostre as informações atualizadas, de acordo com a Tabela 17.

Tabela 17 – Atualizar imc

NOME	CPF	ALTURA	PESO	IMC	RESULTADO
JOANA	181	1.8	94.2	29.1	SOBRE PESO
THIAGO	224	1.9	79.9	22.1	PESO NORMAL

Acervo pessoal

FORMULA IMC = PESO / ALTURA X ALTURA

Tabela 18 – Use a seguinte tabela como base para calcular o resultado

IMC	SITUAÇÃO
< 16	SUBPESO SEVERO
16 a 19,9	SUBPESO
20 a 24,9	NORMAL
25 a 29,9	SOBREPESO
30 a 39,9	OBESO
> 40	OBESO MÓRBIDO

Acervo pessoal

Crie uma função responsável por informar qual a pessoa que possui o maior IMC e imprima da seguinte forma, como é visto na Tabela 19.

Tabela 19 – Maior imc

Saída formatada
Olá JOANA, SEU IMC é 29.1. Isso significa que você está com SOBRE PESO.

Acervo pessoal

1.5 Grupo E: projeto pizzaria paulistinha

Você foi contratado por uma pizzaria para automatizar o processo de cadastros de alimentos. Assim, faz-se necessário que você crie um programa que receba uma lista de itens e suas respectivas quantidades disponíveis, cada item deve ser salvo em um dicionário e após ser armazenado em uma lista. Todos esses itens fazem com que seja possível criar várias pizzas diferentes, como: pizza de calabresa, portuguesa e quatro queijos.

Parâmetros de entrada “n” que identifica quantos itens serão cadastrados e as informações são: nome_item, quantidade, como pode ser observado na Tabela 20.

Tabela 20 – Formato de saída

NOME ITEM	QUANTIDADE
TOMATE	10
QUEIJO	30
CALABRESA	10
OVOS	10
MASSA	10

Acervo pessoal

Crie uma função responsável por solicitar uma quantidade de itens para a criação de uma pizza. Para realizar o teste dessa função, use como exemplo a pizza de calabresa e a de portuguesa, uma após a outra.

Parâmetros de entrada, nome_item e quantidade. Caso o item solicitado não tenha em estoque, informe a mensagem: “Item indisponível”, caso a quantidade informada não esteja disponível informe a seguinte mensagem: “Quantidade solicitada indisponível”.

Para poder ser criada a pizza de calabresa é necessário: cinco tomates, cinco calabresas, quinze queijos, cinco cebolas e cinco massas.

Já para criar a pizza portuguesa, é necessário: cinco tomates, cinco queijos, dez ovos, cinco cebolas e cinco massas.

Quanto à criação da pizza de quatro queijos, é necessário dez queijos e dez massas.

A cada pizza criada deve-se ser subtraído da lista de item as quantidades solicitadas, no exemplo acima a pizza de quatro queijos não poderá ser criada porque não haverá disponibilidade de massas.

A cada pizza criada deve ser informada a seguinte mensagem: “Pizza criada com sucesso!”

Para encerrar a chamada dessa função, deve-se informar a quantidade do item como 0 ou informado no nome do item como encerrar, ou ainda quando não houver mais item disponível para ser criado uma pizza. Por fim, imprima todo o estoque disponível como na primeira questão. Saída formatada, como é visto na Tabela 21.

Tabela 21 – Formato de saída

NOME ITEM	QUANTIDADE
TOMATE	0
QUEIJO	10
CALABRESA	5
OVOS	0
MASSA	0

Acervo pessoal

I.6 Grupo F: projeto Formula 1

Você foi contratado para trabalhar no sistema da formula 1, sua missão é desenvolver um programa capaz de gerenciar todos os pilotos e seus respectivos carros e equipes. Ao final, identificar todas as posições de cada piloto ao final da corrida.

Parâmetros de Entrada: posicao, nome_piloto, equipes, numero_carro, a saída da impressão deve ser da seguinte forma, como pode ser visto na Tabela 22.

Tabela 22 – Formato de saída

POSIÇÃO	PILOTO	EQUIPE	NÚMERO CARRO
1	LEWIS HAMILTON	MERCEDES	1
2	MAX VERSTAPPEN	RBR-HONDA	2
3	LANDO NORRIS	MCLAREN-MERCEDES	1
4	LANCE STROLL	ASTON MARTIN-MERCEDES	2
5	FERNANDO ALONSO	ALPINE-RENAULT	1
6	CHARLES LECLERC	FERRARI	2

Acervo pessoal

Observação: o exemplo acima é o grid de largada de cada piloto, por isso é informado as posições de cada um no início também.

Crie uma função responsável por atualizar as posições de cada piloto. Para tanto, você precisa informar o nome do piloto e a posição em que ele ficou ao final da corrida.

Parâmetros de Entrada: posicao, nome_piloto, vide o exemplo a baixo depois de atualizar, vide a Tabela 23.

Tabela 23 – Formato de saída

POSIÇÃO	PILOTO	EQUIPE	NÚMERO CARRO
1	LEWIS HAMILTON	MERCEDES	1
2	MAX VERSTAPPEN	RBR-HONDA	2
3	LANDO NORRIS	MCLAREN-MERCEDES	1
4	LANCE STROLL	ASTON MARTIN-MERCEDES	2
5	FERNANDO ALONSO	ALPINE-RENAULT	1
6	CHARLES LECLERC	FERRARI	2

Acervo pessoal

I.7 Grupo H: projeto aluguel de apartamentos

Você foi contratado para trabalhar no sistema de um condomínio. Sua missão é criar um programa capaz de cadastrar todos os apartamentos e seus respectivos blocos – e se tem ou não morador, aqueles apartamentos em que estiverem sem moradores deve-se informar: disponível para aluguel, vide o exemplo abaixo:

Parâmetros de Entrada: numero_ap, bloco, existe_morador. Crie uma função de impressão que retorne os valores da seguinte forma, como pode ser visto na Tabela 24.

Tabela 24 – Formato de saída

APARTAMENTO	BLOCO	MORAM PESSOAS
602	ALFREDO MALETE	SIM
100	TAIERA	SIM
401	REISADO	DISPONÍVEL PARA ALUGUEL
575	CACUMBI	SIM

Acervo pessoal

Crie uma função responsável por alugar um apartamento. Para tanto, deve-se pesquisar pelo número do apartamento e pelo o bloco, bem como verificar se está disponível e, se estiver, deve-se atualizar a existe_morador para sim. Contudo, caso a variável existe_morador já possua moradores, deve-se informar a mensagem: “Apartamento indisponível para locação”. Por fim, chame a função de impressão: numero_ap, bloco, vide a Tabela 25 que mostra como deve sair o resultado.

Tabela 25 – Formato de saída

APARTAMENTO	BLOCO	MORAM PESSOAS
602	ALFREDO MALETE	SIM
100	TAIERA	SIM
401	REISADO	SIM
575	CACUMBI	SIM

Acervo pessoal

APÊNDICE J – Projetos colaborativos no EAD

J.1 Grupo 1: Projetos colaborativos modelo 1

O grupo um escolheu o modelo de projeto um, em que eles puderam desenvolver três ideias diferentes e conseguiram trabalhar em equipe, ao construir códigos distintos e terem que ajudar uns aos outros.

Relatório para definição do problema

Este projeto foi construído com base no modelo 1, ofertado pelo docente. Portanto, cada integrante ficou responsável pela elaboração de sua problemática, o mesmo foi dividido em três questões, sendo Aluno um responsável pela questão 1, Aluna dois pela questão 2 e Aluna três pela questão 3. Dessa forma, cada componente ficou encarregado de elaborar um código utilizando listas compostas para salvar várias informações, dicionários para salvar uma única informação, listas para salvar os dicionários, blocos de condições, laços de repetição e as seguintes funções próprias: função cadastrar, função pesquisar, função imprimir, função remover e a função atualizar. Além disso, foram criadas funções adicionais necessárias para a solução de cada problema atendendo suas respectivas particularidades.

Durante duas semanas, realizamos reuniões pelo Google Meet, tiramos dúvidas com o professor pelo WhatsApp e marcamos uma reunião com o professor no sábado, dia 24 de junho, para resolvermos alguns problemas dos nossos códigos. Durante o desenvolvimento dos problemas, ajudamos uns aos outros na criação do enunciado, como também na hora do desenvolvimento dos códigos.

J.1.1 Treino crossFit

Um treino de crossFit é dividido em três partes: Aquecimento (Preparo do corpo), Técnica (Como executar os exercícios) e Wod (Workout of the Day, “Missão do dia”). Sabendo disso, um Coach (treinador) deseja criar um programa que possa cadastrar e acompanhar os exercícios de seus alunos, com o intuito de ratificar o treino de cada atleta. Este programa deve conter as opções de cadastrar, pesquisar, imprimir, remover e atualizar qual exercício, as repetições ou a parte que se encaixam, como pode ser visto na Tabela 26:

Tabela 26 – Exercício do dia

PARTE	EXERCÍCIO	REPETIÇÕES
Aquecimento	Polichinelos	20
Aquecimento	Deslocamentos	10
Aquecimento	Lunges	12
Técnica	Clean	2
Técnica	Hang Clean	1
Técnica	Jerk	3
Wod	Single Unders	200
Wod	Deadlift	8
Wod	Toes To Bar	15

Acervo pessoal

Na figura 58 temos apresentação do Aluno um do grupo um, referente a parte prática com a solução proposta.

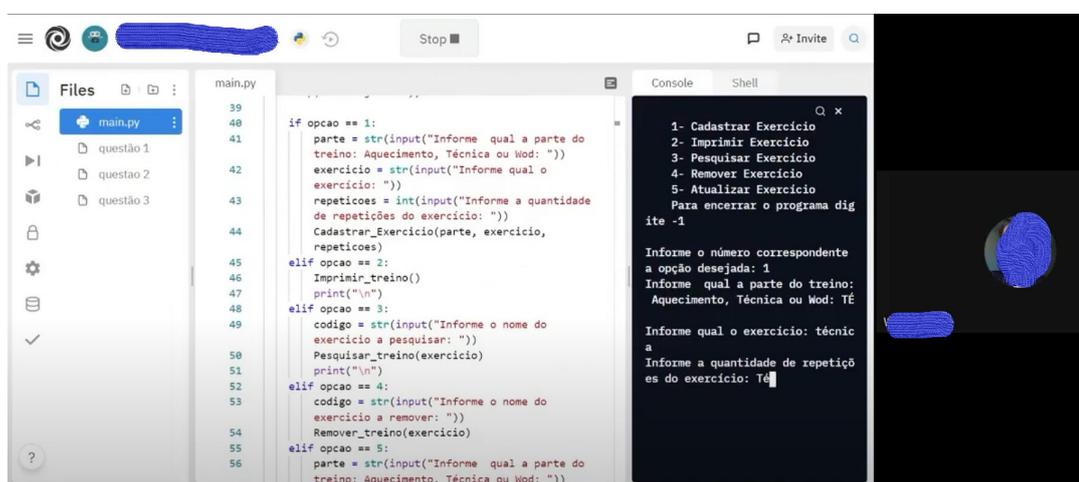


Figura 58 – Apresentação projeto crossfit

J.1.2 Disciplinas

Problema: Um aluno pretende se matricular em disciplinas sem pré-requisitos, mas, para isso, ele precisa saber quais são essas disciplinas. Ou seja, para que ele não perca tempo procurando uma a uma em meio a imensa quantidade de disciplinas do seu curso, ajude-o criando um programa que mostre a ele somente tais disciplinas.

Objetivo: Criar um programa que cadastre disciplinas, seus respectivos códigos e pré-requisitos (se possuir) e informar ao estudante se há alguma disponível sem pré-requisito. Caso exista, apresente uma mensagem na tela informando que ele poderá ser matriculado nessa(s) disciplina(s).

Entrada: nome_disciplina, código, pre_requisito, da seguinte forma, como pode ser visto na Tabela 27:

Tabela 27 – Entrada:

CÓDIGO	DISCIPLINA	PRÉ-REQUISITO
C001	Cálculo 1	Não possui
F001	Física 1	Cálculo 1 e Vetores
Q001	Química geral I	Não possui
QO22	Química orgânica	Química geral I

Saída: “Existe disciplina sem pré-requisito” ou “Você poderá se matricular em: Cálculo 1 e Química geral I”.

Na figura 59 temos apresentação da Aluna dois do grupo um, referente a parte prática com a solução proposta.

```

55 print("\n")
56 elif opcao == 3:
57     codigo = str(input("Informe o código da
58     disciplina a pesquisar: "))
59     Pesquisar_Disciplina(codigo)
60     print("\n")
61 elif opcao == 4:
62     codigo = str(input("Informe o código da
63     disciplina a remover: "))
64     Remover_Disciplina(codigo)
65 elif opcao == 5:
66     codigo = str(input("Informe o código da
67     disciplina: "))
68     nome = str(input("Informe o nome da disciplina:
69     "))
70     preRequisitos = str(input("Informe os pré
71     requisitos (Ex: D001, D002. Caso não possua,
72     deixe em branco: "))
73     Atualizar_Disciplina(codigo, nome, preRequisitos)
74 elif opcao == 6:
75     Mostrar_Disciplinas_Sem_Prerequisitos()
76 else:
77     print("Escolha uma opção válida, ou digite -1
78     para encerrar\n")
79

```

```

Disciplina removida!
1- Cadastrar Disciplina
2- Imprimir Disciplinas
3- Pesquisar Disciplina
4- Remover Disciplina
5- Atualizar Disciplina
6- Verificar se há disciplinas sem p
ré-requisitos
Escolha uma das opções (Caso queira encerr
rar o programa digite -1): 3
Informe o código da disciplina a pesquis
ar: 12
1- Cadastrar Disciplina
2- Imprimir Disciplinas
3- Pesquisar Disciplina
4- Remover Disciplina
5- Atualizar Disciplina
6- Verificar se há disciplinas sem p
ré-requisitos
Escolha uma das opções (Caso queira encerr

```

Figura 59 – Apresentação projeto disciplinas

J.1.3 GetTickets

A empresa GetTickets precisa que você crie um programa para automatizar o processo de vendas de ingressos. Será necessário criar uma matriz que armazene os códigos, as quantidades e os valores. Assim, utilizando as seguintes funções:

1. Cadastrar_usuários(), para receber e armazenar as informações dos usuários.
2. Remover_usuario(usuário), para remover o usuário que deseja comprar uma quantidade de ingressos acima do limite.
3. Atualizar_usuario(), adicionar um novo usuário para substituir o usuário removido.
4. Valor_total(), para calcular o valor total, da compra de cada usuário.
5. Verificar_limite(), para verificar a quantidade de ingressos disponíveis.

6. Imprimir_Resultado(), para imprimir os resultados de forma padronizada, como o exemplo.
7. Pesquisar_usuario(), para pesquisar o usuário que comprou ingressos indisponíveis. Sendo que, o limite é de 100 ingressos.

Tabela 28 – Entrada:

CÓDIGO	NOME DO ARTISTA	QUANTIDADE	VALOR
1	BOY PABLO	20	R\$ 100,00
2	AURORA	10	R\$ 250,00
3	CLAIRO	5	R\$ 250,00
4	JOJI	50	R\$ 350,00

Tabela 29 – Saída

CÓDIGO	NOME DO ARTISTA	VALOR TOTAL
1	BOY PABLO	R\$ 2.000,00
2	AURORA	R\$ 2.500,00
3	CLAIRO	R\$ 1.250,00
4	JOJI	R\$ 17.500,00

Na figura 60 temos apresentação da Aluna três do grupo um, referente a parte prática com a solução proposta.

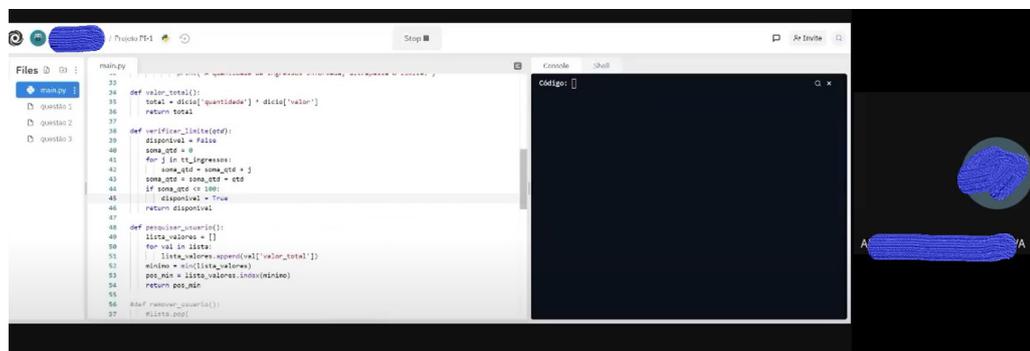


Figura 60 – Apresentação projeto geticketes

J.2 Grupo 2: Projetos colaborativos modelo 1

O grupo 2 também escolheu o modelo de projeto um, em que eles puderam desenvolver três ideias diferentes e conseguiram trabalhar em equipe, ao construírem códigos distintos e terem que ajudar uns aos outros.

Relatório para definição do problema

Durante a definição de quais problemas iríamos desenvolver, optamos no primeiro momento em realizar um debate apenas no grupo do whatsapp, pedindo auxílio ao professor com relação a saber se o problema estava ou não de acordo com o que era esperado. Assim,

realizamos reuniões virtuais por meio do Google Meet apenas para ajudarmos uns aos outros em problemas no código fonte.

Desta forma, desenvolvemos três problemas que podem ser vistos a seguir.

J.2.1 Aulas particulares

João é um rapaz muito ocupado e, ao mesmo tempo, muito metódico. Porém, por dar aulas particulares, seu horário semanal muda constantemente, sendo difícil de lembrar de tudo. Por isso, João solicitou para que você elabore um código que possibilite cadastrar, remover, alterar, imprimir seu horário ou pesquisar nele por algum horário, para checar se já está ocupado. Será que Pedro poderá ajudá-lo?

Requisitos técnicos:

Construir uma matriz 19x7, utilizando dicionários para cadastrar as atividades, separadas de hora em hora, e conter as funções cadastrar, remover, alterar (altera um horário escolhido pelo usuário), imprimir (o horário semanal completo) e pesquisar (pesquisar por um horário específico para checar as atividades). O horário deve ser impresso em formato de tabela, mostrando o horário de cada atividade na primeira coluna e o nome de cada dia na primeira linha da tabela. Além disso, o código deve fornecer autonomia ao usuário para decidir o que fazer em seguida.

- O cadastro será feito quando o usuário acessar o aplicativo, no qual ele irá inserir informações, de hora em hora, em cada dia da semana, de segunda à domingo, das 5 às 23:59. Caso o usuário não tenha nenhuma atividade naquele horário, ele deve ser orientado a inserir um (-) para marcar o horário vazio.
- A função remover, quando solicitada, substituirá a informação no dia e hora solicitados e fornecidos pelo usuário por (-).
- A função alterar, quando solicitada, substituirá a informação no dia e hora solicitados e fornecidos pelo usuário por outra informação também solicitada ao usuário.
- A função pesquisar imprime a atividade naquele horário solicitado ao usuário. Caso esteja vazia, será impresso 'Horário vago'.
- A função imprimir deve imprimir todo o horário semanal em formato de tabela, com o nome de cada dia na primeira linha e os horários na primeira coluna, seguindo uma formatação específica ("====" entre cada item da tabela).

Como teste, João cadastrou todos os dias completamente, no entanto, percebeu que preencheu de maneira errada no domingo a partir de uma pesquisa que ele executou para o horário das 15:00 e tentou remover essa atividade para imprimir o horário, ele conseguiu?

Na figura 61 temos apresentação do Aluno um do grupo 2, referente a parte prática com a solução proposta.

```

main.py
70 print('Horas ---- Segunda ---- Terça ---- Quarta ---- Sexta ---- Sábado ---- Domingo')
71 for i in range(10):
72     print(' ' + str(horas[i]) + ' ---- ' + str(segunda[i]) + ' ---- ' + str(terca[i]) + ' ---- ' + str(quarta[i]) +
73         ' ---- ' + str(quinta[i]) + ' ---- ' + str(sexta[i]) + ' ---- ' + str(sabado[i]) + ' ---- ' + str(domingo[i]) + ')')
74
75
76 #interface
77 print('bem vindo ao seu organizador semanal vamos começar com o cadastro das atividades!o')
78 cronograma = cadastrar()
79
80 while True:
81    acao = int(input('o que deseja fazer? Para remover alguma atividade, digite 1; para alterar alguma
82     atividade, digite 2; para pesquisar por algum horário, digite 3; para imprimir seu horário, digite
83     4; para finalizar a operação, digite 0'))
84     if acao == 1:
85         print(remover(cronograma))
86     elif acao == 2:
87         print(atualizar(cronograma))
88     elif acao == 3:
89         print(pesquisar(cronograma))
90     elif acao == 4:
91         print(imprimir(cronograma))
92     else:
93         print('até logo!')
94         break

```

```

Console
14 às 14:59:
-
15 às 15:59:
-
16 às 16:59:
-
17 às 17:59:
-
18 às 18:59:
-
19 às 19:59:
-
20 às 20:59:
-
21 às 21:59:
-
22 às 22:59:
-
23 às 23:59:
-
Dia adicionado com sucesso!
Quinta:
5 às 5:59:

```

Figura 61 – Apresentação projeto aulas particulares

J.2.2 Delta construtora

Adriano, funcionário da Delta construtora, é responsável pelo estoque de materiais da empresa. Ele atualiza, semanalmente, esse controle manual em planilhas. Devido à demanda dos serviços terem aumentado, ele percebeu a necessidade de pensar em um método menos cansativo, mais rápido, prático e preciso. Para resolver este problema, ele decidiu pedir ajuda ao seu amigo programador Fabiano. Este propôs que criasse um programa no qual ele poderia armazenar os materiais por código.

Os requisitos técnicos básicos que serão utilizados: listas, dicionários, blocos de condições, laço de repetição e criação de funções. Construir uma matriz utilizando dicionários para guardar os materiais cadastrados. Para cadastrar cada produto será necessário: o código, o nome, a quantidade em estoque e a marca. Os códigos e as quantidades devem ser usados inteiramente e para os nomes e as marcas devem ser utilizados string. Caso haja erro de informações digitadas, deve informar uma mensagem falando em qual variável está o erro e pedir para digitar novamente.

Criar a função `cadastrear_material()`, a função `pesquisar()`, a função `atualizar()`, função `remover()` e a função `imprimir_estoque()`. A função `pesquisar()` irá pesquisar os materiais por código e deverá mostrar ao final: a quantidade, o nome e a marca. Ademais, a cada material cadastrado deverá aparecer a mensagem “cadastrado com sucesso!”. Para saber quantos itens há no estoque, será necessário a criação da função `soma_itens()`

Na figura 62 temos apresentação da Aluna dois do grupo 2, referente a parte prática com a solução proposta.

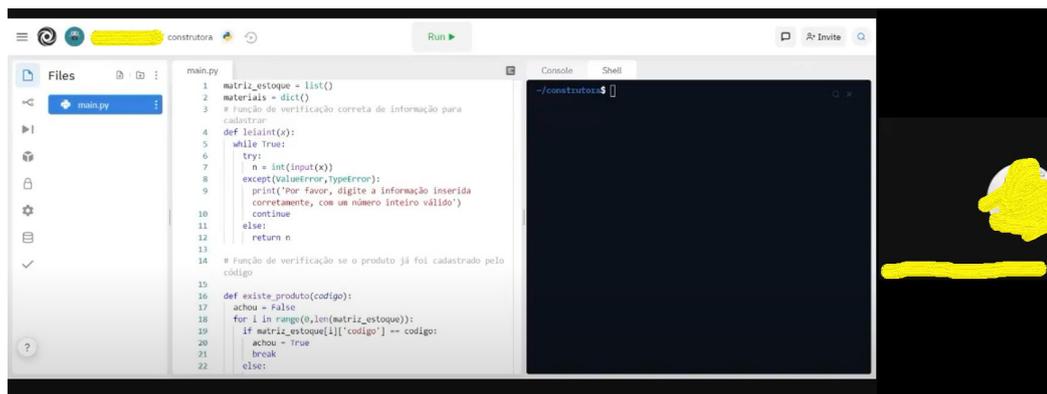


Figura 62 – Apresentação projeto aulas particulares

J.2.3 Moto boy

Wally é motoboy que trabalha no iFood. Nas suas viagens de trabalho, utiliza uma moto para fazer as entregas. Porém, ele está sem controle dos seus gastos com gasolina. Para resolver isso, desenvolva um programa que o ajude no controle dos seus gastos no qual ele adicionaria o dia do abastecimento e o valor. Já quanto ao fim da semana, o quanto ele tem abastecido. Funções: seu programa deverá conter uma função adicionar_dia função que aciona um dia da semana; já para ser atualizada com os devidos abastecimentos diários, uma função remover_dia e remover um dia da semana, caso seja adicionado incorretamente uma função; atualizar_dia função, que atualiza o abastecimento do motoboy; uma função resultado, que soma todos os abastecimentos da semana e imprima na tela para o usuário.

Na figura 63 temos apresentação do Aluno três do grupo 2, referente a parte prática com a solução proposta.

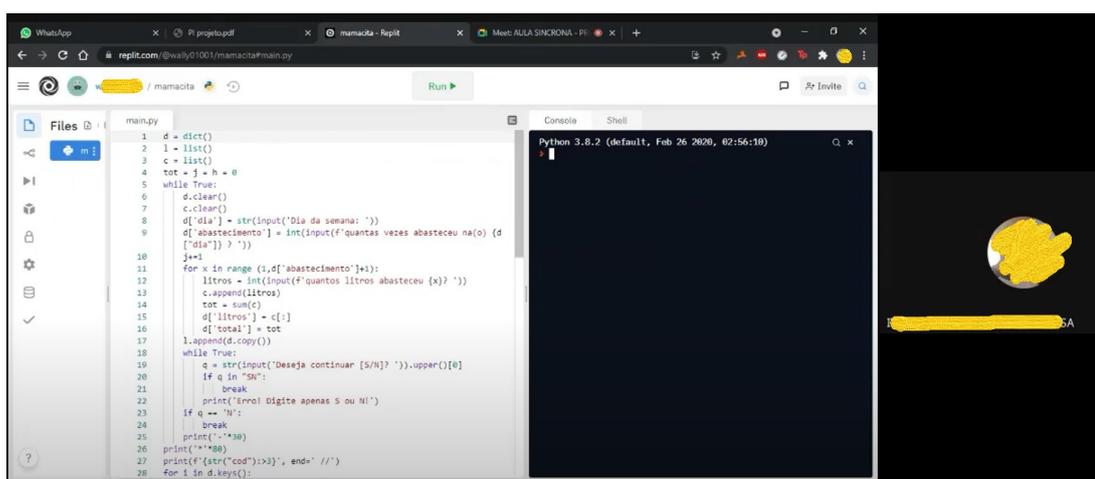


Figura 63 – Apresentação projeto moto boy

J.3 Grupo 3: Projetos colaborativos modelo 2

J.3.1 Estoque de produtos - Empresa SinteQui

Relatório para definição do problema

Inicialmente, o grupo optou pelo modelo 2 do projeto. Assim sendo, a Aluna um pensou em fazer algo relacionado à química, pois todos os integrantes do grupo cursam Química Industrial. Feito isso, Aluna dois deu a ideia de fazermos um programa que realizasse o estoque de produtos químicos, por ser algo que vemos com frequência no cotidiano da indústria. Desse modo, o grupo chegou ao consenso de que seria realizado o estoque de produtos de acordo com o pH e Aluna um definiu que a capacidade máxima do estoque seria de acordo com o volume. Aluna quatro definiu que a função remover seria para retirar o produto que possuísse menor volume, e logo depois foi decidido em grupo que seria o de maior volume por ser mais coeso. Aluno cinco definiu que a função pesquisar seria para pesquisar quais produtos são ácidos e Aluna três decidiu que a função atualizar mostraria o estoque em que seria armazenado o produto, ou seja, atualizaria a situação de “indisponível”. Aluno cinco ficou responsável pela digitação do problema e organização do arquivo. Por fim, cada integrante criou duas funções extras que complementassem o problema principal.

Funções Extras:

- Aluna dois: Nocividade do produto, Definir a coloração dos produtos de acordo com a escala de pH.
- Aluno cinco: Calcular espaço total ocupado; Calcular espaço médio
- Aluna quatro: Ordenar os produtos em ordem alfabética; Padronizar a saída com letras maiúsculas.
- Aluna três: Porcentagem da quantidade dos volumes; Informar o pOH do produto.
- Aluna um: Informar se o produto existe no estoque ou não; Verificar se há espaço no estoque.

Problema: Estoque de Produtos - Empresa SinteQui

Um químico da empresa SinteQui, ficou responsável por realizar o estoque dos produtos químicos e agrupá-los de acordo com o nível de alcalinidade dos produtos, sendo que o estoque tem capacidade máxima de até 1000 mL no total.

Assim, enquanto houver espaço podem ser adicionados produtos no estoque. Entretanto, se não houver mais espaço, deve-se informar a mensagem: “Não é possível adicionar mais produto, estoque cheio”.

Por ser um trabalho exaustivo, criaremos um programa que faça isso para ele de maneira eficiente. Sendo assim, o programa terá como entrada o nome do produto, seu pH e o volume em mL do mesmo. Exemplo: Cloreto de Sódio, 7, 50. Os dados de cada produto serão salvos em um dicionário e depois serão armazenados em uma lista, como pode ser visto na Tabela 30.

Tabela 30 – Saída:

PRODUTO	PH	VOLUME	ESTOQUE
CLORETO DE SÓDIO	7	50	INDISPONÍVEL
ÁCIDO SULFÚRICO	2	75	INDISPONÍVEL
HIDRÓXIDO DE SÓDIO	8	100	INDISPONÍVEL

Será necessário criar uma função que atualize a situação de ‘INDISPONÍVEL’ para o tipo de estoque (ÁCIDOS, BÁSICOS, NEUTROS) em que o produto será armazenado, outra função que pesquise quais produtos são ácidos e uma que remova o produto com o maior volume no estoque. Exemplo: O produto com maior volume é o Hidróxido de Sódio.

Crie também uma função que imprima em qual estoque será armazenado cada produto. Como no exemplo que é visto na Tabela 31.

Tabela 31 – Saída:

PRODUTO	PH	VOLUME	ESTOQUE
CLORETO DE SÓDIO	7	50	NEUTROS
ÁCIDO SULFÚRICO	2	75	ÁCIDOS

Após a impressão, utilize a função remover e remova o produto de maior volume no estoque e chame a função imprimir para mostrar os resultados atualizados como mostrado acima. Depois disso, tente atualizar o produto Cloreto de Sódio para o volume de 2000 mL, essa tentativa não poderá ser aceita pelo programa, pois vai exceder a capacidade máxima do estoque, isto é, deverá informar que não tem espaço disponível.

Na figura 64, é apresentado um print tirado na hora da apresentação referente ao projeto pratico feito em conjunto no Replit.

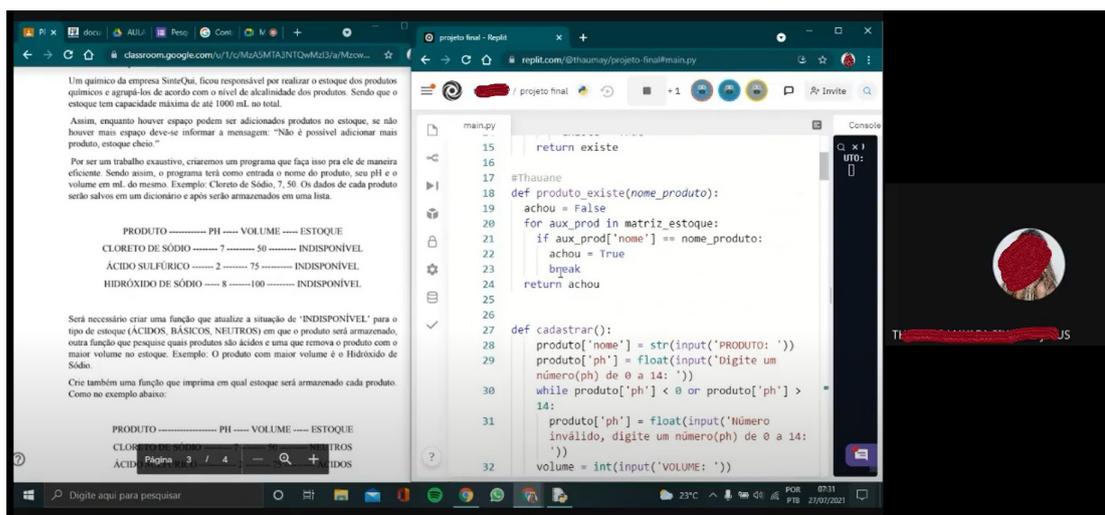


Figura 64 – Grupo 3: apresentando o projeto estoque produtos empresa SenteQui

Relatório final para apresentação do projeto

Na sexta-feira, dia 23/ 07/2021, começamos o processo de criação do código baseado no nosso projeto inicial apresentado dia 20/07/2021, cujo título é “Estoque de Produtos”, tendo esse por objetivo realizar o estoque de produtos químicos na indústria. Entramos em reunião no Google Meet para isso e foi decidido, em grupo, que primeiro seriam feitas as funções do problema base. No mesmo dia em que foram feitas as funções, foram criadas em conjunto, com exceção das funções: `remover_maior_volume` (`volume_produto`) e a `pesquisar_menores_ph()`.

No sábado dia 24/07/2021, cada integrante fez a sua respectiva função extra, mesmo algumas (`cor_do_produto()`, `maiusculas()`, `porcento_volume()`, `poh_produto()` e `produto_existe` (`nome_produto`)) apresentando erros. Sendo assim, no domingo, dia 25/07/2021, realizamos uma reunião com o professor para orientação nas funções que apresentavam erros. No mesmo dia, conseguimos corrigir o código e finalizá-lo.

Portanto, no dia 26/07/2021, realizamos uma última reunião em grupo para elaboração das considerações finais e revisão do projeto.

J.4 Grupo 4: Projetos colaborativos modelo 2

J.4.1 Cadastro de viagem

Enunciado do Problema

Fernanda decidiu que iria viajar para Espanha nas suas férias de verão, junto com a sua família e por ser uma grande quantidade de malas e não querer pagar excesso de bagagem, resolveu organizar os pertences que a família iria levar. Desse modo, usando seus conhecimentos de programação, decidiu desenvolver um aplicativo para que lhe ajudasse a lembrar o que seria útil levar na viagem e se está dentro dos limites de peso de bagagem.

Entrada:

- Mostre o nome do aplicativo;
- Login e/ou cadastro;
- Perguntar destino, se é um lugar quente ou frio;
- Quais peças ela vai querer;
- Peso da mala depois de guardar tudo.

Saída:

- Mostrar qual o usuário, itens na sua mala e taxa de bagagem

Na figura 65, é apresentado um print tirado na hora da apresentação referente ao projeto pratico feito em conjunto no Replit.

The screenshot shows a Replit IDE with a Python script named 'main.py' and its console output. The code defines lists for accessories, summer items, and winter items, and a dictionary for taxes. It prompts the user for item selection, item details, and travel type. The console output shows the user's selections and the program's responses.

```

1 -----LISTAS/DIC-----
2 lista_usuarios = list()
3 lista_malas = list()
4 lista_taxas = list()
5 dict_usuario = dict()
6 dict_mala = dict()
7 dict_taxa = dict()
8 dados_verao = {'ACESSÓRIOS':('Brinco', 'Pulseira',
9               'Colar', 'Relógio', 'Bone', 'Chapéu'), 'CALÇADOS':('Chinelo',
10              'Tênis', 'Sandálias', 'Sapato Social', 'Salto Alto'),
11              'COSMÉTICOS':('Sabonete', 'Shampoo', 'Condicionador',
12              'Creme de Pentear', 'Pasta de Dente', 'Escova', 'Pente',
13              'Fio Dental', 'Toalhas'), 'DOCUMENTOS':('Identidade',
14              'Passaporte', 'CM', 'MOPAS':('Vestido', 'Blusa', 'Meia',
15              'Calça', 'Short', 'Biquíni', 'Maiô', 'Sunga', 'Canga', 'Body',
16              'Camisa', 'Gravata'))}
17 dados_inverno = {'ACESSÓRIOS':('Brinco', 'Pulseira',
18               'Colar', 'Relógio', 'Touca', 'Chapéu'), 'CALÇADOS':
19               ('Chinelo', 'Tênis', 'Sandálias', 'Sapato Social', 'Salto
20               Alto', 'Contorno'), 'COSMÉTICOS':('Sabonete', 'Shampoo',
21               'Condicionador', 'Creme de pentear', 'Pasta de Dente',
22               'Escova', 'Pente', 'Fio Dental', 'Toalhas'), 'DOCUMENTOS':
23               ('Identidade', 'Passaporte', 'CM'), 'MOPAS':('Vestido',
24               'Blusa', 'Meia', 'Calça', 'Short', 'Biquíni', 'Body', 'Camisa',
25               'Casaco', 'Noletom', 'Sobretudo')}
26 -----FUNÇÕES-----

```

```

Qual item deseja adicionar?
(Informe "q" se deseja finalizar sua mala.)
Informe o item: identidade
Informe o item: pente
Informe o item: 6
MALA: ['Identidade', 'pente', 'tenis']

A viagem é Nacional ou Internacional?
[1] Nacional
[2] Internacional
[3]
1
Digite o comprimento da mala em cm: 45
Digite a altura da mala em cm: 34
Digite o peso da mala em kg: 23
A dimensão da mala é: 45,8 cm
A taxa a ser paga pelas dimensões elevadas da mala é: R$ 298,6

Bem-vindo a Mala Online
[1] - Login
[2] - Cadastar
[0] - Sair
[ ]:

```

Figura 65 – Grupo 4: apresentando o projeto cadastro de viagem

Relatórios:

- 13/07 - Primeira reunião: Discutimos e escolhemos o tema.
- 14/07 - Segunda reunião: Estruturou a questão, como ela seria feita. 15/07 - Envio do rascunho para o professor no grupo.
- 19/07 - Terceira reunião: Dividimos tarefas para cada pesquisador:
 - Aluno um: Pesquisar como fazer a login e cadastro;

- Aluna dois: Pesquisar como fazer os cálculos de excesso de bagagem;
 - Aluna três: Pesquisar como fazer import
 - Aluna quatro: Pesquisar como fazer banco de dados.
- 20/07 - Quarta reunião: Discussão da pesquisa
 - Tentamos desenvolver as tarefas.
- Na parte do banco de dados e import, percebemos que estão bem interligados e que precisaríamos que o banco de dados fosse online para poder rodar em qualquer computador. Assim, olhamos o JSON, mas desistimos dele porque precisaria do entendimento de java. Olhamos o SQLite, porém, não daria certo por não ser uma plataforma 100% online. Vimos por alto o banco de dados da Amazon e do Replit, e ficamos de aprofundar melhor sobre.
- Na parte dos cálculos, devido ao fato das empresas possuírem variados preços para pagar de excesso, decidimos estipular dois valores médios (nacional e internacional) para a realização dos cálculos de excesso se a mala do passageiro exceder o peso de 23kg.
- Na parte do login e cadastro tentamos fazer uma tela.
- 21/07 - Quinta reunião: Continuamos desenvolvendo as atividades da reunião anterior
 - Optamos por deixar de lado o banco de dados com SQLite e simulamos o BD diretamente no Replit por meio de dicionários e listas;
 - Início do desenvolvimento código para login e cadastro;
 - Início do desenvolvimento do código para a realização dos cálculos;
- 22/07 - Sexta reunião: Reunião com o professor
 - Ajustes de código e orientação para desenvolvimento do projeto.
- 24/07 - Sétima reunião: Resolução dos problemas de furo no código.
 - Fizemos ajustes no código para calcular a taxa de excesso.
 - Ajuste da função cadastrar mala.
- 25/07 - Oitava reunião: Finalização do código.
 - Fizemos a junção dos códigos na main do projeto;
 - Organizamos a main do projeto.

Nota do grupo:

Nosso trabalho foi inteiramente feito por todos da equipe, por meio de reuniões diárias no discord e Google Meet. Assim, não existiu bem uma divisão, já que todos participaram de todas as etapas de confecção do código, de criação das funções, até consertando os erros nos condicionais que deram bastante trabalho por acharmos que funcionava melhor assim depois da reunião de discussão, que aconteceu no dia 20 de julho. Por isso não sabemos ao certo quem fez determinada parte.

Como o grupo explicou o que fez em cada parte do código além da apresentação, colocando no documento entregue, então será mostrado a seguir apenas o nome do método e a explicação dada por eles.

Explicando o Código

- **Parte 1, Declarações de variáveis:** Na parte introdutória do código, nós organizamos as listas vazias e os dicionários que iriam servir de banco de armazenamento ao decorrer do código e adicionamos listas compostas já com dados, sendo que o nosso aplicativo funciona como um lembrete de itens úteis para se colocar em uma mala.
- **Parte 2, Função `existe_usuario(user)`:** Essa função tem como dever verificar se existe um usuário cadastrado na lista de usuários (`lista_usuarios`) com as informações respectivas registradas na tela de cadastro para evitar repetições de dados de registro. Foi utilizado um mecanismo de operadores booleanas, definindo `existe` inicialmente como falso, até que conste o contrário usando o `'for'`, buscando na lista de usuários um `user` correspondente ao inserido até que o condicional `'if'` seja verdade e o código continue.
- **Parte 3, Função `login(user, password)`:** Verifica se consta, em cadastro, o usuário e senha digitados no login para dar acesso ao resto do aplicativo, utilizando o mesmo mecanismo da parte 2 e aplicando operadores booleanos e condicionais.
- **Parte 4, Função `cadastrar(user, password)`:** Essa função serve para cadastrar novos usuários na lista de usuários (`lista_usuarios`), recebe usuário e senha registrando no dicionário de usuário (`dict_usuario`), em seguida a partir do `'if'` verifica se `lista_usuarios` está vazia. Isto é, caso esteja vazia, ela cadastra; caso não esteja, ela entra em outro condicional `'if'` que puxa `def existe_usuario(user)` e, dessa forma, dependendo do resultado cadastra o usuário ou imprime que já existe esse usuário.
- **Parte 5, Função `existe_mala(user)`:** Usando dos mesmos recursos que a parte 2 e a parte 3 ele verifica se existe uma mala (`dict_mala`) registrada na lista de malas (`lista_malas`) para no `'while'` puxar o cadastro da mala (`cadastrar_mala(user, lista_itens)`), se não existir, ou imprimir a mala (`dict_mala`) se existir.

- **Parte 6, Função `cadastrar_mala(user, lista_itens)`:** Essa parte do código é responsável por adicionar as malas nas respectivas contas e adicionar os dados na lista de malas (`lista_malas`) e apresentar na tela. Isso é feito com uma cópia do `dict_mala` usando `‘.copy()’` e adicionando na lista de mala pelo `‘.append’`.
- **Parte 7, Função `cadastrar_itens(user)`:** Essa parte do código tem como função preencher uma lista vazia a partir da adição dos itens que o usuário deseja e inserindo no `cadastrar_mala(user, lista_itens.copy())`. Para tanto, usa-se o laço de repetição `‘while’` que fornece a chance de o usuário colocar os itens na mala até que o condicional seja zero e o looping se encerre.
- **Parte 8, Função `valor_taxa_mala(user, taxa)`:** Funciona da mesma forma que a parte 6, diferindo-se por, dessa vez, adicionar o `user` e a `taxa` – que vai ser determinada na função `cálculo()` e adiciona-se ambas as variáveis no dicionário de `taxa (dict_taxa)` e, posteriormente, na lista de `taxas (lista_taxas)`.
- **Parte 9, Função `calcula()`** Essa função é responsável por calcular a taxa de excesso de bagagem em que será inserida no `valor_taxa_mala (user, taxa)`. Ou seja, de acordo com os parâmetros que foram estabelecidos pela altura (cm), comprimento (cm), largura (cm) e o peso (kg) da mala. Essa função foi feita embasada em condicionais `‘if’` que vão ter caminhos diferentes ao depender da necessidade de quem está usando o aplicativo e caso forem feitas ações erradas, recebendo avisos de erro.
- **Parte 10, Testando o código com laço de repetição `while`:** Essa parte é responsável por rodar o código inteiro em um laço de repetição. Aqui ela solicita variáveis que irão determinar o caminho do usuário pelo programa a partir de condicionais. Inicia solicitando login ou cadastro, para cadastro o programa pede o usuário e senha e, logo após, chamando o `def cadastrar(user, password)` para verificar disponibilidade de usuário e registrar as informações. Para o login, solicita o usuário e a senha para verificar se já existe o cadastro. Para caso afirmativo, o login é efetuado com sucesso. Em seguida, utilizando o `existe_mala(user)`, observa-se se já existe uma mala registrada; se sim, o nome de usuário, mala e taxa são exibidos na tela; se não, é iniciado o processo de cadastro da mala. Assim, começa solicitando o tipo de clima do destino da viagem, sendo verão ou inverno, para decidir qual lista de itens será exibido na tela. Após adicionar os itens na mala, usando a função `cadastrar_itens(user)`, inicia-se o cálculo para a taxa de excesso de bagagem e, assim, finaliza-se o código.

J.5 Grupo 5: Projetos colaborativos modelo 2

J.5.1 Estoque de concessionária

Enunciado do Problema:

A concessionária de veículos multimarcas GJPM de São Cristóvão/SE possui problemas na organização do seu estoque. Desse modo, contratou sua equipe de programadores para solucionar os seguintes empecilhos: cadastrar as informações de modelo, ano, cor, fabricante, km, quantidade, preço de custo e preço de venda; pesquisar e consultar os veículos na base de dados; atualizar o estoque fazendo novos cadastros; alterar o registro de estoque adicionando ou removendo itens; atualizar o estoque fazendo novos cadastros.

Para realizar o teste, você deverá utilizar listas compostas para salvar as informações e dicionários para salvar uma única informação. As listas devem ser usadas para salvar os dicionários. É necessário, também, que sejam utilizados blocos de condicionais e laços de repetição na resolução desse problema.

Seu programa deverá conter as seguintes funções:

FUNÇÕES PRINCIPAIS:

1. CADASTRAR_VEICULO():

- Recebe as informações, cria o dicionário e salva;
- Cadastrar informações de modelo, ano, cor, fabricante, classificação e preço.
- Tratar os erros de informação digitada;
- Limite de 50 veículos no estoque e caso ultrapasse o limite do estoque, será informado uma mensagem na tela: “Seu estoque chegou ao limite de 50 carros, remova algum veículo!”;
- Caso seja efetuado o cadastramento, será emitida a seguinte mensagem: “Veículo cadastrado com sucesso!”.

2. PESQUISAR():

- Irá pesquisar os veículos de acordo com o modelo digitado;
- Caso o veículo informado não seja encontrado será emitida a mensagem: “Nenhum veículo encontrado para o termo!”;
- Limite de 50 veículos no estoque e caso ultrapasse o limite do estoque, será informado uma mensagem na tela: “Seu estoque chegou ao limite de 50 carros, remova algum veículo!”;
- Caso não tenha nenhum veículo cadastrado no sistema, será informada a seguinte mensagem: “Não é possível pesquisar, estoque está vazio.”.

3. REMOVER():

- Com essa função será possível remover os modelos de veículos contidos no estoque;

- Com o item removido do estoque, será informada a mensagem: “Carro removido com sucesso!”;
- Caso o sistema não encontre o veículo será emitida a mensagem: “Carro não encontrado.”.

4. IMPRIMIR():

- Irá imprimir, em uma tabela, todos os veículos cadastrado no sistema e mostrar.

5. ATUALIZAR():

- Irá atualizar as informações do veículo escolhido a partir da posição; Caso a posição informada não esteja de acordo com a quantidade de veículos dentro do estoque, será informada a mensagem: “Posição informada não existe no estoque. Tente novamente.”;
- Se a informação digitada não for compatível com o que foi pedido, será emitida a mensagem: “Não foi possível atualizar.”.

FUNÇÕES EXTRAS:

1. MENU_PRINCIPAL():

- Irá mostrar a interface de todas as opções contidas no código.

2. ORGANIZAR_ANO():

- Organiza os veículos por ano.

3. VENDAS():

- Irá selecionar o veículo vendido e removê-lo do estoque.

4. DATA_ACAO():

- Irá colocar a data do dia em que o estoque está sendo utilizado.

5. RELATORIO_VENDAS():

- Fará um relatório de todas as vendas de veículos.

6. PRECO_ORDEM_CRESCENTE():

- Irá organizar a tabela de preço do mais caro para o mais barato do estoque.

Na figura 66, é apresentado um print tirado na hora da apresentação referente ao projeto pratico feito em conjunto no Replit.

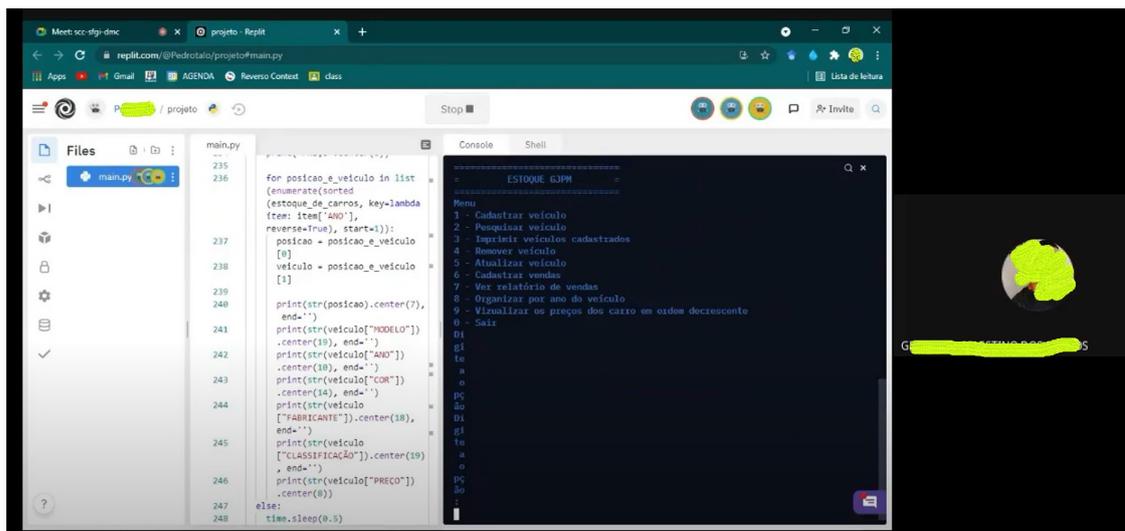


Figura 66 – Grupo 5: apresentando o projeto concessionária

Relatório:

ATIVIDADES DESENVOLVIDAS EM CONJUNTO

- 13/07/2021: Reunimos no Meet para discussão do tema principal do projeto.
- 15/07/2021: Com o tema no projeto decidido, desenvolvemos melhor as ideias do que teria em cada def (função) principal e como poderíamos trabalhar com essas informações para as def (funções) extras dentro do código. Além de também apresentá-las ao professor por meio de relatório escrito enviado no grupo.
- 20/07/2021: Desenvolvemos melhor a ideia do projeto proposto, com suas def principais e extras definidas e também do que cada integrante iria ficar encaminhado de fazer dentro do projeto. Assim, ficando definido que def principais: iria ser trabalhado em conjunto por meio de chamadas no Meet; def extras: iria ser trabalhado de maneira individual e, em caso de dúvidas, consultar os demais integrantes ou professor.
- 22/07/2021: Iniciamos a criação do código às 14h do dia presente por meio de chamada no Meet. Nesse dia, conseguimos desenvolver a função cadastrar do código e demos início à função pesquisar. No entanto, não a concluímos.
- 23/07/2021: Demos início à outra chamada no Meet para continuarmos o desenvolvimento do código. Vale salientar que tivemos algumas dificuldades em desenvolver essa função, uma vez que sempre dava erro. Então, fomos pesquisar string, funções e dicionários para que pudessem nos ajudar no desenvolvimento do código. Assim, conseguimos desenvolver a função atualizar nesse dia. Ademais, também concluímos a função imprimir.
- 24/07/2021: Começamos a desenvolver a função remover, porém, também estávamos com dúvidas em relação a como iríamos dar prosseguimento nessa função. Então, entramos em

chamada via Meet com o professor, de forma que foi possível esclarecer nossas dúvidas, além dele nos “dar o caminho” de como poderia ser feito a função pesquisar. Feita a função remover, partimos para fazer a função pesquisar em que pudemos obter êxito no mesmo dia.

- 25/07/2021: Demos ao desenvolvimento das def extras.
- Observações: Vale ressaltar que, no decorrer do projeto, existiram algumas mudanças de funções extras dentro do código proposto.

ATIVIDADES FEITAS INDIVIDUALMENTE

- Aluno um:
 - Desenvolvimento inicial da função cadastrar_veiculo(), com a utilização de listas e dicionários, e da função imprimir. Criação da variável limite estoque e implementação dela no cadastro para limitar o cadastramento de veículos.
- Aluna dois:
 - Inicialmente, sugeri a ideia de como seria o projeto, em que propus que poderíamos desenvolver um sistema de gestão de estoque a fim de organizar de maneira mais eficiente.
 - Ademais, criei duas funções que pudessem organizar as vendas e fazer um balanço mensal, chamadas: vendas e relatório_vendas.
 - Após ter criado essas duas funções, criei uma outra função que retornasse à data do dia a fim de implementar ainda mais o projeto e, por fim, sugeri a ideia de usar a função sleep para deixar o código mais organizado.
- Aluna três:
 - Criação da def menu_principal() para que o código tivesse uma interface fácil de usar, informativa e mais interativa para quem estivesse utilizando o programa;
 - Adicionei cor ao código, visando a estética da interface;
 - Desenvolvimento da def organizar_ano() que tem como finalidade ordenar os carros por lançamento, ou seja, do carro mais atual para o mais antigo dentro do estoque.
- Aluno quatro:
 - A priori, quando estávamos dialogando no que tange ao tema do projeto, propus uma ideia de estoque de veículos. A posteriori, o Aluno um sugeriu que ficaria mais assertivo implementar o modelo de uma concessionária de carros. Deve-se ressaltar que a ideia de estoque foi desenvolvida pela Aluna dois. Contudo, todos apoiaram o tema e demos início ao projeto, de fato.

- Desenvolvimento da def `preco_ordem_crescente()` com apoio do código da Aluna três.

J.6 Grupo 6: Projetos colaborativos modelo 2

J.6.1 Meu Delivery

Enunciado do Problema:

Maria quer pedir um lanche pelo iFood, mas primeiramente precisa saber se a região em que se encontra tem a disponibilidade do serviço de entrega. Para que essa verificação seja concretizada, será solicitado o nome da cidade em que a cliente está. Sendo que, nas cidades de Aracaju, Barra dos Coqueiros, Nossa Senhora do Socorro e São Cristóvão a entrega poderá ser feita, mas com uma diferença no valor da entrega, que será R\$15,00 para Aracaju e R\$ 20,00 para as demais localidades.

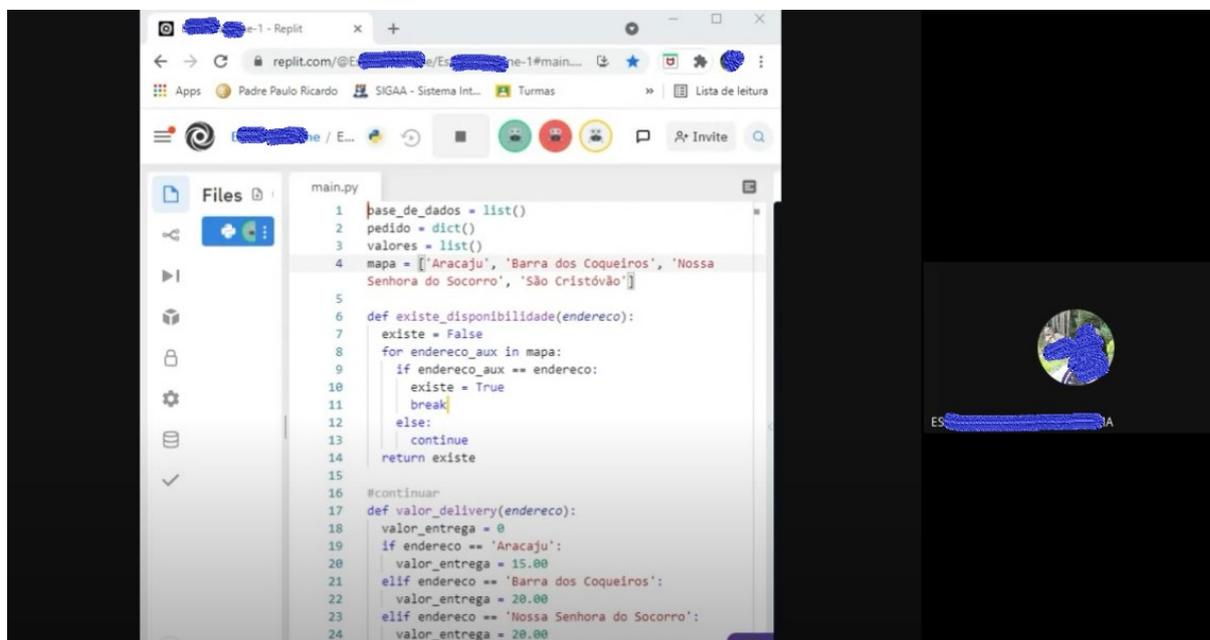
Após a confirmação da disponibilidade de entrega na região indicada, o aplicativo pedirá alguns dados pessoais como: nome e telefone. Com esses dados cadastrados, Maria poderá escolher o lanche que deseja comprar e o aplicativo indicará o valor total do pedido, se o somatório desse valor for de R\$ 60,00 ou mais o aplicativo disponibilizará um cupom de 30% de desconto para a compra. Já com o valor final da compra, o aplicativo irá solicitar que o cliente informe a forma de pagamento que deseja utilizar, sendo dinheiro ou cartão as opções disponíveis. Depois disso, o pedido será finalizado e o restaurante irá receber a impressão dos dados do cliente, lanches escolhidos e o valor total com desconto, caso seja atingido o valor mínimo para este ser adquirido.

Funções necessárias:

1. `Existe_disponibilidade` (Aluna um)
2. `Realizar_pedido` (Aluna dois)
3. `Escolher_lanche` (Aluna três)
4. `Valor_lanche` (Aluna três)
5. `Valor_pedido_com_desconto`(Aluna quatro)
6. `Valor_delivery` (Aluna um/Aluna três)
7. `Salvar_pedido` (Aluna dois/Aluna quatro)
8. `Imprimir_pedido`(Aluna um/Aluna dois/Aluna quatro)

Observações: Para a construção e desenvolvimento do projeto, todas as componentes participaram de maneira significativa e igual.

Na figura 67, é apresentado um print tirado na hora da apresentação referente ao projeto pratico feito em conjunto no Replit.



```

1 base_de_dados = list()
2 pedido = dict()
3 valores = list()
4 mapa = [['Aracaju', 'Barra dos Coqueiros', 'Nossa
5 Senhora do Socorro', 'São Cristóvão']]
6
7 def existe_disponibilidade(endereco):
8     existe = False
9     for endereco_aux in mapa:
10        if endereco_aux == endereco:
11            existe = True
12            break
13        else:
14            continue
15    return existe
16
17 #continuar
18 def valor_delivery(endereco):
19     valor_entrega = 0
20     if endereco == 'Aracaju':
21         valor_entrega = 15.00
22     elif endereco == 'Barra dos Coqueiros':
23         valor_entrega = 20.00
24     elif endereco == 'Nossa Senhora do Socorro':
25         valor_entrega = 20.00
  
```

Figura 67 – Grupo 6 apresentado o código fonte de seu projeto

Relatório:

Abaixo está indicando o que ficou separado para cada um no código. Todavia, durante a desenvolvimento dele, cada participante do grupo acabou contribuindo com a parte do outro.

- Aluna um:
 - Deu a ideia do desconto aplicado a compra se atingido um valor mínimo.
 - Criação da função Existe_disponibilidade. (Disponibilidade de entrega na cidade da cliente)
 - Criação da função Valor_delivery. (Atribuir valores de entrega para as regiões disponíveis)
 - Criação da função Imprimir_pedido. (Forma de impressão do pedido e dados dos clientes)
- Aluna dois:
 - Deu a ideia do frete para cada região de entrega.
 - Criação da função Realizar_pedido. (Pedir dados do cliente para o pedido ser feito)

- Criação da função Salvar_pedido . (Salvar pedido para ser finalizado e feito pelo restaurante)
- Criação da função Imprimir_pedido. (Forma de impressão do pedido e dados dos clientes)
- Aluna três:
 - Deu a ideia da questão no geral.
 - Criação da função Escolher_lanche. (Selecionar os itens)
 - Criação da função Valor_lanche. (Atribuir valores aos itens)
 - Criação da função Valor_delivery. (atribuir valores de entrega para as regiões disponíveis)
- Aluna quatro:
 - Deu a ideia da forma de impressão do pedido
 - Criação da função Valor_pedido_com_desconto. (Valor total da compra e se atingir o valor mínimo de R\$ 60,00 o desconto é dado)
 - Criação da função Salvar_pedido. (Salvar pedido para ser finalizado e feito pelo restaurante)
 - Criação da função Imprimir_pedido. (Forma de impressão do pedido e dados dos clientes)

Vale ressaltar que realizamos encontros do grupo para definir a parte teórica e realizar o código. Assim, foram feitas reuniões pelo Google Meet, já no grupo do whatsapp, diariamente compartilhávamos conteúdos para contribuir na realização do projeto. Utilizamos materiais passados pelo professor durante o período e pesquisas individuais em sites. O código foi feito pelo Replit, pois ele disponibilizava total acesso a todas as integrantes do grupo.

J.7 Grupo 7: Projetos colaborativos modelo 2

J.7.1 Hortifruti delivery

O problema proposto diz respeito ao sistema de gerenciamento de um empreendimento comercial do tipo delivery. Para tanto, é preciso lidar com clientes, fornecedores, controle de estoque e controle de entregas. Desta forma, o propósito deste projeto é criar um sistema unificado, capaz de realizar a comunicação entre estes setores, visando otimizar o tempo de processamento de pedidos e entregas, além de solucionar problemas que possam estar ligados ao controle de estoque.

Onde tem quatro problemas:

- P1 - Cadastro de Clientes :
 - Ter um cadastro de clientes auxilia no processamento do pedido, visto que não será necessário realizar o cadastro sempre que o cliente efetuar uma compra no estabelecimento. Bastando, portanto, efetuar uma busca no sistema e proceder para os produtos da entrega.
 - Algumas informações importantes, como: Nome, CPF, Idade e Cidade (Endereço) farão parte do cadastro básico do cliente, podendo ser inseridos outras informações em atualizações futuras do sistema.
- P2 - Cadastro de Fornecedores:
 - O gerenciamento de fornecedores é outro ponto crucial na empresa, em que é preciso ter os melhores fornecedores com cadastro ativo para que a reposição de estoque possa ser realizada em um tempo hábil. Esta área do negócio está diretamente ligada ao setor de produtos/estoque, no qual é preciso saber quais fornecedores disponibilizam cada tipo de produto, ressaltando a importância de se manter um cadastro atualizado. Como informações básicas, inicialmente será feito um cadastro do fornecedor contendo dados da empresa como Nome, CNPJ e os produtos por ela oferecidos.
- P3 - Controle de Estoque (Produtos):
 - O controle de estoque precisa estar sempre atualizado, desde o momento em que um pedido é realizado ao momento de compra de insumos para a sua reposição. Manter os produtos com código de etiquetagem garante melhor eficiência no processo de busca. Outras informações importantes são o preço de venda, a quantidade vendida e em estoque. Estes dados podem ser utilizados posteriormente para gerar estatísticas de faturamento de cada produto, ou de todos, de modo a auxiliar a administração no processo de direcionamento dos planos futuros da empresa. Em atualizações futuras, podem ser anexadas funções gráficas que permitam melhor visualização.
- P4 - Controle de Entregas:
 - Saber quem foi o cliente, o que comprou e onde será entregue é a principal atividade deste setor. Para isso, é necessário consultar a lista de produtos para saber se tem estoque suficiente antes de gerar o pedido, atualizando o estoque sempre que houver alterações. O estoque diminuirá quando um pedido for efetuado, mas se o pedido for cancelado, o estoque deverá retornar ao estado anterior. O controle de entregas é, então, o principal mecanismo do sistema, partindo do ponto em que interliga o gerenciamento de produtos e clientes. O acompanhamento dos pedidos é importante, já que monitora as alterações do gerenciamento de estoque.

- Observação:
 - Pode-se ser usado Status na parte de Entrega: Cadastro, concluído, entregue.

Para Cadastro de Clientes: Foi apresentado o problema que envolve cadastro de clientes e auxílio no processo de pedidos. Como informações básicas, inicialmente foi feito um cadastro do fornecedor, contendo dados da empresa como Nome, CNPJ e os produtos por ela oferecidos. Para tanto, foram usadas as seguintes funções:

1. Cadastrar
2. Pesquisar
3. Imprimir
4. Remover
5. Atualizar
6. Pedidos Anteriores
7. Melhores Clientes

Essa parte do Projeto foi feita pelo Aluno um. A parte responsável já feita foi de apresentar o problema e também a das funções, em que foi dito cada uma usada no problema envolvido.

Para Cadastro de Fornecedores: Foi apresentado o problema que envolve o Gerenciamento de Fornecedores. Isto é, um ponto de suma importância para esse problema, porque para que a reposição de estoque possa ser realizada em um tempo hábil, são necessários ótimos fornecedores e com cadastros ativos. Para isso, foram usadas as seguintes funções:

1. Cadastrar
2. Pesquisar
3. Imprimir
4. Remover
5. Atualizar
6. Buscar por Produto
7. Solicitar Compra

Essa parte do Projeto foi feita pelo Aluno dois, em que apresenta o problema citado acima e fica responsável pela parte já feita no qual mostra cada código utilizado (Do 1 ao 7) e apresentá-los.

Para Controle de Estoque: Foi apresentado o problema que envolve manter sempre atualizado um controle de estoque, desde o momento em que um pedido é realizado ao momento de compra de insumos para sua reposição. Em suma, nesse caso basicamente tudo vai girar em torno de preço de vendas, quantidade de vendas e o que está em estoque. Foram utilizadas as seguintes funções:

1. Cadastrar
2. Pesquisar
3. Imprimir
4. Remover
5. Atualizar
6. Ver Estoque
7. Mais Vendido

Essa parte do Projeto foi feita pela Aluna três. Foi apresentado, por ela, o problema citado acima e ficou responsável pela parte já feita em que mostra todos os códigos utilizados e, assim, apresentá-los.

Para Controle de Entregas: Foi apresentado o problema que envolve Detecção de clientes, detecção de produtos e em que será feito a entrega. O controle de entregas é, então, o principal mecanismo do sistema, partindo do ponto em que interliga o gerenciamento de produtos e clientes. Foram usadas as seguintes funções:

1. Cadastrar
2. Pesquisar
3. Imprimir
4. Remover
5. Atualizar
6. Detalhar Entrega
7. Status da Entrega

Essa parte foi feita pela Aluna quatro em que foi apresentado por ela o problema citado acima e ficou responsável pela parte já feita mostrar todos os códigos utilizados e apresentá-los.

Na figura 68, é apresentado um print tirado na hora da apresentação referente ao projeto pratico feito em conjunto no Replit.

```

Files
  main.py
  BD_Clientes.json
  BD_Entregas.json
  BD_Fornecedores...
  BD_Produtos.json

main.py
1 report_json
2
3 ===== PULID
4
5
6 def Telainicio():
7     erroopcao = True
8     while (erroopcao):
9         print("=====")
10        print("#  HORTIFRUTI DELIVERY  #")
11        print("#                               #")
12        print("#  Sistemas Disponíveis  #")
13        print("#                               #")
14        print("#  Tecla 1 para Produtos  #")
15        print("#  Tecla 2 para Clientes  #")
16        print("#  Tecla 3 para Fornecedores #")
17        print("#  Tecla 4 para Entregas  #")
18        print("#                               #")
19        print("#  Tecla X para encerrar! #")
20        print("#                               #")
21        print("=====")
22        opcaoEscolhida = input("Digite a opção desejada")
23        (ex: 1):
24        if opcaoEscolhida.upper() in ["1", "2", "3", "4",
25        "X"): # Verifica se foi inserida uma opção válida
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

Console
Shell

Erro!
Digite apenas números!
Informe o Código do Produto que deseja Remover (ex: 1): 1
CÓDIGO      1
NOME        UVA
PREÇO       3.5
Confirma Remoção da base de dados? (Y/N): y
=====
#  Gerenciamento de Produtos  #
#                               #
#  Opções Disponíveis         #
#                               #
#  Tecla C para Cadastrar     #
#  Tecla P para Pesquisar     #
#  Tecla I para Imprimir      #
#  Tecla R para Remover       #
#  Tecla A para Atualizar     #
#  Tecla E para Ver Estoques  #
#  Tecla V para Mais Vendido  #
#                               #
#  Tecla X para encerrar!     #
#                               #
=====
Digite a opção desejada (ex: P):
  
```

Figura 68 – Grupo 7 apresentado o código fonte de seu projeto

Observação: Além das funções exigidas, cada integrante acrescentou duas outras mais funções pedidas de acordo com o Projeto. E nas funções exigidas: Cadastrar, Pesquisar, Imprimir, Remover e Atualizar, nós discutimos e implementamos soluções similares nas quatro partes do projeto, ou seja, utilizando o mecanismo de validação de dados através de laços While e condicionais if, else. Além disso, foram criadas listas para cada etapa e cada produto armazenado no dicionário e, posteriormente, o dicionário armazenado na lista com as informações tanto para o Produto, para o Cliente, para os Fornecedores e para as entregas. Como por exemplo: Nome do cliente, CPF, Cidade são informações que são salvas em um dicionário e esse dicionário é, em seguida, salvo em uma lista.

J.8 Grupo 8: Projetos colaborativos modelo 2

J.8.1 Campeonato de várzea

Enunciado do problema:

O nosso projeto tem por finalidade apresentar um código direcionado às situações do cotidiano, dentre elas, a apuração de resultados em um campeonato de futebol com 5 equipes. Nesse sentido, foi utilizado o programa em Python para montar a estrutura desses códigos.

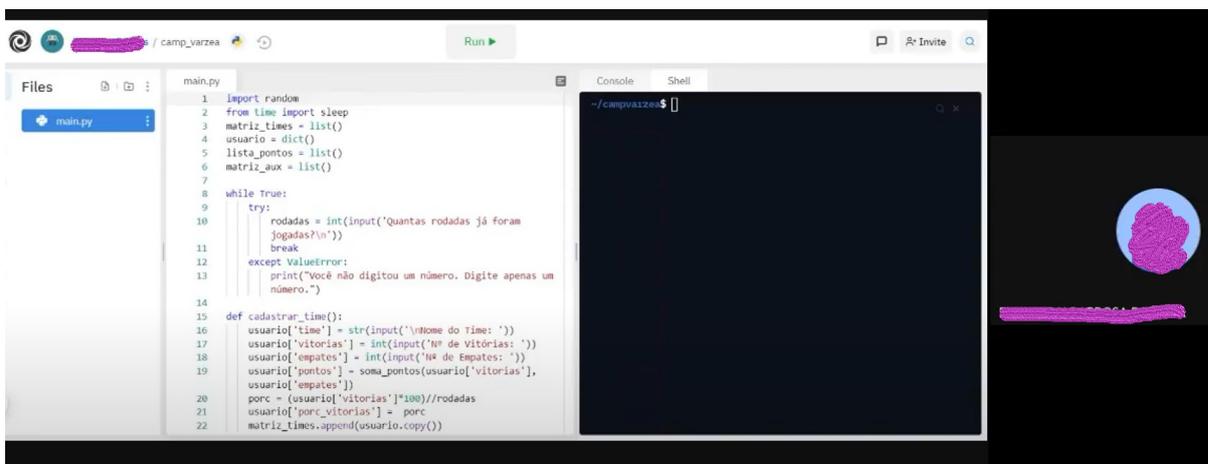
Partindo desse princípio, executamos as seguintes funções com suas respectivas funcionalidades:

1. Cadastrar_time() (Responsável principal todos os integrantes):

- Responsável por receber os nomes, quantidade de vitórias e empates, calculando automaticamente suas % de vitórias.
2. `Atualizar_time()` (Responsável principal todos os integrantes):
 - Utilizado apenas para corrigir algum valor digitado equivocadamente no código anterior.
 3. `Soma_pontos()` (Responsável principal Aluna um):
 - Este, calcula a pontuação total baseada nas vitórias (3 pontos) e nos empates (1 ponto) de cada equipe.
 4. `Pesquisar()` (Responsável principal todos os integrantes):
 - Realiza a procura de um determinado time que a pessoa solicitar antes de sair a classificação total das equipes.
 5. `Classificacao()` (Responsável principal Aluno dois):
 - Gera a ordem do melhor para o pior classificado dentre os times.
 6. `Pos_time_menos_pontos()` (Responsável principal Aluna três):
 - Esta função procura o time pior classificado para posteriormente ser removido do campeonato.
 7. `Remover_time()` (Responsável principal todos os integrantes):
 - Remove o time pior classificado.
 8. `Mata_mata()` (Responsável principal Aluno quatro):
 - Serão selecionados os 4 melhores times para jogar entre eles e decidir quais jogarão a final do campeonato.
 9. `Final_camp()` (Responsável principal Aluna cinco):
 - Demonstra quais são os 2 times classificados para jogar a final e determinar respectivamente quem será o campeão e o vice.
 10. `Imprimir_resultados()` Esta função será utilizada algumas vezes neste projeto. Com as seguintes finalidades (Responsável principal todos os integrantes):
 - Mostrar as equipes cadastradas
 - Caso alguma equipe precise atualizar a quantidade de vitórias ou empates com o novo resultado

- Classificação das equipes e seus respectivos pontos
- Remoção da equipe com pior desempenho
- Demonstrar as classificadas para o mata-mata
- Informar quem irá participar da final do campeonato
- Mostrará quem será a equipe campeã.

Na figura 69, é apresentado um print tirado na hora da apresentação referente ao projeto prático.



```
1 import random
2 from time import sleep
3 matriz_times = list()
4 usuario = dict()
5 lista_pontos = list()
6 matriz_aux = list()
7
8 while True:
9     try:
10        rodadas = int(input("Quantas rodadas já foram
11        jogadas?\n"))
12        break
13    except ValueError:
14        print("Você não digitou um número. Digite apenas um
15        número.")
16
17 def cadastrar_time():
18     usuario['time'] = str(input('Nome do Time: '))
19     usuario['vitorias'] = int(input('Nº de vitórias: '))
20     usuario['empates'] = int(input('Nº de Empates: '))
21     usuario['pontos'] = soma_pontos(usuario['vitorias'],
22     usuario['empates'])
23     porc = (usuario['vitorias']*100)//rodadas
24     usuario['porc_vitorias'] = porc
25     matriz_times.append(usuario.copy())
```

Figura 69 – Grupo 8: apresentado o projeto campeonato de varzea

Relatório de atividades:

- Reunião 1 – Discutimos acerca do tema que seria trabalhado no projeto e concluímos que uma situação interessante a ser trabalhada seria um campeonato de futebol.
- Reunião 2 – Iniciamos o trabalho colocando as funções básicas obrigatórias no projeto, foram elas: Cadastrar times, pesquisar e imprimir, as funções foram escritas pelo Aluno quatro e o restante do grupo foi apoiando e discutindo quais métodos seriam utilizados e quais códigos seriam adequados para cada situação do problema.
- Reunião 3 – Pensamos e discutimos quais seriam as funções extras que seriam desenvolvidas no projeto e a partir disso definimos quem ficaria responsável por cada uma das funções.
- Reunião 4 – Pesquisamos e colocamos em prática algumas novas ideias e conceitos sobre determinadas funções, uma vez que tínhamos um pouco de dificuldades em associar listas compostas com as funções básicas do problema e, desse modo, conseguíamos avançar aos poucos.

- Reunião 5 – Uma das funções extras do projeto foi: A soma de pontos, desenvolvida pela Aluna um. Essa função que teve a finalidade de calcular a pontuação final de cada time foi muito essencial, uma vez que outras funções dependeram desta para ficar completa.
- Reunião 6 – Na função em que a Aluna três é responsável (posição do time com menos pontos), foram encontrados alguns percalços, uma vez que não estávamos conseguindo desenvolver uma maneira de classificar os times do melhor pontuado para o pior pontuado. Porém, o professor nos orientou com relação à função lambda para a resolução deste problema.
- Reunião 7 – Continuamos trabalhando nas funções extras, no entanto, em decorrência às dificuldades encontradas na função classificação (função essa que o Aluno dois é o responsável), o professor nos auxiliou nos problemas que não estávamos conseguindo solucionar, nisso, funções novas foram pesquisadas, outros métodos foram encontrados e utilizados para que pudéssemos dar continuidade ao projeto e, dessa forma, conseguimos resolver.
- Reunião 8 – Nesta reunião, estávamos buscando a melhor forma de desenvolver as últimas funções extras, sendo elas: Mata-Mata (Aluno quatro) e Final (Aluna cinco). Novamente, encontramos alguns problemas e foram pesquisados vários métodos a fim de encontrar uma solução.
- Reunião 9 – Com praticamente tudo resolvido, fizemos alguns aprimoramentos no código para deixar de maneira melhor apresentável e compreensível aos olhos de quem fosse presenciar pela primeira vez.
- Reunião 10 – Reunião feita para elaborar o relatório do trabalho realizado pelo grupo.