

## Análise dos erros mais comuns de aprendizes de programação que utilizam a linguagem Python

Galileu Santos de Jesus<sup>1</sup>, Kleber Santos<sup>1</sup>, Jaine Conceição<sup>1</sup>,  
Alberto Costa Neto<sup>1</sup>.

<sup>1</sup>Departamento de Computação – Universidade Federal de Sergipe.  
Av. Marechal Rondon, Jardim Rosa Elze, São Cristóvão - SE, 49100-000.

galilasmb@gmail.com, klebertarcisio@yahoo.com.br

jainecs@dcomp.ufs.br, alberto@ufs.br

**Abstract.** *This paper presents an analysis of the user database of the on-line judge The Huxley, illustrating the most frequent run-time errors committed by the programming learners of the Python programming language. 143 types of errors were found distributed in 20 classes. The most common errors were of invalid syntax, when the student misuses some commands, such as if, for, while, input and print and forgets the token ':' of the conditional command if, variable declaration, when using an undeclared variable and type errors when operations are performed with different types of data. Knowing the most common errors makes it possible for teachers to emphasize them during their classes.*

**Resumo.** *Este trabalho apresenta uma análise da base de dados dos usuários do juiz on-line The Huxley, ilustrando os erros em tempo de execução mais cometidos pelos aprendizes de programação da linguagem de programação Python. Foram encontrados 143 tipos de erros, distribuídos em 20 classes, sendo os erros mais comuns são de sintaxe inválida, quando o aluno faz o uso incorreto de alguns comandos, como if, for, while, input e print e esquece o token ':' do comando condicional if, declaração de variáveis, quando é feito o uso de uma variável que não foi definida anteriormente e erros de tipos, quando são realizadas operações com tipos diferentes de dados. Conhecer os erros mais comuns permite aos professores enfatizá-los durante suas aulas.*

### 1. Introdução

Várias são as dificuldades enfrentadas pelos alunos no decorrer do processo de ensino-aprendizagem de programação. O trabalho de (LAHTINEN; ALA-MUTKA; JÄRVINEN, 2005) cita que a retirada de erros, modularização e conceitos como recursão e ponteiros são algumas das dificuldades. Todavia, a maioria está relacionada à combinação e à utilização apropriada dos conceitos básicos de programação, de acordo com Caspersen e Kolling (2009). Segundo Miyadera, Huang e Yokoyama (2000) uma das maiores dificuldades de um estudante de programação é entender cada passo da execução do programa. Assim como entender e aprender a gramática de uma linguagem de programação, consertar erros de sintaxe em um programa, desenvolver novos algoritmos, escrever um novo programa, depurar e consertar erros em um programa, em ordem crescente de dificuldade. Assim, os alunos parecem entender os conceitos e estruturas que compõem uma

linguagem de programação, mas não sabem como utilizar corretamente, ou seja, entendem como é a parte léxica de uma linguagem, mas não sabem como utilizar de forma correta a parte sintática ou semântica.

Alguns pesquisadores catalogaram alguns erros mais comuns de programação. O trabalho de Schorsch (1995) relata 111 mensagens de diagnóstico diferentes (incluindo 62 para erros de sintaxe, 21 para erros de lógica e 28 para erros léxicos) da linguagem Pascal. Tais erros foram recolhidos e classificados através de um estudo informal de todos os 520 alunos matriculados no curso de Introdução à Ciência da Computação - CS110. Já o trabalho de Hristova et al. (2003) elaborou uma lista com 62 erros de programação da linguagem Java, dos quais 20 foram identificados como essenciais. O trabalho de Ahmadzadeh, Elliman e Higgins (2005) utilizou a linguagem Java para analisar 108.652 registros de erros dos estudantes, tendo como resultado 36 erros de sintaxe, 63 semânticos e 1 léxico, constatando que o erro mais comum foi o de não definir uma variável.

Este trabalho traz como contribuição uma análise dos erros mais comuns em Python, cometidos pelos usuários da ferramenta The Huxley<sup>1</sup> - um ambiente virtual de aprendizagem e juiz *on-line*, agrupando-os como erros de execução e sintaxe. Conhecendo os erros mais comuns, professores podem enfatizá-los em suas aulas introdutórias.

## 2. Extração de dados

Neste trabalho foi utilizado o juiz *on-line* The Huxley por disponibilizar uma API de acesso à sua base de dados relacional, permitindo acesso a diversas funções disponíveis no sistema, possibilitando assim a extração de diversos dados acerca das submissões de programas de aprendizes para realização de uma análise estatística.

Diante da base de dados da ferramenta The Huxley, temos como questão central de pesquisa a seguinte pergunta: *Quais os erros mais comuns cometidos pelos usuários da ferramenta que utilizam a linguagem Python?* Para responder a essa questão, foram definidas as seguintes questões específicas de pesquisa:

Questão de Pesquisa 1: Qual o percentual de submissões por linguagem?

Questão de Pesquisa 2: Qual o percentual de submissões corretas e não corretas por linguagem?

Questão de Pesquisa 3: Qual o resultado das submissões realizadas na linguagem Python?

Foram obtidas as estatísticas das submissões, assim como a base de dados com os erros das submissões realizadas na linguagem Python através de requisições no padrão REST com JSON, como ilustra o arquivo localizado no caminho *Codes/baseErros.py* disponível em um compartilhamento do Google Drive<sup>2</sup>.

A Figura 1 ilustra a notação da mensagem de erro mapeada, onde, inicialmente é identificada a linha que a contém, posteriormente sua classe e por fim, o subtipo do erro com sua descrição. Esta notação é específica da linguagem Python.

Foi criado um *script* de acesso à API da ferramenta, para acessar a base de submissões realizadas desde a sua concepção. Os dados extraídos da base de dados

---

<sup>1</sup>[www.thehuxley.com](http://www.thehuxley.com)

<sup>2</sup>[gg.gg/galileuFiles](https://gg.gg/galileuFiles)

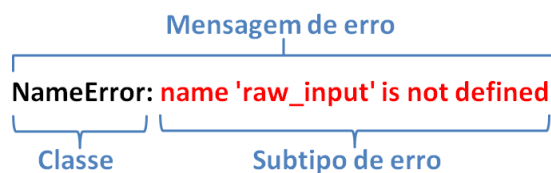


Figura 1. Mensagem de erro mapeada.

encontram-se no caminho *Codes/baseErros.txt* e os *scripts* em *Codes/Cod\_errorMsg.py* e *Codes/RequestData.py*. Em Jan/2017 foram encontradas 667.836 submissões.

### 3. Resultados

Foi realizada a análise em toda a base, obtendo-se 20 classes de erros, subdivididas em 143 mensagens de erros, como ilustra a planilha disponível *on-line*<sup>3</sup>. Ao analisar e catalogar os 143 subtipos de erros, foi feita a pesquisa do significado dos erros na internet e a verificação dos códigos-fontes com os respectivos erros, disponível em<sup>2</sup>. Para cada subtipo de erro, que provém de um tipo de mensagem de erro, foi associada uma descrição que visa explicar o subtipo do erro.

Para responder à questão de pesquisa 1: Qual o percentual de submissões por linguagem? A Figura 2, ilustra o percentual de submissões por linguagem, onde C possui o maior percentual, com 53,44%, seguida por Python, C++, Java, Octave e Pascal.

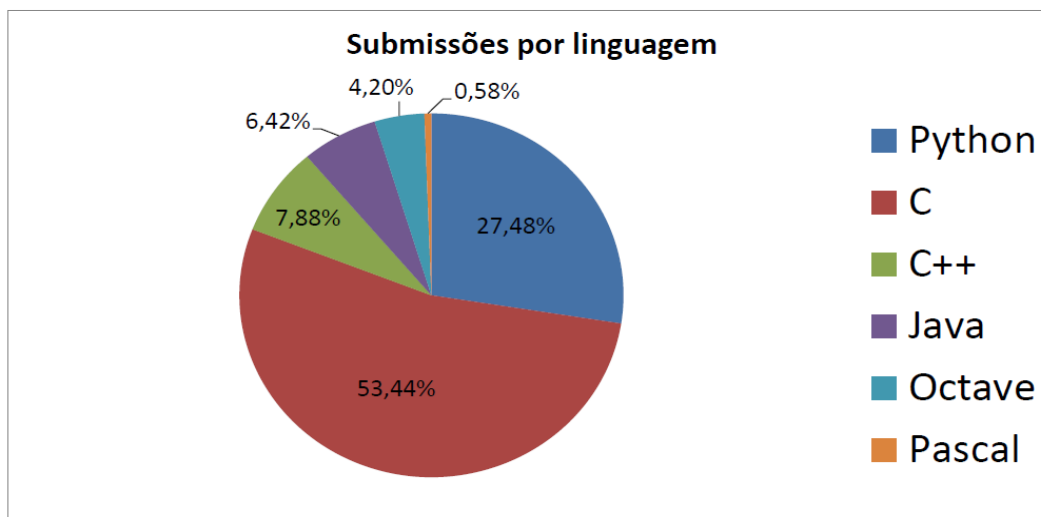
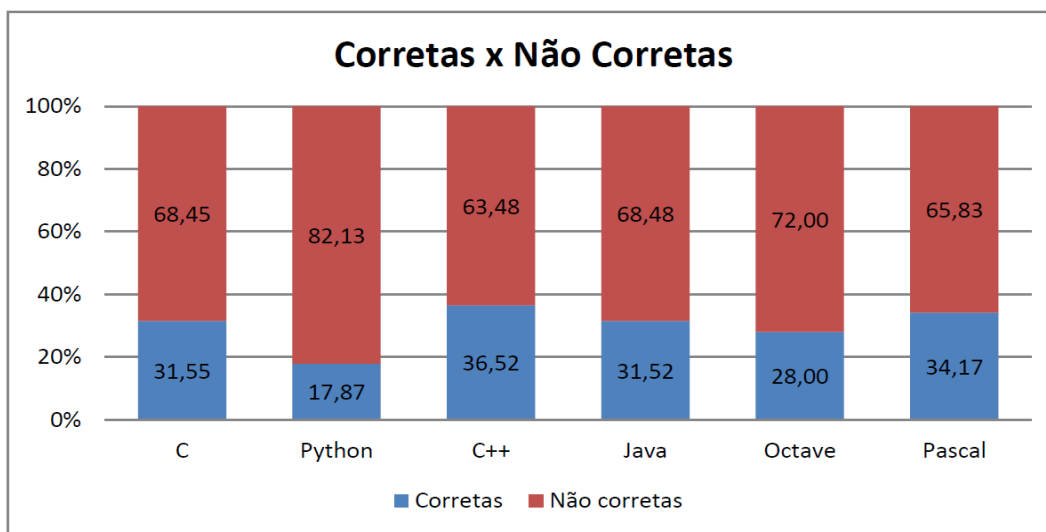


Figura 2. Submissões por linguagem.

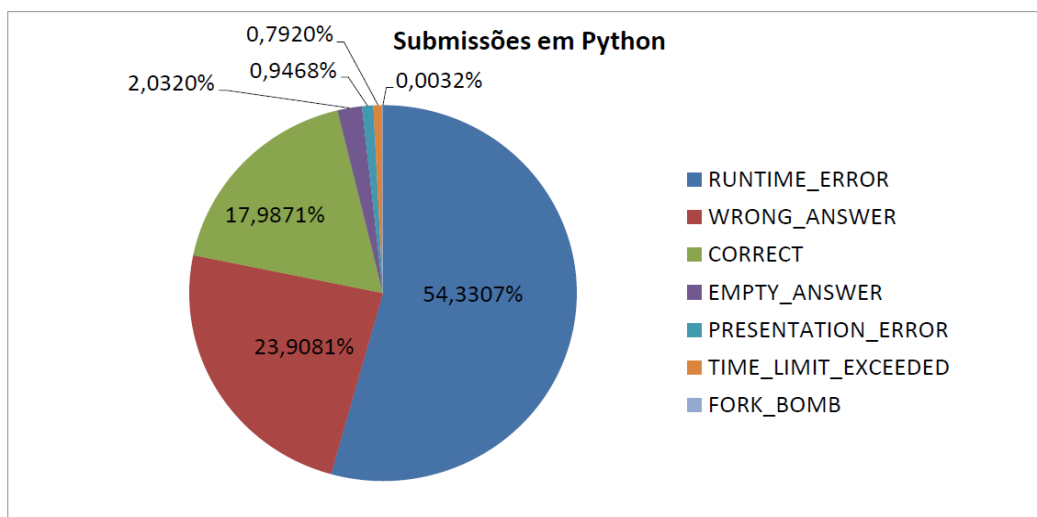
Para responder à questão de pesquisa 2: Qual o percentual de submissões corretas e erradas por linguagem? Foi feita a análise de todas as submissões erradas, correspondendo a cerca de 71,97% do total. A Figura 3 ilustra as submissões corretas e erradas de acordo com cada linguagem possível para submissão no The Huxley. Python se destaca pelo maior percentual não aceito, já que concentra 82,13% das submissões erradas contra 17,87% corretas. Na sequência, estão Octave, Java, C, Pascal e C++.

<sup>3</sup>[gg.gg/requi](http://gg.gg/requi)



**Figura 3. Submissões consideradas correta e não corretas de acordo com cada linguagem.**

Para responder à questão de pesquisa 3: Qual o resultado das submissões realizadas na linguagem Python? Foi realizada a busca pelo resultado das submissões referentes somente à linguagem Python. Obteve-se que a maior porcentagem são de *RUNTIME\_ERROR*, com cerca de 54,33%, seguidas de *WRONG\_ANSWER*, *CORRECT*, *EMPTY\_ANSWER*, *PRESENTATION\_ERROR*, *TIME\_LIMIT\_EXCEEDED* e *FORK\_BOMB*, como ilustra a Figura 4. A explicação de cada categoria de erro usada nos juízes *on-line* está disponível em<sup>4</sup> e no trabalho de Revilla, Manzoor e Liu (2008).



**Figura 4. Resultado das submissões realizadas em Python.**

A linguagem Python possui maior percentual de submissões não corretas, conforme ilustrado na Figura 3, mesmo não sendo a linguagem com maior quantidade de submissões, como ilustra a Figura 2. Dentre as não corretas, 54,33% de suas submissões

<sup>4</sup>[maratona.ime.usp.br/manualBOCA.html](http://maratona.ime.usp.br/manualBOCA.html)

são do tipo (*RUNTIME\_ERROR*), ou seja, com erro em tempo de execução.

#### 4. Considerações Finais e Trabalhos Futuros

Foram mapeados todos os erros encontrados na base de dados do juiz *on-line* The Huxley, sendo aprofundado o estudo nos erros da Linguagem Python por esta ter apresentado uma incidência de erros bem maior que as outras. Obteve-se 143 mensagens de erros, agrupadas em 20 classes.

Há fortes indícios de que os erros mais comuns são: (a) de sintaxe, quando o aluno faz o uso incorreto de alguns comandos, como *if*, *for*, *while*, *input* e *print* e esquece o token ':' do comando condicional *if*; (b) declaração de variáveis, quando é feito o uso de uma variável que não foi definida anteriormente e (c) erros de tipos, quando são realizadas operações com tipos diferentes de dados, em sua maioria, usando um literal como inteiro. É de suma importância que os professores enfatizem os erros mais comuns, explicando melhor o uso do ':' em Python, assim como os tipos de dados, conversão entre eles e sua compatibilidade. É importante salientar que os erros catalogados podem não refletir com a mesma proporção entre aprendizes não usuários do The Huxley.

Como trabalho futuro, será feito o mesmo levantamento nas outras linguagens de programação aceitas pelo The Huxley, como C, C++, Java, Octave e Pascal.

#### Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

#### Referências

- AHMADZADEH, M.; ELLIMAN, D.; HIGGINS, C. An analysis of patterns of debugging among novice computer science students. *ACM SIGCSE Bulletin*, ACM, v. 37, n. 3, p. 84–88, 2005.
- CASPERSEN, M. E.; KOLLING, M. Stream: A first programming process. *ACM Transactions on Computing Education (TOCE)*, ACM, v. 9, n. 1, p. 4, 2009.
- HRISTOVA, M. et al. Identifying and correcting java programming errors for introductory computer science students. In: ACM. *ACM SIGCSE Bulletin*. [S.l.], 2003. v. 35, n. 1, p. 153–156.
- LAHTINEN, E.; ALA-MUTKA, K.; JÄRVINEN, H.-M. A study of the difficulties of novice programmers. In: ACM. *Acm Sigcse Bulletin*. [S.l.], 2005. v. 37, p. 14–18.
- MIYADERA, Y.; HUANG, N.; YOKOYAMA, S. A programming language education system based on program animation. In: *Proceedings of Educational Uses of Information and Communication Technologie, in IFIP 16th World Computer Congress*. [S.l.: s.n.], 2000. p. 258–261.
- REVILLA, M. A.; MANZOOR, S.; LIU, R. Competitive learning in informatics: The uva online judge experience. *Olympiads in Informatics*, v. 2, p. 131–148, 2008.
- SCHORSCH, T. Cap: an automated self-assessment tool to check pascal programs for syntax, logic and style errors. In: ACM. *ACM SIGCSE Bulletin*. [S.l.], 1995. v. 27, n. 1, p. 168–172.