

EMILIANA MARTINS GONÇALVES

**AS DIFICULDADES DE APRENDIZAGEM DE CONCEITOS DE
PROGRAMAÇÃO DOS ALUNOS DE PRIMEIROS ANOS EM
COMPUTAÇÃO**

Assis

2011

EMILIANA MARTINS GONÇALVES

**AS DIFICULDADES DE APRENDIZAGEM DE CONCEITOS DE
PROGRAMAÇÃO DOS ALUNOS DE PRIMEIROS ANOS EM
COMPUTAÇÃO**

Projeto de pesquisa apresentado ao curso de Bacharelado em Ciência da Computação do Instituto Municipal de Ensino Superior de Assis – IMESA e a Fundação Educacional do Município de Assis – FEMA, como requisito parcial à obtenção do Certificado de Conclusão.

Orientado: Emilana Martins Gonçalves

Orientador: Luiz Ricardo Begosso

Assis

2011

FICHA CATALOGRÁFICA

GONÇALVES, Emiliana Martins.

As Dificuldades de Aprendizagem de conceitos de Programação dos alunos de primeiros anos em Computação

52p.

Orientador (a): Luiz Ricardo Begosso

Trabalho de Conclusão de Curso – Instituto Municipal de Ensino Superior de Assis - IMESA

1. Dificuldades. 2. Linguagem de Programação. 3. Python.

CDD:XXX XXX

Biblioteca FEMA

**AS DIFICULDADES DE APRENDIZAGEM DE CONCEITOS DE
PROGRAMAÇÃO DOS ALUNOS DE PRIMEIROS ANOS EM
COMPUTAÇÃO**

EMILIANA MARTINS GONÇALVES

Trabalho de Conclusão de Curso
apresentado ao Instituto Municipal Ensino
Superior de Assis, como requisito do Curso
de Bacharelado de Ciência da Computação
analisado pela seguinte comissão
examinadora.

Orientador (a): _____

Analisador (a): _____

Assis

2011

DEDICATÓRIA

Dedico esse trabalho a minha mãe Ester Martins dos Santos que me deu a oportunidade de cursar uma faculdade e todo apoio necessários. Obrigada.

AGRADECIMENTOS

Agradeço primeiramente a Deus, por me acompanhar e estar comigo em todos os momentos dessa jornada.

Ao professor Luiz Ricardo Begosso, pela orientação durante o trabalho me ajudando em todos os momentos ao decorrer desse trabalho.

Queria agradecer também aos Professores Domingos de Carvalho Villela Junior pelo convite para ministrar o mini curso de introdução a Python e também a Professora Cláudia Patrícia Candia Maciel professora do Etec Pedro D'Arcádia Neto - Centro Paula Souza - Assis/SP, que disponibilizou de suas aulas para realização do estudo de caso, para a conclusão do trabalho.

Gostaria de agradecer umas pessoas em especiais, meus amigos Mariana, Thamiris, Thais, Luciana, Natália, Camila, Patrícia, Laís, Raissa e o Rafael que de uma forma ou de outra me ajudaram na conclusão desse trabalho e por todo apoio dado. Também a todos aquelas pessoas que estiveram presente direto e indiretamente presente ao decorrer do curso e na execução desse trabalho.

Agradeço também a minha mãe em especial, por todo esforço feito para a conclusão desse trabalho, podendo assim finalizar o curso.

RESUMO

Com o avanço tecnológico, o mercado de trabalho disponibiliza inúmeras vagas para profissionais capacitados na área da tecnologia da informação. O profissional de TI requer um grande conhecimento, para que possa entrar no mercado de trabalho com um diferencial entre os demais.

Porém, além da falta de profissionais qualificados serem grande, cada vez mais aumenta o número de alunos desmotivados a finalizar os cursos onde são lecionadas linguagens de programação, pelo elevado nível de abstração do conteúdo aplicado. Por essa razão, o presente trabalho tem por objetivo apresentar as dificuldades dos alunos ao decorrer do curso, os métodos de ensino aplicados e a utilização de Python como primeira linguagem, para proporcionar uma eficácia no entendimento desses alunos.

Palavras-chave: Dificuldades; Linguagem de Programação; Python;

ABSTRACT

With technological advancement, the job market offers numerous vacancies for qualified professionals in the field of information technology. The IT professional requires a great knowledge, so you can enter the job market with a gap between the others.

However, besides the lack of qualified professionals is great, more and more increases the number of unmotivated students to complete courses which are taught programming languages, the high level of abstraction of content applied. Therefore, this paper aims to present the students' difficulties during the course, teaching methods applied and the use of Python as first language, to provide an understanding of the efficacy of these students.

Keywords: difficulty; programming language; Python;

LISTA DE ILUSTRAÇÕES

Figura 3-1 Modo do preparo de um bolo	17
Figura 3-2 Tipos de representação gráfica	18
Figura 3-3 Representação gráfica	19
Figura 3-4 Representação Pseudocódigo	20
Figura 4-1 Python no Menu iniciar do Windows 7	24
Figura 4-2 Janela Principal e janela de edição de textos IDLE	26
Figura 4-3 Janela para criar um arquivo Python	27
Figura 4-4 Janela para desenvolvimento Código Python	27
Figura 4-5 Janela para execução do código Python	27
Figura 4-6 Representação Olá mundo no interpretador Python	29
Figura 4-7 Interpretador Python verificando os tipos de variáveis	30
Figura 4.8 Palavras reservadas Python	32
Figura 4.9 Operadores Aritméticos.....	33
Figura 4.10 Representação em Python para atribuição de um valor á uma variável.	33
Figura 4.11 Operadores de Comparação.....	34
Figura 4.12 Representação em Python para entrada e saída de dados.....	34

SUMÁRIO

1. INTRODUÇÃO	11
1.1. OBJETIVOS	12
1.2. JUSTIFICATIVA	13
1.3. MOTIVAÇÃO	13
1.4. ESTRUTURA DO TRABALHO	14
2. A DIFICULDADE DE APRENDIZADO EM LINGUAGEM DE PROGRAMAÇÃO..	15
3. COMPREENDENDO ALGORITMOS E LINGUAGEM DE PROGRAMAÇÃO.....	17
3.1 ALGORITMOS.....	17
3.1.1 FORMAS DE REPRESENTAÇÕES DE ALGORITMOS.....	17
3.2 DEFINIÇÃO DE LINGUAGEM DE PROGRAMAÇÃO.....	22
3.2.1 LINGUAGEM DE BAIXO NÍVEL.....	22
3.2.2 LINGUAGEM DE ALTO NÍVEL.....	22
4. LINGUAGEM DE PROGRAMAÇÃO PYTHON	23
4.1 PYTHON	23
4.1.1 VANTAGENS	22
4.1.2 DESVANTAGENS.....	23
4.2 INTERPRETADOR.....	25
4.2.1 INSTALANDO O INTERPRETADOR	25
4.3 A INTERFACE GRÁFICA COM O USUÁRIO.....	26
4.4 INICIANDO PROGRAMAÇÃO PYTHON	29
4.4.1 OLÁ MUNDO	29
4.4.2 VARIÁVEIS.....	30
4.4.3 OPERADORES.....	32

4.4.4 ENTRADA E SAIDA DE DADOS.....	34
4.4.5 DESVIOS.....	35
5. ESTUDO DE CASO.....	36
5.1. PROPOSTA.....	36
5.2. ESTUDO REALIZADO.....	38
5.3. EXERCICIOS APLICADOS	48
5.4. QUESTINONÁRIO	45
5.5. CONCLUSÃO SOBRE O TRABALHO REALIZADO	46
6. CONCLUSÕES.....	47
7. REFERÊNCIAS.....	48

1. INTRODUÇÃO:

Estudos revelam que alunos que ingressam nos cursos em que são lecionados conceitos básicos de programação têm um índice elevado de desistências. [1] “Um estudo da Associação das Empresas do Setor mostra que 82% dos alunos de TI abandonam seus cursos, fazendo com que haja uma falta de profissionais qualificados, que pelos cálculos do setor sobram 92 mil vagas nessa área em todo país, e este número pode dobrar até o ano de 2013. Muitos alunos abandonam o curso porque não conseguem acompanhar assuntos como matemáticos, lógica, química e física. Além disso, os cursos são regulares e não acompanham a velocidade do desenvolvimento tecnológico, causando uma defasagem em relação as necessidades do mercado de trabalho (Jornal Nacional, Abril/ 2011)”.

Esses índices podem ser identificados por vários motivos, como pela falta de interesse por parte dos alunos, falta de conhecimento prévio, compreensão e motivação. As Linguagens de programação possuem uma sintaxe complexa, lhas, no qual requer um grande empenho por parte do aluno, que nem sempre está preparado para enfrentar as dificuldades do curso, proveniente do elevado nível de abstração dos conteúdos ensinados. Por ser o ensino de algoritmos fundamental nos cursos de computação, é preciso que seja feito, desde o início, um desenvolvimento no raciocínio lógico para a introdução dos conceitos e práticas de programação. Visando reduzir as dificuldades apresentadas pelos alunos, vários métodos de ensino foram criados para minimizar o elevado nível de desistências e assim aumentar o interesse e motivação para disciplinas onde são lecionadas linguagem de programação.

Existe uma grande diversidade de linguagens de programação, sendo que as mais utilizadas nos cursos que ensinam programação são C, C++, C# e Java, na qual o C++ vem sendo ensinado em 22% dos cursos de computação (McCauley & Manaris, 1998), como primeira linguagem.

Uma linguagem pouco discutida e que vem ganhando seu espaço é Python, criada em 1989 por Guido VanRossum, hoje funcionário da Google (Python Brasil). Python é uma linguagem de fácil aprendizagem, possui uma sintaxe simples, sendo

considerada ideal para alunos que estão aprendendo a programar. Python é uma linguagem de altíssimo nível, sendo possível utilizá-la em diferentes tipos de projetos e também multi-plataforma. Essa linguagem suporta tanto programação procedural, composto de procedimentos e funções, como orientada a objetos que combinam dados e funcionalidades. Comparada a outras linguagens, Python possui uma maneira mais simples ao lidar com orientação a objetos. É classificado como uma linguagem interpretada necessitando de um interpretador, no qual seu código não é compilado diretamente para linguagem de máquina, fazendo com que os programas sejam portáteis, independentemente da arquitetura do computador.

1.1 OBJETIVO

O objetivo desse trabalho é realizar um estudo de caso da viabilidade no ensinamento de Python como primeira linguagem, já que é uma linguagem que possui metodologia de fácil aprendizagem, verificando a possibilidade de ser adotada como alternativa para minimizar as dificuldades apresentadas pelos alunos e conseqüentemente o número de desistências nos cursos nos quais são lecionados linguagens de programação. Será realizado levantamento de dados e serão apresentados os motivos das desistências dos alunos, os estudos feitos referentes a esse problema, as alternativas apresentadas para redução dos números apresentados, Python como primeira linguagem, as vantagens e comparações a outras linguagens.

Será desenvolvido um estudo de caso que será aplicado uma introdução sobre a linguagem Python para alunos do curso Técnico em Informática do módulo I do Centro Paula Souza da cidade de Assis – SP, utilizando o material coletado em livros e pela internet, para que possamos concluir de forma concreta a finalidade do trabalho realizado.

1.2 JUSTIFICATIVA

Devido aos inúmeros casos de desistências registrados nos cursos de Ciência da Computação, desmotivados pelo método de ensino do curso, que possui a linguagem de programação como matéria básica e específica do curso, utilizando de linguagens complexas de difícil entendimento dos alunos que iniciam o curso como primeira linguagem, principalmente daqueles que não possuem conhecimento prévio sobre o assunto são os mais afetados. É necessário que seja aplicado para esses alunos afetados metodologias alternativas, para que eles tenham menos dificuldades para o aprendizado de programação, diminuindo assim o número de desistências nos curso de ciência da computação.

Por não ser muito utilizada e pouco conhecida em comparação a outras linguagens tradicionais, a linguagem de programação Python vem com o propósito de facilitar o entendimento desses alunos, ofertando uma linguagem mais simples e objetiva, motivando-os a concluírem os cursos onde são aplicados esse tipo de disciplina.

1.3 MOTIVAÇÃO

Visando a melhora na qualidade de ensino dos alunos que possuem dificuldades no entendimento do conteúdo aplicado em disciplinas de linguagem de programação, a utilização da linguagem Python como primeira linguagem pode ser uma boa alternativa por ser uma linguagem simples e de fácil compreensão, tendo em vista que as linguagens tradicionais aplicadas nos cursos possuem sintaxe complexa de difícil entendimento.

Outra motivação é a possibilidade de conhecer uma linguagem nova, que não é aplicada no calendário da grade curricular.

Sendo assim o trabalho será desenvolvido baseado nessa linguagem pouco conhecida e pouco discutida, contribuindo para passar o conhecimento adquirido

adiante, mostrando um pouco do que o curso oferece e a abrangência do conteúdo do curso com relação a várias linguagens disponibilizadas.

1.4 ESTRUTURA DO TRABALHO

Este trabalho está estruturado nas seguintes partes: o Capítulo 1 contém a introdução ao projeto proposto. O Capítulo 2 apresenta as dificuldades de aprendizagem de linguagens de programação dos alunos das séries iniciais dos cursos de computação. O Capítulo 3 apresenta os conceitos de linguagens de programação, que são adotados para o ensino de alunos das séries iniciais. O Capítulo 4 faz uma introdução à Linguagem Python, descrevendo com certos detalhes as formas de desenvolvimento de programas simples que poderiam ser adotados para o ensino de alunos em séries iniciais. O Capítulo 5 apresenta o Estudo de Caso proposto para este trabalho, sendo aqui enfatizada a grande contribuição deste projeto, e o Capítulo 6 apresenta as conclusões do trabalho.

2. AS DIFICULDADES DE APRENDIZAGEM EM LINGUAGEM DE PROGRAMAÇÃO

Com o elevado nível de desistência dos alunos que cursam faculdades onde linguagem de programação é a principal disciplina, há de se manter uma atenção especial para tal fato. Aprender a programar é um processo difícil e exigente. Para se tornarem bons programadores os alunos têm de adquirir um conjunto de conhecimentos que vão muito além de conhecer a sintaxe e a semântica de uma linguagem de programação (Gomes, 2000; Takahashi, 1998).

“As linguagens de programação são incorporadas como conteúdo de ensino no laboratório de informática, por que são consideradas uma ferramenta que permite ajudar a melhorar o pensamento e acelerar se desenvolvimento cognitivo (”LITWIN - 1997).

O ensino de linguagem de programação tem como finalidade de desenvolver a capacidade do aluno para resolver problemas simples, para que o mesmo possa abstrair um conhecimento que o leve a progredir para os mais elevados níveis de dificuldades.

Com o decorrer do curso, os níveis de dificuldades vão aumentando, o que leva a um grande número de desistência dos alunos, que são desmotivados pelo grau de abstração das disciplinas. E para que esse elevado nível seja minimizado, os pontos que influenciam tais desistências devem ser identificados. Inúmeros estudos foram feitos levando em conta as dificuldades apresentadas por alunos ao entendimento de linguagem de programação.

Os alunos, ao ingressam em faculdades onde são lecionados linguagem de programação, apresentam características variadas de dificuldades, sendo necessário que haja um enfoque no método de ensino, visado melhorar o entendimento dos alunos. O aprendizado de linguagens de programação requer uma atenção aos detalhes, exigindo um empenho e dedicação por parte do aluno.

Necessitando de um elevado nível de conhecimento na resolução de problemas, muitos alunos são desmotivados por não possuírem um conhecimento prévio que o deixe familiarizado com a disciplina e desenvolva práticas e técnicas com facilidade.

As disciplinas introdutórias nos cursos de informática, denominadas algoritmos e programação possuem sintaxe complexa de difícil entendimento, requerendo que o conteúdo aplicado seja desenvolvido pelos educadores de forma diversificada, devido ao ritmo diferenciado retornado pelos alunos.

Não sendo possível acompanhar individualmente o desempenho desses alunos, devido ao grande número de alunos que iniciam os cursos, os educadores terão de traçar metas e estratégias que atinjam os alunos que possuem dificuldade de aprendizado, fazendo com que os mesmos criem um ritmo de aprendizagem equivalente, com relação aos outros alunos com conhecimento mais avançado.

Dificuldades muitas vezes apresentadas por esses alunos é a compreensão sobre o conteúdo aplicado, sendo que muitas vezes é aplicado conteúdo de codificação, fazendo com que o aluno sinta uma dificuldade maior, pelo mesmo não possuir uma concepção clara sobre o que realmente é solicitado para resolução de um determinado problema, identificados somente nessa fase.

Diante das dificuldades apresentadas pelos alunos, os educadores estão cada vez mais procurando novas estratégias para melhorar a qualidade de ensino e diminuindo assim a quantidade de desistência e insucessos nas disciplinas de programação.

Visando essa melhora, atualmente existem vários estudos nos quais o enfoque é a metodologia aplicada no ensino de linguagens de programação, objetivando atingir as expectativas dos alunos que possuem dificuldades no aprendizado. Dentre os vários estudos feitos, foram desenvolvidos diversos aplicativos de apoio à aprendizagem de programação, com o enfoque de minimizar as dificuldades desses alunos. Por meio dessas ferramentas o aluno desenvolve práticas que auxiliam na compreensão a resolução de problemas com mais facilidade. Porém os mesmos estudos revelam que essas ferramentas disponíveis suprem completamente as exigências da aprendizagem de programação (GOMES, 2008, p.97).

Vale destacar a existência de uma diversidade de linguagens que são aplicadas para o ensino de primeira linguagem, porém em uma grande maioria, essas linguagens possuem um conteúdo complicado e abstrato dificultando o entendimento e aprendizagem dos alunos novatos.

3. Compreendendo Algoritmos e Linguagem de Programação

Nesse capítulo será feita uma breve apresentação sobre a importância dos algoritmos, os tipos de representações. Será apresentada uma definição sobre linguagem de programação, abordando-se linguagem de baixo e alto nível.

3.1 Algoritmos

Um algoritmo é um procedimento computacional bem definido que toma algum valor ou conjunto de valores como **entrada** e produz algum conjunto de valores como **saída**. Portanto um algoritmo é uma seqüência de passos computacionais que transformam a entrada em saída (CORMEN, 2002, p.4).

Os algoritmos têm com características, por serem representados de forma *finita*, terminar em um tempo finito, não podendo ser ambíguos, ou seja, não dar margem a uma dupla interpretação. Um algoritmo tem que ser capaz de receber, gerar e fornecer informações.

3.1.1 Formas de Representações de Algoritmos

A representação de algoritmos pode ser dividida em três formas, na representação *Narrativa*, na representação *Gráfica* e na representação *Pseudocódigo* (linguagem estruturada). De acordo com os 3 tipos de representações apresentadas, a média aritmética será reproduzida em ambas as 3 formas de representação, mostrando que podemos resolver o mesmo problema de diferentes formas

Representação Narrativa: Nesse tipo de representação, o algoritmo é expresso na forma da linguagem natural, dessa forma tem-se o algoritmo representado em seqüência, onde cada linha representa uma ação a ser executada. Como exemplo, pode-se exemplificar a preparação de um bolo:

Resolução do Problema:

- Informe a média 1;
- Informe a média 2;
- Some a média 1 + a média 2;
- O resultado divida por 2;
- Se o resultado > 7 aluno aprovado;
- Senão aluno reprovado;

Figura 3-1 – Representação na forma narrativa para calcular a média aritmética.

Representação Gráfica: Já nessa representação, os algoritmos são representados com formas geométricas, cada forma representa uma ação específica a ser executada no desenvolver do algoritmo. Segue abaixo a tabela na qual representa os símbolos e suas ações:

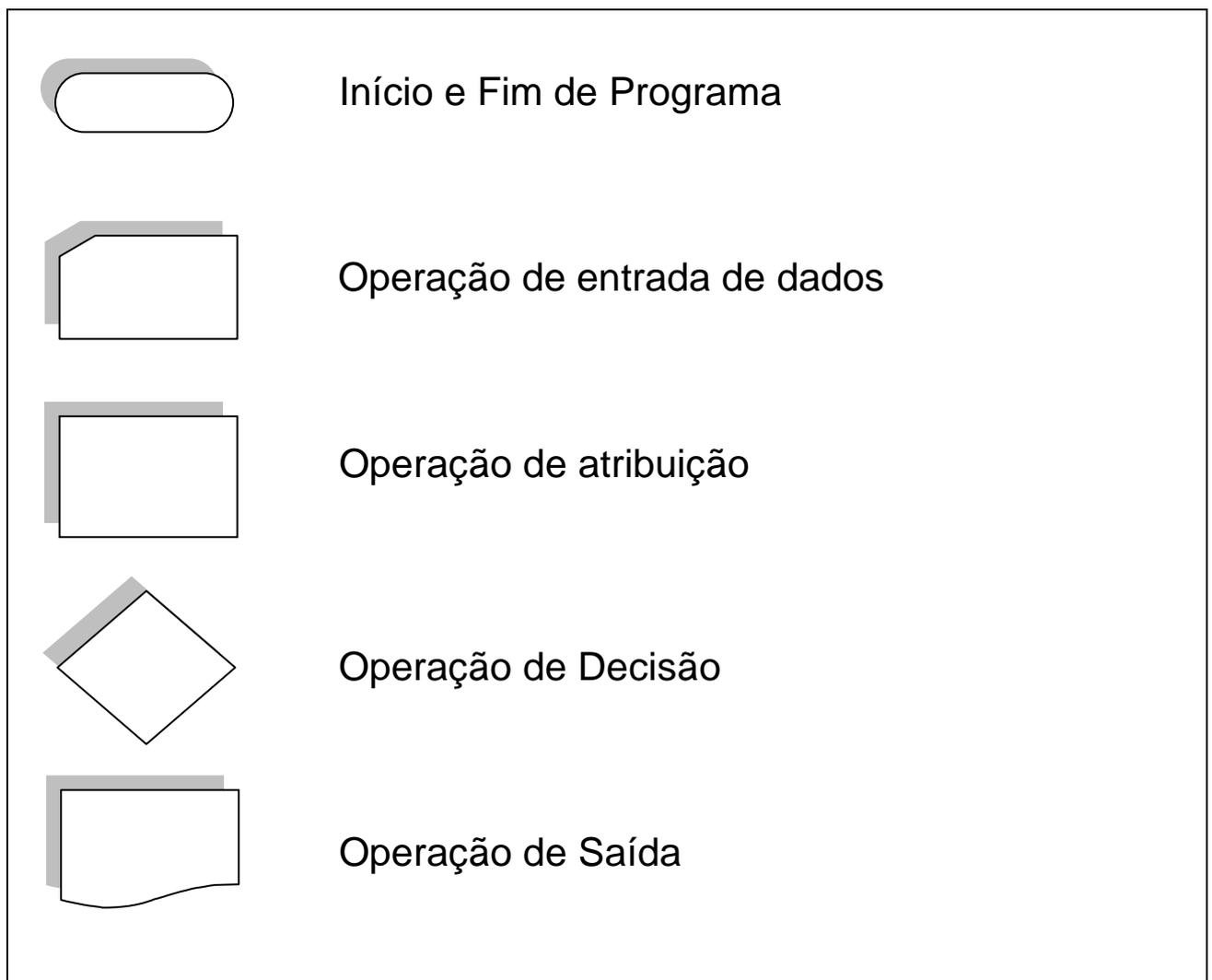
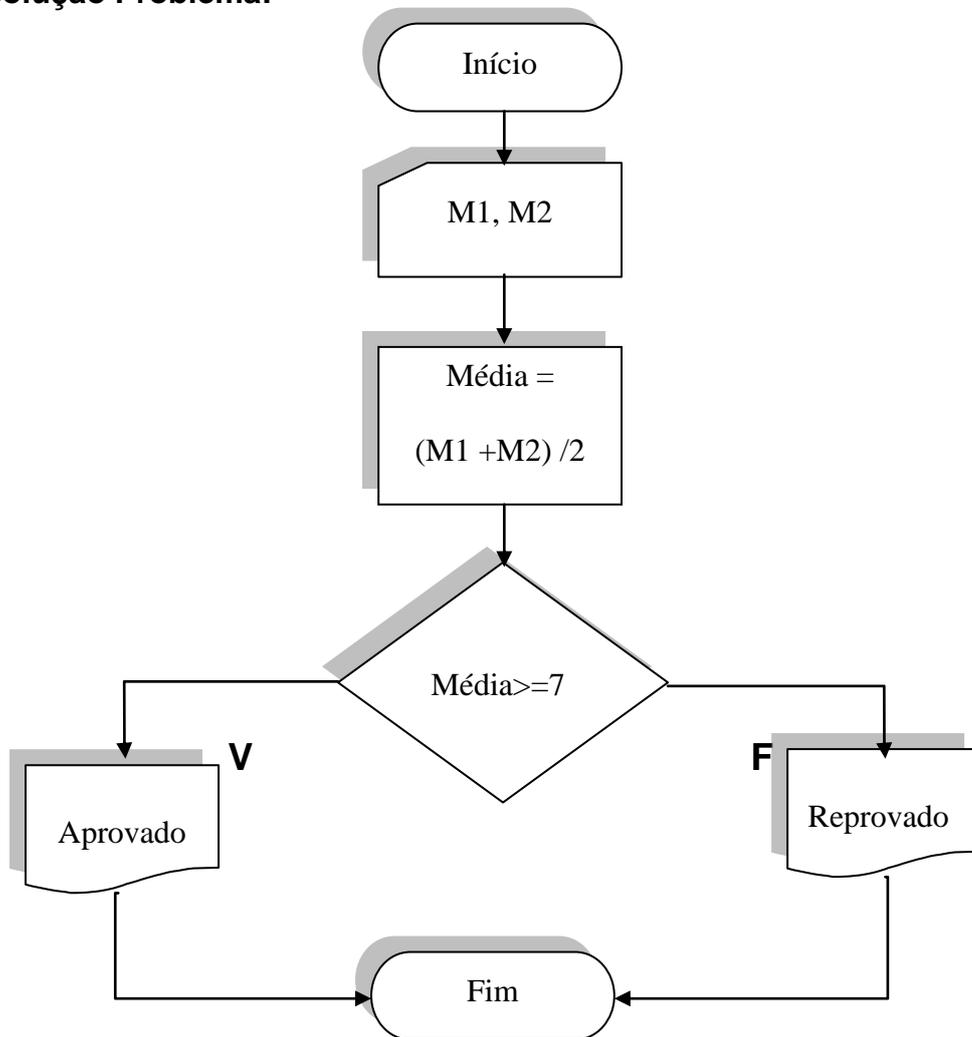


Figura 3-2 – Tipos de representação gráfica

Resolução Problema:**Figura 3-3 – Representação Gráfica**

Representação Pseudocódigo: Essa forma de representação são palavras em português que de acordo como informada, executa uma determinada ação. A forma que é escrita essa representação aproxima-se a forma na qual os programas são escritos em linguagem de máquina. Esse tipo de representação facilita o entendimento das linguagens de programação. Podem ser representados com algumas palavras:

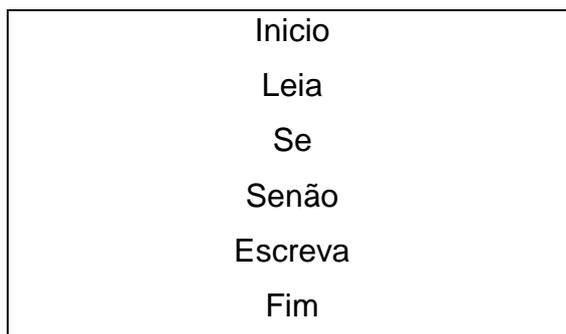


Figura 3-4 –Representação Pseudocódigo

Resolução Problema:

```
var real m1, m2, media
Inicio
    Leia (m1,m2);
    Media = (m1+m2)/2;
Se media >=7
    então escreva “Aprovado”
Senão escreve “ Reprovado”
Fim
```

3.2 Definições de linguagem de programação

Existem várias definições para linguagem de programação. Em informática, o termo “Linguagem de Programação” pode ser definido como um conjunto de símbolos e regras, utilizados em um processo de comunicação (Oliveira, 1999, p.7).

Já para Lucchesi, linguagem de programação é definida por um conjunto de símbolos, chamado *alfabeto*, que podem ser usados na representação de procedimentos, e por um conjunto de regras que especificam como compor estas representações e quais são ações associadas a estas representações. Uma

seqüência de símbolos de uma linguagem e programação que representa um ou mais procedimentos será chamada de *programa* (Lucchesi, 1979, p10). Existem vários tipos de linguagem de programação, duas propriamente ditas e claramente diferenciadas: as linguagens de baixo nível e as de alto nível.

3.2.1 Linguagem de Baixo Nível

Linguagem de baixo nível é totalmente dependente da máquina, é a linguagem de programação escrita em código de máquina não sendo possível que seu código seja migrado para outras máquinas. Constituída por números em base binária, ou seja, identificam somente Zero (0) e Um (1) de difícil entendimento. Esse tipo de linguagem é executada diretamente no computador, tendo um tempo de resposta rápido. Um exemplo de linguagem de baixo nível é o *Assembler*.

3.2.2 Linguagem de Alto Nível

Linguagem de alto nível é o tipo de linguagem que mais se aproxima da concepção humana, totalmente independente da arquitetura do computador, seu código é escrito em forma de algoritmos, podendo ser migrado de uma plataforma para outra. Já esse tipo de linguagem necessita de um tradutor, no qual interpreta a linguagem de alto nível para a linguagem de baixo nível (linguagem de máquina).

4. LINGUAGEM DE PRORGAMAÇÃO PYTHON

A finalidade desse capítulo é apresentar a linguagem Python, destacando sua importância, ambiente de desenvolvimento, vantagens e desvantagens.

4.1 Python

O Python é uma linguagem de programação de altíssimo nível que começou a ser desenvolvida no final dos anos 80 na Holanda, por Guido Van Rossum, cuja primeira versão foi publicada em 1991. Como qualquer linguagem, Python visa à qualidade de software, “o enfoque dá à legibilidade, coerência e qualidade de software em geral, o distingue das linguagens estilo [2]” Kitchen sink “*(Lutz, 2007, p.29). Python é uma linguagem não muito conhecida, possui características que a torna uma linguagem simples e eficiente, sendo fácil de manter, indicada para pessoas que estão aprendendo programar, seu código é reutilizável e possui programação orientada a objetos.

4.1.2 Vantagens

Por possuir suporte para programação Orientada a Objetos (POO), Python suporta polimorfismo, sobrecarga de operadores e herança múltipla, sendo de fácil aplicação desses métodos, como JAVA ou C++ (Lutz, 2007, p.36).

O autor ainda ressalta que Python é uma linguagem com código aberto, totalmente gratuito, não existindo restrições para utilização, sendo possível vender código fonte. Essa linguagem possui um número de linhas de código menor comparada a outras linguagens, fazendo que haja menos digitação e uma manutenção mais fácil.

Ressalta também a portabilidade da linguagem, multiplataforma permitindo que sua compilação possa ser executada em diferentes plataformas, não necessitando que seu código seja alterado (Lutz, 2007, p.30).

Essa linguagem de programação permite também o desenvolvimento de aplicações, podendo ser utilizada em vários tipos de projetos, em gerenciamentos de conteúdos,

documentos eletrônicos, servidores de aplicações, Telefonia Móvel (Lutz, 2007, p.30).

Python é utilizado como linguagem de Script, sendo possível que seja incorporado em seu código funcionalidades, utilizando mecanismos de integração, por exemplo: Python pode chamar bibliotecas C e C++, entre outros.

Lutz destaca também a tipagem dinâmica da linguagem, no qual suas variáveis são determinadas automaticamente, em tempo de execução e não em respostas às declarações presentes em seu código (Lutz, 2007, p.91).

O uso interativo do Python permite testar os comandos antes de executá-los em programas mais complexos.

Tudo isso fazendo com que o Python seja uma linguagem simples e objetiva, de fácil entendimento.

4.1.3 Desvantagens

Segundo Lutz, Python diferente das outras linguagens como, por exemplo, Java, compila (traduz) o código em formato intermediário conhecido como *codigo-byte* e depois interpretam esse *codigo-byte*, fazendo com que sua execução não seja tão rápida com relação às linguagens compiladas como C, C++ (Lutz, 2007, p. 32).

4.2 Interpretador

O interpretador é um tipo de programa que executa outros programas. Quando você escreve um programa em Python, o interpretador lê o seu programa e executa as instruções que nele contém. Na verdade o interpretador é uma camada de software lógico entre seu código e o hardware do computador em sua máquina (LUTZ, 2007, p.41).

4.2.1 Instalando o Interpretador

Será apresentado o passo a passo para instalação do interpretador Python para o Sistema Operacional (SO), Windows 7, nesse tipo de plataforma ele vem como um programa executável.

O interpretador Python pode ser adquirido através do site www.python.org na opção de downloads, selecionando assim a plataforma utilizada no estudo. Caso o programador for utilizar de outro SO selecione o arquivo de acordo com sua plataforma utilizada. O roteiro de instalação abaixo apresentado faz a instalação padrão do Interpretador Python.

Após a finalização do Download execute-o, em seguida selecione em quais usuários de sua maquina deseja que o interpretador Python seja instalado e selecione o botão "NEXT". A próxima seleção é de escolha do diretório a ser salvo o Python, selecione o diretório desejado e clique em "NEXT", em seguida será solicitado que selecione a maneira que os recursos do interpretador sejam instalados, clicando em seguida em "NEXT". Será iniciado assim o processo de instalação do Interpretado Python no SO Windows 7. Depois de completado o processo de instalação clique em "FINISH" para concluir a instalação do Interpretador.

Depois de instalado o Python localiza-se no Menu >> Todos os Programas >> Python. A figura abaixo mostra onde o Python aparece no SO utilizado.

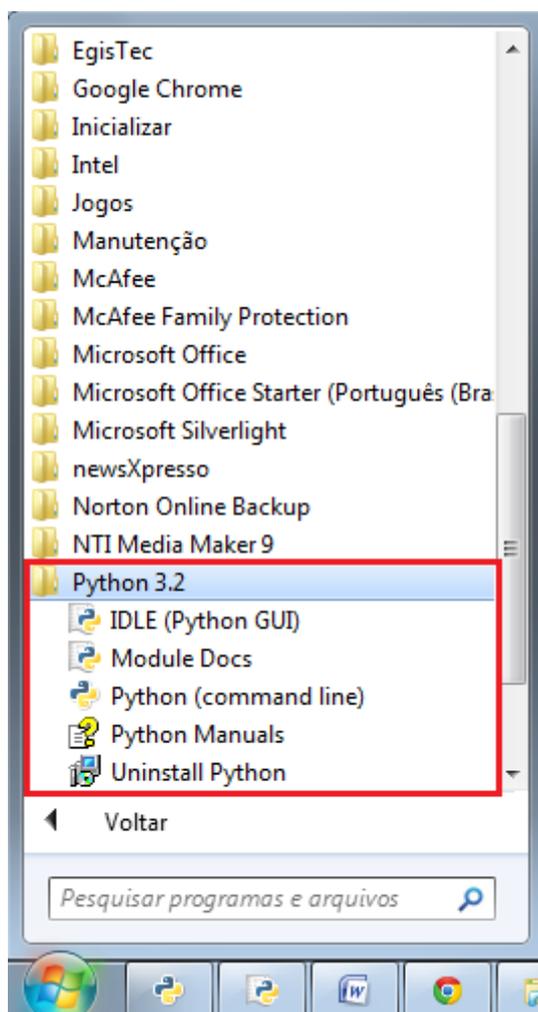


Figura 4-1- Python no Menu Iniciar do Windows 7.

O menu do Python tem cinco entradas, a IDLE interface com o usuário, ler a documentação de módulo, iniciar uma sessão interativa, ler os manuais padrões do Python em um navegador e desinstalar(LUTZ, 2007 – p. 516).

4.3 A interface gráfica com o usuário

O IDLE (Ambiente de desenvolvimento Integrado) é uma interface gráfica com o usuário (GUI- *Graphical User Interface*) para o desenvolvimento em Python, e é uma parte padrão e gratuita do sistema Python. Em resumo o IDLE é um a GUI que permite editar, executar, navegar e depurar programas em Python (Lutz, 2007, p 64).

No Windows 7 podemos acessá-lo **Iniciar >> Todos os Programas >> Python >> IDLE**.

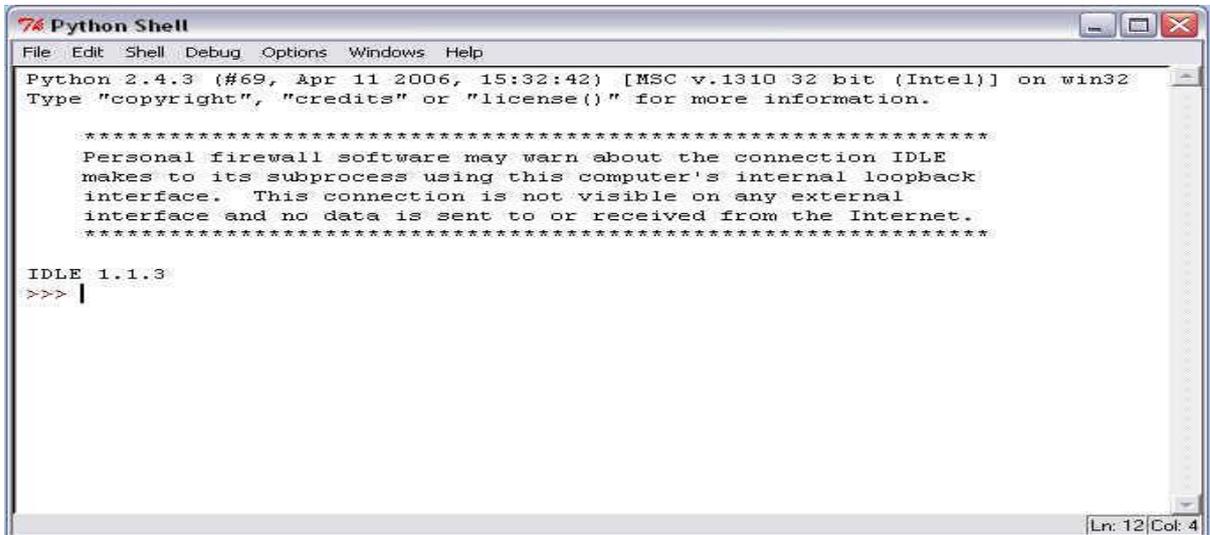


Figura 4-2 Janela Principal e Janela de edição de textos do IDLE.

Porém vem à seguinte pergunta quando fechado o interpretador, como fazer para recuperar o código? Dentro do interpretador clica-se em File/ New Window.

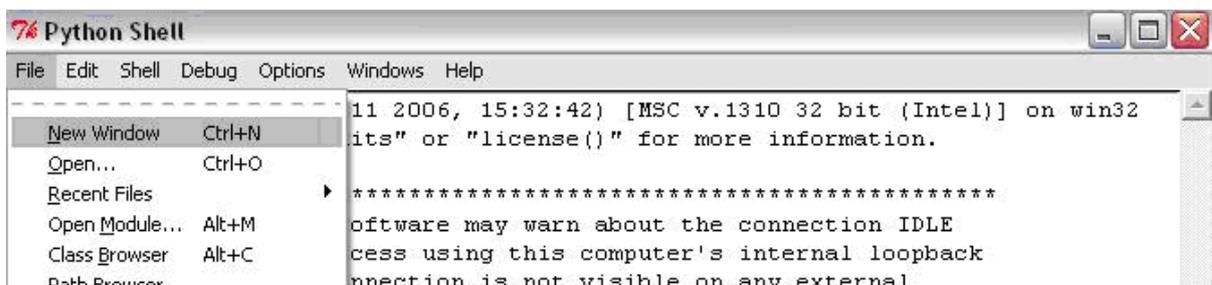


Figura 4-3 Janela para criar um arquivo Python.

Resultando na abertura da tela onde se digita o código Python, possibilitando que o código possa ser salvo para ser executado quando desejar.

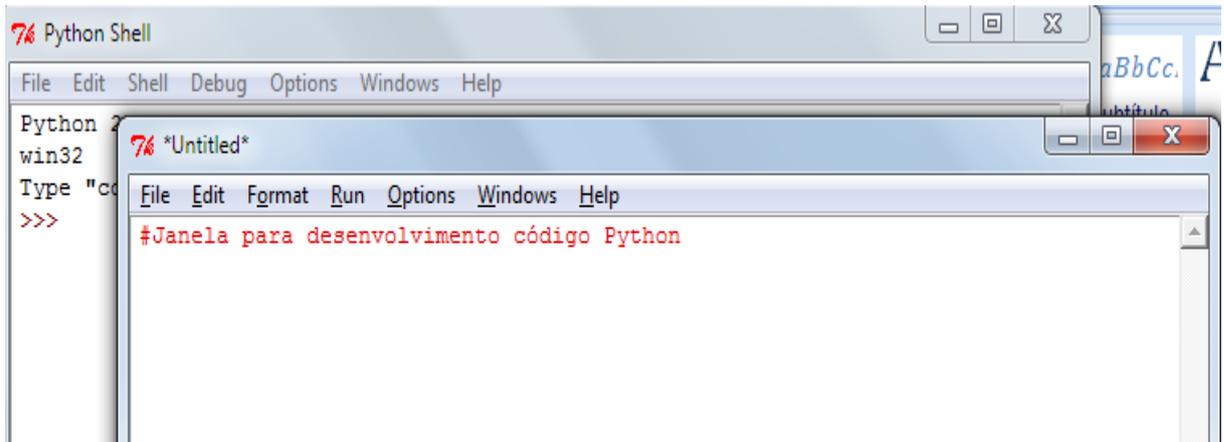


Figura 4-4 Janela para desenvolvimento código Python

Para executar o código existente dentro da IDLE Python, é necessário acessar Run/Run Module F5.

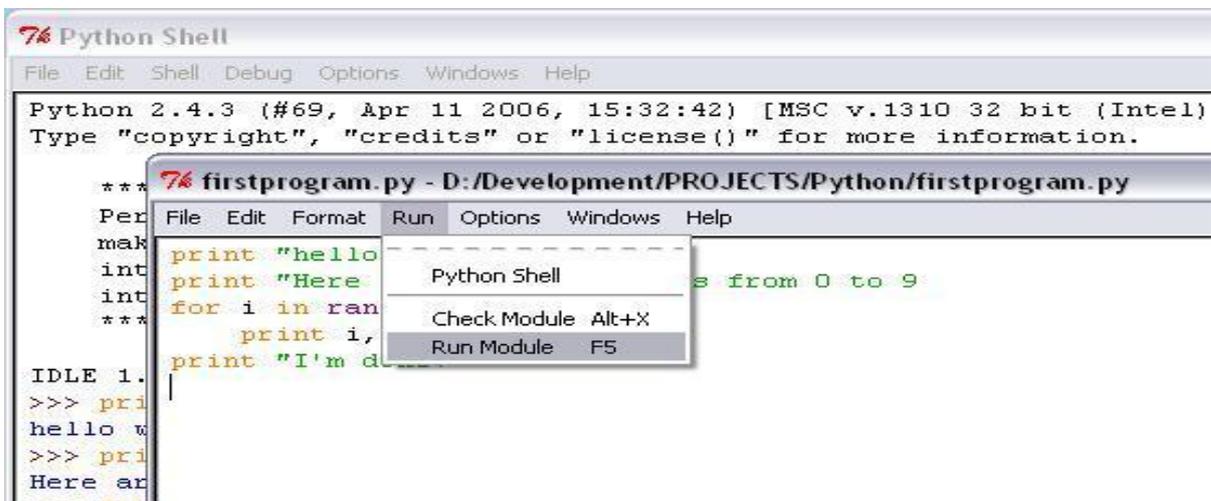


Figura 4-5 - Janela para executar seu código Python

4.4 Iniciando Programação Python

Para utilizarmos Python como primeira linguagem, será definida alguns princípios básicos e particularidades da linguagem. Para que possamos iniciar um estudo sobre a introdução da linguagem nos cursos de ciência da computação. As definições são de acordo com o estudo que será realizado e o material disponibilizado pela Professora Cláudia P. C. Maciel com o conteúdo aplicado no Módulo I do Centro Paula Souza de Assis.

4.4.1 Olá Mundo

Ao iniciar o estudo de linguagem de programação nos cursos de informática o primeiro programa a ser desenvolvida na linguagem estudada, geralmente a linguagem inicial é o C, C++, é o “HELLO WORD”, mas conhecido como “Olá mundo. Os alunos iniciam com simples linhas de comando para que execute a impressão na tela da frase “Olá Mundo”.

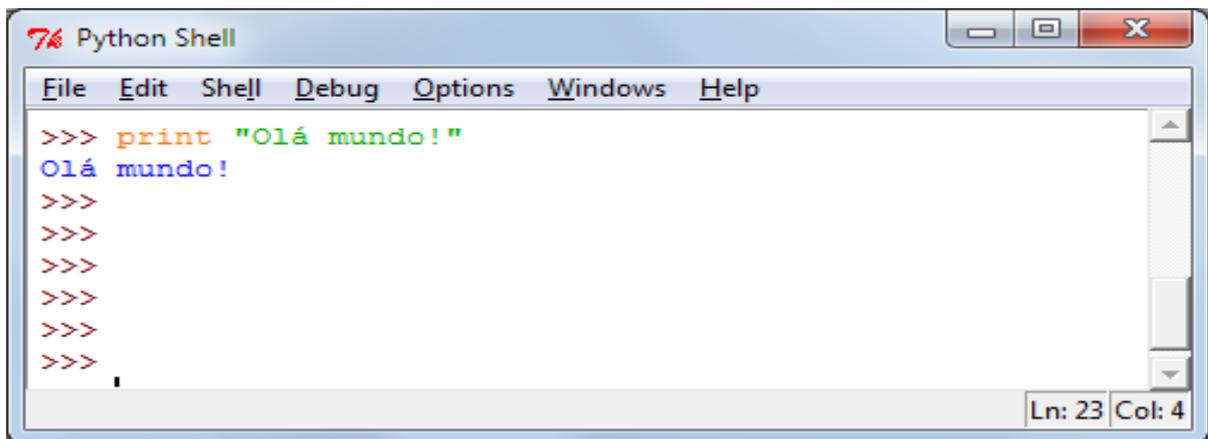
Em C++, a primeira linguagem utilizada na maioria dos cursos de informática seria desenvolvida da seguinte forma:

```
#include <iostream.h>
void main()
{
cout << “Olá Mundo. “ << endl ;
}
```

Podemos observar que para exibir o simples “Olá mundo” na linguagem C++, temos que expressar com vários comandos, que para quem não possui um conhecimento prévio sobre a linguagem ficaria completamente assustado, por já ter ouvido anteriormente que linguagem de programação é difícil de aprender, começando assim sua desmotivação pelo assunto.

Já em Python, podemos representar o mesmo código apenas com um parágrafo, no qual é simples e objetivo e de fácil entendimento para que esteja iniciando o trabalho com linguagem de programação. Em Python “Olá mundo” é expresso da seguinte forma:

```
print “Olá mundo. “
```

A screenshot of a Python Shell window. The title bar reads "Python Shell". The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Windows", and "Help". The main text area shows the following code and output:

```
>>> print "Olá mundo!"  
Olá mundo!  
>>>  
>>>  
>>>  
>>>  
>>>  
>>>  
>>>  
>>>
```

The status bar at the bottom right indicates "Ln: 23 Col: 4".

Figura 4-6- Representação “Olá Mundo” no interpretador Python

Dessa maneira, inicialmente podemos notar uma facilidade de programação em Python.

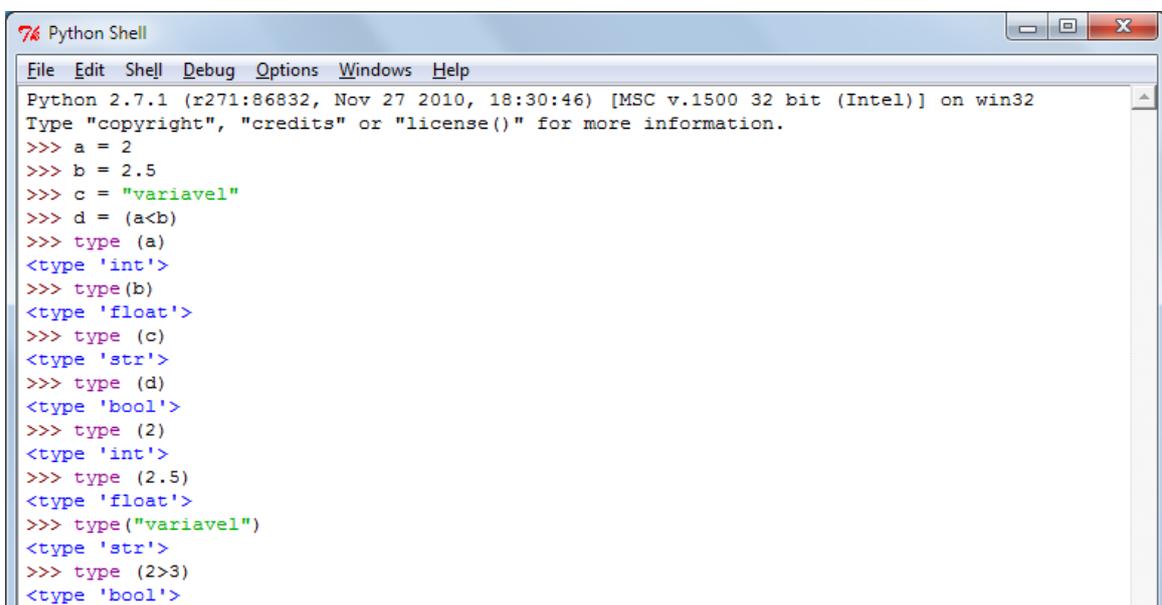
4.4.2 Variáveis

Variáveis podem ser definidas como itens que podem ser alterados. Variáveis referem-se a um valor que lhe é atribuído. Usar variáveis em Python é muito fácil. Em Python não somos obrigados a declarar uma variável, ou seja, não há necessidade de especificar o tipo da variável como é de costume fazer em outras linguagens orientadas a objeto. No Python o tipo da variável é o tipo do seu valor automaticamente sem precisar declarar. Segundo Lutz, “é utilizado o modelo de Tipagem Dinâmica, esse modelo faz com que os tipos das variáveis sejam

determinados automaticamente em tempo de execução, e não em resposta de declarações presentes em seu Código (LUTZ, 2007, p.91)”.

Em Python podemos identificar o tipo do valor ou da variável utilizada. Para que seja possível saber qual o tipo de um determinado valor utilizamos o comando `type(x)`, na qual retorna o tipo do objeto.

Na figura abaixo é representado como verificar o tipo de variável no interpretador Python.

A screenshot of a Python Shell window titled "Python Shell". The window has a menu bar with "File", "Edit", "Shell", "Debug", "Options", "Windows", and "Help". The main area shows the following text:

```
Python 2.7.1 (r271:86832, Nov 27 2010, 18:30:46) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> a = 2
>>> b = 2.5
>>> c = "variavel"
>>> d = (a<b)
>>> type(a)
<type 'int'>
>>> type(b)
<type 'float'>
>>> type(c)
<type 'str'>
>>> type(d)
<type 'bool'>
>>> type(2)
<type 'int'>
>>> type(2.5)
<type 'float'>
>>> type("variavel")
<type 'str'>
>>> type(2>3)
<type 'bool'>
```

Figura 4-7- Interpretador Python verificando os tipos de variáveis

Na representação acima declaramos o para as variáveis `a = 2`, `b = 2.5`, `c = "variável"` e para `d = (a<b)`, em seguida foi verificado através do comando "type" qual o tipo respectivo retornando para "a" o tipo int (inteiro), "b" o tipo float (real), "c" o tipo str (string) e para "d" o tipo bool (booleano).

Em Python pode-se também verificar o tipo do objeto, informando o respectivo no momento da verificação como no exemplo acima. O comando `type(2)` retorna o tipo int representando inteiro.

O nome das variáveis em Python tem suas particularidades, podem conter tanto letras quanto números, mas obrigatoriamente deve começar com letras. Não tem

restrições para letras maiúsculas e minúsculas, porém por conveniência não é utilizada. Caso vier a utilizar lembre-se, Variavel e variável são diferentes. Não se podem utilizar caracteres especiais como \$, !, #, @ entre outros, no qual o interpretador causara erro de sintaxe. Não pode utilizar palavras reservadas da linguagem para a declaração de variáveis.

Python possui 29 palavras reservadas, a imagem abaixo mostra as palavras reservadas da linguagem Python.

```
and      def      exec     if       not      return
assert   del      finally  import   or       try
break    elif     for      in       pass     while
class    else     from     is       print    yield
continue except    global  lambda   raise
```

Figura 4-8 - Palavras reservadas em Python

4.4.3 Operadores

Definiremos alguns tipos de operadores, no qual é essencial para iniciar o estudo de linguagem de programação.

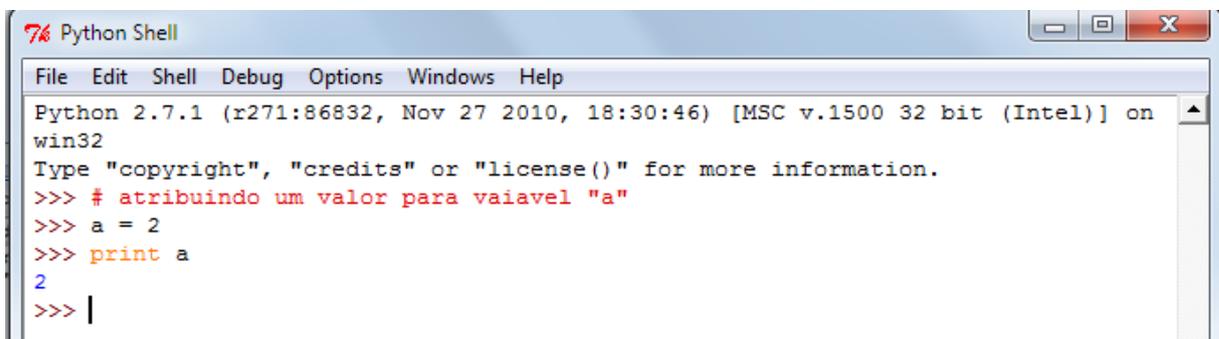
Os operadores são os símbolos ou palavras reservadas da linguagem que representam as expressões a serem executadas.

Em Python, não muito diferente de outras linguagens utilizamos os símbolos matemáticos para representação de uma expressão. A figura abaixo representa a os símbolos aritméticos utilizado em Python e seus respectivos significados.

Operador	Descrição
+	Soma
-	Subtração
*	Multiplicação
/	Divisão (real ou inteira)
//	Divisão inteira
**	Elevar a uma potência
%	Módulo (resto da divisão inteira)

Figura 4-9 – Operadores Aritméticos

Como na matemática o uso de parênteses para agrupar uma determinada expressão. Temos também como operadores, os operadores de atribuição, como já vestem em exemplos anteriores, o símbolo que representa a atribuição é o símbolo de igualdade, =, definindo uma variável e atribuindo um valor a ela um valor. Abaixo é representada a atribuição em Python.



```

Python Shell
File Edit Shell Debug Options Windows Help
Python 2.7.1 (r271:86832, Nov 27 2010, 18:30:46) [MSC v.1500 32 bit (Intel)] on
win32
Type "copyright", "credits" or "license()" for more information.
>>> # atribuindo um valor para variavel "a"
>>> a = 2
>>> print a
2
>>> |

```

Figura 4-10- Representação em Python para atribuição de um valor á uma variável.

Temos também, os operadores de comparação, serve para comparar dois números e determinar algumas expressões. [2]”Os operadores de comparação de Python são os mesmos usados em Java e C++”.

Operador	Descrição
<	Menor
>	Maior
<=	Menor ou igual
>=	Maior ou igual
==	Igual
!=	Diferente

Figura 4-11- Operadores de Comparação

4.4.4 Entrada E Saída de dados

Em linguagem de programação, ao desenvolver alguns tipos de programas, necessitamos que determinadas informações sejam fornecidas a partir de uma unidade de entrada. No mais comum no desenvolvimento de algoritmos essa entrada é atribuída através da instrução “**LEIA**”. Em Python utilizamos o comando **input**, no qual recupera a informação através de entrada para uma determinada variável.

A saída de dados na forma de algoritmos pseudocódigo é representada com a instrução “**IMPRIMA**”, no qual retorna o valor obtido através de uma entrada de dados ou de uma expressão. Representamos em Python com o comando **print** seguindo no nome da variável na qual deseja que seja retornado o seu valor. No interpretador Python representamos da seguinte forma:

```
>>> a = input ("Informe um valor para a: ")
Informe um valor para a: 10
>>> print a
10
>>>
```

Figura 4-12- Representação em Python de entrada e saída de dados.

4.4.5 Desvios

Desvio condicional, é uma maneira de possibilitar tomada de decisão durante a execução do algoritmo. A forma mais simples de tomada de decisão utilizada em pseudocódigo é o “se”. Em Python representamos as tomadas de decisão com as palavras reservadas:

- **IF:** em português conhecida como “se” no qual verifica a condição posta para é verdadeira.
- **ELSE:** já em português esse comando é conhecido como “senão” caso a condição seja feita essa instrução.
- **ELIF:** em Python o elif tem com funcionalidade fazer múltiplos testes, em português seria representado como :

Se <condição>
 <verdadeiro>

Senão

Se <condição>

Em Python o código em português acima em negrito representa o ELIF

if <condição>
 <verdadeiro>
elif <condição>

5 ESTUDO DE CASO

Nesse capítulo será detalhado o estudo de caso realizado no Curso de Técnico em Informática - Módulo I da Etec Pedro D'Arcádia Neto - Centro Paula Souza - Assis/SP.

5.1 Proposta

A proposta do estudo realizado conforme convite do Professor Domingos foi aplicar uma aula introdutória para os alunos do Técnico em Informática do Centro Paula Souza de Assis sobre Python como primeira língua. O estudo teve como finalidade identificar as dificuldades apresentadas por esses alunos, no qual possuem conhecimento apenas de Pseudocódigo ao se depararem pela primeira vez com uma linguagem de programação.

O estudo teve também como proposta levar conhecimento de uma linguagem de programação na qual possui um código simples e de fácil entendimento, na qual é apropriada para ser aplicada para alunos que iniciaram o estudo de linguagem de programação.

Teve como proposta também ressaltar a importância do entendimento e resolução de algoritmos, na qual é a principal base para o desenvolvimento de linguagem de programação.

Através do questionário aplicado para esses alunos, será feito o levantamento das dificuldades apresentadas e Python como primeira linguagem.

5.2 Conteúdo Aplicado

O conteúdo aplicado para os alunos é baseado em todo levantamento bibliográfico realizado para a conclusão dos capítulos anteriores e também no material disponibilizado pela Professora Claudia Patrícia Candia Maciel, na qual ministra matéria de Lógica de Programação. Inicialmente os alunos desenvolvem comandos

em português (Pseudocódigo), e utilizam da ferramenta Portugol 2.3, para testar os algoritmos realizados.

Com base nesses dados coletados, inicialmente foi introduzido um breve conteúdo sobre os tipos de algoritmos acima citados, e a importância do entendimento de algoritmos. Também conceituado para os alunos, os tipos de linguagens de Programação, Linguagem de baixo e auto-nível, no qual Python é pertencente à classificação de alto-nível.

De acordo com o estudo sobre a Linguagem Python, foram passadas para os alunos as motivações sobre o estudo dessa linguagem, o poder da linguagem e a simplicidade da mesma.

O estudo sobre a linguagem se iniciou na orientação dos alunos na instalação do Interpretador Python, disponibilizado no site oficial em www.python.org, baixando o executável para plataforma utilizada e orientando na instalação padrão do mesmo. Depois de instalado, apresentado o interpretador, a IDLE de interface com o usuário e os padrões de execução do código Python.

Olá Mundo, foi o primeiro contato com a linguagem, como em toda linguagem o primeiro código a ser desenvolvido. Dando seqüência na introdução da linguagem, foi abordado sobre as formas de utilização das variáveis, tendo em vista que no Python não há necessidade de declaração do tipo do mesmo, devido a sua tipagem dinâmica.

Os operadores da linguagem são apresentados, e suas respectivas funcionalidades, operadores aritméticos os mesmos utilizados na matemática, o operador de atribuição representado pelo símbolo de igualdade "=" e os operadores de comparação nos quais são fundamentais para as tomadas de decisão. Conforme me passado pela professora Cláudia, ela acabará de introduzir a lógica para tomada de decisão (se, senão), não houve ainda introdução sobre os Laços (para, enquanto, repita), aplicado dois exercícios no qual foi apresentada em Python a lógica para utilização desses laços.

5.3 Exercícios aplicados

Os exercícios aplicados foram baseados no nível de conhecimento dos alunos e nas listas de exercícios disponibilizada pela professora Claudia. A lista de exercícios é composta de exercícios que abrangem desde a verificação dos tipos de variáveis, até a utilização de laços conforme a seqüência estudada no decorrer desse projeto de pesquisa.

Os exercícios aplicados juntamente com suas resoluções foram:

1- Verifique o tipo das variáveis do seguintes dados:

A) 125

type 125

Saída em Python:

<type 'int'>

B) 3,25

type (3.25)

Saída em Python:

<type 'float'>

C) Python

type ('python')

Saída em Python:

<type 'str'>

D) (3<2)

type(3>2)

Saída em Python:

<type 'bol'>

2- Ler dois valores inteiros (A e B) e efetuar as operações de adição, subtração e multiplicação de A por B, apresentando ao final os três resultados obtidos.

```
a = input ("Informe um valor para A: ")
b = input ("Informe um valor para B: ")
c = a+b
d = (a-b)
e = a*b
```

```
print "A Soma de A + B é: ", c
print "A Subtração de A-B é: ", d
print "A Multiplicação de A*B: ", e
```

Saída em Python:

```
Informe um valor para A: 20
Informe um valor para B: 20
A Soma de A + B é: 40
A Subtração de A-B é: 0
A Multiplicação de A*B: 400
```

3- Faça um algoritmo para calcular o triplo de um número.

```
a = input ("Informe o numero que deseja calcular o triplo: ")
b = a * 3
print " O triplo de ", a, "é: ", b
```

Saída em Python:

```
Informe o numero que deseja calcular o triplo: 3
O triplo de 3 é: 27
```

4- Faça um algoritmo para calcular a metade da soma de 3 números.

```
a = input("Informe o número 1: ")
b = input("Informe o número 2: ")
c = input("Informe o número 3: ")
d = (a+b+c)/3
print "A soma dos três números dividido por 3 é: ", d
```

Saída em Python:

```
Informe o número 1: 1
Informe o número 2: 2
Informe o número 3: 3
A soma dos três números dividido por 3 é: 2
```

5- Faça um algoritmo que leia dois números, calcule e mostre um elevado ao outro.

```
a = input ("Informe uma valor para A:")
b = input ("Informe uma valor para B:")
c = a ** b
print "O valor de A elevado a B é: ", c
```

Saída em Python:

```
Informe uma valor para A:2
Informe uma valor para B:2
O valor de A elevado a B é: 4
```

6- Faça um algoritmo que leia um valor em metros e transforme-o para centímetros.

```
a = input ("Informe o valor em metros: ")
b = a/100.00
```

```
print "O valor de ",a, "em centímetros e: ", b
```

Saída em Python:

Informe o valor em metros: 10

O valor de 10 em centímetros e: 0.1

7- Faça um algoritmo que leia um valor em quilometro e transforme-o para metros. Mostre o resultado.

```
a= input("Informe o valor em quilometros: ")
```

```
b = a * 1000
```

```
print "O valor me metros é: ", b
```

Saída em Python:

Informe o valor em quilometros: 10

O valor me metros é: 10000

8- Calcular a área de um retângulo e de um circulo.

```
#Calcular a área de um retângulo
```

```
base = input("Informe o valor da Base: ")
```

```
altura =input ("Informe o valor da altura: ")
```

```
area = base * altura
```

```
print "A area do retangulo é: ", area
```

Saída em Python 1:

Informe o valor da Base: 10

Informe o valor da altura: 20

A area do retangulo é: 200

```
#Programa para calcular a area de um circulo
```

```
raio = input ("Informe o valor do Raio: ")
area = 3,141526 * raio * raio
print "A area do circulo é: ", area
```

Saída em Python 2:

```
Informe o valor do Raio: 25
A area do circulo é: (3, 88453750)
```

9- Calcular os três primeiros múltiplos de um numero.

```
a = int(input ("Informe um número: "))
print "O primeiro multiplo de a é :", (a * 1)
print "O segundo multiplo de a é :", (a * 2)
print "O terceiro multiplo de a é :", (a * 3)
```

Saída em Python:

```
Informe um número: 3
O primeiro multiplo de a é : 3
O segundo multiplo de a é : 6
O terceiro multiplo de a é : 9
```

10- Escreve um programa que calcule o quadrado e o cubo de um número.

```
a = input ("Informe o numero: ")
b = a**2
c = a**3
print "O numero ",a,"elevado ao cubo é: ",c
print "O numero ",a,"elevado ao quadrado é: ",b
```

Saída em Python:

```
Informe o numero: 2
O numero 2 elevado ao cubo é: 8
```

O numero 2 elevado ao quadrado é: 4

11- Faça um algoritmo que leia 3 números e verifique qual o maior dentre eles.

```
a = input ("Informe um valor para A: ")
b = input ("Informe um valor para B: ")
c = input ("Informe um valor para C: ")
if (a > b):
    if(a>c):
        print " O maior valor é: ", a
    else:
        print " O maior valor é: ", c
else:
    if(b>c):
        print " O maior valor é: ", b
    else:
        print " O maior valor é: ", c
```

Saída em Python 1:

```
Informe um valor para A: 1
Informe um valor para B: 2
Informe um valor para C: 3
O maior valor é: 3
```

Saída em Python 2:

```
Informe um valor para A: 3
Informe um valor para B: 2
Informe um valor para C: 1
O maior valor é: 3
```

Saída em Python 3:

Informe um valor para A: 1

Informe um valor para B: 3

Informe um valor para C: 2

O maior valor é: 3

12- Faça um algoritmo que leia duas médias e calcule a média aritmética.

```
a = input('Digite a Nota 1: ')
b = input('Digite a Nota 2: ')
c = (a+b)/2
```

```
if (c >= 7):
```

```
    print "A média do aluno é: " , c
```

```
    print "Aluno aprovado"
```

```
else:
```

```
    print "A média do aluno é: " , c
```

```
    print "Aluno Reprovado"
```

Saída em Python 1:

Digite a Nota 1: 10

Digite a Nota 2: 10

A média do aluno é:10

Aluno aprovado

Saída em Python 2:

 Digite a Nota 1: 5

 Digite a Nota 2: 5

 A média do aluno é: 5

Aluno Reprovado

13- Faça um algoritmo que leia um determinado valor, e calcule a seqüência de Fibonacci até atingir o valor lido.

```
a = input ("Informe o valor maximo: ")
```

```
ant = 0
prox = 1W
atual = 0
while prox < q:
    print prox
    ant = atual
    atual = prox
prox = ant + atual
```

Saída em Python:

Informe o valor maximo: 20

1 1 2 3 5 8 13

5.1 Questionário

O questionário tem como finalidade de levantar os principais motivos desistência nos cursos de computação, e também de uma avaliação sobre o primeiro contato com Python.

1- Qual sua dificuldade no aprendizado de algoritmos?

- Falta de Interesse.**
- Falta de conhecimento.**
- Conteúdo Abstrato**
- Difícil entendimento.**
- Não tenho dificuldades.**

2 - Após a conclusão do Curso de Técnico em informática, você pretende cursar uma graduação na área? Informe o motivo.

- Sim.** **Não.**
-
-

3 – Você acha o conteúdo de algoritmos fácil?

Sim. Não Um pouco Outros. _____

4 - O ensino da linguagem Python é próximo ao desenvolvido em Portugal (Pseudocódigo)?

Sim. Não. Talvez.

5 - O que achou sobre a linguagem de Programação Python?

Fácil Médio Difícil Muito Difícil.

6 – Qual sua opinião sobre a linguagem abordada?

5.2 Conclusão sobre o trabalho realizado

Conforme a proposta abordada, foi aplicado para a Técnico em Informática do Centro Paula Souza de Assis uma introdução sobre a linguagem Python, visando a viabilidade da introdução da linguagem como primeira língua nos cursos relacionados a informática.

Através do estudo realizado podemos identificar que metade dos alunos que participaram do estudo realizado encontram dificuldades no aprendizado de algoritmos, sendo que mais da metade acha que o conteúdo aplicado na matéria de algoritmos possui um conteúdo abstrato, dificultando o entendimento. A maioria dos alunos se identificaram com a linguagem, pois acham que a linguagem é muito próxima ao aplicado na matéria de introdução a computação, na qual eles validam os algoritmos realizados em sala de aula no tipo de representação Pseudocódigo, na ferramenta Portugal. A grande maioria deseja cursar uma faculdade na área da informática. Dê acordo com o apresentado a esses alunos, sobre a linguagem

apresentada os alunos opinaram de acordo com o conteúdo aplicado sobre o mesmo, obtivemos as seguintes repostas.

- Foi uma introdução a matéria;
- O Python, eu gostei muito, pois me ajudou a entender mais os algoritmos;
- Esse programa foi bem legal e interessante do que fazer algoritmo no Portugol;
- Eu gostei porque é mais fácil do que o Portugol;
- Eu achei Python muito mais fácil de ver, eu tenho muita dificuldade no Portugol, e nesse Python aprendi com um pouco mais de facilidade;
- O conteúdo aplicado em aula relacionado a Python é bem interessante e curioso;
- Muito bom, aprendi um pouco mais;
- Gostei pela eficiência e facilidade no desenvolvimento;
- O Python é uma linguagem boa para aprender;
- Foi detalhado, bem explicado, consegui entender o conteúdo o que instrutora quis mostrar;
- O conteúdo relacionado a Python foi uma experiência muito boa e prática;
- Python não sendo tão difícil facilitou a entender o conteúdo. E ainda mais com um ótimo ensino breve;
- Eu achei bem legal e compreensivo e fácil entendimento, apesar da linguagem técnica ser um pouco diferente da usual;
- Eu achei muito interessante, pois não conhecia a linguagem.
- A linguagem é de fácil entendimento para iniciantes, quanto para aqueles que tem grande conhecimento da linguagem;
- Por Python ser mais simples do que o Portugol, o conteúdo apresentado foi de fácil entendimento;
- Conteúdo básico sobre como escrever algoritmos básicos com operações aritméticas e composições lógicas;
- O conteúdo é simples, entendimento de fácil aplicação;
- Uma linguagem fácil e eficiente;
- Ele é mais fácil de manusear, mais fácil que o Portugol. É particularmente prefiro o Python por causa da facilidade de uso;

- Gostei muito mais do que o Portugol, pois em apenas duas aulas consegui compreender o Python;
- Muito interessante esta linguagem é bem mais fácil se comparada às outras;
- Foi uma aula muito interessante, na qual podemos perceber que o Python é muito fácil, prático e simples de uso;
- Gostei muito, pois podemos uma idéia do que é uma faculdade voltada para a área de informática;

Podemos assim concluir através do estudo realizado que a linguagem Python é viável a ser introduzida como primeira linguagem, pois a mesma apresenta uma simplicidade na qual facilita o entendimento de lógica de programação para o desenvolvimento de programas.

6 - CONCLUSÕES

Diante do estudo feito, pode-se concluir com o material coletado da internet como livros em geral e através do estudo de caso realizado, que nos dias atuais muitos alunos que ingressam os cursos onde de Ciência da Computação e áreas da informática em geral, no qual tem como linguagem de programação a principal disciplina, são desmotivados pelos métodos de ensino no qual são aplicado tal matéria. Os mesmos apresentam dificuldades que se não identificadas ao início do curso, levam ao insucesso da conclusão do curso, registrados nos elevados números de desistência.

A proposta do projeto foi identificar as principais dificuldades apresentadas por esses alunos, visando à melhoria na qualidade de ensino e possibilitando que haja uma demanda maior de alunos concluindo os cursos na área de informática.

Através do estudo realizado, podemos por meio da linguagem de programação Python identificar que a mesma é apropriada a ser introduzida na grade curricular dos cursos que possuem linguagem de programação como matéria, tendo em vista a facilidade do seu conteúdo, possuindo uma sintaxe simples de fácil entendimento fazendo assim que diminua o numero de desistência por partes dos alunos que iniciam esses cursos e não possuem um conhecimento prévio sobre o conteúdo aplicado.

Com o estudo do Python, podemos identificar também que a linguagem supriu a proposta abordada, sendo bem recebida pelos alunos e desenvolvida baseando-se na estrutura de aprendizado abordada pela professora Claudia, responsável pelo conteúdo no curso em questão.

A linguagem foi aplicada para alunos que não tiveram nenhum contato com linguagem de programação, tendo apenas como único conhecimento o desenvolvimento de algoritmos desenvolvidos em aula. De acordo com a coleta do material sobre a linguagem e a abordagem da linguagem para esses tipos de alunos, pode ser identificado que os mesmos conseguiram realizar os exercícios propostos apenas com a apresentação da estrutura da linguagem e conhecimentos básicos para o início do desenvolvimento em Python.

Sendo assim, podemos concluir que de acordo com o estudo feito durante o ano, a linguagem possui estrutura para ser aplicada como língua inicial para os cursos de Ciência da Computação e afins, confirmando assim a viabilidade da mesma como língua inicial na matéria de introdução a programação.

REFERENCIAS

[1] NACIONAL Jomal; **Problemas na educação causam falta de profissionais no setor tecnológico.** Disponível em:

<<http://video.globo.com/Videos/Player/Noticias/0,,GIM1488216-7823-PROBLEMAS+NA+EDUCACAO+CAUSAM+FALTA+DE+PROFISSIONAIS+NO+SE TOR+TECNOLOGICO,00.html>> Acesso em: Agosto de 2011.

[2] *N. de RT:Linguagens em que a operação é mais importante do que o efeito obtido.

CORMEN, Thomaz H.; LIESERON, Charles E.; RIVEST, Ronald L.; STEIN, Clifford; Algoritmo Teoria e Pratica - Rio de Janeiro,2002

BRASIL, Python; **Guido Van Rossum** Disponível em: <<http://www.python.org.br/wiki/GuidoVanRossum>> Acesso em: Agosto de 2011.

GOMES, Anabela; HENRIQUES, Joanas; MENDES, Antônio José. **Uma proposta para ajudar alunos com dificuldades na aprendizagem inicial de programação de Computadores.** Disponível <<http://eft.educom.pt/index.php/eft/article/viewFile/23/16>> Acesso em: Fevereiro 20 de 2011.

MENDES, António José Nunes. **Software educativo para apoio à aprendizagem de programação.** Disponível: <http://www.c5.cl/ieinvestiga/actas/tise01/pags/charlas/charla_mendes.htm> Acesso em: 23 Fevereiro 2011.

JESUS, Andréia de; BRITO Gláucia Silvia. **Concepção de Ensino-Aprendizagem de Algoritmos e Programação de Computadores: A Prática Docente.** Disponível

em: <<http://200.201.81.50/enined/anais/enined/A15.pdf>> Acesso em: 23 de Fevereiro 2011.

RAABE, Andre Luis Alice; Silva, Julia Marques Carvalho da. **Um Ambiente para Atendimento as Dificuldades de Aprendizagem de Algoritmos** . Disponível em: <<http://www.unisinos.br/diversos/congresso/sbc2005/dados/anais/pdf/arg0027.pdf>> Acesso em:23 de Fevereiro 2011.

FERNANDES, Claudia Santos; MENEZES, Paulo Blauth. **Metodologia de ensino de Ciência da Computação: uma proposta para crianças**. Disponível em:<<http://pt.scribd.com/doc/5398022/Metodologia-do-Ensino-de-Ciencia-da-Computacao-uma-proposta-para-criancas>> Acesso em: 03 de Março 2011.

LUIZ, Mark;. **Aprendendo Python**. 2ed- Porto Alegre: Bookman, 2007.

ZELLE, John M. **Python as a First Language**. Disponível em: <<http://mcsp.wartburg.edu/zelle/python/python-first.html>> Acesso em 07 de Março 2011.

ESTEVES, Micaela; ANTUNS,Ricardo; MORGADO, Leonel; MARTINS, Paulo; FONSECA, Benjamim. **CONTEXTUALIZAÇÃO DA APRENDIZAGEM DA PROGRAMAÇÃO: ESTUDO EXPLORATÓRIO NO SECOND LIFE®**. Disponível em: <http://www.iadis.net/dl/final_uploads/200713C033.pdf> Acesso em 14 Março 2011.

Norvig, Peter; **Google** Disponível em: <<http://www.python.org/about/quotes/>> Acesso em: 7 de Abril 2011.

LUIZ, Mark;. **Aprendendo Python**. 2ed- Porto Alegre: Bookman, 2007. Pag. 29.

OLIVEIRA, Álvaro Borges de; **Introdução à Programação de Algoritmos** Florianópolis 1999.

LUCCHESI, Claudio L.; SIMON, Imre; SIMON, ISTVAN; SIMON, Janos; KAWALTOWSKI, Tomasz; **ASPECTOS TEÓRICOS DA COMPUTAÇÃO** Rio de Janeiro 1979.