



**UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO**  
**DEPARTAMENTO DE ESTATÍSTICA E INFORMÁTICA**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA APLICADA**

**FRANCIELY ALVES DE SOUZA**

**APLICAÇÃO DE LEARNING ANALYTICS NA ANÁLISE  
AUTOMÁTICA DE ATIVIDADES DE INTRODUÇÃO À  
PROGRAMAÇÃO COM O SCRATCH**

**RECIFE – PE**

**2022**

**FRANCIELY ALVES DE SOUZA**

**APLICAÇÃO DE LEARNING ANALYTICS NA ANÁLISE  
AUTOMÁTICA DE ATIVIDADES DE INTRODUÇÃO À  
PROGRAMAÇÃO COM O SCRATCH**

Dissertação submetida à Coordenação do Programa de Pós-Graduação em Informática Aplicada do Departamento de Estatística e Informática - DEINFO - Universidade Federal Rural de Pernambuco, como parte dos requisitos necessários para obtenção do grau de Mestre.

**ORIENTADOR: Rafael Ferreira Leite de Mello**

**CO-ORIENTADORA: Taciana Pontual da Rocha Falcão**

**RECIFE – PE**

**2022**

Dados Internacionais de Catalogação na Publicação  
Universidade Federal Rural de Pernambuco  
Sistema Integrado de Bibliotecas  
Gerada automaticamente, mediante os dados fornecidos pelo(a) autor(a)

---

- S729a Souza, Franciely Alves de  
Aplicação de Learning Analytics na Análise Automática de atividades de Introdução à Programação com o Scratch /  
Franciely Alves de Souza. - 2022.  
137 f. : il.
- Orientador: Rafael Ferreira Leite de Mello.  
Coorientadora: Taciana Pontual da Rocha Falcao.  
Inclui referências e anexo(s).
- Dissertação (Mestrado) - Universidade Federal Rural de Pernambuco, Programa de Pós-Graduação em Informática Aplicada, Recife, 2022.
1. Programação em Blocos. 2. Scratch e Dr. Scratch. 3. Avaliação Automática. 4. Pensamento Computacional. 5. Learning Analytics. I. Mello, Rafael Ferreira Leite de, orient. II. Falcao, Taciana Pontual da Rocha, coorient. III. Título

**FRANCIELY ALVES DE SOUZA**

**APLICAÇÃO DE LEARNING ANALYTICS NA ANÁLISE  
AUTOMÁTICA DE ATIVIDADES DE INTRODUÇÃO À  
PROGRAMAÇÃO COM O SCRATCH**

Dissertação submetida à Coordenação do Programa de Pós-Graduação em Informática Aplicada do Departamento de Estatística e Informática - DEINFO - Universidade Federal Rural de Pernambuco, como parte dos requisitos necessários para obtenção do grau de Mestre.

Aprovada em:

**BANCA EXAMINADORA**

---

Nome do orientador (Orientador) Universidade Departamento

---

Membro 2 da banca Universidade Departamento

---

Membro 3 da banca Universidade Departamento

Primeiramente a Deus por ter permitido que este trabalho acontecesse. A minha mãe que sempre me deu todo o apoio e me transmitiu valores que levarei para sempre, a minha irmã, meus amigos e meus orientadores que me apoiaram, incentivaram e acreditaram no meu potencial.

## Agradecimentos

Meu agradecimento ao meu orientador Prof. Dr. Rafael Ferreira, pela confiança e oportunidade de ter sido sua orientanda, e a minha co-orientadora Profa. Dra. Taciana Pontual por ter aceitado o convite de estar ao meu lado nesta caminhada, também pela paciência e sabedoria com que me conduziram e pelos conhecimentos compartilhados que contribuíram fortemente para o meu crescimento.

A minha família, em especial à minha mãe, Lucidalva Alves, por todo apoio, paciência, e por sempre ter me incentivado a sempre melhorar cada vez mais como profissional e como pessoa, por todo o amor e suporte dado para a concretização do mestrado.

A todos os meus amigos, em especial à minha amiga, Maria Renay, que sempre torceu por mim e esteve do meu lado ao longo de toda esta trajetória.

Ao meu professor da graduação, Ivaldir Junior, por todos os conselhos, incentivos e por acreditar que eu sou capaz.

Por último, agradeço a todos que contribuíram de alguma forma para a realização deste trabalho.

O sucesso é a soma de pequenos  
esforços repetidos dia após dia.

(Robert Collier)

# Resumo

O ensino de programação na escola é uma tendência impulsionada pelo desenvolvimento tecnológico e pela consolidação do pensamento computacional como habilidade fundamental para todas as pessoas. Linguagens de programação em blocos tornaram-se populares para ajudar a desenvolver o pensamento computacional das crianças, já que se trata de um desafio nas escolas. Este trabalho utiliza dados de uma experiência vivenciada com alunos do 2º ao 5º ano do ensino fundamental que, durante oito meses de aulas remotas, tiveram a oportunidade de aprender conceitos básicos de programação utilizando a ferramenta Scratch. As atividades eram contextualizadas através de storytelling e eram avaliadas de forma manual. Em vista disso, este trabalho tem como objetivo propor visualizações automáticas de avaliações de atividades, com a finalidade de dar suporte para os professores acompanharem o avanço do aprendizado dos alunos em atividades de introdução a programação com o Scratch, a fim de analisar quais conceitos do pensamento computacional estão presentes nas programações, levando em consideração que ainda não existe nenhum método de avaliação definido que possa ajudar os professores nesse processo. Técnicas de learning analytics foram utilizadas na intenção de gerar insumos para a avaliação, e neste trabalho as atividades foram avaliadas automaticamente por meio da ferramenta de código livre Dr. Scratch, que teve seu código modificado para ser acoplado a um banco de dados, com a finalidade de extrair os dados automaticamente e adicioná-los na ferramenta online Data Studio, que gerou visualizações gráficas automaticamente. Tais visualizações foram avaliadas por meio de um formulário que obteve 28 respostas, e alcançou participantes de diversos cursos, que analisaram o nível de compreensão das visualizações geradas. Os resultados demonstram que a partir da avaliação automática, foram geradas visualizações compreensíveis, onde o professor pode analisar de forma rápida e prática, em quais critérios os alunos possuem mais dificuldades, e consequentemente adaptar as aulas para suprir essas dificuldades.

**Palavras-chave:** Ensino fundamental, Scratch, Programação em Blocos, Dr. Scratch, Avaliação Automática, Pensamento Computacional, Learning Analytics.

# Abstract

Teaching programming at school is a trend driven by technological development and the consolidation of computational thinking as a fundamental skill for all people. Block programming languages have become popular to help develop children's computational thinking. This work uses data from an experience with students from the 2nd to the 6th year of elementary school who, during eight months of remote classes, had the opportunity to learn basic programming concepts using the Scratch tool. The activities were contextualized through storytelling and evaluated manually. In view of this, this work aims to propose automatic visualizations of evaluations of activities, in order to support teachers to monitor the progress of student learning in introductory activities to programming with Scratch, and learning analytics techniques were used. with the intention of generating inputs for the evaluation. In this work the activities were automatically evaluated using the free code tool Dr. Scratch, which had its code modified to be coupled to a database, in order to automatically extract the data and add it to the Data Studio online tool, which automatically generated graphical views. Such visualizations were evaluated through a form that obtained 28 responses, and reached participants of several courses, who analyzed the level of understanding of the generated visualizations. The results show that from the automatic evaluation, understandable views were generated, where the teacher can analyze quickly and practically, in which criteria the students have more difficulties, and consequently adapt the classes to overcome these difficulties.

**Keywords:** Elementary School, Scratch, Block Programming, Dr. Scratch, Automatic Assessment, Computational Thinking, Learning Analytics.

# Lista de Figuras

Figura 1 - Tela de programação de Scratch. Fonte: Scratch . . . . .	22
Figura 2 - Função de Compartilhamento no Scratch. Fonte: Scratch . . . . .	23
Figura 3 - Tela que disponibiliza o link do projeto compartilhado. Fonte: Scratch. . . . .	24
Figura 4 - Exemplo de Algoritmo. Fonte: Elaborada pela autora. . . . .	24
Figura 5 - Interface inicial da ferramenta Dr. Scratch. Fonte: Dr. Scratch. . . . .	25
Figura 6 - Pontuação dos conceitos avaliados pelo Dr. Scratch. Fonte: Dr. Scratch . . . . .	25
Figura 7 - Três definições de projetos apresentados pelo Dr. Scratch. Fonte: Dr. Scratch . . . .	27
Figura 8 - Funcionamento da estratégia de avaliação de atividades por (Eloy, 2019). . . . .	29
Figura 9 - Exemplo de storytelling utilizado nas aulas. Fonte: Scratch. . . . .	36
Figura 10 - Exemplo de atividade avaliada manualmente. Fonte: Elaborada pela autora. . . . .	36
Figura 11 - Visualização das tabelas do Dr. Scratch, por meio do BD. . . . .	39
Figura 12 - Visualização das pontuações avaliadas pelo Dr. Scratch, diretamente no BD. . . . .	40
Figura 13 - Exemplo de visualização e questionamento presente no formulário. . . . .	41
Figura 14 - Visualização do 1º Trimestre. . . . .	65
Figura 15 - Visualização do 2º Trimestre. . . . .	65
Figura 16 - Visualização do 3º Trimestre. . . . .	66
Figura 17 - Estilo de gráfico escolhido como melhor visualização. . . . .	67
Figura 18 - Visualização da análise por faltas. . . . .	68
Figura 19 - Visualização do 2º Ano. . . . .	69
Figura 20 - Visualização do 3º Ano. . . . .	71
Figura 21 - Visualização do 4º Ano. . . . .	73
Figura 22 - Visualização do 5º Ano. . . . .	73
Figura 23 - Estilo de gráfico escolhido como melhor visualização entre as quatro turmas. . . . .	75
Figura 24 - Visualização que mais apresentou atividades. . . . .	75
Figura 25 - Visualização referente ao 1º Aluno. . . . .	76
Figura 26 - Visualização referente ao 2º Aluno. . . . .	77
Figura 27 - Visualização referente ao 3º Aluno. . . . .	78
Figura 28 - Visualização referente ao 4º Aluno. . . . .	79
Figura 29 - Visualização referente ao 5º Aluno. . . . .	79
Figura 30 - Visualização referente ao 6º Aluno. . . . .	80
Figura 31 - Visualização referente ao 7º Aluno. . . . .	81

## Lista de Figuras

Figura 32 - Visualização referente ao 8º Aluno.....	82
Figura 33 - Visualização referente ao 9º Aluno.....	83
Figura 34 - Visualização referente ao 10º Aluno.....	83
Figura 35 - Visualização referente ao 11º Aluno.....	84
Figura 36 - Visualização referente ao 12º Aluno.....	85
Figura 37 - Visualização referente ao 13º Aluno.....	86
Figura 38 - Visualização referente ao 14º Aluno.....	87
Figura 39 - Visualização referente ao 15º Aluno.....	88
Figura 40 - Visualização referente ao 16º Aluno.....	89
Figura 41 - Determinado como melhor estilo de gráfico.....	90
Figura 42 - Visualização com muitas informações.....	91

## **Lista de Tabelas**

Tabela 1 - Conceitos avaliados pela ferramenta Dr. Scratch.Fonte:Dr. Scratch. ....	.27
Tabela 2 - Resumo dos principais dados dos trabalhos relacionados.....	32
Tabela 3 - Quantidade de alunos por turma.....	34
Tabela 4 - Cronograma utilizado para as aulas.....	34
Tabela 5 - Número de faltas de cada aluno no ano letivo. ....	61
Tabela 6 - Graduação dos participantes que responderam o formulário. ....	65

## Lista de Gráficos

Gráfico 1 - Dados dos alunos das duas turmas do 2º Ano	42
Gráfico 2 - Dados dos alunos do 3º Ano	43
Gráfico 3 - Dados dos alunos do 4º Ano	44
Gráfico 4 - Dados dos alunos do 5º Ano	45
Gráfico 5 - Dados do 1º Aluno	46
Gráfico 6 - Dados do 2º Aluno	47
Gráfico 7 - Dados do 3º Aluno	47
Gráfico 8 - Dados do 4º Aluno	48
Gráfico 9 - Dados do 5º Aluno	49
Gráfico 10 - Dados do 6º Aluno	49
Gráfico 11 - Dados do 7º Aluno	50
Gráfico 12 - Dados do 8º Aluno	51
Gráfico 13 - Dados do 9º Aluno	51
Gráfico 14 - Dados do 10º Aluno	52
Gráfico 15 - Dados do 11º Aluno	52
Gráfico 16 - Dados do 12º Aluno	53
Gráfico 17 - Dados do 13º Aluno	54
Gráfico 18 - Dados do 14º Aluno	54
Gráfico 19 - Dados do 15º Aluno	55
Gráfico 20 - Dados do 16º Aluno	55
Gráfico 21 - Dados do 1º Trimestre de Aula	56
Gráfico 22 - Dados do 2º Trimestre de Aula	57
Gráfico 23 - Dados do 3º Trimestre de Aula	58
Gráfico 24 - Dados de alunos com no mínimo quatro faltas	58
Gráfico 25 - Avaliação das atividades entregues	59
Gráfico 26 - Dados dos alunos com 0 a 2 faltas	62
Gráfico 27 - Dados dos alunos com 3 a 4 faltas	63
Gráfico 28 - Dados dos alunos com 5 a 6 faltas	64
Gráfico 29 - Respostas à pergunta: Qual conceito foi menos pontuado no primeiro trimestre?	65
Gráfico 30 - Respostas à pergunta: Qual conceito foi melhor pontuado no segundo trimestre?	66
Gráfico 31 - Respostas à pergunta: Quais conceitos mais pontuaram no terceiro trimestre?	67

## Lista de Gráficos

Gráfico 32 - Respostas à pergunta: Qual é o melhor estilo de gráfico?.....	68
Gráfico 33 - Respostas à pergunta: Qual conceito que se manteve com pontuação constante na maioria das atividades?.....	68
Gráfico 34 - Respostas à pergunta: Em apenas 5 atividades a pontuação de lógica não é 0?..	69
Gráfico 35 - Respostas à pergunta: Na análise do segundo ano, quais conceitos tiveram pontuação 2?.....	70
Gráfico 36 - Respostas à pergunta: Observando o gráfico do 2º ano, é verdade que 9 atividades não pontuaram no conceito sincronização?.....	70
Gráfico 37 - Respostas à pergunta: Observando o gráfico do 2º ano, é verdade que nenhuma atividade pontuou mais que 1, no conceito representação de dados?.....	71
Gráfico 38 - Respostas à pergunta: Na análise do 3º ano, quais conceitos não pontuaram em apenas 1 atividade?.....	72
Gráfico 39 - Respostas à pergunta: Nas atividades do 3º ano, é verdade que em apenas 1 atividade houve pontuação em lógica?.....	72
Gráfico 40 - Respostas à pergunta: Nas atividades do 4º ano, quais dos conceitos pontuou em um número maior de atividades?.....	73
Gráfico 41 - Respostas à pergunta: Nas atividades do 5º ano, qual conceito não houve pontuação em 6 atividades?.....	74
Gráfico 42 - Respostas à pergunta: Referente ao 5º ano, é verdade que apenas uma atividade não pontuou no conceito sincronização?.....	74
Gráfico 43 - Respostas à pergunta: Qual dos 4 estilos de gráficos ficou mais fácil de visualizar?.....	75
Gráfico 44 - Respostas à pergunta: Qual dos anos, teve o maior número de atividades avaliadas?.....	76
Gráfico 45 - Respostas à pergunta: Nas atividades do 1º aluno, qual pontuação está sendo representada na atividade 7.....	76
Gráfico 46 - Respostas à pergunta: Nas atividades do 2º aluno, é verdade que em pelo menos 1 atividade o conceito de controle de fluxo foi zero?.....	77
Gráfico 47 - Respostas à pergunta: Nas atividades do 3º aluno, quais conceitos tiveram a pontuação zerada na atividade 7?.....	78

## Lista de Gráficos

Gráfico 48 - Respostas à pergunta: Nas atividades do 4º aluno, é verdade que o maior número de atividades seguidas com pontuação zero no conceito de lógica é 3? .....	79
Gráfico 49 - Respostas à pergunta: Nas atividades do 5º aluno, quais conceitos estão presentes em todas as atividades? .....	80
Gráfico 50 - Respostas à pergunta: Nas atividades do 6º aluno, é verdade que nenhuma atividade pontuou nos 7 conceitos? .....	81
Gráfico 51 - Respostas à pergunta: Na análise dos dados do 7º aluno, quais conceitos estão presentes em todas as atividades? .....	81
Gráfico 52 - Respostas à pergunta: Na análise dos dados do 8º aluno, o que ocorre com os dados da atividade 4? .....	82
Gráfico 53 - Respostas à pergunta: Na análise dos dados do 9º aluno, é verdade que desde a primeira atividade o conceito de sincronização apenas progrediu na pontuação? .....	83
Gráfico 54 - Respostas à pergunta: Na análise dos dados do 10º aluno, qual conceito não foi pontuado em nenhuma atividade? .....	84
Gráfico 55 - Respostas à pergunta: Na análise dos dados do 11º aluno, qual resposta está correta? .....	85
Gráfico 56 - Respostas à pergunta: Na análise dos dados do 12º aluno, quantas atividades pontuaram no conceito abstração? .....	85
Gráfico 57 - Na análise dos dados do 13º aluno, quais conceitos pontuaram em todas as atividades? .....	86
Gráfico 58 - Respostas à pergunta: Na análise dos dados do 14º aluno, é verdade que em 4 atividades não houve pontuação zero em nenhum conceito? .....	87
Gráfico 59 - Respostas à pergunta: Na análise dos dados do 15º aluno, é verdade que pelo menos em uma atividade todos os conceitos tiveram pontuação zero? .....	88
Gráfico 60 - Respostas à pergunta: Na análise dos dados do 16º aluno, é verdade que a primeira pontuação para o conceito de sincronização foi na atividade 2 e não houve declínio menor do que a pontuação inicial? .....	89
Gráfico 61 - Respostas à pergunta: Na análise das pontuações, qual é o conceito que menos pontuou nas atividades? .....	89

# Lista de Siglas

LA	<i>Learning Analytics</i>
MIT	<i>Instituto de Tecnologia de Massachusetts</i>
URL	<i>Uniform Resource Locator</i>
PC	<i>Pensamento Computacional</i>
BD	<i>Banco de Dados</i>
API	<i>Application Programming Interface</i>
JSON	<i>JavaScript Object Notation</i>
CSV	<i>Comma-Separated-Values</i>
S.V	<i>Sem Visualizações</i>

# Sumário

<b>1</b>	<b>Introdução</b>	<b>14</b>
1.1	Pergunta de Pesquisa	16
1.2	Objetivos	16
1.2.1	Objetivo Geral	16
1.2.2	Objetivo Específico	16
1.3	Organização do Trabalho	16
<b>2</b>	<b>Fundamentação Teórica</b>	<b>18</b>
2.1	Pensamento Computacional	19
2.2	Programação para Crianças	20
2.3	Scratch	21
2.3.1	Dr. Scratch	25
2.4	Avaliação da Aprendizagem	27
2.4.1	Tipos de Avaliação	28
2.5	Learning Analytics	29
<b>3</b>	<b>Trabalhos Relacionados</b>	<b>30</b>
<b>4</b>	<b>Metodologia</b>	<b>34</b>
4.1	Realização das Aulas	34
4.1.1	Processo da Elaboração das Atividades	35
4.2	Análise manual das Atividades	36
4.3	Análise por meio do Dr. Scratch	37
4.4	Análise Automática dos Dados	38
4.5	Avaliação das Visualizações	40
<b>5</b>	<b>Proposta de Análise de Visualização</b>	<b>42</b>
<b>5.1</b>	<b>Proposta para Análise Automática das Atividades</b>	<b>42</b>
5.1.1	Análise por Turma	42
5.1.2	Análise por Aluno	45
5.1.3	Análise por Trimestre	56
5.1.4	Análise por Faltas	58
<b>6</b>	<b>Resultados</b>	<b>59</b>

# Sumário

6.1	Resultado da Análise Manual por meio do Dr. Scratch .....	59
6.2	Resultado da Análise dos Gráficos.....	64
6.3	Conclusões sobre a análise dos Gráficos .....	90
<b>7</b>	<b>Considerações Finais .....</b>	<b>91</b>
7.1	Limitações e Trabalhos Futuros .....	92
	<b>Referências .....</b>	<b>94</b>
	<b>ANEXOS .....</b>	<b>98</b>
	Anexo 1 .....	98
	Anexo 2 .....	108
	Anexo 3.....	109

# 1 Introdução

Os avanços das tecnologias computacionais impulsionam mudanças, transformando o mundo em que vivemos e possibilitando meios para adquirir diferentes informações. Tais tecnologias possibilitam aos estudantes vastas fontes de dados que podem ser utilizados para estudos sobre diferentes áreas, inclusive para obter conhecimentos que os tornem aptos a lidar com o uso dessas inovações. A tecnologia está levando a grandes mudanças na economia, na nossa forma de nos comunicarmos e nos relacionarmos com os outros, e cada vez mais no modo como aprendemos (BATES, 2017).

Dentre as inovações tecnológicas que regem o século XXI, está a popularização da habilidade de desenvolver programas computacionais por meio de algoritmos escritos em linguagens de programação. A área de programação é um dos pilares da profissionalização em informática e de grande importância nas matrizes curriculares (BINI e KOSCIANSKI, 2009). Desde a década de 90, uma das motivações para a introdução da computação ou programação na Educação Básica consiste na demanda crescente por profissionais com a capacidade de entender e produzir novas tecnologias digitais (MANYIKA, 2017). Entretanto, aprender programação é um processo complexo, que exige competências como raciocínio lógico e abstração (BITTENCOURT, 2013).

A aprendizagem de programação é um processo que requer muita dedicação e prática para obter bons resultados, e na educação básica é fundamental aplicar metodologias que possam engajar os alunos, facilitando a compreensão dos conceitos abordados e tornando as aulas mais atrativas. De acordo com Marcon Júnior e Boniatti (2015), é necessária a criação ou a utilização de metodologias diferenciadas que apresentem benefícios ao ensino-aprendizagem de programação, em especial, no desenvolvimento do raciocínio lógico. Neste sentido, é importante considerar a habilidade de Pensamento Computacional, vislumbrada por estudiosos como uma ferramenta que se tornará indispensável no futuro. De acordo com Wing (2006):

Esse tipo de pensamento será parte do conjunto de habilidades não somente de outros cientistas, mas de todas as pessoas. A computação ubíqua está para o hoje assim como o pensamento computacional está para o amanhã. A computação ubíqua era o sonho de ontem que se tornou a realidade de hoje; pensamento computacional é a realidade do amanhã. (WING, 2006, p.4).

Segundo Wing (2006), o PC é um modo de pensamento humano, que envolve diversas habilidades, o que requer diferentes níveis de abstração. O Pensamento Computacional aparece

como um dos três eixos da proposta da (SBC, 2017) para Computação na Educação Básica, junto com Mundo Digital e Cultura Digital.

Valente (2016) destaca que a necessidade de ter um desenvolvimento efetivo do PC na educação básica tem levado países a reformularem os currículos de ensino. Onde pretende-se buscar caminhos com o intuito de produzir o PC de forma clara e consistente no ensino para as crianças. Dentro deste contexto, Araújo, Andrade e Guerreiro (2016) mapearam a literatura brasileira a respeito do PC, a partir disso constataram que a abordagem mais empregada a fim de estimular o PC é a programação.

Neste contexto, as linguagens de programação em blocos foram desenvolvidas para estudantes iniciantes, com intuito de facilitar o aprendizado inicial de lógica de programação (MARJI, 2014). Em particular, a ferramenta de programação em blocos Scratch se tornou extremamente popular e tem sido usada com crianças mundialmente, inclusive fomentando a criatividade. Moran (2018) apresenta o Scratch como exemplo de recurso para aplicação de metodologia ativa para usuários iniciantes, podendo ser utilizada para a construção de jogos ou narrativas de histórias que podem ser contadas pelos próprios alunos, através de programação em blocos. Moretti (2019) afirma que a linguagem Scratch se constitui numa maneira de se desenvolver as habilidades características do pensamento computacional.

Por outro lado, ainda não existe um consenso sobre o processo para avaliar soluções expressas através de linguagens de programação em blocos. Muitas vezes, os professores elaboram seus próprios métodos manuais para avaliar se o aluno atendeu ou não ao objetivo das atividades. Nesse sentido, técnicas de learning analytics possuem o potencial de contribuir na análise de programações desenvolvidas na ferramenta Scratch, sendo um processo demorado quando realizado sem a utilização de técnicas automatizadas. De acordo com Blikstein *et al.* (2014), a utilização de técnicas de learning analytics em aulas introdutórias de programação de computadores, evidenciam o potencial de avaliação dos alunos em programação, analisando dados de sua trajetória e valorizando o processo de construção.

De acordo com Pelle (2018), a avaliação de uma programação desenvolvida por um aluno, se não for realizada automaticamente, o professor passa a realizar por meio da análise manual, verificando o código produzido. Análise de código é o processo de extrair informação sobre um programa através do código fonte (BINKLEY, 2007).

Para códigos desenvolvidos no Scratch, já existe uma ferramenta livre de código aberto, Dr. Scratch, que possibilita as avaliações de forma automática, ajudando os professores a identificarem pontos fracos como códigos mortos e variáveis não utilizadas; e também pontos fortes como a utilização de sete conceitos relacionados ao pensamento computacional (MORENO-LEON *et al.*, 2016). Tendo o pensamento computacional como objeto de avaliação, e fazendo uso das técnicas de learning analytics para gerar dados que facilite a análise dos professores.

Neste contexto, utilizar a ferramenta Dr. Scratch proporciona a obtenção de insumos para análises, já que a ferramenta disponibiliza dados importantes voltados para os conceitos do pensamento computacional, porém não conta com nenhuma função para a extração dos dados das avaliações. Após avaliar uma atividade, os resultados obtidos precisam ser extraídos de forma manual, onde o professor irá precisar observar os resultados e adicioná-los em algum documento por meio da escrita. A princípio, espera-se que a presente pesquisa e seus resultados contribuam para que avaliações de atividades do Scratch tenham seus dados extraídos e analisados de forma automatizada.

## **1.1 Pergunta de Pesquisa**

Quais formatos de visualizações de dados podem ser mais adequados para professores acompanharem o desempenho dos estudantes em atividades de programação com o Scratch?

## **1.2 Objetivos**

A seguir são apresentados o objetivo geral e os objetivos específicos que conduziram esta pesquisa. O objetivo geral define o propósito do estudo e os específicos caracterizam as etapas do projeto.

### **1.2.1 Objetivo Geral**

A presente pesquisa teve como objetivo geral propor visualizações de análise automática de avaliações referente a atividades de introdução à programação com Scratch, a fim de dar suporte para os professores identificarem o avanço do desempenho dos alunos.

### **1.2.2 Objetivos Específicos**

Visando cumprir o objetivo geral apresentado, foram definidos os seguintes objetivos específicos:

- Analisar métodos de avaliação utilizados atualmente em atividades de introdução à programação, incluindo o contexto do Scratch.
- Adaptar uma ferramenta para avaliação automática de atividades de programação feitas no Scratch.
- Definir tipos de visualizações para os dados coletados na avaliação automática.
- Propor uma estratégia automática para visualizar e analisar os dados.

### **1.3 Organização do Trabalho**

Este trabalho está organizado em 7 capítulos. O capítulo 2 apresenta a fundamentação teórica. Explicações sobre Learning Analytics, Programação para Crianças, Pensamento Computacional, Scratch e Dr. Scratch são encontradas neste capítulo. O capítulo 3 relata os trabalhos relacionados ao tema desta pesquisa, descrevendo suas metodologias e resultados. O capítulo 4 relata a metodologia utilizada, a coleta dos dados e como foram utilizados durante a análise. O capítulo 5 relata a proposta de análise dos dados. O capítulo 6 contém os resultados, discussões e conclusões da pesquisa. Já o capítulo 7 refere-se à visão geral da pesquisa abordando suas limitações e trabalhos futuros.

## 2 Fundamentação Teórica

Neste capítulo são discutidos os principais temas de interesse deste trabalho, que são: Learning Analytics, Programação para Crianças, Pensamento Computacional Scratch e Dr. Scratch.

### 2.1 Pensamento Computacional

Wing (2006) publicou um artigo na revista Communications of the ACM, onde ela descreve que considera o PC um conjunto de competências e habilidades atreladas à Ciência da Computação, habilidades estas que os estudantes deveriam incorporar desde os primeiros anos escolares. No artigo, a autora expõe o PC como uma forma de pensar com base em fundamentos da Ciência da Computação, com o objetivo de resolver problemas, desenvolver sistemas e entender o comportamento humano (WING, 2006).

Diversos autores se dedicam a compreender as competências para o desenvolvimento do PC, como CAVALCANTE, COSTA e ARAÚJO (2016):

O pensamento computacional poderá ser desenvolvido a partir de atividades de: Formular problemas de forma que nos permita usar um computador e outras ferramentas para ajudar a resolvê-los; Organizar e analisar dados logicamente; Representar dados através de abstrações tais como modelos e simulações; Propor soluções através do pensamento algorítmico; Identificar, analisar e implementar as soluções combinando recursos de forma eficiente e eficaz; Generalizar e transferir um processo de solução de um problema para outros. (CAVALCANTE, COSTA E ARAÚJO, 2016, p.3).

No ano de 2010, Wing passa a definir o termo PC como o método de pensamento presente na abstração de um problema, bem como suas possíveis soluções de modo que estas sejam representadas de uma forma que pode ser eficientemente executada por um agente processador de informações.

Barr e Stephenson (2011) apresentam o PC voltado para o uso em sala de aula considerando-o como uma metodologia para solucionar problemas, metodologia esta que pode ser automatizada e aplicada para resolver problemas de assuntos diversos, o que permite a aplicação do PC em várias áreas.

De acordo com a The Royal Society (2012) o PC é definido como o processo de identificação de elementos da computação no mundo que nos rodeia, e de aplicar ferramentas e técnicas da Ciência da Computação com o intuito de compreender e analisar sistemas e processos artificiais.

Pensamento Computacional é saber usar o computador como instrumento de aumento do poder cognitivo e operacional humano, aumentando a produtividade, inventividade e criatividade (FRANÇA *et al.*, 2012). O pensamento computacional e o raciocínio lógico devem ser desenvolvidos desde cedo, já que aumentam a capacidade de dedução e resolução de problemas (SICA, 2011). Neste contexto, o ensino de programação para crianças poderia desenvolver o pensamento computacional e passos lógicos para a resolução automatizada de problemas (KAFAI; BURKE, 2013).

## 2.2 Programação para Crianças

Aprender a programar não é útil apenas para quem quer ser um programador no futuro. De acordo com (OLIVEIRA, 2021), o ensino de programação não é uma prática restrita apenas à área da Computação ou a pessoas que se interessam em seguir na área de tecnologia, mas envolve a todos. Programar ajuda “[...] a desenvolver habilidades para auxiliar na resolução de problemas, ser um elemento ativo na construção do conhecimento e entender o contexto tecnológico em que vivemos” (ZANETTI; BORGES; RICARTE, 2016), além de desenvolver o raciocínio lógico e contribuir em outras matérias da grade escolar como, por exemplo, matemática e inglês (CODE, 2015). Para formar jovens criativos e inovadores na era digital, e não apenas consumidores de tecnologias, precisamos ensinar programação para que eles sejam capazes de construir softwares nessa nova era tecnológica (RODRIGUES, 2014).

Programar é usar ferramentas para criar um mundo digital e proporcionar melhorias a partir delas. Uma das alternativas de ensinar programação é por meio de jogos, sendo uma maneira em que a criança foge um pouco dos quadros e carteiras encontrados nas escolas, e acaba numa brincadeira lúdica no computador (CALEGARI *et al.*, 2015).

Quando estudam programação, as pessoas não só aprendem a programar, como também programam para aprender, noções de programação devem ser ensinadas desde cedo às crianças (SERRANO, 2014). As crianças podem se deparar com situações inesperadas e desafios, então é preciso prepará-las para soluções novas e inovadoras. Por isso a necessidade de desenvolver um pensamento criativo: é preciso pensar e agir, e quando uma criança aprende a programar está não apenas aprendendo o código mas também a decodificação, aprendendo a organizar suas ideias (CASTRO, 2017).

Entretanto, aprender a programar pode ser considerado como uma atividade complexa para pessoas iniciantes, e essa ação se torna ainda mais difícil quando se trata de crianças. Ao ensinar programação para crianças, é importante utilizar ferramentas que possam dinamizar o

processo de aprendizagem criativa. Um exemplo de ferramenta com interface voltada para crianças é o Scratch<sup>1</sup>, um projeto do grupo *Lifelong Kindergarten* no *Media Lab* do MIT (Instituto de Tecnologia de Massachusetts), idealizado por Mitchel Resnick. A ferramenta foi projetada para crianças, mas passou a ser utilizada por diferentes públicos, por se tratar de uma ferramenta prática e com diversas funcionalidades, trabalhando com programação baseada em blocos.

De acordo com Kelleher e Pausch (2005), a programação baseada em blocos procura abstrair a sintaxe de forma que os estudantes iniciantes mantenham seu foco apenas na semântica. Ela estimula o uso de estratégias para utilizar a computação criativa no ambiente educacional. Utilizar um ambiente de programação em blocos é fundamental para introduzir noções de linguagem de programação para usuários iniciantes, que não possuem conhecimento prévio sobre linguagem de programação.

Programação em blocos funciona a partir da ação de arrastar e soltar componentes, que se caracterizam como blocos. Na construção de um programa, vários blocos são encaixados uns aos outros como um quebra-cabeça, levando em consideração que cada bloco possui encaixes específicos, já que as linguagens de programação em blocos possibilitam que o usuário consiga visualizar onde cada bloco pode ser encaixado (embora em alguns casos, a depender da ferramenta, seja possível encaixar blocos de forma errônea) (WEINTROP *et al.*, 2017; KELLEHER e PAUSCH, 2005).

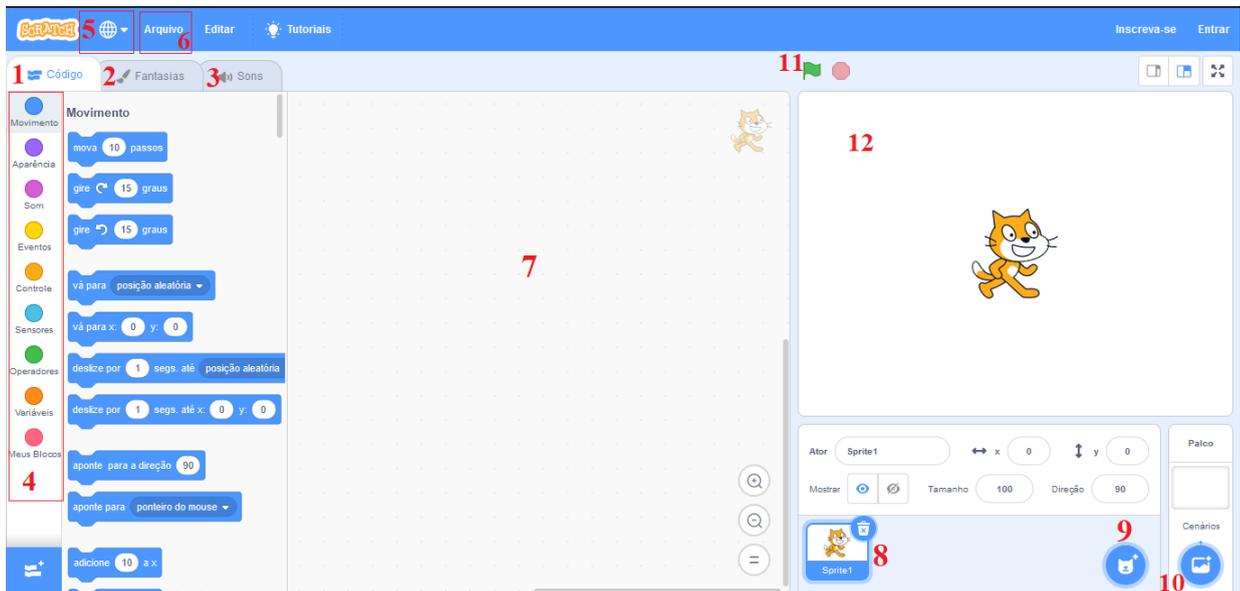
### **2.3 Scratch**

O Scratch é um ambiente de programação visual que possibilita aos usuários desenvolver várias competências como: raciocínio lógico, criatividade e resolução de problemas, além de estimular novas habilidades e ideias tecnológicas através de seus ambientes criativos (MALONEY *et al.*, 2010). Desenvolvido no Media Lab do Instituto de Tecnologia de Massachusetts, esse ambiente introdutório de ensino tem como objetivo facilitar a introdução de conceitos de matemática e computação (Scratch, 2015). A figura 1 mostra a tela de programação e as principais funções disponíveis no Scratch.

---

<sup>1</sup> <https://scratch.mit.edu/>

Figura 1 - Tela de programação de Scratch.



Fonte: Scratch

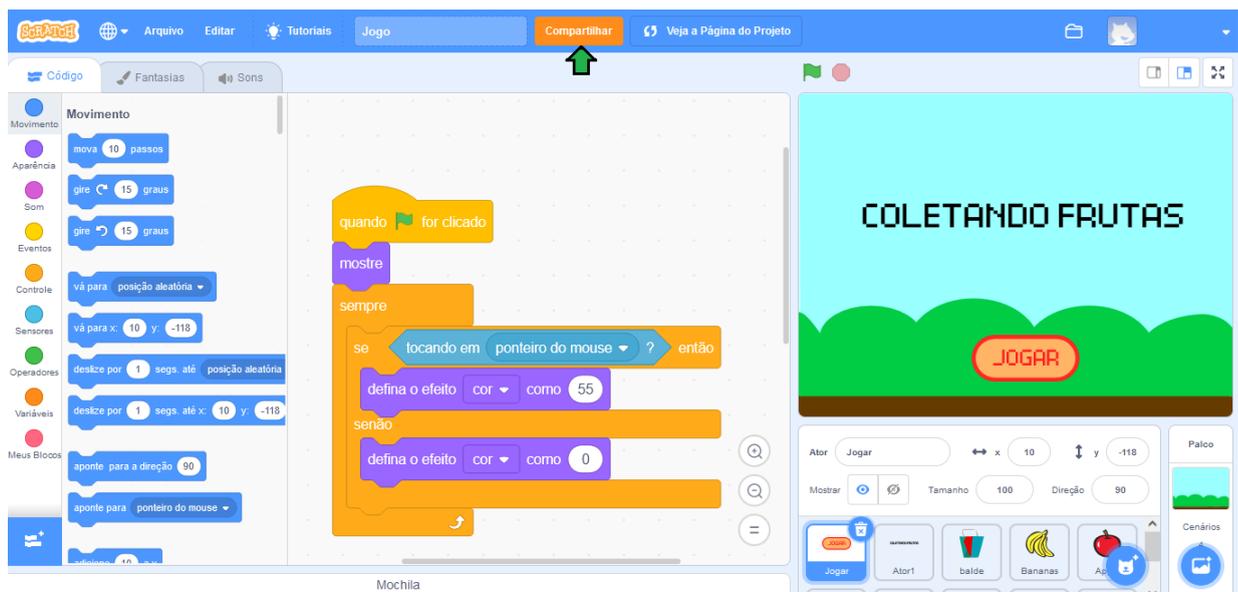
Observando a figura 1, existem algumas numerações indicativas, que destacam funções importantes:

1. Botão que dá acesso às instruções de código;
2. Botão disponível para visualizar e editar as fantasias dos personagens, e também desenhar novas fantasias;
3. Botão disponível para visualizar e editar os sons disponíveis na ferramenta, e também disponibiliza a opção de gravar novos sons;
4. Instruções de código, cada cor representa blocos diferentes;
5. Botão disponível para selecionar o idioma desejado;
6. Botão que possibilita a extração do projeto para o computador e a criação de um novo projeto;
7. Área de programação, onde os blocos precisam ser encaixados;
8. Representação de quais personagens estão no cenário;
9. Botão que dá acesso aos personagens da ferramenta;
10. Botão que dá acesso aos cenários da ferramenta;
11. Botão de execução dos programas;
12. Cenário.

Desde 2013, o Scratch está disponível de forma online, podendo ser acessado de qualquer lugar utilizando apenas um computador ou smartphone com acesso à internet. Todas as produções são salvas de forma automática na conta dos usuários, que pode ser criada apenas com o uso de um e-mail e a definição de uma senha de acesso. Também é possível efetuar o download gratuito da versão desktop para diversos sistemas operacionais.

No site do Scratch, o usuário pode encontrar diferentes versões disponíveis para download, sendo: a primeira versão Scratch 1.4, segunda versão Scratch 2.0, e a versão mais recente Scratch 3.0, que conta com novos blocos de comandos que permitem avanços na programação. A versão online possui um diferencial em comparação com às versões para download, já que ao finalizar um projeto, ele pode ser compartilhado através de um botão que a própria ferramenta disponibiliza, como mostrado pela seta verde na figura 2.

Figura 2 - Função de Compartilhamento no Scratch.



Fonte: Scratch

Com isso, pode-se compartilhar o link do projeto e outras pessoas podem acessar e visualizar qual foi a produção desenvolvida e quais blocos foram utilizados, conforme a figura 3.

Figura 3 - Tela que disponibiliza o link do projeto compartilhado.



Fonte: Scratch

No Scratch existem vários sprites e cenários diferentes que podem ser utilizados, e a programação ocorre pelo encaixe de blocos prontos, que são separados por cor. Cada ação possui sua cor, conforme apresentado no anexo 1.

Todas as categorias de comandos possuem blocos diferentes, que quando encaixados formam algoritmos computacionais. A figura 4 apresenta um exemplo de algoritmo, onde blocos de diferentes categorias foram encaixados para programar um botão com sensor de toque.

Figura 4 - Exemplo de Algoritmo.



Fonte: Elaborada pela autora.

A execução do algoritmo ocorre quando a bandeira verde é clicada, e sempre que o cursor de mouse estiver tocando o botão, o mesmo irá trocar de cor, pois, os blocos de “defina efeito”, estão programados para trocar a cor do botão de acordo com a numeração especificada. O número 55 possibilita que a cor original do botão seja alterada para verde, e se o cursor do mouse não estiver tocando no botão, nenhuma ação será realizada e a cor do botão continuará a mesma.

### **2.3.1 Dr. Scratch**

O Dr. Scratch<sup>2</sup> é uma ferramenta que pode ser utilizada por alunos e professores para avaliar projetos desenvolvidos no Scratch. A ferramenta foi desenvolvida por [Moreno-León et al. \(2015\)](#), que perceberam a falta de ferramentas que pudessem apoiar os professores no processo de avaliação dos projetos desenvolvidos no Scratch.

[Moreno-León et al. \(2015\)](#) destacam a ferramenta Dr. Scratch como primordial para automatizar a avaliação de projetos no Scratch e fornecer um feedback ao usuário, mostrando em que ponto o projeto precisa ser melhorado e apresentando informações de como desenvolver essas melhorias, com o objetivo de disseminar o pensamento computacional. Desta forma, a ferramenta visa aperfeiçoar o processo de avaliação que, na maioria das vezes, ocorre de forma manual, por meios desenvolvidos pelos próprios professores, e por vezes avaliando-se apenas a presença ou a ausência de estruturas de programação.

O Dr. Scratch é uma aplicação web de código livre, que consiste em analisar projetos a partir de sua URL (acrônimo para *Uniform Resource Locator*), ou através do arquivo do projeto no formato .sb3, que pode ser carregado na plataforma, como apresentado na figura 5.

---

<sup>2</sup> <http://www.drscratch.org/>

Figura 5 - Interface inicial da ferramenta Dr. Scratch.



Fonte: Dr. Scratch

Desta forma, a avaliação fornece uma análise quantitativa e sete conceitos avaliados, que estão descritos no quadro 1. Ao avaliar um projeto, será informado quais desses conceitos estão presentes na programação e a pontuação alcançada, numa escala de 0 a 3, como pode-se observar na figura 6.

Figura 6 - Pontuação dos conceitos avaliados pelo Dr. Scratch.



Fonte: Dr. Scratch

Somando as pontuações em cada conceito, é obtida a nota final do projeto, conforme a figura 7.

Figura 7 - Três definições de projetos apresentados pelo Dr. Scratch.



Fonte: Dr. Scratch

A pontuação máxima é 21, onde projetos com notas de até 7 pontos são considerados simples, entre 8 e 14 pontos são considerados como satisfatórios, e acima de 15 pontos são considerados como ótimos (MORENO-LEON *et al.*, 2016).

#### Quadro 1. Conceitos avaliados pela ferramenta Dr. Scratch.

Fonte: Dr. Scratch

CONCEITOS AVALIADOS PELO DR. SCRATCH	
Lógica	Instruções sobre pensamento lógico podem tornar os projetos dinâmicos, de modo a se comportarem de forma diferente dependendo da situação.
Representação de Dados	Cada projeto Scratch precisa de um conjunto de informações sobre os personagens para funcionar corretamente. Por exemplo, a posição de cada personagem, a direção que está apontando, tamanho, entre outros. Além disso, programadores podem criar novos repositórios de informação para armazenar dados como o nível do jogo, o tempo decorrido, a classificação, as vidas e as recompensas coletadas.
Paralelismo	É a possibilidade de que várias coisas ocorram simultaneamente. Por exemplo, um personagem faz uma ação enquanto outro está fazendo outra coisa.
Sincronização	Estas instruções permitem que o programador organize a ordem em que as

	ações vão ocorrer.
Interatividade com o usuário	Essas instruções podem tornar os projetos Scratch mais interativos, por exemplo, usar o teclado ou mouse para mover um personagem, responder perguntas ou usar a webcam, entre outras possibilidades.
Controle de fluxo	Essas instruções ajudam a controlar o comportamento dos personagens, fazendo, por exemplo, com que alguns blocos repitam funções um número de vezes determinado, até que uma situação ocorra.
Abstração	A habilidade de abstrair e decompor problemas permite a quebrar um problema em partes menores, mais fáceis de entender, programar e depurar (corrigir os erros).

Moreno-León e Robles (2016) não transmitem em seus trabalhos uma definição formal a respeito dos sete conceitos que propõem, porém podemos relacionar alguns destes conceitos às habilidades do pensamento computacional (PC), conforme as concepções apresentadas por Silva (2019), e que são as mais apropriadas para este trabalho.

Com isso, o conceito de Abstração pode ser relacionado com a habilidade de abstração e decomposição de problemas, que são definidos por Silva (2019) como “diminuir a complexidade do problema para poder identificar o elemento principal” é "separar uma tarefa em partes menores e gerenciáveis”.

Referente à habilidade de Paralelismo, que de acordo com Silva (2019) se define como “organizar recursos com o fim de realizar tarefas simultaneamente com o intuito de alcançar um objetivo comum”, é diretamente relacionada ao conceito de Paralelismo no Dr. Scratch.

Por fim, pode-se associar o conceito controle de fluxo do Dr. Scratch com a habilidade de algoritmos, que de acordo com Silva (2019) corresponde a “definir um conjunto de passos para resolver um problema”; e à habilidade automação, expressa pela mesma autora como “fazer uso de computadores para execução de tarefas repetitivas”.

Nesse sentido, métodos de análise automatizada de criações feitas no Scratch podem ser entendidos como uma aproximação ao campo de learning analytics (ELOY, 2019), fornecendo dados sobre o desempenho dos estudantes.

## 2.4 Avaliação da Aprendizagem

No ensino de programação com a ferramenta Scratch, tem-se um grande desafio, já que não existe nenhum método definido para avaliar a aprendizagem obtida por meio de programações geradas na ferramenta. Pois, a partir da avaliação é possível que o educador perceba as aprendizagens que os alunos já desenvolveram e quais dificuldades ainda persistem. Isto é fundamental para ajudar a encontrar mecanismos necessários para auxiliá-los diante das

dificuldades. O processo avaliativo implica em um posicionamento positivo ou negativo, diante do objeto avaliado, sendo assim a avaliação da aprendizagem deve buscar levantar dados, analisá-los e sintetizá-los, possibilitando o diagnóstico dos fatores que interferem no resultado da aprendizagem do aluno (LUCKESI, 2014).

Para Gatti (2009), a avaliação deve ser feita de maneira contínua de modo a acompanhar o desenvolvimento e o processo de aprendizagem do aluno e para isso é necessário que os professores estejam capacitados e aptos a elaborar instrumentos de avaliação condizentes com o trabalho realizado em sala de aula. Pois, o principal objetivo da avaliação é o aperfeiçoamento do ensino, um instrumento eficaz e valioso quando utilizado com o propósito de analisar e compreender o processo de aprendizagem (SILVA; CRUZ; OLIVEIRA, 2016).

### **2.4.1 Tipos de Avaliação**

A avaliação somativa ou classificatória, de acordo com Luckesi (2005); Haydt (2004) e Martins (1993) tem a função de classificar os alunos, no final do período ou ano letivo, selecionando as aprendizagens alcançadas pelo aluno de acordo com o aproveitamento nesse período de tempo determinado, com a finalidade de sua aprovação ou reprovação.

De acordo com Hoffman (2012) e Perrenoud (1999), a avaliação mediadora está ligada ao processo de aprendizagem dos alunos. Com esse contexto, surge a discussão sobre qual a forma mais apropriada de avaliar um aluno.

A avaliação diagnóstica permite ao professor localizar o foco do erro ou do acerto do aluno, podendo analisar em que etapa do processo de construção de conhecimento ele se encontra, adaptando as estratégias de ensino às suas necessidades (NASCIMENTO, 2009).

Já a avaliação formativa auxilia o aluno a aprender e a se desenvolver, colaborando para a regulação das aprendizagens e do desenvolvimento no sentido de um projeto educativo (PERRENOUD, 1999).

No contexto de buscar dados e analisá-los para realizar uma avaliação, Dawson et al. (2014, p. 231) citam que Learning Analytics (LA) trata da “coleta, análise e relato de dados sobre os alunos e seus contextos, com o objetivo de compreender e otimizar o aprendizado e os ambientes em que ele ocorre”. Neste trabalho, foram utilizadas técnicas de LA, onde os dados foram gerados pela ferramenta Dr. Scratch e coletados para a realização de visualizações gráficas, a fim de possibilitar maior visibilidade dos dados para o professor. Desta forma, o objetivo é que a partir de tais visualizações os professores possam realizar uma avaliação diagnóstica dos alunos, analisando em quais conceitos do PC os alunos poderiam obter maiores

pontuações, e identificar as possíveis dificuldades para adaptar suas metodologias.

## 2.5 Learning Analytics

Learning Analytics é uma área que está relacionada à interpretação, detalhamento e descrição de dados e utiliza técnicas de análises de dados, como análise estatística exploratória e até técnicas de mineração de dados (SILVA, 2017). E segundo Siemens *et al.*, (2011), LA visa a análise dos dados de alunos para um melhor entendimento do processo de ensino-aprendizagem. Pois, LA podem contribuir tanto para avaliação da aprendizagem, como também buscar meios para a sua melhoria. Em palestra proferida no LAK — 1st International Conference on Learning Analytics and Knowledge, LA foram definidas por (FERGUSON, 2012), como sendo medida, coleta, análise e relatório dos dados dos alunos bem como seus contextos visando, primeiro, entender e, em seguida, otimizar o aprendizado e o ambiente em que se desenvolve. Reforçando esse conceito, a proposta de LA é identificar, por meio da extração e análise de dados de forma automática, perfis de alunos, problemas ou padrões com um determinado conteúdo programático, reconhecimento entre o sucesso e insucesso de trajetórias de aprendizagem, detecção de lições complexas, entre outras (JOVANOVIC *et al.*, 2008).

Baker e Inventado (2014) complementam a definição de learning analytics ao enfatizar o interesse na análise de dados educacionais para ampliar as possibilidades de interpretação da aprendizagem, empoderando aprendizes e mediadores de informações que os auxiliem a tomar melhores decisões.

Sua importância se dá pela possibilidade de coleta de dados de navegação, o que possibilita múltiplas análises relacionadas a questões comportamentais e de aprendizagem do educando. Compreende-se por dados, as informações inseridas nos programas de forma classificada e ordenada, para que assim, seja possível fazer o processamento das mesmas de acordo com o objetivo pretendido (GIRAFFA, 2015).

Assim, a partir da análise gráfica de todos os dados coletados é possível perceber a evolução do desempenho do aluno, bem como seu progresso, suas dificuldades, sua participação, entre outros fatores (FARIA, 2014). De acordo com Clow (2012), LA é composto por: (1) Alunos, (2) Dados, (3) Análises e (4) Intervenções. Nesse ciclo, a partir da geração de dados pelas interações dos alunos é possível realizar diversas análises, propiciando informações acerca do processo ensino-aprendizagem e, conseqüentemente, favorecendo a tomada de

decisão quanto às intervenções necessárias.

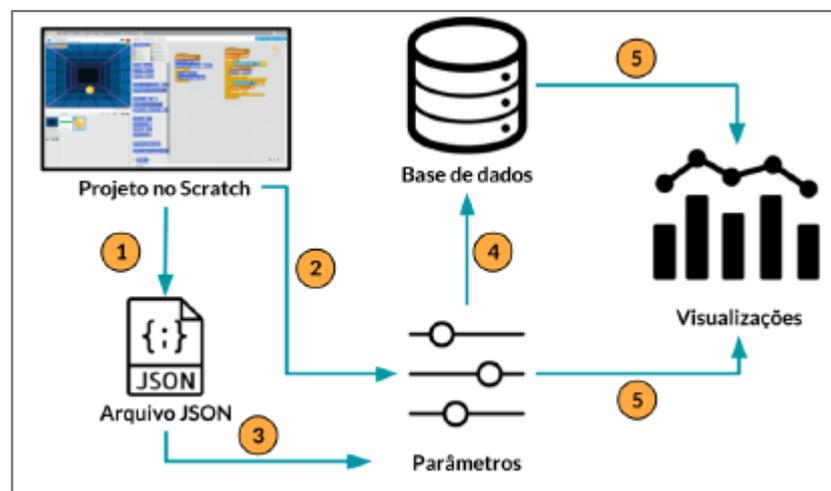
Este processo de análise é muito positivo pois os professores podem pensar em ações de melhorias para serem implementadas nas aulas, a fim de suprir as dificuldades apresentadas. E com o tempo, é possível criar um panorama detalhado da evolução de cada aluno, o que possibilita observar seu desempenho em diversas áreas de conhecimento, em comparação com outros da mesma turma e instituição (DIAS, 2017).

### 3 Trabalhos Relacionados

Esta seção é composta por alguns trabalhos acadêmicos relacionados ao tema da dissertação, que foram selecionados com foco nas ferramentas utilizadas no processo de ensino e aprendizagem e também no processo de avaliação, visando verificar qual ferramenta de programação foi utilizada em suas metodologias, a fim de gerar dados para avaliar o desenvolvimento do aprendizado de programação dos alunos, e conseqüentemente proporcionar um feedback para os professores.

O primeiro trabalho a ser mencionado é o de Eloy (2019). Este trabalho se relaciona bastante com o presente trabalho, uma vez que ele utiliza learning analytics na avaliação de atividades do Scratch e aplicação de questionários para coleta de dados. Na pesquisa relacionada, técnicas learning analytics são utilizadas para coletar e organizar dados, na intenção de verificar se as produções realizadas na ferramenta Scratch estão sendo relevantes para o desenvolvimento do conhecimento e aprendizado de programação. A proposta para a avaliação das atividades de introdução à programação está presente na figura 8.

Figura 8 - Funcionamento da estratégia de avaliação de atividades por (Eloy, 2019).



O autor determina que para a avaliação ocorrer, uma URL do projeto Scratch precisa ser utilizada, ou um arquivo Scratch exportado. As informações dos projetos são coletadas por uma API (acrônimo para *Application Programming Interface*) de código Python, que armazena todos os dados na base de dados e em seguida gera as visualizações necessárias para a análise e debates. De acordo com o autor, análises automatizadas das atividades, podem contribuir para uma melhor compreensão das necessidades de aprendizagem, além de conseguir observar graficamente o processo de aprendizagem percorrido pelos alunos. Porém, ao realizar algumas

etapas com foco na pesquisa, sendo: levantamentos bibliográficos, revisões sistemáticas e entrevistas com professores, o autor consegue concluir que outros fatores, principalmente a visão humana, são imprescindíveis para uma visão ampla do processo de desenvolvimento dos alunos. Desta forma deixando claro que as avaliações automáticas podem contribuir fortemente para uma análise mais robusta, onde pode-se obter e armazenar registros de aprendizados de cada aluno.

O trabalho de [Araújo et al. \(2020\)](#) também se assemelha a este trabalho por avaliar atividades Scratch, tendo a intenção de verificar se habilidades do PC são desenvolvidas nas programações. Para o processo de avaliação, foi construída uma ontologia de nome “OntoScratch”, que é dividida em classes que representam o projeto Scratch e classes que representam a avaliação do PC a ser realizada, com a finalidade de avaliar as habilidades do PC no projetos Scratch e acompanhar o progresso dos alunos nas programações. A cada novo bloco adicionado na programação, a ontologia Scratch cresce, pois possibilita a criação de novas classes. Para verificar a eficiência da ontologia na avaliação de habilidades do PC, as mesmas atividades que foram avaliadas na OntoScratch, foram submetidas a uma avaliação no Dr. Scratch, o que resultou em dados equivalentes. Consequentemente provando que a representação obtida pela OntoScratch se mostra eficaz, e também permite armazenar de maneira estruturada o percurso dos alunos durante a realização dos projetos no Scratch, visando analisar os processos de aprendizagens relacionadas ao PC.

A pesquisa de [AIVALOGLOU et al. 2017](#)) relata avaliações de projetos Scratch a partir da análise do código fonte. Os trabalhos avaliados foram extraídos da própria ferramenta Scratch web, sendo selecionados apenas novos projetos que foram compartilhados mais recentemente. Após a seleção, os projetos foram analisados a partir de arquivo JSON (acrônimo para *JavaScript Object Notation*), que resultou em quais blocos cada usuário utilizou em sua programação. Após a análise dos blocos utilizados, os projetos foram analisados na ferramenta Dr. Scratch, onde o código-fonte foi inspecionado e pontuações foram atribuídas, algo que possui uma grande semelhança com esse trabalho. Na análise mostrada pelo Dr Scratch, além de obter as pontuações referentes aos conceitos do pensamento computacional, os autores buscaram verificar a quantidade de clonagem de código, que representa quantas vezes o mesmo código foi repetido. Ao finalizar todas as análises, os dados foram armazenados em um banco de dados (BD), podendo ser identificados a partir de um ID.

Este trabalho destaca quais descrições de cada projeto o BD armazena, sendo elas: p\_id

(ID do projeto Scratch), nome-do-projeto (nome dado ao projeto), nome de usuário (nome de usuário do Scratch do autor), total-views (número de visualizações do projeto), total-remixes (número de remixes do projeto), total de favoritos (total de usuários favoritos), total-loves (usuários 'amando' o projeto), is-remix (de valor “true” se o projeto é um remix de outro), além das pontuações disponibilizadas pela ferramenta Dr. Scratch. Todas essas análises foram realizadas pelos autores, com o intuito de disponibilizar o BD, com todas as diversas informações sobre os projetos Scratch, com a finalidade de poder ajudar pesquisadores que pesquisam sobre temas como a avaliação das habilidades de programação, projetos de programadores iniciantes e progressões de aprendizagem em programação.

Outro trabalho que pode ser referenciado, que possui muitas semelhanças com a presente pesquisa, é o trabalho de Hoover *et al.* (2016), que relata a análise automática de jogos desenhados por meninas no Scratch, a fim de avaliar se o PC está inserido no contexto do jogo. Os desenhos dos jogos foram produzidos durante um workshop que utilizou o Scratch web, os desenhos tiveram a temática de mudanças climáticas. Este trabalho avaliou os projetos de duas maneiras, primeiro fazendo uso do Dr. Scratch, com intuito de verificar as pontuações voltadas para os sete conceitos do PC. A segunda avaliação foi uma análise qualitativa de todos os projetos, buscando verificar quais interfaces de jogos possuíam melhor interação com o usuário, narrativas e instruções mais elaboradas. Com os dois tipos de avaliações foi possível ter visões bem distintas, onde a análise qualitativa aponta claramente que os conceitos do PC são mais presentes em projetos com interfaces mais elaboradas. Já a avaliação com o Dr. Scratch foram negativas, pois blocos importantes que são avaliados pela ferramenta não estavam sendo utilizados pelos alunos na construção das interfaces, com isso as pontuações ficaram abaixo do esperado.

Com foco em se ter avaliações justas para as interfaces dos jogos, mesmo não fazendo uso de alguns blocos importantes, os autores implementaram um algoritmo que calcula conhecimentos básicos, e integraram ao Dr. Scratch, algoritmo esse que não depende de utilização de muitos blocos para apresentar uma pontuação, pois o intuito é avaliar as interfaces. Com isso, os autores concluíram que o Dr. Scratch necessita da inclusão de novas métricas, para realizar uma boa avaliação não só na programação, mas também no design produzido pelos alunos, já que ao fazer a utilização do Dr. Scratch sem nenhuma modificação, os resultados não foram relevantes para a pesquisa, pois as pontuações referentes aos projetos seriam melhoradas apenas com a implementação de um código mais robusto no Scratch, não

focando nas interfaces produzidas, sendo esse o foco da pesquisa.

A pesquisa de Niemela, (2017) pode ser referenciada por citar a importância da utilização de uma linguagem visual no processo de aprendizagem de programação para programadores iniciantes, com o Scratch como ferramenta utilizada em um curso de programação, e o Dr. Scratch como ferramenta de avaliação. O curso foi planejado para motivar alunos a estudarem programação. Dez alunos conseguiram iniciar o aprendizado de programação na ferramenta Scratch, e foram avançando de acordo com o nível de aprendizado, que foi analisado a partir dos conceitos do Dr. Scratch. A partir do momento que os alunos demonstraram progresso nos conceitos do PC, foram orientados a sair da programação em blocos e focar o aprendizado em linguagens textuais como Python. Diante disto, o Dr. Scratch foi utilizado para monitorar o avanço da aprendizagem dos alunos, evidenciando a importância de iniciar com uma linguagem em blocos para posteriormente desenvolver as mesmas programações em outras linguagens.

Observando a tabela 1, pode-se visualizar as principais características dos trabalhos referenciados. Os trabalhos que não propõem visualizações, receberam a nomenclatura S.V (Sem Visualizações).

Tabela 1 - Resumo dos principais dados dos trabalhos relacionados.

<b>Ano</b>	<b>Título do Trabalho</b>	<b>Autores</b>	<b>Metodologia</b>	<b>Ferramentas/ Avaliações</b>	<b>Visualização proposta.</b>
2016	Assessing Computational Thinking in Students' Game Designs	Hoover et al.	Análise de projetos Scratch.	Dr. Scratch e análise qualitativa.	S.V
2017	A Dataset of Scratch Programs: Scraped, Shaped and Scored	Aivaloglou et al.	Análise de projetos Scratch.	Dr. Scratch	S.V
2017	All rosy in scratch lessons: No bugs but guts with visual programming.	Niemela	Análise de projetos Scratch.	Dr. Scratch	S.V

2019	Contribuições para aplicação de learning analytics no apoio à avaliação em atividades de introdução à programação com Scratch.	Eloy	Levantamentos bibliográficos, revisões sistemáticas, entrevistas com professores e análise de projetos Scratch.	Código fonte extraído por API de código Python	S.V
2020	OntoScratch: ontologias para a avaliação do ensino de Pensamento Computacional através do Scratch.	Araújo et al.	Análise de projetos Scratch.	OntoScratch e Dr. Scratch	A visualização ocorre por meio da ontologia.

O presente trabalho difere dos demais por apresentar uma versão do Dr. Scratch modificada, instalada localmente, em que os dados extraídos são automaticamente adicionados a um BD, onde a análise das atividades de cada aluno fica armazenada, podendo ser acessada a qualquer momento, não sendo necessário que a mesma atividade seja reavaliada caso o professor necessite visualizar estes dados em outro momento. Desta forma, o professor também obtém a liberdade de extrair os dados em formato CSV (*Comma-Separated-Values*), na intenção de gerar uma visualização gráfica, o que até então nenhum trabalho referenciado utilizou.

Sendo assim, pode-se constatar pelos trabalhos relacionados que, embora utilizem a mesma ferramenta de análise, os meios para obter e visualizar os dados não abordaram as mesmas técnicas. Sendo os dados dessa pesquisa, algo inédito em relação aos trabalhos citados. Outra justificativa que deve ser mencionada é que existem poucos trabalhos brasileiros com foco nessa pesquisa, tanto que a maioria dos trabalhos citados não faz parte da literatura brasileira.

## 4 Metodologia

Esta seção descreve a metodologia utilizada para a coleta e análise dos dados para a pesquisa.

### 4.1 Realização das Aulas

Os dados utilizados nesta pesquisa foram coletados no período de maio a dezembro do ano de 2020, nas aulas de introdução à programação que ocorreram remotamente devido ao período de isolamento decorrente da pandemia do covid-19. As aulas foram ministradas em cinco turmas do ensino fundamental em uma escola privada do estado de Pernambuco, onde existia o clubinho de programação, e os pais poderiam efetuar a matrícula dos alunos interessados. As turmas eram de 2º, 3º, 4º e 5º ano.

A autora deste trabalho é graduada em Licenciatura em Computação, e atuou como professora em todas as turmas citadas, com isso conseguiu coletar os dados de suas próprias atividades aplicadas a cada turma. Os dados utilizados para adquirir os resultados tanto na análise manual com o Dr. Scratch, como na análise automática (seções 4.2 a 4.4), foram liberados pela escola a partir da assinatura da diretora no documento presente no anexo 2, onde é informado que todas as informações serão mantidas em anonimato. A tabela 2 mostra a quantidade de alunos por turma.

Tabela 2 - Quantidade de alunos por turma.

<b>Ano</b>	<b>Quantidade de Alunos</b>
2º	28 Alunos(as) distribuídos em duas turmas.
3º	15 Alunos(as)
4º	13 Alunos(as)
5º	20 Alunos(as)
<b>Total</b>	<b>76 Alunos(as)</b>

As aulas ocorriam uma vez por semana, de forma remota, por meio da plataforma Zoom. Nas aulas, os alunos utilizavam a versão online da ferramenta Scratch, onde podiam realizar as atividades e compartilhá-las para a correção. As atividades sempre seguiam o mesmo padrão semanal, conforme tabela 3.

Tabela 3. Cronograma utilizado para as aulas.

<b>CRONOGRAMA DE ATIVIDADES</b>	
1- Vídeo com explicação da atividade por meio de storytelling. 2- Disponibilização da atividade por meio de arquivo PDF anexado no aplicativo da escola. 3- Aula via plataforma Zoom.	Quarta-feiras
4- Liberação do formulário da semana, por meio de google forms.	Após a finalização da aula.
5- Último dia para responder o formulário, desta forma fazendo a entrega da atividade concluída.	Segunda-feiras

A cada atividade semanal finalizada, um formulário era respondido pelos alunos, onde era adicionado o link de compartilhamento do projeto, para as atividades serem acessadas e avaliadas.

#### **4.1.1 Processo de Elaboração das Atividades**

As atividades eram propostas aos alunos de forma contextualizada por meio de storytelling (contação de histórias). No contexto desse trabalho, storytelling é a contação de histórias com o auxílio de ferramentas digitais com o propósito de obter a atenção do público (DOMINGOS, 2009; PALACIOS; TERENCE, 2016). O uso de storytelling no ensino e aprendizagem pode proporcionar momentos lúdicos e engajadores, onde a contrapartida para obter a solução de um problema é visualizar a história e abstrair tudo que não se faz necessário para a resolução de um determinado problema. Segundo Chung (2006), o ato de narrar histórias contribui para uma melhor compreensão de ideias complexas, assim como de conceitos e informações.

As atividades além de serem apresentadas durante a aula remota, também eram apresentadas por meio de vídeos, onde eram contextualizadas através de uma narração de história (storytelling) com apresentação de imagens. Pois, nas primeiras aulas que as narrativas passaram a ser utilizadas, observou-se que os alunos apresentaram mais entusiasmo em assistir às aulas e participar dos momentos de discussões, então percebe-se que o storytelling contribuiu para as aulas se tornarem mais lúdicas e atrativas em todas as turmas. E por haver apenas uma aula por semana, aqueles alunos que eventualmente faltassem poderiam sentir dificuldades na resolução da atividade, mesmo seguindo as instruções disponibilizadas no

arquivo PDF, então poderiam assistir a vídeo aula e verificar toda a explicação da atividade por meio do storytelling, este foi o meio encontrada para proporcionar para os alunos um aula mais divertida e de fácil compreensão. A figura 9 apresenta um exemplo de atividade, onde o desafio foi construir um cenário e desenvolver uma programação onde o personagem estivesse praticando esporte. Na programação, a bola precisaria ter seus movimentos programados para chegar até a cesta, e os pontos deveriam ser armazenados em uma variável.

Figura 9 - Exemplo de storytelling utilizado nas aulas.



Fonte: Scratch.

Os storytelling eram utilizados na intenção de possibilitar a imersão em um mundo lúdico, e proporcionar uma melhor compreensão da atividade. A partir da história narrada, os alunos precisavam observar a problemática apresentada e resolvê-la.

#### **4.2 Análise manual das Atividades**

A professora avaliava as atividades de forma manual, nenhuma ferramenta era utilizada neste processo. A avaliação ocorria de acordo com o projeto entregue, observando se atendia aos requisitos solicitados na atividade. Um exemplo está apresentado na figura 10.

Figura 10 - Exemplo de atividade avaliada manualmente.



**ATIVIDADE**

Tocar música é muito divertido e alegre, na escola onde você estuda tem aula prática de música?

Na atividade de hoje você terá o desafio de construir um programa, onde um ator esteja tocando algum instrumento musical.

Fonte: Elaborada pela autora.

Na atividade apresentada na figura 10, os alunos necessitavam programar um personagem tocando um instrumento musical (o instrumento poderia ser escolhido pelo aluno).

As dicas para resolver a atividade eram:

- Utilize 2 atores;
- Você consegue resolver o problema utilizando 4 categorias de comandos, sendo elas: Eventos, aparência, controle e som;

Após as atividades serem entregues, a avaliação ocorria a partir da análise do projeto por completo, desde o design à programação, os pontos avaliados eram:

- Atores utilizados;
- Os atores coincidem com o cenário utilizado;
- Tipos de blocos utilizados;
- Quantidade de blocos utilizados;
- A execução do projeto, verificando se a programação consegue realizar as ações necessárias.

Na avaliação manual o foco se concentra em analisar a resolução do problema, onde os alunos precisavam pensar para encontrar uma ou mais soluções, independente de qual fosse o formato final da solução.

### 4.3 Análise por meio do Dr. Scratch

Antes mesmo de iniciar esta pesquisa, a autora já atuava com as turmas fazendo uso da

ferramenta Scratch, coletando dados e os analisando manualmente, com isso, se deparou com a dificuldade em realizar as avaliações, já que demandava muito tempo analisar diversas atividades e verificar se as programações possuíam blocos voltados para os conceitos do PC. Com isso, a estratégia utilizada para a avaliação de forma automática foi colocada em prática após os alunos desenvolverem as atividades, onde passou-se a utilizar a ferramenta Dr. Scratch na avaliação das atividades.

Ao longo das experiências de avaliação manual de atividades produzidas no Scratch, a autora se deparou com a necessidade de utilizar uma ferramenta que pudesse facilitar a correção de atividades com foco em 7 conceitos do PC. Com a ferramenta, a análise do código produzido se tornou mais fácil, por conseguir visualizar quais conceitos do PC estavam presentes nas programações. Com a avaliação das atividades por meio do Dr. Scratch, obtiveram-se resultados a partir de uma análise manual, onde a ferramenta apresenta as pontuações obtidas em cada atividade e esses dados precisam ser extraídos de forma manual. No momento que a ferramenta disponibiliza as pontuações referentes a cada atividade, é preciso copiar os resultados e anexar em uma planilha externa, que não tem nenhuma conexão com a ferramenta, para então conseguir gerar as visualizações e gráficos. Desta forma, o processo de extrair os dados e gerar as visualizações é completamente manual. Este foi o primeiro estudo realizado, cujos resultados são apresentados na subseção 6.1.

Para gerar tais resultados, foram utilizadas atividades de 17 alunos de turmas distintas, que entregaram a mesma quantidade de atividades, no mesmo período de entrega. Passaram pela avaliação 6 atividades de cada aluno. Foram selecionadas 2 atividades do início das aulas, 2 atividades no período intermediário e 2 atividades do fim das aulas, somando um total de 102 atividades avaliadas.

As atividades foram realizadas durante aulas remotas que ocorreram devido ao período de isolamento, com isso ocorreram alguns problemas no processo de entrega de atividades semanais, pois apesar de uma grande quantidade de alunos matriculados nas turmas, não foi obtido um grande número de atividades para análise, devido a:

- Falta de entrega das atividades;
- Atividades entregues incompletas;
- URLs sem acesso;
- Quem não conseguiu, por algum eventual motivo, entregar as atividades por

meio da URL, passou a entregar a execução do projeto por vídeo e o código por foto, sendo inviável avaliar no Dr. Scratch;

- Atividades que foram compartilhadas e no momento da análise, já apresentaram compartilhamento não disponível.

Após essa análise manual realizada com os dados obtidos pelo Dr. Scratch, percebeu-se que tal análise se torna impraticável, pois analisar repetidamente atividades de cada aluno ou turma demanda um grande período de tempo utilizado para extração e organização dos dados, a fim de gerar os resultados.

Com isto, foi vista a necessidade de realizar um segundo estudo, onde os dados obtidos por meio da ferramenta fossem analisados de forma mais sistemática, e a partir de tal estudo chegamos aos resultados presentes na subseção 6.2.

#### **4.4 Análise Automática dos Dados**

Com foco em propor uma análise sistemática dos dados obtidos pelo Dr. Scratch, a ferramenta passou por uma adaptação, a fim de torná-la apta para extrair automaticamente os dados das atividades de introdução a programação com o Scratch, com a intenção de gerar visualizações dos resultados de forma automática, desta forma proporcionando visualizações rápidas, que sem esta adaptação seriam produzidas manualmente.

A análise automática se concretizou a partir de uma adaptação no código do Dr. Scratch, já que o mesmo possui código livre em linguagem Python, fazendo uso do framework Django e containers Docker. Para conseguir implantar o Dr. Scratch localmente, um BD precisa ser utilizado para guardar os resultados das análises, e para realizar essas instalações e configurações, as indicações podem ser acessadas no github da autora<sup>3</sup>, que apresenta os passos necessários.

Para conseguir extrair os dados automaticamente, um sistema de gerenciamento de banco de dados MySQL foi acoplado aos containers Docker do Dr. Scratch, desta forma sendo possível visualizar por meio do BD, total as tabelas inclusas na codificação do Dr. Scratch, conforme mostrado na figura 11.

Figura 11 - Visualização das tabelas do Dr. Scratch, por meio do BD.

---

<sup>3</sup> <https://github.com/FrancielyAlves/Adaptacao-do-Dr.-Scratch.git>

The screenshot shows a list of tables in a database. The table 'app\_file' is highlighted with a green box. The list includes:

#	Table Name
1	app_activity
2	app_coder
3	app_comment
4	app_csvs
5	app_discuss
6	app_file
7	app_organization
8	app_organizationhash
9	app_stats
10	app_student
11	app_teacher
12	auth_group
13	auth_group_permissions
14	auth_permission
15	auth_user
16	auth_user_groups
17	auth_user_user_permissions
18	django_admin_log
19	django_content_type
20	django_migrations
21	django_session

Fonte: Elaborada pela autora.

A tabela (app\_file) guarda os dados das pontuações referentes aos conceitos avaliados. Ao adicionar uma URL do projeto Scratch, a ferramenta avalia, e as pontuações são salvas na tabela (app\_file), que pode ser acessada diretamente no BD, como por exemplo na figura 12.

Figura 12 - Visualização das pontuações avaliadas pelo Dr. Scratch, diretamente no BD.

id	filename	method	time	abstraction	dataRepresentation	deadCode	flowControl	initialization	logic	parallelization	score	spriteNaming	synchronization	userInteractivity	duplication
1	623712816.sb3	url	2022-0...	1	2	0	3	0	2	3	15	0	2	2	2
2	623707472.sb3	url	2022-0...	1	2	1	3	0	2	3	15	0	2	2	2
3	623704003.sb3	url	2022-0...	1	2	0	3	0	2	3	15	0	2	2	3
4	623835325.sb3	url	2022-0...	1	2	0	3	0	2	3	15	0	2	2	3
5	623710537.sb3	url	2022-0...	1	2	1	3	0	2	3	15	0	2	2	1
6	633634517.sb3	url	2022-0...	1	2	0	2	0	2	1	12	0	2	2	0
7	623725079.sb3	url	2022-0...	1	2	0	3	0	2	3	15	0	2	2	2
8	630924569.sb3	url	2022-0...	1	2	0	3	0	2	3	15	0	2	2	2
9	623703919.sb3	url	2022-0...	1	2	0	3	0	2	3	15	0	2	2	2

Fonte: Elaborada pela autora.

Com a figura 12 pode-se observar várias informações referentes aos projetos avaliados. Para gerar as visualizações gráficas, foram utilizadas as informações de:

- Id = Número de projeto avaliado, no exemplo em questão foram avaliados 9 projetos.
- Abstraction = Pontuação referente ao conceito de Abstração.
- Data Representation = Pontuação referente ao conceito de Representação de Dados.

- Flow Control = Pontuação referente ao conceito de Controle de Fluxo.
- Logic = Pontuação referente ao conceito de Lógica.
- Parallelization = Pontuação referente ao conceito de Paralelismo.
- Synchronization = Pontuação referente ao conceito de Sincronização.
- User Interactivity = Pontuação referente ao conceito de Interação com o Usuário.
- Score = Pontuação total do obtida por cada projeto.

Após identificar a tabela que guarda os dados necessários para efetuar as análises, as pontuações passaram a ser extraídas automaticamente em formato CSV para gerar as visualizações gráficas. Após a extração, o arquivo CSV foi adicionado na ferramenta online Google Data Studio, que converte os dados em gráficos, desta forma gerando as visualizações que facilitam a interpretação dos dados obtidos. Tais visualizações geradas estão presentes na subseção 6.2.

#### **4.5 Avaliação das visualizações**

Após os dados serem convertidos em gráficos, foi produzido um formulário que está presente no anexo 3, e foi construído primeiramente para passar por uma análise piloto, onde foi respondido por apenas duas pessoas, na intenção de obter sugestões de melhorias para então ser liberado definitivamente para obter respostas. As sugestões de melhorias obtidas foram:

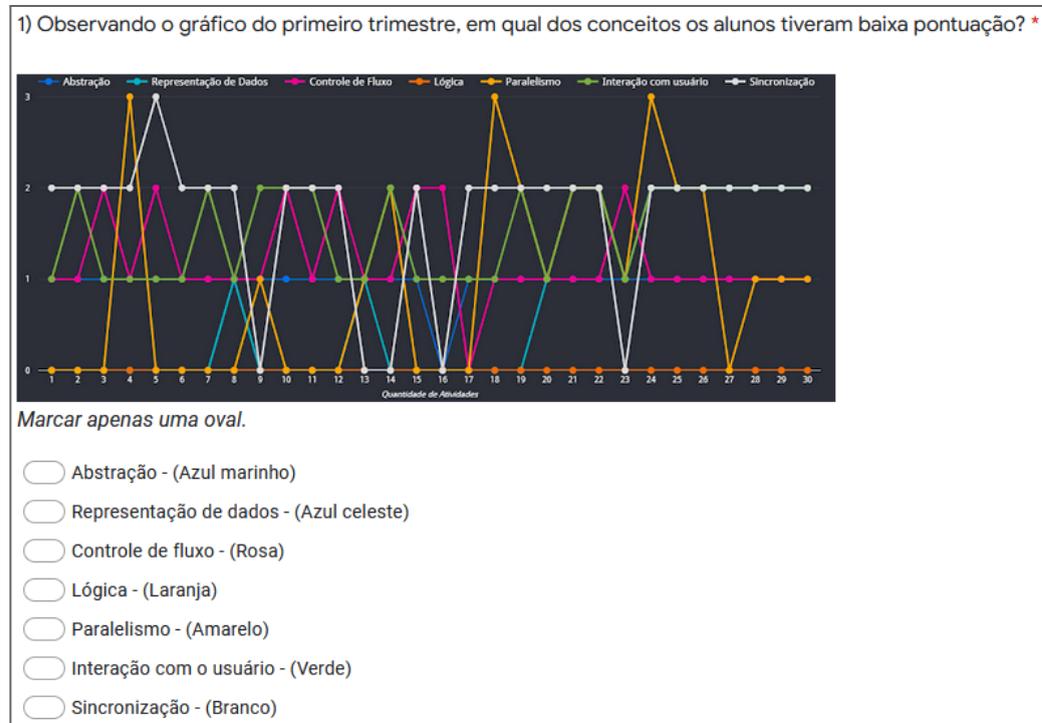
- Padronizar o idioma, pois os gráficos apresentavam algumas palavras em inglês, mas as opções para resposta estavam em português;
- Aumento no tamanho dos gráficos para uma melhor visualização;

As melhorias foram aplicadas e o formulário foi divulgado via redes sociais, sendo: E-mail, Instagram, WhatsApp e Telegram, com isso conseguiu alcançar 28 respostas. Sua finalidade foi avaliar se as visualizações gráficas geradas pelo Google Data Studio são apresentadas de forma clara e objetiva.

O exemplo mostrado da figura 13 apresenta um dos gráficos gerados e um questionamento para ser analisado, a fim de avaliar os dados presentes, sendo que a pontuação de 0 à 3, presente no eixo Y representa a pontuação que cada atividade pode atingir, já as numerações presentes no eixo X, representam a quantidade de atividades analisadas. Vale salientar que as visualizações são apresentadas em diferentes estilos de gráficos, a fim de

verificar qual estilo de gráfico é considerado melhor para apresentar os resultados, de acordo com as respostas dos participantes do formulário.

Figura 13 - Exemplo de visualização e questionamento presente no formulário.



Fonte: Elaborada pela autora.

Para validar a proposta de visualizações, os dados foram validados por meio do formulário, na intenção de verificar se os dados obtidos na avaliação automática são de fácil compreensão, pois são esses dados que podem auxiliar os professores na avaliação de atividades de introdução a programação com o Scratch.

## 5 Propostas de Análise de Visualização

Esta seção apresenta as análises realizadas através do Dr. Scratch adaptado, e as visualizações gráficas a partir do Data Studio.

### 5.1 Proposta para Análise Automática das Atividades

Foram determinadas quatro tipos de visualizações gráficas, na intenção de entender o avanço do aprendizado dos alunos, tanto de forma individual como em turmas. Foram avaliadas atividades de alunos de cinco turmas, sendo: duas turmas de segundo ano, uma turma do terceiro, quarto e quinto ano. Apenas os alunos que entregaram, no mínimo, quatro atividades, entraram nesta análise.

### 5.1.1 Análise por Turma

Nesta análise buscou-se verificar graficamente o avanço do aprendizado dos alunos por turma ao longo dos 8 meses de aula. Referente à turma do segundo ano, foram analisadas 52 atividades, é válido informar que esse total de atividades corresponde a entrega de apenas sete alunos. Observando o gráfico 1, pode-se verificar que várias programações evoluíram em vários conceitos, e se mantiveram em pontuações 2 ou 3.

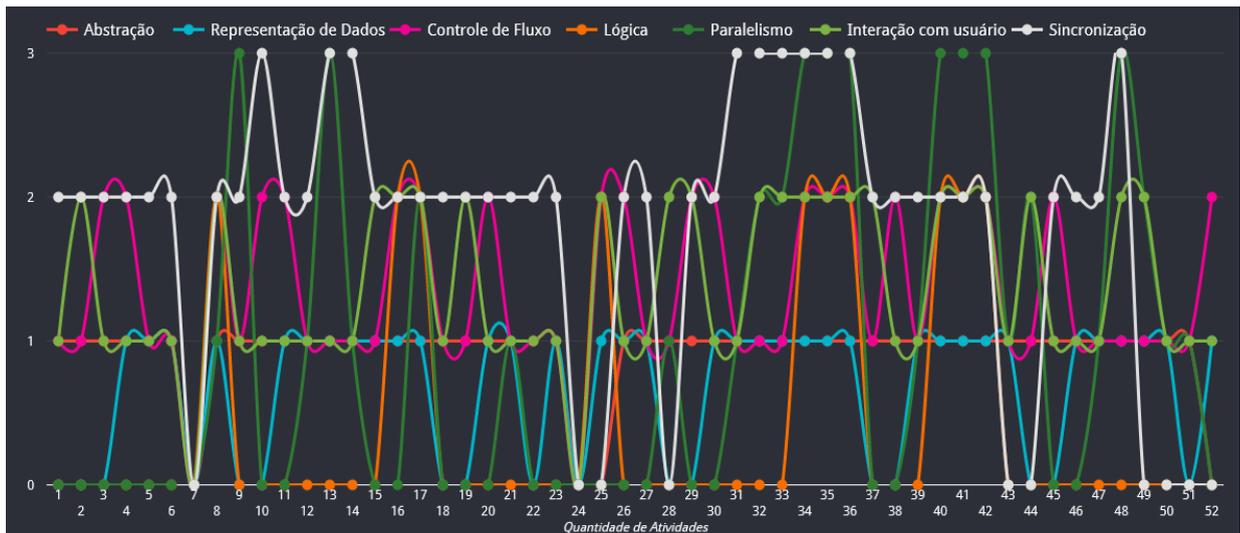


Gráfico 1 - Dados dos alunos das duas turmas do 2º Ano.

O conceito de Paralelismo foi zero em 25 atividades, pois este conceito só pontua quando existem programações que são executadas simultaneamente, desta forma 25 atividades possuem uma programação simples que conta com apenas a execução de um único código. Este tipo de projeto é típico quando o aluno programa apenas um personagem em um único cenário. Desta forma não ocorre nenhuma troca de comunicação entre programações de outros personagens. Porém 8 atividades atingiram a pontuação máxima, isso significa que a programação contou com várias partes de código que foram executadas ao mesmo tempo, isso pode ocorrer por exemplo quando na programação: um personagem está andando sozinho pelo cenário, de forma coordenada programada anteriormente ou de forma aleatória, e ao mesmo tempo existe uma personagem no cenário que só consegue se mover com o uso das teclas do teclado.

O conceito que menos pontuou foi Lógica, tendo 42 atividades com a pontuação zerada, isso ocorre devido o conceito exigir programações mais robustas para pontuar, utilizando os blocos de controle que programam condições a serem seguidas, e essa programação é bastante utilizada no desenvolvimento de grandes jogos, o que não foi o caso da turma do segundo ano,

pois o foco se concentrou em desenvolvimento de animações, diálogos entre personagens e jogos com pequeno grau de dificuldade, tanto no desenvolvimento, como em jogabilidade. As atividades que pontuaram no conceito de lógica foi devido a programações de pequenos jogos, onde os alunos programaram que (se o personagem tocar em um determinado objeto, este será coletado e escondido do cenário e a quantidade de objetos coletados seriam adicionados a uma variável).

Um conceito que pontuou na maioria das atividades foi a Sincronização, e em muitas atividades obteve a pontuação máxima. Este conceito é utilizado quando o aluno programa diferentes partes de código e determina a sincronização na qual a programação será executada. Um exemplo que representa este conceito é a programação de troca de mensagens entre personagens quando alguma condição passa a ser atendida.

Houve poucas atividades para realizar a análise do terceiro ano, sendo apenas 9 atividades, de dois alunos. Pode-se observar as pontuações dos conceitos no gráfico 2, onde alternam muito entre as atividades, exceto o conceito de Lógica que se manteve zerado na maioria delas. Com isso, significa que não foram utilizados blocos de condição nas programações.

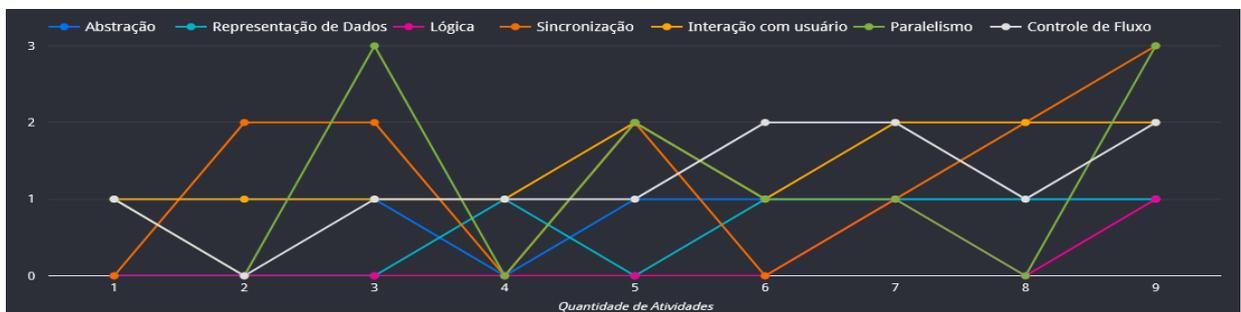


Gráfico 2 - Dados dos alunos do 3º Ano.

O único conceito que não ficou zerado em nenhuma atividade foi Interação com Usuário. Para obter pontuação neste conceito, a programação precisa ser desenvolvida para que o usuário ao executar o programa, possa interagir com o projeto, por exemplo: O programa pode solicitar que o usuário utilize as teclas do teclado para mover um personagem ou utilizar o cursor do mouse para clicar em algum botão no cenário. Desta forma interagindo com o programa.

O gráfico 3 representa o resultado da análise das atividades do quarto ano, sendo 26 atividades de 5 alunos. O conceito de lógica novamente foi pontuado em poucas atividades, sendo 20 atividades que não pontuaram. Os blocos de controle são os principais para gerar pontuações de lógica nas atividades, e a função de tais blocos foram bastante enfatizadas com a

turma em questão, o fato é que ao invés de programar ações para os personagens utilizando condições sequenciadas, por exemplo: se o personagem tocar na borda ou nos inimigos do jogo, então adicione -1 na quantidade de vidas, se as vidas forem igual a 0, então mude para cenário game over e pare tudo. Os alunos optaram por programar várias cópias utilizando a condição (se então) para ao terminar a execução de uma, já iniciar a outra, tornando assim a programação com scripts duplicados.

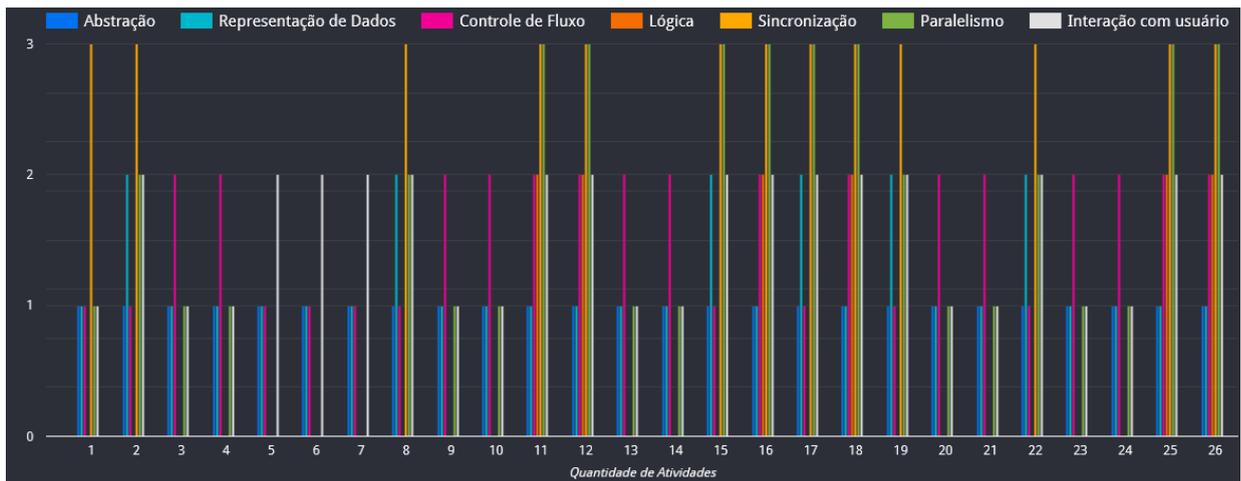


Gráfico 3 - Dados dos alunos do 4º Ano.

Com esta análise acredita-se que as atividades tiveram os conceitos de Sincronização e Paralelismo bem trabalhados, já que a maioria das atividades pontuou nestes conceitos e boa parte delas obteve a pontuação máxima. Desta forma as programações foram desenvolvidas para serem executadas de forma paralela, onde diferentes partes de códigos ficam em funcionamento ao mesmo tempo, e as programações são sincronizadas onde existem diferentes partes de códigos que passam a ser executados apenas quando recebem alguma instrução para iniciar e finalizar seu funcionamento.

Analisando os dados do gráfico 4, pode-se perceber que foram analisadas 17 atividades do quinto ano, sendo referente a entrega de dois alunos. Observando o gráfico é notório que apenas dois conceitos ficam oscilando frequentemente entre as atividades, sendo o Paralelismo e Lógica, o restante dos conceitos se manteve sempre com pontuações relevantes, exceto Sincronização, que obteve zero em uma das atividades.

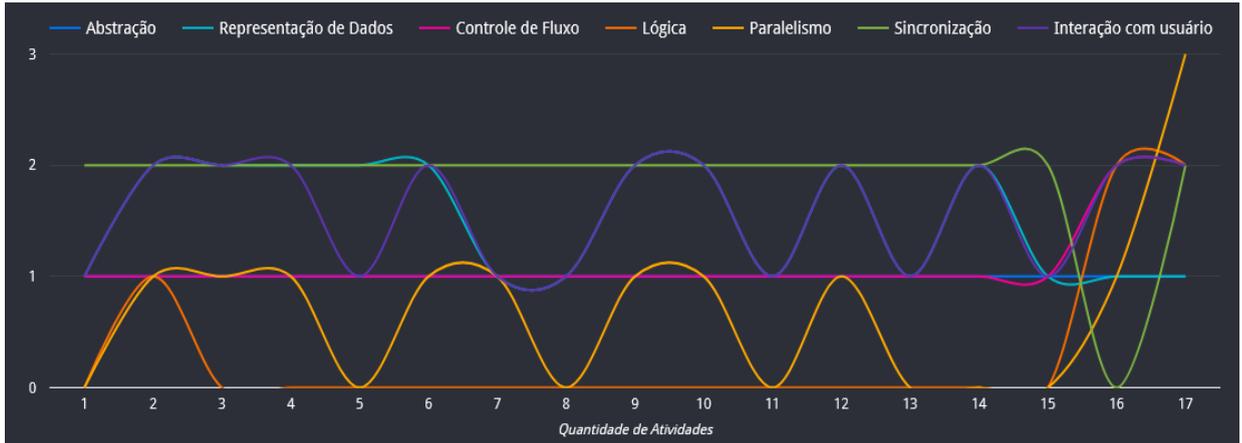


Gráfico 4 - Dados dos alunos do 5º Ano.

As atividades que obtiveram pontuações no conceito de lógica, tiveram o bloco de condição (se então, se não) programado, sendo o bloco mais simples a ser utilizado para conseguir uma pontuação.

### 5.1.2 Análise por Aluno

Nos dados obtidos por aluno, a análise foi realizada utilizando todas as atividades que os alunos entregaram durante todo o período de aula. O gráfico 5 representa a análise das atividades do primeiro aluno, onde é possível observar que três dos conceitos iniciaram zerados nas atividades, sendo: Lógica, Paralelismo e Representação de dados. Os três conceitos conseguiram pontuar posteriormente após algumas atividades, isso mostra uma evidência do melhoramento das programações.



Gráfico 5 - Dados do 1º Aluno.

A partir da atividade 3, o conceito de Representação de dados passou a obter pontuação, e este conceito é trabalhado quando programa-se para que características de personagens possam ser modificadas ao longo da execução do programa e também quando utiliza-se

mecanismos para guardar informações, um exemplo disto é o uso das variáveis que podem guardar um valor que representa diferentes tipos de dados, sendo: pontos ganhos e quantidades de vidas.

Observando a atividade 7, vemos um ocorrido que está presente em algumas atividades em diferentes análises, onde é notório que não houve nenhuma pontuação adquirida para esta atividade, todos os conceitos foram zerados. Isto ocorre quando o aluno realiza a montagem do design do projeto porém não programa nenhuma ação para os personagens, pois a ferramenta Dr. Scratch só avalia projetos que contêm programações (se houver só o design, é irreconhecível para a ferramenta). Neste contexto, vale salientar a importância da análise manual citada na subseção 4.2, onde aspectos importantes de design podem ser analisados pelo professor, desta forma se torna imprescindível o uso de ambas as estratégias de avaliação.

Nos dados do gráfico 6, é possível notar o quanto as atividades do segundo aluno obtiveram oscilação nos conceitos, apresentando sempre as pontuações bem distribuídas. O conceito que menos pontuou foi Lógica, pois obteve pontuação apenas nas últimas atividades, com isso pode-se entender o avanço do aprendizado sobre os blocos que representam este conceito, pois passaram a ser utilizados após o aluno adquirir uma bagagem de conhecimentos que o ajudou a melhorar seu desempenho na entrega das atividades.



Gráfico 6 - Dados do 2º Aluno.

O conceito de Sincronização se manteve sempre com boa pontuação em todas as atividades, com isso pode-se entender que todas as atividades foram programadas para houvesse uma sincronia na execução do código.

O gráfico 7 apresenta os dados das oito atividades do terceiro aluno, que estão com as pontuações bem distribuídas, tendo dois conceitos que pontuaram em todas as atividades, sendo: Interação com Usuário e Controle de Fluxo que é programado através de blocos que

controlam o comportamento dos personagens, fazendo que funções sejam repetidas até que uma ação já determinada aconteça.

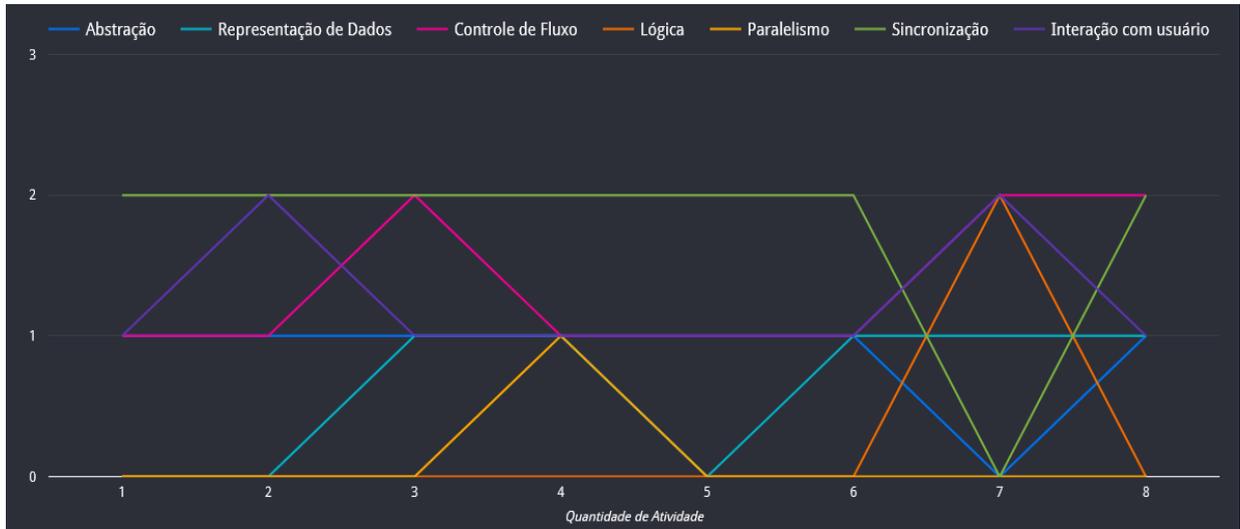


Gráfico 7 - Dados do 3º Aluno.

Os conceitos de Lógica, Paralelismo e Representação de Dados, foram pouco utilizados nas primeiras atividades, passando a pontuar posteriormente, com isso percebe-se que em algumas atividades as programações passaram a ser mais elaboradas utilizando blocos de diferentes funções.

A análise das atividades do quarto aluno mostra bastante oscilação das pontuações de acordo com a representação do gráfico 8. Conceitos que iniciaram com pontuações baixas foram elevando o nível em atividades posteriores, como pode-se ver na linha verde que representa a sincronização, que esteve em baixa em uma atividade e logo depois foi o conceito que mais obteve a pontuação máxima.



Gráfico 8 - Dados do 4º Aluno.

Diferente da análise das atividades dos alunos anteriores, este aluno entregou atividades com bastante variação nas pontuações, o que pode-se observar o avanço de alguns conceitos a

partir da sétima atividade, onde Lógica vinha sempre com a pontuação zero, isso passou a ocorrer devido a variações de blocos que passaram a ser utilizados, após feedback que demonstravam pontos onde a programação precisaria ser melhorada. Observa-se que conceito de Abstração manteve a mesma pontuação da primeira atividade até a última, e os blocos que a representam programam clones, se tratando de um abstração utilizada para programar um objeto e criar quantas cópias precisar, evitando que vários personagens sejam programados com o mesmo código, pois a técnica de repetição de código dificulta mudanças, edições, por necessitar que todos os personagens fossem alterados.

O gráfico 9 apresenta a análise das atividades do quinto aluno, onde pode-se observar um grande avanço nas pontuações, já que inicialmente alguns conceitos possuem zero na pontuação, mas a partir da segunda atividade os conceitos começaram a pontuar gradativamente. Com esta análise é nítido o desenvolvimento do aprendizado do aluno, onde ele passa a programar com diferentes blocos que satisfaz o resultado esperado, obtendo pontuação em todos os conceitos.



Gráfico 9 - Dados do 5º Aluno.

É fato que para resolver os problemas presentes nas atividades, nem todas exigem a utilização de todos os blocos que representam os conceitos para serem resolvidas, mas é importante que com os conhecimentos prévios adquiridos as programações passem a ser mais robustas, se tornando um código menor que executa a mesma função, e foi exatamente que ocorreu com as atividades deste aluno.

Os dados apresentados no gráfico 10 são referentes a um dos alunos que mais entregou atividades ao longo do período de aulas. Porém, o sexto aluno mesmo entregando tantas atividades não pontuou no conceito de lógica em nenhuma delas.

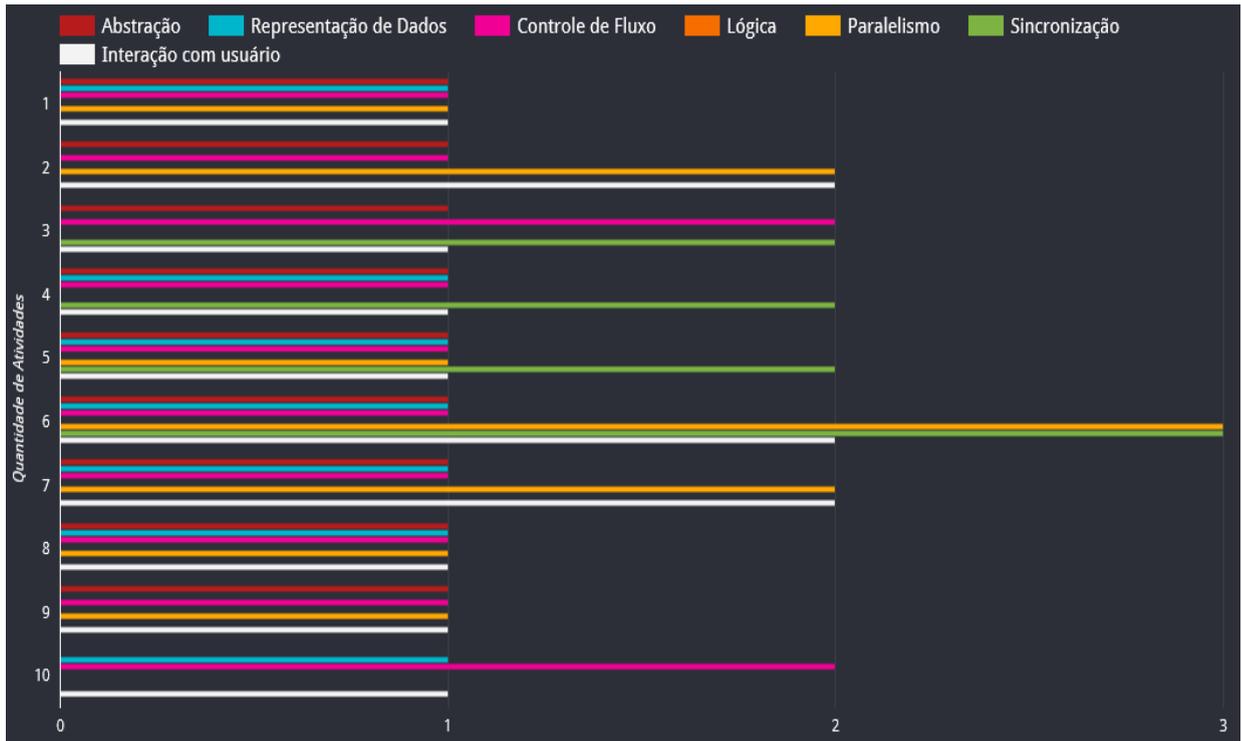


Gráfico 10 - Dados do 6º Aluno.

Lógica é um conceito que diversos alunos apresentaram dificuldades para desenvolver, por se tratar de um conceito que necessita de códigos refinados para ser presente. Programações simples como diálogos não necessitam de diferentes ações, desta forma não utilizam os blocos de lógica, e muitas vezes por não conseguir construir uma programação com diferentes blocos, os alunos acabavam duplicando códigos que ao final algumas vezes conseguiam resolver a problemática mesmo sem a função de lógica.

Analisando o gráfico 11, os dados do sétimo aluno demonstram que o mesmo obteve baixas pontuações nas atividades, embora tenha avançado na quantidade de conceitos pontuados, já que iniciou com pontuação em apenas 2 conceitos.

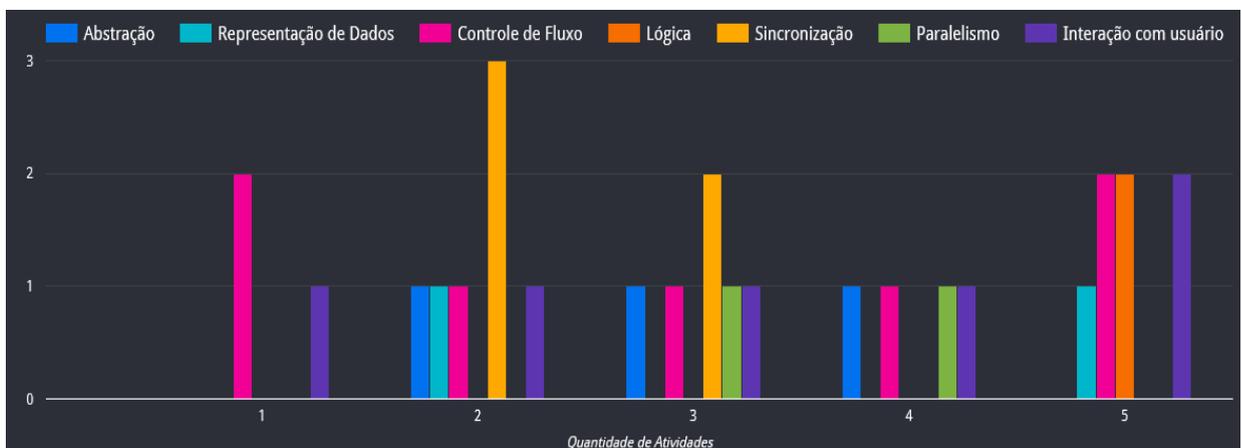


Gráfico 11 - Dados do 7º Aluno.

Os únicos conceitos que pontuaram em todas as atividades foram Controle de Fluxo e Interação com Usuário. Pelas baixas pontuações percebe-se que foram entregues projetos com simples programações, porém que exigiam interação do usuário para funcionar, sendo cliques em botões ou movimentos de personagens, além de programar os personagem com suas ações bem definidas, não havendo código repetido para executar alguma função, já que o controle de fluxo foi bastante utilizado.

O oitavo aluno entregou a quantidade mínima determinada de atividades para a análise. E entre elas, consta novamente uma atividade que possui apenas o design produzido, pois não houve nenhuma pontuação para a atividade 4. O conceito de lógica não obteve pontos em nenhuma atividade, conforme mostrado no gráfico 12, considerando assim que as atividades foram entregues com programações simples.

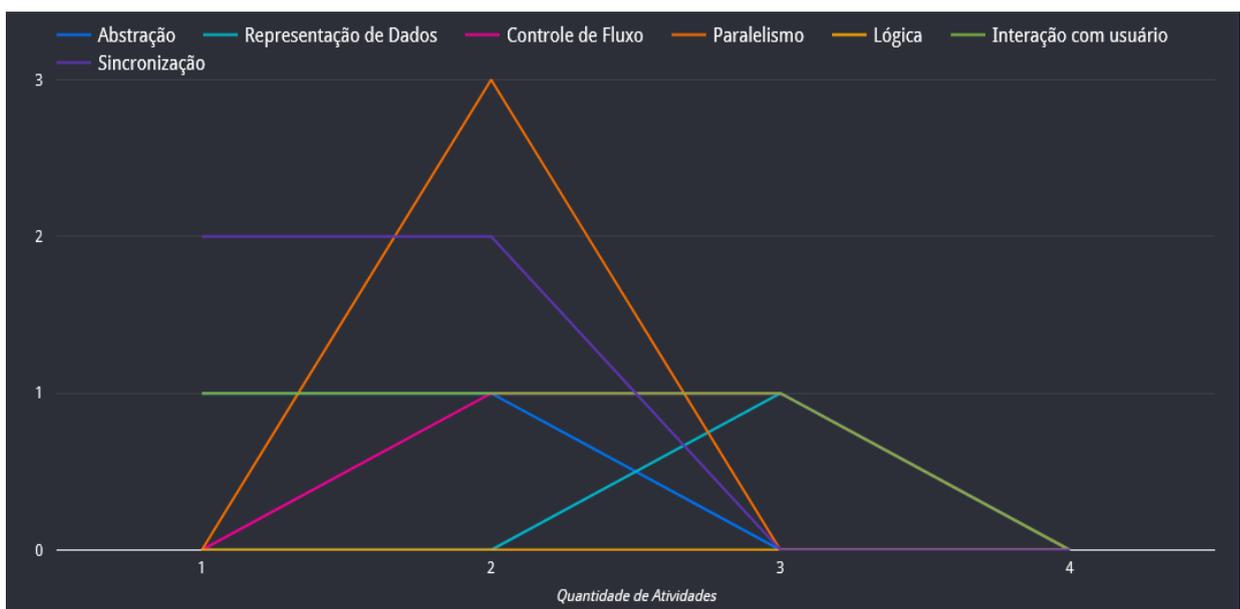


Gráfico 12 - Dados do 8º Aluno.

Diferente da análise dos alunos anteriores, que iniciavam com baixas pontuações, neste gráfico pode-se perceber que as duas primeiras atividades possuem melhor rendimento, havendo uma queda nas duas últimas, devido este aluno conter um número elevado de faltas em comparação com o restante já analisado.

A análise representada no gráfico 13 é referente ao nono aluno, que apresentou boas pontuações nas atividades, onde a maioria dos conceitos não obteve pontuações zeradas. São projetos que necessitam fortemente da interação com o usuário para funcionar, pois este conceito alcançou boas pontuações.

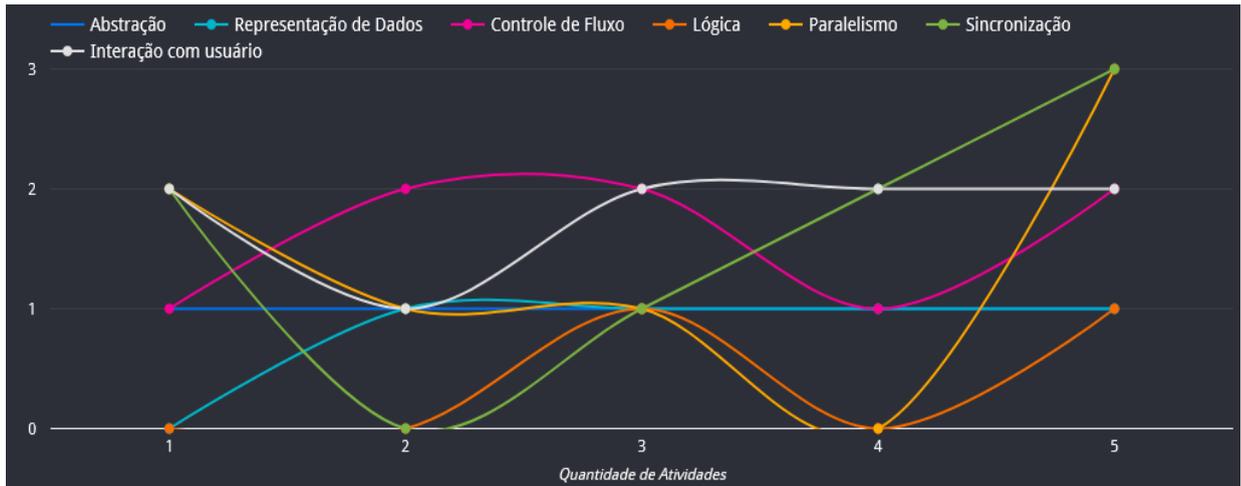


Gráfico 13 - Dados do 9º Aluno.

Como se trata de projetos que o usuário precisa interagir para funcionar, o conceito de sincronização foi bem pontuado na maioria das atividades, já que as ações ocorrem de forma organizada a partir das interações que determinam a ordem de funcionamento.

O décimo aluno não obteve pontuação no conceito de lógica em nenhuma atividade, como mostrado no gráfico 14, e o conceito de sincronização foi pouco utilizado, desta forma pode-se entender que se trata de projetos básicos de programação simples.

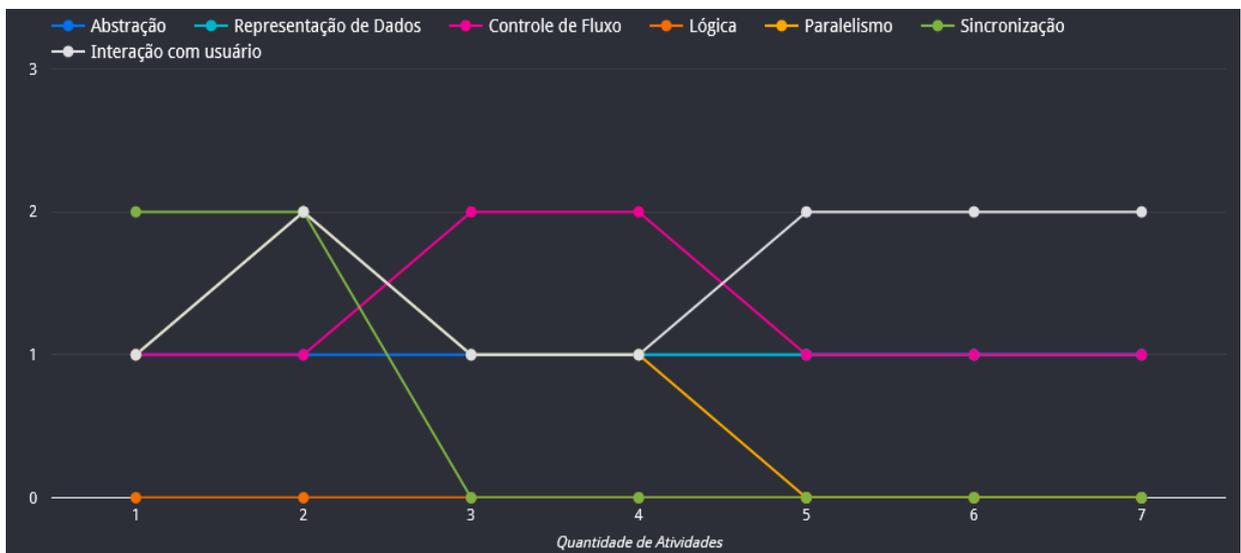


Gráfico 14 - Dados do 10º Aluno.

No gráfico 15, está nítido o avanço das pontuações do décimo primeiro aluno, que após adquirir o conhecimento prévio de programação em blocos, passou a entregar as atividades sem nenhuma pontuação zerada, de onde pode-se entender que os códigos passaram a ser mais elaborados.



Gráfico 15 - Dados do 11º Aluno.

Um grande código pode funcionar e executar as ações esperadas, porém quando o código passa a ser elaborado com lógica além da programação conter menos blocos, as outras pontuações passam a melhorar, pois os blocos que representam o restante dos conceitos podem ser programados acoplados dentro dos blocos de lógica, e foi que ocorreu com as programações deste aluno.

As programações que apresentaram pontuação de interação com usuário em todas as atividades, como mostrado no gráfico 16, na maioria das vezes são personagens que foram programados para se movimentar a partir das teclas do teclado sem pressionadas, desta forma o décimo segundo aluno programou para que seus personagens se movam a partir da interação do usuário.



Gráfico 16 - Dados do 12º Aluno.

Novamente o conceito de lógica se manteve com a pontuação zeradas em todas as atividades, desta forma não existem os blocos de condição nas programações.

A análise das atividades do décimo terceiro aluno demonstra o quanto as pontuações se mantiveram equilibradas, não havendo uma grande quantidade seguida de atividades com pontuações zeradas, como mostrado no gráfico 17.

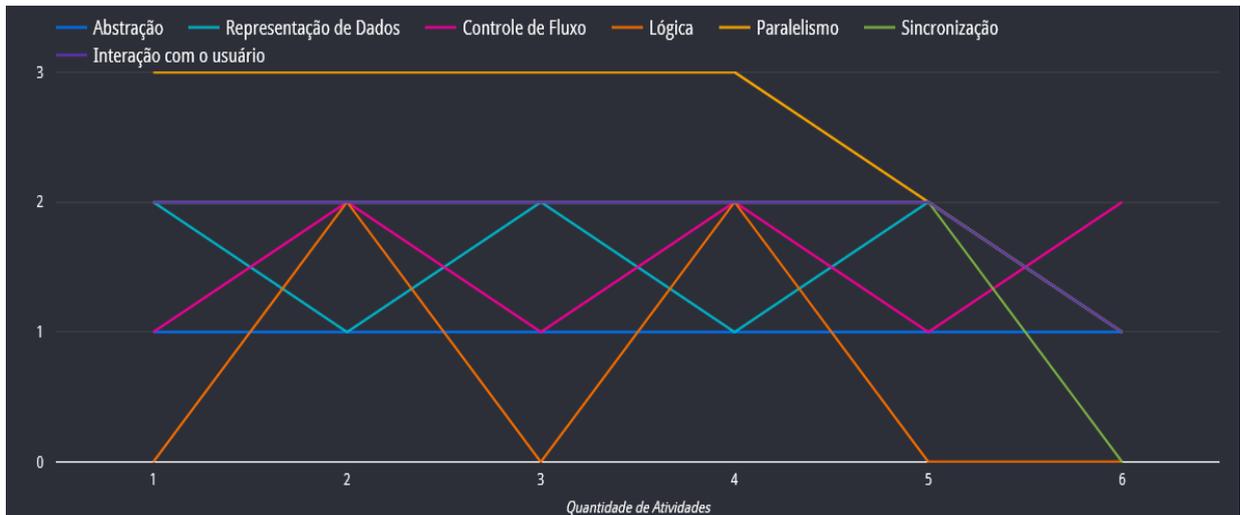


Gráfico 17 - Dados do 13º Aluno.

Com estes dados, pode-se entender que as atividades possuem muitos códigos executados de forma paralela, já que na maioria das atividades obtiveram 3 pontos no conceito de paralelismo. Em um projeto, ao adicionar um som para ser executado (para sempre) após a bandeira verde ser clicada, e adicionar personagens que andam a partir da interação com o usuário, já se caracteriza o conceito de paralelismo, pois tais ações estarão sendo executadas ao mesmo tempo.

Os dados do gráfico 18 demonstram que as atividades do décimo quarto aluno possuem pontuações regulares, já que alguns conceitos se encontram em baixa, mas em outras atividades possuem melhoramento, como pode-se perceber no conceito de lógica que ficou oscilando entre as atividades.

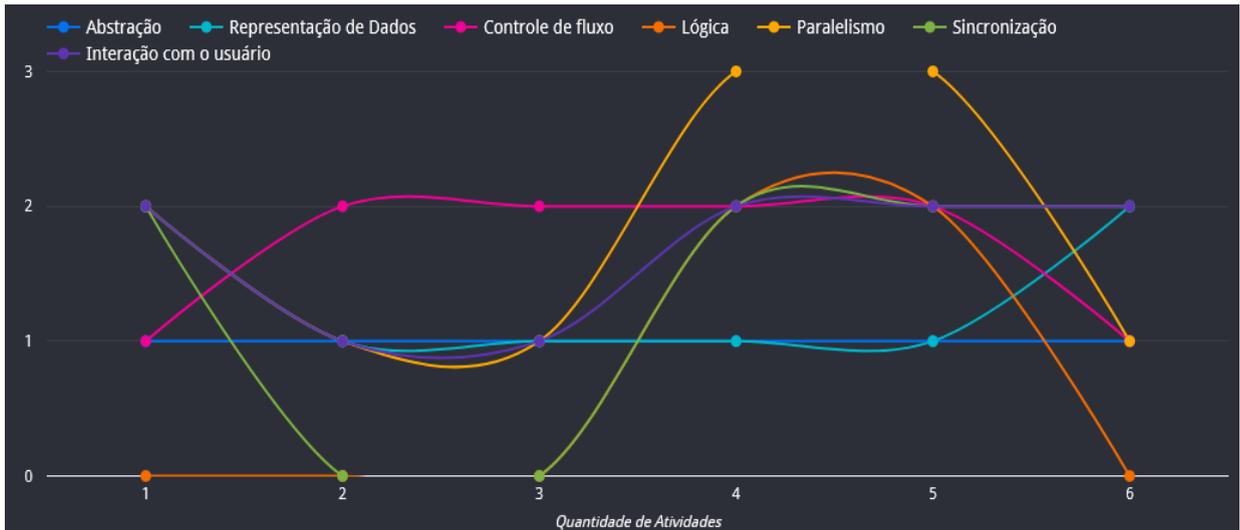


Gráfico 18 - Dados do 14º Aluno.

Novamente, são apresentadas atividades que não possuem nenhum conceito zerado, sendo algo positivo que demonstra que houve a compreensão de como utilizar diferentes blocos para resolver os problemas.

De acordo com os dados anteriores, é notório quais conceitos estão sempre com melhores pontuações, e entre eles estão: Paralelismo e Interação com Usuário, e isto não foi diferente na análise das atividades do décimo quinto aluno, como mostrado no gráfico 19.

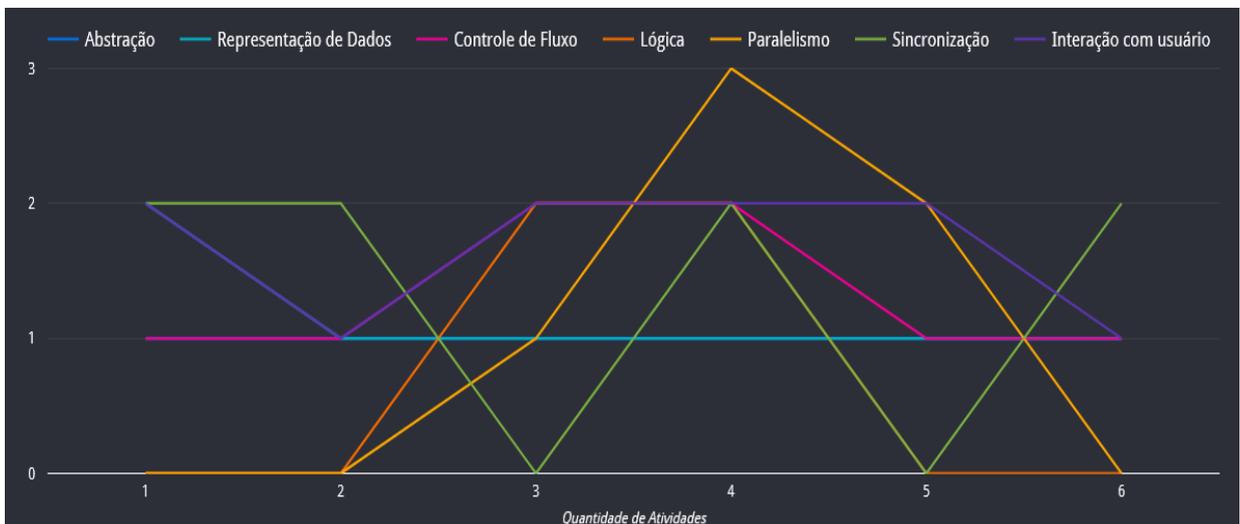


Gráfico 19 - Dados do 15º Aluno.

Tais conceitos na maioria das vezes obtêm boas pontuações devido a programação de movimentos dos personagens em todos os projetos, sendo movimentos para a direção que o usuário determina, obtendo assim códigos executando paralelamente.

Nas atividades avaliadas do décimo sexto aluno, houve pontuação no conceito de lógica apenas em uma atividade, como mostrado no gráfico 20. Já os conceitos de sincronização e interação com usuário foram bastante empregados nas programações, assim como visto em análises anteriores.

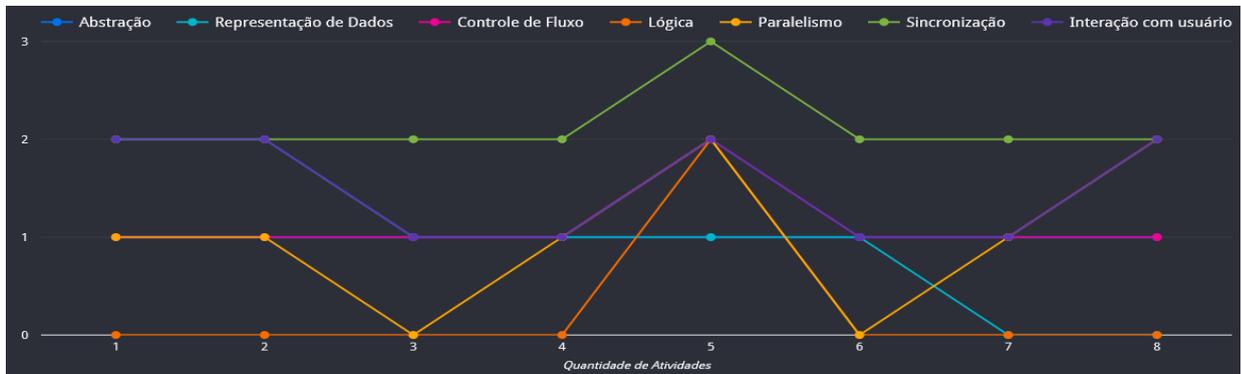


Gráfico 20 - Dados do 16º Aluno.

A análise das atividades de cada aluno favorece dados que podem ser analisados pelos professores na intenção de entender quais pontos os alunos conseguem compreender, e em quais precisam melhorar seu desempenho, pois a representação gráfica possibilita facilmente esta visualização.

A partir da análise por aluno, é notório o quanto as pontuações do conceito de lógica oscilaram nesta análise, onde alguns alunos possuem melhores pontuações do que outros. Isso pode ter ocorrido devido a:

- Dificuldades de compreensão por parte dos alunos;
- Conceito pouco trabalhado nas turmas de 2º ano;
- O aluno optou por resolver as problemáticas com códigos muitas vezes repetitivos, não fazendo uso dos blocos que alavancam o conceito de lógica.

Tais pontos ajudam a compreender o motivo do conceito de lógica ter obtido pontuações em algumas atividades, porém muito abaixo em comparação com os outros conceitos, que muitas vezes se apresentaram com pontuações crescentes.

### 5.1.3 Análise por Trimestre

Para esta análise, foi selecionada a mesma quantidade de atividades para cada semestre, a fim de analisar se houve melhoramento dos conceitos a partir do conhecimento prévio adquirido nos três primeiros meses de aula. Alguns conceitos ficaram com baixas pontuações

nas primeiras atividades avaliadas, e o intuito é verificar se houve um crescente aumento no decorrer das aulas.

Observando o gráfico 21, o conceito de lógica não foi pontuado em nenhuma atividade do primeiro trimestre, pois inicialmente as atividades não exigiam códigos complexos para a resolução das problemáticas, por se tratar de atividades simples, não havendo nenhum desenvolvimento com foco em jogos.

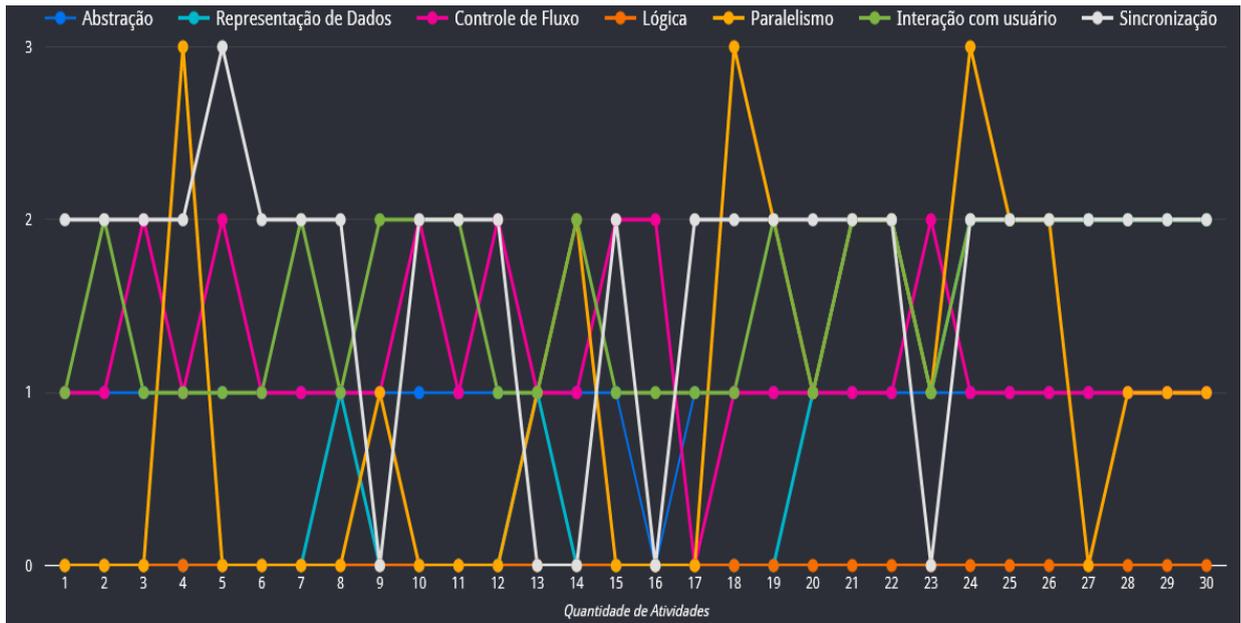


Gráfico 21 - Dados do 1º Trimestre de Aula.

Durante todo o período de aulas, funções de interação com o usuário foram abordadas, utilizando o teclado ou o cursor do mouse, por isso este conceito sempre possui boas pontuações.

Comparando a análise do primeiro e segundo semestre, é de fácil visualização identificar no gráfico 22 quais conceitos obtiveram melhores pontuações, onde conceito de lógica passou a ser pontuado, demonstrando assim um avanço na utilização de diferentes blocos.

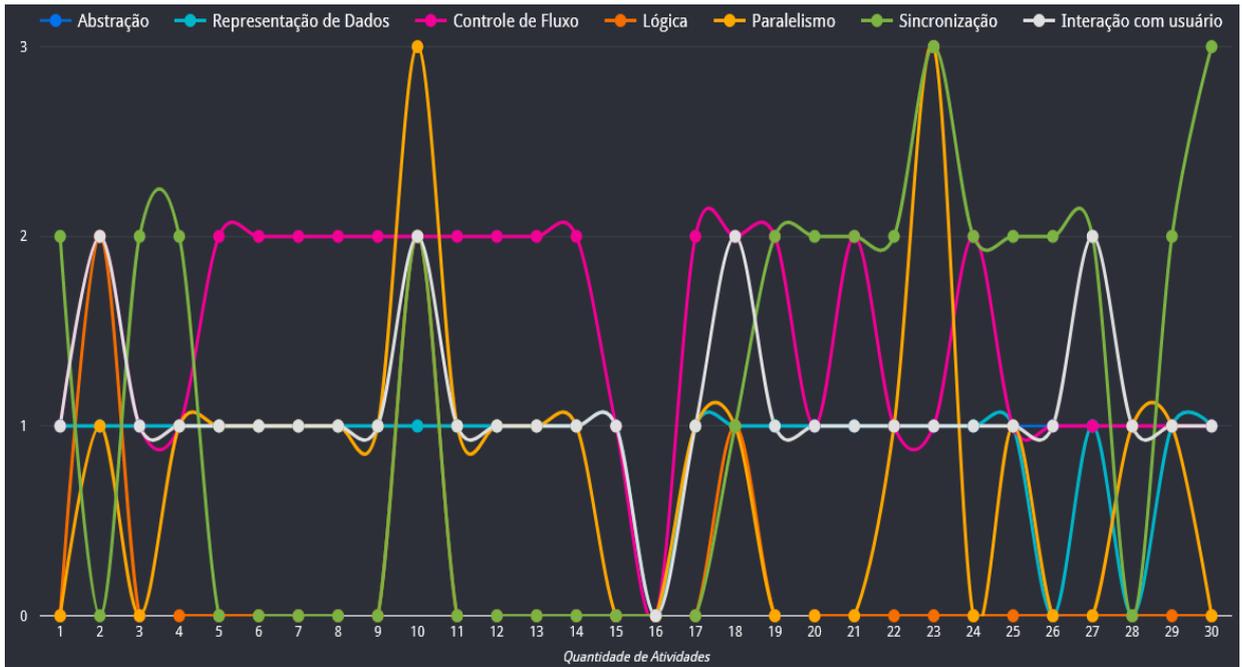


Gráfico 22 - Dados do 2º Trimestre de Aula.

A análise contém uma atividade que não pontuou em nenhum conceito, pois nesta atividade apenas o design foi elaborado, como já visto em análises anteriores.

Nos dados do gráfico 23, referente ao terceiro trimestre, existem algumas atividades que não pontuaram em nenhum conceito, mas apesar deste ponto negativo, pode-se perceber o quanto as pontuações melhoraram nas demais atividades, contendo o conceito de lógica com pontuações crescente, em várias atividades seguidas, algo que não ocorreu nas análises dos outros trimestres.

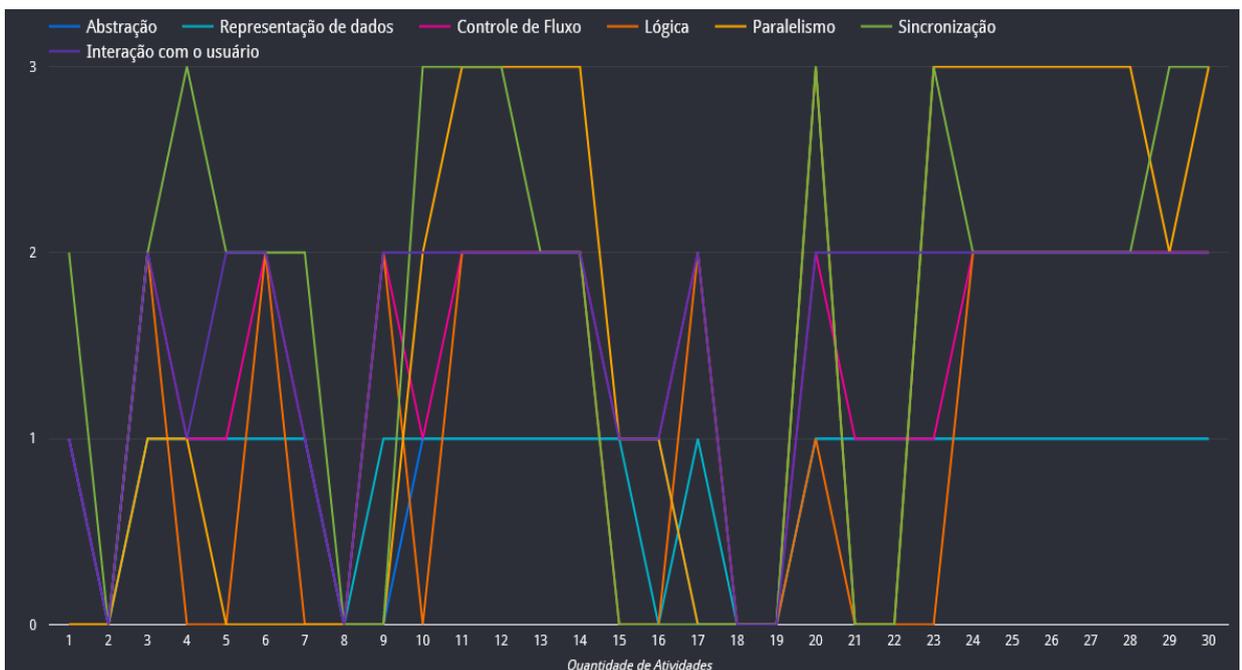


Gráfico 23 - Dados do 3º Trimestre de Aula.

### 5.1.4 Análise por Faltas

Para esta análise, foram utilizadas as atividades dos alunos que tiveram o maior número de faltas, a fim de analisar se este fator influencia no resultado das atividades, por se tratar de aulas remotas, os alunos poderiam consultar o material da aula e resolver as atividades, desta forma não havendo interferência nos resultados. O intuito é verificar se houve queda nas pontuações, e quais são os conceitos que mais precisam ser trabalhados.

Nas análises anteriores, foi fácil perceber que o conceito de lógica obteve pontuação em poucas atividades, e nesta análise não é diferente, conforme apresentado no gráfico 24, pois tal conceito é mais difícil de ser trabalhado, principalmente com alunos que não possuem nenhum conhecimento prévio do conteúdo, e neste caso o número de faltas influenciou bastante para esta análise negativa de tal conceito.

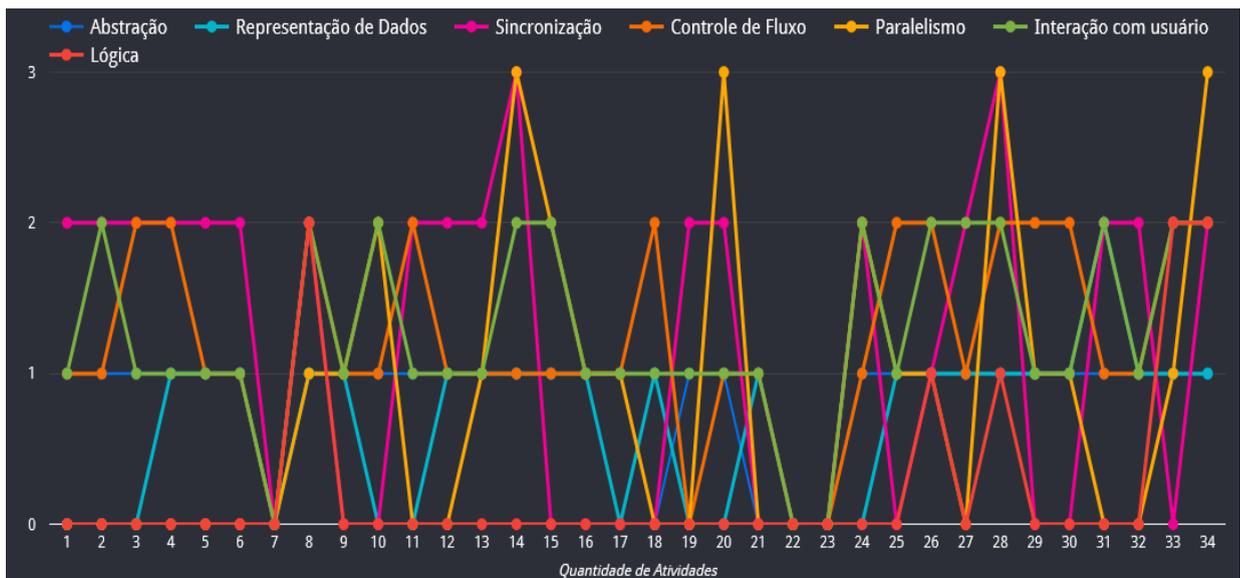


Gráfico 24 - Dados de alunos com no mínimo quatro faltas.

Já os conceitos de sincronização e paralelismo continuam em alta, sendo os mais implementados juntamente com a interação do usuário.

De fato, essas visualizações podem orientar os professores em tomadas de decisão de como abordar determinados conceitos em sala de aula, podendo acompanhar o desenvolvimento de cada turma, cada aluno individualmente, e entregas de atividades realizadas em um período de tempo já determinado. Com estes dados, pode-se observar em

quais conceitos os alunos tiveram o menor rendimento para então buscar metodologias e estratégias que os façam melhorar o desempenho.

## 6 Resultados

Nesta seção estão presentes os resultados obtidos manualmente por meio do Dr. Scratch, e os resultados do questionário respondido por profissionais de diferentes áreas, a fim de avaliar qual tipo de visualização consegue apresentar os dados de forma clara e objetiva.

### 6.1 Resultado da Análise Manual por meio do Dr. Scratch

Inicialmente, as atividades foram avaliadas na ferramenta Dr. Scratch, e os resultados foram extraídos do site de forma manual, e adicionados a uma planilha, onde foram utilizados para gerar gráficos a fim de exibir graficamente os resultados obtidos nas atividades.

Para gerar a representação mostrada no gráfico 25, foram consideradas as pontuações de todas as 102 atividades citadas na subseção 4.3. A partir do gráfico, é possível observar a evolução das programações entregues nas atividades, de acordo com a avaliação dos conceitos analisados pelo Dr. Scratch.

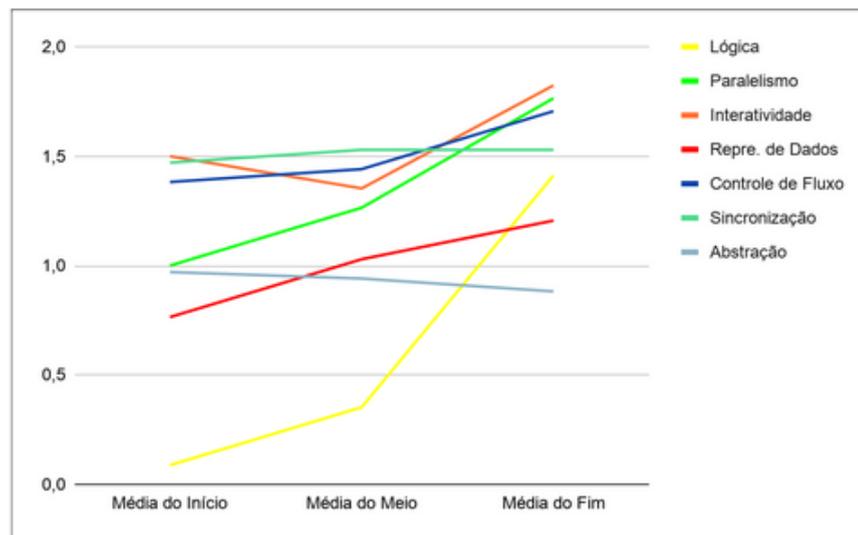


Gráfico 25 - Avaliação das atividades entregues.

O conceito de lógica se encontra em baixa no início das atividades, pois o Dr. Scratch avalia se na programação existem os blocos de condição (se, então, senão e operadores lógicos), e nas primeiras atividades poucos alunos fizeram uso desses blocos em suas programações, por terem sido pouco enfatizados nas primeiras aulas. O conceito de interatividade com o usuário esteve em diferentes posições, isso é algo comum de ocorrer, já que os blocos que representam esse conceito geralmente são utilizados com frequência, pois

são blocos considerados de início, por esperarem uma ação do usuário para então iniciar a execução, sendo: bloco da bandeira verde. Vários desses blocos podem ser utilizados no mesmo programa, porém é necessário utilizar pelo menos um, e o momento de queda deste conceito ocorreu devido aos alunos terem utilizado estes blocos numa quantidade mínima permitida.

Visualizando a análise geral das atividades, é nítida a evolução da maioria dos conceitos, comparada a média de início, com a média do meio, e podemos observar como o conceito de representação de dados se manteve crescente. Este conceito é representado pelo uso de variáveis e listas, que no decorrer das atividades passaram a ser utilizadas rapidamente pelos alunos, desta forma adquirindo um aumento significativo da utilização desses blocos na programação. Por ocasião das atividades realizadas no meio do ano, é considerado que a turma já havia adquirido uma bagagem de conhecimentos iniciais que ajudavam a pensar em soluções mais elaboradas utilizando diferentes encaixes de blocos. Outro grande aumento ocorreu no conceito de paralelismo, que ocorre quando os alunos programam diferentes ações para serem executadas ao mesmo tempo. Essa programação consiste na utilização de blocos que possibilitem o clicar com o mouse ou clique na tecla do teclado, desta forma é acionada mais de uma ação ao mesmo tempo.

Analisando a média das atividades finais, é possível observar que o aumento na maioria dos conceitos continuou, isso significa que os alunos já estavam habituados a utilizar blocos de diferentes funções. Exceto para abstração, que em nenhum momento esteve em alta, pois o bloco clássico que representa este conceito é o clone, que pode ser utilizado para criar cópias de partes da programação, evitando que o mesmo código precise ser programado, e essa funcionalidade por ser avançada, foi enfatizada apenas nas atividades finais, desta forma sendo pouco utilizada.

A cada aula, uma nova problemática era apresentada, e no decorrer do desenvolvimento das programações, as atividades passaram a ser entregues mais rápido e com códigos mais refinados, desta forma possibilitando um grande avanço na aprendizagem de programação em blocos, já que a cada aula novos blocos passaram a ser utilizados e novas funcionalidades implementadas.

Por se tratar de 8 meses de atividades, as faltas obtidas pelos alunos são consideradas um número baixo, porém esse é um fator que influenciou nos resultados obtidos. A tabela 5

mostra o número de faltas que cada um dos 17 alunos acumulou (A1 significa aluno 1, e assim sucessivamente).

Tabela 5. Número de faltas de cada aluno no ano letivo.

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16	A17
5	3	3	0	3	4	3	4	4	1	3	4	2	3	3	6	1

Após a análise geral, os dados foram organizados em 3 grupos de acordo com o número de faltas de cada aluno, com a intenção de analisar se houve algum impacto devido à diferença de número de faltas de alguns alunos. O primeiro grupo analisado foram os alunos com 0 a 2 faltas, que apresentaram um bom desempenho nas 3 etapas das atividades, possibilitando um nítido avanço no gráfico, finalizando com o conhecimento bem distribuído entre os 7 conceitos. Uma possível justificativa para o crescimento gradativo no aprendizado desses alunos é o pequeno número de faltas acumuladas durante o ano, lhes proporcionando muitos momentos de discussões e perguntas durante as aulas online. O gráfico 26 mostra a evolução dos conceitos abordados nas programações. O conceito que passou pela maior queda de acordo com os dados, foi a interatividade, que consiste na utilização de blocos de sensores onde a programação pode ser desenvolvida para que o usuário se comunique com o programa. Responder perguntas é um exemplo do que a interatividade proporciona como programação.

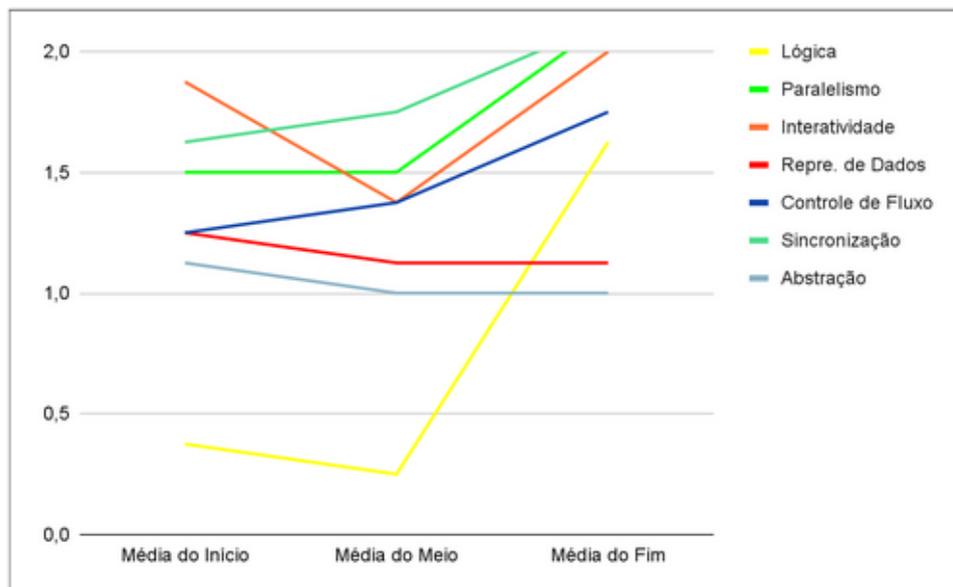


Gráfico 26 - Dados dos alunos com 0 a 2 faltas.

O segundo grupo analisado foram os alunos com 3 e 4 faltas, que iniciaram com pontuação zerada para o conceito de lógica, mas que passaram a evoluir a partir das atividades intermediárias, mediante a presença de alguns pais que passaram a acompanhar todas as aulas

até a conclusão do ano letivo. A primeira aula que contou com a presença de pais chamou a atenção da turma, e a partir disto vários alunos passaram a presenciar a aula acompanhados, o que possivelmente os engajou para adquirir um melhor desempenho, que resultou na evolução do aprendizado, como mostrado no gráfico 27, onde vários conceitos iniciaram em baixa e foram melhorando gradativamente. Exceto o conceito de abstração, que não manteve o valor inicial, sendo possível identificar que os blocos que elevam a pontuação desse conceito não foram utilizados muitas vezes nas programações.

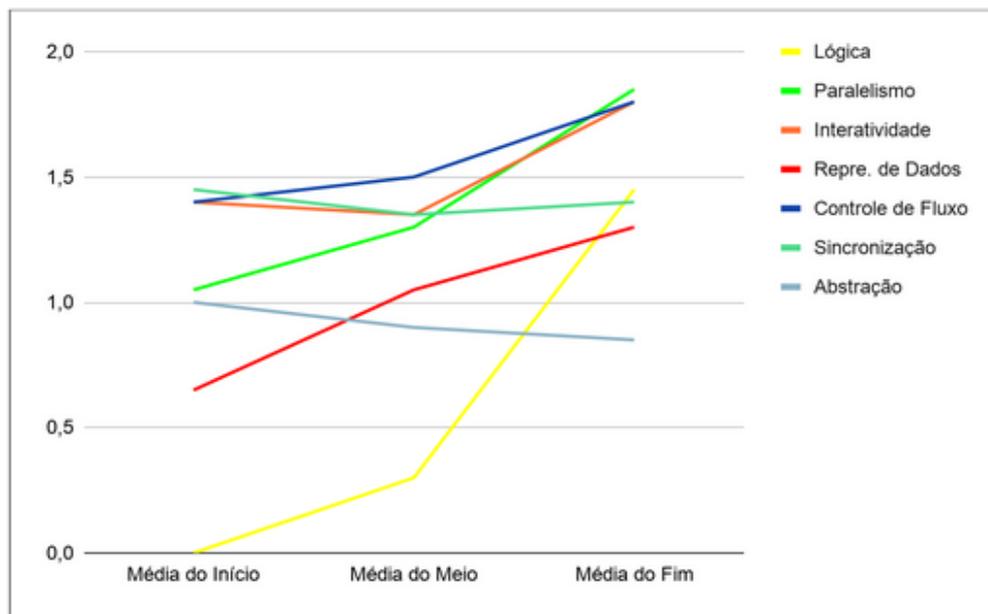


Gráfico 27 - Dados dos alunos com 3 a 4 faltas.

O último grupo analisado foram os alunos com os maiores números de faltas, sendo 5 e 6, apresentando resultados bem abaixo em relação aos outros grupos, pois iniciaram com atividades que não apresentaram pontuação para dois dos conceitos do Dr. Scratch, sendo eles: lógica e paralelismo. Com isso, significa que não foram utilizados blocos de condição e nem programadas funções para serem executadas ao mesmo tempo. No gráfico 28, apesar do grande número de faltas, nas últimas atividades houve um grande avanço na maioria dos conceitos. Uma justificativa para este ocorrido é o número de presenças nas últimas aulas, pois após observarem que as entregas das atividades passaram a conter feedback solicitando melhorias na programação, os alunos passaram a não faltar nas últimas aulas, e aproveitar o momento da aula para esclarecer eventuais dúvidas. Os alunos passaram a assistir às aulas acompanhados dos pais, na intenção de acompanhar o conteúdo que já estava avançado, isso resultou em

aspectos positivos. Os alunos se tornaram mais participativos e conseguiram desenvolver o aprendizado com mais facilidade.

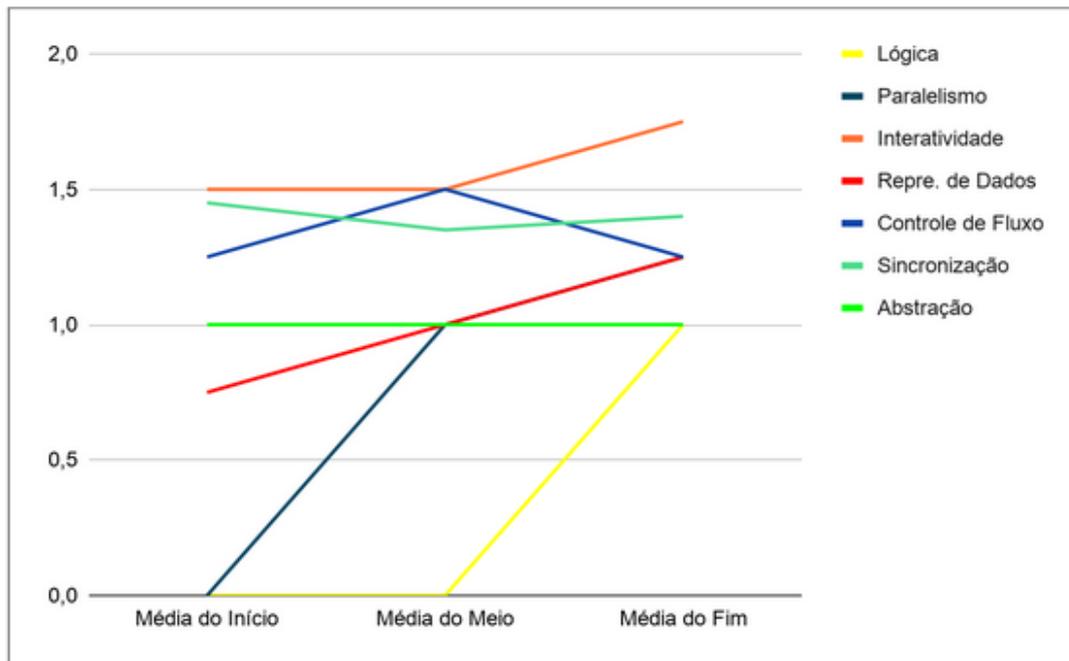


Gráfico 28 - Dados dos alunos com 5 a 6 faltas.

Neste último grupo analisado, o conceito que passou por uma queda foi o controle de fluxo, que é acionado quando a programação é construída para que haja controle de fluxo das ações do personagem, por exemplo determinando quantas vezes ele poderá repetir uma ação.

Com esta análise é possível verificar que no conceito de interatividade os alunos conseguiram ter as melhores pontuações, e torna perceptível em quais precisam melhorar, sendo fundamental para que o professor possa visualizar os progressos dos alunos nas atividades, podendo adaptar suas aulas utilizando diferentes metodologias para suprir as dificuldades apresentadas de acordo com os resultados das avaliações. Porém o processo realizado para chegar a esses resultados se torna inviável quando se trata de uma grande quantidade de turmas e alunos, pois a extração e organização dos dados é um processo demorado que demanda muito tempo, algo que precisaria ser prático para passar a ser utilizado com frequência.

## 6.2 Resultado da Análise dos Gráficos

A fim de suprir as dificuldades de extração e visualização dos dados adquiridos por meio da ferramenta Dr. Scratch, esta pesquisa apresenta uma proposta de visualização de dados que são extraídos automaticamente de tal ferramenta. Sendo assim, os resultados que serão

apresentados nesta seção foram obtidos a partir da aplicação de um formulário, onde os participantes visualizaram todos os gráficos gerados com os resultados das atividades, e avaliaram se essas visualizações gráficas são de fácil entendimento.

Foram obtidas 28 respostas para o formulário, sendo participantes de diversos cursos, conforme apresentado na tabela 6.

Tabela 6 - Graduação dos participantes que responderam o formulário.

<b>Curso</b>	<b>Quantidade</b>
Licenciatura em Computação	10
Ciência da Computação	8
Sistema de Informação	1
Licenciatura em Pedagogia	3
Teologia	1
Economia	1
Licenciatura em Geografia	1
Medicina veterinária	1
Licenciatura em Filosofia	1
Psicologia	1

No formulário, foram apresentados os gráficos obtidos na análise automática, e nestes resultados, a visualização será apresentada juntamente com o gráfico de resultado. Na análise por trimestre, o conceito que menos pontuou no primeiro trimestre foi lógica, e tal visualização presente na figura 14, foi apresentada para que os participantes pudessem visualizar e interpretar, na intenção de verificar a clareza da apresentação dos dados, com isso obteve-se 57,1% de respostas corretas, como podemos verificar no gráfico 29. Já o conceito que mais pontuou foi controle de fluxo, obtendo pontuações 1 ou 2, nas trintas atividades apresentadas no gráfico, tal conceito obteve pontuação zero apenas uma vez.

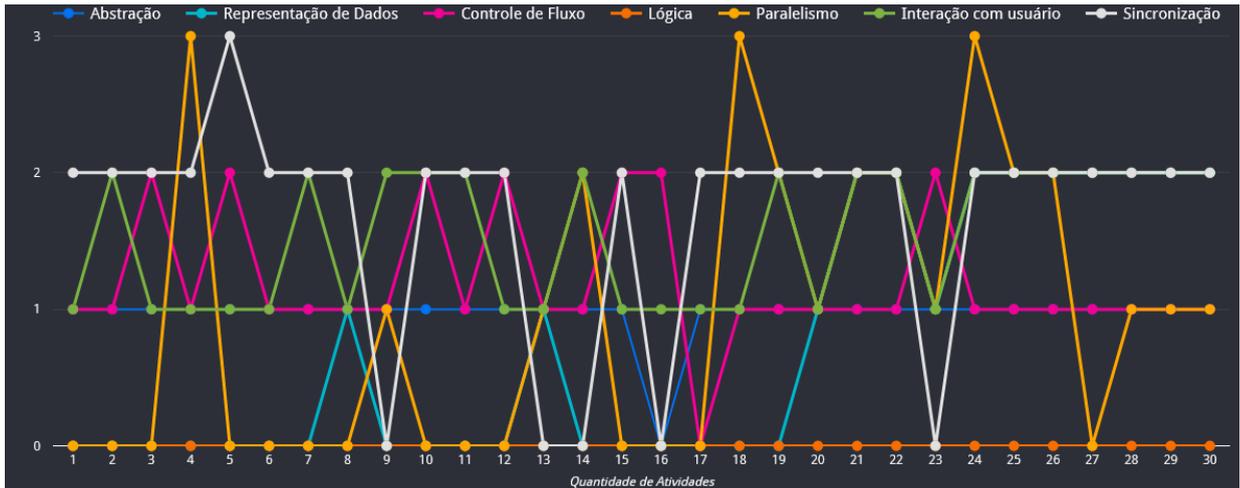


Figura 14 - Visualização do 1º Trimestre.

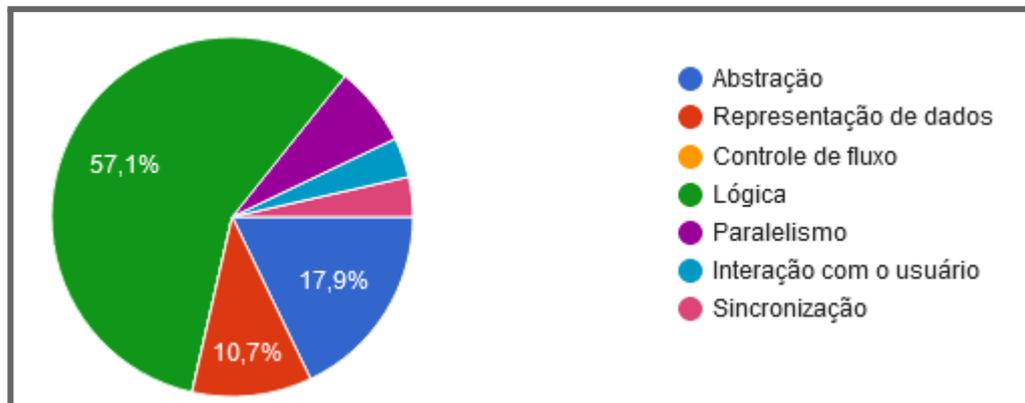


Gráfico 29 - Respostas à pergunta: Qual conceito foi menos pontuado no primeiro trimestre?

Nos dados do segundo trimestre, o conceito que apresentou boa pontuação foi controle de fluxo, conforme representado na visualização da figura 15, e o público precisou analisar e visualizar esta questão, obtendo-se a maioria das respostas corretas, como visto no gráfico 30.

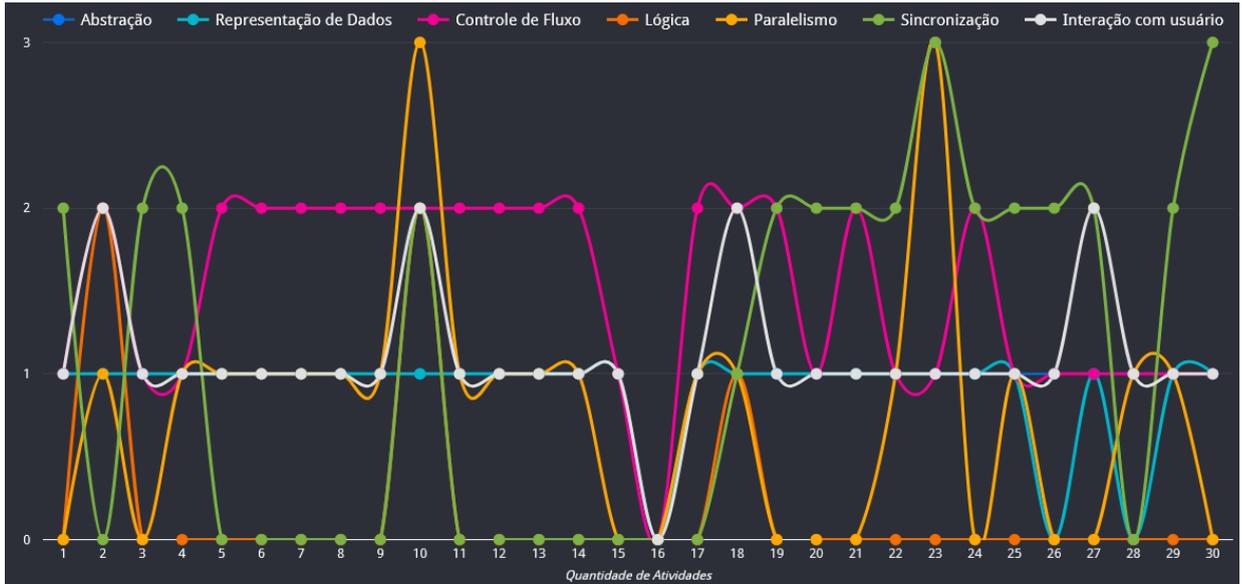


Figura 15 - Visualização do 2º Trimestre.

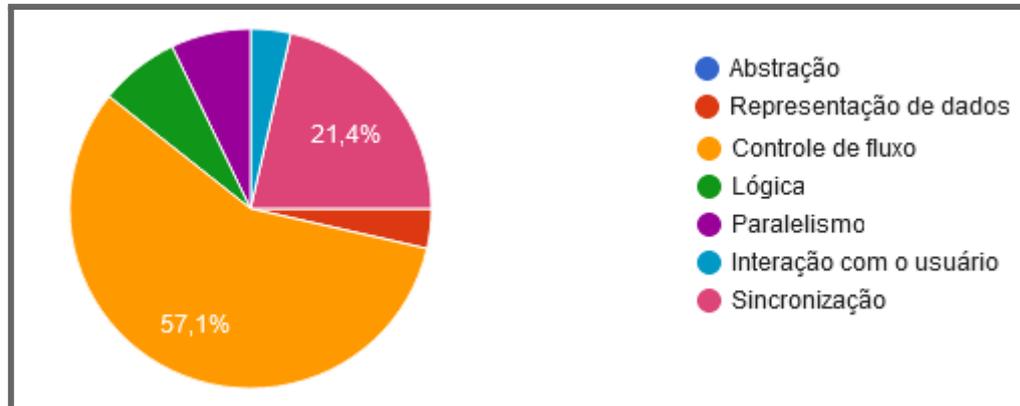


Gráfico 30 - Respostas à pergunta: Qual conceito foi melhor pontuado no segundo trimestre?

Algumas das perguntas possuíam mais de uma alternativa correta, esse detalhe pode ser conferido no formulário presente nos anexos. Para interpretar corretamente o gráfico do terceiro trimestre, os participantes precisavam assinalar as alternativas que continham os conceitos que mais pontuaram neste trimestre, sendo: sincronização e paralelismo, conforme a figura 16, desta forma podemos concluir que os dados foram apresentados com facilidade de interpretação, pois mas uma vez a maioria das respostas foram voltadas para esses dois conceitos, de acordo com o gráfico 31.

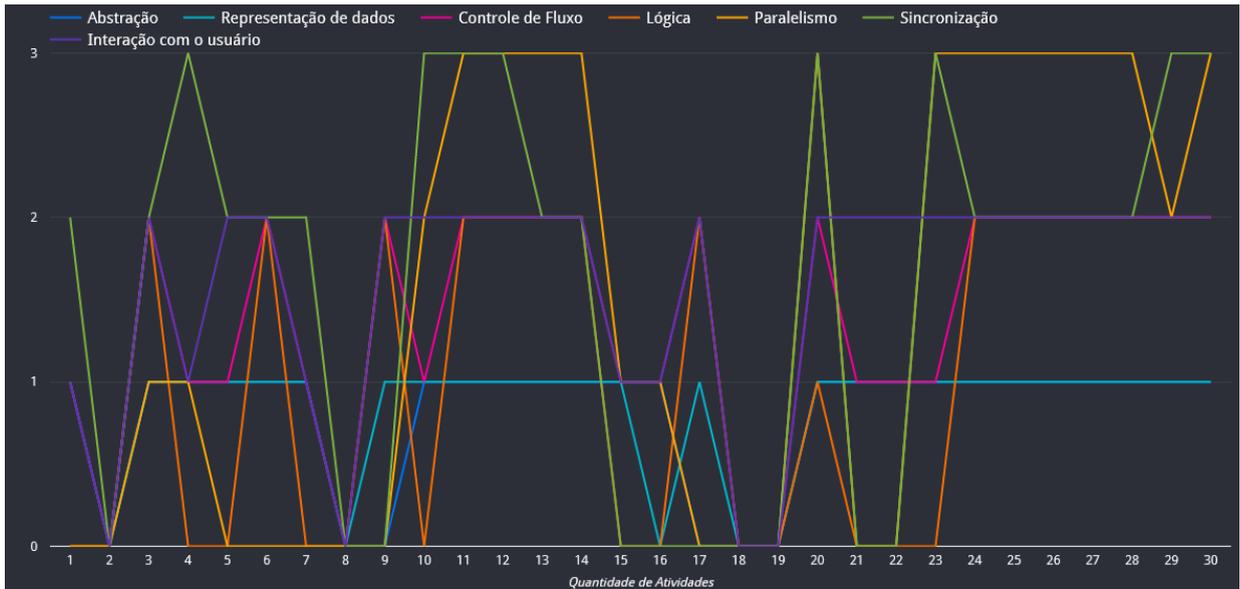


Figura 16 - Visualização do 3º Trimestre.

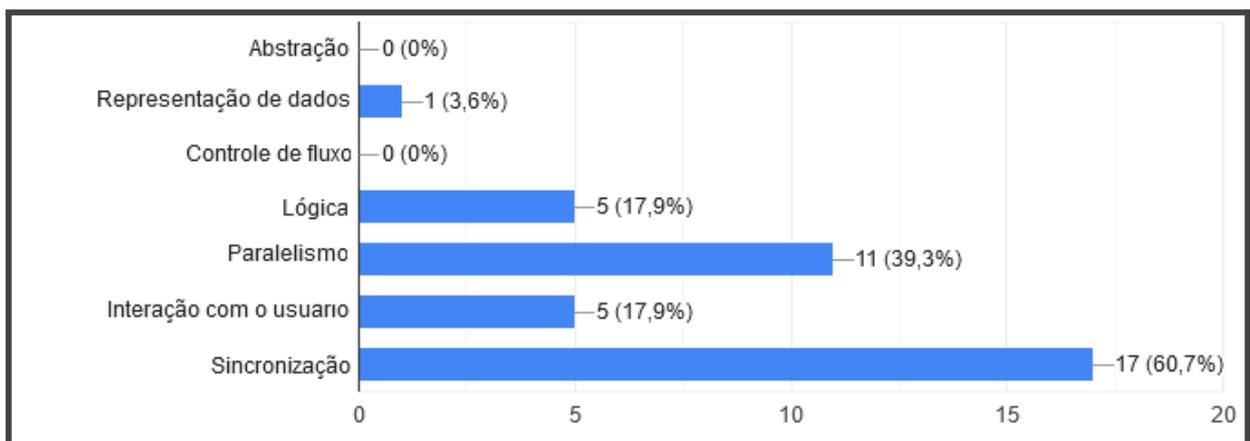


Gráfico 31 -Respostas à pergunta: Quais conceitos mais pontuaram no terceiro trimestre?

Os dados dos três trimestres foram apresentados em gráficos de estilos diferentes, na intenção de verificar qual gráfico é mais fácil de analisar. Os participantes escolheram o gráfico que representou o segundo trimestre, conforme a figura 17, sendo um gráfico de linhas onduladas com pontos exibidos, como o mais simples para visualizar as informações, como mostrado no gráfico 32.

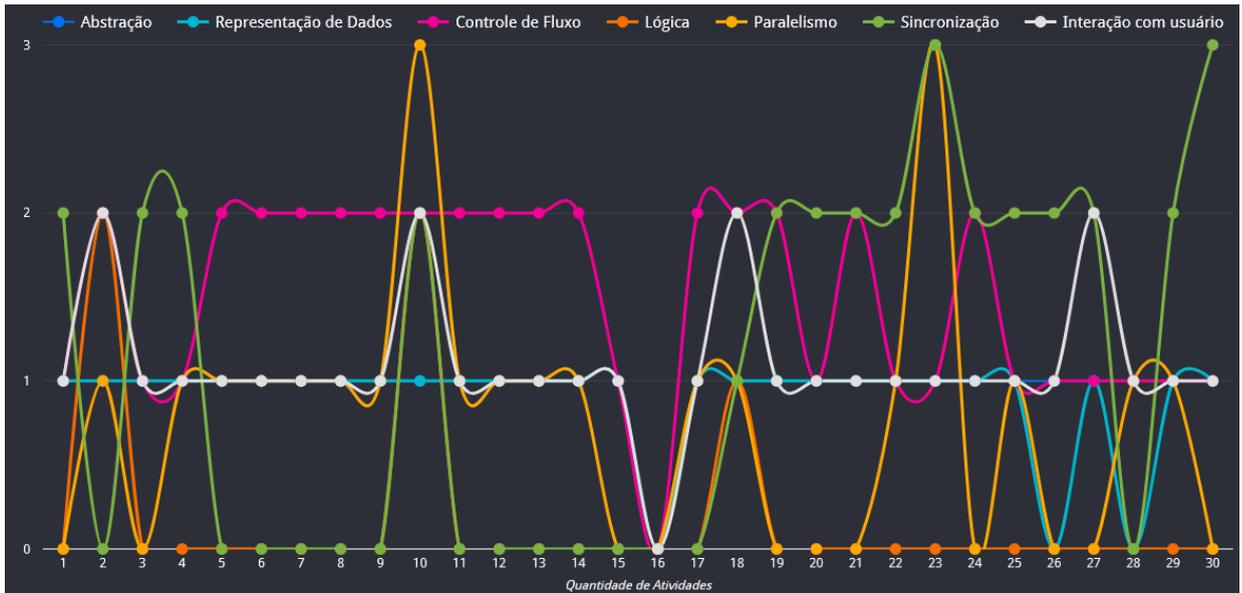


Figura 17 - Estilo de gráfico escolhido como melhor visualização.

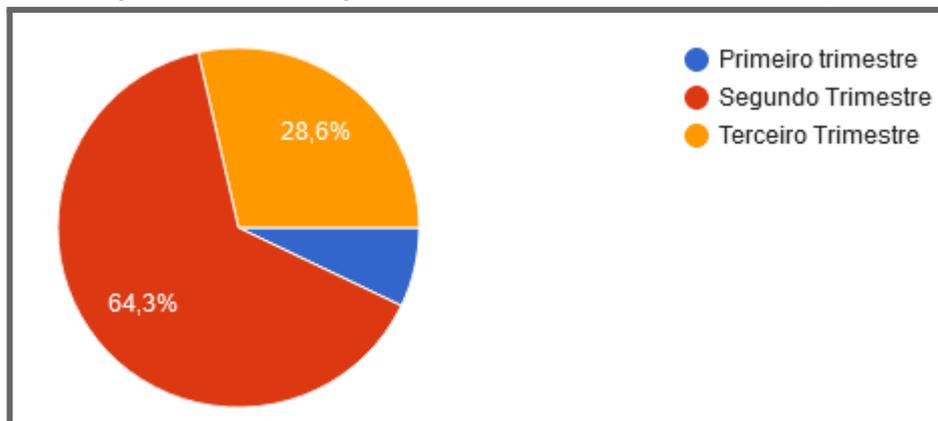


Gráfico 32 - Respostas à pergunta: Qual é o melhor estilo de gráfico?

Analisando a análise dos dados dos alunos que tiveram os maiores números de faltas, representado os participantes precisaram identificar qual conceito apresentou uma pontuação constante na maioria das atividades, sendo o conceito de lógica, que pouco pontuou, tendo uma pontuação constante que se manteve zerada na maioria das atividades, conforme a figura 18. O resultado pode ser observado no gráfico 33.

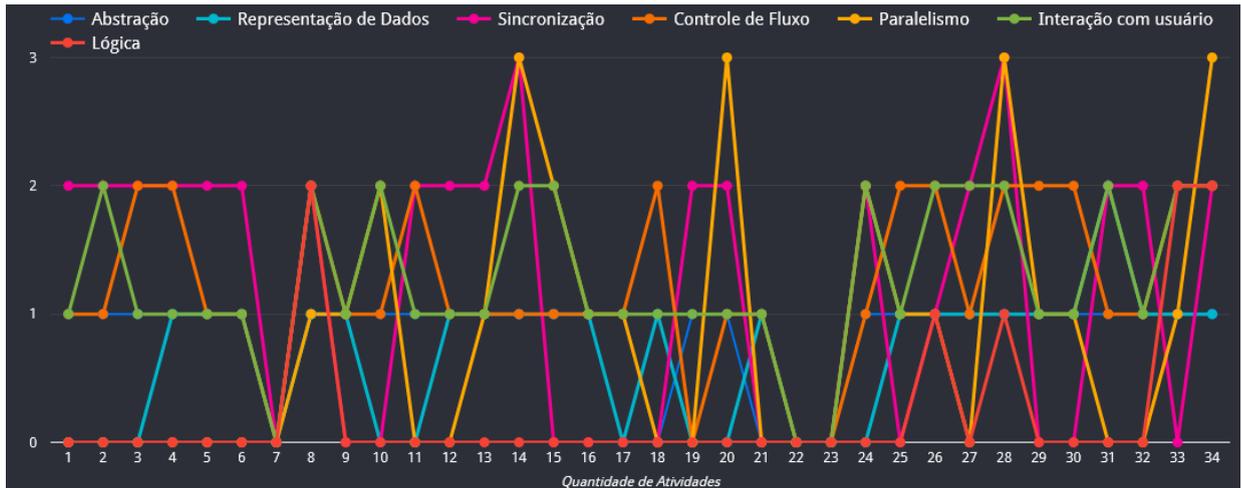


Figura 18 - Visualização da análise por faltas.

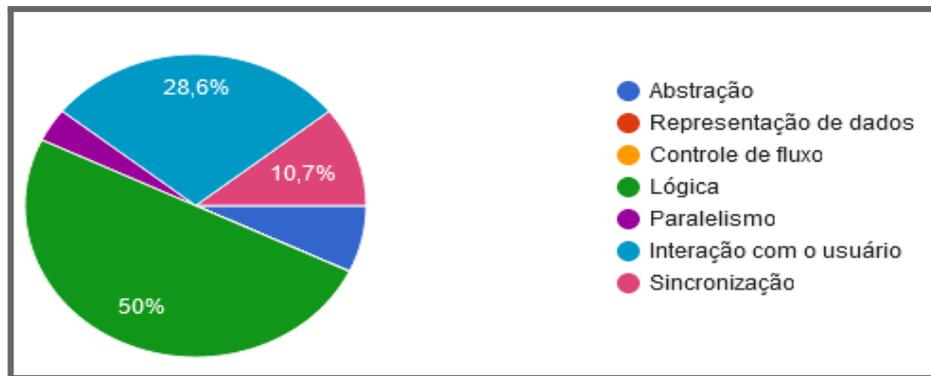


Gráfico 33 - Respostas à pergunta: Qual conceito que se manteve com pontuação constante na maioria das atividades?

Após identificar qual conceito se manteve com a pontuação constante no gráfico de dados dos alunos que tiveram os maiores números de faltas, os participantes analisaram no mesmo gráfico, quantas atividades pontuaram no conceito de lógica, sendo que precisavam assinalar sim ou não, para a pergunta “Analisando o gráfico, é verdade que em apenas 5 atividades a pontuação de lógica não é 0?”, a maioria das respostas foi correta como visto no gráfico 34. Apenas 5 atividades pontuaram no conceito de lógica.

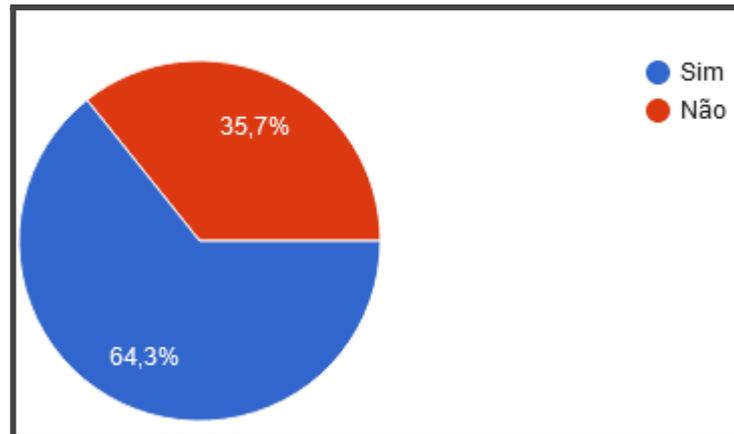


Gráfico 34 - Respostas à pergunta: Em apenas 5 atividades a pontuação de lógica não é 0?

Foram analisados os dados por turma, onde cada representação visual foi interpretada individualmente. Nos dados do segundo ano, os participantes precisaram identificar quais conceitos tiveram pontuação 2, sendo: sincronização, interação com usuário, paralelismo, controle de fluxo e lógica, como podemos observar na figura 19. De acordo com o resultado do gráfico 35, os participantes conseguiram analisar e identificar tais conceitos.

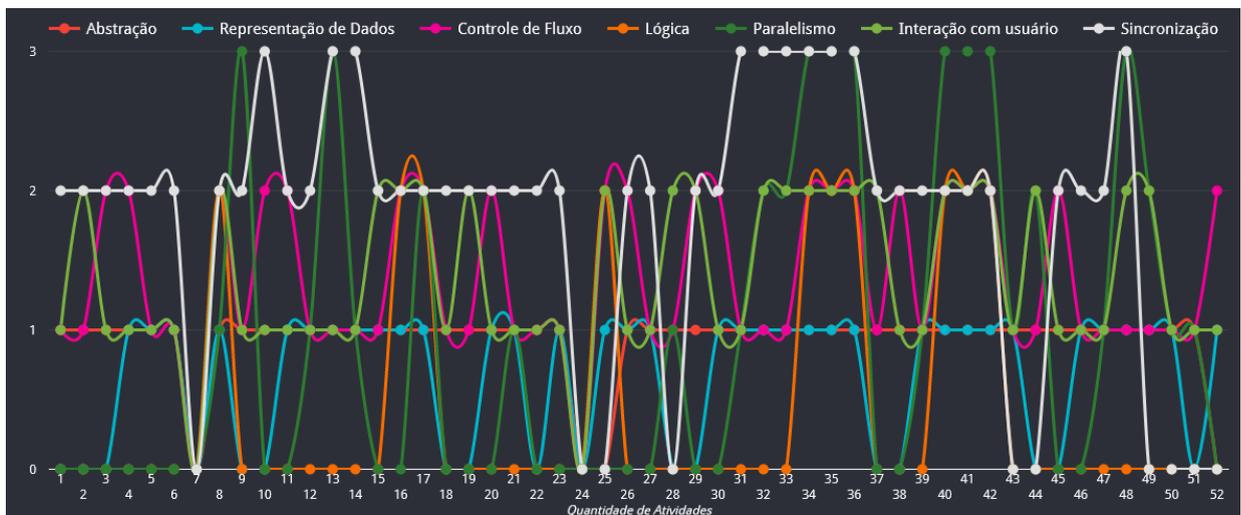


Figura 19 - Visualização do 2º Ano.

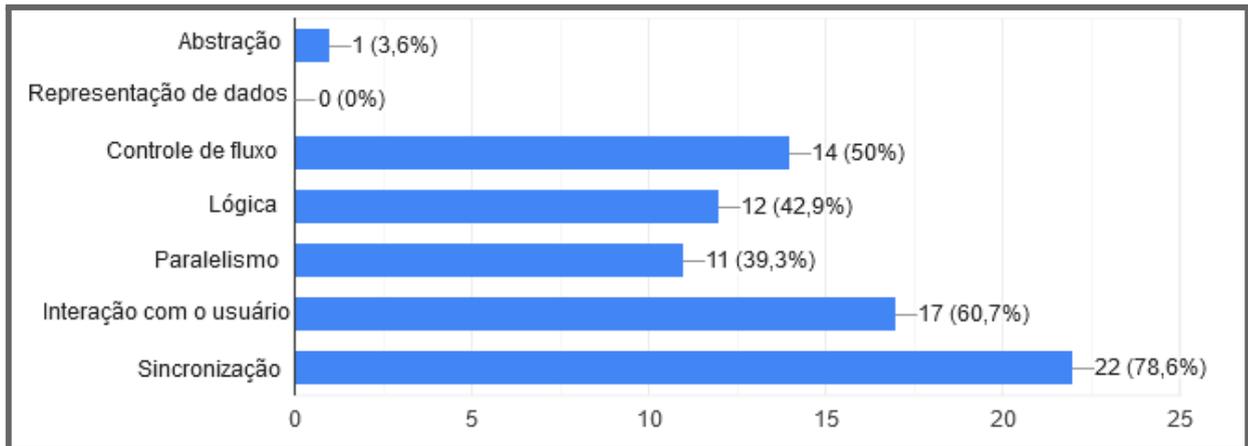


Gráfico 35 - Respostas à pergunta: Na análise do segundo ano, quais conceitos tiveram pontuação 2?

Analisando os dados do segundo ano, os participantes verificaram que a resposta da pergunta “é verdade que 9 atividades não pontuaram no conceito sincronização?” é negativa, pois foram 10 atividades que não pontuaram, desta forma a maioria das respostas foi correta, conforme o gráfico 36.

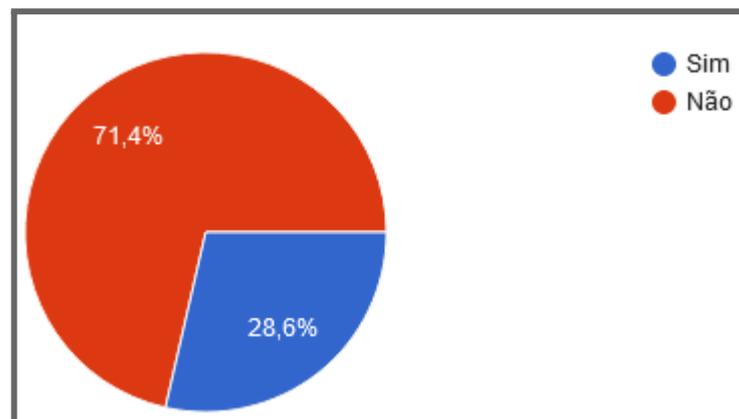


Gráfico 36 - Respostas à pergunta: Observando o gráfico do 2º ano, é verdade que 9 atividades não pontuaram no conceito sincronização?

Ainda na análise do segundo ano, os participantes possivelmente não se atentaram à cor que representa o conceito de representação de dados, pois tal conceito não pontuou mais que 1 em nenhuma atividade, e ao responder a pergunta “é verdade que nenhuma atividade pontuou mais que 1, no conceito representação de dados?”, as respostas ficaram divididas em 50% para cada opção como visto no gráfico 37, sendo que a resposta é sim, pois não houve pontuação maior que 1.

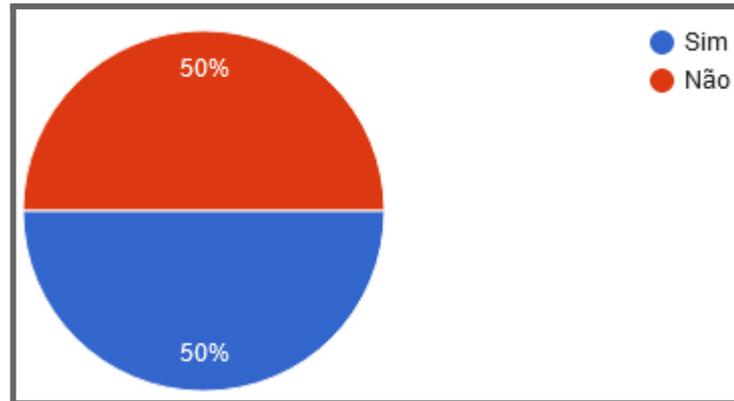


Gráfico 37 - Respostas à pergunta: Observando o gráfico do 2º ano, é verdade que nenhuma atividade pontuou mais que 1, no conceito representação de dados?

Nos dados apresentados da análise do terceiro ano, apenas uma atividade não pontuou em vários conceitos, com isso os participantes precisavam analisar e assinalar todos os conceitos que obtiveram pontuação zero em uma única atividade, sendo eles: lógica, abstração, sincronização e paralelismo, conforme a visualização da figura 20. De acordo com o gráfico 38, os resultados foram negativos, já que o conceito com maior porcentagem não está incluso entre as opções corretas.

O gráfico que apresenta os dados do terceiro ano não está no estilo escolhido pelos participantes como melhor opção de visualização, este é um fator que pode ter influenciado as respostas para as questões desta turma.



Figura 20 - Visualização do 3º Ano.

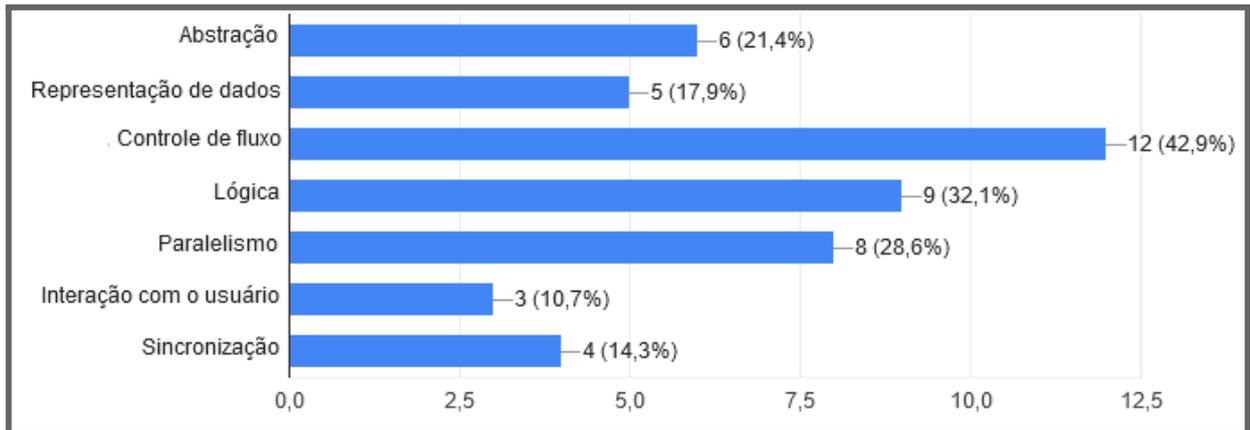


Gráfico 38 - Respostas à pergunta: Na análise do 3º ano, quais conceitos não pontuaram em apenas 1 atividade?

Um novo gráfico foi gerado com dados da turma do terceiro ano, onde os participantes precisavam analisar a visualização das pontuações das atividades e verificar quantas atividades pontuaram no conceito de lógica, sendo apenas duas atividades. Os participantes responderam a pergunta “é verdade que em apenas 1 atividade houve pontuação em lógica?” obtendo 50% para cada opção de resposta, conforme apresentado no gráfico 39. Com isso pode-se entender que as perguntas para tal turma obtiveram resultados intermediários. Mesmo a visualização sendo simples e com poucas atividades, houve dificuldade na interpretação dos dados.

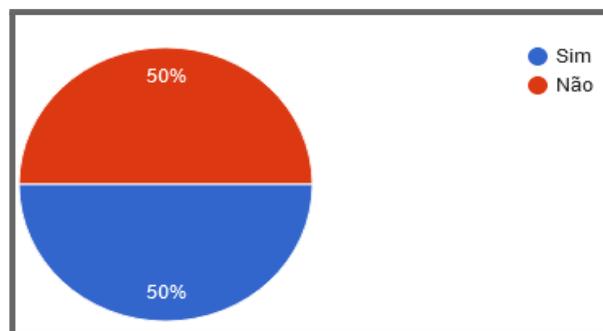


Gráfico 39 - Respostas à pergunta: Nas atividades do 3º ano, é verdade que em apenas 1 atividade houve pontuação em lógica?

Analisando os dados das atividades do quarto ano, os participantes precisaram verificar quais foram os conceitos que mais pontuaram nas atividades, sendo: interação com o usuário, controle de fluxo, representação de dados e abstração, que pontuaram em todas as 26 atividades, conforme a figura 21. De acordo com o resultado do gráfico 40, a análise dos participantes foi intermediária, pois o conceito de paralelismo obteve uma grande porcentagem de resposta, e tal conceito não pontuou em muitas atividades. Os dados desta turma não foram apresentados no estilo de gráfico com melhor visualização, conforme a opinião dos participantes.

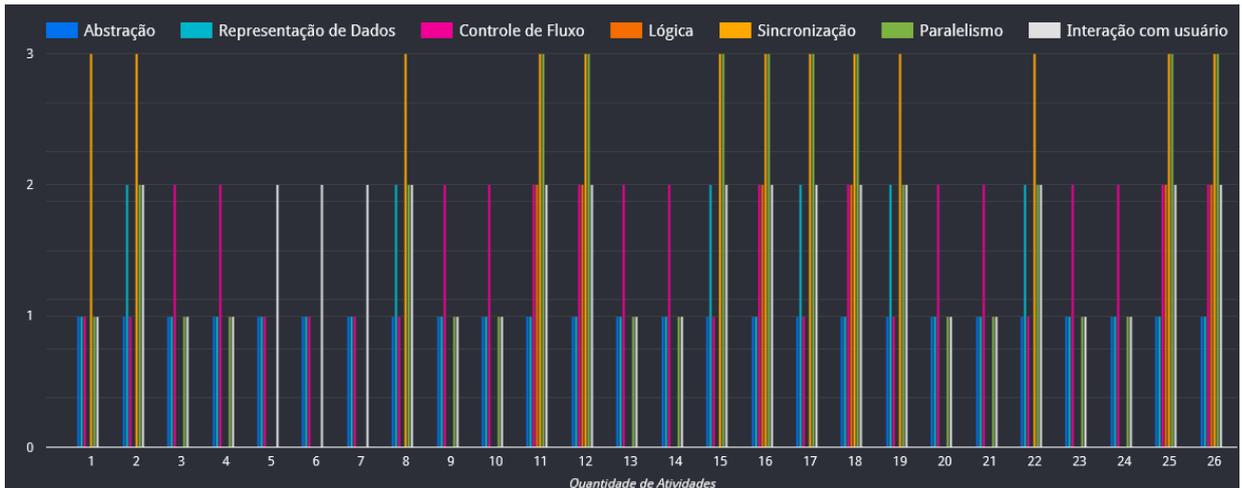


Figura 21 - Visualização do 4º Ano.

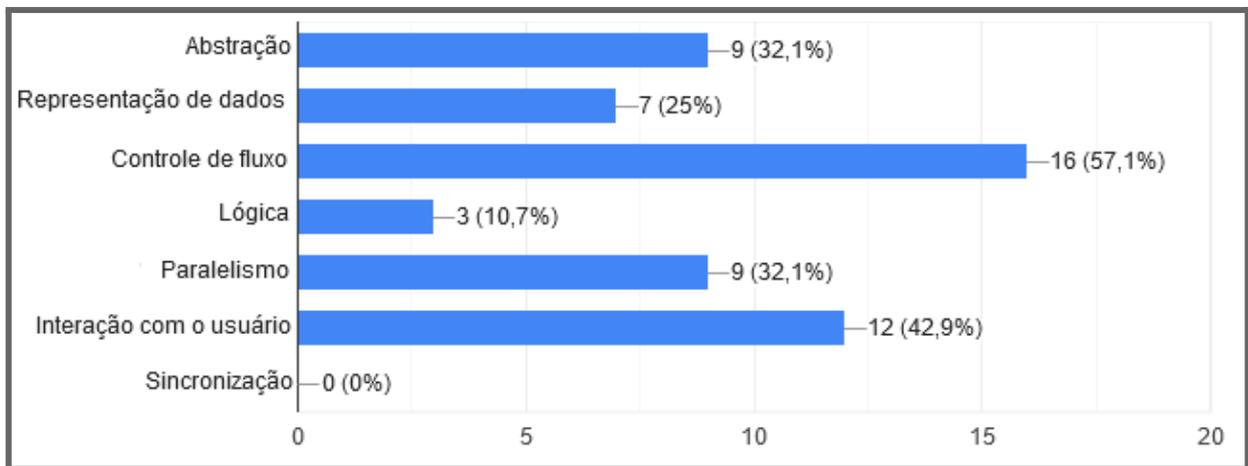


Gráfico 40 - Respostas à pergunta: Nas atividades do 4º ano, quais dos conceitos pontuou em um número maior de atividades?

Os dados do quinto ano representam poucas atividades, e foram apresentadas no melhor estilo de gráfico escolhido pelos participantes, conforme a figura 22. Foi necessário observar qual conceito não pontuou em 6 atividades, sendo o paralelismo. Conforme os resultados do gráfico 41, a maioria das respostas foi correta, onde 46,4% dos participantes conseguiram identificar corretamente o conceito.

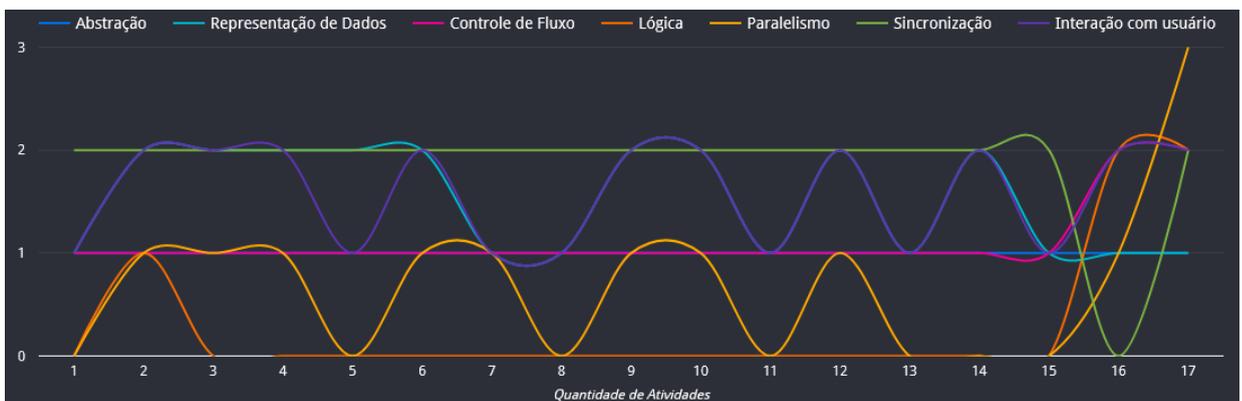


Figura 22 - Visualização do 5º Ano.

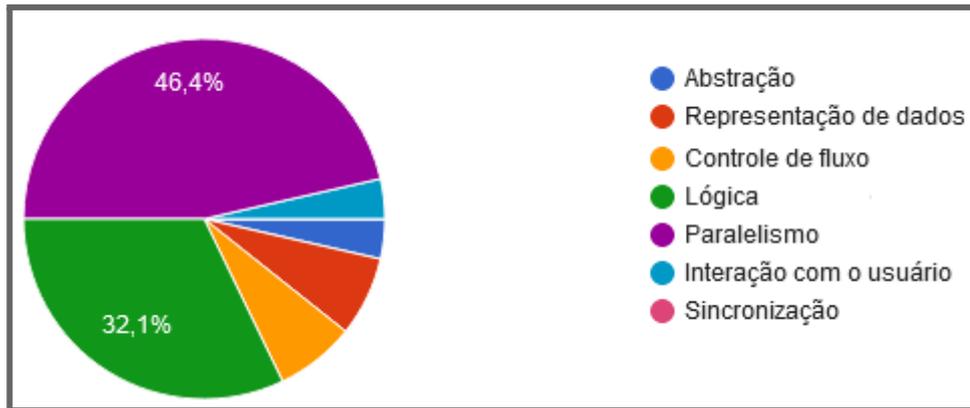


Gráfico 41 - Respostas à pergunta: Nas atividades do 5º ano, qual conceito não houve pontuação em 6 atividades?

Os dados apresentam de forma clara qual que apenas uma atividade não pontuou no conceito de sincronização, com isso os participantes precisaram observar e responder a pergunta “é verdade que apenas uma atividade não pontuou no conceito sincronização?”. Conforme o resultado do gráfico 42, 89,3% das respostas foram corretas, indicando que sim, apenas uma atividade não pontuou em tal conceito.

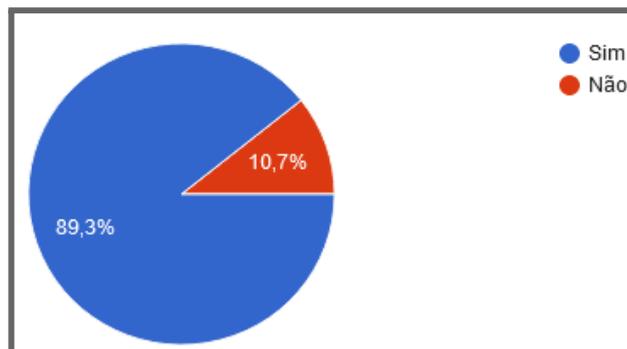


Gráfico 42 - Respostas à pergunta: Referente ao 5º ano, é verdade que apenas uma atividade não pontuou no conceito sincronização?

Na intenção de verificar qual estilo de gráfico é mais simples de analisar os dados, cada turma teve suas informações apresentadas em gráficos diferentes, onde o estilo de visualização é diferente, e os participantes foram questionados sobre qual estilo de gráfico apresentou melhor os dados, tal pergunta havia sido respondida apenas na análise das atividades por trimestre. Neste caso, foram dados de 4 turmas, sendo quatro visualizações diferentes, e os participantes escolheram que a representação da turma do quinto ano, apresentada na figura 23, foi a mais clara e objetiva, conforme o gráfico 43.

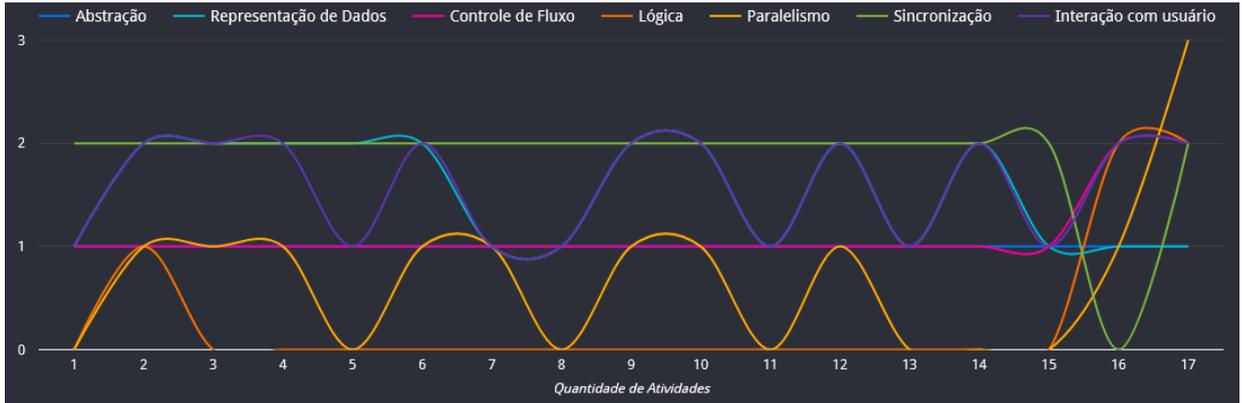


Figura 23 - Estilo de gráfico escolhido como melhor visualização entre as quatro turmas.

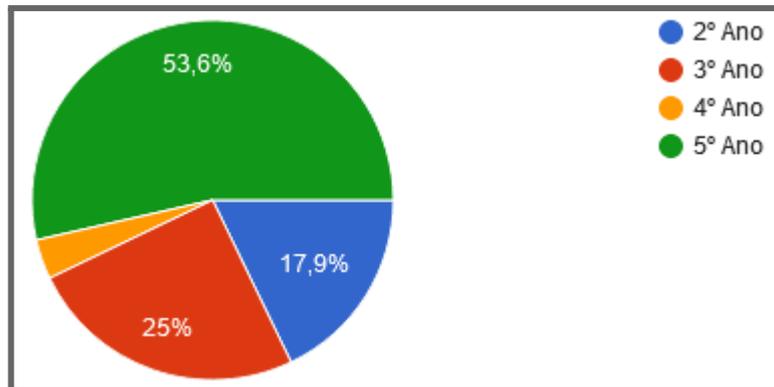


Gráfico 43 - Respostas à pergunta: Qual dos 4 estilos de gráficos ficou mais fácil de visualizar?

Diversas atividades tiveram seus dados apresentados, cada turma com uma quantidade diferente de atividades. Os participantes foram questionados sobre qual turma apresentou o maior número de atividades, sendo o segundo ano com 52 atividades, presente na figura 24. Desta forma, obteve-se 57,1% de respostas corretas, conforme o gráfico 44.

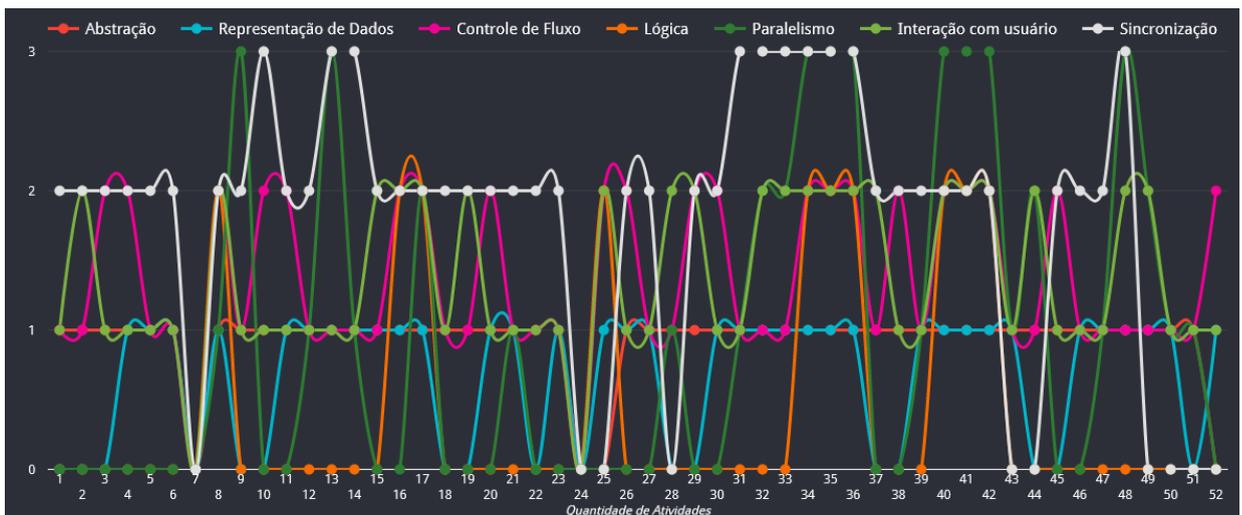


Figura 24 - Visualização que mais apresentou atividades.

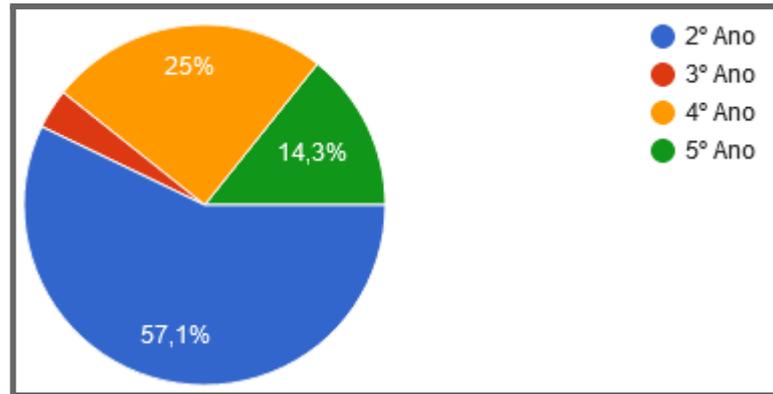


Gráfico 44 - Respostas à pergunta: Qual dos anos, teve o maior número de atividades avaliadas?

Os dados das atividades de cada aluno também fizeram parte das perguntas no questionário. Analisando o gráfico de atividades do primeiro aluno, a atividade 7 não pontuou em nenhum conceito, pelo projeto só apresentar a parte gráfica desenvolvida, conforme a figura 25, com isso o questionamento para os participantes foi “qual pontuação está sendo representada na atividade 7?”. 53,6% das respostas foram corretas, como apresentado no gráfico 45.

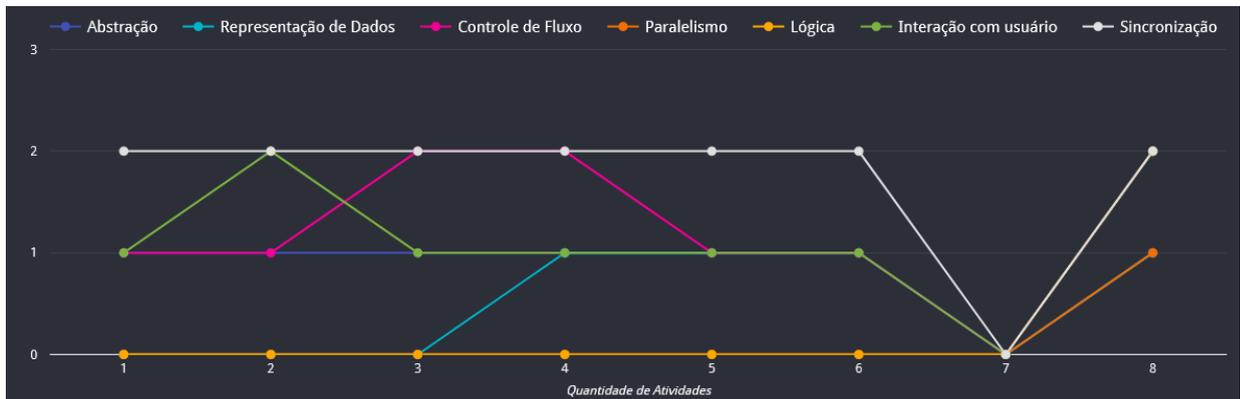


Figura 25 - Visualização referente ao 1º Aluno.

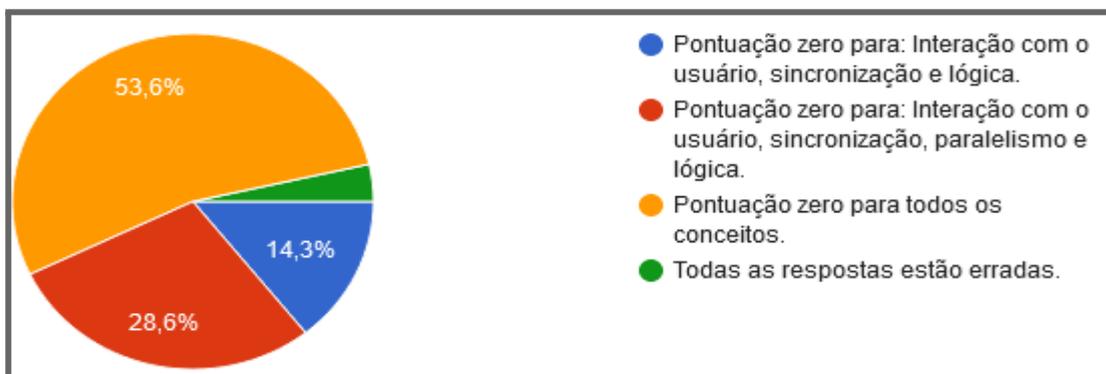


Gráfico 45 - Respostas à pergunta: Nas atividades do 1º aluno, qual pontuação está sendo representada na atividade 7.

Analisando o gráfico de atividades do segundo aluno, todas as atividades pontuaram no conceito de controle de fluxo, conforme apresentado na figura 26, ao serem questionados sobre “é verdade que em pelo menos 1 atividade o conceito de controle de fluxo foi zero?”, 78,6% dos participantes conseguiram identificar a resposta correta, conforme o gráfico 46.

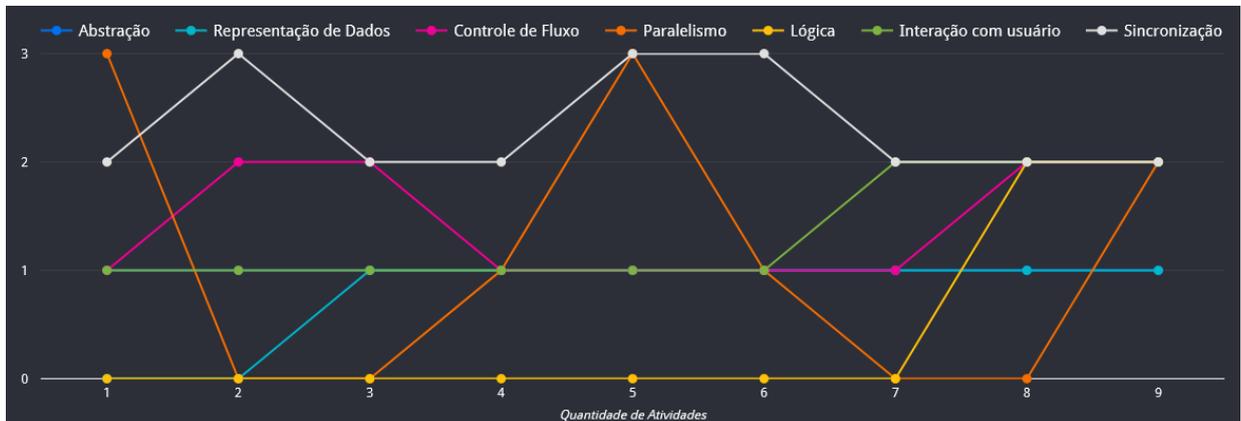


Figura 26 - Visualização referente ao 2º Aluno.

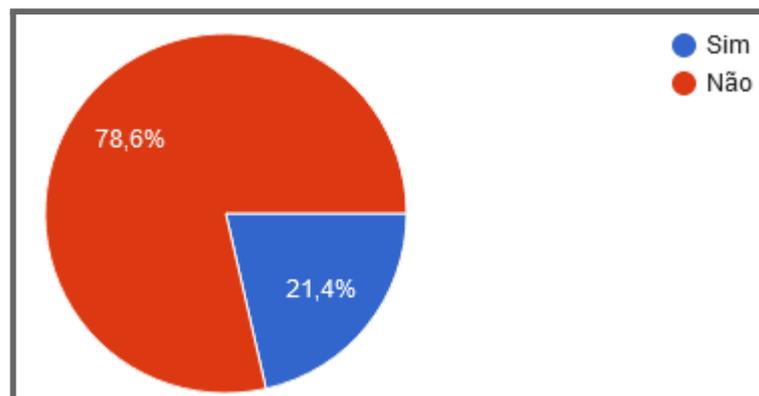


Gráfico 46 - Respostas à pergunta: Nas atividades do 2º aluno, é verdade que em pelo menos 1 atividade o conceito de controle de fluxo foi zero?

Nas atividades do terceiro aluno, os conceitos de abstração, paralelismo e sincronização obtiveram pontuação zero na atividade 7, conforme a figura 27, com isso os participantes foram questionados sobre quais conceitos não pontuaram em tal atividade. Verificando o resultado das respostas no gráfico 47, é nítido que os participantes conseguiram identificar os conceitos corretamente.

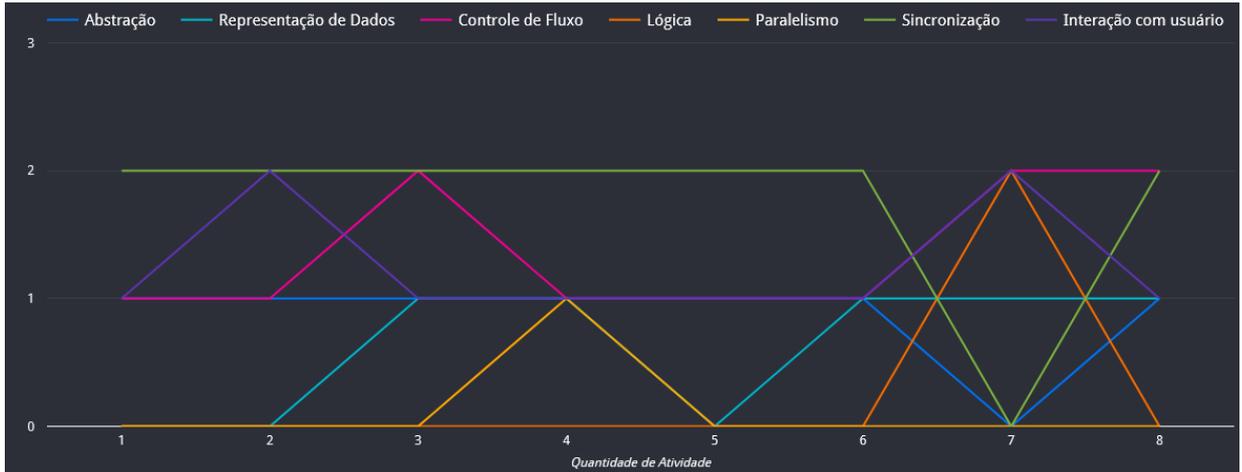


Figura 27 - Visualização referente ao 3º Aluno.

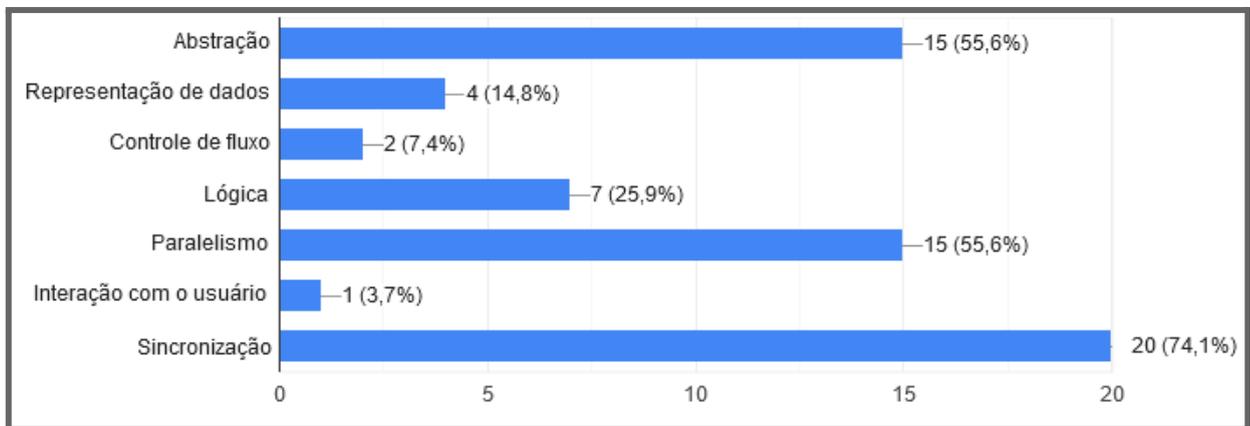


Gráfico 47 - Respostas à pergunta: Nas atividades do 3º aluno, quais conceitos tiveram a pontuação zerada na atividade 7?

O quarto aluno obteve pontuação no conceito de lógica em poucas atividades, onde a pontuação se manteve zerada por 6 atividades seguidas, conforme a figura 28, com isso os participantes precisaram analisar a visualização dos dados e verificar se “é verdade que o maior número de atividades seguidas com pontuação zero no conceito de lógica é 3?”. De acordo com o gráfico 48, a maioria das respostas comprova que a representação dos dados foi clara, já que as respostas coincidem com o esperado.



Figura 28 - Visualização referente ao 4º Aluno.

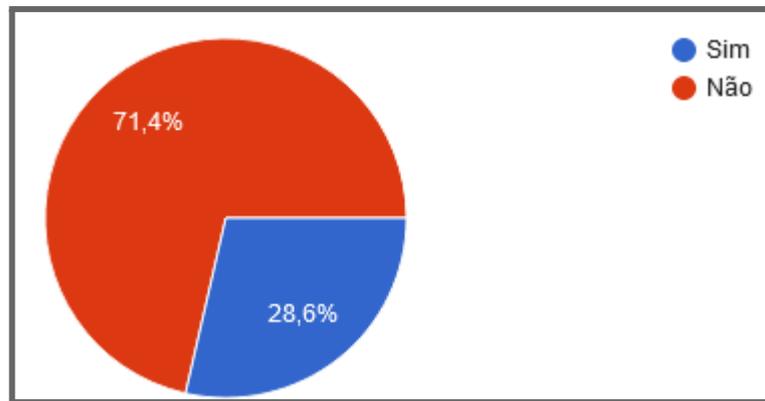


Gráfico 48 - Respostas à pergunta: Nas atividades do 4º aluno, é verdade que o maior número de atividades seguidas com pontuação zero no conceito de lógica é 3?

Analisando o gráfico de atividades do quinto aluno, foram poucas atividades entregues, porém vários conceitos estavam presentes em todas as atividades, sendo: abstração, controle de fluxo, interação com usuário e sincronização, conforme a figura 29. Os participantes verificaram os dados para responder o seguinte questionamento: "quais conceitos estão presentes em todas as atividades?". Conforme os resultados presentes no gráfico 49, pode-se observar que as respostas foram em sua maioria positivas, onde os conceitos corretos possuem as maiores porcentagens.

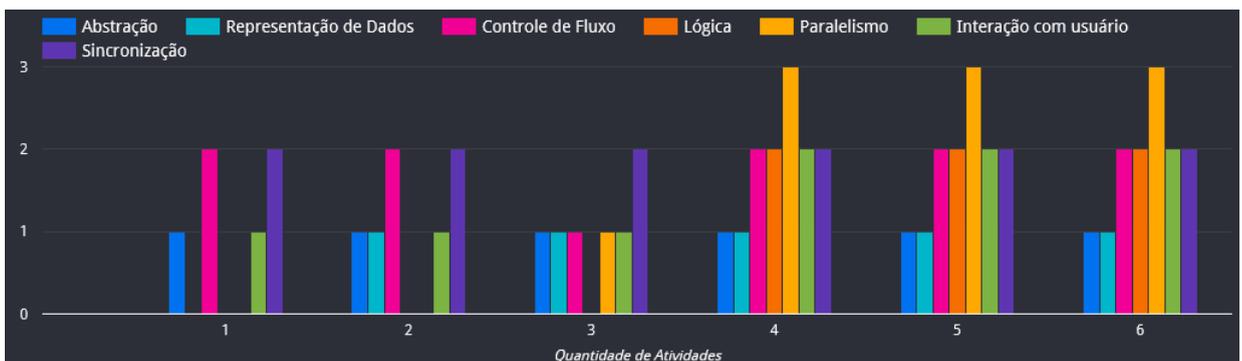


Figura 29 - Visualização referente ao 5º Aluno.

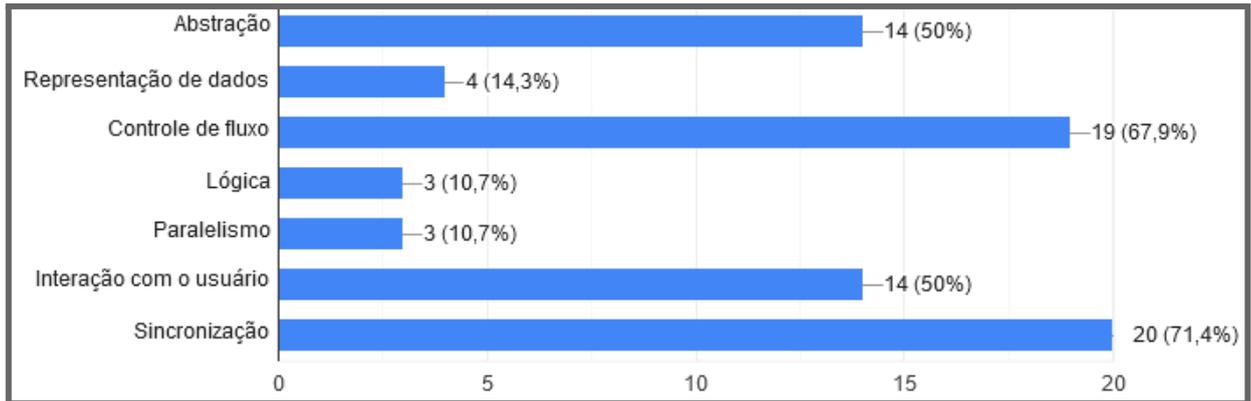


Gráfico 49 - Respostas à pergunta: Nas atividades do 5º aluno, quais conceitos estão presentes em todas as atividades?

O sexto aluno não pontuou no conceito de lógica em nenhuma atividade, conforme a figura 30, desta forma não obteve nenhuma atividade com pontuação nos 7 conceitos, levando em consideração esse ponto a ser observado, os participantes foram questionados sobre “é verdade que nenhuma atividade pontuou nos 7 conceitos?”. A resposta correta seria “sim”, porém a maioria das respostas foi “não”, conforme o gráfico 50. Desta forma, fica claro que o estilo de gráfico que apresentou os dados de tal aluno não proporcionou uma boa visualização para análise.

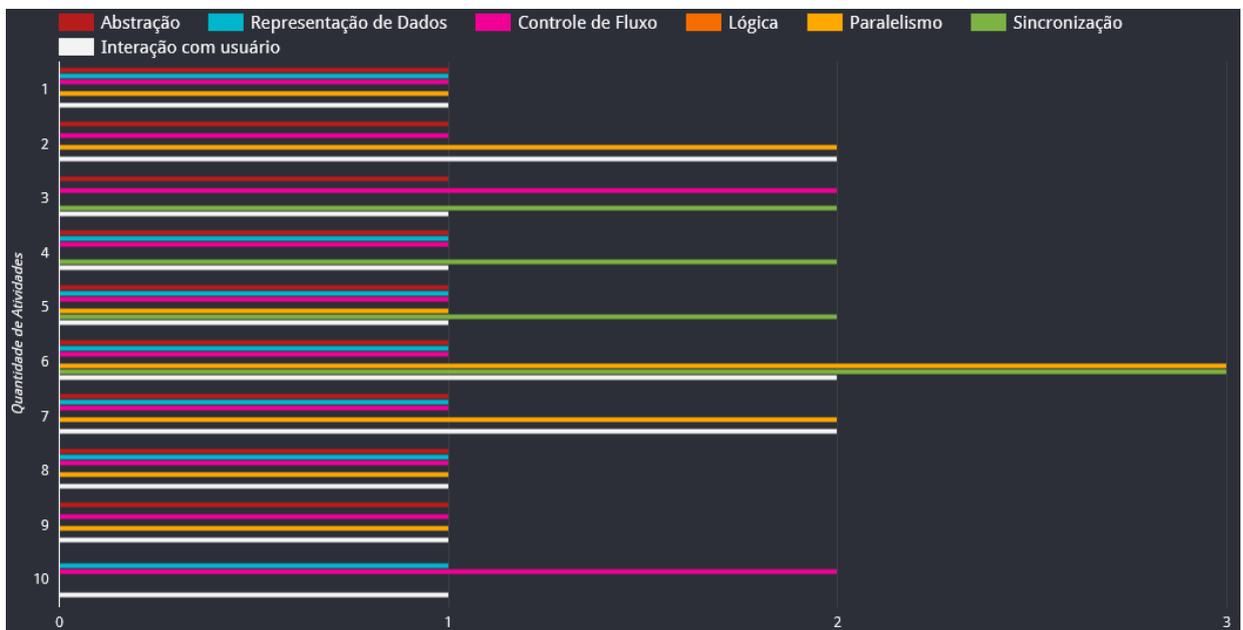


Figura 30 - Visualização referente ao 6º Aluno.

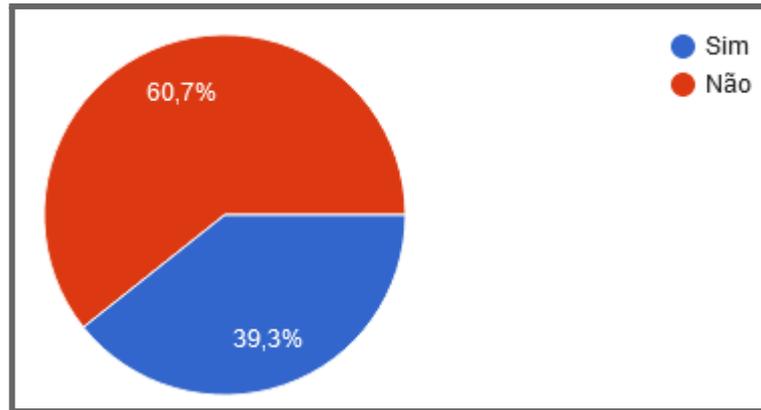


Gráfico 50 - Respostas à pergunta: Nas atividades do 6º aluno, é verdade que nenhuma atividade pontuou nos 7 conceitos?

Analisando o gráfico de atividades do sétimo aluno, dois conceitos predominaram em todas as atividades, sendo: controle de fluxo e interação com o usuário, conforme a figura 31. Os participantes precisaram verificar quais conceitos estavam presentes em todas as atividades. Com isso, pode-se observar que os resultados apresentados no gráfico 51 correspondem às respostas esperadas, onde os participantes conseguiram identificar os dois conceitos em questão.

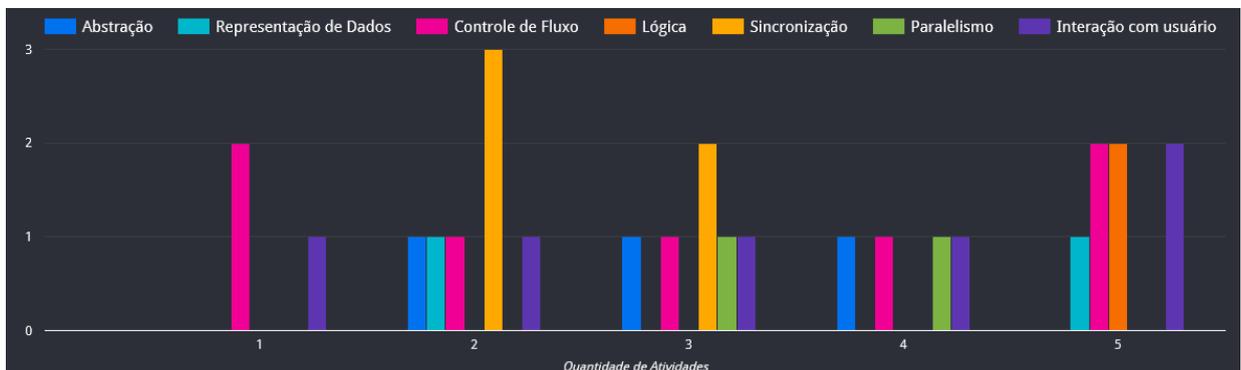


Figura 31 - Visualização referente ao 7º Aluno.

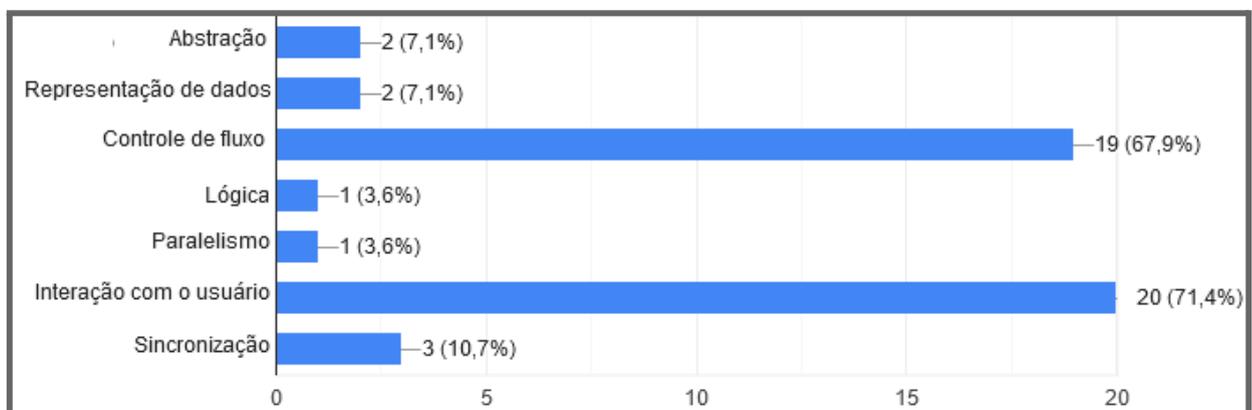


Gráfico 51 - Respostas à pergunta: Na análise dos dados do 7º aluno, quais conceitos estão presentes em todas as atividades?

O oitavo aluno entregou uma atividade sem nenhuma programação, apenas com o design montado, com isso obteve pontuação zero para todos os conceitos na atividade 4, conforme a figura 32. Os participantes foram questionados sobre o que ocorreu com os dados da atividade 4, e de acordo com os resultados apresentados no gráfico 52. A maioria das respostas coincidiu com o esperado, já que 53,6% conseguiram analisar e identificar corretamente o fator ocorrido com tal atividade.

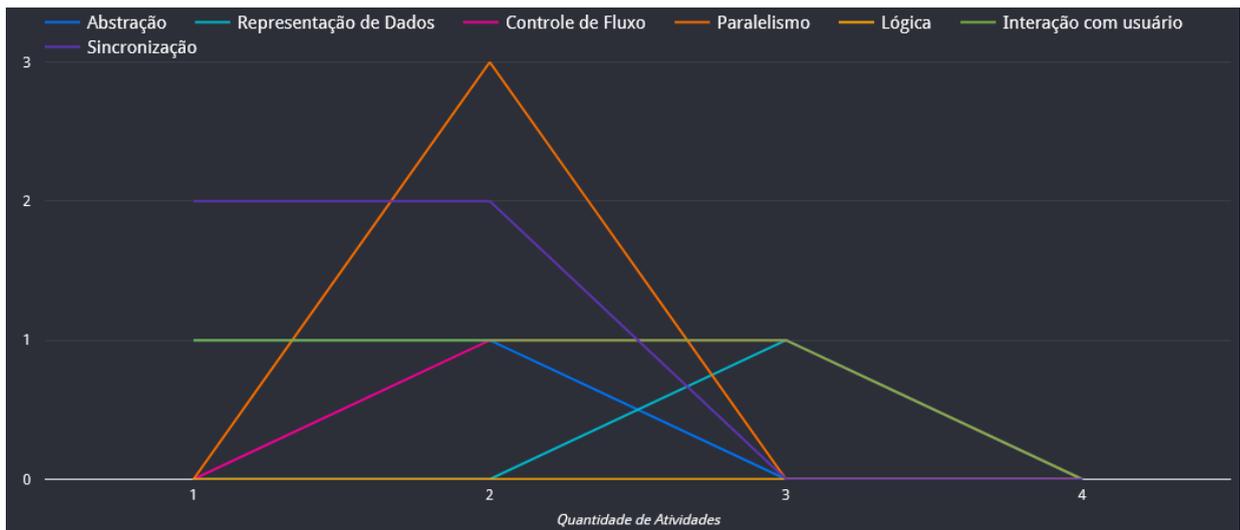


Figura 32 - Visualização referente ao 8º Aluno.

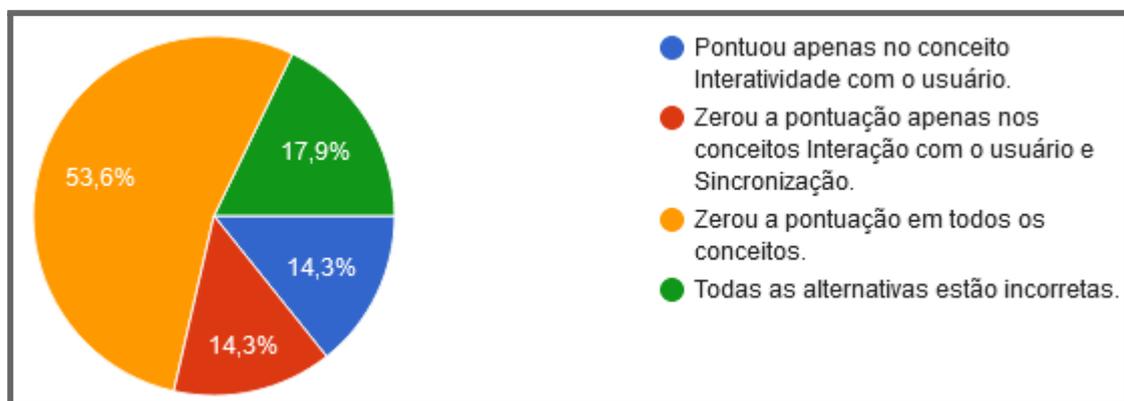


Gráfico 52 - Respostas à pergunta: Na análise dos dados do 8º aluno, o que ocorre com os dados da atividade 4?

Analisando o gráfico de atividades do nono aluno, o conceito de sincronização progrediu na pontuação a partir da segunda atividade, onde passou por um declínio para em seguida apenas progredir nas seguintes atividades, observe na figura 33. Visualizando este ponto, os participantes foram questionados sobre “é verdade que desde a primeira atividade o conceito de sincronização apenas progrediu na pontuação?”. Conforme o resultado do gráfico 53, os participantes conseguiram verificar claramente, pois a maioria das respostas foi correta.

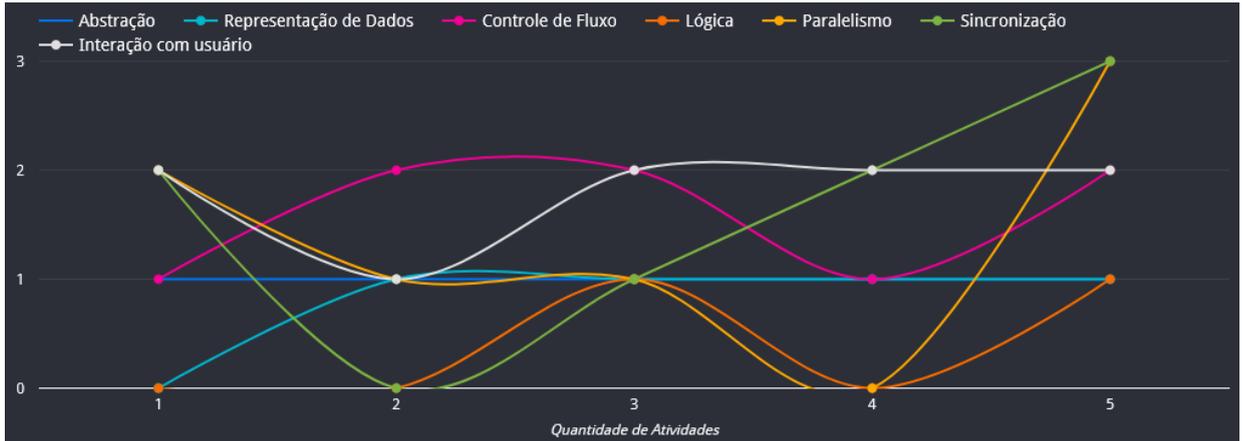


Figura 33 - Visualização referente ao 9º Aluno.

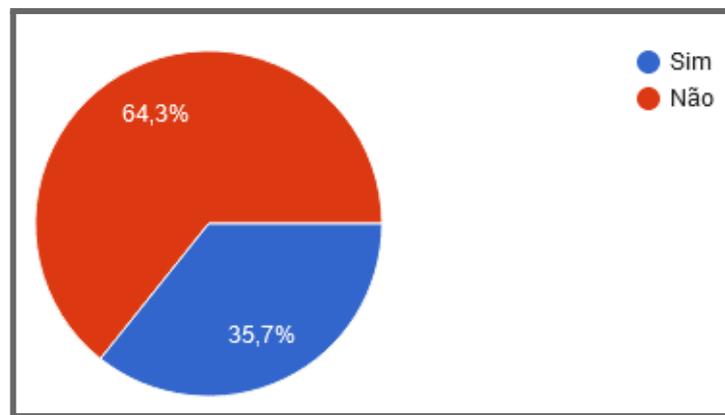


Gráfico 53 - Respostas à pergunta: Na análise dos dados do 9º aluno, é verdade que desde a primeira atividade o conceito de sincronização apenas progrediu na pontuação?

Nos dados das atividades do décimo aluno, o conceito de lógica não pontuou em nenhuma atividade, sendo o único com esta representação, presente na figura 34. Os participantes precisaram verificar qual conceito não pontuou em nenhuma atividade, e os resultados do gráfico 54 demonstram que 71,4% dos participantes conseguiram analisar corretamente os dados apresentados, identificando o conceito correto.



Figura 34 - Visualização referente ao 10º Aluno.

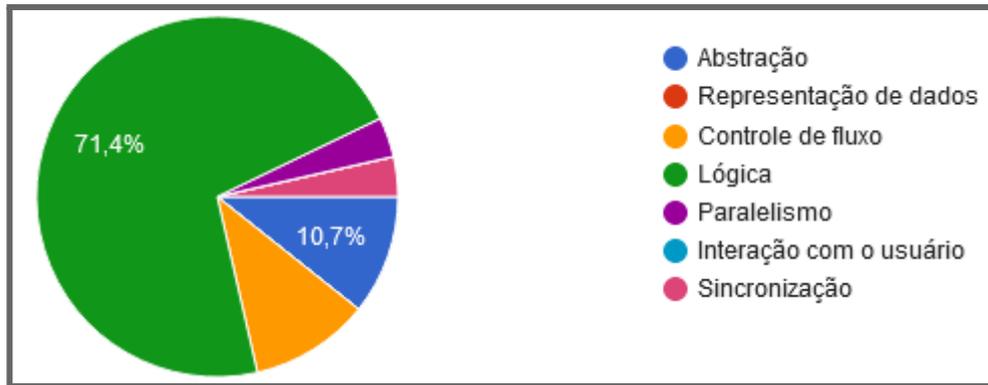


Gráfico 54 - Respostas à pergunta: Na análise dos dados do 10º aluno, qual conceito não foi pontuado em nenhuma atividade?

A partir do gráfico de atividades do décimo primeiro aluno, foram apresentadas várias afirmações. Os participantes precisavam analisar os dados presentes na figura 35 e responder “qual resposta está correta?”. Conforme os resultados obtidos no gráfico 55, houve divergências de opiniões e as porcentagens ficaram bem distribuídas, sendo que todas as alternativas estavam corretas, e a maioria das respostas foi voltada para outras alternativas. Desta forma, possivelmente os participantes se atentaram apenas para a primeira informação que visualizaram, algo que os motivou a selecionar a alternativa correspondente para a primeira informação adquirida.

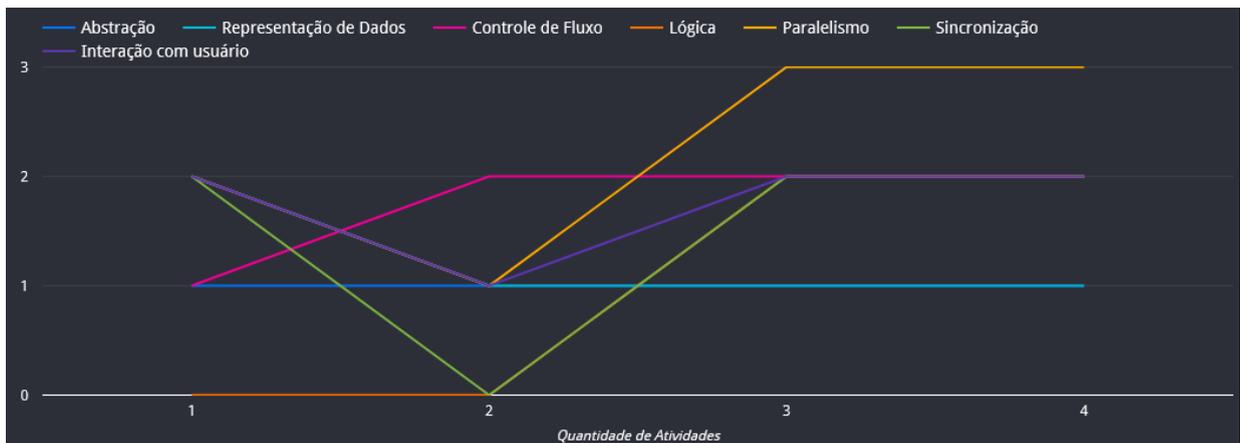


Figura 35 - Visualização referente ao 11º Aluno.



Gráfico 55 - Respostas à pergunta: Na análise dos dados do 11º aluno, qual resposta está correta?

Nos dados das atividades do décimo segundo aluno, é possível observar na figura 36 que todas as atividades pontuaram no conceito de abstração. Trata-se de uma visualização simples com 6 atividades, onde os participantes precisavam analisar e responder a seguinte pergunta: "quantas atividades pontuaram no conceito abstração?". Conforme os resultados mostrados no gráfico 56, a maioria dos participantes conseguiu visualizar a resposta correta.



Figura 36 - Visualização referente ao 12º Aluno.

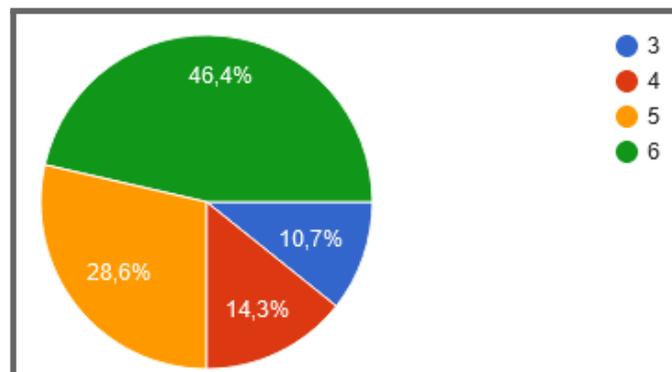


Gráfico 56 - Respostas à pergunta: Na análise dos dados do 12º aluno, quantas atividades pontuaram no conceito abstração?

Analisando o gráfico de atividades do décimo terceiro aluno, quase todos os conceitos pontuaram em todas as atividades, conforme a figura 37, e pode ser verificado no gráfico 57 o resultado, onde os conceitos de abstração, representação de dados, controle de fluxo, paralelismo e interação com o usuário, então com as maiores porcentagens, pois os participantes precisaram analisar os dados e verificar quais conceitos pontuaram em todas as atividades. Com isso obteve-se o resultado esperado, onde os 5 conceitos foram identificados pelos participantes.

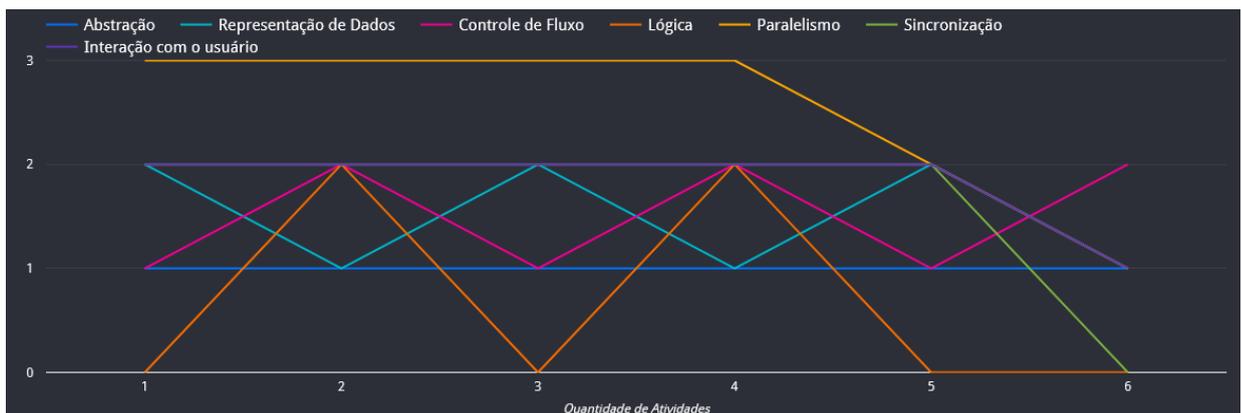


Figura 37 - Visualização referente ao 13º Aluno.

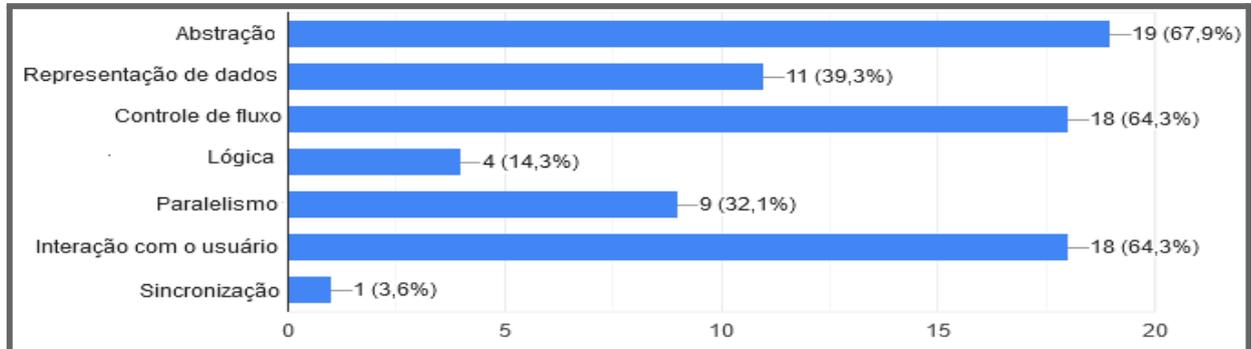


Gráfico 57 - Na análise dos dados do 13º aluno, quais conceitos pontuaram em todas as atividades?

Observando o gráfico de atividades do décimo quarto aluno, os participantes precisaram verificar quantas atividades não obtiveram pontuação zero em nenhum conceito, para então responder a seguinte pergunta “é verdade que em 4 atividades não houve pontuação zero em nenhum conceito?”. O resultado detalhado no gráfico 58 é positivo, pois foram apenas duas atividades que não zeraram tais conceitos, conforme a figura 38.

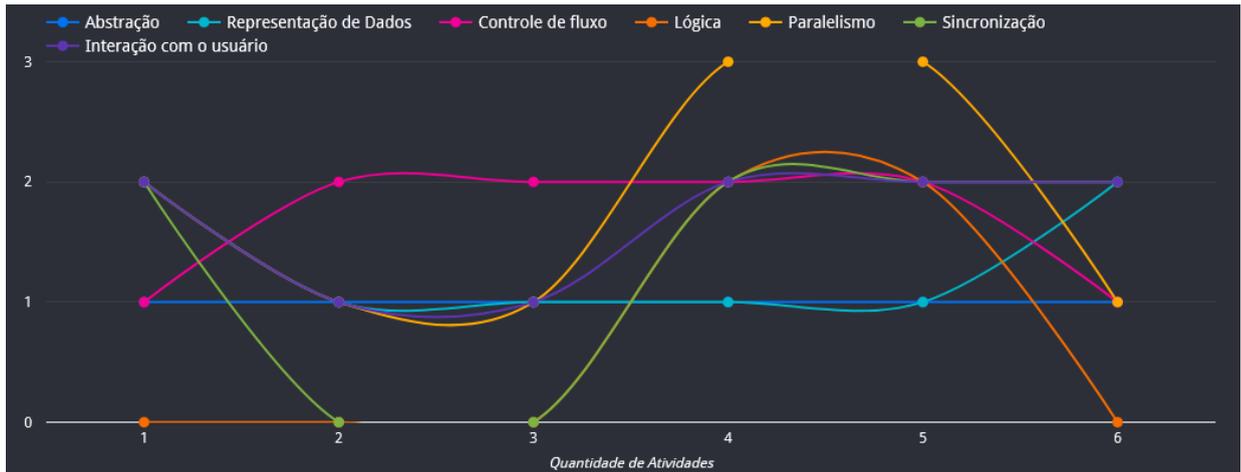


Figura 38 - Visualização referente ao 14º Aluno.

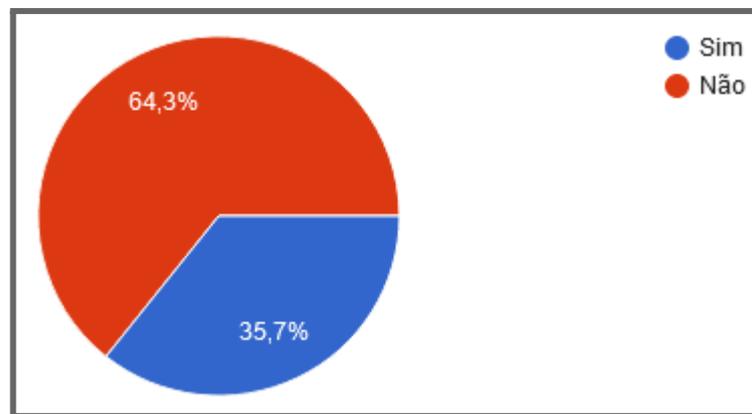


Gráfico 58 - Respostas à pergunta: Na análise dos dados do 14º aluno, é verdade que em 4 atividades não houve pontuação zero em nenhum conceito?

Nos dados das atividades do décimo quinto aluno, pode-se observar na figura 39 que todas as atividades obtiveram pontuações, este é um fator que precisou ser verificado pelos participantes para então responder a seguinte pergunta: “é verdade que pelo menos em uma atividade todos os conceitos tiveram pontuação zero?”. 75% das respostas foram “não”, desta forma considera-se que os dados foram apresentados de forma clara, onde uma grande quantidade de respostas coincide com o esperado, conforme detalhado no gráfico 59.

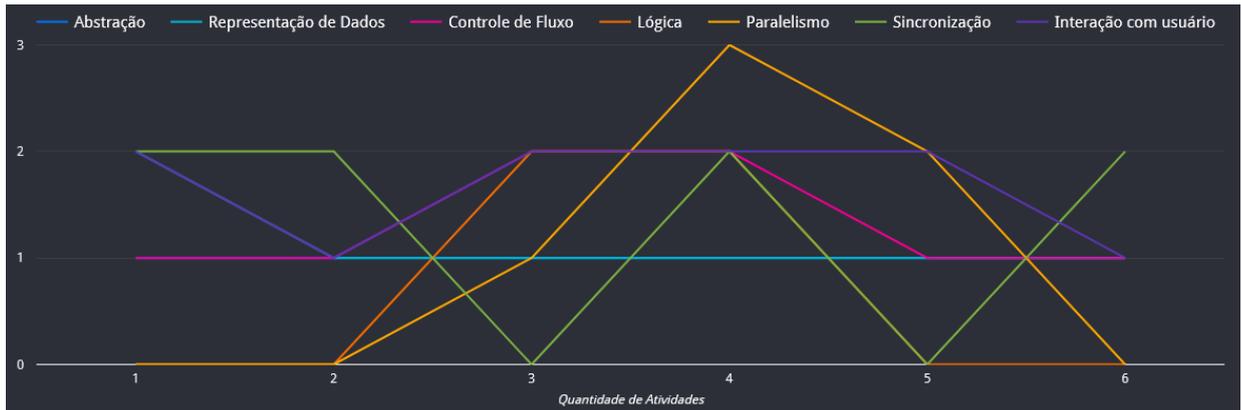


Figura 39 - Visualização referente ao 15º Aluno.

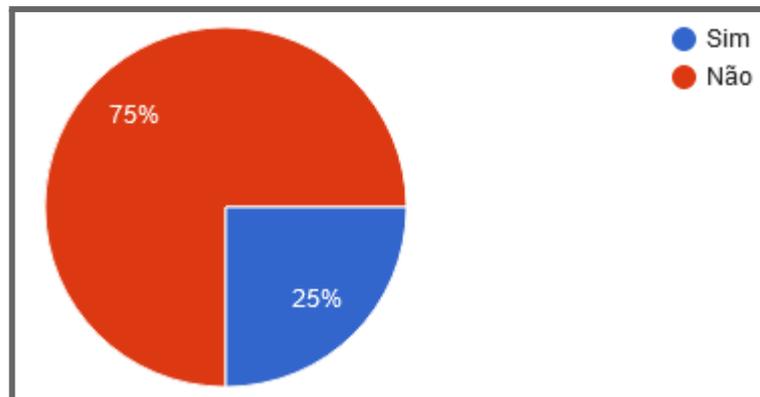


Gráfico 59 - Respostas à pergunta: Na análise dos dados do 15º aluno, é verdade que pelo menos em uma atividade todos os conceitos tiveram pontuação zero?

Analisando o gráfico de atividades do décimo sexto aluno, o conceito de sincronização se manteve com boa pontuação em todas as atividades, não passando por nenhum declínio, onde a pontuação se manteve 2 ou maior, conforme mostrado na figura 40. Levando isso em consideração, os participantes precisaram verificar a trajetória de tal conceito para responder a devida questão: “é verdade que a primeira pontuação para o conceito de sincronização foi na atividade 2 e em seguida não houve nenhum declínio menor do que a pontuação inicial?”. Com o resultado apresentado no gráfico 60, pode-se perceber que as opiniões foram divididas, obtendo 50% para cada resposta. Com isso pode-se entender que não houve uma fácil interpretação dos dados de tal aluno, pois a resposta “sim” deveria ter prevalecido com uma porcentagem maior de respostas.

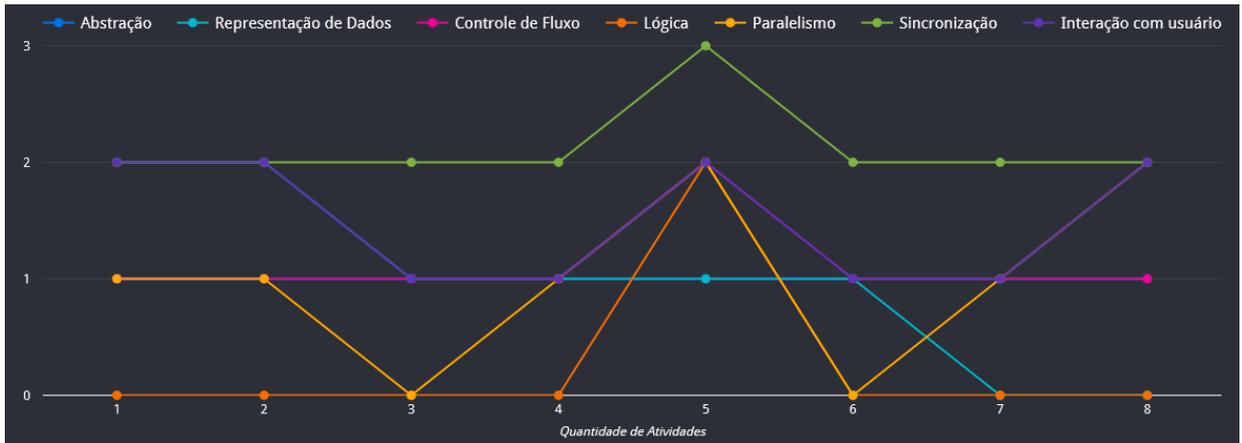


Figura 40 - Visualização referente ao 16º Aluno.

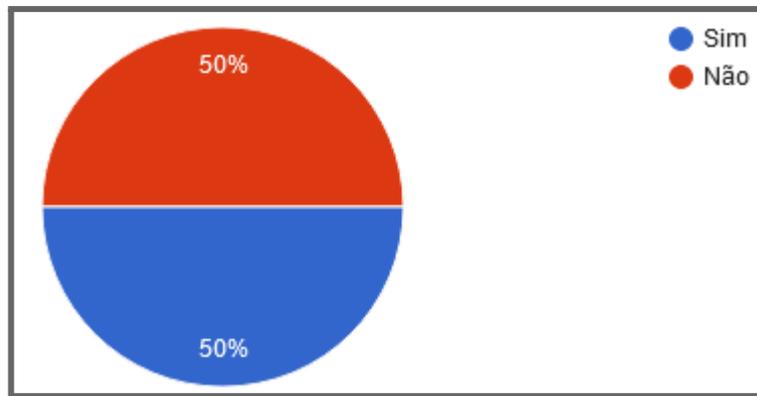


Gráfico 60 - Respostas à pergunta: Na análise dos dados do 16º aluno, é verdade que a primeira pontuação para o conceito de sincronização foi na atividade 2 e não houve declínio menor do que a pontuação inicial?

Foram muitos dados analisados pelos participantes, mas certamente ficou evidente qual conceito não obteve boas pontuação nas atividades de todos os alunos, sendo o conceito de lógica que menos pontuou em todos os gráficos apresentados. 75% dos participantes identificaram corretamente qual conceito foi menos pontuado. O gráfico 61 apresenta o resultado detalhado.

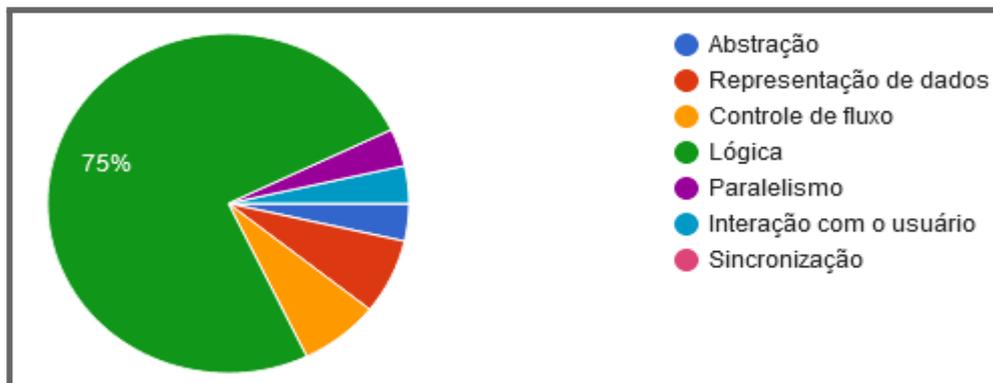


Gráfico 61 - Respostas à pergunta: Na análise das pontuações, qual é o conceito que menos pontuou nas atividades?

### 6.3 Exemplo de Aplicação da Proposta

A professora Franciely irá utilizar uma das visualizações gráficas de um aluno para gerar insight dos resultados obtidos, e a partir disso poderá guiar suas tomadas de decisões a fim de suprir as dificuldades apresentadas pelo aluno.



Gráfico 62 - Visualização gerada com atividades de aluno.

Nesta visualização, o aluno obteve pontuação em 6 dos 7 conceitos, iniciou bem na pontuação de sincronização e parou de pontuar a partir da terceira atividade, já o conceito de paralelismo parou de pontuar a partir da quinta atividade, e o conceito de lógica não obteve pontuação em nenhuma atividade. Com isso, são três conceitos que precisam de atenção, e uma análise das causas que proporcionaram essas baixas pontuações precisa ser realizada.

Em sete atividades o aluno não obteve pontuação máxima em nenhum conceito, sendo esse mais um questionamento a ser verificado, já que as atividades deste aluno necessitavam de vários blocos para solucionar as problemáticas.

### 6.4 Conclusões sobre a análise dos Gráficos

De acordo com os resultados obtidos pelo formulário, a maioria foi positiva, o que indica que os dados coletados de forma automática e apresentados em visualização gráfica podem ajudar no acompanhamento do desenvolvimento dos conceitos nas atividades, possibilitando que o professor escolha qual tipo de visualização pretende obter a partir das atividades já coletadas.

Para apresentar os dados, é importante escolher o melhor estilo de gráfico que facilite a visualização das informações, evitando possíveis interferências nos resultados. Com os resultados adquiridos nesta análise, o estilo de gráfico em linhas com marcação dos pontos,

conforme a figura 41, foi determinado pelos participantes como o melhor para apresentar as visualizações.

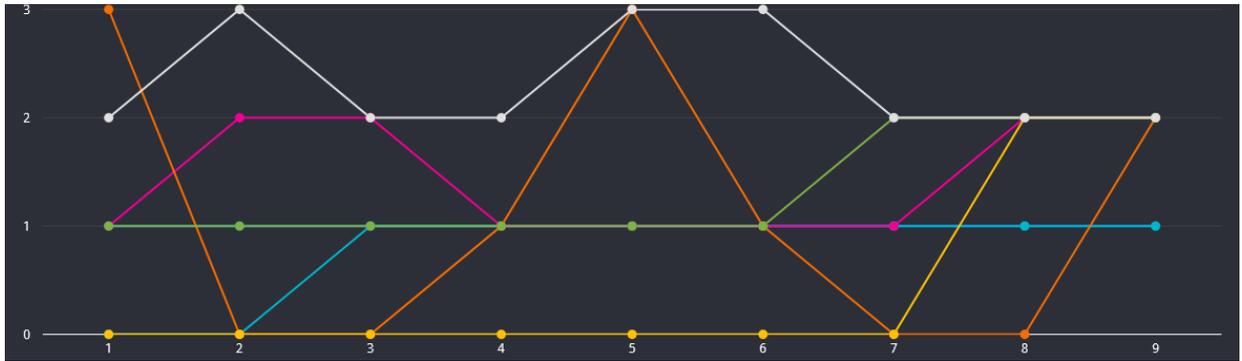


Figura 41 - Determinado como melhor estilo de gráfico.

Alguns dos resultados, que foram negativos, partiram da análise de um gráfico diferente do representado na figura 41. Com isso, pode-se entender que possivelmente os participantes encontraram dificuldades em visualizar informações importantes.

Bastante visualizações foram analisadas, para obter tais resultados a partir do formulário, e todas essas visualizações foram geradas automaticamente por meio do Data Studio. Contudo, tal processo seria impraticável se os dados fossem extraídos de forma manual, pois, de cada aluno diversas atividades foram analisadas, e o processo de coletar as pontuações para cada atividade seria dificultoso e demorado. Além de exigir mais tempo dedicado para gerar os gráficos manualmente, para então poder analisar o avanço das pontuações.

Além de determinar qual estilo de gráfico é mais apropriado para análise, acredita-se que apresentar visualizações com poucos dados proporciona uma melhor visualização, podendo obter melhores resultados. Pois, as visualizações que apresentaram muito dados, e foram avaliadas a partir do formulário, pelos resultados adquiridos, acredita-se que houve dificuldades em identificar os dados claramente. Um exemplo de má visualização está presente na figura 42.

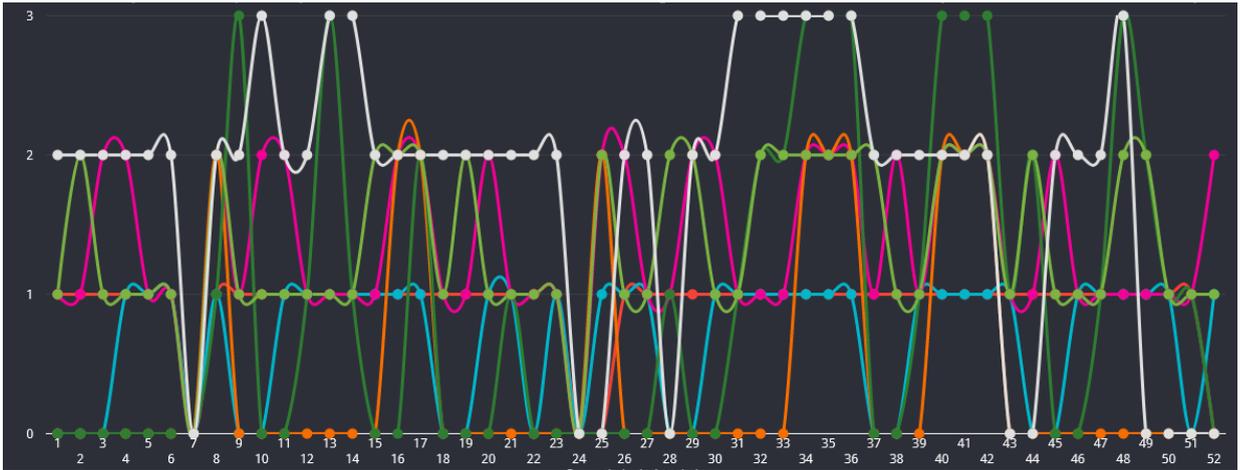


Figura 42 - Visualização com muitas informações.

Então pode-se concluir que não basta definir um bom estilo de gráfico para apresentar os dados, é preciso identificar que uma grande quantidade de informações acaba por atrapalhar a interpretação. Os dados do 2º ano, por exemplo, representam duas turmas, mas seria mais indicado dividir os dados para visualização distintas, o que poderia ajudar os participantes a terem uma melhor visualização dos dados.

## 7 Considerações Finais

A presente pesquisa apresentou uma proposta de procedimento de extração automática de dados utilizando uma versão modificada da ferramenta Dr. Scratch, onde foram avaliadas atividades do Scratch, que geraram visualizações gráficas dos dados, em busca de ajudar os professores na análise do desempenho de alunos em atividades de introdução a programação com o Scratch.

A pesquisa apresenta algumas contribuições voltadas para a análise automática dos dados, como:

- A proposta de visualização apresenta os dados, objetivando permitir as inferências acerca do desempenho dos alunos em geral no período de trimestres diferentes comparando com a análise das turmas, onde pode-se verificar quais conceitos foram mais ou menos pontuados, ao longo das atividades.
- A análise de atividades por aluno proporciona que o professor possa visualizar os dados e analisar quais conceitos precisam ser melhor trabalhados para atingir pontuações positivas, podendo verificar e moldar suas estratégias de ensino para uma melhor compreensão do conteúdo.
- Diferentes estilos de gráficos para a visualização dos dados, que podem ser

alterados facilmente por meio do Data studio, para apresentar os resultados da melhor forma, proporcionando uma fácil visualização dos dados.

Nosso objetivo de pesquisa é propor as visualizações automáticas, onde o professor pode visualizar os dados das atividades de cada aluno, permitindo-lhe analisar o desenvolvimento do aprendizado, pois de acordo com os resultados que são obtidos, se necessário o professor poderá planejar novas metodologias que impactem na melhoria do desempenho dos alunos, conseqüentemente buscando contribuir na perspectiva de atingir os objetivos e metas relacionadas ao aprendizado de introdução à programação com o Scratch.

Os objetivos desta pesquisa foram atingidos, as visualizações foram geradas, analisadas e discutidas, e pode-se concluir que tais visualizações podem ajudar no acompanhamento do avanço do desempenho da turma e dos alunos individualmente, proporcionando dados para que o professor possa produzir orientações que alavanquem os resultados obtidos nas atividades.

A proposta de visualização foi avaliada com a aplicação do questionário online, e os resultados indicaram que os benefícios das visualizações dependem de como os dados são apresentados, buscando sempre auxiliar na visão e performance das informações obtidas nas atividades de cada aluno. Dados apresentados por uma visualização gráfica ruim podem fortemente contribuir para resultados negativos.

## **7.1 Limitações e Trabalhos Futuros**

O presente trabalho teve algumas limitações. Entre elas, pode-se destacar a dificuldade em encontrar pesquisas brasileiras que se assemelham com esse trabalho, a fim de utilizá-las como trabalhos relacionados, tendo como foco avaliar automaticamente atividades desenvolvidas em programações em blocos.

Outra limitação se constitui no pequeno número de alunos que entregaram dados para a análise, pois as aulas ocorreram remotamente e este panorama fez com que poucas atividades fossem entregues.

Como possíveis trabalhos futuros associados a esta pesquisa, pode-se descrever:

- As análises propostas nesta pesquisa poderão servir de motivação para o desenvolvimento de uma ferramenta, com foco em dar suporte à metodologia do professor, contendo uma interface de usuário, podendo ser acessada tanto por professores como por alunos, onde cada professor poderá efetuar seu login e realizar suas análises de atividades, e os alunos ao efetuar login poderão

verificar suas atividades realizadas e as metas cumpridas e quais precisam ser alcançadas ao longo das aulas, desta forma podendo realizar uma autoavaliação do seu aprendizado.

- A realização de uma pesquisa de opinião dos professores de introdução a programação na educação básica, a fim de verificar a contribuição da avaliação automática de atividades e seus resultados de visualização.
- As análises propostas nesta pesquisa poderão servir de motivação para que atividades e materiais didáticos sejam propostos para a melhoria do rendimento dos alunos no aprendizado de programação, focando nos 7 conceitos do PC que foram apresentados.
- A realização de uma análise a fim de verificar como a proposta presente neste trabalho poderá se conectar com os currículos escolares.
- Com os resultados deste trabalho, pretende-se analisar quais conceitos do Pensamento Computacional são apropriados para serem abordados em cada turma, levando em consideração que alguns conceitos podem ser mais difíceis de serem compreendidos por turmas de anos iniciais.

## Referências

- Aivaloglou, F. Hermans, J. Moreno-Leon and G. Robles, "A Dataset of Scratch Programs: Scraped, Shaped and Scored," 2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR), 2017, pp. 511-514, doi: 10.1109/MSR.2017.45.
- Amy K. Hoover, Jackie Barnes, Borna Fatehi, Jesús Moreno-León, Gillian Puttick, Eli Tucker-Raymond, and Casper Hartevelde. 2016. Assessing Computational Thinking in Students' Game Designs. In Proceedings of the 2016. Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts (CHI PLAY Companion '16). ACM, New York, NY, USA, 173–179.
- ARAÚJO, Nicolas de; PRIMO, Tiago Thompsen; PERNAS, Ana Marilza. OntoScratch: ontologias para a avaliação do ensino de Pensamento Computacional através do Scratch. *In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO*, 31. , 2020, Online. **Anais [...]**. Porto Alegre: Sociedade Brasileira de Computação, 2020. p. 1823-1832.
- ARAÚJO, Ana Liz Souto Oliveira; ANDRADE, Wilkerson L.; GUERRERO, Dalton D. Serey. Um Mapeamento Sistemático sobre a Avaliação do Pensamento Computacional no Brasil. V Congresso Brasileiro de Informática na Educação (CBIE 2016). Rio Tinto, PB.
- Baker, R. S., e Inventado, P.S. (2014). Educational data mining and learning analytics. In Learning analytics (pp. 61-75). Springer, New York, NY.
- Blikstein, P., Worsley, M., Piech, C., Sahami, M., Cooper, S., & Koller, D. (2014), programming pluralism: Using learning analytics to detect patterns in the learning of computer programming, *Journal of the Learning Sciences*, 23(4), 561-599.
- BARR, V.; STEPHENSON, C. Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community? *ACM Inroads*, v. 2, n. 1, p. 48, 2011.
- BATES, Tony. Educar na era digital. São Paulo: Artesanato Educacional, 2017. (Coleção tecnologia educacional; 8). Livro Digital.
- BINI, Elena Mariele; KOSCIANSKI, André. O Ensino de Programação de Computadores em um Ambiente Criativo e Motivador. Florianópolis, 2009.
- BINKLEY, D. 2007. Source Code Analysis: A Road Map. In 2007 Future of Software Engineering (FOSE '07). IEEE Computer Society, Washington, DC, USA, 2007.
- BITTENCOURT, Roberto A. et al. Aprendizagem de Programação Através de Ambientes Lúdicos em um Curso de Engenharia de Computação: Uma Primeira Incursão. Feira de Santana – BA, 2013.
- Calegari, P., Santos, T. N., Pozzebon, E. e Frigo, L. B. (2015) Utilizando a robótica para o ensino de lógica computacional com crianças do ensino fundamental. In: *Revista Novas Tecnologias na Educação*, v. 13, n. 2.
- Castro, Adriane de. O uso da programação scratch para o desenvolvimento de habilidades em crianças do ensino fundamental. Dissertação (Mestrado em Ensino de Ciência e Tecnologia) -

Programa de Pós-Graduação em Ensino de Ciência e Tecnologia. Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2017.

CAVALCANTE, Ahemenson Fernandes; COSTA, Leonardo dos Santos; ARAÚJO, Ana Liz Souto Oliveira. Um Estudo de Caso Sobre Competências do Pensamento Computacional Estimuladas na Programação em Blocos no Code.Org. V Congresso Brasileiro de Informática na Educação, Rio Tinto, PB, 2016.

Code. (2015) “Code Club Brasil”, <http://codeclubbrasil.org>.

CHUNG, S. (2006) ‘Digital Storytelling in Integrated Arts Education’, *The International Journal of Arts Education*, 33: 33-50.

CLOW, D. The learning analytics cycle: closing the loop effectively. In *Proceedings of the 2nd international conference on learning analytics and knowledge*. Vancouver - Canadá: ACM.. p. 134-138, 2012.

DAWSON, S. et al. *Current state and future trends: a citation network analysis of the learning analytics field*. 2014, New York, New York, USA: ACM Press, 2014.

DIAS, Robson dos Santos. Caracterização do Learning Analytics na Educação a Distância. *Anais do I Seminário de Pesquisa e Inovação Tecnológica – SEPIT/ IFTM, Uberlândia/MG*, v.1, n. 1, jun., 2017.

DOMINGOS, Adenil Alfeu. *Storytelling: Evolução, Novas Tecnologias e Mídia*. Intercom – Sociedade Brasileira de Estudos Interdisciplinares da Comunicação, Curitiba, set. 2009.

ELOY, Adelmo Antonio da Silva. *Contribuições para aplicação de learning analytics no apoio à avaliação em atividades de introdução à programação com Scratch*. 2019. Dissertação (Mestrado em Sistemas Eletrônicos) - Escola Politécnica, Universidade de São Paulo, São Paulo, 2019.

FARIA, Susana. *Educational Data Mining e Learning Analytics na melhoria do ensino online*.2014.138f. Dissertação de Mestrado em Estatística Computacional - Universidade Aberta, fev, 2014.

FERGUSON, Rebecca et al. *Research evidence on the use of learning analytics: Implications for education policy*. 2016.

FRANÇA, R. S. de; SILVA, W. C. da; AMARAL, H. J. C. (2012) *Ensino de Ciência da Computação na Educação Básica: Experiências, Desafios e Possibilidades*. In: *Anais do XX WEI (Workshop sobre Educação em Computação)* Curitiba.

GATTI, A. Bernardete. *O Professor e a Avaliação em Sala de Aula*. *Revista Brasileira de Docência*. Vol. 1, n.1, p.61-77, Maio de 2009.

GIRAFFA, Lúcia M. M. Recursos de Learning Analytics para compor indicadores auxiliares na avaliação dos estudantes. *Em Rede: Revista de Educação a Distância, UniRede*. v. 2, n. 2, 2015.

HAYDT, Regina Cazaux. *Avaliação do processo ensino-aprendizagem*. São Paulo, Ática, 2004.

HOFFMANN, Jussara. *Avaliação mediadora: uma pratica em construção da pré-escola à*

universidade. 32<sup>a</sup> ed. Porto Alegre, Mediação, 2012.

JOVANOVIC, J.; Gasevic, D.; Brooks, C.; Devedzic, V.; Hatala, M.; Eap, T.; Richards, G. (2008) “LOCO-Analyst: semantic web technologies in learning content usage analysis” *Int. J. Cont. Engineering Education and Lifelong Learning*, Vol. 18, Nº 1. Pag. 54 – 76.

KAFAI, Y. B. BURKE, Q. (2013). Computer Programming Goes Back to School. In: *Education Week*, set.

KELLEHER, C. and Pausch, R. (2005). Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys (CSUR)*, 37(2):83–137. Publisher: ACM New York, NY, USA.

LAK — 1st International Conference on Learning Analytics and Knowledge. Banff, Alberta, 2011.

LUCKESI, C. C. *Avaliação da aprendizagem escolar: estudos e proposições*. [s.l.]: Cortez editora, 2014.

LUCKESI, Cipriano Carlos. *Avaliação da aprendizagem escolar: Estudos e Preposições*. 22<sup>a</sup> ed. São Paulo, Cortez, 2005.

MALONEY, J. et al. (2010) The scratch programming language and environment. In: *Communications of The ACM*, 2010. v. 10, n. 4, Article 16.

MANYIKA, J., Chui, M., Madgavkar, A., & Lund, S. (2017). *Technology, jobs, and the future of work*. McKinsey Global Institute.

MARCON JÚNIOR, Rogério Paulo; BONIATI, Bruno Batista. *LogicBlocks: Uma Ferramenta para o Ensino de Lógica de Programação*. Rio Grande do Sul, 2015.

MARJI, Majed. *Aprenda a programar com Scratch: Uma introdução visual à programação com jogos, arte, ciência e matemática*. São Paulo: Novatec, 2014. Tradução: Lúcia Kinoshita.

MORAN, José. Metodologias ativas para uma aprendizagem mais profunda. In: BACICH, Lilian; MORAN, José. *Metodologias ativas para uma educação inovadora: uma abordagem teórico-prática*. Porto Alegre: Penso, 2018. Cap. 1. p. 1- 25. (Desafios da educação).

MORENO-LEÓN, J., Robles, G. e Román-González, M. (2015). Dr. Scratch: Automatic Analysis of Scratch Projects to Assess and Foster Computational Thinking. *RED-Revista de Educación a Distancia*.

MORENO-LEON, Jesus; ROBLES, Gregorio; ROMAN-GONZALEZ, Marcos. Comparing computational thinking development assessment scores with software complexity metrics. *IEEE Global Engineering Education Conference, EDUCON*, v. 10-13-April-2016, n. April, p. 1040–1045, 2016.

MORETTI, Vinícius Fernandes. *O pensamento computacional no ensino básico: potencialidades de desenvolvimento com o uso do Scratch*. 2019. Trabalho de conclusão de curso (Graduação) - Universidade Federal do Rio Grande do Sul.

MARTINS, José do Prado. *Didática Geral: fundamentos, planejamento, metodologia, avaliação*. 2<sup>a</sup> ed. São Paulo, Atlas, 1993.

NASCIMENTO, R. P. do. Preparando professores para promover a inclusão de alunos com necessidades educacionais especiais. Londrina: PDE/UEL, 2009.

Niemelä, Pia. (2017). All rosy in scratch lessons: No bugs but guts with visual programming. 1-9. 10.1109/FIE.2017.8190612.

Oliveira, A. M. de. (2021). Ensino de programação para crianças e o desenvolvimento de pensamento computacional: Algumas reflexões. Colloquium Humanarum. ISSN: 1809-8207, 18(1). 100-13.

PALACIOS, Fernando; TERENCE, Martha. O Guia Completo do Storytelling. Alta Books Editora, 2016.

PELLE, Rafael. Desenvolvimento de um analisador de código para snap! voltado ao ensino de computação na educação básica. Universidade Federal de Santa Catarina, Departamento de Informática e Estatística, FLORIANÓPOLIS, 2018.

PERRENOUD, Philippe. *Avaliação: excelência à regulação das aprendizagens – Entre duas lógicas*. Artmed, Porto Alegre 1999.

Rodrigues, B. (2014) “Por que ensinar programação nas escolas?”, [https://medium.com/@brunorodrigues\\_270/por-que-ensinar-programacao-nas-escolas-3238def95e7](https://medium.com/@brunorodrigues_270/por-que-ensinar-programacao-nas-escolas-3238def95e7).

SCRATCH. About Scratch (Scratch Documentation Site, 2015).

SERRANO, F. Geração Geek. Revista Exame Informática. São Paulo: Editora Abril, 2014.

Siemens, G., Long, P.. Penetrating the Fog: Analytics in Learning and Education. EDUCAUSE review, vol. 46, n. 5, p. 31-40, 2011.

SILVA, Leandro Augusto et al. Ciência de Dados Educacionais: definições e convergências entre as áreas de pesquisa. In: Anais dos Workshops do Congresso Brasileiro de Informática na Educação. p. 764, 2017.

SILVA, Leonardo Cintra Lopes. A relação do Pensamento Computacional com o Ensino de Matemática na Educação Básica. Orientador: Prof. Dr. Ronaldo Celso Messias Correia. 2019. 131 f. Dissertação (Mestrado) - Universidade Estadual Paulista Júlio de Mesquita Filho, Presidente Prudente – São Paulo, SP – Brasil, 2019.

SILVA, M. D. M. da; CRUZ, R. S. da S. da; OLIVEIRA, A. DO C. B. A avaliação escolar como ferramenta de exclusão. Vivências Educacionais, v. 1, n. 1, p. 25–34, 2016.

SICA, C. (2011). Ciência da Computação no Ensino Básico e Médio. Disponível em <<http://www.odiarior.com/blogs/carlossica/2011/10/07/ciencia-da-computacao-no-ensino-medio/>>

SBC. Referenciais de formação em Computação: educação básica. Porto Alegre, 2017. Documento aprovado pela Comissão de Educação e apresentado no CSBC 2017 durante as Assembleias do WEI e da SBC.

THE ROYAL SOCIETY. Shut down or restart?: the way forward for computing in UK schools. London: The Royal Society, 2012.

VALENTE, José Armando 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 [en linea]. 2016,

WEINTROP, D., Shepherd, D. C., Francis, P., and Franklin, D. (2017). Blockly goes to work: Block-based programming for industrial robots. In 2017 IEEE Blocks and Beyond Workshop (B B), pages 29–36.

WING, J. Computational thinking. Communications of the ACM, v. 49, n. 3, p. 33-35, 2006.

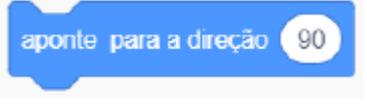
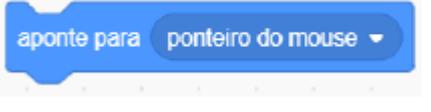
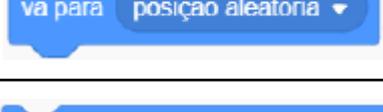
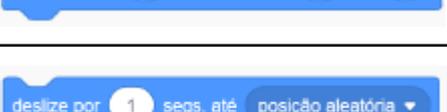
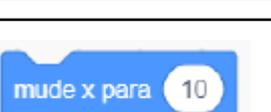
WING, J. PENSAMENTO COMPUTACIONAL: Um conjunto de atitudes e habilidades que todos, não só cientistas da computação, ficaram ansiosos para aprender e usar. Revista Brasileira de Ensino de Ciência e Tecnologia, v. 9, n. 2, 2016.

WING, J. M. Computational thinking: what and why? 17 Nov. 2010. Disponível em: <http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf>.

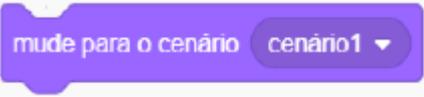
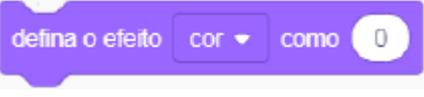
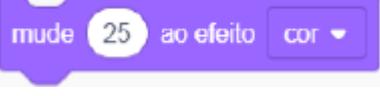
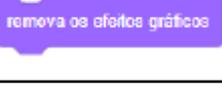
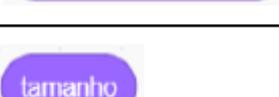
ZANETTI, Humberto Augusto Piovesana; BORGES, Marcos Augusto Francisco; RICARTE, Ivan Luiz Marques. Pensamento Computacional no Ensino de Programação: uma revisão sistemática da literatura. In: Congresso Brasileiro de Informática na Educação, 5, 2016, Uberlândia, Minas Gerais.

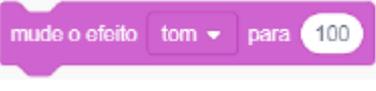
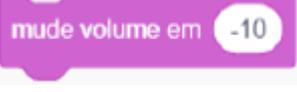
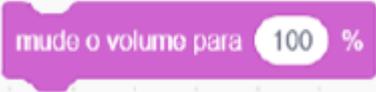
## ANEXOS

## Anexo 1 - Blocos do Scratch e suas funções.

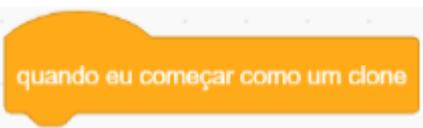
BLOCOS DO SCRATCH	
Blocos de Movimento	Função
	Move o ator um determinado número de passos.
	Gira para a direita.
	Gira para a esquerda.
	Estabelece a direção para qual o ator irá se direcionar.
	Aponta o ator na direção do mouse.
	Vá para a posição no palco nas posições: x e y.
	Vai para a localização do ponteiro do mouse ou aleatória.
	Desliza ao longo do tempo para a posição (x; y).
	Desliza ao longo do tempo para a posição aleatória ou ponteiro do mouse.
	Adiciona esta quantidade a coordenada x.
	Define a posição x de um ator.

	Adiciona esta quantidade a coordenada y.
	Define a posição y de um ator.
	Se tocar na borda de cenário, volta.
	Informa a posição x de um ator.
	Informa a posição y de um ator.
	Informa a direção atual de um ator.
<b>Blocos de Aparência</b>	<b>Função</b>
	Mostra o texto em um balão de fala durante um número de segundos.
	Mostra um texto em um balão de fala.
	Exibe palavras em um balão de pensamento por um número de segundos.
	Mostra palavras em um balão de pensamento.
	Faz o ator aparecer no cenário.
	Faz o ator desaparecer do cenário.
	Muda para uma fantasia específica do ator.

	Muda para a próxima fantasia da lista de fantasias do ator.
	Muda para um cenário específico.
	Muda o efeito selecionado para o valor especificado.
	Adiciona o valor especificado ao efeito selecionado.
	Remove todos os efeitos do ator.
	Adiciona o valor especificado ao tamanho do ator.
	Muda o tamanho do ator para a porcentagem selecionada.
	Coloca um ator na frente ou atrás de todos os outros.
	Trás o ator para frente ou para trás do número especificado de camadas.
	O número ou nome da fantasia.
	O número ou nome do cenário.
	O tamanho do ator.
<b>Blocos de Som</b>	<b>Função</b>
	Toca o som.

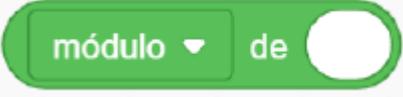
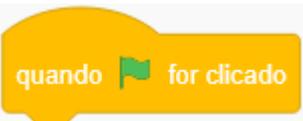
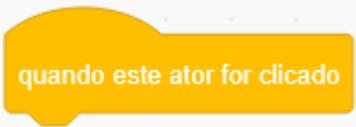
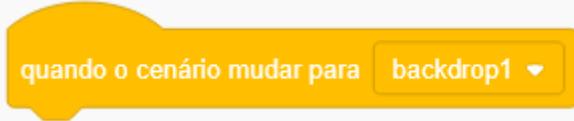
	Reproduz um som e espera até que o som termine.
	Para de tocar todos os sons.
	Muda o tom de um som para grave ou agudo.
	Altera o volume do som do ator em uma quantidade específica.
	Altera o volume do som do ator em uma porcentagem específica.
	Informa o volume do som do ator.
<b>Blocos de Variáveis e Lista</b>	<b>Função</b>
	Informa o valor da variável.
	Atribui a variável o valor específico.
	Altera o valor da variável em um valor especificado.
	Mostra as variáveis criadas no cenário.
	Ocultas as variáveis criadas no cenário.
	Informa todos os itens da lista.
	Adiciona um item especificado ao final da lista.

	Exclui um ou todos os itens da lista.
	Insere um novo item na posição especificada da lista.
	Substitui o valor de um item da lista pelo valor especificado.
	Coloca o item na posição especificada da lista.
	Informa quantos itens há na lista.
	Tem o valor verdadeiro se a lista tiver um valor específico.
	Mostra a lista no cenário.
	Ocultas as listas criadas do cenário.
<b>Blocos de Controle</b>	<b>Função</b>
	Espera um número específico de segundos, continuando depois o comando do próximo bloco.
	Executa os blocos envolvidos um determinado número de vezes.
	Executa repetidamente e para sempre os comandos do seu interior.

	<p>Se a condição for verdadeira, executa os blocos envolvidos.</p>
	<p>Se a condição for verdadeira, executa os comandos envolvidos pelo bloco (se); se não for, executa os comandos envolvidos no bloco (senão).</p>
	<p>Espera até que a condição seja verdadeira, depois executa o restante dos comandos.</p>
	<p>Repete os blocos que se seguem até que a condição seja verdadeira.</p>
	<p>Para todos os scripts em todos os atores ou apenas um script ou outros scripts do ator.</p>
	<p>Programa comportamento de um clone.</p>
	<p>Cria um clone de ator especificado.</p>
	<p>Exclui um clone atual de um ator.</p>
<p style="text-align: center;"><b>Blocos de Sensores</b></p>	<p style="text-align: center;"><b>Função</b></p>
	<p>O valor é verdadeiro se o ator estiver tocando em outro ator já especificado, na borda ou no cursor do mouse.</p>

	O valor é verdadeiro se o ator estiver tocando na cor definida.
	Tem o valor verdadeiro se a primeira cor estiver tocando a segunda cor.
	Informa a distância até o ator especificado ou até o cursor do mouse.
	Faz uma pergunta na tela e armazena a entrada do teclado em resposta.
	A entrada de resposta mais recente.
	Tem valor verdadeiro se a tecla especificada estiver sendo pressionada.
	Tem o valor verdadeiro se o botão de mouse estiver pressionado.
	Tem o valor da coordenada x da posição de ponteiro do mouse.
	Tem o valor da coordenada y da posição de ponteiro do mouse.
	Informa a intensidade (entre 1 e 100) dos sons captados pelo microfone do computador.
	Define o ator como arrastável ou não arrastável.
	Informa o valor do cronômetro em segundos.
	Zera o cronômetro.
	A posição X, posição Y, direção, traje, tamanho ou volume do palco ou um ator.

	Tem o valor do ano, mês, data, dia da semana, hora, minuto, segundo atual.
	Informa o nome do usuário que está usando o projeto.
<b>Blocos de Operadores</b>	<b>Função</b>
	Calcula adição.
	Calcula subtração.
	Calcula multiplicação
	Calcula divisão
	Escolhe um valor inteiro aleatório em um intervalo determinado.
	Tem o valor de verdadeiro se o primeiro valor for menor que o segundo
	Mostrar verdadeiro se os dois valores forem iguais.
	Tem o valor verdadeiro se o primeiro valor for maior que o segundo
	Tem o valor verdadeiro se ambas as condições forem verdadeiras
	Tem o valor verdadeiro se alguma das condições for verdadeira.

	Tem o valor verdadeiro se a condição for falsa e tem o valor falso se a condição for verdadeira.
	Concatena, junta cadeias de caracteres.
	Apresenta o caractere na posição especificada da cadeia de caracteres.
	Informa o número de caracteres em uma cadeia de caracteres.
	Calcula o valor do resto da divisão do primeiro número pelo segundo número.
	Calcula o valor do número inteiro mais próximo do número dado.
	Calcula uma função.
<b>Blocos de Eventos</b>	<b>Função</b>
	Bloco de início, a programação é executada ao clicar na bandeira verde.
	Bloco de início, a programação é executada ao clicar na tecla do teclado que foi especificada.
	Bloco de início, a programação é executada ao clicar no ator que possuir essa programação.
	Bloco de início, a programação é executada quando o cenário mudar.

A yellow Scratch block with a semi-circular top-left corner. It contains the text 'quando' followed by a dropdown menu showing 'ruído', a greater-than sign '>', and a white circle containing the number '10'.	Bloco de início, a programação é executada quando o número do ruído ou cronômetro for maior que o número especificado.
A yellow Scratch block with a semi-circular top-left corner. It contains the text 'quando eu receber' followed by a dropdown menu showing 'mensagem 1'.	Bloco de início, a programação é executada quando o personagem recebe uma mensagem especificada.
A yellow Scratch block with a semi-circular top-left corner. It contains the text 'transmita' followed by a dropdown menu showing 'mensagem 1'.	Transmite uma mensagem entre personagens.
A yellow Scratch block with a semi-circular top-left corner. It contains the text 'transmita' followed by a dropdown menu showing 'mensagem 1' and the text 'e espere'.	Transmite uma mensagem entre personagens e espera para dar continuidade na execução.

## Anexo 2 - Termo de Autorização para Pesquisa.

**TERMO DE AUTORIZAÇÃO PARA PESQUISA**

Prezado-(a) Senhor-(a)

Eu, Franciely Alves de Souza, responsável pelo projeto de pesquisa Aplicação de Learning Analytics para Avaliação do Aprendizado de Programação em Blocos do Mestrado em Informática Aplicada (PPGIA-UFRPE), sob orientação do professor Rafael Ferreira e co-orientação da professora Taciana Pontual Falcão, venho pelo presente, solicitar vossa autorização para utilizar dados educacionais coletados por meio da ferramenta de programação em blocos Scratch, durante o ano de 2020 no Colégio [REDACTED]

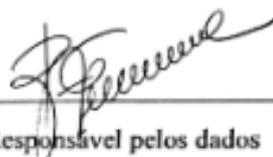
Esta pesquisa tem como objetivo identificar e analisar a progressão do aprendizado de programação no ensino fundamental. O procedimento adotado será a análise automática de atividades de programação realizadas na ferramenta Scratch, utilizada por alunos do 2º ao 6º ano, no período de maio a dezembro de 2020.

Espera-se com esta pesquisa, contribuir para um maior conhecimento sobre as práticas de avaliação automática de atividades no Scratch. Os dados obtidos nesta pesquisa serão utilizados de forma anônima na publicação de artigos científicos, garantindo-se o sigilo, em relação ao nome da escola, nomes de alunos, endereço e quaisquer outras informações de identificação.

Desta forma, uma vez tendo lido e entendido tais esclarecimentos, caso esteja de acordo solicitamos a sua concordância e assinatura:

Eu, [REDACTED], autorizo a pesquisadora Franciely Alves de Souza a usar os dados educacionais coletados por meio da ferramenta de programação em blocos Scratch no Colégio [REDACTED], de maio a dezembro de 2020.

Recife, 31 de maio de 2021.



Responsável pelos dados

Pesquisadora Responsável

**DIRETORA**  
Autorização nº 019/21-GRE Norte  
Cadastro Escolar P.000.455

### Anexo 3 - Questionário para Compreensão dos Gráficos.

## TESTE DE COMPREENSÃO DE GRÁFICOS

Olá, sou Franciely Alves, aluna do Mestrado em Informática Aplicada (PPGIA-UFRPE), estou realizando uma pesquisa sobre os métodos de avaliação de atividades desenvolvidas na ferramenta de programação em blocos Scratch. O objetivo desse formulário é analisar dados apresentados em gráficos que foram gerados a partir de avaliações automáticas, fornecendo uma análise quantitativa e categorizada em sete conceitos: abstração, paralelismo, raciocínio lógico, sincronização, controle de fluxo, interatividade com o usuário e representação de dados. Agradeço desde já ao responder este formulário.

---

\*Obrigatório

1. \*

*Marque todas que se aplicam.*

Concordo em participar voluntariamente da pesquisa.

### IDENTIFICAÇÃO

2. Nome: \*

---

3. Informe seu e-mail: \*

---

## 4. 1 - Qual é o seu curso de graduação? \*

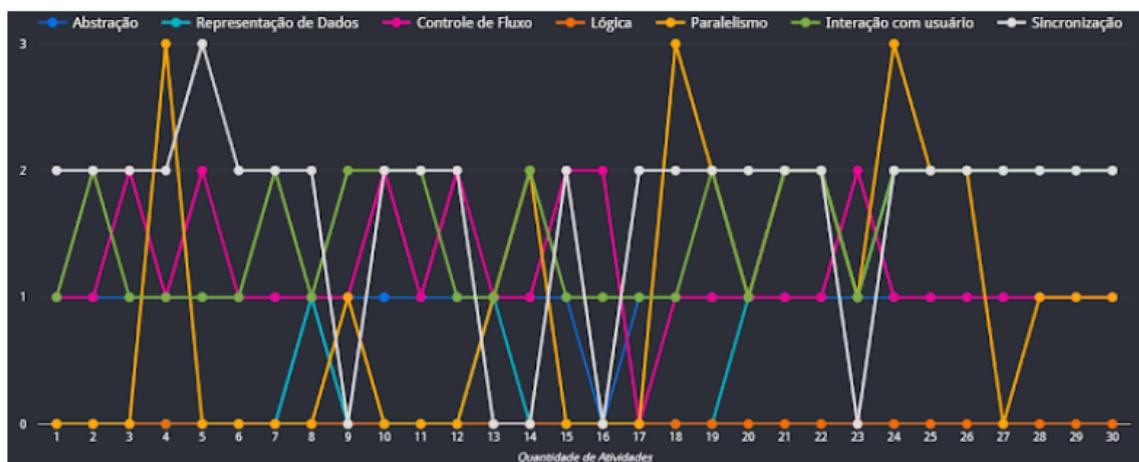
Marcar apenas uma oval.

- Licenciatura em Computação
- Ciência da Computação
- Sistema de Informação
- Outro: \_\_\_\_\_

### ANÁLISE DE ATIVIDADES POR TRIMESTRE

Os alunos fizeram uso da linguagem de programação em blocos durante o período de 9 meses. Sabendo disso, antes das atividades serem avaliadas automaticamente, foram agrupadas por trimestres, sendo: 3 trimestre. A partir dessa informação, podemos observar o gráfico de cada trimestre.

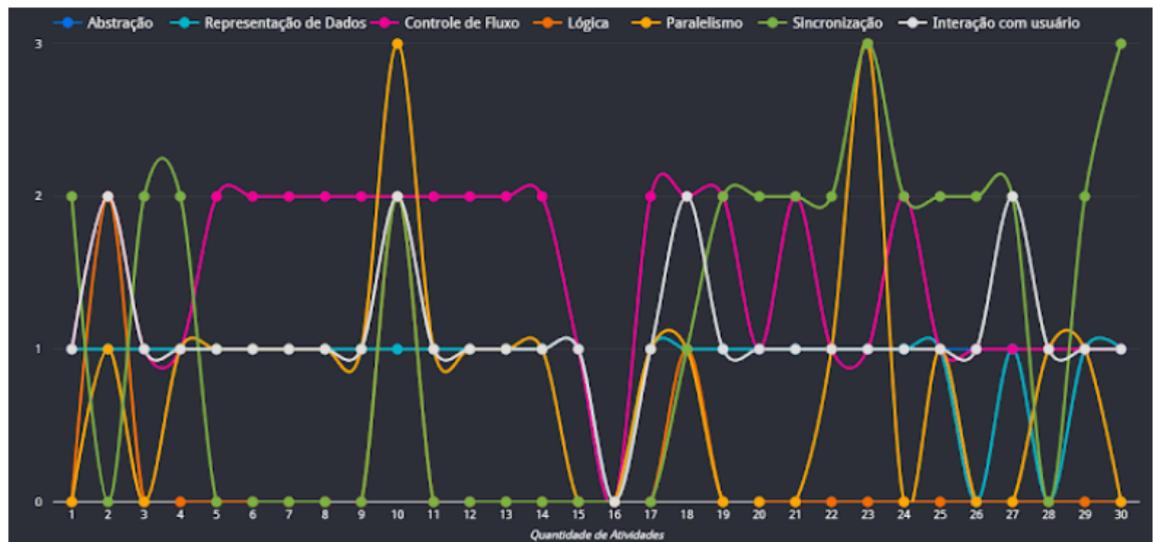
## 5. 1) Observando o gráfico do primeiro trimestre, em qual dos conceitos os alunos tiveram baixa pontuação? \*



Marcar apenas uma oval.

- Abstração - (Azul marinho)
- Representação de dados - (Azul celeste)
- Controle de fluxo - (Rosa)
- Lógica - (Laranja)
- Paralelismo - (Amarelo)
- Interação com o usuário - (Verde)
- Sincronização - (Branco)

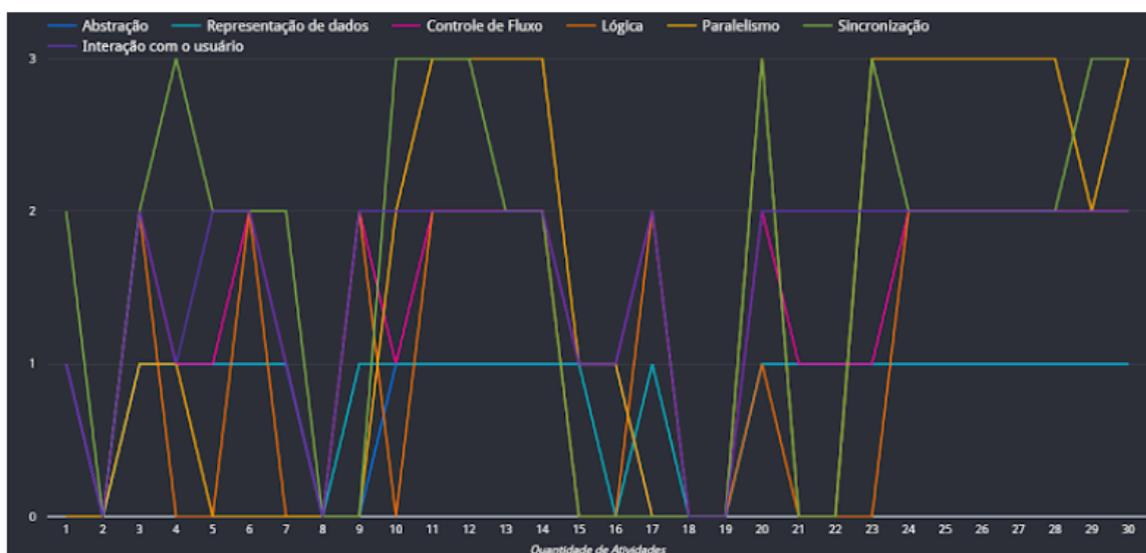
6. 2) Observando o gráfico, em qual dos conceitos a maioria dos alunos mantiveram um bom desempenho durante todo o segundo trimestre? \*



Marcar apenas uma oval.

- Abstração - (Azul marinho)
- Representação de dados - (Azul celeste)
- Controle de fluxo - (Rosa)
- Lógica - (Laranja)
- Paralelismo - (Amarelo)
- Interação com o usuário - (Branco)
- Sincronização - (Verde)

7. 3) Observando o gráfico do terceiro trimestre, quais dos conceitos os alunos tiveram a melhores pontuações? \*



Marque todas que se aplicam.

- Abstração - (Azul marinho)
- Representação de dados - (Azul celeste)
- Controle de fluxo - (Rosa)
- Lógica - (Laranja)
- Paralelismo - (Amarelo)
- Interação com o usuário - (Roxo)
- Sincronização - (Verde)

8. Qual estilo de gráfico ficou mais fácil de visualizar? \*

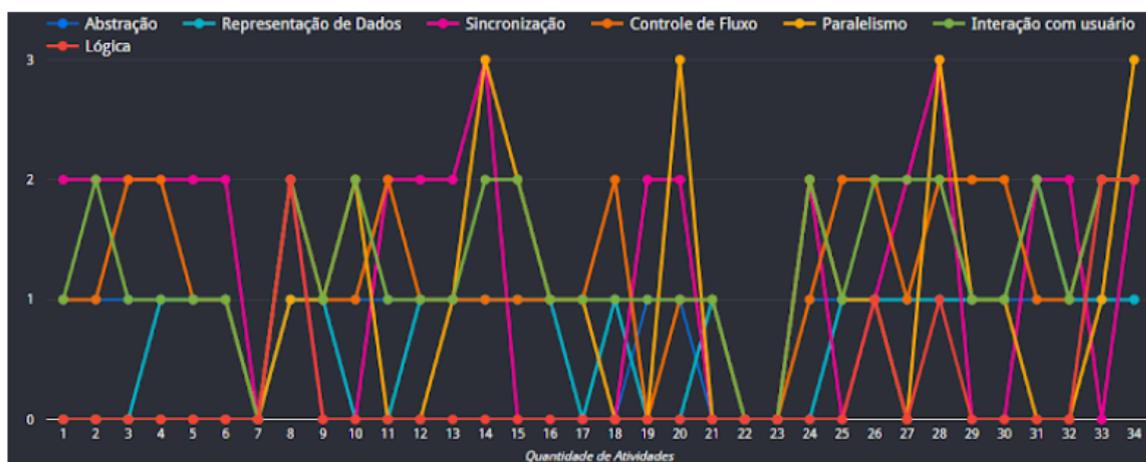
Marcar apenas uma oval.

- Primeiro trimestre
- Segundo Trimestre
- Terceiro Trimestre

ANÁLISE DE  
ATIVIDADES POR  
FALTAS

As aulas ocorriam 1 vez por semana, e as atividades analisadas foram de alunos que faltaram no mínimo 4 vezes.

9. 1) Analisando o gráfico de dados dos alunos que tiveram os maiores números de faltas, qual dos conceitos se manteve com uma pontuação constante na maioria das atividades, desde do início? \*



Marcar apenas uma oval.

- Abstração - (Azul marinho)  
 Representação de dados - (Azul celeste)  
 Controle de fluxo - (Laranja)  
 Lógica - (Vermelho)  
 Paralelismo - (Amarelo)  
 Interação com o usuário - (Verde)  
 Sincronização - (Rosa)
10. 2) Analisando o gráfico, é verdade que em apenas 5 atividades a pontuação de lógica não é 0? \*

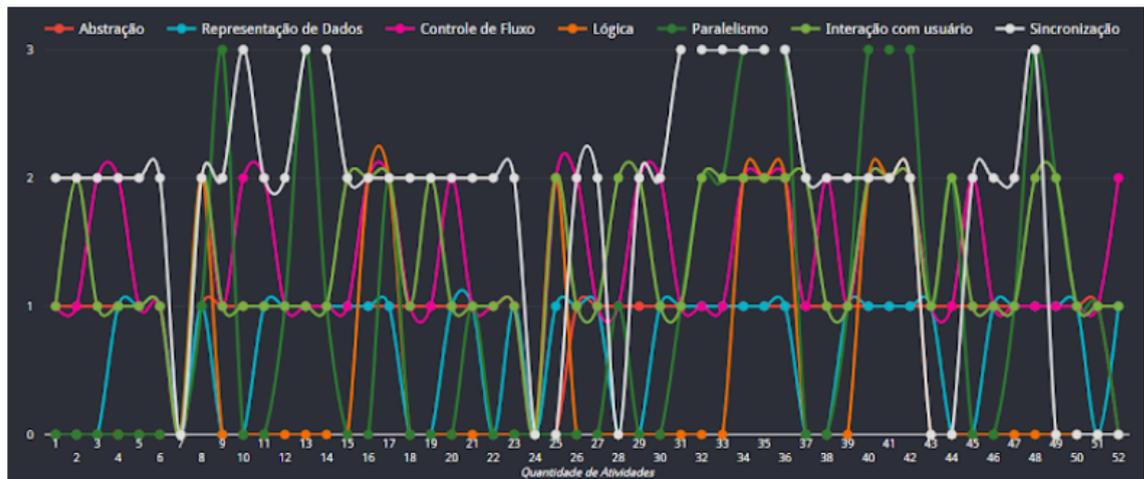
Marcar apenas uma oval.

- Sim  
 Não

ANÁLISE DE ATIVIDADES POR ANO

As aulas ocorriam no ensino fundamental, sendo: 2º, 3º, 4º e 5º ano.

11. 1) Analisando o gráfico do 2º ano, quais conceitos tiveram pontuação 2? \*



Marque todas que se aplicam.

- Abstração - (Vermelho)  
 Representação de dados - (Azul celeste)  
 Controle de fluxo - (Rosa)  
 Lógica - (Laranja)  
 Paralelismo - (Verde em tom escuro)  
 Interação com o usuário - (Verde em tom claro)  
 Sincronização - (Branco)
12. Observando o gráfico do 2º ano, é verdade que 9 atividades não pontuaram no conceito sincronização? \*

Marcar apenas uma oval.

- Sim  
 Não
13. Observando o gráfico do 2º ano, é verdade que nenhuma atividade pontou mais que 1, no conceito representação de dados? \*

Marcar apenas uma oval.

- Sim  
 Não

14. 2) Observando o gráfico das atividades do 3º ano, quais conceitos não pontuaram em apenas 1 atividade? \*



Marque todas que se aplicam.

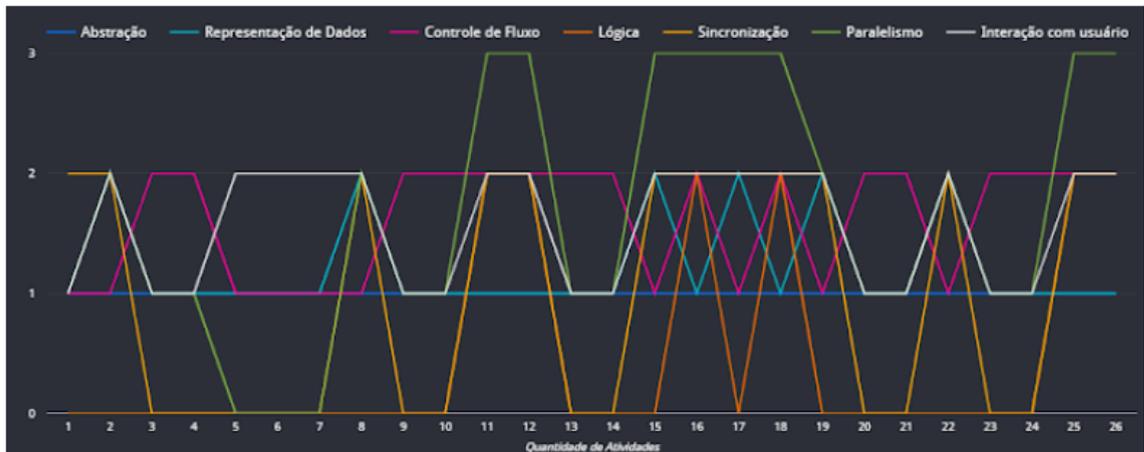
- Abstração - (Azul marinho)
- Representação de dados - (Azul celeste)
- Controle de fluxo - (Branco)
- Lógica - (Rosa)
- Paralelismo - (Verde)
- Interação com o usuário - (Amarelo)
- Sincronização - (Laranja)

15. Observando o gráfico das atividades do 3º ano, é verdade que em apenas 1 atividade houve pontuação em lógica? \*

Marcar apenas uma oval.

- Sim
- Não

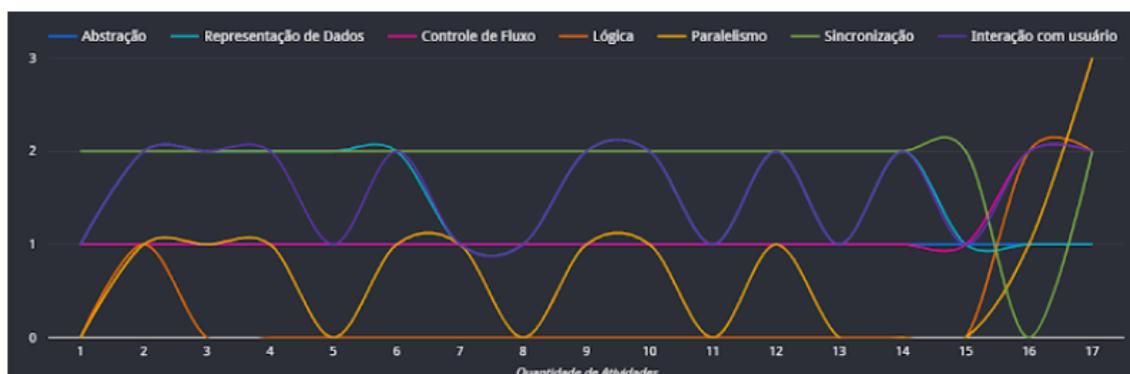
16. 3) Analisando o gráfico das atividades do 4º ano, quais dos conceitos pontuou em um número maior de atividades? \*



Marque todas que se aplicam.

- Abstração - (Azul marinho)
- Representação de dados - (Azul celeste)
- Controle de fluxo - (Rosa)
- Lógica - (Laranja)
- Paralelismo - (Verde)
- Interação com o usuário - (Branco)
- Sincronização - (Amarelo)

17. 4) Analisando o gráfico das atividades do 5º ano, qual conceito não houve pontuação em 6 atividades? \*



Marcar apenas uma oval.

- Abstração - (Azul marinho)
- Representação de dados - (Azul celeste)
- Controle de fluxo - (Rosa)
- Lógica - (Laranja)
- Paralelismo - (Amarelo)
- Interação com o usuário - (Roxo)
- Sincronização - (Verde)

18. Analisando o gráfico do 5º ano, é verdade que apenas uma atividade não pontou no conceito sincronização? \*

*Marcar apenas uma oval.*

- Sim
- Não

19. Qual dos 4 estilos de gráficos ficou mais fácil de visualizar? \*

*Marcar apenas uma oval.*

- 2º Ano
- 3º Ano
- 4º Ano
- 5º Ano

20. Qual dos anos, teve o maior número de atividades avaliadas? \*

*Marcar apenas uma oval.*

- 2º Ano
- 3º Ano
- 4º Ano
- 5º Ano

## ANÁLISE DE ATIVIDADES POR ALUNO(A)

Atividades de 16 alunos(as) foram avaliadas de forma automática. Para as atividades passarem por esse processo de avaliação, cada aluno(a) entregou no mínimo 4 atividades.

21. 1) Analisando o gráfico de atividades do primeiro aluno(a), qual pontuação está sendo representada na atividade 7? \*



Marcar apenas uma oval.

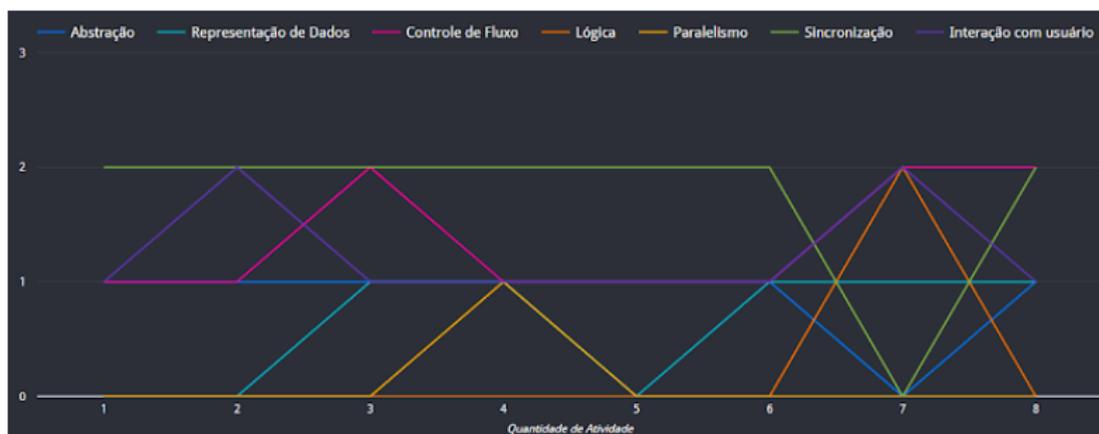
- Pontuação zero para: Interação com o usuário, sincronização e lógica.  
 Pontuação zero para: Interação com o usuário, sincronização, paralelismo e lógica.  
 Pontuação zero para todos os conceitos.  
 Todas as respostas estão erradas.
22. 2) Analisando o gráfico de atividades do segundo aluno(a), é verdade que em pelo menos 1 atividade o conceito de controle de fluxo foi zero? \*



Marcar apenas uma oval.

- Sim  
 Não

23. 3) Analisando o gráfico de atividades do terceiro aluno(a), quais conceitos tiveram a pontuação zerada na atividade 7?



Marque todas que se aplicam.

- Abstração - (Azul marinho)  
 Representação de dados - (Azul celeste)  
 Controle de fluxo - (Rosa)  
 Lógica - (Laranja)  
 Paralelismo - (Amarelo)  
 Interação com o usuário - (Roxo)  
 Sincronização - (Verde)
24. 4) Analisando o gráfico de atividades do quarto aluno(a), é verdade que o maior número de atividades seguidas com pontuação zero no conceito de lógica é 3? \*



Marcar apenas uma oval.

- Sim  
 Não

25. 5) Analisando o gráfico de atividades do quinto aluno(a), quais conceitos estão presente em todas as atividades? \*



Marque todas que se aplicam.

- Abstração - (Azul marinho)
- Representação de dados - (Azul celeste)
- Controle de fluxo - (Rosa)
- Lógica - (Laranja)
- Paralelismo - (Amarelo)
- Interação com o usuário - (Verde)
- Sincronização - (Roxo)

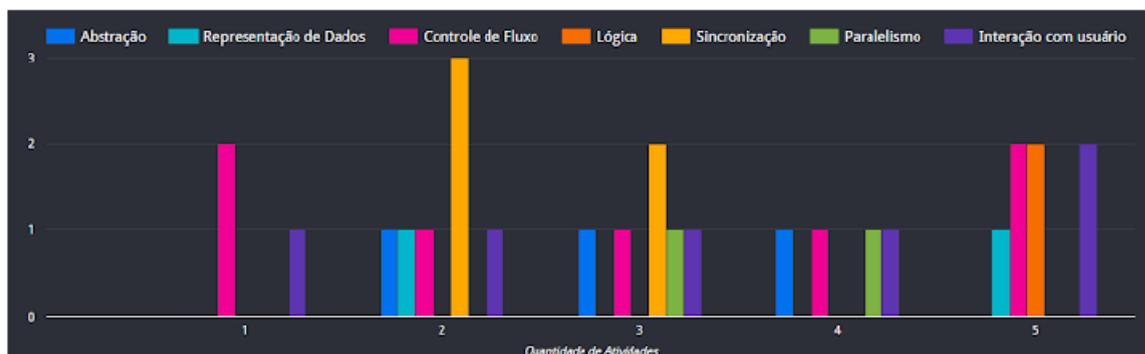
26. 6) Analisando o gráfico de atividades do sexto aluno(a), é verdade que nenhuma atividade pontuou nos 7 conceitos? \*



Marcar apenas uma oval.

- Sim
- Não

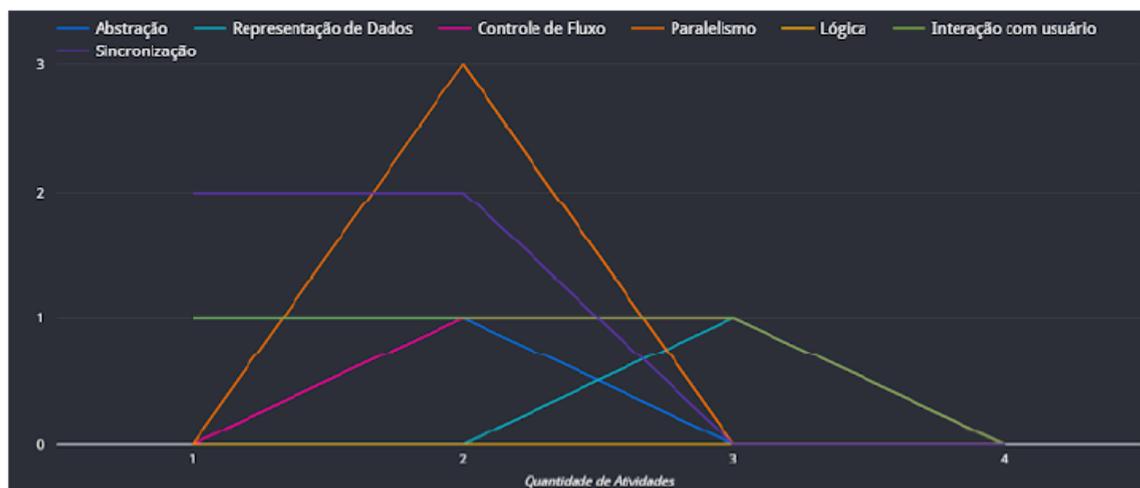
27. 7) Analisando o gráfico de atividades do sétimo aluno(a), quais conceitos estão presente em todas as atividades? \*



Marque todas que se aplicam.

- Abstração - (Azul marinho)
- Representação de dados - (Azul celeste)
- Controle de fluxo - (Rosa)
- Lógica - (Laranja)
- Paralelismo - (Verde)
- Interação com o usuário - (Roxo)
- Sincronização - (Amarelo)

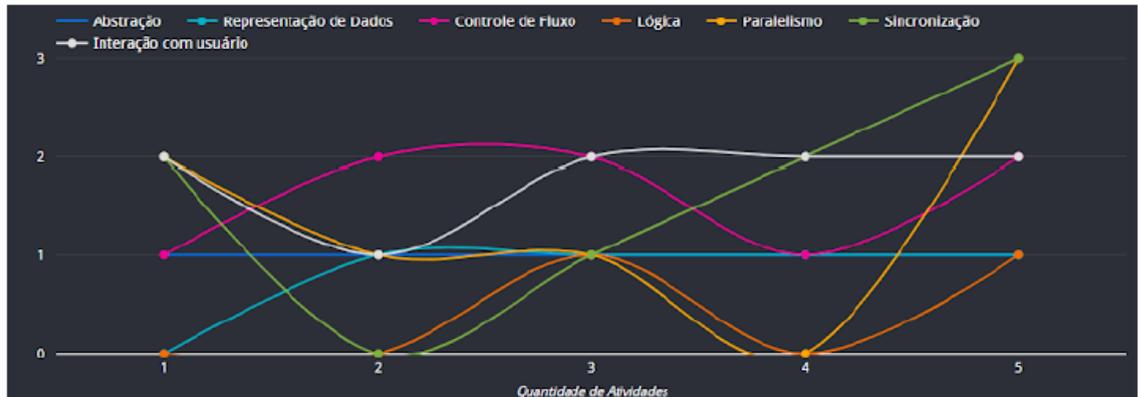
28. 8) Analisando o gráfico de atividades do oitavo aluno(a), o que ocorre com os dados da atividade 4? \*



Marcar apenas uma oval.

- Pontuou apenas no conceito Interatividade com o usuário.
- Zerou a pontuação apenas nos conceitos Interação com o usuário e Sincronização.
- Zerou a pontuação em todos os conceitos.
- Todas as alternativas estão incorretas.

29. 9) Analisando o gráfico de atividades do nono aluno(a), é verdade que desde a primeira atividade o conceito de sincronização apenas progrediu na pontuação? \*

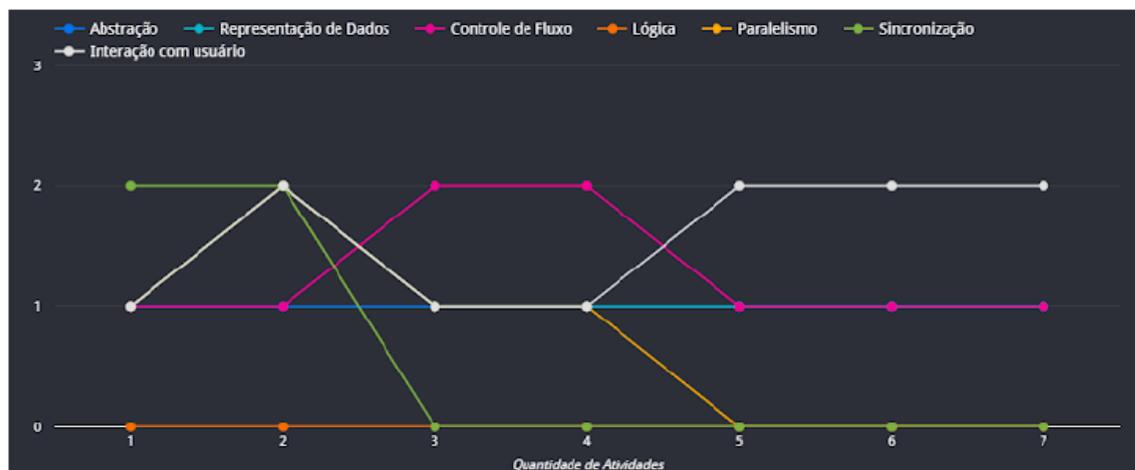


Marcar apenas uma oval.

Sim

Não

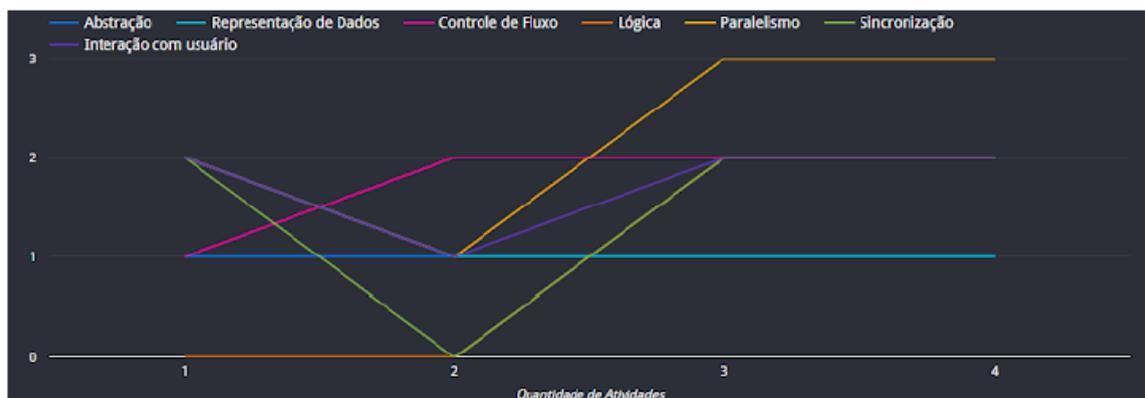
30. 10) Analisando o gráfico de atividades do décimo aluno(a), qual conceito não foi pontuado em nenhuma atividade? \*



Marcar apenas uma oval.

- Abstração - (Azul marinho)
- Representação de dados - (Azul celeste)
- Controle de fluxo - (Rosa)
- Lógica - (Laranja)
- Paralelismo - (Amarelo)
- Interação com o usuário - (Branco)
- Sincronização - (Verde)

31. 11) Analisando o gráfico de atividades do décimo primeiro aluno(a), qual resposta está correta? \*



Marcar apenas uma oval.

- Obteve pontuação zero em dois conceitos.
- Nos conceitos de representação de dados e abstração, obteve pontuação 1.
- Não pontuou no conceito de lógica.
- Todas as alternativas estão corretas.

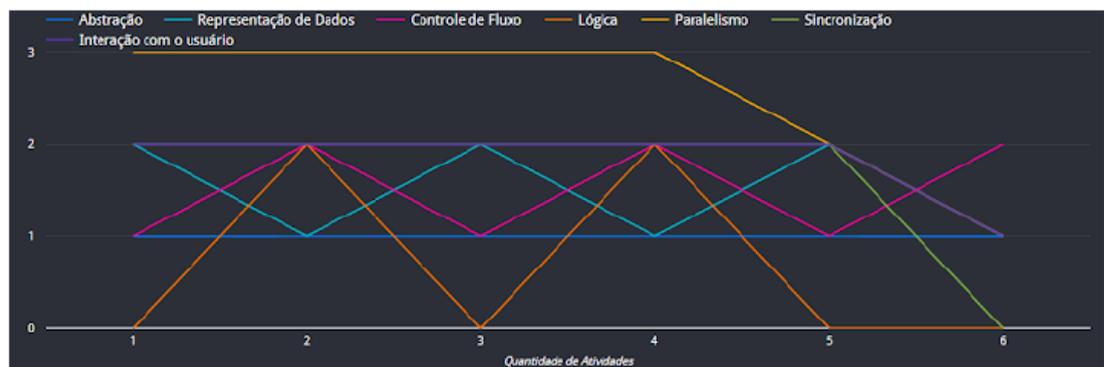
32. 12) Analisando o gráfico de atividades do décimo segundo aluno(a), quantas atividades pontuaram no conceito abstração? \*



Marcar apenas uma oval.

- 3
- 4
- 5
- 6

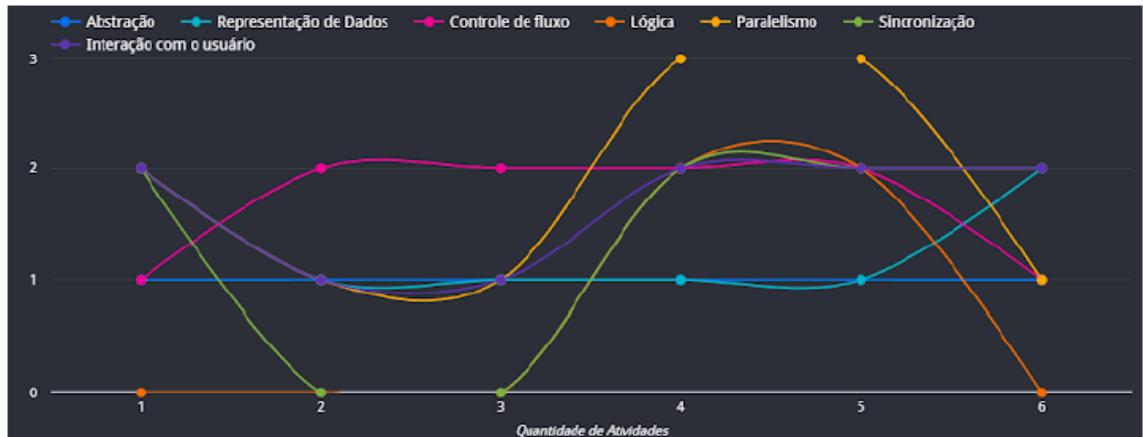
33. 13) Analisando o gráfico de atividades do décimo terceiro aluno(a), quais conceitos pontuaram em todas as atividades? \*



Marque todas que se aplicam.

- Abstração - (Azul marinho)
- Representação de dados - (Azul celeste)
- Controle de fluxo - (Rosa)
- Lógica - (Laranja)
- Paralelismo - (Amarelo)
- Interação com o usuário - (Roxo)
- Sincronização - (Verde)

34. 14) Analisando o gráfico de atividades do décimo quarto aluno(a), é verdade que em 4 atividades não houve pontuação zero em nenhum conceito? \*

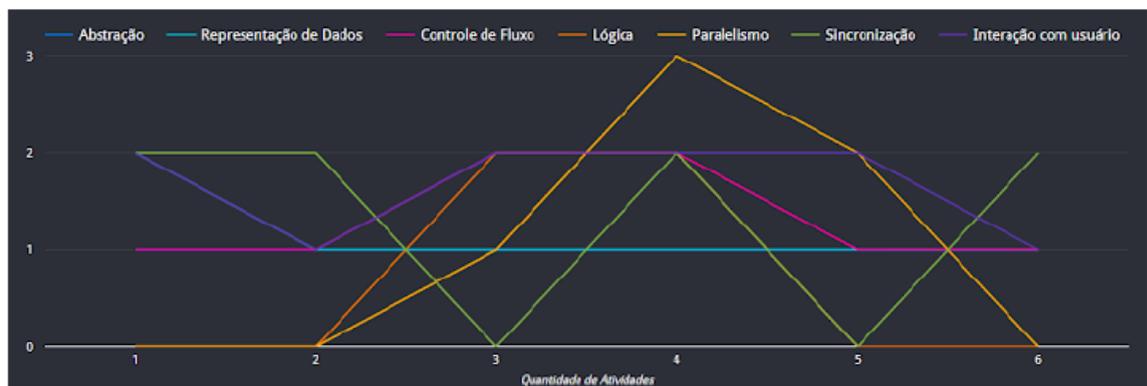


Marcar apenas uma oval.

Sim

Não

35. 15) Analisando o gráfico de atividades do décimo quinto aluno(a), é verdade que pelo menos em uma atividade todos os conceitos tiveram pontuação zero? \*

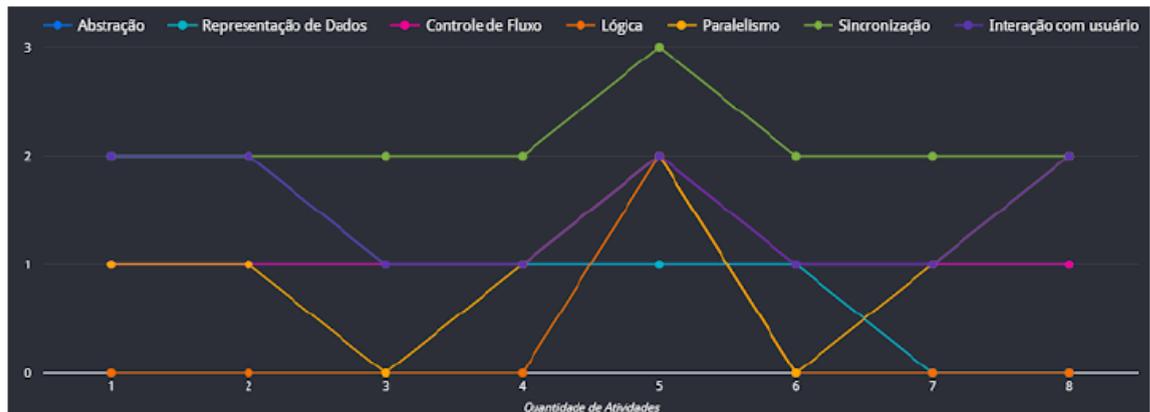


Marcar apenas uma oval.

Sim

Não

36. 16) Analisando o gráfico de atividades do décimo sexto aluno(a), é verdade que a primeira pontuação para o conceito de sincronização foi na atividade 2 e em seguida não houve nenhum declínio menor do que a pontuação inicial? \*



Marcar apenas uma oval.

Sim

Não

37. Após observar os gráficos de todos os alunos, qual é o conceito que menos pontuou nas atividades? \*

Marcar apenas uma oval.

Abstração

Representação de dados

Controle de fluxo

Lógica

Paralelismo

Interação com o usuário

Sincronização