

## CHATBOT COMO ESTRATÉGIA MULTIETÁRIA DE ENSINO

Submetido em: 19 jan. 2022. Aceito: 16 jan. 2024.

Débora Liliane de Souza Renó<sup>1</sup>  
Cátia dos Santos Espina<sup>2</sup>  
Ricardo Rodrigues dos Santos<sup>3</sup>  
William Nunes Ferreira<sup>4</sup>  
Lucas Santos Medina de Oliveira<sup>5</sup>  
Olivia Ines Alves Morales<sup>6</sup>

### RESUMO

A sensibilização de estudantes contemporâneos para o processo de aprendizagem esbarra na oferta abundante e acelerada de recursos tecnológicos. Este cenário obriga o ensino a ser atraente, dinâmico e interativo ao longo da edificação do conhecimento autônomo e significativo. Desta forma, este trabalho objetiva através da metodologia ativa “Aprendizagem Baseada em Projetos” conectar através do ensino da robótica educacional estudantes de diferentes níveis de escolaridade e propiciar a eles, através da confecção e execução de *chatbot* uma nova ferramenta de estudo autônoma, interativa e integrada a fatores socioemocionais. O *StudyChat* SESI-SP 113 foi desenvolvido por estudantes do 3º ano do Ensino Médio pelo método *Scrum* e é oferecido a estudantes do 5º ano do Ensino Fundamental como novo recurso de estudo multidisciplinar.

**Palavras-chave:** *Chatbot*. Aprendizagem Baseada em Projetos. Robótica Educacional. Cultura Digital.

### ABSTRACT

The awareness of contemporary students for the learning process comes up against the abundant and accelerated supply of technological resources. This scenario forces teaching to be attractive, dynamic and interactive throughout the construction of autonomous and meaningful knowledge. In this way, this work aims through the

---

<sup>1</sup>Doutora em Biosistemas pela Universidade Federal do ABC; Professora de Programação e Robótica da Rede SESI-SP; Mogi das Cruzes, São Paulo, Brasil. E-mail: deborareno@gmail.com

<sup>2</sup>Licenciada em Química pela Universidade de Mogi das Cruzes; Professora de Educação Básica III da Rede SESI-SP; Mogi das Cruzes, São Paulo, Brasil. E-mail: catia.espina@sesisp.org.br

<sup>3</sup>Bacharel em Engenharia Elétrica pela Universidade Cruzeiro do Sul; Orientador Educação Digital da Rede SESI-SP; Mogi das Cruzes, São Paulo, Brasil. E-mail: ricardo.santos@sesisp.org.br

<sup>4</sup>Estudante do 3º ano do Ensino Médio do Centro Educacional SESI-SP 113 “Milton Sobrosa Cordeiro”; Estudante da Rede SESI-SP; Mogi das Cruzes, São Paulo, Brasil. E-mail: william.nunes3@senaisp.edu.br

<sup>5</sup>Estudante do 3º ano do Ensino Médio do Centro Educacional SESI-SP 113 “Milton Sobrosa Cordeiro”; Estudante da Rede SESI-SP; Mogi das Cruzes, São Paulo, Brasil. E-mail: lucas.oliveira25@portalsesisp.org.br

<sup>6</sup>Estudante do 3º ano do Ensino Médio do Centro Educacional SESI-SP 113 “Milton Sobrosa Cordeiro”; Estudante da Rede SESI-SP; Mogi das Cruzes, São Paulo, Brasil. E-mail: olivia.morales@portalsesisp.org.br

active methodology "Project Based Learning" to connect through the teaching of educational robotics students of different levels of education and provide them, through the making and execution of chatbot, a new tool for autonomous study, interactive and integrated to socio-emotional factors. StudyChat SESI-SP 113 was developed by 3rd year high school students using the Scrum method and offered to 5th year elementary school students as a new multidisciplinary study resource.

**Keywords:** Chatbot. Project Based Learning. Educational Robotics. Digital Culture.

## 1 INTRODUÇÃO

A necessidade da padronização normativa para a elaboração de currículos nacionais de instituições de ensino públicas e privadas promotoras de ensino de nível infantil, fundamental, médio e superior existe desde a promulgação da constituição em 1988. Leis subsequentes à carta magna como Lei de Diretrizes e Bases da Educação Nacional em 1996, Lei dos Parâmetros Curriculares Nacionais do Ensino Fundamental em 1997, para séries iniciais e, em 1999, para séries finais e dos Parâmetros Curriculares Nacionais para o Ensino Médio em 2000 complementam a legislação específica, detalhando como deve ser o processo de ensino-aprendizagem nacional (BITTENCOURT, 2019).

A promulgação da Base Nacional Comum Curricular para a Educação Básica em 2018 encerra momentaneamente, este conjunto normativo, estabelecendo dez competências educacionais gerais desmembradas nas seguintes áreas de conhecimento: Linguagens e suas Tecnologias, Matemática e suas Tecnologias, Ciências da Natureza e suas Tecnologias e Ciências Humanas e Sociais Aplicadas (BRASIL, 2018).

A inovação estabelecida na BNCC constitui a promoção do desenvolvimento de competências e habilidades socioemocionais, além das curriculares convencionais estabelecidas no sistema educacional brasileiro (LISBOA & ROCHA, 2020; PALAVRAS PROJETOS EDITORIAIS, 2022). Este ambiente é propício ao estabelecimento do ensino da Robótica Educacional cuja metodologia educacional prioriza a aprendizagem significativa a partir da pesquisa, descoberta e construção de conhecimento por meio de atividades plugadas e desplugadas realizadas,

geralmente, em equipes (BRAUN, 2020).

O ensino do pensamento computacional requer estratégias diversificadas e efetivas para promover a construção do conhecimento autônomo. Dessa forma, o uso de metodologias ativas como a Aprendizagem Baseada em Projetos torna-se uma ferramenta comum e eficiente ao ensino da Robótica Educacional (IBARRA & SOARES, 2022).

Criado para facilitar a aprendizagem, a partir das metodologias ativas, e auxiliar o professor dentro da sala de aula, o *StudyChat* SESI-SP 113 é uma nova ferramenta de estudo autônomo, que permite ao estudante realizar suas atividades sem necessitar do auxílio simultâneo de um mentor, dentro da plataforma. Ele foi desenvolvido por estudantes do 3º ano do Ensino Médio para ter aplicabilidade no estudo autônomo de estudantes do 5º ano do Ensino Fundamental.

O aplicativo permite que o professor crie tarefas baseadas nas expectativas de ensino e aprendizagem determinadas para cada série no ensino da Robótica Educacional da Rede SESI-SP de Educação (SESI-SP, 2021) e as disponibilize para os estudantes os quais podem remotamente acessar esta atividade e executá-la. De forma interativa, o chat funciona como uma conversa na qual é enviada a pergunta e as alternativas de resposta para os estudantes. Ao escolherem uma das alternativas, o software analisa os resultados de forma ágil e retorna sua assertividade, fazendo com que as dificuldades e pontos fortes de cada estudante, ou da turma no geral, sejam atendidas e observadas, tornando as aulas mais eficazes, assim como o processo de avaliação, que podem facilmente ser adequados às necessidades de cada estudante.

## 2 METODOLOGIA

A interação entre docentes do componente curricular Programação e Robótica do Centro Educacional 113 “Milton Cordeiro Sobrosa” da Rede SESI-SP proporcionou a elaboração do projeto intitulado “CHATBOT COMO ESTRATÉGIA MULTIETÁRIA DE ENSINO” a partir da metodologia ativa Aprendizagem Baseada em Projetos (IBARRA & SOARES, 2022).

A criação de questionário conteudista direcionado ao processo de aprendizagem de estudantes do 5º ano do Ensino Fundamental pela docente responsável forneceu aos estudantes do 3º ano do Ensino Médio material para a criação e desenvolvimento de aplicativo baseado em inteligência artificial, *StudyChat* SESI-SP 113, sob orientação dos docentes envolvidos (DORF, 1998; AFARI, 2017).

O desenvolvimento do *Front-End* iniciou-se na criação do *layout* por meio da ferramenta Figma, plataforma colaborativa para realizar a construção de *layout* de aplicações. Posteriormente, a codificação foi desenvolvida em React.js, biblioteca utilizada para desenvolvimento de interfaces de usuário. Dentre outros apoios utilizados estão, jQuery, biblioteca JavaScript que facilita a manipulação do HTML e de eventos, Axios, cliente HTTP que facilita o envio de requisições e tratamento da resposta e *React-simple-Chatbot*, biblioteca responsável pela geração de um componente de chat.

O *Back-End* foi desenvolvido na Linguagem C# com o Framework .NET tendo como base a arquitetura de software API RESTful, caracterizando-se por uma API, *Application Programming Interface*, que segue os padrões impostos pelo estilo de arquitetura REST, *Representational State Transfer*, o qual impõe restrições arquitetônicas que, ao serem implementadas, colocam em evidência o potencial de crescimento de interações de componentes, generalização de interfaces, implementação independente de componentes intermediários para reduzir a latência de interação, reforçar a segurança e encapsular sistemas antigos, também chamados de Sistema Legado (SAUDATE, 2013). Também foi utilizado o *Clean Code*, um conjunto de regras que organizam o código de maneira limpa, mantendo um alto nível de manutenibilidade, além das regras citadas anteriormente (BALTIERI, 2022). Em sua importância evidencia o uso dos cinco princípios de design de código POO, Programação Orientada a Objetos, denominado pela sigla S.O.L.I.D, a qual é composta por *Single Responsibility*, princípio que diz que uma classe deve ter apenas uma responsabilidade, *Open-Close*, que propõe a ideia de que uma classe está aberta para extensão, mas fechada para modificação (THELMA, 2022).

Liskov Substitution diz que se determinada classe é um subtipo de outra, objetos

da classe subtipo que podem substituir a classe principal sem alterar nenhuma de suas propriedades, *Interface Segregation*, uma classe não deve ser forçada a depender de métodos que ela não utiliza, *Dependency Inversion*, classes de alto nível não devem depender de classes de baixo nível, ambos devem derivar da abstração de maneira que determinada funcionalidade da classe de baixo nível depende da mesma (AL-AHMAD, 2006). E, de modo a acrescentar uma camada de segurança além dos itens descritos anteriormente, foi implementado o JWT, JSON Web Token, que fornece um modo compacto e independente de transmitir informações por meio de um objeto JSON, trazendo confiabilidade já que o Token possui uma assinatura digital que utiliza como algoritmo o HMAC, Código de autenticação baseado em Hash, que evita que as informações sejam adulteradas (BALTIERI, 2022).

Já em relação à persistência de dados, utilizamos o banco de dados relacional SQL Server hospedado na Azure, e durante sua criação foi necessário, além de definir a regra de negócio, passar por etapas de criação para um banco normalizado (MICROSOFT, 2022). Como primeira etapa, foi necessário dar início à modelagem do banco com a criação de um MER, Modelo de Entidade e Relacionamento, e posteriormente DER, Diagrama de Entidade Relacionamento. A partir de então, foi possível progredir para criação dos *scripts* iniciando pelo DDL, *Data Definition Language*, um conjunto de comandos responsável pela criação e manipulação de tabelas no banco, seguido por DML, *Data Manipulation Language* conjunto de comandos responsáveis pelo acesso e manipulação dos dados contidos em uma tabela, e DQL, *Data Query Language* tendo como objetivo a consulta dos dados (DATE, 2004).

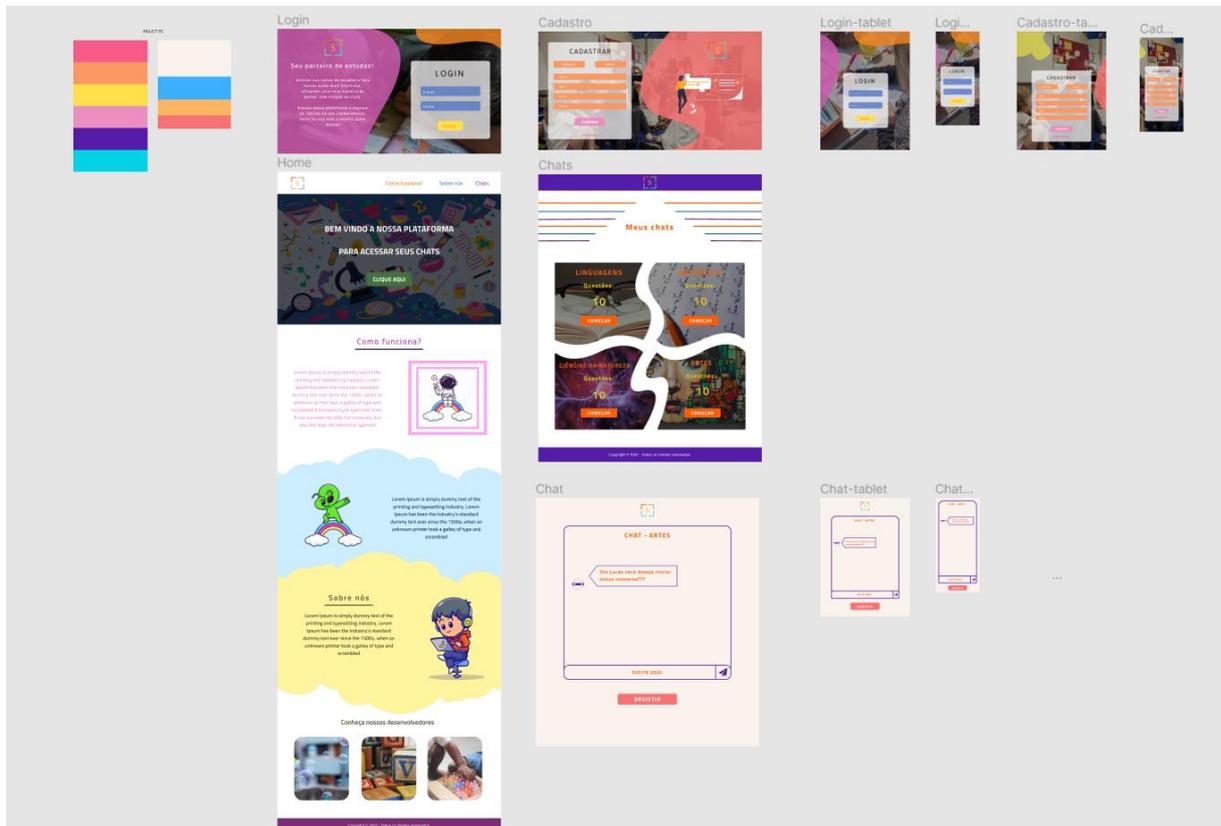
O teste do aplicativo *StudyChat* SESI-SP 113 foi realizado por meio de uma aula experimental que os desenvolvedores apresentaram aos estudantes do 5º ano do Ensino Fundamental sob supervisão dos professores orientadores.

### 3 RESULTADOS E DISCUSSÃO

A criação do layout da plataforma a partir da ferramenta Figma tornou a ideia de *design* do *StudyChat* uma realidade mais centralizada. Com ela foi possível diagnosticar a responsividade, bem como a disposição dos elementos de cena,

criando um *design* de produto mais aprimorado e chamativo. Simuladores de conversação utilizam desta mesma ferramenta estratégica de estilização atraente a seus usuários.

Figura 1 – Modelagem do sistema e sua responsividade.



Com o esquemático do *StudyChat*, paralelamente foi possível diagnosticar a elaboração das bibliotecas do *Front-End*, utilizando parte dos componentes da biblioteca React, conforme Figura 2, implementação possível a partir da Orientação a Objetos e, também pela chamada dos serviços de API, corroborando com a metodologia adotada por serviços de atendimento ao cliente e serviços de mensagens instantâneas baseadas na nuvem (GUIGIDI & MATTOS, 2018; ALVES, 2021).

Figura 2 – Importação de Serviços e APIs.

```
import React, { Component } from 'react';
import PropTypes from 'prop-types';
import ChatBot from 'react-simple-chatbot';
import css from './teste.css'
import logo from '../assets/logo.png'
import api from '../services/api'
```

Idealizando melhor comunicação entre os setores da aplicação, utilizou-se a biblioteca Axios, responsável pelo fluxo de informações de estudantes e professores entre Front-End e API na plataforma *StudyChat*. No contexto da programação, Figura 3, parametrizou-se a utilização do método 'setState', assumindo que, caso falte alguma informação no cadastro, o sistema automaticamente detecta e solicita a correção.

Figura 3 – Biblioteca Axios consumindo o back-end

```
cadastrarAluno = (event) => {
  event.preventDefault();

  this.setState({ erroMensagem: "", isLoading: true })

  api.post('/api/Estudantes', {
    nome: this.state.nome,
    idSerie: this.state.idSerie,
    idUsuarioNavigation: {
      email: this.state.email,
      senha: this.state.senha,
    }
  })

  .then(resposta => {
    if (resposta.status === 201) {
      this.setState({ isLoading: false });
      this.setState({ erroMensagem: "Cadastrado com sucesso" });

      this.props.history.push("/")
    }
  })
  .catch((error) => {
    this.setState({ erroMensagem: error.response.data, isLoading: false });
  })
}
```

Para aumentar o nível de segurança e credibilidade do aplicativo, foi adotado o JSON Web Token, conhecido como JWT. Essa estratégia possibilita a geração de códigos personalizados de autenticação, permitindo que tanto o cliente quanto o servidor tenham seu acesso e credibilidade reconhecidos. Isso contribui para anular a entrada não autorizada de possíveis ataques. Na Figura 4, é possível observar a composição do JWT em três partes, conforme descrito por Saudate (2013), assegurando que o aplicativo e suas autorizações de entrada sejam protegidos.

Figura 4 – JWT contendo o payload do *StudyChat*.

The image shows a web-based JWT decoder interface. On the left, under the 'Encoded' tab, a long alphanumeric string is displayed. On the right, under the 'Decoded' tab, the token's structure is shown in three sections: 'HEADER: ALGORITHM & TOKEN TYPE', 'PAYLOAD: DATA', and 'VERIFY SIGNATURE'. The payload data is a JSON object containing user information and token metadata.

```

Encoded
PASTE A TOKEN HERE

Decoded
EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE
{
  "alg": "HS256",
  "typ": "JWT",
  "cty": "JWT"
}

PAYLOAD: DATA
{
  "email": "a@a",
  "jti": "5",
  "http://schemas.microsoft.com/ws/2008/06/identity/claims/role": "Z",
  "exp": 1664859751,
  "iss": "Robotica.webAPI",
  "aud": "Robotica.webAPI"
}

VERIFY SIGNATURE
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  your-256-bit-secret
)  secret base64 encoded
  
```

Em conjunto com JWT, foi também essencial a utilização da API Rest (FIELDING, 2000), responsável pelo tratamento de dados da aplicação, perfazendo com que os dados sigam o fluxo do Front-End ao Back-End. Com isso é possível que cliente e servidor sejam conectados ao banco de dados principal. Dessa forma, nas Figuras 5 e 6 podemos observar a codificação da linguagem, que determina a centralização e integração das escolhas de questionários. Na Figura 6, é possível observar o resultado final na parte de interação.

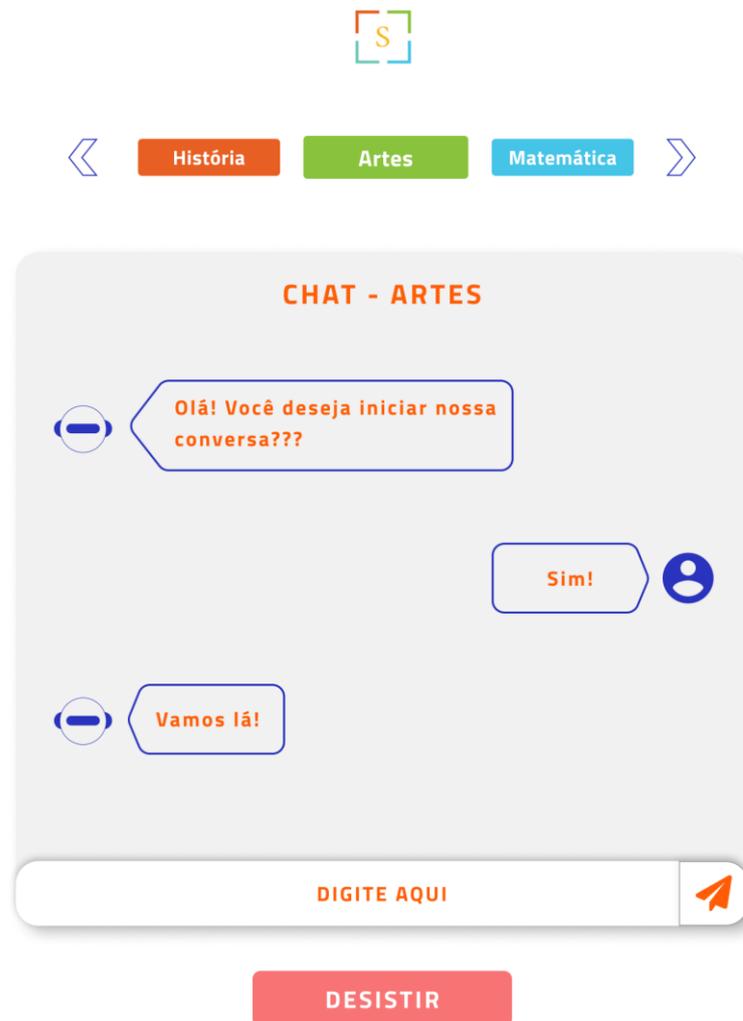
Figura 5 – API Rest contendo as opções de escolha.

```

1  using Microsoft.AspNetCore.Mvc;
2  using System;
3  using System.Collections.Generic;
4  using System.IdentityModel.Tokens.Jwt;
5  using System.Linq;
6  using System.Threading.Tasks;
7  using WebApi_Robotica.Domains;
8  using WebApi_Robotica.Interfaces;
9
10 namespace WebApi_Robotica.Controllers
11 {
12     [Produces("application/json")]
13
14     [Route("api/{controller}")]
15     [ApiController]
16     public class QuestionariosController : Controller
17     {
18         private readonly IQuestionarioRepository _questionarioRepository;
19         private readonly IEstudanteRepository _estudanteRepository;
20
21         public QuestionariosController(IQuestionarioRepository contexto, IEstudanteRepository ctx)
22         {
23             _questionarioRepository = contexto;
24             _estudanteRepository = ctx;
25         }
26
27         [HttpGet]
28         public IActionResult Listar()
29         {
30             try
31             {
32                 int idUsuario = Convert.ToInt32(HttpContext.User.Claims.First(c => c.Type == JwtRegisteredClaimNames.Jti).Value);
33                 Estudante e = _estudanteRepository.BuscarPorIdUser(idUsuario);
34
35                 return Ok(_questionarioRepository.Listar(e.IdSerie));
36             }
37             catch (Exception ex)
38             {
39                 return StatusCode(500, ex.InnerException.Message);
40             }
41         }
42
43         [HttpPost]
44         public IActionResult CriarQuestionario(Questionario q)
45         {
46             try
47             {
48                 _questionarioRepository.Cadastrar(q);
49                 return StatusCode(200, "Questionario criado com sucesso");
50             }
51             catch (Exception ex)
52             {
53                 return BadRequest(ex);
54             }
55         }
56
57         [HttpDelete("{Id}")]
58         public IActionResult Delete(int Id)
59         {
60             try
61             {
62                 _questionarioRepository.Deletar(Id);
63                 return StatusCode(200, "Questionario apagado com sucesso");
64             }
65             catch (Exception ex)
66             {
67                 return StatusCode(500, ex.InnerException.Message);
68             }
69         }
70
71         [HttpPut("{Id}")]
72         public IActionResult Put(Questionario QuestionarioAtualizado, int Id)
73         {
74             try
75             {
76                 _questionarioRepository.Atualizar(QuestionarioAtualizado, Id);
77                 return StatusCode(200, "Alterado com sucesso");
78             }
79             catch (Exception Erro)
80             {
81                 return BadRequest(Erro);
82             }
83         }
84     }
85 }

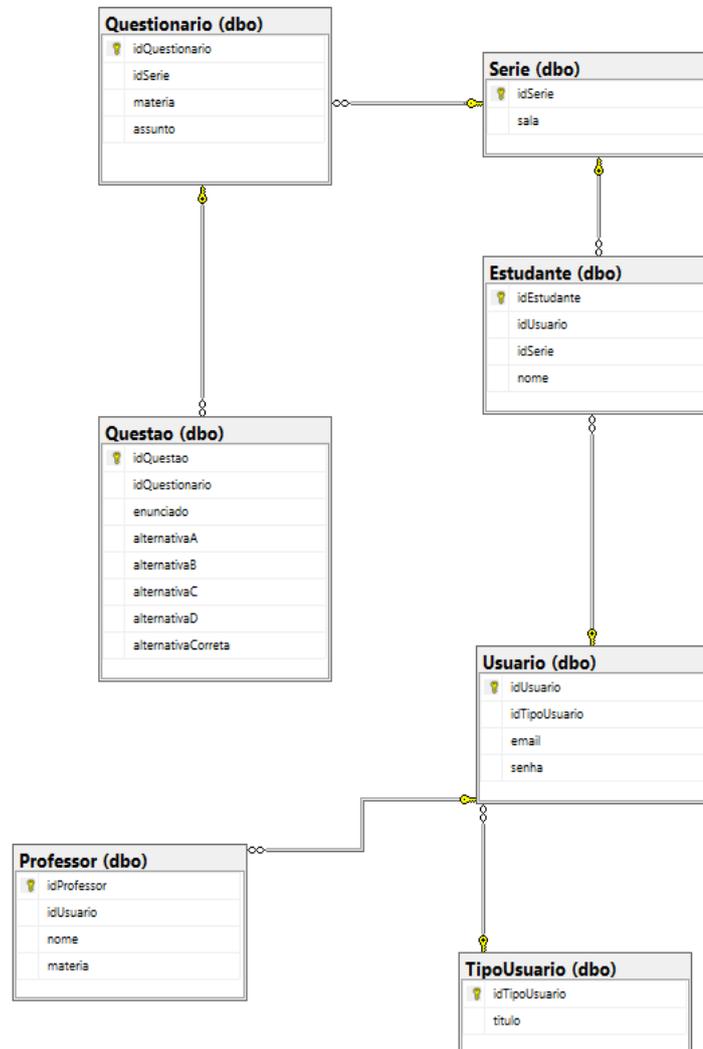
```

Figura 6 – Interação com o usuário *StudyChat*.



Para que a implantação do aplicativo respeite as regras de negócio solicitadas a partir da proposta acadêmica, criada por professores e estudantes, analisou-se a criação de integridades relacionais, utilizando Banco de Dados (Figura 7), conforme DATE (2004).

Figura 7 – Relacionamento das Entidades Relacionais - Banco de Dados.



Após uma análise experimental com estudantes do 5º ano do ensino fundamental, é possível visualizar um aspecto positivo do processo. Inicialmente foi aplicada uma roda de conversa explicativa, na qual foi possível diagnosticar o trabalho com o *chat*, mostrando também um pouco sobre o universo da programação, citando cada uma das etapas e suas funções. Com a execução da ferramenta (Figura 8) foi possível mostrar a forma com que o *StudyChat* SESI-SP 113 se apresenta, tornando as aulas mais dinâmicas e interessantes, eficiência similar à obtida em *chatbots* direcionados à aprendizagem de idiomas ou de habilidades socioeducacionais (DA SILVEIRA, 2019; SILVA et al, 2020).

Figura 8 - Tela de interação com o usuário



Esperamos que *StudyChat* seja uma plataforma de auxílio ao professor, criando um ecossistema que facilite avaliações, bem como a otimização do tempo, ponto forte que destaca a qualidade em estudos futuros.

#### 4 CONSIDERAÇÕES FINAIS

O incentivo da aprendizagem autônoma e significativa por meio da aplicação de metodologias ativas, como Aprendizagem Baseada em Projetos, torna-se promissor à medida que favorece o protagonismo natural dos estudantes envolvidos na construção do conhecimento, estabelecendo, além das habilidades curriculares, competências socioemocionais.

A interação entre estudantes de diferentes idades proporcionou uma vivência ímpar no processo de ensino e aprendizagem notabilizada na ampliação de vocabulário relacionado a linguagem de programação, no posicionamento questionador e argumentador dos estudantes envolvidos e na prática da programação responsável pela criação de um produto real e aplicável – o *StudyChat* SESI-SP 113.

A cooperação ética entre profissionais da educação é uma estratégia promissora e eficiente para a criação de novas ferramentas e metodologias de ensino como evidencia nosso projeto.

## REFERÊNCIAS

AL-AHMAD, Walid. **A framework for conceptual modeling in OOP**. [S. l.], 21 fev. 2006. Disponível em: <https://doi.org/10.1016/j.jfranklin.2006.02.035>. Acesso em: 26 set. 2022.

AFARI, E. & KHINE, M. S., 2017. **Robotics as an Educational Tool: Impact of Lego Mindstorms**. Disponível em: <http://www.ijiet.org/vol7/908-T108.pdf>. Acesso em: 20 dez. 2017.

ALVES, A. O. **Uso de chatbots para apoio ao atendimento de clientes no aplicativo Telegram**. 2021. 24f. Trabalho de conclusão de curso (Bacharelado em Ciência da Computação) — Universidade de Brasília, Brasília.

BITTENCOURT, Jane. **Educação Integral no contexto da BNCC**. Revista e-Curriculum, São Paulo, v.17, n.4, 2019, p. 1759-1780.

BRAUN, Júlio César. **Robótica Educacional: A Possibilidade de múltiplas aprendizagens no espaço escolar**. In: CONGRESSO LATINO-AMERICANO DE SOFTWARE LIVRE E TECNOLOGIAS ABERTAS (LATINOWARE), 17, 2020, Online. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2020. p. 164-167.

BRASIL. Ministério da Educação. **Base Nacional Comum Curricular**. Brasília, 2018.

DATE, C. J. **Introdução a Sistemas de Bancos de Dados**. 1ª edição. ed. [S. l.]: GEN LTC, 2004. 896 p. ISBN 8535212736.

DA SILVEIRA, C.; SILVA, A. R.; HERPICH, F.; TAROUCO, L. M. R. Uso de Agente conversacional como recurso de aprendizagem sócio-educacional. **RENOTE**, Porto Alegre, v. 17, n. 3, p. 668–678, 2019.

DIAS, T. M. **Desenvolvimento de um chatbot para Apoio a Vítimas de Violência Doméstica**. 2022. 181 f. Trabalho de conclusão de curso (Graduação em Sistemas de Informação) - Faculdade em Sistemas de Informação da UNISUL, Universidade do Sul de Santa Catarina, Florianópolis.

DORF, Richard C. & BISHOP, Robert H. **Sistemas de Controle Modernos**. Rio de Janeiro: LTC Editora, 1998. p. cap. 1-7-12.

FIELDING, Roy Thomas. **Representational State Transfer (REST)**. [S. l.], 1 jan. 2000. Disponível em: [https://www.ics.uci.edu/~fielding/pubs/dissertation/rest\\_arch\\_style.htm](https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm). Acesso em: 26 set. 2022.

FIELDING, Roy Thomas. **ASP.NET Core 3 – Autenticação e Autorização com Bearer e JWT**. [S. l.], 26 set. 2022. Disponível em: <https://balta.io/artigos/aspnetcore-3-autenticacao-autorizacao-bearer-jwt>. Acesso em: 26 set. 2022.

GUIGIDI, I. & MATTOS, W. W. **Desenvolvimento e aplicação de um Chatbot para auxiliar o atendimento ao cliente**. 2018. 75f. Monografia (Graduação em Sistemas de Informação) - Faculdade em Sistemas de Informação da UNISUL, Universidade do Sul de Santa Catarina, Florianópolis.

IBARRA, Luis Antonio Ccopa & SOARES, Marisa. **A robótica e o pensamento computacional na educação: Uma proposta de avaliação da aprendizagem baseada em projetos**. Dialogia, São Paulo, n.40, p. 1-26, 2022.

LISBOA, A. C. & ROCHA, P. A. M. **Competências Socioemocionais e Docência: A BNCC e as novas exigências na formação de professores**. Bahia: Editora Realize, 2020. p. 1-26.

MENEZES, Ana Lívia Ferreira. **EVA : uma proposta de design digital para aplicativos de monitoramento do ciclo menstrual**. 2022. 90 f. Trabalho de Conclusão de Curso (Graduação em Design Digital)- Universidade Federal do Ceará, Campus de Quixadá, Quixadá, 2022.

MICROSOFT, Brasil. **Documentação de dados .NET**. Disponível em: <https://learn.microsoft.com/pt-br/ef/dotnet-data/>. Acesso em 26 set. 2022.

PALAVRAS PROJETOS EDITORIAIS. **Conheça as 10 Competências da BNCC**. 2022. Disponível em: [https://www.palavraseeducacao.com.br/home/conheca-as-10-competencias-da-bncc?gclid=Cj0KCQjwkruVBhCHARIsACVliOxSEYXqvVMJOP46xdrVTTLjeSqmaINtPd-683UrwgtjDj24kqQaA70aAuzNEALw\\_wcB](https://www.palavraseeducacao.com.br/home/conheca-as-10-competencias-da-bncc?gclid=Cj0KCQjwkruVBhCHARIsACVliOxSEYXqvVMJOP46xdrVTTLjeSqmaINtPd-683UrwgtjDj24kqQaA70aAuzNEALw_wcB). Acesso em: 19/06/22.

SAUDATE , Alexandre. **REST. Construa API's Inteligentes de Maneira Simples**. 1ª edição. ed. [S. l.]: Casa do Código, 2013. 314 p. ISBN 978-8566250374. SESI-SP. **Aulas de Programação e Robótica: Orientações Gerais**, 2021.

SILVA, S. H. C., TOLEDO, M. V. de S., CIZILIO, L. C. de B., & LACERDA, I. I. **O uso de chatbot no auxílio de estudantes do ensino médio para o aprendizado em inglês / The use of chatbot in the help of high school students for english learning**. Brazilian Journal of Development, 6(1), 2933–2944. 2020.

SILVA, K. G. G. **Deteção Automática de Conteúdos Preconceituosos utilizando Técnicas de Classificação de Textos**. 2021. 32f. Monografia (Iniciação Científica em Ciência da Computação) - Centro Universitário Sagrado Coração, UNISAGRADO, Bauru.

THELMA, Ugonna. **The S.O.L.I.D Principles in Pictures**. [S. l.], 18 maio 2022. Disponível em: <https://medium.com/backticks-tildes/the-s-o-l-i-d-principles-in-pictures-b34ce2f1e898>. Acesso em: 26 set. 2022.

## AGRADECIMENTOS

O presente trabalho foi realizado com o apoio da Rede de Educação SESI-SP pela disponibilidade de diferentes recursos educacionais em nosso centro educacional e, principalmente, por autorizar a divulgação deste projeto através da equipe gestora da unidade.

Agradecemos, especialmente, a professora de Língua Portuguesa do Ensino Fundamental II e Médio do SESI 113 Milton Sobrosa Cordeiro, Adriana Edargo Ribeiro, pela correção gramatical deste texto.