

**CAMPUS PORTO VELHO ZONA NORTE**  
**CURSO SUPERIOR DE TECNOLOGIA EM SISTEMAS PARA INTERNET**

LUKAS ROCHA RODRIGUES  
MARINA REGINATO

**PORTUPY: UMA PSEUDO LINGUAGEM BASEADA NA SINTAXE DE  
PYTHON PARA O ESTUDO E APRENDIZAGEM DE LÓGICA DE  
PROGRAMAÇÃO**

**PORTO VELHO**  
**2024**

LUKAS ROCHA RODRIGUES  
MARINA REGINATO

**PORTUPY: UMA PSEUDO LINGUAGEM BASEADA NA SINTAXE DE  
PYTHON PARA O ESTUDO E APRENDIZAGEM DE LÓGICA DE  
PROGRAMAÇÃO**

Trabalho de Conclusão de Curso apresentado como requisito parcial para obtenção do título de Tecnólogo em Sistemas para Internet pelo Instituto Federal de Educação, Ciência e Tecnologia de Rondônia – Campus Porto Velho Zona Norte.

**PORTO VELHO  
2024**

Ficha catalográfica elaborada pelo Sistema Gerador de Ficha Catalográfica do IFRO, com dados informados pelo(a) próprio(a) autor(a).

Rodrigues, Lukas Rocha.

Portupy: uma pseudo linguagem baseada na sintaxe de python para o estudo e aprendizagem de lógica de programação / Lukas Rocha Rodrigues, Marina Reginato, Porto Velho-RO, 2024.  
18 f. : il.

Orientador(a): Prof. M<sup>a</sup> Juliana Braz da Costa.

Trabalho de Conclusão de Curso (Superior de Tecnologia em Sistemas para Internet) – Instituto Federal de Educação, Ciência e Tecnologia de Rondônia - IFRO, Porto Velho-RO, 2024.

1. PortuPy. 2. Programação. 3. Pseudo linguagem. I. Reginato, Marina. II. Costa, Juliana Braz da (orient.). III. Instituto Federal de Educação, Ciência e Tecnologia de Rondônia - IFRO. IV. Título.

**Bibliotecário(a) Responsável:** Celia Reis Sales, CRB-CRB11/955 (Campus Porto Velho Zona Norte)

*CAMPUS* PORTO VELHO ZONA NORTE  
CURSO SUPERIOR DE TECNOLOGIA EM SISTEMAS PARA INTERNET  
**PORTUPY: UMA PSEUDO LINGUAGEM BASEADA NA SINTAXE DE PYTHON  
PARA O ESTUDO E APRENDIZAGEM DE LÓGICA DE PROGRAMAÇÃO**

**Lukas Rocha Rodrigues<sup>1</sup>, Marina Reginato<sup>2</sup>, Juliana Braz da Costa<sup>3</sup>**

Acadêmicos do Curso Superior de Tecnologia em Sistemas para Internet, Turma 2022/1.  
<sup>1</sup>lkscomka@gmail.com, <sup>2</sup>reginato0909@gmail.com, <sup>3</sup>juliana.costa@ifro.edu.br

### RESUMO

A lógica de programação é a base para quem deseja seguir carreira na área de tecnologia da informação. No entanto, muitos estudantes encontram dificuldades em entender a lógica por trás da programação e em aplicá-la de maneira efetiva. Alguns professores buscam formas de mitigar essa dificuldade optando por usar pseudo linguagens para familiarizar os estudantes à lógica e em algum momento posterior migrar para uma linguagem de programação. Nesse cenário, este trabalho apresenta o desenvolvimento da pseudo linguagem "PortuPy", uma ferramenta projetada para facilitar o ensino e a aprendizagem de lógica de programação. Baseada na sintaxe da linguagem Python, desenvolvida para diminuir as dificuldades da transição para uma linguagens de programação real, nesse caso o Python. Além disso, um protótipo de ambiente integrado de desenvolvimento (IDE) é desenvolvido para permitir aos estudantes praticar e aplicar conceitos lógicos em um ambiente prático.

**Palavras-chave:** pseudo linguagem. PortuPy. Programação.

### ABSTRACT

Programming logic is the foundation for those who wish to pursue a career in information technology. However, many students find it challenging to understand the logic behind programming and to apply it effectively. Some teachers seek ways to mitigate this difficulty by opting to use pseudolanguages to familiarize students with logic and then later transition to a real programming language. In this context, this work presents the development of the pseudolanguage "PortuPy," a tool designed to facilitate the teaching and learning of programming logic. Based on Python's syntax, it was developed to ease the transition to a real programming language, in this case, Python. Additionally, a prototype of an integrated development environment (IDE) is developed to allow students to practice and apply logical concepts in a practical environment.

**Keywords:** Pseudolanguages. PortuPy. Programming.

## 1. INTRODUÇÃO

O ramo da tecnologia da informação está em constante evolução, e a programação tornou-se uma habilidade fundamental para aqueles que desejam seguir carreira nessa área. No entanto, muitos estudantes enfrentam dificuldades em compreender a lógica por trás da programação e em aplicá-la de maneira eficaz. Para mitigar essas dificuldades, algumas estratégias educacionais incluem o uso de pseudo linguagens, que são linguagens simplificadas que permitem aos estudantes se familiarizarem com os conceitos de lógica de programação antes de migrar para uma linguagem de programação real (Noschang et al., 2014).

As pseudo linguagens oferecem uma abordagem intermediária que facilita a transição para linguagens mais complexas, permitindo que os alunos desenvolvam uma base sólida em lógica de programação sem a sobrecarga inicial das complexidades de uma linguagem de programação completa. Segundo Cristovão (2008), pseudo linguagens podem ser ferramentas eficazes no ensino de programação, especialmente para iniciantes, pois proporcionam um ambiente mais controlado e menos intimidador.

A escolha da linguagem de programação para iniciar os estudantes em conceitos fundamentais de lógica de programação desempenha um papel crucial no desenvolvimento de habilidades computacionais e na formação de uma base sólida para o aprendizado futuro. Reconhecido por sua simplicidade, clareza e ampla aplicabilidade, o Python tornou-se uma escolha popular em ambientes educacionais.

Este trabalho tem como objetivo desenvolver uma pseudo linguagem, baseada na sintaxe de Python, para facilitar o ensino e a aprendizagem de lógica de programação. A intenção é minimizar as barreiras encontradas por estudantes, especialmente aquelas relacionadas ao uso do inglês em ferramentas de programação e à complexidade das Interfaces de Desenvolvimento Integradas (IDEs). A criação da pseudo linguagem PortuPy é uma solução que surgiu para atender a esses desafios e melhorar a experiência educacional. Além disso, a criação de um protótipo de ambiente integrado de desenvolvimento (IDE) específico para o PortuPy permitirá que os estudantes pratiquem e apliquem conceitos lógicos em um ambiente prático e acessível.

Para alcançar esse objetivo, será necessário: reduzir as barreiras linguísticas no ensino de programação, especialmente relacionadas ao uso do inglês em ferramentas de programação; simplificar o uso de interfaces de desenvolvimento integradas (IDEs), tornando-as mais acessíveis para estudantes que estão aprendendo lógica de programação;

e promover uma experiência de aprendizagem mais envolvente e eficiente, por meio da utilização de uma pseudo linguagem que facilite o processo de ensino e aprendizagem de programação. A importância de ferramentas acessíveis e intuitivas no ensino de programação é amplamente reconhecida (Barbosa & Terra, 2016), e espera-se que a PortuPy possa contribuir significativamente para a melhoria desse processo educacional.

## **2. REFERENCIAL TEÓRICO**

### **2.1. IMPORTÂNCIA DA LÓGICA DE PROGRAMAÇÃO E DIFICULDADE NO APRENDIZADO**

O ensino de programação tem se tornado crucial no contexto educacional moderno, preparando os estudantes para desafios tecnológicos e promovendo habilidades de pensamento crítico e resolução de problemas (Resnick et al., 2009; Wing, 2006). Além disso, muitas disciplinas de cursos da tecnologia de informação apresentam como foco central a programação (Santos et al., 2015).

No entanto, aspectos do ensino de lógica de programação, como a escolha inicial da ferramenta, o uso predominante do inglês nas ferramentas, o processo mental para os alunos novatos que muitas vezes resulta em resistência ao aprendizado, juntamente com a metodologia de ensino empregada, podem tornar o processo de aprendizagem mais difícil.

Noschang et al. (2014) argumentam que as IDEs profissionais são inadequadas para alunos iniciantes em programação, citando características como a predominância do inglês nas ferramentas, a presença de múltiplas janelas e opções de configuração, mensagens de erro complexas e a utilização de depuradores complexos, o que pode ser intimidante para os alunos. Além disso, o suporte dessas IDEs apenas para linguagens de programação profissionais, como Python, Java, JavaScript e C++, podem representar mais uma dificuldade para os alunos iniciantes em programação.

A aprendizagem de algoritmos representa um desafio para os alunos iniciantes em computação, já que representa um processo mental novo para eles. Essa novidade frequentemente resulta em resistência ao aprendizado que pode ser intensificada pela metodologia de ensino utilizada. Na sala de aula, é comum que todos os alunos recebam a mesma abordagem de ensino, dificultando ao professor considerar individualmente o perfil, as metas, as necessidades, as expectativas e o nível de conhecimento de cada um, a fim de proporcionar uma aprendizagem adaptada (Falckembach e Araújo, 2006).

Ademais, diversas pesquisas e autores apontam dificuldades encontradas no processo de aprendizagem, sendo recorrente a ideia de que a capacidade de dominar a lógica de programação é determinante para o fracasso ou sucesso de um aluno ao longo do curso, principalmente durante o período inicial de aprendizagem dessa disciplina (Cristovão, 2008). A superação desses desafios contribuirá não apenas para uma melhor aprendizagem da programação, mas também para o desenvolvimento de habilidades essenciais, como pensamento crítico, criatividade e resolução de problemas, que são cada vez mais valorizadas no mercado de trabalho atual.

## 2.2. LINGUAGENS DE PROGRAMAÇÃO PARA INICIANTES E RECURSOS LÚDICOS NO ENSINO DE PROGRAMAÇÃO

Linguagens para iniciantes devem ser simples, legíveis e permitir um aprendizado progressivo. Segundo Cristovão (2008, p. 30), existem debates acerca da escolha da linguagem inicial para aprendizado e do paradigma a ser adotado, seja ele procedural, lógico, orientado a objetos, entre outros. Além disso, há consensos sobre a importância de uma abordagem inicial que utilize ferramentas acessíveis, permitindo que os alunos experimentem, descubram, testem e cometam erros durante o processo de aprendizagem. Do mesmo modo, a escolha de qual linguagem influencia diretamente na eficiência do aprendizado.

Diante disso, muitos educadores buscam experimentar diferentes abordagens no ensino de programação, especificamente da lógica de programação, como pseudo linguagens de alto nível (Portugol) ou ambientes de programação em blocos como Scratch, MIT App Inventor (Barbosa & Terra, 2016).

Portanto, a busca por ferramentas mais acessíveis e intuitivas pode ser fundamental para melhorar o processo de aprendizagem. O desenvolvimento de ambientes de programação mais amigáveis e adaptados ao nível de conhecimento dos alunos pode contribuir significativamente para tornar a aprendizagem de programação mais eficaz e envolvente.

## 2.3. PYTHON COMO LINGUAGEM DE ENSINO

Ao familiarizar-se com os conceitos de lógica de programação usados em pseudo linguagens como o Portugol ou outras ferramentas, os estudantes migram progressivamente para linguagens de programação reais. Diante disso, muitos educadores optam por apresentar o Python, cuja sintaxe clara e concisa, juntamente com uma

comunidade ativa e bibliotecas abrangentes, contribuem para seu sucesso como linguagem de ensino (Python Software Foundation, 2021).

Entretanto, usar ferramentas alternativas e migrar para o Python pode ser desafiador. O Portugol, por exemplo, segundo a documentação oficial do Portugol Studio Web (2024), é uma ferramenta que apresenta semelhanças significativas com a linguagem C, mas com a distinção de ser escrita em português. Nessa transição, os alunos podem deparar-se com algumas diferenças marcantes, já que o Python possui características próprias. Um exemplo notável é a utilização da indentação como método para delimitar blocos de código, ao contrário do C e do Portugol, que empregam chaves. Além disso, na linguagem C e no Portugol, é necessário declarar o tipo das variáveis, enquanto o Python realiza essa tarefa automaticamente por meio de inferência de tipos.

### **3. METODOLOGIA**

Este estudo centrou-se na criação e implementação de uma pseudo linguagem de programação chamada PortuPy, com o objetivo de facilitar o ensino e a aprendizagem de lógica de programação. A metodologia adotada envolveu cinco etapas interligadas que garantiram a concretização desse objetivo.

Dessa forma, foi adotada a metodologia qualitativa, conforme a classificação de Creswell e Clark (2015), para obter dados que contribuam para a compreensão das dificuldades enfrentadas pelos estudantes ao aprender lógica de programação. Os objetivos da pesquisa são descritivos, segundo Marconi e Lakatos (2017), já que buscam descrever as barreiras encontradas no processo de ensino e aprendizagem de programação, além de compreender a importância do uso de pseudo linguagens como ferramenta facilitadora.

A primeira fase do desenvolvimento consistiu na concepção da pseudo linguagem. Isso envolveu a análise de publicações científicas relativas ao tema nos últimos 5 anos, centrando-se nas dificuldades comuns enfrentadas por estudantes ao aprenderem lógica de programação, especialmente aqueles que não possuem nível avançado no conhecimento de linguagens de programação. Com base nessa análise, decidiu-se criar uma linguagem que utiliza sintaxe da linguagem de programação Python, mas em português.

Após a concepção inicial, a próxima etapa foi a definição da sintaxe da PortuPy. Isso envolveu a tradução de 214 instruções do Python para o Português, incluindo exceções nativas, palavras reservadas e funções nativas. Essas traduções foram então armazenadas em um arquivo JSON, que serve como a base de dados para a pseudo linguagem. O

formato JSON foi escolhido devido à sua simplicidade e eficiência na troca de dados, sendo fácil de interpretar por máquinas e de ler e escrever por seres humanos (Introdução ao JSON, 2024).

Com a sintaxe definida, a terceira etapa foi o desenvolvimento de um ambiente de desenvolvimento integrado (IDE) para a PortuPy, utilizando Python e a biblioteca gráfica Tkinter. Este ambiente foi projetado para ser intuitivo e de fácil utilização, a fim de conferir maior dinamicidade no uso por alunos que estão iniciando em cursos técnicos e de graduação da área de informática.

O núcleo do funcionamento do PortuPy é o seu mecanismo de tradução, que processa os códigos escritos em português e os converte para Python. Este mecanismo utiliza o arquivo JSON como referência para substituir as instruções em português por suas equivalentes em Python. O processo de tradução é realizado em três etapas principais: remoção de strings; substituição das instruções; e, restauração das strings. Esse método garante que a sintaxe do PortuPy seja próxima à do Python, facilitando a transição dos estudantes para uma linguagem de programação real.

Na quarta etapa, o PortuPy passou por uma fase de testes, nesta etapa foram desenvolvidos algoritmos que incluíam uma ampla variedade de exceções nativas, palavras reservadas e funções nativas escritas em PortuPy. O objetivo desses testes era verificar se o resultado produzido pelo PortuPy correspondia exatamente ao resultado obtido no Python, permitindo identificar e corrigir quaisquer bugs ou inconsistências. Essa fase foi fundamental para assegurar que a ferramenta estivesse apta a atender às necessidades educacionais dos estudantes.

Por fim, foi elaborada uma documentação que explica a sintaxe e o uso da PortuPy. A ferramenta e a documentação foram disponibilizadas para download: <https://github.com/lkscomk/doc-portupy>, com o objetivo de facilitar o acesso e incentivar o uso em ambientes educacionais. Portanto, a concentração deu-se para a produção de uma pseudo linguagem de programação, desenvolvendo o PortuPy com o intuito de oferecer uma ferramenta prática e acessível para o ensino da lógica de programação, especialmente voltada para o ensino-aprendizado nesta área da informação.

## **4. RESULTADOS E DISCUSSÃO**

### **4.1. ANÁLISE DA SINTAXE DA LINGUAGEM PYTHON**

Considerando as afirmativas de Resnick e seus colaboradores (2009) acerca de que o ensino de programação tem se tornado crucial no contexto educacional moderno, a criação do PortuPy baseou-se na necessidade de uma ferramenta específica incorporando conceitos familiares de Python. Além disso, os objetivos incluem simplificar a compreensão de estruturas lógicas, facilitar a transição para Python e proporcionar uma experiência educacional envolvente e oferecer uma abordagem única, proporcionando familiaridade com Python desde o início.

O desenvolvimento do PortuPy envolveu uma análise detalhada da sintaxe da linguagem de programação Python, com o objetivo de simplificar conceitos complexos e criar uma interface interativa voltada para estudantes. O processo começou com a identificação de uma ampla gama de instruções e palavras reservadas que poderiam ser traduzidas para o português, facilitando a compreensão do significado de cada componente por parte dos usuários. Durante essa fase, foram examinadas 214 instruções, incluindo exceções nativas, palavras reservadas e funções nativas. Esse trabalho detalhado permitiu criar uma base sólida para a conversão de códigos Python para PortuPy.

A partir dessa análise, foi criado um arquivo JSON para armazenar as relações entre as instruções em português e suas equivalentes em Python. Esse arquivo JSON foi integrado na aplicação, de modo a garantir que os usuários não pudessem acessá-lo diretamente, protegendo a integridade do sistema e mantendo a experiência do usuário fluida e segura.

Na Figura 1, pode-se observar um exemplo básico de equivalência entre as instruções em Python e suas contrapartes em PortuPy. É interessante notar que, ao contrário do Python, o PortuPy é sensível à acentuação, o que implica que instruções com acentos devem ser digitadas corretamente. Essa sensibilidade à acentuação pode ser vista como uma vantagem significativa, pois aproxima ainda mais a pseudo linguagem do Português. Essa aproximação facilita a compreensão e torna o aprendizado mais intuitivo, uma vez que os alunos podem reconhecer e utilizar termos familiares.

**Figura 1:** Comparativo entre Python e PortuPy

Python	PortuPy
while	enquanto
continue	continue
if	se
else	senao
elif	e se
for	para
try	tentar
def	funcao
not	nao
break	quebrar
or	ou
in	em
and	e
is	é
return	devolva

**Fonte:** Elaborado pelos autores, 2024.

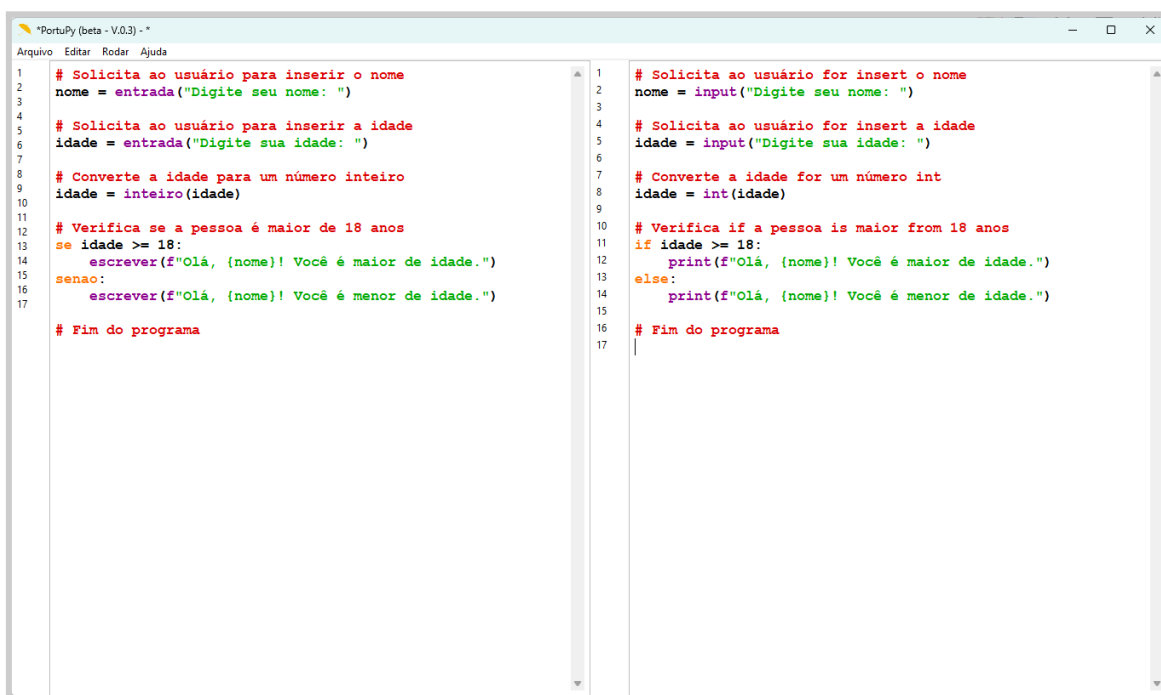
Dessa forma, a inclinação do PortuPy para o uso correto da acentuação e outras particularidades linguísticas do português reflete um cuidado em criar uma ferramenta de ensino que verdadeiramente capte o interesse dos estudantes. Ao utilizar palavras e sintaxes que são naturais para os falantes de português, o PortuPy não apenas ensina programação, mas também o faz de uma maneira que respeita e valoriza o idioma dos alunos. Esse enfoque ajuda a reduzir barreiras de entrada para novos programadores, tornando o aprendizado de conceitos de programação uma experiência mais acessível e menos intimidante.

#### 4.2. IMPLEMENTAÇÃO DO AMBIENTE DE DESENVOLVIMENTO (IDL)

O ambiente integrado de desenvolvimento (IDL) do PortuPy foi desenvolvido integralmente com Python e a biblioteca gráfica Tkinter, segundo a documentação oficial, Tkinter é a interface gráfica padrão do Python, conhecida por sua simplicidade e facilidade de uso. A inclusão de Tkinter permitiu criar uma interface de usuário intuitiva e funcional, facilitando a interação dos estudantes com o ambiente de desenvolvimento. Este IDL foi projetado para ser fácil de usar e é dividido em duas seções principais, como ilustrado na Figura 2: à esquerda, um espaço dedicado à edição de códigos em PortuPy; à direita, uma área para exibição instantânea da tradução desses códigos para Python. Vários recursos

foram integrados a este ambiente, como salvar e abrir arquivos de texto, realizar buscas, ocultar a tela do Python, escrever e transcrever código entre Python e PortuPy, ajustar o tamanho da fonte, desfazer e refazer ações, corrigir indentação e executar o código. Esses recursos são essenciais para proporcionar uma experiência de aprendizado abrangente e eficiente, permitindo que os alunos explorem e pratiquem conceitos de programação de forma prática e acessível.

**Figura 2:** Ambiente de desenvolvimento do PortuPy



```

PortuPy (beta - V.0.3) - *
Arquivo  Editar  Rodar  Ajuda
1  # Solicita ao usuário para inserir o nome
2  nome = entrada("Digite seu nome: ")
3
4
5  # Solicita ao usuário para inserir a idade
6  idade = entrada("Digite sua idade: ")
7
8  # Converte a idade para um número inteiro
9  idade = inteiro(idade)
10
11
12 # Verifica se a pessoa é maior de 18 anos
13 se idade >= 18:
14     escrever(f"Olá, {nome)! Você é maior de idade.")
15 senao:
16     escrever(f"Olá, {nome)! Você é menor de idade.")
17
18 # Fim do programa

1  # Solicita ao usuário for insert o nome
2  nome = input("Digite seu nome: ")
3
4
5  # Solicita ao usuário for insert a idade
6  idade = input("Digite sua idade: ")
7
8  # Converte a idade for um número int
9  idade = int(idade)
10
11
12 # Verifica if a pessoa is maior from 18 anos
13 if idade >= 18:
14     print(f"Olá, {nome)! Você é maior de idade.")
15 else:
16     print(f"Olá, {nome)! Você é menor de idade.")
17
18 # Fim do programa

```

**Fonte:** Elaborado pelos autores, 2024.

Além disso, o pacote PyInstaller foi empregado para compilar o aplicativo Python em um único arquivo executável, conforme descrito no Manual do PyInstaller de 2024. Isso simplifica o processo de distribuição e instalação do ambiente de desenvolvimento, tornando-o mais acessível a todos os usuários, independentemente do seu nível de experiência com programação.

Essa integração entre Python, Tkinter e PyInstaller, ao proporcionar um ambiente robusto e fácil de usar, permite aos estudantes poderem focar em aprender e aplicar conceitos de programação sem se preocupar com as complexidades de configuração e instalação. Este ambiente de desenvolvimento IDL é, portanto, uma ferramenta poderosa para a educação em programação, permitindo que os alunos pratiquem e aprofundem seus conhecimentos de maneira prática e intuitiva.

#### 4.3. FUNCIONAMENTO INTERNO DO PORTUPY: TRADUÇÃO DE CÓDIGOS PORTUGUÊS PARA PYTHON

O funcionamento interno do PortuPy baseia-se na capacidade de processar códigos escritos em português, realizando iterações que substituem as instruções na linguagem PortuPy por suas equivalentes em Python. Esse mecanismo de tradução cria uma estreita semelhança entre a sintaxe do PortuPy e a do Python. Internamente, o algoritmo efetua a transcrição direta das instruções do PortuPy para Python, permitindo que os usuários possam escrever códigos em Python diretamente no ambiente do PortuPy.

Durante o processamento, são realizados três laços de repetição. Na Figura 3, há um exemplo de entrada de um código PortuPy e, na Figura 4, um exemplo do primeiro laço de repetição, que tem como objetivo remover todas as strings (cadeias de caracteres) do código, substituindo-as por valores numéricos aleatórios e armazenando temporariamente as strings em uma estrutura de dados.

**Figura 3:** Código de entrada no PortuPy

```
Código PortuPy - entrada  
  
escrever("ola mundo")  
idade = inteiro(entrada("escreva sua idade: "))
```

**Fonte:** Elaborado pelos autores, 2024.

**Figura 4:** Código do primeiro tipo de laço de repetição no PortuPy

```
Primeiro laço de repetição  
  
escrever(125487)  
idade = inteiro(entrada(365478))
```

**Fonte:** Elaborado pelos autores, 2024.

Na segunda iteração, o código em português é substituído por códigos em Python, com a relação de códigos e instruções que foi armazenada em um arquivo JSON, como ilustrado na Figura 5.

**Figura 5:** Código do segundo laço de repetição no PortuPy

```
Segundo laço de repetição  
  
print(125487)  
idade = int(input(365478))
```

**Fonte:** Elaborado pelos autores, 2024.

Por fim, na terceira iteração, o algoritmo devolve os valores das cadeias de caracteres aos locais de onde foram removidos no primeiro laço, restaurando assim o código original em PortuPy, conforme ilustrado na Figura 6.

**Figura 6:** Código do terceiro laço de repetição no PortuPy

```
Terceiro laço de repetição  
  
print("ola mundo")  
idade = int(input("escreva sua idade: "))
```

**Fonte:** Elaborado pelos autores, 2024.

Além disso, para retomar a ideia de semelhança entre a linguagem sintaxe dos códigos de Python e PortuPy, foi elaborado um código simples que utiliza conceitos fundamentais de lógica de programação, como exceções nativas, palavras reservadas e funções nativas. O objetivo do código é converter um número de dias fornecido pelo usuário em uma representação de idade em anos, meses e dias.

A Figura 7 apresenta o código em Python, enquanto na Figura 8 apresenta o mesmo código em PortuPy.

Figura 7: Exemplo de código em Python

```

1  def converter_dias_para_idade(dias):
2      anos = dias // 365
3      meses = (dias % 365) // 30
4      dias_restantes = (dias % 365) % 30
5
6      return f"{anos} anos, {meses} meses e {dias_restantes} dias"
7
8  # Faz validação de entrada
9  def obter_dias_usuario():
10     while True:
11         try:
12             dias = int(input("Digite um número de dias: "))
13             if dias <= 0:
14                 print("Por favor, digite um número inteiro positivo.")
15             else:
16                 return dias
17         except ValueError:
18             print("Por favor, digite um número inteiro válido.")
19
20 # Obtém os dias do usuário
21 dias = obter_dias_usuario()
22
23 idade = converter_dias_para_idade(dias)
24 print(idade)
25

```

Fonte: Elaborado pelos autores, 2024.

Figura 8: Exemplo de código em PortuPy

```

1  funcao converter_dias_para_idade(dias):
2      anos = dias // 365
3      meses = (dias % 365) // 30
4      dias_restantes = (dias % 365) % 30
5
6      devolva f"{anos} anos, {meses} meses e {dias_restantes} dias"
7
8  # Faz validação de entrada
9  funcao obter_dias_usuario():
10     enquanto Verdadeiro:
11         tentar:
12             dias = inteiro(entrada("Digite um número de dias: "))
13             se dias <= 0:
14                 escrever("Por favor, digite um número inteiro positivo.")
15             senao:
16                 devolva dias
17         deuerro ErroDeValor:
18             escrever("Por favor, digite um número inteiro válido.")
19
20 # Obtém os dias do usuário
21 dias = obter_dias_usuario()
22
23 idade = converter_dias_para_idade(dias)
24 escrever(idade)
25

```

Fonte: Elaborado pelos autores, 2024.

A função “converter\_dias\_para\_idade” realiza a conversão dividindo o número de dias por 365 para obter a quantidade de anos e, em seguida, divide o restante por 30 para obter a quantidade de meses. Os dias restantes são calculados como o restante da divisão por 30.

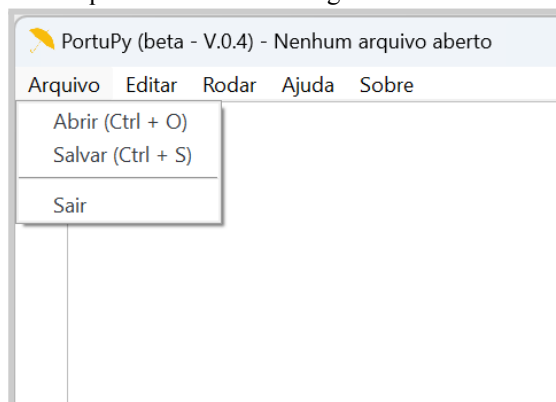
A função “*obter\_dias\_usuario*” é responsável por validar a entrada do usuário, garantindo que seja fornecido um número inteiro positivo. O código utiliza um laço “*while True*” para continuar solicitando a entrada até que um valor válido seja fornecido.

Dessa forma, o PortuPy incorpora integralmente as características do Python, incluindo a indentação, a não necessidade de declarar o tipo das variáveis ao inicializá-las e outras particularidades. Essa funcionalidade é essencial para proporcionar aos estudantes uma experiência de aprendizado fluida e intuitiva, facilitando a transição para o uso de linguagens de programação reais.

#### 4.4. PRINCIPAIS RECURSOS

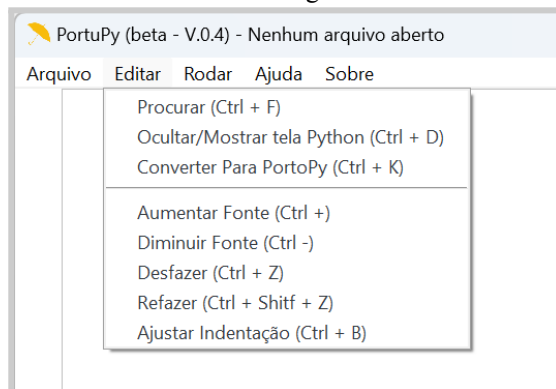
Os recursos implementados no ambiente de desenvolvimento estão ilustrados nas figuras: Figura 9, salvar e abrir arquivos de texto (com instruções PortuPy); Figura 10, ferramenta de procura, ocultar a tela do lado direito (tela do Python), escrever código em Python e transcrever para PortuPy, ajustar tamanho de fonte, recursos de desfazer e refazer, corrigir a indentação; Figura 11, rodar o código; e Figura 12, a documentação da ferramenta.

**Figura 9:** Menu de Arquivo do ambiente integrado de desenvolvimento do PortuPy

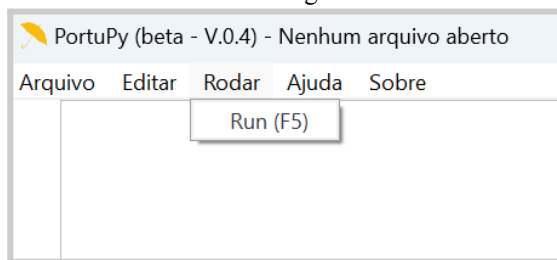


**Fonte:** Elaborado pelos autores, 2024.

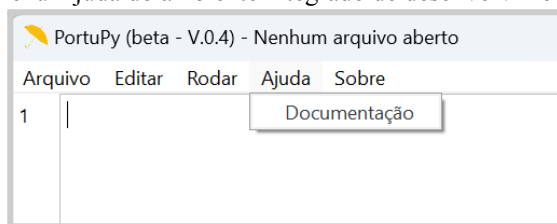
**Figura 10:** Menu Editar do ambiente integrado de desenvolvimento do PortuPy



**Fonte:** Elaborado pelos autores, 2024.

**Figura 11:** Menu Rodar do ambiente integrado de desenvolvimento do PortuPy

**Fonte:** Elaborado pelos autores, 2024.

**Figura 12:** Menu Ajuda do ambiente integrado de desenvolvimento do PortuPy

**Fonte:** Elaborado pelos autores, 2024.

Esses recursos são fundamentais para garantir uma experiência de aprendizado completa e eficiente, permitindo que os estudantes explorem e pratiquem conceitos de programação de maneira prática e acessível. Daí é que se pode conceber a ideia de que a implementação de tais funcionalidades pode contribuir significativamente para o processo educacional, tornando o aprendizado de lógica de programação mais envolvente e eficaz.

#### 4.5. DOCUMENTAÇÃO DA FERRAMENTA

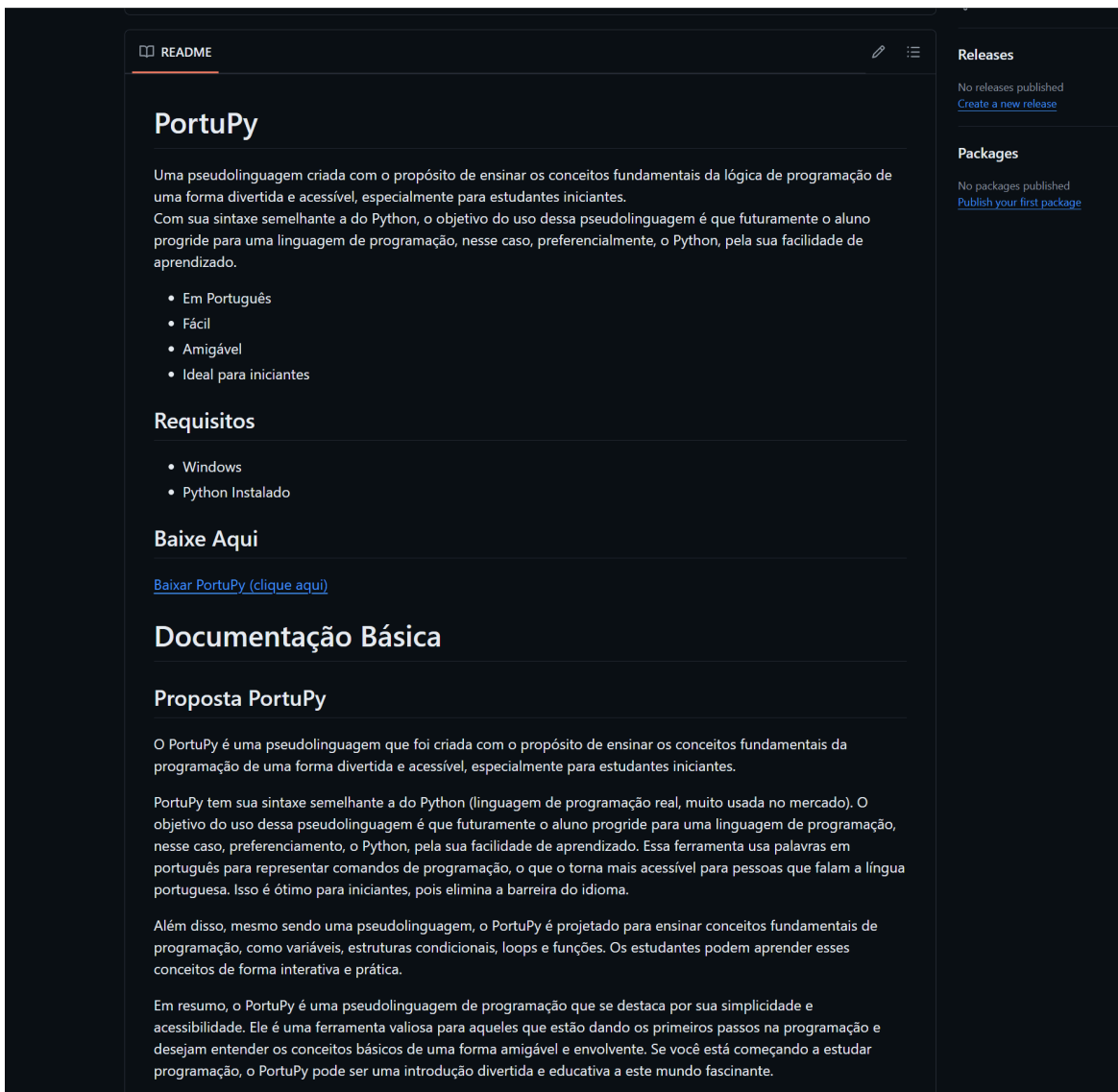
A documentação do PortuPy foi desenvolvida de forma simples e está disponível link: <https://github.com/lkscomk/doc-portupy>, conforme ilustrado na Figura 13. Neste mesmo site, os usuários poderão baixar o software (atualmente o PortuPy está disponível apenas para o Windows). A documentação abrange os principais aspectos da pseudo linguagem, como sua sintaxe, estruturas de controle e exemplos práticos.

Além disso, o projeto PortuPy está disponível no repositório GitHub, onde o código completo pode ser acessado através do link: <https://github.com/lkscomk/PortuPy>. O objetivo deste repositório é permitir que outros desenvolvedores contribuam com melhorias para a ferramenta. A estrutura do projeto está organizada de forma a facilitar a compreensão e o desenvolvimento contínuo, promovendo a colaboração na comunidade de desenvolvedores.

Com tudo isso, o desenvolvimento do programa revela traços experimentais capazes de alcançar os diversos níveis de usuários, permitindo tanto o acesso de iniciantes no aprendizado das linguagens de programação quanto de usuários avançados com

conhecimento necessário para a ampliação e desenvolvimento do programa, com possibilidades de implementações futuras, ao exemplo de seções para a comunidade de usuários no site, oferecendo suporte e recursos adicionais para facilitar o aprendizado e a utilização do PortuPy. O que se alcançou de todo o percurso entre a ideia e sua concretização foi o desenvolvimento de um instrumento capaz de auxiliar.

**Figura 13:** Imagem da documentação do projeto do PortuPy no GitHub



**Fonte:** Elaborado pelos autores, 2024.

## 5. CONSIDERAÇÕES FINAIS

Este estudo teve como objetivo desenvolver a pseudo linguagem PortuPy para facilitar o ensino e a aprendizagem de lógica de programação, abordando as dificuldades enfrentadas por estudantes iniciantes. O artigo buscou descrever as barreiras no

aprendizado de programação e compreender a importância do uso de pseudo linguagens como ferramenta educacional. Os resultados indicam que os objetivos foram alcançados, pois a "PortuPy" demonstrou ser uma ferramenta eficaz para simplificar a compreensão de estruturas lógicas, facilitar a transição para a linguagem Python e proporcionar uma experiência de aprendizagem mais acessível e intuitiva para os alunos.

No entanto, algumas limitações foram identificadas durante o desenvolvimento da ferramenta, uma delas é a capacidade de identificar e corrigir erros de sintaxe e lógica. Rastrear e corrigir erros específicos pode se tornar um desafio para os usuários, especialmente para aqueles com menos experiência em programação. O sistema de detecção de erros no PortuPy requer um alto nível de programação e conhecimento para ser eficaz. Isso ocorre porque o software precisa analisar o código de forma abrangente e identificar possíveis problemas de sintaxe, lógica ou semântica. Além disso, o sistema deve ser capaz de interpretar corretamente a indentação do usuário e fornecer mensagens de erro claras e precisas.

Essa complexidade torna o desenvolvimento do sistema desafiador, exigindo habilidades avançadas de programação e um profundo entendimento da linguagem e de seus possíveis erros. Embora o PortuPy seja capaz de identificar e corrigir alguns erros de sintaxe e lógica, a complexidade de rastrear erros específicos em códigos extensos pode ser um obstáculo. Em muitos casos, o PortuPy não consegue localizar diretamente o erro, deixando para o usuário a tarefa de identificar a linha problemática. Isso pode demandar mais tempo e esforço por parte do usuário, especialmente em códigos extensos ou complexos. Essa dificuldade em rastrear erros pode impactar a experiência do usuário, tornando a depuração de código uma tarefa mais desafiadora no processo de aprendizagem e uso do PortuPy.

Além disso, outra área que pode ser aprimorada é a documentação do PortuPy. Uma documentação mais detalhada e abrangente pode ajudar os usuários a compreender melhor os recursos e funcionalidades da ferramenta, facilitando o seu uso e contribuindo para uma experiência mais satisfatória de aprendizado.

Para futuros trabalhos, recomenda-se a realização de estudos empíricos com alunos para avaliar a eficácia do PortuPy no processo de aprendizagem de programação. Além disso, sugere-se a continuidade do desenvolvimento da ferramenta, com foco na melhoria da precisão da tradução e na correção de possíveis erros, a fim de torná-la ainda mais útil e eficaz para educadores e alunos.

Apesar das limitações e áreas de melhoria identificadas, o desenvolvimento do PortuPy marca um avanço significativo na busca por tornar o ensino de programação mais acessível e compreensível para um público mais amplo. Com a implementação contínua de melhorias e a condução de estudos adicionais, o PortuPy tem o potencial de se tornar uma ferramenta valiosa para educadores e alunos, facilitando o aprendizado da lógica de programação e preparando os estudantes para os desafios tecnológicos do futuro.

## REFERÊNCIAS

BARBOSA, Lucas Lemos; TERRA, Ricardo. PortuCol: uma pseudo linguagem inspirada em C ANSI para o Ensino de Lógica de Programação e Algoritmos. Lavras: Departamento de Ciência da Computação, Universidade Federal de Lavras (UFLA), Brasil, 2016. Disponível em: [http://professores.dcc.ufla.br/~terra/publications\\_files/students/2016\\_ufla\\_lemos.pdf](http://professores.dcc.ufla.br/~terra/publications_files/students/2016_ufla_lemos.pdf). Acesso em: 21 ago. 2024.

CRESWELL, John W.; CLARK, Vicki L. Plano. Pesquisa de Métodos Mistos: Série Métodos de Pesquisa. Penso Editora, 2015. Disponível em [https://books.google.com.br/books?hl=pt-BR&lr=&id=HPyzCAAAQBAJ&oi=fnd&pg=PA7&dq=classifica%C3%A7%C3%A3o+metodo1%C3%B3gica+de+creswell&ots=ZV6RqL2wBI&sig=cGl4Wy4mtKhK\\_hPWj9DKj0aYXW8](https://books.google.com.br/books?hl=pt-BR&lr=&id=HPyzCAAAQBAJ&oi=fnd&pg=PA7&dq=classifica%C3%A7%C3%A3o+metodo1%C3%B3gica+de+creswell&ots=ZV6RqL2wBI&sig=cGl4Wy4mtKhK_hPWj9DKj0aYXW8). Acesso em 20 jun. 2023.

CRISTOVÃO, Henrique Monteiro. Aprendizagem de Algoritmos num Contexto Significativo e Motivador: Um Relato de Experiência. In: Anais do XXVIII Congresso da SBC. 2008. Disponível em: <https://www2.sbc.org.br/esbc2008/pdf/arq0123.pdf>. Acesso em: 21 ago. 2024.

ECMA INTERNATIONAL. Documentação oficial do formato JSON. [S.l.], 2024. Disponível em: <https://www.json.org/json-pt.html>. Acesso em: 22 ago. 2024.

FALCKEMBACH, G. A. M., & Araujo, F. V. (2006). Aprendizagem de algoritmos: dificuldades na resolução de problemas. Universidade Luterana do Brasil ULBRA – Santa Maria – RS - Brasil. Disponível em: <https://www.periodicos.unesc.net/ojs/index.php/sulcomp/article/view/916/909>. Acesso em: 21 ago. 2024.

MARCONI, M. de A.; LAKATOS, E. M. de. Metodologia do trabalho científico: projetos de pesquisa / pesquisa bibliográfica / teses de doutorado, dissertações de mestrado, trabalhos de conclusão de curso. 8. ed. São Paulo: Atlas, 2017.

NOSCHANG, Luiz F.; PELZ, Fillipi; JESUS, Elieser A. de; RAABE, André L. A. Portugol Studio: Uma IDE para Iniciantes em Programação. In: XXXIV Congresso da Sociedade Brasileira de Computação – CSBC, 2014. Disponível em: <https://sol.sbc.org.br/index.php/wei/article/view/10954/10824>. Acesso em: 21 ago. 2024.

PORTUGOL STUDIO WEB, Portugol Studio Web. [S.l.], 2024. Disponível em: <https://dgadelha.github.io/Portugol-Webstudio/>. Acesso em: 21 ago. 2024.

PYINSTALLER DEVELOPMENT TEAM. Documentação oficial do PyInstaller. [S.l.], 2024. Disponível em: <https://pyinstaller.org/en/stable/>. Acesso em: 22 ago. 2024.

PYTHON SOFTWARE FOUNDATION. Python Software Foundation. [S.l.], 2021. Disponível em: <https://www.python.org/>. Acesso em: 21 ago. 2024.

PYTHON SOFTWARE FOUNDATION. Documentação oficial do Tkinter. [S.l.], 2024. Disponível em: <https://docs.python.org/pt-br/3/library/tkinter.html>. Acesso em: 22 ago. 2024.

RESNICK, M., MALONEY, J., MONROY-HERNÁNDEZ, A., RUSK, N., EASTMOND, E., BRENNAN, K., MILLNER, A., ROSENBAUM, E., SILVER, J., SILVERMAN, B., & KAFAI, Y. (2009). Scratch: Programming for All. *Communications of the ACM*, 52(11), 60–67. Disponível em: <https://web.media.mit.edu/~mres/papers/Scratch-CACM-final.pdf>. Acesso em: 21 ago. 2024.

SANTOS, Antunes; GORGÔNIO, Arthur; LUCENA, Amarildo; GORGÔNIO, Flavius. A Importância do Fator Motivacional no Processo Ensino-Aprendizagem de Algoritmos e Lógica de Programação para Alunos Repetentes. *In: WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO (WEI)*, 23. , 2015, Recife. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2015. p. 168-177. ISSN 2595-6175. DOI: <https://doi.org/10.5753/wei.2015.10233>. Acesso em: 21 ago. 2024.