

Universidade de Brasília – UnB
Faculdade UnB Gama – FGA
Engenharia de Software

Enpyre: uma plataforma web interativa e lúdica para o ensino de Python

Autores: Lucas A. de Victor Coutinho e Vitor N. A. Ribeiro
Orientador: Prof. Dr. Wander C. M. Pereira da Silva

Brasília, DF
2023



Lucas A. de Victor Coutinho e Vitor N. A. Ribeiro

Enpyre: uma plataforma web interativa e lúdica para o ensino de Python

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Universidade de Brasília – UnB

Faculdade UnB Gama – FGA

Orientador: Prof. Dr. Wander C. M. Pereira da Silva

Brasília, DF

2023

Lucas A. de Victor Coutinho e Vitor N. A. Ribeiro
Enpyre: uma plataforma web interativa e lúdica para o ensino de Python/
Lucas A. de Victor Coutinho e Vitor N. A. Ribeiro. – Brasília, DF, 2023-
54 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Wander C. M. Pereira da Silva

Trabalho de Conclusão de Curso – Universidade de Brasília – UnB
Faculdade UnB Gama – FGA , 2023.

1. Python. 2. Ensino. I. Prof. Dr. Wander C. M. Pereira da Silva. II.
Universidade de Brasília. III. Faculdade UnB Gama. IV. Enpyre: uma plataforma
web interativa e lúdica para o ensino de Python

CDU 02:141:005.6

Lucas A. de Victor Coutinho e Vitor N. A. Ribeiro

Enpyre: uma plataforma web interativa e lúdica para o ensino de Python

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Trabalho aprovado. Brasília, DF, 01 de junho de 2013 – Data da aprovação do trabalho:

**Prof. Dr. Wander C. M. Pereira da
Silva**
Orientador

Prof.^a Dr.^a Elaine Venson
Convidado 1

**Prof. Dr. John Lenon Cardoso
Gardenghi**
Convidado 2

Brasília, DF
2023

Agradecimentos

Os autores deste trabalho contaram com o auxílio de diversas pessoas.

Lucas Amoêdo:

À Juliana, que sempre esteve comigo.

Vitor Nere:

À minha esposa, Ana Paula, companheira de vida que me apoiou, não desistiu e me incentivou a não desistir nos momentos onde beirei a volta da depressão. À minha mãe, Maria Aparecida, que sempre me incentivou nos momentos difíceis e compreendeu a minha ausência enquanto eu me dedicava à realização da graduação. E à minha falecida avó, Maria do Socorro, que investiu nos meus estudos, tornando possível o meu ingresso na graduação em Engenharia de Software na FGA, além de sempre me apoiar e me incentivar a ser uma pessoa melhor.

Ambos:

Por fim, agradecemos ao professor orientador Wander Cleber, por emprestar um pouco de seu conhecimento e por viabilizar o sonho da formatura.

Resumo

Este trabalho apresenta uma proposta de ferramenta para minimizar as dificuldades de ensino e aprendizado de programação. A ferramenta é uma plataforma online que permite a execução de código Python diretamente no navegador, eliminando a necessidade de instalação de requisitos técnicos na máquina do aluno. Além disso, a plataforma oferece uma abordagem lúdica para o ensino de programação, através de quiz para fixação de conteúdo e da criação de jogos para exercitar a lógica de programação. A ferramenta foi desenvolvida utilizando a biblioteca Pyodide, que permite a execução de código Python no navegador através da tecnologia Web Assembly. A interface da ferramenta foi desenvolvida utilizando a biblioteca PixiJS, que permite a criação de jogos em 2D no navegador. A ferramenta foi desenvolvida como um Trabalho de Conclusão de Curso e está disponível em <https://enpyre-play.vercel.app/>.

Palavras-chave: Python; Ensino de Programação; Aprendizado de Programação; Web Assembly; PixiJS; Pyodide.

Abstract

This paper presents a proposal for a tool to minimize the difficulties of teaching and learning programming. The tool is an online platform that allows the execution of Python code directly in the browser, eliminating the need for technical requirements installation on the student's machine. In addition, the platform offers a playful approach to programming teaching, through quizzes to fix content and game creation to exercise programming logic. The tool was developed using the Pyodide library, which allows the execution of Python code in the browser using Web Assembly technology. The interface of the tool was developed using the PIXIJS library, which allows the creation of 2D games in the browser. The tool was developed as a Final Course Work and is available at <https://enpyre-play.vercel.app/>.

Key-words: Python; Programming Learning; Programming Teaching; Web Assembly; PIXIJS; Pyodide.

Lista de ilustrações

Figura 1 – Cronograma	22
Figura 2 – Backlog de desenvolvimento	23
Figura 3 – Representação de pacotes e tecnologias do servidor	27
Figura 4 – Representação de pacotes e tecnologias do cliente	28
Figura 5 – Modelo de dados do sistema	29
Figura 6 – Tela de boas-vindas	36
Figura 7 – Tela de lista de projetos	36
Figura 8 – Tela do projeto	37
Figura 9 – Tela de lista de quiz	37
Figura 10 – Tela do quiz	38
Figura 11 – Métricas de qualidade do código	40
Figura 12 – Visão Geral - Parte 1	50
Figura 13 – Visão Geral - Parte 2	51
Figura 14 – Move Ball Game	52

Sumário

1	INTRODUÇÃO	11
2	REFERENCIAL TEÓRICO	12
2.1	Programação	12
2.2	Linguagens de Programação	12
2.3	Métodos de Ensino de Programação	13
2.3.1	Escolha da linguagem de programação	13
2.3.2	Abordagem Instrucionista	13
2.3.3	Abordagem Construtivista	14
2.3.4	Outros Métodos	14
2.4	Dificuldades no Aprendizado	14
2.5	Ensino com Jogos	15
2.6	Plataformas existentes	16
2.6.1	Code.org	16
2.6.2	Scratch	16
2.6.3	Blockly	16
2.6.4	CodeCombat	16
2.6.5	Outros	17
3	METODOLOGIA	18
3.1	Metodologia de Desenvolvimento de Software	18
3.1.1	Metodologia Ágil	18
3.1.1.1	Scrum	18
3.1.1.2	Kanban	18
3.1.2	Metodologia de Desenvolvimento Utilizada	19
3.2	Ferramentas Utilizadas	19
3.2.1	Gerenciamento do Projeto	19
3.2.1.1	Trello	19
3.2.1.2	Slack	19
3.2.1.3	Google Drive	19
3.2.2	Desenvolvimento	20
3.2.2.1	Figma	20
3.2.2.2	GitHub e GitHub Actions	20
3.2.2.3	Docker	20
3.2.2.4	Visual Studio Code	21
3.3	Processo de Desenvolvimento	21

3.3.1	Cronograma	21
3.4	Cronograma	21
3.4.1	Backlog de desenvolvimento	22
4	SOLUÇÃO DESENVOLVIDA	24
4.1	Visão Geral	24
4.2	Requisitos	24
4.3	A Distribuição do Software	25
4.4	Arquitetura	26
4.4.1	Servidor	26
4.4.2	Cliente	27
4.4.3	Modelo de Dados	28
4.4.4	Configuração do Ambiente de Produção	30
4.4.4.1	Servidor	30
4.4.4.2	Cliente	31
4.4.5	Configuração do Ambiente de Desenvolvimento	32
4.4.5.1	Servidor	32
4.4.5.2	Cliente	34
4.5	A Ferramenta	35
4.5.1	Disponibilidade	35
4.5.2	Tela de boas-vindas	36
4.5.3	Tela de lista projetos	36
4.5.4	Tela de quiz	37
4.6	Métricas de Qualidade	38
5	CONCLUSÃO	41
	REFERÊNCIAS	42
	APÊNDICES	44
	APÊNDICE A – REFERENCIAL TECNOLÓGICO DA POC	45
A.1	WebAssembly	45
A.1.1	Definição	45
A.1.2	Vantagens	45
A.1.3	Limitações	45
A.2	Pyodide	46
A.2.1	Definição	46
A.2.2	Vantagens	46

A.2.3	Limitações	46
A.3	PixiJS	46
A.3.1	Definição	46
A.3.2	Vantagens	47
A.3.3	Limitações	47
A.3.4	Outras Tecnologias	47
A.3.4.1	Canvas API	47
A.3.4.2	WebGL API	47
A.3.4.3	HTML Canvas, DOM, Event, Document	47
	APÊNDICE B – ARQUITETURA DA PROVA DE CONCEITO	49
B.1	Visão Geral	49
B.2	Arquitetura	49
B.2.1	Camada Browser	49
B.2.1.1	Camada HTML	49
B.2.2	Camada JavaScript	49
B.2.3	Camada Python	49
	APÊNDICE C – RESULTADOS DA PROVA DE CONCEITO	52
C.1	Aparencia	53
C.2	Nome da Ferramenta: Enpyre	53
C.3	Estrutura Visual	53
C.4	Resultados	53

1 Introdução

A programação tem se tornado cada vez mais essencial no mundo moderno, tanto no âmbito profissional quanto pessoal. No entanto, aprender a programar pode ser um desafio para muitas pessoas, especialmente para iniciantes que não possuem conhecimentos prévios sobre o assunto. Uma abordagem promissora para tornar o aprendizado de programação mais fácil e envolvente é a utilização de jogos, que permitem aos usuários aprender conceitos de programação em um contexto lúdico e visual.

A linguagem Python tem se tornado cada vez mais popular como uma linguagem de programação para iniciantes, devido à sua simplicidade e facilidade de uso. A combinação do Python com uma abordagem baseada em jogos pode ser particularmente eficaz para o ensino de programação para jovens estudantes e adultos iniciantes. A aprendizagem baseada em jogos pode aumentar a motivação e a retenção do conhecimento, tornando o processo de aprendizado mais envolvente e eficaz.

O estudo de [Pereira e Souza \(2020\)](#) levantou que “as ferramentas para programação estão entre os três maiores problemas relacionados ao ensino e aprendizagem do assunto”. Além disso, “várias ferramentas e estratégias de ensino são implementadas na preparação de estudantes de programação, por meio de contos narrativos, jogos, simulações e técnicas de visualização”.

Com isto posto, este trabalho tem por objetivo propor uma ferramenta que asmitigue parte das dificuldades no ensino e aprendizagem de programação. A ferramenta proposta deve ser capaz de auxiliar no conteúdo teórico e possuir interface gráfica com elementos interativos relacionados a jogos. O teste da ferramenta em ambiente de aprendizado real está além do escopo deste trabalho.

2 Referencial Teórico

2.1 Programação

Programação é uma das habilidades fundamentais do mundo moderno, com aplicações em diversas áreas, incluindo ciência, tecnologia e negócios. Em termos simples, a programação envolve a criação de algoritmos que orientam um computador a executar uma tarefa específica.

De acordo com [Roy e Haridi \(2004\)](#), programação pode ser definida como “o processo de especificar um algoritmo computacional para resolver um problema”. Um algoritmo é uma sequência lógica de instruções que descrevem como um problema deve ser resolvido. Essas instruções são escritas em uma linguagem de programação que é compreensível tanto para humanos quanto para computadores.

A programação é uma habilidade valiosa para o mundo moderno, pois permite que as pessoas criem soluções personalizadas para problemas específicos. Ela é utilizada em uma ampla variedade de aplicações, incluindo desenvolvimento de software, análise de dados, inteligência artificial, jogos, aplicações móveis e muito mais. Além disso, a programação ajuda a desenvolver habilidades importantes, como resolução de problemas, pensamento lógico e criatividade.

2.2 Linguagens de Programação

Uma linguagem de programação é uma linguagem formal que permite a um programador expressar instruções precisas para um computador ou outro dispositivo. As linguagens de programação são usadas para desenvolver software, aplicativos e sistemas, e são criadas com base em uma série de regras sintáticas e semânticas que permitem que os programadores criem códigos que possam ser entendidos e executados pelo computador. [Goodrich \(2013\)](#) define linguagem de programação como “uma notação para descrever algoritmos e estruturas de dados”, dizendo também que são “uma coleção de símbolos, palavras-chave e regras sintáticas que definem como esses símbolos podem ser combinados para criar programas de computador”.

Uma das barreiras no aprendizado de programação é a sintaxe da linguagem escolhida para estudo. [Stefik e Siebert \(2013\)](#) concluiu em sua investigação empírica sobre sintaxes de linguagem de programação que linguagens com sintaxe derivada ou semelhante a C, como C++ e Java, podem apresentar aspectos que dificultam o aprendizado de novos programadores, como a necessidade de inicializar e declarar o tipo de variáveis, necessidade

de se usar chaves, ponto e vírgula e expressões verbosas (MANNILA; PELTOMAKI; SALAKOSKI, 2006).

Python é uma linguagem de programação que abandonou as familiaridades de sintaxe com C (STEFIK; SIEBERT, 2013), optando por usar a indentação ao invés de chaves para delimitar blocos de código. Python também foi alvo de um estudo por Jayal et al. (2015) acerca de seu uso no ensino de programação introdutória, onde se concluiu que houveram melhores resultados no aprendizado quando os conceitos introdutórios (fluxos de controle e uso de bibliotecas, por exemplo) foram ensinados com Python, em comparação ao Java.

Ao adotar a indentação como elemento de sintaxe, Python tornou seus programas mais legíveis e fáceis de compreender. Além disso, é multiplataforma, possuindo suporte para sistemas como Windows, Linux, MacOS, etc; e possui módulos prontos em sua biblioteca principal para realização de várias tarefas comuns de programação (MORAIS; PIRES, 2002).

De acordo com Fangohr (2004), esta linguagem é indicada para disciplinas introdutórias e seu ambiente auxilia no aprendizado prático, seja na modalidade à distância ou em laboratórios.

2.3 Métodos de Ensino de Programação

Silva e Aranha (2014) realizaram uma revisão sistemática da literatura que identificou diversas abordagens e estratégias utilizadas para o ensino de programação.

2.3.1 Escolha da linguagem de programação

A maioria dos estudos analisados por Silva e Aranha (2014) utilizou linguagens de programação específicas para o ensino de programação, como Scratch, Alice e Python. Essas linguagens são escolhidas por serem simples e intuitivas, permitindo que alunos iniciantes possam aprender a programar de maneira mais fácil e rápida. Além disso, elas são visualmente atrativas, o que torna o processo de aprendizado mais lúdico e divertido.

2.3.2 Abordagem Instrucionista

A abordagem instrucionista, baseada em aulas expositivas e resolução de exercícios, ainda é a mais utilizada pelos professores brasileiros. De acordo com Silva e Santos (2018), essa abordagem é criticada por não incentivar a criatividade e a autonomia dos alunos, limitando-se apenas à reprodução de exemplos dados em sala de aula. No entanto, ela ainda é utilizada por ser considerada uma forma simples e eficaz de transmitir o conhecimento.

2.3.3 Abordagem Construtivista

A abordagem construtivista, na qual os alunos são encorajados a resolver problemas reais e a trabalhar em projetos de programação que envolvam outras áreas do conhecimento, é vista como uma alternativa à abordagem tradicional. Segundo [Silva e Santos \(2018\)](#), essa abordagem permite que os alunos desenvolvam habilidades de pensamento crítico e de trabalho em equipe, além de estimular a criatividade e a autonomia.

2.3.4 Outros Métodos

Além das abordagens mencionadas anteriormente, [Silva e Santos \(2018\)](#) identificaram outros métodos de ensino de programação utilizados no Brasil, como a utilização de jogos educativos, a programação por blocos e a programação por agentes. Esses métodos são considerados alternativas às abordagens mais tradicionais e podem ser utilizados para tornar o processo de aprendizado de programação mais atrativo e dinâmico.

2.4 Dificuldades no Aprendizado

O ensino e aprendizado de programação apresentam diversos desafios e dificuldades. Dentre eles, destaca-se a abstração, que pode ser considerada como uma das principais barreiras para a aprendizagem de programação ([SILVA; ARANHA, 2014](#)). De acordo com os autores, a habilidade de abstração é fundamental para o desenvolvimento de programas eficientes e de alta qualidade. Entretanto, a capacidade de abstrair conceitos complexos não é inata e pode ser um desafio para muitos estudantes.

Outra dificuldade importante no ensino e aprendizado de programação é a falta de motivação dos estudantes ([SILVA; ARANHA, 2014](#)). Muitos alunos não conseguem visualizar a utilidade da programação em suas vidas e, portanto, não se sentem motivados a aprender. Segundo os autores, é importante que os professores expliquem a relevância e a aplicabilidade da programação para ajudar os alunos a se engajarem com o conteúdo.

Além disso, a falta de experiência prévia em lógica de programação pode dificultar a compreensão dos conceitos ([SILVA; ARANHA, 2014](#)). Para muitos estudantes, a lógica por trás da programação pode não ser óbvia e pode exigir uma compreensão abstrata. Nesse sentido, é importante que os professores trabalhem com exemplos e exercícios que possam ajudar os alunos a compreender os conceitos fundamentais.

A complexidade das linguagens de programação, especialmente para iniciantes, também pode representar um desafio significativo ([SILVA; ARANHA, 2014](#)). As linguagens de programação podem incluir uma ampla variedade de recursos e comandos, e a compreensão de sua sintaxe pode ser difícil para quem está começando. Para contornar

essa dificuldade, é recomendado que os professores utilizem linguagens de programação mais simples e adequadas para iniciantes.

2.5 Ensino com Jogos

De acordo com [Moreira \(2018\)](#), os jogos digitais podem ser uma excelente forma de engajar os alunos no aprendizado de programação. Diferentemente dos métodos tradicionais de ensino, que muitas vezes são enfadonhos e pouco interativos, os jogos digitais podem tornar o processo de aprendizado mais dinâmico e divertido, além de estimular o pensamento crítico e o raciocínio lógico dos alunos.

Ao mesmo tempo, a construção de jogos é uma abordagem interessante para o ensino de programação, uma vez que permite que os alunos apliquem seus conhecimentos na prática, criando seus próprios jogos. De acordo com o artigo “O Uso de Jogos Digitais para o Ensino de Programação” [Moreira \(2018\)](#), a construção de jogos é uma forma atraente e motivadora para ensinar programação aos alunos, pois permite que eles criem algo concreto e apliquem seus conhecimentos de programação na prática.

Para ensinar programação através da construção de jogos, é importante que os alunos tenham um conhecimento prévio de conceitos fundamentais de programação, como variáveis, loops e condicionais, para que possam aplicá-los na criação de seus jogos. O artigo “Jogos Digitais como Ferramenta de Apoio ao Ensino de Programação” [Pereira \(2016\)](#) destaca a importância do uso de jogos na fase inicial de aprendizado da programação, uma vez que o ato de jogar pode ajudar os alunos a compreender melhor a lógica de programação e a entender como os conceitos são aplicados na prática.

Já o ato de jogar pode ser utilizado como uma ferramenta complementar ao ensino de programação. Conforme destacado por [Pereira \(2016\)](#), jogar jogos pode ajudar os alunos a compreender melhor a lógica de programação e a entender como os conceitos são aplicados na prática. Além disso, jogar jogos pode ser uma forma interessante de estimular o interesse dos alunos pela programação e de motivá-los a aprender mais sobre o tema.

Para utilizar o ato de jogar como ferramenta pedagógica, é importante escolher jogos adequados e que possam ser utilizados para ensinar conceitos de programação. Por exemplo, jogos de raciocínio lógico e de resolução de problemas podem ser utilizados para ensinar conceitos como condicionais, loops e variáveis, enquanto jogos de plataforma podem ser utilizados para ensinar conceitos de física e mecânica de movimento.

2.6 Plataformas existentes

2.6.1 Code.org

Um exemplo de jogo digital que pode ser utilizado para o ensino de programação é o [Code.org](#), uma plataforma que oferece uma série de jogos e atividades voltados para o aprendizado da lógica de programação. Os jogos do Code.org são projetados para incentivar os alunos a resolver problemas de programação de uma forma mais criativa e interativa, ajudando-os a desenvolver habilidades de programação mais avançadas ao longo do tempo.

2.6.2 Scratch

Outra plataforma muito utilizada no ensino de programação é o [Scratch](#), um software gratuito desenvolvido pelo MIT Media Lab. O Scratch é um ambiente de programação visual que permite aos alunos criar seus próprios jogos, histórias e animações. Os alunos podem arrastar e soltar blocos de programação para criar suas próprias sequências de comandos, sem precisar digitar códigos complexos. Essa abordagem torna o processo de aprendizado mais acessível para alunos iniciantes e ajuda a estimular a criatividade e o pensamento lógico.

2.6.3 Blockly

O [Blockly](#) é outro exemplo de plataforma de programação visual que pode ser utilizada para o ensino de programação. Ele permite que os alunos criem seus próprios programas, usando blocos de código que podem ser facilmente arrastados e soltos para criar uma sequência de comandos. O Blockly é um ambiente de programação flexível, que pode ser usado para criar jogos, animações, aplicativos e muito mais.

2.6.4 CodeCombat

A [CodeCombat](#) é uma plataforma de aprendizado de programação que ensina programação através de jogos. Os jogadores são desafiados a escrever código para controlar seus personagens e completar os níveis do jogo. A CodeCombat suporta várias linguagens de programação, incluindo Python, JavaScript, Java, Lua e CoffeeScript, e tem conteúdo adequado para jogadores de diferentes níveis de habilidade. A plataforma é usada por escolas, professores e alunos, bem como por programadores iniciantes e experientes que desejam melhorar suas habilidades de programação de forma divertida e interativa.

2.6.5 Outros

Além desses exemplos, há uma grande variedade de jogos digitais que podem ser utilizados no ensino de programação. Jogos como o [Kodu](#), [Minecraft](#), [Lightbot](#), entre outros, podem ser usados como ferramentas pedagógicas para o ensino de programação em diferentes níveis e contextos educacionais.

As plataformas existentes cumprem o propósito de apresentar a lógica de programação ao aluno, mas não necessariamente através de uma linguagem de programação específica. A Code.org, o Scratch e o Blockly, por exemplo, apresentam contato mínimo com uma linguagem de programação real, focando suas atividades de resolução de problemas em uma abordagem de arrastar blocos.

A ferramenta CodeCombat, diferentemente, traz o contato direto com uma linguagem de programação, mas é uma plataforma com conteúdo pago.

3 Metodologia

3.1 Metodologia de Desenvolvimento de Software

Um processo ou metodologia de desenvolvimento de software é uma coleção de atividades e resultados relacionados que auxiliam na criação de software. Por exemplo, análise e codificação de requisitos são duas das muitas atividades associadas. (SOARES, 2004)

Há vários tipos de metodologia de desenvolvimento de software como o Scrum, PMBOK, Kanban, Ágil, Ciclo de Vida Único, Modelo em cascata, entre outras. Cada uma delas possui suas próprias características e abordagens para gerenciamento de projetos, mas todas elas visam ajudar as equipes a entregar projetos de software de qualidade dentro do prazo, orçamento e com as funcionalidades esperadas. Algumas metodologias são mais estruturadas e baseadas em fases, enquanto outras são mais flexíveis e baseadas em fluxo de trabalho, o que torna importante escolher a metodologia mais adequada para o projeto específico. (SILVA; SOUZA; CAMARGO, 2013)

3.1.1 Metodologia Ágil

As metodologias ágeis são um conjunto de abordagens para gerenciamento de projetos que se concentram em fornecer flexibilidade, adaptabilidade e colaboração entre equipes. Elas foram desenvolvidas originalmente para o desenvolvimento de software, mas agora são amplamente utilizadas em muitos outros campos. (LIBARDI; BARBOSA, 2010)

3.1.1.1 Scrum

Scrum é uma metodologia ágil específica que foi projetada para ajudar equipes a gerenciar projetos de desenvolvimento de software complexos e altamente incertos. Ele se concentra em entregar valor rapidamente e incrementos regulares, enquanto a equipe colabora e se adapta a mudanças no projeto. (STOPA; RACHID, 2019)

3.1.1.2 Kanban

Kanban é uma metodologia que se concentra em tornar visível o fluxo de trabalho, limitando o número de tarefas em andamento e permitindo que a equipe se adapte às mudanças de forma flexível. Ele é amplamente utilizado em conjunto com outras metodologias ágeis, como Scrum, para ajudar a equipe a aumentar a eficiência e a entrega de valor. (JUNIOR; FILHO, 2008)

3.1.2 Metodologia de Desenvolvimento Utilizada

Kanban e Scrum são ambas metodologias ágeis, mas possuem abordagens diferentes para o gerenciamento de projetos. Enquanto o Scrum é uma metodologia mais estruturada e baseada em sprints, Kanban é uma metodologia mais flexível e baseada no fluxo de trabalho. Ambos têm seus próprios benefícios e podem ser usados de forma complementar para ajudar a equipe a alcançar seus objetivos de projeto. (KNIBERG; SKARIN, 2010)

Sendo assim, o projeto foi desenvolvido utilizando ambas as metodologias, Kanban e Scrum, para que fosse possível atingir os objetivos propostos. O Kanban foi utilizado para gerenciar o fluxo de trabalho, enquanto o Scrum foi utilizado para gerenciar as sprints.

3.2 Ferramentas Utilizadas

3.2.1 Gerenciamento do Projeto

3.2.1.1 Trello

[Trello](#) é uma plataforma que auxilia equipes a organizarem e priorizarem projetos. Utiliza quadros de tarefas, onde cada tarefa é representada por um cartão, que pode ser movido e organizado em listas para indicar o progresso e o estado do projeto. Ele é uma ferramenta útil para gerenciar tarefas do projeto, estabelecer prazos e definir prioridades, além de acompanhar o andamento de cada Sprint.

3.2.1.2 Slack

[Slack](#) é uma ferramenta de comunicação de equipe que permite que as equipes criem canais de bate-papo para discutir projetos, compartilhar arquivos e se comunicar de forma rápida e eficiente.

No contexto do desenvolvimento de software, Slack pode ajudar a equipe a se comunicar de forma mais eficiente e colaborativa, permitindo que desenvolvedores, gerentes de projetos e outros membros da equipe discutam e compartilhem informações de forma rápida e fácil. Ele pode ser usado para discutir problemas técnicos, atribuir tarefas, acompanhar o progresso do projeto e compartilhar arquivos, tudo em um único lugar. Além disso, ele pode ser integrado com outras ferramentas, como o GitHub, o Google Drive e o Trello, para ajudar a equipe a automatizar tarefas e trabalhar de forma mais eficiente.

3.2.1.3 Google Drive

[Google Drive](#) é um serviço de armazenamento e compartilhamento de arquivos na nuvem fornecido pelo Google. Ele permite que os usuários armazenem arquivos, como do-

cumentos, fotos e vídeos, e compartilhem esses arquivos com outras pessoas. Ele também oferece recursos de colaboração, como edição em tempo real de documentos, comentários e histórico de versões.

3.2.2 Desenvolvimento

3.2.2.1 Figma

[Figma](#) é uma ferramenta de design colaborativo que permite que equipes criem e compartilhem arquivos de design. Ele permite que os usuários criem protótipos de aplicativos e sites, além de permitir que eles compartilhem esses protótipos com outras pessoas para que possam colaborar e fazer comentários. Ele também permite que os usuários criem e compartilhem arquivos de design, como imagens, ícones e paletas de cores, que podem ser usados por outros membros da equipe.

3.2.2.2 GitHub e GitHub Actions

[GitHub](#) é uma plataforma de desenvolvimento de software que permite que os desenvolvedores armazenem, rastreiem e colaborem em projetos de código-fonte. Ele é baseado no sistema de controle de versão Git, que permite que os desenvolvedores façam alterações no código e versionem essas alterações, facilitando a colaboração e o controle de versão.

GitHub Actions é uma ferramenta de automação de fluxo de trabalho do GitHub. Ele permite que os desenvolvedores criem scripts de trabalho automatizados, chamados “Ações”, que podem ser disparados por eventos, como o envio de código para o repositório ou a criação de uma nova issue. Isso permite automatizar tarefas comuns, como compilação, teste e implantação, e integrar com outras ferramentas, como ferramentas de integração contínua e de monitoramento de desempenho.

3.2.2.3 Docker

[Docker](#) é uma plataforma de virtualização de aplicativos que permite que os desenvolvedores embalem e distribuam facilmente aplicativos em contêineres. Um contêiner é uma forma de isolamento de sistemas operacionais que permite que um aplicativo seja embalado com todas as suas dependências, bibliotecas e configurações, de forma que possa ser executado de forma consistente em diferentes ambientes.

Docker permite que os desenvolvedores criem e gerenciem contêineres, e que esses contêineres sejam implantados em diferentes ambientes, incluindo computadores locais, nuvens públicas e privadas. Isso permite que os desenvolvedores criem aplicativos de forma mais eficiente e confiável, e que esses aplicativos possam ser executados de forma

consistente em diferentes ambientes. Além disso, ele também permite a colaboração entre equipes, e aumenta a segurança, escalabilidade e gerenciamento do aplicativo.

3.2.2.4 Visual Studio Code

VSCode (**Visual Studio Code**) é um editor de código-fonte desenvolvido pela Microsoft. Ele é uma ferramenta gratuita e de código aberto, que suporta diversas linguagens de programação, incluindo JavaScript, Python, C++, Java e muitas outras. Ele oferece recursos avançados de edição, como sugestão de código, depuração e gerenciamento de versão. Além disso, ele também possui uma ampla variedade de extensões e plugins desenvolvidos pela comunidade, que podem ser usadas para adicionar recursos adicionais, como integração com ferramentas de teste, análise de código e integração com outras ferramentas.

3.3 Processo de Desenvolvimento

3.3.1 Cronograma

3.4 Cronograma

O cronograma foi dividido em 6 fases, sendo que cada fase possui um prazo de 2 semanas. A Figura 1 mostra o cronograma completo do projeto.

- Fase 1: Reestruturação do trabalho existente; Levantamento de requisitos; criação do modelo de dados, arquitetura e pacotes.
- Fase 2, 3 e 4: Desenvolvimento do sistema.
- Fase 5: Correção de bugs e ajustes finais; Análise e coleta de resultados.
- Fase 6: Documentação e apresentação do projeto.

CRONOGRAMA DE PROJ ETO

TÍTULO DO PROJ ETO	Plataforma web para auxiliar no ensino da linguagem de programação Python de forma mais interativa e divertida	NOME DA EMPRESA	Enpyre
GERENTE DO PROJ ETO	Vitor Nere	DATA	09/01/23



Figura 1 – Cronograma

3.4.1 Backlog de desenvolvimento

O backlog de desenvolvimento foi definido na Fase 1 do cronograma, e é composto por 3 pacotes, sendo que cada pacote possui um conjunto de tarefas. A Figura 2 mostra o backlog de desenvolvimento completo do projeto.

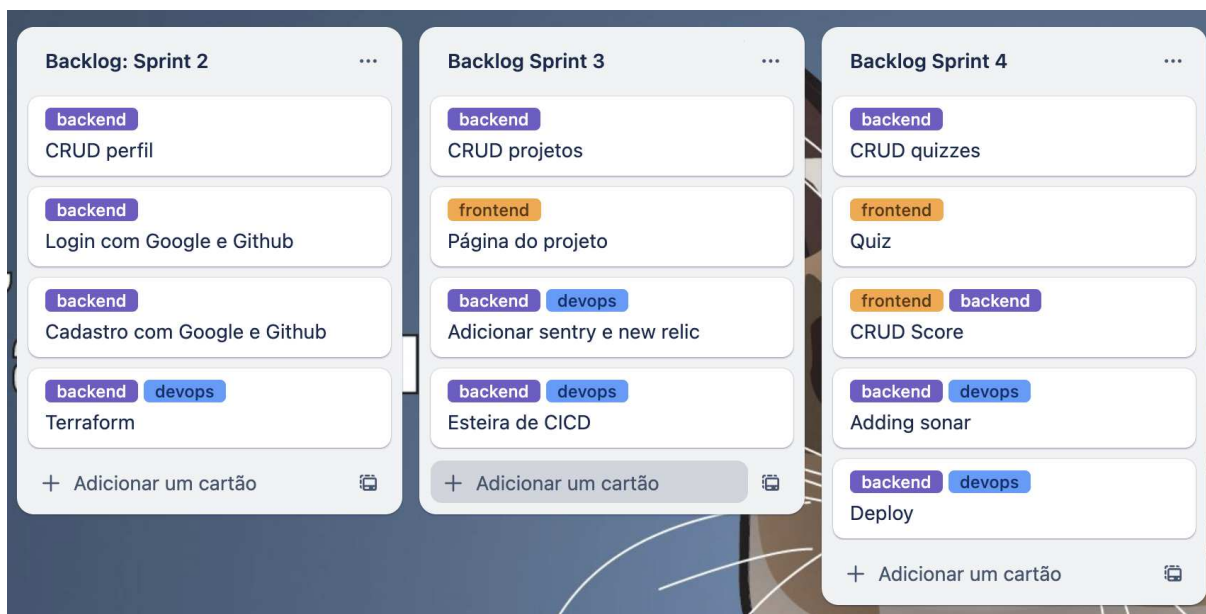


Figura 2 – Backlog de desenvolvimento

4 Solução Desenvolvida

4.1 Visão Geral

O projeto consistiu em uma plataforma de aprendizado de programação em Python, com foco em iniciantes. A plataforma é composta por um conjunto de quizzes, desafios e projetos, que são utilizados para fixar o conteúdo aprendido. A plataforma também conta com um sistema de pontuação, que é utilizado para incentivar os usuários a realizarem os quizzes, desafios e projetos, além de compartilharem seus projetos com outros usuários. A plataforma também conta com um sistema de ranking, que é utilizado para mostrar aos usuários o desempenho deles em relação aos outros usuários.

4.2 Requisitos

Os requisitos funcionais da plataforma foram definidos com base nos objetivos do projeto, que são: ensinar programação em Python para iniciantes, de forma lúdica e interativa. Sendo que foram priorizados os requisitos de RF01 a RF09 para a primeira versão da plataforma. Os requisitos da plataforma são:

- RF01: Cadastro de usuários: o sistema deve permitir que os usuários se cadastrem na plataforma.
- RF02: Login de usuários: o sistema deve permitir que os usuários façam login na plataforma.
- RF03: Visualização de quizzes e desafios: o sistema deve permitir que os usuários visualizem quizzes e desafios.
- RF04: Realização de quizzes: o sistema deve permitir que os usuários realizem quizzes para fixar o conteúdo.
- RF05: Realização de desafios: o sistema deve permitir que os usuários realizem desafios para fixar o conteúdo.
- RF06: Criação de projetos: o sistema deve permitir que os usuários criem projetos em Python.
- RF07: Compartilhamento de projetos: o sistema deve permitir que os usuários compartilhem projetos através de links.

- RF08: Pontuação: o sistema deve permitir que os usuários acumulem pontos ao realizar quizzes, desafios, projetos e compartilhar projetos.
- RF09: Visualização da pontuação: o sistema deve permitir que os usuários visualizem sua pontuação.
- RF09: Visualização de ranking: o sistema deve permitir que os usuários visualizem o ranking de pontos.
- RF10: Visualização do histórico de pontuação: o sistema deve permitir que os usuários visualizem o histórico de pontuação.
- RF11: Visualização do perfil: o sistema deve permitir que os usuários visualizem seu perfil.
- RF12: Edição do perfil: o sistema deve permitir que os usuários editem seu perfil.
- RF13: Exclusão do perfil e dados: o sistema deve permitir que os usuários excluam seu perfil e dados.

4.3 A Distribuição do Software

Com o propósito de reduzir as dificuldades associadas com o preparo do ambiente de desenvolvimento, desejou-se que a ferramenta proposta possuísse o mínimo de dependências possível, e que fosse compatível com qualquer sistema operacional. Uma forma de se atingir este objetivo foi através do uso de uma solução que fosse executada à partir do navegador.

A distribuição padrão do Python, conhecida como CPython, é implementada em C, e não roda nativamente no navegador. Existem poucas soluções que viabilizem isso, sendo o WebAssembly a mais completa. Trata-se de um padrão que define um formato de código binário de baixo nível capaz de ser executado à partir de um navegador. É possível compilar o interpretador de Python para WebAssembly, tornando possível sua execução em ambiente Web.

No entanto, existem vários desafios de integração entre o código do interpretador e o ambiente do navegador, e por isso não basta uma simples recompilação. O projeto Pyodide não só prepara a compilação do interpretador do Python, como também tem uma série de funções para integrar melhor os dois ambientes e promover contexto compartilhado entre Python e JavaScript.

Buscou-se então uma biblioteca JavaScript capaz de desenhar em um canvas com HTML5. Dentre as opções, escolheu-se o PixiJS. O objetivo passou a ser invocar o PixiJS à partir de código Python, de modo que o estudante pudesse desenhar formas geométricas, linhas e vértices com linhas de código simples.

4.4 Arquitetura

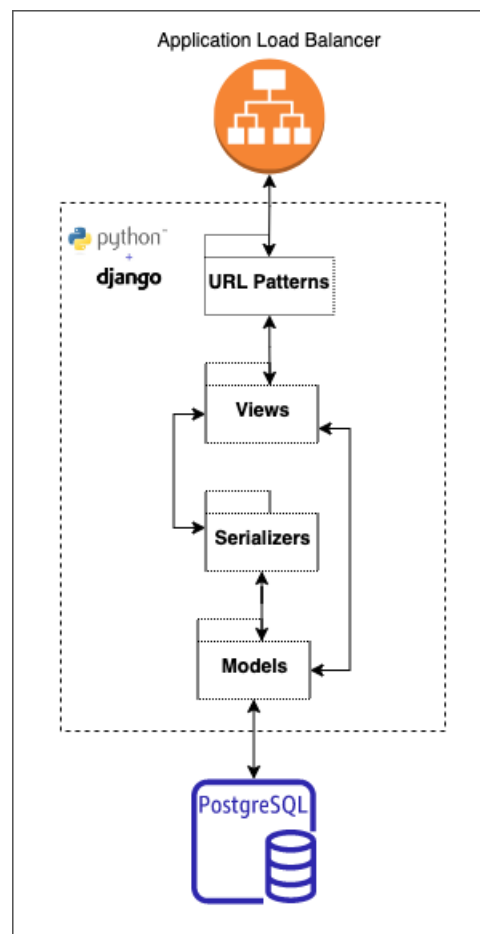
A arquitetura do sistema é composta por três componentes principais: o servidor, o cliente e o banco de dados. O servidor é responsável por receber as requisições do cliente e enviar as respostas. O cliente é responsável por enviar as requisições ao servidor e receber as respostas. O banco de dados é responsável por armazenar os dados do sistema.

4.4.1 Servidor

O servidor é responsável por receber as requisições do cliente e enviar as respostas. O servidor é composto por um conjunto de módulos que são responsáveis por realizar as tarefas do sistema. Cada módulo é responsável por uma tarefa específica. Os pacotes utilizados no servidor são:

- **URL Patterns:** são padrões de URLs utilizados pelo Django Rest Framework para identificar as requisições recebidas pelo servidor.
- **Views:** são funções que são executadas quando uma requisição é recebida pelo servidor. As views são responsáveis por realizar as tarefas do sistema.
- **Serializers:** são classes que são utilizadas para serializar e desserializar os dados recebidos e enviados pelo servidor. As classes de serialização são responsáveis por converter os dados recebidos e enviados pelo servidor para o formato JSON.
- **Models:** são classes que são utilizadas para representar os dados do sistema. As classes de modelo são responsáveis por representar os dados do sistema no banco de dados.

Figura 3 – Representação de pacotes e tecnologias do servidor



Fonte: Autores

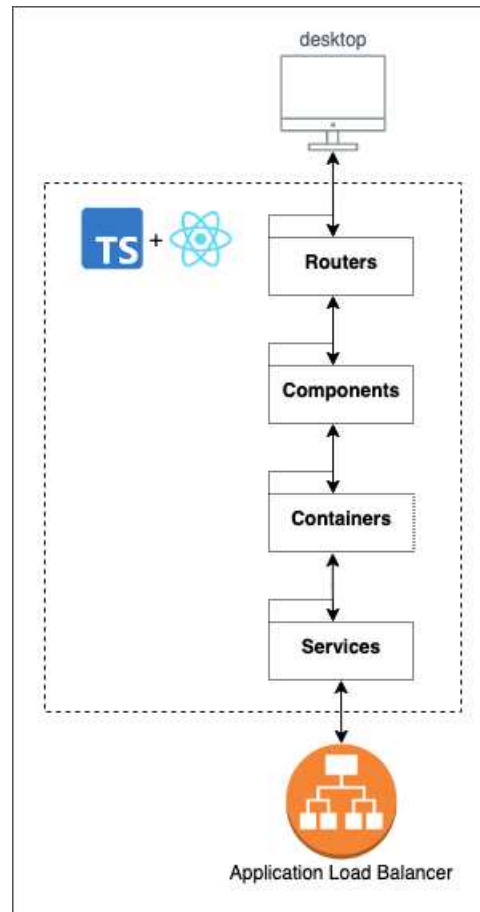
4.4.2 Cliente

O cliente é responsável por enviar as requisições ao servidor e receber as respostas. O cliente é composto por um conjunto de módulos que são responsáveis por realizar as tarefas do sistema. Cada módulo é responsável por uma tarefa específica. Os pacotes utilizados no cliente são:

- **Routers:** são classes que são utilizadas para definir as rotas da aplicação. Através das rotas é renderizado o componente correto para cada rota.
- **Components:** são classes que são utilizadas para definir os componentes da aplicação. Os componentes possuem uma estrutura HTML e são responsáveis por renderizar a interface do usuário.
- **Containers:** são classes que são utilizadas para definir os containers da aplicação. Os containers possuem a lógica de negócio da aplicação.

- **Services:** são classes que são utilizadas para definir os serviços da aplicação. Os serviços são responsáveis por realizar as requisições ao servidor.

Figura 4 – Representação de pacotes e tecnologias do cliente



Fonte: Autores

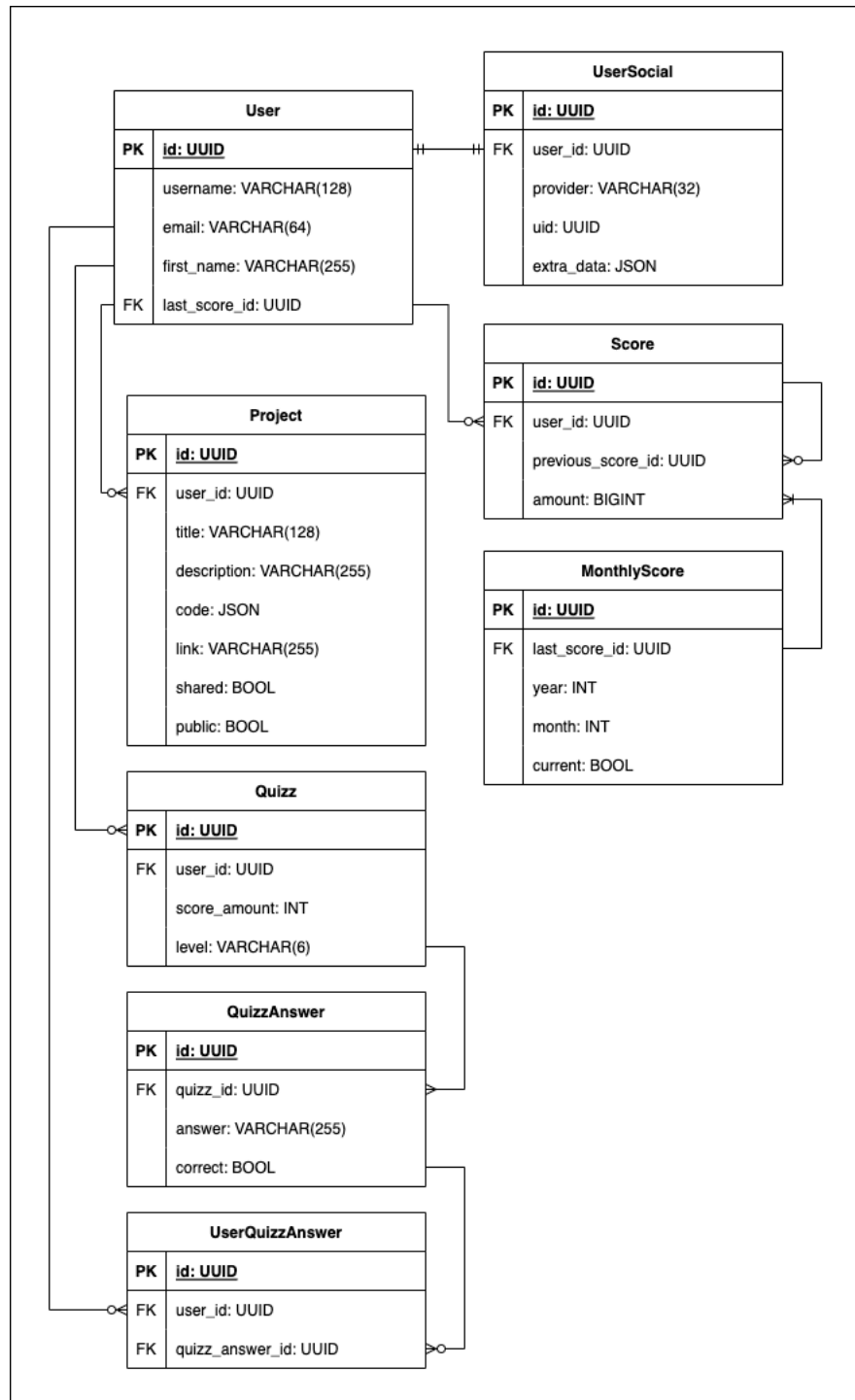
4.4.3 Modelo de Dados

Um modelo de dados é uma representação abstrata de como os dados são estruturados e como eles se relacionam entre si. Ele é utilizado para descrever a estrutura lógica e conceitual dos dados, independentemente da forma como eles são armazenados ou implementados em um sistema. Existem vários tipos de modelos de dados, como o modelo relacional, o modelo de objeto-relacional e o modelo de dados NoSQL.

A escolha do modelo de dados é uma decisão importante, pois ela afeta diretamente a forma como os dados são armazenados e manipulados. O modelo de dados escolhido deve ser capaz de atender as necessidades do sistema. O modelo de dados escolhido para o sistema é o modelo relacional, pois ele é o modelo de dados mais utilizado no mundo e é capaz de atender as necessidades do sistema.

Abaixo é apresentado o modelo de dados do sistema. O modelo de dados é composto por 4 entidades: *User*, *Project*, *Quiz* e *Score*. Além das entidades, o modelo de dados é composto por 4 relacionamentos entre as entidades: *SocialUser*, *QuizAnswer*, *UserQuizAnswer* e *MonthlyScore*:

Figura 5 – Modelo de dados do sistema



Fonte: Autores

4.4.4 Configuração do Ambiente de Produção

4.4.4.1 Servidor

O ambiente de produção do servidor é composto por um conjunto de ferramentas que são utilizadas para garantir a disponibilidade, escalabilidade, segurança e desempenho dos sistemas em execução. As ferramentas utilizadas no ambiente de produção do servidor são:

- **Python:** é uma linguagem de programação de alto nível, interpretada, de uso geral, criada por Guido van Rossum em 1989. Ela se destaca pela sua sintaxe clara e concisa, além de possuir uma grande variedade de bibliotecas e frameworks, tornando-a uma linguagem versátil e popular para desenvolvimento de diversas aplicações.
- **Django Rest Framework:** é um conjunto de ferramentas para criar aplicativos RESTful com o framework web Django. Ele fornece uma série de recursos para facilitar o desenvolvimento de APIs, como serialização automática de objetos do Django, autenticação, permissões, paginação e suporte a diferentes formatos de dados (JSON, XML, etc).
- **Docker:** é uma plataforma de software que permite aos desenvolvedores e administradores de sistemas embalar, distribuir e executar aplicativos em contêineres. Um contêiner é uma forma de isolamento de sistema, onde a aplicação e suas dependências são embaladas em um pacote, junto com tudo o que é necessário para rodá-lo, incluindo o sistema operacional, bibliotecas e configurações.
- **PostgreSQL:** é um sistema gerenciador de banco de dados relacional de código aberto. Ele é conhecido por sua robustez, escalabilidade e suporte a recursos avançados, como transações de nível de banco de dados, gatilhos, armazenamento de objetos, entre outros.
- **Route 53:** é um serviço de DNS gerenciado pela Amazon Web Services (AWS) que permite aos desenvolvedores criar domínios personalizados e hospedar sites na nuvem. Ele fornece alta disponibilidade e tolerância a falhas, além de suporte a várias regiões geográficas.
- **Elastic Load Balancing (ELB):** é um serviço de balanceamento de carga gerenciado pela Amazon Web Services (AWS) que distribui o tráfego entrante em várias instâncias de aplicativos em execução em um ou mais servidores. Ele permite aos desenvolvedores escalar horizontalmente suas aplicações, garantindo alta disponibilidade e tolerância a falhas, sem precisar gerenciar manualmente as configurações de balanceamento de carga.

- **Amazon Elastic Container Service (ECS):** é um serviço de nuvem gerenciado pela Amazon Web Services (AWS) que permite aos desenvolvedores executar e escalar aplicativos em contêineres. Ele fornece uma abstração para gerenciar e escalar clusters de contêineres, além de integração com outros serviços da AWS, como o Elastic Load Balancing (ELB) e o Elastic Block Store (EBS)
- **Amazon Elastic Container Registry (ECR):** é um serviço de nuvem gerenciado pela Amazon Web Services (AWS) que permite aos desenvolvedores armazenar e gerenciar imagens de contêineres Docker. Ele fornece uma abstração para armazenar e gerenciar imagens de contêineres, além de integração com outros serviços da AWS, como o Amazon Elastic Container Service (ECS) e o Amazon Elastic Kubernetes Service (EKS).
- **Amazon Relational Database Service (RDS):** é um serviço de nuvem gerenciado pela Amazon Web Services (AWS) que permite aos desenvolvedores executar e escalar bancos de dados relacionais em nuvem. Ele fornece uma abstração para gerenciar e escalar clusters de bancos de dados relacionais, além de integração com outros serviços da AWS, como o Elastic Load Balancing (ELB) e o Elastic Block Store (EBS).
- **Terraform:** é uma ferramenta de automação de infraestrutura como código (IaC) que permite aos desenvolvedores declarar a infraestrutura de seus sistemas em um arquivo de configuração e, em seguida, executar esse arquivo para provisionar e gerenciar a infraestrutura de seus sistemas na nuvem.

4.4.4.2 Cliente

O ambiente de produção do cliente é composto por um conjunto de ferramentas que são utilizadas para garantir a disponibilidade, escalabilidade, segurança, desempenho e compatibilidade com diferentes dispositivos e navegadores dos sistemas web em execução. As ferramentas utilizadas no ambiente de produção do cliente são:

- **JavaScript:** é uma linguagem de programação de alto nível, interpretada e de uso geral, que é principalmente utilizada para criar scripts para páginas da web e aplicativos web. Ele permite que os desenvolvedores adicionem interatividade e dinamicidade às páginas da web, tornando-as mais atraentes e intuitivas para os usuários.
- **ReactJS:** ReactJS é um biblioteca JavaScript open-source para construir interfaces de usuário (UI). Ele foi desenvolvido pelo Facebook e tem sido amplamente adotado por outras empresas e comunidades. Ele permite aos desenvolvedores criar componentes de UI reutilizáveis e gerenciar seu estado de forma eficiente. ReactJS utiliza uma abordagem de programação baseada em componentes, onde cada componente

é responsável por renderizar uma parte específica da UI e pode ser facilmente combinado com outros componentes para construir interfaces complexas. Ele também tem uma funcionalidade chamada de “Virtual DOM”, que otimiza o desempenho das atualizações na tela.

- **Material UI:** Material-UI é um framework de interface de usuário de código aberto para React que segue as diretrizes de design do Material Design, um conjunto de diretrizes de design criadas pelo Google. Ele fornece uma coleção de componentes prontos para uso, como botões, caixas de seleção, tabelas e formulários, além de fornecer suporte a temas personalizados e adaptabilidade a diferentes tamanhos de tela. Ele é projetado para ser fácil de usar e personalizar, e se integra facilmente com outras bibliotecas e frameworks.
- **Amplify Hosting:** é uma funcionalidade dentro do Amplify que permite aos desenvolvedores hospedar facilmente aplicativos web estáticos, como páginas SPA (Single Page Application), sites estáticos e landing pages. Ele oferece uma forma simples de configurar e gerenciar o armazenamento e a distribuição de arquivos estáticos, além de suportar recursos avançados como roteamento personalizado, suporte ao CloudFront e integração com o CloudFormation.

4.4.5 Configuração do Ambiente de Desenvolvimento

4.4.5.1 Servidor

O ambiente de desenvolvimento do servidor além de ser composto pelas ferramentas utilizadas no ambiente de produção 4.4.4.1, também inclui ferramentas adicionais para facilitar o processo de desenvolvimento, como ferramentas de gerenciamento de código-fonte, integração contínua, testes automatizados, depuração e monitoramento. Além disso, é comum que o ambiente de desenvolvimento seja configurado para simular o ambiente de produção o mais fielmente possível, de forma a permitir que os desenvolvedores testem o código com precisão antes de lançá-lo em produção. As ferramentas utilizadas no ambiente de desenvolvimento do servidor são apresentadas a seguir:

- **Pylint:** é uma ferramenta de análise de código para a linguagem de programação Python. Ele verifica o código para seguir as boas práticas de estilo, detecta erros e problemas de qualidade, e gera relatórios sobre o desempenho do código. Ele pode ser integrado com diversas IDEs e ferramentas de gerenciamento de código-fonte, e é amplamente utilizado para melhorar a qualidade e a manutenibilidade do código.
- **Mypy:** é uma ferramenta de verificação de tipos para a linguagem Python. Ele verifica o código para garantir que os tipos sejam usados corretamente, detectando erros de tipo em tempo de desenvolvimento, antes que eles sejam executados. Ele

pode ser usado como uma ferramenta de linha de comando ou integrado com outras ferramentas de desenvolvimento, como IDEs e sistemas de gerenciamento de código-fonte.

- **Black:** Black é uma ferramenta de formatação automática de código para Python. Ele segue as boas práticas de estilo recomendadas pela comunidade Python, e formata o código de acordo com essas regras, garantindo que o código seja consistente e fácil de ler. Ele pode ser integrado com ferramentas de gerenciamento de código-fonte, como o Git, e usado como parte do fluxo de trabalho de desenvolvimento para garantir que o código seja formatado corretamente antes de ser enviado para o repositório.
- **Pre-commit:** Pre-commit é uma ferramenta de gerenciamento de configurações de hooks de commit para várias ferramentas de desenvolvimento. Ele permite que os desenvolvedores configurem várias verificações, como testes automatizados, análise de código e formatação, para serem executadas antes de cada commit. Isso garante que o código enviado para o repositório esteja de acordo com as boas práticas e padrões estabelecidos, e evita problemas de qualidade e compatibilidade.
- **Docker Compose:** é uma ferramenta de linha de comando para definir e executar aplicativos multi-contêiner usando o Docker. Ele permite que os desenvolvedores escrevam uma configuração simples em um arquivo YAML, descrevendo os contêineres, volumes e redes necessários para o aplicativo. Com essa configuração, os desenvolvedores podem facilmente criar e gerenciar os contêineres e suas dependências em ambiente de desenvolvimento, teste ou produção.
- **GitHub:** é uma plataforma de hospedagem de código-fonte baseada na web que fornece controle de versão, gerenciamento de projetos e colaboração para desenvolvedores. Ele é baseado no sistema de controle de versão Git e permite que os desenvolvedores armazenem, compartilhem e colaborem em projetos de software usando o Git. Ele também fornece recursos adicionais, como gerenciamento de tarefas, relatórios de erros, documentação e integração com outras ferramentas de desenvolvimento.
- **GitHub Actions:** é uma funcionalidade do GitHub que permite que os desenvolvedores criem e gerenciem fluxos de trabalho automatizados diretamente no GitHub. Ele permite que os desenvolvedores automatizem tarefas como testes, implantações, construções e entregas de software, sem precisar deixar a plataforma do GitHub. Ele fornece uma ampla variedade de ações pré-construídas e integrações com outras ferramentas, além de suportar scripts personalizados. Isso permite que os desenvolvedores aumentem a eficiência e reduzam o tempo de entrega, além de monitorar e auditar facilmente os fluxos de trabalho.

- **SonarCloud:** SonarCloud é uma plataforma de análise de código baseada na nuvem que fornece ferramentas de análise de qualidade de código, métricas e relatórios para desenvolvedores. Ele é projetado para ajudar os desenvolvedores a melhorar a qualidade, manutenibilidade e segurança do código. Ele suporta uma variedade de linguagens de programação, incluindo Java, C#, JavaScript, TypeScript, Python, entre outras. Ele permite que os desenvolvedores façam análise estática de código, análise de segurança, monitoramento de cobertura de testes e outras verificações de qualidade.

4.4.5.2 Cliente

O ambiente de desenvolvimento do cliente além de ser composto pelas ferramentas utilizadas no ambiente de produção 4.4.4.2, também inclui ferramentas adicionais para facilitar o processo de desenvolvimento. As ferramentas utilizadas no ambiente de desenvolvimento do servidor são apresentadas a seguir:

- **TypeScript:** é uma linguagem de programação de código aberto desenvolvida e mantida pela Microsoft. Ele é uma super set do javascript, isso significa que todo o código javascript é válido em TypeScript. Ele adiciona tipagem estática ao javascript, o que permite aos desenvolvedores escrever código mais confiável e fácil de manter, além de oferecer recursos avançados como inferência de tipos, classes, interfaces, além de suporte a orientação a objetos. TypeScript compila para javascript, o que significa que pode ser executado em qualquer lugar onde javascript é suportado.
- **ESLint:** é uma ferramenta de análise de código estática para JavaScript e outras linguagens baseadas em JavaScript, como JSX, TypeScript. Ele ajuda a identificar e corrigir problemas de qualidade de código, como erros de sintaxe, padrões de codificação não seguidos, e boas práticas de desenvolvimento.
- **Prettier:** é uma ferramenta de formatação de código para JavaScript e outras linguagens baseadas em JavaScript, como JSX, TypeScript. Ele segue uma série de regras de formatação predefinidas para garantir que o código seja consistente e fácil de ler.
- **Husky:** é uma ferramenta de gerenciamento de ganchos de commit para Git. Ele permite que os desenvolvedores configurem scripts para serem executados automaticamente antes de cada commit, teste, push e outras operações do Git. Ele é especialmente útil para automatizar tarefas como testes automatizados, análise de código e formatação, garantindo que o código enviado para o repositório esteja de acordo com as boas práticas e padrões estabelecidos.

- **ViteJS:** ViteJS é uma ferramenta de desenvolvimento web de código aberto que foi projetada para ser rápida e fácil de usar. Ele é um servidor de desenvolvimento baseado em JavaScript que permite aos desenvolvedores iniciar rapidamente projetos e experimentar novas ideias sem a necessidade de configuração complexa. Ele usa o recurso ES modules nativo do navegador para carregar módulos, o que permite aos desenvolvedores desfrutar de carregamento de módulos instantâneo sem a necessidade de configurar um transpilador como o Babel ou o Webpack. ViteJS também inclui recursos como live-reloading, suporte a TypeScript e JSX, e suporte a plugins. Ele é projetado para trabalhar especialmente bem com frameworks modernos como React, Vue e Svelte.
- **Yarn:** é um gerenciador de pacotes para JavaScript, similar ao npm (o gerenciador de pacotes padrão para Node.js), mas com algumas funcionalidades adicionais. Ele permite que os desenvolvedores instalem e gerenciem pacotes JavaScript e suas dependências de forma eficiente e segura. Ele oferece uma série de recursos adicionais, como gerenciamento de cache, garantia de reproducibilidade, e melhorias de desempenho no gerenciamento de dependências.

4.5 A Ferramenta

A ferramenta apresentada por este trabalho chama-se Enpyre, uma plataforma de aprendizado de programação focada exclusivamente na linguagem Python, que oferece uma abordagem baseada em jogos para o ensino de programação. A ferramenta permite aos usuários criar jogos de forma fácil e intuitiva, utilizando uma interface gráfica e um editor de texto para escrever código Python. Além disso, a ferramenta utiliza a biblioteca PixiJS para fornecer uma interface gráfica para os jogos criados.

4.5.1 Disponibilidade

A ferramenta está disponível em: <https://enpyre-play.vercel.app/>.

O repositório do trabalho está disponível em: <https://github.com/Enpyre>.

4.5.2 Tela de boas-vindas

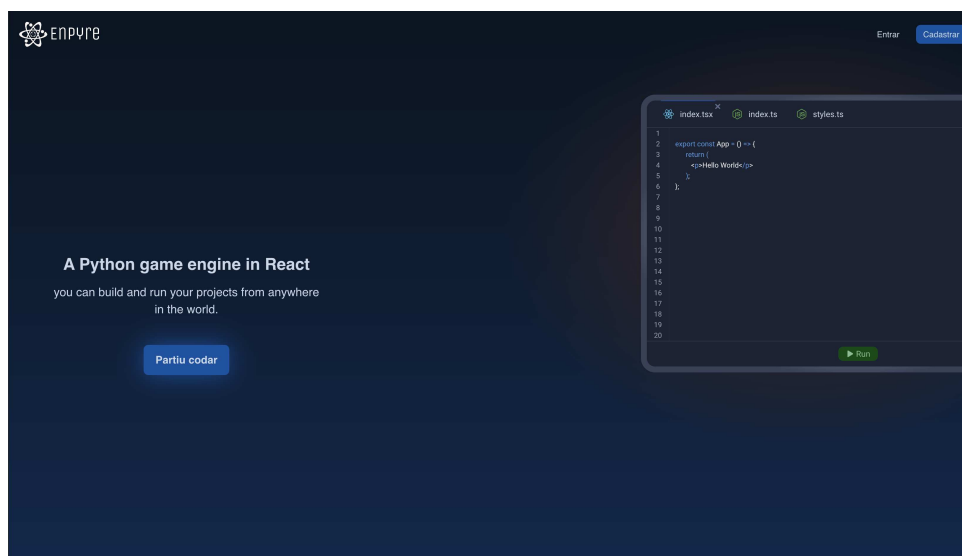


Figura 6 – Tela de boas-vindas

4.5.3 Tela de lista projetos

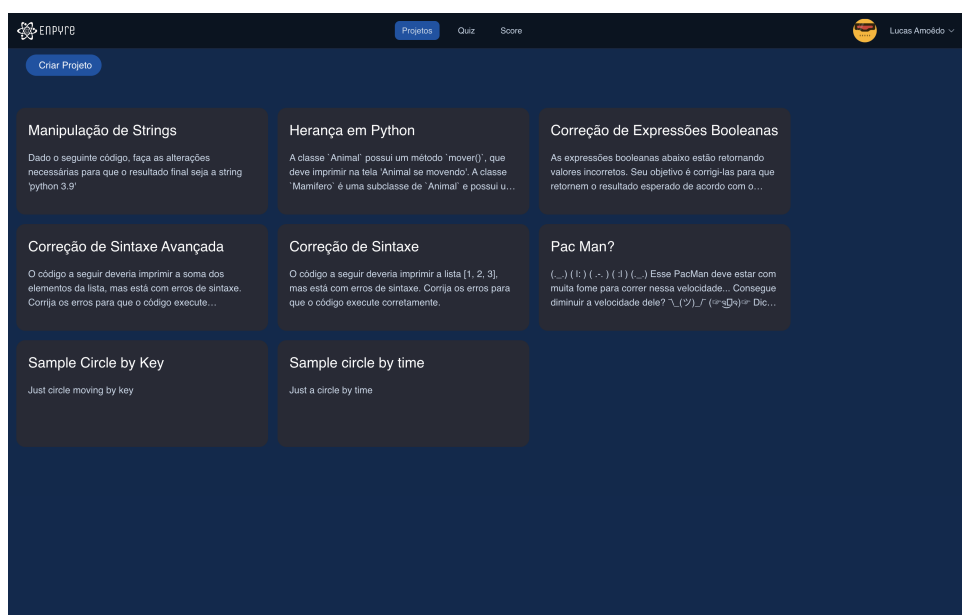


Figura 7 – Tela de lista de projetos

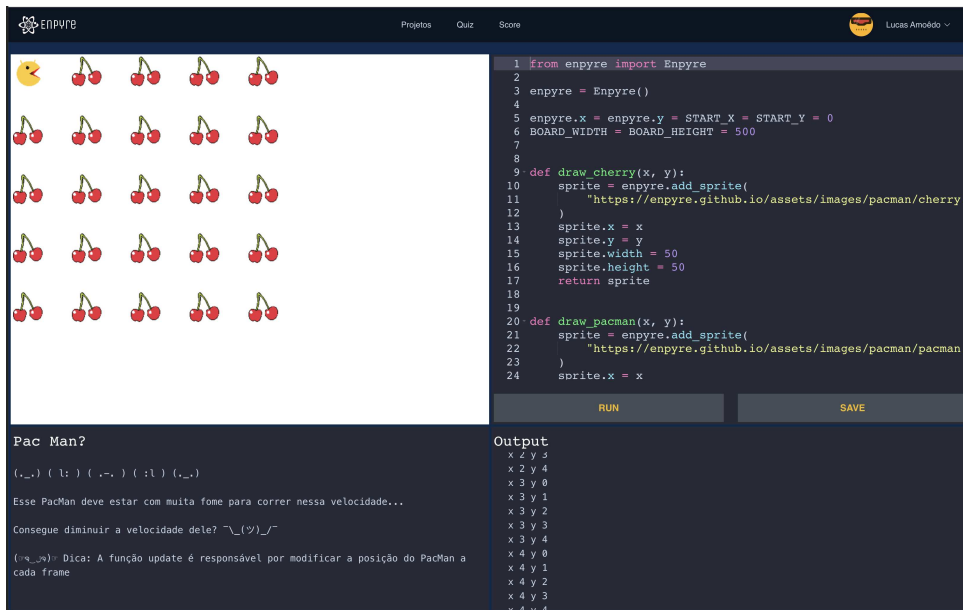


Figura 8 – Tela do projeto

4.5.4 Tela de quiz

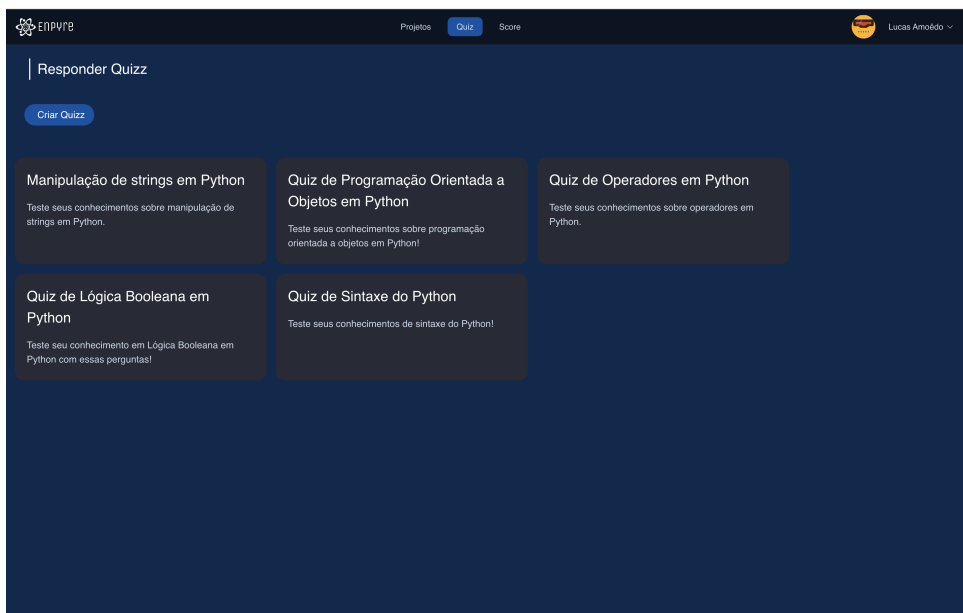


Figura 9 – Tela de lista de quiz

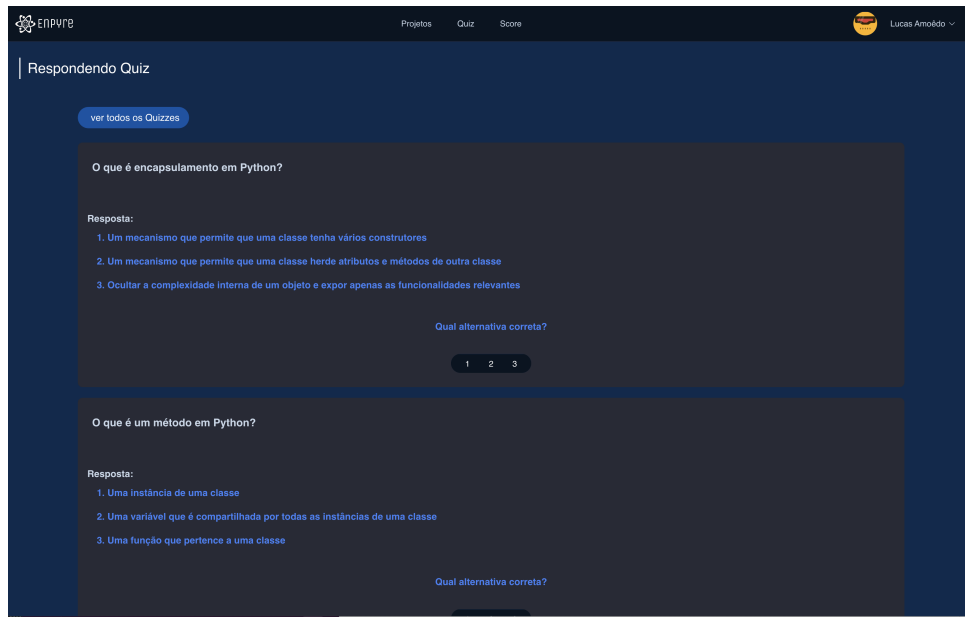


Figura 10 – Tela do quiz

4.6 Métricas de Qualidade

Foi utilizado o [SonarCloud](#) para realizar a análise estática de código e obter métricas de qualidade do código. As métricas finais obtidas podem ser vistas na imagem 11. A descrição das métricas de qualidade do código são apresentadas a seguir:

- **Reliability:** A métrica “Reliability” se refere à confiabilidade do código, ou seja, a capacidade do software em executar suas funcionalidades de forma consistente e livre de erros. Essa métrica avalia a estabilidade e robustez do código, medindo a probabilidade de ocorrência de falhas, erros de lógica, exceções não tratadas e comportamentos inesperados durante a execução do programa. É uma medida importante para garantir a qualidade do software, pois um código confiável é menos propenso a causar falhas e interrupções indesejadas.
- **Maintainability:** A métrica “Maintainability” se refere à manutenibilidade do código, ou seja, a facilidade com que o código pode ser mantido, modificado e aprimorado ao longo do tempo. Essa métrica avalia a qualidade do código em termos de legibilidade, modularidade, organização e documentação. Um código com alta manutenibilidade é mais fácil de entender, depurar, estender e adaptar a novos requisitos ou correções. Isso é fundamental para permitir que equipes de desenvolvimento trabalhem de forma eficiente e reduzir o risco de introdução de erros durante as atividades de manutenção e evolução do software.
- **Security:** A métrica “Security” se refere à segurança do código, ou seja, a capacidade do software em proteger dados sensíveis, prevenir vulnerabilidades e evitar

ataques cibernéticos. Essa métrica avalia a robustez das medidas de segurança implementadas no código, como autenticação, autorização, criptografia e proteção contra ameaças conhecidas. Um código com alta segurança é menos suscetível a brechas de segurança e protege as informações confidenciais dos usuários e do sistema como um todo. Garantir a segurança do código é fundamental para evitar violações de segurança e proteger a integridade e privacidade dos dados.

- **Security Review:** A “Security Review” é uma avaliação de segurança realizada no código ou em um sistema de software. Ela envolve a revisão minuciosa do código-fonte, infraestrutura e configurações de segurança para identificar possíveis vulnerabilidades, brechas ou falhas de segurança. Durante a revisão de segurança, são aplicadas técnicas e ferramentas específicas para analisar o código em busca de problemas de segurança conhecidos, padrões inseguros de programação e configurações inadequadas. O objetivo é identificar e corrigir quaisquer pontos fracos que possam ser explorados por hackers ou representar riscos à segurança do sistema.
- **Coverage:** “Coverage” se refere à cobertura de código, uma métrica que mede a quantidade de código que foi exercitada por testes automatizados. Ela indica a porcentagem de linhas de código, ramificações condicionais ou funcionalidades que foram testadas pelo conjunto de testes. Quanto maior a cobertura, maior a confiança de que o código foi amplamente testado e que as diferentes partes do software estão funcionando corretamente. A cobertura de código é uma medida importante para avaliar a qualidade dos testes e a eficácia dos esforços de garantia de qualidade no desenvolvimento de software.
- **Duplications:** “Duplications” se refere a duplicações de código, ou seja, trechos de código que aparecem repetidamente em diferentes partes do software. Essas duplicações podem ocorrer por várias razões, como falta de modularização, cópia e colagem excessiva de código, ou falta de reutilização de componentes. Identificar e analisar duplicações de código é importante para a manutenibilidade e qualidade do software, pois elas podem levar a problemas como dificuldade na manutenção, inconsistências, propagação de bugs e aumento da complexidade do código. Ao eliminar duplicações, é possível melhorar a legibilidade, eficiência e facilidade de manutenção do código.

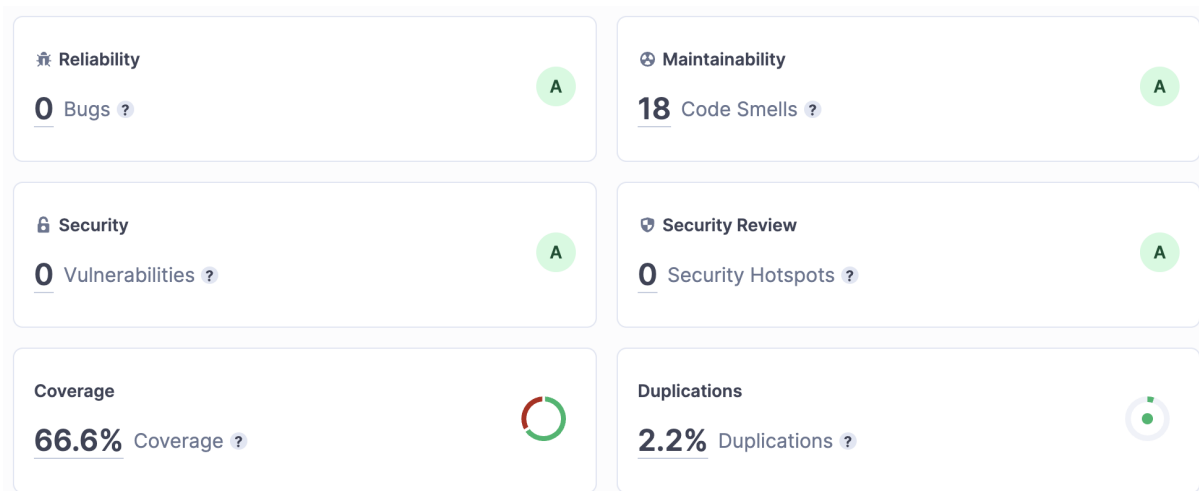


Figura 11 – Métricas de qualidade do código

5 Conclusão

Em conclusão, este trabalho apresentou uma solução para mitigar as dificuldades no ensino e aprendizagem de programação através da abordagem lúdica com a criação de jogos utilizando a linguagem Python. A ferramenta Enpyre foi desenvolvida com o intuito de oferecer uma opção acessível e prática para professores e alunos que buscam uma forma mais atrativa e eficaz de ensinar e aprender programação. Acreditamos que sua criação pode ser uma importante contribuição para a área de ensino de programação e que ela pode ser útil para aqueles que buscam uma abordagem mais dinâmica e envolvente para o aprendizado de programação. Esperamos que este trabalho possa incentivar novas pesquisas e contribuir para o desenvolvimento de ferramentas cada vez mais eficientes e acessíveis para o ensino de programação.

Apesar de fugir do escopo deste trabalho, os próximos passos naturais da Enpyre seriam testar a ferramenta em um ambiente real, de forma a mensurar sua capacidade de ensino de programação.

Por fim, é importante destacar que a aprendizagem de programação é um processo contínuo e que novas metodologias e ferramentas devem ser constantemente avaliadas e atualizadas para garantir que os estudantes tenham acesso ao melhor ensino possível.

Referências

- FANGOHR, H. *A Comparison of C, MATLAB, and Python as Teaching Languages in Engineering*. [S.l.], 2004. Citado na página 13.
- GOODRICH, M. T. *Estruturas de Dados e Algoritmos em Java*. [S.l.: s.n.], 2013. Citado na página 12.
- JAYAL, A. et al. *Innovation in Teaching and Learning in Information and Computer Sciences*. [S.l.], 2015. Citado na página 13.
- JUNIOR, M. L.; FILHO, M. G. Adaptações ao sistema kanban: revisão, classificação, análise e avaliação. 2008. Citado na página 18.
- KNIBERG, H.; SKARIN, M. *Kanban and Scrum-making the most of both*. [S.l.]: Lulu, 2010. Citado na página 19.
- LIBARDI, P. L.; BARBOSA, V. Métodos Ágeis. 2010. Citado na página 18.
- MANNILA, L.; PELTOMAKI, M.; SALAKOSKI, T. *What about a simple language? Analyzing the difficulties in learning to program*. [S.l.], 2006. Citado na página 13.
- MORAIS, P.; PIRES, J. N. *Python: Curso Completo*. 1st. ed. [S.l.]: FCA – Editora de Informática, Lda., 2002. Citado na página 13.
- MOREIRA, L. B. O uso de jogos digitais para o ensino de programação. 2018. Citado na página 15.
- PEREIRA, R. D. S. D. E. F.; SOUZA, A. D. de. Ferramentas de apoio ao ensino introdutório de programação: um mapeamento sistemático. 2020. Citado na página 11.
- PEREIRA, S. F. G. Jogos digitais como ferramenta de apoio ao ensino de programação. 2016. Citado na página 15.
- ROY, P. V.; HARIDI, S. *Concepts, Techniques and Models of Computer Programming*. [S.l.]: The MIT Press, 2004. Citado na página 12.
- SILVA, D. E. S.; SOUZA, I. T.; CAMARGO, T. Metodologias Ágeis para o desenvolvimento de software: Aplicação e o uso da metodologia scrum em contraste ao modelo tradicional de gerenciamento de projetos. *Revista de Computação Aplicada - UNG - SER*, v. 2, 2013. Citado na página 18.
- SILVA, J. C. F. R. R.; SANTOS, R. P. dos. Panorama da utilização de jogos digitais no ensino de programação no nível superior na Última década: Uma revisão sistemática da literatura. 2018. Citado 2 vezes nas páginas 13 e 14.
- SILVA, T. J. M. Thiago Reis da; ARANHA, E. H. da S. Jogos digitais para ensino e aprendizagem de programação: uma revisão sistemática da literatura. 2014. Citado 2 vezes nas páginas 13 e 14.

SOARES, M. S. Comparação entre metodologias Ágeis e tradicionais para o desenvolvimento de software. *INFOCOMP Journal of Computer Science*, v. 3, n. s. 1., 2004. Citado na página 18.

STEFIK, A.; SIEBERT, S. *An Empirical Investigation into Programming Language Syntax*. [S.l.], 2013. Citado 2 vezes nas páginas 12 e 13.

STOPA, G. R.; RACHID, C. L. Scrum: Metodologia ágil como ferramenta de gerenciamento de projetos. *CES Revista*, v. 33, 2019. Citado na página 18.

Apêndices

APÊNDICE A – Referencial Tecnológico da PoC

O presente capítulo lista as tecnologias escolhidas ou envolvidas na criação da ferramenta. da ferramenta Enpyre. Para as ferramentas de escolha, são apresentadas suas definições, suas vantagens e suas limitações. Para as tecnologias associadas ao projeto por consequência, são apresentadas suas definições.

A.1 WebAssembly

A.1.1 Definição

[WebAssembly](#) é um bytecode portátil de baixo nível para Web. É desenvolvido de modo a ser independente de hardware e plataforma. Com o WebAssembly, existe a possibilidade de se compilar código de diversas linguagens em código de baixo nível. Atualmente existe suporte para C, C++, Rust entre outros.

O WebAssembly foi proposto de forma colaborativa por engenheiros de quatro grandes distribuidores de navegadores: Google, Microsoft, Mozilla e Apple. A implementação atual já é capaz de compilar uma quantidade grande de programas, mas existem ainda várias propostas em estudo para estender as funcionalidades da linguagem e a integração com JavaScript.

A.1.2 Vantagens

O WebAssembly é seguro, rápido, independente de hardware e plataforma, além de ser portátil e compacto. Em comparação, outras tecnologias que tentam ou tentaram a execução de código nativo na Web, como o ActiveX da Microsoft, o Native Client da Google, o Java da Oracle e o Flash da Adobe, falham ao alcançar pelo menos uma destas características.

A.1.3 Limitações

Dentre as limitações presentes no WebAssembly, está sua incapacidade de acessar diretamente o DOM, então qualquer manipulação de DOM deve ser feita com JavaScript. Além disso, o WebAssembly funciona apenas com uma única thread, e não tem acesso a todas as funções do navegador.

A.2 Pyodide

A.2.1 Definição

O [Pyodide](#) é uma ferramenta que contém o interpretador do Python, em sua versão 3.9, sendo compilado para WebAssembly. Como resultado, é possível executar Python na Web, à partir de um navegador.

A ferramenta foi criada em 2018 por Michael Droettboom, desenvolvedor da Mozilla, inicialmente como parte do projeto Iodide. Este projeto propunha um ambiente de desenvolvimento baseado em navegadores, voltado para computação e programação científica e execução de notebooks do Jupyter. O projeto Iodide foi encerrado em Setembro de 2020, ainda podendo ser encontrado, mas não mais sendo mantido.

A.2.2 Vantagens

O Pyodide permite troca de informações entre os contextos JavaScript e Python. Por exemplo, é possível, usando JavaScript, manipular o DOM com informações obtidas do contexto do Python.

Para que a troca de informações entre contextos funcione, o Pyodide possui integrada uma conversão de tipos, que atua automaticamente durante o acesso de contexto do JavaScript para o Python ou vice-versa.

Além disso, é possível instalar dependências usando uma abstração do gerenciador de pacotes do Python, chamada “micropip”. É possível baixar dependências tanto do repositório oficial, quanto dependências hospedadas em outros ambientes, contanto que estejam no formato “wheel”.

A.2.3 Limitações

Dentre as limitações do Pyodide, está o fato de que não é capaz de guardar cache dos pacotes baixados durante sua execução.

Além disso, não é possível trabalhar com multiprocessamento, threading ou sockets, devido a limitações do WebAssembly.

A.3 PixiJS

A.3.1 Definição

O [PixiJS](#) é um sistema de renderização para Web que usa WebGL (ou Canvas, opcionalmente) para a exibição de conteúdo visual 2D.

A.3.2 Vantagens

O PixiJS foi construído com o intuito de maximar a performance de execução no navegador do usuário.

Além da renderização de imagens, a biblioteca oferece a renderização de objetos primitivos como linhas, círculos, polígonos, bem como textos e sprites.

Diferente do Unity e do Flash, o PixiJS não requer a instalação de um plugin ou aplicação para funcionar, dependendo apenas do navegador do usuário.

A.3.3 Limitações

O PixiJS se limita a renderização de imagens. Não é possível usá-lo para renderizações 3D, armazenamento de dados e reprodução de áudio. O PixiJS não é um framework para jogos e não possui interface gráfica.

A.3.4 Outras Tecnologias

As escolhas do PixiJS e do Piodide para o projeto trazem interação com as seguintes tecnologias:

A.3.4.1 Canvas API

Trata-se de uma API que viabiliza o desenho de gráficos 2D à partir do JavaScript. É possível construir animações, gráficos para jogos, manipulação de fotos, processamento de vídeo, visualização de dados etc. Os elementos gerados pela Canvas API são renderizados no elemento “canvas” do HTML.

A.3.4.2 WebGL API

Semelhante à [Canvas API](#), a [WebGL API](#) permite o desenho de gráficos em um elemento “canvas” do HTML. Diferente da Canvas API, no entanto, os gráficos gerados pela WebGL API podem ser 2D e 3D, e contam com aceleração de hardware.

A.3.4.3 HTML Canvas, DOM, Event, Document

- **HTML Canvas:** elemento HTML usado para renderizar gráficos da Canvas API ou WebGL API.
- **Event:** interface que representa um evento que ocorre no DOM. Um evento pode ser resultado de interação do usuário (clique de botão, por exemplo), gerado por APIs para notificar andamento de tarefas assíncronas (carregamento de página concluído, por exemplo) ou invocado programaticamente via JavaScript.

- **DOM:** significa *Document Object Model*, e representa o HTML carregado na forma de uma árvore de nós no navegador, onde cada nó representa uma parte do documento *Web*.
- **Document:** trata-se de um objeto que representa a página *Web* carregada no navegador, sendo a raiz do DOM.

APÊNDICE B – Arquitetura da Prova de Conceito

B.1 Visão Geral

B.2 Arquitetura

B.2.1 Camada Browser

Representa a comunicação entre as sub-camadas HTML, JavaScript e Python. Com o python sendo interpretado para código C e compilado para Web Assembly pelo Pyodide.

B.2.1.1 Camada HTML

Representa a comunicação entre dois principais componentes HTML Canvas e DOM.

- **HTML Canvas:** responsável por possibilitar a renderização de objetos 2D na DOM.
- **DOM:** responsável por representar os objetos e captura de interações do usuário.

B.2.2 Camada JavaScript

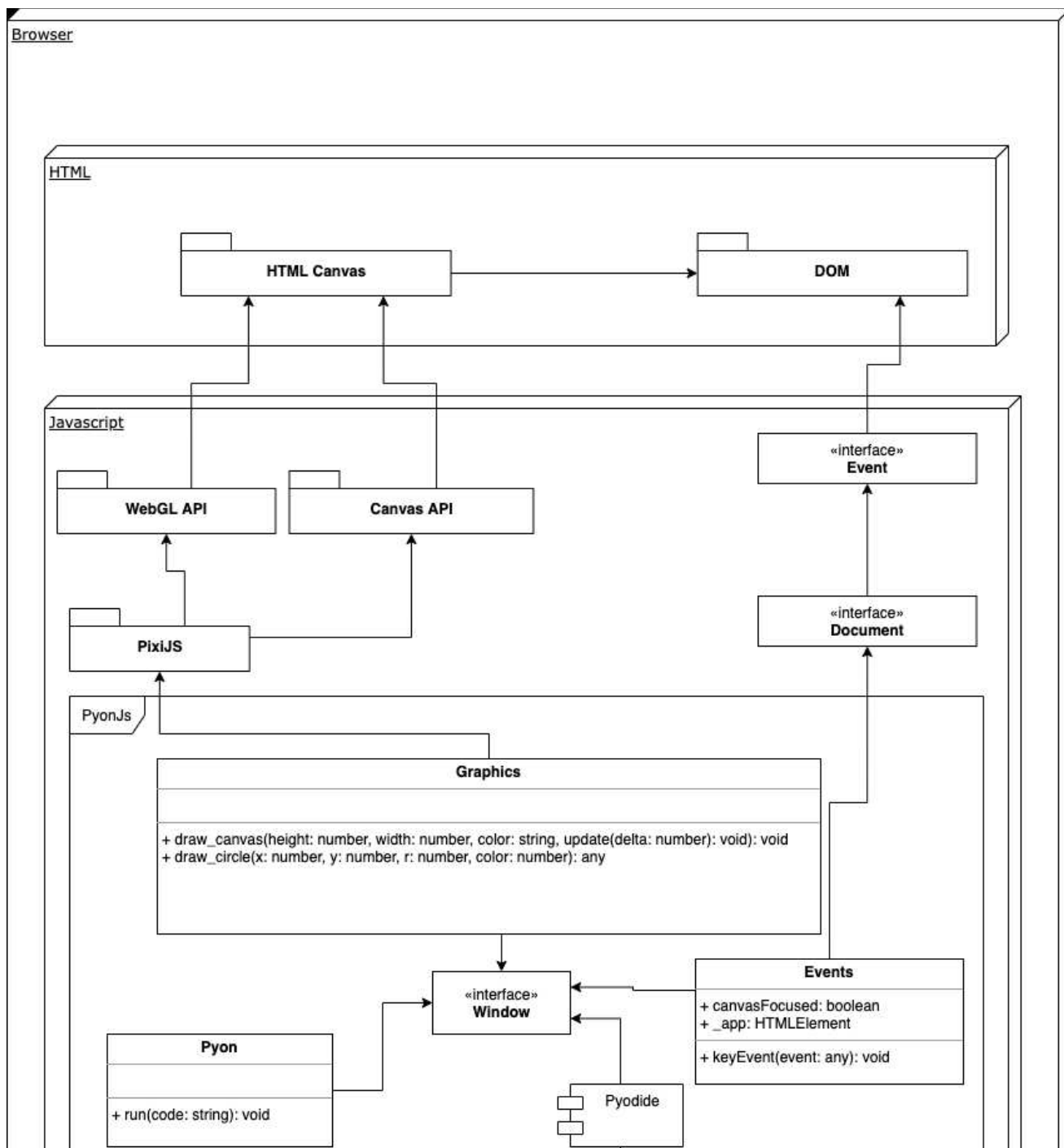
Representa a comunicação da Engine na camada JavaScript. [EnpyreJS](#)

- **Graphics:** Interface de comunicação entre a engine e o PixiJS, com suas funções disponibilizadas de forma global no Window.
- **Events:** Interface para monitoramento dos eventos de interação com a DOM, sendo o estado de cada evento atualizado de forma global no Window.
- **Enpyre:** Interface da engine onde seu propósito é compilar o código criado pelo aluno através da instância disponível no Window do Pyodide.

B.2.3 Camada Python

Representa a comunicação da engine na camada Python. Todo o código do usuário vai interagir com esta camada, e não com o JavaScript. [EnpyrePy](#)

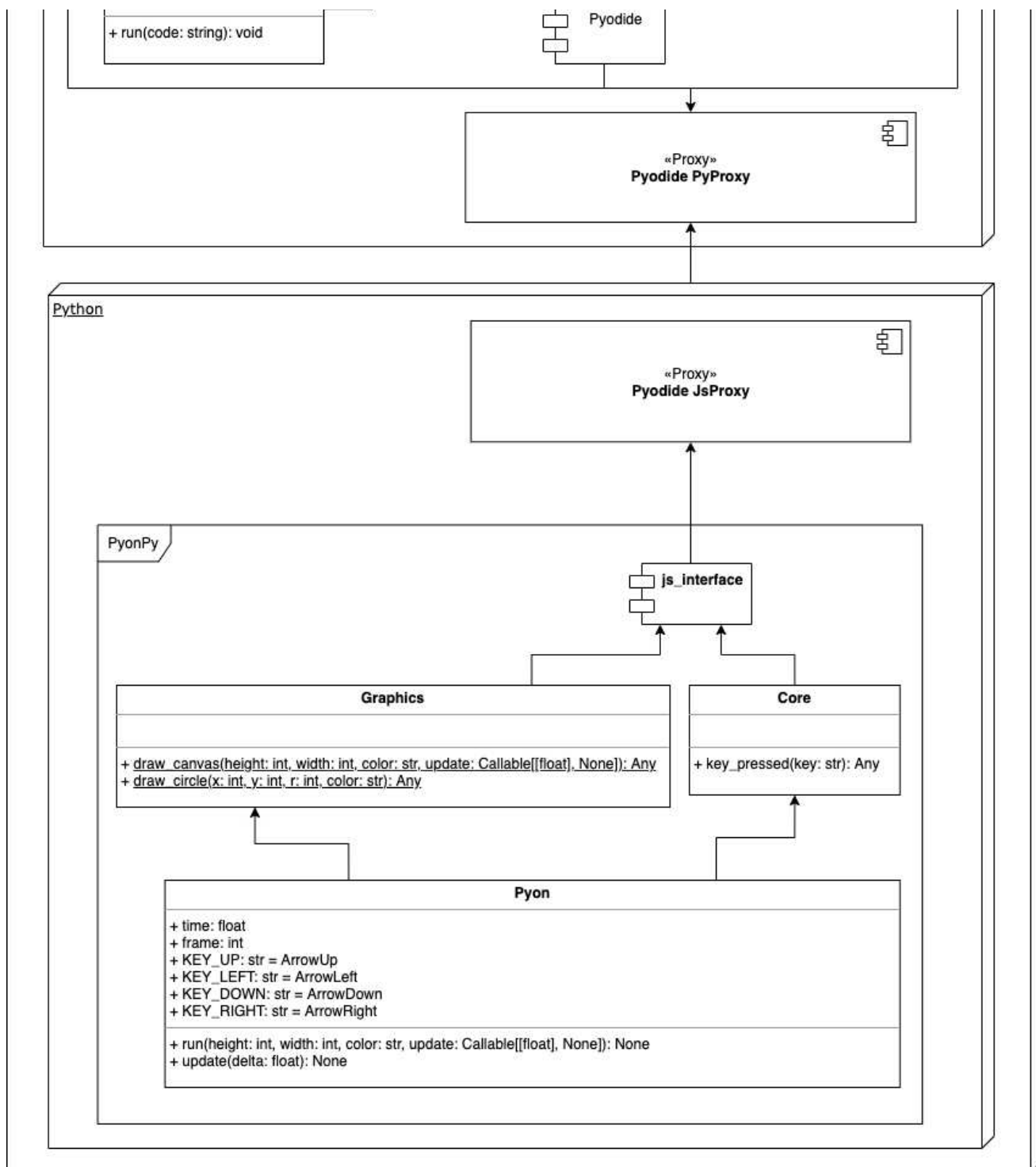
Figura 12 – Visão Geral - Parte 1



Fonte: Autores

- **Graphics:** Interface de comunicação entre a engine e o Graphics da camada de JavaScript.
- **Core:** Interface para monitoramento dos eventos de interação entre a engine e o Events da camada JavaScript.
- **Enpyre:** Interface que implementa atributos e funcionalidades que estarão disponí-

Figura 13 – Visão Geral - Parte 2



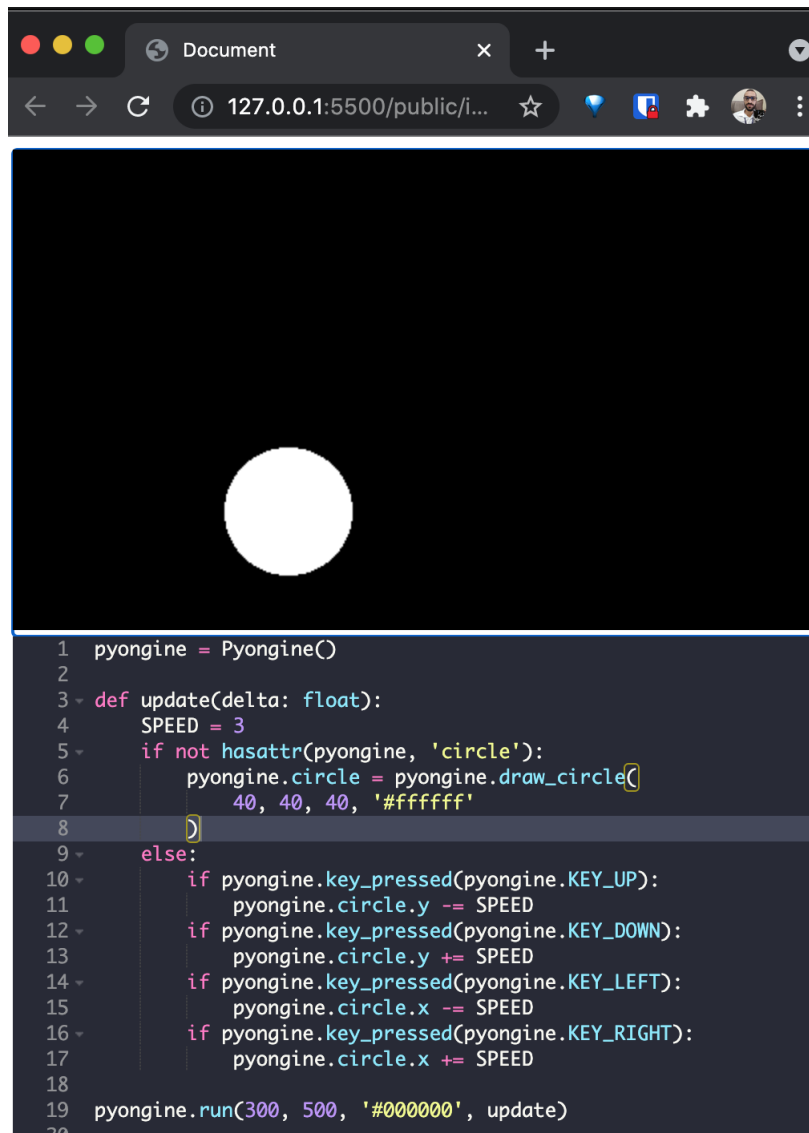
Fonte: Autores

veis durante a execução do código recebido pelo Enpyre da camada JavaScript.

APÊNDICE C – Resultados da Prova de Conceito

O presente capítulo exibe os resultados obtidos após a concepção da Prova de Conceito.

Figura 14 – Move Ball Game



```
1 pyongine = Pyongine()
2
3 def update(delta: float):
4     SPEED = 3
5     if not hasattr(pyongine, 'circle'):
6         pyongine.circle = pyongine.draw_circle(
7             40, 40, 40, '#ffffff'
8         )
9     else:
10        if pyongine.key_pressed(pyongine.KEY_UP):
11            pyongine.circle.y -= SPEED
12        if pyongine.key_pressed(pyongine.KEY_DOWN):
13            pyongine.circle.y += SPEED
14        if pyongine.key_pressed(pyongine.KEY_LEFT):
15            pyongine.circle.x -= SPEED
16        if pyongine.key_pressed(pyongine.KEY_RIGHT):
17            pyongine.circle.x += SPEED
18
19 pyongine.run(300, 500, '#000000', update)
20
```

Fonte: Autores

C.1 Aparência

A Figura ?? exibe um navegador aberto e dois elementos principais: um canvas, acima, contendo a saída do programa; e um editor de texto, abaixo, contendo o código escrito pelo usuário. O resultado do programa permite que o usuário interaja com o círculo desenhado, usando as teclas direcionais para alterar a direção em que a bola se move.

C.2 Nome da Ferramenta: Enpyre

Inicialmente foi dado o nome Pyon, que teve a origem descrita a seguir. Primeiro, pensou-se na ferramenta como um “motor” (*engine*, em inglês) de jogos escrito em Python. Este conceito foi alterado no princípio do trabalho, mas por sua razão adotou-se o nome provisório **Pyengine**.

Posteriormente, quando se consolidou o uso da biblioteca Pyodide, o nome provisório foi alterado para **Pyongine**, uma mescla do nome anterior com o nome da biblioteca.

Buscando-se simplificar o nome que se usava até então, reduziu-se Pyongine para **Pyon**, acreditando-se que sua pronúncia fosse mais agradável.

Por fim, o nome foi alterado para **Enpyre**, uma mescla de “engine”, “Python” e “React” (a biblioteca JavaScript usada para construir a interface do usuário).

C.3 Estrutura Visual

A aplicação é constituída por duas partes principais: um canvas, onde elementos podem ser desenhados, e um editor de texto integrado, com realce de sintaxe, onde o código Python pode ser escrito e executado.

C.4 Resultados

A prova de conceito comprovou a viabilidade do projeto, que mistura elementos e contextos do JavaScript e Python graças a dois proxies existentes no Pyodide.

Foi possível desenhar círculos coloridos no Canvas à partir de código Python escrito e executado à partir do editor de texto integrado. A execução segue a ordem a seguir.

1. O código Python é extraído do editor de texto via JavaScript.
2. O código Python então é interpretado pelo Pyodide, e executado em seguida.
3. A execução do código Python inclui o download, via gerenciador de pacotes do Python (pip), do módulo Enpyre.

4. As funções constantes neste módulo são capazes de acessar, graças ao Pyodide, a interface `document` do JavaScript.
5. Através desta interface, as funções do PixiJS podem ser executadas. E finalmente, pode-se desenhar gráficos 2D no Canvas do DOM. As funções da parte Python do módulo Enpyre expõe estas funcionalidades de uma maneira conveniente.