



UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE CIÊNCIAS EXATAS E NATURAIS
FACULDADE DE COMPUTAÇÃO
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Renan Filip Costa Balieiro

LABORATÓRIO DE PROGRAMAÇÃO EM PYTHON: LABPY

Belém

2018

Renan Filip Costa Balieiro

LABORATÓRIO DE PROGRAMAÇÃO EM PYTHON: LABPY

Trabalho de conclusão de curso de graduação apresentado ao Instituto de Ciências Exatas e Naturais da Universidade Federal do Pará como um dos requisitos para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Eloi Luiz Favero

Belém

2018

Renan Filip Costa Balieiro

LABORATÓRIO DE PROGRAMAÇÃO EM PYTHON: LABPY

Trabalho de conclusão de curso de graduação apresentado ao Instituto de Ciências Exatas e Naturais da Universidade Federal do Pará como um dos requisitos para a obtenção do título de Bacharel em Ciência da Computação de Computação.

Aprovado em: ____ de _____ de _____.

BANCA EXAMINADORA

Profª. Dra. Marcelle Pereira Mota – UFPA

Prof. Dr. Raimundo Viegas Junior – UFPA

Prof. Dr. Eloi Luiz Favero – UFPA (orientador)

AGRADECIMENTOS

Acima de tudo a Deus pelo dom da vida, por sempre me dar forças para continuar a caminhada e por ter me dado grandes amigos e pais incríveis que tornaram esta jornada mais prazerosa.

Aos meus pais, que sempre me apoiaram incondicionalmente, e sei que continuarão me apoiando em meus objetivos. Agradecer por terem sido os responsáveis por eu conhecer o mundo da computação, pois foram eles que me deram o meu primeiro computador, no natal de 2002, e foi o melhor presente que já recebi em minha vida e que foi o divisor de águas para eu me interessar por computação.

A minha mãe Maria do Socorro, que sempre esteve ao meu lado e fez de tudo para que eu tivesse uma vida boa, espero algum dia poder retribuir tudo o que ela fez por mim e ainda faz até hoje. Ao meu pai Rubens, que apesar de distante, algumas vezes, nunca esteve ausente e sempre me incentivou a estudar e também a ter o hábito da leitura.

A minha avó Ana, que é a minha segunda mãe e sempre cuidou de mim, como se fosse seu filho, sempre me deu apoio e conselhos valiosos que nunca esquecerei.

Ao meu tio Marco Antônio, que foi minha inspiração para ingressar no curso, que sempre me deu apoio, dicas e conselhos valiosos que levarei para toda a minha vida, e também contribuiu através de livros que me deu de presente, e que foram essenciais para a minha formação.

A minha tia Círia que me deu conselhos, me apoiou e me acolheu em sua casa nos momentos mais complicados em minha vida.

Ao meu primo João Victor, o qual considero um irmão, por sempre estar ao meu lado e sempre me ajudar nos momentos em que pensei que nada ia dar certo, por sua confiança em mim e na minha capacidade.

Aos amigos que conheci ao longo desses quatro anos de curso, em especial Adriano, Filipe, Giordanna, George, Ricardo, Ramón, Yvan e Paulo, que sempre me ajudaram e sei que

sempre poderei contar com eles. Além de muitos outros que não foram citados mas que também são muito importantes pra mim.

Por fim, mas não menos importante, agradeço a todos os professores que conheci durante o curso, que me ensinaram tudo o que sei sobre computação e que se tornaram grandes amigos. Em especial ao professor Elói, meu orientador, que permitiu que todo este trabalho valesse a pena, foi um verdadeiro amigo, ajudou quando eu mais precisei me apoiando e acreditando em meu potencial, fico honrado de poder ter trabalhado ao seu lado no desenvolvimento deste trabalho, e ter sido seu aluno durante o curso.

Todos os que conheci e fizeram parte desta jornada, mesmo que não lembrados aqui, amigos, familiares e professores, foram de alguma forma importantes e possuem minha gratidão por tudo o que puderam ensinar. Sendo assim, não me resta mais nada a dizer a vocês, a não ser o meu mais sincero obrigado!

“A sabedoria suprema é ter sonhos bastante grandes para não se perderem de vista enquanto os perseguimos”

(Willian Faulkner)

RESUMO

As disciplinas de algoritmos e programação são essenciais para os estudantes dos cursos de computação, já que estas são a base para diversas outras matérias. No entanto, o ensino e aprendizagem de programação ainda constituem um grande desafio, para os professores e alunos, isso se deve a fatores como, por exemplo, método de estudo inadequado, dificuldades na abstração da linguagem, estratégias de ensino dos professores não se adequam a todos os alunos, entre outros. Os Ambientes Virtuais de Aprendizagem (AVA) ao longo dos anos vem se tornando um modelo promissor para solucionar problemas durante o processo de ensino-aprendizagem de algoritmos e programação de computadores. Estas ferramentas oferecem suporte ao gerenciamento de informações, postagem de materiais de estudo, seja pelo professor ou aluno, entre outras características, para que professores e alunos possam estabelecer uma melhor interação, tornando o ensino mais adequado às necessidades de ambos. Com base nesses problemas, este trabalho propõe um AVA, como um laboratório virtual de programação via Web, para auxiliar o ensino de programação em Python. Ele apresenta uma abordagem de avaliação automática das atividades de programação, dando um feedback imediato (com múltiplas re-submissões) ao estudante, auxiliando assim o estudante em seu processo de aprendizagem e o professor na correção e gerenciamento de listas de exercícios e provas, seja na modalidade à distância ou em aulas laboratoriais. Uma proposta de avaliação foi elaborada, levando-se em consideração aspectos de qualidade de *software* e pedagógicos presentes no sistema, e será aplicada posteriormente em uma turma programação, para que assim seja possível obter resultados mais adequados.

Palavras-chave: Ambiente virtual de aprendizagem, Avaliação automática, Python, Qualidade de Software.

ABSTRACT

The algorithm and programming disciplines are essential for students in computer courses, as these are the basis for many other subjects. However, the teaching and learning of programming are still a great challenge, for teachers and students, this is due to factors such as inadequate method of study, difficulties in language abstraction, teacher teaching strategies do not fit to all students, among others. The Virtual Learning Environments (AVA) over the years has become a promising model for solving problems during the teaching-learning process of algorithms and computer programming. These tools support the management of information, posting study materials, either by the teacher or student, among other characteristics, so that teachers and students can establish a better interaction, making teaching more suited to the needs of both. Based on these problems, this work proposes an AVA, as a virtual web programming laboratory, to aid in the teaching of programming in Python. It presents an approach of automatic evaluation of programming activities, giving immediate feedback (with multiple re-submissions) to the student, thus helping the student in his / her learning process and the teacher in correcting and managing lists of exercises and tests, whether in the distance mode or in laboratory classes. An evaluation proposal was elaborated, taking into account aspects of software and pedagogical quality present in the system, and will be applied later in a programming class, so that better results can be obtained.

Keywords: Automatic evaluation, Python, Software quality, Virtual Learning Environments.

LISTA DE FIGURAS

Figura 1 - Ranking das linguagens mais populares	15
Figura 2 - Representação dos quatro eixos em um Ambiente Virtual de Aprendizagem	16
Figura 3 - Página principal do livro interativo	21
Figura 4 - Página de conteúdo presente no livro interativo	22
Figura 5 - Exemplo de uso do Skulpt	23
Figura 6 - Saída gerada pelo Skulpt.....	24
Figura 7 - Esquema de funcionamento do WebSQL.....	26
Figura 8 - Exemplo de fluxograma do sistema de feedback automático.....	28
Figura 9 - Estrutura da Arquitetura do LabSQL.....	29
Figura 10 - Arquitetura do WeFiLab.....	31
Figura 11 - Resultados obtidos a partir do sistema JavaTool.....	34
Figura 12 - Funcionamento do Sistema The Huxley.....	37
Figura 13 - Representação geral do Labpy.....	40
Figura 14 - Disposição geral dos módulos no sistema	42
Figura 15 - Área de cadastro de questões	43
Figura 16 - Exercícios cadastrados no sistema.....	44
Figura 17 - Exemplo de execução de um exercício.....	45
Figura 18 - Feedback fornecido ao aluno após a execução do programa.....	45
Figura 19 - Área do aluno.....	46
Figura 20 - Área principal do administrador	47
Figura 21 - Seção de administração de turmas	47
Figura 22 - Seção de administração de usuários.....	48
Figura 23 - Seção de administração de exercícios.....	48
Figura 24 - Seção de administração de módulos	49
Figura 25 - Seção de administração de provas	49
Figura 26 - Área de login no sistema.....	50
Figura 27 - Área de provas	50
Figura 28 - Área de exercícios.....	51
Figura 29 - Vídeo disponível no sistema	52
Figura 30 - Saída gerada pela execução de um programa em um módulo de estudo.....	52
Figura 31 - Alerta de erro de sintaxe no LabPy.....	53
Figura 32 - Resultado gerado pelo sistema.....	54
Figura 33 - Esquema de Qualidade Web	56

LISTA DE TABELAS

Tabela 1 - Comparação entre os trabalhos relacionados e esta proposta.....	38
Tabela 2 - Comparação entre os trabalhos relacionados e esta proposta.....	38
Tabela 3 - Questões sobre usabilidade	57
Tabela 4 - Questões sobre funcionalidade	58
Tabela 5 - Questões sobre satisfação pedagógica.....	59
Tabela 6 - Questões sobre eficiência	60
Tabela 7 - Questões sobre confiabilidade.....	60
Tabela 8 - Questão discursiva.....	60
Tabela 9 - Questões sobre aspectos gerais da linguagem Python.....	61
Tabela 10 - Detalhes de implementação.....	68

LISTA DE SIGLAS

AVA	Ambiente Virtual de Aprendizagem
IDLE	Integrated Development and Learning Environment
PHP	PHP: Hypertext Preprocessor
CSS	Cascading Style Sheets
HTML	HiperText Markup Language
IEEE	Instituto de Engenheiros Eletricistas e Eletrônicos
SBC	Sociedade Brasileira de Computação
UFPA	Universidade Federal do Pará
USP	Universidade de São Paulo

SUMÁRIO

1 INTRODUÇÃO	13
1.1 VISÃO GERAL.....	13
1.1.1 Linguagens de Programação.....	14
1.1.2 Ambientes Virtuais de Aprendizagem.....	15
1.2 MOTIVAÇÕES E JUSTIFICATIVA	17
1.3 OBJETIVOS	17
1.3.1 Objetivo geral	17
1.3.2 Objetivos específicos.....	17
1.4 METODOLOGIA.....	18
1.5 CONTRIBUIÇÃO	18
1.6 ESTRUTURA DO TRABALHO	19
2 REFERENCIAL TEÓRICO	20
2.1 LINGUAGEM PYTHON.....	20
2.2 COMO PENSAR COMO UM CIENTISTA DA COMPUTAÇÃO.....	21
2.3 AMBIENTE VIRTUAL DE APRENDIZAGEM	22
2.4 SKULPT	23
2.5 CONCLUSÕES DO CAPÍTULO	24
3 TRABALHOS RELACIONADOS	25
3.1 WEBSQL.....	25
3.2 AMBIENTE PARA ENSINO DE PROGRAMAÇÃO COM FEEDBACK AUTOMÁTICO DE EXERCÍCIOS	27
3.3 LABSQL.....	28
3.4 WEFILAB: A WEB-BASED WIFI LABORATORY PLATFORM FOR WIRELESS NETWORKING EDUCATION.....	30
3.5 NETAULA	32
3.6 AMBIENTE INTEGRADO À PLATAFORMA MOODLE PARA APOIO AO DESENVOLVIMENTO DAS HABILIDADES INICIAIS DE PROGRAMAÇÃO	33
3.7 TSTVIEW.....	35
3.8 THE HUXLEY	36
3.9 CONCLUSÕES DO CAPÍTULO	37
4 LABPY.....	39

4.1 PROPOSTA DO LABPY	39
4.2 ESTRUTURA DO CONTEÚDO	41
4.2.1 Exercícios	43
4.3 ÁREA DO PROFESSOR	46
4.4 ÁREA DO ALUNO	49
4.5 EXECUÇÃO DE PROGRAMAS EM PYTHON	53
4.6 CONCLUSÕES DO CAPÍTULO	54
5 ANÁLISE DA PROPOSTA	55
5.1 AVALIAÇÃO DO LABPY	55
5.2 PROPOSTA DE AVALIAÇÃO DO LABPY	55
5.2.1 Usabilidade	56
5.2.2 Funcionalidade	57
5.2.3 Satisfação Pedagógica	58
5.2.4 Confiabilidade e Eficiência	59
5.2.5 Aspectos Gerais da Linguagem Python	60
5.3 CONCLUSÕES DO CAPÍTULO	61
6 CONSIDERAÇÕES FINAIS	62
6.1 SUGESTÕES PARA TRABALHOS FUTUROS	63
REFERÊNCIAS BIBLIOGRÁFICAS	64
APÊNDICE A – Diagrama Entidade-Relacionamento do Sistema	67
APÊNDICE B – Trechos de Código do Sistema	68
APÊNDICE C – Trecho de Código sobre Módulo	71
APÊNDICE D – Trecho de Código sobre Banco de Dados	78

1 INTRODUÇÃO

1.1 VISÃO GERAL

A programação é uma disciplina imprescindível para os estudantes dos cursos de computação. Isso se deve muito ao fato de estabelecer a base para diversos campos nos quais a informática se aplica, proporcionar uma melhor compreensão e utilização de tecnologias computacionais atuais e também incentivar o desenvolvimento do raciocínio lógico e da capacidade na resolução de problemas (Moreira & Favero, 2009).

Porém, o ensino e aprendizagem de programação ainda constituem um grande desafio, para os professores e alunos. Uma série de problemas estão associados a estas dificuldades, em Gomes et al. (2008) estas são citadas as seguintes:

- O método de ensino tradicional não se adequa às necessidades dos alunos em diversos casos, devido a fatos como: ausência de feedback e supervisão personalizada as necessidades de cada aluno; estratégias elaboradas por professores que não contemplam os estilos de aprendizagem dos alunos; ensino de conceitos dinâmicos, realizados através de materiais estáticos, como apresentações de slide, desenhos no quadro, textos, entre outros.
- O método de estudo realizado por vários alunos não é adequado ao aprendizado de programação, já que exige estudo prático e intensivo. Diversos alunos se limitam apenas a assistir às aulas e estudar por livros texto, deixando a parte prática de lado.
- As habilidades e algumas atitudes dos alunos também estão relacionadas as dificuldades de aprendizagem, devido a fatores como: dificuldade na resolução de problemas, que necessitam de diversas competências a exemplo da compreensão do problema, relacionar o problema com os já conhecidos, reflexão mais elaborada; falta de persistência, já que os alunos tendem a desistir de problemas em que a solução não é encontrada de forma simples e rápida.
- A natureza da programação também envolve uma série de problemas, já que a programação requer uma hierarquia de competências como um elevado nível de abstração, generalização, transferência e pensamento crítico. Algumas linguagens de programação possuem uma forte complexidade, devido ao fato delas possuírem um propósito profissional e não de aprendizado, sendo extensivas e com diversos detalhes sintáticos complexos, que requer do aluno uma concentração na construção do algoritmo e também na recordação de regras de sintaxe.

Diante dos problemas e dificuldades expostos, fica claro que ainda há muito a se fazer para melhorar o ensino de programação nos cursos de computação. Sendo assim, existe a necessidade de obter soluções que possam se adequar ao período atual, auxiliar o trabalho dos professores e o progresso de seus alunos na disciplina de programação.

1.1.1 Linguagens de Programação

Existem diversos tipos de linguagens de programação que permitem aos alunos aprenderem as técnicas e exercitarem a lógica aprendida na disciplina de algoritmos. Dentre as linguagens mais utilizadas e relevantes atualmente estão: *C*, *C++*, *Java*, *C#*, *Python*, *Javascript*, *PHP*, entre outras. Porém, diversas delas envolvem uma complexidade alta de aprendizado devido a sua sintaxe e regras que estão atreladas a linguagem (Silva & Trentin, 2016), fator que constitui um problema durante a fase de aprendizagem de programação.

Dentre as diversas linguagens utilizadas, uma das mais importantes e que vem em crescente é a linguagem de programação *Python*. São diversos os fatores que a tornam uma das mais populares, seja entre os aprendizes ou entre desenvolvedores avançados. A Figura 1 apresenta uma tabela contendo dados estatísticos das linguagens de programação mais utilizadas em Fevereiro de 2017 e em Fevereiro de 2018. *Python* está localizada, atualmente, na quarta posição. Dentre estes fatores Grandell et al. (2006) cita alguns, os mais relevantes:

- Python possui uma sintaxe mais intuitiva e simples, comparada a linguagens como o Java e o C++;
- Python é uma linguagem dinamicamente tipada, que reduz sua notação assim como sua complexidade;
- Python propõe uma maneira estruturada e indentada na escrita de programas, fator que torna os códigos semelhantes a um pseudo-código, facilitando seu entendimento;
- Python possui ampla comunidade open-source, pois é uma linguagem gratuita e amplamente utilizada, que trás um ambiente integrado IDLE (Integrated Development and Learning Environment), e uma grande quantidade de tutoriais, livros, materiais de cursos, exercícios além de uma documentação extensa disponível na web.

Figura 1 - Ranking das linguagens mais populares

Fev 2018	Fev 2017	Alteração	Linguagem de Programação	Classificações	Alteração
1	1		Java	14.988%	-1.69%
2	2		C	11.857%	+3.41%
3	3		C++	5.726%	+0.30%
4	5	▲	Python	5.168%	+1.12%
5	4	▼	C#	4.453%	-0.45%
6	8	▲	Visual Basic .NET	4.072%	+1.25%
7	6	▼	PHP	3.420%	+0.35%
8	7	▼	JavaScript	3.165%	+0.29%
9	9		Delphi/Object Pascal	2.589%	+0.11%
10	11	▲	Ruby	2.534%	+0.38%

Fonte: (TIOBE Software BV, 2000), adaptada

Na Figura 1, pode-se perceber que de Fevereiro de 2017 a Fevereiro de 2018, houve um aumento na utilização de *Python*, subiu de quinto lugar para quarto no ranking das linguagens mais utilizadas. Portanto, através das características expostas somadas a sua ampla utilização, seja no meio empresarial ou educacional, ela caracteriza-se como uma linguagem de fácil assimilação tornando-se uma boa candidata para o ensino de programação nos momentos iniciais dos cursos, diferente das linguagens C, C++ ou Java que são linguagens mais voltadas aos usuários mais avançados em programação.

1.1.2 Ambientes Virtuais de Aprendizagem

Com o avanço da Internet, da tecnologia e da mídia, houve a possibilidade de novos caminhos de acesso a informações e produção de conhecimento, fator que impactou os métodos de ensino e aprendizagem. Além disso, o surgimento das redes sociais exerceu um forte impacto em escolas, na educação à distância e na educação presencial (Pereira & França, 2013). Sendo assim, ferramentas de ensino tais como os AVAs, tornaram-se importantes para gerar novas metodologias de ensino, auxiliando o trabalho dos educadores.

Os AVAs constituem os componentes nos quais aprendizes e professores participam de interações online dos mais variados tipos, sendo estas interações consideradas de aprendizado online (Sneha J. M. & Nagaraja G. S., 2013). Estas ferramentas oferecem suporte ao gerenciamento de informações, postagem de materiais de estudo, seja pelo professor ou aluno, bem como permitem a comunicação síncrona e assíncrona através de fórum, chat, entre outros (Pereira & França, 2013). Sendo assim, professores e alunos podem estabelecer uma

interação simultânea, tornando o processo de ensino mais adequado às necessidades de ambos os lados.

Segundo Cybis Pereira et al. (2007), os principais recursos tecnológicos que são normalmente utilizados nestes ambientes são divididos em quatro eixos, Figura 2, sendo estes:

- Informação e documentação, que permite apresentar informações institucionais do curso, veicular conteúdos e materiais didáticos, fazer *upload* e *download* de arquivos e oferecer suporte ao uso do ambiente;
- Comunicação, que proporciona ao ambiente uma facilidade de comunicação síncrona (Fórum) e assíncrona (chat);
- Gerenciamento pedagógico e administrativo, que permite ao professor acessar as avaliações e o desempenho dos alunos, realizar consultas, entre outros;
- Produção que permite o desenvolvimento de atividades e resoluções de problemas dentro do ambiente.

Figura 2 - Representação dos quatro eixos em um Ambiente Virtual de Aprendizagem



Fonte: (Pereira, Schmitt, & Dias, 2007)

As características e recursos de um AVA são propícias a melhorar a qualidade do ensino dos alunos, facilitando a comunicação entre os aprendizes e seu professor. No quesito ensino de programação, um AVA é importante ferramenta, já que um de seus eixos permite a resolução prática de exercícios dentro do próprio ambiente. Além disso, com as propriedades dos outros três eixos pode sanar diversas das dificuldades apresentadas anteriormente.

1.2 MOTIVAÇÕES E JUSTIFICATIVA

O ensino de programação nos cursos de computação, ainda possui algumas deficiências. Além das mencionadas na Seção 1.1, os conceitos iniciais de programação são de difícil assimilação para os aprendizes, já que estes não estão familiarizados com conceitos abstratos, tais como: *arrays*, construções de *loops*, estruturas de decisões, entre outros.

Outra importante dificuldade a ser considerada, é a comunicação entre o professor e seus alunos, já que diversas vezes o professor não consegue dar atenção a todos os alunos, e nem atender às necessidades específicas de cada um. Na maioria dos casos alguns alunos entram nos cursos de computação sem conhecimento prévio da matéria. Eles demoram um tempo para se adaptar a rotina de aprendizado, fator que gera problemas no entendimento final dos assuntos mais importantes durante o ensino de programação.

Sendo assim, diante dos problemas expostos, os alunos não se sentem incentivados a ter o hábito de pesquisar e deixam a prática de programação de lado. Através deste trabalho, um AVA é desenvolvido para facilitar o ensino dos conceitos básicos de programação utilizando a linguagem *Python*, para auxiliar professor e estudante no processo de aprendizagem.

1.3 OBJETIVOS

1.3.1 Objetivo geral

O objetivo deste trabalho é propor uma ferramenta Web para o ensino de programação em *Python*. Este objetivo geral é dividido em objetivos específicos que devem ser explorados e cumpridos.

1.3.2 Objetivos específicos

Os objetivos específicos deste projeto de pesquisa são:

- Realização de estudos mais elaborados de trabalhos relacionados a AVAs, buscando adquirir um domínio mais amplo nesta área, conhecendo os pontos positivos e negativos de cada trabalho;
- Elaboração de um AVA utilizando tecnologias recentes, permitindo uma mobilidade do ambiente, de Desktops e a dispositivos móveis;
- Utilização de um conteúdo atualizado e revisado para que os aprendizes tenham uma experiência e aprendizado mais efetivo da linguagem *Python*;

- Avaliação automática de programas feitos em linguagem *Python*, informando a aproximação da solução do aluno com a “solução de referência”;
- Integração entre o ensino de conceitos de programação com exemplos de fragmentos de programas executáveis, permitindo a interatividade no ambiente;

Este trabalho é um esforço para obter um ambiente de auxílio no ensino da linguagem de programação *Python*, facilitando o trabalho do professor nas atividades práticas de programação como listas de exercícios e avaliações, e no gerenciamento de suas turmas.

1.4 METODOLOGIA

Após um levantamento de dados do trabalho, é necessário evidenciar a metodologia utilizada ao longo de todo o processo. Inicialmente foi feito um trabalho para obtenção de conhecimento sobre laboratórios virtuais de programação e materiais para ensino de *Python*, a partir de um levantamento bibliográfico, incluindo artigos de conferências, a exemplo da SBC e IEEE. Focamos em AVAs, ambientes de programação e tecnologias recentes para implementação.

A partir do levantamento bibliográfico foi elaborada uma proposta de ambiente. Na proposta foram consideradas várias tecnologias: o PHP, o JavaScript, o HTML5, o CSS3 e o framework Bootstrap. A proposta foi implementada e testada pelo desenvolvedor. Na implementação foi utilizada uma abordagem de desenvolvimento ágil, centrada em prototipação.

Uma proposta de avaliação foi elaborada com o intuito de avaliar o sistema através da aplicação de um questionário contendo 30 questões que englobam seis categorias distintas (usabilidade, funcionalidade, satisfação pedagógica, confiabilidade, eficiência e aspectos gerais da linguagem Python). Testes com usuários finais serão realizados em uma turma de programação da UFPA, como atividade prática, para que resultados mais adequados possam ser obtidos.

1.5 CONTRIBUIÇÃO

O trabalho tem o intuito de ajudar os alunos iniciantes na aprendizagem de programação tendo como foco os conceitos iniciais da matéria, visa incentivar um estudo mais prático e intensivo de programação permitindo que o aluno procure respostas mais adequadas para resolver os exercícios.

Facilita o trabalho do professor, através do gerenciamento fácil de suas turmas, da correção automática de provas e do acompanhamento através do relatório de atividades dos

estudantes. Estas características permitem que o professor possa localizar os alunos que estão com mais dificuldades na matéria e dar uma atenção especial.

1.6 ESTRUTURA DO TRABALHO

Feitas as asserções iniciais acerca do trabalho, o restante do mesmo será constituído de mais 5 capítulos, que serão distribuídos como segue:

- **Capítulo 2: Referencial Teórico.**
Aborda as tecnologias e técnicas utilizadas para constituir a base de pesquisa do trabalho, apresentando um breve histórico e descrição de cada conceito.
- **Capítulo 3: Trabalhos Relacionados.**
Aborda os trabalhos relacionados mais relevantes para o desenvolvimento do ambiente LabPy proposto, não serão apresentados todos os lidos, mas os que tiveram uma influência significativa e que se assemelham ao que foi proposto no trabalho. Sempre expondo as peculiaridades presentes em cada um dos sistemas e realizando uma comparação entre estes e o que foi desenvolvido.
- **Capítulo 4: LabPy.**
Descreve conceitualmente o ambiente de ensino LabPy, apresentando a organização do conteúdo didático proposta para que o sistema possa proporcionar aos alunos uma maior interação com os tópicos de programação em Python e a análise detalhada de diversas atividades de programação, que ajuda o estudante na resolução de diversos problemas propostos.
- **Capítulo 5: Análise da Proposta.**
Analisa a proposta de avaliação do sistema LabPy, utilizando o questionário elaborado.
- **Capítulo 6: Considerações Finais.**
Apresenta as considerações finais acerca das ideias desenvolvidas ao longo do trabalho, onde os resultados coletados durante a implementação dos estudos de caso serão avaliados.

2 REFERENCIAL TEÓRICO

Para o desenvolvimento deste trabalho, foram utilizados diversos conhecimentos que foram fundamentais ao longo de sua construção. O primeiro foi a linguagem de programação Python, que é a base e inspiração para a criação do sistema de ensino; tem como base um estudo de caso de ensino de programação e o ambiente do livro interativo “Como Pensar Como um Cientista da Computação”, tradução do livro “How to Think Like a Computer Scientist: Interactive Version”, de Brad Miller e David Ranun, feita pela USP (Universidade de São Paulo) em sua plataforma Panda (DCC, IME & USP, 2015), que é muito utilizado como base de conteúdo em diversos ambientes virtuais; o conceito de AVA serviu para a construção do ambiente; por fim utilizamos diversas tecnologias para desenvolvimento web.

Uma visão geral de todos estes referenciais é descrita nos tópicos subsequentes. É importante ressaltar que foi utilizada uma máquina com Sistema Operacional Ubuntu, com uma versão de Kernel 4.4.0-93, para o desenvolvimento da proposta, e o ambiente Web foi projetado usando a pilha LAMP (Linux, Apache, MySQL, PHP) por ser de fácil manutenção, robusta e com uma comunidade de desenvolvimento ativa.

2.1 LINGUAGEM PYTHON

Python é uma linguagem de programação desenvolvida com o objetivo de ser produtiva e legível, ou seja, que tenha uma facilidade para produção de um código bom, simples de compreender e manter.

Entre suas principais características estão: o uso de poucos caracteres especiais, fator que permite a linguagem se aproximar de um pseudo-código; suporte a múltiplos paradigmas de programação, como por exemplo orientação a objetos e programação funcional; uma grande biblioteca padrão, possuindo classes e métodos para diversos domínios de aplicação; por fim, Python é uma linguagem livre e multi-plataforma, ou seja, programas que são escritos em uma plataforma podem ser executados na maioria das plataformas existentes sem modificações.

Foi a linguagem escolhida para ser o tema de ensino por ser uma linguagem expressiva e que mais se aproxima da linguagem algorítmica, sendo assim mais fácil de aprender e dominar os assuntos. Permitir que o aluno tenha um foco maior em dominar a lógica de programação, dando a possibilidade dele se concentrar especificamente no problema a ser resolvido e ter menos preocupações com a estrutura de comandos presentes na linguagem.

Outro fator que influenciou a escolha da linguagem como instrumento de aprendizado, foi sua comunidade ativa que dá um suporte aos usuários iniciantes e avançados, o que

permite aos usuários colaborarem uns com os outros em diversos aspectos. A linguagem também serve para propósito geral como, por exemplo, criar planilhas, utilizar um banco de dados remoto, ler uma página na Internet, entre outros que tornam a linguagem mais atrativa a um público geral.

2.2 COMO PENSAR COMO UM CIENTISTA DA COMPUTAÇÃO

Diante das especificações elucidadas anteriormente, a linguagem de programação Python foi escolhida para ser o principal instrumento de ensino presente no ambiente Web projetado neste trabalho, tendo seu conteúdo baseado no livro *How to Think Like a Computer Scientist*. Este é um dos principais livros sobre o ensino de Python, e é utilizado como referência em diversos sites que são voltados ao ensino da linguagem, como é o exemplo da plataforma Panda da USP, a qual este ambiente de ensino Web também foi baseado.

O livro está disponível de forma digital através do site (<http://interactivepython.org/runestone/static/thinkcspy/index.html>), onde o aluno pode seguir o conteúdo presente no site, sem ter a necessidade de seguir uma ordem específica. O site possui diversos exercícios e exemplos que estão disponíveis aos usuários para que possam exercitar e fixar o conteúdo que foi explicado durante a seção.

A Figura 3 apresenta a página principal do livro interativo digital, onde é possível perceber a estruturação do conteúdo feita para que o aluno tenha a possibilidade de escolher a qual achar necessidade.

Figura 3 - Página principal do livro interativo

Como Pensar Como um Cientista da Computação: Edição Interativa

Sobre este Projeto

Tabela de Conteúdo

- [Declarações](#)
- [1. Introdução Geral](#)
 - [1.1. O Caminho do Programa](#)
 - [1.2. Algoritmos](#)
 - [1.3. A Linguagem de Programação Python](#)
 - [1.4. Executando Python neste Livro](#)
 - [1.5. Mais Sobre Programas](#)
 - [1.6. O que é Depuração?](#)
 - [1.7. Erros de Sintaxe](#)
 - [1.8. Erros de Execução](#)
 - [1.9. Erros de Semântica](#)
 - [1.10. Depuração Experimental](#)
 - [1.11. Linguagens Formais e Naturais](#)
 - [1.12. Um Primeiro Programa Típico](#)

Fonte: (Miller, 2012), adaptada

Na Figura 4 há um exemplo de uma página de conteúdo presente no livro. É possível perceber uma amostra de código, onde pode-se executar no próprio site; tem o propósito de demonstrar o comportamento e funcionamento das instruções presentes na linguagem, assim como apresentar a lógica presente no código; deixando de ser um estudo só mental para ser um estudo prático vivencial pela animação do código.

Figura 4 - Página de conteúdo presente no livro interativo

2.2. Variáveis e Tipos de Dados

Um **valor** um valor é uma das coisas fundamentais — como uma palavra ou número — que um programa manipula. Os valores que vimos até o momento são `5` (o resultado quando fazemos a adição `2 + 3`), e `"Hello, World!"`. Nós frequentemente nos referimos a esses valores como **objetos** e usaremos as palavras valor e objeto indiscriminadamente.

Nota

Na verdade, o `2` e o `3` que são parte da adição acima são também valores (objetos).

Esses objetos são classificados em **classes**, ou **tipos de dados** diferentes: `4` é um *inteiro*, e `"Hello, World!"` é uma *string*, que recebe esse nome pois contém uma sequência de letras ou caracteres. Você (e o interpretador) podem identificar strings pois estes estão envolvidos por aspas.

Fonte: (Miller, 2012), adaptada

2.3 AMBIENTE VIRTUAL DE APRENDIZAGEM

AVAs são softwares educacionais utilizados para o apoio no ensino dos alunos em diversos temas, auxiliando o professor dentro e fora da sala de aula. Estes AVAs são ambientes para exposição de conteúdo, atividades laboratoriais, avaliação e acompanhamento da situação dos estudantes de uma turma.

Os AVAs são de fácil uso e gerência por parte dos professores, facilitando diversas tarefas como, por exemplo, correção de atividades, identificação de problemas específicos dos alunos por parte do professor, entre outros (Pereira, Schmitt, & Dias, 2007). Algumas das vantagens de utilizar o AVA são:

1. interação entre o aluno e o computador;
2. possibilidade de dar atenção individual ao aluno;
3. controle do ritmo de aprendizagem.

O uso dos AVA vem sendo cada vez mais comum, seja no ensino médio e fundamental quanto superior, na graduação e até na pós-graduação. Isso se deve ao fato da Internet estar cada vez mais integrada a vida dos estudantes. Utilizar AVAs para fins educacionais tem sido benéfico ao ensino, pois agrega valor ao aprendizado do aluno.

2.4 SKULPT

Skulpt é um interpretador feito em Javascript, que tem por objetivo rodar programas em linguagem Python no próprio Browser, ou seja, não necessita de pré-processadores, plugins ou suporte do lado do servidor. Ele é muito utilizado por diversos ambientes que possuem o intuito de ensinar a linguagem, como por exemplo, o Panda da USP e o livro interativo *How to Think Like a Computer Scientist*, entre outros.

O Skulpt está disponível gratuitamente pelo site (<http://www.skulpt.org/>); é um projeto Open Source; basta apenas importar diretamente no documento HTML (ou baixar os arquivos pelo gerenciador npm). A Figura 5 apresenta um exemplo de funcionamento presente no site do Skulpt, onde há uma função, *genr*, que tem o objetivo de gerar números de 1 a n através do laço *while*; inserir os itens em uma lista e imprimir estes na tela. A Figura 6 apresenta a saída gerada a partir da execução deste trecho de código.

Figura 5 - Exemplo de uso do Skulpt

Demo

O código é executado inteiramente no seu navegador, portanto, não se sinta obrigado a "travar o servidor", você só vai tropeçar. [Ajuda](#), ou exemplos: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#). Ctrl-Enter para executar.

```

1 def genr(n):
2     i = 0
3     while i < n:
4         yield i
5         i += 1
6
7 print list(genr(12))
8

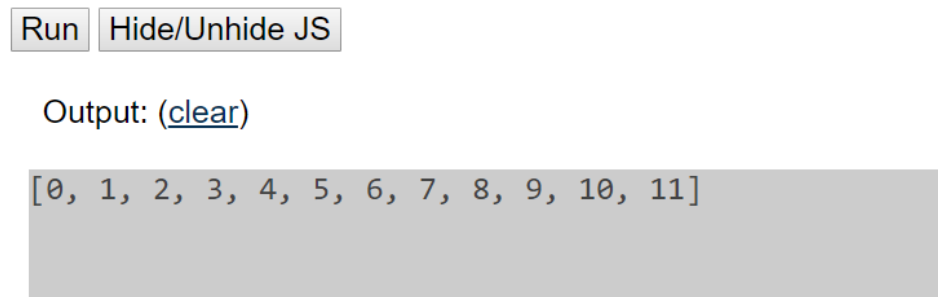
```

Depurador Skulpt

 digite 'help' para verificar os comandos

Fonte: (Skulpt, s.d.), adaptada

Figura 6 - Saída gerada pelo Skulpt



Fonte: (Skulpt, s.d.)

Sendo assim, o Skulpt foi escolhido para ser o interpretador da linguagem Python; ele fornece uma interpretação dos programas através do próprio Browser.

2.5 CONCLUSÕES DO CAPÍTULO

A linguagem Python que vem ganhando uma grande notoriedade dentre as principais linguagens utilizadas no mercado (TIOBE Software BV, 2000). Devido ao fato de ser uma linguagem de fácil compreensão e robusta, com um aprendizado rápido.

Com o auxílio do interpretador Skulpt é possível obter um laboratório no AVA que permite aos estudantes executarem seus programas e observarem o seu comportamento.

3 TRABALHOS RELACIONADOS

Este capítulo aborda os trabalhos relacionados à nossa proposta, ou seja, alguns trabalhos que dizem respeito à AVA para ensino de programação.

3.1 WEBSQL

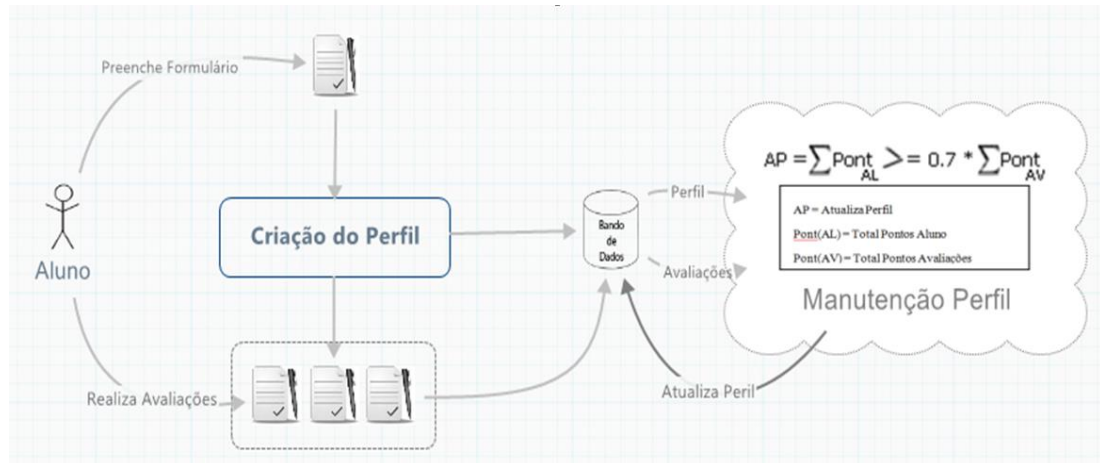
Este ambiente (Raiol, Natal, & Favero, 2014), propõe uma ferramenta de apoio ao processo de ensino e aprendizagem de SQL baseado na recomendação de objetos de aprendizagem e nas competências do aluno em SQL. Na proposta, um AVA é desenvolvido com o intuito de ser utilizado como suporte metodológico durante o processo de ensino-aprendizagem da linguagem SQL. Um repositório de 138 questões, em 4 módulos, é utilizado para compor os exercícios e as avaliações dos alunos.

Este sistema utiliza um procedimento para recomendar as questões que mais se adequam a cada aluno, baseado em suas competências em relação a linguagem SQL. Outra funcionalidade importante é a possibilidade de realização de exercícios, atividades em grupo e avaliações, com feedback em tempo real das soluções que foram propostas. Ele possui um recurso de avaliação que motiva o aluno a buscar soluções mais adequadas para realizar as consultas.

Para modelar o perfil do aluno, o autor entende como competência a capacidade que o aluno possui para realizar consultas, utilizando a linguagem SQL. O perfil do usuário é obtido, inicialmente, de forma explícita a partir da aplicação de um questionário. É utilizado o algoritmo de aprendizado de máquina para realizar a classificação do aluno num perfil, buscando por integrantes vizinhos mais próximos. Após o perfil inicial ser estabelecido, a manutenção é feita de forma automática através do monitoramento das interações com o ambiente.

No sistema de avaliação, todas as pontuações obtidas pelos alunos são somadas, se este resultado for maior ou igual a 70% do total de pontos das atividades e avaliações, o sistema irá atualizar seu perfil, classificando o usuário em um nível superior ao que está no momento da avaliação. A Figura 7 apresenta um esquema de como estes procedimentos ocorrem no sistema.

Figura 7 - Esquema de funcionamento do WebSQL



Fonte: (Raiol, Natal, & Favero, 2014)

Outro mecanismo presente no WebSQL é o de recomendações, onde é feita de forma colaborativa, em que a interação do aluno na resolução de suas atividades é monitorada, controlando a quantidade de vezes que pode tentar para chegar à solução ótima de determinada questão. Após verificar que o aluno já utilizou 70% das possibilidades e ainda não chegou a uma solução ideal, o sistema busca os alunos que já resolveram a questão e que possuem o mesmo perfil do aluno alvo da indicação, levando em consideração a questão, a quantidade de vezes tentadas e se o aluno já chegou à solução ótima. Ele indica os cinco primeiros alunos encontrados através de uma lista contendo o nome e o e-mail dos alunos recomendados.

Outra forma de recomendação utilizada é a baseada em conteúdo, a qual é apresentada no sistema através da indicação de questões específicas, para constituir a avaliação do aluno, baseadas em seu perfil. Através desta funcionalidade, o professor cadastra as questões no repositório de questões do ambiente, após gerar a avaliação através de recomendação, o professor seleciona a turma, e quantidade de questões que deseja na avaliação. O módulo Recomendação obtém o perfil de cada aluno e filtra as questões que mais se adequam ao seu perfil, produzindo avaliações personalizadas e individuais para cada aluno.

Como experimento, os autores testaram inicialmente a utilização do sistema com 10 alunos de graduação da disciplina de Banco de Dados do curso de Sistemas de Informação, e em um segundo momento com mais duas turmas da graduação, dos cursos de Análise e Desenvolvimento de Sistemas, na Disciplina de Banco de Dados, com 25 alunos, e a outra do curso de Ciência da Computação, na disciplina de Linguagem de Programação para Banco de Dados, com 50 Alunos, onde o comportamento do ambiente foi observado, para o feedback das suas funcionalidades.

Os resultados obtidos pelo ambiente se mostraram satisfatórios perante os testes, sendo geradas 2 avaliações, através de recomendação, contendo 12 questões cada, onde alcançaram um total de 248 respostas.

3.2 AMBIENTE PARA ENSINO DE PROGRAMAÇÃO COM FEEDBACK AUTOMÁTICO DE EXERCÍCIOS

Este artigo, (Moreira & Favero, 2009), apresenta um estudo sobre técnicas para realizar a avaliação automática de algoritmos que possibilitam ao estudante um feedback imediato, sendo capaz de auxiliar o professor na correção e avaliação de exercícios de programação, seja no contexto de ambiente virtual ou em aulas em laboratório.

No sistema, há a utilização de dois modelos de regressão linear, para que seja possível complementar a abordagem de testes. O primeiro é baseado em indicadores compostos por métricas de engenharia de software: quantidade de uso de funções da linguagem, número de palavras reservadas, número de declarações, número de linhas do programa, complexidade de McCabe e volume de Halstead. O segundo utilizou indicadores de similaridade estrutural baseado em n-gramas. Em ambas as técnicas o objetivo é medir o quanto a solução de um estudante se aproxima em termos de complexidade da resposta supostamente ideal.

Os testes realizados foram compostos por respostas e exercícios de programação em Java básico, sem levar em consideração recursos de orientação a objetos.

Foram selecionados 20 exercícios para compor o treinamento e para realizar a escolha destes foram levadas em consideração as seguintes características: presença de estruturas de controle, estruturas de repetição, declarações, uso de funções da linguagem e respostas com um tamanho razoável. Em cada atividade um grupo de 5 respostas foram coletadas, sendo uma destas a resposta principal, que vale 10 pontos, totalizando 100 respostas. As respostas foram avaliadas e atribuídas notas por 2 professores, sendo a nota final a média aritmética entre estas 2 notas. Em caso de discrepância entre elas, as respostas foram revisadas novamente até que os professores entrassem em consenso.

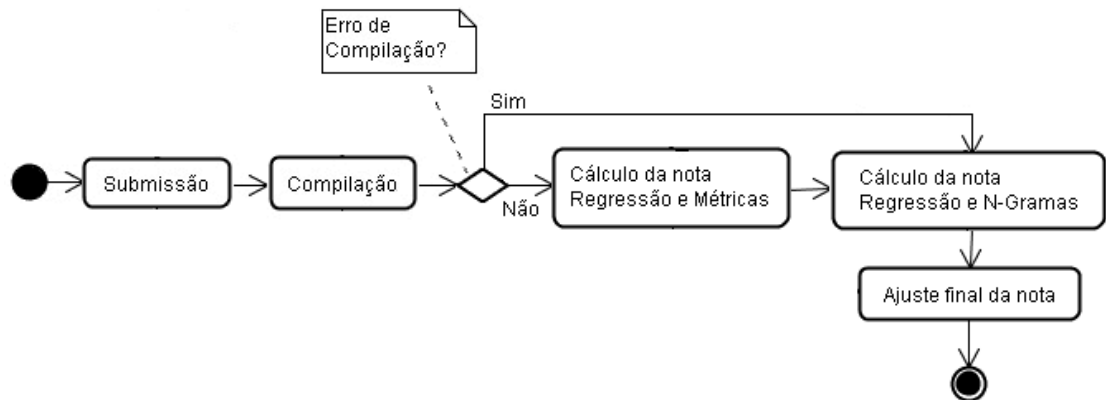
Após a comparação das duas formas de avaliação automática de programação, concluiu-se que ambas as medidas são consideradas boas na avaliação automática de programação, acertando na maioria das vezes na atribuição de notas.

A proposta foi implementada no Moodle, um sistema que realiza o gerenciamento de cursos para aprendizagem online, a partir do módulo Tarefas foi desenvolvida a categoria de submissão de exercícios em Java com avaliação automática online. Perante a necessidade de

compilar os exercícios submetidos pelos alunos, em Java, foi desenvolvida uma solução que se comunica com o sistema através de Web Service.

O lado servidor é um sistema Java, que aguarda uma solicitação do lado cliente, PHP. No momento em que um aluno submete uma resposta, o PHP invoca uma chamada ao Java, após a compilação deste exercício, caso não haja erros a nota da regressão será calculada através das métricas propostas, e caso haja erros apenas a regressão por n-gramas será calculada, para indicar ao aluno se está seguindo o caminho correto. Após passar por estas etapas, o sistema retorna ao PHP as notas finais do aluno, exibindo na tela. A Figura 8 representa o fluxo geral do funcionamento do sistema.

Figura 8 - Exemplo de fluxograma do sistema de feedback automático



Fonte: (Moreira & Favero, 2009)

Este artigo não leva em consideração o sorteio de vários exercícios para compor uma lista ou prova e também não propõe a identificação de situações de plágio. Outra funcionalidade ausente é um ambiente Web próprio que permita ao professor acompanhar o andamento da turma.

3.3 LABSQL

O sistema LabSQL (Lino, Silva, Harb, Favero, & Brito, 2007) é um ambiente interativo para auxiliar o estudante no aprendizado da linguagem SQL e também é utilizado como uma ferramenta de apoio ao professor na realização automática de avaliações em atividades laboratoriais.

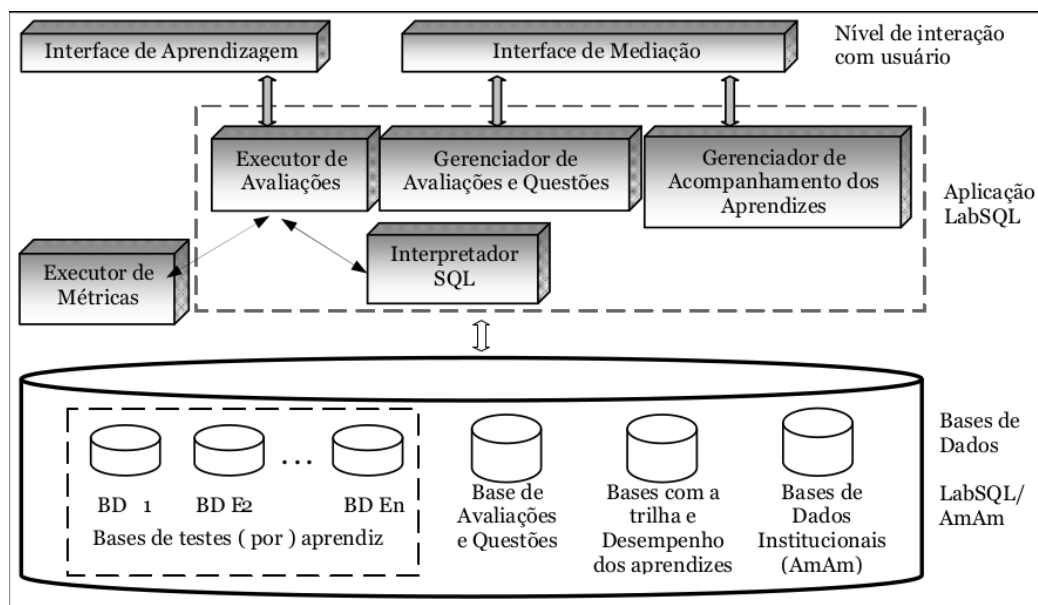
Os aspectos explorados pela ferramenta são: interatividade nos tópicos de programação, onde os exemplos que aparecem na tela de ensino podem ser executados pelo aluno; flexibilidade no sequenciamento de conteúdo, onde o estudante não precisa seguir linearmente o conteúdo ou execução dos exercícios; avaliação automática com um feedback

imediatamente; flexibilidade no monitoramento do aprendiz - o professor tem a possibilidade de acompanhar o desempenho de seus alunos.

A Figura 9 demonstra uma visão geral da arquitetura do LabSQL, onde o professor define as questões e avaliações; as questões servem para treinamento ou para avaliação. A seleção de questões é realizada previamente pelo professor ou é sorteada automaticamente pelo sistema. O acompanhamento do aluno pelo professor é realizado com base na trilha percorrida e no desempenho alcançado; como feedback, o aprendiz recebe a avaliação automática das consultas, sugerindo melhorias quando apropriado e comentários do professor acerca das suas soluções. As informações gerais do curso e de seus participantes estão na base de dados institucionais do AVA integrado ao LabSQL.

A estrutura do conteúdo do LabSQL está organizada em 5 módulos distintos, tendo um avanço gradativo de um para o outro, e em cada um destes há um determinado número de sessões, as quais o usuário pode escolher, independente de linearidade.

Figura 9 - Estrutura da Arquitetura do LabSQL



Fonte: (Lino, Silva, Harb, Favero, & Brito, 2007)

O professor é responsável por gerenciar questões (podendo ser: objetivas de múltipla escolha, objetivas (V/F), programação SQL e questões discursivas), conteúdo, avaliação e acompanhar o desenvolvimento dos alunos através de relatórios como: frequência de utilização do ambiente; avaliação detalhada mostrando notas de cada exercício por estudante; desempenho individual dos aprendizes.

Os usuários são direcionados a realizar as atividades a serem desenvolvidas, assim o aprendiz ao efetuar o login no sistema possui dois modos: avaliação, em que o sistema

redireciona o aprendiz para a área de avaliações, onde são listadas todas as avaliações disponíveis; e estudo onde o aprendiz visualiza a área de estudo, que permite a leitura acerca do conteúdo disponível.

Ao realizar um exercício SQL o aluno submete uma consulta que é avaliada pelo interpretador do SGBD retornando o resultado desta, utilizando uma base de testes que é gerada e mantida para cada estudante. O interpretador retorna o resultado da consulta que pode ser correto ou incorreto em relação à sintaxe e o resultado da consulta. Caso o resultado da consulta do aprendiz seja correto, então o comando SQL passa a ser avaliado em termos de complexidade, por um módulo avaliador de métricas que mede a distância da consulta do aprendiz em relação à do professor.

A avaliação do sistema foi composta de dez questões para a categoria usabilidade, seis para funcionalidade, duas para confiabilidade, duas para a eficiência do sistema, treze para a satisfação, um para o nível de satisfação subjetiva. Elas permitiram identificar o grau de satisfação do usuário num intervalo de 0 a 5. Esta análise foi realizada com 30 usuários, estudantes do curso de Sistemas de Informação da UFPA.

Assim como o ambiente WebSQL, o LabSQL se propõe a ser um sistema focado em ensino de consultas SQL. O sistema não leva em conta o uso de responsividade para adaptar o ambiente Web a dispositivos móveis.

3.4 WEFILAB: A WEB-BASED WIFI LABORATORY PLATFORM FOR WIRELESS NETWORKING EDUCATION

Neste artigo, (Cui, Tso, Yao, & Jia, 2012), uma plataforma de laboratório on-line prática, o Laboratório Wi-Fi baseado na Web (WeFiLab), é apresentada. Este sistema se concentra principalmente em fornecer aos alunos uma experiência prática de fazer experimentos em dispositivos reais através de um ambiente Web.

No ambiente, a estrutura de operações de dois níveis é utilizada, para aumentar a escala de dispositivos sem fio e permitir que sejam estendidos para operações mais complicadas. Os esquemas de programação do WeFiLab permitem que mais alunos compartilhem e usem dispositivos sem fio. A proposta tem o intuito de abranger alunos iniciantes e avançados.

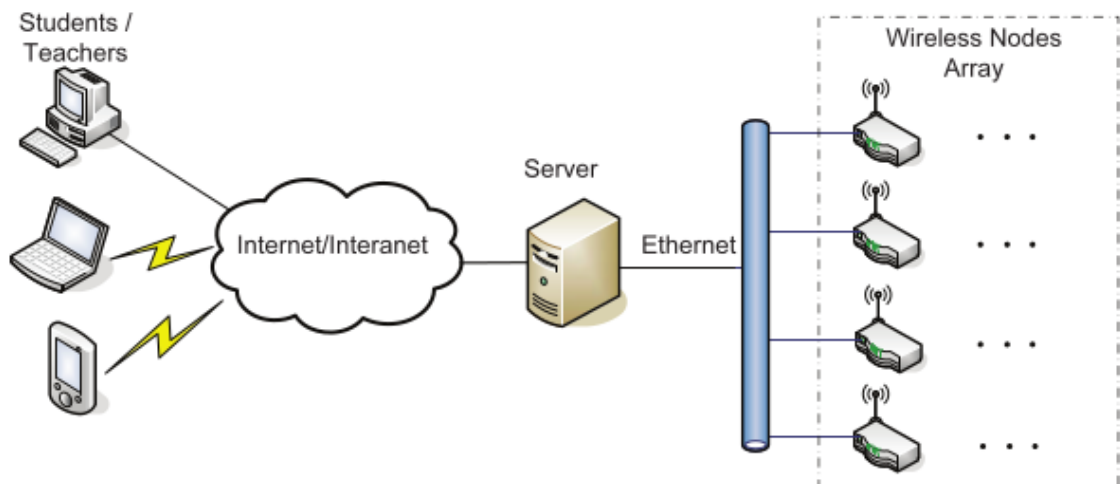
Estas atividades de aprendizagem e experiências são realizadas em dispositivos sem fio reais. Os alunos podem obter acesso instantâneo ao WeFiLab para treinamento prático, de modo a consolidar o conhecimento que já obtiveram em aulas expositivas. Professores / tutores podem projetar experimentos, avaliar o desempenho e revisar os registros dos alunos.

Todas essas informações tem o intuito de ajudar os professores a entender o desempenho dos alunos e coordenar suas atividades de ensino.

Uma peculiaridade do sistema é que seu design esconde os detalhes das configurações complexas em dispositivos sem fio, através da estrutura de operação de dois níveis, para que os alunos possam se concentrar em suas experiências, que são conduzidas completamente na página. WeFiLab também permite que o administrador adicione novos dispositivos sem fio sempre que necessário para suportar mais alunos.

A ilustração da arquitetura WeFiLab é mostrada na Figura 10. Todo o sistema WeFiLab consiste em um servidor, matriz de nós sem fio e terminais de cliente de borda.

Figura 10 - Arquitetura do WeFiLab



(a) System Architecture

Fonte: (Cui, Tso, Yao, & Jia, 2012)

Todos alunos, professores e tutores podem ser usuários do WeFiLab. Os navegadores da Web que executam seus terminais conectando-os ao WeFiLab são os clientes do sistema.

O servidor desempenha dois papéis importantes na plataforma WeFiLab. Um deles é atuar como um servidor web para fornecer uma GUI amigável ao usuário baseada em Ajax permitindo a interatividade entre estudantes e professores. Outro papel do servidor é atuar como um gateway para retransmitir mensagens de controle e resposta entre clientes e dispositivos sem fio.

Além de aproveitar a diversidade do perfil dos alunos, WeFiLab também usa a restrição de bloqueio em dispositivos sem fio e o esquema de cronograma para garantir a capacidade dos alunos.

A propriedade baseada em Web do WeFiLab permite que os alunos não precisem fazer as experiências durante o tempo do curso. Além disso, os alunos também não são obrigados a fazer os experimentos no campus.

Como base de testes, foram utilizados 16 roteadores sem fio para formar a matriz de nós. Vários dispositivos sem fio podem ser usados, pois o servidor esconde a diferença dos usuários através das operações de dois níveis. O sistema foi avaliado através de dois estudos de caso sendo estes: no curso CS4284-Mobile Computing e o curso CS4289-Pervasive Computing durante dois anos, que incluiu um total de 315 alunos de ciências da computação, na City University de Hong Kong.

No primeiro caso, 146 estudantes de informática de graduação estão envolvidos na experiência. Depois de aperfeiçoar o design e implementação do WeFiLab, o sistema foi utilizado novamente para a experiência do curso no semestre CS4284 2011B e CS4289 2011A, que envolveu 116 e 53 estudantes de graduação, respectivamente.

Os resultados dos estudos de caso mostraram que o WeFiLab melhorou a compreensão dos alunos sobre o WiFi, e também oferece uma experiência prática adequada sobre o design da rede WiFi, sendo assim uma boa ferramenta para o ensino acerca do assunto.

Este sistema, apesar de oferecer uma experiência gráfica útil didaticamente aos alunos, não leva em consideração a característica de avaliação com feedback automático através do sistema, não há funcionalidade para que o professor possa cadastrar provas, e também tem um foco específico na programação de redes WiFi para atividades laboratoriais, sem levar em consideração um conteúdo expositivo acerca do assunto.

3.5 NETAULA

Este artigo (kampff & Júnior, 2010) tem como objetivo apresentar a construção de um sistema de geração automática de avaliações integrado ao módulo de avaliação do AVA NetAula. O sistema permite criar provas, exercícios ou testes para os estudantes, de forma simples e ágil, utilizando questões previamente cadastradas em um banco de dados e catalogadas quanto à dificuldade, tipo e módulo de aula, que serão selecionadas conforme os critérios estabelecidos no momento de criação. As avaliações são realizadas pelos alunos por meio de um ambiente Web.

Este AVA é uma plataforma de Ensino a Distância que contempla funcionalidades como Fórum, Bate Papo, Avaliação, Correio Eletrônico e Mural de Recados.

O módulo de avaliação do AVA NetAula permite que o professor crie provas, exercícios e simulados para suas turmas. As questões das avaliações são criadas no momento

da geração da mesma (não há banco de questões) e podem ser dos tipos: Dissertativas, Objetivas, Múltipla Escolha e Associativas. Cada questão deve possuir um peso, medida utilizada para composição da nota do aluno. Além de gerar avaliações, o módulo fornece também opções de correção de avaliações, gerar notas e gabarito.

Para a geração de avaliações automáticas, é necessária que existam uma quantidade mínima de questões cadastradas no repositório. Essa quantidade mínima é o resultado de todas as combinações possíveis de questões por dificuldade e tipo, para cada questão a ser gerada de um determinado módulo.

Para realizar o processo de validação do sistema foi criado um banco de questões para as disciplinas de Cultura Religiosa e Comunicação e Expressão.

Esta proposta não leva em consideração o cadastro de questões por parte dos professores. Outra característica ausente é o sorteio de questões para compor as provas, além disso é um sistema focado no ensino fundamental e não são em matérias de programação.

3.6 AMBIENTE INTEGRADO À PLATAFORMA MOODLE PARA APOIO AO DESENVOLVIMENTO DAS HABILIDADES INICIAIS DE PROGRAMAÇÃO

Este trabalho (Mota, Brito, Favero, & Moreira, 2009) apresenta uma ferramenta que tem por objetivo ser um ambiente de exercício, simulação e avaliação para apoiar o processo de ensino-aprendizagem em cursos de algoritmos e programação, possui um visualizador de programas e um avaliador automático.

No projeto, um ambiente de aprendizagem de programação é integrado à plataforma Moodle, para isso foi criado um ambiente de aprendizagem que utiliza um visualizador de programas, denominado JavaTool, em exemplos, tarefas e questionários de programação. Além disso, quando o estudante submete suas soluções ao ambiente, é realizada uma avaliação, que pode ser automática, reduzindo a sobrecarga do professor no acompanhamento das soluções dos estudantes e agilizando o *feedback* necessário para o estudante reavaliar a sua solução, A Figura 11 apresenta um exemplo de resultados obtidos a partir do sistema.

Figura 11 - Resultados obtidos a partir do sistema JavaTool

The screenshot displays the JavaTool interface within a Moodle course. At the top, it shows the course name 'Curso de Programação' and the user's login 'Você acessou como teste teste2 (Sair)'. The navigation path is 'MMM > CCTurma2 > Questionários de Programação > Questionário de Programação'. The main title is 'Questionário de Programação' with the instruction 'Dado um retângulo, determine o perímetro e a área.' Below this is a toolbar with a play button, a stop button, a refresh button, and a 'Salvar Alterações' button. The interface is divided into several sections:

- Left Panel (B):** A code editor window titled 'Resposta.java' containing the following Java code:


```
1 int a = readInt();
2 int b = readInt();
3 int p = 2 * (a+b);
4 int area = a*b;
5 print("perímetro = ");
6 println(p);
7 print("área = ");
8 println(area);
```
- Right Panel (E):** An animation window titled 'Animação' with a 'Historico' tab. It displays four variable declarations with their values:

DEC	int	a	5
DEC	int	b	6
DEC	int	p	22
DEC	int	area	30
- Bottom Panel (F):** A console window titled 'Console' with a 'Histórico Textual' tab. It shows the output of the program:


```
perímetro = 22
área = 30
```

 The interface also includes a 'CONSTANTES' section with a box icon and a 'Salvar Alterações' button.

Fonte: (Mota, Brito, Favero, & Moreira, 2009)

A integração à plataforma Moodle permite que o ambiente de programação seja acessado e utilizado por meio desta. O Moodle possibilita a inserção de recursos e atividades, onde os recursos disponibilizam conteúdo e atividades promovem interação com o usuário. Foi desenvolvido um recurso denominado tutoria de programação e dois novos tipos de atividades (tarefa e questionário de programação).

Nas tutorias de programação, o professor armazena o código exemplo e o estudante visualiza a animação do código exemplo. A tarefa de programação é incluída pelo professor como um problema que deve ser resolvido pelo estudante. O questionário de programação consiste de um conjunto de questões-problemas de programação, selecionadas pelo professor. O estudante pode consultar as questões e o histórico de suas tentativas. A nota final é uma média aritmética das maiores notas entre as tentativas de cada questão. O sistema de avaliação permite que o professor avalie as respostas dos estudantes manualmente ou possibilita um *feedback* imediato por meio de um avaliador automático.

A proposta possui diversas funcionalidades como resolução de exercícios (atividades laboratoriais), possui conteúdo acerca da lógica de algoritmos e da linguagem em si, área específica para que o professor possa acompanhar o progresso da turma e também gerar provas e listas de exercícios, bem como um ambiente Web em que foi aplicado na plataforma Moodle, porém é centrada na linguagem de programação Java.

3.7 TSTVIEW

O TSTView (Gaudencio, et al., 2013) é um ambiente Web que permite visualizar o desempenho dos estudantes em um dado período de tempo e/ou em qualquer subconjunto dos exercícios. A observação dos resultados dos testes realizados sobre todas as soluções enviadas para cada exercício permite, por exemplo, que o professor monitore em tempo real o desenvolvimento dos estudantes frente aos exercícios planejados.

Ele se propõe a resumir os dados relativos às submissões das resoluções dos exercícios de programação realizados pelos estudantes. Cada resolução de exercício é um programa Python submetido a um testador automático que aplica os testes criados pelos professores para as questões propostas. Os dados de controle e os resultados dos testes aplicados às submissões compõem a base de dados para o TSTView. A ferramenta agrupa tais informações e as apresenta em diferentes perspectivas para professores e alunos.

O sistema possui dois módulos principais, o módulo do professor e o módulo do aluno. O primeiro apresenta os dados recentes dos alunos que estão realizando alguma atividade no momento da consulta, dando uma visão mais geral da turma, a aplicação disponibiliza filtros que permitem o foco da avaliação dos dados sobre um ou mais alunos, sobre turmas específicas, sobre determinados horários e datas, e sobre uma ou mais questões. Já o segundo, apresenta as informações em forma de um relatório histórico individual para cada aluno. Cada módulo requer acesso controlado mediante login que é efetuado através da plataforma Google App Engine, onde o sistema executa.

Esta ferramenta foi utilizada na disciplina Programação 1, do segundo semestre do curso de Bacharelado em Ciência da Computação da Universidade Federal de Campina Grande. No total, 105 alunos, fizeram parte desta avaliação divididos em 5 turmas práticas. A lista de exercícios foi composta de 258 questões. Destas 258 questões, 68 representaram mini-testes presenciais aplicados para contabilização de nota para a disciplina (média de 13,6 questões por turma). Durante os mini-testes, os alunos utilizaram um ambiente protegido que permitia o acesso apenas ao enunciado, ao serviço de submissão de respostas e aos resultados de teste.

Este sistema tem um foco apenas em atividades laboratoriais, através da correção de exercícios e avaliação automática, ou seja, não leva em consideração a exposição de conteúdo acerca da linguagem Python.

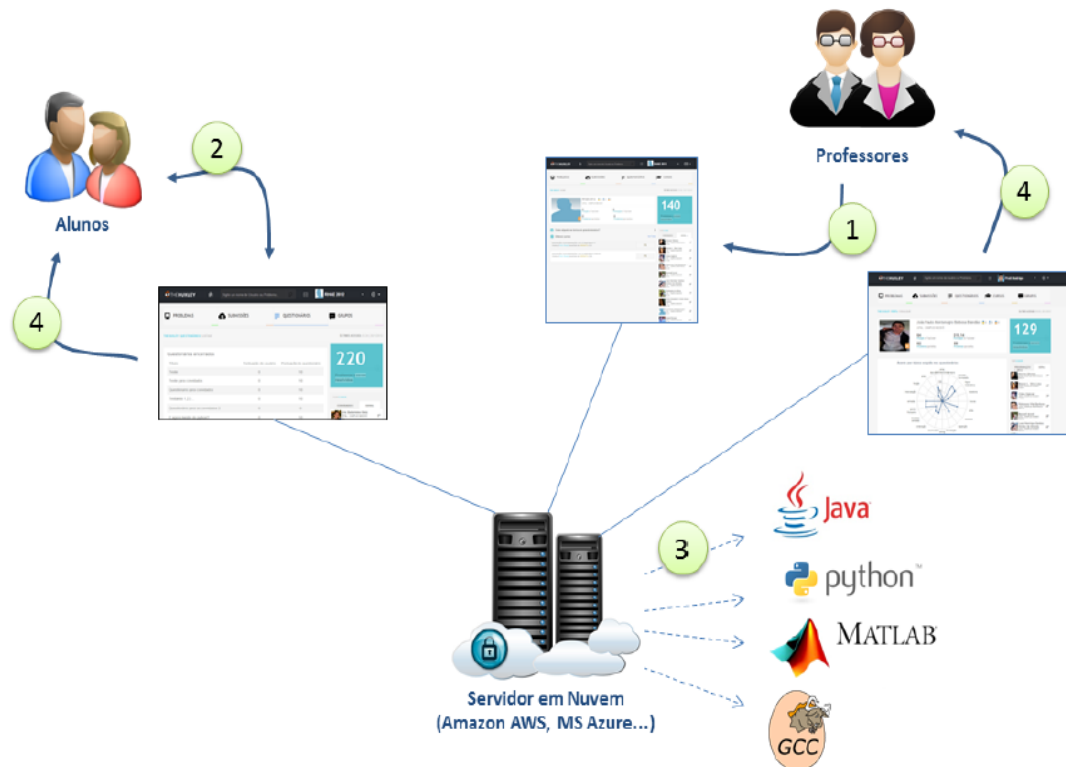
3.8 THE HUXLEY

The Huxley é uma ferramenta para avaliação do aprendizado de alunos em disciplinas de programação de computadores (Paes, Malaquias, Guimarães, & Almeida, 2013). É um ambiente web que permite aos alunos submeterem código em diversas linguagens de programação como respostas a exercícios de uma base de centenas de problemas. Para cada submissão, o aluno recebe feedback da correção automática pelo sistema através de análise sintática do código e testes de aceitação.

O professor tem uma visão mais analítica do desempenho de seus alunos, incluindo a quantidade de problemas resolvidos, porcentagem de acertos/erros, tipos de problemas com mais erros, detecção de plágio e erros específicos de cada aluno. Do ponto de vista prático, o The Huxley possui uma interface web, com acesso via login e senha, disponível em www.thehuxley.com. Professores e alunos possuem diferentes visões, ambas baseadas no conceito de *dashboard*, que permite uma visualização global do status de exercícios, avaliações e conteúdo já na página principal do sistema.

No ambiente The Huxley, o professor cria questionários e avaliações a partir de uma base de centenas de exercícios existentes, podendo também criar seus próprios exercícios, estes incluem descrição e testes de aceitação do resultado. Os alunos acessam os exercícios definidos pelos professores, desenvolvem os programas e os submetem para análise. Para cada programa em linguagem específica, The Huxley compila e testa, armazenando o resultado para consulta posterior do aluno e do professor, este resultado é visualizado em forma de planilha e gráfico pelos professores, que podem atribuir nota diretamente com base na avaliação do sistema. Os alunos também recebem o resultado para que possam submeter novamente a solução caso haja algum problema de compilação ou nos testes de aceitação. A Figura 12 expõe uma visão geral acerca do funcionamento do sistema.

Figura 12 - Funcionamento do Sistema The Huxley



Fonte: (Paes, Malaquias, Guimarães, & Almeida, 2013)

Como estudo de caso, foram analisados dados de 332 estudantes, alunos do primeiro semestre do curso de Ciência da Computação. O objetivo principal desse estudo realizado é identificar como a ferramenta impactou o desempenho dos estudantes.

O propósito da ferramenta é de apenas gerar avaliações e listas de exercícios, e promover uma análise detalhada do desempenho dos alunos, porém não leva em consideração uma linguagem específica para ensino, além disso não possui uma área específica para exposição de conteúdo para o ensino de algoritmos.

3.9 CONCLUSÕES DO CAPÍTULO

Este capítulo abordou um estudo acerca dos trabalhos relacionados aos AVA. É possível notar que existem diversos ambientes que possuem foco em ensino de programação. Independente da linguagem ou assunto cada um possui suas particularidades, vantagens e desvantagens.

A Tabela 1 e a Tabela 2 contém um apanhado geral dos objetivos de cada trabalho relacionado apresentado ao longo deste capítulo, a fim de ilustrar a diferença do foco da proposta desenvolvida neste trabalho. É perceptível que a maioria deles não possuem o foco no ensino de *Python* (a exceção da ferramenta *TSTView*).

A maioria dos sistemas estudados não focam no conteúdo sobre ensino de programação; mas focam na avaliação de exercícios. Este trabalho inclui também o conteúdo a ser exibido ao aluno sobre os assuntos de programação.

Tabela 1 - Comparação entre os trabalhos relacionados e esta proposta

	LabSQL	WebSQL	WeFiLab	NetAula	LabPy
Linguagem	SQL	SQL	Redes	Outros	Python
Avaliação Automática	X	X		X	X
Provas	X	X		X	X
Lista de Exercícios	X	X		X	X
Conteúdo para Estudo	X				X
Ambiente Web Próprio	X	X	X	X	X
Área do Professor	X	X	X	X	X
Área do Aluno	X	X	X	X	X
Ensino de Programação	X	X	X	X	X
Ano	2007	2008	2011	2010	2018
Referência	(LINO, <i>et al.</i> , 2007)	(RAIOL, 2014)	(CUI, TSO, <i>et al.</i> , 2012)	(KAMPFF, 2010)	

Fonte: Própria

Tabela 2 - Comparação entre os trabalhos relacionados e esta proposta

	JAVATOOL	TSTView	The Huxley	Feedback	LabPy
Linguagem	Java	Python	Várias	Java	Python
Avaliação Automática	X	X		X	X
Provas	X		X		X
Lista de Exercícios	X		X		X
Conteúdo para Estudo	X				X
Ambiente Web Próprio	X	X	X		X
Área do Professor	X	X	X		X
Área do Aluno	X	X	X		X
Ensino de Programação	X	X	X	X	X
Ano	2009	2013	2013	2009	2018
Referência	(MOTA, 2009)	(GAUDENCIO, <i>et al.</i> , 2013)	(PAES, <i>et al.</i> , 2013)	(MOREIRA, 2009)	

Fonte: Própria

4 LABPY

Este capítulo tem por objetivo apresentar a proposta do LabPy, elucidando a partir dos dados e informações levantadas com os trabalhos relacionados a sua utilização, assim como as funcionalidades propostas para o sistema, na visão do professor ou do estudante.

4.1 PROPOSTA DO LABPY

O LabPy é um ambiente interativo que tem o intuito de auxiliar os estudantes no aprendizado de lógica de programação focando na linguagem Python. Além disso, ele também pode ser utilizado como uma ferramenta de auxílio ao professor para realizar atividades práticas de laboratório com avaliação automática. O sistema LabPy teve sua concepção baseado na premissa de resolver problemas durante as aulas práticas de programação. Em sua versão mais recente, os seguintes aspectos-chave são explorados:

- conteúdo didático para consulta: o ambiente fornece uma série de conteúdos acerca da linguagem Python e sobre programação, para que assim o aluno possa tirar suas dúvidas a partir do próprio material presente no site;
- tópicos de programação interativos: em cada tópico presente no sistema, existe uma diversidade de exemplos que podem ser executados para que o estudante tenha uma visualização e possa realizar uma análise acerca do funcionamento do programa;
- conteúdo não-linear: o sistema possui um comportamento flexível, permitindo que o aluno tenha a possibilidade de seguir o conteúdo de forma não-linear, ou seja, sem ter a obrigatoriedade de passar por várias sessões anteriores para chegar a que pretende;
- avaliação automática: proporciona aos alunos a possibilidade de uma avaliação com feedback imediato, para que possam observar seu desempenho e tentar melhorar gradativamente;
- ambiente adequado para acompanhamento dos alunos: o professor tem a possibilidade de acompanhar o desempenho de cada um dos seus alunos ou da turma inteira.

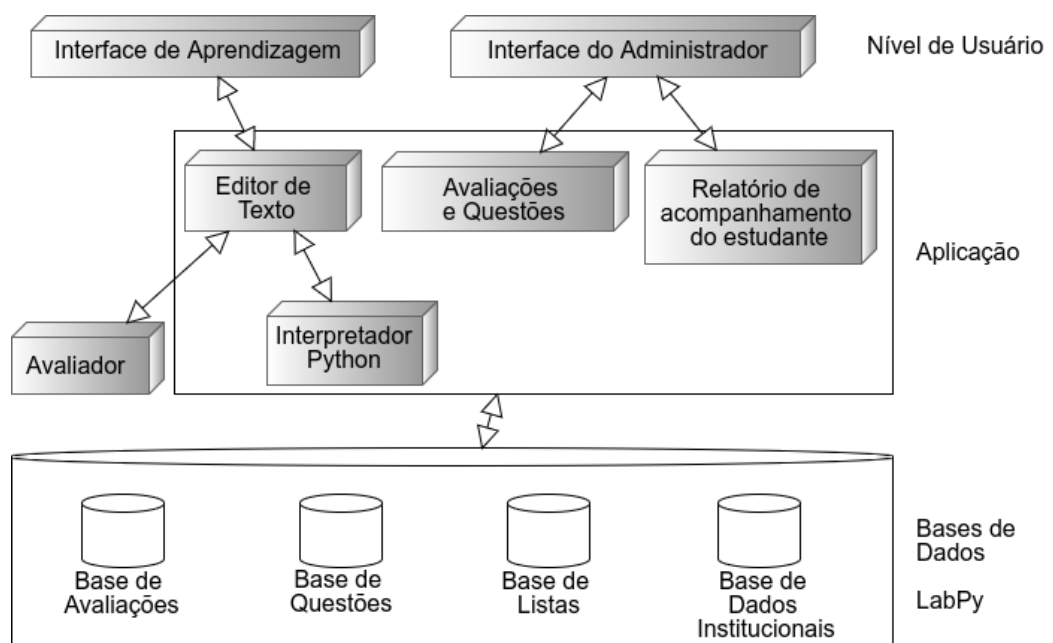
Uma visão geral da arquitetura do sistema LabPy está representada na Figura 13, a qual o professor a partir de uma **interface**, define as **avaliações e exercícios**, assim como as suas respostas. As questões podem ser apenas para o treino dos alunos ou utilizadas como parte de uma avaliação.

Em uma das possibilidades de prova ou lista de exercícios o sistema realiza um sorteio automaticamente para cada um dos estudantes. O aluno através da **interface de aprendizagem** pode interagir com o sistema com a finalidade de solucionar os exercícios disponibilizados. O **editor de texto** envia uma requisição ao **Interpretador Python**, que executa o script e retorna a saída para o usuário na tela. Caso a execução do programa feito pelo aluno retorne uma saída semelhante ao retorno da solução do professor, então o script passa a ser avaliado, através do **algoritmo n-gramas**, para verificar se a resposta do aluno possui uma quantidade de palavras menor ou igual a do professor. Caso contrário, o aluno é convidado a aprimorar a sua solução, escrevendo mais uma versão da solução com um menor número de palavras.

O algoritmo n-gramas é um pedaço de n-caracteres, extraído de uma cadeia de caracteres maior (palavra ou frase). Normalmente, pode-se utilizar n assumindo o valor 1 (unigrama), 2 (bigrama) ou 3 (trigrama). Ao quebrar duas cadeias de caracteres em n-gramas, é possível medir quantos n-gramas são compartilhados entre eles, assim pode-se determinar o quanto dois textos são similares (Moreira & Favero, 2009).

Sempre que o aluno submete uma resposta ao sistema imediatamente ele recebe o *feedback* através do avaliador automático. Através do **relatório de acompanhamento do estudante**, o professor pode obter a situação do aluno em termos de notas, número de tentativas, entre outros. Por fim, a **base de dados** contém todas as informações referentes ao curso e sobre seus participantes.

Figura 13 - Representação geral do LabPy



Fonte: Própria

4.2 ESTRUTURA DO CONTEÚDO

Para uma melhor organização do conteúdo exposto através do sistema, o LabPy foi segmentado em módulos, baseados em assuntos específicos. Os seguintes tópicos podem ser observados na versão mais recente do sistema:

- **Introdução:** apresenta uma introdução a linguagem Python, dando um apanhado geral acerca da instalação, modo de uso do IDLE e como exemplos de alguns comandos iniciais;
- **Variáveis, expressões e comandos:** apresenta uma explicação sobre variáveis, expressões aritméticas, e comandos tais como: uso do operador de atribuição =; uso das funções `input()`, `print()` e `type()`;
- **Módulos:** apresenta um apanhado geral acerca do uso dos módulos presentes em *Python*, exemplos de uso destes tais como o *Math()*, que contém diversas funções matemáticas comumente encontradas em calculadoras e algumas constantes, e o *random()*, utilizado para gerar números aleatórios;
- **Funções (parte 1):** apresenta de forma didática uma elucidação acerca de funções em *Python*, demonstrando conceitos como retorno, escopo de função, entre outros;
- **Funções (parte 2):** da continuidade ao módulo **Funções (parte 1)**, expõe acerca do desenvolvimento de funções, composição e função *booleana*;
- **Decisões e seleção:** neste módulo há um foco decisões e seleção, exemplificando de forma didática acerca da estrutura condicional *if*, *if-else* e *if-elif-else*;
- **Mais sobre iteração:** apresenta uma explicação acerca dos principais comandos de repetição tais como *for* e *while*, apresentando diversos exemplos para fixação de conteúdo;
- **Strings:** apresenta conceitos sobre manipulação de Strings, citando as melhores práticas para se utilizar este padrão;
- **Listas:** explica sobre Listas e suas diversas funcionalidades, como concatenação e repetição, obter o comprimento de uma lista, fatias de listas, apelidos e clones de listas;
- **Arquivos:** apresenta os principais conceitos sobre manipulação de arquivos, explicando sobre como encontrar um arquivo em disco, como abrir um arquivo, entre outros conceitos;
- **Dicionários:** apresenta os conceitos gerais sobre dicionários em *Python*, como por exemplo operações com dicionários e métodos de dicionários;

- **Recursão:** aborda uma visão geral sobre recursão, ensina a formular programas de forma recursiva, entender e aplicar as leis da recursão, entender a recursão como uma forma de iteração, implementar a formulação recursiva de um problema e entender como uma recursão é implementada por um sistema computacional;

A Figura 14 ilustra a forma como os módulos estão dispostos no ambiente. Em cada um destes há um determinado número de tópicos, representados em formato de link, em que o aluno pode clicar para ter acesso ao conteúdo.

Figura 14 - Disposição geral dos módulos no sistema

Índice

Introducao
Variaveis, Expressoes e Comandos
Modulos
Funcoes Parte 1
Funcoes Parte 2
Decisoes e Selecao
Mais Sobre Iteracao
Strings
Listas
Arquivos
Dicionarios
Recursao

Fonte: Própria

Cada módulo presente no sistema possui diversas questões para fixação de conceitos acerca do que foi estudado ao longo do conteúdo, além disso, ao final de cada assunto um link é disponibilizado para que o aluno tenha acesso aos exercícios práticos referentes ao que foi aprendido, e assim possa ser avaliado corretamente.

4.2.1 Exercícios

O LabPy possui diversos exercícios, estes são voltados apenas para a programação prática. Tanto as perguntas quanto as respostas são previamente cadastradas pelo professor, que o faz através da área do administrador presente no sistema. As respostas submetidas pelos estudantes são avaliadas automaticamente pelo sistema. A Figura 15 demonstra um exemplo da área de cadastro de questões.

Figura 15 - Área de cadastro de questões

Dados da Questão

Módulo

Variaveis, Expressoes e Comandos ▾

Dificuldade

Facil ▾

Enunciado

Faça um programa para somar dois números.

Resposta

```
a = 1
b = 2
soma = a + b
print "A soma vale: ", soma
```

Fonte: Própria

Nesta área o professor deve escolher um dos módulos para inserir o exercício, sua dificuldade, que pode ser fácil, médio e difícil, o enunciado da questão, a resposta esperada para a questão, sua saída e a nota, então deverá clicar no botão “ADD”, para submeter o exercício. A Figura 16 apresenta alguns exercícios cadastrados no sistema.

Figura 16 - Exercícios cadastrados no sistema

Exercícios sobre Variáveis, Expressões e Comandos

Questões

Questao 10
Questao 11
Questao 12
Questao 13
Questao 14
Questao 15
Questao 16
Questao 17
Questao 18

Fonte: Própria

Estando cadastrada, a questão pode ser acessada com apenas um clique. O aluno tem acesso a questão via um editor de texto com a função *syntax highlighting*, que permite que os principais comandos em *Python* fiquem em evidência. A Figura 17 apresenta um exemplo de acesso a um exercício e sua execução.

Figura 17 - Exemplo de execução de um exercício

Questão 56 - Faça um programa que receba dois números inteiros e gere os números inteiros que estão no intervalo compreendido por eles.

```
1 inicial = int(raw_input('Informe o valor inicial: '))
2 final = inicial
3 while (final <= inicial):
4     final = int(raw_input('Informe o valor final: '))
5     if (final <= inicial):
6         print 'O valor final deve ser maior que o valor inicial!'
7
8 for i in range(inicial, final + 1):
9     print i
```

Executar

Resultado da execução

```
1
2
3
4
5
```

Fonte: Própria

A execução da questão é feita através do editor de texto, onde o aluno entra com a sua resposta e sua saída é gerada, como no exemplo da Figura 17, os valores de entrada do programa foram 1 e 5, e a saída gerada foi a sequência de inteiros 1, 2, 3, 4 e 5; a Figura 18 apresenta o feedback fornecido ao aluno.

Figura 18 - Feedback fornecido ao aluno após a execução do programa

```
Resultado final
Saída correta!
Precisão da Solução:
100%
```

Fonte: Própria

4.3 ÁREA DO PROFESSOR

No seu perfil o professor é o responsável por fazer a gerência das questões, avaliações e acompanhar a evolução de sua turma através de relatórios acerca do desempenho e situação de cada aluno.

O professor também tem a permissão de cadastrar novos alunos no sistema, que pode ser através da inserção de uma turma completa, feita através de uma tabela *excel* (arquivo no formato .xml), ou de um usuário específico, sendo que nesse caso é necessária a inserção dos dados do aluno, como nome, matrícula, turma, entre outros.

O professor tem a possibilidade de visualizar e acessar seis opções, cada uma destas referente a uma funcionalidade específica, sendo estas:

- **Área do aluno:** encaminha o professor para a área principal do site, que contém os módulos presentes no sistema, em que os alunos possuem acesso também, Figura 19;

Figura 19 - Área do aluno

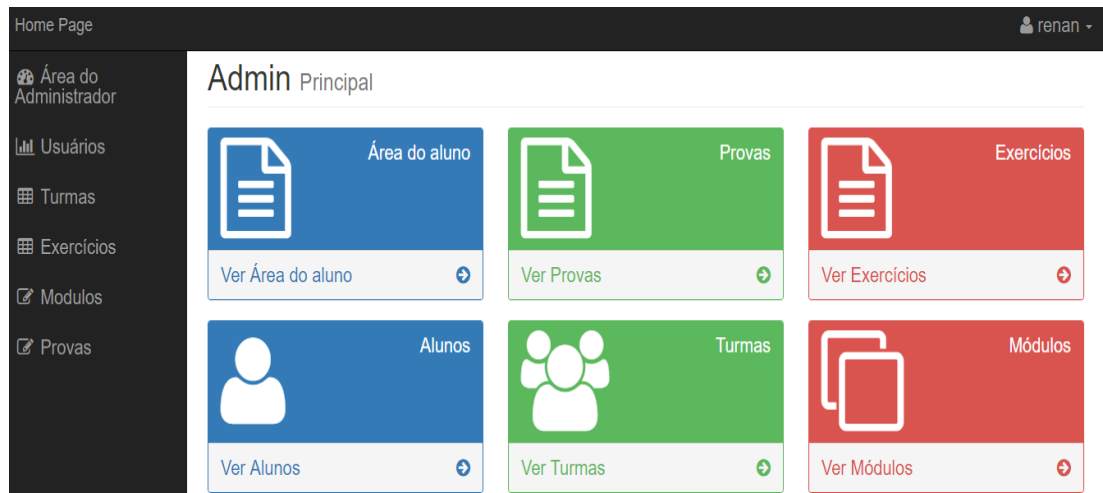
The screenshot shows a web interface for a Python course. At the top, there is a navigation bar with links for 'Home Page', 'Provas', and 'Admin', and a user profile for 'renan'. The main content area is titled 'Introdução a Computação com Python' and features a table of contents under the heading 'Índice'. To the right, there is a search bar and a sidebar menu with links to 'Conteúdo', 'Introdução à Computação com Python', 'Livro Interativo', 'Índice', 'Apêndices', and 'Índices e tabelas'.

Índice	
Introducao	
Variaveis, Expressoes e Comandos	
Modulos	
Funcoes Parte 1	
Funcoes Parte 2	
Decisoes e Selecao	
Mais Sobre Iteracao	
Strings	

Fonte: Própria

- **Área principal do administrador:** é uma área onde o professor tem o acesso global a outras seções presentes no sistema, possibilitando a visualização rápida de provas, turmas, exercícios e avaliações, Figura 20;

Figura 20 - Área principal do administrador



Fonte: Própria

- **Administrar Turmas:** esta área permite ao professor gerenciar suas turmas, podendo cadastrar, editar ou excluir alguma turma, Figura 21;

Figura 21 - Seção de administração de turmas

Turmas

Add Turma

ID	Nome da Turma	Quantidade de Alunos
1	Turma 1 Edit Delete	10 Ver Alunos
2	Turma 2 Edit Delete	10 Ver Alunos
3	turma 3 Edit Delete	10 Ver Alunos
4	turma 4 editada Edit Delete	10 Ver Alunos
7	turma 7 Edit Delete	10 Ver Alunos

Fonte: Própria

- **Administrar Usuários:** esta área permite ao professor gerenciar seus estudantes, podendo cadastrar, editar ou excluir algum estudante, Figura 22;

Figura 22 - Seção de administração de usuários

Usuários

Add Usuário		Gerar Notas							
ID	Nome de Usuário	Nome do Aluno	Matricula	Permissão					
1	renan	Renan	123456	professor	Ver Notas				
	Edit	Delete							
52	teste	ADIEL	02	aluno	Ver Notas				
	Edit	Delete							
53	teste	ALAN	09	aluno	Ver Notas				
	Edit	Delete							
54	teste	ANA	21	aluno	Ver Notas				
	Edit	Delete							
55	teste	AUGUSTO	03	aluno	Ver Notas				
	Edit	Delete							
56	teste	DANIEL	18	aluno	Ver Notas				
	Edit	Delete							

Fonte: Própria

- **Administrar Exercícios:** a partir desta funcionalidade, é permitido ao professor gerenciar as questões presentes no sistema cadastrando, editando ou excluindo. Além disso, permite gerenciar as respostas referentes a cada questão, Figura 23;

Figura 23 - Seção de administração de exercícios

Exercícios

Add Exercício							
ID	Pergunta	Resposta	Saídas	Entradas	Módulo	Nota	Dificuldade
1	Faça um Programa que mostre a mensagem 'alo mundo' na...	print 'Alo mundo'...	ola mundo... / alo mundo... / hello	alo mundo... / ola mundo... / hello	1	10	Facil
	Edit	Delete					
2	Faça um Programa que peça um número e então mostre...	x = int(raw_input("Informe um numero: ")) print "O nu...	... / ... / / ... / ...	1	10	Facil
	Edit	Delete					
3	Faça um Programa que peça dois números e imprima a<...	num1 = int(raw_input("Informe um numero: ")) num2 =	... / ... / / ... / ...	1	10	Facil

Fonte: Própria

- **Administrar Módulos:** proporciona ao professor a possibilidade de gerenciar os módulos presentes no sistema, permitindo a criação, edição e exclusão de um determinado módulo, Figura 24;

Figura 24 - Seção de administração de módulos

Modulos

Add Modulo			
ID	Nome	Editar	Deletar
1	Introducao	Edit	Delete
2	Variaveis, Expressoes e Comandos	Edit	Delete
3	Modulos	Edit	Delete
4	Funcoes Parte 1	Edit	Delete
5	Funcoes Parte 2	Edit	Delete
6	Decisoes e Selecao	Edit	Delete
7	Mais Sobre Iteracao	Edit	Delete
8	Strings	Edit	Delete
9	Listas	Edit	Delete

Fonte: Própria

- **Administrar Provas:** o professor pode gerenciar as avaliações, criando, editando ou excluindo uma avaliação por completo, ou também pode realizar estas operações para questões individualmente, Figura 25.

Figura 25 - Seção de administração de provas

Provas

Add Prova						
ID	Nome	Modulos	Dificuldade	Quantidade	Editar	Deletar
1	Prova 1	1,2	Facil	13	Edit	Delete
2	Prova 2	1,2	Facil,Medio	10	Edit	Delete
3	Prova 3	12	Facil,Medio	5	Edit	Delete
4	Prova 4	1		4	Edit	Delete
5	Prova 4	1		4	Edit	Delete
6	Prova 5	1	Facil,Medio,Dificil	3	Edit	Delete
ID	Nome	Modulos	Dificuldade	Quantidade	Editar	Deletar

Fonte: Própria

4.4 ÁREA DO ALUNO

É uma área presente no sistema em que o aluno ao efetuar seu login tem acesso ao conteúdo, listas de exercícios e provas cadastradas pelo professor. A Figura 26 contém uma ilustração da visão do aluno ao efetuar o login no sistema, sendo este redirecionado para a área de seleção de módulos.

Figura 26 - Área de login no sistema

Digite seu Nome de Usuário e sua Senha.

renan

...

Lembre-me

Login

Fonte: Própria

A Figura 27 apresenta a área de avaliações que é visualizada pelos estudantes ao selecionar a opção Provas. Quando uma avaliação é disponibilizada no sistema, o LabPy apresenta cada tarefa na tela através de um link, para que assim o aluno possa ter uma flexibilidade na escolha das atividades que acharem mais importantes, sem a necessidade de seguir uma ordem específica.

Figura 27 - Área de provas

Provas

Prova 1
Prova 2
Prova 3
Prova 4
Prova 4
Prova 5

Fonte: Própria

Na área de estudo, o aluno visualiza a página principal, que possibilita a leitura referente ao conteúdo disponibilizado e links que permitem o acesso a outras funcionalidades, sendo estes:

- **Provas:** é uma área destinada a hospedar todas as provas cadastradas pelo professor, ao clicar no link o aluno é redirecionado para área de avaliação, como representado na Figura 25;

- **Exercícios:** está disponível para o acesso do aluno aos exercícios ao final de cada sessão, a Figura 28 representa esta tela;

Figura 28 - Área de exercícios

Exercícios sobre Funcoes Parte 1

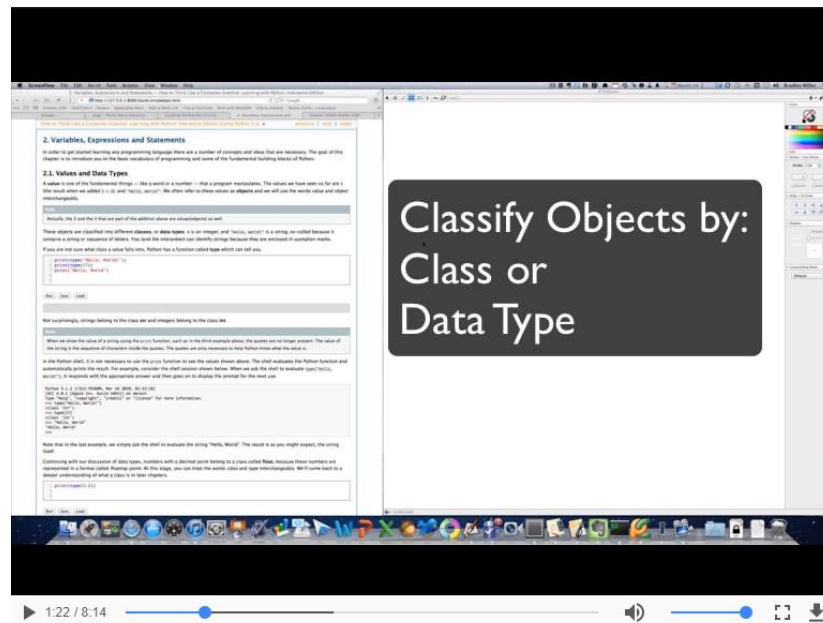
Questões

Questao 28
Questao 29
Questao 30
Questao 31
Questao 32
Questao 33
Questao 34
Questao 35
Questao 36

Fonte: Própria

- **Vídeos:** quando disponível, disponibiliza aos estudantes alguns links contendo vídeos externos sobre o assunto, para que assim haja uma forma extra de aprendizado mais interativa, e torna mais fácil o ensino principalmente para os alunos que não compreenderam a explicação do material ou não estavam presentes nas aulas onde a matéria foi explicada, a Figura 29 apresenta um exemplo de vídeo disponível no sistema;

Figura 29 - Vídeo disponível no sistema



Fonte: Própria

- **Referências Externas:** assim como vídeos e fontes de materiais externos, a sessão de conteúdo também conta com referências externas de artigos ou livros;

Além disso, ao longo de cada módulo existe uma série de exemplos que podem ser executados na própria sessão, sendo assim, quando o aluno clica em executar é possível compilar e visualizar a saída gerada pelo código em questão. A Figura 30 representa um exemplo de saída gerada pelo sistema.

Figura 30 - Saída gerada pela execução de um programa em um módulo de estudo

Algumas funções são definidas com mais do que um argumento. Por exemplo, o módulo matemático contém uma função chamada `pow` que usa dois argumentos, a base e o expoente.

```

1 import math
2 print(math.pow(2, 3))
3
4 print(math.pow(7, 4))
5
6

```

Run Save Load

```

8.0
2401.0

```

Note que já vimos que elevar uma base a um expoente pode ser feito com o operador `**`.

Fonte: Própria

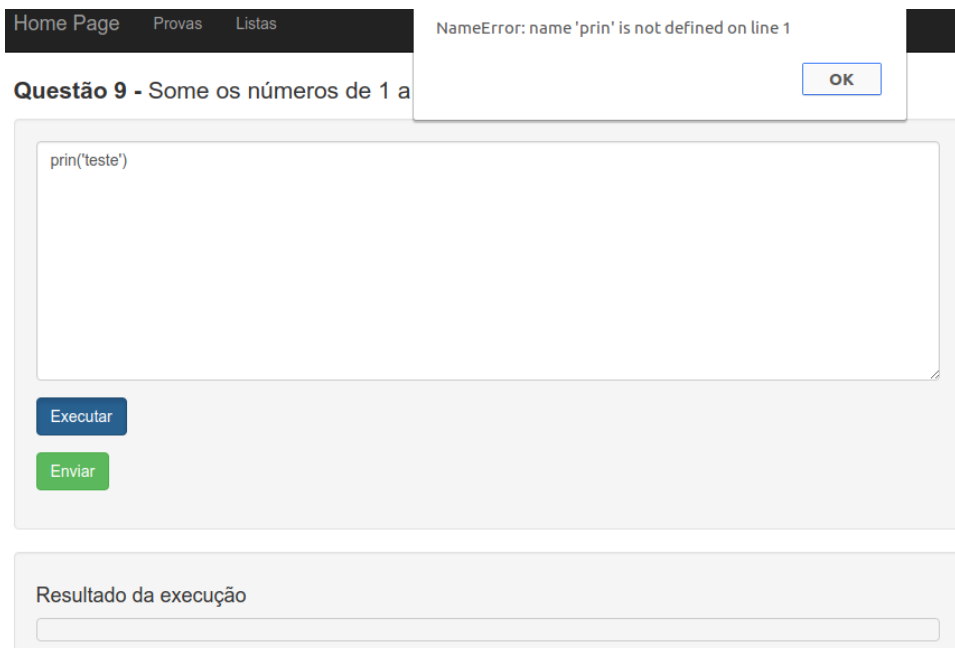
4.5 EXECUÇÃO DE PROGRAMAS EM PYTHON

Após responder um exercício em Python, o aluno tem a possibilidade de executar o programa escrito para que seja avaliado pelo interpretador Python, presente no browser; o resultado da saída é retornada ao estudante, podendo estar correta ou incorreta em relação a cadastrada pelo professor. Caso esta saída esteja correta é feita uma avaliação do código a partir de um avaliador que utiliza o algoritmo n-gramas.

Um programa em Python pode ser avaliado diante de diversas métricas, de acordo com o nível este *feedback* pode ser diferente:

1. **Sintaxe:** O resultado emitido pode ser correto ou incorreto, caso o resultado seja incorreto, uma mensagem de erro é emitida através do interpretador presente no próprio browser. A Figura 31 representa uma mensagem que indica o erro ao utilizar um comando específico da linguagem Python, `print(“”)`, de forma incorreta.

Figura 31 - Alerta de erro de sintaxe no LabPy



Fonte: Própria

2. **Resultado da Execução:** após ser compilado, o resultado do programa é avaliado pelo sistema LabPy, caso esteja certo o sistema compara a quantidade de palavras utilizadas na resposta do aluno com a resposta ideal cadastrada pelo professor na base de dados, caso dê diferente o aluno recebe uma mensagem informando que sua solução ainda pode ser aperfeiçoada; a Figura 32 apresenta um resultado gerado pelo sistema.

Figura 32 - Resultado gerado pelo sistema

Questão 19 - Faça um Programa que peça dois números e imprima o maior deles.

The image shows a code editor window with the following Python code:

```

1 num1 = int(raw_input('Informe um numero: '))
2 num2 = int(raw_input('Informe outro numero: '))
3
4 if (num1 > num2):
5     print num1, 'eh maior que', num2
6 elif (num1 < num2):
7     print num2, 'eh maior que', num1
8 else:
9     print 'Os numeros sao iguais'

```

Below the code is a green button labeled "Executar". To the right of the code editor is a box titled "Resultado final" containing the following text:

```

Resultado final
Saída correta!
Precisão      da
Solução:
106.06299212598%

```

Fonte: Própria

4.6 CONCLUSÕES DO CAPÍTULO

Este capítulo teve como objetivo apresentar o AVA LabPy proposto neste trabalho, bem como a visão geral de seu funcionamento. A exposição das características presentes no sistema assim como as telas de execução foi essencial para um melhor entendimento de suas peculiaridades.

Nossa proposta tem como principal objetivo ser uma ferramenta para ensino de programação utilizando a linguagem *Python*. Além disso, este trabalho se propõe a auxiliar o professor e seus alunos em sala de aula através de diversas características presentes no sistema:

1. Gerar listas de exercícios, onde o professor pode cadastrar exercícios de acordo com cada módulo, levando em consideração o nível de dificuldade;
2. Compor provas e lista de exercícios a partir da base de questões;
3. Avaliar de forma automática as respostas dos alunos, para as listas e provas;
4. Gerar notas para verificar o desempenho dos estudantes ao longo da disciplina;
5. Criar trechos de código executáveis ao longo do texto dos módulos, tornando possível uma aprendizagem mais dinâmica aos alunos.

Sendo assim, através das considerações feitas acerca deste trabalho, pode-se perceber que existem características que o distingue dos demais trabalhos apresentados no Capítulo 3.

5 ANÁLISE DA PROPOSTA

Uma vez conhecidas as características técnicas da ferramenta proposta, este capítulo apresenta uma proposta de avaliação para obtenção de resultados a partir da metodologia utilizada no trabalho.

5.1 AVALIAÇÃO DO LABPY

Lino (2007) avaliou o LabSQL baseando-se na abordagem de qualidade proposta por Kemczinski (2005). Ele estendeu a proposta de Kemczinski incorporando aspectos pedagógicos proporcionando assim uma visão mais detalhada das funcionalidades do sistema.

A abordagem de Werneck e Moraes (2003) também é levada em consideração no trabalho de Lino. Ele apresenta uma proposta de avaliação de qualidade de aplicações na Web utilizando normas de avaliação de qualidade de *software* para produtos e *softwares*, e tem como base as normas de qualidade ISO/IEC 9126-1 e um conjunto de características que estão presentes em Pressman (2016), Lima et al (2005) e Reis et al (2002).

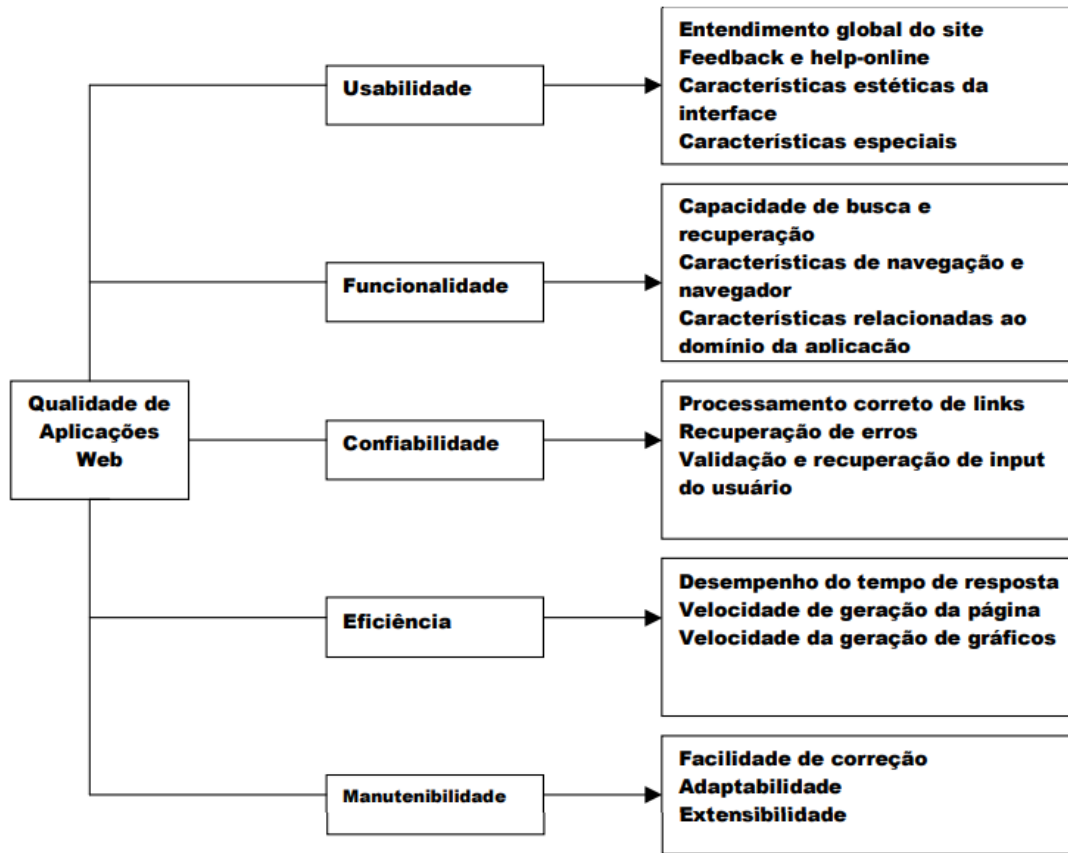
Nesta proposta, em cada item da avaliação há uma pontuação relacionada aos requisitos de qualidade (usabilidade, funcionalidade, confiabilidade, eficiência, manutenibilidade, portabilidade e segurança) em uma escala de 0 (sem solução) a 5 (excelente).

Este trabalho propõe uma abordagem para avaliar o sistema Labpy baseado em um formulário com 30 questões, levando-se em consideração aspectos de qualidade de *software* e aspectos pedagógicos presentes no sistema. Os aspectos referentes a linguagem *Python* também serão avaliados através de questões específicas.

5.2 PROPOSTA DE AVALIAÇÃO DO LABPY

Esta proposta de avaliação leva em consideração um formulário com 30 questões, organizadas em cinco categorias estabelecidas por Werneck e Moraes (2003), que definem um conjunto de características para realizar a avaliação de qualidade na Web e abrangem aspectos relacionados ao uso da tecnologia, assim como ao conteúdo disponibilizado. As categorias propostas podem ser observadas na Figura 33.

Figura 33 - Esquema de Qualidade Web



Fonte: (Werneck & Moraes, 2003)

A partir do esquema definido na Figura 32, a avaliação do sistema foi composta por: sete questões para a categoria usabilidade; quatro para a funcionalidade; dez para a satisfação pedagógica; duas para confiabilidade; duas para eficiência do sistema; e uma para o nível de satisfação subjetiva. Este questionário tem o objetivo de observar o grau de satisfação do usuário, em um intervalo de 0 a 5.

Além disso, uma categoria extra sobre os aspectos gerais da linguagem foi adicionada, para que fosse possível medir o grau de dificuldade dos alunos em relação ao aprendizado de Python. As subseções a seguir abordam as questões propostas em cada categoria apresentada anteriormente.

5.2.1 Usabilidade

Para Nielsen (1993) a usabilidade é definida em função de vários componentes e é relacionada aos seguintes atributos:

- Facilidade de aprendizagem: tempo que o usuário leva para assimilar o sistema e começa a utilizar com um grau de produtividade aceitável;

- Facilidade de memorizar: facilidade de lembrar o funcionamento de um sistema ao utilizá-lo depois de um determinado tempo;
- Satisfação: medida da satisfação do usuário ao utilizar o sistema.

A Tabela 3 demonstra as questões da categoria usabilidade, apresentando as alternativas utilizadas para a avaliação do sistema. Tem como objetivo medir a facilidade de uso do sistema e descobrir o nível de facilidade de memorização das operações, para que assim seja possível descobrir o grau de produtividade deste usuário

Tabela 3 - Questões sobre usabilidade

Questão	Pergunta
1	O sistema foi de fácil utilização? (0, 1, 2, 3, 4, 5) = (Difícil ... Fácil)
2	Os ícones e imagens utilizadas são representativos quanto às funções desempenhadas? (0, 1, 2, 3, 4, 5) = (Pouco ... Muito)
3	As identificações dos links e botões, são adequados? (0, 1, 2, 3, 4, 5) = (Pouco ... Muito)
4	As telas do sistema são atrativas? (0, 1, 2, 3, 4, 5) = (Pouco ... Muito)
5	A sequência de telas é clara e consistente? (0, 1, 2, 3, 4, 5) = (Pouco ... Muito)
6	Os vídeos disponíveis no sistema auxiliaram no entendimento do assunto? (0, 1, 2, 3, 4, 5) = (Pouco ... Muito)
7	O editor para executar programas em Python é útil? (0, 1, 2, 3, 4, 5) = (Pouco ... Muito)

Fonte: Própria

5.2.2 Funcionalidade

Esta categoria é definida como a capacidade do ambiente satisfazer as necessidades especificadas, segundo a NBR ISO/IEC 9126-1 (2001). Os seguintes atributos são especificados nesta categoria:

- Adequação: a ferramenta se propõe a fazer o que é apropriado?
- Acurácia: a ferramenta gera resultados corretos ou conforme o estabelecido?
- Interoperabilidade: a ferramenta é capaz de interagir com os sistemas definidos?
- Conformidade: a ferramenta está de acordo com normas e convenções previstas em leis, normas e descrições similares?
- Segurança de acesso: a ferramenta evita o acesso não autorizado ou acidental à programa e dados?

As questões referentes a funcionalidade e suas alternativas, podem ser observadas na Tabela 4. O objetivo desta categoria é avaliar se as necessidades específicas são satisfeitas, além disso pretende analisar a necessidade real do usuário diante das funcionalidades existentes e de possíveis implementações. Uma funcionalidade ausente no LabPy é um mecanismo busca de informações no ambiente, portanto uma questão específica para verificar se há necessidade foi adicionada ao formulário.

Tabela 4 - Questões sobre funcionalidade

Questão	Pergunta
8	As informações disponíveis no seu perfil são suficientes? (Por exemplo: nome; endereço; e-mail) (0, 1, 2, 3, 4, 5) = (Insuficiente ... Suficiente)
9	Os recursos disponíveis no LabPy (editor Python, lista de exercícios, vídeo aula, conteúdo didático) favorecem seu aprendizado? (0, 1, 2, 3, 4, 5) = (Pouco ... Muito)
10	A disposição das informações (menus, botões, questões, mensagens, etc) nas telas são claras e organizadas? (0, 1, 2, 3, 4, 5) = (Confusas ... Claras)
11	É necessário recursos de busca e localização de informações no LabPy? (N, S) = (Não ... Sim)

Fonte: Própria

5.2.3 Satisfação Pedagógica

Este tipo de avaliação tem o intuito de julgar o valor pedagógico do AVA, determinado por seus objetivos de formação e aprendizagem, o conteúdo apresentado e as estratégias aplicadas. Instrumentos e metodologias de avaliação qualitativa, como análise de necessidades, questionários, instrumentos psicológicos, técnicas de observação e de entrevista estruturada, assim como métodos quantitativos de estatísticas de sistemas e testes de performance de usuário tem sido a metodologia predominante das avaliações (Lino, Silva, Harb, Favero, & Brito, 2007).

Lino (2007) define a existência de 17 critérios pedagógicos a se levar em consideração em avaliações de AVAs, são: estruturação do conteúdo; sistemas de ajuda; objetivos de aprendizagem; clareza dos conteúdos; validade do conteúdo; estratégias didáticas; métodos pedagógicos; motivação; experiência do estudante; estilos de aprendizagem; compatibilidade; componente prática; avaliação do processo de aprendizagem; tutoria; coerência com a

proposta pedagógica; filosofia pedagógica. A Tabela 5 apresenta o questionário sobre a categoria satisfação pedagógica.

Os principais objetivos de realizar a avaliação da satisfação pedagógica são:

- Identificar e apresentar o grau de satisfação do usuário diante das mudanças identificadas na primeira avaliação do sistema, e verificar se essas mudanças são favoráveis a um aprendizado melhor;
- Medir a satisfação dos usuários acerca dos itens: feedback, conteúdo proposto, questões e tipos de questões, exercícios e recomendação do sistema por parte do usuário;

Tabela 5 - Questões sobre satisfação pedagógica

Questão	Pergunta
12	A partir da primeira avaliação do sistema, ocorreram melhoras significativas no LabPy? (N, S) = (Não ... Sim)
13	Em relação ao processo de ensino-aprendizagem tradicional (ex: aulas em laboratório), a utilização do LabPy aumentou seu aprendizado? (satisfação pedagógica) (N, S) = (Não ... Sim)
14	O conteúdo disponível, em módulos, contribui para o seu aprendizado? (0, 1, 2, 3, 4, 5) = (Pouco ... Muito)
15	Você considera o conteúdo bem elaborado, claro e objetivo? (0, 1, 2, 3, 4, 5) = (Pouco ... Muito)
16	O feedback oferecido pelo LabPy no momento da execução de programas em Python auxiliam no seu aprendizado? (0, 1, 2, 3, 4, 5) = (Pouco ... Muito)
17	O número de exercícios por módulos foi suficiente para avaliação? (Satisfação) (0, 1, 2, 3, 4, 5) = (Insuficiente ... Suficiente)
18	Os exemplos disponíveis no LabPy são satisfatórios (trechos de programas, ilustrações)? (0, 1, 2, 3, 4, 5) = (Pouco ... Muito)
19	O tipo de questão utilizado no sistema (prática de programação Python) foi suficiente? (0, 1, 2, 3, 4, 5) = (Pouco ... Muito)
20	Você recomendaria esse sistema a um amigo? (0, 1, 2, 3, 4, 5) = (Pouco ... Muito)
21	Você considera o conteúdo satisfatório? (0, 1, 2, 3, 4, 5) = (Pouco ... Muito)

Fonte: Própria

5.2.4 Confiabilidade e Eficiência

Medir a eficiência estabelece a capacidade do sistema apresentar desempenho apropriado relativo à quantidade de recursos utilizados, sob condições específicas em um

determinado período. A confiabilidade apresenta características similares, porém tem o foco relacionado a maturidade do sistema, se o mesmo apresenta falhas e como se comporta quando ocorrem e a recuperação do sistema em relação a estas, ou seja, a possibilidade de um usuário poder parar sua atividade em um determinado ponto e reiniciar do mesmo ponto onde parou (Lino, NBR ISO/IEC 9126-1).

A Tabela 6 e 7 contém as questões referentes as categorias eficiência e confiabilidade, respectivamente. A Tabela 8 apresenta a questão discursiva elaborada.

Tabela 6 - Questões sobre eficiência

Questão	Pergunta
22	O tempo de resposta do sistema às suas ações (transições entre as páginas, execução de programas em Python) é satisfatório? (0, 1, 2, 3, 4, 5) = (Lento ... Rápido)
23	Quantas vezes o sistema travou inesperadamente ou se comportou de forma incompreensível? (0, 1, 2, 3, 4, 5) = (Muito ... Pouco)

Fonte: Própria

Tabela 7 - Questões sobre confiabilidade

Questão	Pergunta
24	Os erros são tratados corretamente pelo sistema (falha de login, término de sessão, recuperação de informação)? (0, 1, 2, 3, 4, 5) = (Ruim ... Bom)
25	Os links funcionam corretamente (confiabilidade)? (0, 1, 2, 3, 4, 5) = (Ruim ... Bom)

Fonte: Própria

Tabela 8 - Questão discursiva

Questão	Pergunta
26	Exponha sua opinião em relação ao LabPy (indique pontos positivos, negativos e melhorias).

Fonte: Própria

5.2.5 Aspectos Gerais da Linguagem Python

Além da avaliação do sistema LabPy, seguindo as categorias expostas anteriormente, é necessário avaliar a opinião dos estudantes em relação a linguagem de programação Python, levando-se em consideração os aspectos gerais da linguagem, tais como: estruturas de sequência, seleção, repetição; abstração (funções, dados); entre outras características. Sendo

assim, quatro questões foram elaboradas para que seja possível verificar a situação dos alunos em relação ao Python. A Tabela 9 apresenta este questionário.

Tabela 9 - Questões sobre aspectos gerais da linguagem Python

Questão	Pergunta
27	Qual o grau de dificuldade das estruturas (repetição, seleção) em Python? (0, 1, 2, 3, 4, 5) = (Fácil ... Difícil)
28	Qual o nível de dificuldade da abstração em relação a linguagem Python? (0, 1, 2, 3, 4, 5) = (Fácil ... Difícil)
29	Como você considera seu aprendizado da linguagem Python? (0, 1, 2, 3, 4, 5) = (Insuficiente ... Suficiente)
30	Você voltaria a programar em Python futuramente? (N, S) = (Não ... Sim)

Fonte: Própria

5.3 CONCLUSÕES DO CAPÍTULO

Neste capítulo, foi elaborada uma proposta de avaliação para coletar os resultados específicos do uso do ambiente por parte dos alunos. Seis categorias foram levadas em consideração (Usabilidade, Funcionalidade, Satisfação pedagógica, Confiabilidade, Eficiência e Aspectos gerais da linguagem Python) e 30 questões foram elaboradas com base nessas categorias, para que resultados mais específicos e adequados possam ser avaliados.

Com base nesse questionário pretende-se avaliar e aprimorar o ambiente proposto, tendo como propósito se adequar às necessidades dos estudantes e tornar o sistema mais prático e útil, para que os usuários tenham uma experiência de ensino-aprendizagem melhor. A opinião dos alunos é coletada, juntamente com as propostas de ajustes e mudanças. A avaliação será feita no semestre seguinte em uma turma do curso de Ciência da Computação da UFPA.

6 CONSIDERAÇÕES FINAIS

Ao longo dos anos os AVA ganharam relevância e seu uso se intensificou em universidades, isto se deve ao fato do avanço de novas tecnologias, como a Web, que permitiram a criação de sistemas interativos e eficientes, para auxiliar estudantes e professores no processo de ensino-aprendizagem. Além disso, a demanda por formatos inovadores de conteúdo no campo da educação aumentou consideravelmente, em especial em cursos de computação como na disciplina de programação que necessita de novos métodos de estudo e de laboratório.

Nesse contexto, este trabalho apresenta uma proposta de ferramenta para o ensino a distância e presencial, com o objetivo de auxiliar nos problemas relacionados ao ensino-aprendizado de algoritmos e programação. O LabPy fornece um ambiente com conteúdo focado no ensino de programação através da linguagem Python, ele:

- Proporciona um ambiente com conteúdo interativo, no qual o estudante tem acesso ao conteúdo e pode exercitar a prática laboratorial de forma presencial e/ou a distância;
- Avalia o aluno através de provas e listas de exercícios com feedback automático, apresentando um retorno imediato ao aluno; permite múltiplas re-submissões, possibilitando ajustes na resposta do estudante;
- Proporciona um ambiente gerenciável por parte do professor, que faz o cadastro da turma, tem acesso a notas dos alunos, cadastra exercícios, disponibiliza provas e faz controle de seus alunos.

Um sistema deste tipo sistema para atividades práticas de laboratório bem como para avaliações, proporciona benefícios para alunos e professores:

- Todo exercício receberá um feedback imediato; o estudante ganha tempo; não existe a necessidade de uma espera para saber se ele está indo na direção certa;
- Em tempo hábil, o professor identifica os alunos que possuem dificuldades em certos exercício, podendo assim tomar as providencias necessárias;
- Ambiente flexível, o aluno pode praticar os exercícios a qualquer hora em qualquer local;
- Reduz a carga de trabalho do professor em relação a correção dos exercícios e provas.

O LabPy foi implementado em PHP, utilizando tecnologias Web como o HTML5, JavaScript, CSS3 e Bootstrap. A ferramenta apresenta 12 classes, e um banco de dados contendo 9 tabelas.

Uma proposta de avaliação baseada em um questionário contendo 30 questões foi elaborada buscando testar e aperfeiçoar o sistema. A avaliação deverá ser aplicada em uma turma de programação do curso de Ciência da Computação da UFPA. Assim é possível obter resultados mais adequados do ambiente, observar seu real comportamento e realizar modificações de acordo com a satisfação dos estudantes.

O desenvolvimento deste projeto irá contribuir na compreensão durante a elaboração dos códigos, reduzir o tempo de *feedback* que será retornado ao estudante. O professor terá acesso ao relatório de desempenho dos estudantes, que permite identificar os estudantes que tem uma dificuldade maior na matéria, sendo assim o educador pode adotar estratégias de ensino mais adequadas.

6.1 SUGESTÕES PARA TRABALHOS FUTUROS

Como trabalhos futuros, há planos para:

- Realizar testes práticos em uma turma de programação através da aplicação da proposta de avaliação tendo como base o questionário elaborado, para que dados mais específicos e a opinião dos alunos sejam levadas em consideração;
- Aperfeiçoar a funcionalidade de feedback automático utilizando métricas mais precisas;
- Criar um fórum em que os estudantes e seu professor possam interagir, trocar conhecimentos e materiais de estudo
- Permitir outros estilos de exercícios no sistema, como por exemplo: questões de múltipla escolha e questões discursivas;
- Considerar outras formas de pontuação na avaliação dos alunos;
- Estudar o problema do plágio/cola.

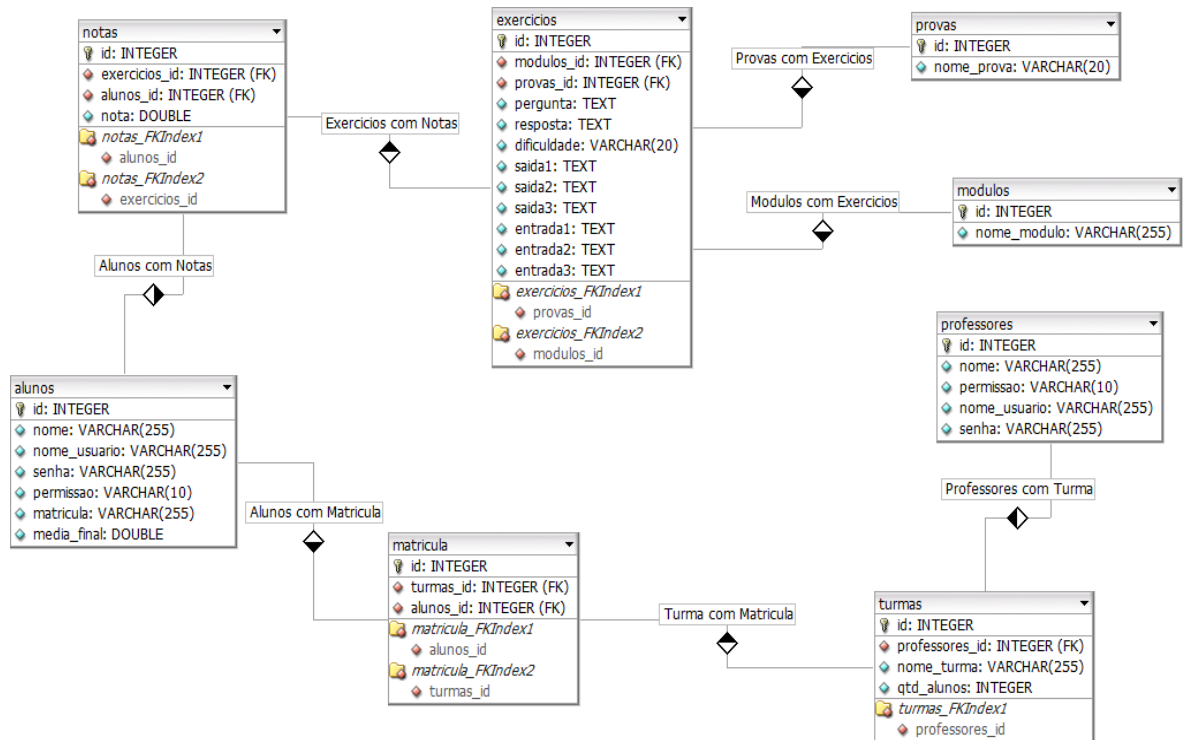
REFERÊNCIAS BIBLIOGRÁFICAS

- Cui, L., Tso, F., Yao, D., & Jia, W. (2012). WeFiLab: A Web-Based WiFi Laboratory Platform for Wireless Networking Education. *IEEE Transactions on Learning Technologies*, pp. 291-303.
- DCC; IME; USP;. (2015). *Panda - Cursos de Computação*. Acesso em 10 de Outubro de 2017, disponível em IME: <https://panda.ime.usp.br/panda/python/index>
- Gaudencio, M., Wanderley, L., Lemos, F., Araújo, E., Figueiredo, J., & Guerrero, D. (2013). Eu Sei o que Vocês Fizem (Agora e) na Aula Passada: o TSTView no Acompanhamento de Exercícios de Programação. *II Congresso Brasileiro de Informática na Educação*, pp. 204-213.
- Gomes, A., Areias, C., Henriques, J., & Mendes, A. (2008). Aprendizagem de Programação de Computadores: Dificuldades e Ferramentas de Suporte. *Revista Portuguesa de Pedagogia*, pp. 161-179.
- Grandell, L., Peltomaki, M., Back, R. J., & Salakoski, T. (2006). Why Complicate Things? Introducing Programming in High School Using Python. *Proceedings of the 8th Australian Conference on Computing Education*, 52, pp. 71-80.
- Junior, S. S., & França, S. A. (2017). Programação para todos: Análise Comparativa de Ferramentas Utilizadas no Ensino de Programação. *25º WEI - Workshop sobre Educação em Computação*, pp. 2199-2208.
- kampff, A. C., & Júnior, G. A. (2010). NetAula - Geração automática de avaliações. *CINTED-UFRGS*, pp. 1-14.
- Kemczinski, A. (2005). Método de Avaliação para Ambientes e-learning. *Tese (Doutorado)*.
- Kemczinski, A., Gasparini, I., & Hounsell, M. S. (2006). Avaliação da Usabilidade do Ambiente SIA-AE: Sistema Interativo de Avaliação de Ambientes E-learning. *International Conference IADIS CIAWI*.
- Lino, A., Silva, A., Harb, T., Favero, E., & Brito, S. (2007). Avaliação automática de consultas SQL em ambiente virtual de ensino-aprendizagem. *Conferencia Iberica de Sistemas y Tecnologias de la Informacion*.
- M., S. J., & Nagaraja, G. S. (2013). Virtual Learning Environments – A Survey. *International Journal of Computer Trends and Technology*, IV, pp. 1705-1709.
- Miller, B. (2012). *How to Think Like a Computer Scientist: Interactive Edition*. Acesso em 10 de Outubro de 2017, disponível em Interactive Python: <http://interactivepython.org/runestone/static/thinkcspy/index.html>

- Moreira, M. P., & Favero, E. L. (2009). Um Ambiente para Ensino de Programação com Feedback Automático de Exercícios. *WEI - XVII Workshop sobre Educação em Computação*, pp. 429-438.
- Mota, M. P., Favero, E. L., Brito, S., & Moreira, M. (2009). Ambiente Integrado à Plataforma Moodle para Apoio ao Desenvolvimento das Habilidades Iniciais de Programação. *SBC - Anais do XVII Congresso da SBC*, pp. 127-136.
- Nielsen, J. (1993). *Usability Engineering*.
- Paes, R., Malaquias, R., Guimarães, M., & Almeida, H. (2013). Ferramenta para a Avaliação de Aprendizado de Alunos em Programação de Computadores. *II Congresso Brasileiro de Informática na Educação*, pp. 203-212.
- Pereira, A. C., Schmitt, V., & Dias, M. Á. (2007). *Ambientes Virtuais de Aprendizagem: Em Diferentes Contextos*. Rio de Janeiro: Editora Ciência Moderna.
- Pereira, L. d., & França, G. (2013). Os Ambientes Virtuais de Aprendizagem (AVA): Um Estudo do MOODLE no Curso de Pedagogia da UFT. *InterSciencePlace*, pp. 77-92.
- Pressman, R. (2016). *Software Engineering: A Practioner's Approach* (8ª ed.). McGraw Hill.
- PyScience. (s.d.). *Python: O que é? Por que usar?* Acesso em 10 de Outubro de 2017, disponível em PyScience-Brasil: <http://pyscience-brasil.wikidot.com/python:python-oq-e-pq>
- Python Software Foundation. (2001). *Aprenda a Programar*. Acesso em 10 de Outubro de 2017, disponível em Python Brasil: https://wiki.python.org.br/AprendaProgramar#Porqu.2BAOo_Python
- Python Software Foundation. (2001). *Python 3.6.4 documentation*. Acesso em 10 de Outubro de 2017, disponível em Python: <https://docs.python.org/3/>
- Raiol, E. M., Natal, I., & Favero, E. (2014). WebSQL: Uma ferramenta de apoio ao processo de ensino e aprendizagem de SQL baseado na recomendação de objetos de aprendizagem e competências do aluno em SQL. *CINTED*, pp. 1-10.
- Reis, T., Castro, J., & Olsina, L. (2002). Medição de Qualidade de Aplicações Web na Fase de Requisitos. *Simpósio Brasileiro de Engenharia de Software*, pp. 162-174.
- Ribeiro, E. N., Mendonça, G. A., & Mendonça, A. F. (2007). A Importância dos Ambientes Virtuais de Aprendizagem na Busca de Novos Domínios de EAD. *Congresso Internacional ABED de Educação*, pp. 1-10.
- Silva, B. S., & Trentin, M. A. (2016). Dificuldades no Ensino-Aprendizagem de Programação de Computadores: Contribuições para a sua Compreensão e Resolução. *Simpósio Nacional de Ensino de Ciência e Tecnologia*.

- Silva, C. (2002). Maep: um método ergopedagógico interativo de avaliação para produtos educacionais informatizados. *Programa de Pós-graduação em Engenharia de Produção. Tese (Doutorado)*.
- Skulpt. (s.d.). *Python. Client side*. Acesso em 8 de Setembro de 2017, disponível em Skulpt: <http://www.skulpt.org/>
- TIOBE Software BV. (2000). *TIOBE Index for February*. Acesso em 17 de Fevereiro de 2018, disponível em TIOBE - The Software Quality Company: <https://www.tiobe.com/tiobe-index/#null>
- Werneck, V., & Moraes, E. (2003). Uma Abordagem de Avaliação de Qualidade de Aplicações Web. Acesso em 25 de Janeiro de 2018, disponível em Uma Abordagem de Avaliação: https://www.researchgate.net/profile/Vera_Werneck/publication/237352146_Uma_Abordagem_de_Avaliacao_de_Qualidade_de_Aplicacoes_Web/links/542c885c0cf277d58e8c81a4/Uma-Abordagem-de-Avaliacao-de-Qualidade-de-Aplicacoes-Web.pdf

APÊNDICE A – Diagrama Entidade-Relacionamento do Sistema



APÊNDICE B – Trechos de Código do Sistema

A Tabela 10 apresenta informações sobre os detalhes técnicos de implementação do sistema, como linguagem de programação, número de linhas de código e quantidade de classe utilizadas.

Tabela 10 - Detalhes de implementação

Linguagem	Número de linhas de código	Quantidade de classes
PhP	552	12
HTML	7000	0
CSS	105	0
JavaScript	103	2

Fonte: Própria

JavaScript:

```

var codeMirrorElement;
function outf(text) {
  var mypre = document.getElementById("dynamicframe");
  text = text.replace(/</g, '&lt;');
  mypre.innerHTML = mypre.innerHTML + text;
}

function builtinRead(x) {
  if (Sk.builtinFiles === undefined || Sk.builtinFiles["files"][x] === undefined)
    throw "File not found: " + x + "";
  return Sk.builtinFiles["files"][x];
}

function runit() {
  var prog = codeMirrorElement.getValue();
  var mypre = document.getElementById("dynamicframe");
  mypre.innerHTML = "";
  Sk.pre = "dynamicframe";
  Sk.configure({
    inputfun: function(prompt) {
      return window.prompt(prompt);
    },
    inputfunTakesPrompt: true,
    output: outf,
    read: builtinRead
  });
  var myPromise = Sk.misceval.asyncToPromise(function() {
    return Sk.importMainWithBody("<stdin>", false, codeMirrorElement.getValue(),
true);
  });

```

```

});
myPromise.then(function(mod) {
    console.log('success');
    },
    function(err) {
        sErr = e.toString();
        alert(sErr);
        outf(sErr);
    });

var id;
var user_href;
var user_href_splitted;

user_href = $("#user-id").prop('href');
user_href_splitted = user_href.split("=");
id = user_href_splitted[user_href_splitted.length - 1];

var teste = document.getElementById("dynamicframe");
$(document).ready(function() {
    $.get("ajax.php",
        { x: teste.innerHTML,
          id: id,
          prog: prog },
        function (data) {
            document.getElementById("preview").innerHTML = data;
        }, "html");
    });
}
window.onload = function() {
    codeMirrorElement = CodeMirror.fromTextArea(document.getElementById('textbox'), {
        mode: {
            name: "python",
            version: 3,
            singleLineStringErrors: false
        },
        lineNumbers: true,
        indentUnit: 4
    });
}

```

PHP, Correção de Exercícios:

```

<?php
ob_start();
require_once("admin/includes/init.php");
$id = $_GET['id'];
$x = $_GET['x'];
// $a = $_GET['a'];
$resp = $_GET['prog'];

```

```

$cont = 0;
$result = 0;
$exercicio = Exercicios::seleciona_por_id($id);
$stri1 = Exercicios::Ngrams($exercicio->resposta, 2);
$stri2 = Exercicios::Ngrams($resp, 2);

if (trim($x) == $exercicio->saida1 || trim($x) == $exercicio->saida2 || trim($x) ==
$exercicio->saida3) {
    echo "Saída correta!";
    $result = (count($stri2)*100)/count($stri1);
    if (count($stri2) < count($stri1)) {
        $result += 10;
    } elseif (count($stri2) > count($stri1)) {
        $result -= 7;
    }
    echo "<br>Precisão da Solução: $result%";
} else {
    echo "Saída incorreta!<br>Continue tentando!<br>";
    if (count($stri2) >= count($stri1)) {
        $result = (count($stri1)*100)/count($stri2);
        $result -= 30;
    } else {
        $result = (count($stri2)*100)/count($stri1);
        $result -= 30;
    }
    echo "<br>Precisão da Solução: $result%";
    // echo "<br>Resposta esperada: $exercicio->resposta";
}
?>

```

PHP, Classe Exercícios:

```

<?php
class Exercicios extends Db_object {
    protected static $db_table = "exercicios";
    protected static $db_table_fields = array('pergunta', 'resposta', 'modulo_id',
'nota_exercicio', 'dificuldade', 'saida1', 'saida2', 'saida3', 'entrada1', 'entrada2', 'entrada3');
    public $id;
    public $pergunta;
    public $resposta;
    public $modulo_id;
    public $dificuldade;
    public $saida1;
    public $saida2;
    public $saida3;
    public $entrada1;
    public $entrada2;
    public $entrada3;
    public $nota_exercicio = 0;
}

```

```

public static function encontrar_exercicios($modulo_id = 0, $lim) {
    global $database;
    $sql = "select * from " . self::$db_table . " ";
    $sql .= "where modulo_id = " . $database->escape_string($modulo_id) . " ";
    $sql .= "order by RAND() limit " . $lim . """;
    return self::encontra_por_query($sql);
}

public static function buscar_exercicios($modulo_id = 0) {
    global $database;
    $sql = "select * from " . self::$db_table . " ";
    $sql .= "where modulo_id = " . $database->escape_string($modulo_id) . """;
    return self::encontra_por_query($sql);
}

public static function conta_exercicios($modulo_id = 0) {
    global $database;
    $sql = "select count(*) from " . self::$db_table . " ";
    $sql .= "where modulo_id = " . $database->escape_string($modulo_id) . """;
    return self::encontra_por_query($sql);
}

public static function Ngrams($word,$n=3){
    $len=strlen($word);
    $ngram=array();

    for($i=0;$i+$n<=$len;$i++){
        $string="";
        for($j=0;$j<$n;$j++){
            $string.=$word[$j+$i];
        }
        $ngram[$i]=$string;
    }
    return $ngram;
}
}
?>

```

HTML:

```

<nav class="navbar navbar-inverse navbar-fixed-top" role="navigation">
  <?php include("includes/top_nav.php"); ?>
  <?php include("includes/side_nav.php"); ?>
</nav>
<div id="page-wrapper">
  <div class="container-fluid">
    <!-- Page Heading -->
    <div class="row">

```

```

<div class="col-lg-12">
  <h1 class="page-header">
    Cadastrar um Novo Exercício <br>
    <small>Preencha os campos abaixo</small>
  </h1>
  <form action="" method="post" class="form">
    <div class="col-md-6 col-md-offset-3">
      <h3>Dados da Questão</h3>
      <div class="form-group">
        <label for="modulo">Módulo</label>
        <select class="form-control" id="modulo" name="modulo" required>
          <option></option>
          <option value="0" class="mdl-menu__item">Nenhuma</option>
          <?php foreach ($modulos as $modulo) : ?>
            <option value="<?php echo $modulo->id; ?>" class="mdl-
menu__item"><?php echo $modulo->nome_modulo; ?></option>
          <?php endforeach; ?>
        </select>
      </div>
      <div class="form-group">
        <label for="prova">Dificuldade</label>
        <select class="form-control" id="dificuldade" name="dificuldade"
required>
          <option></option>
          <option value="Facil" class="mdl-menu__item">Facil</option>
          <option value="Medio" class="mdl-menu__item">Medio</option>
          <option value="Dificil" class="mdl-menu__item">Dificil</option>
        </select>
      </div>
      <div class="form-group">
        <label for="AboutMe">Enunciado</label>
        <textarea class="form-control" type="text" rows="5" cols="66"
id="AboutMe" name="pergunta" required></textarea>
      </div>
      <div class="form-group">
        <label for="AboutMe">Resposta</label>
        <textarea class="form-control" type="text" rows="5" cols="66"
id="AboutMe" name="resposta" required></textarea>
      </div>
      <div class="form-group">
        <label for="entrada1">Entrada 1</label>
        <input class="form-control" type="text" id="entrada1" name="entrada1"
required/>
      </div>
      <div class="form-group">
        <label for="entrada2">Entrada 2</label>

```

```

        <input class="form-control" type="text" id="entrada2" name="entrada2"
required/>
    </div>

    <div class="form-group">
        <label for="entrada3">Entrada 3</label>
        <input class="form-control" type="text" id="entrada3" name="entrada3"
required/>
    </div>

    <div class="form-group">
        <label for="saida1">Saida 1</label>
        <input class="form-control" type="text" id="saida1" name="saida1 "
required/>
    </div>

    <div class="form-group">
        <label for="saida2">Saida 2</label>
        <input class="form-control" type="text" id="saida2" name="saida2"
required/>
    </div>

    <div class="form-group">
        <label for="saida3">Saida 3</label>
        <input class="form-control" type="text" id="saida3" name="saida3"
required/>
    </div>

    <div class="form-group">
        <label for="secondName">Nota do Exercício</label>
        <input class="form-control" type="number" id="secondName" min="0"
max="10" value="10" name="nota_exercicio" required/>
    </div>
    <button id="submit_button" class="btn btn-primary pull-right" name="criar">
        ADD
    </button>
</div>
</form>
</div>
</div>
</div>
</div>

```

CSS:

```

.admin-photo-thumbnail {
    width: 200px;
    border-radius: 5px;
}

```

```
.user_image {  
  width: 170px;  
  height: 100px;  
}  
  
.photo-info-box {  
  border: 1px solid #e5e5e5;  
  -webkit-box-shadow: 0 1px 1px rgba(0, 0, 0, .04);  
  box-shadow: 0 1px 1px rgba(0, 0, 0, .04);  
  background-color: #fff;  
}  
  
.info-box-header {  
  padding: 6px;  
  border-bottom: 1px solid #e5e5e5;  
  -webkit-box-shadow: 0 1px 1px rgba(0, 0, 0, .04);  
  box-shadow: 0 1px 1px rgba(0, 0, 0, .04);  
  background-color: #fff;  
}  
  
.photo-info-box .box-inner {  
  border-top: 1px solid #e5e5e5;  
  background-color: #fff;  
  padding: 6px;  
}  
  
.photo-info-box .box-inner .text {  
  padding: 7px 9px 19px;  
}  
  
.photo-info-box .box-inner .data {  
  font-weight: bold;  
}  
  
.photo-info-box .info-box-footer {  
  border-top: 1px solid #ddd;  
  background-color: #ddd;  
  padding: 10px;  
}  
  
.modal-dialog {  
  width: 90% !important;  
}  
  
.modal-body {  
  overflow: auto;  
  height: 500px !important;  
  min-height: 300px !important;  
}
```

```
.modal_thumbnails {  
  width: 150px;  
  height: 100px !important;  
}  
  
#set_user_image {  
  margin: 10px;  
}  
  
#modal_sidebar {  
  position: fixed !important;  
}  
  
#login-id label {  
  color: #fff;  
}
```

APÊNDICE C – Trecho de Código sobre Módulo

```

<?php include("includes/header.php"); ?>
<?php if (!$session->is_signed_in()) { redirect("admin/login.php"); } ?>
<script type="text/javascript" src="js/default.js"></script>
<style media="screen">
  ol {
    font-size: 20px;
  }
  pre {
    font-size: 20px;
  }
</style>
<div class="row">
  <div class="col-md-9">
    <div class="section" id="modulos-e-obtendo-ajuda">
      <h1>Módulos e Obtendo Ajuda<a class="headerlink" href="#modulos-e-obtendo-ajuda" title="Permalink to this headline"></a></h1>
      <div class="section" id="modulos">
        <h2>Módulos<a class="headerlink" href="#modulos" title="Permalink to this headline"></a></h2>
        <a id="inputvid_thumb" ></a>
        <div id="inputvid" class="video_popup" >
          <video controls >
            <source src="http://media.interactivepython.org/thinkcsVideos/modules.mov"
type="video/mp4"></source>
            <source src="http://media.interactivepython.org/thinkcsVideos/modules.webm"
type="video/webm"></source>
            No supported video types
          </video>
        </div>
        <script>
          jQuery(function($) {
            $('#inputvid_thumb').click(function(e) {
              $('#inputvid').show();
              $('#inputvid_thumb').hide();
              logBookEvent({'event':'video','act':'play','div_id': 'inputvid'});
            });
          });
        </script>
        <p>Um <strong>módulo</strong> é um arquivo contendo definições e comandos em Python para serem usados em outros programas em Python. Há diversos módulos do Python que fazem parte da <strong>biblioteca padrão</strong>. Lembre-se que uma vez que o módulo é importado, podemos utilizar as coisas que estão definidas dentro dele.</p>
        <p>A documentação para a versão 3 do Python está disponível (em inglês) no sítio <a class="reference external" href="http://docs.python.org/py3k/">http://docs.python.org/py3k/</a> (a página é mostrada

```

abaixo). Essa é uma referência muito útil sobre todos os aspectos do Python. Esse sítio contém uma listagem de todos os módulos padrões disponíveis no Python (veja [Global Module Index](http://docs.python.org/py3k/py-modindex.html)).

Observe também que existe um manual de referência à linguagem ([Language Reference](http://docs.python.org/py3k/reference/index.html)) e um tutorial ([Tutorial](http://docs.python.org/py3k/tutorial/index.html)), bem como instruções para instalação, dicas de como fazer, e respostas a perguntas frequentes. Recomendamos que você tente se familiarizar com essas informações e se acostume a utilizá-las com frequência.

```

```

Se você ainda não o fez, dê uma olhada na lista de módulos ([Global Module Index](http://docs.python.org/py3k/py-modindex.html)). Nessa lista você vai encontrar, em ordem alfabética, todos os módulos disponíveis na biblioteca padrão.

```

```

```
<div class="admonition-nota-modulos-do-python-e-limitacoes-do-activecode admonition">
```

```
<p class="first admonition-title">Nota: Módulos do Python e limitações do activecode</p>
```

```
<p class="last">Em todos os capítulos desse livro, as janelas do activecode permitem que você pratique as partes do Python que você está aprendendo.</p>
```

Nós mencionamos no primeiro capítulo que a atividade de programação é normalmente realizada dentro de um ambiente de desenvolvimento e que o activecode utilizado aqui serve apenas para nos ajudar a aprender. Essa não é a maneira de desenvolver programas reais.

```
</p>
</div>
```

```
<div class="section" id="mais-sobre-o-uso-de-modulos">
<h2>Mais sobre o uso de módulos<a class="headerlink" href="#mais-sobre-o-uso-de-modulos" title="Permalink to this headline"></a></h2>
```

Antes de prosseguir com a exploração dos outros módulos, nós vamos falar um pouco mais sobre o que são módulos e como são tipicamente usados. Uma das coisas mais importantes a saber sobre módulos é o fato de que eles são objetos de dados, assim como qualquer outro tipo de dado em Python. Objetos do tipo `module` simplesmente contém outros elementos do Python.

A primeira coisa que precisamos fazer quando queremos usar um módulo é fazer um `<code class="docutils literal">import</code>`.

Agora vamos mostrar alguns outros módulos que você pode achar útil.

```
<a id="randmodvid_thumb" > </a>
```

```
<div id="randmodvid" class="video_popup" >
```

```
<video controls >
```

```
<source
```

```
src="http://media.interactivepython.org/thinkcsVideos/mathrandommodule.mov"
type="video/mp4"></source>
```

```

    <source
src="http://media.interactivepython.org/thinkcsVideos/mathrandommodule.webm"
type="video/webm"></source>
    No supported video types
</video>
</div>
<script>
jQuery(function($) {
    $('#randmodvid_thumb').click(function(e) {
        $('#randmodvid').show();
        $('#randmodvid_thumb').hide();
        logBookEvent({'event':'video','act':'play','div_id': 'randmodvid'});
        // Log the run event
    });
});
</script>
</div>
<div class="section" id="o-modulo-math">
<h2>O módulo <code>math</code><a class="headerlink" href="#o-modulo-math"
title="Permalink to this headline"></a></h2>
<p>O módulo <code class="docutils literal"><span
class="pre">math</span></code> contém funções matemáticas que você costuma encontrar
em calculadoras e algumas constantes matemáticas como <code>pi</code> e <code>e</code>.
Como vimos acima, ao importar o módulo <code class="docutils literal"><span
class="pre">math</span></code>, obtemos uma referência para um objeto
<code>module</code> que contém esses elementos.</p>

<p>Mostramos a seguir como usar alguns itens do módulo math. Se você quiser
saber mais detalhes, dê uma olhada na documentação do módulo em
<a class="reference external"
href="http://docs.python.org/py3k/library/math.html#module-math">Math Module</a>.</p>

<div id="chmodule_02" >
<textarea cols="50" rows="12" id="chmodule_02_code" class="active_code">
import math

print(math.pi)
print(math.e)

print(math.sqrt(2.0))

print(math.sin(math.radians(90))) # seno de 90 graus

</textarea>
<p class="ac_caption"><span class="ac_caption_text"> (chmodule_02)</span>
</p>
<button onclick="runit('chmodule_02',this, undefined);">Run</button>
<button class="ac_opt" onclick="saveEditor('chmodule_02');">Save</button>
<button class="ac_opt" onclick="requestCode('chmodule_02');">Load</button>
<br />

```

```
<canvas id="chmodule_02_canvas" height="400" width="400" style="border-style: solid; display: none"></canvas>
```

```
<pre id="chmodule_02_pre" class="active_out">
```

```
</pre>
```

```
</div>
```

<p>Funções matemáticas não precisam ser construídas. Elas simplesmente executam uma tarefa. Elas ficam todas alojadas no módulo chamado <code>math</code>. Uma vez que o módulo math tenha sido importado, qualquer coisa definida lá pode ser usada no nosso programa. Observe que usamos sempre o nome do módulo seguido de um <code>ponto</code> seguido pelo item específico do módulo (<code class="docutils literal">math.sqrt</code>). Você pode pensar nisso como sobrenome.nome onde o sobrenome é o módulo

(família) e o nome indica o item individual no módulo.</p>

<p>Se você não tiver feito isso ainda, dê uma olhada na documentação do módulo <code>math</code>.</p>

```
<div class="section" id="o-modulo-random">
```

```
<h2>O módulo <code>random</code><a class="headerlink" href="#o-modulo-random" title="Permalink to this headline"></a></h2>
```

<p>Muitas vezes queremos usar números aleatórios em programas. Aqui estão alguns usos típicos:</p>

```
<ul class="simple">
```

```
<li>Para jogar um jogo de azar em que o computador precisa jogar alguns dados, escolher um número, ou jogar uma moeda,</li>
```

```
<li>Para embaralhar um baralho de cartas aleatoriamente,</li>
```

```
<li>Para permitir que uma nova nave inimiga apareça aleatoriamente e atire em você,</li>
```

```
<li>Para simular a precipitação de chuva em um modelo computacional para estimar o impacto ambiental da construção de uma barragem,</li>
```

```
<li>Para criptografar sua sessão bancária na Internet.</li>
```

```
</ul>
```

<p>O Python fornece um módulo <code class="docutils literal">random</code> que ajuda com tarefas como esta. Você pode dar uma olhada na documentação. Aqui estão as principais coisas que podemos fazer com esse módulo.</p>

```
<div id="chmodule_rand" >
```

```
<textarea cols="50" rows="12" id="chmodule_rand_code" class="active_code">
import random
```

```

prob = random.random()
print(prob)
```

```

lanceDado = random.randrange(1,7) # retorna um int, dentre 1,2,3,4,5,6
print(lanceDado)
```

```
</textarea>
```

```

</p>
<p class="ac_caption"><span class="ac_caption_text"> (chmodule_rand)</span>
</p>
<button onclick="runit('chmodule_rand',this, undefined);">Run</button>
<button class="ac_opt" onclick="saveEditor('chmodule_rand');">Save</button>
<button class="ac_opt" onclick="requestCode('chmodule_rand');">Load</button>
<br />

<canvas id="chmodule_rand_canvas" height="400" width="400" style="border-
style: solid; display: none"></canvas>

<pre id="chmodule_rand_pre" class="active_out">

</pre>

</div>

```

Pressione o botão **Run** algumas vezes. Note que os valores se modificam

a cada vez. Esses números são aleatórios.

A função `randrange` gera um inteiro no intervalo definido pelo menor e maior argumento, usando a mesma semântica do `range` ou seja, o valor inicial é incluído mas o final é excluído. Todos os valores tem a mesma probabilidade de ocorrência (ou seja, os resultados são **uniformemente** distribuídos).

A função `random()` retorna um número em ponto flutuante no intervalo [0.0, 1.0); o colchete significa intervalo fechado à esquerda; e o parênteses significa intervalo aberto à direita. Em

outras palavras, 0.0 é possível, mas todos os números retornados serão estritamente menores que 1.0. É comum **escalar** os resultados após chamar esse método para transformá-los para dentro de um intervalo apropriado para a sua aplicação.

Nesse caso, nós convertemos o resultado da chamada do método em um número no intervalo [0.0, 5.0). Novamente, esses números são uniformemente distribuídos; números perto do 0 são tão prováveis de aparecer quanto o 0.5, ou números perto de 1.0. Se você continuar a pressionar o botão **Run**, você verá valores aleatórios de 0.0 até o limite de, mas sem incluir o valor, 5.0.

```

<div id="chmodule_rand2" >
<textarea cols="50" rows="12" id="chmodule_rand2_code" class="active_code">
import random

prob = random.random()
resultado = prob * 5
print(resultado)

</textarea>

```

```

</p>
<p class="ac_caption"><span class="ac_caption_text"> (chmodule_rand2)</span>
</p>
<button onclick="runit('chmodule_rand2',this, undefined);">Run</button>
<button class="ac_opt" onclick="saveEditor('chmodule_rand2');">Save</button>
<button class="ac_opt" onclick="requestCode('chmodule_rand2');">Load</button>
<br />

<canvas id="chmodule_rand2_canvas" height="400" width="400" style="border-
style: solid; display: none"></canvas>

<pre id="chmodule_rand2_pre" class="active_out">

</pre>

</div>

```

É importante notar que geradores de números aleatórios são baseados em um algoritmo **determinístico** e reproduzível e previsível.

Por isso eles são chamados de geradores **pseudo-aleatórios**; pois eles não são verdadeiramente aleatórios.

Eles começam com um valor de **semente**. Cada vez que você pedir um outro número aleatório, você vai receber um com base no valor atual da semente, e o estado da semente (que é um dos atributos do gerador) é atualizado. A boa notícia é que cada vez que você executar o seu programa, o valor da semente é provavelmente diferente, significando que apesar dos números aleatórios estarem sendo criados usando algoritmos determinísticos, você vai obter um comportamento aleatório cada vez que você executar.

```

<div class="section" id="glossario">
<h2>Glossário<a class="headerlink" href="#glossario" title="Permalink to this
headline"></a></h2>
<dl class="glossary docutils">
<dt id="term-biblioteca-padrao">biblioteca padrão</dt>
<dd>Uma coleção de módulos que faz parte de uma instalação padrão do
Python.</dd>
<dt id="term-deterministico">determinístico</dt>
<dd>Um processo que é reproduzível e previsível.</dd>
<dt id="term-documentacao">documentação</dt>
<dd>Um lugar onde você pode pegar informação detalhada sobre aspectos da
linguagem de programação.</dd>
<dt id="term-gerador-de-numeros-aleatorios">gerador de números aleatórios</dt>
<dd>Uma função que te fornece números aleatórios, geralmente entre 0.0 e
1.0.</dd>
<dt id="term-modulo">módulo</dt>
<dd>Um arquivo contendo definições e comandos em Python para serem
usados em outros programas em Python. Os conteúdos de um
módulo são disponibilizados para outro programa pelo uso do
comando import.</dd>
<dt id="term-numero-aleatorio">número aleatório</dt>
<dd>Um número que é gerado de forma a exibir aleatoriedade estatística.</dd>
<dt id="term-numero-pseudo-aleatorio">número pseudo-aleatório</dt>

```

```

    <dd>Um número que não é genuinamente aleatório mas criado a partir de um
    algoritmo.</dd>
  </dl>
</div>
<div class="section" id="exercicios">
  <h2>Exercícios<a class="headerlink" href="#exercicios" title="Permalink to this
  headline"></a></h2>
  <div class="toctree-wrapper compound">
  </div>
</div>
<?php $id_mod = $_GET['id']; ?>
<div class="list-group">
  <a href="exercicio_lista.php?id=?php echo $id_mod; ?>" class="list-group-
  item"><h4>Exercícios Módulo <?php echo $id_mod; ?></h4></a>
  </div>
</div>
</div>
</div>
</div>
</div>
</div>
<?php include("includes/sidebar.php"); ?>
</div>
<?php include("includes/footer.php"); ?>

```

APÊNDICE D – Trecho de Código sobre Banco de Dados

```

<?php
class Db {
    public static function seleciona_tudo() {
        return static::encontra_por_query("select * from " . static::$db_table . " ");
    }

    public static function seleciona_por_id($id) {
        global $database;
        $array_resultado = static::encontra_por_query("select * from " . static::$db_table . "
where id = $id limit 1");
        return !empty($array_resultado) ? array_shift($array_resultado) : false;
    }

    public static function encontra_por_query($sql) {
        global $database;
        $result_set = $database->query($sql);
        $objeto_array = array();
        while ($row = mysqli_fetch_array($result_set)) {
            $objeto_array[] = static::instanciar($row);
        }
        return $objeto_array;
    }

    public static function instanciar($the_record) {
        $calling_class = get_called_class();
        $objeto = new $calling_class;
        foreach ($the_record as $the_attribute => $value) {
            if ($objeto->tem_atributo($the_attribute)) {
                $objeto->$the_attribute = $value;
            }
        }
        return $objeto;
    }

    private function tem_atributo($the_attribute) {
        $object_propriedades = get_object_vars($this);
        return array_key_exists($the_attribute, $object_propriedades);
    }

    protected function propriedades() {
        $propriedades = array();
        foreach (static::$db_table_fields as $db_field) {
            if (property_exists($this, $db_field)) {
                $propriedades[$db_field] = $this->$db_field;
            }
        }
        return $propriedades;
    }
}

```

```

protected function limpa_propriedades() {
    global $database;
    $limpa_propriedades = array();
    foreach ($this->propriedades() as $key => $value) {
        $limpa_propriedades[$key] = $database->escape_string($value);
    }
    return $limpa_propriedades;
}

public function save() {
    return isset($this->id) ? $this->update() : $this->create();
}

public function create() {
    global $database;
    $propriedades = $this->limpa_propriedades();
    $sql = "insert into " . static::$db_table . " (" . implode(",", array_keys($propriedades))
    . ") ";
    $sql .= "values (" . implode("','", array_values($propriedades)) . ")";
    if ($database->query($sql)) {
        $this->id = $database->id_inserido();
        return true;
    } else {
        return false;
    }
}

public function update() {
    global $database;
    $propriedades = $this->limpa_propriedades();
    $propriedades_pairs = array();
    foreach ($propriedades as $key => $value) {
        $propriedades_pairs[] = "{$key}='{$value}'";
    }
    $sql = "update " . static::$db_table . " set ";
    $sql .= implode(" , ", $propriedades_pairs);
    $sql .= " where id = " . $database->escape_string($this->id);

    $database->query($sql);

    return (mysqli_affected_rows($database->connection) == 1) ? true : false;
}

public function delete() {
    global $database;

    $sql = "delete from " . static::$db_table . " ";
    $sql .= "where id = " . $database->escape_string($this->id) . " ";
    $sql .= "limit 1";
}

```

```
$database->query($sql);  
return (mysqli_affected_rows($database->connection) == 1) ? true : false;  
}
```

```
public static function count_all() {  
    global $database;  
    $sql = "select count(*) from " . static::$db_table;  
    $result_set = $database->query($sql);  
    $row = mysqli_fetch_array($result_set);  
    return array_shift($row);  
}  
}
```

?>