Fatec Diadema Luigi Papaiz

Centro Paula Souza

SÃO PAULO GOVERNO DO ESTADO

Secretaria de Desenvolvimento Econômico

Sigla do Projeto - Nome do Projeto – Parte 2

Documento de Arquitetura de Software

Versão 1.0

Histórico de Revisões			
Versão	**Data**	**Descrição**	**Autor**

Sumário

Documento de Arquitetura de Software

1. Introdução

Este documento oferece uma visão geral da arquitetura do sistema, utilizando diversas visões arquiteturais para representar diferentes aspectos dele. O objetivo é comunicar as decisões significativas que foram tomadas.

O projeto "Jornada da Inclusão" tem como foco identificar e atender às necessidades específicas de crianças, de até oito anos de idade, que possuem dificuldades de inclusão em ambientes escolares públicos. Este projeto visa criar uma plataforma que combine avaliações interativas e personalizadas, além de métodos de ensino, que utilizem recursos e atividades focados nos processos de alfabetização e identificação numérica. O objetivo é cumprir as ODS 4 e 10, que abordam, respectivamente, a promoção de uma educação de qualidade e a redução da desigualdade educacional.

2. Identificação do Projeto

Projeto	Jornada da Inclusão
Requisitante	Projeto Integrador do curso de Desenvolvimento de Software Multiplataforma, da Fatec Diadema
Responsável do Projeto	Gabriel Dourado dos Santos Luciana Guedes de Araújo Manuela Tenório da Silva Marcos Vinicius de Oliveira Pedro Henrique Santos Bernardo Renato Winicius de Lima Jacob

3. Representação Arquitetural

A representação arquitetural deste software tem como objetivo não só auxiliar os desenvolvedores no entendimento e implementação dele, mas também facilitar a comunicação entre as partes interessadas. Ela é composta por diferentes visões que permitem examinar o *software* sob diversos ângulos. Cada uma destacando aspectos distintos.

Visão de Casos de Uso - Descreve o comportamento do site, do ponto de vista dos usuários finais e de outras partes externas que interagem com ele. Os elementos desta visão são os atores, os casos de uso e as relações. Os atores representam entidades

externas que interagem com o site; os casos de uso descrevem as funcionalidades oferecidas aos atores; as relações definem as formas como os atores interagem com os casos de uso através de associações, extensões e inclusões, por exemplo.

Visão Lógica - Apresenta os componentes estruturais, geralmente representados por diagramas de classes ou diagramas de componentes. Também aborda a organização dos módulos e as relações entre eles. Possui como elementos as classes, as interfaces, os pacotes e as relações. As classes definem as estruturas principais do sistema, seus atributos e métodos. As interfaces representam contratos que as classes ou componentes devem seguir. Os pacotes são agrupamentos de classes e componentes que têm responsabilidades relacionadas. Por último, as relações podem ser de associação, herança e dependência entre classes ou pacotes.

Visão de Processos - Representa os processos do software através do modelo BPMN. O BPMN (*Business Process Model and Notation*) é uma notação padrão para modelar processos de negócio de forma gráfica, tornando-os compreensíveis para profissionais de TI e *stakeholders* de negócios. Os principais elementos dele são as piscinas, as raias, os eventos, as atividades, as tarefas, os *gateways*, os fluxos e os artefatos.

Visão de Implementação - Trata dos aspectos relacionados à organização do código-fonte, estruturas de arquivos e dependências de bibliotecas. Alguns elementos são os módulos, as bibliotecas, as dependências e os *scripts* de *build*. Os módulos são os agrupamentos de código-fonte, organizados em arquivos ou pacotes. As bibliotecas são componentes externos, incluídos no projeto para fornecer funcionalidades específicas. As dependências são relações entre diferentes módulos e bibliotecas. Os *scripts* de *build* definem as regras de compilação, integração e empacotamento do *software*.

4. Metas e Restrições da Arquitetura

Metas da Arquitetura

O objetivo principal do sistema IntegraKids é fornecer soluções personalizadas e inclusivas para ajudar no desempenho escolar de crianças com dificuldade de inclusão. Isso significa que o sistema deve ajustar as atividades e avaliações para atender às necessidades de cada aluno. Garantir que o sistema seja interativo e envolvente, incluindo atividades lúdicas e jogos que atraiam as crianças, é outro objetivo importante. Além disso,

o sistema deve ser escalável para que possa ser utilizado por diversos alunos sem comprometer o desempenho.

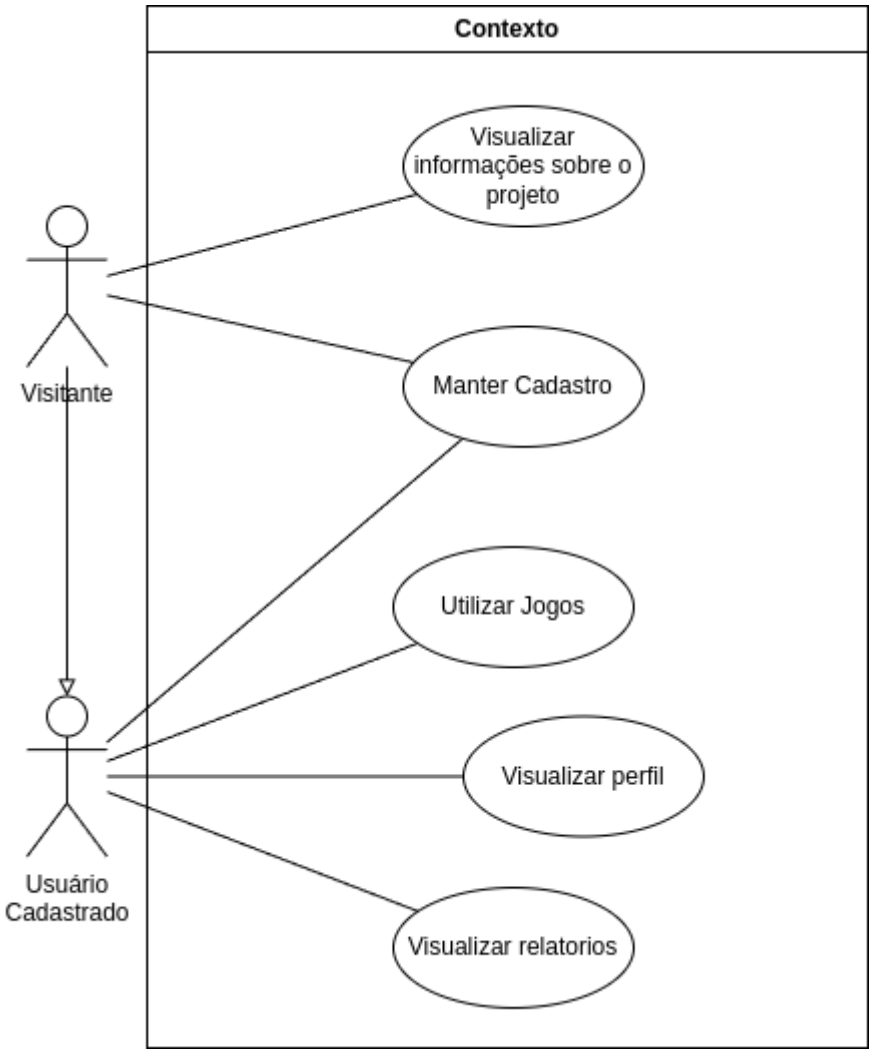
O objetivo é tornar o projeto acessível em uma variedade de plataformas, como computadores, *tablets* e *smartphones*, para que possa ser usado em uma variedade de contextos educacionais. Por tanto, de acordo com os ODS 4 (Educação de Qualidade) e 10 (Redução das Desigualdades), a estrutura do sistema deve ser projetada para promover a educação de alta qualidade e fornecer uma ferramenta gratuita e acessível a todos, garantindo assim a inclusão de todas as crianças no âmbito de ensino escolar.

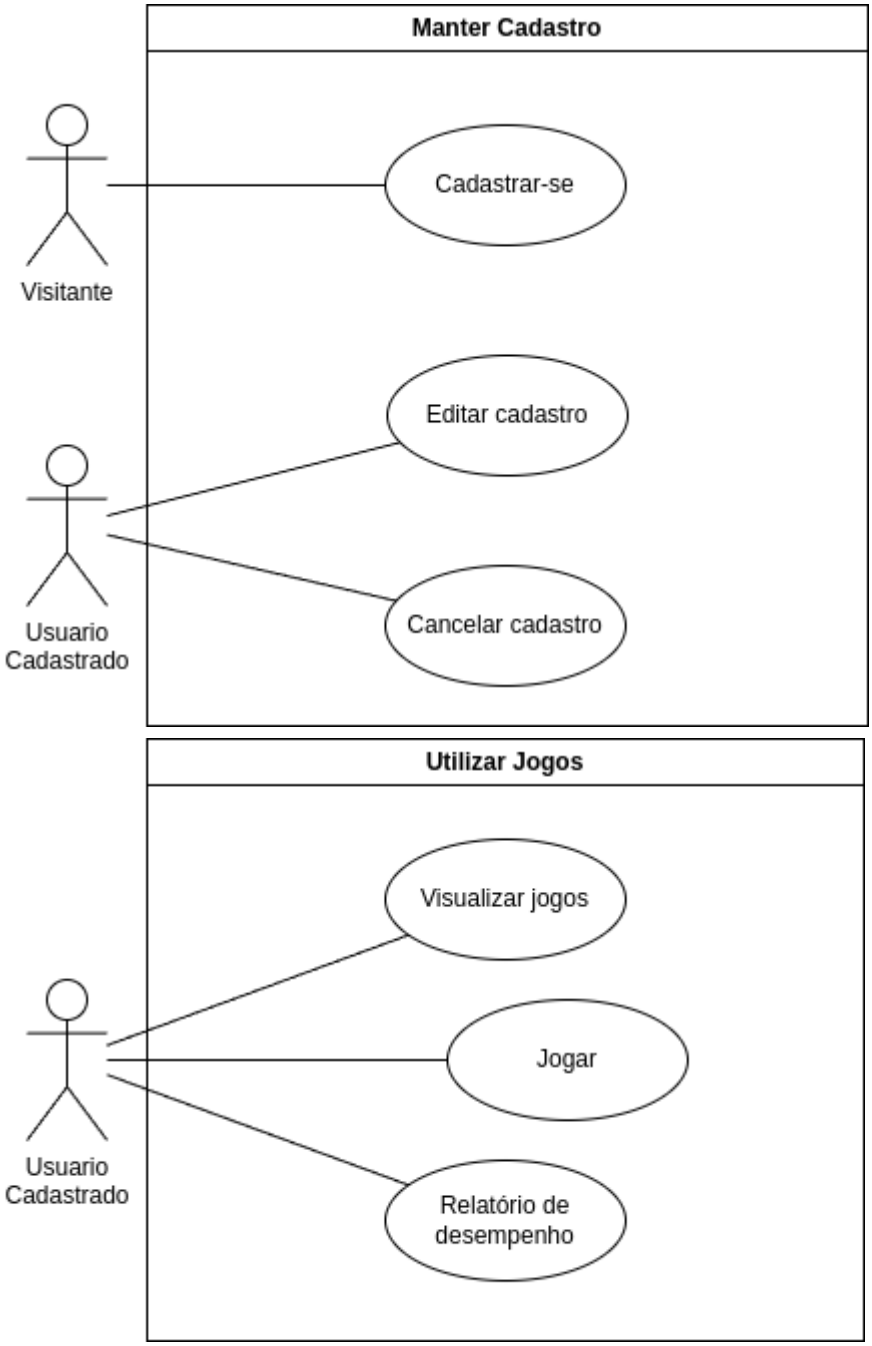
Restrições da Arquitetura

A arquitetura do sistema deve ser projetada considerando as limitações dos recursos tecnológicos de cada família e escola. Isso significa que o sistema deve funcionar bem em dispositivos com capacidades de processamento reduzidas e funcionar com eficiência mesmo em conexões de internet lentas. Ele deve ser compatível com várias plataformas, como Windows, Android e iOS, para que possa ser usado em vários dispositivos. A segurança e a privacidade dos dados dos alunos são muito importantes, e a arquitetura deve seguir a LGPD. Além disso, a arquitetura deve ser fácil de manter, permitindo atualizações regulares sem interromper o funcionamento. Para garantir que o sistema seja gratuito em longo prazo, é fundamental que a solução tenha baixos custos operacionais.

5. Visão de Casos de Uso

5.1 Realizações de Casos de Uso





6. Visão Lógica

O objetivo do sistema IntegraKids é oferecer um ambiente digital que possa identificar e atender às necessidades únicas das crianças que enfrentam dificuldades de inclusão nas escolas. Em conformidade com os Objetivos de Desenvolvimento Sustentável (ODS) 4 e 10, o sistema fornecerá atividades centradas na alfabetização e na identificação de números e cores, para promover a educação de alta qualidade e no apoio à redução das desigualdades por meio de avaliações personalizadas e interativas.

6.1 Visão Geral

O sistema será composto por várias camadas, que serão organizadas em subsistemas e pacotes para garantir a escalabilidade. A arquitetura será dividida em várias camadas, que vão desde a interface com o usuário até a persistência de dados, cada uma com responsabilidades bem definidas. A estrutura permite personalizar e adaptar o conteúdo educacional de acordo com as necessidades de cada criança.

6.2 Pacotes de Design Significativos do Ponto de Vista da Arquitetura

[Para cada pacote significativo, inclua uma subseção com o respectivo nome, uma breve descrição e um diagrama com todos os pacotes e classes significativos nele contidos.
Para cada classe significativa no pacote, inclua o respectivo nome, uma breve descrição e, opcionalmente, uma descrição de algumas das suas principais responsabilidades, operações e atributos.]

6.2.1 Pacote br.com.integrakids.presentation

Descrição: Esse pacote contém as classes responsáveis pela interface de usuário e pela experiência visual do sistema. Ele inclui componentes que tratam da interação com o usuário e da apresentação de dados.

Classes Significativas:

Login: Classe responsável pela exibição da tela de login e pelo gerenciamento das interações de autenticação do usuário.

Responsabilidades: Renderizar a tela de login, capturar os dados de entrada e enviar solicitações de autenticação para a camada de negócio.

Operações: exibirTela(), validarEntrada(), enviarAutenticacao().

Avaliacao: Classe que gerencia a exibição das avaliações interativas para as crianças.

Responsabilidades: Exibir os jogos, capturar as respostas e enviar os resultados para a camada de lógica de negócio.

Operações: mostrarAvaliacao(), capturarNota(), enviarResultados().

6.2.2 Pacote br.com.integrakids.business

Descrição: Contém as classes que implementam a lógica de negócio do sistema. Esse pacote é responsável pelo gerenciamento de usuários, avaliações e análises de desempenho.

Classes Significativas:

UsuarioPrincipal: Classe que gerencia as operações de usuários, como cadastro, autenticação e atualização de perfil.

Responsabilidades: processar e validar as solicitações de cadastro e autenticação, gerenciar dados de perfis.

Operações: autenticar(), **registrarUsuario()**, **atualizarPerfil().**

AvaliaçãoPrincipal: gerencia a criação, modificação e análise das avaliações.

Responsabilidades: criar avaliações, obter resultados e processar relatórios de desempenho.

Operações: criarAvaliacao(), **obterResultados()**, **gerarRelatorio()**.

6.2.3 Pacote br.com.integrakids.persistence

Descrição: esse pacote encapsula as classes que gerenciam a persistência de dados. Ele contém as classes que interagem diretamente com o banco de dados para armazenar, recuperar e atualizar informações.

Classes Significativas:

UsuarioBD: classe responsável pelas operações no banco de dados relacionadas aos usuários.

Responsabilidades: Salvar, buscar, atualizar e excluir registros de usuários.

Operações: inserirUsuario(), buscarUsuarioPorID(), atualizarUsuario().

AvaliacaoBD: gerencia a persistência dos dados das avaliações.

Responsabilidades: Inserir novas avaliações, buscar resultados e atualizar informações das avaliações.

Operações: inserirAvaliacao(), buscarResultadosPorUsuario(), atualizarAvaliacao().

6.3 Camadas

[Para cada camada, inclua uma subseção com o respectivo nome, uma lista dos subsistemas localizados na camada e um diagrama de componentes.]

6.3.1 Camada de Apresentação

Subsistemas Localizados:

- **Interface do Usuário:** Responsável pela interação visual e pela experiência do usuário, permitindo que os usuários acessem e utilizem as funcionalidades do sistema.
- **Responsividade e Acessibilidade:** Garantia de que o sistema seja acessível em diferentes dispositivos e adaptável às necessidades de todos os usuários, incluindo crianças com dificuldades de inclusão.

6.3.2 Camada de Lógica de Negócio

Subsistemas Localizados:

- Gerenciamento de Usuários: Controla as operações relacionadas ao cadastro, autenticação e autorização de usuários.

- Gerenciamento de Avaliação: Cria, modifica e analisa as avaliações aplicadas às crianças para identificar suas dificuldades educacionais.

Descrição: esta camada contém a lógica que processa as informações e regras de negócio do sistema. Ela é responsável por gerenciar a comunicação entre a camada de apresentação e a camada de persistência de dados.

6.3.3 Camada de Persistência de Dados

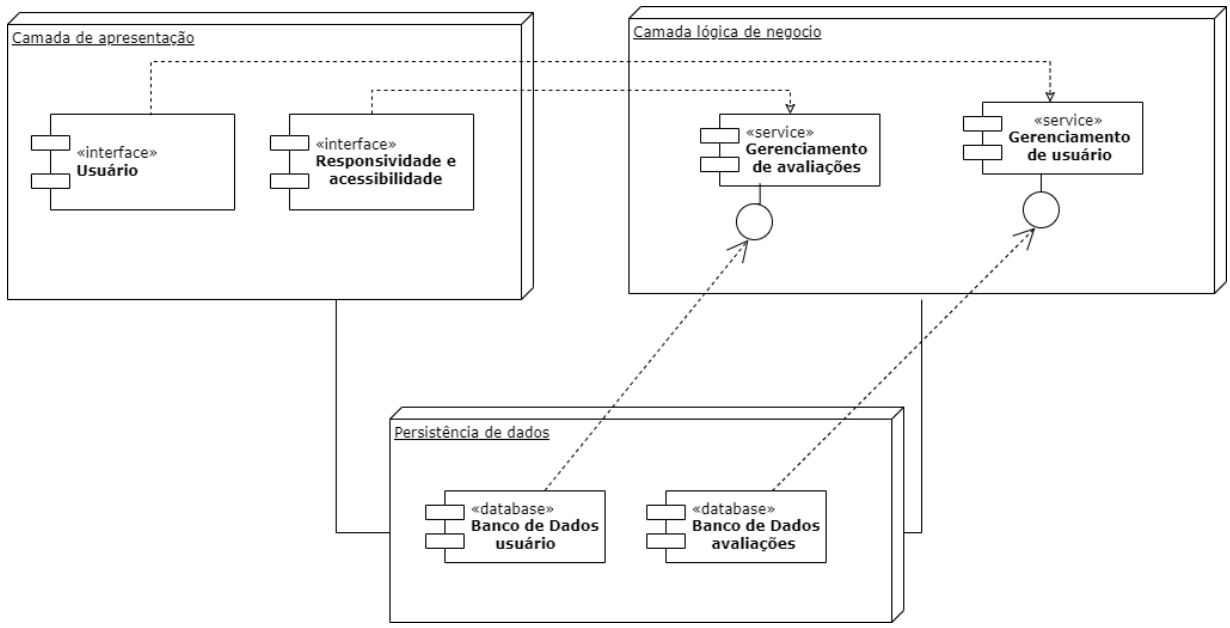
Subsistemas Localizados:

- Banco de Dados de Usuários: Armazena informações sobre todos os usuários cadastrados, suas credenciais e perfis.

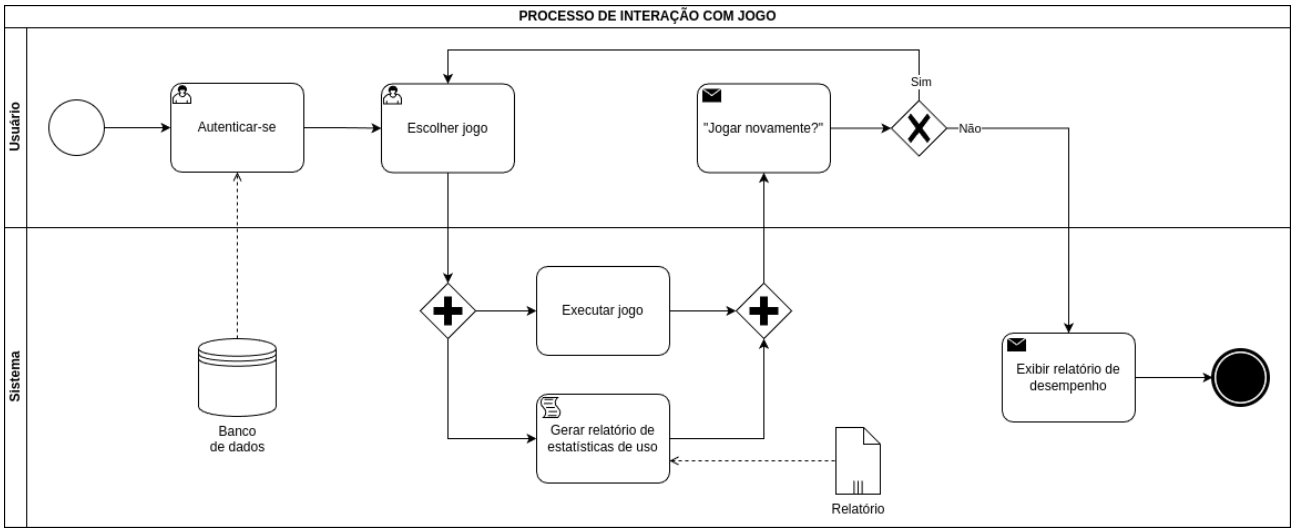
- Banco de Dados de Avaliações: Contém dados relacionados às avaliações, incluindo jogos e resultados.

Descrição: esta camada é responsável pela persistência e recuperação dos dados do sistema. Ela gerencia as interações com o banco de dados e assegura a integridade e segurança das informações.

Essas camadas e pacotes estruturam o sistema IntegraKids de forma modular e escalável, permitindo a adaptação e evolução contínua do ambiente digital para atender às necessidades educacionais das crianças.



7. Visão de Processos



Segundo o Object Management Group (2010), o BPMN (*Business Process Model Notation*) é um diagrama que detalha as etapas de um processo de negócio de modo compreensível tanto aos stakeholders quanto aos desenvolvedores do projeto, criando uma ponte entre a modelagem dos processos do negócio e seu desenvolvimento e implementação em software.

8. Visão de Implantação

[Esta seção descreve uma ou mais configurações da rede física (hardware) na qual o software é implantado e executado. Ela é uma visão do Modelo de Implantação. No mínimo, para cada configuração, ela deve indicar os nós físicos (computadores, CPUs) que executam o software e suas interconexões (barramento, LAN, ponto a ponto, etc.) É incluído também um mapeamento dos processos da Visão de Processos nos nós físicos.]

9. Visão da Implementação

A estrutura da implementação possui bibliotecas e ferramentas utilizadas para facilitar o processo de desenvolvimento. Cada uma delas está presente no arquivo package.json, que está dentro do projeto.

Node.js - um ambiente de tempo de execução de JavaScript que permite a execução da linguagem em servidores. É ideal para a construção de ambientes *back-end* e APIs. Utiliza o motor V8 do Chrome, que é rápido e eficiente.

React.js - uma biblioteca JavaScript criada pelo Meta (anteriormente Facebook). Ela possibilita a construção de interfaces de usuário através da criação de componentes

reutilizáveis e o gerenciamento do estado da aplicação de forma eficaz. Assim facilitando o desenvolvimento de interfaces dinâmicas.

Vite - uma ferramenta de build rápida para *front-end* e que pode ser utilizada junto de bibliotecas, como o React.js, ou *frameworks*, como o Vue.js. Utiliza módulos ES e um servidor de desenvolvimento rápido, para recarregar as mudanças do projeto instantaneamente.

ESLint - um *linter* de JavaScript que auxilia o programador a manter o código limpo e padronizado por meio da identificação de erros e sugestão de correções. Como resultado, há uma maior legibilidade do código e a prevenção de *bugs*.

Arquivo Package.json - abaixo está uma imagem do arquivo package.json.

```
{
  "name": "pi-jornadainclusao",
  "private": true,
  "version": "0.0.0",
  "type": "module",
  "scripts": {
    "dev": "vite",
    "build": "vite build",
    "lint": "eslint .",
    "preview": "vite preview"
  },
  "dependencies": {
    "react": "^18.3.1",
    "react-dom": "^18.3.1",
    "react-router-dom": "^6.26.2"
  },
  "devDependencies": {
    "@eslint/js": "^9.9.0",
    "@types/react": "^18.3.3",
    "@types/react-dom": "^18.3.0",
    "@vitejs/plugin-react": "^4.3.1",
    "eslint": "^9.9.0",
    "eslint-plugin-react": "^7.35.0",
    "eslint-plugin-react-hooks": "^5.1.0-rc.0",
    "eslint-plugin-react-refresh": "^0.4.9",
    "globals": "^15.9.0",
    "vite": "^5.4.1"
  }
}
```

9.1. Visão Geral

[Esta subseção nomeia e define as diversas camadas e o seu conteúdo, as regras que determinam a inclusão em uma camada específica e as fronteiras entre as camadas. Inclua um diagrama de componentes ou pacotes que mostre os relacionamentos entre as camadas.]

10. Visão de Dados (opcional)

[Uma descrição da perspectiva de armazenamento de dados persistentes do sistema. Esta seção será opcional se os dados persistentes forem poucos ou inexistentes ou se a conversão entre o Modelo de Design e o Modelo de Dados for trivial.]

- *Ver se é aplicável no nosso PI – Ver com Pedro*

11. Qualidade

Garantir acessibilidade, disponibilizando uma interface amigável de navegação clara e concisa, de acordo com a faixa etária, com legendas, audiodescrição ou uso de sinais visuais para usuários com deficiência auditiva, personalização de tamanho de fonte, cores e contrastes para atender outros tipos de necessidades, como por exemplo, daltonismo.

O conteúdo disponível no sistema de acordo com a faixa etária, respeitando o ciclo de desenvolvimento cognitivo de cada usuário, utilização de recursos visuais, sonoros e participativo, de forma a desenvolver as várias formas de aprendizagem: visual, auditiva e cinestésica, uso de gamificação com técnicas lúdicas para desenvolver a aprendizagem, atividades com níveis de habilidades de acordo com cada usuário e atividades diferenciadas.

A interatividade fornecendo um feedback em tempo imediato sobre desempenho, melhorando a aprendizagem individual e corrigindo erros, atividades de exploração englobando experimentos e descobertas novas de forma descontraída.

Áreas para responsáveis para controle de monitoramento do progresso da criança e personalização de seu aprendizado, emitindo relatórios de desempenho, mostrando as áreas em que a criança precisa de mais atenção, instruções e manual para a utilização do sistema web com mais praticidade e eficiência.

O sistema web funcionando em quaisquer dispositivos como computadores, smartphones, tablets, atendendo os diferentes meios de uso da tecnologia.

Os dados dos usuários garantidos pela lei nº 13.709/2018, a LGPD (Lei Geral de Proteção de Dados), com ambiente seguro e controlável, evitando conteúdos inadequados, e garantido o acesso aos responsáveis de forma segura e exclusiva.

Conteúdos desafiadores e lúdicos para manter o interesse e a motivação das crianças, suporte tanto para os usuários quanto aos seus responsáveis.

Disposição de área para que todos os usuários possam avaliar e sugerir melhorias na plataforma.

Alinhar conteúdos e atividades de acordo com as diretrizes pedagógicas da escola, para assim, caminhar junto ao aprendizado de sua entidade de ensino.

12. Anexos

[Relacione aqui eventuais documentos que fazem parte do presente artefato, tais como atas de reunião, cronograma e outros.]

(etapa final do projeto - atas de reunião, cronogramas e outros)

13. Referências

OBJECT MANAGEMENT GROUP. **Business Process Model Notation**. 2010.

14. Aprovações

Aprovações		
Participante	**Assinatura**	**Data**

- *O que já foi aprovado no projeto*
  Ex: sprints, backlogs.....