

CENTRO ESTADUAL DE EDUCAÇÃO PAULA SOUZA

ETEC ZONA LESTE

Mtec Desenvolvimento De Sistemas

João Pedro Bispo de Biasi

Lucas Bonfim Vilela

Neluma Siqueira Lopes

BLOOMY: Sistema Interativo Para Amparo Educacional.

São Paulo

2025

João Pedro Bispo de Biasi

Lucas Bonfim Vilela

Neluma Siqueira Lopes

**BLOOMY: Sistema Interativo No Amparo Educacional Do Ensino
Básico.**

Trabalho de Conclusão de Curso apresentado ao
Mtec Desenvolvimento de Sistemas AMS da Etec
Zona Leste, orientado pelo Prof. Esp. Jeferson
Roberto de Lima, como requisito parcial para a
obtenção do título de técnico em Desenvolvimento
de Sistemas.

São Paulo

2025

BLOMMY

Sistema Interativo No Amparo Educacional Do Ensino Básico.

João Pedro Bispo de Biasi

Lucas Bonfim Vilela

Neluma Siqueira Lopes

Aprovada em __/__/____.

BANCA EXAMINADORA

Prof. Esp. Jeferson Roberto de Lima

Universidade do Jeferson

Prof. (Professor avaliador)

Universidade do Avaliador

Prof. (Professor avaliador)

Universidade do Avaliador

RESUMO

Contrary to popular belief, Lorem Ipsum is not simply random text. It has roots in a piece of classical Latin literature from 45 BC, making it over 2000 years old. Richard McClintock, a Latin professor at Hampden-Sydney College in Virginia, looked up one of the more obscure Latin words, consectetur, from a Lorem Ipsum passage, and going through the cites of the word in classical literature, discovered the undoubtable source. Lorem Ipsum comes from sections 1.10.32 and 1.10.33 of "de Finibus Bonorum et Malorum" (The Extremes of Good and Evil) by Cicero, written in 45 BC. This book is a treatise on the theory of ethics, very popular during the Renaissance. The first line of Lorem Ipsum, "Lorem ipsum dolor sit amet..", comes from a line in section 1.10.32.

The standard chunk of Lorem Ipsum used since the 1500s is reproduced below for those interested. Sections 1.10.32 and 1.10.33 from "de Finibus Bonorum et Malorum" by Cicero are also reproduced in their exact original form, accompanied by English versions from the 1914 translation by H. Rackham

Palavras-Chave: lore; ipsum; lore lore ipsum; ipsum ipsum; ipsum lore.

ABSTRACT

Contrary to popular belief, Lorem Ipsum is not simply random text. It has roots in a piece of classical Latin literature from 45 BC, making it over 2000 years old. Richard McClintock, a Latin professor at Hampden-Sydney College in Virginia, looked up one of the more obscure Latin words, consectetur, from a Lorem Ipsum passage, and going through the cites of the word in classical literature, discovered the undoubtable source. Lorem Ipsum comes from sections 1.10.32 and 1.10.33 of "de Finibus Bonorum et Malorum" (The Extremes of Good and Evil) by Cicero, written in 45 BC. This book is a treatise on the theory of ethics, very popular during the Renaissance. The first line of Lorem Ipsum, "Lorem ipsum dolor sit amet..", comes from a line in section 1.10.32.

Keywords: lore; ipsum; lore lore ipsum; ipsum ipsum; ipsum lore.

LISTA DE ILUSTRAÇÕES

Figura 1 - Diagrama de caso de uso	16
Figura 2 - Diagrama de sequência	20
Figura 3 - Diagrama de classe	21
Figura 4 - Diagrama de Máquina de Estados	22
Figura 5 - Diagrama de atividade	23
Figura 6 - Wireframe de baixa fidelidade.....	24
Figura 7 - Wireframe de alta fidelidade	25
Figura 8 - Exemplo de código em C++	30

LISTA DE ABREVIATURAS E SIGLAS

LISTA DE TABELAS

LISTA DE SÍMBOLOS

SUMÁRIO

1	INTRODUÇÃO	13
2	REFERENCIAL TEÓRICO.....	14
	2.1 Desamparo Educacional Sentido Nos Alunos do Ensino Básico	14
	2.2 Tecnologias utilizadas.....	15
	2.2.1 Design Thinking.....	15
	2.2.2 Unified Modeling Language (UML).....	15
	2.2.3 Wireframes	24
	2.2.4 HTML.....	25
	2.2.5 CSS	26
	2.2.6 JS	26
	2.2.7 React.....	27
	2.2.8 React Native	27
	2.2.9 Typescript.....	28
	2.2.10 Banco de dados	28
	2.2.11 Firebase	28
	2.2.12 Construct.....	29
	2.2.13 C++.....	29
	2.2.14 Internet of Things (IOT)	31
	2.2.15 ESP32	31
	2.2.16 Modelagem 3D	32
	2.2.17 Impressão 3D	32
3	DESENVOLVIMENTO.....	33
4	CONSIDERAÇÕES FINAIS	34
	REFERÊNCIAS	35

1 INTRODUÇÃO

2 REFERENCIAL TEÓRICO

Este capítulo discorre sobre toda a fundamentação teórica pesquisada para basear a construção do projeto.

2.1 Desamparo Educacional Sentido Nos Alunos do Ensino Básico

Um fato sobre o baixo desempenho acadêmico, discute a relação da má vontade de absorver os saberes passados pelo educador, que não ampara as vivências únicas do aluno, dificultando a trilha para uma educação libertadora. (FREIRE, 1996) Além disso, sabe-se que a escola é o meio essencial para um futuro de qualidade, mas pouca importância é dada para o aprendizado; muitos estudantes julgam a escola como insatisfatória, sem sentido, cansativa e conflituosa. (MARQUES; CASTANHO, 2011)

Segundo Ana Parisotto e Renata Rinaldi (2016), os professores do ensino básico estão com menos respaldo acadêmico, expondo a dificuldade das matérias em serem compreendidas por si só quando há carência de outras formas educacionais. Com isso, de acordo com Samara de Sena *et. al.*, as mudanças do século XXI tornam evidente que a abordagem educacional deve ser revista e jogos epistêmicos é uma aposta promissora. (SENA, *et. al*, 2016)

2.1.1 Jogos sérios

Como dito por Vasconcelos *et. al* (2017), estudiosos buscaram formas de vincular o brincar dos jogos com o aprendizado, chamados "jogos sérios", onde planejam usar de ferramentas lúdicas promovendo habilidades autônomas aos alunos. Segundo Paula (2015), jogos trazem um mundo imersivo – trilha para o subconsciente das pessoas – e qualquer mensagem, para ele é passível de reflexões intimistas pelo jogador, criando e desenvolvendo o pensamento crítico.

Porém, como visto atualmente, a incidência do uso excessivo de aparelhos eletrônicos cresceu, o que leva todo o foco e interesse por tarefas que te desafiam a diminuir em comparação com as atividades contrárias, como jogos. (MINAS, 2023) Tal atividade, traz ao cotidiano das crianças uma gama de estímulos, pondo-as num comodismo, devido a recorrentes atividades superficiais e pouco desafiadoras,

levando a um baixo rendimento e interesse nas aulas. (G1, 2024)

Considerando os fatos, como uma plataforma de jogos-sérios pode amenizar o baixo rendimento escolar?

2.2 Tecnologias utilizadas

Para a confecção do Bloomy, foram usadas tecnologias pensadas para a solução do problema proposto, a seguir teremos o referencial teórico sobre essas ferramentas que serão utilizadas.

2.2.1 Design Thinking

De acordo com Talita Pagani, o design tem a função de desenhar um caminho para resolução de um problema – de maneira humana e com atenção individual acima das ferramentas de resolução (PAGANI, 2018).

Como visto na Revista eletrônica de biblioteconomia e ciência da informação, Design Thinking é um método de inovação com a função de entender e solucionar problemas de forma criativa em diversas condições. (FEDERAL DE SANTA et al., 2022).

É apontado no livro Design Thinking: Inovação em Negócios:

Ao desafiar os padrões de pensamento, comportamento e de sentimento 'Design Thinkers' produzem soluções que geram novos significados e que estimulam os diversos aspectos (cognitivo, emocional e sensorial) envolvidos na experiência humana. (VIANNA et al., 2012, p. 14).

Em serviços que lidam com pessoas, o modelo tradicional de resolução de problemas, sistemático e objetivo, não é adequado, é necessário se moldar às necessidades, emoções e desejos de cada indivíduo para entrega de requisitos (PAGANI, 2018).

2.2.2 Unified Modeling Language (UML)

É uma linguagem – uma ferramenta de modelagem – que visa analisar e projetar sistemas orientados a objeto, limitando e direcionando os objetivos do projeto, buscando atender o melhor resultado em todas as áreas possíveis. (BOOCH; RUMBAUGH; JACOBSON, 2006)

Segundo Guedes (2009), a UML é uma linguagem de modelagem, logo possui a meta de auxiliar engenheiros de software para a construção – caracterização e detalhamento – da aplicação, não sendo exclusiva à apenas eles.

Conforme descrito por Larman (2000), essa ferramenta oferece padrões, que torna possível a visualização e disposição de soluções mais fácil – não prescreve passo a passo, mas oferece estrutura e seriedade ao seu projeto, tornando-o manutenível.

Dentro desse modelo, temos padrões normativos para cada especificação do projeto – não necessariamente equivalentes ao código, mas aproximados – não sendo um modelo rígido, apenas um norte do que deve ser feito. (FOWLER, 2007)

2.2.2.1 Diagrama de Casos de Uso

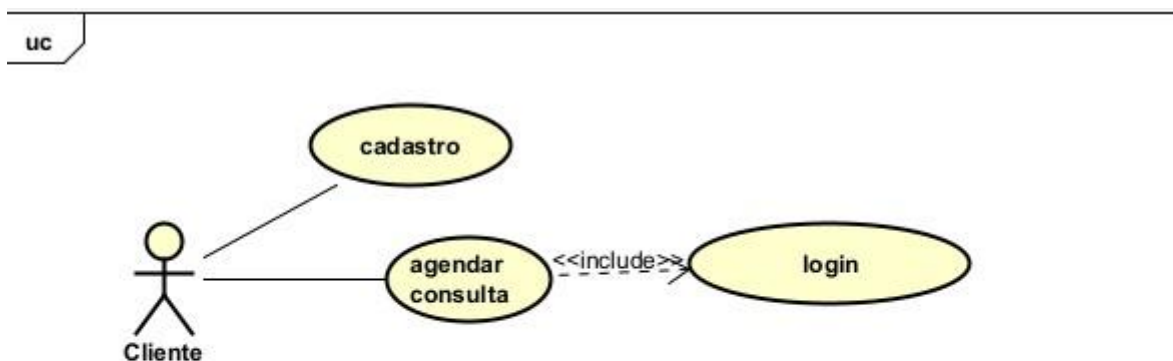
Assim como outros diagramas dinâmistas informais, o de Casos de Uso modela a atitude do sistema – como ele deve agir, que comportamento ele precisa ter – envolvendo contexto, e requisitos da aplicação. (BOOCH; RUMBAUGH; JACOBSON, 2006)

Como aponta Guedes (2009), objetiva exemplificar as funcionalidades do sistema, moldando de forma geral, o que ocorrerá – a cadeia de eventos vinculada a cada usuário e função – e baseando outros diagramas.

Em outras palavras, converge um ator e os processos que ele desencadeia na aplicação, baseando a construção do software –sendo atores os usuários e os casos de uso às funções – de fácil entendimento prévio da narração dos requisitos.

(LARMAN, 2000)

Figura 1 - Diagrama de caso de uso



Fonte: Autoria Própria, 2025

O exemplo acima, demonstra o modelo simples de casos de uso, tendo um ator chamado "Cliente", e os casos de uso "Cadastro", "Agendar Consulta" e "Login". O Cliente pode interagir com todos os processos, ou seja, ele pode se cadastrar, e em

seguida gerenciar seu perfil; é possível que o ator “Cliente” agende uma consulta, mas apenas após efetuar o login, como mostra pela seta tracejada “<<include>>” – elemento condiciona uma ação antes que o caso de uso principal seja realizado.

2.2.2.2 Documentação dos Casos de Uso

Para Booch, Rumbaugh e Jacobson (2005), os comportamentos dos Casos de Uso é o conjunto de ações que são utilizadas para visualizar e documentar o sistema e em função do que foi estimado, convertendo tais requisitos em detalhes fixos.

Contudo, Guedes (200) afirma que a documentação do caso de uso, é a maneira de simplificar as funções gerais expostas no diagrama – tendo o contexto dos processos, os atores que interagem, e as funções que irão ocorrer.

Para Larman (2000), o documento narrativo formal e sucinto – tem as mesmas funções que o diagrama derivado, só que de maneira detalhista, onde aponta as

Tabela 1 - Tabela de ações de caso de uso

Nome do Caso de Uso	Marcar Consulta.
Ator Principal	Cliente.
Atores Secundários	Nenhum.
Resumo	Este caso de uso expressa o fluxo de um cliente marcando uma consulta.
Pré-Condições	É preciso efetuar o log-in.
Pós-Condições	Uma consulta é cadastrada vinculada ao cliente,
Fluxo Principal	
Ações do Ator	Ações do Sistema

1. Solicitar Login.	
	2. Inserir informações do Cliente.
3. Informar a senha da conta para efetuar login.	4. Consultar cliente com base na senha fornecida e CPF.
5. Solicitar página de marcar consulta.	
6. Passa parâmetros para efetuar consulta.	
	7. Inserir consulta.
Restrições/ Validações	1. Para se cadastrar é preciso ser maior de idade. 2. Para marcar uma consulta, é preciso efetuar login.

Fonte: Autoria Própria, 2025

Explicação do exemplo:

Define-se o nome do caso de uso, chamado “Marcar Consulta”.

Então é informado o caso de uso geral, no entanto não há nenhum, então não fora especificado.

Logo, frisado o ator principal e o secundário, respectivamente, é aquele que mais interage com o caso de uso documentado, sendo ele o “Cliente”; não possui, logo não foi especificado. E também um resumo sobre a função desse caso de uso em específico.

Em seguida é definido as pré-condições e pós-condições, as etapas que ocorrem anterior e seguinte ao caso de uso, sendo elas “Efetuar login” e “Cadastrar Consulta” subsequentemente.

Após isso, temos o fluxo principal – onde mostra a cadeia de eventos principais que regem o caso de uso, separado por etapas dos atores e do sistema – como no exemplo, o login é solicitado pelo “Cliente”, que envia suas informações para o sistema inserir e fazer a verificação; com isso é possível fazer o processo de cadastrar consulta, onde o “Cliente” novamente informa seus dados para cadastrar e o sistema além de inserir, retorna à consulta finalizada.

2.2.2.3 Diagrama de Sequência

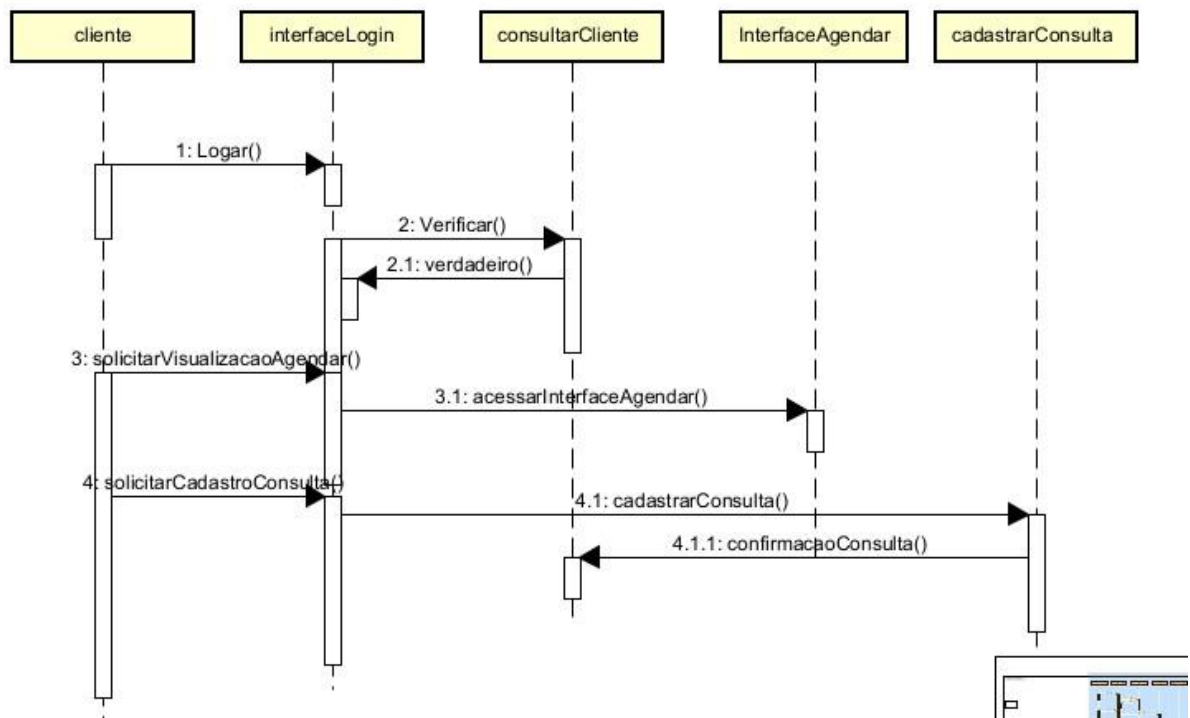
Segundo Fowler (2007), é a fragmentação de um caso de uso baseada em cenários – o que acontecerá para o usuário em determinada situação – mostrando possíveis resultados para cada situação.

Como explicado por Guedes (2009), é apoiado nos processos desencadeados pelo usuário, determinando o desenrolar do cenário – validando e complementando o diagrama de classes, possibilitando uma aplicação mais sólida e precisa.

Com uma ordenação de mensagens e eventos, o diagrama de sequência dispõe os objetos que reagem e afetam outros, prevendo a duração da interação de usuário com objetos, chamadas de funções e valores de retorno. (BOOCH; RUMBAUGH; JACOBSON, 2006)

Exemplo:

Figura 2 - Diagrama de sequência



Fonte: Autoria Própria, 2025

Explicação do exemplo:

Podemos ver os retângulos de ativação – atores ou funcionalidades ativas no diagrama – sendo eles “Cliente”, “InterfaceLogin”, “consultarCliente”, “interfaceAgendar” e “cadastrarConsulta”.

O “Cliente”, na sua linha de vida – linha tracejada abaixo dos retângulos – pode efetuar o login, para ser encaminhado para “interfaceLogin”. Após essa solicitação, os dados fornecidos são levados para “consultarCliente” – onde as informações são validadas – que caso seja verdadeiro, ou seja, esse usuário está cadastrado no sistema, é possível ver outras funcionalidades.

O “Cliente” pode navegar pela “interfaceAgendar”, onde é cabível o cadastro de consulta, que retornará o resultado “confirmacaoConsulta”.

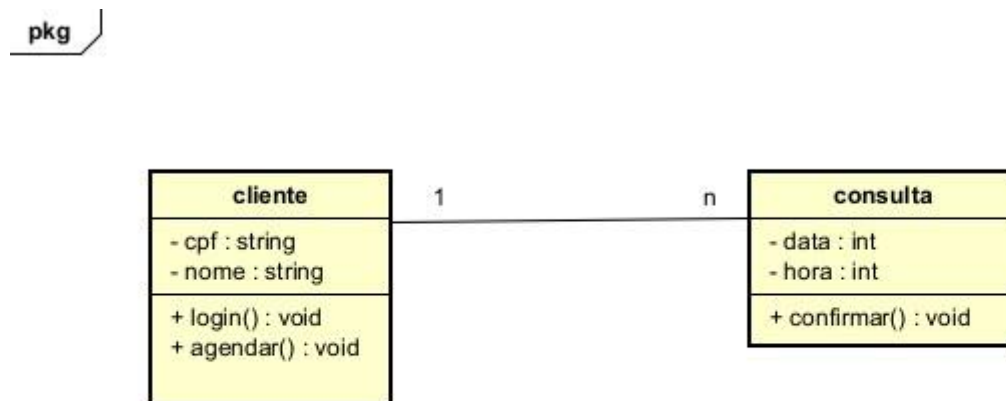
2.2.2.4 Diagrama de Classe

Fowler (2005), é um modelo que apresenta de forma clara, componentes relacionados às funções do sistema, descrevendo os presentes objetos – o que eles compõem, como interagem entre si, e como operam.

Preocupa-se com a criação da estrutura lógica, definindo os comportamentos das características gerais dos sistemas orientados a objeto – foca em apresentar as necessidades pertinentes do projeto – como afirma Guedes (2009).

Um dos diagramas mais utilizados na modelagem de orientados a objeto, facilitando o entendimento da aplicação, ajudando na redução de erros, e mapeando o sistema – como uma planta baixa para os arquitetos (BOOCH; RUMBAUGH; JACOBSON, 2006)

Figura 3 - Diagrama de classe



Fonte: Autoria Própria, 2025

Explicação do exemplo:

Está sendo apresentado duas classes – representadas por esses retângulos – chamadas “Cliente” e “Consultas”. Para a classe “Cliente” efetuar suas ações ele precisa dos atributos – objetos dentro da classe, onde são guardados os dados – que irão ser recebidos pelos métodos – ações que ocorrem dentro de uma ou mais classes que podem ou não retornar um resultado – chamadas “Login” e “Agendar”; seguindo o mesmo princípio, a classe “Consulta” precisa de “data” e “hora” para efetuar a confirmação se houve ou não uma consulta marcada.

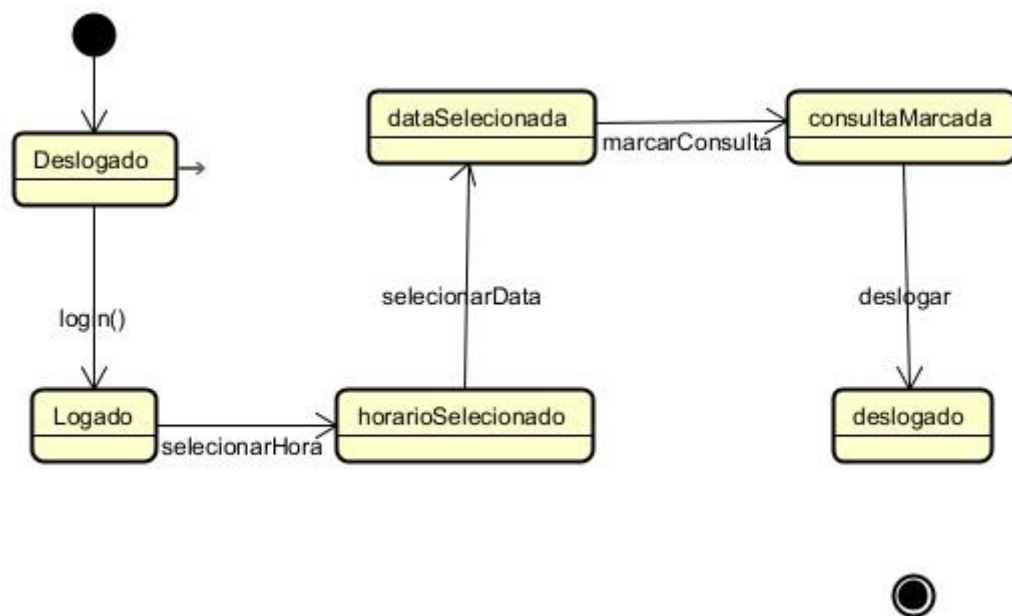
2.2.2.5 Diagrama de Máquina de Estados

De acordo com Fowler (2007), é um diagrama que explana as mudanças de estados em uma máquina – seja devido a imposições ou condições – também definindo uma base de dependência do seu objeto sobre o estado atual.

Usado para modelar as atividades dinâmicas de cada aspecto individual da vida de um sistema, como uma cadeia de eventos determinados pelo estado do objeto — que causará uma ou mais reações. (BOOCH; RUMBAUGH; JACOBSON, 2006)

Larman Craig (2000), explica que o diagrama de estados modela apenas o que é necessário, indo de detalhista à simplista de maneira arbitrária — de acordo com os requisitos do sistema.

Figura 4 - Diagrama de Máquina de Estados



Fonte: Autoria Própria, 2025

Explicação do exemplo:

O início do diagrama é marcado pelo círculo preto.

Como explicado anteriormente, esse diagrama demonstra estados – retângulos amarelos, que representam a reação após ações das classes e objetos – sendo eles “Deslogado”, “Logado”, “horarioSelecioneado”, “dataSelecioneada”, “consultaMarcada” e novamente “Deslogado”.

As setas indicam eventos, como métodos – ações que afetam os estados – sendo elas “login”, “slecionarHora”, “selecionarData”, “marcarConsulta” e “deslogar”.

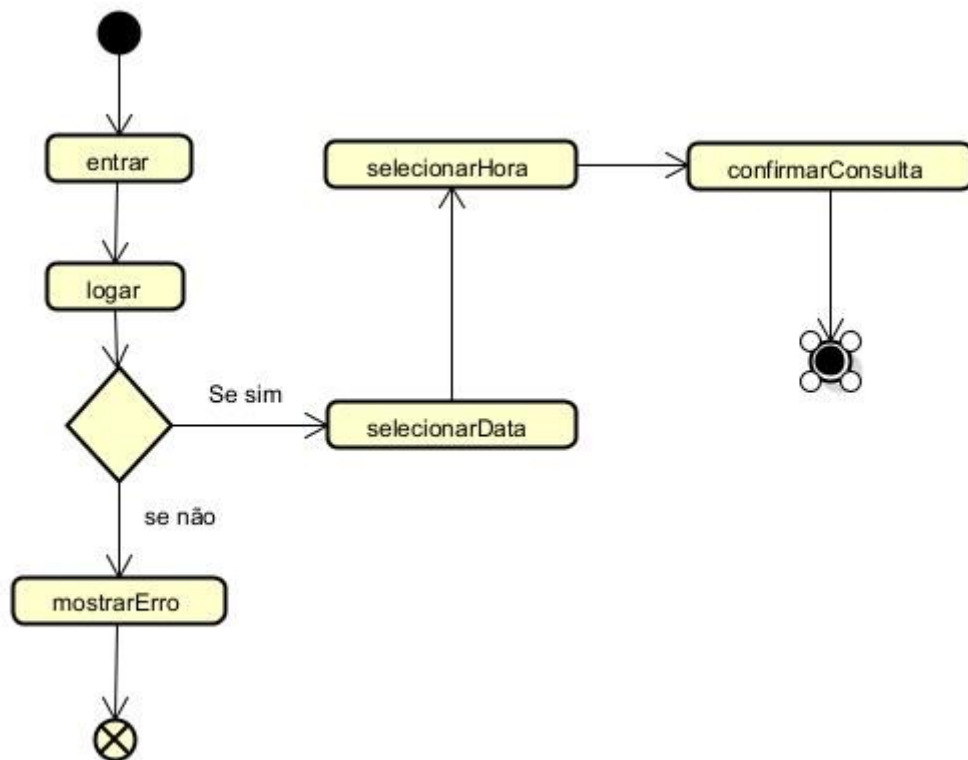
O diagrama é finalizado pelo círculo preto dentro de outro.

2.2.2.6 Diagrama de Atividade

Fowler (2007) expõe, é a ferramenta que indica o fluxo – espécie de fluxograma, onde registra a cadeia de atividades lógicas – simultâneas ou intercaladas, que ocorrerão durante o uso do sistema.

Elucide Guedes (2009), que este modelo enfatiza o curso das operações que serão realizadas pelos métodos – ou seja, um conjunto de ações que expressam as funcionalidades, podendo ou não finalizar a atividade.

Figura 5 - Diagrama de atividade



Fonte: Autoria Própria, 2025

Explicação do exemplo:

O diagrama se inicia com o círculo preto preenchido. segue para as atividades “entrar” e “login”, onde ocorre uma validação, se “login” retorna falso, o sistema mostrar

invalidez e fecha atividade; caso retorne verdadeiro, o fluxo continua possibilitando marcar consulta, pelo caminho “selecionarData” e “selecionarHorario”, podendo assim “confirmarConsulta” e encerrar as atividades como um todo.

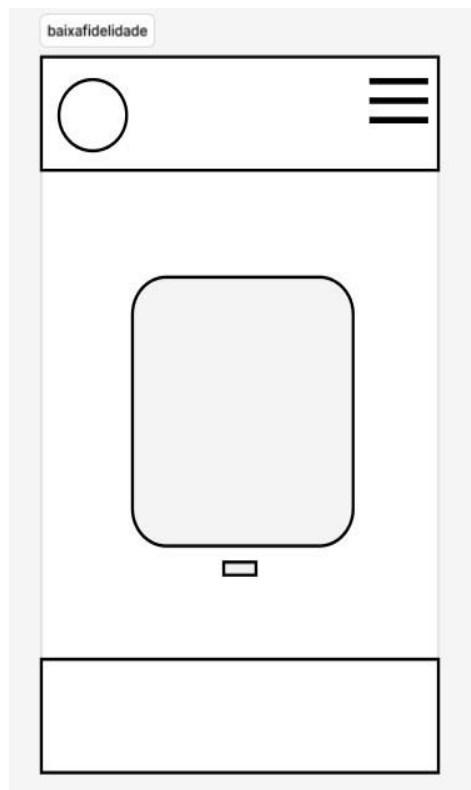
2.2.3 Wireframes

Exemplifica Caio Oliveira, (2021) que wireframes são como as interfaces irão ser montadas – representam a aparência, navegação e funcionalidades básicas do site – tornando mais concretos os eventos feitos em diagramas.

Como desenhos, afirma Fabrício (2014), os wireframes podem ser mais ou menos elaborados, depende de quão perto do design final o produto se encontra, dito isso – wireframes de baixa fidelidade são como rascunhos, passíveis de mudanças.

De acordo com Rogério (2018), os wireframes de alta fidelidade, são os mais próximos de um resultado – quanto mais específico for, menos chances de explorar criativamente se tem, abaixando as possibilidades de uma identidade visual própria.

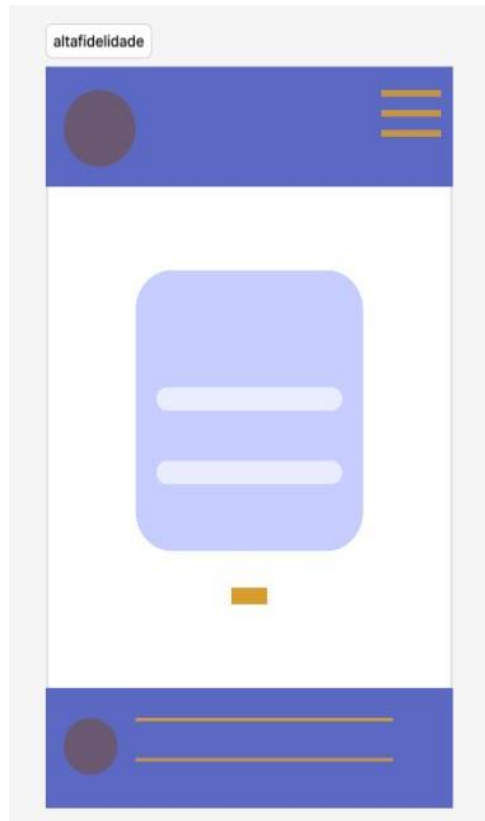
Figura 6 - Wireframe de baixa fidelidade



Fonte: Autoria Própria, 2025

Para modelar interfaces, pode ser utilizado o Figma, um software que auxilia no processo de desenvolvimento em wireframes, na aparência e na navegabilidade. (ACETI *et. al*, 2023)

Figura 7 - Wireframe de alta fidelidade



Fonte: Autoria Própria, 2025

2.2.4 HTML

De acordo com Fábio Flatschart (2011), o HTML, ou Marcação de Linguagem de Hipertexto (tradução de HyperText Markup Language), é a principal linguagem para construção de páginas na internet.

Segundo Maurício Samy (2008), Hipertexto é todo conteúdo que pode ser inserido em páginas da internet e que possuam a possibilidade de se interligarem com outras aplicações da web, utilizando de links para a sua construção.

Na linguagem HTML, utilizamos de *tags* para a escritura da linguagem, que usam das sintaxes “< >” ao início e “</ >” ao fim de cada linha. Na construção de aplicações web, as *tags* servem para estruturar como os elementos são inseridos na página

(FLATSCHART, 2011). As principais *tags* utilizadas no HTML para a construção de aplicações são:

- “<!DOCTYPE html>” informa a versão utilizada na marcação ao navegador, permitindo que o mesmo processe a página da melhor forma na internet;
- “<html> & </html>” criado logo após a tag DOCTYPE, esta tag detém todo o código da página, indicando o início e o fim do mesmo;
- “<head> & </head>” está localizado dentro da tag <html>, serve como cabeçalho da aplicação, além de deter as configurações da página;
- “<body> & </body>” está localizado dentro da tag <html>, todo o código dos elementos visíveis na página é escrito dentro desta tag.

2.2.5 CSS

Folhas de Estilo em Cascata, ou apenas CSS, é a linguagem de estilização utilizada para a formatação do layout de páginas. O CSS facilita a estilização das páginas criando um arquivo geral para o projeto, economizando tempo formatando os arquivos de e facilitando suas atualizações (JOBSTRAIBIZER, 2009).

Segundo Evandro Manara Miletto e Silvia de Castro Bertagnolli (2014), a linguagem CSS é o que possibilita a criação de estilos personalizados para títulos, imagens, listas e outros, além de permitir a criação de paleta de cores, fontes para texto, alinhamento dos objetos, entre outras características relacionada à aparência da aplicação.

Afirma Felipe Alex Scheidt (2015) que o termo “Folha de Estilo” possui o significado de um conjunto de regras que permitem definir a aparência de uma aplicação web, onde as características da página como cores e posições dos objetos são definidas através dessas regras.

2.2.6 JS

Baseado em David Flanagan (2013), a linguagem JavaScript é utilizada pela ampla maioria dos sites e navegadores modernos. Junto ao HTML e CSS, o JavaScript é uma

das três linguagens que o desenvolvedor web deve conhecer, pois JavaScript é a linguagem que especifica o comportamento das páginas na internet, as tornando dinâmicas.

Na programação, o JavaScript é uma linguagem dinâmica, portanto sua execução é interpretada por interpretadores em tempo de execução, sem necessariamente uma compilação prévia (CONTRIBUTORS, 2021 apud MORÓRÓ, 2024).

Na construção de um site, o JavaScript é representado pela tag “<script>”, geralmente localizada dentro da tag “<head>”, permitindo a associação de arquivos externos ao HTML, mas sempre apontam para arquivos JavaScript (PUREWAL, 2014).

2.2.7 React

React é um framework de JavaScript fundamentado pela Meta, sendo extremamente popular, de código aberto e frequentemente atualizada, seu principal foco é construir interfaces de usuário com atualizações rápidas e reativas (NEVES, 2023).

A construção de telas pelo React é feita com componentes e elementos, tendo uma segmentação maior do código. Suas interações são baseadas em estados e eventos, assim gerando códigos reativos e dinâmicos (AZZOLINI, 2021)

React é especificamente utilizado para criar interfaces de usuário com interações reativas e de atualização imediata, seu uso se tornou imensamente popular no desenvolvimento web, e, hoje, é um dos frameworks mais utilizados (SILVA; OLIVEIRA, 2023)

2.2.8 React Native

Embora React Native tenha suas semelhanças com React isso não descarta suas diferenças, a linguagem é executada com o JavaScript Core com uma ponte para tradução do código para renderização em qualquer sistema operacional mobile.

(SILVA; DE SOUZA, 2019)

Como apontado por Bruna Escudelario e Diego Pinho, React Native é um framework que permite aplicativos híbridos mobile sejam desenvolvidos com as mais sofisticadas ferramentas front-end, com uma compatibilidade com iOS e Android. (ESCUDELARIO; PINHO, 2021)

Segundo Franklyn Seabra Rogério Bezerra, React Native não funciona assim como um sistema web em WebView, e sim como um elemento nativo do sistema do

dispositivo, assim se desempenhando como um aplicativo na camada nativa. (BEZERRA, 2021)

2.2.9 Typescript

TypeScript é uma linguagem de programação desenvolvida com base no JavaScript padrão, adicionando recursos avançados para a linguagem. Foi projetada com o principal intuito de solucionar limitações que o JavaScript possui (MORÓRÓ, 2024).

De acordo com Francisco Souza, Edilson Lima e Elda Caridade (2022), a linguagem TypeScript é fortemente tipada e mais escalável que o JavaScript – muito utilizada por frameworks de FrontEnd – algo benéfico para desenvolvedores FullStack.

Afirma Jadson Faustino Moróró (2024) que a linguagem TypeScript é muito utilizada por sua sintaxe e estrutura de serem as similares as de JavaScript, mas com a adição de tipos estáticos opcionais.

2.2.10 Banco de dados

De acordo com C.J. Date (2004), um sistema de Banco de Dados é apenas uma coleção computadorizada de gerenciamento de registros, servindo como um repositório para arquivos de dados digitais.

Os bancos de dados possuem um grande impacto em áreas que utilizam de computadores, pois sua capacidade de abrigar dados de qualquer tamanho e complexidade torna os bancos benéficos ao trabalho (ELMASRI; NAVATHE, 2011).

O banco de dados é quem molda a interface, afirma Lev Manovich (2015); a estrutura de armazenamento de dados é quem dita como deverá ser o funcionamento de uma página, definindo os elementos que precisaram ser criados para aquela tela.

2.2.11 Firebase

“O Firebase é uma ferramenta que oferece serviços em nuvem para a construção completa de uma infraestrutura backend para aplicações mobile e web. (FIREBASE, 2021 apud RODRIGUES, 2021)”.

A utilização do Firebase é benéfica por trazer uma facilidade de entendimento e utilização, uma escalabilidade boa para a demanda, além de seus serviços oferecidos e possuir um desenvolvimento mais rápido (ALURA, 2024).

Entre os serviços oferecidos pelo Firebase, estão incluídos um serviço de banco de dados Cloud Firestore, autenticação de usuários, envio de notificações e armazenamento de arquivos, diz Amanda Anjolin Rodrigues (2021).

2.2.12 Construct

Como notado por Rodrigo Ribeiro Silva, Luis Rivero e Rodrigo Pereira dos Santos, Construct possui muitas vantagens para o desenvolvimento rápido de jogos, possui uma baixa complexidade, baixo custo, comunidade de suporte e flexibilidade nas plataformas onde o jogo pode ser disponibilizado. (SILVA; RIVERO; SANTOS, 2021).

Construct é um motor gráfico para jogos em duas dimensões baseado em HTML5, assim, criando a possibilidade de executá-lo em aplicações web, de forma nativa ou por navegadores. (CAPTERRA, 2021 apud CARVALHO; ARANHA; CARDIA NETO, 2022).

De muitas ferramentas de desenvolvimento de jogos Construct mostra uma simplicidade e didática em sua construção, assim, facilitando o seu desenvolvimento e implementação de ideias no desenvolvimento de jogos. (ALVES; AZEVEDO; COSTA JUNIOR, 2019).

As ferramentas e tecnologias do Construct se mostram mais eficazes em prototipação e facilidade de desenvolvimento em comparação a similares, tendo recursos mais adequados para uma equipe onde nem todos possuem o mesmo conhecimento técnico. (CARVALHO; ARANHA; CARDIA NETO, 2022).

2.2.13 C++

A linguagem de programação C, a qual serviu de base para a construção da linguagem C++, foi inicialmente criada para conseguir ser interpretada por máquinas de forma eficiente e rápida. (HORSTMANN, 2008)

Como descrito por Diego Rodrigues, C++ surgiu como uma extensão da linguagem C, incluindo estruturas de programação orientada a objeto junto com o controle de baixo nível do C, formando assim uma linguagem muito versátil e poderosa. (RODRIGUES, 2024)

Conforme Chen (2010 apud MARTINS, 2013) e Prechelt (2000, p. 29 apud MARTINS, 2013), concluiu-se que as linguagens C e C++ desempenham significativamente

melhor em tempo real e economizam memória quando comparadas com outras linguagens.

Abaixo você pode ver um código em C++, criado para ligar e desligar um LED ao apertar de um botão, a estrutura utilizada é específica para funcionar em microcontroladores.

Figura 8 - Exemplo de código em C++

```

1  const int botao = 2; //Define a porta para o botão
2  const int led = 13; //Define a porta para o LED
3
4  int estadoDoBotao = 0;
5
6  void setup() {
7      pinMode(botao, INPUT);
8      pinMode(led, OUTPUT);
9  }
10
11 void loop() {
12     estadoDoBotao = digitalRead(botao);
13
14     if (estadoDoBotao == HIGH) {
15         digitalWrite(led, HIGH); //Liga o LED
16     }
17     else {
18         digitalWrite(led, LOW); //Desliga o LED
19     }
20 }

```

Fonte: Autoria Própria, 2025

1ª Linha: É declarada uma constante de tipo inteiro armazenando o valor 2, de nome “botao”, esta linha se refere à porta do microcontrolador onde o botão está conectado.

2ª Linha: É declarada uma constante de tipo inteiro armazenando o valor 13, de nome “led”, esta linha se refere à porta do microcontrolador onde o LED está conectado. 4ª

Linha: É declarada uma variável de tipo inteiro inicializada com o valor 0, de nome “estadoDoBotao”, esta linha receberá o estado atual do botão.

6ª Linha: É criado uma função de nome “setup”, esta função é imediatamente executada ao código ser interpretado pela máquina.

7ª Linha: A porta que o botão está conectado é definido como uma entrada.

8ª Linha: A porta que o LED está conectado é definido como uma saída.

11ª Linha: É definida uma função de nome “loop”, esta função sera executada repetidamente e indeterminadamente.

12ª Linha: A variável “estadoDoBotao” recebe o valor do botão.

14ª Linha: É criado um operador lógico “*if*”, o código em seu interior será executado apenas se o botão estiver ligado.

15ª Linha: Liga o LED.

17ª Linha: É criado um operador lógico “*else*”, o código em seu interior será executado apenas se as condições para o último operador lógico “*if*” não forem atingidas.

18ª Linha: Desliga o LED.

2.2.14 Internet of Things (IOT)

A internet das coisas – é um conceito, onde dispositivos físicos, por meio de sensores, se conectam à internet – sendo um facilitador no cotidiano atual, defende Eduardo.

(2018)

Segundo Corneto e Márcia (2024), a junção vinculada ao termo IoT, interage com o ambiente e com outros dispositivos, podendo coletar uma gama de dados, o que serve como base e meio para a criação de outros sistemas.

2.2.15 ESP32

CPU (Unidade Central de Processamento) e memória em uma pequena placa de circuito impresso que contém diversas capacidades de comunicação sem fio. (BERTOLETI, 2019).

Segundo José V. S. Moraes, ESP32 veio como uma versão mais poderosa do microcontrolador, já consolidado no mercado, ESP8266, tendo dois núcleos de processamento, mais periféricos e Bluetooth, continuando com um preço competitivo. (MORAIS, 2023).

De acordo com Allyson Nascimento, o microcontrolador ESP32, especialmente ESPWROOM-32, possui suporte para Wi-Fi, BLE (Bluetooth Low Energy) e Ethernet, além de possuir grande eficiência energética. (NASCIMENTO et al (2023).

2.2.16 Modelagem 3D

A modelagem, como explicado por Priscila Argoud (2024) – é uma técnica de recriar o mundo real de forma virtual, criando objetos tridimensionais (3D) nos meios digitais mimetizando a realidade – aplicando conceitos físicos e geoespaciais.

Contudo, por meio da modelagem se busca o detalhamento durante a construção, possibilitando uma imersão ao usuário – trazendo o virtual para algo mais concreto e agradável. (VILELA; LOPES; LIMA, 2015).

2.2.17 Impressão 3D

Como dito por Moisés Miranda e Gustavo Henrique (2020), a impressão tridimensional – é a capacidade de modelar fisicamente, objetos e seres inanimados, de diversos tipos – sejam eles detalhados, simples, robustos, grandes ou pequenos.

É explicado por Silva *et. al* (2020), que o processo de impressão é feito em camadas – mais especificamente fatias baseadas no formato do modelo que está sendo impresso – que são adicionadas uma junta da outra, criando espessura e relevo.

Afirma Leonardo Gomez, que a impressão 3D, atraí alta atenção de quem a manuseia, apesar de complexas, ao transformar modelos escaneados em atrações físicas, estimula bastante a criatividade e pensamento livre.

3 DESENVOLVIMENTO

4 CONSIDERAÇÕES FINAIS

REFERÊNCIAS

ACETI, L. B.; BRITO, L. T.; OLIVEIRA, D. C. M.; PINA, D. B.; KUNSTMANN, L. N. O. Desenvolvimento de uma Aplicação Web para Instrumentação de Scripts para a Ferramenta DfAnalyzer. 2023. Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) - Universidade Federal Fluminense, Niterói, 2023.

ALURA. 2024. O que é Firebase? Para que serve, principais característica e um Guia dessa ferramenta Google. Disponível em: <https://www.alura.com.br/artigos/firebase?srsItid=AfmBOopem7O3TvcbgX7pGKqJmj6aesJI7WIJtDFoEtQfEpk0PCVOWqyC>. Acesso em: 28 Abril, 2025.

DATE, C. J. Tradução de Daniel Vieira. Introdução a sistemas de bancos de dados. Rio de Janeiro, RJ: Elsevier Brasil, 2004.

Acesso em: 28 de Abril, 2025

ANACLETO CHICCA JUNIOR, N.; GOMEZ CASTILLO, L. Os desafios em utilizar a impressão 3D no processo ensino-aprendizagem de design. RENOTE, Porto Alegre, v. 16, n. 1, 2018. DOI: 10.22456/1679-1916.86040. Disponível em: <https://seer.ufrgs.br/index.php/renote/article/view/86040>. Acesso em: 29 abr. 2025.

FROBOESE DA SILVA, Eloar; MENIN DA SILVA, Lucas; GONÇALVES DEON, Vinícius; TOSO, Marcelo André. IMPRESSÃO 3D APLICADA À TECNOLOGIA ASSISTIVA. Revista Destaques Acadêmicos, [S. l.], v. 12, n. 4, 2020. DOI: 10.22410/issn.2176-3070.v12i4a2020.2657. Disponível em: <https://www.univates.br/revistas/index.php/destaques/article/view/2657>. Acesso em: 29 abr. 2025.

ARGOUD, P. Modelagem 3D. São Paulo, SP: SENAC, 2024.
<https://play.google.com/books/reader?id=iCgGEQAAQBAJ&pg=GBS.PP1&hl=ptBR> .

BERTOLETI, P. Projetos com ESP32 e LoRa. São Paulo: INSTITUTO NEWTON C. BRAGA., 2019.

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. UML: guia do usuário. 2.ed. Rio de Janeiro: Elsevier, 2006.

ELMASRI, Ramez; NAVATHE, Shamkant B.. SISTEMAS DE BANCO DE DADOS. 6. ed. São Paulo, SP: Addison Wesley, 2011.

EM. Uso excessivo de telas pode causar demência digital, diz neurocirurgião. 6 out. 2023. Disponível em: (https://www.em.com.br/app/noticia/saude-e-bem-viver/2023/10/06/interna_bem_viver,1572765/uso-excessivo-de-telas-pode-causar-demencia-digital-diz-neurocirurgiao.shtmlhttps://www.em.com.br/app/noticia/saude-e-bem-viver/2023/10/06/interna_bem_viver,1572765/uso-excessivo-de-telas-pode-causar-demencia-digital-diz-neurocirurgiao.shtmlcausar-demencia-digital-diz-neurocirurgiao.shtml). Acesso em: 19 abr. 2025.

FEDERAL DE SANTA, U. et al. Revista eletrônica de biblioteconomia e ciência da informação: O MÉTODO DESIGN THINKING E A PESQUISA EM CIÊNCIA DA INFORMAÇÃO Encontros Bibli: revista eletrônica de biblioteconomia e ciência da informação. v. 27, p. 1–21, 2022.

FLANAGAN, D. Tradução de João Eduardo Nóbrega Tortello. JavaScript: o Guia Definitivo - 6° ed. Porto Alegre, RS: Bookman, 2013.

FLATSCHART, Fábio. HTML 5 - Embarque Imediato. Rio de Janeiro, RJ: Brasport, 2011.

Acesso em: 22 de Abril, 2025.

FOWLER, Martin. UML essencial : um breve guia para a linguagem-padrão de modelagem de objetos. 3. ed. Porto Alegre: Bookman, 2005.

FREIRE, Paulo. Pedagogia da autonomia: saberes necessários à prática educativa. São Paulo: Paz e Terra, 1996.

FROBOESE DA SILVA, Eloar; MENIN DA SILVA, Lucas; GONÇALVES DEON, Vinícius; TOSO, Marcelo André. IMPRESSÃO 3D APLICADA À TECNOLOGIA ASSISTIVA. Revista Destaques Acadêmicos, [S. l.], v. 12, n. 4, 2020. DOI: 10.22410/issn.2176-3070.v12i4a2020.2657. Disponível em: <https://www.univates.br/revistas/index.php/destaques/article/view/2657>. Acesso em: 29 abr. 2025.

GUEDES, Gilleanes T. A. UML 2: uma abordagem prática. 3.ed. São Paulo: Novatec, 2009.

G1. 'Cérebro podre': entenda o que é 'brain rot', expressão do ano eleita pelo Dicionário Oxford. G1 Educação, 02 dez. 2024. Disponível em: <https://g1.globo.com/educacao/noticia/2024/12/02/entenda-o-que-e-brain-rothttps://g1.globo.com/educacao/noticia/2024/12/02/entenda-o-que-e-brain-rot-cerebro-podre-expressao-do-ano-eleita-pelo-dicionario-oxford.ghtml>

podre-expressao-do-ano-eleita-pelo-dicionario-oxford.ghhtml . Acesso em: 12 maio. 2025.

HORSTMANN, Cay. Conceitos de computação com o essencial de C++.

Tradução de Carlos Arthur Lang Lisbôa e Mária Lúcia Blanck Lisbôa. 3ªed.

Porto Alegre, RS: ARTMED, p. 30- 2008. Disponível em: https://play.google.com/store/books/details?id=qSn8wEMsolMC&rdid=bookhttps://play.google.com/store/books/details?id=qSn8wEMsolMC&rdid=book-qSn8wEMsolMC&rdot=1&source=gbs_vpt_read&pcampaignid=books_booksearch_viewportqSn8wEMsolMC&rdot=1&source=gbs_vpt_read&pcampaignid=books_booksearch_viewport. Acesso em 26 Abr. 2025.

JOBSTRAIBIZER, Flávia. Criação de Sites CSS. São Paulo, SP: Digerati Books, 2009.

Acesso em: 27 de Abril, 2025.

LARMAN, Craig. Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objeto. Porto Alegre: Bookman, 2000.

MAGRANI, E. A internet das coisas. Rio de Janeiro: FGV Editora, 2018. 192 p. ISBN 978-85-225-2005-3.

MANOVICH, L. Banco de Dados. Revista Eco-Pós, [S. l.], v. 18, n. 1, p. 7–26, 2015. DOI: 10.29146/eco-pos.v18i1.2366. Disponível em: https://ecopos.emnuvens.com.br/eco_pos/article/view/2366. Acesso em: 28 abr. 2025.

MARQUES, P. B.; CASTANHO, M. I. S. O que é a escola a partir do sentido construído por alunos. Psicologia Escolar e Educacional, v. 15, n. 1, jun. 2011. DOI: (<https://doi.org/10.1590/S1413-85572011000100003>).

MARTINS, Mateus Rodrigues. Desenvolvimento de uma biblioteca em C++ para modelagem e controle cinemático de robôs manipuladores usando quatérnios duais. 2013. Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Controle e Automação) — Universidade Federal de Minas Gerais, Belo Horizonte, 2013.

MILETTO, E. M.; DE CASTRO BERTAGNOLLI, S. Desenvolvimento de Software II: Introdução ao Desenvolvimento Web com HTML, CSS, JavaScript e PHP - Eixo: Informação e Comunicação - Série Tekne. Porto Alegre, RS: Bookman Editora, 2014.

MIRANDA MORANDINI, M.; DEL VECHIO, G. H. IMPRESSÃO 3D, TIPOS E POSSIBILIDADES: uma revisão de suas características, processos, usos e tendências. Revista Interface Tecnológica, Taquaritinga, SP, v. 17, n. 2, p. 67– 77, 2020. DOI: 10.31510/infa.v17i2.866. Disponível em:

<https://revista.fatectq.edu.br/interfacetecnologica/article/view/866>. Acesso em: 29 abr. 2025.

MORAIS, J. V. S. ESP32 com IDF. São Paulo: Editora NCB, 2023.

MORORÓ, J. F. Um estudo comparativo entre JavaScript e TypeScript. 2024. 154 f. Trabalho de Conclusão de Curso (Graduação em Engenharia de Computação) – Campus de Sobral, Universidade Federal do Ceará, Sobral, 2024. Acesso em: 28 de Abril, 2025.

NASCIMENTO, A.; SOUSA, C.; OLIVEIRA, G.; OLIVEIRA, F.; CAMPOS, G. Controle de Iluminação Através da Internet Utilizando as Tecnologias ReactJS, Firebase e ESP32. Anais do Encontro de Computação do Oeste Potiguar ECOP/UFERSA (ISSN 2526-7574), [S. l.], v. 1, n. 6, p. 46–49, 2023. Disponível em: <https://periodicos.ufersa.edu.br/ecop/article/view/11835>. Acesso em: 26 abr. 2025.

PAGANI, T. Design thinking. [s.l.] SENAC, p. 3–9, 2018. Disponível em: <https://play.google.com/store/books/details?id=Iz1MDwAAQBAJ&rdid=bookhttp>

s://play.google.com/store/books/details?id=lz1MDwAAQBAJ&rdid=book-lz1MDwAAQBAJ&rdot=1&source=gbs_vpt_read&pcampaignid=books_booksearch_viewportlz1MDwAAQBAJ&rdot=1&source=gbs_vpt_read&pcampaignid=books_booksearch_viewport . Acesso em: 17 maio. 2025.

PARISOTTO, A. L. V.; RINALDI, R. P. Ensino de língua materna: dificuldades e necessidades formativas apontadas por professores na Educação Fundamental. Educação e Revista, n. 60, abr./jun. 2016. DOI: (https://doi.org/10.1590/0104-4060.45125).

PAULA, B. H. de. Jogos digitais como artefatos pedagógicos: o desenvolvimento de jogos digitais como estratégia educacional. 2015. Dissertação (Mestrado em Artes Visuais) - Instituto de Artes, Universidade Estadual de Campinas, Campinas, 2015.

PUREWAL, S. Tradução de Lúcia Kinoshita. Aprendendo a desenvolver aplicações web: Desenvolva rapidamente com as tecnologias JavaScript mais modernas. São Paulo, SP: Novatec Editora, 2014.

Acesso em: 28 de Abril, 2025.

RODRIGUES, Amanda Anjolin. Desenvolvimento de aplicativo Android utilizando os serviços de firebase, 2021. Trabalho de conclusão de curso (Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas) Faculdade de Tecnologia de São Paulo, São Paulo, SP, 2021.

Acesso em: 28 de Abril, 2025.

RODRIGUES, D. APRENDA C++ MODERNO: Desenvolva Aplicações Performáticas com Recursos Avançados. Dos Fundamentos às Aplicações Práticas. [s.l.] StudioD21, 2024.

SCHEIDT, F. A. Fundamentos de CSS: criando design para sistemas web. Foz do Iguaçu, PR: [s.n.], 2015. Acesso em: 27 de Abril, 2025.

SENA, S. de; SCHMIEGELOW, S. S.; PRADO, G. M. B. C. do; PERASSI, R.; FIALHO, F. A. P. Aprendizagem baseada em jogos digitais: a contribuição dos jogos epistêmicos na geração de novos conhecimentos. CINTED-UFRGS Novas Tecnologias na Educação, v. 14, n. 1, jul. 2016.

SILVA, Eduardo Corneto; ESPEJO, Márcia M. S. B. Adoção da Internet das Coisas (IoT) na agropecuária: uma revisão sistemática sobre as possibilidades de adoção no ambiente produtivo rural brasileiro. Interações, Campo Grande, v. 25, n. 4, 2024. DOI: (<https://doi.org/10.20435/inter.v25i4.4024>).

SILVA, Maurício Samy. HTML 5: A LINGUAGEM DE MARCAÇÃO QUE REVOLUCIONOU A WEB. São Paulo, SP: Novatec Editora, 2011. Acesso em: 22 de Abril, 2025.

SOUZA, Francisco Moreira Calado; LIMA, Edilson Carlos Silva; CARIDADE, Elda Regina de Sena. CRIANDO SISTEMA ESCALÁVEL DE AGENDAMENTOS UTILIZANDO TYPESCRIPT COM NESTJS NO BACKEND E NEXTJS NO FRONTEND . Revista Ibero-Americana de Humanidades, Ciências e Educação, [S. l.], v. 8, n. 12, p. 43–57, 2022. DOI: 10.51891/rease.v8i12.7986. Disponível em: <https://periodicorease.pro.br/rease/article/view/7986>. Acesso em: 29 abr. 2025.

VASCONCELLOS, M. S. de; CARVALHO, F. G. de; BARRETO, J. O.; ATELLA, G. C. As Várias Faces dos Jogos Digitais na Educação. Informática na Educação: teoria & prática, Porto Alegre, v. 20, n. 4, p. 203-218, ago. 2017.

VIANNA, M. et al. Design Thinking: Inovação em Negócios. 1. ed. Rio de Janeiro, RJ: MJV Press, 2012.

VILELA, João Paulo; LOPES, Ricardo; LIMA, Fernando. Modelagem 3d de edifícios históricos: a influência do lod no processo de reconstrução virtual. 2020. Trabalho de Graduação (Graduação em Engenharia Civil) - Universidade Estadual Paulista, Guaratinguetá, 2015.

