



UNIVERSIDADE ESTADUAL DO PIAUÍ
CENTRO DE TECNOLOGIA E URBANISMO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Quemia Caroline Alves de Oliveira

**SHAPELAB – Uma Abordagem Visual para o Ensino de
Programação a Alunos Surdos por Meio do *Shape Coding* e
MDE**

TERESINA

2025

Quemia Caroline Alves de Oliveira

**SHAPELAB – Uma Abordagem Visual para o Ensino de
Programação a Alunos Surdos por Meio do *Shape Coding* e
MDE**

Monografia de Trabalho de Conclusão de Curso
apresentado na Universidade Estadual do Piauí -
UESPI como parte dos requisitos para a conclu-
são do Curso de Bacharel em Ciência da Com-
putação.

Orientador: Lianna Mara Castro Duarte

TERESINA

2025

048s Oliveira, Quemia Caroline Alves de.

SHAPELAB - Uma Abordagem Visual para o Ensino de Programação a Alunos Surdos por Meio do Shape Coding e MDE / Quemia Caroline Alves de Oliveira. - 2025.

51f.: il.

Monografia (Graduação) - Universidade Estadual do Piauí - UESPI, Centro de Tecnologia e Urbanismo, Bacharelado em Ciência da Computação, 2025.

"Orientadora: Dr.ª Lianna Mara Castro Duarte".

1. Shape-Coding. 2. Programação. 3. Surdo. I. Duarte, Lianna Mara Castro . II. Título.

CDD 004.678

SHAPELAB – Uma Abordagem Visual para o Ensino de Programação a Alunos Surdos por Meio do *Shape Coding* e MDE

Quemia Caroline Alves de Oliveira

Monografia de Trabalho de Conclusão de Curso apresentado na Universidade Estadual do Piauí - UESPI como parte dos requisitos para a conclusão do Curso de Bacharel em Ciência da Computação.

Documento assinado digitalmente

 LIANNA MARA CASTRO DUARTE
Data: 10/07/2025 13:26:24-0300
Verifique em <https://validar.iti.gov.br>

Dra. Lianna Mara Castro Duarte
Orientadora

Nota da Banca Examinadora: 9,0

Banca Examinadora:

Documento assinado digitalmente

 LIANNA MARA CASTRO DUARTE
Data: 10/07/2025 13:11:24-0300
Verifique em <https://validar.iti.gov.br>

Dra. Lianna Mara Castro Duarte
Presidente

Documento assinado digitalmente

 JOSE DE RIBAMAR MARTINS BRINGEL FILHO
Data: 11/07/2025 15:35:43-0300
Verifique em <https://validar.iti.gov.br>

Dr. José de Ribamar Martins Bringel Filho
Membro

Documento assinado digitalmente

 CARLOS GIOVANNI NUNES DE CARVALHO
Data: 10/07/2025 12:29:36-0300
Verifique em <https://validar.iti.gov.br>

Dr. Carlos Giovanni Nunes de Carvalho
Membro

*Este trabalho é dedicado à minha mãe,
por todas as vezes em que dormi antes de a senhora chegar
e acordei depois de a senhora sair.*

AGRADECIMENTOS

Agradeço à Deus e a meus guias pela minha vida e pela vida de todas as pessoas que fazem e fizeram parte dela.

A minha mãe Rosileide e meu pai Francisco por me incentivarem, pelo suporte incondicional e compreensão durante essa jornada acadêmica com paciência, amor e suporte; aos meus irmãos, avós e namorado pelo apoio e compreensão das muitas noites de mal humor resolvendo atividades, e para os que mesmo não estando mais aqui, sei que ficariam muito felizes com essa conquista.

Em especial quero agradecer a minha mãe, que como educadora, me mostrou desde cedo o valor, a recompensa e a importância da educação na vida de uma pessoa, e que todos merecem ter o melhor acesso possível a informação, a ideia desse trabalho surgiu quando eu estava ajudando a senhora a cortar aquelas dezenas formas com desenhos para auxiliar na alfabetização dos seus alunos.

Aos colegas de curso que tornaram a caminhada durante a graduação mais leve, em especial ao João, Leonardo e Luan, com quem compartilhei melhores momentos durante a minha trajetória na UESPI.

Aos meus professores que me incentivaram ao longo do curso e em especial a professora Lianna Duarte por aceitar ser minha orientadora e me mostrar um lado mais humano que a programação pode tomar. A todos que direta ou indiretamente fizeram parte da minha formação, *"Muito Obrigada"*.

Por fim, agradeço a mim também, a Quemia criança ficaria muito orgulhosa da gente ter tornado nosso sonho de infância real e não ter desistido de ser uma cientista.

“Libraries were full of ideas—perhaps the most dangerous and powerful of all weapons.”
(Maas, Sarah J. ; Throne of Glass, 2012)

RESUMO

Observa-se no Brasil mudanças significativas na percepção das pessoas surdas como indivíduos. Atualmente, é notável o aumento da presença de estudantes com diferentes formas de deficiência nas universidades, com um crescimento anual dos estudantes surdos. Com o aumento no número desses estudantes, cresce a necessidade de adaptação das instituições para receber esse grupo de alunos. A principal dificuldade surge da divergência entre as linguagens utilizadas por docentes e discentes. Isso pode acarretar prejuízos em diversas disciplinas ao longo dos cursos, caso os alunos não compreendam plenamente o conteúdo ministrado nas disciplinas. Este trabalho tem como objetivo o desenvolvimento de uma abordagem metodológica de apoio docente, voltada para a disciplina de Programação Estruturada. Para isso, adotou-se a metodologia do *Shape Coding*, visando aproximar a didática e a apresentação do conteúdo à forma como os estudantes surdos aprendem, promovendo, assim, uma prática pedagógica mais inclusiva. A proposta baseia-se na adaptação de estruturas da linguagem de programação a elementos visuais, como formas e cores, conforme os princípios do *Shape Coding*. Para demonstrar a viabilidade dessa abordagem, foram elaborados protótipos gráficos no ambiente Eclipse Sirius, permitindo a modelagem visual de algoritmos e a validação conceitual da proposta. Além disso, a iniciativa considerou diretrizes de acessibilidade e a compatibilidade com softwares previamente utilizados pelos alunos, com o objetivo de garantir maior usabilidade e familiaridade no processo de aprendizagem.

Palavras-chaves: Programação, Surdo, Ensino-superior, *Shape-Coding*.

ABSTRACT

Significant changes have been observed in Brazil regarding the perception of deaf individuals as full members of society. Currently, there is a noticeable increase in the presence of students with various types of disabilities in higher education, with a growing number of deaf students enrolling each year. As this number rises, so does the need for institutions to adapt to adequately accommodate this group. One of the main challenges lies in the divergence between the languages used by instructors and deaf students, which can hinder comprehension and lead to academic difficulties across multiple disciplines. This study aims to develop a methodological teaching approach to support instructors in the context of the “Structured Programming” course. To achieve this, the Shape Coding methodology was adopted to align instructional strategies and content presentation with the way deaf students learn, thereby promoting a more inclusive pedagogical practice. The proposed approach is based on adapting programming language structures into visual elements—such as shapes and colors—according to the principles of Shape Coding. In order to demonstrate the feasibility of this approach, graphical prototypes were developed using the Eclipse Sirius environment, allowing for the visual modeling of algorithms and conceptual validation of the methodology. Furthermore, the initiative took into account accessibility guidelines and the compatibility with software tools already familiar to students, aiming to ensure greater usability and a more effective learning experience.

Keywords: Programming, Deaf, College-Education, Shape-Coding.

LISTA DE ILUSTRAÇÕES

Figura 1 – Metodologia da Pesquisa	19
Figura 2 – Sistema do <i>Shape Coding</i>	24
Figura 3 – Fluxograma - Média	27
Figura 4 – <i>Shape Coding</i> - Média	27
Figura 5 – Fluxo de Trabalho do EMF	28
Figura 6 – <i>Snowballing procedure</i>	30
Figura 7 – Diagrama Metamodelo A da Linguagem	37
Figura 8 – Representação do Algoritmo utilizando o <i>Sirius</i>	38
Figura 9 – <i>Sirius Model</i>	39
Figura 10 – Diagrama Metamodelo B da Linguagem	40
Figura 11 – Representação do Algoritmo com <i>Shape Coding</i>	41
Figura 12 – Representação da Propriedade de <i>Operation</i>	42
Figura 13 – Representação da Propriedade de <i>Output</i>	42
Figura 14 – <i>Sirius Model</i>	43
Figura 15 – Representação do Algoritmo SOMA com <i>Shape Coding</i>	44
Figura 16 – Representação do Algoritmo “MaiorQue” com <i>Shape Coding</i>	45
Figura 17 – Algoritmo utilizando laço <i>while</i>	46

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

DLD	Transtorno do Desenvolvimento da Linguagem
EAD	Educação a Distância
EMF	<i>Eclipse Modeling Framework</i>
L1	Língua Primária
LIBRAS	Língua Brasileira de Sinais
LP	Língua Portuguesa
MDE	<i>Model-driven Engineering</i>
MOF	<i>Meta Object Facility</i>
UESPI	Universidade Estadual do Piauí
UML	<i>Unified Modeling Language</i>
XML	<i>eXtensible Markup Language</i>

LISTA DE QUADROS

Quadro 1 – Palavras-chaves e sinônimos.	32
Quadro 2 – Cores, Formas e Significados do <i>Shape Coding</i>	36
Quadro 3 – Diferença entre Variable e Declare	40

SUMÁRIO

1	INTRODUÇÃO	16
1.1	Justificativa	17
1.2	Objetivos	18
1.2.1	Objetivo Geral	18
1.2.2	Objetivos Específicos	18
1.3	Metodologia da Pesquisa	18
1.4	Estrutura dos Capítulos	20
2	REFERENCIAL TEÓRICO	22
2.1	LIBRAS e a Tridimensionalidade	22
2.2	Visualidade na educação de surdos	23
2.3	<i>Shape Coding</i>	24
2.4	Disciplina de Programação	25
2.5	Fluxograma e <i>Shape Coding</i>	26
2.5.1	<i>Model-Based Engineering</i>	27
2.5.2	<i>Eclipse Modeling Framework</i>	28
3	REVISÃO DA LITERATURA	30
3.1	Protocolo de Revisão	30
3.2	Questões de Pesquisa	31
3.3	Fontes de Pesquisa	31
3.4	Definição do Conjunto Inicial	31
3.4.1	Palavras-Chaves	32
3.4.2	<i>Strings</i> de Busca	32
3.5	Seleção dos Projetos	33
3.6	Resultados da Revisão de Literatura	33
4	<i>SHAPELAB</i>	35
4.1	Adaptação do <i>Shape Coding</i> para o ensino de programação	35
4.2	Metamodelos Criados	36
4.2.1	Metamodelo A	36
4.2.1.1	Representação no editor gráfico	37
4.2.1.2	Editor Gráfico Metamodelo A	38
4.2.2	Metamodelo B	39
4.2.2.1	Editor Gráfico Metamodelo B	41
4.2.2.2	Resultado do Editor Gráfico	42

5	RESULTADOS	44
5.1	Resultado do Metamodelo A	44
5.2	Resultado do Metamodelo B	44
5.3	Prova de Conceito	45
6	CONCLUSÃO	47
6.1	Resultados Obtidos	47
6.2	Trabalhos Futuros	47
	Referências Bibliográficas	49

1 INTRODUÇÃO

No Brasil, observamos significativos avanços nas políticas públicas voltadas para a inclusão das pessoas surdas, as quais são reconhecidas não como indivíduos deficientes, mas sim como cidadãos com características distintas (Carvalho et al., 2020). Conforme demonstrado por um estudo realizado em parceria entre o Instituto Locomotiva e a Semana da Acessibilidade Surda, em 2019, a população de deficientes auditivos no Brasil alcançava a marca de 10,7 milhões de pessoas. Dentre esse contingente, 2,3 milhões enfrentavam deficiência auditiva severa, o que significa que não tinham qualquer capacidade de audição.

No contexto atual, a presença de estudantes com alguma forma de deficiência nas universidades é notável, de acordo com o Censo do Ensino Superior de 2022, o número era de 79.262 ingressantes, correspondendo a 0.8% do total de alunos matriculados. Dentre esse contingente, é importante ressaltar que os deficientes auditivos compreendem uma parcela significativa, totalizando 8.722 ingressantes (Brasil, 2022). A Lei de Educação Bilíngue de Surdos trouxe modificações à Lei de Diretrizes e Bases da Educação Nacional, reconhecendo a Língua Brasileira de Sinais (LIBRAS) como a Língua Primária (L1) e a Língua Portuguesa (LP) como a Segunda Língua (Brasil, 2021). Ainda assim, há quem não compreenda que a LP não está distante da vida do surdo como o Inglês, o Espanhol ou ainda o Francês pode estar da vida de um aluno ouvinte (Viana, 2019).

Diante disso, uma das principais dificuldades que os alunos surdos enfrentam é a fluência em uma língua diferente daquela usada pelo professor, o que resulta em desafios adicionais na compreensão do conteúdo apresentado. Saussure discorre em seu livro o conceito de signo linguístico que em línguas orais, como a LP, o signo linguístico une não uma coisa e uma palavra, mas um conceito e uma imagem acústica (Saussure, 2008). Por meio dessa definição, verificamos que o signo é uma entidade psíquica de duas faces: o conceito e a imagem acústica. A primeira refere-se à imagem mental, ao referente que temos para designar o signo e a segunda refere-se à sequencia fônica que utilizamos para designar o signo (Decian; Méa, 2005).

A particularidade distintiva da LIBRAS, como anteriormente destacado, reside em sua natureza tridimensional, na qual a representação visual substitui a representação acústica. Se por sua natureza auditiva o significante acústico se articula de forma linear, o significante das línguas gesto-visorais assume a natureza tridimensional do espaço visual em que se articula (LESSA-DE-OLIVEIRA, 2013). Por exemplo uma palavra em LIBRAS tem como seus elementos constitutivos - mão, olho, movimento - que ocupam o espaço tridimensional, de forma simultânea. A presença de cada elemento é considerada no espaço, sem o desaparecimento, como na realização de signos acústicos, onde um fonema vai dando lugar a outro para sua construção.

A incorporação de recursos visuais no ensino de surdos desempenha um papel de ex-

trema relevância, auxiliando na interpretação e na atribuição de significado (Carvalho et al., 2020). Nesse contexto, partindo do pressuposto apresentado por Costa de que ”a língua de sinal é visual, o surdo não”, surge uma oportunidade de explorar a dimensão visual dessa língua, permitindo a criação de materiais de apoio que capitalizem todo o potencial visual da LIBRAS (Costa, 2020).

1.1 Justificativa

A comunidade surda tem seus direitos garantidos por lei a muito tempo, porém é nítida as dificuldades no processo de inclusão dos alunos surdos no processo escolar como um todo (Moret; Mendonça, 2023). Boa parte das tecnologias assistivas para os deficientes é direcionada à tradução do Português para LIBRAS.

Porém no contexto em que a disciplina é aplicada de forma presencial, não existe possibilidade de utilizar esse recurso. Tendo em vista que o aprendizado do surdo se dá por uma comunicação gesto-visuais e que durante a maior parte da trajetória escolar deste alunado, eles tiveram pouco ou nenhum contato com materiais acessíveis, faz-se necessário buscar práticas pedagógicas inclusivas.

No entanto, de acordo com Moret e Mendonça (2023), estudos demonstram que os processos de ensino e a aprendizagem das pessoas surdas não estão relacionados diretamente as limitações pedagógicas, mas aos processos de comunicação e expressão na língua portuguesa.

Neste contexto, torna-se necessário a criação de espaços educacionais onde as diferenças sejam respeitadas, onde exista empatia, desejo de fazer melhorias, de transformar a educação e sobretudo, um olhar sob a ótica do aluno e suas necessidades.

Na Universidade Estadual do Piauí (UESPI), o curso de Ciência da Computação possui em sua grade curricular a disciplina de Programação Estruturada, onde o aluno entende conceitos de uma linguagem de programação estruturada de alto nível e suas particularidades.

As aulas, em sua maioria, são geralmente ministradas no laboratório de informática, tendo a prática como didática principal e a teoria em sala de aula como reforço.

De acordo com o decreto nº 5.626, de dezembro de 2005, seu artigo 23, diz: As instituições federais de ensino, de educação básica e superior, devem proporcionar aos alunos surdos equipamentos e tecnologias que viabilizem o acesso à comunicação, à informação e à educação, porém essa não é a realidade presenciada na universidade (Brasil, 2005).

Certificando o que foi mencionado, é importante destacar que uma das principais ferramentas de acessibilidade utilizadas para o ensino de surdos é o tradutor Português-LIBRAS. Entretanto, ao considerar a abordagem de ensino atualmente adotada, a utilização de tradutores, como comumente ocorre na Educação a Distância (EAD), torna-se inviável.

Isso se deve à necessidade de assegurar que o aluno comprehenda plenamente o conteúdo

e seja capaz de executar as tarefas de forma precisa e adequada.

A maior parte dos educadores estão presos em metodologias e técnicas de ensino voltadas aos ouvintes, no entanto, os surdos possuem sua cultura, identidade e sua forma de construir o seu conhecimento com o uso da língua de sinais (Carvalho et al., 2020).

Partindo do pressuposto de Moret e Mendonça (2023) “quando há um complemento visual, a aula se torna mais atrativa e desperta no aluno o interesse pelo conhecimento, assim contribui na construção da aprendizagem”. Desta forma, o uso da metodologia do *Shape Coding* é um recurso a ser considerado.

1.2 Objetivos

1.2.1 Objetivo Geral

O objetivo geral deste trabalho é o desenvolvimento de uma abordagem focada no ensino de Programação Estruturada para alunos surdos utilizando a metodologia *Shape Coding*. A partir dela, os professores poderão lecionar a matéria de forma inclusiva para todos os alunos.

1.2.2 Objetivos Específicos

Para alcançar o objetivo geral pretendido, buscar-se-á atingir os seguintes objetivos específicos:

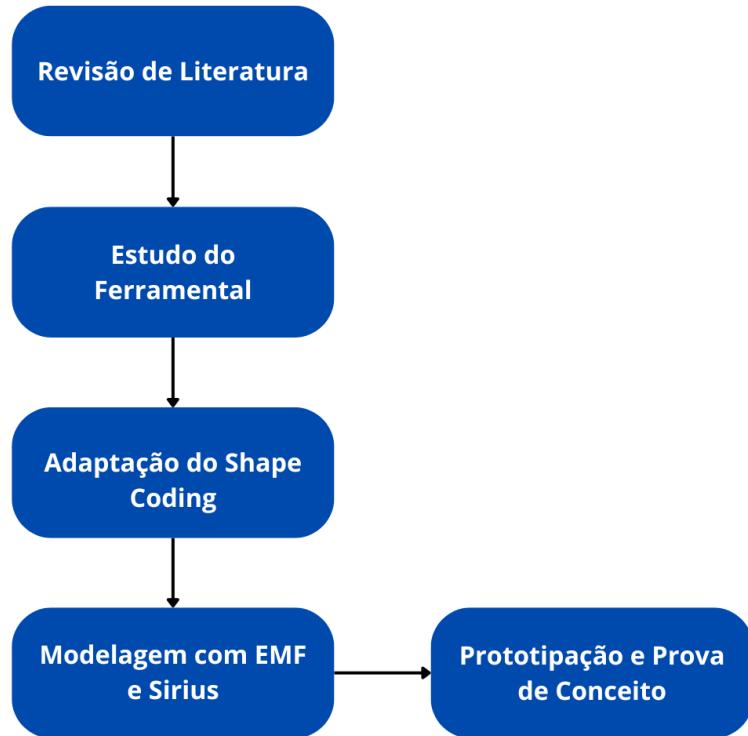
- Entender como os surdos aprendem uma linguagem e adequar o ensino da programação a este aprendizado.
- Pesquisar formas de amparo tecnológico com eficácia comprovada no ensino de surdos.
- Estruturar os conceitos da abordagem do *Shape Coding* para o ensino de programação para surdos, adaptando o conteúdo que será ofertado na disciplina, podendo incluir exemplos visuais, descrições claras e ilustrativas.
- Identificar e documentar as características essenciais para o desenvolvimento da estrutura da abordagem metodológica, optando, para a prototipagem da proposta, pela utilização de softwares previamente utilizados pelos alunos..

1.3 Metodologia da Pesquisa

A metodologia deste trabalho foi estruturada em etapas que visam responder às três questões de pesquisa apresentadas no Capítulo 3, com foco na aplicação do *Shape Coding* como recurso didático acessível no ensino de Programação Estruturada para estudantes surdos. A abordagem adotada é aplicada e qualitativa, com caráter exploratório.

Para alcançar os objetivos específicos e, por fim, o objetivo geral desta pesquisa prática, os passos de pesquisa foram realizados de forma ordenada como ilustra a Figura 1.

Figura 1 – Metodologia da Pesquisa



Fonte: Autoria própria

1. Revisão de Literatura

Foi realizada uma revisão sistemática da literatura utilizando o método *Snowballing*, conforme proposto por Wohlin (2014). Essa etapa teve como objetivo mapear trabalhos relacionados ao ensino de surdos, visualidade na educação, metodologias bilíngues e aplicação do *Shape Coding* no contexto educacional.

A revisão também permitiu identificar tecnologias assistivas e práticas pedagógicas eficazes para o público-alvo.

2. Estudo e Utilização da Ferramenta

Com base nas evidências coletadas, foram definidos os requisitos pedagógicos e visuais necessários para adaptar o ensino de programação à realidade dos alunos surdos, aprofundando o conhecimento em *Model-driven Engineering* (MDE) e interfaces gráficas acessíveis segundo as diretrizes do W3C, com foco na integração desses elementos para atender às demandas de acessibilidade e usabilidade.

3. Adaptação do *Shape Coding*

A metodologia original do *Shape Coding*, voltada ao ensino da gramática inglesa, foi reinterpretada e adaptada ao contexto computacional.

Foram definidas cores, formas e significados associados a comandos comuns da lógica de programação, como *declare*, *assign*, *input*, *output* e *operation*, conforme apresentado no Quadro 2.

4. Metamodelagem com *Eclipse Modeling Framework* (EMF)

Dois metamodelos foram criados com o EMF, um abstrato presente na Figura 7, focado em representar comandos genéricos, e outro mais detalhado na Figura 10, incluindo estruturas como variáveis, operações e declarações.

A partir desses metamodelos, editores gráficos foram gerados com o uso da ferramenta Sirius, permitindo a construção visual de algoritmos por meio de formas e cores.

5. Prototipação e Prova de Conceito

Foi elaborada uma prova de conceito com base no Metamodelo da Figura 10. O objetivo foi validar a viabilidade da modelagem visual com base no *Shape Coding* como ferramenta de apoio ao ensino. O experimento contemplou a construção de algoritmos simples, como a soma de valores e o uso de estruturas de repetição com *while*.

1.4 Estrutura dos Capítulos

Este trabalho está estruturado em capítulos distintos que abordam diferentes aspectos do estudo. O Capítulo 1 consiste na introdução, em que é apresentado o contexto do estudo, evidenciando a relevância social e acadêmica da pesquisa. São definidos os objetivos gerais e específicos, bem como descrita a metodologia adotada para o desenvolvimento da pesquisa.

O Capítulo 2 aborda o referencial teórico, apresentando os conceitos fundamentais sobre o tema. Este capítulo reúne os principais conceitos que sustentam a pesquisa, como LIBRAS, tridimensionalidade, visualidade na educação de surdos, o sistema *Shape Coding*, além de aspectos da disciplina de Programação Estruturada e do uso de metamodelagem com *EMF*.

O Capítulo 3 contém a revisão da literatura, onde é apresentado o protocolo metodológico utilizado para a seleção dos materiais bibliográficos, com base na técnica de *snowballing*. São detalhadas as questões de pesquisa, as fontes consultadas, os critérios de inclusão e exclusão, bem como os principais achados que reforçam a pertinência da proposta.

O Capítulo 4 apresenta a solução proposta para o problema, além de detalhar a abordagem desenvolvida. Neste capítulo, são descritos os metamodelos desenvolvidos, os recursos visuais aplicados conforme os princípios do *Shape Coding* e o processo de criação do editor gráfico por meio das ferramentas *EMF* e *Sirius*. Também é apresentada uma prova de conceito com exemplos de algoritmos simples modelados visualmente, validando a aplicabilidade da aborda-

gem.

Por fim, o Capítulo 6 apresenta os resultados alcançados após a execução do plano delineado, além de incluir sugestões para trabalhos futuros que possam contribuir para o avanço da área abordada neste trabalho.

2 REFERENCIAL TEÓRICO

Neste capítulo, serão introduzidos os conceitos abordados neste trabalho, tais como LIBRAS e a Tridimensionalidade (*i*), Visualidade na educação de surdos (*ii*), Sistema de *Shape Coding* (*iii*), Disciplina de Programação (*iv*) e o Fluxograma e *Shape Coding* (*v*).

2.1 LIBRAS e a Tridimensionalidade

A Língua Brasileira de Sinais conforme Nóbrega et al. (2021) se constitui como língua, gramática e linguisticamente, pois utiliza dos canais visual e gestual para assegurar a comunicação, bem como facilitar o acesso à informação, ao ensino e à aprendizagem. Uma frase negativa, por exemplo, é compreendida mediante movimento da cabeça, enquanto uma pergunta é identificada pela expressão facial de dúvida. Essas expressões faciais são consideradas complementos essenciais para a compreensão das frases em LIBRAS, e é por essa razão que essa língua se enquadra na modalidade gestual-visual (Silva, 2023).

De acordo com Capovilla e Capovilla (2002), a pessoa ouvinte e falante mantém uma continuidade entre os três contextos comunicativos básicos: comunicação interna (pensamento), comunicação interpessoal direta (fala) e comunicação escrita. Isso permite que todo o processamento linguístico se concentre na língua falada primária, usando as mesmas palavras para pensar, comunicar e escrever. Dessa forma, ao escrever, pode fazer uso intuitivo das propriedades sonoras das palavras da mesma língua usada para pensar e se comunicar. No caso da pessoa surda, são impostas expectativas adicionais. Ela pensa e se comunica com a língua de sinais, que é visual e para escrever, espera-se que o faça utilizando a LP.

Para compreendermos a natureza tridimensional da LIBRAS, é essencial reconhecer que a LP se constitui por meio de um sistema de signos, nos quais qualquer palavra com significado pode ser considerada um signo. Nesse contexto, introduzimos o conceito de "significante", que é a representação mental do som, ou seja, o reconhecimento cognitivo de um som específico.

No entanto, quando exploramos o conceito de "Linearidade do Significante", observamos que os signos na LP são considerados em uma única dimensão, em que um signo segue o outro de forma sequencial, em uma ordem específica. É importante ressaltar que essa linearidade não se aplica à LIBRAS, que é uma língua tridimensional, onde os signos podem ocorrer simultaneamente em diferentes planos e não estão estritamente vinculados à sequência linear, estabelecendo, assim, uma distinção fundamental entre as duas línguas.

"Na maioria dos indivíduos, as impressões visuais são mais nítidas e mais duradouras que as impressões acústicas; dessarte, eles se apegam, de preferência, às primeiras."(Saussure, 2008)

A visualidade da pessoa surda está ligada diretamente ao seu modo de estar no mundo, de se comunicar por meio da LIBRAS e de perceber as informações pelo canal visual. Sendo assim, a visualidade surda se baseia na apreensão que estes têm sobre o mundo, mediante o uso de uma língua visual. Desde o seu nascimento, as crianças surdas interpretam tudo desta forma, os ensinamentos familiares, sociais e ainda escolares, mesmo quando não são incluídos de forma satisfatória. Desta forma, a educação deveria se munir desta característica e trazer significado visual e imagético para a sala de aula (Viana; Lebedeff, 2021).

2.2 Visualidade na educação de surdos

Campello et al. (2008) argumenta que na sociedade, a comunicação não se limita mais a discursos apenas orais ou escritos, mas se baseia na visibilidade, incluindo elementos imagéticos. Segundo Skliar (2001), a visão do surdo como um "sujeito visual" está relacionada principalmente ao âmbito comunicacional, ou seja, no contexto de compreensão e produção de informações em língua de sinais. No entanto, a experiência do surdo com a visualidade envolve todo tipo de questões comunitárias e culturais (Lebedeff, 2015).

Aspectos da visualidade na educação de Surdos, ou pedagogia surda é assim denominada considerando-se que a mesma pode ser compreendida como aquela que se ergue sobre os pilares da visualidade, ou seja, que tem no signo visual seu maior aliado no processo de ensinar e aprender. Sendo assim, temos o conceito de visualidade e visibilidade, onde, de acordo com Campello et al. (2008), o primeiro se refere à relação entre a percepção e a imagem, que é moldada pelas qualidades do sinal visual.

Por outro lado, a segunda categoria não está diretamente vinculada à imagem, mas se desenvolve a partir dela. Isso ocorre porque, por meio da iconicidade do sinal visual, são estabelecidas relações plausíveis por meio de "descrições imagéticas". Essas descrições possibilitam o surgimento de signos mais complexos, que se baseiam nas representações das informações registradas e visuais, bem como na construção mental da imagem.

As técnicas, recursos e perspectivas utilizados nos aspectos da visualidade na educação de Surdos, estão relacionados com o uso da "visão", em vez da "audição", sendo que a imagem na "apreensão do estímulo visual" e perspectiva emergem de acordo com forças bidimensionais e tridimensionais.

(Campello et al., 2008)

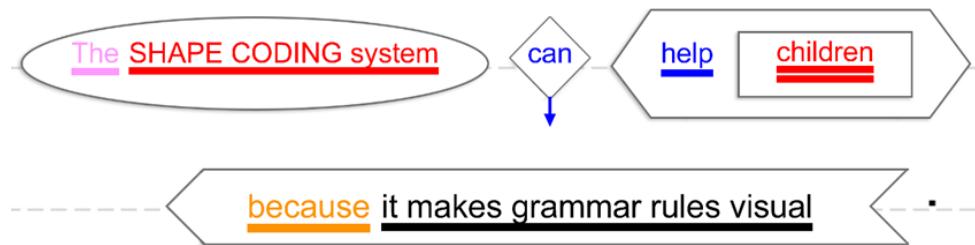
Aliado ao contexto de compreensão pela visualidade, pode-se refletir sobre a importância do letramento visual como uma condição para o processo de aprendizagem das pessoas surdas. As modalidades das duas línguas envolvidas são diferentes, ou seja, a forma viso-espacial da LIBRAS se difere do aspecto oral auditivo das línguas faladas, assim, o processo de desenvolvimento da leitura e escrita dos surdos seria organizado, primeiramente, a partir da compreensão pela visualidade (Sampaio, 2018).

2.3 Shape Coding

O *Shape Coding*, desenvolvido pela fonoaudióloga Dra. Susan Ebbels, foi criado inicialmente com o propósito de ensinar gramática inglesa falada e escrita a crianças em idade escolar com Transtorno do Desenvolvimento da Linguagem (DLD). No entanto esse método se mostrou eficaz não somente para essa finalidade, mas também na aplicação a indivíduos com deficiência auditiva, síndrome de Down e adultos com afasia adquirida.

Essa abordagem utiliza um sistema de codificação visual para ilustrar as regras de como as palavras se organizam em frases, com o intuito de aprimorar a compreensão e o uso da gramática, o que resulta em uma comunicação mais eficaz. O sistema incorpora como pode ser visto na Figura 2, o uso de formas (para representar estruturas sintáticas), cores (para identificar classes de palavras), linhas (para diferenciar singular de plural e gênero) e setas (para indicar tempo verbal e aspecto).

Figura 2 – Sistema do *Shape Coding*



Fonte: (College, 2023)

Conforme Viana (2019), ao conectar as ideias do sistema com os princípios da educação bilíngue para surdos, identificou-se que uma metalinguagem visual para analisar a sintaxe e a gramática da Língua Portuguesa poderia representar uma ferramenta pedagógica de apoio ao professor. Nesse contexto, o elemento visual central é a língua de sinais do sujeito surdo, que se destaca, com a experiência visual servindo como base para as criações, dado que essa característica é intrínseca à surdez.

Em vários países da Europa, o *Shape Coding* é utilizado com crianças com limitações na linguagem escrita. São treinamentos com estratégias visuais de estudos linguísticos explícitos, ou seja, entendendo o funcionamento da língua, apropriando-se dos conceitos gramaticais por meio de figuras metalingüísticas (Viana, 2019).

O *Shape Coding* pode entrar na educação de surdos como alternativa para a reflexão linguística, baseando-se nas concepções que Viana (2019) pontuou:

- A LIBRAS e a visualidade surda devem ser respeitadas e valorizadas dentro do contexto da educação bilíngue para surdos (Campello et al., 2008; Capovilla; Capovilla, 2002; Capovilla; Capovilla, 2006; Lebedeff, 2010)

- O uso de uma consciência metalinguística pode ser compreendida como uma estratégia de reflexão linguística, ocupando momentos específicos do desenvolvimento da competência comunicativa dos alunos e não deve ser a base do ensino de uma segunda língua (Barbosa, 2015)
- O uso do *Shape Coding* pode ser desenvolvido dentro de uma metodologia mais abrangente de ensino de línguas.

Os surdos vivem o bilinguismo diariamente, no qual as relações de transferência linguística se dão a todo o momento com o agravante da descontinuidade das línguas envolvidas - a LIBRAS, visuo-gestual e a LP, oral-auditiva (Campello et al., 2008). A proposta é adaptar e utilizar o sistema do *shape coding*, aproveitando todo o seu potencial visual para correlacioná-lo com a LIBRAS na compreensão da língua portuguesa, a fim da melhor comunicação do professor com o aluno.

2.4 Disciplina de Programação

Assim como observa Moret e Mendonça (2023), a limitação na adaptação das aulas pode prejudicar o aprendizado do aluno surdo, uma vez que ele depende do componente visual para aprender e isso pode resultar na perda de informações cruciais devido à falta de recursos visuais adequados.

Os autores fomentam o seguinte questionamento:

Quais tecnologias podem ajudar na prática pedagógica da educação inclusiva para surdos? Essa pergunta se torna pertinente uma vez que durante a escolarização dos surdos é explícita a falta de recursos didáticos, tecnológicos e visuais nas salas de aula. (Moret; Mendonça, 2023)

Discursos e posicionamentos políticos, educacionais e acadêmicos enfatizam que as necessidades de aprendizes surdos são contempladas a partir de materiais didático-pedagógicos adequados, sendo os recursos visuais ou a visualidade fatores essenciais ou pré-requisitos para a satisfação dessa necessidade, porém essa temática pode ser questionada, uma vez que materiais visuais, por si sós, não levam o aprendiz surdo ao letramento visual. É preciso que as imagens e os signos façam sentido e sejam pertinentes (Sá; Cruz; Pinheiro, 2020).

Existe ainda outro fator a ser considerado: as habilidades e competências dos alunos ingressantes no curso superior. Alguns já possuem noções básicas de programação, mas frequentemente não compreendem os conceitos de forma completa, passando diretamente para a prática sem a fundamentação necessária. Isso pode gerar frustração ao enfrentar dificuldades na resolução de problemas.

Conforme Medeiros (2019), a aprendizagem de programação necessita ser progressiva. O problema identificado é que os estudantes frequentemente iniciam imediatamente a atividade de codificação, sem compreender totalmente o problema em questão. É sugerido que o aluno

desenvolva maturidade para compreender primeiramente a sintaxe básica da linguagem, seguida pela semântica, estrutura e, por fim, o estilo, para finalmente passar a fase de codificação.

Este trabalho concentra-se principalmente na disciplina de Programação Estruturada lecionada na UESPI, que normalmente é ministrada no segundo semestre acadêmico. Essa disciplina serve como pré-requisito para a maioria das áreas da programação e a falta de compreensão efetiva pode ter consequências negativas, como reprovações, desmotivação e até mesmo o abandono do curso.

Reiterando como já foi mencionado, a disciplina é ministrada inicialmente em sala de aula, utilizando a teoria como base para entendimento de uma linguagem de programação compilada e estruturada. Após parte do conteúdo teórico ser passado, as aulas seguem no laboratório de informática, onde os alunos são submetidos a atividades a serem desenvolvidas utilizando o que foi lhes passado em sala de aula, sempre contando com o apoio do professor para tirar dúvidas.

Analizando a forma como a teoria e a prática ocorre, pode-se perceber uma falha na questão da acessibilidade quanto, um aluno enquanto surdo, até poderia entender a parte teórica sem muitos problemas, porém teria mais dificuldade em sanar suas dúvidas quanto a parte prática, uma vez que as ferramentas utilizadas não contam com acessibilidade adequada, como tradutores automáticos virtuais.

2.5 Fluxograma e *Shape Coding*

O fluxograma é uma representação gráfica de um processo ou algoritmo, comumente utilizado na disciplina de introdução a ciência da computação, para ilustrar sequência de passos ou operações de um algoritmo permitindo a visualização da sequência de passos de um processo ou algoritmo. Por outro lado, o *Shape Coding* é um sistema baseado em cores e formas para representar informações ou conceitos. Ele usa cores e formas específicas para associar significados a diferentes elementos, permitindo uma codificação visual de informações.

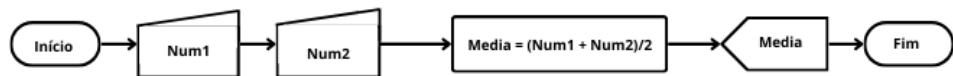
O *Shape Coding* é comumente usado para auxiliar na compreensão de conceitos e no ensino, fornecendo uma representação visual para facilitar a associação de ideias e conceitos. Logo, principal diferença está no modo como cada um é estruturado e utilizada: o fluxograma é mais voltado para representar sequências lógicas e algoritmos, enquanto o *Shape Coding* é mais orientado para a representação de informações por meio de cores e formas específicas.

Ambas as estruturas podem ser empregadas para associar estruturas de código. No entanto, observa-se que o *Shape Coding* apresenta um maior potencial visual no ensino de estudantes surdos. Isso se deve à capacidade de criar formas e cores para estruturas específicas, tornando-o mais preciso e facilmente distinguível do que uma estrutura de fluxograma.

A diferença no potencial visual entre as duas estruturas pode ser observada na Figura 3

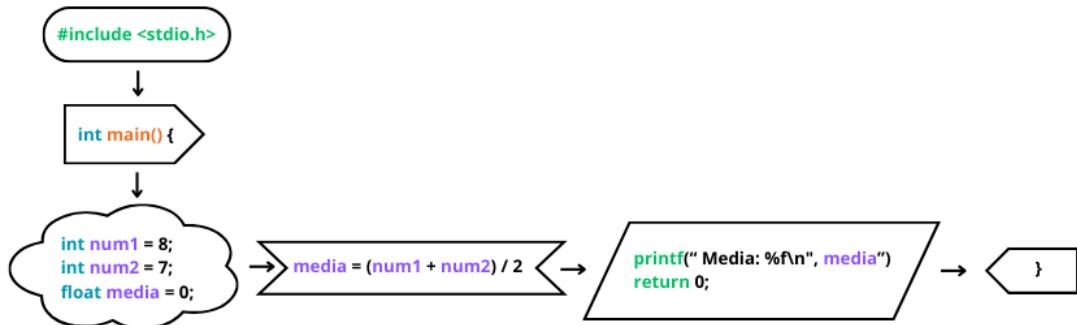
e na Figura 4. Ambas são representações visuais, porém o *Shape Coding* se mostra mais específico, explicativo e de fácil associação. No caso do fluxograma, se um aluno não conhecer o significado de cada forma usada, isso se torna um conhecimento adicional que ele precisa adquirir. Por outro lado, no *Shape Coding*, as formas são flexíveis e sua definição pode ser estabelecida entre professor e aluno para determinar o significado de cada uma e a melhor forma de representação. Isso transmite uma sensação de maior controle ao aluno sobre suas escolhas, facilitando a memorização.

Figura 3 – Fluxograma - Média



Fonte: Autoria própria

Figura 4 – *Shape Coding* - Média



Fonte: Autoria própria

2.5.1 Model-Based Engineering

Sommerville (2010) elucida em seu livro que a MDE é uma abordagem do desenvolvimento de software onde os modelos, em vez de programas, são as saídas principais do processo de desenvolvimento. Os programas executados em um *hardware/software* são, então, gerados automaticamente a partir dos modelos.

De acordo com Kleppe, Warmer e Bast (2003), o modelo define quais elementos podem existir em um sistema. Logo, todo tipo de elemento que o modelador pode usar, é definido pelo metamodelo da linguagem que o projetista usa.

A MDE permite que engenheiros de software trabalhem em um nível mais alto de abstração, sem se preocuparem, inicialmente, com os detalhes de implementação. Sua ideia central é possibilitar a transformação automática de modelos em código-fonte, o que requer a construção de modelos gráficos com semântica bem definida (Sommerville, 2010).

Podemos descrever uma linguagem por meio de um modelo, que define os elementos válidos e as regras de composição dessa linguagem (Kleppe; Warmer; Bast, 2003). Esse modelo é também conhecido como metamodelo. Essa modelagem pode ser feita com o uso da *Unified Modeling Language* (UML) originalmente concebida como uma linguagem de apoio à especificação e documentação de projetos de *software*, e não como uma linguagem de programação.

2.5.2 Eclipse Modeling Framework

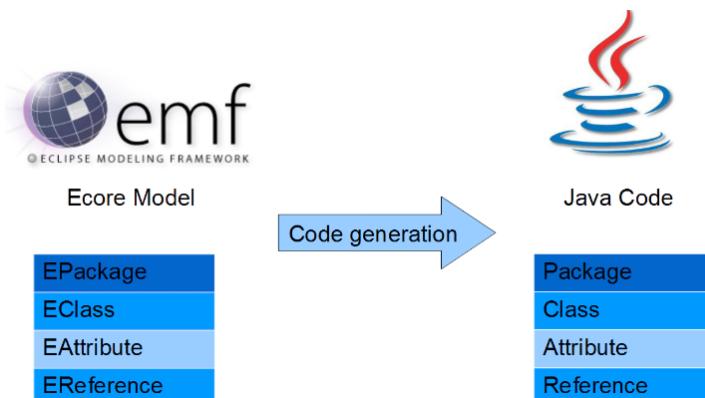
Eclipse Modeling Framework (EMF) é uma estrutura de modelagem que explora os recursos fornecidos pelo Eclipse (Steinberg et al., 2008).

Em seu livro Steinberg et al. (2008) nos apresenta o EMF como um *framework* e um recurso para geração de código que permite definir um modelo em diferentes formatos como Java, *eXtensible Markup Language* (XML) ou UML.

De acordo como o Eclipse (2024), o EMF fornece ferramentas e suporte em tempo de execução para produzir um conjunto de classes Java para o modelo, a partir de um modelo descrito em XML. Ele cria um conjunto de classes adaptadoras que permitem a visualização e edição do modelo.

A partir desses modelos (metamodelos), os desenvolvedores podem gerar automaticamente outros artefatos, como classes de implementação, facilitando o desenvolvimento de software baseado em modelos (Steinberg et al., 2008).

Figura 5 – Fluxo de Trabalho do EMF



(EclipseSource, 2024)

A Figura 5 ilustra o fluxo de trabalho básico do EMF, no qual um modelo é criado e definido no formato Ecore — um subconjunto de diagramas de classes UML. A partir desse modelo, é possível gerar automaticamente código Java (EclipseSource, 2024). Cada metaclasses no metamodelo é mapeada diretamente para uma classe Java correspondente.

O Ecore, por sua vez, é baseado no *Meta Object Facility* (MOF), um metametamodelo padronizado (Steinberg et al., 2008). Dessa forma, o EMF adota os conceitos e estruturas definidos pelo MOF para construir sua linguagem de modelagem e viabilizar a geração automática de código a partir dos modelos definidos (Steinberg et al., 2008).

3 REVISÃO DA LITERATURA

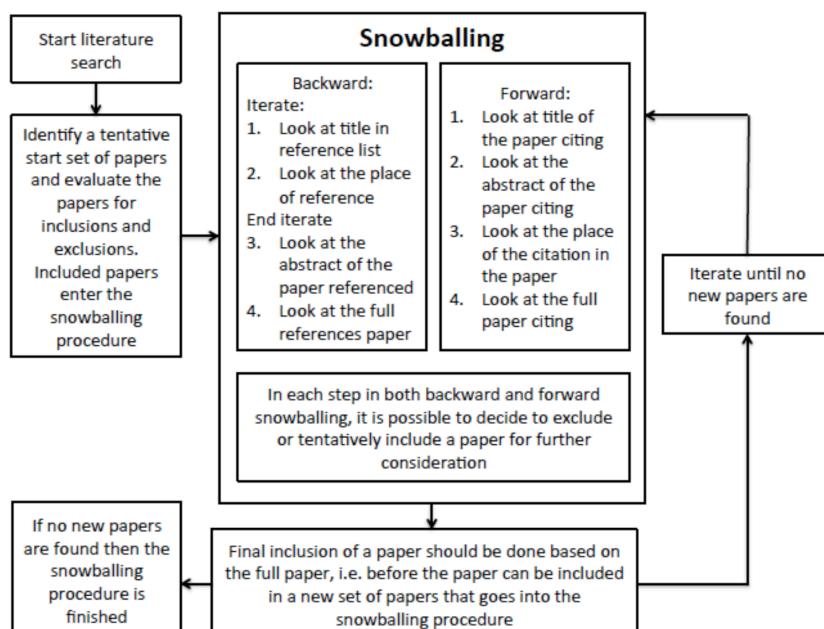
Neste capítulo, apresentaremos o protocolo empregado para conduzir a revisão de literatura, que serviu de base para este trabalho.

3.1 Protocolo de Revisão

O planejamento da revisão da literatura foi elaborado utilizando a técnica de *snowballing* proposta por Wohlin (2014). O método *Snowballing*, como demonstrado na Figura 6, inicia-se com a formulação de *strings* de busca na etapa inicial, com o propósito de obter um conjunto inicial de documentos.

Após identificar esse primeiro conjunto, realiza-se uma pesquisa nas referências em busca de trabalhos que atendam aos critérios de exclusão, como tema abordado e ano de publicação; essa fase é denominada *Backward Snowballing*. Em seguida, ocorre a segunda busca por trabalhos que citaram alguns dos documentos contidos no conjunto inicial, podendo utilizar ferramentas como o *Google Scholar*; esse processo é conhecido como *Forward Snowballing*. Ao término da primeira iteração, os novos trabalhos encontrados são incorporados ao conjunto, e o processo é repetido em iterações subsequentes.

Figura 6 – *Snowballing procedure*



Fonte: (Wohlin, 2014)

3.2 Questões de Pesquisa

As questões que nortearam o desenvolvimento da pesquisa nas bases bibliográficas foram:

Q1. - Como podemos fornecer amparo tecnológico eficaz para um aluno com surdez matriculado no curso de Ciência da Computação, a fim de assegurar sua compreensão eficiente do conteúdo da matéria de Programação Estruturada?

O intuito dessa questão é identificar recursos e estratégias que possam ajudar os educadores no ensino de alunos surdos, concentrando-se na melhor maneira do professor se comunicar de forma eficaz com esses alunos, mesmo que não compartilhem o mesmo idioma.

Q2. - Como o surdo aprende uma linguagem e como o *Shape Coding* se encaixa no papel de facilitador deste aprendizado?

Esta questão visa explorar estudos destinados a compreender o processo de aprendizagem das pessoas surdas, incluindo sua percepção do mundo. Uma vez compreendido esse aspecto, o objetivo é identificar e implementar métodos de ensino que se alinhem de maneira mais efetiva com as formas de aprendizagem fundamentais para esse grupo.

Q3. - Como o uso do *Shape Coding* pode trazer alguma melhora para o ensino da programação?

Esta pergunta procura investigar e revisar estudos, especialmente no campo da ciência da computação, que empregam a metodologia do *Shape Coding* para o ensino de programação.

3.3 Fontes de Pesquisa

As bases bibliográficas selecionadas para a busca foram:

- ACM Digital Library <<https://dl.acm.org>>
- EduCAPES <<https://educapes.capes.gov.br/>>
- Anais do CEIE <<http://milanesa.ime.usp.br/rbie/>>
- Google Scholar <<https://scholar.google.com.br/?hl=pt>>

3.4 Definição do Conjunto Inicial

Inicialmente, realizou-se uma busca manual de artigos relacionados ao tema diretamente no *Google Scholar*. Após obter o primeiro conjunto de artigos, foram identificadas as palavras-chave, seguindo o fluxo do *snowballing*.

3.4.1 Palavras-Chaves

A Quadro 1 apresenta uma lista de palavras-chave utilizadas na busca deste trabalho, juntamente com seus respectivos sinônimos e termos relacionados.

Quadro 1: Palavras-chaves e sinônimos.

Palavras-Chaves	Sinônimos ou Termos Equivalentes
Surdo	Comunidade surda, <i>Deaf, Hearingimpaired students</i>
LIBRAS	Língua Brasileira de Sinais, Língua de Sinais <i>Language, Sign</i>
Aplicativo Educacional	Jogos educacionais digitais
Educação	Educação de Surdos, Ensino Superior Educação de sujeitos Surdos
Computação	Ciências da computação, <i>Computational, computer programming education</i>
Visualidade	Descrição Imagética
<i>Shape Coding</i>	<i>Shape Coding</i> adaptado

Fonte: Autoria própria

3.4.2 *Strings* de Busca

Cada *string* corresponde a uma questão de pesquisa diferente, onde utilizando das palavras-chave, sinônimos e termos equivalentes já definidos, temos as seguintes *strings* de busca:

- A busca por métodos, ou ferramentas auxiliares que utilizem do *Shape Coding* como instrumento de ensino, ou apoio educacional na ciência da computação.
 - (Computação OR ``Ciência Da Computação'' OR *Computational*)
AND (``Educação de Surdos'' OR Educação)
AND (*Shape-coding* OR *shape-coding* adaptado)
- A procura equipamentos e técnicas de apoio docente com o intuito de lecionar de forma eficiente a alunos surdos no curso superior de ciência da computação.
 - (``Ciência Da Computação'' OR *Computational*)
AND (``Educação de Surdos'' OR Educação)
AND (``Ensino Superior'')
- Entendendo como o processo de aprendizagem da linguagem dos surdos funciona, quais as características e particularidades que os alunos possuem.
 - (*Sign* OR *language*)
AND (*Deaf* OR Surdos OR *Hearingimpaired students*)
AND (*Shape-coding* OR *shape-coding* adaptado)

3.5 Seleção dos Projetos

Foram estabelecidos os seguintes critérios de inclusão e exclusão de estudos:

- Critérios de Inclusão

1. Artigos, Teses ou Dissertações completas em inglês ou português;
2. Trabalhos, pesquisas ou livros que tratem de linguística em geral e de linguística para surdos;
3. Estudos que tratem de educação para surdos em todos os níveis de ensino e em diversas áreas aplicadas;
4. Trabalhos e pesquisas que tratem de educação para surdos no ensino superior;
5. Trabalhos e pesquisas que tratem de aplicativos ou técnicas de apoio docente a alunos com surdez em todos os níveis de ensino;
6. Artigos publicados de 2019 e 2023;

- Critérios de Exclusão

1. Artigos e trabalhos duplicados que tiveram sua versão mais recente e não tem informações complementares;
2. Estudos cujo texto completo não estava disponível;
3. Materiais que não se enquadram no escopo da revisão, como resumos, relatórios e *keynote*;
4. Estudos capacitistas em relação a surdez
5. Estudos que não estão em português ou inglês.

3.6 Resultados da Revisão de Literatura

Por meio da revisão da literatura, foram identificados diversos estudos vinculados ao tópico do ensino de indivíduos surdos. Essas pesquisas compreendem artigos científicos, publicações e teses, bem como dissertações que abordam uma ampla variedade de estratégias e abordagens para o processo de aprendizagem dos surdos e métodos aprimorados de instrução.

No método de ensino direcionado a esses estudantes, vários estudos exploram a visualidade e o potencial imagético dos tópicos e conteúdos a serem ensinados, como é evidenciado no contexto dos jogos pedagógicos. Outras abordagens relacionadas ao tema empregam tradutores de português para LIBRAS como meio de comunicação entre os professores e os alunos.

Nos artigos e pesquisas voltadas para a linguagem e aprendizado de alunos surdos, constatou-se uma ampla adoção de estratégias visuais, incluindo o uso extensivo de imagens

e a adaptação da informação de maneira altamente visual. Essa abordagem visa proporcionar uma compreensão mais acessível e eficaz, reconhecendo a importância da comunicação visual para os esses acadêmicos. Essa prática destaca a necessidade de métodos pedagógicos que atendam às especificidades da audiência surda, promovendo uma experiência de aprendizado mais inclusiva e efetiva.

Em algumas dissertações e artigos, observou-se a aplicação do *Shape Coding* como uma metodologia para aprimorar o ensino da gramática portuguesa. Enquanto a gramática normativa emprega a sintaxe para analisar e atribuir sentido às frases, garantindo o correto ordenamento das palavras dentro da oração, a linguagem de programação compartilha de um princípio semelhante. A sintaxe estruturada na programação contribui para tornar o código mais legível. Após a escrita, o programa-fonte é executado por um compilador, responsável por traduzir todas as instruções do programa. Essa tradução é comparável ao funcionamento do nosso cérebro, possibilitando, assim, a associação entre essas duas formas de linguagem.

4 *SHAPELAB*

Neste capítulo, são apresentadas as atividades realizadas ao longo deste trabalho, com foco na exploração da abordagem *SHAPELAB*. A proposta central consiste em demonstrar, por meio da construção de um protótipo ilustrativo, como é possível integrar a técnica de *Shape Coding* ao Desenvolvimento Orientado a Modelos, com o objetivo de apoiar o ensino da disciplina de Programação Estruturada para alunos com deficiência auditiva.

Assim, foi desenvolvido um metamodelo, utilizando a plataforma EMF e o editor gráfico com o Eclipse Sirius, usado como prova de conceito para demonstrar a viabilidade dessa integração. Nos tópicos a seguir, serão detalhados os processos de modelagem, as decisões técnicas adotadas, os elementos do metamodelo proposto e resultados no editor gráfico.

4.1 Adaptação do *Shape Coding* para o ensino de programação

Nesta seção, abordaremos os métodos usados na adaptação do *shape coding*, visando responder o 3º questionamento de pesquisa, mencionado na seção 3.2.

Q3. - Como o uso do *Shape Coding* pode trazer alguma melhora para o ensino da programação?

Para isso, foram feitas adaptações de cores e formas, as quais representam aspectos da linguagem. Em seguida, foi feita a aplicação prática do sistema, ilustrando o seu uso para representar um algoritmo. Para explorar o potencial imagético de uma linguagem, utilizaram-se cores para diferenciar aspectos específicos da linguagem, permitindo que os alunos façam associações mais rápidas.

O Quadro 2 mostra como o uso de cores e formas auxiliou esse processo. Ao associar a cor laranja ou a forma circular a uma operação, por exemplo, o aluno adquire uma compreensão aprimorada do funcionamento dessas estruturas. Isso proporciona ao professor uma melhor capacidade de compreender, explicar e corrigir possíveis erros do aluno. Esse recurso facilita a comunicação entre professor e aluno, mesmo que utilizem idiomas diferentes, como Português e LIBRAS.

Quadro 2: Cores, Formas e Significados do *Shape Coding*

Cor	Forma	Significado
Rosa		Program
Azul		Declare
Amarelo		Assign
Roxo		Input
Verde		Output
Laranja		Operation

Fonte: Autoria Própria

Reiterando o que foi dito na introdução, a utilização de recursos visuais no ensino para surdos é fundamental, pois ajuda na interpretação e na atribuição de significado, uma vez que eles “aprendem a aprender” dessa maneira, o uso de formas é aliado nesta forma de aprendizagem.

4.2 Metamodelos Criados

Para ilustrar a adaptação da abordagem para uma linguagem de programação, foram elaborados dois metamodelos: um com maior nível de abstração, como mostrado na Figura 7, e outro com foco em estruturas mais explícitas, como mostrado na Figura 10.

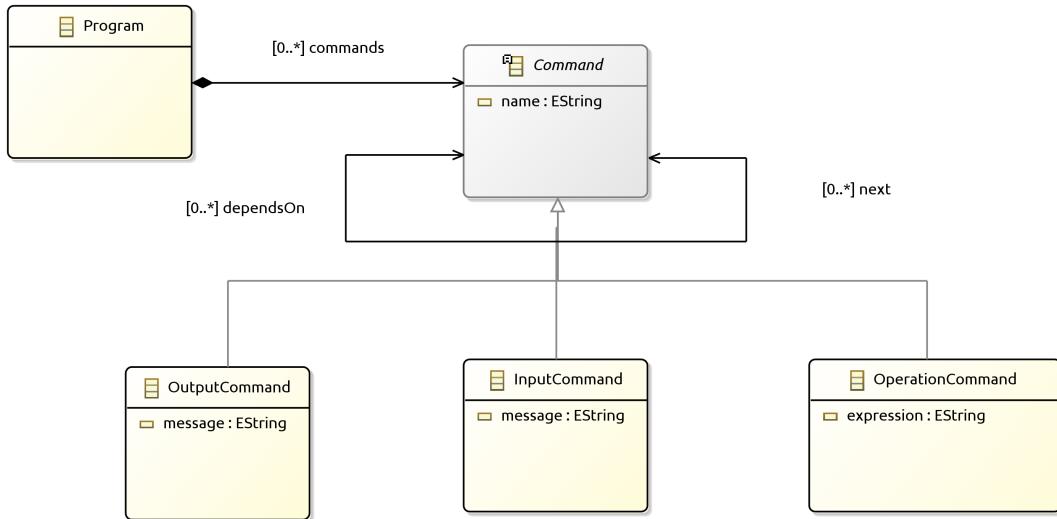
Tendo em mente que os metamodelos são a especificação de um modelo, ou seja, um metamodelo conduz como um modelo deve ser expresso (Velasco; Carvalho, 2022), temos a definição dos metamodelos.

4.2.1 Metamodelo A

Conforme apresentado no referencial teórico, podemos descrever uma linguagem por meio de um modelo, que define os elementos válidos e as regras de composição dessa linguagem. Seguindo esse princípio, o Metamodelo representado na Figura 7 foi projetado para representar, com maior nível de abstração, os elementos fundamentais de um programa simples, voltado ao ensino de programação.

A classe central é *Program*, que atua como contêiner do conjunto de comandos a serem executados. Por meio da relação de composição com *Commands*, um programa pode conter diversos comandos de diferentes tipos, que serão interpretados na sequência definida.

Figura 7 – Diagrama Metamodelo A da Linguagem



Fonte: Autoria própria

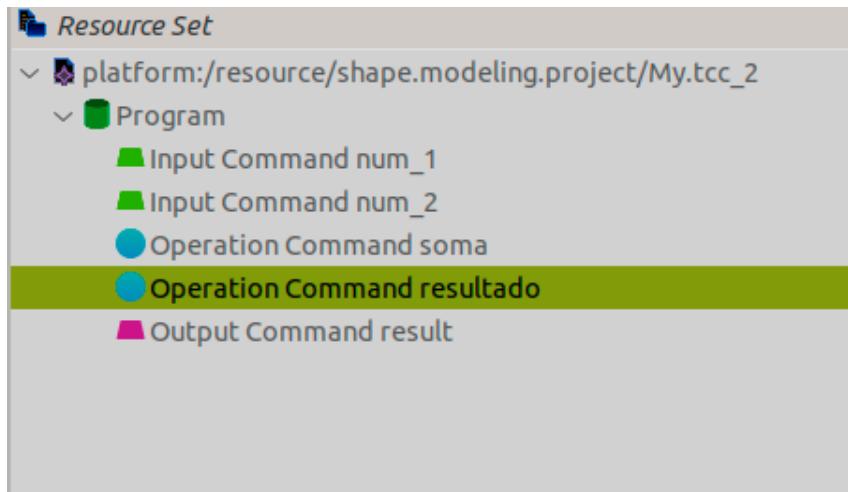
A classe *Command* é utilizada como superclasse para todos os comandos específicos. Ela generaliza comportamentos comuns, como o atributo *name*, que é utilizado para identificar e rotular o comando, além dos relacionamentos *dependsOn* e *next*, que tornam possível a definição de dependências lógicas e da sequência de execução, respectivamente.

Dessa forma, mesmo sendo um metamodelo abstrato, ele permite a representação de fluxos e relações entre as instruções. A partir dessa classe base, derivam três subclasses: *InputCommand*, *OutputCommand* e *OperationCommand*.

A classe *InputCommand* representa comandos de entrada de dados e contém o atributo *message*, que define a mensagem exibida ao solicitar a entrada. Já a classe *OutputCommand*, *OutputCommand*, modela comandos de saída, usados para exibir mensagens ou resultados ao usuário, também possui o atributo *message* como conteúdo de saída. Por fim, a classe *OperationCommand* representa comandos responsáveis por realizar operações lógicas ou matemáticas, armazenando a expressão a ser avaliada por meio do atributo *expression*.

4.2.1.1 Representação no editor gráfico

A Figura 8 apresenta a estrutura hierárquica de um modelo construída no editor do EMF, conforme o metamodelo de linguagem. O exemplo representa um programa simples, que realiza a soma de dois números. Os elementos do metamodelo foram associados às representações de cores, formas e significados do *Shape Coding*, conforme o Quadro 2.

Figura 8 – Representação do Algoritmo utilizando o *Sirius*

Fonte: Autoria própria

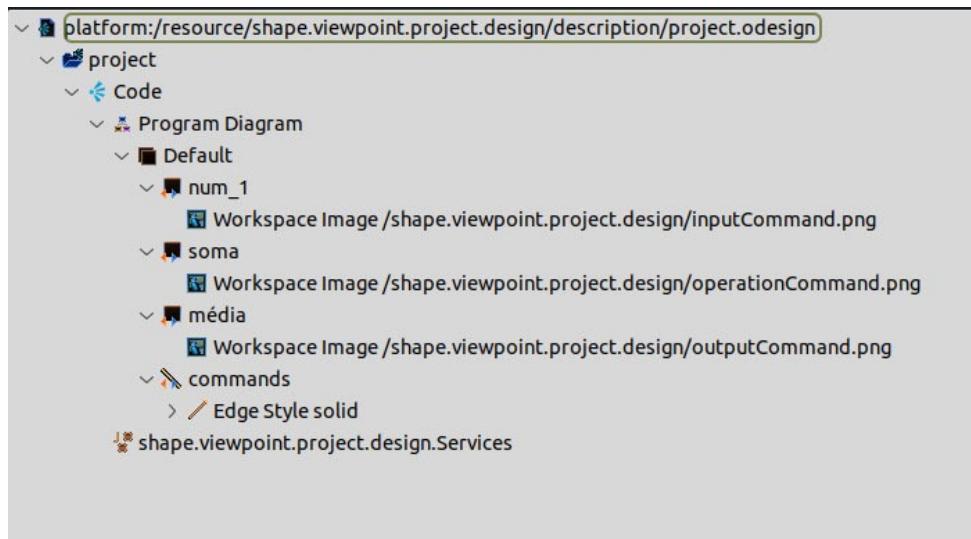
Por meio da árvore de elementos, é possível observar a sequência dos comandos modelados: os elementos contidos em *input* são responsáveis por receber os valores de entrada; soma e resultado realizam os cálculos intermediários; e, por fim, *result* é encarregado de exibir a saída do programa. Esse editor em árvore, gerado pelo EMF, permite ao usuário visualizar, organizar e editar os elementos do modelo de forma sequencial e hierárquica, definindo com clareza a lógica do programa.

4.2.1.2 Editor Gráfico Metamodelo A

Com o uso do Sirius, é possível gerar um editor gráfico personalizado com base no metamodelo definido, o que permite uma visualização intuitiva da estrutura e do fluxo de execução do programa. Como resultado da configuração do editor gráfico, tem-se a visualização do diagrama na Figura 9 presente no ambiente *.odesign*, o qual é aberto por meio do painel de representações fornecido pelo *Sirius*.

Esse ambiente permite a criação visual dos elementos definidos no metamodelo, refletindo a lógica construída previamente no editor do EMF. Nessa estrutura, foram definidas as regras para a apresentação visual a partir dos elementos estruturais da linguagem proposta pelos metamodelos. Desta forma, a partir do editor, poderemos desenhar de forma gráfica diversos programas que contenham os elementos definidos.

Figura 9 – Sirius Model



Fonte: Autoria própria

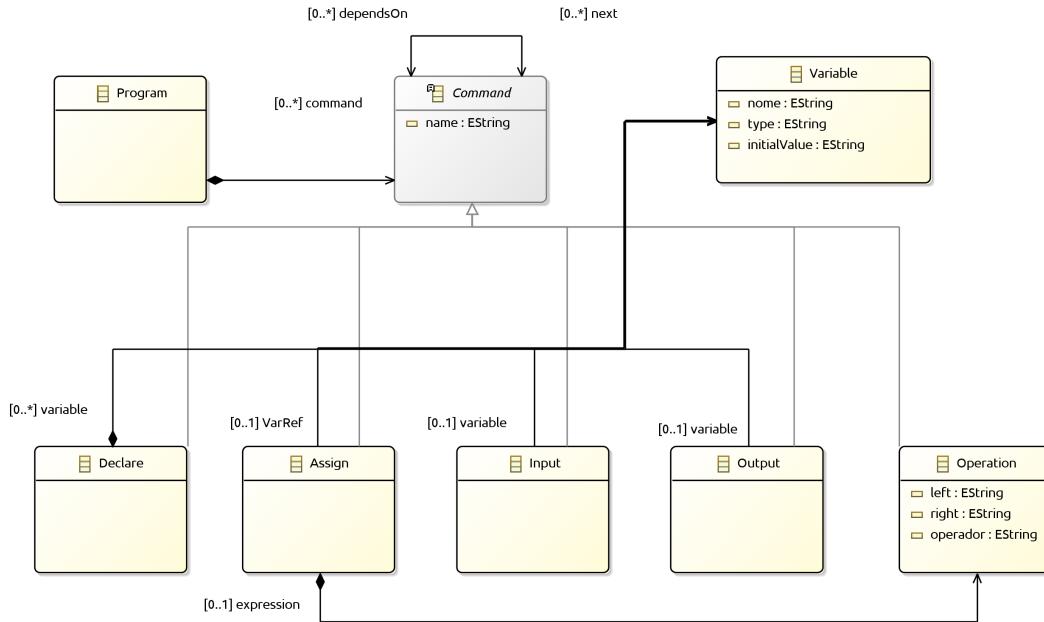
4.2.2 Metamodelo B

O metamodelo presente em Figura 10 evolui o Metamodelo A e apresenta uma representação mais completa da estrutura dos programas, incorporando explicitamente o uso de variáveis, operações, expressões e comandos fundamentais para o ensino de programação. A classe principal, *Program*, atua como contêiner do conjunto de comandos que compõem o fluxo de execução do programa.

A classe abstrata *Command* representa a base de todos os comandos possíveis e inclui o atributo *name*, que identifica cada comando individualmente. Essa classe define dois relacionamentos importantes, como já foi explicitado no metamodelo anterior da Figura 7. Os tipos específicos de comandos são derivados da classe *Command*, cada um com funcionalidades distintas.

A classe *Declare*, que representa a declaração de variáveis, por meio da associação *variable* representa a ação de declarar uma nova variável, com atributos de nome, tipo e valor inicial (definidos na classe *Variable*). Diferentemente de *Variable*, que corresponde à variável em si, como ilustrado no Quadro 3, a classe *Declare* representa o ato de declarar uma variável no código.

Figura 10 – Diagrama Metamodelo B da Linguagem



Fonte: Autoria própria

Quadro 3: Diferença entre Variable e Declare

Classe	Papel no Metamodelo
<i>Variable</i>	Representa um modelo de dado
<i>Declare</i>	Representa a ação declarar a variável no código

Fonte: Autoria própria

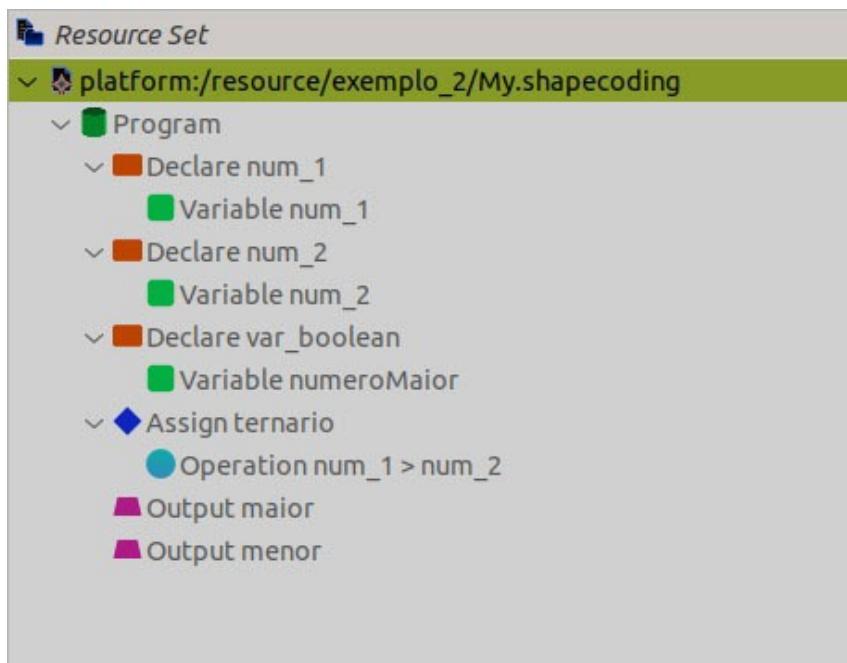
O componente *Input* representa comandos de entrada de dados, associados a uma variável que receberá o valor fornecido. E a classe *Output* representa comandos de saída, associados a uma variável cujo valor será exibido. A classe *Assign* modela a atribuição de valores, relacionando-se a uma variável e a uma expressão definida por meio da classe *Operation*, que representa operações lógicas ou matemáticas. Esta última possui os atributos *left*, *right* e *operator*, responsáveis por definir os operandos e o tipo de operação a ser realizada. Complementarmente, a relação *VarRef* permite referenciar variáveis previamente declaradas no programa.

Esse metamodelo amplia significativamente a capacidade de modelagem de programas, tornando possível representar com mais precisão os aspectos estruturais e semânticos de um código em linguagem imperativa, além de favorecer sua aplicação pedagógica por meio da associação visual entre comandos e variáveis.

4.2.2.1 Editor Gráfico Metamodelo B

A Figura 11 apresenta a árvore de elementos criada no ambiente .aird, em que os elementos foram organizados para representar um programa que compara dois números e identifica o maior entre eles. Por meio da árvore de elementos, é possível observar a sequência dos comandos modelados: os elementos contidos em *input* são responsáveis por receber os valores de entrada; soma e resultado realizam os cálculos intermediários; e, por fim, *result* é encarregado de exibir a saída do programa.

Figura 11 – Representação do Algoritmo com *Shape Coding*



Fonte: Autoria própria

Nesse exemplo, são declaradas três variáveis, *num_1* e *num_2*, ambas do tipo numérico, e *numeroMaior*, do tipo *booleano*. Em seguida, um comando *Assign*, nomeado como ternário, utiliza a operação lógica *num_1 > num_2* para realizar a comparação entre os valores de entrada. O resultado da operação é atribuído à variável *numeroMaior*, sendo posteriormente exibido pelos comandos *Output maior* e *Output menor*, conforme a lógica definida no modelo.

A Figura 12 mostra as propriedades da instância da classe *Operation* associada ao comando *Assign*. Nela, observa-se que o campo *Left* refere-se ao primeiro operando (*num_1*), enquanto *Right* representa o segundo operando (*num_2*). O campo *Operator* contém o símbolo lógico utilizado na operação ($>$), indicando que se trata de uma comparação.

Figura 12 – Representação da Propriedade de *Operation*

Property	Value
Depends On	
Left	
Name	num_1
Next	num_1 > num_2
Operador	
Right	

Fonte: Autoria própria

Já o campo *Name* identifica a operação (*num_1 > num_2*), sendo útil para referência visual no modelo. Além disso, as propriedades *Depends On* e *Next* podem ser utilizadas para definir o fluxo de execução e dependência entre comandos, embora estejam vazias neste caso.

Figura 13 – Representação da Propriedade de *Output*

Property	Value
Depends On	Assign ternario
Name	maior
Next	
Variable	Variable numeroMaior

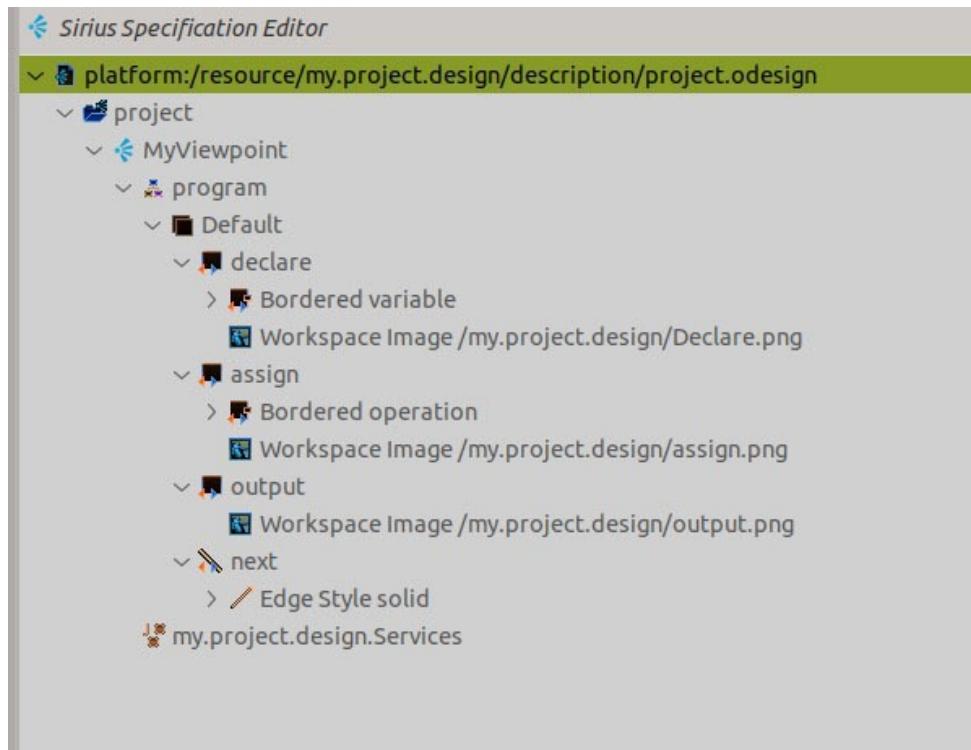
Fonte: Autoria própria

Já a Figura 13 apresenta as propriedades do comando *Output*, nomeado como *maior*, cuja função é exibir o valor da variável *numeroMaior* como saída. Essa instância está relacionada diretamente ao comando *Assign ternário* por meio da propriedade *dependsOn*, garantindo que a saída só será executada após a atribuição do valor lógico.

Esse tipo de visualização gráfica permite não apenas organizar os elementos do programa de forma clara e sequencial, mas também reforçar a associação entre forma, cor e significado conforme proposto pela abordagem do *Shape Coding*, contribuindo para um processo de ensino mais acessível e visual.

4.2.2.2 Resultado do Editor Gráfico

A Figura 14 ilustra a evolução do diagrama gerado no ambiente *.odesign*, acessado por meio do painel de representações definido no *Sirius*. Esse ambiente gráfico reflete diretamente a lógica estabelecida no editor EMF, permitindo a construção visual dos elementos do programa conforme as estruturas modeladas. As regras de representação visual foram definidas a partir dos componentes da linguagem especificados nos metamodelos, possibilitando ao usuário desenhar graficamente algoritmos compostos pelos elementos propostos, de maneira didática e acessível.

Figura 14 – *Sirius Model*

Fonte: Autoria própria

5 RESULTADOS

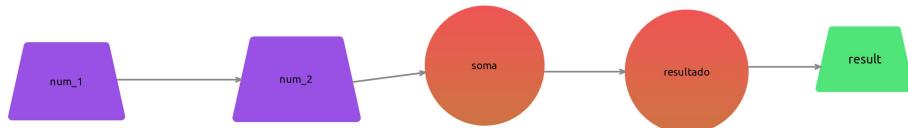
Esta seção apresenta os principais resultados obtidos a partir dos metamodelos desenvolvidos e da prova de conceito implementada, com foco na aplicação dos princípios do *Shape Coding* para o ensino de programação a estudantes surdos.

5.1 Resultado do Metamodelo A

A Figura 15 apresenta a visualização gráfica de um algoritmo simples estruturado a partir do Metamodelo A, modelado previamente na árvore hierárquica representada pela Figura 9. A representação foi gerada conforme as regras de exibição definidas no arquivo `.odesign`, utilizando o editor gráfico da ferramenta *Sirius*.

Essa representação gráfica reafirma a capacidade do *Sirius* de transformar estruturas modeladas no EMF em visualizações didáticas e acessíveis. A conexão entre os elementos é possível graças aos relacionamentos *next* e *dependsOn*, definidos na superclasse *Command* e herdados por suas subclasses no metamodelo proposto.

Figura 15 – Representação do Algoritmo SOMA com *Shape Coding*



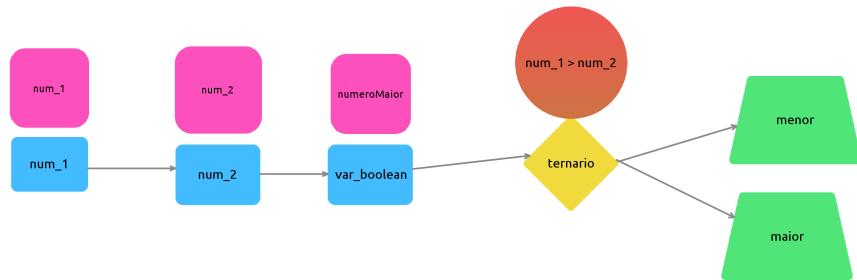
Fonte: Autoria própria

5.2 Resultado do Metamodelo B

A Figura 16 apresenta a representação visual de um algoritmo construído com base no Metamodelo B, conforme a estrutura definida na Figura 11. Essa visualização foi gerada a partir das regras estabelecidas no arquivo `.odesign` (Figura 14), também utilizando o editor gráfico do *Sirius*.

Assim como no metamodelo anterior, os relacionamentos *next* e *dependsOn* permitem representar, de forma visual e sequencial, o fluxo de execução entre os comandos. Isso contribui para uma melhor organização e entendimento da lógica do programa, especialmente em um contexto educacional voltado para alunos surdos.

Figura 16 – Representação do Algoritmo “MaiorQue” com *Shape Coding*



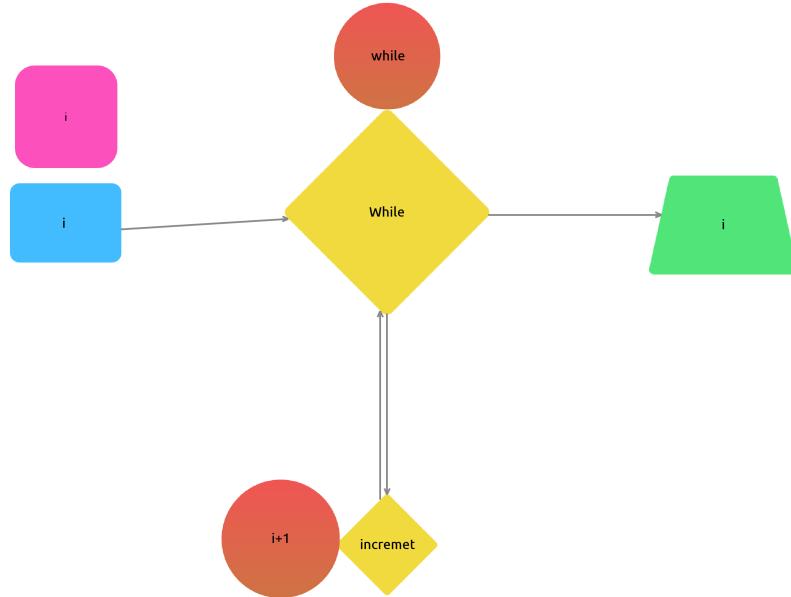
Fonte: Autoria própria

5.3 Prova de Conceito

Para validar a viabilidade da abordagem proposta, foi desenvolvida uma prova de conceito com base no Metamodelo B (Figura 10), utilizando o ambiente gráfico do *Sirius*. O objetivo foi avaliar se a combinação entre formas, cores e estruturas visuais, conforme os princípios do *Shape Coding*, favorece a aprendizagem de conceitos fundamentais da programação.

A Figura 17 apresenta um exemplo de algoritmo que utiliza uma estrutura de repetição do tipo *while*, permitindo testar a representação gráfica de comandos mais complexos.

Figura 17 – Algoritmo utilizando laço while



Fonte: Autoria própria

A modelagem evidenciou que a proposta baseada no *Shape Coding* é capaz de representar algoritmos simples de maneira clara, didática e compatível com os objetivos pedagógicos do ensino de lógica de programação. A utilização de elementos visuais facilita a identificação e distinção entre diferentes tipos de comandos, promovendo uma experiência de aprendizagem mais acessível — especialmente para estudantes com deficiência auditiva, que tendem a se beneficiar de recursos visuais reforçados.

Adicionalmente, a construção do algoritmo por meio de modelagem gráfica permite ao estudante concentrar-se nos conceitos fundamentais da lógica computacional sem a complexidade inicial das linguagens textuais, tornando o processo de aprendizagem mais visual, progressivo e inclusivo.

6 CONCLUSÃO

Este capítulo apresenta a conclusão deste trabalho, detalhando os resultados alcançados durante o desenvolvimento. Além disso, são fornecidas sugestões para futuros trabalhos no projeto.

6.1 Resultados Obtidos

A implementação do *Shape Coding* no ensino de programação estruturada para alunos surdos resultou em uma abordagem visualmente acessível e coerente com as necessidades desse público.

A prova de conceito desenvolvida por meio do editor gráfico criado com as ferramentas EMF e Sirius demonstrou que é possível representar algoritmos simples utilizando formas, cores e estruturas visuais.

A partir dos metamodelos elaborados, foi possível construir visualmente algoritmos que incluíam entrada, processamento, saída, estrutura de repetição e operações lógicas.

A organização hierárquica dos comandos no ambiente gráfico contribuiu para uma compreensão mais clara da lógica computacional, eliminando a complexidade inicial da linguagem textual.

Dessa forma, os resultados confirmaram que o uso de elementos visuais facilita a assimilação dos conceitos de programação por parte de estudantes com deficiência auditiva, promovendo uma experiência de aprendizagem mais inclusiva, intuitiva e significativa.

A modelagem visual mostrou-se eficaz tanto para representar a estrutura de algoritmos quanto para permitir o acompanhamento sequencial do fluxo de execução, reforçando a aplicabilidade da metodologia proposta.

6.2 Trabalhos Futuros

Como continuidade deste trabalho, propõem-se as seguintes direções para pesquisas futuras:

- Desenvolver e validar a abordagem metodológica em ambiente real de sala de aula, com a participação de estudantes surdos e professores da área de computação, a fim de avaliar a usabilidade e a eficácia pedagógica do modelo proposto.

- Expandir o metamodelo para contemplar estruturas mais complexas, como condicionais encadeadas, vetores, funções e laços aninhados, ampliando assim a abrangência do ensino de programação com base em *Shape Coding*.
- Desenvolver extensões ou adaptações que permitam integrar a abordagem a plataformas de ensino online ou sistemas híbridos, respeitando diretrizes de acessibilidade digital.
- Investigar o impacto da utilização contínua da ferramenta no desempenho acadêmico dos alunos surdos ao longo de disciplinas introdutórias de programação, por meio de estudos longitudinais.
- Explorar a adaptação da metodologia para outras áreas da computação ou outras linguagens de programação, analisando a generalização do modelo proposto.

REFERÊNCIAS BIBLIOGRÁFICAS

BARBOSA, A. V. O papel da consciência (meta) linguística no ensino da língua portuguesa a alunos chineses. *Letras & Letras*, v. 31, n. 2, p. 111–127, 2015. Citado na página 25.

BRASIL. Regulamenta a lei 10.436, de 24 de abril de 2002, que dispõe sobre a língua brasileira de sinais - libras, e o artigo 18 da lei 10.098, de 19 de dezembro de 2000. *Diário Oficial da União de 23/12/2005] (p. 28, col. 2)*, 2005. Disponível em: <https://www.planalto.gov.br/ccivil_03/_ato2004-2006/2005/decreto/d5626.htm>. Citado na página 17.

BRASIL. Lei nº 14.191, de 3 de agosto de 2021. altera a lei nº 9.394, de 20 de dezembro de 1996 (lei de diretrizes e bases da educação nacional), para dispor sobre a modalidade de educação bilíngue de surdos. *Diário Oficial da União - Seção 1 - 04/08/2021, Página 1*, Brasília, DF, 2021. Disponível em: <http://www.planalto.gov.br/ccivil_03/_Ato2011-2014/2012/Lei/L12651.htm>. Citado na página 16.

BRASIL. *Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (Inep). Resumo Técnico: Censo Escolar da Educação Superior 2022*. Brasília, DF: Inep, 2022. Disponível em: <<https://www.gov.br/inep/pt-br/centrais-de-conteudo/acervo-linha-editorial/publicacoes-institucionais/estatisticas-e-indicadores-educacionais/censo-da-educacao-superior-2022-notas-estatisticas>>. Citado na página 16.

CAMPELLO, A. R. et al. Aspectos da visualidade na educação de surdos. Florianópolis, SC, 2008. Citado 3 vezes nas páginas 23, 24 e 25.

CAPOVILLA, F. C.; CAPOVILLA, A. G. Educação da criança surda: o bilingüismo e o desafio da descontinuidade entre a língua de sinais e a escrita alfabetica. *Revista Brasileira de Educação Especial*, v. 8, n. 02, p. 127–156, 2002. Citado 2 vezes nas páginas 22 e 24.

CAPOVILLA, F. C.; CAPOVILLA, A. G. *Leitura de estudantes surdos: desenvolvimento e peculiaridades em relação à de ouvintes*. [S.l.]: ETD, 2006. Citado na página 24.

CARVALHO, G. B. d. et al. Elaboração de um modelo conceitual de ferramenta de auxílio nas práticas de ensino-aprendizagem para estudantes surdos. Universidade Federal Rural do Semi-Árido, 2020. Citado 3 vezes nas páginas 16, 17 e 18.

COLLEGE, M. H. S. . *Sistema SHAPE CODING™ – Uma maneira visual de ensinar gramática falada e escrita*. 2023. <<https://shapecoding.com/>>. (Accessed on 10/18/2023). Citado na página 24.

COSTA, O. S. Uma ideia na mão e uma câmera na cabeça: cinema na educação bilíngue de surdos e surdas. Universidade Federal de São Carlos, 2020. Citado na página 17.

DECIAN, M. M.; MÉA, C. H. P. D. O signo lingüístico: de saussure a benveniste. *Disciplinarum Scientia| Artes, Letras e Comunicação*, v. 6, n. 1, p. 93–109, 2005. Citado na página 16.

ECLIPSE. *Eclipse Modeling Project | The Eclipse Foundation*. 2024. (Accessed on 04/07/2024). Disponível em: <<https://eclipse.dev/modeling/emf>>. Citado na página 28.

- ECLIPSESOURCE. *EMF Tutorial*. 2024. (Accessed on 04/07/2024). Disponível em: <<https://eclipsesource.com/blogs/tutorials/emf-tutorial>>. Citado na página 28.
- KLEPPE, A. G.; WARMER, J. B.; BAST, W. *MDA explained: the model driven architecture: practice and promise*. [S.l.]: Addison-Wesley Professional, 2003. Citado 2 vezes nas páginas 27 e 28.
- LEBEDEFF, T. B. Aprendendo a ler “com outros olhos”: relatos de oficinas de letramento visual com professores surdos. *Cadernos de Educação*, n. 36, 2010. Citado na página 24.
- LEBEDEFF, T. B. Práticas bilíngues em escolas de surdos pennsylvania school for the deaf e oak lodge school. *Revista Espaço*, p. 68–85, 2015. Citado na página 23.
- LESSA-DE-OLIVEIRA, A. S. C. *Libras escrita: o desafio de representar uma língua tridimensional por um sistema de escrita linear*. *ReVEL*, v. 10, n. 19, 2012. 2013. Citado na página 16.
- MEDEIROS, R. P. Hello, world: uma análise sobre dificuldades no ensino e na aprendizagem de introdução à programação nas universidades. Universidade Federal de Pernambuco, 2019. Citado na página 25.
- MORET, M. C. F. F.; MENDONÇA, J. G. R. O uso das tecnologias para acessibilidade da comunidade surda no processo educacional brasileiro. *Revista Educação Especial em Debate*, v. 8, n. 15, p. 180–193, 2023. Citado 3 vezes nas páginas 17, 18 e 25.
- NÓBREGA, Y. S. et al. Ensino da libras através de recurso pedagógico digital: Educalibras. Universidade Federal da Bahia, 2021. Citado na página 22.
- SÁ, O. M. de; CRUZ, S. da; PINHEIRO, V. D. S. Visualidade, língua de sinais e conhecimento prévio: Pilares no ensino para aprendizes surdos. *Communitas*, v. 4, n. 7, p. 312–326, 2020. Citado na página 25.
- SAMPAIO, C. C. d. R. d. *O uso do Shape Coding no ensino de Língua Portuguesa como segunda língua para Surdos: um estudo sobre variação temporal*. Dissertação (Mestrado) — Universidade Federal de Pelotas, 2018. Citado na página 23.
- SAUSSURE, F. D. *Curso de linguística geral*. [S.l.]: Editora Cultrix, 2008. Citado 2 vezes nas páginas 16 e 22.
- SILVA, D. N. *Língua Brasileira de Sinais (Libras) - Brasil Escola*. 2023. Disponível em : <<https://brasilescola.uol.com.br/educacao/lingua-brasileira-sinais-libras.htm>>. Acesso em: 15 out. 2023. Citado na página 22.
- SKLIAR, C. Perspectivas políticas e pedagógicas da educação bilíngüe para surdos. *SILVA, Shirley; VIZIM, Marli. Educação Especial: múltiplas leituras e diferentes significados*. Campinas: Mercado de Letras/ALB, p. 85–110, 2001. Citado na página 23.
- SOMMERRVILLE, I. *Software engineering, 9thed*. England: Education Limited, 2010. Citado na página 27.
- STEINBERG, D. et al. *EMF: eclipse modeling framework*. [S.l.]: Pearson Education, 2008. Citado 2 vezes nas páginas 28 e 29.

- VELASCO, G. C.; CARVALHO, S. T. Uma abordagem dirigida por modelo para desenvolvimento de contratos inteligentes na ethereum virtual machine. In: SBC. *Anais da X Escola Regional de Informática de Goiás*. [S.l.], 2022. p. 106–117. Citado na página 36.
- VIANA, J. M. *Adaptação do Shape Coding para o ensino de Língua Portuguesa para surdos do sexto ano do Ensino Fundamental*. Dissertação (Mestrado) — Universidade Federal de Pelotas, 2019. Citado 2 vezes nas páginas 16 e 24.
- VIANA, J. M.; LEBEDEFF, T. B. O sistema shape coding como ferramenta para a reflexão linguística em língua portuguesa para surdos. *Educação de surdos, linguagens e experiências* (1a ed., pp. 166-181). Uberlândia, MG: Editora Navegando, 2021. Citado na página 23.
- WOHLIN, C. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: *Proceedings of the 18th international conference on evaluation and assessment in software engineering*. [S.l.: s.n.], 2014. p. 1–10. Citado 2 vezes nas páginas 19 e 30.