USO DA PROGRAMAÇÃO SCILAB NO ENSINO DE FÍSICA: MODELANDO FENÔMENOS DE CINEMÁTICA

JOSÉ RICARDO DOS SANTOS FREITAS

Marabá-Pará
2019

USO DA PROGRAMAÇÃO SCILAB NO ENSINO DE FÍSICA: MODELANDO FENÔMENOS DE CINEMÁTICA

JOSÉ RICARDO DOS SANTOS FREITAS

Orientador: Dr. Narciso das Neves Soares.

Dissertação de Mestrado submetida ao Programa de Pós-Graduação no Curso de Mestrado Profissional de Ensino de Física (MNPEF), como parte dos requisitos necessários à obtenção do título de Mestre em Ensino de Física.

Aprovada por:

_____
Prof. Dr. Narciso das Neves Soares. (Orientador)

_____
Prof. Dr. Luiz Moreira Gomes (Membro Interno)

_____
Prof. Dr. Francisco Ferreira de Sousa (Membro Externo - UFPA)

Marabá-Pará
2019

Ficha Catalográfica

Dados Internacionais de Catalogação-na-Publicação (CIP)
Biblioteca II da UNIFESSPA. Marabá, PA

Freitas, José Ricardo Santos
Uso da programação em SCILAB no ensino de física / José Ricardo Santos Freitas ; orientador, Narciso Soares. — Marabá : [s. n.], 2019.

Dissertação (Mestrado) – Universidade Federal do Sul e Sudeste do Pará, Instituto de Ciências Exatas - ICE, Mestrado Nacional em Ensino de Física - MNPEF, Marabá, 2019.

1. Cinemática. 2. Modelagem. 3. Física – Estudo e ensino. 4 Scilab (Programa de computador). I. Soares, Narciso, orient. II. Universidade Federal do Sul e Sudeste do Pará. III. Título.

CDD: 22. ed.: 531.112

Elaborado por Nádia Lopes Serrão CRB2/575

Os **SÁBIOS** perdoam, mas nem sempre **TOLERAM**.

Pedro Cruz

DEDICATÓRIA

Dedico este trabalho a meu
irmão Elken Freitas (*in memoriam)*.

AGRADECIMENTOS

Agradeço antes de tudo ao Senhor meu Deus que me concedeu o dom da vida, sempre está comigo me guiando e me ajudando a tomar as melhores decisões, me tratando com misericórdia em todos os momentos difíceis.

Agradeço imensamente à minha família, por sempre me incentivarem e não terem medido esforços para me ajudar em todos os percalços ao longo desta e de outras conquistas. Agradeço a meu pai José Freitas pelo o apoio e a minha mãe Raimundinha Santos pelo suporte e orações constantes e a um pedaço de cada um de minha família chamado Ezequiel Freitas eu agradeço por ser o motivo para eu querer construir cada vez mais. Tenho uma dívida de gratidão e especial a meu irmão Elken Freitas, com quem pude contar em todas as nossas realizações e hoje sei que ele está sendo honrado através dessa conquista que ele tem ativa participação.

Agradeço a o Mestrado Nacional Profissional em Ensino de Física, promovido pela Sociedade Brasileira de Física (SBF), polo Universidade Federal do Sul e Sudeste do Pará (UNIFESSPA), a CAPES pelo apoio. Agradeço a todo corpo Docente do MNPEF da UNIFSSPA, por retransmitirem seu conhecimento e orientação a todos os Mestrandos do programa.

Agradeço a meu Orientador e Prof. Dr. Narciso das Neves Soares, pela parceria de trabalho e pelo suporte técnico no tocante a confecção do presente trabalho, que sempre se deu com competência e seriedade almejando sinceramente o êxito do seu orientando. Não poderia deixar de fazer um agradecimento especial a todos os meus colegas de turma do MNPEF 2017, onde passamos muitos momentos alegres e outros tensos.

Por derradeiro, mas não menos importantes gostaria de agradecer a todos os meus amigos pelas palavras de incentivos mutuamente, entre os quais vou citar em especial meu grande amigo Pedro Cruz, que tanto me ensinou sobre programação e do Software MatLAb, o que viabilizou a construção do presente trabalho. Também tenho a satisfação de citar meus amigos Bento Cleiton e Matias Formachare, colegas de graduação que até hoje têm a paciência de me ajudar a desenrolar minhas questões, amigos valiosos. Gostaria de agradecer ainda a meu amigo Juninho por uma dica essencial para a construção do trabalho.

USO DA PROGRAMAÇÃO SCILAB NO ENSINO DE FÍSICA: MODELANDO FENÔMENOS DE CINEMÁTICA

JOSÉ RICARDO SANTOS FREITAS

RESUMO

O presente trabalho traz a abordagem do tema Cinemática, tratada no 1º ano do Ensino Médio, através da utilização de um produto educacional baseado no emprego de programação computacional na modelagem numérica e gráfica de fenômenos relacionados ao tema. O software de programação utilizado no trabalho foi o SciLab, escolhido por ser livre, de código aberto, e ter versões para Linux, Windows e Mac OS. Na composição do trabalho o principal teórico levado em consideração foi Seymour Papert com a Teoria do Construcionismo, que defende a construção do conhecimento por quem aprende, ou seja, o discente tornasse o construtor do próprio conhecimento de maneira prática e através do computador. Para verificação da eficácia do produto desenvolvido ministramos um minicurso para alunos de 1° ano do ensino Médio de uma escola pública no município de Marabá no Pará, realizado no laboratório de informática do Instituto de Ciências Exatas da Universidade Federal do Sul e Sudeste do Pará (UNIFESSPA). O minicurso foi ministrado em etapas de execução através de um roteiro onde mostramos comandos fundamentais no software e passamos a modelar os tipos de movimentos estudados na Cinemática. O modelo de metodológico do trabalho foi de abordagem qualitativa, e a avaliação realizada levou em consideração a satisfação e o interesse dos alunos no assunto trabalhado sob a utilização da programação. Os resultados foram satisfatórios e o produto mostrou-se um bom recurso didático para ensino de Física.

PALAVRAS-CHAVE: Cinemática, Modelagem, Ensino de Física, SciLab.

ABSTRACT

The present work brings the approach of the kinematic theme in the 1st year of high school through the use of an educational product based on the use of computational programming in the numerical and graphical modeling of phenomena related to the theme. The programming software used in the work was SciLab. We chose SciLab because it is free (open source) software, there are versions for Linux, Windows and Mac OS. In the composition of the work the main theoretical taken into consideration was Sgmon Papert with the Theory of Constructionism, that theoretician defends the construction of knowledge by those who learn, that is, the student would make the builder of his own knowledge in a practical way and through the computer. To verify the efficacy of the product developed, we taught a mini-course with students of the 1st year of high school in a public school in the municipality of Marabá, Pará. It was taught in the computer lab of the Institute of Exact Sciences of the Federal University of South and Southeast of Pará (UNIFESSPA). The mini course was taught in execution stages through a script where we showed fundamental commands in the software and started to model the types of movements studied in Kinematics. The methodological model of the work was a qualitative approach, and the evaluation carried out took into account the satisfaction and interest of the students in the subject worked using the programming. The results were highly satisfactory and the product is a good didactic resource for physics teaching.

KEYWORDS: Kinematics , Modeling, Physics Teaching, SciLab

SUMÁRIO

LISTA DE FIGURAS

CAPÍTULO 1

1.1 Introdução

A Física é uma importante disciplina na formação do Ensino Médio. Ela consente distinguir as leis gerais da Natureza que gerem o desenvolvimento das ações que se podem medir, quantificar ou apreciar no universo macroscópico e microscópico. A Física se concentra em esclarecer as leis que regem o universo baseando-se em argumentos matemáticos.

Segundo Silva, Prates e Ribeiro (2016), no ensino de Ciências da Natureza e Física há o desafio de formar o aluno de maneira que o mesmo possa interagir com diversas situações e ambientes diferentes, reconhecendo e associando conhecimentos científicos. Partindo desse princípio os profissionais do ensino de Física tem a importante missão de encontrar meios inovadores para estarem constantemente renovando o conteúdo e dinamizando as aulas dos diversos temas possíveis. Neste contexto, em Brasil (2006) menciona que:

> A Física deve apresentar-se, portanto, como um conjunto de competências específicas que permitam perceber e lidar com os fenômenos naturais e tecnológicos, presentes tanto no cotidiano mais imediato quanto na compreensão do universo distante, a partir de princípios, leis e modelos por ela construídos.
> (BRASIL, 2006. p.02)

O presente trabalho traz a implementação de um Produto Educacional no ensino de Física. Ele consiste no ensino de Física através da utilização de um software de programação computacional, o SciLab. Esse é um software numérico e gráfico bastante simples de ser utilizado se comparado com linguagens mais clássicas como Fortran ou C. O Produto consiste, inicialmente, em ensinar aos alunos, sob orientação do professor, comandos básicos de entrada e saída de dados em um script ou no executor de tarefas do software. Após essa etapa o aluno aprenderá a utilizar esses comandos na construção de programas que modelem fenômenos de Física observados no estudo do campo da Cinemática. Esses programas são construídos passo a passo por eles e sua aplicação será analisada através da verificação dos possíveis valores obtidos com a modelagem que fizeram. Após aprendizagem do método de programação para modelar fenômenos do campo da Cinemática que foram utilizados como exemplos, os alunos são

instigados a buscar equações dentro do mesmo conteúdo para que eles possam construir o conhecimento através de scripts executados no software criado por eles no editor do SciLab.

1.2 **Justificativa**

Uma das funções do professor é mediar a captação de informações do conteúdo pelo aluno. Para isso ele pode usar os recursos disponíveis ou criar meios adequados que facilitem esse trabalho. Fazer uso das tecnologias buscando facilitar o entendimento pode ser uma estratégia didático pedagógica que venha desenvolver no aluno senso crítico, uma vez que eles realizam reflexões sobre os conteúdos abordados (Oliveira e Moura, 2015).

Baseando na missão da escola, que é formar e capacitar os alunos, segundo os Parâmetros Curriculares Nacionais (PCN) de 1998, deve-se promover ações inovadoras que possibilitem e destaquem o uso da cidadania, fazendo com que os alunos através dessas atitudes se integrem no processo de transformação e construção da sociedade que faz parte, aceitando os fatos e promovendo mudanças em um processo harmonioso. Há, para tanto, a necessidade de que os professores busquem meios de incorporar as novas tecnologias à sala de aula. Essa introdução da tecnologia no meio educacional tem como foco ampliar os recursos didáticos e simultaneamente promover a integração dos meios educacionais, tecnológicos e sociais.

No entanto, existem ainda muitas dificuldades na implementação de Tecnologias da Informação e Comunicação (TIC) como, a falta de recursos nas escolas e a não aceitação de alguns professores ao uso de recursos de tecnologia em sala de aula. O computador, assim como as Ciências Exatas, é ainda um pouco estigmatizado, como se criasse uma ligação com o adjetivo “difícil”. É necessário que toda a comunidade escolar compreenda que o uso da computação, de jogos virtuais e da própria programação computacional são atrativos modernos e podem melhorar substancialmente o nível de aprendizagem (STINGHEN, 2016).

Segundo dados do Centro Regional de Estudos para o Desenvolvimento da Sociedade da Informação de 2017, há laboratórios de informática em 81% das escolas públicas, mas somente 59% são usados. No entanto, o número de professores que utilizam a internet em atividades com os alunos cresceu de 39%

para 49%, mas a baixa velocidade da conexão ainda é um desafio para as escolas públicas.

Desse modo o ensino de Física utilizando o software de programação SciLab vem como uma alternativa para implementar as aulas do tema cinemática. A utilização desse recurso propõe que o conhecimento seja alcançado de forma que tenha compreensão da interação entre o modelo matemático e o fenómeno físico. Para Moreira (2019), o professor deve observar que os recursos que envolvam tecnologias da informação não devem se tornar o objeto de estudo da aula, mas sim auxiliar a compreensão do conteúdo para o qual está sendo executado.

1.3 Objetivo geral

Propor e implementar um produto educacional baseado na teoria do Construcionismo de Papert através do uso de programação computacional utilizando o software SciLab na construção de scripts que modelem gráfica e numericamente conceitos e problemas no estudo de Cinemática, que está elencado ao conteúdo de 1°ano do Ensino Médio.

1.4 Organização da Dissertação

O presente trabalho está organizado de seguinte maneira, Capítulo I, com introdução, justificativa e objetivo, Capítulo II, onde se apresenta uma definição de modelagem e a Teoria do Construcionismo de Seymour Papert, que versa sobre a construção do conhecimento por quem aprende, através de algo palpável que desperte o interesse de quem o desenvolve por meio do uso do computador.

O Capítulo III traz algumas definições fundamentais e abordagens mais usuais no estudo da Cinemática, assunto elencado no conteúdo programático do primeiro ano do Ensino Médio.

No Capítulo IV, é apresentado o referencial de modelagem computacional usando um software de programação SciLab. Onde se mostrará os principais comandos do console para a modelagem de problemas numéricos e gráficos.

O capitulo V, traz a implementação do Produto Educacional baseado na construção do conhecimento fundamentado na realização de uma ação concreta

que resulta em um produto visível, com a utilização do SciLab. E no Capítulo VI, apresenta-se as considerações finais.

CAPÍTULO II: MODELAGEM E O CONSTRUCIONISMO SOB A ÓTICA DE SEYMOUR PAPERT

2.1 Modelagem

Segundo Brandt, Burak e Klüber (2016) a modelagem e a simulação são eficazes ferramentas para o aluno, pois ao permitir que o aluno possa realizar experiências conceituais através dos softwares ele se aproxima de um conhecimento designado "*descoberta*". Nesta feita, o aluno utilizando a modelagem poderá construir um ambiente virtual próximo da realidade. Esse ambiente pode ser chamado de "micromundo". Assim, ao utilizar simulações de problemas que envolvam conceitos físicos o aluno irá, basicamente, alterar valores de variáveis ou parâmetros de entrada e verificar as alterações no resultado.

2.2 O Construcionismo de Papert

O principal teórico levado em consideração neste trabalho foi Seymour Papert com a teoria que ele intitulou Construcionismo. O teórico trabalha ainda, com a construção do conhecimento sob a produção de algo concreto e visível, basicamente com a utilização das Tecnologias da Informação e Comunicação (TIC). Segundo Nunes e Santos (2013):

> O Construcionismo é uma reconstrução teórica a partir do construtivismo piagetiano, proposta por Seymour Papert (1994 e 1986), originalmente em 1980. Papert concorda com Piaget (1976), em que a criança é um "'ser pensante" e construtora de suas próprias estruturas cognitivas, mesmo sem ser ensinada. Porém, se inquietou com a pouca pesquisa nesta área e levantou a seguinte interrogação: Como criar condições para que mais conhecimento possa ser adquirido por este aluno? A atitude construcionista implica na meta de ensinar, de forma a produzir o máximo de aprendizagem, com o mínimo de ensino. A meta do Construcionismo é alcançar meios de aprendizagem fortes que valorizem a construção mental do sujeito, apoiada em suas próprias construções no mundo. Dizer que estruturas intelectuais são construídas pelo aluno, ao invés de ensinadas por um professor não significa que elas sejam construídas do nada. Pelo contrário, como qualquer construtor, a criança se apropria, para seu próprio uso, em materiais que ela encontra e, mais significativamente, em modelos e metáforas sugeridas pela cultura que a rodeia (NUNES, SANTOS, 2013).

Segundo Brasão(2010), construcionismo é uma teoria que diz respeito à construção do conhecimento baseada na realização de uma ação concreta que resulta em um produto palpável, desenvolvido com o concurso do computador, que seja de interesse de quem o produz. A esse termo frequentemente se associa o

adjetivo contextualizado, na perspectiva de destacar que tal produto - seja um texto, uma imagem, um mapa conceitual, uma apresentação em slides - deve ter vínculo com a realidade da pessoa ou com o local onde será produzido e utilizado. O Construcionismo implica numa interação aluno-objeto, mediada por uma linguagem de programação.

Segundo Covo (2016), a tecnologia da qual dispomos nos dias de hoje é sustentada por modelos físicos e matemáticos que descrevem com precisão os diversos fenômenos da natureza. Esses fenômenos podem ser compreendidos e manipulados através de argumentos matemáticos acessíveis a nível médio como sequências numéricas, matrizes, funções trigonométricas, funções reais e complexas, amostragem, noções de limite e continuidade e outros conceitos matemáticos.

Unindo esses fatos chegamos a inevitável necessidade de usar meios para expandir o acesso e o conhecimento dos alunos na utilização de meios tecnológicos para manipulação desses conceitos matemáticos dentro da Física. Uma das vias de acesso a esse conhecimento é a utilização de aplicativos prontos para determinados temas da Física como conversores de graus, Mobile Science Aceleration, Simphyfsics, Weight Mass And Force Of Gravit entre outros.

Outra via é a utilização de softwares, que pode ser considerado um suporte lógico de programação, o qual possui um conjunto de programas, métodos, procedimentos, regras e documentação relacionados com o funcionamento e manejo de um sistema de dados. Ou seja, o usuário precisa conhecer o sistema de manipulação de entrada e saída de dados para que possa executar o que se pretende.

Para Wirth (1989) A programação é tida como “arte ou técnica de construir ou formular algoritmos de uma maneira sistemática”, onde algoritmos são códigos escritos nas diferentes linguagens de programação. Essa técnica pode ser desenvolvida ou/e aperfeiçoada através dos métodos de aplicação que consistem em manipular os softwares de programação e blocos lógicos.

Segundo Vieira (2006), é partindo das perspectivas de que o aluno seja o programador do software, que o computador poderá se colocar no lugar de aprendiz, de quem recebe as instruções, deixando para o aluno a função de ser pensante que tem a missão de comandar. Dessa forma, o indivíduo vai aprender através dos próprios ensinamentos e “descobertas”. Nesta perspectiva, o educando

adquire a habilidade de produção e manejo do seu próprio conhecimento viabilizando a construção da melhor forma para sua aprendizagem. Com base nisso evidencia-se que o termo “construcionismo” advém da necessidade de se relacionar a interação aluno-objeto, intermediada pela linguagem de programação. O professor/instrutor que conhece o software com o qual se está trabalhando atua nesse caso como mediador dessa interação.

Há um conjunto de softwares considerados educacionais, esses têm por objetivo trazer uma facilitação no processo de ensino–aprendizagem. Para Martins (2013), um conjunto de recursos computacionais é considerado software educacional quando este é projetado para auxiliar no processo de ensino e aprendizagem. Isso acaba despertando no aluno o maior interesse no objeto de estudo o que acarreta na obtenção do conhecimento cognitivo.

Nesse sentido Toledo (2015), afirma que o processo de ensino pode ser implementado através do uso de novas tecnologias como softwares educacionais, que podem acrescentar novos métodos, práticas e materiais além ainda de poderem remodelar práticas já utilizadas em sala de aula, facilitando com isso o aprendizado do aluno.

Tomaz, Da Silva, e Dutra (2017) indicam que o uso de diferentes ferramentas de ensino se torna cada vez mais viável frente à evolução tecnológica atual. As práticas de ensino que envolvem a programação surgem como uma alternativa quando se deseja desenvolver raciocínio lógico.

Segundo as Diretrizes Curriculares Nacionais para o Ensino Médio elaboradas pelo MEC (1998), a escola deve organizar-se de maneira a possibilitar um desenvolvimento da base lógica-matemática dos educandos ainda no período do ensino médio, proporcionando meios de o aluno ter o conhecimento reconstruído e estimulado o raciocínio a experimentação e a resolução de problemas utilizando expressões cognitivas superiores, ou seja, expressar melhor suas ideias, raciocínio, tomadas de decisão, entre outras, em particular, para resolução de problemas de Física.

CAPÍTULO III: ESTUDO DE CINEMÁTICA E SUAS APLICAÇÕES

O presente Capítulo traz uma breve abordagem do estudo da Cinemática. Trataremos do movimento sem considerar as forças que o provocam. Utilizamos na parte bibliográfica obras como Resnick, Halliday, e Krane(2008); Young e Freedman (2008) e Alonso e Finn(2012).

Segundo Youg e Freedman (2008) a Cinemática é a parte da Física constituída pelo estudo dos movimentos mais simples, em que uma partícula está em movimento retilíneo, mas sem se ater aos fatores que os acarretam. É ainda, a parte da Mecânica que estuda e descreve o movimento, determinando a posição, a velocidade e a aceleração de um corpo em cada instante do tempo.

Nesse ramo da Física, não são consideradas para efeito de cálculos a massa dos corpos nem as forças que são aplicadas sobre eles. Os corpos em estudo, denominados móveis, são considerados pontos materiais, que são corpos cujas dimensões não interferem no estudo do fenômeno considerado. O tempo é uma noção aceita sem definição, fundamental na descrição de qualquer movimento.

A cinemática trata de movimentos que acontecem sem a necessidade de estimar a direção e o sentido**.** Em vista dessas condições, os cálculos se tornam mais simples. Em seguida veremos alguns dos conceitos fundamentais para o estudo Cinemática.

3.1 Espaço

Intervalo entre um ponto e outro; Distância percorrida por um ponto em movimento. Quando um corpo cai livremente, os espaços que ele percorre são proporcionais aos quadrados dos tempos empregados para percorrê-los.

3.2 Deslocamento Escalar

Deslocamento escalar (Δs) é a variação da posição de um corpo móvel dentro da trajetória, ou seja, é a diferença entre a posição final (s) do móvel e a posição inicial (s_0) do móvel.

$$\Delta s = s - s_0 \tag{1.1}$$

Podemos encontrar o deslocamento através da integral definida da função velocidade. Digamos que uma partícula se mova em linha reta, com velocidade de $v(t) = 5 - t$ metros por segundo no qual t é dado em segundos.

Para analisar o deslocamento escalar de uma partícula que se move de 0 a 10 unidades de tempo (u) com velocidade constante podemos ter:

$$\int_0^{10} |v(t)|dt \Longrightarrow \begin{cases} 5-t, se\ t<5 \\ t-5, se\ t>5 \end{cases}$$

$$\int_0^5 5-t\ dt + \int_5^{10} -5+t\ \ dt$$

$$\left[25-\frac{25}{2}\right]+\left[(-50+50)-(-25+\frac{25}{2})\right]$$

$$25u$$

A velocidade escalar é a taxa de variação na *distância total*, portanto sua integral definida nos dará a distância total percorrida, independentemente da posição.

3.3 Velocidade Média Escalar

Segundo Alonso e Finn (2012, p.82), a Velocidade média (V_m) é a razão entre o deslocamento escalar $\Delta S = S - S_0$ e o correspondente intervalo de tempo $\Delta t = t - t_0$, que o móvel leva para percorrê-lo.

$$V_m = \frac{\Delta S}{\Delta t} \qquad (1.2)$$

Ex.: Você dirige sua BMW por uma estrada em linha reta por 5,2 milhas a 43mi/h, quando sua gasolina acaba. Então, você caminha 1,2 milha a mais de até o posto. De gasolina mais próximo, em 27 minutos. Qual a sua velocidade média desde o instante em que o carro parti até a sua chegada ao posto de gasolina? (RESNICK, HALLIDAY, e KRANE, 2008, p. 27)

É possível determinar a velocidade média a partir da equação xx conhecendo o $\Delta S\ e\ \Delta t$. Que são respectivamente

$$\Delta S = 5{,}2mi + 1{,}2mi = 6{,}4mi$$

$$\Delta t = \frac{5{,}2mi}{43mi/h} + 27min \Longrightarrow 7{,}3min + 27min = 0{,}57h$$

Usando a equação (1.2) temos

$$V_m = \frac{\Delta S}{\Delta t} = \frac{6{,}4mi}{0{,}57h} = 11{,}2mi/h$$

3.4 Velocidade Escalar Instantânea

A velocidade escalar instantânea é considerada um limite da velocidade escalar média V_m, quando o intervalo de tempo tender a 0. A velocidade escalar instantânea é totalmente derivada do espaço, em relação ao tempo. Essa “derivação” pode ser representada pela equação:

$$V_{inst} = \lim_{\Delta t \to 0} V_m = \lim_{\Delta t \to 0} \frac{\Delta S}{\Delta t} \tag{1.3}$$

Onde delta ΔS é o deslocamento de tempo e delta Δt é o deslocamento de tempo. E que, por definição, é a derivada de S em relação a t;

$$V_{inst} = \frac{dS}{dt} \tag{1.4}$$

Ex.: Uma partícula move-se ao longo do eixo x de modo que sua posição em qualquer instante é dada por $S = 5t^2 + 1$, onde S é dado em metros e t em segundos. Calcular a sua velocidade escalar instantânea no instante $t = 2\ seg$ (ALONSO E FINN, 2012, p. 84).

Fazendo a derivada de S em relação a t na função, obtemos o valor da velocidade instantânea.

$$S = 5t^2 + 1$$

$$\frac{dS}{dt} = 10t$$

$$dS = 10tdt$$

$$v_{inst} = \frac{dS}{dt} = \frac{10tdt}{dt} = 10t$$

Substituindo o valor t por 2 temos:

$$v_{inst} = 10 * 2 = 20m/s$$

3.5 Aceleração Média Escalar

A aceleração escalar média é considerada a relação entre a variação de velocidade (V) e o intervalo de tempo (t) que ocorreu esta variação. Velocidade de uma partícula é a razão, segundo a qual, sua posição varia com o tempo. Seja uma partícula situada num ponto α, num instante t_1. Em um instante posterior, t_2, a partícula estará num ponto β. A equação da aceleração escalar média é:

$$a_m = \frac{\Delta V}{\Delta t} \Longrightarrow a_m = \frac{V - V_0}{t - t_0} \tag{1.5}$$

(Unimep-SP) Uma partícula parte do repouso e em 5 segundos percorre 100 metros. Considerando o movimento retilíneo e uniformemente variado, podemos afirmar que a aceleração da partícula é de:

Podemos utilizar a equação $\Delta S = V_0 t + \frac{1}{2} a t^2$, onde

$$100 = \frac{1}{2} a 5^2 \Longrightarrow a = \frac{100}{12{,}5} = 8 m/s^2$$

OBS.:

i) Seja aceleração **a** constante, temos:

$$a = \frac{dv}{dt}$$

$$dv = adt$$

$$\int dv = \int adt = a \int dt$$

$$v = at + C \tag{1.6}$$

Determinando o valor de C.

Fazendo *t* = 0, instante em que $v = v_0$, substituindo em 1.6, temos:

$$v_0 = a(0) + C = C, \tag{1.7}$$

Substituindo em 1.6,obtemos que:

$$v = v_0 + at \tag{1.8}$$

ii) Seja agora:

$$v = \frac{dS}{dt}$$

$$ds = vdt$$

$$\int ds = \int vdt = \int (v_0 + at)dt = \int v_0 dt + \int atdt$$

$$\int ds = v_0 \int dt + a \int tdt$$

$$s = v_0 t + a\frac{t^2}{2} + C' \quad (1.9)$$

Determinando o valor de C.

Fazendo *t* = 0, instante em que $s = s_0$, substituindo em 1.9, temos:

$$s_0 = v_0(0) + a\frac{(0)^2}{2} + C' = C', \quad (1.10)$$

Substituindo em 1.9, obtemos que:

$$s = v_0 t + a\frac{t^2}{2} + s_0 \quad (1.11)$$

De 1.8 e 1.10, podemos determinar com álgebra básica as seguintes equações:

$$v^2 = {v_0}^2 + 2a(s - s_0) \quad (1.12)$$

$$s = s_0 + \frac{1}{2}(v_0 + v)t \quad (1.13)$$

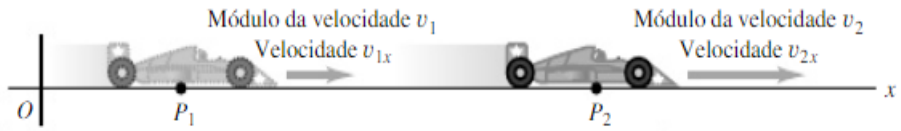
$$s = s_0 + vt - \frac{1}{2}at \quad (1.14)$$

3.6 Aceleração Escalar Instantânea

Quando a aceleração escalar média chega ao seu limite, temos a aceleração escalar instantânea, que determina a aceleração do corpo em um determinado momento, isto é, quando o intervalo de tempo tende a zero; $a_m = \lim_{t \to 0} \frac{\Delta v}{\Delta t}$. A velocidade escalar instantânea é a derivada da função horária de sua velocidade escalar $a_m = \frac{dv}{dt} = \frac{d}{dt}\left(\frac{dS}{dt}\right) = \frac{d^2S}{dt^2}$.

Suponha que a velocidade V_x do carro da figura 01 em qualquer instante t seja dada pela equação $V_x = 60m/s + (0{,}5m/s^2)t^2$ (YOUNG E FREEDMAN, 2008, p. 43).

Figura 01
Nenhuma entrada de índice de ilustrações foi encontrada.



Fonte: Young e Freedman (2012, p. 43)

Ache a aceleração instantânea do carro para $t_1 = 1{,}0\ s$.

Se

$$V_x = 60 + \frac{1}{2}t^2 \Longrightarrow \frac{dv}{dt} = t$$

Para $t = 1$, temos:

$$a_m = 1m/s^2$$

3.7 Sobre os Movimentos:

Movimento Acelerado**:** Um movimento será acelerado quando os vetores velocidade e aceleração possuírem o mesmo sentido, isto é, o módulo da velocidade escalar aumenta com o tempo.

Movimento Retardado: Quando os sentidos dos vetores velocidade e aceleração forem opostos o módulo da velocidade diminui com o tempo, trata-se de um movimento retardado.

Movimento Uniforme (MU): O movimento uniforme acontece com velocidade constante. Nesse tipo de movimento, não há aceleração.

Movimento uniformemente variado (MUV)**:** É um movimento cuja velocidade aumenta ou diminui de maneira constante**.**

Gráficos Representativos dos Movimentos: Os gráficos dos movimentos são, no geral o gráfico de uma função, onde, podemos ter a velocidade variando em função do tempo, a posição em função do tempo, da aceleração instantânea em função da velocidade, entre outros.

CAPÍTULO IV: O SCILAB PARA O ENSINO E APRENDIZAGEM DE FÍSICA

Para Marinho (2014), o ensino de Física pode se tornar bem mais interessante para os alunos se estiver aliado ao uso de novas Tecnologias de Informação e Comunicação (TIC), como um software de modelagem computacional para simular modelos físicos que podem ser utilizados pelos professores e alunos em suas aulas, proporcionando assim uma forma de facilitar a aprendizagem e despertar o interesse dos educandos pelo estudo da Física.

O trabalho aqui pretende utilizar o software SciLab como plataforma de programação para o aluno de Física do Ensino Médio. SciLab é uma plataforma de software para computação numérica desenvolvida na França em 1990 pelo INRIA - Institut National de Recherche en Informatique et en Automatique (Instituto Nacional de Pesquisa em Informática e em Automação) e, pelo ENPC - École Nationale des Ponts et Chaussées (Escola Nacional de Pontes e Estradas). Ele conta com um dos melhores ambientes de computação, sendo um software aberto, de código fonte livre para os usuários, ou seja, ele pode ser compartilhado livremente e/ou modificado para se adaptar às necessidades podendo ainda ter essas modificações livremente compartilhada com outros usuários.

Esse software existe em diversas versões como Linux, Windows e Mac OS X. O SciLab foi desenvolvido baseado no software MATLAB. A sintaxe do SciLab é baseada no MatLab e assim como ele usa o '*prompt (*-->)' para indicar a interpretação dos scripts.

O SciLab tem como diferencial uma maior facilidade na criação de seus programas. Isso se caracteriza por uma linguagem mais simples e disponibilização das estruturas de outras linguagens de programação mais convencionais. O SciLab por ser um interpretador de comando leva um tempo mais elevado para executar tarefas como simulações otimizações e até mesmo plotagem de gráficos do que softwares com linguagens compiladas. Todavia como o software em questão pode executar programas feitas em linguagens como Fortran, C, C++ ou MatLab isso pode ser uma alternativa para sanar essa diferênça no tempo.

Neste Capítulo será abordado a linguagem do software SciLab e seus principais comandos de uso na modelagem de problemas numéricos e construção de gráficos com 2 e 3 dimensões.

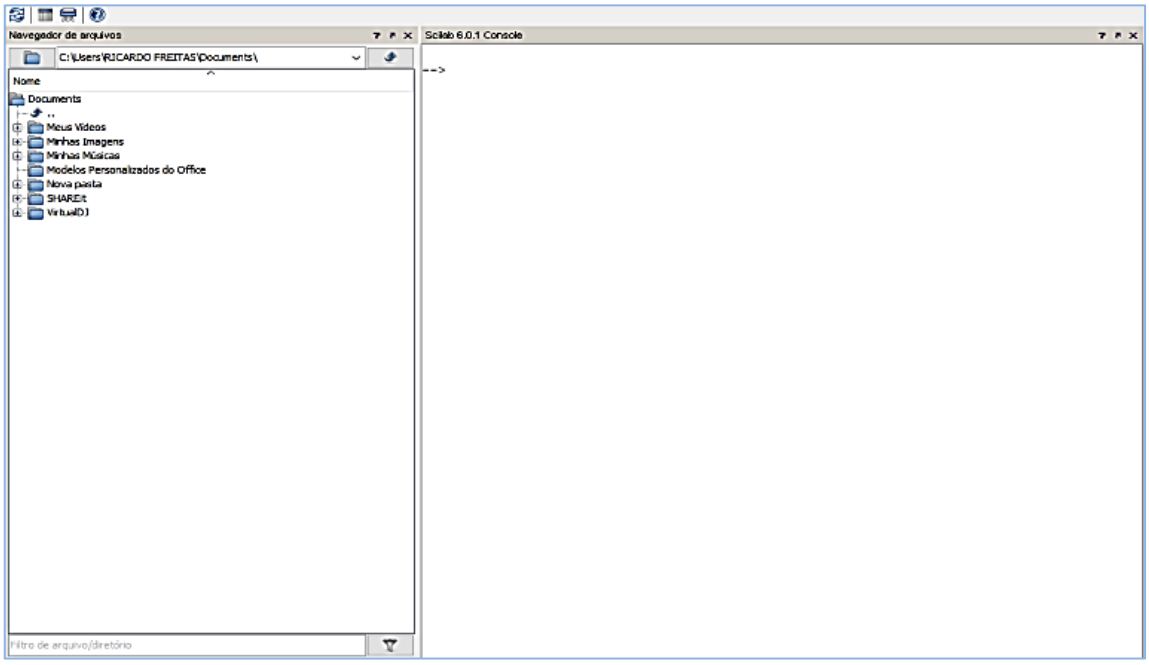
4.1 Linguagem do Software SciLab.

Nesse software a linguagem e o sistema possuem o mesmo nome, “SciLab”. Ele tem um interpretador de comandos em seu ambiente de programação numérica, onde disponibiliza um editor para a construção de programas através de uma linguagem de programação simples e de fácil aprendizado baseada em argumentos matemáticos (SciPad). Com ele pode-se trabalhar com rotinas escritas em outras linguagens como FORTRAN, C ou C**++**.

4.2 Layout do Software

O SciLab tem uma área de trabalho principal chamado *“console”* e um “editor de arquivos” onde podem ser salvos arquivos em formato de extensão *.sci.* É no *console* que os programas e cálculos numéricos são executados. O *console* trabalha com o “*prompt*”, indicado pelo sinal *(−−>),* e o local de inserção e execução imediata do comando ordenado pelo programador.

Figura 02



Fonte: o autor

4.3 Principais Funções

Umas das funções elementares do SciLab é um sistema de auxílio do software chamado “*help”.* Que permite ao usuário receber ajuda sobre como utilizar qualquer comando a ser inserido na área de execução de tarefas, bastando o

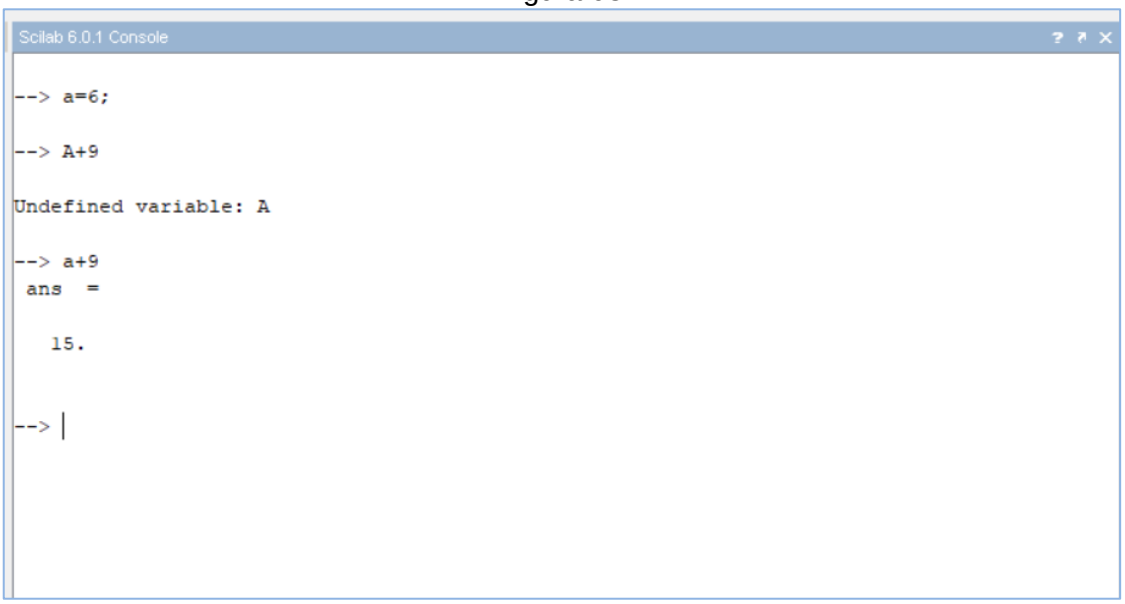
usuário entrar com "*help + comando desejado". O "help"* sem argumentos fornece a página de hipertextos dos capítulos de ajuda, que é fundamentalmente útil.

Várias Variáveis podem ser definidas no *console* ou no editor. Essas variáveis são reconhecidas e gerenciadas por caracteres pelo software. Uma letra minúscula é diferente da mesma letra maiúscula e qualquer uma delas pode ser substituída por um novo valor a partir do momento que o usuário a redefina com um valor diferente. Na linguagem SciLab o (;) é usado para suprimir o valor (resposta) de uma variável ou de uma expressão.

A variável (ans) é utilizada pra expressar resultados de expressões ou variáveis não previamente definidas.

Ex.: Atribuímos a variável $\boldsymbol{a}$ o valor 6, mas ao executarmos no *prompt A+9*, não teremos resposta definida, pois a variável $\boldsymbol{A}$ é diferente da variável $\boldsymbol{a}$. Porém, ao executarmos $a + 9$, obtemos a resposta $ans = 15$, pois foi definido o valor 6 para variável $\boldsymbol{a}$ como pode ser vista a seguir:

Figura 03



Fonte: o autor

No console ou no editor de programas o comando duas barras "//" é utilizado para adicionar um texto explicativo ou um comentário ao que se pretende fazer, ou seja, o texto após duas barras não será executado pelo software.

A tecla ↑ é utilizada no *"prompt"* para chamar o comando mais recente executado. Dai em diante as teclas ↑ e ↓ navegam pelos comandos recém executados no *"prompt" e as teclas* ← *e* → movem o cursor entre os caracteres do comando.

Para limpar a área de trabalho do console é utilizado o comando *"clc"*. Outro comando importante é o *"tohome" usado* para posicionar a área de trabalho com cursor do *"prompt"* no canto superior esquerdo.

Na linguagem SciLab podem ser inseridas três formas distintas de constantes: numéricas, literais e lógicas.

A numérica é formada por caracteres numéricos munidos de um ponto decimal, se o número for decimal, tendo necessariamente um sinal caso seja negativa ou um fator multiplicativo negativo isolado, podendo ainda ter uma potência de dez como fator multiplicativo. A potência de dez é indicada por qualquer uma das letras $\{e, E, d, D\}$ seguida de outros caracteres numéricos, que indicam o fator em potência de 10 pelo qual o número está multiplicado. Assim como $17D5$, que indica 17×10^5 ou $97e11$ que indica 97×10^{11}.

No tratamento de números complexos o SciLab oferece uma facilidade não muito comum nos softwares de programação. Ele trabalha com complexo do mesmo modo que trabalha com números reais, basta acrescentar a indicação $(* \%i)$ ao final da parte imaginaria dos números complexos.

Além do comando *"help",* que é utilizado para consultar um comando do SciLab, há também outra importante ferramenta de consulta no software, o *"apropos".* Ele é usado para pesquisar sobre um argumento ou operação não necessariamente como é escrito o comando no SciLab. Ele mostra na biblioteca de informações do software todos os comandos que fazem menção ao comando ou operação sobre o qual a pesquisa é feita.

4.4 Expressões Aritméticas

Na linguagem do software SciLab para realizar expressões aritméticas os operadores são fixos. Os valores operados podem ser constantes ou variáveis definidas, que podem ser chamadas de incógnitas. Esses operadores são utilizados na definição de fórmulas, receitas ou equações. Ainda podem haver simples cálculos numéricos como uma calculadora numérica tradicional, a tabela a seguir mostra exemplos de como podem ser escritas na linguagem algumas operações simples.

Quadro 01

Operação	*Operador*	*Uso*
adição	$+$	$a + b$

$subtração$	$-$	$a - b$
$multiplicação$	$*$	$a * b$
$Divisão$	$/$	a/b
$Potência$	^	$a\hat{}b$
$Radiciação$	$sqrt()$	$sqrt(a)$

Fonte: o autor

4.2 – Janela Gráfica

SciLab possui várias funções gráficas integradas, para a apresentação visual dos dados processados pelo software. Utilizando SciLab pode-se plotar gráficos de funções em duas ou três dimensões. Os gráficos feitos pelo SciLab podem ser programados no editor de arquivos ou diretamente na área de execução de comandos, o *prompt*.

4.2.1 - Comando $Plot()$

$Plot()$ é o meio mais simples de plotar um gráfico de uma função usando a linguagem Scilab. utilizando o comando $plot(x, y)$, no prompt ou no editor de programas, para um valor de x sendo um vetor que pode variar em um intervalo definido da seguinte forma:

$$--> x = [1:0.1:10]$$

Onde 0.1 entre dois sinais $(:)$, representa o incremento de interação, ou seja, o intervalo de interações do programa. A cada décimo, como foi escolhido no exemplo, haverá a geração de um ponto no gráfico.

Podemos ainda executar um intervalo na seguinte forma: $x = [1:10]$. Onde os dois pontos (:) entre os dois escalares indica um intervalo monótono, ou seja, a variação é de uma unidade.

O valor de y é dado em função de x, no tipo $y = f(x)$.

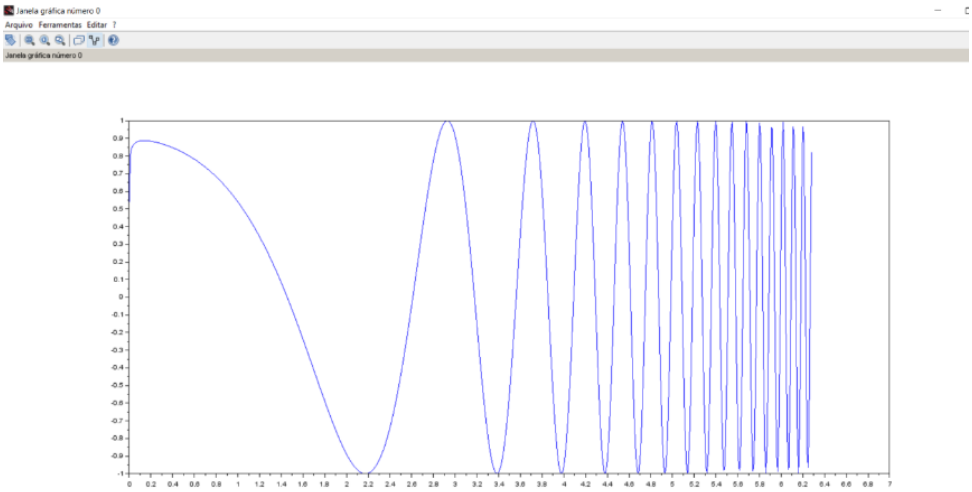
Os valores se associam num plano de eixos perpendiculares e ordenados. Por exemplo:

$--> x = [0:2 * \%pi];$

$--> y = sin(x);$

$--> plot(x, y)$

Figura 04



Fonte: o autor

CAPITULO 5: IMPLEMENTAÇÃO DO PRODUTO EDUCACIONAL

O processo de aprendizagem depende da razão que motiva a busca do conhecimento. Nos trabalhos de Pezzini e Szymanski (2019), encontramos que uma maneira prática de incentivar os alunos a buscarem conhecimento é o desenvolvimento da autonomia. A autonomia, neste sentido, é outorgar ao aluno a capacidade de desenvolver e criar também pela oportunidade de participação.

Com a utilização de linguagens de programação o aluno assume a posição de orientador do computador. Com o auxílio constante do professor responsável ele pode ser capaz de dar os comandos necessários para que a máquina execute o que deve para modelar com perfeição o fenômeno tratado em cada momento.

Este Capítulo é sistematicamente organizado com o intuito de fornecer a descrição de como ocorreu a implementação do produto, estabelecendo a ordem cronológica em que ocorreram as ações inerentes, seguindo a Sequência Didática, preparada para tal.

5.1 Sequência Didática

1. Identificação

Escola:

Disciplina: Física

Série:1° ano

Data:

Tempo previsto: 5horas (ou 6 aulas de 50min)

2. Tema: Modelagem de Fenômenos Físicos Utilizando Software Numérico e Gráfico SciLab.

3. Conteúdo

- Movimentos: variações e conservações;
- M.U.V, M.R.U, Estudo dos movimentos, Queda livre, lançamento vertical, deslocamento.

4. Objetivo

Sequência didática para modelar numérica e graficamente fenômenos de Cinemática com uso de um Produto Educacional (Apêndice I) baseado no software de programação SciLab sob a perspectiva da teórica do Construcionismo de Papert.

5. Conhecimentos Prévios

- Equações do primeiro e do segundo grau;
- Funções de primeiro grau e segundo grau.

6. Recursos

Para execução do produto educacional serão necessários:

- Físicos/Concretos: Laboratório de informática, Datashow, quadro branco, marcador, papel e caneta;
- Digitais: O software SciLab instalado nas maquinas do LAB.

O software está disponível para download gratuito no site oficial do grupo SciLab Enterprises em: http://www.SciLab.org/en/download/6.0.1 .

7. Desenvolvimento

1ª Etapa. APRESENTAÇÃO DA AULA

Tempo previsto: 20min.

- A aula será aberta com uma palestra a respeito da importância do uso e do aprendizado da tecnologia, sobretudo na área de programação em linguagens tradicionais;
- Nesse momento o aprendente receberá informações a respeito do que é fundamentalmente programação computacional e o tipo de programação que irá ser mostrada a seguir;

- Ainda será aberto espaço durante esse tempo para ouvir alguma dúvida ou perspectiva dos alunos no desenvolver da aula.

2ª Etapa: AMBIENTAÇÃO

Tempo previsto: 15min

- Mostrar como funciona a primeira parte do Softwares que é o Executor de comandos chamado *prompt;*
- O Aluno perceberá através de explicação que o software também pode ser utilizado como uma calculadora numérica obedecendo a comandos simples de aritmética.

3ª Etapa: FUNDAMENTAÇÃO BÁSICA

Tempo previsto: 40 min

- Incentivar o aluno a iniciar o software, que está previamente instalado nos computadores do LAB, afim de verificar e realizar operações aritméticas diversas no executor de programas do Software;
- Mostrar o editor de programas do SciLab, onde os programas que executam funções são escritos. Professor irá passar comandos básicos de entrada e saída de comandos no software. Disponível em: https://help.SciLab.org/docs/5.3.0/pt_BR/index.html;
- Mostrar alguns programas feitos com o SciLab que modelam fenômenos dentro do conteúdo que a aula é destinada:

4ª Etapa: EXECUÇÃO

Tempo previsto: 3 horas divididas em intervalos de aulas diferentes

- Pedir aos alunos que baseado no que viram até ali utilizem o software para modelar fenômenos da física que envolvam movimentos com velocidade constante ou aceleração constante;
- Diversos programas vão ser criados ao vivo utilizando o projetor para servirem de base e exemplo para aos alunos;
- Após os alunos compreenderem os comandos de geração numérica da modelagem o professor irá também com uso do projetor ensinar a plotagem dos gráficos dos fenômenos que os alunos acabaram de modelar;
- Apresentação de Programas do SciLab que serão implementados em sala:

- Velocidade em função da posição
- Gráfico da posição em função do tempo
- Posição em função do tempo
- Deslocamento em função do tempo e velocidade Posição final em função do tempo com aceleração constante
- Distância em queda livre ao longo do tempo
- Raízes de equação do segundo grau
- Gráfico do deslocamento em função do tempo e velocidade
- Gráfico da posição final ao longo do tempo com aceleração constante
- Gráfico da velocidade em função do tempo
- Gráfico da queda livre

5ª Etapa: AVALIAÇÃO

Tempo previsto: 45min

A o final do período de execução o professor irá avaliar a funcionalidade dos programas escritos pelos alunos e verificar se as fórmulas foram corretamente empregadas.

Essa avaliação será no tocante a realização do que cada programa construído se propõe a fazer. O professor irá executar cada programa de cada aluno e verificar a funcionalidade dos mesmos.

O professor irá avaliar os programas para plotagem dos gráficos dos fenômenos se os mesmos descrevem o comportamento da variação de tempo velocidade ou aceleração.

Ao final uma avaliação a respeito de cada programa será feita contendo as seguintes perguntas:

5.2 Descrição da Implementação do Produto

Para implementação do produto educacional, foi realizado um minicurso, com participação de 12 estudantes do 1º ano do Ensino Médio, vespertino, da Escola José Cursino de Azevedo, do município de Marabá-Pará nos dias 18 e 19/12/2019. O Professor de Física dos alunos também participou como observador e auxiliar no minicurso, que foi realizado no Laboratório de Informática do Instituto de Ciências Exatas da Universidade Federal do Sul e Sudeste do Pará, Marabá-PA. O produto educacional versa sobre a utilização de uma linguagem de

programação computacional como ferramenta didática para o ensino de fenômenos da Cinemática.

Na primeira etapa foi realizado a introdução ao tema programação computacional conforme apresentado no Produto Educacional (Apêndice I). Em seguida foi apresentado o software SciLab utilizado para realização do minicurso.

Em seguida, na segunda etapa foi mostrado a execução de comandos simples no console e, após isso, mostrados os principais comandos de entrada, saída e interação dos dados. Posteriormente a essa etapa os alunos puderam utilizar o software para executar operações matemáticas.

Conforme a Sequência Didática a terceira etapa compreendeu com que o aluno utilizasse as instruções vistas até aqui para escrever scripts que modelassem numérica ou graficamente os fenômenos do campo da cinemática solicitados pelo professor.

5.3 Introdução à Programação no SciLab (Etapa de Apresentação)

A turma foi inserida no contexto de programação, na etapa de Fundamentação, afim de despertar o interesse e aumentar o grau de conhecimento na área, através de uma apresentação em slides que é apresentada na forma textual como veremos a seguir.

A programação de computadores tem linguagens tradicionais, ou seja, mais utilizadas por programadores ao longo do tempo, como Fortran, Pascal, C, C++, Mathlab, Phayton e o SciLab, esta última bastante popular por sua facilidade de uso. É importante conhecermos um pouco daquelas que podemos configurar como as principais, dentre estas.

Os programadores desenvolveram as linguagens de programação que são conjuntos de padrões e comandos usados para dar ordens aos processadores. As linguagens de programação usadas pelos programadores passam por um compilador, que é usado para transformar os códigos escritos na linguagem de programação em uma linguagem que o processador entenda.

Uma das primeiras linguagens de programação se chama Assembly. Ela é uma montagem utilizada para programar códigos entendidos por dispositivos computacionais.

O Fortran é uma das primeiras linguagens de alto nível desenvolvidas. Ela inovou muito ao trazer comandos e funções prontas para serem compiladas. A

linguagem **C** combina especificidades das primeiras linguagens com a evolução de linguagens mais modernas, de alto nível. Essa linguagem é ainda muito utilizada. O **C++** é um aperfeiçoamento da linguagem **C** mantendo suas características só que aumentando seu nível e simplificando a sua escrita pelo programador.

Python é uma linguagem moderna e simples muito utilizada por programadores iniciantes. Com ela pode-se facilmente construir blocos lógicos usados em jogos. MatLab é um poderoso software numérico e gráfico utilizado em diversas tarefas no campo da programação e da modelagem. Sua principal característica é a facilidade de escrita dos seus scripts. Esse software tem um layout simples e agradável.

O SciLab é um software de programação numérica e gráfico bastante utilizado para fins acadêmicos. Suas principais características são: ser um programa aberto, livre, ou seja, grátis e de fácil manuseio pelo usuário, não sendo necessário que esse seja um programador experiente. A partir do momento que se aprende a usar suas ferramentas ele se torna um poderoso aliado na vida do aluno podendo realizar tarefas dignas de grandes outras linguagens de programação.

Esta última linguagem de programação, dadas suas caraterísticas, é a que apresentaremos como ferramenta para modelagem para o ensino de Física, em particular, para resolução de problemas de Cinemática.

5.4 Conhecendo o Software (Etapa de Ambientação)

Esta etapa de ambientação consistiu em mostrar como funciona a primeira parte do Software, que é o executor de comandos chamado "*prompt".*

Figura 05

Fonte: o autor

Após clicarem no *"prompt"*, os alunos foram convidados a analisar e descobrir que o executor de programas numéricos do software pode ser utilizado como uma calculadora numérica avançada. Eles puderam executar operações no "*prompt*" do tipo

Raiz quadrada de 81:

$$--> sqrt(81)$$

$$ans \ =$$

$$9.$$

12 elevado a 7:

$$--> 12\text{^}7$$

$$ans \ =$$

$$35831808.$$

Criar um polinômio de quarto grau, ou qualquer outro grau, e calcular as raízes:

$$--> p = poly([12,-20,-11,5,2],'x','coeff')$$

$$p \ =$$

$$12 \ - 20x \ - 11x^2 \ + 5x^3 \ + 2x^4$$

$$--> roots(p)$$

$$ans \ =$$

$$-3.$$

$$-2.$$

$$2.$$

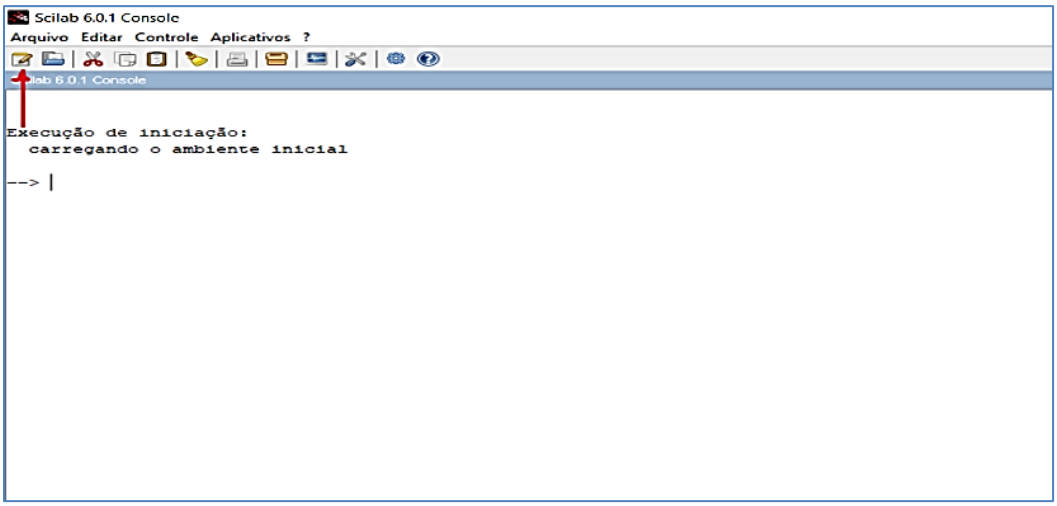
$$0.5$$

Onde, p=poly([12,-20,-11,5,2],'x','coeff') representa a equação:

"$2x^4 + 5x^3 - 11x^2 - 20x + 12 = 0$ " *e as raízes são:* $x' = -3, x'' = -2$, $x''' = 2$ *e* $x'''' = \frac{1}{2}$.

Os alunos foram ainda orientados a abrirem o editor de scripts no canto superior esquerdo do console. Mostramos o editor de programas do SciLab, onde os programas que executam funções são escritos. Passamos comandos básicos de entrada, saída e processamento de dados no software, disponíveis no endereço oficial em: https://help.SciLab.org/docs/5.3.0/pt_BR/index.html.

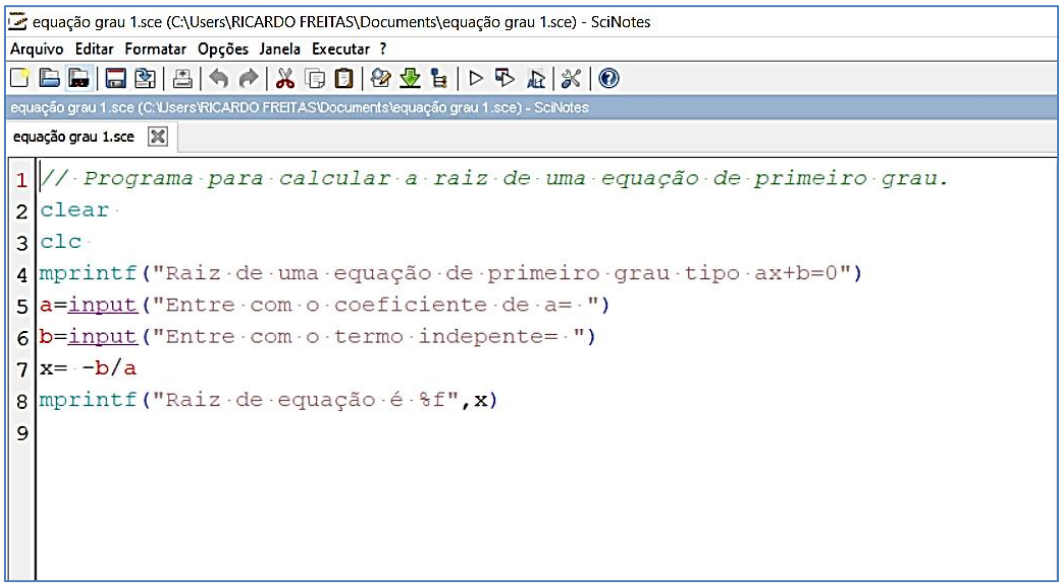
Figura 06



Fonte: o autor

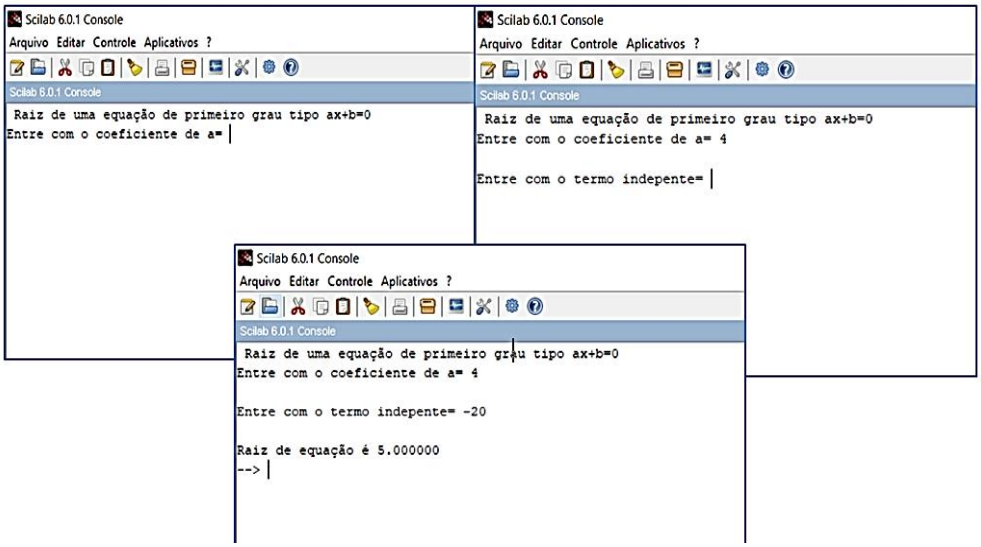
Após, os educandos, foram familiarizados com o uso do software através da execução de dois programas simples de cálculo numérico que construímos passo a passo acompanhando os alunos. Os programas utilizados como demonstração foram programas para mostrar as raízes de equações do primeiro e do segundo grau. Os comandos na Figura 07 foram usados para criar um programa capaz de resolver numericamente uma equação de primeiro grau, servindo como exemplo e incentivo aos alunos.

Figura 07



Fonte: o autor

Figura 08

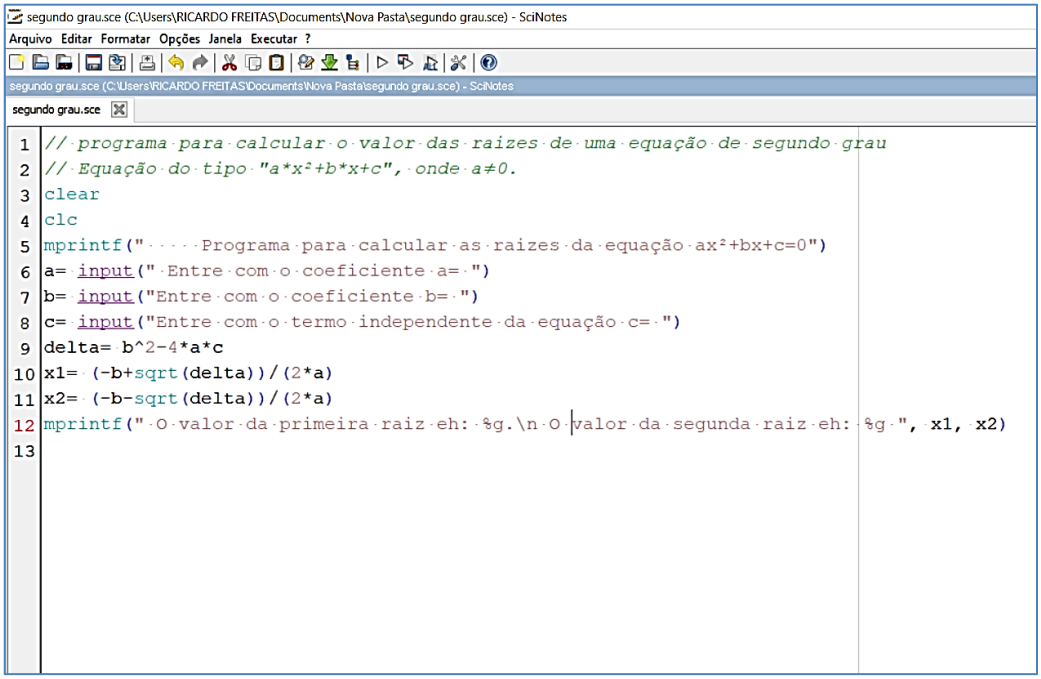


Fonte: o autor

Os valores inseridos na etapa de execução do programa foram escolhidos aleatoriamente pelo professor ou por sugestão dos próprios educandos.

Programa simples para calcular as raízes de uma equação do segundo grau.

Figura 09

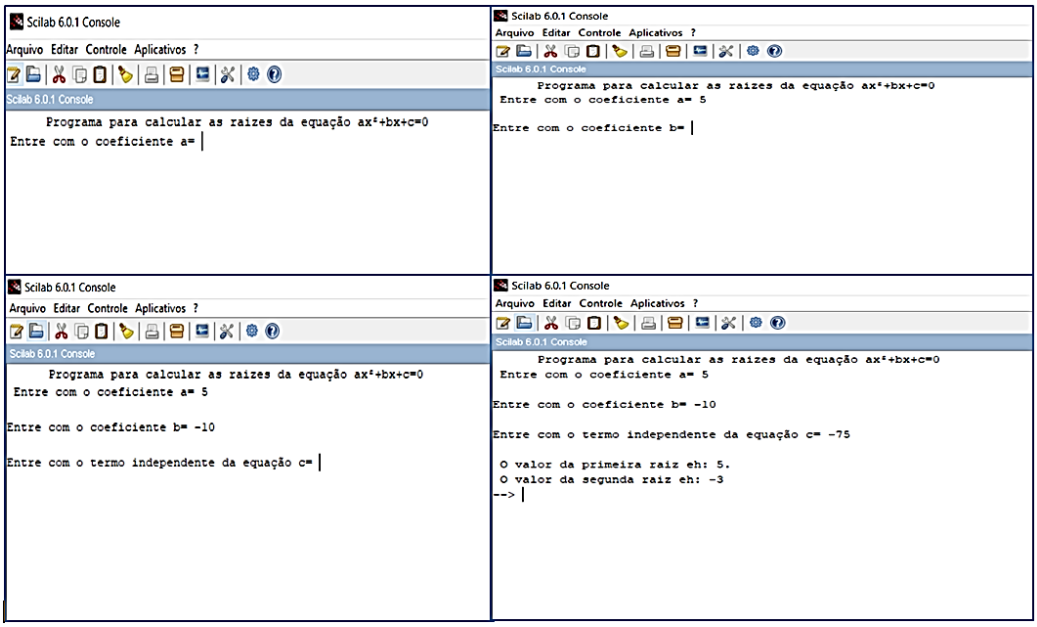


Fonte: o autor

Após escrevermos o programa acima pedimos que os alunos sugerissem equações do segundo grau para serem resolvidas pelo programa. Essa fase gerou maior interesse e participação dos alunos no desenvolvimento de um conhecimento que eles puderam adquirir e podem levar para outros ramos da Física.

Na figura 10 temos a execução do programa.

Figura 10



Fonte: o autor

Os valores inseridos na execução do programa também podem ser retirados de livros didáticos.

5.5 Construindo e executando um script. (Fundamentação Básica)

Nessa etapa do trabalho pedimos aos alunos para acompanharem no software a construção de um script no editor de arquivos. O arquivo foi nomeado e salvo em um local acessível ao software. As ideias fundamentais de programação ensinadas aos alunos foram três; algoritmo de entrada de dados, algoritmo de processamento de dados e algoritmo de saída.

Usamos basicamente os seguintes comandos de entrada, saída e processamento de dados:

a) ***clc*** para limpar o executor de arquivos antes da nova execução;

b) ***clear*** função para remover itens da área de trabalho, liberando memória do sistema;

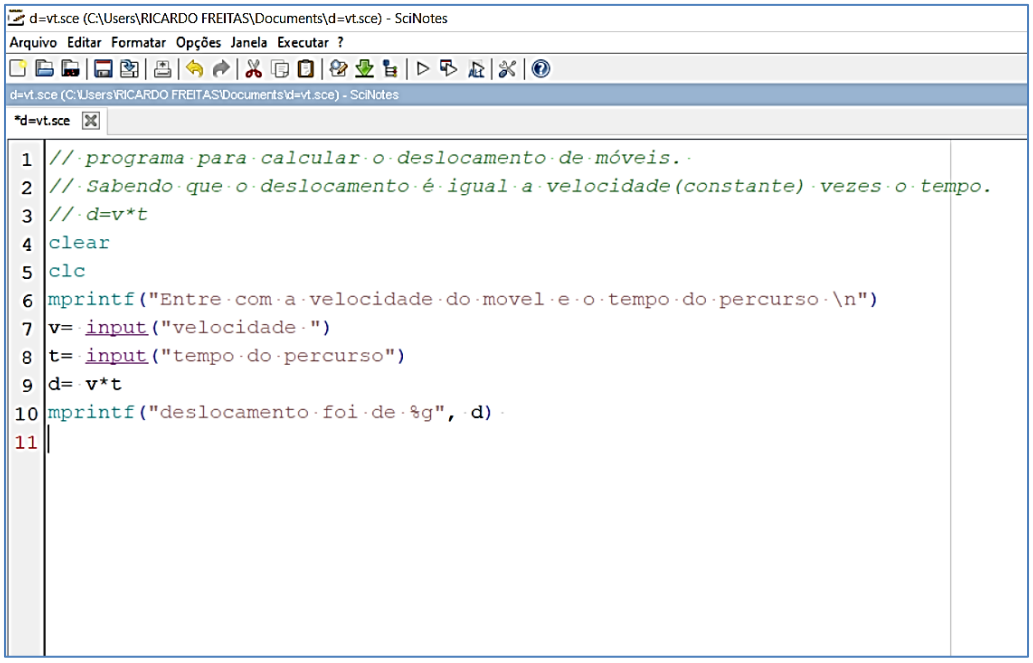
c) **mprintf(“ “)** é função de saída. Imprime um texto e/ou o valor resultado de alguma variável definida no script executado;

d) **input ( )** fornece ao programa executado o “prompt” para a inserção de valores a alguma variável do problema, que irá ser processada pelo script e dará como resposta a solução do problema para aquele valor.

5.6 Modelagem de Fenômenos de Cinemática (Etapa de Execução)

Como exemplo inicial de modelagem construímos passo a passo o seguinte programa de modelagem da equação da Distância.

Figura 11



Fonte: o autor

O programa construído se inicia com o uso de “//”, que como visto, indica que o texto após sua inserção não será executado pelo software. Apenas servirá como instrução e definição de para que o script pode ser usado**.**

Em seguida usa-se os comandos clc e clear, com o intuito de limpar o executor onde se localiza o prompt e liberar memória do sistema.

O comando mprintf("Entre com a velocidade do móvel e o tempo do percurso \n") insere uma fala durante a execução do script, dando instruções e indicando as próximas etapas da execução. O \n indica que o próximo comando será executado na linha abaixo.

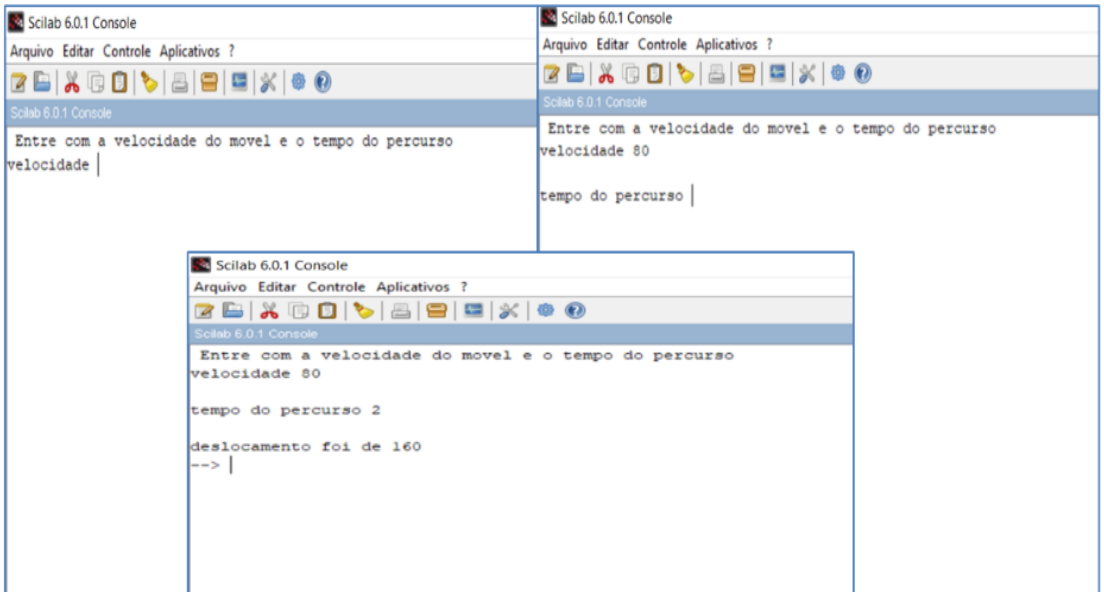
A seguir foram definidos dois fatores no problema, a velocidade(v) e o tempo(t). Note que ambas foram definidas com input( ). Isso é por que são

variáveis de possíveis problemas, logo serão determinados seus valores no momento de execução do programa. Ressaltamos aos alunos que observem sempre a unidade de medida. Se a velocidade for expressa em quilômetros por hora o tempo deve ser expresso em horas, ou seja, sempre na mesma unidade.

Em seguida foi escrita a equação que processa os dados inseridos e modela numericamente o fenômeno. Nesse caso foi definida a variável $d = v \times t$.

Para fechar o programa usou-se: mprintf("deslocamento foi de %g", d), onde o comando foi usado para um título à resposta final pro programa. Essa parte de texto sempre vem entre aspas(“”), exceto o sinal **%g** que indica que não deve ser expressa casas decimais no resultado. Se houver a necessidade de saber o valor das casas decimais esse comando deve ser trocado por **%f**. Após a vírgula vem a variável chave do problema, aquela que está se querendo conhecer o valor**.**

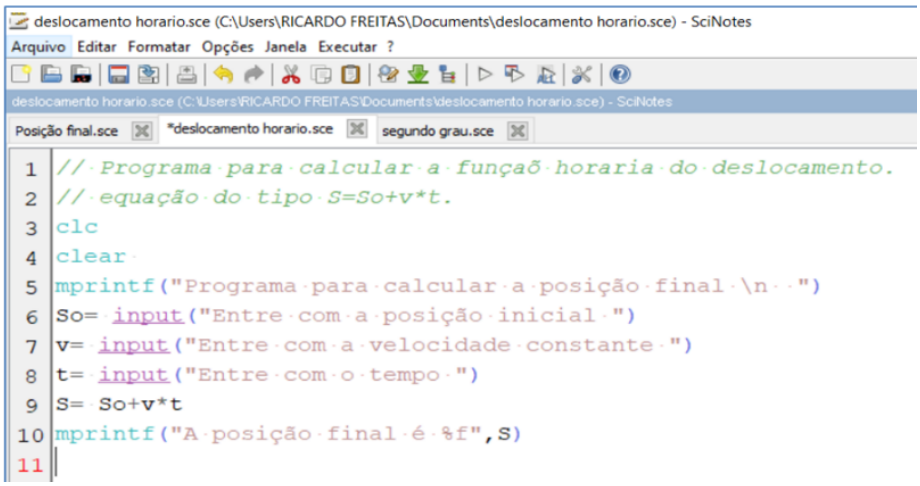
Figura 12



Fonte: o autor

Outro exemplo simples que foi construído passo a passo com os alunos é o da função horaria do deslocamento.

Figura 13



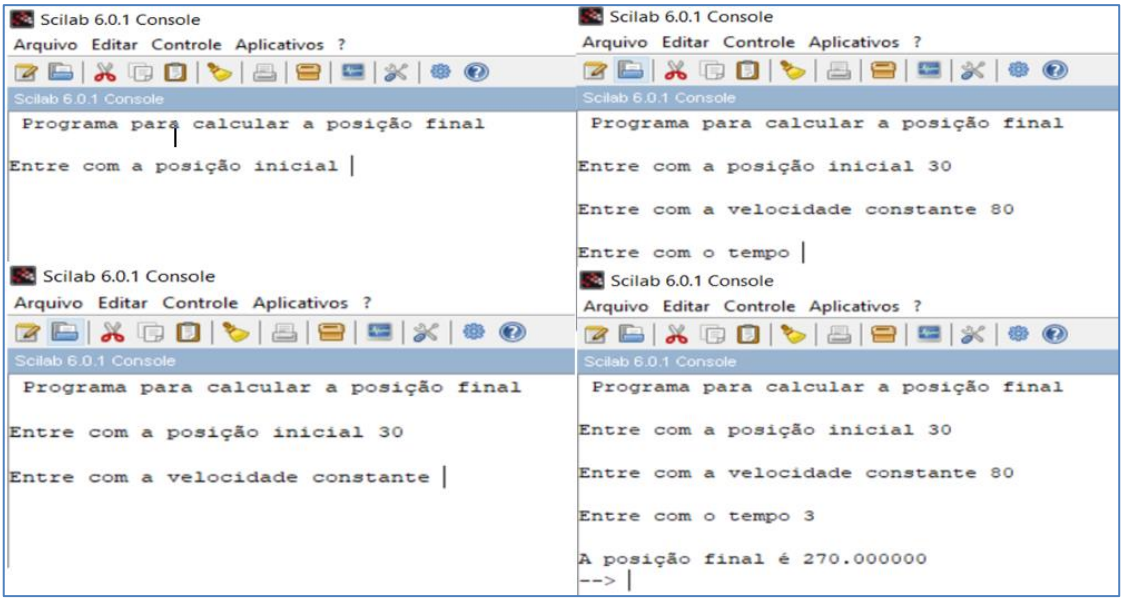
Fonte: o autor

Além dos comandos citados no exemplo anterior foram inseridos três variáveis com o comando de entrada de dados $\boldsymbol{input}(\text{""})$. Observe que dentro do parênteses do comando $\boldsymbol{input}(\text{""})$ é usado um comentário entre “”, para contextualizar qual é aquela variável. As variáveis usadas foram: S_0, posição inicial ; v, velocidade; t, tempo.

A equação usada para processar os dados de entrada nesse problema foi a definição da variável S, definida como: $\boldsymbol{S = S_0 + v \times t}$.

Foi usado o comando $mprintf()$, mas agora para imprimir o resultado de $\boldsymbol{S}$ e uma fala explicativa.

Figura 14

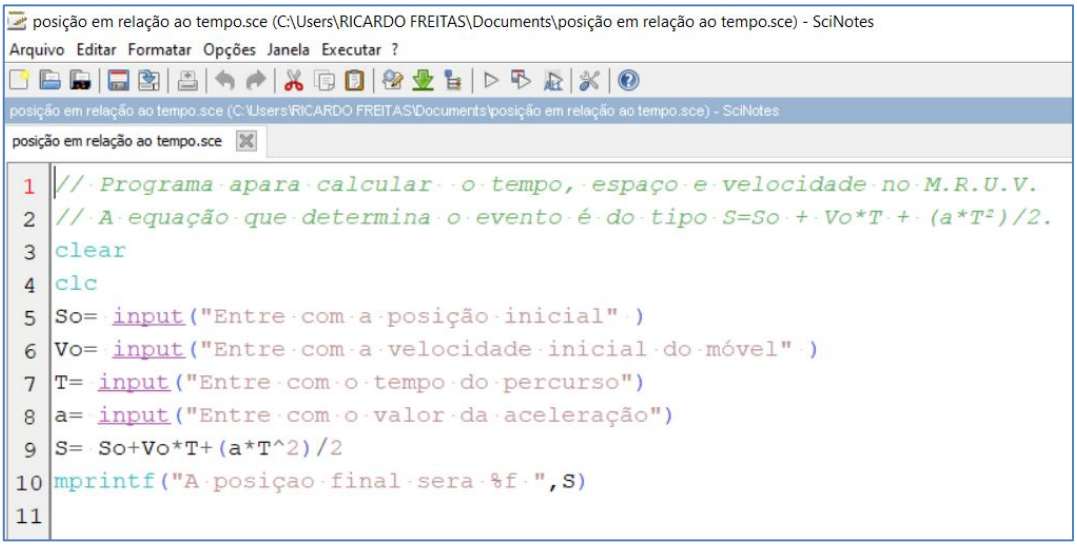


Fonte: o autor

Outro exemplo construído com os alunos foi sobre o estudo do Movimento Retilíneo Uniformemente Variado(M.R.U.V). Nesse momento construímos juntos no SciLab um programa para calcular a posição de um móvel em relação ao tempo. A equação utilizada foi:

$$S = S_0 + v_0 t + \frac{at^2}{2}$$

Figura 15

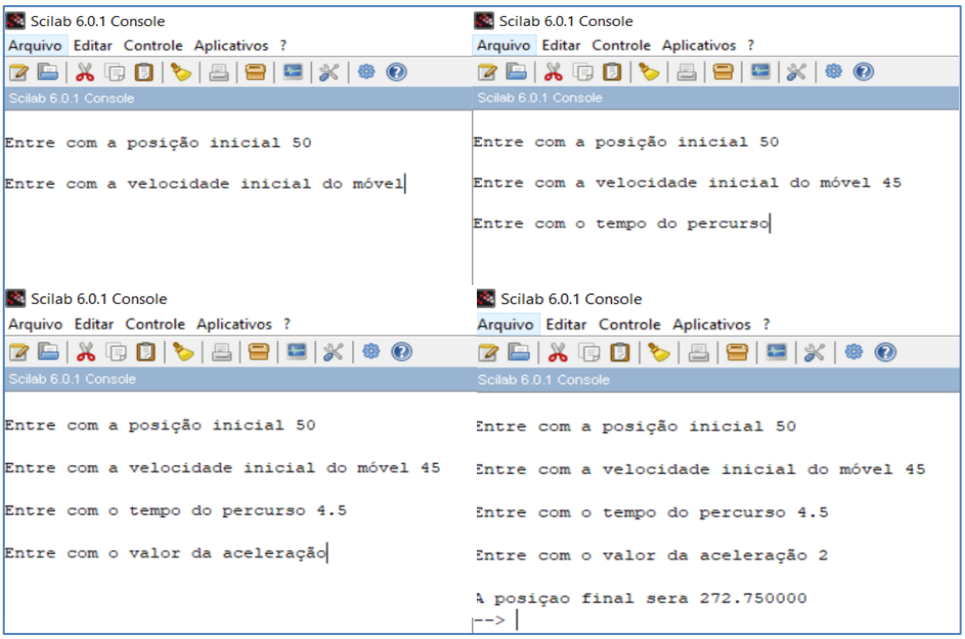


Fonte: o autor

Na construção desse programa pedimos aos alunos que observassem as variáveis do problema que poderiam ser inseridas no momento da execução do mesmo. Eles já foram capazes de entender e dizer onde usaríamos o comando $\boldsymbol{input}(\text{""})$.

Os alunos, analisando o problema, usaram o referido comando para os valores de; Posição inicial S_0; Velocidade inicial V_0; Tempo do percurso t e a aceleração a. O valor de S foi dado em função dos valores que eram condições do problema.

Figura 16



Fonte: o autor

Diversos programas podem ser criados ao vivo utilizando o projetor para servirem de base e exemplo para os alunos:

5.7 Gráficos no SciLab

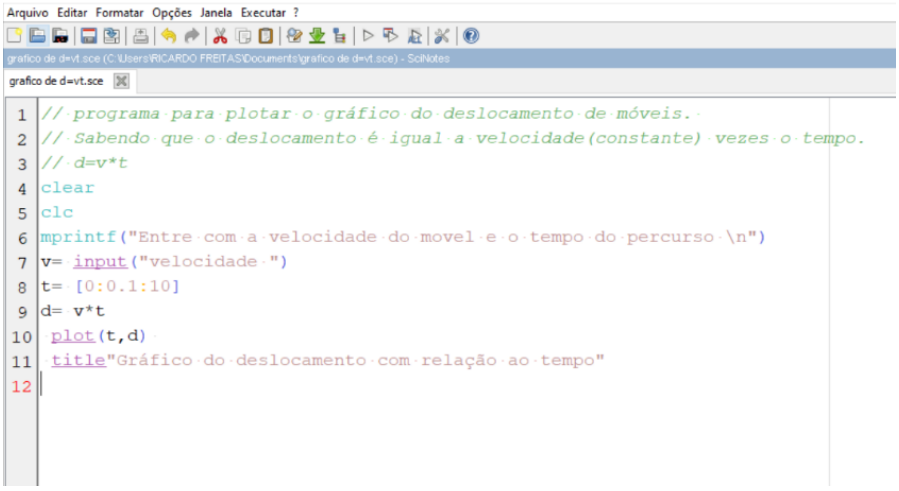
Após os alunos compreenderem os comandos de geração numérica da modelagem, também com uso do projetor ensinamos a plotagem dos gráficos dos fenômenos de Cinemática que os alunos acabaram de modelar. Esse passo a passo foi fundamental, pois o aluno pode agora ver os gráficos das variações de posição tempo e velocidade que eles estavam fazendo.

Para construir um script que plote um gráfico abre-se um novo documento no editor de arquivos.

Como exemplo inicial de modelagem construímos passo a passo o seguinte programa para plotar o gráfico do deslocamento de móveis.

Sabendo que o deslocamento é igual à velocidade vezes o tempo:

Figura 17



Fonte: o autor

O programa construído se inicia com o uso de "//", seguido do comentário pertinente ao caso**.**

Usa-se os comando $clc\ e\ clear$.

O comando $mprintf$("Entre com a velocidade do móvel e o tempo do percurso \n").

Na etapa seguinte é inserida uma variável com o comando de entrada de dados $\boldsymbol{input}$(""). A variável usada é a velocidade v.

Definimos o intervalo de tempo usando colchetes, como já visto.

$$t = [1{:}\,0.1{:}\,10]$$

Onde t é um intervalo de tempo de 0 a 10, com pontos de interação a cada décimo.

A equação que vai processar os dados de entrada nesse problema é a variável d que é definida como:

$$\boldsymbol{d = v \times t}.$$

O próximo comando inserido é o comando $plot(t, d)$, que ira imprimir um gráfico do deslocamento em função do tempo na janela gráfica do software.

Para adicionar um título ao gráfico usa-se o comando $title$"Gráfico d$o\ deslocameno\ em\ função\ do\ tempo$".

Figura 18

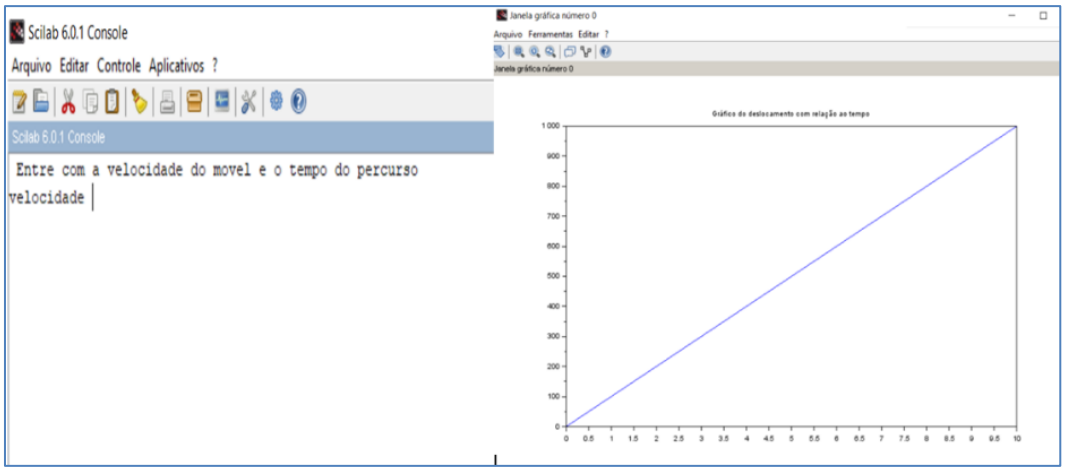
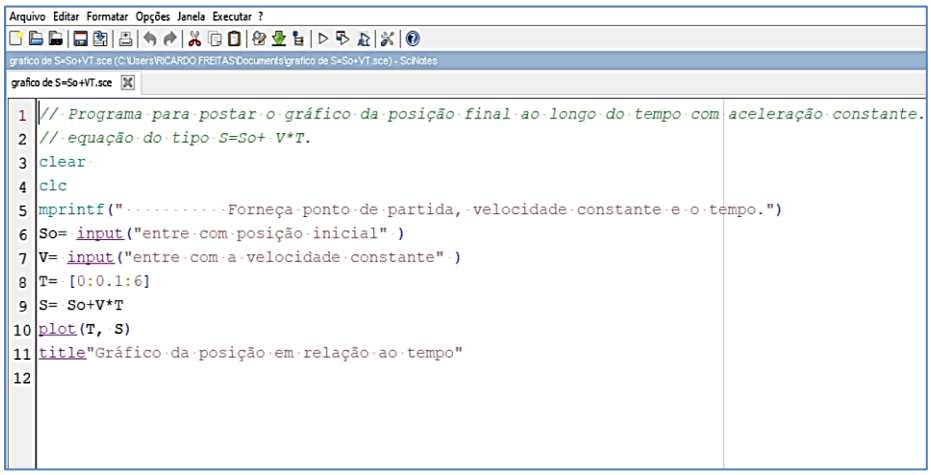


Figura: o autor

Outro exemplo usado para esclarecer os alunos sobre fenômenos de Cinemática foi o programa para plotar o gráfico da posição final em um intervalo de tempo com aceleração constante.

Figura 19



Fonte: o autor

O programa construído se inicia com o uso de “//”, seguido do comentário pertinente ao caso**.**

Usa-se os comando $clc\ e\ clear$.

Os comandos $\boldsymbol{mprintf}$("Forneça ponto de partida, velocidade constante e o tempo." \n")

So= input("entre com posição inicial")

V= input("entre com a velocidade constante")

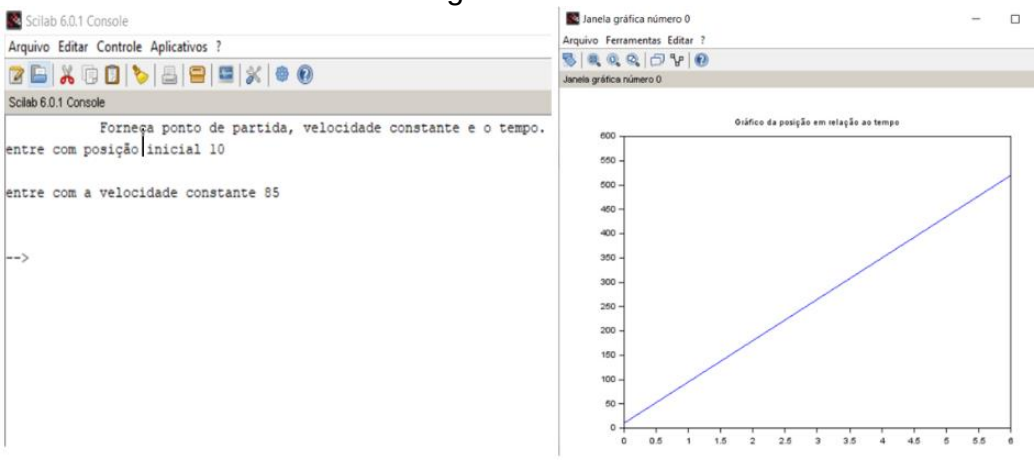
T= [0:0.1:6], intervalo de 0 a 6 com interação de 0,1.

S= So+V*T, equação que modela o fenômeno

plot(T, S), para plotar o gráfico da posição em função do tempo

title"Gráfico da posição em relação ao tempo"

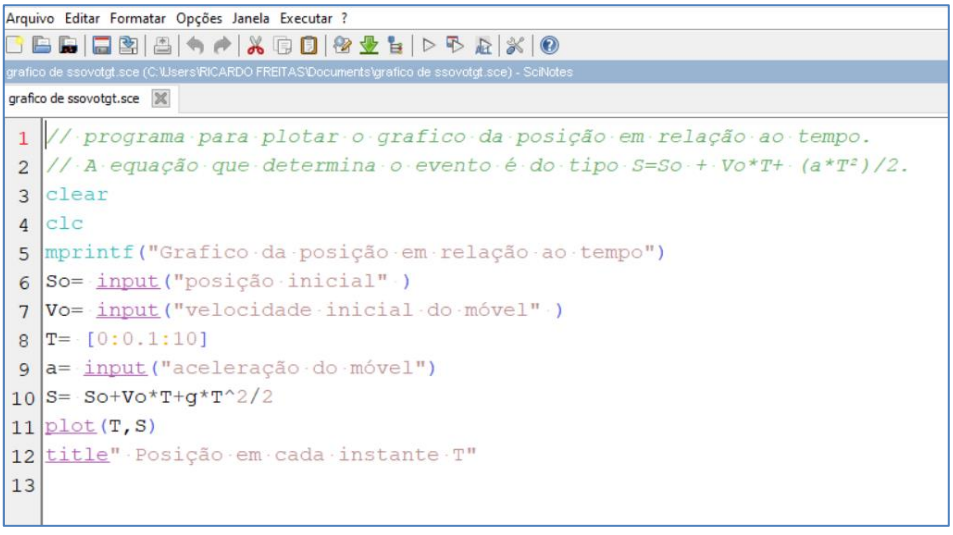
Figura 20



Fonte: o autor

Outro Programa construído com os alunos foi para plotar o gráfico da posição em relação ao tempo com variação de aceleração.

Figura 21



Fonte: o autor

O programa construído se inicia com o uso de “//”, seguido do comentário pertinente ao caso**.**

Usa-se os comando $clc\ e\ clear$.

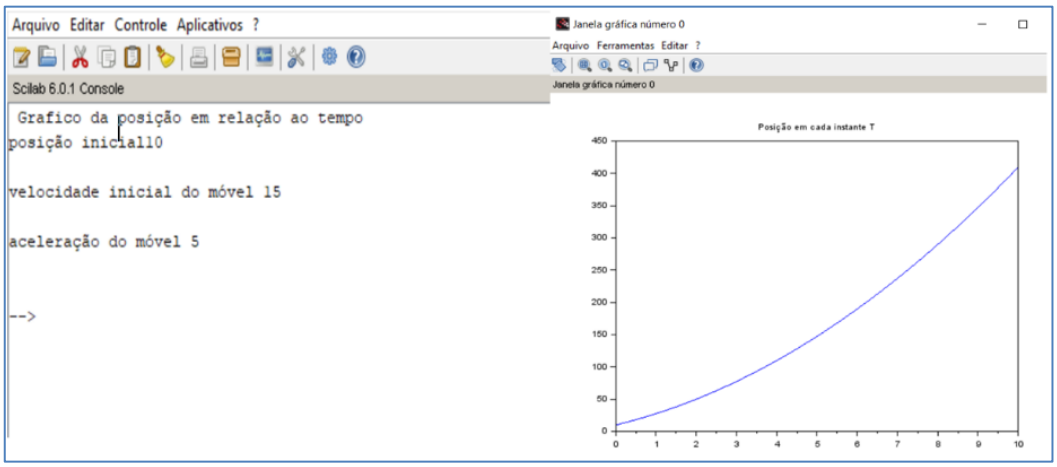
Os comandos:

mprintf("Grafico da posição em relação ao tempo")

So= input("posição inicial")

```
Vo= input("velocidade inicial do móvel" )
T= [0:0.1:10]
a= input("aceleração do móvel")
S= So+Vo*T+g*T^2/2
plot(T,S)
title" Posição em cada instante T"
```

Figura 22



Fonte: o autor

Problemas diversos de física no campo da Cinemática podem ser modelados e resolvidos através dessas programações, bastando para tanto o aluno entender a lógica usada para escrever os presentes modelos. O aluno, ao passo que aprende a manipular as equações isolando as variáveis necessárias para resolver cada situação problema utilizando o software, avança no conhecimento da física envolvida no processo.

O professor tem a opção de mostrar o uso do SciLab utilizando exemplos bastante simples que envolvam movimento e passar a introduzir questões mais complexas à medida que os alunos forem se envolvendo na utilização de programação. Podemos resolver questões de vestibulares conceituados no pais.

Pedimos aos alunos que baseado no que viram até ali utilizassem o software para modelar fenômenos da física que envolvem movimentos com velocidade constante ou aceleração constante.

Diversos programas podem ser criados pelos alunos com o auxílio e orientação do professor responsável durante seu trabalho com o tema de Cinemática nas Séries do Ensino Médio. No Apêndice II podemos encontrar vários modelos sugeridos e construídos pelo autor para auxiliar nesta tarefa.

5.8 Verificação e Análise dos Resultados

Esta seção é destinada à coleta e tratamento de informações para verificar a satisfação de aprendizagem e evolução do aluno com o uso da ferramenta Programação em SciLab para o ensino de Cinemática. Importante salientar que durante o Minicurso levamos sempre em consideração os seguintes fatores: compreensão dos fenômenos abordados e interesse e participação ativa do aluno. Para tanto, ao final da realização do minicurso foram feitas as seguintes perguntas aos alunos:

1°) Quais equações você utilizou na programação?

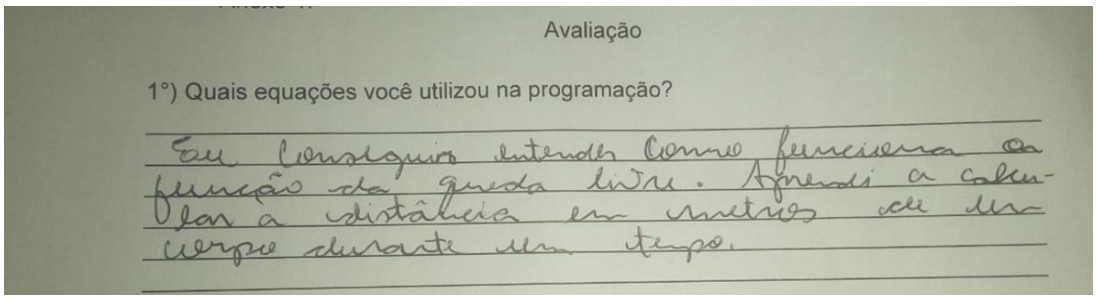
2°) Nessas equações o que estava variando em função de que respectivamente?

3°) Qual sua avaliação pessoal sobre a utilização do software de programação numérica SciLab na aprendizagem de física?

Como houve a participação durante o minicurso do Professor de Física da turma, lhe fizemos o seguinte questionamento: O produto utilização de programação no ensino de Física mostrou-se um bom recurso didático?

A experiência permitiu aos discentes um novo conhecimento no campo da TIC, pelo fato de ter possibilitado sua aproximação com um recurso de modelagem computacional de um conteúdo dito como mecânico dentro do estudo de Física. No primeiro questionamento feito, deixam claro um entendimento melhor sobre aplicações de conceitos de Física, o que já representa um grande diferencial proporcionado pelo processo de modelagem.

Figura 23

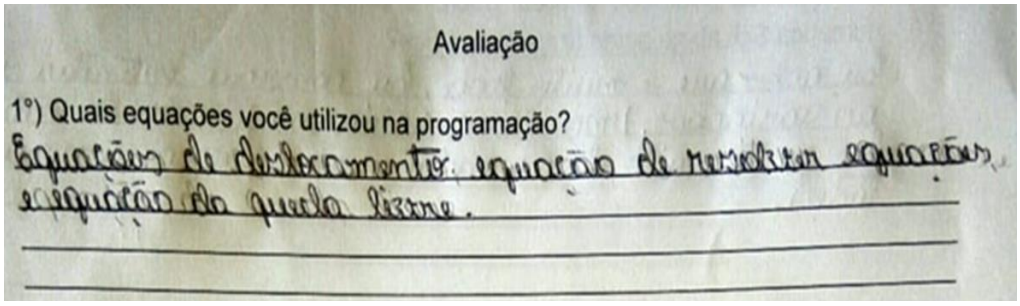

Avaliação

1°) Quais equações você utilizou na programação?

Eu consegui entender como funciona a função da queda livre. Aprendi a calcular a distância em metros de um corpo durante um tempo.

Fonte: o autor

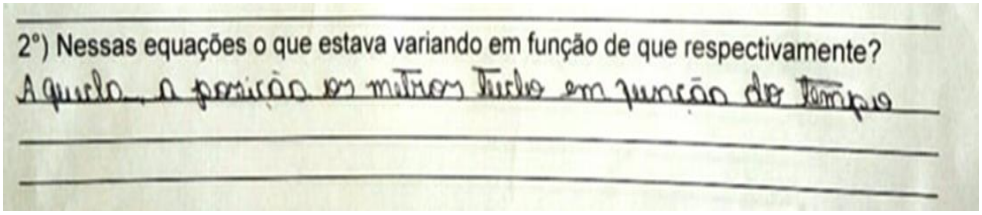
Figura 24



Fonte: o autor

Em um segundo questionamento feito, procuramos identificar se os alunos conseguiam fazer relações entre as variáveis envolvidas no processo de modelagem, e se pode perceber que, no processo de construção dos modelos, ele sente a necessidade saber identificar as variáveis e as relações de grandezas, demonstrando sua compreensão em torno dos problemas de Cinemática que ele manipulou durante o mini curso.

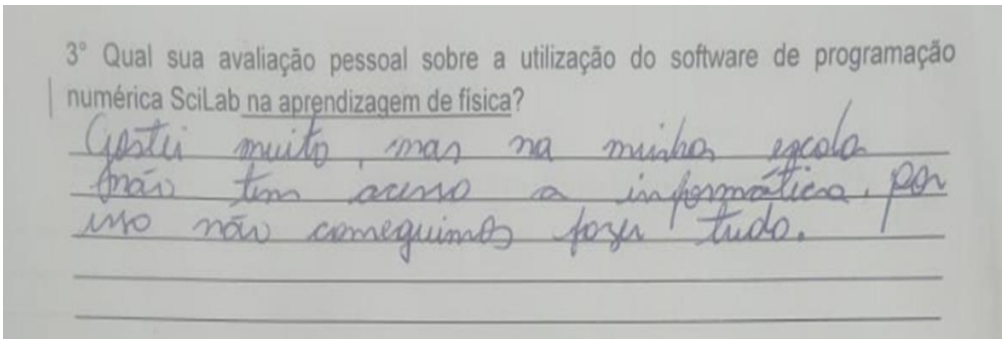
Figura 25



Fonte: o autor

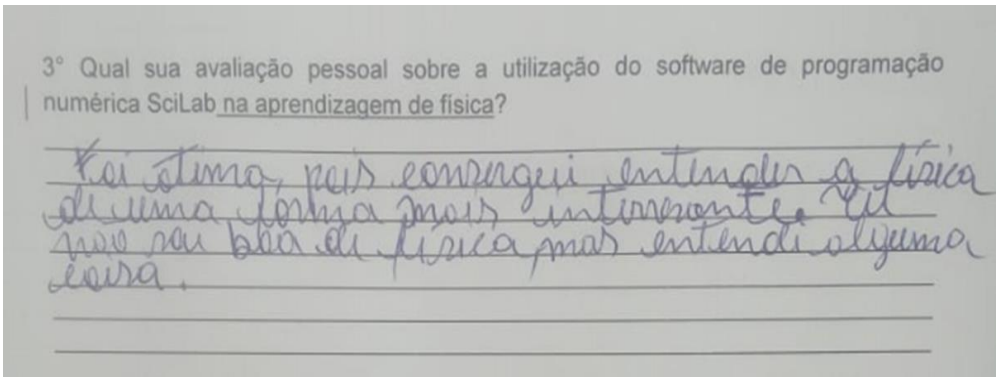
Identificamos ainda as dificuldades, adaptações e empatia dos alunos ao uso do método computacional. A partir da compreensão de sua realidade, entende-se a ausência de ligação entre práticas escolares e uso de TIC, como também a ausência de estratégias elaboradas no âmbito escolar para promover essa interação. Isto pode se verificar em algumas respostas sobre o uso do software:

Figura 26



Fonte: o autor

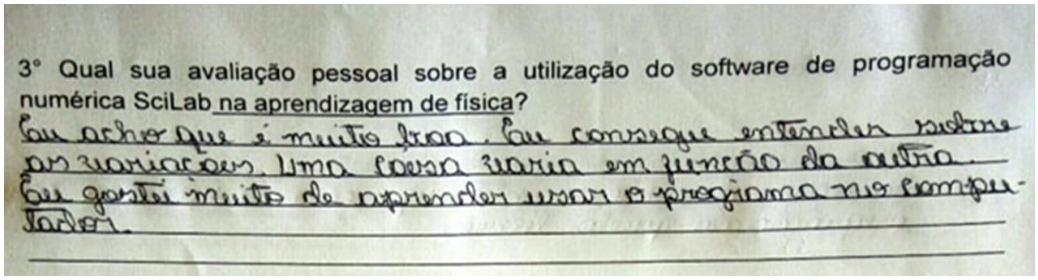
Figura 27



Fonte: o autor

Em geral as respostas mostraram como os alunos ficaram satisfeitos quanto ao nível de aprendizagem dos fenômenos de física modelados durante o minicurso.

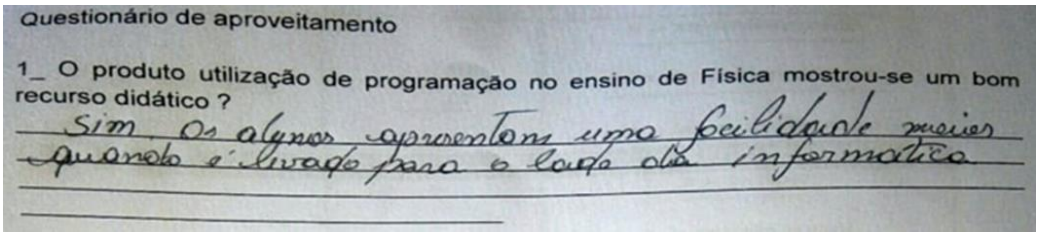
Fígura 28



Fonte: o autor

O professor de física no Ensino Médio que acompanhou a implementação do produto educacional relatou ter percebido entusiasmo nos alunos e uma melhora substancial no nível de compreensão dos mesmos sobre o tema.

Figura 29



Fonte: o autor

O uso informatica, segundo o relato, trás uma novidade, desperta o interresse do aluno para o que está se fazendo na sala de aula, logo, compreendemos ser proveitoso o ensino do conteudo abordado atraves do uso da programação em SciLab.

CAPITULO 6: CONSIDERAÇÕES FINAIS

O presente trabalho traz dados que apontam para a necessidade de uma inovação nos métodos de ensino e aprendizagem da educação nacional. A sociedade está em quase todas as partes do mundo cada vez mais tecnológica, mais conectada e rápida no tráfego de informações e comunicação. A educação, não somente no tocante ao ensino de Física, precisa acompanhar a revolução tecnológica dos dias atuais, através da criação de métodos inovadores, incorporar a tecnologia à vivencia de sala de aula.

Como vimos no trabalho, há uma dificuldade nas escolas em dar o suporte necessário para viabilização do acesso por parte dos alunos a recursos tecnológicos. A tecnologia pode ser uma grande aliada no processo de ensino e aprendizagem dos educandos. Ela tem o potencial de despertar no aluno a curiosidade e o interesse. Utilizando para isso ferramentas como jogos digitais, simulações computacionais, realidade aumentada, uso de aplicativos desenvolvidos para Android ou IOS, possibilidades de diversos recursos multimídia como vídeo aulas disponíveis na internet, e a programação computacional, como vista e proposta neste trabalho, entre diversas outras. Para irmos nessa direção é necessário possibilitar aos alunos o ambiente favorável, que consiste no acesso a recursos tecnológicos e a materiais elaborados para sua utilização no processo de aprendizagem. Tem-se com isso, a curto prazo, a possibilidade de o discente absorver uma gama de conhecimento que nenhuma outra geração experimentou.

O presente trabalho teve como objetivo elaborar um produto para o ensino de Cinemática utilizando como recurso o uso de programação computacional que desperte no aluno o interesse necessário para que o mesmo possa construir o próprio saber, dizendo o que o computador deve efetuar, verificando, com isso, a Teoria do Construcionismo de Papert. O discente pode assumir o protagonismo no desenvolvimento do seu processo de aprendizagem. A modelagem utilizando programação computacional pode trazer ao aluno uma visão concreta de fenômenos que outrora poderiam parecer muito abstratos, pode construir uma ponte entre o conhecimento de Cinemática e o próprio cotidiano do aluno.

O produto foi testado com alunos da rede pública estadual na cidade de Marabá PA, e apresentou resultados satisfatórios quanto à aprendizagem dos alunos ao participarem da atividade oferecida, bem como pelo nível de interesse dos mesmos. O trabalho levou em consideração sempre uma análise qualitativa

dos dados coletados, dado que uma das principais incógnitas levantadas ao longo do mesmo é a falta de interesse dos alunos no tipo de aulas que temos atualmente.

No embasamento teórico trouxemos ao trabalho a teoria de Papert na qual o aluno atua como construtor do próprio conhecimento, sendo ativo no processo de aprendizagem e não passivo esperando que o professor lhe encha de informações. Ainda como base teórica do trabalho apresentamos algumas definições e equações fundamentais no estudo da Cinemática, que estuda os movimentos de corpos ou partículas sem considerar as forças que agem sobre ele.

O trabalho foi ministrado em etapas de forma que inicialmente foi introduzido o tema programação computacional para os alunos através de uma palestra sobre linguagens de programação. Sequencialmente veio a etapa de contextualização, onde mostramos a área principal do software e seus comandos mais básicos. A parte posterior foi a socialização e ambientação, na qual os alunos foram incentivados a inicializarem o software SciLab e usarem livremente recursos como operações simples. Nessa mesma etapa mostramos o layout do software, funções essenciais do mesmo e mostramos a execução de alguns programas simples feitos no editor de scripts. Essas foram as partes preparatórias para a junção do uso de programação computacional com o estudo de Cinemática. A partir daí trouxemos o assunto de modelagem por programação para o conteúdo de Física abordado.

Os alunos criaram programas que modelam numérica e graficamente movimentos no estudo de Cinemática. Durante cada explicação sobre os tipos de movimentos que seriam imediatamente modelados por um programa, a atenção de todos os participantes foi constante, eles, naquele momento, desejavam aprender sobre aquelas equações que descreviam cada tipo movimento, pois cada equação seria utilizada no script que o aluno estava desenvolvendo. O aluno se sentia instigado a fazer o programa dar certo.

A utilização de programação computacional em SciLab segundo os indicativos do trabalho tem um potencial elevado na possibilidade de melhora no processo de ensino e aprendizagem do conteúdo de Cinemática, dado que o aluno passa a interagir ativamente na construção do próprio conhecimento, bem como para qualquer outro assunto da Física. De certo, como o processo de modelagem computacional, é possível se obter uma melhora significativa no nível de entendimento e correlação das definições físicas, e que se queira ainda nas

relações com o cotidiano do aluno, diminuindo a falsa impressão de que os conceitos físicos sejam abstratos e sem emprego no dia a dia.

Como uma perspectiva futura indicamos que o material aqui produzido, em software livre e gratuito, possa ser adaptado para outras áreas do conhecimento de Física, pois seu manejo uma vez compreendido pelo aluno pode o ajudar a compreender e interpretar qualquer conteúdo de Física do Ensino Médio, através da modelagem dos fenômenos.

BIBLIOGRAFIA

BRANDT, C. F., BURAK, D., and KLÜBER, T. E., orgs. Modelagem matemática: perspectivas, experiências, reflexões e teorizações [online]. 2nd ed. rev. and enl. Ponta Grossa: Editora UEPG, 2016, 226 p. ISBN 978-85-7798-232-5. Available from: doi: 10.7476/9788577982325. Also available in ePUB from: http://books.scielo.org/id/b4zpq/epub/brandt-9788577982325.epub.

BRASIL. Ministério da Educação. Secretaria de Educação Média e Tecnológica. Orientações Educacionais Complementares aos Parâmetros Curriculares Nacionais (PCN+): Física.Brasília: MEC/SEMTEC, 2006. Disponível em: <http://portal.mec.gov.br/seb/arquivos/pdf/CienciasNatureza.pdf>. Acesso em: 29 abr. 2018.

BRASÃO, Mauricio dos Reis. LOGO – UMA LINGUAGEM DE PROGRAMAÇÃO VOLTADA PARA A EDUCAÇÃO. em :< http://www.fucamp.edu.br/wp-content/uploads/2010/10/5%23U00c2%23U00ba-MAUR%23U00c3%23U008dCIO-DOS-REIS-BRAS%23U00c3%23U0192O1.-LOGO.pdf> acesso em 05/01/2019.

COVO ,Carlos Cesar de Carvalho; Modelagem Matemática e Computacional de Efeitos em Ondas Sonoras. Disponível em:< https://docplayer.com.br/26933567-Modelagem-matematica-e-computacional-de-efeitos-em-ondas-sonoras-por.html> . Acesso em 20/11/2008.

RESNICK, R. HALLIDAY, D. e KRANE, K. S. **Física 1**, 5º edição, v. 1, Rio de Janeiro: LTC, 2008, 390p

JOSEPETTI, Daniela; LOPES, Mariana; Fernanda, NICOLE; MICHELON, Vinicios; ALMEIDA, Wellick. **SCILAB NÍVEL BÁSICO AO INTERMEDIÁRIO, 2015,** disponível em: <SciLab Nível Básico ao Intermediário> acessado em 27/12/2018.

MOREIRA, Elaine Ribeiro. Metodologias Ativas para o Ensino de Física: criando uma webquest para o ensino das leis de newton e suas aplicações. Dissertação de Mestrado. MNPEF-UNIFESSPA, 2019.

NUNES, Sergio da Costa; SANTOS, Renato Pires dos. O Construcionismo de Papert na criação de um objeto de aprendizagem e sua avaliação segundo a taxionomia de Bloom. Disponível em: < http://www.fisica-interessante.com/files/artigoconstrucionismo_papert_objeto_de_aprendizagem.pdf >. Acesso em: 20/12/2018.

PCNs-Física-Disponível em: http://www.sbfisica.org.br/arquivos/PCN_FIS.pdf. Acesso em 12/01/2018.

PEZZINI, Clenilda Cazarin. SZYMANSKI, Maria Lidia Sica. FALTA DE DESEJO DE APRENDER: Causas e Consequências. Disponivel em:< http://www.diaadiaeducacao.pr.gov.br/portals/pde/arquivos/853-2.pdf> acesso em 14/05/2109.

SILVA, Ione de Cássia Soares. PRATES, Tatiana da Silva. RIBEIRO, Lucineide Fonseca Silva. **As Novas Tecnologias e aprendizagem**: desafios enfrentados pelo professor na sala de aula. Disponível em: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=2ahUKEwi2nbK_q-biAhU9HrkGHZ2kB-oQFjAAegQIAxAC&url=https%3A%2F%2Fperiodicos.ufsc.br%2Findex.php%2Femdebate%2Farticle%2Fdownload%2F1980-3532.2016n15p107%2F33788&usg=AOvVaw0KGp-6N0Dku6GnGVOPeGMi Acesso em 26/02/2019.

STINGHEN, Regiane Santos. TECNOLOGIAS NA EDUCAÇÃO: DIFICULDADES ENCONTRADAS PARA UTILIZÁ-LA NO AMBIENTE ESCOLAR FLORIANÓPOLIS, SC. 2016. Disponível em :< https://repositorio.ufsc.br/bitstream/handle/123456789/169794/TCC_Stinghen.pdf?s>
Acesso em 28/02/2019.

OLIVEIRA, Cláudio de. MOURA, Samuel Pedrosa. TIC'S NA EDUCAÇÃO: A UTILIZAÇÃO DAS TECNOLOGIAS DA INFORMAÇÃO E COMUNICAÇÃO NA APRENDIZAGEM DO ALUNO. Disponível em:< http://periodicos.pucminas.br/index.php/pedagogiacao/article/viewFile/11019/8864

TOLEDO, Bruno de Souza. O USO DE SOFTWARES COMO FERRAMENTA DE ENSINO-APRENDIZAGEM NA EDUCAÇÃO DO ENSINO MÉDIO/TÉCNICO NO INSTITUTO FEDERAL DE MINAS GERAIS. Disponível em: < http://www.fumec.br/revistas/sigc/article/view/3163>. Acesso em 25/02/2019.

VIEIRA**,** Zacarias Nascimento de lima. Informática na Educação. 2006, disponível em:http://www.avm.edu.br/monopdf/31/ZACARIAS%20NASCIMENTO%20DE%20LIMA%20VIEIRA.pdf, acessado 27/12/2018.

WIRTH, N. Algoritmos e Estruturas de Dados. LTC Informática-Programação, 1989.

YOUNG, H. D. e FREEDMAN, R. A. Física (Pearson Education do Brasil, São Paulo, 2009), v. 1, 12ª ed.

APÊNDICE I – PRODUTO EDUCACIONAL

MNPEF
Mestrado Nacional Profissional em Ensino de Física

UNIFESSPA
UNIVERSIDADE FEDERAL DO SUL E SUDESTE DO PARÁ

INSTITUTO DE CIÊNCIAS EXATAS

PROGRAMA NACIONAL DE MESTRADO PROFISSIONAL EM ENSINO DE FÍSICA

UTILIZAÇÃO DE PROGRAMAÇÃO EM SCILAB NO ENSINO DE FÍSICA: MODELANDO FENÔMENOS DE CINEMÁTICA

PROF. JOSÉ RICARDO DOS SANTOS FREITAS

1. Apresentação

O presente trabalho traz o modelo de implementação de um produto educacional para o ensino de Física desenvolvido no âmbito do curso de Mestrado Nacional Profissional no Ensino de Física Polo 29, na Universidade Federal do Sul e Sudeste do Pará – UNIFESSPA, Campus II, Marabá-PA. O produto educacional mostrado a seguir versa sobre a utilização de uma linguagem de programação computacional como ferramenta didática para o ensino de fenômenos da Cinemática.

O produto educacional a seguir é proposto sob a ótica do Construcionismo de Papert, que visa possibilitar que o aluno seja construtor do próprio conhecimento com auxílio de Tecnologias de Informação e Comunicação - TIC no processo de ensino e aprendizagem. Nesta perspectiva, vislumbra-se que com a utilização de linguagens de programação, o aluno é quem diz como e o quê o computador deve fazer, tendo o professor como mediador deste processo.

O produto apresentado foi desenvolvido para ser utilizado com alunos do Ensino Médio, mas também pode ser trabalhado com alunos do 8º ano do Ensino Fundamental e está organizado de uma maneira sequencial lógica; o que fazer, como fazer e resultados.

Primeira Etapa:

Na primeira etapa é feita a introdução ao tema programação computacional, feita através de uma apresentação com data show a respeito das linguagens tradicionais de programação. Em seguida será apresentado o software SciLab onde serão mostrados o layout, inicialização e principais partes do software. O SicLab está disponível gratuitamente com versões para Windows; Vista, 7, 8, 10. GNU / Linux e Mac OS X no endereço oficial eletrônico <https://www.scilab.org/download/6.0.2>.

Na sequência serão mostrados a execução de comando simples no console e após isso mostrados os principais comandos de entrada, saída e interação dos dados. Após esse passo os alunos poderão construir alguns exemplos de modelagem por programação em SciLab.

Segunda Etapa:

Esta etapa compreende que o aluno deva utilizar as instruções vistas até o momento para escrever scripts que descrevam numérica ou graficamente os fenômenos do campo da cinemática solicitados pelo professor.

Terceira Etapa:

A última etapa é constituída pela avaliação que o professor deve fazer com a turma sobre o grau de aprendizado e satisfação com a utilização do produto aplicado.

2. Primeira Etapa

Será apresentado a seguir orientações para elaboração dos slides de motivação e apresentação das linguagens de programação pelo Professor.

A turma deve ser inserida no contexto de programação, a fim de despertar o interesse e aumentar o grau de conhecimento na área, através de uma apresentação que pode ser na forma de slides de modo simplificado como veremos a seguir.

Texto de motivação:

Introdução à Programação

Os computadores são funcionários quase perfeitos, pois eles executam com certa precisão aquilo que são mandados fazer. No entanto eles falam uma linguagem diferente da nossa, ou seja, da linguagem humana, denominada de alto nível. Considerando que o processador do computador fala em linguagem binária, isto dificulta enormemente a comunicação, entre homens e máquinas, nessa linguagem, pois o homem teria que decorar milhares de sequências de códigos binários.

Por isso os programadores desenvolveram as linguagens de programação que são conjuntos de padrões e comandos usados para facilitar a comunicação entre homem e máquina. As linguagens de programação usadas pelos programadores passam por um compilador, que é usado para transformar os códigos escritos na linguagem de programação em uma linguagem que o processador entenda.

A programação de computadores tem linguagens tradicionais, ou seja, mais utilizadas por programadores ao longo do tempo, como Fortran, Pascal, C, C++, Mathlab, Phayton e o SCilab, esta última bastante popular por sua facilidade de uso. É importante conhecermos um pouco daquelas que podemos configurar como as principais, dentre estas.

O Fortran é uma das primeiras linguagens desenvolvidas. Ela inovou muito ao trazer comandos e funções prontas para serem compiladas. A linguagem **C** combina especificidades das primeiras linguagens com a evolução de linguagens mais modernas, de alto nível. Essa linguagem é ainda muito utilizada. O **C++** é um aperfeiçoamento da linguagem **C** mantendo suas características só que aumentando seu nível e simplificando a sua escrita pelo programador.

Python é uma linguagem moderna e simples muito utilizada por programadores iniciantes. Com ela pode-se facilmente construir blocos lógicos usados em jogos.

Uma linguagem muito usada na academia é o MatLab, que se trata de um poderoso software numérico e gráfico utilizado em diversas tarefas no campo da programação e da modelagem. Sua principal característica é a facilidade de escrita dos seus scripts. Esse software tem um layout simples e agradável.

Outro software de programação numérica e gráfico bastante utilizado para fins acadêmicos é o SciLab, suas principais características são: ser um programa aberto, livre, ou seja, gratuito e de fácil manuseio pelo usuário, não sendo necessário que seja um programador experiente. A partir do momento que se aprende a usar suas ferramentas ele pode se tornar um poderoso aliado na aprendizagem do aluno.

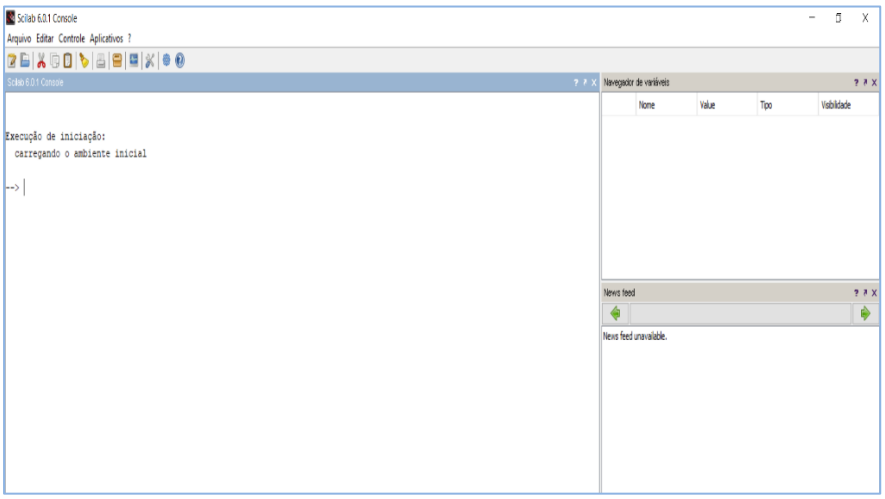
Esta última linguagem de programação, dadas suas caraterísticas, é a que apresentaremos como ferramenta para modelagem no ensino de Física, em particular, para resolução de problemas de Cinemática.

3. Segunda Etapa

3.1 Conhecendo o ambiente de trabalho do Scilab

A segunda etapa consiste em mostrar como funciona o Software SciLab. Então, após a instalação nos computadores o professor inicia apresentando o Executor de comandos chamado "*prompt"* (-->).

Figura 01



Fonte: o autor

Os alunos devem ser incentivados a analisar e descobrir que o "*prompt*" é o executor de programas numéricos do software e também pode ser utilizado com uma calculadora numérica avançada. Eles podem executar operações aritméticas no "*prompt*" como nos exemplos a seguir:

1) Raiz quadrada de 81:

$--> sqrt(81)$, tecle enter, e se obtém como resposta

$$ans =$$

$$9.$$

2) 12 elevado a 7:

$$--> 12\text{^}7$$

$$ans =$$

$$35831808$$

O professor pode deixar os alunos livres para usarem o Scilab com uma calculadora realizando outras operações aritméticas, isto ajudará no início do processo de ambientação.

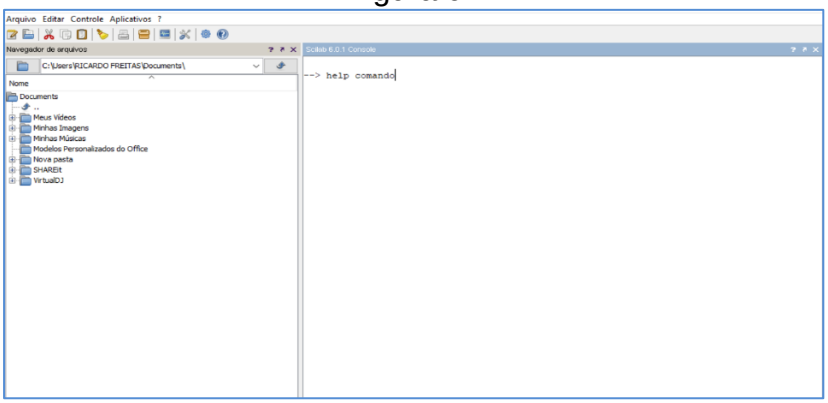
3.2 AMBIENTAÇÃO

O próximo passo é a ambientação, onde os alunos são incentivados a abrirem o editor de scripts no canto superior esquerdo do console. Mostrar o editor de programas do SciLab, onde os programas que executam funções são escritos. Professor irá passar comandos básicos de entrada e saída de comandos no software. Um tutorial com os comandos está disponível no endereço oficial em: https://help.scilab.org/docs/5.3.0/pt_BR/index.html.

3.3 Principais comandos

Umas das funções elementares do Scilab é um sistema de auxílio do software chamado "*help*". Que permite ao usuário receber ajuda sobre como utilizar qualquer comando a ser inserido na área de execução de tarefas, que é o "*prompt*", bastando o usuário entrar com "*help + comando desejado*". O "*help*" sem argumentos fornece a página de hipertextos dos capítulos de ajuda, que é fundamentalmente útil.

Figura 02



Fonte: o autor

Várias Variáveis podem ser definidas no *console* ou no editor. Essas variáveis são reconhecidas e gerenciadas por caracteres pelo software. Uma letra minúscula é diferente da mesma letra maiúscula e qualquer uma delas pode ser

substituída por um novo valor a partir do momento que o usuário a redefina com um valor diferente. Na linguagem Scilab o (;) é usado para suprimir o valor (resposta) de uma variável ou de uma expressão.

A variável (*ans*) é utilizada para expressar resultados de expressões ou variáveis não previamente definidas.

Ex.: Atribuímos a variável $\boldsymbol{a}$ o valor 6, mas ao executarmos no prompt $A + 9$, não teremos resposta definida, pois a variável $\boldsymbol{A}$ é diferente da variável $\boldsymbol{a}$. Porém, ao ser executado $a + 9$, obtemos a resposta $ans = 15$, pois foi definido o valor 6 para variável $\boldsymbol{a}$ como pode ser vista a seguir:

Figura 03

Fonte: o autor

No console ou no editor de programas o comando duas barras "$//$" é utilizado para adicionar um texto explicativo ou um comentário sobre o que se pretende fazer. Este texto após duas barras não será executado pelo software.

A tecla ↑ é utilizada no *"$prompt$"* para chamar o comando mais recente executado. Dai em diante as teclas ↑ e ↓ navegam pelos comandos recém executados no *"prompt" e as teclas* ← *e* → movem o cursor entre os caracteres do comando.

Para limpar a área de trabalho do console é utilizado o comando *"clc"*. Outro comando importante é o *"$tohome$" usado* para posicionar a área de trabalho com cursor do *"$prompt$"* no canto superior esquerdo.

Na linguagem Scilab podem ser inseridas três formas distintas de constantes: numéricas, literais e lógicas.

A numérica é formada por caracteres numéricos munidos de um ponto decimal, se o número for decimal, tendo necessariamente o sinal (-) caso seja negativo. A potência de dez é indicada por qualquer uma das letras $\{e, E, d, D\}$ seguida de outros caracteres numéricos, que indicam o fator em potência de 10 pelo qual o número está multiplicado, por exemplo, $17D5$, que indica 17×10^5, ou $97e11$ que indica 97×10^{11}.

No tratamento de números complexos o Scilab oferece uma facilidade não muito comum nos softwares de programação. Ele trabalha com complexo do mesmo modo que trabalha com números reais, basta acrescentar a indicação $(* \%i)$ ao final da parte imaginaria dos números complexos.

Além do comando *"help",* que é utilizado para consultar um comando do Scilab, há também outra importante ferramenta de consulta no software, o *"apropos",* usado para pesquisar sobre um argumento ou operação, como é escrito o comando no Scilab.

O software mostra como utilizar cada comando e o que eles representam, facilitando assim o uso pelo programador.

3.4 **Expressões Aritméticas**

No Scilab para realizar expressões aritméticas são usados os seguintes operadores.

Tabela 01

Operação	*Operador*	*Uso*
adição	+	$a + b$
subtração	−	$a - b$
multiplicação	*	$a * b$
Divisão	/	a/b
Potência	^	$a\hat{}b$
Radiciação	*sqrt*()	*sqrt*(*a*)

Fonte: o autor

3.5 Construindo e executando um SCRIPT.

Para construir um script abre-se um novo documento no editor de arquivos (fig. 04). Em seguida o arquivo deve ser nomeado e salvo em um local acessível ao software. As ideias fundamentais de programação são três; algoritmo de entrada de dados, algoritmo de processamento de dados e algoritmo de saída.

No SciLab usaremos basicamente os seguintes comandos de entrada, saída e processamento de dados:

$$\boldsymbol{clc}$$

Para limpar o executor de arquivos antes da nova execução

$$\boldsymbol{clear}$$

Função Matlab Remover itens da área de trabalho, liberando memória do sistema.

$$\boldsymbol{mprintf}(\text{" "})$$

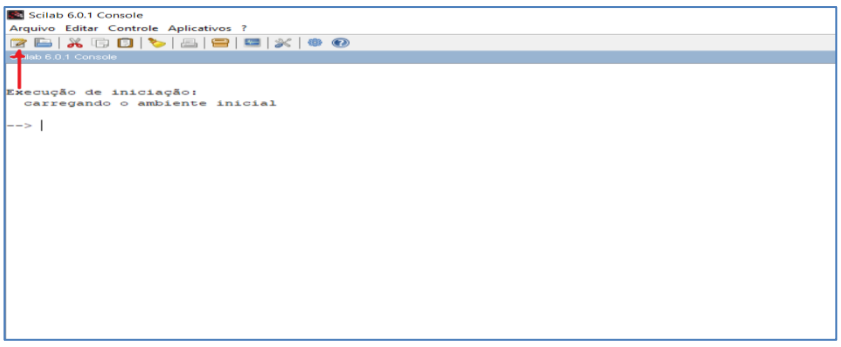
Função de saída. Imprime um texto e/ou o valor resultado de alguma variável definida no script executado.

$$\boldsymbol{input}(\ )$$

Funções de entrada/saída: Fornece ao programa executado o “prompt” para a inserção de valores a alguma variável do problema, que irá ser processada pelo script e dará como resposta a solução do problema para aquele valor.

Os educandos nessa etapa serão familiarizados através da execução de dois programas simples de cálculo numérico.

Figura 04

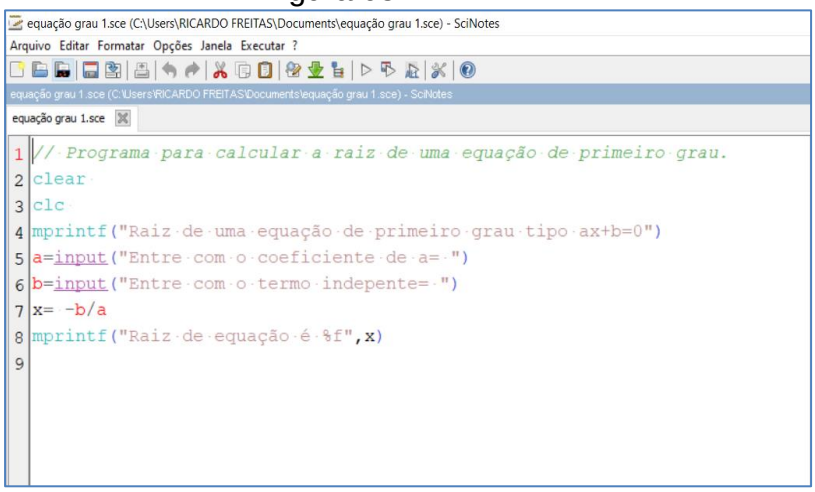


Fonte: o autor

Os programas utilizados como demonstração do uso dos comandos básicos são para calcular raízes de equações do primeiro e do segundo grau, onde serão mostrados exemplos de utilização da programação numérica antes de se começar a modelar os fenômenos de física.

Ex. 01) Programa para calcular a raiz de uma equação do primeiro grau.

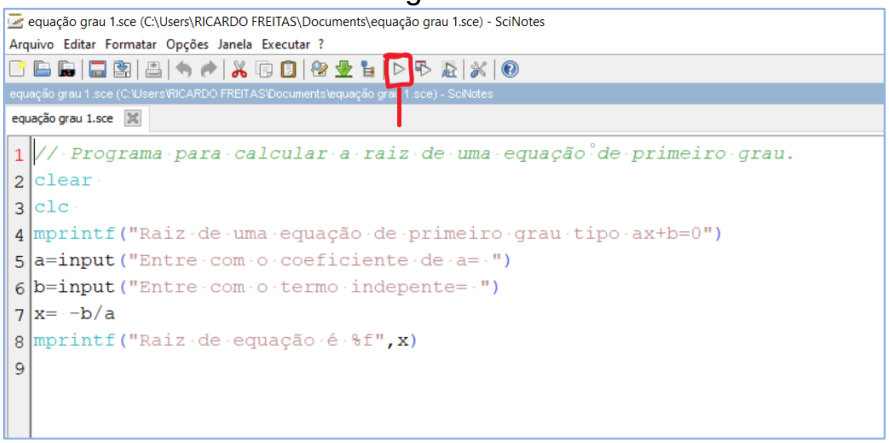
Figura 05



Fonte: o autor

Para execução do programa basta clicar no ícone, como indicado na figura 04, neste momento o software pergunta onde você deseja salvar o programa, daí você escolhe uma pasta onde deseja que fique seu programa.

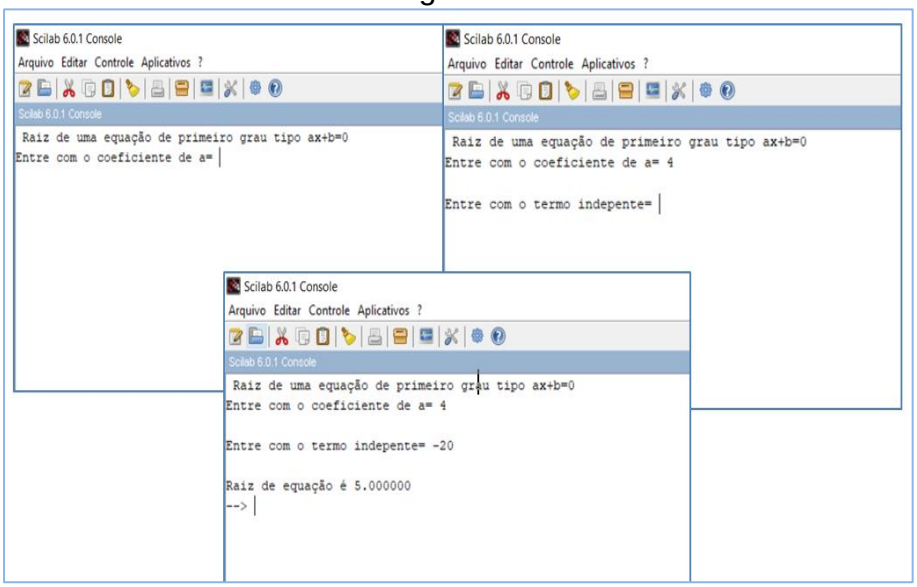
Figura 06



Fonte: o autor

Após clicar no ícone para execução, as telas a seguir são apresentadas sequencialmente para inserção dos dados e resolução do problema, no caso, a determinação da raíz de uma equação do primeio grau.

Figura 07

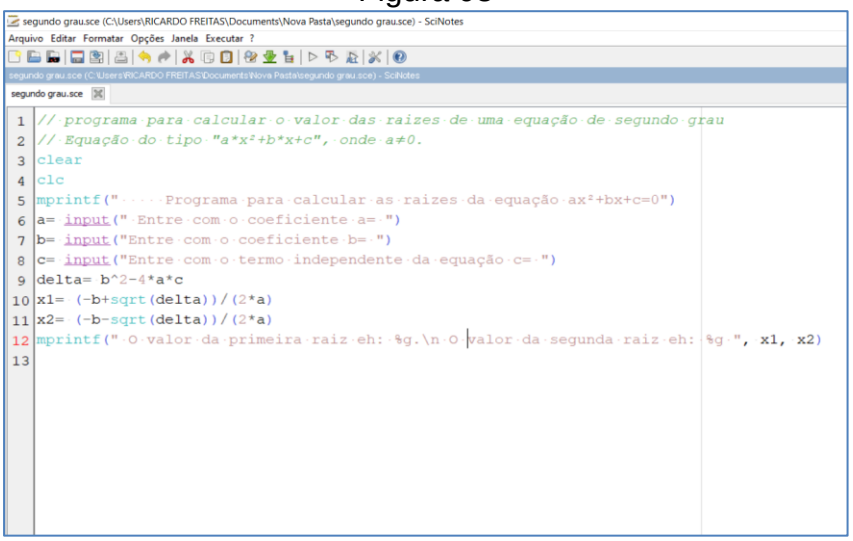


Fonte: o autor

Observe, que o método utilizado aqui, é o aluno aprender fazendo, o que vai ao encontro da teoria Construcionista de Papert.

Ex.: 02) Programa para calcular as raízes de uma equação do segundo grau.

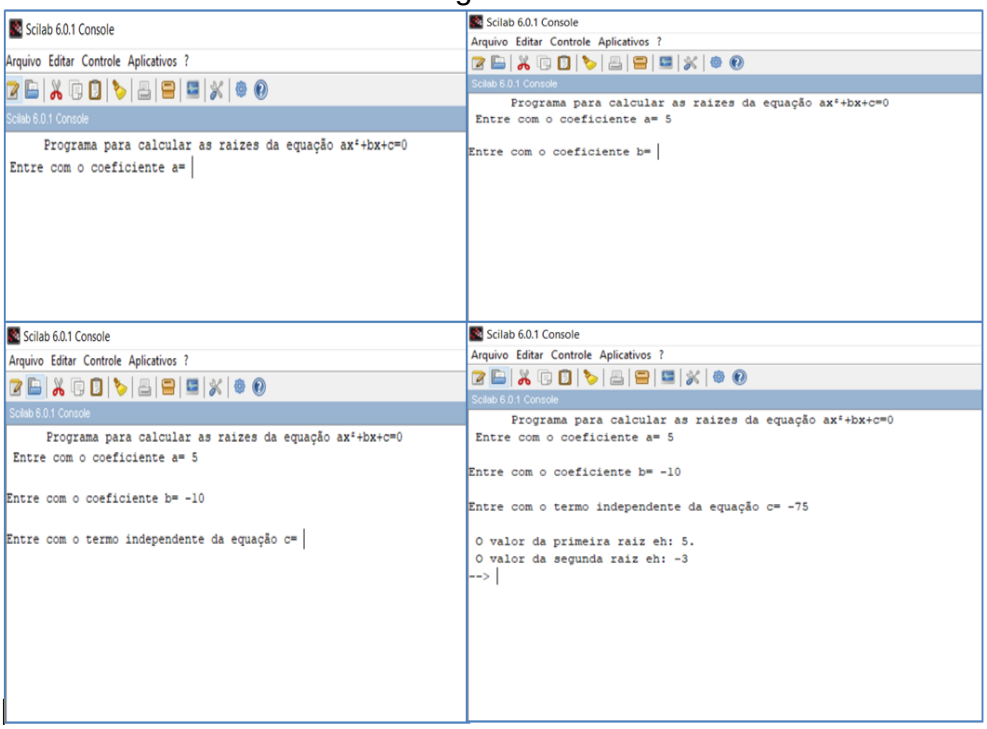
Figura 08



Fonte: o autor

Como a inserção dos dados se obtém a seguinte solução:

Figura 09



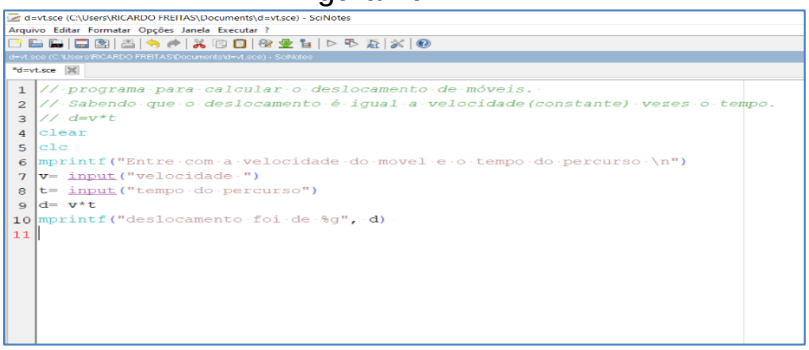
Fonte: o autor

4. **Construção de Modelos para resolução de problemas de Cinemática**

4.1 Modelando a Equação do Deslocamento = velocidade×tempo (d = v x t)

Como exemplo inicial de modelagem o professor responsável pode construir passo a passo o seguinte programa de modelagem da equação do Deslocamento. Chamando atenção que este processo deve se dá após o Professor ter explanado o assunto, e o recurso entra como motivador e parte do processo de aprendizagem.

Figura 10



Fonte: o autor

O programa construído se inicia com o uso de “//”, que como visto, indica que o texto após sua inserção não será executado pelo software. Apenas servirá como instrução e definição de para que o script pode ser usado**.**

Em seguida usa-se os comandos clc e clear, com o intuito de limpar o executor onde se localiza o prompt e liberar memória do sistema.

O comando mprintf("Entre com a velocidade do movel e o tempo do percurso \n") insere uma fala durante a execução do script, dando instruções e indicando as próximas etapas da execução.

\n indica que o próximo comando será executado na linha abaixo.

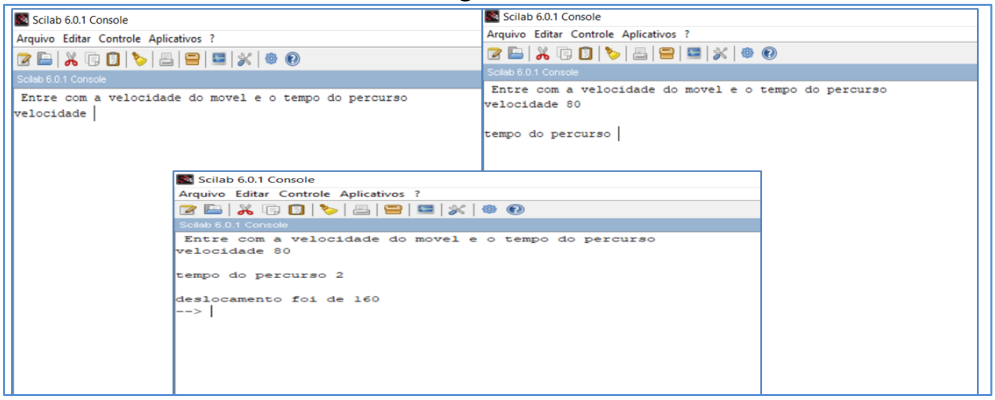
A seguir são definidos dois fatores no problema, a velocidade(v) e o tempo(t). Note que ambas são definidas com input( ). Isso é por que são variáveis de possíveis problemas e logo serão determinados seus valores no momento de execução do programa feito. O professor deve ficar atento quanto à questão de grandezas na mesma unidade, se a velocidade for expressa em quilômetros por hora o tempo deve ser expresso em horas, ou seja, sempre na mesma unidade.

Em seguida foi escrita a equação que processa os dados inseridos e modela numericamente o fenômeno. Nesse caso foi definida a variável $d = v \times t$.

Para fechar o programa usou-se mprintf("deslocamento foi de %g", d)

O comando foi usado para gerar um título à resposta final ao programa. Essa parte do texto sempre vem entre aspas, exceto o sinal **%g** que indica que não deve ser expressa casas decimais no resultado. Se for necessário saber o valor das casas decimais esse comando deve ser trocado por **%f**. Após a virgula vem a variável chave do problema, aquela que está se querendo conhecer o valor **d.**

Figura 11



Fonte: o autor

Um exemplo de aplicação a ser resolvido a partir do modelo no Scilab:

"Um carro anda durante 3 horas seguidas com velocidade de 60 km/h. Quantos quilômetros o carro percorreu?"

Resposta:

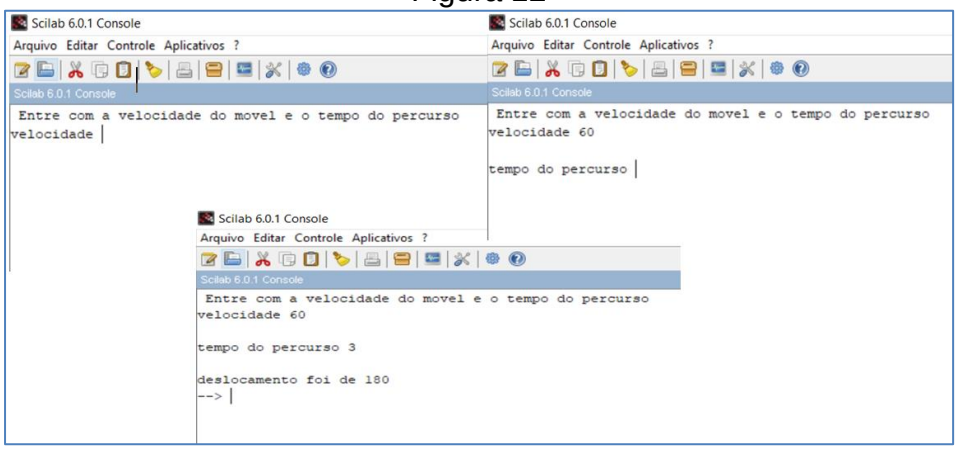
v=60km/h

t= 3h

d= v*t

Substituindo as variáveis no programa feito utilizando SciLab temos:

Figura 12

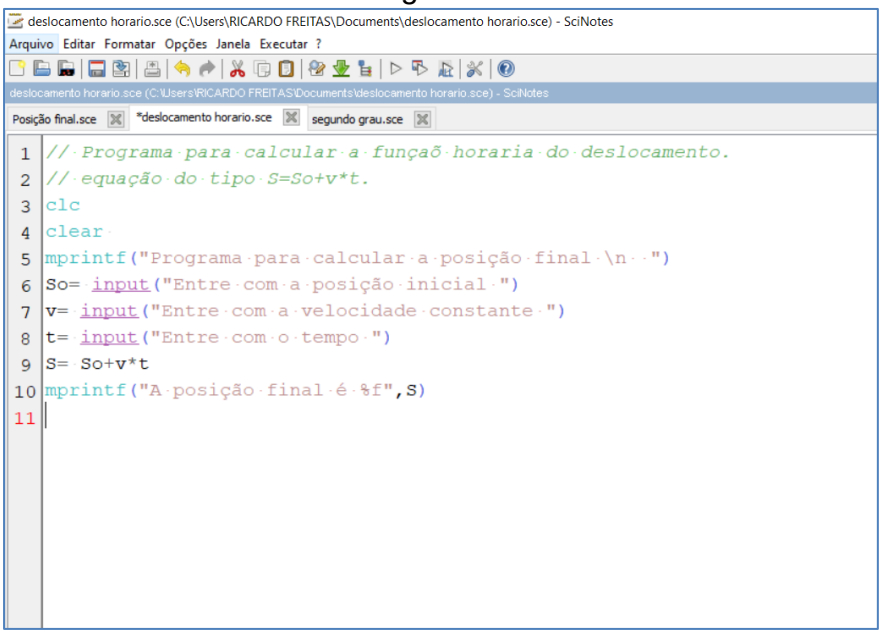


Fonte: o autor

4.2 Modelagem da equação posição = posição inicial + velocidade × tempo.

Outro exemplo simples que pode ser construído passo a passo com os alunos é o da função horaria do deslocamento.

Figura 13



Fonte: o autor

O programa construído se inicia com o uso de "//" para indica que o texto após sua inserção não será executado pelo software. Apenas servirá como instrução e definição de para que o script pode ser usado**.** Em seguida usam-se os comandos clc e clear, com o intuito de limpar o executor onde se localiza o prompt e liberar memória do sistema.

O comando $\boldsymbol{mprintf}$("Programa para calcular a posição final ") insere um título durante a execução do script, dando instruções e indicando as próximas etapas da execução.

Na etapa seguinte são inseridos três variáveis com o comando de entrada de dados $\boldsymbol{input}()$. Observe que dentro do parênteses do comando $\boldsymbol{input}()$ é usado um comentário entre "", para contextualizar qual é aquela variável. As variáveis usadas são: So, posição inicial; v, velocidade; t, tempo.

A equação que vai processar os dados de entrada nesse problema é a variável S que é definida como:

$$\boldsymbol{S = S_0 + v \times t}.$$

Novamente é usado o comando $mprintf()$, mas agora para imprimir o resultado de $\boldsymbol{S}$ e uma fala explicativa.

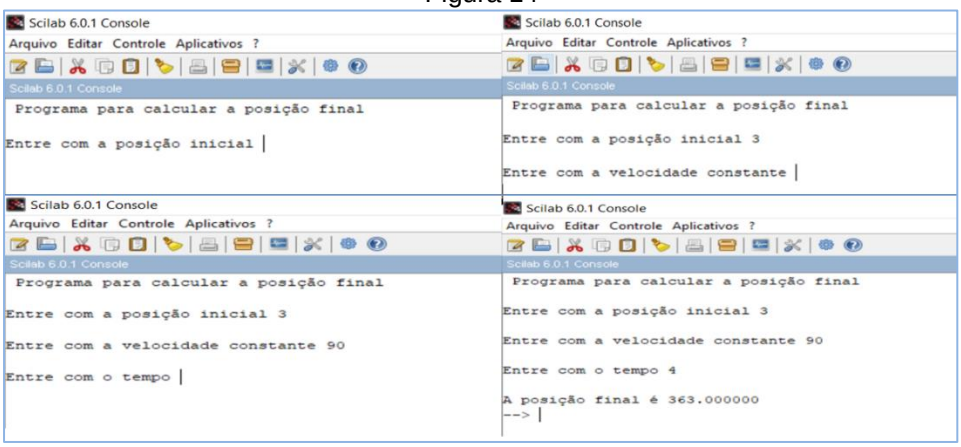
Como exemplo para utilização dessa modelagem mostraremos o seguinte problema: "Um carro está no quilômetro 3 de uma rodovia a uma velocidade

constante de 90Km/h. Determine a posição em que ele estará após um tempo de 4 horas mantendo a mesma velocidade durante todo o trajeto.

Resposta: S_0= 3km v= 90 km/h t= 4h

Utilizando os valores no programa construído temos as seguintes etapas de execução:

Figura 14



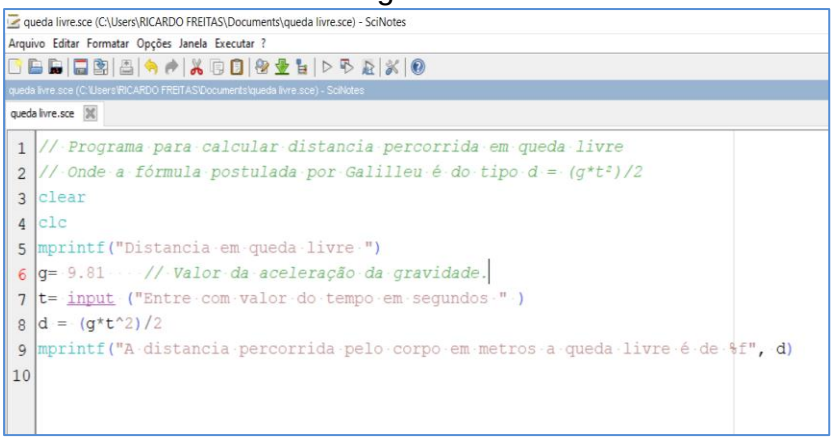
Fonte: o autor

4.3 **Modelagem da equação:**

distância = [aceleração da gravidade × (**tempo)^2] ÷ 2.**

Ainda construindo passo a passo com os alunos o professor pode construir um programa para calcular a distância em queda livre percorrida por um corpo em função do tempo.

Figura 15



Fonte: o autor

Nesse programa se inicia com o uso de “//” para indica que o texto após sua inserção não será executado pelo software. Apenas servirá como instrução e definição de para que o script pode ser usado**.**

Usa-se os comando clc e clear, com o intuito de limpar o executor onde se localiza o $prompt$ e liberar memoria do sistema.

O comando $\boldsymbol{mprintf}$("Distância em queda livre ") insere um título durante a execução do script.

Logo após foi definida uma constante, que é a da gravidade, com valor igual a 9,81m/s².

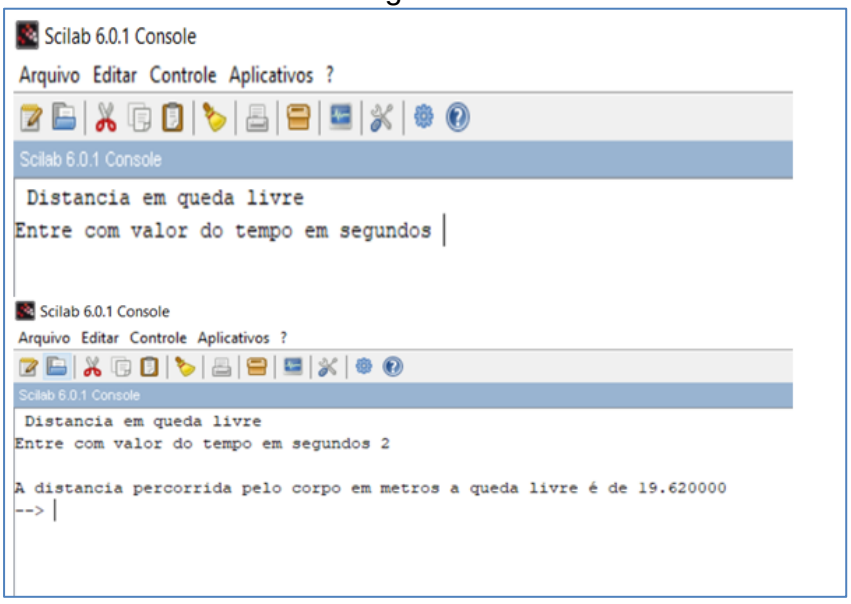
Define-se a variável do tempo (t) usando o $\boldsymbol{input}()$ para que o valor seja atribuído pelo usuário durante a execução do programa, afim de resolver problemas diversos.

O processamento dos dados inseridos efetuado no programa se dá, nesse caso, pela formula de Galileu definida como:

$$\boldsymbol{d = \frac{(g * t^2)}{2}}$$

Novamente é usado o comando $mprintf()$ para imprimir o valor resposta, que é distância de queda em metros e uma fala explicativa.

Figura 16



Fonte: o autor

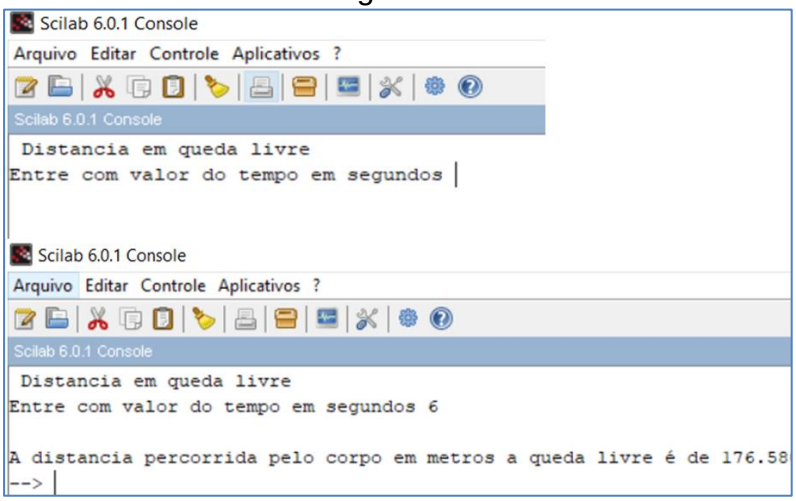
Exemplo de aplicação do programa construído:

Um objeto é abandonado do topo de um edifício levando 6 segundos para tocar o solo. Determine a altura do edifício utilizando a equação da queda livre dos corpos. Adote g = 9,8 m/s².

Resposta: g= 9,8 m/s^2 t= 6 seg

Aplicando esses valores na programação construída temos:

Figura 17



Fonte: o autor

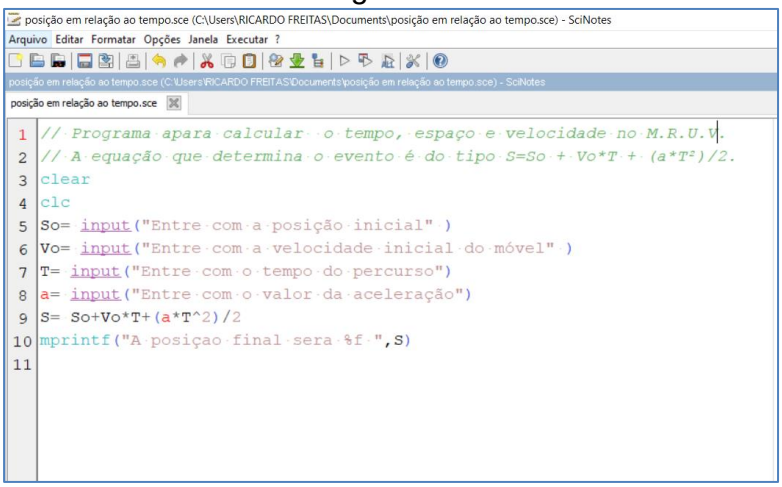
5. EXECUÇÃO

Esse momento compreende pedir aos alunos que baseado no que viram até o momento utilizem o software SciLab para modelar fenômenos da física que envolvam movimentos com velocidade ou aceleração constantes ou variáveis.

Diversos programas podem ser criados pelos alunos com o auxílio e orientação do professor responsável durante seu trabalho com o tema de Cinemática no Ensino Médio.

Programa para calcular a posição em relação ao tempo e às condições de aceleração, velocidade inicial e posição inicial.

Figura 18



Fonte: o autor

Exemplo de utilização do script:

Um móvel parte do repouso de uma posição inicial igual a 30m. Sabendo que sua velocidade inicial é de 5m/s, e sua aceleração é de 4m/s², pede-se:

Posição final do móvel após 10s.

Resposta.

$S_0 = 30$

$v_0 = 0$

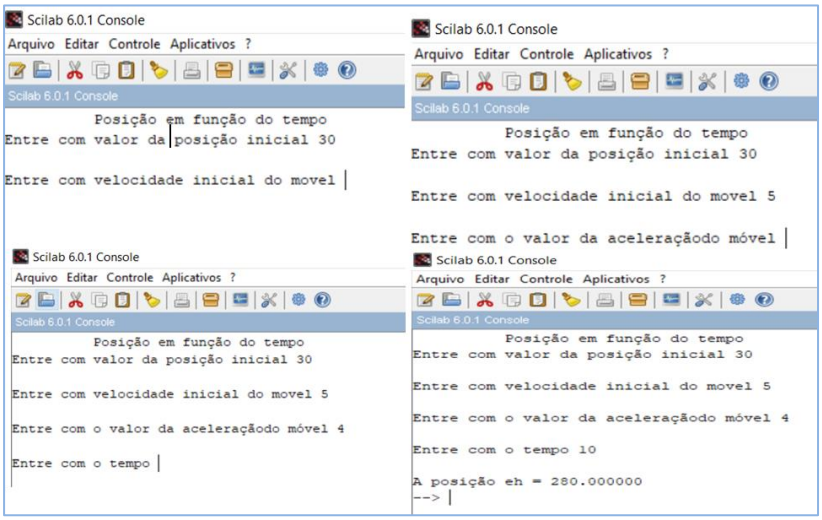
t =5

a = 4m/s²

$S = S_0 + v_0 t + at^2/2$

Inserindo os valores no programa construídos temos:

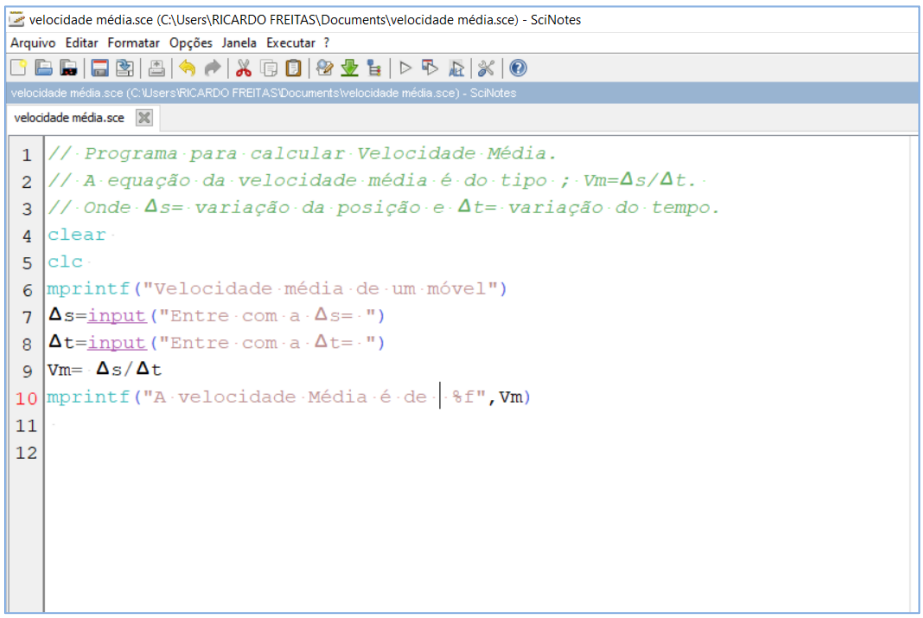
Figura 19



Fonte: o autor

Programa para calcular a velocidade média dada pelo quociente da variação de posição pela variação de tempo.

Figura 20



Fonte: o autor

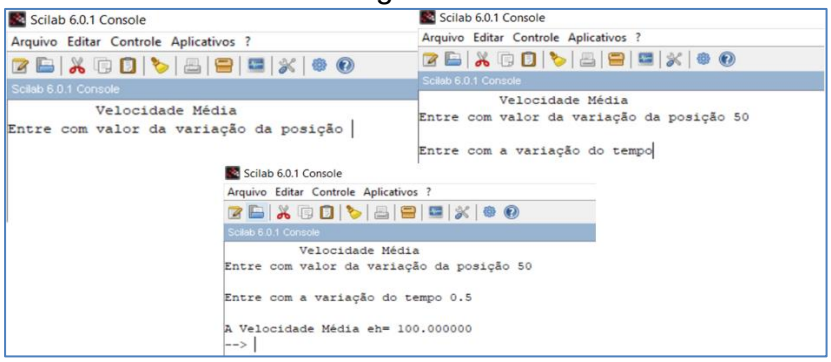
Podemos utilizar o seguinte exemplo de aplicação do Script:

Vamos supor que um trecho de 50 km de uma rodovia seja monitorado pela Polícia Rodoviária. A velocidade máxima permitida no trecho é de 80 km/h. Um motorista percorreu os 50 km do trecho em 30 minutos. Calcule se ele ultrapassou o limite de velocidade permitido. Disponível em: https://brasilescola.uol.com.br/matematica/equacoes-no-calculo-velocidade-media-um-veiculo.htm

Resposta: $\Delta S = 50km \quad \Delta t = 0{,}5h \quad Vm = \frac{\Delta S}{\Delta t}$

Aplicando os valores no script temos:

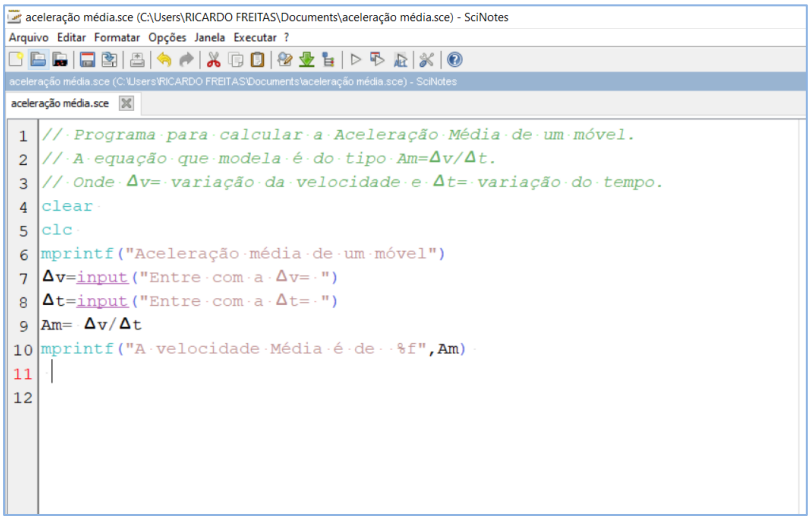
Figura 21



Fonte: o autor

Programa para calcular a aceleração média dada pelo quociente da variação de velocidade pela variação de tempo.

Figura 22



Fonte: o autor

Como exemplo de aplicação do script podemos ter o seguinte problema:

Qual a aceleração média de um ciclista que ao avistar uma situação de perigo passou da velocidade de 10m/s para 20m/s em 20s?

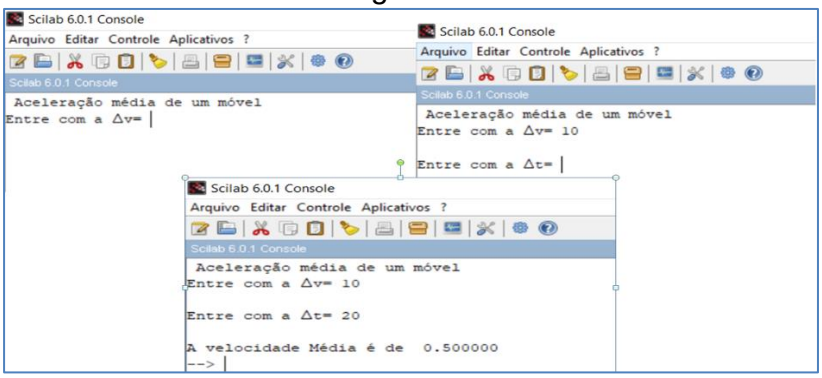
Resposta:

$\Delta v = (20 - 10)$m/s

$\Delta t = (20 - 0)$s

Aplicando os dados no script temos:

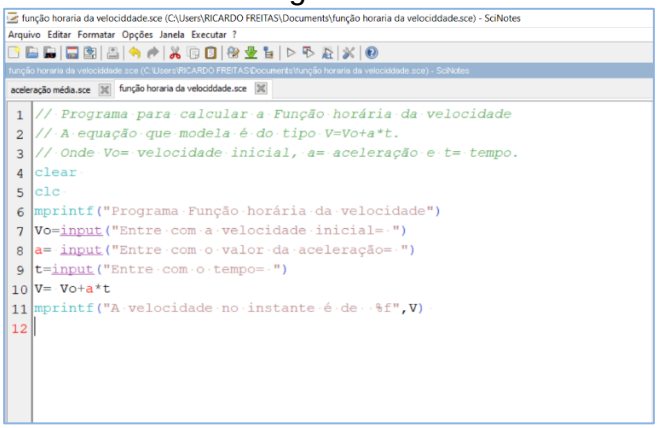
Figura 23



Fonte: o autor

Programa para calcular a velocidade em função da velocidade inicial mais o produto da aceleração pelo tempo.

Figura 24



Fonte: o autor

Como exemplo de aplicação do script podemos usar o seguinte problema:

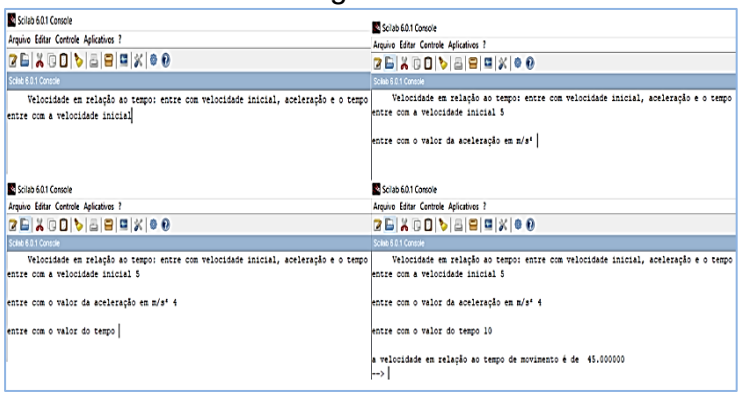
Um móvel parte do repouso de uma posição igual a 30m. sabendo que sua velocidade inicial é de 5 m/s e sua aceleração é de 4m/s², pede-se:

Velocidade após 10s? Disponível em: http://exerciciosdefisica.blogspot.com/2009/10/exercicio-16.html

Resposta: $v_0 = 5\text{m/s} \quad a = 4\text{m/s}^2 \quad t = 10s$

Usando os valores no script temos:

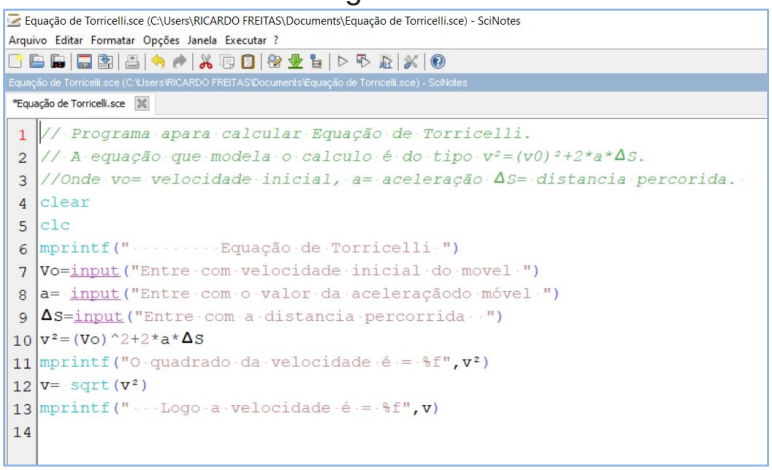
Figura 25



Fonte: o autor

Equação de Torricelli utilizada para calcular a velocidade de um corpo em relação ao espaço que ele percorre.

Figura 26



Fonte: o auto

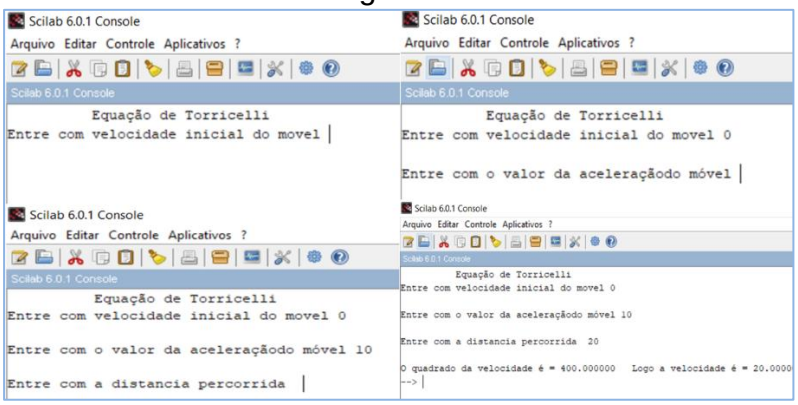
Como exemplo de aplicação desse script podemos utilizar o seguinte problema:

(UEPI) Um corpo é abandonado de uma altura de 20 m num local onde a aceleração da gravidade da Terra é dada por g = 10 m/s². Desprezando o atrito, o corpo toca o solo com velocidade: disponível em: https://www.todamateria.com.br/equacao-de-torricelli/

Resposta: $v_0 = 0 \quad a = 10m/s^2 \quad \Delta S = 20m$

Substituindo os valores no programa obtemos:

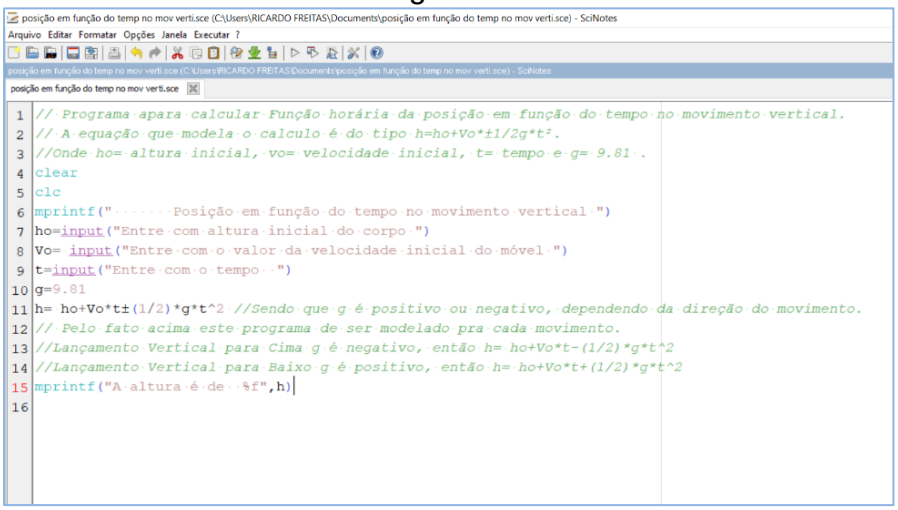
Figura 27



Fonte: o autor

Programa para calcular o Movimento Vertical.

Figura 28



Fonte: o autor

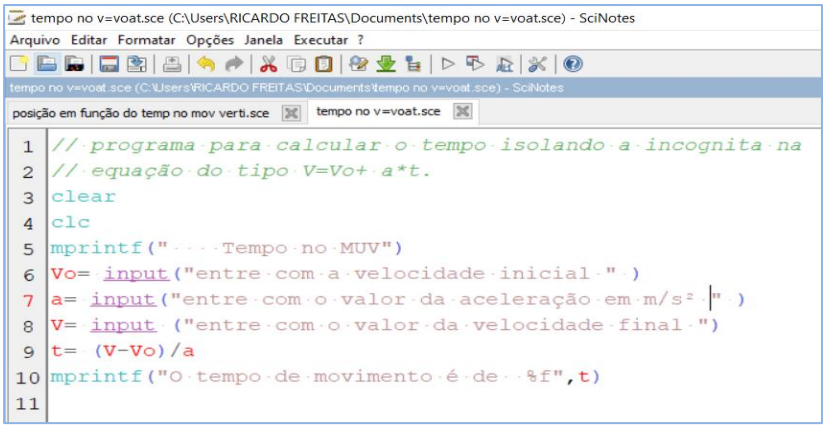
Como exemplo da aplicação desse script podemos usar o seguinte problema:

Uma bola de futebol é chutada para cima com velocidade igual a 20m/s. Qual a altura máxima atingida pela bola? Dado g=10m/s².

Resposta:

Inicialmente vamos calcular o tempo utilizando o programa exemplificado anteriormente fig. 24, no entanto alterando seus parâmetros e isolando a variável t no primeiro membro da igualdade.

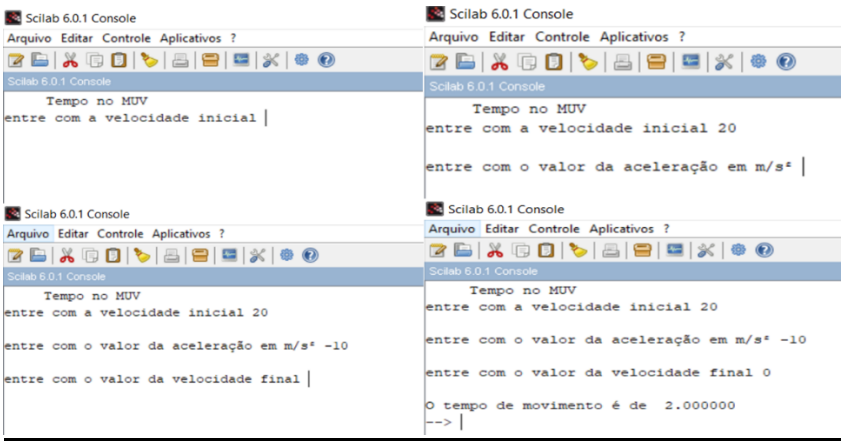
Figura 29



Fonte: o autor

Aplicando os dados do problema temos:

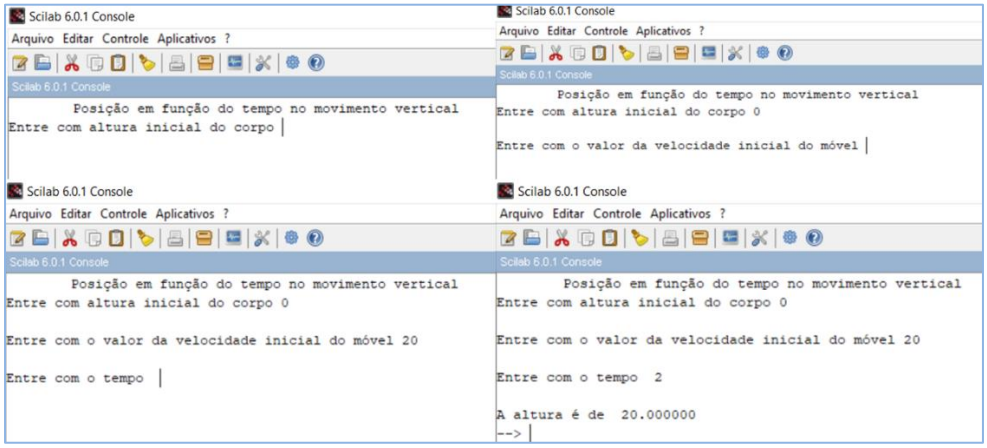
Figura 30



Fonte: o autor

O tempo então é igual a 2s. Com isso aplicamos no script fig. 28 temos:

Figura 31



Fonte: o autor

6. Plotagem de Gráficos no SciLab.

O Scilab possui várias funções gráficas integradas, para a apresentação visual dos dados processados pelos modelos desenvolvidos. Pode-se plotar gráficos de funções em duas ou três dimensões. Os gráficos podem ser programados no editor de arquivos ou diretamente na área de execução de comandos, o $'\boldsymbol{prompt}'$.

Comando $\boldsymbol{Plot}()$

$Plot()$ é o meio mais simples de plotar um gráfico de uma função usando a linguagem Scilab. utilizando o comando $plot(x, y)$, no prompt ou no editor de programas, para um valor de x sendo um vetor que pode variar em um intervalo definido da seguinte forma:

$$--> x = [1{:}0.1{:}10]$$

Onde 0.1 entre dois sinais $(:)$, representa o incremento de interação, ou seja, o intervalo de interações do programa. A cada décimo, como foi escolhido no exemplo, haverá a geração de um ponto no gráfico.

Podemos ainda executar um intervalo na seguinte forma:

$$x = [1{:}10]$$

Onde os dois pontos (:) entre os dois escalares indica um intervalo monótono, ou seja, a variação é de uma unidade.

O valor de y é dado em função de x, no tipo $y = f(x)$.

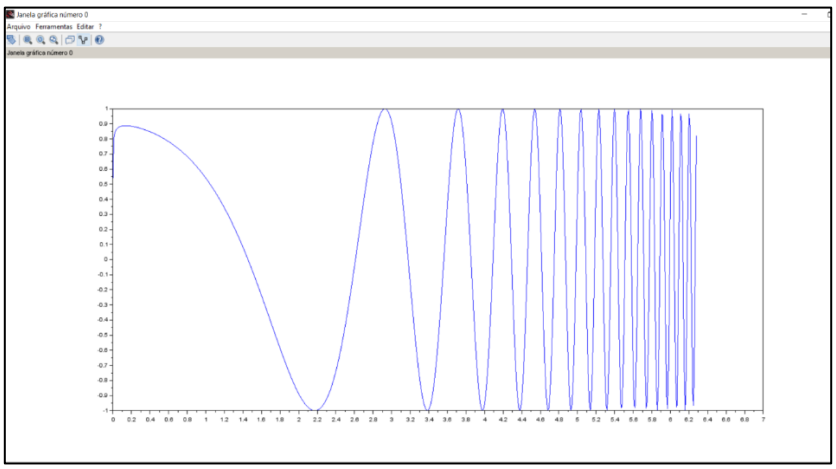
Os valores se associam num plano de eixos perpendiculares e ordenados. Por exemplo, a construção do gráfico da função $y = f(x), onde\ f(x) = sen(x)$, com x variando de $0\ a\ 2\pi$. Esses comandos podem ser executados no prompt ou no editor de arquivos.

$$-->\ x = [0{:}2 * \%pi];$$

$$-->\ y =\ sin(x);$$

$$-->\ plot(x, y)$$

Figura 32



Fonte: o autor

Para plotar gráficos o usuário pode recorrer a ajuda "*help plot".*

Os tipos de gráficos mais usuais nessa linha de comando são:

- $xtitle -$ adiciona títulos a janelas de gráficos.
- $plot2d$ - esboço 2d.

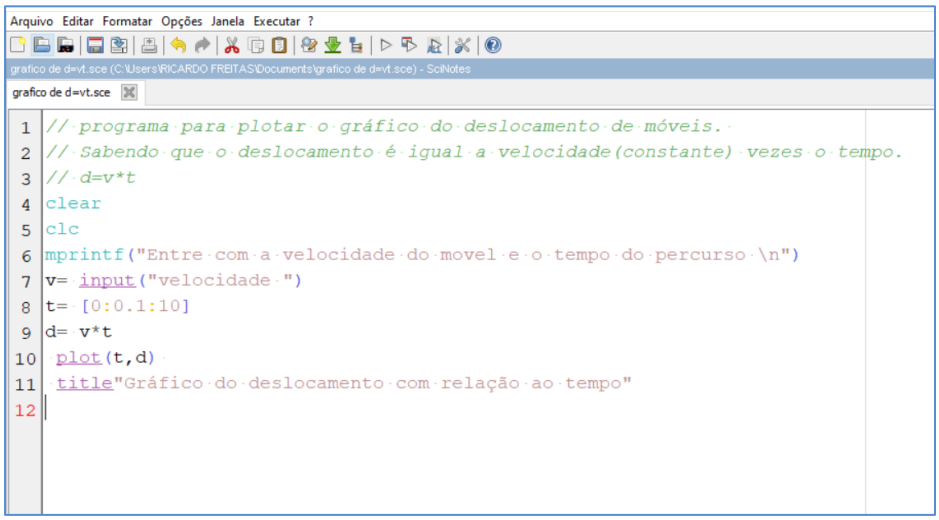
- $plot2d2$ - esboço 2d (funções de degraus).
- $plot2d3$ - esboço 2d (barras verticais).
- $plot2d4$ - esboço 2d (setas).
- $fplot2d$ - esboço 2d de uma curva definida por uma função.
- $xgrid$ - adiciona um grid em um esboço 2d.
- $errbar$ - adiciona barras de erro verticais a um esboço 2d.
- $histplot$ - esboça um histograma.
- $Matplot$ - esboço 2d de uma matriz utilizando-se cores

Para construir um script que plote um gráfico dos modelos da cinemática, por exemplo, abre-se um novo documento no editor de arquivos.

Como exemplo inicial de modelagem o professor pode construir passo a passo o seguinte programa para plotar o gráfico do deslocamento de móveis.

Sabendo que o deslocamento é igual à velocidade (constante) vezes o tempo:

Figura 33



Fonte: o autor

O programa construído se inicia com o uso de “//”, seguido do comentário pertinente ao caso**.**

Usa-se os comando $clc\ e\ clear$.

O comando $\boldsymbol{mprintf}$("Entre com a velocidade do móvel e o tempo do percurso \n").

Na etapa seguinte é inserida uma variável com o comando de entrada de dados $\boldsymbol{input}$( ). A variáveis usadas é v, velocidade.

É definida aqui então um intervalo de tempo t, da seguinte forma

$$t = [1{:}\,0.1{:}\,10],$$

onde t é um intervalo de tempo de 0 a 10, com pontos de interação a cada décimo.

A equação que vai processar os dados de entrada nesse problema é a variável d que é definida como:

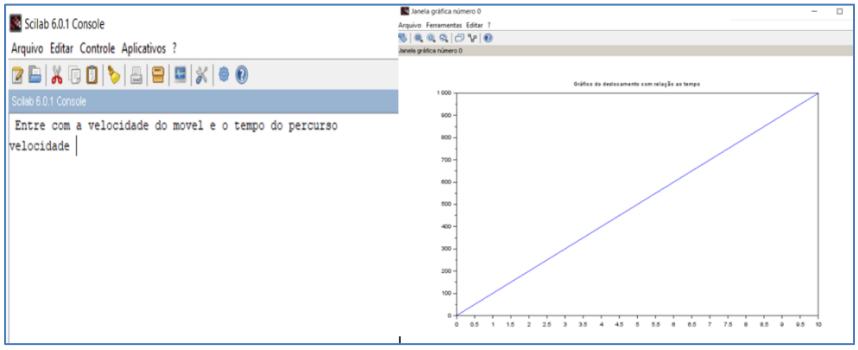
$$\boldsymbol{d = v \times t}.$$

O próximo comando inserido é o comando $plot(t, d)$, que ira imprimir um gráfico do deslocamento em função do tempo na janela gráfica do software.

Para adicionar um título ao gráfico usa-se o comando

$title$"Gráfico d$o\ deslocameno\ em\ função\ do\ tempo$".

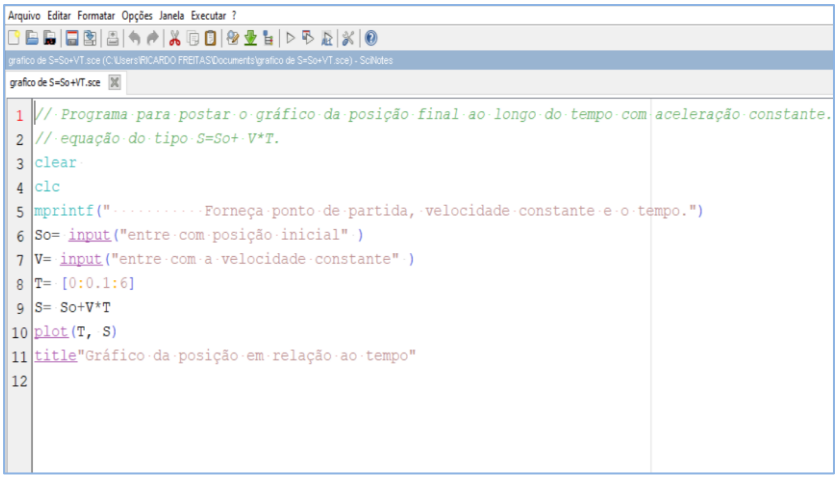
Figura 34



Fonte: o autor

Programa para plotar o gráfico da posição final em um intervalo de tempo com aceleração constante.

Figura 35



Fonte: o autor

O programa construído se inicia com o uso de "//", seguido do comentário pertinente ao caso**.**

Usa-se os comando $clc\ e\ clear$.

$\boldsymbol{mprintf}$("Forneça ponto de partida, velocidade constante e o tempo." \n")

So= input("entre com posição inicial")

V= input("entre com a velocidade constante")

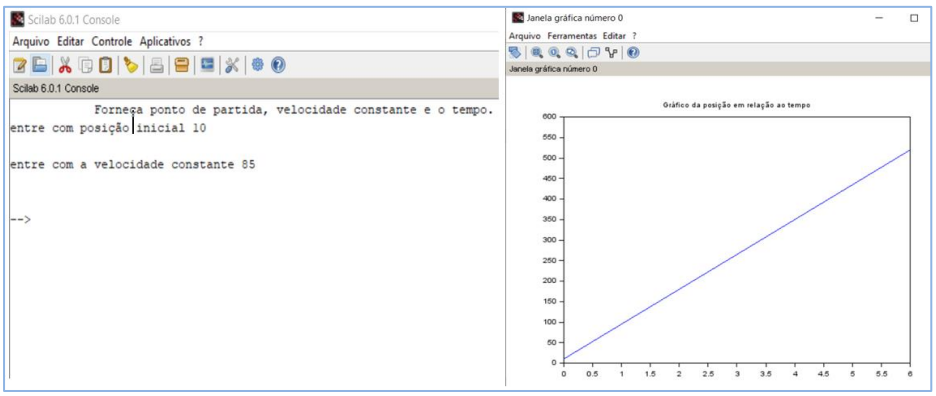
T= [0:0.1:6], intervalo de 0 a 6 com interação de 0,1.

S= So+V*T, equação que modela o fenômeno .

plot(T, S), para plotar o gráfico da posição em função do tempo

title"Gráfico da posição em relação ao tempo"

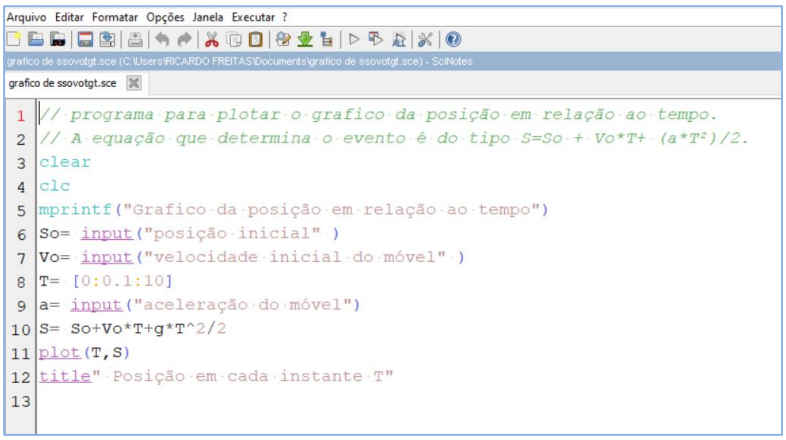
Figura 36



Fonte: o autor

Programa para plotar o gráfico da posição em relação ao tempo com variação de aceleração.

Figura 37



Fonte: o autor

O programa construído se inicia com o uso de “//”, seguido do comentário pertinente ao caso**.**

Usa-se os comando $clc\ e\ clear$.

Os comandos:

mprintf("Grafico da posição em relação ao tempo")

So= input("posição inicial")

Vo= input("velocidade inicial do móvel")

T= [0:0.1:10]

a = input("aceleração do móvel")

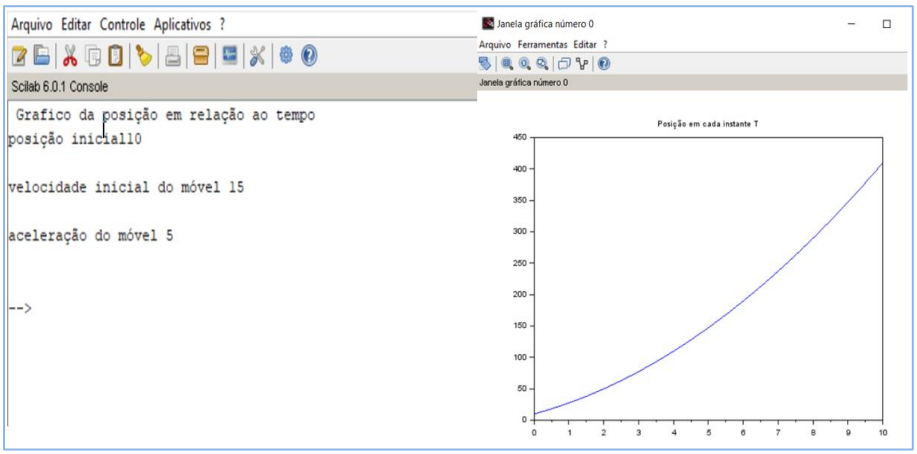
S= So+Vo*T+g*T^2/2

plot(T,S)

title" Posição em cada instante T"

Em seguida se insere os dados para plotar o gráfico desejado.

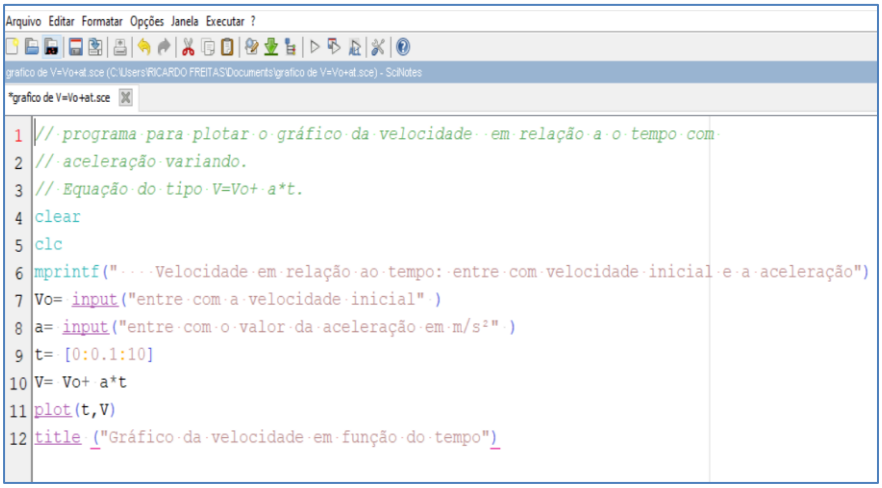
Figura 38



Fonte: o autor

Programa para plotar o gráfico da velocidade em relação ao tempo com variação de aceleração.

Figura 39



Fonte: o autor

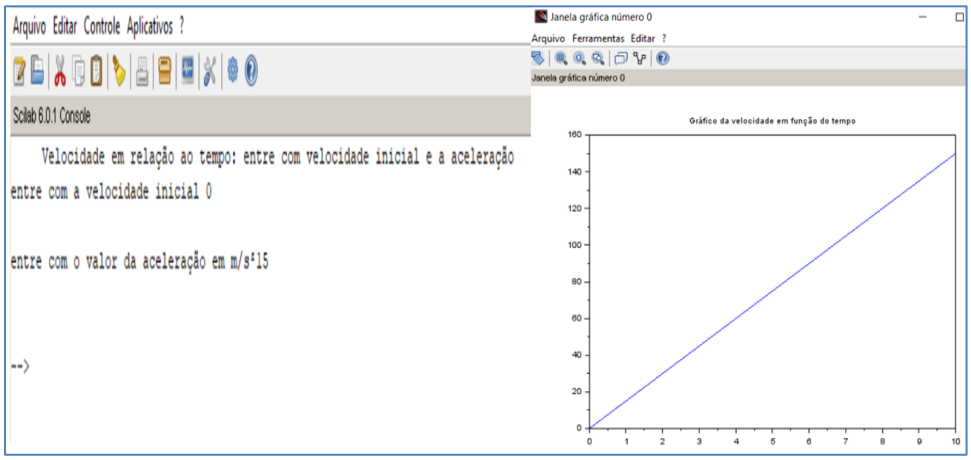
O programa construído se inicia com o uso de “//”, seguido do comentário pertinente ao caso**.**

Usa-se os comando $clc\ e\ clear$.

Os comandos:

```
mprintf("    Velocidade em relação ao tempo: entre com velocidade inicial, aceleração e o tempo")
Vo= input("entre com a velocidade inicial" )
a= input("entre com o valor da aceleração em m/s²" )
t= [0:0.1:10]
V= Vo+ a*t
plot(t,V)
title ("Gráfico da velocidade em função do tempo")
```

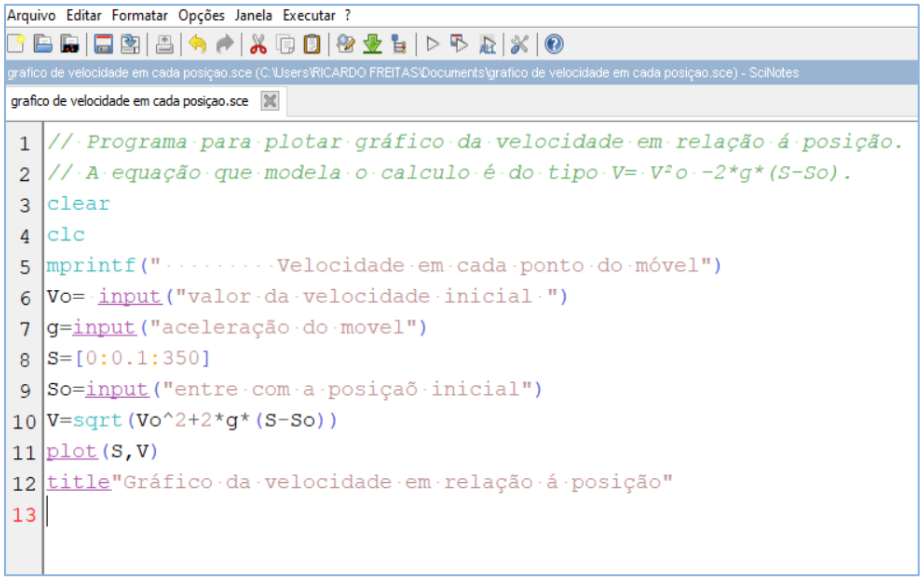
Figura 40



Fonte: o autor

Programa para plotar gráfico da velocidade em relação à posição

Figura 41



Fonte: o autor

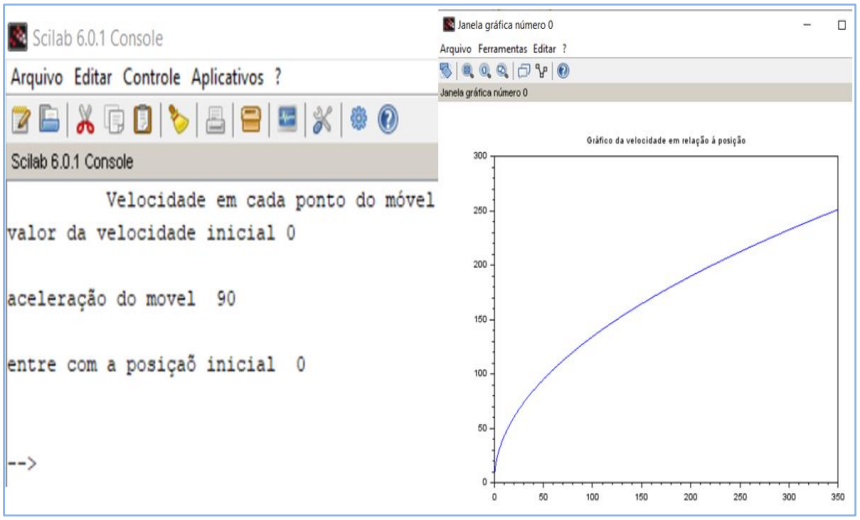
O programa construído se inicia com o uso de “//”, seguido do comentário pertinente ao caso**.**

Usa-se os comando $clc\ e\ clear$.

Os comandos:

```
mprintf("        Velocidade em cada ponto do móvel")
Vo= input("valor da velocidade inicial ")
g=input("aceleração do movel")
S=[0:0.1:350]
So=input("entre com a posiçaõ inicial")
V=sqrt(Vo^2+2*g*(S-So))
plot(S,V)
title"Gráfico da velocidade em relação á posição"
```

Figura 42



Fonte: o autor

7. Sugestão de Avaliação e Conclusão

A o final do período de execução o professor irá avaliar a funcionalidade dos programas escritos pelos alunos e verificar se as fórmulas foram corretamente empregadas.

Essa avaliação será no tocante a realização do que cada programa construído se propõe a fazer. O professor irá executar cada programa de cada aluno e verificar a funcionalidade dos mesmos.

O professor irá avaliar os programas para plotagem dos gráficos dos fenômenos se os mesmos descrevem o comportamento da variação de tempo velocidade ou aceleração.

O professor irá avaliar os programas feitos afim de auxiliar em algum parâmetro.

Cada um dos programas vistos anteriormente pode ser indicado para que o aluno possa recriá-lo de maneira que o professor, munido dessas instruções, lhe oriente, esclarecendo alguma dúvida quando necessário.

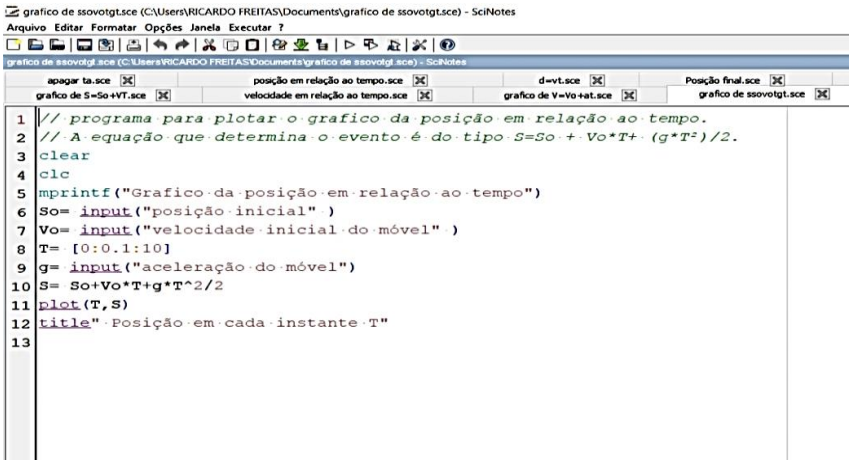
Problemas diversos de física no campo da Cinemática podem ser modelados e resolvidos através dos scripts apresentados, bastando para tanto, que o aluno entender a lógica usada para escrever os presentes modelos. O aluno, ao passo que aprende a manipular as equações isolando as variáveis necessárias para resolver cada situação problema pode avançar no conhecimento teórico da Física.

O professor tem a opção de mostrar o domínio do Scilab utilizando exemplos bastante simples que envolvam movimento e passar a introduzir questões mais complexas à medida que os alunos forem se envolvendo na utilização de programação. Podemos resolver questões de vestibulares conceituados no pais.

APÊNDICE II - **Modelos de Programas do SciLab a fim de serem implementados em sala.**

1)

Figura 23



Fonte: o autor

2)

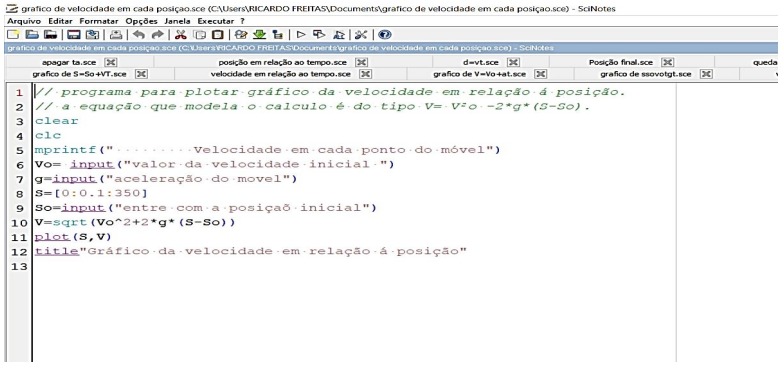


grafico de velocidade em cada posiçao.sce (C:\Users\RICARDO FREITAS\Documents\grafico de velocidade em cada posiçao.sce) - SciNotes

Arquivo Editar Formatar Opções Janela Executar ?

grafico de velocidade em cada posiçao.sce (C:\Users\RICARDO FREITAS\Documents\grafico de velocidade em cada posiçao.sce) - SciNotes

apagar ta.sce | posição em relação ao tempo.sce | d=vt.sce | Posição final.sce | queda

grafico de S=So+VT.sce | velocidade em relação ao tempo.sce | grafico de V=Vo+at.sce | grafico de ssovotgt.sce

```
1  // programa para plotar gráfico da velocidade em relação á posição.
2  // a equação que modela o calculo é do tipo V= V²o -2*g*(S-So).
3  clear
4  clc
5  mprintf("·····Velocidade em cada ponto do móvel")
6  Vo= input("valor da velocidade inicial ")
7  g=input("aceleração do movel")
8  S=[0:0.1:350]
9  So=input("entre com a posiçaõ inicial")
10 V=sqrt(Vo^2+2*g*(S-So))
11 plot(S,V)
12 title"Gráfico da velocidade em relação á posição"
13
```

Gráfico da posição em função do tempo

velocidade em relaçao a posicao.sce (C:\Users\RICARDO FREITAS\Documents\velocidade em relaçao a posicao.sce) - SciNotes

Arquivo Editar Formatar Opções Janela Executar ?

velocidade em relaçao a posicao.sce (C:\Users\RICARDO FREITAS\Documents\velocidade em relaçao a posicao.sce) - SciNotes

apagar ta.sce | posição em relação ao tempo.sce | d=vt.sce

segundo grau.sce | grafico de d=vt.sce | grafico de S=So+VT.sce | velocidade em relação ao tempo.sce | grafico de V=Vo+at.

```
1  // programa apara calcular a velocidade em relação á posição.
2  // a equação que modela o calculo é do tipo V= V²o -2*g*(S-So).
3  clear
4  clc
5  mprintf("·····Velocidade em cada ponto do móvel")
6  Vo= input("valor da velocidade inicial ")
7  g=input("aceleração do movel")
8  S=input("entre com a posição final")
9  So=input("entre com a posiçaõ inicial")
10 V=sqrt(Vo^2+2*g*(S-So))
11 mprintf("a velocidade final eh de %f",V)
12
```

velocidade em função da posição

grafico de ssovotgt.sce (C:\Users\RICARDO FREITAS\Documents\grafico de ssovotgt.sce) - SciNotes

Arquivo Editar Formatar Opções Janela Executar ?

grafico de ssovotgt.sce (C:\Users\RICARDO FREITAS\Documents\grafico de ssovotgt.sce) - SciNotes

apagar ta.sce | posição em relação ao tempo.sce | d=vt.sce | queda livre.sce | segundo grau.sce | grafico de d=vt.sce | grafico de S=So+VT.sce | velocidade em relação ao tempo.sce

```
1  // programa para plotar o grafico da posição em relação ao tempo.
2  // A equação que determina o evento é do tipo S=So + Vo*T+ (g*T²)/2.
3  clear
4  clc
5  mprintf("Grafico da posição em relação ao tempo")
6  So= input("posição inicial" )
7  Vo= input("velocidade inicial do móvel" )
8  T= [0:0.1:10]
9  g= input("aceleração do móvel")
10 S= So+Vo*T+g*T^2/2
11 plot(T,S)
12 title" Posição em cada instante T"
13
```

gráfico da posição em função do tempo

posição em relação ao tempo.sce (C:\Users\RICARDO FREITAS\Documents\posição em relação ao tempo.sce) - SciNotes

Arquivo Editar Formatar Opções Janela Executar ?

posição em relação ao tempo.sce (C:\Users\RICARDO FREITAS\Documents\posição em relação ao tempo.sce) - SciNotes

apagar ta.sce | posição em relação ao tempo.sce | d=vt.sce | Posição final.sce | queda livre.sce | segundo grau.sce | grafico de d=vt.sce | grafico de S

```
1  // programa apara calcular a posição em relação ao tempo.
2  // A equação que determina o evento é do tipo S=So + Vo*T+ (g*T²)/2.
3  clear
4  clc
5  So= input("posição inicial" )
6  Vo= input("velocidade inicial do móvel" )
7  T= input("o tempo do percurso")
8  g= 9.81
9  S= So+Vo*T+g*T^2/2
10 mprintf("posiçao final sera %f ",S)
11
```

Posição em função do tempo

d=vt.sce (C:\Users\RICARDO FREITAS\Documents\d=vt.sce) - SciNotes

Arquivo Editar Formatar Opções Janela Executar ?

d=vt.sce (C:\Users\RICARDO FREITAS\Documents\d=vt.sce) - SciNotes

apagar ta.sce | d=vt.sce | Posição final.sce | queda livre.sce | segundo grau.sce | grafico de d=vt.sce | grafico de S=So+VT.sce | velocidade em rela

```
1  // programa para calcular o deslocamento de móveis.
2  // Sabendo que o deslocamento é igual a velocidade(constante) vezes o tempo.
3  // d=v*t
4  clear
5  clc
6  mprintf("Entre com a velocidade do movel e o tempo do percurso \n")
7  v= input("velocidade ")
8  t= input("tempo do percurso")
9  d= v*t
10 mprintf("deslocamento foi de %f", d)
11
12
```

deslocamento em função do tempo e velocidade

Posição final.sce (C:\Users\RICARDO FREITAS\Documents\Posição final.sce) - SciNotes

Arquivo Editar Formatar Opções Janela Executar ?

Posição final.sce (C:\Users\RICARDO FREITAS\Documents\Posição final.sce) - SciNotes

apagar ta.sce | Posição final.sce | queda livre.sce | segundo grau.sce | grafico de d=vt.sce | grafico de S=So+VT.sce | velocidade em relação ao tempo.sce | grafico de V=Vo+at.sce

```
1  // Programa pra calcular a posição final ao longo do tempo com aceleração constante.
2  // equação do tipo S=So+ V*T.
3  clear
4  clc
5  mprintf("..........Forneça ponto de partida, velocidade constante e o tempo.")
6  So= input("entere com posição inicial" )
7  V= input("entre com a velocidade constante" )
8  T= input("entre com o tempo na mesma unidade que a velocidade constante" )
9  S= So+ V*T
10 mprintf("A posicção final foi de %f", S)
11
```

Posição final em função do tempo com aceleração constante

queda livre.sce (C:\Users\RICARDO FREITAS\Documents\queda livre.sce) - SciNotes

Arquivo Editar Formatar Opções Janela Executar ?

queda livre.sce (C:\Users\RICARDO FREITAS\Documents\queda livre.sce) - SciNotes

apagar ta.sce | queda livre.sce | segundo grau.sce | grafico de d=vt.sce | grafico de S=So+VT.sce | velocidade em relação ao tempo.sce | grafico de V=Vo+at.sce

```
1  // Programa para calcular distancia percorrida em queda livre
2  // onde a fórmula postulada por Galilleu é do tipo d = (g*t²)/2
3  clear
4  clc
5  mprintf("Distancia em queda livre")
6  g= 9.81
7  t= input ("Valor do tempo em segundos" )
8  d = (g*t^2)/2
9  mprintf("A distancia percorrida pelo corpo em metros a queda livre é de %f", d)
10
```

Distância em queda livre ao longo do tempo

segundo grau.sce (C:\Users\RICARDO FREITAS\Documents\segundo grau.sce) - SciNotes

Arquivo Editar Formatar Opções Janela Executar ?

segundo grau.sce (C:\Users\RICARDO FREITAS\Documents\segundo grau.sce) - SciNotes

apagar ta.sce | segundo grau.sce | grafico de d=vt.sce | grafico de S=So+VT.sce | velocidade em relação ao tempo.sce | grafico de V=Vo+at.sce

```
1  // programa para calcular o valor das raizes de uma equação de segundo grau
2  // Equação do tipo "a*x²+b*x+c", onde a≠0.
3  clear
4  clc
5  mprintf("            Raizes da equação ax²+bx+c=0")
6  a= input(" coeficiente a= ")
7  b= input(" coeficiente b= ")
8  c= input(" termo independente da equação= ")
9  delta= (b^2)-4*a*c
10
11 x1= -b+ sqrt(delta)/(2*a)
12 x2= -b- sqrt(delta)/(2*a)
13 mprintf(" o valor da primeira raiz eh: %g.\nOvalor da segunda raiz eh: %g ", x1, x2)
14 //mprintf("xlinha %f", x1)
15 //mprintf("x2linha %f", x2)
16 //disp("1°raiz da equação ",x1)
17 //disp("2° raiz da equação ",x2)
18
19
```

Raízes de equação do segundo grau

grafico de d=vt.sce (C:\Users\RICARDO FREITAS\Documents\grafico de d=vt.sce) - SciNotes

Arquivo Editar Formatar Opções Janela Executar ?

grafico de d=vt.sce (C:\Users\RICARDO FREITAS\Documents\grafico de d=vt.sce) - SciNotes

apagar ta.sce | grafico de d=vt.sce | grafico de S=So+VT.sce | velocidade em relação ao tempo.sce | grafico de V=Vo+at.sce

```
1  // programa para plotar o gráfico do deslocamento de móveis.
2  // Sabendo que o deslocamento é igual a velocidade(constante) vezes o tempo.
3  // d=v*t
4  clear
5  clc
6  mprintf("Entre com a velocidade do movel e o tempo do percurso \n")
7  v= input("velocidade ")
8  t= [0:0.1;10]
9  d= v*t
10  plot(t,d)
11  title"Gráfico do deslocamento com relação ao tempo"
12
```

Gráfico do deslocamento em função do tempo e velocidade

grafico de S=So+VT.sce (C:\Users\RICARDO FREITAS\Documents\grafico de S=So+VT.sce) - SciNotes

Arquivo Editar Formatar Opções Janela Executar ?

grafico de S=So+VT.sce (C:\Users\RICARDO FREITAS\Documents\grafico de S=So+VT.sce) - SciNotes

apagar ta.sce | grafico de S=So+VT.sce | velocidade em relação ao tempo.sce | grafico de V=Vo+at.sce

```
1  // Programa para postar o gráfico da posição final ao longo do tempo com aceleração constante.
2  // equação do tipo S=So+ V*T.
3  clear
4  clc
5  mprintf("          Forneça ponto de partida, velocidade constante e o tempo.")
6  So= input("entre com posição inicial" )
7  V= input("entre com a velocidade constante" )
8  T= [0:0.1:6]
9  S= So+V*T
10 plot(T, S)
11 title"Gráfico da posição em relação ao tempo"
12
```

Gráfico da posição final ao longo do tempo com aceleração constante

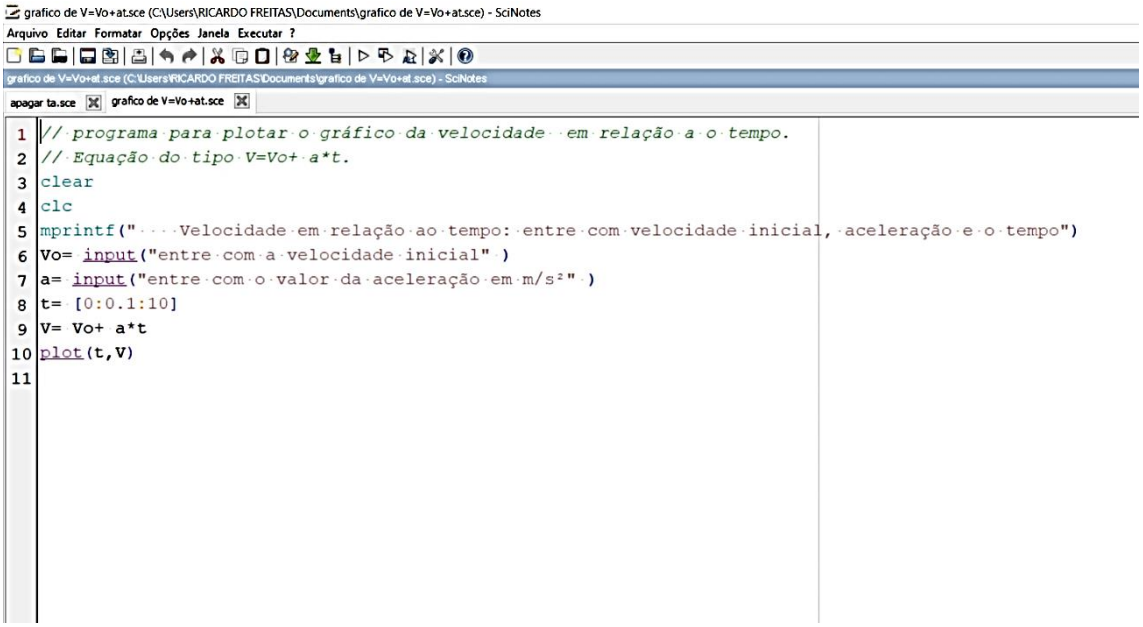


Gráfico da velocidade em função do tempo

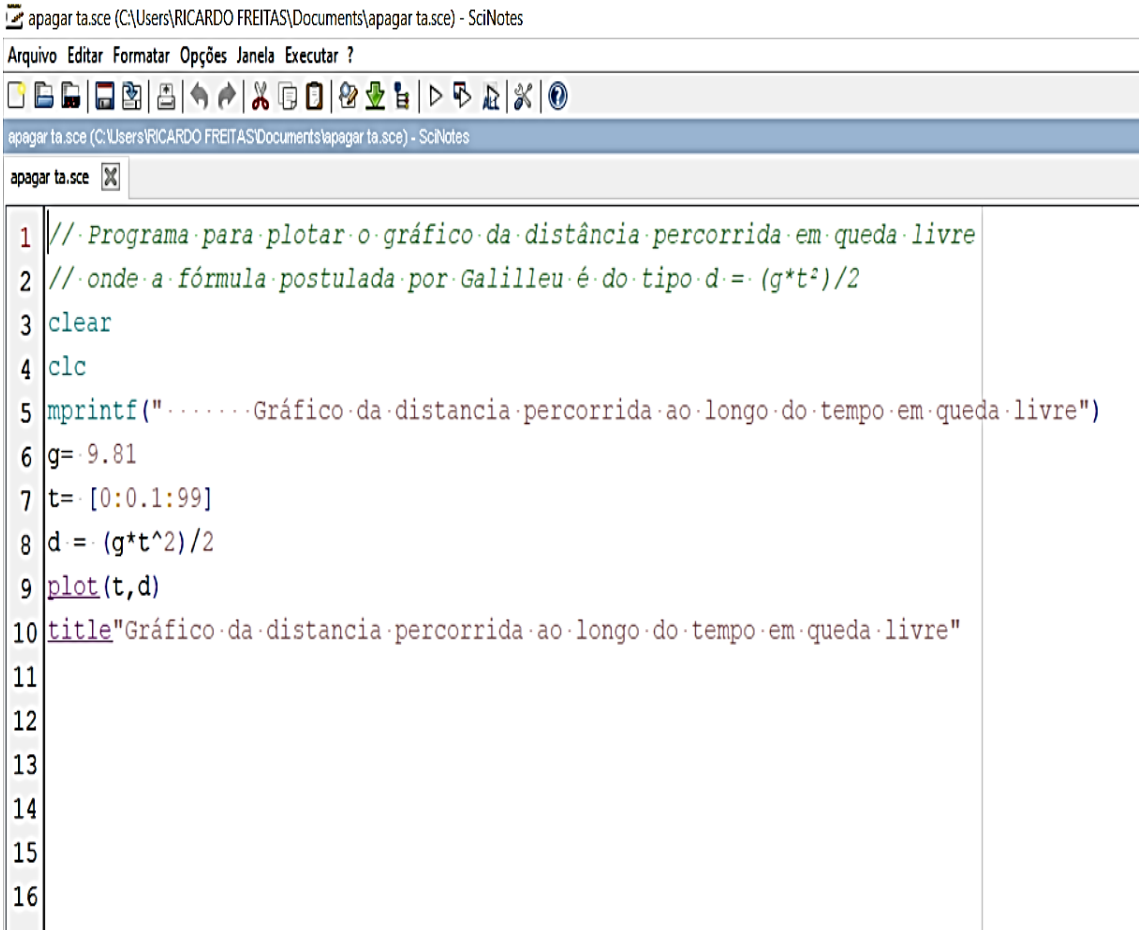


Gráfico da queda livre