Fernando J. Capeletto Neto
(No USP 2370518) (fc@usp.br)

LabTex - Interface entre LaTeX e LabView

São Paulo - SP, Brasil

<11 de dezembro de 2009 (Data da defesa da Monografia)>

Fernando J. Capeletto Neto
(No USP 2370518) (fc@usp.br)

LabTex - Interface entre LaTeX e LabView

Monografia apresentada na disciplina de Projeto de Formatura II (PSI2594) para obtenção do título de Engenheiro Elétricista com ênfase em Sistemas Eletrônicos pela Escola Politécnica da Universidade de São Paulo.

Orientador:
João Eduardo Kogler Jr. (kogler@lsi.usp.br)

Departamento de Sistemas Eletrônicos
PSI
Escola Politécnica da USP

São Paulo - SP, Brasil
<11 de dezembro de 2009 (Data da defesa da Monografia)>

Sumário

OK

Lista de Figuras

1 *Introdução*

Neste capítulo são apresentados objetivos e estrutura deste trabalho.

1.1 Objetivos

Desenvolver ferramenta de interface para LabVIEW que permitirá tornar executáveis expressões selecionadas e inicializadas dentro de um texto em LaTeX, renderizando resultados e gráficos gerados pelo LabVIEW dentro do documento LaTeX, visando a criação de arquivos técnicos ativos.

1.2 Introdução

As atividades nos campos da ciência e tecnologia se estabelecem e renovam-se por meio da pesquisa de novos métodos e conceitos. Em conseqüência disto há também a necessidade de documentar e divulgar os resultados e os avanços.

O projeto busca criar ponte facilitadora a essas atividades no campo da engenharia e das ciências. O objetivo é fazer a ponte tecnológica entre duas estruturas largamente utilizadas em cada um desses campos, que por suas características, oferecem caminho lógico da documentação científica ao experimento e dele novamente à documentação.

Será desenvolvido portanto um processador léxico para LaTeX, implementado em instrumento virtual de LabVIEW, com recursos e metodologias que serão descritos a seguir e com auxilio da criação de um pacote de biblioteca LaTeXcom a definição das tags que transmitirão as instruções adequadas ao funcionamento do processador.

LaTeXtem sido a via mais universalmente aceita e utilizada para a documentação científica nas últimas três décadas enquanto que no âmbito da experimentação científica, controle e teste industriais, o LabVIEW tem sido amplamente utilizado há 22 anos e seu uso continua crescendo.

Se por um lado o LaTeXé ferramenta ubíqua e de fácil uso para se expressar o pensamento científico, o LabVIEW mostra-se de modo complementar como uma ferramenta que idealmente facilita a transformação desse pensamento criativo em aplicações reais.

Por meio dessa interface de softwares uniremos processamento ao LaTeX(ferramenta de documentação) e documentação ao LabVIEW, ferramenta de processamento.No decorrer

deste trabalho serão descritas as características de ambas as estruturas, como são orientadas e tipadas de modo que permitem essa componentização, destacando seus aspectos complementares e definindo as estratégias e planos de trabalho para a execução do projeto em PSI2594, como será feita essa ponte tecnológica, necessidades que motivam este projeto, aspectos inovadores e vantagens na escolha dessas estruturas (LaTeXe LabView) como seus componentes.

1.3 Justificativas

Existe a necessidade de investigar o conhecimento documentado do mesmo modo que existe a necessidade de documentar o conhecimento adquirido, este projeto atua nessa mão de via dupla.

Existem no mercado ferramentas de software que se propõe a realizar em parte, o objetivo desse projeto, no tocante ao processamento de simulações.Uma referencia é o Mathcad - Enginering Calculation Software da Mathsoft, um software proprietário e que é um híbrido de documentação técnica com cálculo numérico e simbólico que permite explorar problemas, formular idéias, analizar dados, modelar e verificar cenários, determinar soluções; bem como também documentar, apresentar e comunicar resultados. Existem desvantagens no uso do Mathcad e entre elas está a dificuldade de integração com dispositivos de hardware, o fato de ter seu funcionamento sustentado em macros, o que faz com que o software seja interpretado e não compilado, gerando problemas com execução em tempo real. Existem outros softwares como o Maple da Maplesoft ou o Mathematica da Wolfram, porém são mais limitados também fogem da possibilidade de interação com operações em tempo real.

O projeto oferece uma alternativa que se baseia em código aberto na interface de documentação (TeX) e embora use um software proprietário na interface de processamento (LabVIEW) é ferramenta que tem sido das melhores e mais difundidas plataformas de desenvolvimento de aplicações científicas e tecnológicas e é existente na escola. São plataformas suficientemente difundidas para que um trabalho nessas justifique utilização no escopo desse projeto, embora haja conhecimento da existência de outras ferramentas e que no caso de uma implementação comercial este seja um tópico a ser explorado.

Em disciplinas da graduação os alunos da Escola Politécnica tem utilizado o LabVIEW nos experimentos de laboratórios e em pesquisas, em projetos de aplicação nas aulas e em projetos de formatura, desenvolvidos no curso de engenharia elétrica, entre eles onde esse projeto agrega. De maneira geral, o LabVIEW está disponível para utilização em todo o campus da Universidade de São Paulo (Campus Capital) por meio de parceria da Universidade com o serviço de relações acadêmicas da National Instruments (NI).

Este projeto justifica-se na necessidade de avanço nos processos documentacionais relacionados à instrumentação virtual e que auxiliem sua reprodução com maior facilidade e eficiência. E supre necessidades quando oferece a alternativa de maneira colaborativa e integrada aos padrões das comunidades difundidas.

Um caráter inovador identificado é a geração de documentos ativos por meio da criação de um arquivo técnico que pode ser dito 'vivo' porque possui existência ativa dentro do escopo das variáveis simuladas ou adquiridas em tempo de execução.

2 O Processador LaTeX

Neste tópico abordaremos o contexto no qual se enquadra o projeto a ser desenvolvido do ponto de vista técnico/científico, descrevendo as caracteristitcas principais dos elementos que o envolvem, ou seja, os objetos constituintes, tanto em estrutura como em componentes.

2.1 Estruturas LaTeX

2.1.1 Formatação (Typesetting)

A formatação envolve a apresentação do material textual em forma gráfica em papel ou algum outro meio de comunicação. Antes do advento da publicação por meio de desktops, tipografia de material impresso foi produzida por compositores de trabalhos manuais, e mais tarde por máquinas.

O princípio geral da tipografia independente do contexto permanece sendo a composição de glifos em linhas para formar organismo: assunto, títulos, legendas e outros pedaços de texto que compõem uma imagem de página, e imprimir ou transferir a imagem da página para o papel ou outras mídias.

Durante a época da impressão tipográfica, tipos móveis foram compostos à mão para cada página. Caracteres de metal eram compostos em palavras e linhas de texto e fortemente ligadas entre si para criar a imagem de uma página denominada 'forma', com todas as faces das letras possuindo o mesmo tamanho para formar uma superfície do mesmo tipo. A forma era montada em uma prensa com tinta e a impressão feita em papel.

Com a evolução dos tipos de tipografia acompanhando a era digital , a disponibilidade de criação de fontes por meio de procedimentos mais simples e baratos ou gratuitos, abriu um fosso entre designers profissionais e amadores. Com o advento do PostScript, complementado pelo formato de arquivo PDF, providenciou-se um método universal de verificação de desenhos e esquemas e legível na maioria dos computadores e sistemas operacionais.

O sistema TeX[11], desenvolvido por Donald E. Knuth, no final dos anos 70, é outro amplo e poderoso sistema de tipografia automatizada que estabeleceu padrões elevados, especialmente para tipografia de matemática. Sobre ele abordamos na seqüência.

2.1.2 TeX Typesetting system

Comandos TeXnormalmente começam com uma barra invertida ('\') e são agrupados com chaves. Entretanto, quase todas as propriedades sintáticas do TeXpodem ser modificadas instantâneamente o que torna suas entradas complexas para analizar por qualquer

sistema a não ser por ele próprio.

TEXé uma linguagem baseada em macros e símbolos: vários comandos, incluindo definições de usuário, são expandidas instantâneamente até que se encontrem apenas símbolos inexpansíveis remanescentes enquanto são executados.

A expansão própria é praticamente livre de efeitos colaterais, a recursão de macros não onera em memória e construtores do tipo if/else estão disponíveis. Isso torna TEXuma linguagem de Turing mesmo no nível expandido. (Turing é uma linguagem de programação baseada em Pascal desenvolvida em 1982 por Ric Holt e James Cordy, depois da Universidade de Toronto, Canadá. È uma linguagem descendente de Euclides, Pascal e SP/k e se caracteriza por uma sintaxe limpa e semântica precisa independente de máquina.)

O sistema pode ser dividido em quatro níveis: no primeiro, caracteres são lidos do arquivo de entrada e recebem a atribuição de um código de categoria (chamado de 'catcode'). Combinações da barra invertida ('\' - qualquer caracter da categoria zero) seguido por letras (caracteres da categoria 11) ou um único outro caracter, são substituídos por um marcador controlador de seqüência.

Nesse contexto, essa etapa é como análise lêxica, no próximo estágio, seqüências de controle expansíveis (como as condicionais e macros definidas) são substituídas por seus textos de substituição. A entrada do terceiro estágio é então uma seqüência de caracteres (incluindo aqueles com significado especial) e seqüências de controle inexpansíveis (tipicamente sinalizações e comandos visuais). Aqui caracteres são montados dentro de um parágrafo. O quarto estágio insere as listas de linhas verticais e outros materiais dentro das páginas.

O sistema TEXpossui conhecimento preciso dos tamanhos de todos os caracteres e símbolos e usando essa informação, processa o arranjo ótimo de letras por linha e de linhas por página. Isso então produz um arquivo DVI ("Device Independent - Independente do Dispositivo) contendo a localização final de todos os caracteres. Esse arquivo dvi pode ser impresso diretamente fornecido um driver de impressão apropriado, ou pode ser convertitdo para outros formatos. Nos tempos atuais é geralmente usado o PDFTex que processa a geração do DVI para PDF em conjunto.

2.1.3 Linguagens de Macro

TeXoferece uma linguagem de macro não usual, a definição de uma macro não apenas inclui uma lista de comandos mas também a sintaxe de chamada. Macros são integradas em larga escala com a linguagem interpretada em tempo de compilação o que também guia o processamento.

O nivel da operação da macro TeXé léxico, mas é construído sobre suas facilidades que fazem uso da interpretação sintática.A linguagem macro em TeXfoi utilizada com sucesso para extender TeXpor exemplo para LaTeX.

2.1.4 LaTeX

LaTeXé mais amplamente utilizada por matemáticos, cientistas, engenheiros, filósofos, estudantes acadêmicos e no mundo comercial e outros segmentos. é utilizado devido sua alta qualidade de formatação atingida pelo TeX. O sistema de formatação oferece ferramentas programáveis de publicação em desktops e facilidades extendidas para automação da maioria dos aspectos de formatação e publicação em desktops, incluindo numeração e referência cruzada, tabelas e figuras, layout de páginas e bibliografias.

LaTeXfoi criado com a intenção de oferecer uma linguagem de alto nível que acessasse o poderio do TeX. LaTeXessencialmente compreende uma coleção de macros em TeXe um programa para processar documentos LaTeX. Foi originalmente escrito no inicio dos anos 80 por Leslie Lamport [14] na SRI International, e se tornou o método dominante para o uso do TeX, poucas pessoas relativamente escrevem em TeXpuro ainda. A versão corrente é chamada LaTeX2_ε.

O termo LaTeXrefere-se apenas a linguagem na qual os documentos são escritos e não ao editor usado para escrever estes documentos. Para criar um documento em LaTeX, um arquivo do tipo .tex precisa ser criado usando algum tipo de editor de texto. Enquanto vários editores de texto servem ao propósito, várias pessoas preferem utilizar editores desenvolvidos especificamente para trabalhar com LaTeX.

Distribuído sob os termos da licença 'LaTeXProject Public License' (LPPL), LaTeXé software livre, detalharemos a questão da licensa na Análise de Viabilidade.

2.1.5 Sistema de Formatação LaTeX

LaTeXé baseado na idéia de que autores devem estar aptos a focar no conteúdo que estão escrevendo sem serem distraídos pela sua apresentação visual. Ao preparar um documento LaTeX, o autor especifica a estrutura lógica utilizando conceitos familiares como 'capítulo', 'seção', 'tabela', 'figura', etc; e deixa que o sistema LaTeXpreocupe-se a respeito da apresentação dessas estruturas.

Isso encoraja portanto a separação de layout do conteúdo enquanto permite ajustes de formatação quando necessário. é similar ao mecanismo de estilos permitidos em vários processadores de texto, definidos globalmente para um documento inteiro, similar ao mecanismo do CSS usado pelo HTML.

O exemplo abaixo mostra a entrada LaTeXe a correspondente saida:

```
\documentclass[12pt]{article}
\usepackage{amsmath}
\title{\LaTeX}
\date{}
\begin{document}
\maketitle
 \LaTeX{} is a document preparation system for the \TeX{}
  typesetting program. It offers programmable desktop publishing
  features and extensive facilities for automating most aspects of
  typesetting and desktop publishing, including numbering and
  cross-referencing, tables and figures, page layout, bibliographies,
  and much more. \LaTeX{} was originally written in 1984 by Leslie
  Lamport and has become the dominant method for using \TeX; few
  people write in plain \TeX{} anymore. The current version is
  \LaTeXe.
  % This is a comment, it is not shown in the final output.
  % The following shows a little of the typesetting power of LaTeX
  \begin{align}
   E&=mc^2 \\
```

```
    m&=\frac{m_0}{\sqrt{1-\frac{v^2}{c^2}}}
  \end{align}
\end{document}
```

LaTeX

LaTeX is a document preparation system for the TeX typesetting program. It offers programmable desktop publishing features and extensive facilities for automating most aspects of typesetting and desktop publishing, including numbering and cross-referencing, tables and figures, page layout, bibliographies, and much more. LaTeX was originally written in 1984 by Leslie Lamport and has become the dominant method for using TeX; few people write in plain TeX anymore. The current version is LaTeX 2_ε.

$$E = mc^2 \tag{1}$$

$$m = \frac{m_0}{\sqrt{1 - \frac{v^2}{c^2}}} \tag{2}$$

Figura 1: Figura proveniente do arquivo `exemplo01.jpg` gerado da compilação do código da página anterior

2.1.6 Apanhado Geral da Estrutura LaTeX.

Em resumo, o centro de processamento do LaTeXé a linguagem de programação chamada TeXque provê várias instruções de formatações. Em conjunto com o TeXvem um conjunto de fontes denominadas 'Computer Modern (CM). As CM fontes e a linguagem TeX formam a base de um sistema TeXtípico.

TeXé expansível, ou seja, comandos adicionais podem ser definidos em termos de outros mais básicos. Uma das melhores conhecidas expansões do TeXé o LaTeXque introduz a idéia de unidade lógica.

O layout visual no LaTeXé determinado pela classe do documento. Expansões do LaTeXsão denominadas pacotes, nas seções a seguir abordamos os elementos de um documento LaTeX, sua estrutura e componentização passando ao contexto específico dos pacotes AMS - LaTeX (amssymb e amsmath), apropriados para a representação tipográfica matemática e científica em geral. Ao final do capítulo são apresentadas agumas primeiras estruturas num apanhado das mais representativas da biblioteca.

A componentização resumida dos elementos da estrutura LaTeXé ilustrada na figura abaixo. Essa figura sugere que, em ordem para trabalhar com um documento LaTeX, é necessário primeiro instalar o TeXe as fontes 'Computer Modern', então instalar o LaTeXe finalmente especificar a classe de documento e os pacotes necessários.

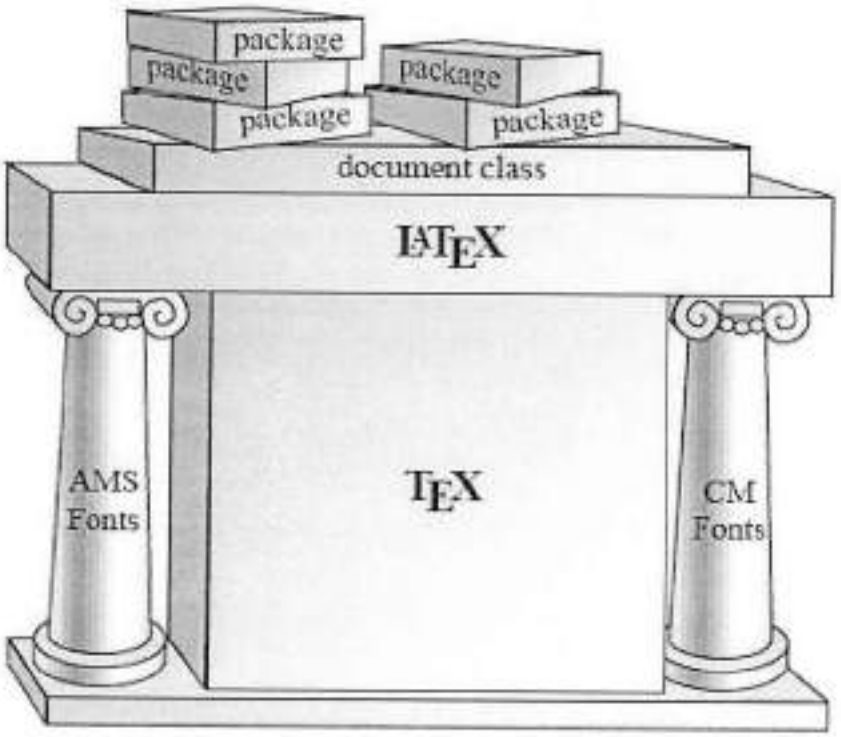


Figura 2: Estrututra de relação entre o TeX e o LaTeX. O TeX é a fundação sobre a qual vários pacotes utilitários - extensões do LaTeX - são construídos.

2.1.7 Classes e Pacotes LaTeX

LaTeXem sua versao atual, faz distinção entre as macros que definem o layout de um documento, e as macros que extendem as funcionalidades para fornecer o que o autor realmente quer.

A idéia é que um documento de classe LaTeXdiz sobre qual a natureza do documento, enquanto os pacotes do documento carregam comandos que refinam sobre a especificação global.

No disco, os arquivos só aparecem diferentes em virtude da extensão - arquivos de classes são do tipo *. cls enquanto arquivos de pacotes são do tipo *.sty. Desse modo, encontramos o arquivo classe padrão para artigo no disco com o nome de 'article.cls', enquanto que o pacote footmisc (que refina as definições de notas de rodapé dos artigos) é representado no disco por um arquivo chamado footmisc.sty

O usuário define a classe do seu documento com o comando `\documentclass`(normalmente o primeiro comando em um documento), e carrega pacotes com o comando `\usepackage`. Um documento pode ter vários comandos usepackage, mas pode ter apenas um `\documentclass`. Existem versões da interface de programação de ambos os comandos, já que uma classe pode escolher carregar outra classe para aperfeiçoar as suas capacidades, e ambas as classes e pacotes podem optar por carregar outros pacotes.

2.1.8 Ajustando comandos existentes

No caso geral, colocar uma sobredefinição no meio de um comando existente é difícil. No entanto, para adicionar algum código no início ou no fim de um comando existente, é conceitualmente muito fácil. Para definir uma versão de um comando que faz alguma pequena extensão da sua definição original podemos escrever:

```
\renewcommand{\splat}{\mumble\splat}
```

Entretanto, isso não deverá funcionar: uma chamada a `\splat` executaria `\mumble` , e a chamada que redefine `\splat` novamente; Isso é um loop infinitamente recursivo, que irá rapidamente exaurir a memória do TeX. Felizmente, o comando TeX`\let` nos

permite ter um 'instantâneo' do estado atual de um comando, podemos, então, usa-lo na sua redefinição. Então:

```
\let\OldSmooth\smooth
```

e

```
\renewcommand{\smooth}{\mumble\OldSmooth}
```

possuem os mesmos efeitos da correção necessária, de forma segura. Adicionando coisas no fim de um comando funciona de modo similar. Se `\smooth` tem argumentos, eles devem ser passado por referência:

```
\renewcommand{\smooth}[2]{\mumble\OldSmooth{#1}{#2}}.
```

O caso geral pode ser alcançado de duas maneiras. Em primeiro lugar, pode-se usar o comando LaTeX `\CheckCommand`; este compara um comando existente com a definição que você der a ele, e emite um aviso se dois não são iguais. Sua utilização é, portanto:

```
\CheckCommand{\complex}{<original-definition>}
```

```
\renewcommand{\complex}{<new-definition>}.
```

Esta técnica é, obviamente, um pouco trabalhosa, mas se o comando original vem de uma fonte que é passível de mudança em âmbito do contexto, ele ao menos avisa que seu patch está em perigo de funcionar equivocadamente.

Existe a possibilidade de se usar pacotes específicos para a finalidade de redefinição de comandos:

O pacote patchcmd restringe o conjunto de comandos que você pode ajustar; O pacote ted [19] é um 'editor de lista de tokens' e provê um comando `\Substituteo` qual ajusta o conteúdo de uma macro, colocando o resultado em uma lista de tokens ou opcionalmente usando o resultado para (re)definir a macro. O pacote pode ser usado também como ferramenta de debugging.

O pacote etoolbox [16] provê o comando `\patchcmdo` qual realizará uma 'substituição de string' na definição de uma macro. O pacote também prove commandos que adicionam pré-definições (`\pretocmd`) ou pós-definições (`\apptocmd`) a uma definição de um comando.

Nesse estágio do trabalho abordamos superficialmente alguns dos comandos e pacotes que julgamos que serão úteis ao desenvolvimento das tags do LabTeX devido suas funcionalidades que permitem flexibilização sobre a sintaxe das macros LaTeXe oferecem suporte ao ensaio das tags escolhidas como marcadores para o LabTeX, inclusive com possibilidade de debugging. São recursos que auxiliam a metodologia e no decorrer do desenvolvimento do projeto em PSI2594, quando forem utilizados, cada pacote será apresentado e mencionado em seu contexto de uso.

2.1.9 Analise de Viabilidade TeXe LaTeX

Nesta seção analizaremos questões que envolvem os componentes e suas características que viabilizam o projeto, como integração, aceitabilidade, estabilidade, escalabilidade.

TeXfoi desenvolvido com dois objetivos principais em mente: permittir que qualquer pessoa produzisse livros de alta qualidade usando quantidade de esforço razoável e prover um sistema que pudesse gerar exatamente o mesmo resultado em todos os computadores, agora e no futuro. Tex possui excelentes capacidades de formatação e trabalha com fórmulas matemáticas tão bem como texto.

Um grande atrativo da linguagem TeX é que seu arquivo código é puro texto, algumas vezes denominado um arquivo 'ASCII'. Portanto, artigos contendo mesmo as mais complicadas expressões matemáticas podem ser legíveis transmitidas eletronicamente, para colegas, co-autores, journals, editores e publicadores.

TeXé independente da plataforma, admite desenvolvimento em um Macintosh por exemplo e seu co-autor pode fazer melhorias no mesmo arquivo usando um computador pessoal compatível com IBM; o publicador do journal pode usar um DEC minicomputador, essa diferença de hardware não muda em nada o processo.

LaTeXé muito mas fácil para trabalhar do que TeX, existe grande número de ferramentas seguras construídas internamente e um enorme conjunto de mensagens de erro.

LaTeXdiminui o tédio na escrituração de tarefas. Considere um artigo completo com teoremas e equações numeradas e apropriadamente referenciadas cruzadamente. Ao fim da leitura algumas mudanças precisam ser feitas, por exemplo, a seção 4 precisa ser colocada

após a seção 7 e um novo teorema precisa ser inserido em algum lugar no meio.Esses tipos de modificações no processamento documentacional convencional pode causar transtornos enquanto que com LaTeXisso se torna simples pois automaticamente refaz todas as numerações e referencias cruzadas.

LaTeXé uma linguagem universal e estabelecida, acompanhada por communidades científicas que desde os fins dos anos 60 pesquisavam alternativas que pudessem oferecer recursos ao desejo de expressão. É acessível, possui protocolo aberto e poucas tecnicalidades de modo que o leigo instruído é capaz de desenvolver e de modo geral preservar, documentar e publicar conhecimento.

Em diversos campos técnicos em particular a ciência da computação, matemática e físicas, LaTeXtem se tornado padrão. Vários milhares de livros tem sido publicados usando LaTeX, incluindo livros publicados pela Addison-Wesley, Cambridge University Press, Elsevier, Oxford University Press and Springer.Diversos journals nesses campos são produzidos usando TeX or LaTeX, permitindo que autores possam submeter seus trabalhos manuscritos diretamente em TeX, o que é inclusive eletronicamente mais leve quanto a transmissão eletrônica e armazenamento em base de dados.

A estabilidade do TeXtambêm merece ser ressaltada, desde a versão 3, TeXusa um idiosincrasso sistema de numeração de versão onde atualizações são indicadas pela adição de um dígito extra ao final do decimal, de modo que a versão do número assintoticamente se aproxime de 'pi'. Isso é uma reflexão do fato que TeX é atualmente muito estável e apenas mínimas atualizações são antecipadas. A versão atutal do TeXé 3.1415926; que foi atualizada pela última vez em Março de 2008.

O criador do TeX, Donald Knuth, mantevem um registro muito detalhado dos bugs corrigidos e mudanças efetuadas desde 1982. A lista atualizada em 2008 contém 427 registros. Knuth ofereceu premios em dinheiro às pessoas que encontrassem e reportassem um bug no TeX. O premio por bug começou com U$ 2.56 (um 'dolar hexadecimal') e dobrava todo ano até que seu valor foi congelado para o valor atual de U$ 327.68. Knuth entretanto, perdeu relativamente pouco dinheiro com isso pois muitos poucos bugs foram anunciados.

Quanto à Licença LaTeXé tipicamente distribuído com texto puro, isso é distribuído sob uma licença de software livre, a 'LaTeX Project Public Licença (LPPL). A LPPL não

é compatível com a GNU - Licença Pública Geral, pois a primeira requer que os arquivos modificados sejam claramente diferenciados um do outro (normalmente mudando o nome do arquivo). Para reforçar essa regra, qualquer implementação no sistema precisa passar por bateria de testes antes de ser liberada para ser chamada TeX.

Quanto ao software para codificação, LaTeXé disponível na maioria dos sistemas operacionais incluindo Unix, Linux e BSDs, Windows, Mac OSx, RISC OS e AmigaOS.

Outro aspecto que fortalece a viabilidade e a escolha do LaTeXcomo componente é a organização das comunidades ao redor dessas tecnologias. Entidades da comunidade TeXfazem parte do TeXUsers Group o qual publica o 'TUGboat' e 'The PracTeX Journal' abrangendo um vasto leque de temas relevantes para a tipografia digital TeX. O 'The Deutschsprachige Anwendervereinigung TeX' é um grande grupo de usuários na Alemanha. O 'TeXUsers Group' foi fundado em 1980 para fins educacionais e científicos, possibilita uma organização para aqueles que possuem algum interesse em tipografia e design de fontes, e são usuários do sistema tipográfico TeX.

2.2 AMS-LaTeX

O pacote amsmath é um pacote LaTeXque oferece diversos acessórios para melhorar a estrutura da informação e impressão de documentos que contêm fórmulas matemáticas. $\mathcal{AMS}$-LaTeXé uma coleção de classes de documento e pacotes em LaTeXelaborados pela American Mathematical Society ($\mathcal{AMS}$) e que provê soluções e recursos para diversos fins de documentação científica em diversos segmentos.

Entre os acessórios matemáticos há uma grande variedade de ferramentas tipográficas. $\mathcal{AMS}$-LaTeXprove diversas classes incluindo a classe de documentos para artigos da $\mathcal{AMS}$, a classe amsart e disponibiliza fontes para centenas de operadores binários, relações binárias, relacionamentos negativos binários, símbolos negritos, setas, setas extensíveis, e assim por diante, disponibilizados pelo $\mathcal{AMS}$-LaTeXe que também disponibiliza alfabetos matemáticos adicionais como Blackboard negrito, Euler Fraktur, Euler Script, e matemático negrito itálico.

O pacote amsmath é distribuido em conjunto com alguns pequenos pacotes auxiliares,

outro pacote que sera útil no escopo do projeto pois dará liberdade para a criação de funcionalidades do LabTeX será o pacote amsopn que provê `\DeclareMathOperator` para definição de novos 'nomes de operadores' como `\sin` e `\lim` .

O pacote amsmath provê uma série de outras estruturas de equação além das fornecidas no LaTeXbásico. O conjunto adicional inclui: equation , equation* , align , align*, gather, gather*, flalign, flalign*, multline, multline*, alignat, alignat* e split. [6]

2.2.1 Classes de operação dos símbolos matemáticos

Os simbolos de uma fórmula matemática são classificados em diferentes categorias que correspondem mais ou menos à parte da fala ou elemento léxico que cada símbolo teria se a fórmula estivesse expressas em palavras,(por exemplo: nomes, conjunções e verbos dispostos numa declaração, só que trata-se de 'declaração matemática') Certos modos de espaçamentos e posicionamentos são tradicionalmente utilizados para as diferentes classes de símbolos para aumentar a legibilidade das fórmulas.

Class number	Mnemonic	Description (part of speech)	Examples
0	Ord	simple/ordinary ("noun")	$A\ 0\ \Phi\ \infty$
1	Op	prefix operator	$\sum\ \prod\ \int$
2	Bin	binary operator (conjunction)	$+\ \cup\ \wedge$
3	Rel	relation/comparison (verb)	$=\ <\ \subset$
4	Open	left/opening delimiter	$(\ [\ \{\ \langle$
5	Close	right/closing delimiter	$)\ ]\ \}\ \rangle$
6	Pun	postfix/punctuation	$,\ .\ ;\ !$

Figura 3: Correspondência das categorias de simbolos e seus códigos de operação.[6]

A distinção entre a classe 0 e uma adicional classe 7 é unicamente com os aspectos de seleção de fonte e não se aplica aqui. Simbolos da classe binária são automatitcamente forçados à classe 0 (sem espaços) se eles não possuírem um operando à esquerda. Números arábicos (0-9) são classe 0, assim como as letras gregas, outros símbolos alfabéticos e diversos símbolos acessórios simples também são da classe 0 (ordinária) o que significa

que eles não têm qualquer base de espaçamento. Lexicamente correspondem a nomes e pronomes no 'discurso',sob o ponto de vista dessa metáfora.

Simbolos de operadores binarios são classe 2 (conjunções), simbolos relacionais são classe 3(verbos). Operadores cumulativos (de tamanho variável) são da classe 1. Símbolos de pontuação são da classe 6, pares delimitadores são classe 4 (os símbolos de abertura) e classe 5 (os símbolos de fechamento). Os acentos serão úteis para especificadores de arrays, vetores e multidimensoes no LabTeX por exemplo. Nomes de operadores existentes são representados por abreviações. Existem muitos símbolos adicionais disponíveis para uso no LaTeX. Existe uma lista de símbolos detalhada em [18].

Observa-se portanto uma relação entre a classe dos comandos $\mathcal{AMS}$-LaTeXe a natureza desses comandos quanto a sua função, espaçamento e necessidade de parâmetros construtores, o que será explorado na construção dos 'métodos-subVIs' de 'seleção de operação' (opcode 'fetch'), ou seja, os sub instrumentos virtuais do LabTeX capaz de identificar o tipo do comando/operação matemática para posterior preparação (parse) e execução, conforme é apresentado durante a metodologia. 3.1.1

2.3 Gráficos LabTeX: Recursos pesquisados

LaTeXnasceu em 1984 com um ambiente nativo de desenho que permitiu desenhar linhas, vetores, círculos e 'ovais' com algumas limitações; fontes especiais foram utilizadas para desenhar todos os objetos gráficos. Esta foi uma grande limitação, porque no surgimento do LaTeXtodas as fontes utilizáveis comportavam apenas 128 glifos; Isso aconteceu com as fontes de texto, bem como com as fontes especiais para desenho em LaTeX. Com isso só haviam duas espessuras disponíveis.[4]

A primeira extensão gráfica disponível no mercado foi PiCTeX[29]. Foi concebido para trabalhar com TeXpuro, em uma fase inicial da história do TeX. Após LaTeXser disponibilizado, algumas macros foram criadas para permitir que LaTeXimportasse as macros do PiCTeXe pudesse utilizá-las com quase toda eficiencia. Nesses tempos, TeXe LaTeXeram executados em mainframes e PCs estavam na sua infância, o sistema TeXjá estava disponível para os pequenos PCs, mas as limitações de memória eram tão fortes que era normal obter

a mensagem de excesso de memória e processamento abortado.

Em 1986 Sunil Podar publicou seu pacote de extensão gráfica chamado epic [20], para o ambiente gráfico LaTeX convencional; em 1988 Conrad Kwok publicou o seu reforço eepic [13]. Ambos foram concebidos de forma a aliviar as fortes limitações do ambiente gráfico padrão do LaTeX, nomeadamente a limitação das encostas de linha e dos vectores e a gama limitada de raios de circulo. Ambos os programas foram concebidos para utilização no âmbito do 'velho' LaTeX, hoje conhecido como LaTeX 209, em contraste com o 'novo' LATEX2# (que não é mais tão novo uma vez que posui mais de uma década de existência).

Outro pacote gráfico muito poderoso que necessita da utilização de um software externo atualmente para renderizar objetos gráficos é XY-pic, [21]. Este pacote é projetado para trabalhar com LaTeX, TEXpuro, e $\mathcal{A}_{\mathcal{M}}\mathcal{S}$-TEX.

Leslie Lamport, em sua segunda edição do manual LaTeX [15], fixou a sintaxe de um novo ambiente de desenho extendido, (pict2e e seu pacote de extensão curve2e [3] extensões) onde a maioria se não todas as limitações da implementação padrão puderam ser superados: ilimitadas encostas das linhas e vetores, qualquer raio de círculo, linhas de espessura arbitrária também para linhas curvas, etc.

2.3.1 Pacotes adotados: PGF/TikZ e xcolor

A sigla PGF significa 'Formato Gráfico Portável'; o pacote PGF [25], implementa ou re-implementa todos os comandos do ambiente padrão de desenho LaTeX, de modo a criar um conjunto coerente de comandos capazes de fazer a maioria das operações gráficas no LaTeXe no pdfLATEX.

Vários comandos foram adicionados ao conjunto fundamental para extender as capacidades gráficas do LaTeXe suas capacidades são ampliadas mais uma vez por meio do pacote xcolor que por sua vez, amplia as possibilidades de manipulação de cores oferecido pelo pacote padrão.

Esse pacote automaticamente examina os arquivos de configuração padrões e insere nos arquivos de saída o comando `\special` adequado ao driver de saída.

O novo pacote PGF, versão 1.10, contem um novo pacote e seu novo ambiente gráfico

que é chamado TikZ; acrônimo que significa 'TikZ ist kein Zeichenprogramm' (TikZ não é um programa de desenho). PGF e seu novo pacote tikz possuem inúmeras interfaces de desenho que podem produzir quase qualquer coisa. O manual explica em detalhes o que não pode ser feito com a interface de PostScript e explica que o formato PGF e o programa foram especificamente desenvolvidos para serem utilizados com pdfLATEX, onde possui seu melhor funcionamento.

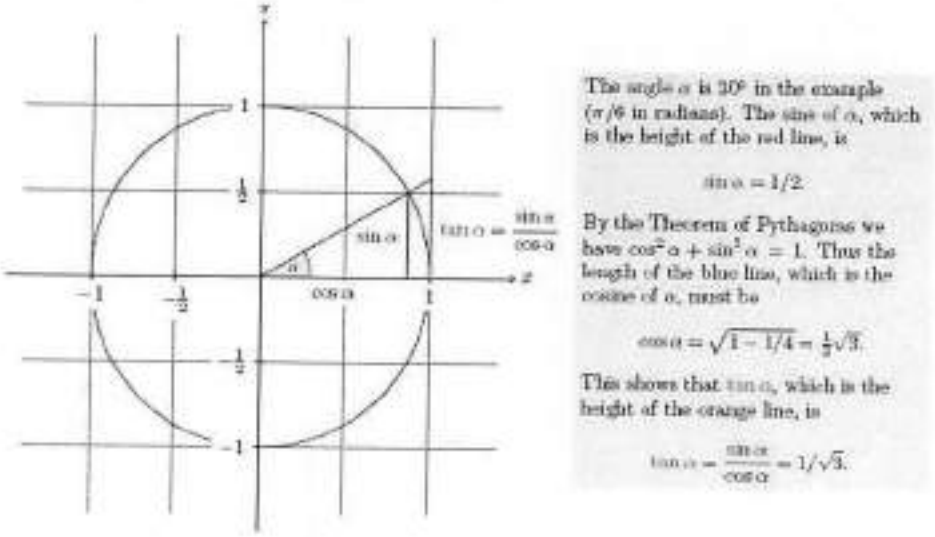


Figura 4: Um exemplo retirado do Manual de Tantau [25]

Tantau diz que TikZ possui a intenção de oferecer ao usuário uma interface uniforme e simplificada que permita a composição de desenhos com o mínimo possível de estardalhaço; em seu manual ele escreveu uma pequena introdução ensinando como produzir demonstrações como a figura produzida acima. O código não é transcrito neste trabalho pois pode ser encontrado por qualquer um pelo manual [25], e que deverá ser aproveitado intensamente para o desenvolvimento e implementações dos métodos-SubVIs de geração de código gráfico, devido à multiplicidade de desenho e comandos disponíveis para qualquer eventual situação.

PGF/TikZ possui extensa documentação. Para a versão 2.0 o manual possui 560 páginas. Como nota-se na figura, o programa permite gráficos em preto e branco ou coloridos, e o texto usa as mesmas fontes que as padrões no documento PDF pois TikZ é completamente integrado com (pdf)LaTeX.

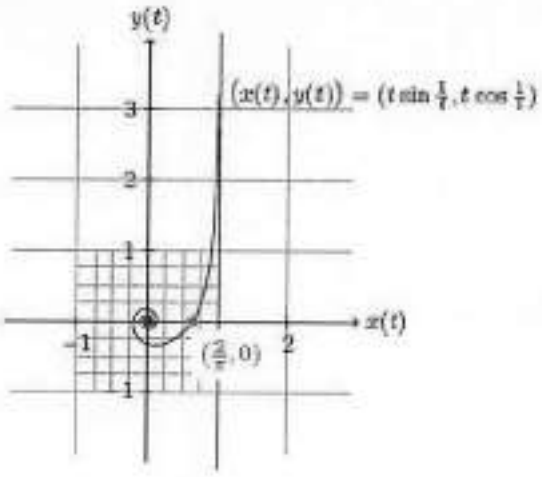


Figura 5: Uma espiral plotada pelo TikZ cujas coordenadas foram calculadas pelo GnuPlot. Manual de Tantau [25].

O pacote PGF contem numeras bibliotecas de commandos adicionais; Existe grande variedade de tipos de setas, variedades de panos de fundo, oferece diagramas de relacionamento de entidades, diagramas para mapas mentais, opções de escolha para padrões de panos de fundo para desenhos técnicos, desenhos de redes de Petri e mais.

O pacote manipula o desenho de gráficos que podem ser definidos como uma serie de dois ou três pares ou tripletes coordenados para fazer desenhos 3D. Uma novidade é a possibilidade de crar desenhos 'inline' simples sem precisar abrir um ambiente. A sintaxe TikZ é simples; suas instruções iniciam com um comando, continuam com opções e coordenadas e terminam com um ponto-e-virgula.

Certamente o ambiente básico padrão LaTeX para desenhos sofre em comparação de performance com o pacote PGF, mesmo se o ambiente padrão for evoluído com os pacotes pic2e e curve2e, essas ferramentas simples podem ser utilizadas para aprender apenas poucos commandos.TikZ possui largo conjunto de funcionalidades mas possui tambem uma curva de aprendizado mais íngreme.

Nesta seção, pesquisamos o histórico e peculiariedades de algumas soluções existentes para geração de desenhos, gráficos e diagramas com LaTeXe nesta etapa do trabalho por comparação das estruturas e disponibilidade de recursos, escolheu-se pautar o desenvolvi-

mento das tags gráficas LabTeX sobre os protótipos do pacote PGF/TikZ por acreditar que ele prove desenhos técnicos cuja estrutura não foram encontradas em outras interfaces, durante a comparação dos resultados desses diversos pacotes. No decorrer dos artefatos de PSI2594 serão escolhidas e apresentadas as estruturas disponíveis para ativação do LabTeX, por meio específico do SubVI-método de concatenação de gráficos, conforme é apresentado durante a metodologia em 3.2.

2.4 Estruturas LabVIEW

2.4.1 Instrumentação virtual

Uso de software customizavel e módulos de hardware de medidas para criar sistemas de medição definidos pelo usuário, chamados de 'instrumentos virtuais'.

Tradicionalmente sistemas de instrumentação de hardware são constituídos por componentes de hardware pré-definidos como os multímetros digitais e osciloscópios que são completamente específicos quanto a natureza do estímulo, análise, função ou medição. Devido às suas funções dedicadas de hardware, esses sistemas são mais limitados em versatilidade do que sistemas de instrumentação virtual

A diferença primária entre instrumentação via hardware e instrumentação virtual é que são utilizados componentes de software para substituir uma grande quantidade de hardware. O software permite que complexos e custosos sistemas em hardware sejam substituídos por dispositivos computacionais já existentes, por exemplo conversores analógico-digitais podem atuar como hardware complementar de um osciloscópio virtual.

2.4.2 LabVIEW

LabVIEW (abreviação para *Lab*oratory *V*irtual *I*nstrumentation *E*ngineering *W*orkbench) é uma plataforma e ambiente de desenvolvimento para uma linguagem de programação visual da National Instruments. A linguagem gráfica é denominada 'G'. Originalmente lançada pela Apple Macintosh em 1986, LabVIEW é commene utilizado para aquisição de dados, controle de instrumentos e automação industrial numa variedade de plataformas incluindo Microsoft Windows, UNIX, Linux e Mac OS. A ultima versão do LabView é a 8.6.1 lançada em fevereiro de 2009.

2.4.3 Programação em Fluxo de Dados

A linguagem de programação utilizada no LabVIEW, também referenciada como 'G', é uma linguagem de programação de fluxo de dados. A execução é determinada pela estrutura de um diagrama em bloco gráfico (o código fonte em LabVIEW) sobre o qual o

programador conecta diferentes nós de funções por meio de conectores desenhados. Esses conectores propagam variáveis e qualquer nó pode executar tão logo todas as entradas de dados tornem-se disponíveis. Desde que isso pode ser o caso de múltiplos nós simultâneos, a linguagem G é inerentemente capaz de execução paralela. Multi-processamento e hardware multi-threading é automaticamente explorado pelo escalador de tarefas (built-in scheduler) o qual multiplexa múltiplas threads do sistema operacional sobre os nós prontos para execução.

2.4.4 Programação Gráfica

LabVIEW enlaça a criação de interfaces de usuário (denominada painel frontal) ao redor do ciclo de desenvolvimento.Programas e Subrotinas em LabVIEW são denominadas 'instrumentos virtuais' (Vis). Cada VI possui três componentes: um diagrama em blocos, um painel frontal e um painel de conectores. O ultimo é usado para representar o VI dentro do diagrama em blocos de outro, chamando VIs.

Controles e indicadores no painei frontal permitem que um operador insira dados ou extraia dados de um instrumento virtual em execução. Entretanto, o painel frontal tambem pode servir como uma interface programada. A abordagem gráfica também permite que não programadores possam construir programas simplesmente arrastando e soltando representações virtuais de equipamento laboratorial com o qual eles já estão familiarizados.

O ambiente de programação do LabVIEW, que inclui exemplos e documentação, torna simples a criação de pequenas aplicações. Esse é um beneficio por um lado mas existe também um certo perigo ao subestimar o conhecimento necessário para a boa qualidade da programação 'G'. Para algoritmos complexos ou códigos de alta escala, é imporatnte que o programador possua um extenso conhecimento de sintaxes especiais de LabVIEW e a topologia do gerenciamento de memória.

LabView é utilizado também para comunicação com hardware como aquisição de dados, aquisição visual e dispositivos de controle de movimento, e dispositivos GPIB, PXI, VXI, RS-232 e RS-484. LabVIEW tambem possui ferramentas construidas para conectar sua aplicação à Web usando o LabVIEW Web Server e padrões de softtware como TCP/IP, redes e ActiveX. Como isso é possível a criação de aplicações distribuídas, as

quais comunicam-se por um esquema cliente/servidor, e portanto facilitam a implementação de código paralelo inerente ao código em linguagem 'G'.

Usando LabVIEW pode-se criar, testar e medir, aquisição de dados, instrumentos de controle, analise de medidas e aplicações de geração de relatórios,entre outras aplicações.

2.4.5 Analise de Viabilidade

Nesta seção estudamos as viabilidades do LabVIEW suas características de orientação a objeto, programação multithreading, licenças, portabilidade, escalabilidade, e outros aspectos relacionados.

LabVIEW foi desenvolvido pela National Instruments há mais de 20 anos e está presente em diversos cenários, desde automação industrial, aplicações biomédicas, de sistemas em campo a sistemas embarcados, a grandes laboratórios e empreendimentos. Seu fôlego inicial surgiu entre os profissionais da área de biológicas que buscavam uma maneira de descrever suas necessidades com programação rápida e facil e resultados imediatos em tempo real, instrumentaçao.

O LabVIEW busca o caminho da conversao de ideias em realidade mas não no paradigma da documentação (como os que desenvolveram e desenvolvem as estruturas em LaTeX) e sim no aspecto da representação de suas necessidades instrumentais. Desse modo novamente esses dois componentes apresentam-se caminhando em paralelo no progresso científico e na medida que as necessidades de um lado são supridas pelas viabilidades do outro componente do par, está criado um caminho para a implementação do projeto.

Quanto a Programação Modular e a natureza expansivel, novamente os dois componentes possuem similaridade quanto a encapsulação em pacotes, no caso do LabVIEW, seu poder reside na natureza hierárquica do VI. Depois de criar um VI, ele pode ser utilizado no bloco diagrama de outro VI. Não há limite no número de camadas na hierarquia.O uso de programação modular ajuda a gerir as alterações e depurar o diagrama em blocos rapidamente. Um subVI corresponde a uma subrotina em texto baseado em linguagens de programação, na forma de um bloco interno a outro VI.

O painel frontal inclui controles e indicadores com aparência familiar, o diagrama em

blocos inclui conexões, ícones do painel frontal, funções, possivelmente subVIs e outros objetos do LabView também familiares e é possivel simplificar o diagrama em blocos de um VI pela conversão de seções do diagrama em subVIs.

Outro benefício do LabVIEW sobre outros ambientes de desenvolvimento é o extensivo suporte para acessar hardwares de instrumentação. Drivers e layers de abstração para diversos titpos de instrumentos e vias são incluídas ou estão disponíveis para inclusão. Estes apresentam-se com conectores gráficos. Os layers de abstração oferecem interfaces padrões de software para comunicação com dispositivos de hardware.

Em termos de performance, LabVIEW inclui um compilador que produz código nativo para a CPU. O código gráfico é traduzido para linguagem de máquina pela interpretação da sintaxe e pela compilação. A sintaxe LabVIEW é estritamente executada durante o processo de edição e compilação para a linguagem de máquina quando é requisitada a execução ou durante o salvamento. Neste caso, o executável e o código fonte são mesclados em um único arquivo. O executável funciona com a ajuda do script de tempo de execução do LabVIEW que contem alguns códigos pré-compilados para desenvolver tarefas comuns que são definidas pela Linguagem 'G'. O script em tempo de execução reduz o tempo de compilação e também provê interface consistente para vários sistemas operacionais, sistemas gráficos, hardware, componentes, etc. O ambiente em tempo de execução torna o código portável através de plataformas.

Outro ponto a ser destacado e que auxilia a viabilidade do projeto é a escalabilidade das estruturas e a capacidade de encapsulamento, tanto do LaTeX por meio dos seus pacotes, como no LabVIEW, por meio da sua estrutura de orientado a objeto do código LabVIEW, que permite reutilização de código sem modificações : tão logo os tipos de dados e saídas sejam consistentes, dois subVIs são intercambiáveis.

Outro ponto em comum em ambas as estruturas é a Portabilidade tecnológica. LaTeXe LabVIEW são portáveis para quase todos os sistemas operacionais, guardam entre si ainda uma peculiariedade, do mesmo modo que as versões atuais de LabVIEW possuem parte de seu código desenvolvido em sua própria linguagem LabVIEW, o TeXtambém possui componente de seu código produzido diretamente em TeX. Isso ocorre pois o código fonte do programa atual TeX é escrito em 'WEB', uma mistura de documentação escrita em

TeX com subinstruções em Pascal o que garante a portabilidade.

Como no Latex, novamente a estrutura é suportada por várias bibliotecas com um enorme número de funções para aquisição de dados, geração de sinais, matemática, estatística, condicionamento de sinais, analise, etc, elementos são disponibilizados em pacotes LabVIEW junto com numerosas interfaces gráficas. O número de blocos matemáticos avançados para funções como integração, filtros e outras capacidades especializadas normalmente associadas com captura de dados de sensores em hardware é imensa. OpenG, conhecido como LAVA Code Repository (LAVAcr), oferece repositório para uma larga faixa de aplicações e bibliotecas LabVIEW.

Em adição, LabVIEW inclui um programa componente baseado em texto chamado MathScript com funcionalidades adicionais para processamento de sinais, analise e matemática. MathScript pode ser integrado com programação gráfica usando nós de script e usa sintaxe geralmente compaivel com MATLAB.

O Sistema de Desenvolvimento Profissional do LabVIEW permite a criação de executáveis 'stand-alone' executáveis e bibliotecas compartilhadas, como DLL, pois LabVIEW é um compilador real de 32-bits e o programa executável resultante pode ser distribuído um ilimitado número de vezes.

O script em tempo de execução e suas bibliotecas podem ser providenciados livremente junto com o executável.

Existe uma edição do LabVIEW para Estudantes com baixo custo destinadas a estabelecimentos de ensino para fins de aprendizagem. Existe também uma ativa comunidade de usuários LabVIEW que comunicam através de vários grupos de e-mail e Internet fórums.

3 *Materiais e Métodos*

Nesta seção apresentamos a metodologia de execução do projeto e questões quanto à infra-estrutura, os materiais e os meios necessários para sua implementação e as abordagens para assimilação das tecnologias. Na sequência estabelecemos o plano de trabalho para o projeto, e apresentamos os componentes desenvolvidos, o cronograma definitivo e os resultados obtidos frente aos testes efetuados.

3.1 Metodologia

O primeiro passo foi o ensaio de documentos ativos que servissem de modelos para aplicação dos teste de conceito. Para cada documento, foram ensaiadas expressões que precisariam ser mapeadas e processadas, acompanhadas das especificações dos valores ou conjuntos de valores de seus parâmetros e variáveis.

As expressões ativas (calculadas numericamente ou processadas graficamente) devem ser identificadas no corpo do documento, bem como as especificações de seus valores, através de marcadores (TAGS) que possam ser interpretados pelo processador LaTeX e sejam identificáveis pelo LabVIEW.

Inicialmente o texto escrito em LaTeX contendo esses itens marcados é lido pelo LabVIEW, que se vale dos TAGS como guias para filtrar as expressões e valores de interesse, gerando para cada expressão uma referência de posição no texto (Anexo 05 A).

A seguir, um parser, implementado dentro do LabVIEW como sub-vi do mesmo, interpreta as expressões filtradas e prepara uma tabela de execução (Anexo 06 A) . Esta tabela contém em cada linha os resultados fornecidos pelo parser para cada expressão identificada no documento, acompanhada de um vetor de valores a serem aplicados à mesma. Em seguida, essa tabela é lida iterativamente e as entradas são calculadas uma a uma, montando-se uma tabela de saída que contém os resultados postos em correspondência com as referências de posição de cada expressão no texto (Anexos 07 A).

Finalmente, a tabela de resultados é desmembrada e casada com o texto, gerando-se um novo documento LaTeX contendo os resultados (Anexo 17 A). O novo documento é então processado pelo LaTeX, gerando-se a sua renderização gráfica (Anexo 19 A).

Seguindo este roteiro foram criados de forma modular sub-instrumentos virtuais em LabVIEW relacionados às operações que compõem essa tabela de instruções e associados a cada um desses blocos lógicos.

Foi então desenvolvida a aplicação principal em instrumento virtual (.vi) (Anexo 01.d A) que, dado o arquivo/roteiro de entrada em padrão látex,ou via entrada por linha de comando, e com o diagrama em blocos correspondente (Anexo 01 A) para o processamento

do documento, torna automático o roteiro proposto de acordo com os dados de entradas inseridos.

Esse instrumento virtual possui um Painel de Controle (Anexo 01.c A), chamado de interface do usuário onde o usuário carrega o documento ao mesmo tempo em que processa os cálculos numéricos e gera os gráficos associados, gerando ao fim do processo um relatório completo do documento.

3.1.1 Metodologia das Tags e Códigos de Operação

Na integração dessas tecnologias, parte considerável do trabalho será na definição dos comandos e macros da biblioteca LabTeX, esses comandos devem ser compreendidos pelos instrumentos virtuais (subVIs) sob o aspecto de instruções lógicas (como criação e atribuição de variáveis, cálculos aritméticos, entre outras, como pré-classificado na Planilha de Mapeamento de Métodos-SubVIs a seguir, mas devem ser ignorados pela geração de layout na compilação do documento .tex correspondente.

A estratégia para isso é a utilização de redefinição de comandos e macros 2.1.8,e a definição de novos, que se utilizem das definições de layout dos comandos atuais mas que possam extendê-los de modo a serem reconhecidos como ativos pelo LabVIEW.

3.1.2 Cronograma

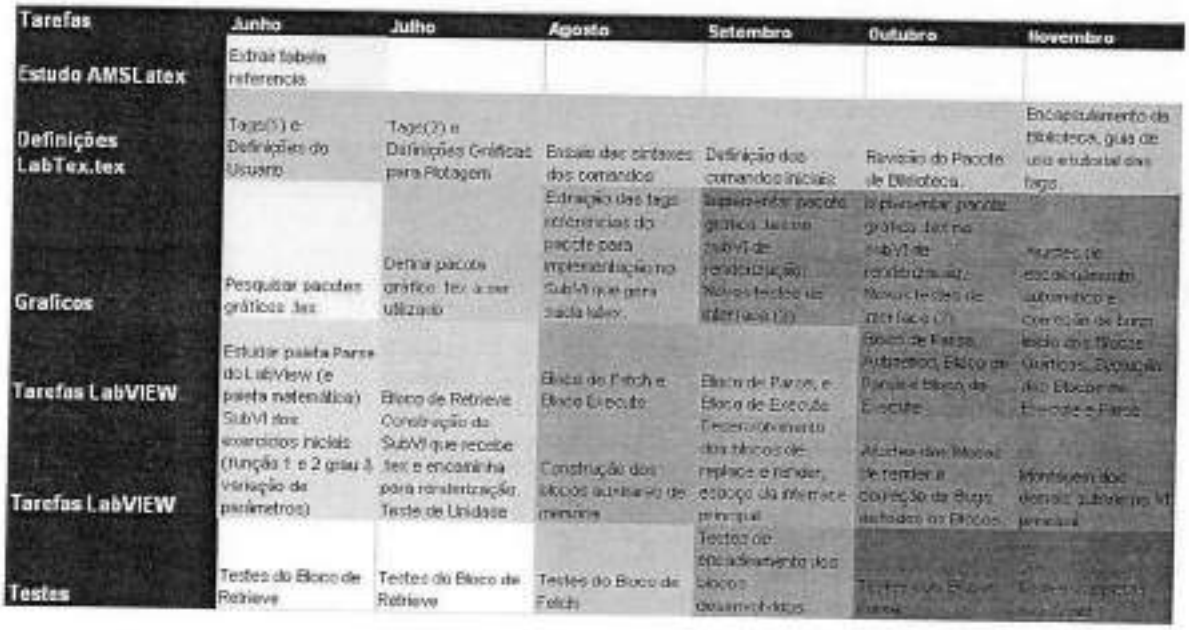


Figura 6: Cronograma Efetuado

3.2 Desenvolvimento: Objetos e Componentes

Classes Labview suportam a orientação a objeto, o que será demasiado útil na implementação componentizada deste projeto com subVIs especificos para cada uma das etapas do processador LabTeX.Esses subVIs-métodos compartilharão o acesso ao mesmo objeto definido pela classe documento ativo ('activeDocument') que define as propriedades necessárias ao objeto 'documento ativo' para a identificação das instruções, tipos de operadores, parâmetros, variáveis, constantes , posições e todo vetor de informações que precisar ser compartilhado e manipulado pelos subVIs do projeto, que neste escopo da orientação a objeto, assumem o papel de métodos e apenas por meio deles será possível manipular os dados da classe privada.

O objeto documentoativo (Anexo 21 A) tem sua classe definida pelas propriedades que definem o objeto. Essas propriedades são compartilhadas por cada bloco componente do processamento desde a etapa de captura das tags (Anexo 05 A) onde são utilizadas as propriedades startIndexTag, endIndexTag (para as coordenadas das tags) e activeTag para o próprio teor da instrução, passando pela etapa de fetch (Anexo 06 A), na qual essas Tags são processadas e realimentam os parâmetros argId (contador), argC (nome do argumento), argV (valor do argumento), argT (tipo da tag) e argDim (quantidade de tags ativas presentes no documento).

Na sequencia o bloco de execução (Anexo 07 A) compartilha do objeto processando os argumentos e colocando os resultados das expressões em resultTag (array de resultados) e por fim o bloco de substituição (Anexo 17 A), acessa novamente a classe para substituir nas posições de startIndexTag e endIndexTag o conteúdo do vetor resultTag realizado na etapa anterior.

O resultado geral do documento é salvo na propriedade resultLatex que é acessado pelo bloco de renderização (Anexo 19 A) e de exibição do texto na aplicação principal (Anexo 18 A).

Todos os blocos desenvolvidos no projeto são apresentados a seguir, sua listagem completa está disponibilizada no Anexo 20 A bem como a listagem de suas dependencias, blocos nativos disponibilizados nas bibliotecas básicas do LabVIEW (Anexo 20.b A

3.2.1 Aplicação Principal

A partir desta subseção e nas demais, passamos a descrever a engenharia do sistema por meio da explicação do fluxo dos blocos e de suas funcionalidades e modos de operação, que corresponde ao desenvolvimento componentizado proposto.

O instrumento virtual Main.vi (Anexo 01 A) corresponde à aplicação principal, na parte superior a área para digitação dos comandos e os botões para seleção de arquivo e página. Abaixo o resultado renderizado e mais abaixo o resultado.tex completo em cinza (Anexo 01.d A).

No diagrama em blocos da aplicação principal (Anexo 01 A) estão envolvidos os diversos SubVIs correlacionados que são descritos de acordo com sua aparição no contexto dessa descrição funcional. Na seção de ?? estão os diagramas dos blocos componentes, bem como Painel Frontal da Aplicação e de alguns subcomponentes, guiando essa descrição funcional.

O primeiro laço (à esquerda) da aplicação é responsável pelo controle da interface e envio de comandos, seleção de entrada ('Arquivo' ou 'Linha de comando'), e troca de páginas. O programa executa enquanto o usuário não apertar 'Stop' e, ao clicar em Carregar, o usuário dispara o processo de execução do documento ativo.

Para trocar de página o usuario deve selecionar seu número e clicar no botão carregar correspondente (Anexo 01.c A). O bloco subsequente processa as alternativas e, entre eles, está o subVI 'getInput' (Anexo 02 A), que recebe os caminhos dos arquivos de cabeçalhos e includes .tex do sistema e o caminho do arquivo de entrada selecionado pelo usuário, bem como o texto digitado na linha de comando, e os encaminha para escolha de entrada de acordo com a opção escolhida no botão [T:File / F: CommandLine], ou seja, posição Verdadeiro para entrada em arquivo, e posição Falso para entrada por texto em linha de comando.

Quando o responsável por sair do laço de controle for o botão 'Página', a subrotina do Anexo 01.b (A) é executada, nela o subVI 'ChangePag'(Anexo 03 A) carrega o resultado previamente processado em outra página. O conhecimento sobre o número de páginas ocorre no momento do processamento do LaTeX na etapa de renderização (Anexo 19 A) quando o aplicativo auxiliar Ghostscript 3.3 é chamado pelo sistema.

Quando o botão 'Página' é selecionado com um número de página igual ao atual, nada acontece. É possivel também iniciar novo processamento e iniciar a visualização pela página na qual se está ou na qual se deseja. Quando LTChangePag.vi é acionado, recebe o valor numérico da página selecionada e efetua a troca da imagem pelo arquivo que fora gerado durante a compilação. O formato adotado é o JPEG e zoom de 50%.

Caso o ato de sair do laço de controle da aplicação principal tenha sido ocasionado pelo botão 'Carregar', este levará o sinal Verdadeiro para o macro-bloco seguinte que é a sequencia básica de processamento propriamente dita.

3.2.2 Busca (retrieve) e Seleção (fetch)

Ao entrar no laço principal de processamento o primeiro passo é o subVI 'LTreadInput'(Anexo 04 A) que recebe o caminho do diretorio de entrada e salva o documento ativo a ser processado na pasta da trabalho, de modo que todo ensaio no sistema gera código fonte de documentação .tex de entrada e não apenas de saída. A pasta de trabalho foi setada para o subdiretorio 'render' do disco principal e os arquivos de entrada salvos sob o nome de original.tex.

A seguir é evocado o bloco LTretrieveTag.vi (Anexo 05 A), responsável pela detecção das Tags ativas do documento, captura as tags iniciadas com \LT e suas respectivas coordenadas de posicionamento inicial e final dentro do documento para posterior substituição dos resultados.

Essas tags e posições constituem vetores que são armazenados no objeto e passados adiante para os demais blocos como propriedades do documento ativo. 3.2

A interface deste bloco (Anexo 05.b A) é apresntada com respectivos vetores de instruções e suas respectivas coordenadas de início e fim dentro do arquivo original.tex

Após serem detectadas no bloco de retrieve o documento ativo é encaminhado para a etapa de fetch/seleção (Anexo 06 A), este é o bloco responsável por fazer a classificação das Tags, recebe como parâmetros o vetor de expressões gerados em LTretrieveTag.vi e separa-as em argumento (argC), valor do argumento (argV) e tipo (argT), criando outro triplete de vetores de expressões, esses já pré-processados e com a marcação a que Tag/função direciona.

Neste mesmo bloco, nota-se a esquerda o array de comandos disponíveis do momento da documentação do projeto, a cada novo comando implementado basta adicionar um elemento para detecção nessa lista e adicionar o caso correspondente no switch, que faz a leitura dos parâmetros internos aos simbolos de '{' e '}'.

É apresentada a interface do bloco de seleção das tags (Anexo 06 A), com respectivos vetores de argumentos, valores e tipos das instruções para processamento pelo bloco de execução. No documento em teste foram encontradas 81 Tags (argDim), em destaque as Tags de 57 a 76.

3.2.3 Preparo e Execução das Tags

Depois dos blocos de busca e seleção, os vetores de expressões ativas são encaminhados para o processamento em LTexecuteTag.vi (Anexo 07 ate Anexo 07.1 A), esse bloco é um case/select onde de acordo com o tipo do comando uma subrotina é efetuada.

No caso do comando LTsetvar, responsável por fazer a atribuição de variáveis, os valores dos argumentos do comando são lidos e primeiramente processados pelo bloco de Parse que efetua a tradução do LaTeX para linguagem aritmética (Anexo 08 A), na sequencia a definição da variável é calculada no bloco ParseArithm (Anexo 10 A) que recebe as variáveis atuais na memória como entrada (vetores HeaderDefs, HeaderVars e HeaderValues) para processar o valor da variável em atribuição, por fim o valor da variável atribuída é também guardada na memória por meio do bloco 'PushHeader' (Anexo 11 A). O bloco pullHeader (Anexo 12 A) nesse caso é a prova real, devolvendo ao usuário a confirmação dos valores registrados na memória do sistema.

O resultado das Tags processadas é armazenado no vetor de propriedade 'resultTags' da classe activeDocument. Cada Tag portanto que ja houvera sido adicionada no vetor de propriedades do objeto na etapa de retrieve (Anexo 05 A) ao lado de suas coordenadas de posicionamento, agora recebem a companhia do respectivo resultado que será lido na etapa de substituição de dados pelo bloco LTreplaceData (Anexo 17 A).

Os Anexos 07.b (A) em diante descrevem as outras operações implementadas no bloco LTexecuteTag. Na seção de resultados 3.4 são apresentadas as funções disponibilizadas, modo de uso e resultados obtidos.

O Comando LTsetfx (Anexo 07.b A) é similar ao LTsetvar com a diferença que por se tratar de uma função com variável independente, não é feito um primeiro cálculo do valor inicial da função, como é feito para o caso de variáveis.

O Comando LTgetvar (Anexo 07.c A) retorna o valor literal de definição da variável, usando para isso o bloco auxiliar de pullHeader (Anexo 12 A)

O Comando LTeval (Anexo 07.d A) efetua o cômputo da variável selecionada recalculando-a a partir de sua definição no vetor de memória HeaderVars, para os valores existentes dos parâmetros em HeaderValues, como veremos na seção de resultados 3.4, isso faz com que

no caso de uma variável recursiva, essa função retorne um novo valor após cada novo processamento.

O Comando LTevalfx (Anexo 07.e A) efetua o cálculo da Função no ponto solicitado a partir de suas definições nos vetores de cabeçalho.

Nessa função há o parâmetro 'exceto' para troca de variável independente, permitindo assim que uma função a vários parâmetros possa ser analizada por meio de suas variações em separado.

Assim, se F(x,y,z) = w, é possivel ensaiar F(x) dados y e z, ou F(y) dados x e z, ou F(z) dados y e z, e assim sucessivamente. Por tratar-se de uma operação de uma dimensão, apenas uma variável pode ser independente nesse contexto.

No procedimento essa operação se utiliza dos blocos de LTreplaceVars (Anexo 15 A) , LTparseLatex (Anexo 08 A), LTparseArithm (Anexo 10 A) e o bloco auxiliar LTpullHeader (Anexo 12 A).

O Comando LTprevalfx (Anexo 07.f A) efetua o précálculo da Função no ponto solicitado a partir de suas definições nos vetores de cabeçalho, retornando como resultado o valor das variáveis nas posições adequadas dos parâmetros da função. É uma operação de apoio a documentação e que dá suporte a LTevalfx, essa função também utiliza o parâmetro 'exceto' para troca de variável independente. Essa operação também faz uso dos blocos de 'replaceVars' e 'pullHeader'.

O Comando LTgetfx (Anexo 07.g A) retorna o valor de definição literal da função solicitada, utilizando-se para isso também do bloco auxiliar LTpullHeader alimentado pelos vetores de cabeçalho, que são a memória do sistema.

O Comando LToperfx (Anexo 07.h A) efetua a composição de duas ou mais funções existentes, atribuindo sua composição a uma função resultante. No procedimento essa operação se utiliza dos blocos de LTreplaceFunction (Anexo 15.b A) e do bloco auxiliar LTpullHeader (Anexo 12 A).

O Comando LTpreval (Anexo 07.i A) efetua o précálculo da variável a partir de suas definições nos vetores de cabeçalho, retornando como resultado o valor das literais nas posições adequadas dos parâmetros da variável. Trata-se também de operação de apoio a

documentação que dá suporte a LTeval. No procedimento essa operação também se utiliza dos blocos de LTreplaceVars e LTpullHeader.

O Comando LTgetval (Anexo 07.j A) busca o valor na memória para a variável selecionada. É similar à LTeval com a diferença de não efetuar novo cálculo durante o processo. Essa operação se restringe a buscar o valor atual da variável ao operar o bloco auxiliar LTpullHeader.

O comando LTplot (Anexo 07.k A) é responsável por gerar o gráfico de uma função diretamente, sem que essa exista necessariamente no escopo da memória do sistema. Os parâmetros do comando são selecionados com a ajuda do bloco auxiliar LTgetParams (Anexo 09 A) e depois é tratada inicialmente em LTreplaceVars (Anexo 15 A) onde tem seus literais substituídos exceto o argumento informado como variável independente neste caso.

A expressão é então traduzida do LaTex para modo Aritmético em LTparseLatex (Anexo 08 A) e tem seus demais parâmetros de plotagem definidos em LTprepare Plot (Anexo 16 A), ao final a string de renderização gráfica é armazenada no vetor de resultados para ser processada na etapa de renderização.

Existe a possibilidade de trocar a variável independente no momento da plotagem, assim se definimos uma variável x como sendo a composição de outras duas, por exemplo: $x = \frac{a}{b}$ com valores estabelecidos $a = 1$ e $b = 2$, por exemplo, então podem ser levantadas as curvas tanto de $x(a)$ (ou seja $f(a) = \frac{a}{2}$) ou $x(b)$ (ou seja $f(b) = \frac{1}{b}$).

O comando LTplotfx é similar a LTplot e também responsável por gerar o gráfico de uma função, com a diferença que esta função precisa ter sido definida antes pela utilização dos comandos LTsetfx (Anexo 07.b A) ou LToperfx (Anexo 07.h A).

É um atalho de utilização que permite que o comando seja chamado em sua sintaxe sem a necessidade de passar novamente a definição da função, bastando referenciá-la. O restante do funcionamento é o mesmo que o de LTplot (Anexo 07.k A) e da mesma forma que em LTplot, existe a possibilidade de trocar a variável independente no momento da plotagem, mais detalhes de utilização estão especificados no documento de resultados 3.4 aos testes efetuados.

3.2.4 Preparo e Execução das Expressões Aritméticas

O bloco responsável pela tradução da expressão Latex para linguagem Aritmética é o LTparseLatex.vi (Anexo 08 A) e é utilizado como componente em diversos comandos do sistema. É de natureza expansível, cada novo comando a ser traduzido deve ser adicionado nos vetores de inicialização à esquerda, onde constam respectivamente a sintaxe no LaTeX, a sintaxe correspondente matemática, e o número de argumentos que se utiliza.

O bloco varre a tabela de expressões e para cada uma, opera as substituições na expressão até que todas tenham sido traduzidas. Os casos de '1' a '5' tratam especificamente cada operador, sendo mínima a variação de algoritmo mas necessária face as sintaxes de cada operador em particular.

Os argumentos das operações são retirados pelo subVI 'getParams' (Anexo 09 A) que detecta os macro argumentos contidos entre { e } mesmo em casos de outros argumentos compostos e internos.

Esse bloco é utilizado em todos os tratamentos de parâmetros e foi componentizado a partir de sua estrutura existente inicialmente no bloco de LTfechtTag (Anexo 06 A). Nesse anexo em destaque o símbolo detectado é o delimitador { (opção/case '0') enquanto no (Anexo 09.b A) é detectado o delimitador } (opção/case '1').

Quando esses símbolos são detectados é efetuada uma contagem da ordem de recursão do argumento para que seja detectado o argumento que contem a maior ordem.

Após a tradução do LaTeX, nas operações de cálculo a expressão é direcionada ao bloco LTparseArithm.vi (Anexo 10 A) que é o bloco de Processamento Aritmético do sistema, utilizado por vários comandos, possui uma lista de operadores compreensíveis e sua respectiva precedência.

Cada operador ou número ou literal da expressão é avaliado, a precedencia do novo operador é comparada com a precedencia do operador no topo da pilha e se o novo operador for de menor precedencia então o bloco LTProcessOperators (Anexo 13 A) efetua o cálculo e substitui os operandos na pilha pelo seu resultado.

Após isso o novo operando ou operador é colocado na pilha de acordo com seu tipo: em

caso de numeral, é interpretado e adicionado à pilha de valores; em caso de literal (Case '1'), é verificado se trata-se de variável (SubCase '-1') (Anexo 10.b A) e então o valor da variável é substituido pelo valor presente em memória nos vetores de HeaderVars e Header Values por meio do bloco de LTPullHeader (Anexo 12 A), em caso de função (SubCase '0,...') então a mesma é adicionada ao topo da pilha de operadores (Anexo 10.c A)

Por fim, caso se trate de operador simbólico e não literal (Case '2,..) (Anexo 10.d A) então o operador é simplesmente colocado no topo da pilha de operadores.

3.2.5 Blocos de Apoio

Nesta seção descrevo os blocos de apoio que são utilizados pelos demais blocos funcionais do sistema e que aparecem diversas vezes no código e nas descrições associadas aos outros blocos.

O bloco LTpushHeader (Anexo 11 A) serve de apoio ao transporte de variáveis, valores e definições. Atua como pilha de memória do sistema em tempo de execução em par com o bloco correspondente PullHeader (Anexo 12 A).

Em cada operação de cálculo ou atribuição de váriáveis ou funções o bloco PushHeader é responsável pela gravação dos dados nos vetores de memória (HeaderDefs, HeaderVars e HeaderValues) verificando se a função ou variável já não existe no escopo das pilhas para atualização, caso contrario adiciona como novo elemento.

O bloco LTpullHeader por sua vez é responsável pela busca dos dados nos vetores de memória (HeaderDefs, HeaderVars e HeaderValues) em cada operação de busca de definições ou valores de váriáveis ou funções.

A chave binária 'Function' seleciona o modo de operação (variável ou função). Se for função efetua apenas a busca pelos literais da definição, se for variável, busca os valores dos literais. Essa particularidade permite seu uso na operação LToperfx (Anexo 07.h A)

O bloco LTProcessOperators.vi (Anexo 13 A) efetua a operação no topo da pilha de operadores entre os dois valores no topo da pilha numérica. Se o operador topo é o sinal de fecha parentesesis, o processamento continua processando os operadores até encontrar o sinal de abre parenteses correspondente. A execução final é feita no bloco LTexecArithm (Anexo 14 A). Chamado unicamente pelo LTProcessOperators.vi, esse bloco é o responsável final pelo cálculo de expressões, ele verifica o operador de entrada e substitui os dois valores do topo da pilha com o resultado. Esse subVI, inicialmente disponibilizado na paleta do LabVIEW, foi modificado (Anexo 14.b A) com a inclusão de outros operadores: foram incluidas as operações trigonometricas seno, cosseno e tangente e a operação de exponenciação.

Ha a possibilidade de inclusão de inúmeras inclusive as que permitam a operacionalização de nûmeros complexos, é bom ressaltar que nesse projeto foram implementadas apenas

essas poucas pela necessidade de desenvolver as macro-estruturas que tornassem o projeto funcional, e que por tratar-se portanto de um teste de conceito, a prioridade nesse momento não foi focar no enriquecimento da paleta de operações o que é sem sombra de dúvida um ponto a evoluir para a implementação em sala de aula como ferramenta de apoio.

O subVI LTreplaceVars (Anexo 15 A) é o bloco responsável pela substituição de variáveis pelos seus respectivos valores em memória. Possui uma entrada 'exceto' para informação da variável independente, e nesses casos, computa a substituição dos outros parâmetros das variáveis e funções que não os contidos nessa entrada. É utilizado em praticamente todas as funções como nas de preval , eval, get e plot.

Como no descrito nos blocos de parse, este bloco compara a entrada com os literais, operadores e váriáveis na memória, preservando a variável independente (se houver um parâmetro 'x' nessa hora é substituido por seu valor nesta etapa e a váriável colocada em 'exceto' é transformada em 'x'.)

O subVI LTreplaceFunction (Anexo 15.b A) é similar ao LTreplaceVars, sendo responsável pela substituição dos literais, mas no caso apenas dos literais das definições das funções, sem a substituição dos literais das variáveis por seus valores, e portanto não possui entrada 'exceto'. É utilizado na função de operação entre funções (LToperfx) (Anexo 07.h A)

LTpreparePlot (Anexo 16 A) é o subVI responsável pela préformatação do gráfico a ser plotado. Este bloco chamado pelas funções gráficas (LTplot, LTplotfx) efetua o ajuste de escala de eixos 'x' e 'f(x)' de acordo com o tamanho desejado para a imagem e com o domínio selecionado, e obtem amostras da função no intervalo para determinar valores máximos e mínimos e calcular a escala mais apropriada para o eixo de f(x).

Configurado inicialmente para efetuar amostragens com passos de 0.01, pode ser aumentada sua precisão no entanto podendo comprometer o desempenho.

As funções de plotagem trabalham em conjunto com o componente livre Gnuplot 3.3 que gera em tempo de compilação do latex, os pontos para plotagem, de acordo com os parâmetros selecionados nesse bloco. O Gnuplot trabalha em conjunto com o pacote gráfico PGF/TikZ do LaTeX.

No subVI LTxscaleGraph (Anexo 16.b A) é selecionada a escala mais adequada de grid. Mais detalhes sobre a utilização das funções gráficas são demonstrados no roteiro de testes. 3.4

O bloco LTxscaleGraph presta-se ao escalonamento de grid nos eixos utilizado pelo LTprepareGraph. O comprimento desejado para o gráfico em questão é comparado com o domínio desejado para a função para estabelecimento da melhor proporção para o grid, na imagem do diagrama em bloco temos um case/select de uma única possibilidade entre as diversas escalas pre-estabelecidas. Quando uma escala é verdadeira, as demais são falsas, fazendo com que os blocos em 'False' funcionem como um by-pass.

3.2.6 Geração dos resultados

Nas etapas finais temos os blocos de substituição , leitura de saída e renderização. O bloco LTreplaceData.vi (Anexo 17 A) é responsável pela substituição das Tags ativas por seus resultados processados em LTexecuteTags (Anexo 07 A), usando para isso as coordenadas obtidas no bloco inicial de LTretrieveTags (Anexo 05 A).

O resultado é salvo em arquivo (configurado no bloco Principal para chamar-se resultado.tex' e encaminhado para os blocos de readOutput (Anexo 18 A) e LTrenderLatex (Anexo 19 A)

Na etapa de conclusão do processamento principal (Anexo 01 A), o bloco LTreadOutput acessa o atributo resultLaTex, do documento ativo e encaminha para a saída texto da interface (Anexo 01.d A).

O Bloco LTrenderLatex.vi é responsável por gerar saida gráfica. Primeiro efetua a compilação do arquivo resultado.tex, por meio de uma chamada ao sistema do compilador latex, gerando resultado.pdf que é armazenado no diretório de trabalho.

Em seguida por meio do utilitário GNU-GPL Ghostscript 3.3, este pdf é convertido para jpg e encaminhado para a interface principal, na segunda chamada ao sistema. As chamadas ao sistema fazem com que esses componentes de renderização sejam requisitos de funcionamento. Em caso de substituição por outro método de saída visual ou renderização, basta alterar as configurações deste bloco.

Esse componente externo, assim como o Gnuplot utilizado em LTprepare Plot (Anexo 16 A) são livres e de fácil obtenção na internet e portáveis para quaisquer plataformas, características dos aplicativos baseados na licença GNU-GPL. Quaisquer troca de tecnologias associadas nesse contexto implica apenas na reconfiguração desses blocos.

3.3 Recursos Auxiliares

Para elaboração do projeto foram incluidas duas ferramentas auxiliares para obtenção dos gráficos e para conversao dos arquivos finais. O LaTeXcompila seu código para pdf mas o LabVIEW não oferece suporte nativo a esse tipo de arquivo, então utilizou-se do

aplicativo livre Gnu-GPL GhostScript [31] para conversão do pdf em formato jpg, este por sua vez aceito pelo LabVIEW.

Ghostscript é um aplicativo em linha de comando que provê um interpretador para pdf com a habilidade de converte-lo para diversas saídas gráficas, é escrito completamente em C e foi desenvolvido com a preocupação de que fosse executado de forma apropriada em uma variedade de sistemas incluindo Windows, Unix, Apple MacOS, etc. É um aplicativo protegido por licença GNU 'General Public License' o que permite seu livre uso, cópia e redistribuição e existe versão comercial para o caso de necessidade de licença comercial

No tocante aos gráficos, por meio da integração com o pacote gráfico de LaTeX, o TikZ/PGF [25] , foi implementada a conexão com o aplicativo Gnuplot [32] para suporte ao desenvolvimento de gráficos, devido suas caracteristicas portáveis e simples e nativamente integradas aos comandos do TikZ.

Gnuplot é um aplicativo também distribuido sob livre licença e é executado em linha de comando de diversos sistemas operacionais como os citados anteriormente para o Ghostscript, foi originalmente criado para permitir que cientistas e estudantes visualizassem funções matemáticas e dados interativamente, porém tem crescido seu uso no suporte a diversos usos como os scripts web, também é utilizado como componente de plotagem em aplicações de terceiros. Suporta diversos formatos de arquivos e é extensível para a inclusão de outros tipos de saídas. Suporta vários tipos de gráficos entre 2D e 3D, e permite integração com o LaTeXque foi o que motivou seu uso neste projeto.

3.4 Testes e Resultados

A seguir apresento um roteiro de ensaio das operações implementadas. Este roteiro é composto por um arquivo original.tex que é o nosso guia interativo de uso dos comandos, ou seja, é feito o convite para que ele seja executado com outros valores e definições de modo que ele possa servir como base de composição para documentos ativos. Num primeiro bloco, é adicionado abaixo o código de original.tex 3.4.1 e o seu respectivo efeito renderizado 3.4.2. Na sequência são apresentados os arquivos resultado.tex e a sua respectiva renderização obtida em tempo de execução, no LabVIEW por meio de imagens jpeg mas o sistema também gera o pdf associado e este é anexado na sequencia desta seção a este trabalho.

Uma observação importante da natureza do projeto e da realização dos testes: se alguma operação ocasionar NaN,ou Inf isso significa que o argumento de alguma raiz é negativo ou que ocorreu erro de divisão por zero, por exemplo. No teste de conceito deste projeto não foi meu foco o tratamento de situações de excessão, me concentrei em levantar possibilidades de utilização como ferramenta de apoio à documentação científica, é obvio no entanto que existem situações especiais que merecem tratamento e que o uso do sistema e exploração das estruturas permitirá avaliar quais os erros que devem ser tratados preponderantemente.

Outra observação a ressaltar é a natureza interativa do processo de escolha do tamanho e dominio adequados à ilustração da função. O algoritmo atual , como é explicado na seção sobre o modo de funcionamento dos subVIs componentes , trabalha com deteção de máximo e mínimo da função para composição do eixo y, usando para isso um passo de amostragem inicialmente configurado para 0.01, o que durante o funcionamento no computador pessoal do aluno, não causou demora na execução mas restringe o 'zoom-in' de plotagem a uma escala de 100. Foram efetuados testes com passos de 0.001 causando sensivel lentidão durante a execução, em computadores com melhor desempenho,o aumento da resolução para esse valor de amostragem implicará na possibilidade de um zoom de até x1000.

3.4.1 original.tex : Código Tex

```
%& --shell-escape dubble
```

```
\documentclass[10pt,letterpaper]{article}
\usepackage[brazil]{babel}
\usepackage[latin1]{inputenc} %codificacao para latin.
\usepackage{a4wide}
\usepackage{hyperref}
\usepackage{amssymb,amsmath}
\usepackage{datetime}
\usepackage{tikz}
\everymath{\displaystyle}
\title{LabTeX - Modo Linha de Comando}
%\author{Fernando J. Capeletto}
\begin{document}
%\maketitle
\thispagestyle{empty}
\newcommand{\matpar}[2]{\raisebox{1pt}{\small
\ensuremath{#1}}$/$\raisebox{-2pt}{\small \ensuremath{#2}}}
% Comandos LabTeX
\newcommand{\LTsetvar}[2]{ $ #1\Leftarrow\left\langle#2\right\rangle $}
\newcommand{\LTpreval}[1]{$\left\langle #1 \right\rangle $}
\newcommand{\LTeval}[1]{$\left\langle #1 \right\rangle $}
\newcommand{\LTgetval}[2]{$#1 = \left(#2\right)$}
\newcommand{\LTgetvar}[2]{$#1 \equiv (#2)$}
\newcommand{\LTplot}[2] { Plotar \left\langle #1 \right\rangle
$ com os dados: #2$ \overbrace{#2}^{<N>-<W,H><xi:xf><yi:yf>}}
\newcommand{\LTsetfx}[2] { $ #1\left(x\right) \equiv #2 $}
\newcommand{\LTevalfx}[4] { $ #1\left(#2 \rightarrow #3\right) = #4 $}
\newcommand{\LTprevalfx}[4] { $ #1\left(#2 \rightarrow #3\right) = #4 $}
\newcommand{\LTgetfx}[3] { $ #1\left(#2 \right)\equiv #3 $}
\newcommand{\LTplotfx}[2] { $Plotar #1 \left(#2 \right) com os dados: $}
\newcommand{\LToperfx}[3] { $ #1\left(x\right) \equiv #2 \equiv #3$}

\begin{math}
```

```
 $Explicando alguns comandos:  \\
\begin{itemize}
\item \verb \LTsetvar{<var>}{<valor-ou-expressao>};  :
 Ex: (\LTsetvar{a}{5*\tan{\frac{3,1415}{4}}}\;) acabamos de fazer uma
 atribuição à variável 'a'. (\LTsetvar{b}{\sqrt{a+119}}\; ,
   \LTsetvar{d}{\sqrt{\frac{a}{b}}}\;
    e  \LTsetvar{ab}{-1*\sin{\frac{\cos{a}+b}{\tan{100}}}}\; são outros exemplos).
  \emph{As variáveis são definidas em letras minusculas podendo ser seguidas por u
   ou mais numeros e outras letras, também minusculas.}
\item  \verb \LTgetvar{<var>};  Retorna a definição da variável, aplicada à 'a' re
 \LTgetvar{a}\; em b resulta \LTgetvar{b}\; e em ab resulta \LTgetvar{ab}\;
\item \verb \LTpreval{<var>}; : Demostra o pré-calculo da variavel, com os valores
 inseridos no lugar de sua definição. No vaso de ab = \LTpreval{ab}\; \\
\item \verb \LTeval{<var>};  : É um comando para cálculo da variavel, temos:
 a = \LTeval{a}\; e aplicada à 'b' resulta b = \LTeval{b}\;
  e no caso de ab : \LTeval{ab}\;
\item \verb \LTgetval{<var>}; : Outra alternativa ao get e ao getvar,
 só que retorna apenas o valor da variável. No exemplo \LTgetval{b}\;
   \verb \LTeval{<var>}; é parecido com getval mas com uma diferença :
    getval retorna o ultimo valor da variável na memória,
  enquanto eval retorna o valor da variavêl após novo cálculo, isso é preponderant
   em variáveis recursivas pois a cada chamada de eval acarretará novo valor enqua
    getval não.
    Vejamos:
\subitem Primeiro definimos \LTsetvar{c}{\sqrt{a+b}}\; e seu valor de partida \LTg
 pois \LTpreval{c}\;
\subitem Depois definimos \LTsetvar{b}{\frac{\sqrt{c}}{c}}\; e seu novo valor \LTg
\subitem Depois fazemos um primeiro eval : \LTeval{c}\; e \LTeval{b}\;
\subitem Depois fazemos um segundo eval : \LTeval{c}\; e \LTeval{b}\;
\subitem Calculemos agora ab: \LTeval{ab}\;
\subitem Enquanto isso getval retorna o valor anterior: \LTgetval{c}\; e \LTgetval
\subitem Se uma variavel depender dela mesma: \LTsetvar{d}{a+d+1}\; com seu primei
```

```
 valor \LTgetval{d}\;
   Se fizermos seguidos eval dessa variável obteremos seguidos valores:
  \LTeval{d}\; e \LTeval{d}\; e \LTeval{d}\; respectivamente.
\subitem Definamos agora \LTsetvar{a}{\sqrt{5}}\; \LTsetvar{x}{\sqrt{a}}\; e porta
x = (\LTeval{x}\;). \\
\item \verb \LTplot{{<x>}{<expressão>}}{{<N>-<W,H>}{<x1>:<xn>#<step>}}; : Plota a
 de x pela expressão dada, para N pontos de amostragem e com dominio entre
  x1 e xn e discretização 'step'.. Conveniente avisar que ela efetua a substituiçã
   literais na memória exceto pelo argumento informado em {x}, como veremos a segu

\emph{Ao fazer uso dos parametros da função plot, pode-se ajustar ,melhor
o tamanho e o eixo  de acordo com a curva para uma melhor documentação como
segue na sequencia de exemplos a seguir, nos quais apresento algumas
possibilidades e limitações de desempenho.}\\

È interessante ressaltar a natureza interativa do processo de escolha do tamanho
e dominio adequados à ilustração da função. O algoritmo atual , como é explicado
na seção sobre o modo de funcionamento dos subVIs componentes, trabalha com deteçã
de máximo e mínimo da função para composição do eixo y, usando para isso um passo
de amostragem inicialmente configurado para 0.01, o que durante o funcionamento no
computador pessoal do aluno, não causou demora na execução mas restringe o 'zoom-i
de plotagem a uma escala de 100. Foram efetuados testes com passos de 0.001 causan
sensivel lentidão durante a execução.

A seguir uma série de funções e seus respectivos zooms em áreas de interesse para
caracterizar o processo interativo de escolha dos parâmetros mais adequados para
representação (domínio, largura, comprimento e discretização da grade e tomada de
 valores).

\subitem Calculando b= (\LTeval{b}\;) , a = (\LTeval{a}\;) e x = (\LTeval{x}\;).
  e para 50 amostras : \\
\subitem \LTplot{{x}{\sin{a*x}+\cos{b*x}}}{{50-10,5}{0:15#1}}\; \\
```

```
\clearpage

\subitem E aumentando o número de amostras para 500 e aumentando a altura da image
\subitem \LTplot{{x}{\sin{a*x}+\cos{b*x}}}{{500-10,10}{0:15#2}}\; \\
\subitem Restrição do domínio : \\
\subitem \LTplot{{x}{\sin{a*x}+\cos{b*x}}}{{50-10,5}{0:5#1}}\; \\
\subitem \LTplot{{x}{\sin{a*x}+\cos{b*x}}}{{50-10,5}{2:4#0,5}}\; \\
\subitem \LTplot{{x}{\sin{a*x}+\cos{b*x}}}{{50-10,5}{2:2.5#0,2}}\; \\
\subitem \LTplot{{x}{\sin{a*x}+\cos{b*x}}}{{50-10,5}{2.2:2.3#0,02}}\; \\

Como observa-se, a escolha adequada do passo de amostragem permite foco na região
 de interesse sobre o domínio, encontra-se anexo a este roteiro de teste uma
 sequencia de tentativas na obtenção da melhor visualização das funções como algun
dos resultados obtidos pelo comando \verb \LTplot; mas ainda há uma sêrie de
particulariedades a serem descobertas de acordo com o uso e exploração das estrutu
\subitem Também interessante observar que se chamamos o mesmo comando plot para a
 mas passando como referência outra variável independente, o comando
 faz a adequação dos parâmetros :
\subitem \LTplot{{a}{\sin{a*x}+\cos{b*x}}}{{50-8,5}{1.5:4#0,5}}\; \\
\subitem \LTplot{{b}{\sin{a*x}+\cos{b*x}}}{{50-8,5}{2.5:5#0,5}}\; \\

\clearpage
\item \verb \LTsetfx{<F>}{<formula>}; : Cria a função F$\left( x \right)
$ atribuindo a ela a formula: \\
\subitem Seja \LTsetfx{F}{a*x+b*x+c}\; \\
\subitem  e \LTsetfx{G}{\sqrt{d}*x^{3} - 6*x +12}\;  \\
\subitem e \LTsetfx{Z}{\sin{10*x} - \cos{50*x}}\; \\
\subitem Sejam \LTgetval{a}\;, \LTgetval{b}\;, \LTgetval{c}\;, e \LTgetval{d}\; \\
\item \verb \LTgetfx{<F>}{<x>}; : Exibe a definição de F$\left( x \right)$
\subitem Vejamos : \LTgetfx{F}{x}\; \\
\subitem \LTgetfx{G}{x}\; \\
\subitem \LTgetfx{G}{d}\; \\
```

```
\subitem \LTgetfx{Z}{x}\; \\
\item \verb \LTprevalfx{<F>}{<x_0>}; : Exibe o preparo do cálculo de F$\left( x_0 '
\subitem Vejamos : \LTprevalfx{F{x}}{9}\; \\
\subitem \LTprevalfx{G{x}}{1}\; \\
\subitem \LTprevalfx{G{d}}{4}\; \\
\subitem \LTprevalfx{Z{x}}{1}\; \\
\item \verb \LTevalfx{<F>}{<x_0>}; : Calcula F$\left( x_0 \right)$
\subitem Logo: \LTevalfx{F{x}}{9}\;, \LTevalfx{G{x}}{1}\; e \LTevalfx{Z{x}}{1}\; \'
\subitem Mas se houver um 'x' na memoria do sistema (\LTgetval{x}\;)
 pode nos interessar calcular \LTevalfx{G{d}}{4}\;
  por exemplo, trocando a variável independente. \\
\clearpage
\item \verb \LTplotfx{{<F>}{<x>}}{{<N>-<W,H>}{<x1>:<xn>#<step>}}; :\\
 Plota a Função $F\left(x\right)$ para os parâmetros dados:\\
\subitem \LTplotfx{{F}{x}}{{50-10,4}{0:3#0,5}}\; \\
\subitem \LTplotfx{{G}{x}}{{50-10,5}{-0.5:1.5#0,5}}\; \\
\subitem \LTplotfx{{G}{d}}{{50-10,5}{0.5:3.5#0,5}}\; \\
\subitem \LTplotfx{{Z}{x}}{{100-10,4}{0:0.5#0,1}}\; \\

\item \verb \LToperfx{<F>}{<F1-oper-F2-oper-F3>}; realiza a operação entre
as duas ou mais funções da direita e atribui à primeira. \\
\subitem Seja \LToperfx{W}{\frac{F}{G}}\; \\
\subitem Logo \LTprevalfx{W{x}}{5}\; \\
\subitem e \LTevalfx{W{x}}{5}\; \\
\subitem Vamos experimentar plotar a nova função : \\
\subitem \LTplotfx{{W}{x}}{{100-10,5}{.5:4#1}}\; \\
\subitem Vamos ensaiar sua inversa: \LToperfx{P}{\frac{G}{F}}\; \\
\subitem Logo \LTprevalfx{P{x}}{5}\; \\
\subitem e \LTevalfx{P{x}}{5}\; \\
\clearpage
\subitem Vamos experimentar plotar a nova função : \\
\subitem \LTplotfx{{P}{x}}{{100-10,5}{.5:4#1}}\; \\
```

```
\subitem Seja então \LToperfx{Q}{{W}*Z}\; \\
\subitem e \LTprevalfx{Q{x}}{12}\; = \LTevalfx{Q{x}}{12}\; \\
\subitem \LTplotfx{{Q}{x}}{{300-10,8}{2:4#1}}\; \\
\subitem Seja \LToperfx{T}{\sqrt{F} - \cos{a}}\; \\
\subitem e \LToperfx{Y}{\frac{P}{x}}\; \\
\subitem Logo \LTprevalfx{T{x}}{154}\; \\
\subitem e \LTprevalfx{Y{x}}{12}\; \\
\subitem e conferindo: \LTevalfx{T{x}}{154}\; e \LTevalfx{Y{x}}{3}\;
\end{itemize}
\end{math}
\end{document}
```

3.4.2 original.pdf : Visualização

Explicando alguns comandos:

- `\LTsetvar{<var>}{<valor-ou-expressao>};` : Ex: ($a \Leftarrow \left\langle 5 * \tan \frac{3,1415}{4} \right\rangle$) acabamos de fazer uma atribuição à variável 'a'. ($b \Leftarrow \left\langle \sqrt{a+119} \right\rangle$, $d \Leftarrow \left\langle \sqrt{\frac{a}{b}} \right\rangle$ e $ab \Leftarrow \left\langle -1 * \sin \frac{\cos a + b}{\tan 100} \right\rangle$ são outros exemplos). *As variáveis são definidas em letras minusculas podendo ser seguidas por um ou mais numeros e outras letras, também minusculas.*

- `\LTgetvar{<var>};` Retorna a definição da variável, aplicada à 'a' resulta : $a \equiv (\ )$ em b resulta $b \equiv (\ )$ e em ab resulta $ab \equiv (\ )$

- `\LTpreval{<var>};`: Demostra o pré-calculo da variavel, com os valores inseridos no lugar de sua definição. No vaso de ab = $\langle ab \rangle$

- `\LTeval{<var>};` : É um comando para cálculo da variavel, temos: a = $\langle a \rangle$ e aplicada à 'b' resulta b = $\langle b \rangle$ e no caso de ab : $\langle ab \rangle$

- `\LTgetval{<var>};`: Outra alternativa ao get e ao getvar, só que retorna apenas o valor da variável. No exemplo $b = (\,)$ `\LTeval{<var>};`é parecido com getval mas com uma diferença preponderante: getval retorna o ultimo valor da variável na memória, enquanto eval retorna o valor da variavél após novo cálculo, isso é preponderante em variáveis recursivas pois a cada chamada de eval acarretará novo valor enquanto getval não. Vejamos:

  Primeiro definimos $c \Leftarrow \left\langle \sqrt{a+b} \right\rangle$ e seu valor de partida $c = (\,)$ pois $\langle c \rangle$

  Depois definimos $b \Leftarrow \left\langle \frac{\sqrt{c}}{c} \right\rangle$ e seu novo valor $b = (\,)$

  Depois fazemos um primeiro eval : $\langle c \rangle$ e $\langle b \rangle$

  Depois fazemos um segundo eval : $\langle c \rangle$ e $\langle b \rangle$

  Calculemos agora ab: $\langle ab \rangle$

  Enquanto isso um getval retorna o valor anterior: $c = (\,)$ e $b = (\,)$

  Se uma variavel depender dela mesma: $d \Leftarrow \langle a + d + 1 \rangle$ com seu primeiro valor $d = (\,)$ Se fizermos seguidos eval dessa variável obteremos seguidos valores: $\langle d \rangle$ e $\langle d \rangle$ e $\langle d \rangle$ respectivamente.

  Definamos agora $a \Leftarrow \left\langle \sqrt{5} \right\rangle$ $x \Leftarrow \langle \sqrt{a} \rangle$ e portanto x = $(\langle x \rangle\,)$.

- `\LTplot{{<x>}{<expressão>}}{{<N>-<W,H>}{<x1>:<xn>#<step>}};`: Plota a Função de x pela expressão dada, para N pontos de amostragem e com dominio entre x1 e xn e discretização 'step'.. Conveniente avisar que ela efetua a substituição pelos literais na memória exceto pelo argumento informado em x, como veremos a seguir.

  Ao fazer uso dos parametros da função plot, pode-se ajustar ,melhor o tamanho e o eixo de acordo com a curva para uma melhor documentação como segue na sequencia de exemplos a seguir, nos quais apresento algumas possibilidades e limitações de desempenho.

  É interessante ressaltar a natureza interativa do processo de escolha do tamanho e dominio adequados à ilustração da função. O algoritmo atual , como é explicado na

seção sobre o modo de funcionamento dos subVIs componentes, trabalha com deteção de máximo e mínimo da função para composição do eixo y, usando para isso um passo de amostragem inicialmente configurado para 0.01, o que durante o funcionamento no computador pessoal do aluno, não causou demora na execução mas restringe o 'zoom-in' de plotagem a uma escala de 100. Foram efetuados testes com passos de 0.001 causando sensivel lentidão durante a execução.

A seguir uma série de funções e seus respectivos zooms em áreas de interesse para caracterizar o processo interativo de escolha dos parâmetros mais adequados para representação (domínio, largura, comprimento e discretização da grade e tomada de valores).

Calculando b= ($\langle b \rangle$) , a = ($\langle a \rangle$) e x = ($\langle x \rangle$). e para 50 amostras :

Plotar ($x \sin a * x + \cos b * x$) com os dados: 50-10,50:151 $\overbrace{50-10,50:151}^{<N>-<W,H><xi:xf><yi:yf>}$

E aumentando o número de amostras para 500 e aumentando a altura da imagem:

Plotar $\langle x\sin a * x + \cos b * x\rangle$ com os dados: 500-10,100:152 $\overbrace{500-10,100:152}^{<N>-<W,H><xi:xf><yi:yf>}$

Restrição do domínio :

Plotar $\langle x\sin a * x + \cos b * x\rangle$ com os dados: 50-10,50:51 $\overbrace{50-10,50:51}^{<N>-<W,H><xi:xf><yi:yf>}$

Plotar $\langle x\sin a * x + \cos b * x\rangle$ com os dados: 50-10,52:40,5 $\overbrace{50-10,52:40,5}^{<N>-<W,H><xi:xf><yi:yf>}$

Plotar $\langle x\sin a * x + \cos b * x\rangle$ com os dados: 50-10,52:2.50,2 $\overbrace{50-10,52:2.50,2}^{<N>-<W,H><xi:xf><yi:yf>}$

Plotar $\langle x\sin a * x + \cos b * x\rangle$ com os dados: 50-10,52.2:2.30,02 $\overbrace{50-10,52.2:2.30,02}^{<N>-<W,H><xi:xf><yi:yf>}$

Como observa-se, a escolha adequada do passo de amostragem permite foco na regiião de interesse sobre o domínio, encontra-se anexo a este roteiro de teste uma sequencia de tentativas na obtenção da melhor visualização das funções como alguns dos resultados obtidos pelo comando `\LTplot`; mas ainda há uma sêrie de particulariedades a serem descobertas de acordo com o uso e exploração das estruturas.

Também interessante observar que se chamamos o mesmo comando plot para a mesma função, mas passando como referência outra variável independente, o comando faz a adequação dos parâmetros :

Plotar $\langle a\sin a * x + \cos b * x\rangle$ com os dados: 50-8,51.5:40,5 $\overbrace{50-8,51.5:40,5}^{<N>-<W,H><xi:xf><yi:yf>}$

Plotar $\langle b\sin a * x + \cos b * x\rangle$ com os dados: 50-8,52.5:50,5 $\overbrace{50-8,52.5:50,5}^{<N>-<W,H><xi:xf><yi:yf>}$

- `\LTsetfx{<F>}{<formula>};`: Cria a função F(x) atribuindo a ela a formula:

  Seja $F(x) \equiv a * x + b * x + c$

  e $G(x) \equiv \sqrt{d} * x^3 - 6 * x + 12$

  e $Z(x) \equiv \sin 10 * x - \cos 50 * x$

  Sejam $a = (\ )$, $b = (\ )$, $c = (\ )$, e $d = (\ )$

- `\LTgetfx{<F>}{<x>};`: Exibe a definição de F(x)

  Vejamos : $F(x) \equiv$

  $G(x) \equiv$

  $G(d) \equiv$

  $Z(x) \equiv$

- `\LTprevalfx{<F>}{<x_0>};`: Exibe o preparo do cálculo de F(x_0)

  Vejamos : $Fx(9 \rightarrow) =$

  $Gx(1 \rightarrow) =$

  $Gd(4 \rightarrow) =$

  $Zx(1 \rightarrow) =$

- `\LTevalfx{<F>}{<x_0>};`: Calcula F(x_0)

  Logo: $Fx\,(9 \rightarrow\ ) =$, $Gx\,(1 \rightarrow\ ) = e\ \ Zx\,(1 \rightarrow\ ) =$

  Mas se houver um 'x' na memoria do sistema ($x = (\ )$) pode nos interessar calcular $Gd\,(4 \rightarrow\ ) =$ por exemplo, trocando a variável independente.

- `\LTplotfx{{<F>}{<x>}}{{<N>-<W,H>}{<x1>:<xn>#<step>}};`:

  Plota a Função $F\,(x)$ para os parâmetros dados:

  $PlotarFx\,(50 - 10, 40 : 30, 5)\,comosdados$:

  $PlotarGx\,(50 - 10, 5{-}0.5 : 1.50, 5)\,comosdados$:

  $PlotarGd\,(50 - 10, 50.5 : 3.50, 5)\,comosdados$:

  $PlotarZx\,(100 - 10, 40 : 0.50, 1)\,comosdados$:

- `\LToperfx{<F>}{<F1-oper-F2-oper-F3>}`;realiza a operação entre as duas ou mais funções da direita e atribui à primeira.

  Seja $W\,(x) = \dfrac{F}{G} =$

  Logo $Wx\,(5 \rightarrow\ ) =$

  e $Wx\,(5 \rightarrow\ ) =$

  Vamos experimentar plotar a nova função :

$PlotarWx\,(100-10,5.5:41)\,comosdados$:

Vamos ensaiar sua inversa: $P\,(x)\equiv\dfrac{G}{F}\equiv$

Logo $Px\,(5\rightarrow)=$

e $Px\,(5\rightarrow)=$

Vamos experimentar plotar a nova função :

$PlotarPx\,(100-10,5.5:41)\,comosdados$:

Seja então $Q\,(x)\equiv W*Z\equiv$

e $Qx\,(12\rightarrow)==Qx\,(12\rightarrow)=$

$PlotarQx\,(300-10,82:41)\,comosdados$:

Seja $T\,(x)\equiv\sqrt{F}-\cos a\equiv$

e $Y\,(x)\equiv\dfrac{P}{x}\equiv$

Logo $Tx\,(154\rightarrow)=$

e $Yx\,(12\rightarrow)=$

e conferindo: $Tx\,(154\rightarrow)=$ e $Yx\,(3\rightarrow)=$itemize

3.4.3 resultado.tex : Código Tex

```
%& --shell-escape dubble
\documentclass[10pt,letterpaper]{article}
\usepackage[brazil]{babel}
\usepackage[latin1]{inputenc} %codificacao para latin.
\usepackage{a4wide}
\usepackage{hyperref}
\usepackage{amssymb,amsmath}
\usepackage{datetime}
\usepackage{tikz}
\everymath{\displaystyle}
\title{LabTeX - Modo Linha de Comando}
%\author{Fernando J. Capeletto}
\begin{document}
%\maketitle
\thispagestyle{empty}
\newcommand{\matpar}[2]{\raisebox{1pt}{\small
\ensuremath{#1}}$/$\raisebox{-2pt}{\small \ensuremath{#2}}}
% Comandos LabTeX
\newcommand{\LTsetvar}[2]{ $ #1\Leftarrow\left\langle#2\right\rangle $}
\newcommand{\LTpreval}[1]{$\left\langle #1 \right\rangle $}
\newcommand{\LTeval}[1]{$\left\langle #1 \right\rangle $}
\newcommand{\LTgetval}[2]{$#1 - \left(#2\right)$}
\newcommand{\LTgetvar}[2]{$#1 \equiv (#2)$}
\newcommand{\LTplot}[2] { Plotar \left\langle #1 \right\rangle
$ com os dados: #2$ \overbrace{#2}^{<N>-<W,H><xi:xf><yi:yf>}}
\newcommand{\LTsetfx}[2] { $ #1\left(x\right) \equiv #2 $}
\newcommand{\LTevalfx}[4] { $ #1\left(#2 \rightarrow #3\right) = #4 $}
\newcommand{\LTprevalfx}[4] { $ #1\left(#2 \rightarrow #3\right) - #4 $}
\newcommand{\LTgetfx}[3] { $ #1\left(#2 \right)\equiv #3 $}
\newcommand{\LTplotfx}[2] { $Plotar #1 \left(#2 \right) com os dados: $}
```

```
\newcommand{\LToperfx}[3] { $ #1\left(x\right) \equiv #2 \equiv #3$}

 \begin{math}
 $Explicando alguns comandos:  \\
\begin{itemize}
\item \verb \LTsetvar{<var>}{<valor-ou-expressao>};  :
 Ex: (\LTsetvar{a}{5*\tan{\frac{3,1415}{4}}}\;) acabamos de fazer uma atribuição
  à variável 'a'. (\LTsetvar{b}{\sqrt{a+119}}\; ,  \LTsetvar{d}{\sqrt{\frac{a}{b}}}
   e  \LTsetvar{ab}{-1*\sin{\frac{\cos{a}+b}{\tan{100}}}}\; são outros exemplos).
    \emph{As variáveis são definidas em letras minusculas podendo ser seguidas
    por um ou mais numeros e outras letras, também minusculas.}
\item  \verb \LTgetvar{<var>};  Retorna a definição da variável, aplicada à 'a'
 resulta : \LTgetvar{a}{5*\tan{\frac{3,1415}{4}}}\;
  em b resulta \LTgetvar{b}{\sqrt{a+119}}\;
   e em ab resulta \LTgetvar{ab}{-1*\sin{\frac{\cos{a}+b}{\tan{100}}}}\;
\item \verb \LTpreval{<var>}; : Demostra o pré-calculo da variavel, com os valores
inseridos no lugar de sua definição.
 No vaso de ab =
 \LTpreval{-1*\sin{\frac{\cos{(4,999768)}+(11,135518)}{\tan{100}}}}\; \\
\item \verb \LTeval{<var>};  : É um comando para cálculo da variavel, temos:
 a = \LTeval{4,999768}\; e aplicada à 'b' resulta b = \LTeval{11,135518}\;
  e no caso de ab : \LTeval{0,561698}\;
\item \verb \LTgetval{<var>}; : Outra alternativa ao get e ao getvar, só que
 retorna apenas o valor da variável. No exemplo \LTgetval{b}{11,135518}\;
   \verb \LTeval{<var>}; é parecido com getval mas com uma diferença: getval
    retorna o ultimo valor da variável na memória, enquanto eval retorna o
     valor da variavél após novo cálculo, isso é preponderante em variáveis
      recursivas pois a cada chamada de eval acarretará novo valor enquanto
       getval não. Vejamos:
\subitem Primeiro definimos \LTsetvar{c}{\sqrt{a+b}}\; e seu valor de partida
 \LTgetval{c}{4,016875}\; pois \LTpreval{\sqrt{(4,999768)+(11,135518)}}\;
\subitem Depois definimos \LTsetvar{b}{\frac{\sqrt{c}}{c}}\; e seu novo valor
```

```
\LTgetval{b}{0,498949}\;
\subitem Depois fazemos um primeiro eval : \LTeval{2,344934}\; e \LTeval{0,653032}'
\subitem Depois fazemos um segundo eval : \LTeval{2,377562}\; e \LTeval{0,648536}\
\subitem Calculemos agora ab: \LTeval{0,999867}\;
\subitem Enquanto isso um getval retorna o valor anterior: \LTgetval{c}{2,377562}\
 e \LTgetval{b}{0,648536}\;
\subitem Se uma variavel depender dela mesma: \LTsetvar{d}{a+d+1}\;
 com seu primeiro valor \LTgetval{d}{6,669837}\;  Se fizermos seguidos eval dessa
 variável obteremos seguidos valores: \LTeval{12,669605}\; e \LTeval{18,669373}\;
  e \LTeval{24,669141}\; respectivamente.
\subitem Definamos agora \LTsetvar{a}{\sqrt{5}}\; \LTsetvar{x}{\sqrt{a}}\; e
portanto x = (\LTeval{1,495349}\;). \\
\item \verb \LTplot{{<x>}{<expressão>}}{{<N>-<W,H>}{<x1>:<xn>#<step>}}; : Plota
a Função de x pela expressão dada, para N pontos de amostragem e com dominio entre
 x1 e xn e discretização 'step'.. Conveniente avisar que ela efetua a substituição
  pelos literais na memória exceto pelo argumento informado em {x}, como veremos a
   seguir. \\

\emph{Ao fazer uso dos parametros da função plot, pode-se ajustar ,melhor o tamanh
 e o eixo de acordo com a curva para uma melhor documentação como segue na sequenc
  de exemplos a seguir, nos quais apresento algumas possibilidades e limitações de
  desempenho.}\\

É interessante ressaltar a natureza interativa do processo de escolha do tamanho e
 dominio adequados à ilustração da função. O algoritmo atual , como é explicado na
  seção sobre o modo de funcionamento dos subVIs componentes, trabalha com deteção
   de máximo e mínimo da função para composição do eixo y, usando para isso um pas
    de amostragem inicialmente configurado para 0.01, o que durante o funcionament
     no computador pessoal do aluno, não causou demora na execução mas restringe o
      'zoom-in' de plotagem a uma escala de 100. Foram efetuados testes com passos
       0.001 causando sensivel lentidão durante a execução.
```

```
A seguir uma série de funções e seus respectivos zooms em áreas de interesse para
 caracterizar o processo interativo de escolha dos parâmetros mais adequados para
  representação (domínio, largura, comprimento e discretização da grade e tomada
   de valores).

\subitem Calculando b= (\LTeval{0,648536}\;) , a = (\LTeval{2,236068}\;)
 e x = (\LTeval{1,495349}\;).  e para 50 amostras : \\
\subitem   \begin{tikzpicture}[samples=50,domain=0:15,xscale=0.666667,yscale=1.269
\draw[very thin,color=gray,step=2.000000cm,dashed] (0,-1.998952) grid (15,1.940486
    \draw[->] (0,-1.998952) -- (15,-1.998952) node[right] {$x$};
    \draw[->] (0,-1.998952) -- (0,1.940486) node[above] {$f(x)$};
% units for cartesian reference frame
   \foreach \x/\xtext in {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15}
        \draw (\x cm,-1.998952) -- (\x cm,-1.998952)
            node[anchor=north,xshift=-0.15cm] {$ \xtext $};
\newcommand{\tickyi}{3/0.666667}
\newcommand{\tickyf}{1/0.666667}
    \foreach \y/\ytext in {1,1.583718,-0.700959,0.046241,-0.391627,-1.977874,0.019
    -0.116322,-0.361634,1.858611,
    0.618454,0.148754,1.063144,-1.259049,-1.05173,-0.104097}
        \draw (0 cm+\tickyi pt,\y cm) -- (0 cm-\tickyf pt,\y cm) node[anchor=east]
        {$ \ytext $};\draw[color=blue] plot[id=x] function{sin((2.236068)*x)
        +cos((0.648536)*x)} node[above] {$f(x)=\sin{(2,236068)*x}
        +\cos{(0,648536)*x}$};\end{tikzpicture} \\
\clearpage

\subitem E aumentando o número de amostras para 500 e aumentando a altura da image
\subitem   \begin{tikzpicture}[samples=500,domain=0:15,xscale=0.666667,yscale=2.53
\draw[very thin,color=gray,step=2.000000cm,dashed] (0,-1.998952) grid (15,1.940486
    \draw[->] (0,-1.998952) -- (15,-1.998952) node[right] {$x$};
    \draw[->] (0,-1.998952) -- (0,1.940486) node[above] {$f(x)$};
% units for cartesian reference frame
```

```
   \foreach \x/\xtext in {0,2,4,6,8,10,12,14}
        \draw (\x cm,-1.998952) -- (\x cm,-1.998952)
            node[anchor=north,xshift=-0.15cm] {$ \xtext $};
\newcommand{\tickyi}{3/0.666667}
\newcommand{\tickyf}{1/0.666667}
    \foreach \y/\ytext in {1,-0.700959,-0.391627,0.01936,-0.361634,0.618454,1.0631
    -1.05173}
        \draw (0 cm+\tickyi pt,\y cm) -- (0 cm-\tickyf pt,\y cm) node[anchor=east]
        {$ \ytext $};\draw[color=blue] plot[id=x] function{sin((2.236068)*x)
        +cos((0.648536)*x)} node[above] {$f(x)=\sin{(2,236068)*x}
        +\cos{(0,648536)*x}$};\end{tikzpicture} \\
\subitem Restrição do domínio : \\
\subitem   \begin{tikzpicture}[samples=50,domain=0:5,xscale=2.000000,
yscale=1.280564]
\draw[very thin,color=gray,step=0.500000cm,dashed] (0,-1.998952)
grid (5,1.905579);
    \draw[->] (0,-1.998952) -- (5,-1.998952) node[right] {$x$};
    \draw[->] (0,-1.998952) -- (0,1.905579) node[above] {$f(x)$};
% units for cartesian reference frame
   \foreach \x/\xtext in {0,1,2,3,4,5}
        \draw (\x cm,-1.998952) -- (\x cm,-1.998952)
            node[anchor=north,xshift=-0.15cm] {$ \xtext $};
\newcommand{\tickyi}{3/2.000000}
\newcommand{\tickyf}{1/2.000000}
    \foreach \y/\ytext in {1,1.583718,-0.700959,0.046241,-0.391627,-1.977874}
        \draw (0 cm+\tickyi pt,\y cm) -- (0 cm-\tickyf pt,\y cm) node[anchor=east]
        {$ \ytext $};\draw[color=blue] plot[id=x] function{sin((2.236068)*x)
        +cos((0.648536)*x)} node[above] {$f(x)=\sin{(2,236068)*x}
        +\cos{(0,648536)*x}$};\end{tikzpicture} \\
\subitem   \begin{tikzpicture}[samples=50,domain=2:4,xscale=5.000000,
yscale=4.116168]
\draw[very thin,color=gray,step=0.500000cm,dashed] (2,-0.838592)
```

```
grid (4,0.37613);
    \draw[->] (2,-0.838592) -- (4,-0.838592) node[right] {$x$};
    \draw[->] (2,-0.838592) -- (2,0.37613) node[above] {$f(x)$};
% units for cartesian reference frame
   \foreach \x/\xtext in {2,2.5,3,3.5,4}
        \draw (\x cm,-0.838592) -- (\x cm,-0.838592)
            node[anchor=north,xshift=-0.15cm] {$ \xtext $};
\newcommand{\tickyi}{3/5.000000}
\newcommand{\tickyf}{1/5.000000}
    \foreach \y/\ytext in {-0.700959,-0.689382,0.046241,0.356102,-0.391627}
        \draw (2 cm+\tickyi pt,\y cm) -- (2 cm-\tickyf pt,\y cm) node[anchor=east]
         {$ \ytext $};\draw[color=blue] plot[id=x] function{sin((2.236068)*x)
         +cos((0.648536)*x)} node[above] {$f(x)=\sin{(2,236068)*x}
         +\cos{(0,648536)*x}$};\end{tikzpicture} \\
\subitem   \begin{tikzpicture}[samples=50,domain=2:2.5,
xscale=20.000000,yscale=33.509818]
\draw[very thin,color=gray,step=0.050000cm,dashed] (2,-0.838592)
 grid (2.5,-0.689382);
    \draw[->] (2,-0.838592) -- (2.5,-0.838592) node[right] {$x$};
    \draw[->] (2,-0.838592) -- (2,-0.689382) node[above] {$f(x)$};
% units for cartesian reference frame
   \foreach \x/\xtext in {2,2.2,2.4}
        \draw (\x cm,-0.838592) -- (\x cm,-0.838592)
            node[anchor=north,xshift=-0.15cm] {$ \xtext $};
\newcommand{\tickyi}{3/20.000000}
\newcommand{\tickyf}{1/20.000000}
    \foreach \y/\ytext in {-0.700959,-0.83514,-0.779241}
        \draw (2 cm+\tickyi pt,\y cm) -- (2 cm-\tickyf pt,\y cm)
         node[anchor=east] {$ \ytext $};\draw[color=blue] plot[id=x]
          function{sin((2.236068)*x)+cos((0.648536)*x)} node[above] {$f(x)=\sin{(2
\subitem   \begin{tikzpicture}[samples=50,domain=2.2:2.3,
xscale=100.000000,yscale=559.033989]
```

```
\draw[very thin,color=gray,step=0.010000cm,dashed] (2.2,-0.838592)
 grid (2.3,-0.829648);
    \draw[->] (2.2,-0.838592) -- (2.3,-0.838592) node[right] {$x$};
    \draw[->] (2.2,-0.838592) -- (2.2,-0.829648) node[above] {$f(x)$};
% units for cartesian reference frame
   \foreach \x/\xtext in {2.2,2.22,2.24,2.26,2.28,2.3}
       \draw (\x cm,-0.838592) -- (\x cm,-0.838592)
           node[anchor=north,xshift=-0.15cm] {$ \xtext $};
\newcommand{\tickyi}{3/100.000000}
\newcommand{\tickyf}{1/100.000000}
    \foreach \y/\ytext in {-0.83514,-0.837823,-0.838592,-0.837467,
    -0.834476,-0.829648}
       \draw (2.2 cm+\tickyi pt,\y cm) -- (2.2 cm-\tickyf pt,\y cm)
       node[anchor=east] {$ \ytext $};\draw[color=blue] plot[id=x]
       function{sin((2.236068)*x)+cos((0.648536)*x)} node[above] {$f(x)=\sin{(2,2:
```

```
Como observa-se, a escolha adequada do passo de amostragem permite foco
 na regiião de interesse sobre o domínio, encontra-se anexo a este roteiro
  de teste uma sequencia de tentativas na obtenção da melhor visualização
   das funções como alguns dos resultados obtidos pelo comando \verb \LTplot;
    mas ainda há uma sêrie de particulariedades a serem descobertas de acordo
     com o uso e exploração das estruturas. \\
\subitem Também interessante observar que se chamamos o mesmo comando plot para
 a mesma função, mas passando como referência outra variável independente,
 o comando faz a adequação dos parâmetros :
\subitem   \begin{tikzpicture}[samples=50,domain=1.5:4,
xscale=3.200000,yscale=2.805153]
\draw[very thin,color=gray,step=0.500000cm,dashed] (1.5,-0.434523)
 grid (4,1.347911);
    \draw[->] (1.5,-0.434523) -- (4,-0.434523) node[right] {$x$};
    \draw[->] (1.5,-0.434523) -- (1.5,1.347911) node[above] {$f(x)$};
% units for cartesian reference frame
```

```
    \foreach \x/\xtext in {1.5,2,2.5,3,3.5,4}
        \draw (\x cm,-0.434523) -- (\x cm,-0.434523)
            node[anchor=north,xshift=-0.15cm] {$ \xtext $};
\newcommand{\tickyi}{3/3.200000}
\newcommand{\tickyf}{1/3.200000}
    \foreach \y/\ytext in {1.347911,0.715797,0.003493,-0.409019,-0.301682,0.268246,
        \draw (1.5 cm+\tickyi pt,\y cm) -- (1.5 cm-\tickyf pt,\y cm) node[anchor=e
        {$ \ytext $};\draw[color=blue] plot[id=a] function{sin(x*(1.495349))
        +cos((0.648536)*(1.495349))} node[above] {$f(a)=\sin{a*(1,495349)}+\cos{(0
\subitem   \begin{tikzpicture}[samples=50,domain=2.5:5,xscale=3.200000,
yscale=2.736508]
\draw[very thin,color=gray,step=0.500000cm,dashed] (2.5,-1.027886)
 grid (5,0.79926);
    \draw[->] (2.5,-1.027886) -- (5,-1.027886) node[right] {$x$};
    \draw[->] (2.5,-1.027886) -- (2.5,0.79926) node[above] {$f(x)$};
% units for cartesian reference frame
   \foreach \x/\xtext in {2.5,3,3.5,4,4.5,5}
        \draw (\x cm,-1.027886) -- (\x cm,-1.027886)
            node[anchor=north,xshift=-0.15cm] {$ \xtext $};
\newcommand{\tickyi}{3/3.200000}
\newcommand{\tickyf}{1/3.200000}
    \foreach \y/\ytext in {-1.027886,-0.425151,0.2973,0.75407,0.701493,0.167617}
        \draw (2.5 cm+\tickyi pt,\y cm) -- (2.5 cm-\tickyf pt,\y cm) node[anchor=e
        {$ \ytext $};\draw[color=blue] plot[id=b] function{sin((2.236068)*(1.4953
        +cos(x*(1.495349))} node[above] {$f(b)=\sin{(2,236068)*(1,495349)}
        +\cos{b*(1,495349)}$};\end{tikzpicture} \\

\clearpage
\item \verb \LTsetfx{<F>}{<formula>}; : Cria a função F$\left( x \right)
$ atribuindo a ela a formula: \\
\subitem Seja \LTsetfx{F}{a*x+b*x+c}\; \\
\subitem  e \LTsetfx{G}{\sqrt{d}*x^{3}-6*x+12}\;  \\
```

```
\subitem e \LTsetfx{Z}{\sin{10*x}-\cos{50*x}}\; \\
\subitem Sejam \LTgetval{a}{2,236068}\;, \LTgetval{b}{0,648536}\;,
 \LTgetval{c}{2,377562}\;, e \LTgetval{d}{24,669141}\; \\
\item \verb \LTgetfx{<F>}{<x>}; : Exibe a definição de
 F$\left( x \right)$
\subitem Vejamos : \LTgetfx{F}{x}{a*x+b*x+c}\; \\
\subitem \LTgetfx{G}{x}{\sqrt{d}*x^{3}-6*x+12}\; \\
\subitem \LTgetfx{G}{d}{\sqrt{d}*x^{3}-6*x+12}\; \\
\subitem \LTgetfx{Z}{x}{\sin{10*x}-\cos{50*x}}\; \\
\item \verb \LTprevalfx{<F>}{<x_0>}; : Exibe o preparo do cálculo
de F$\left( x_0 \right)$
\subitem Vejamos : \LTprevalfx{F}{x}{9}{(2,236068)*9+(0,648536)
*9+(2,377562)}\; \\
\subitem \LTprevalfx{G}{x}{1}{\sqrt{(24,669141)}*1^{3}-6*1+12}\; \\
\subitem \LTprevalfx{G}{d}{4}{\sqrt{4}*(1,495349)^{3}-6*(1,495349)+12}\; \\
\subitem \LTprevalfx{Z}{x}{1}{\sin{10*1}-\cos{50*1}}\; \\
\item \verb \LTevalfx{<F>}{<x_0>}; : Calcula F$\left( x_0 \right)$
\subitem Logo: \LTevalfx{F}{x}{9}{28,338998}\;, \LTevalfx{G}{x}{1}{10,966804}\;
 e \LTevalfx{Z}{x}{1}{-1,508987}\; \\
\subitem Mas se houver um 'x' na memoria do sistema (\LTgetval{x}{1,495349}\;)
 pode nos interessar calcular \LTevalfx{G}{d}{4}{3,027906}\; por exemplo,
  trocando a variável independente. \\
\clearpage
\item \verb \LTplotfx{{<F>}{<x>}}{{<N>-<W,H>}{<x1>:<xn>#<step>}}; :\\
 Plota a Função $F\left(x\right)$ para os parâmetros dados:\\
\subitem   \begin{tikzpicture}[samples=50,domain=0:3,xscale=3.333333,
yscale=0.462224]
\draw[very thin,color=gray,step=0.500000cm,dashed] (0,2.377562)
 grid (3,11.031374);
    \draw[->] (0,2.377562) -- (3,2.377562) node[right] {$x$};
    \draw[->] (0,2.377562) -- (0,11.031374) node[above] {$f(x)$};
% units for cartesian reference frame
```

```
    \foreach \x/\xtext in {0,0.5,1,1.5,2,2.5,3}
        \draw (\x cm,2.377562) -- (\x cm,2.377562)
            node[anchor=north,xshift=-0.15cm] {$ \xtext $};
\newcommand{\tickyi}{3/3.333333}
\newcommand{\tickyf}{1/3.333333}
    \foreach \y/\ytext in {2.377562,3.819864,5.262166,6.704468,
    8.14677,9.589072,11.031374}
        \draw (0 cm+\tickyi pt,\y cm) -- (0 cm-\tickyf pt,\y cm)
         node[anchor=east] {$ \ytext $};\draw[color=blue] plot[id=x]
          function{(2.236068)*x+(0.648536)*x+(2.377562)} node[above] {$F(x)=(2,236
\subitem   \begin{tikzpicture}[samples=50,domain=-0.5:1.5,
xscale=5.000000,yscale=0.485388]
\draw[very thin,color=gray,step=0.500000cm,dashed] (-0.5,9.461934)
grid (1.5,19.762963);
    \draw[->] (-0.5,9.461934) -- (1.5,9.461934) node[right] {$x$};
    \draw[->] (-0.5,9.461934) -- (-0.5,19.762963) node[above] {$f(x)$};
% units for cartesian reference frame
   \foreach \x/\xtext in {-0.5,0,0.5,1,1.5}
        \draw (\x cm,9.461934) -- (\x cm,9.461934)
            node[anchor=north,xshift=-0.15cm] {$ \xtext $};
\newcommand{\tickyi}{3/5.000000}
\newcommand{\tickyf}{1/5.000000}
    \foreach \y/\ytext in {14.37915,12,9.62085,10.966804,19.762963}
        \draw (-0.5 cm+\tickyi pt,\y cm) -- (-0.5 cm-\tickyf pt,\y cm)
         node[anchor=east] {$ \ytext $};\draw[color=blue] plot[id=x] function{((((:
\subitem   \begin{tikzpicture}[samples=50,domain=0.5:3.5,xscale=3.333333,
yscale=0.241405]
\draw[very thin,color=gray,step=0.500000cm,dashed] (0.5,4.210084)
 grid (3.5,24.92214);
    \draw[->] (0.5,4.210084) -- (3.5,4.210084) node[right] {$x$};
    \draw[->] (0.5,4.210084) -- (0.5,24.92214) node[above] {$f(x)$};
% units for cartesian reference frame
```

```
   \foreach \x/\xtext in {0.5,1,1.5,2,2.5,3,3.5}
        \draw (\x cm,4.210084) -- (\x cm,4.210084)
           node[anchor=north,xshift=-0.15cm] {$ \xtext $};
\newcommand{\tickyi}{3/3.333333}
\newcommand{\tickyf}{1/3.333333}
    \foreach \y/\ytext in {4.210084,6.371609,9.170681,12.485326,16.245053,
    20.402296,24.92214}
        \draw (0.5 cm+\tickyi pt,\y cm) -- (0.5 cm-\tickyf pt,\y cm)
         node[anchor=east] {$ \ytext $};\draw[color=blue] plot[id=d] function{{{{{
\subitem   \begin{tikzpicture}[samples=100,domain=0:0.5,xscale=20.000000,
yscale=1.027327]
\draw[very thin,color=gray,step=0.050000cm,dashed] (0,-1.950127)
grid (0.5,1.943472);
    \draw[->] (0,-1.950127) -- (0.5,-1.950127) node[right] {$x$};
    \draw[->] (0,-1.950127) -- (0,1.943472) node[above] {$f(x)$};
% units for cartesian reference frame
   \foreach \x/\xtext in {0,0.1,0.2,0.3,0.4,0.5}
        \draw (\x cm,-1.950127) -- (\x cm,-1.950127)
           node[anchor=north,xshift=-0.15cm] {$ \xtext $};
\newcommand{\tickyi}{3/20.000000}
\newcommand{\tickyf}{1/20.000000}
    \foreach \y/\ytext in {-1,0.557809,1.748369,0.900808,-1.164885,-1.950127}
        \draw (0 cm+\tickyi pt,\y cm) -- (0 cm-\tickyf pt,\y cm)
         node[anchor=east] {$ \ytext $};\draw[color=blue] plot[id=x]
          function{sin(10*x)-cos(50*x)} node[above] {$Z(x)=
          \sin{10*x}-\cos{50*x}$};\end{tikzpicture} \\

\item \verb \LToperfx{<F>}{<F1-oper-F2-oper-F3>}; realiza a operação entre
as duas ou mais funções da direita e atribui à primeira. \\
\subitem Seja \LToperfx{W}{\frac{F}{G}}{\frac{(a*x+b*x+c)}
{(\sqrt{d}*x^{3}-6*x+12)}}\; \\
\subitem Logo \LTprevalfx{W}{x}{5}{\frac{((2,236068)*5+(0,648536)*5+(2,377562))}
```

```
{(\sqrt{(24,669141)}*5^{3}-6*5+12)}}\; \\
\subitem e \LTevalfx{W}{x}{5}{0,027869}\; \\
\subitem Vamos experimentar plotar a nova função : \\
\subitem   \begin{tikzpicture}[samples=100,domain=.5:4,xscale=2.857143,
yscale=11.334424]
\draw[very thin,color=gray,step=0.500000cm,dashed] (.5,0.045496)
grid (4,0.48663);
    \draw[->] (.5,0.045496) -- (4,0.045496) node[right] {$x$};
    \draw[->] (.5,0.045496) -- (.5,0.48663) node[above] {$f(x)$};
% units for cartesian reference frame
   \foreach \x/\xtext in {0.5,1.5,2.5,3.5}
        \draw (\x cm,0.045496) -- (\x cm,0.045496)
            node[anchor=north,xshift=-0.15cm] {$ \xtext $};
\newcommand{\tickyi}{3/2.857143}
\newcommand{\tickyf}{1/2.857143}
    \foreach \y/\ytext in {0.39704,0.339244,0.128529,0.06116}
        \draw (.5 cm+\tickyi pt,\y cm) -- (.5 cm-\tickyf pt,\y cm)
        node[anchor=east] {$ \ytext $};\draw[color=blue] plot[id=x] function{(((2.:
        **(0.5))*((x)**(3))-6*x+12))} node[above] {$W(x)=\frac{((2,236068)*x+(0,64:
        *x^{3}-6*x+12)}$};\end{tikzpicture} \\
\subitem Vamos ensaiar sua inversa: \LToperfx{P}{\frac{G}{F}}
{\frac{(\sqrt{d}*x^{3}-6*x+12)}{(a*x+b*x+c)}}\; \\
\subitem Logo \LTprevalfx{P}{x}{5}{\frac{(\sqrt{(24,669141)}*5^{3}-6*5+12)}
{((2,236068)*5+(0,648536)*5+(2,377562))}}\; \\
\subitem e \LTevalfx{P}{x}{5}{35,882715}\; \\
\clearpage
\subitem Vamos experimentar plotar a nova função : \\
\subitem   \begin{tikzpicture}[samples=100,domain=.5:4,xscale=2.857143,
yscale=0.250938]
\draw[very thin,color=gray,step=0.500000cm,dashed] (.5,2.054951)
 grid (4,21.980162);
    \draw[->] (.5,2.054951) -- (4,2.054951) node[right] {$x$};
```

```
    \draw[->] (.5,2.054951) -- (.5,21.980162) node[above] {$f(x)$};
% units for cartesian reference frame
   \foreach \x/\xtext in {0.5,1.5,2.5,3.5}
        \draw (\x cm,2.054951) -- (\x cm,2.054951)
            node[anchor=north,xshift=-0.15cm] {$ \xtext $};
\newcommand{\tickyi}{3/2.857143}
\newcommand{\tickyf}{1/2.857143}
    \foreach \y/\ytext in {2.518637,2.94773,7.780347,16.35057}
        \draw (.5 cm+\tickyi pt,\y cm) -- (.5 cm-\tickyf pt,\y cm)
        node[anchor=east] {$ \ytext $};\draw[color=blue] plot[id=x] function{(((((
        (((2.236068)*x+(0.648536)*x+(2.377562)))} node[above] {$P(x)=\frac{(\sqrt{
        *x+(2,377562))}$};\end{tikzpicture} \\
\subitem Seja então \LToperfx{Q}{{W}*Z}{{(\frac{(a*x+b*x+c)}
{(\sqrt{d}*x^{3}-6*x+12)})}*(\sin{10*x}-\cos{50*x})}\; \\
\subitem e \LTprevalfx{Q}{x}{12}{{(\frac{((2,236068)*12
+(0,648536)*12+(2,377562))}
{(\sqrt{(24,669141)}*12^{3}-6*12+12)})}*(\sin{10*12}-\cos{50*12})}\;
 = \LTevalfx{Q}{x}{12}{0,006856}\; \\
\subitem   \begin{tikzpicture}[samples=300,domain=2:4,xscale=5.000000,
yscale=12.372046]
\draw[very thin,color=gray,step=0.500000cm,dashed] (2,-0.274227)
 grid (4,0.372392);
    \draw[->] (2,-0.274227) -- (4,-0.274227) node[right] {$x$};
    \draw[->] (2,-0.274227) -- (2,0.372392) node[above] {$f(x)$};
% units for cartesian reference frame
   \foreach \x/\xtext in {2,3,4}
        \draw (\x cm,-0.274227) -- (\x cm,-0.274227)
            node[anchor=north,xshift=-0.15cm] {$ \xtext $};
\newcommand{\tickyi}{3/5.000000}
\newcommand{\tickyf}{1/5.000000}
    \foreach \y/\ytext in {0.01038,-0.145297,0.011734}
        \draw (2 cm+\tickyi pt,\y cm) -- (2 cm-\tickyf pt,\y cm)
```

```
        node[anchor=east] {$ \ytext $};\draw[color=blue] plot[id=x] function{(((((:
        **(0.5))*((x)**(3))-6*x+12))))*(sin(10*x)-cos(50*x))} node[above] {$Q(x)={
        *x^{3}-6*x+12)})}*(\sin{10*x}-\cos{50*x})$};\end{tikzpicture} \\
\subitem Seja \LToperfx{T}{\sqrt{F}-\cos{a}}{\sqrt{(a*x+b*x+c)}-\cos{(a)}}\; \\
\subitem e \LToperfx{Y}{\frac{P}{x}}{\frac{(\frac{(\sqrt{d}*x^{3}-6*x+12)}
{(a*x+b*x+c)})}{(x)}}\; \\
\subitem Logo \LTprevalfx{T}{x}{154}{\sqrt{((2,236068)*154+(0,648536)
*154+(2,377562))}-\cos{((2,236068))}}\; \\
\subitem e \LTprevalfx{Y}{x}{12}{\frac{(\frac{(\sqrt{(24,669141)}*12^{3}-6*12+12)}
{((2,236068)*12+(0,648536)*12+(2,377562))})}{(12)}}\; \\
\subitem e conferindo: \LTevalfx{T}{x}{154}{21,750341}\; e
\LTevalfx{Y}{x}{3}{3,87089}\;
\end{itemize}
\end{math}
\end{document}
```

3.4.4 resultado.pdf : Visualização

Explicando alguns comandos:

- \LTsetvar{<var>}{<valor-ou-expressao>}; : Ex: ($a \Leftarrow \left\langle 5 * \tan \frac{3,1415}{4} \right\rangle$) acabamos de fazer uma atribuição à variável 'a'. ($b \Leftarrow \left\langle \sqrt{a+119} \right\rangle$, $d \Leftarrow \left\langle \sqrt{\frac{a}{b}} \right\rangle$ e $ab \Leftarrow \left\langle -1 * \sin \frac{\cos a + b}{\tan 100} \right\rangle$ são outros exemplos). *As variáveis são definidas em letras minusculas podendo ser seguidas por um ou mais numeros e outras letras, também minusculas.*
- \LTgetvar{<var>}; Retorna a definição da variável, aplicada à 'a' resulta : $a \equiv (5*\tan \frac{3,1415}{4})$ em b resulta $b \equiv (\sqrt{a+119})$ e em ab resulta $ab \equiv (-1*\sin \frac{\cos a + b}{\tan 100})$
- \LTpreval{<var>};: Demostra o pré-calculo da variavel, com os valores inseridos no lugar de sua definição. No vaso de ab = $\left\langle -1 * \sin \frac{\cos(4,999768) + (11,135518)}{\tan 100} \right\rangle$

- \LTeval{<var>}; : É um comando para cálculo da variavel, temos: a = $\langle 4,999768 \rangle$ e aplicada à 'b' resulta b = $\langle 11,135518 \rangle$ e no caso de ab : $\langle 0,561698 \rangle$
- \LTgetval{<var>};: Outra alternativa ao get e ao getvar, só que retorna apenas o valor da variável. No exemplo $b = (11,135518)$ \LTeval{<var>};é parecido com getval mas com uma diferença preponderante: getval retorna o ultimo valor da variável na memória, enquanto eval retorna o valor da variavél após novo cálculo, isso é preponderante em variáveis recursivas pois a cada chamada de eval acarretará novo valor enquanto getval não. Vejamos:

  Primeiro definimos $c \Leftarrow \left\langle \sqrt{a+b} \right\rangle$ e seu valor de partida $c = (4,016875)$ pois $\left\langle \sqrt{(4,999768)+(11,135518)} \right\rangle$

  Depois definimos $b \Leftarrow \left\langle \frac{\sqrt{c}}{c} \right\rangle$ e seu novo valor $b = (0,498949)$

  Depois fazemos um primeiro eval : $\langle 2,344934 \rangle$ e $\langle 0,653032 \rangle$

  Depois fazemos um segundo eval : $\langle 2,377562 \rangle$ e $\langle 0,648536 \rangle$

  Calculemos agora ab: $\langle 0,999867 \rangle$

  Obs:Se ocasionar NaN, isso significa que o argumento de alguma raiz é negativo e não implementamos ainda o conjunto dos números complexos, uma saída para isso pode ser definir o modulo da função , como veremos mais adiante.

  Enquanto isso um getval retorna o valor anterior: $c = \langle 2,377562 \rangle$ e $b = \langle 0,648536 \rangle$

  Se uma variavel depender dela mesma: $d \Leftarrow \langle a+d+1 \rangle$ com seu primeiro valor $d = (6,669837)$ Se fizermos seguidos eval dessa variável obteremos seguidos valores: $\langle 12,669605 \rangle$ e $\langle 18,669373 \rangle$ e $\langle 24,669141 \rangle$ respectivamente.

  Definamos agora $a \Leftarrow \left\langle \sqrt{5} \right\rangle$ $x \Leftarrow \langle \sqrt{a} \rangle$ e portanto x = $(\langle 1,495349 \rangle)$.

- \LTplot{{<x>}{<expressão>}}{{<N>-<W,H>}{<x1>:<xn>#<step>}};: Plota a Função de x pela expressão dada, para N pontos de amostragem e com dominio entre x1 e xn e discretização 'step'.. Conveniente avisar que ela efetua a substituição pelos literais na memória exceto pelo argumento informado em x, como veremos a seguir.

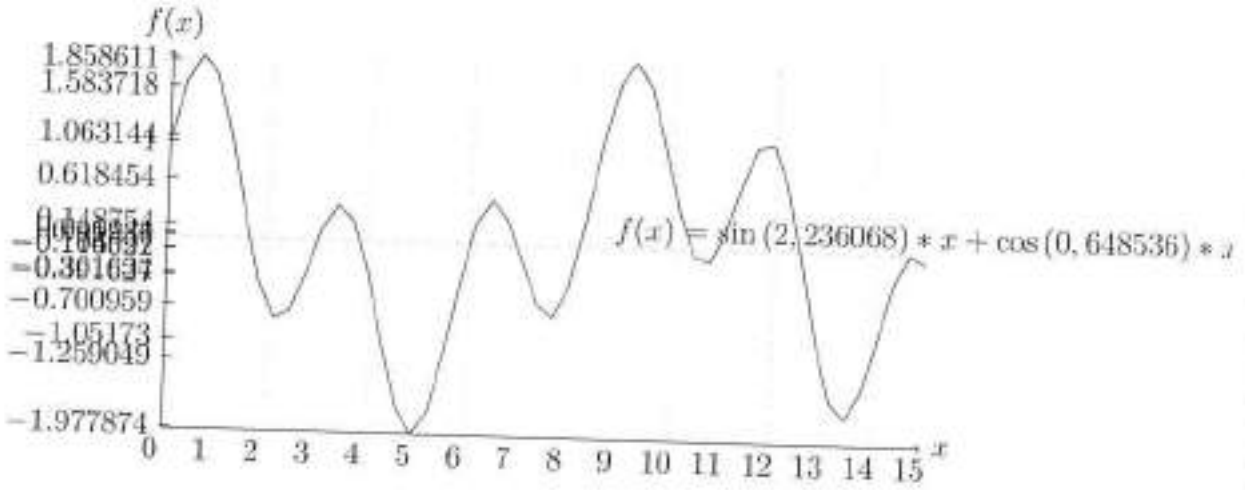
Ao fazer uso dos parametros da função plot, pode-se ajustar ,melhor o tamanho e o eixo de acordo com a curva para uma melhor documentação como segue na sequencia

de exemplos a seguir, nos quais apresento algumas possibilidades e limitações de desempenho.

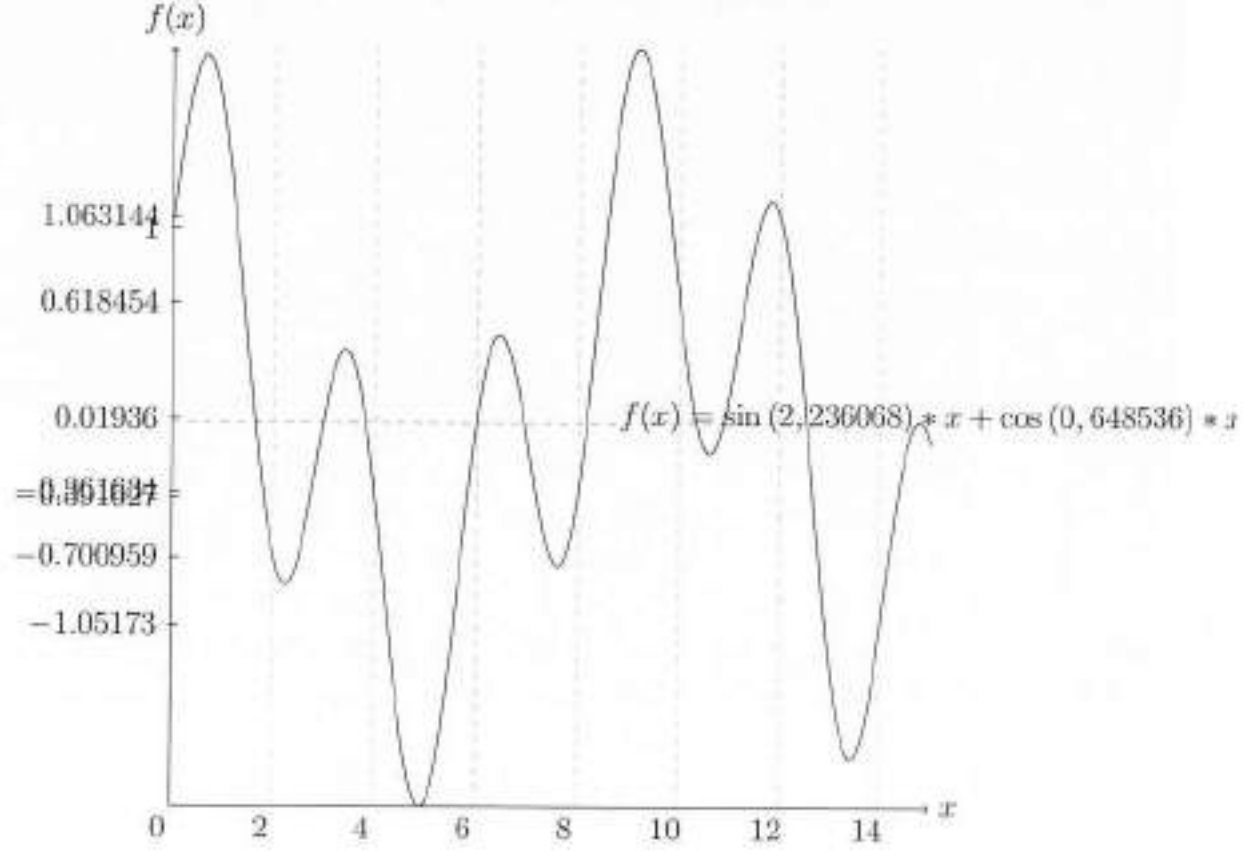
É interessante ressaltar a natureza interativa do processo de escolha do tamanho e dominio adequados à ilustração da função. O algoritmo atual , como é explicado na seção sobre o modo de funcionamento dos subVIs componentes, trabalha com deteção de máximo e mínimo da função para composição do eixo y, usando para isso um passo de amostragem inicialmente configurado para 0.01, o que durante o funcionamento no computador pessoal do aluno, não causou demora na execução mas restringe o 'zoom-in' de plotagem a uma escala de 100. Foram efetuados testes com passos de 0.001 causando sensivel lentidão durante a execução.

A seguir uma série de funções e seus respectivos zooms em áreas de interesse para caracterizar o processo interativo de escolha dos parâmetros mais adequados para representação (domínio, largura, comprimento e discretização da grade e tomada de valores).

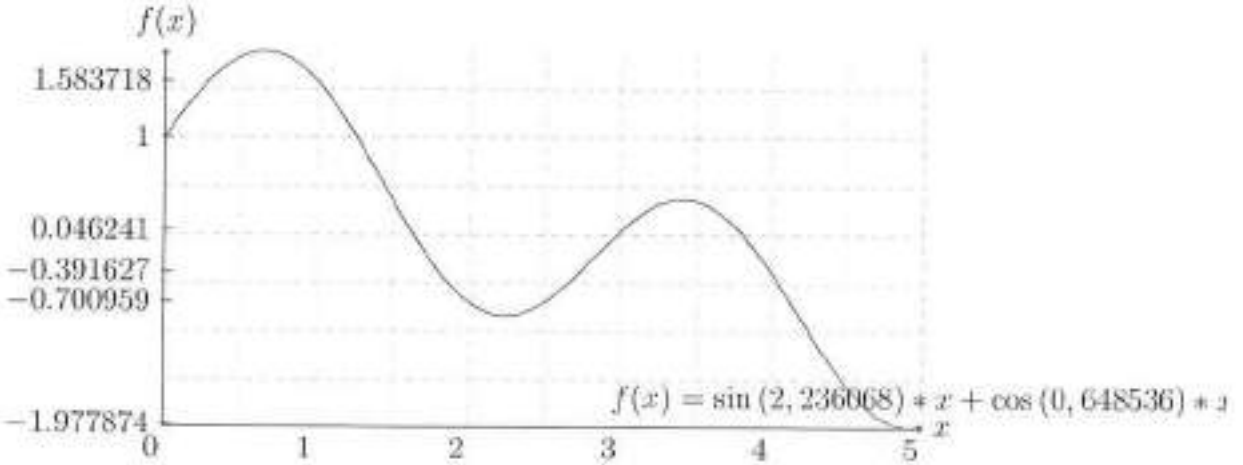
Calculando b= ($\langle 0,648536 \rangle$) , a = ($\langle 2,236068 \rangle$) e x = ($\langle 1,495349 \rangle$). e para 50 amostras :

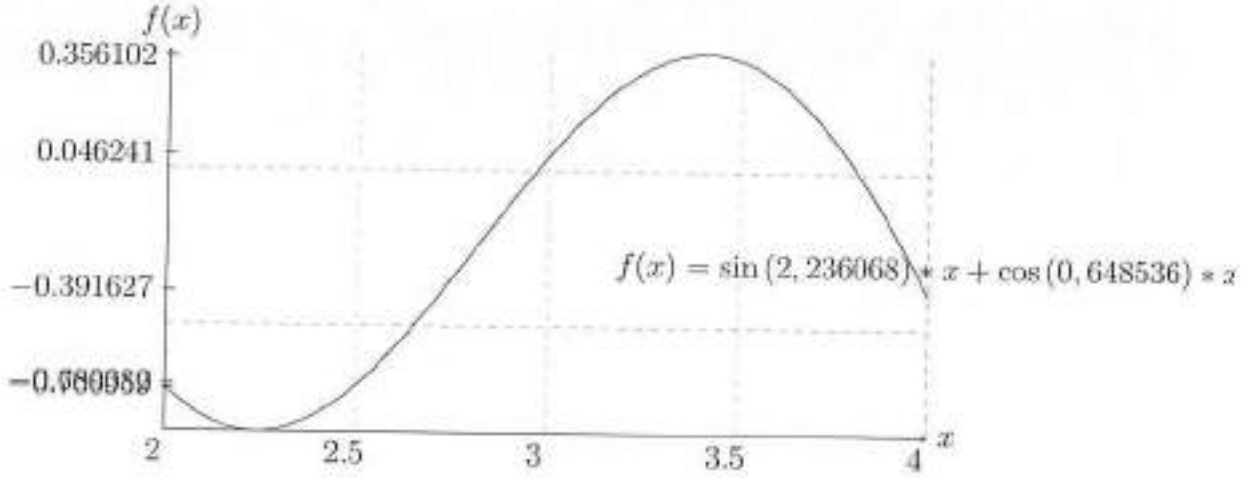


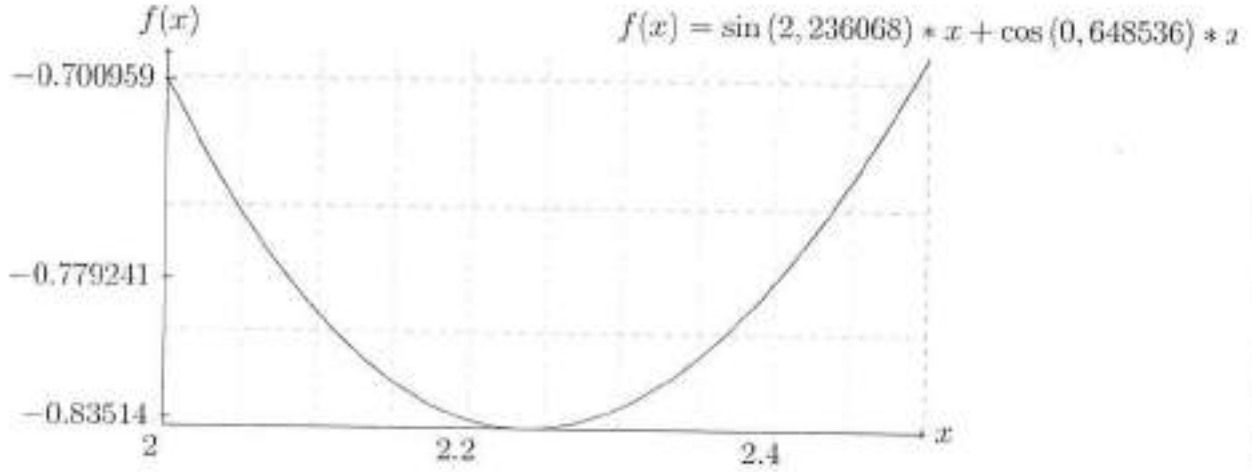
E aumentando o número de amostras para 500 e aumentando a altura da imagem:

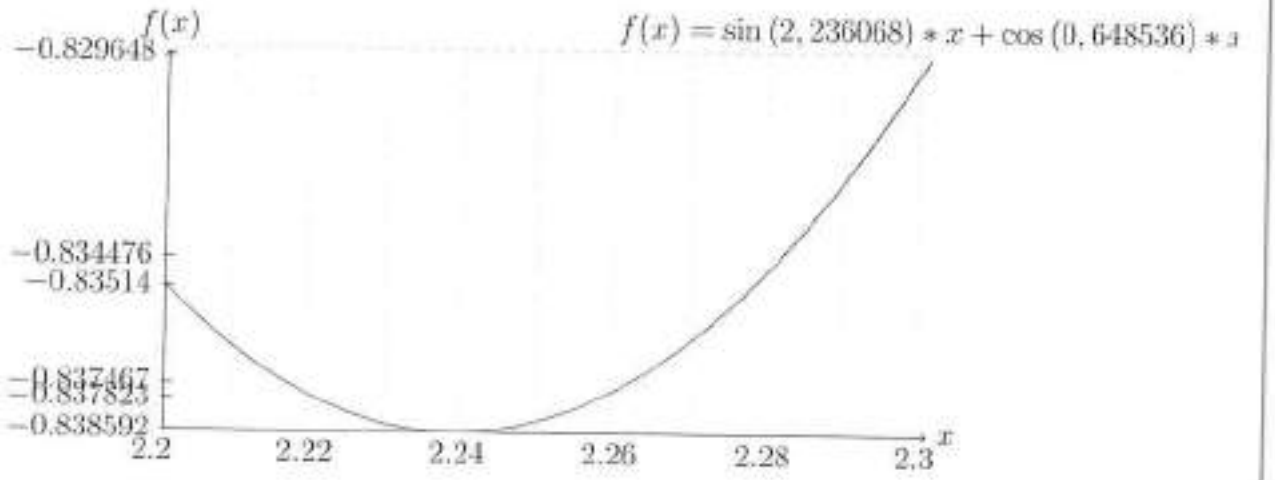


Restrição do domínio :





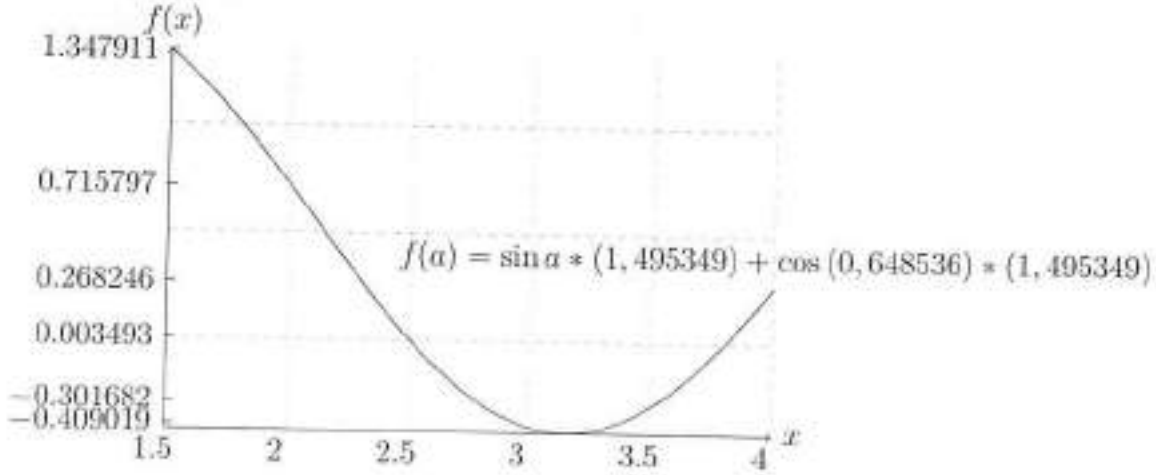


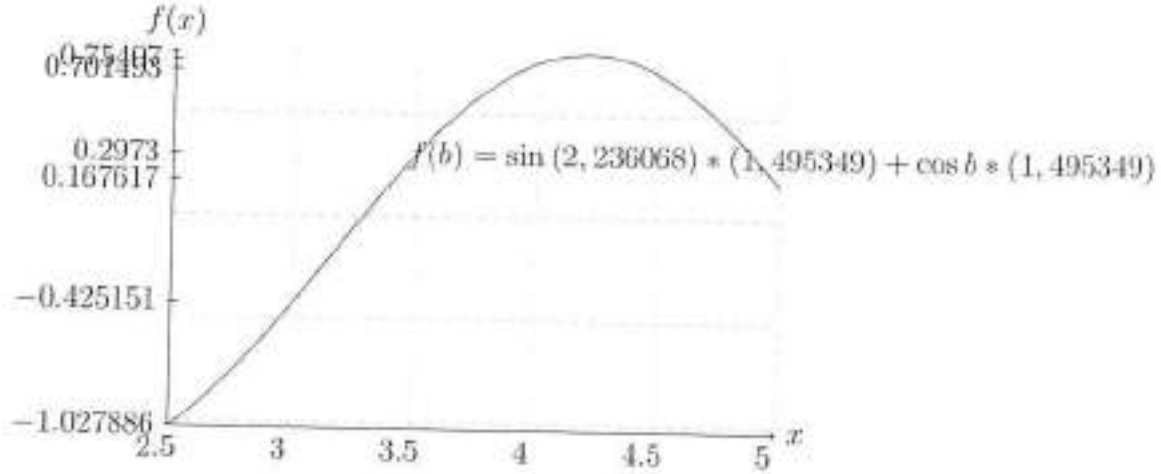


Como observa-se, a escolha adequada do passo de amostragem permite foco na regiião

de interesse sobre o domínio, encontra-se anexo a este roteiro de teste uma sequencia de tentativas na obtenção da melhor visualização das funções como alguns dos resultados obtidos pelo comando \LTplot;mas ainda há uma série de particulariedades a serem descobertas de acordo com o uso e exploração das estruturas.

Também interessante observar que se chamamos o mesmo comando plot para a mesma função, mas passando como referência outra variável independente, o comando faz a adequação dos parâmetros :





- `\LTsetfx{<F>}{<formula>};`: Cria a função F(x) atribuindo a ela a formula:

  Seja $F(x) \equiv a * x + b * x + c$

  e $G(x) \equiv \sqrt{d} * x^3 - 6 * x + 12$

  e $Z(x) \equiv \sin 10 * x - \cos 50 * x$

  Sejam $a = (2,236068)$, $b = (0,648536)$, $c = (2,377562)$, e $d = (24,669141)$

- `\LTgetfx{<F>}{<x>};`: Exibe a definição de F(x)

  Vejamos : $F(x) \equiv a * x + b * x + c$

  $G(x) \equiv \sqrt{d} * x^3 - 6 * x + 12$

  $G(d) \equiv \sqrt{d} * x^3 - 6 * x + 12$

  $Z(x) \equiv \sin 10 * x - \cos 50 * x$

- `\LTprevalfx{<F>}{<x_0>};`: Exibe o preparo do cálculo de F(x_0)

  Vejamos : $F(x \to 9) = (2,236068) * 9 + (0,648536) * 9 + (2,377562)$

  $G(x \to 1) = \sqrt{(24,669141)} * 1^3 - 6 * 1 + 12$

  $G(d \to 4) = \sqrt{4} * (1,495349)^3 - 6 * (1,495349) + 12$

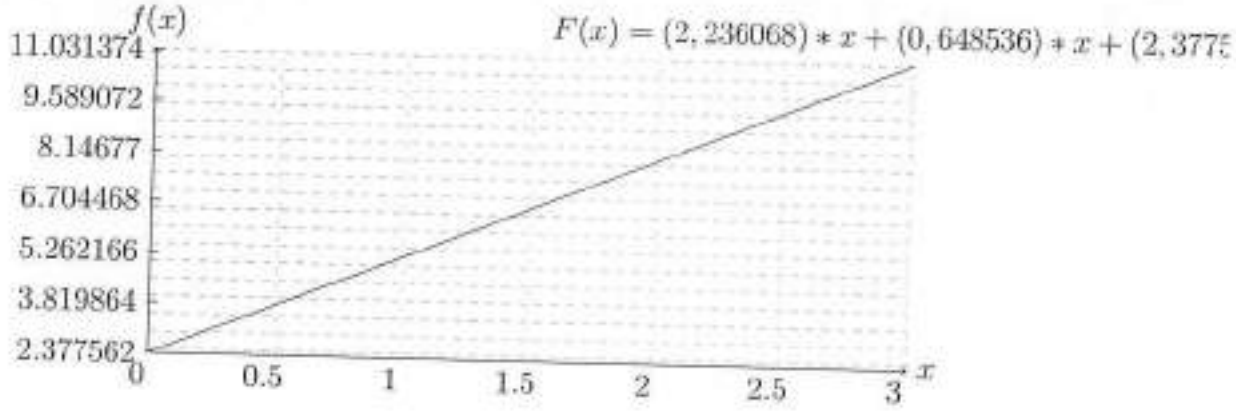
  $Z(x \to 1) = \sin 10 * 1 - \cos 50 * 1$

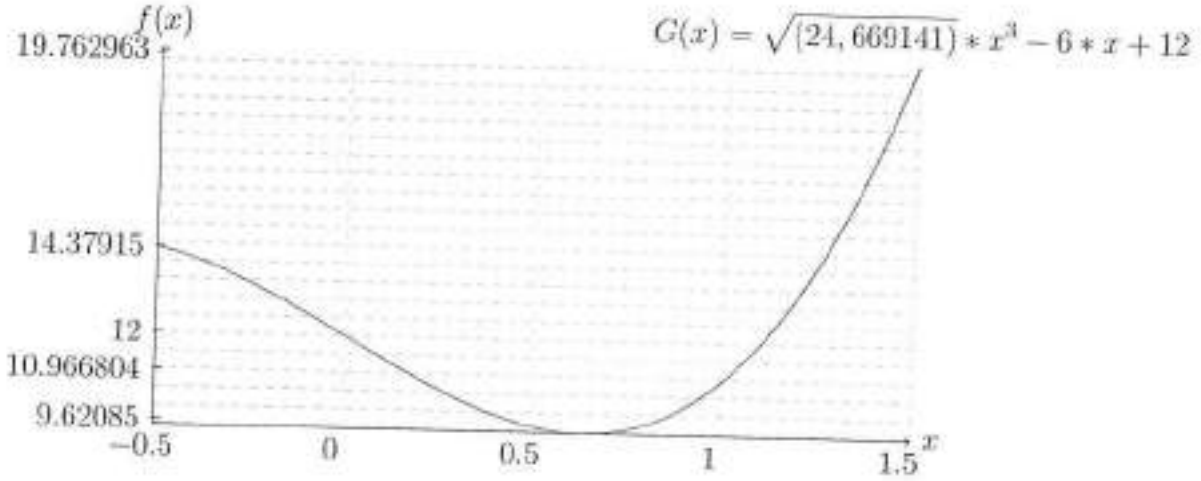
- `\LTevalfx{<F>}{<x_0>};`: Calcula F(x_0)

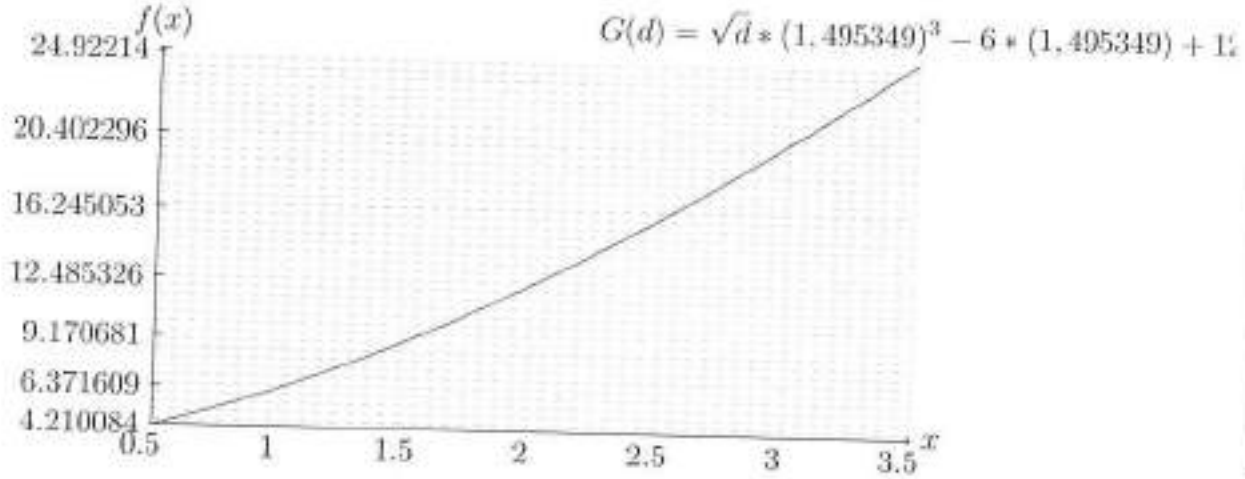
Logo: $F(x \to 9) = 28,338998$, $G(x \to 1) = 10,966804$ e $Z(x \to 1) = -1,508987$

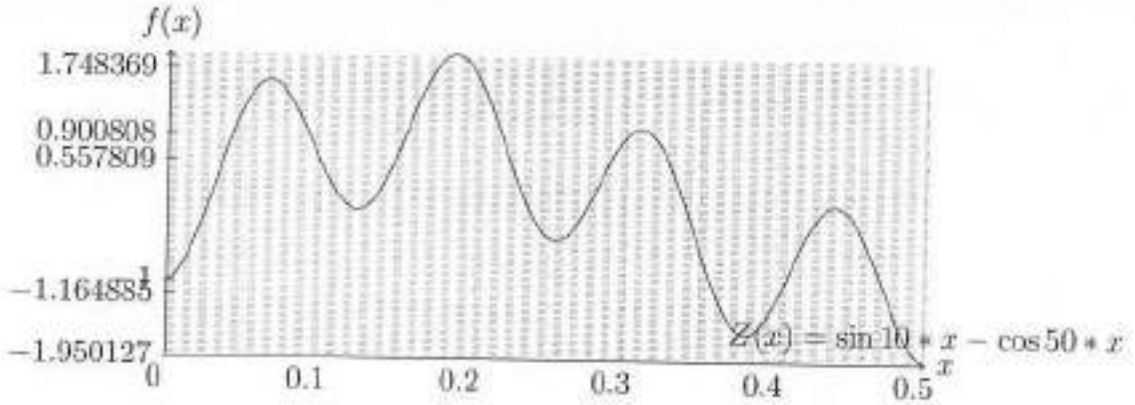
Mas se houver um 'x' na memoria do sistema ($x = (1,495349)$) pode nos interessar calcular $G(d \to 4) = 3,027906$ por exemplo, trocando a variável independente.

- `\LTplotfx{{<F>}{<x>}}{{<N>-<W,H>}{<x1>:<xn>#<step>}};:`
  Plota a Função $F(x)$ para os parâmetros dados:









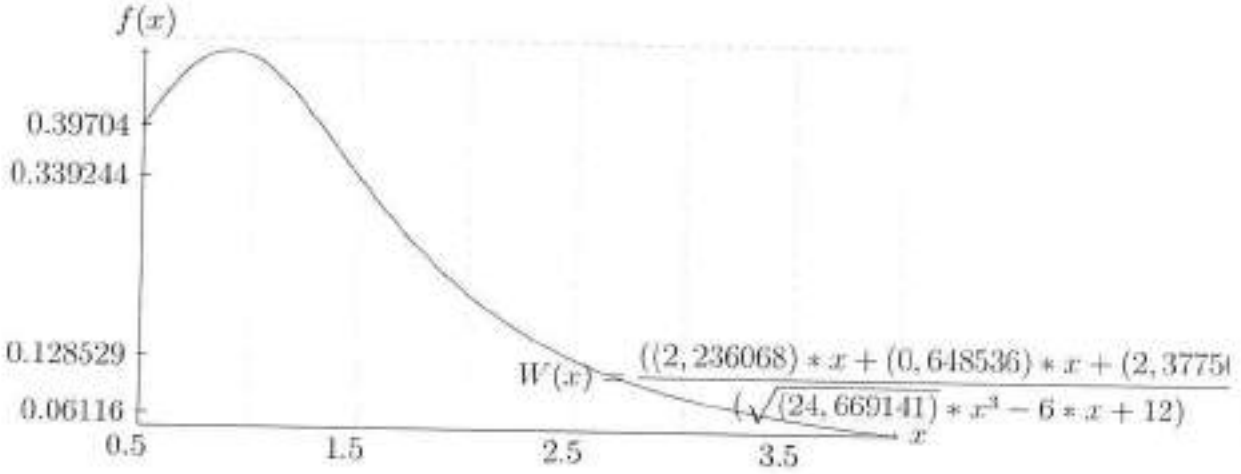
- \LToperfx{<F>}{<F1-oper-F2-oper-F3>};realiza a operação entre as duas ou mais funções da direita e atribui à primeira.

Seja $W(x) \equiv \dfrac{F}{G} \equiv \dfrac{(a*x+b*x+c)}{(\sqrt{d}*x^3-6*x+12)}$

Logo $W(x \to 5) = \dfrac{((2,236068)*5+(0,648536)*5+(2,377562))}{(\sqrt{(24,669141)}*5^3-6*5+12)}$

e $W(x \to 5) = 0,027869$

Vamos experimentar plotar a nova função :

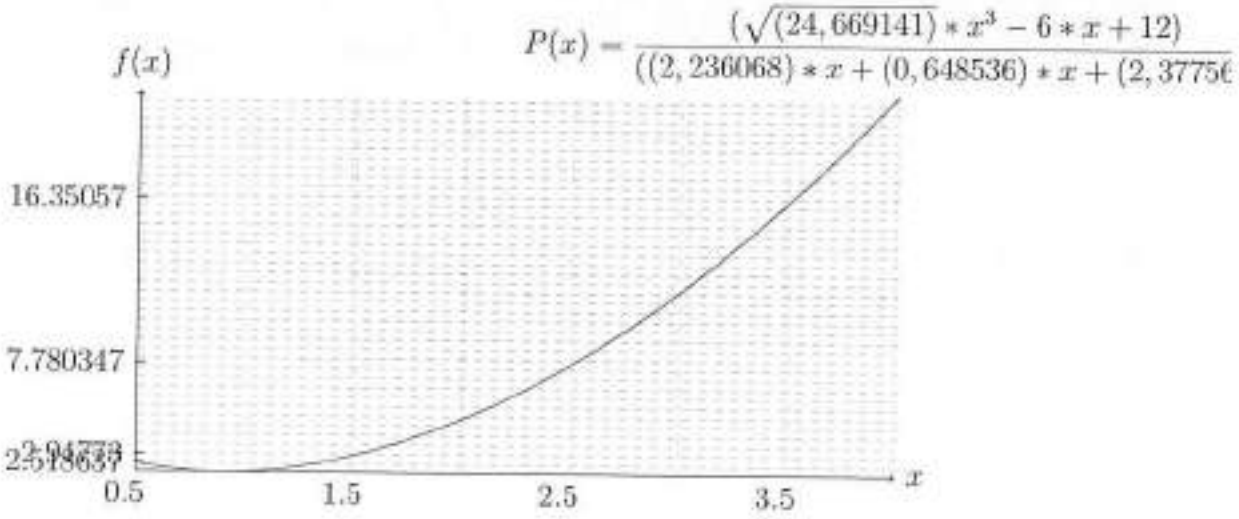


Vamos ensaiar sua inversa: $P(x) \equiv \dfrac{G}{F} \equiv \dfrac{(\sqrt{d} * x^3 - 6 * x + 12)}{(a * x + b * x + c)}$

Logo $P(x \to 5) = \dfrac{(\sqrt{(24,669141)} * 5^3 - 6 * 5 + 12)}{((2,236068) * 5 + (0,648536) * 5 + (2,377562))}$

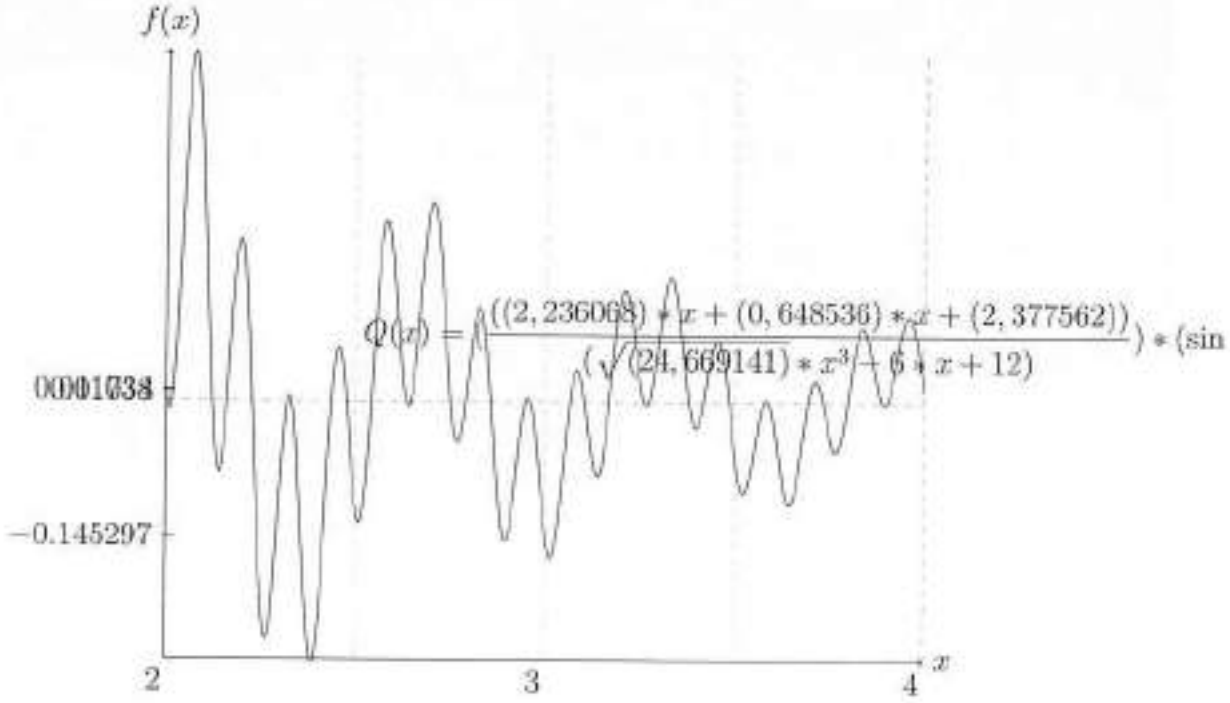
e $P(x \to 5) = 35,882715$

Vamos experimentar plotar a nova função :



Seja então $Q(x) \equiv W * Z \equiv (\frac{(a*x+b*x+c)}{(\sqrt{d}*x^3-6*x+12)}) * (\sin 10 * x - \cos 50 * x)$

e $Q(x \to 12) = (\frac{((2,236068)*12+(0,648536)*12+(2,377562))}{(\sqrt{(24,669141)}*12^3-6*12+12)}) * (\sin 10 * 12 - \cos 50 * 12) = Q(x \to 12) = 0,006856$



Seja $T(x) \equiv \sqrt{F} - \cos a \equiv \sqrt{(a*x+b*x+c)} - \cos(a)$

e $Y(x) \equiv \dfrac{P}{x} \equiv \dfrac{(\frac{(\sqrt{a}*x^3-6*x+12)}{(a*x+b*x+c)})}{(x)}$

Logo $T(x \to 154) = \sqrt{((2,236068)*154+(0,648536)*154+(2,377562))} - \cos((2,23606$

e $Y(x \to 12) = \dfrac{(\frac{(\sqrt{(24,669141)}*12^3-6*12+12)}{((2,236068)*12+(0,648536)*12+(2,377562))})}{(12)}$

e conferindo: $T(x \to 154) = 21,750341$ e $Y(x \to 3) = 3,87089$

3.5 Objetivos

Alguns dos resultados atingidos envolvem a possibilidade de criação de roteiros que gerem relatórios em tempo de execução otimizando sua execução, permitir que o aluno foque-se nos aspectos analíticos e críticos do exercício, possibilidade de refazer ensaios com

facilidade e facilidade para comparar diversas condições de contorno com maior aproveitamento sobre o experimento completo.

Diversificação dos tipos de ensaios devido a facilidade de disponibilizar bases de exercícios definidos nesse conjunto de documento-programa devido ao fato que ambas as estruturas são de padrão aberto e consolidado e o projeto estimula o compartilhamento de exercícios e experimentos científicos entre pares acadêmicos e outras instituições por meio da adoção dessa tecnologia.

Um indicador de desempenho interessante é realizar experiências tradicionais dos cursos de Eletricidade e Eletrônica do curso sob essa tecnologia experimentalmente e comparar os efeitos quanto ao aproveitamento do aluno em relação ao andamento dos experimentos, facilidade de manuseio das estruturas e capacidade de percepção do aluno aos aspectos mais qualitativos e analíticos da experiência abstraindo em compensação de certas rotinas operacionais como operar calculadoras, traçar gráficos ou capturar imagens.

Aumentar o nível de interatividade nos procedimentos laboratoriais pelo oferecimento de uma tecnologia que permite roteiros mais complexos e execução condicionada por etapas, oferecendo uma experiência mais imersiva no processo de investigação e processamento do conhecimento.

Como objetivos locais conseguiu-se praticar os testes de conceito da manipulação de variáveis e valores de referência do LaTeX para o LabVIEW, envolvendo crescente complexidade de construtores durante o processo de definição da 'tabela de instruções' do processador LabTex, ou seja, inicialmente manuseando funções usando o parse próprio do LabView e conforme o grau de complexidade do processador desenvolva-se, criar instruções para utilização de outros construtores e operadores.

Continuar desenvolvend este projeto para homologação e disponibilização os módulos gerados nesse projeto de formatura, tanto o pacote de Biblioteca em LaTeX para as bases do CTAN (Comprechensive TeX Arquive Network), como o pacote de Vis correspondente ao processador propriamente dito em LabVIEW, contribuindo com a OpenG (LAVA Code Repository - LAVAcr), como passo visando ao objetivo geral proposto sobre a colaborção e compartilhamento de roteiros laboratoriais.

4 *Conclusões e trabalhos futuros*

Neste capítulo são apresentadas as conclusões e possibilidades de desenvolvimentos futuros do projeto.

4.1 Conclusão

Cabe aqui ressaltar que o Menu de aplicações do LabVIEW é vastíssimo, só na Paleta Matemática há inúmeras possibilidades, desde as elementares que envolvem funções trigonométricas, exponenciais e de outros 10 tipos, e outras subcategorias como álgebra linear, integração e diferenciação, estatística, equações diferenciais, polinômios, scripts e fórmulas, que por sua vez oferecem funcionalidades de cálculo, zeros de função e calculadores de fórmulas (e só nessa subcategoria são 20 estruturas), portanto, o nosso objetivo durante PSI2594 foi englobar uma quantidade possível dessas estruturas que permitisse a aplicação dos testes de conceito, é ciente por parte do aluno que a paleta atual de funcionalidades do sistema possui poucos recursos, mas é um projeto que pode ser implementado e cujas funcionalidades são desenvolvidas umas sobre as outras e por isso o projeto foi modelado e construído na visão componentizada em camadas, para que fosse atingida escalabilidade suficiente para que o projeto inicie-se funcional desde essas primeiras estruturas de fórmulas implementadas e possa crescer em beta eterno.

Essa arquitetura principal permitir que o conjunto dos métodos de tratamento e execução (parse/execute) possam ser expandidos e adicionados ao sistema de modo contínuo assim que desenvolvidos e disponibilizados pela comunidade de usuários, do mesmo modo que ocorre com os pacotes LaTeX, isso foi aplicando ao trabalho.

4.2 O Futuro do projeto

Como dito na conclusão parcial 4.1, a arquitetura componentizada foi o modo encontrado para realizar o trabalho incremental por camadas de complexidade sem perder a coesão estrutural.

Este é um projeto que admite expansões para instruções de todo tipo em documentos ativos, outras ideias similares incluem executar automaticamente uma partitura feita em latex, em som sintetizado pelo LabVIEW ou outro software.

Na verdade o LabVIEW é uma ponte de processamento escolhida porque no nosso contexto da engenharia e da bancada é muito presente e podemos implementar o projeto

na sala de aula e isso é gancho para contribuição à escola. Mas poderia ser outra ponte, outro processador, por exemplo um hardware similar a um e-book reader com usb para comunicação com um disposiivo de aquisição/controle de sinais e processamento para executar a inteligência instrucional na forma LaTeX. Essa é uma alternativa interessante, e sob esse ponto de vista o projeto criado num instrumento virtual ´pode ser protótipo de uma solução integrada.

O fato estrutural que destaca-se portanto é a ontologia do processo, utilizar uma estrutura sintáxica lógica de tipografia que só era usada até então para exibir os glifos dessa estrutura de maneira adequada e agora esses glifos serão reconhecidos como comandos e operadores dentro de um contexto que é definível e permite mesclar num mesmo arquivo, documento e programa.

O corpo do documento é aquilo que observamos, a estrutura visual pdf do ente, é o instantâneo; As suas instruções de execução são sua 'alma', o rotor do ente, é o que movimenta a execução de processos que podem envolver inclusive interatividade em critérios de escolha do tipo if/else ou chaves de seleção (switches), tornando o documento não apenas ativo, mas com distintas possibilidades de execução de acordo com as variáveis e tomadas de decisão no momento de execução.

Uma possibilidade adicional deverá ser a inclusão futura de um modo 'command echo on/off', para que documentos ativos, ao serem executados gerem documentos que possam herdar também os comandos, ou seja, se a chave de propagação de comandos do LabTeX estiver desativada, a execução do documento ativo efetua apenas a substituição das instruções pelos resultados, e em caso contrario ela mantém as instruções e adiciona em agregação os resultados em seqüência.

Se isso for feito então este ente guardará um histórico das execuções dentro de si, dessa forma ao invés de criar vários documentos relatórios conseqüentes de cada execução, mantemos todos atrelados e passamos a ter dois documentos ambos ativos, com uma única diferença basicamente: um é virgem de históricos, ou seja, o documento técnico ativo sem execuções, apenas instruções; e no outro o mesmo documento possui as instruções mas com o histórico das execuções anteriores agregado.

Isso abre possibilidade para efetuar comparações entre as execuções do experimento

uma vez que mantém em si os dados de diferentes execuções do roteiro. Isso torna-se especialmente útil no campo laboratorial e acredito que também para a engenharia de controle de qualidade, uma vez que rotinas de testes podem ser encapsuladas em um único ente que é documento, programação, histórico e analise comparativa ao mesmo tempo.

Uma outra possibilidade futura mais imediata com a adoção da nova versão do LabVIEW (2009) que permite recursividade, será adaptar os blocos de tratamento das tags para permitir tags compostas (tag dentro de tag), como é usualmente aceito no LaTeX, mas não ainda no LabTeX.

ANEXO A – Anexos

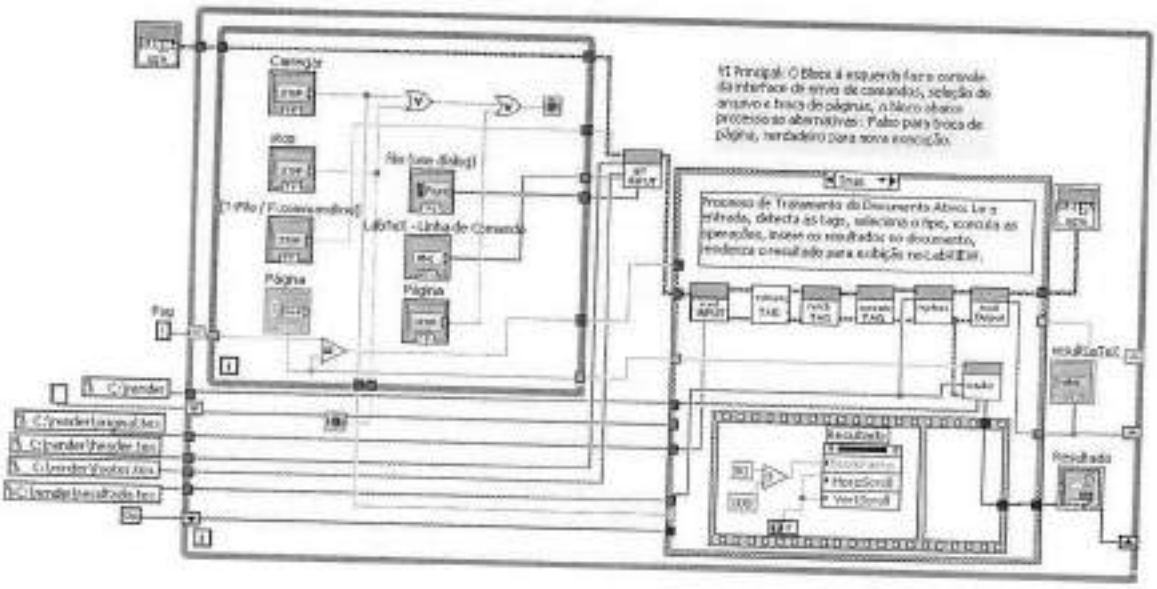


Figura 7: Anexo 01: VI Principal: Opção Carregar 'Verdadeiro'.

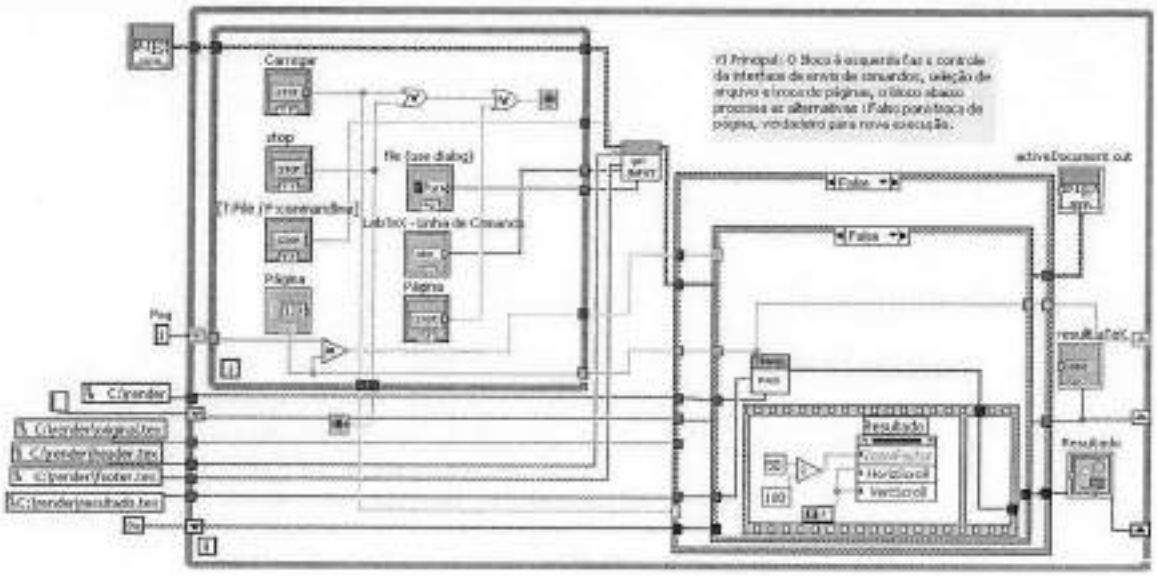


Figura 8: Anexo 01.b : VI Principal : Opção 'Carregar' Falsa.

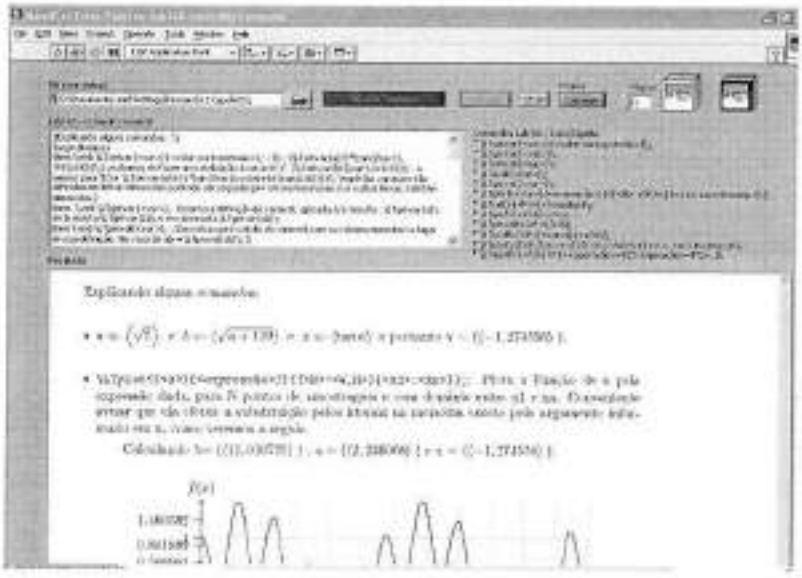


Figura 9: Anexo 01.c: Main.vi : FrontPanel

Page 1

MainVI.vi
C:\Documents and Settings\Fernando J Capeletto\Desktop\Poli\PROJETO\02_Sources\MainVI.vi
Last modified on 29/11/2009 at 15:14
Printed on 29/11/2009 at 16:16

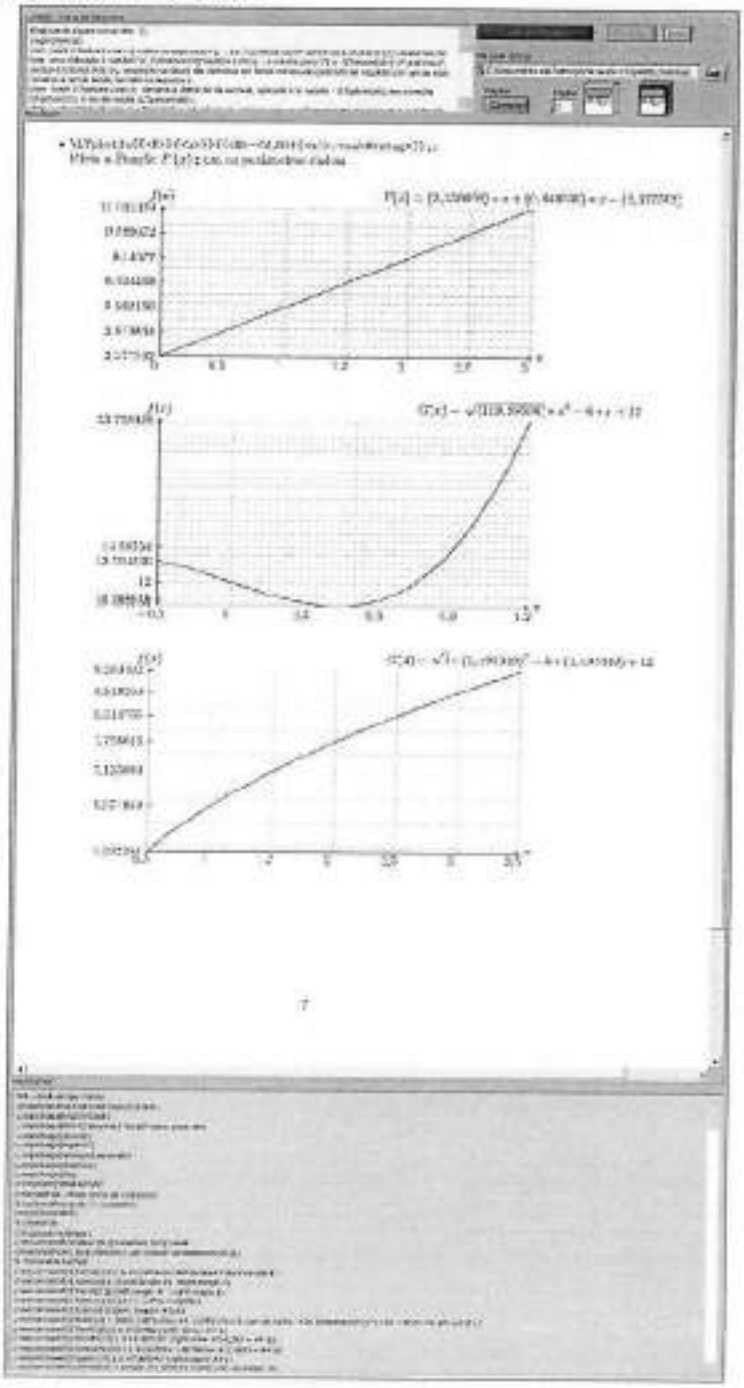


Figura 10: Anexo 01.d - Main.vi - Front Panel Completo.

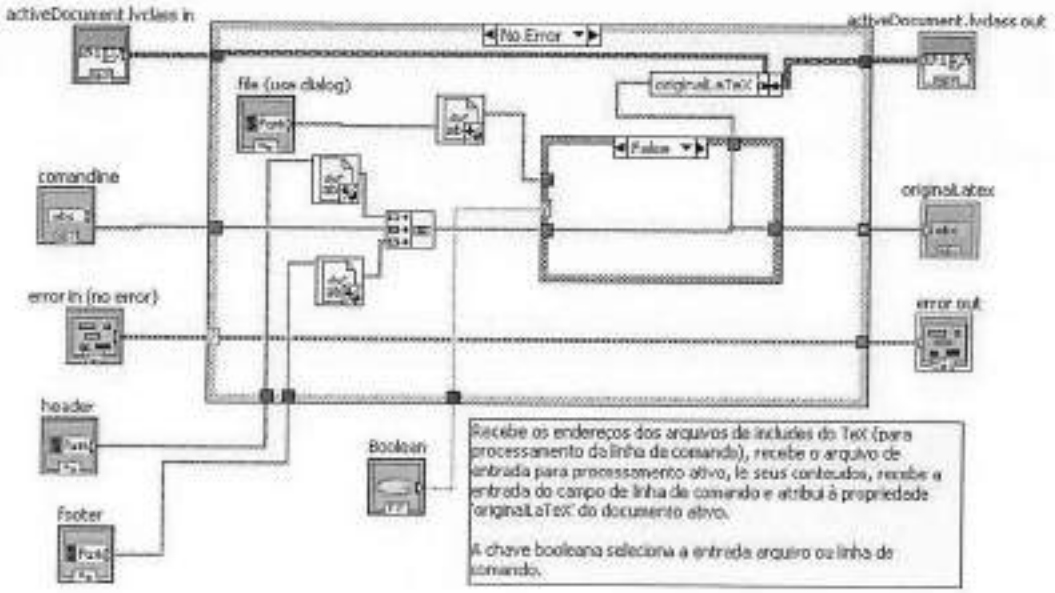


Figura 11: Anexo 02 LTgetInput.vi

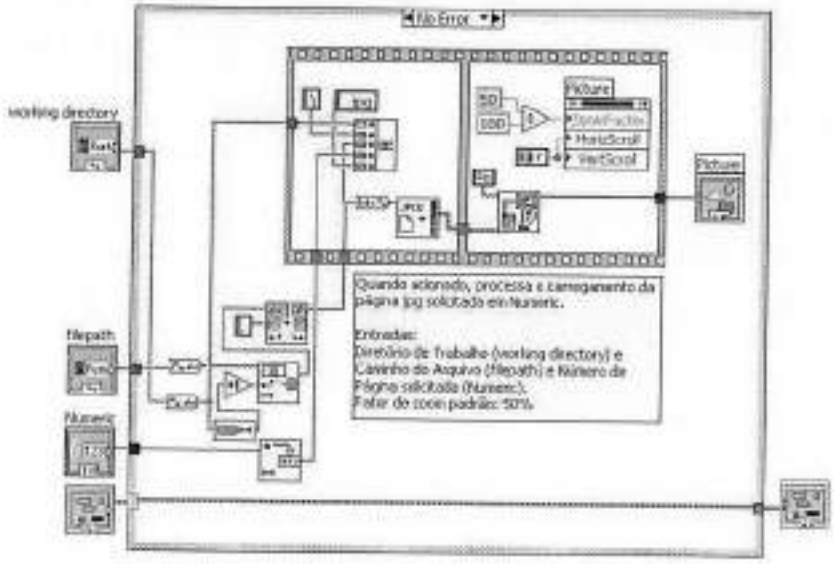


Figura 12: Anexo 03: ChangePag.vi

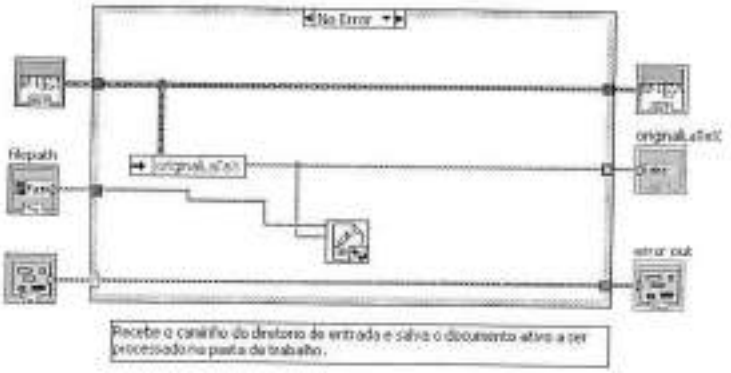


Figura 13: Anexo 04: LTreadInput: Recebe o caminho do diretorio de entrada e salva o documento ativo a ser processado na pasta da trabalho.

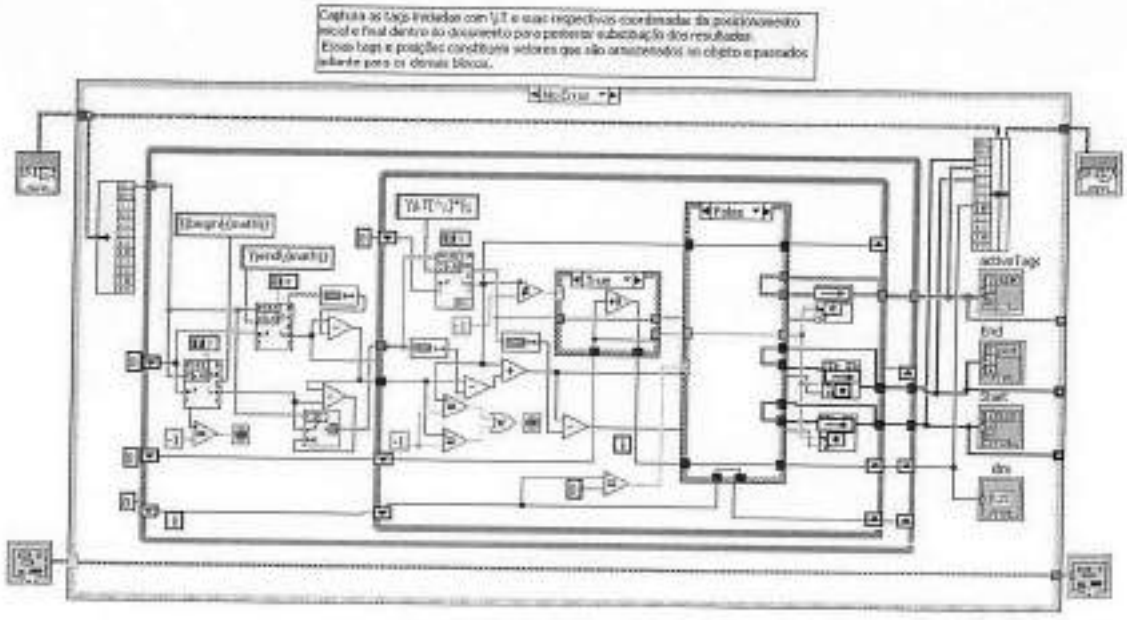


Figura 14: Anexo 05: LTretrieveTag.vi.

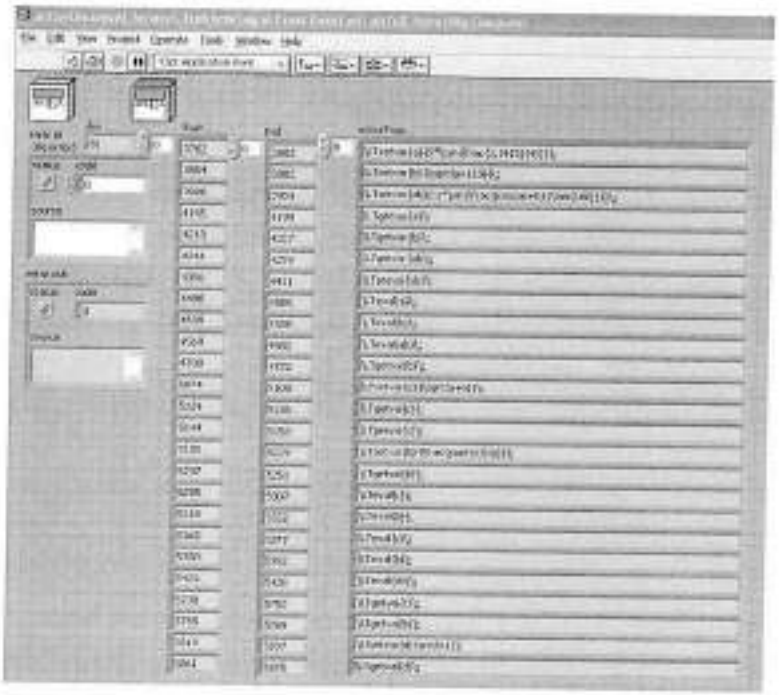


Figura 15: Anexo 05.b: LTretrieveTag.vi - Front Panel.

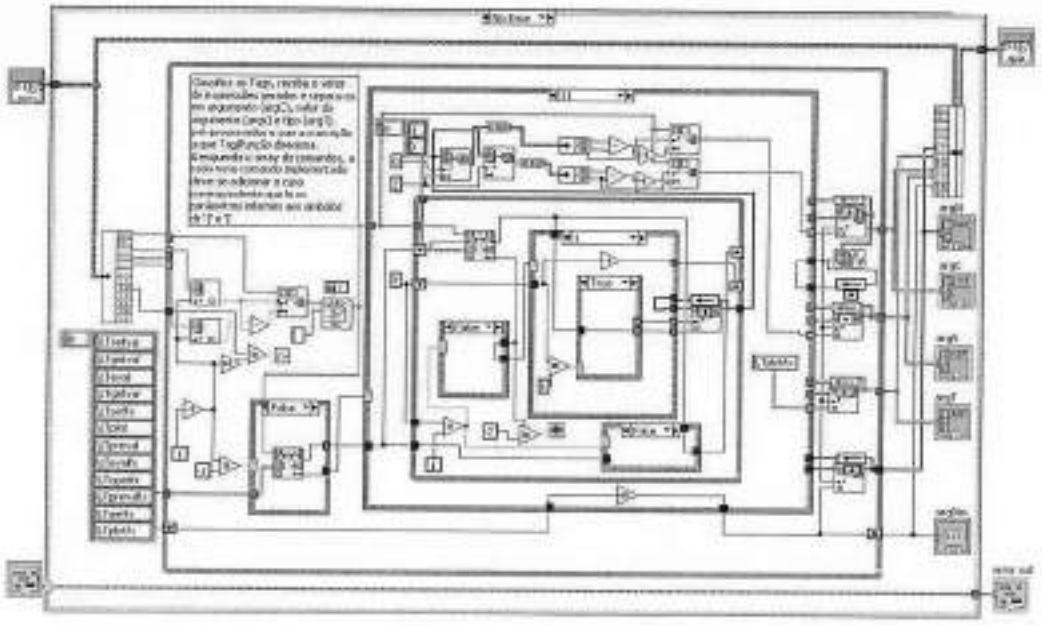


Figura 16: Anexo 06: LTfetchTag.vi

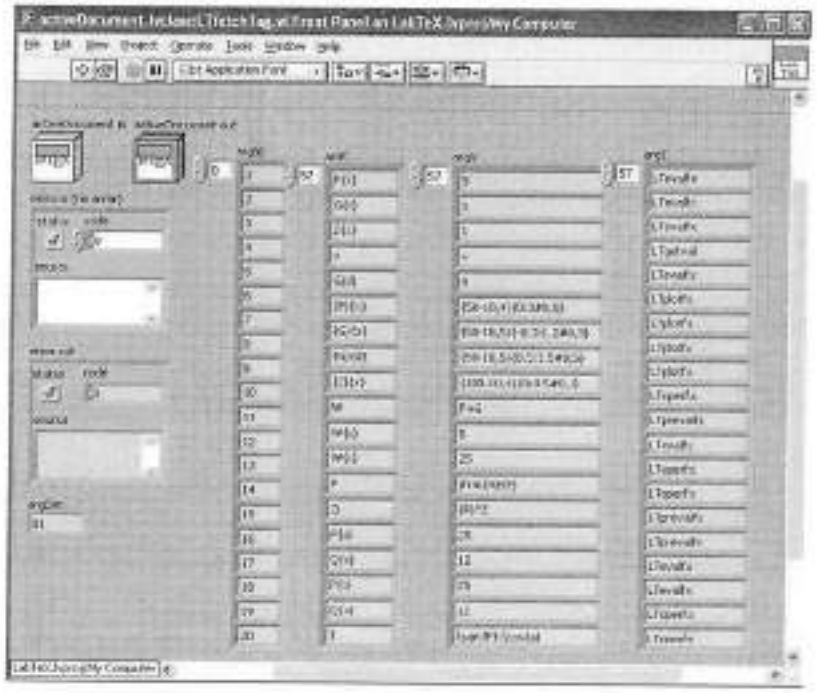


Figura 17: Anexo 06.b: LTfetchTag.vi - Front Panel

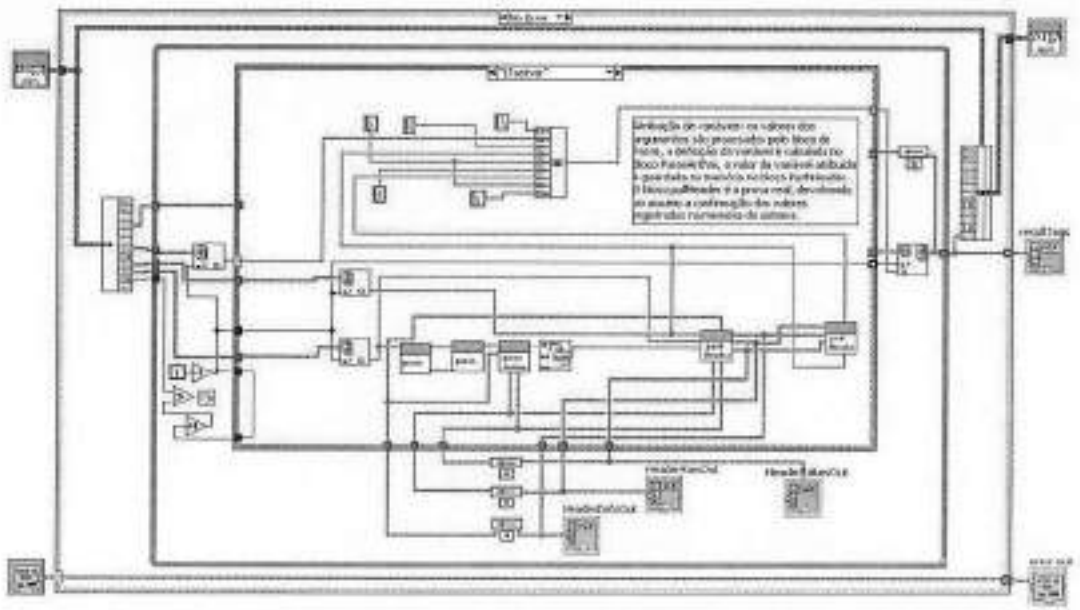


Figura 18: Anexo 07: LTexecuteTag.vi - Comando: LTsetvar

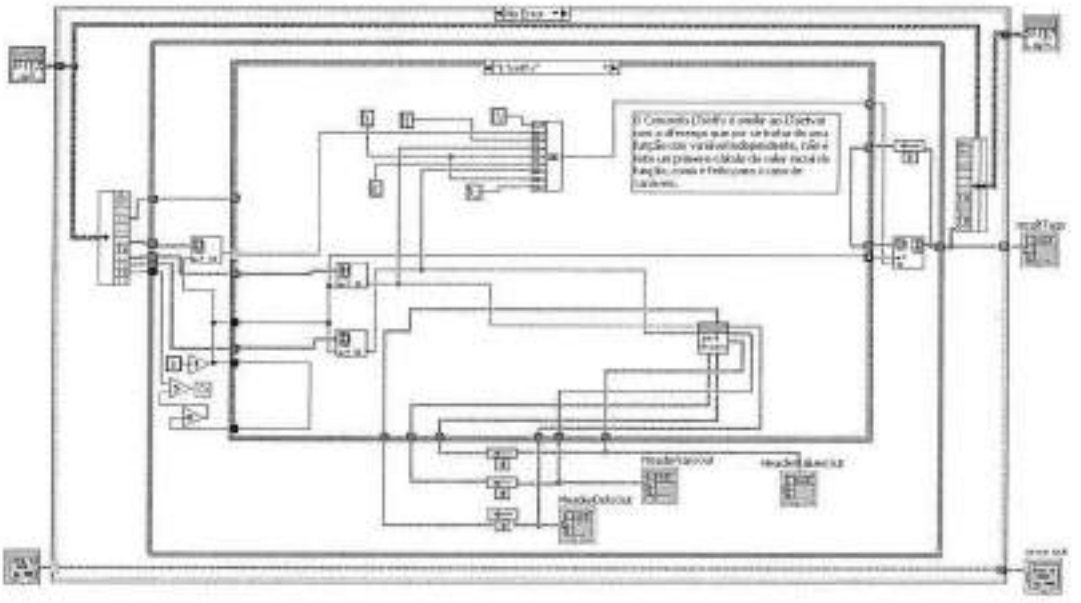


Figura 19: Anexo 7.b : LTexecuteTag.vi - Comando: LTsetfx

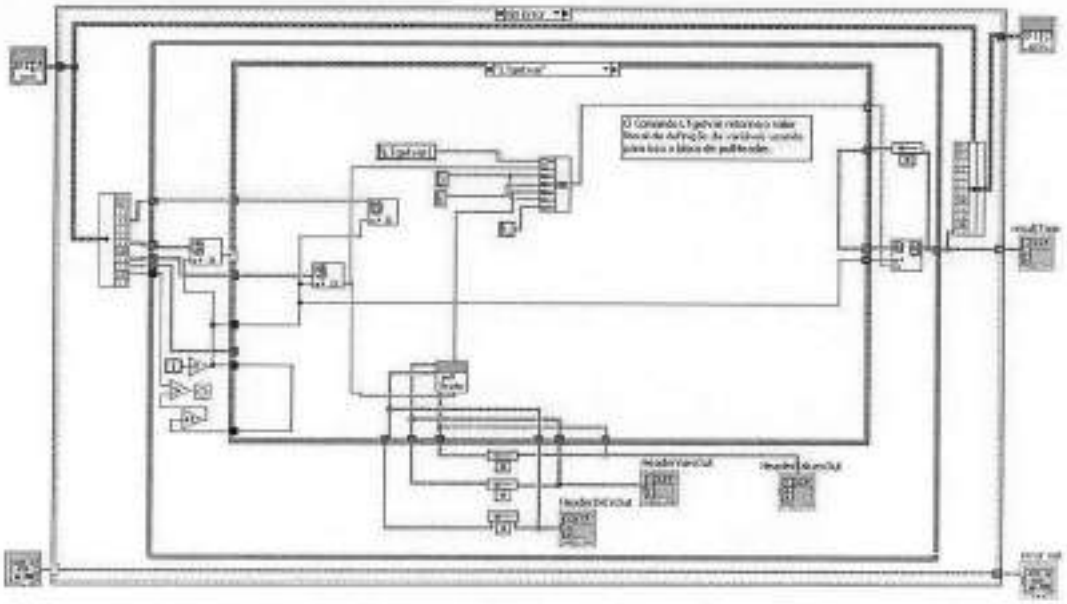


Figura 20: Anexo 7.c : LTexecuteTag.vi - Comando: LTgetvar

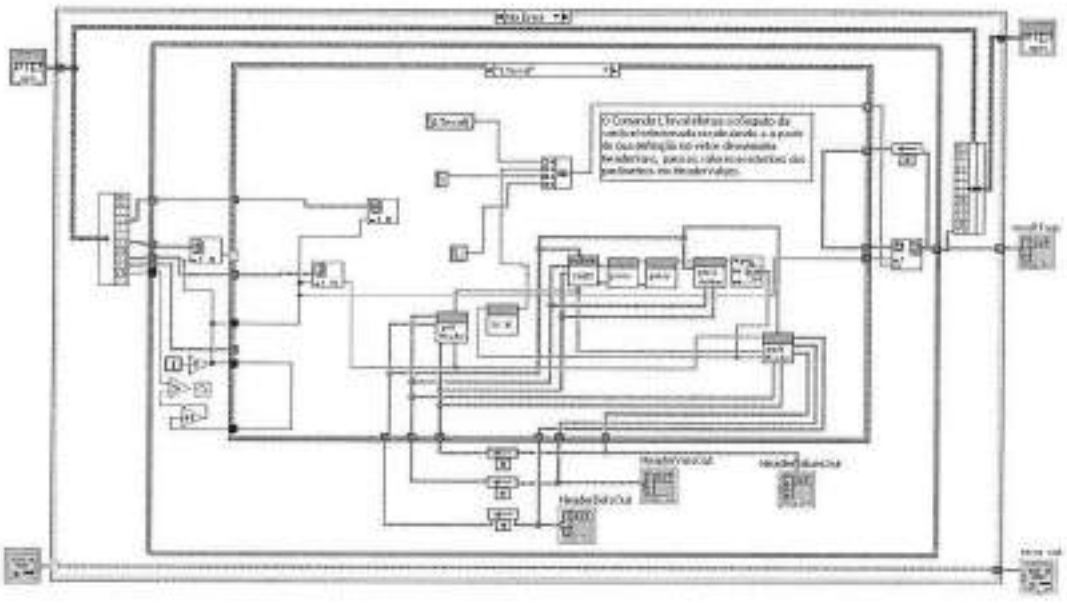


Figura 21: Anexo 7.d : LTexecuteTag.vi - Comando: LTeval

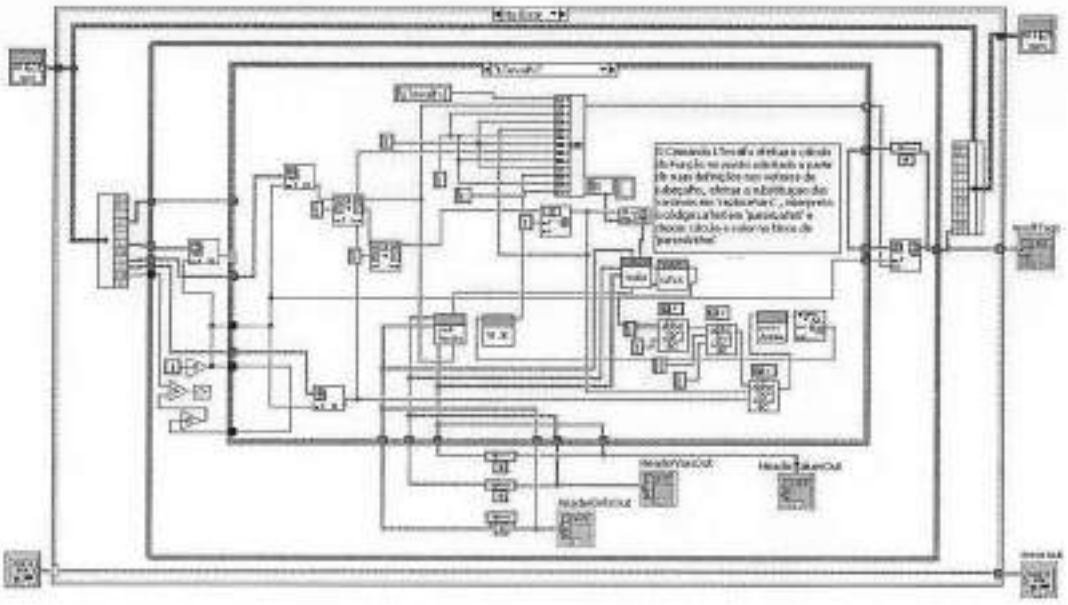


Figura 22: Anexo 7.e : LTexecuteTag.vi - Comando: LTevalfx

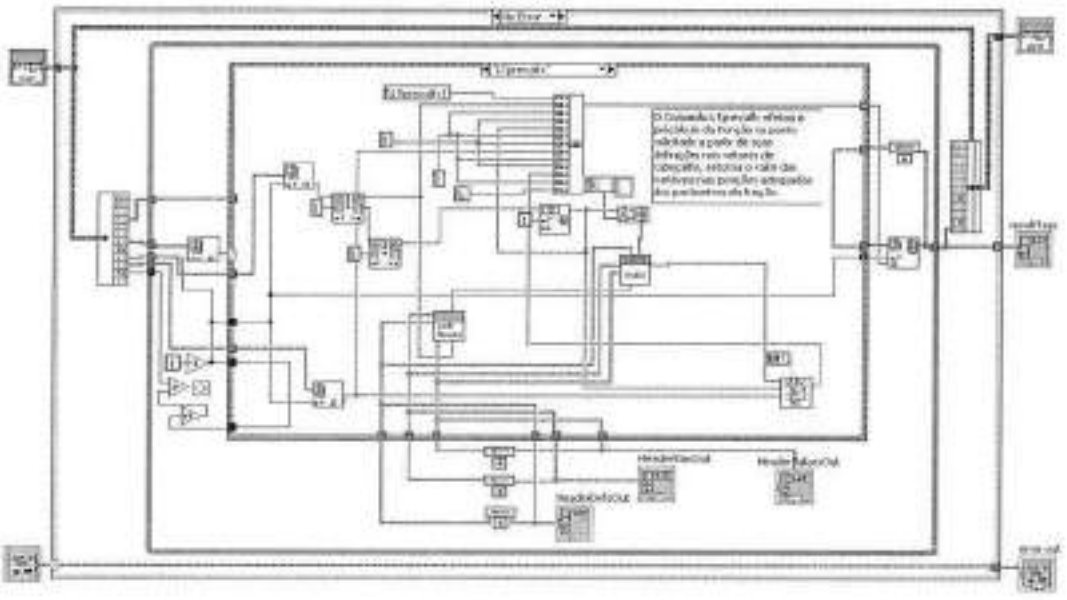


Figura 23: Anexo 7.f : LTexecuteTag.vi - Comando: LTprevalfx

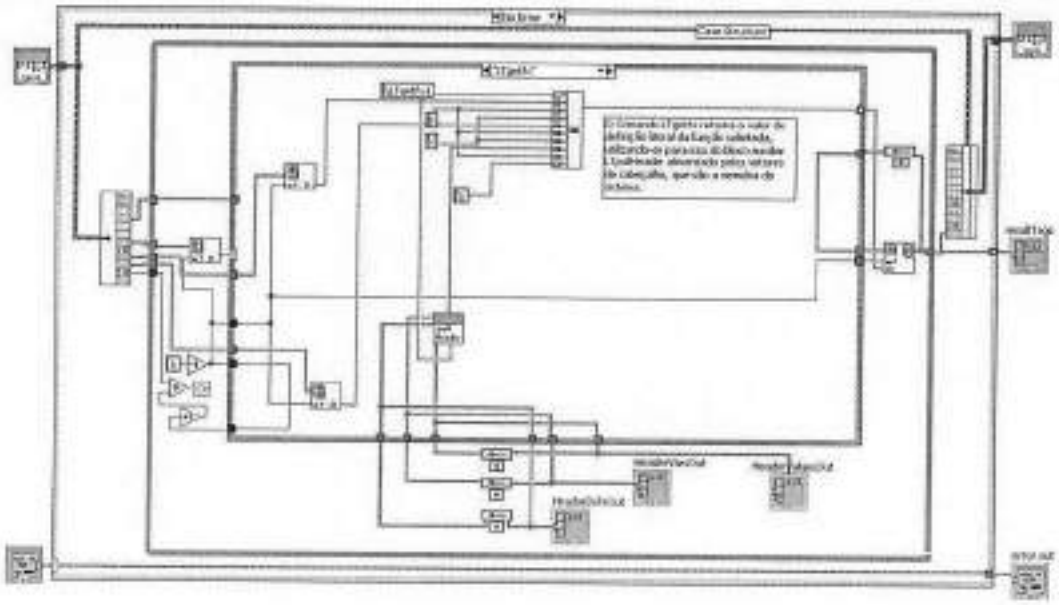


Figura 24: Anexo 7.g : LTexecuteTag.vi - Comando: LTgetfx

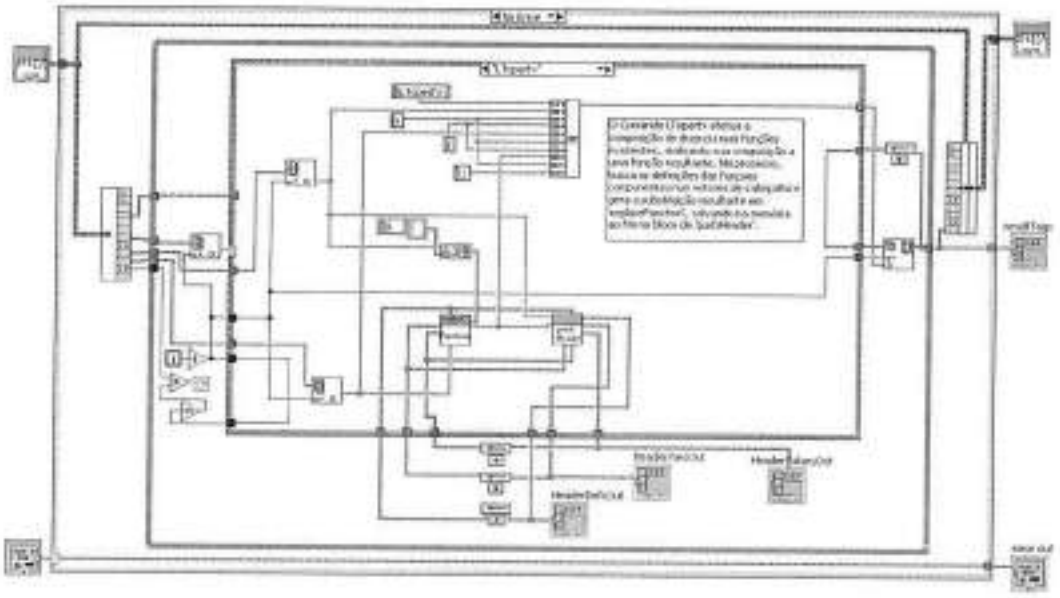


Figura 25: Anexo 7.h : LTexecuteTag.vi - Comando: LToperfx

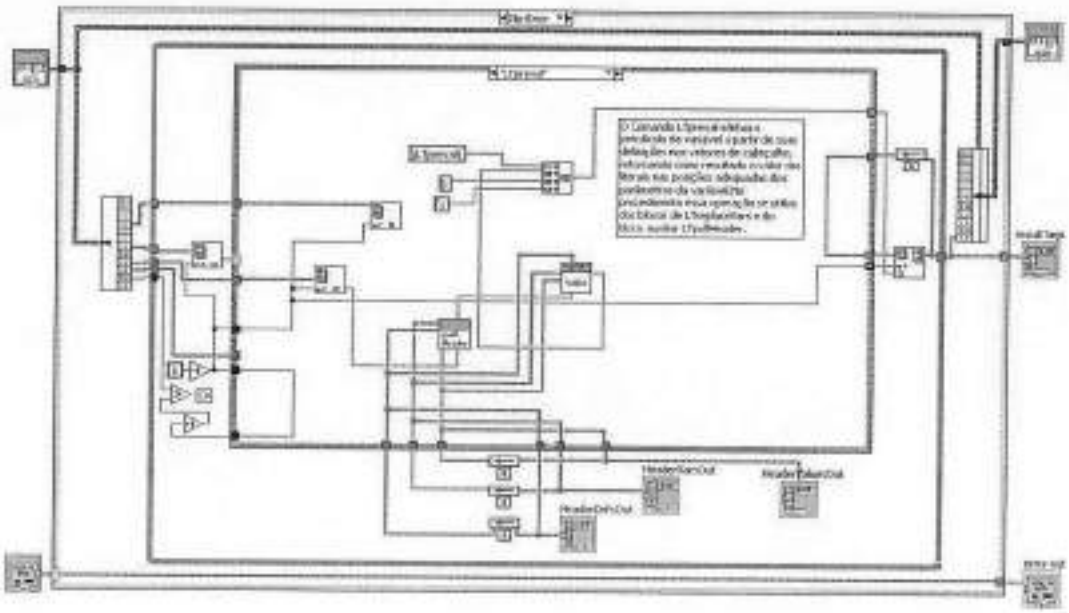


Figura 26: Anexo 7.i : LTexecuteTag.vi - Comando: LTpreval

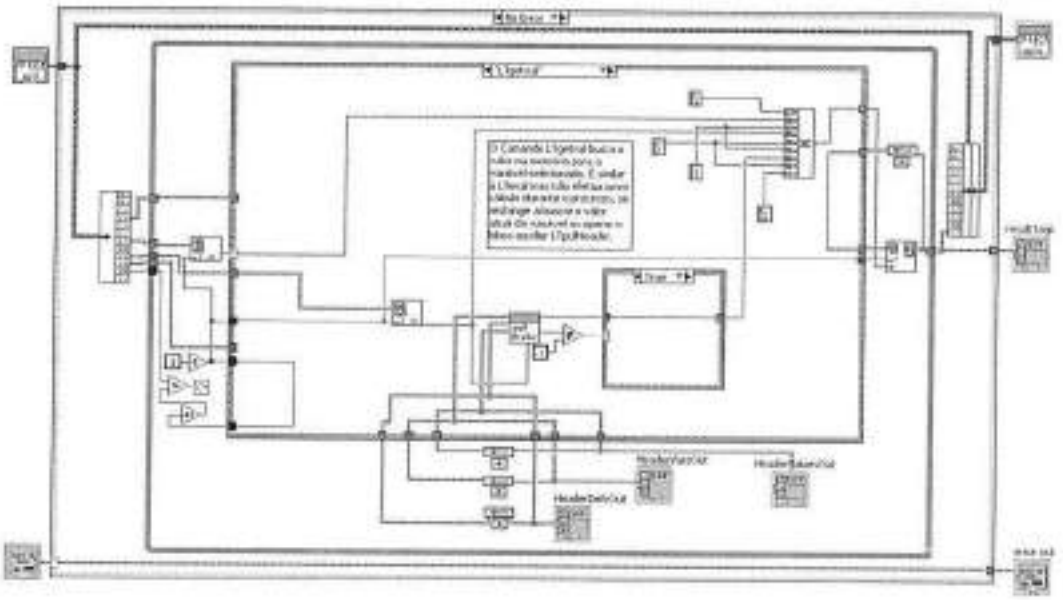


Figura 27: Anexo 7.j : LTexecuteTag.vi - Comando: LTgetval

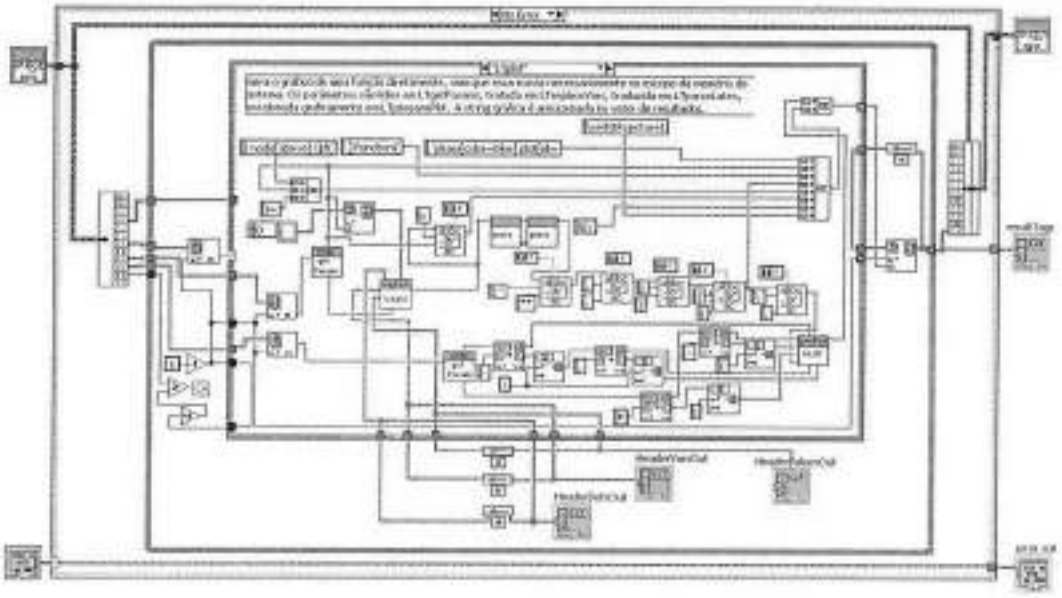


Figura 28: Anexo 7.k : LTexecuteTag.vi - Comando: LTplot

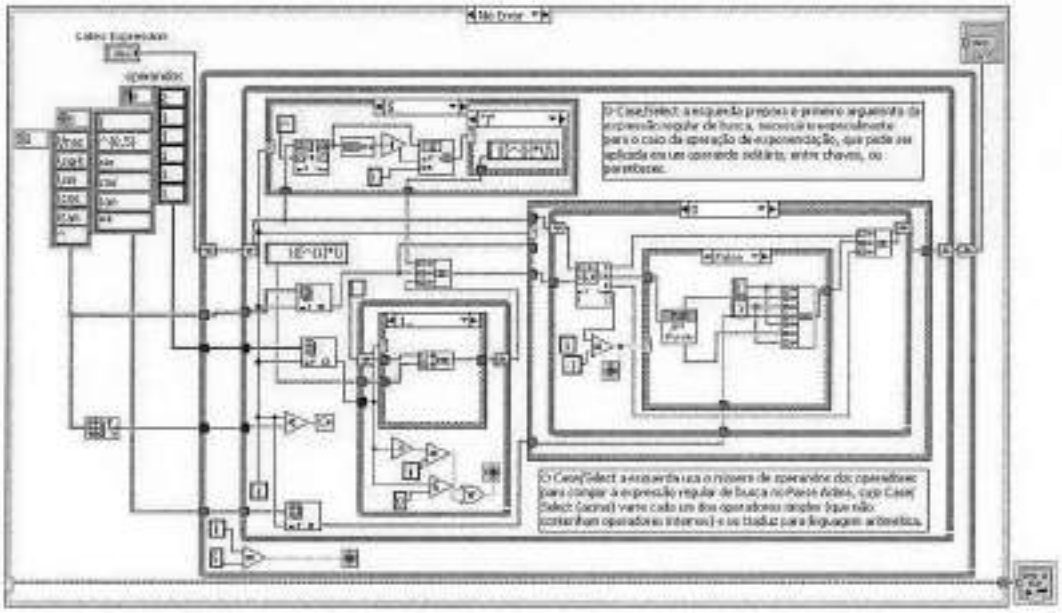


Figura 30: Anexo 08: LTparseLatex.vi - Comando: frac

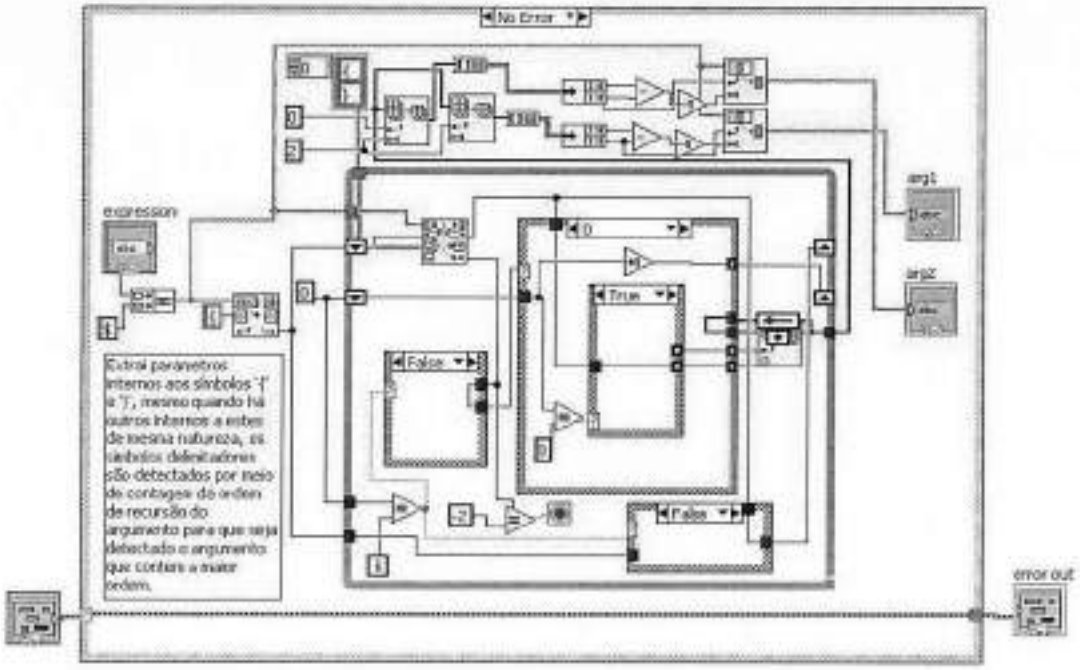


Figura 31: Anexo 09: LTgetParams.vi - Case '0' ('{')

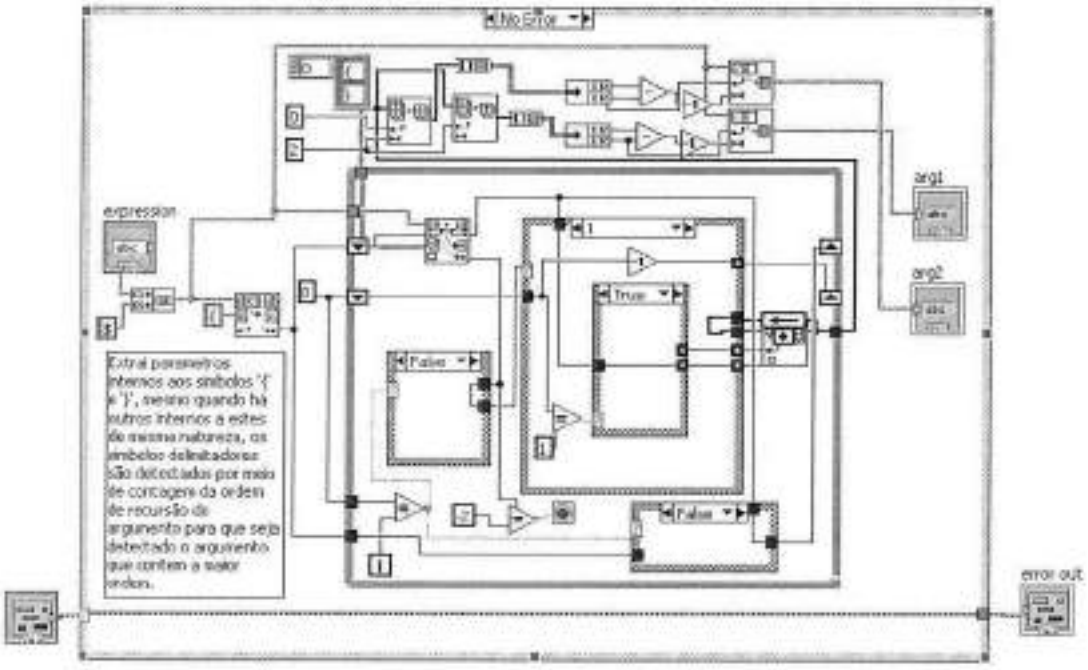


Figura 32: Anexo 09b: LTgetParams.vi - Case '1' ('}')

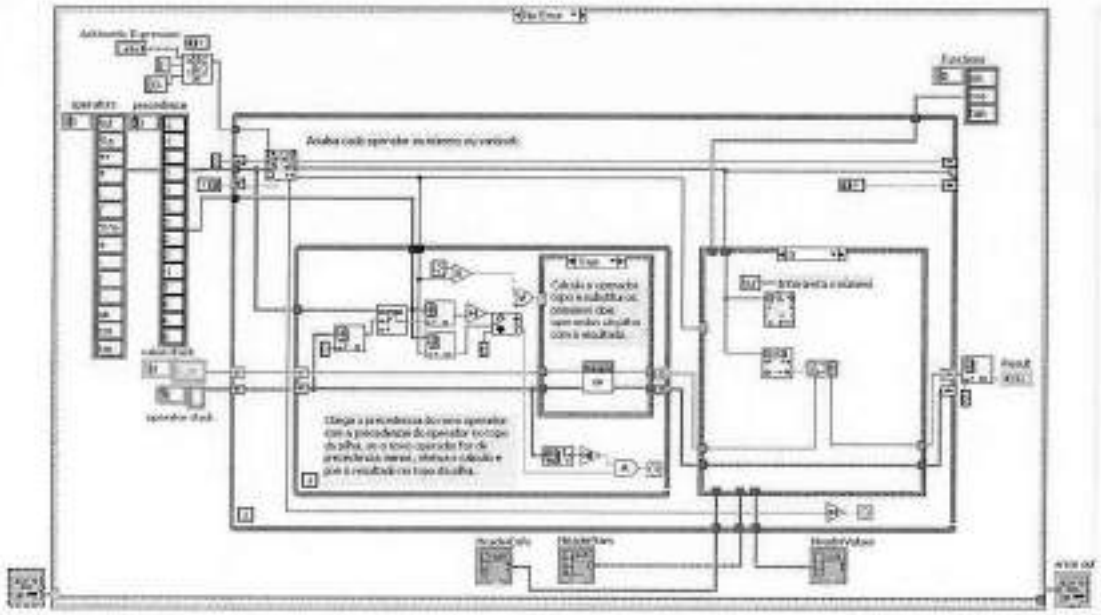


Figura 33: Anexo 10: LTparseArithm.vi - Case 0 (%f)

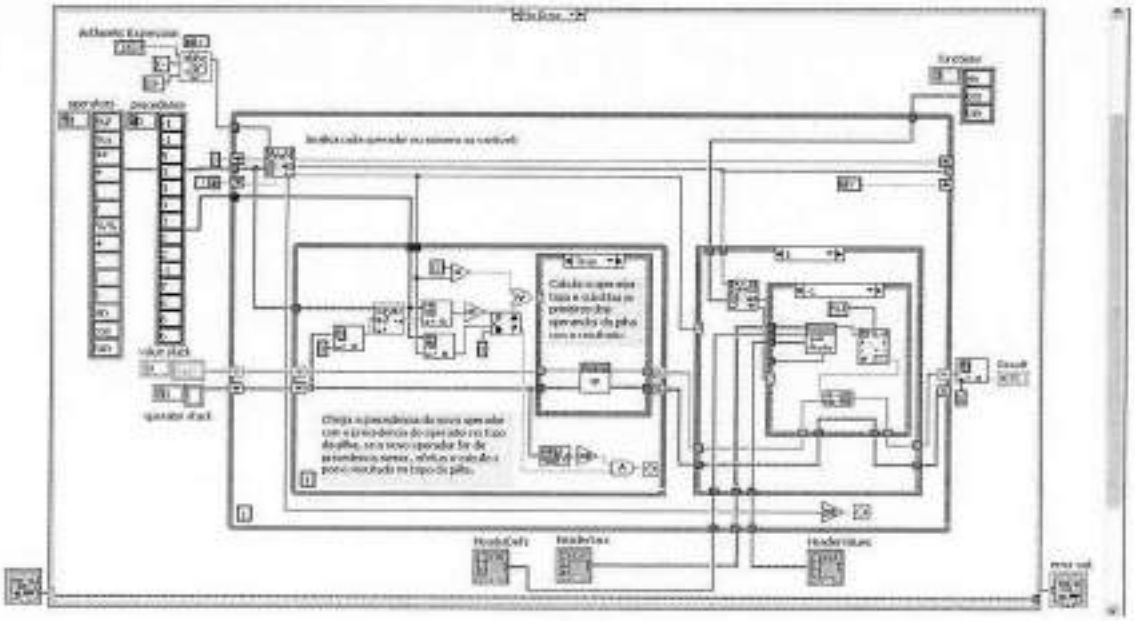


Figura 34: Anexo 10.b: LTparseArithm.vi - Case 1 (%s).

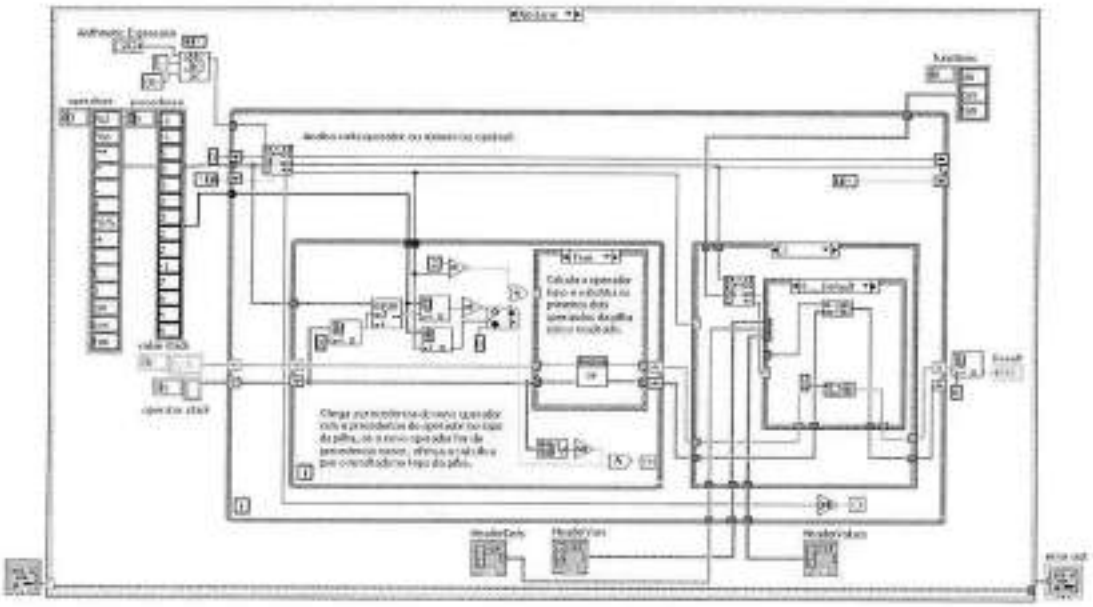


Figura 35: Anexo 10.c: LTparseArithm.vi - Case '1' (%s)

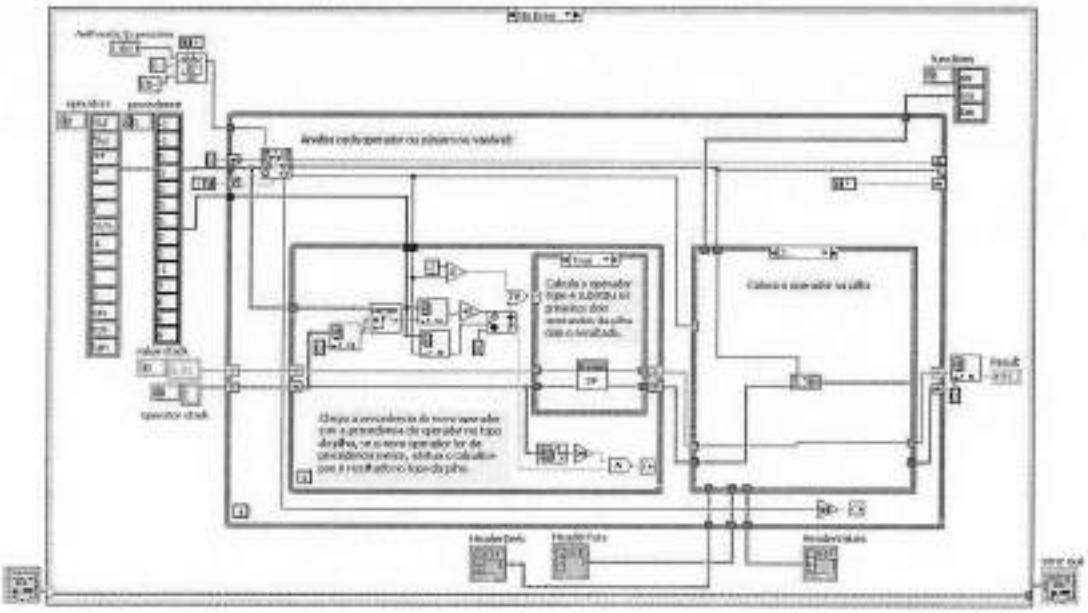


Figura 36: Anexo 10.d: LTparseArithm.vi - Caso '2,..'

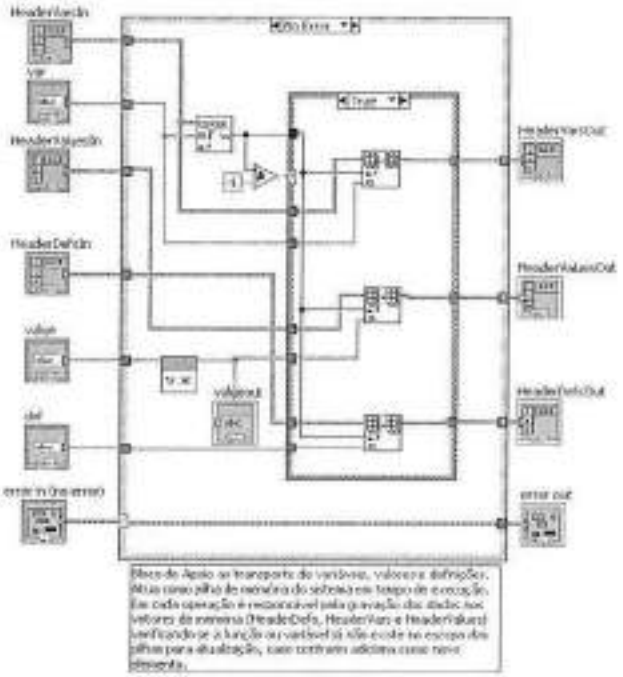


Figura 37: Anexo 11: LTpushHeader.vi

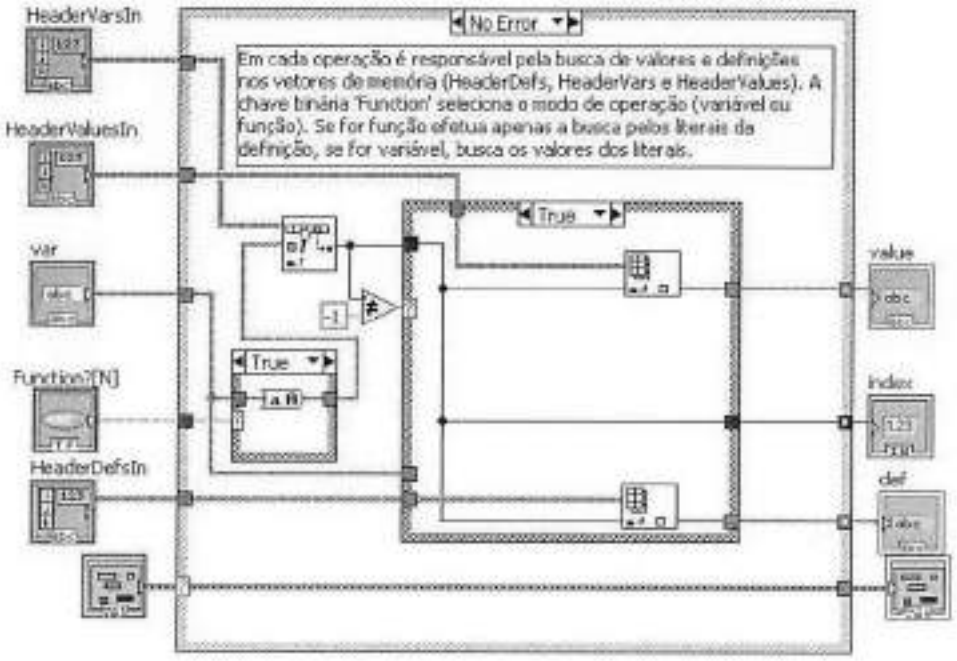


Figura 38: Anexo 12: LTpullHeader.vi

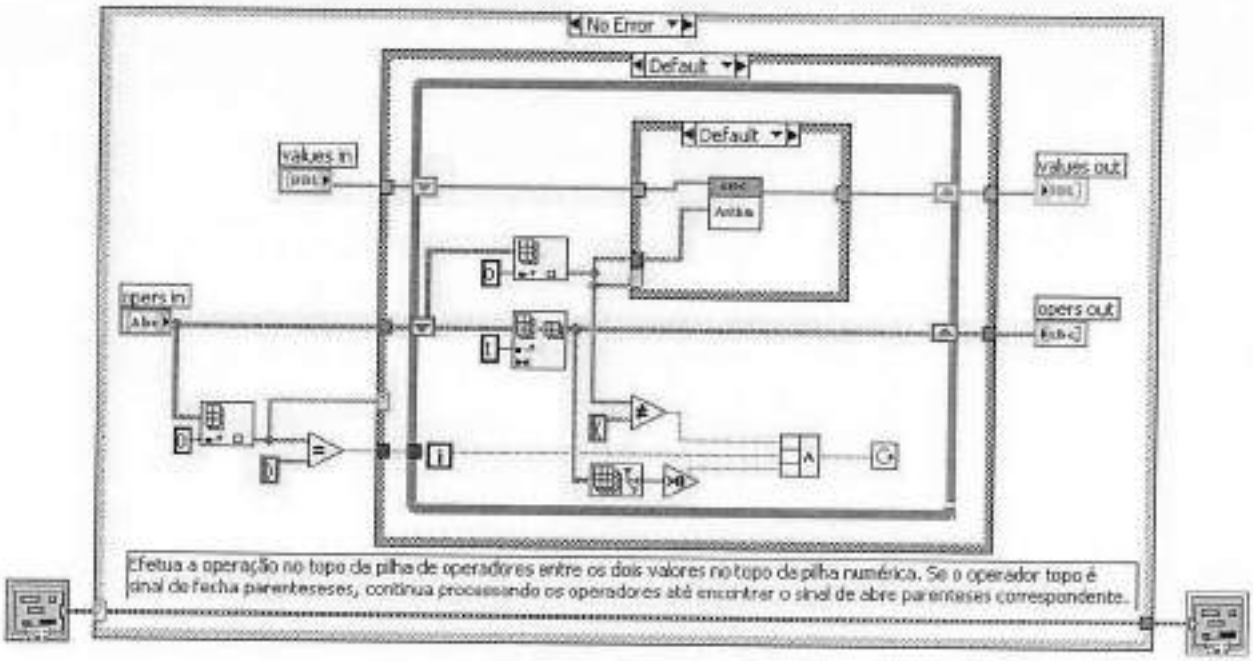


Figura 39: Anexo 13: LTProcessOperators.vi

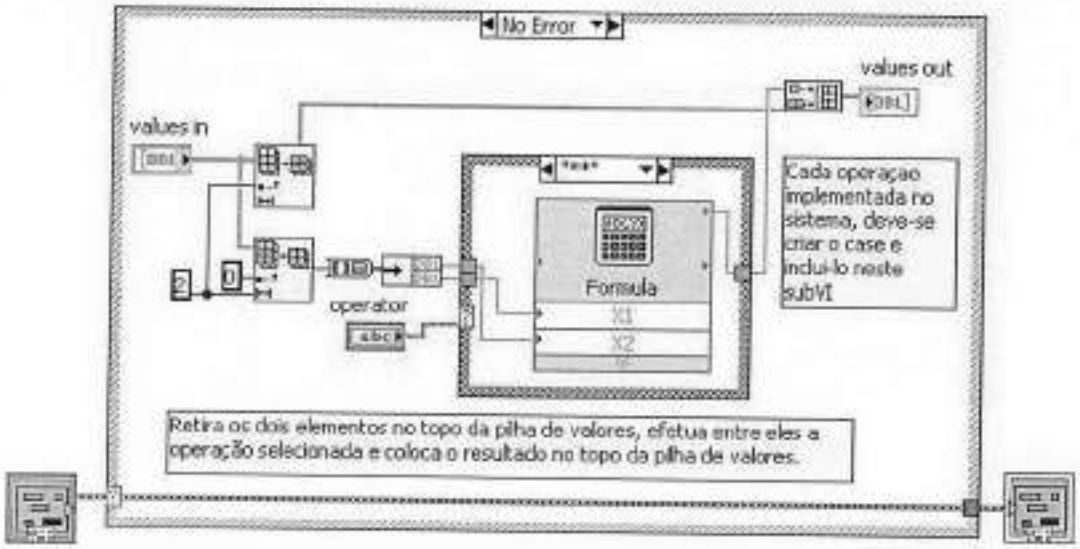


Figura 40: Anexo 14: LTexecArith.vi - Operação Soma (+)

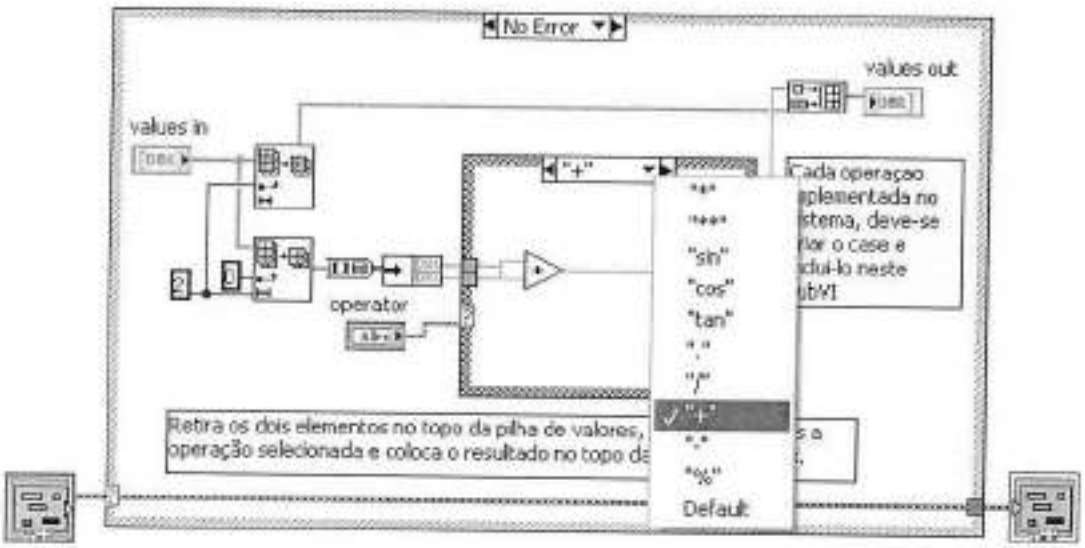


Figura 41: Anexo 14.b : LTexecArith.vi - Operações implementadas.

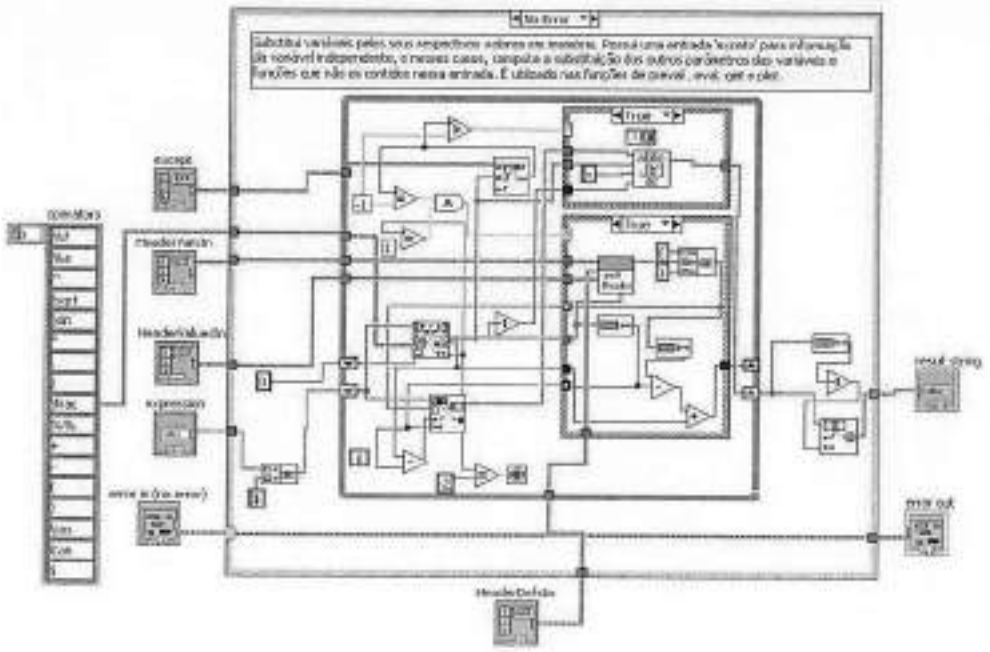


Figura 42: Anexo 15: LTreplaceVars.vi

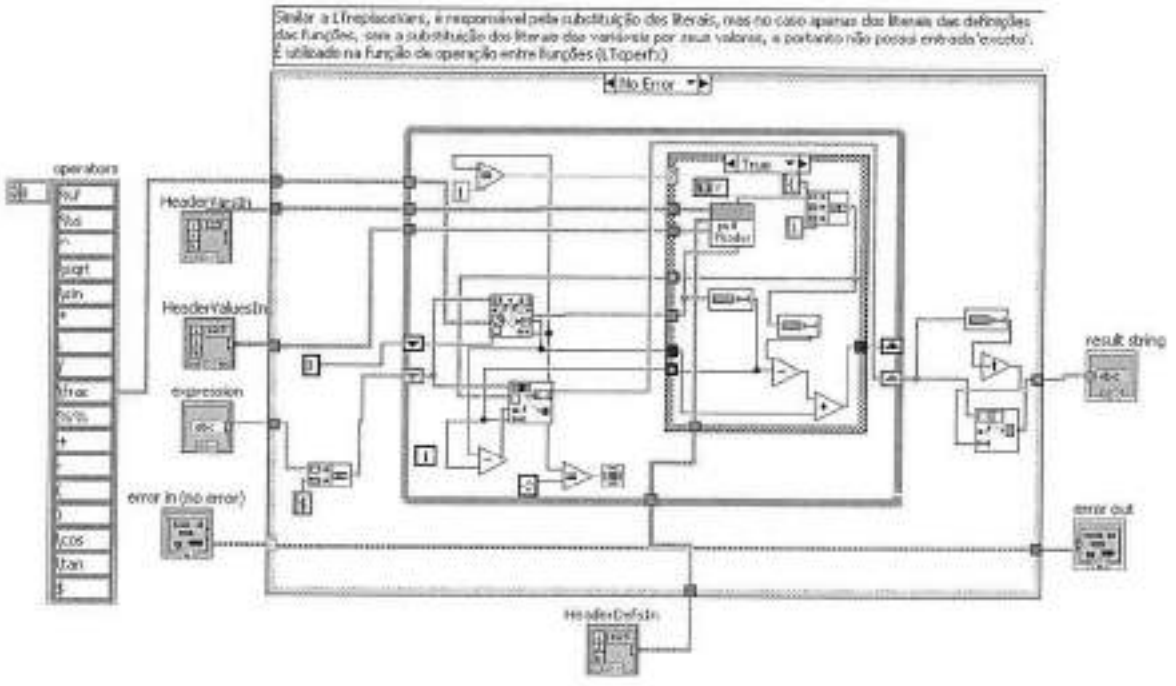


Figura 43: Anexo 15.b - LTreplaceFunction.vi

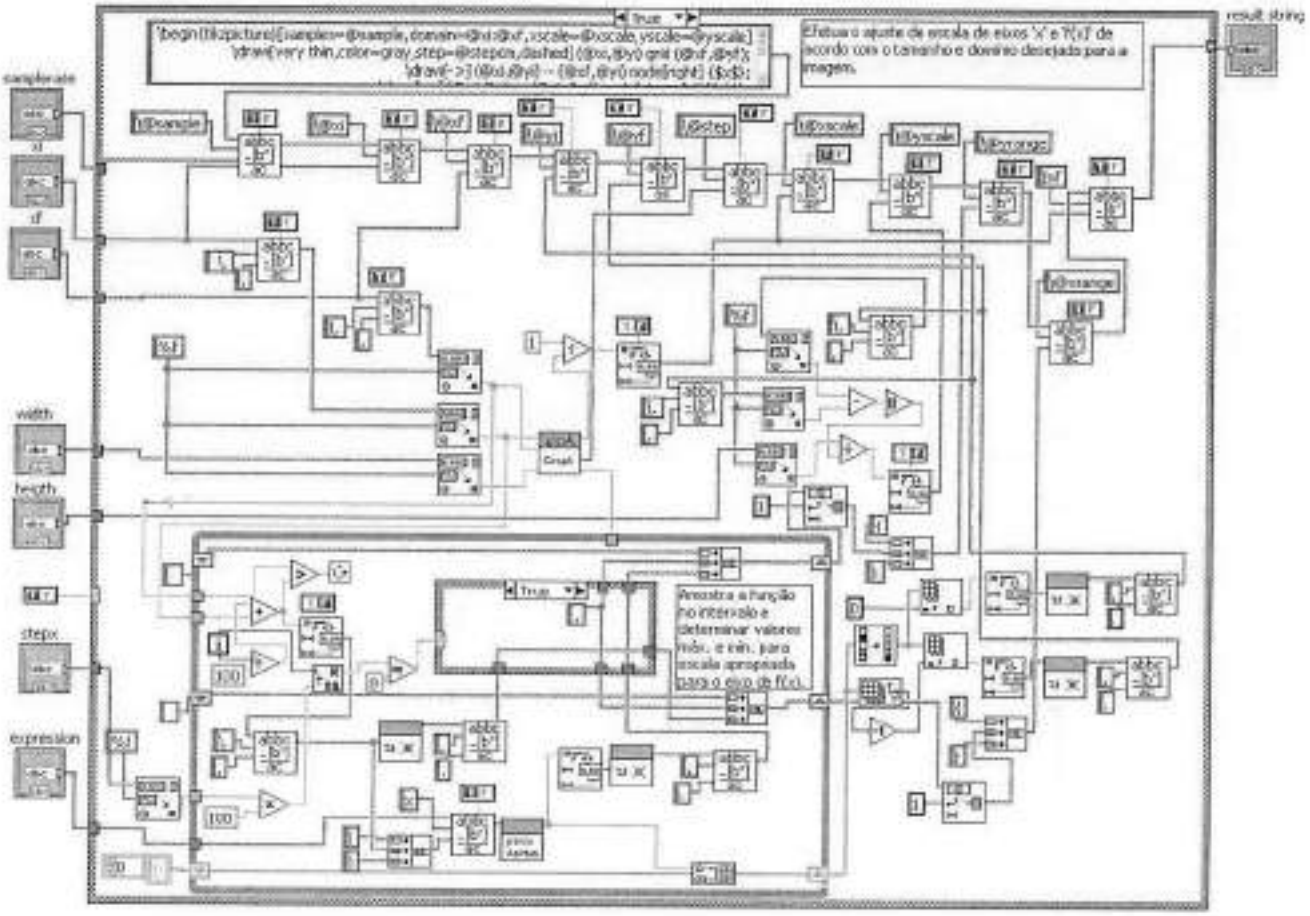


Figura 44: Anexo 16: LTpreparePlot.vi

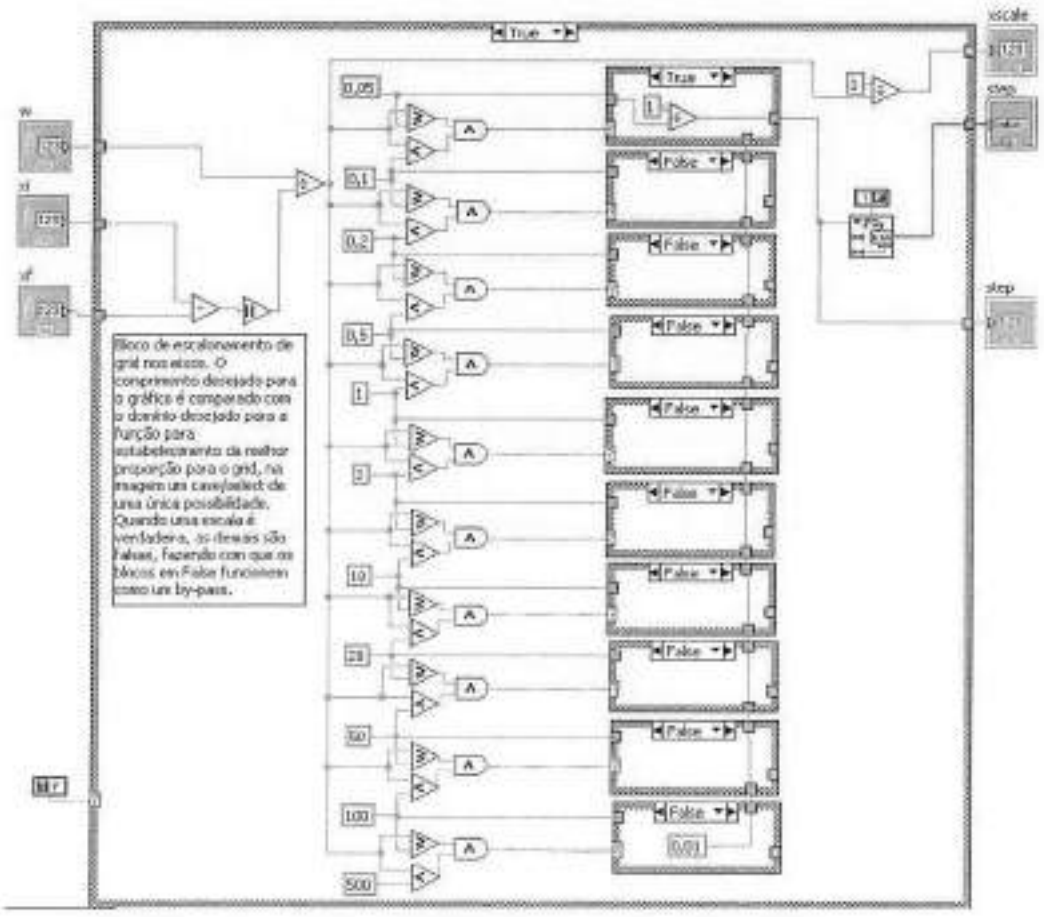


Figura 45: Anexo 16.b - LTxscaleGraph.vi

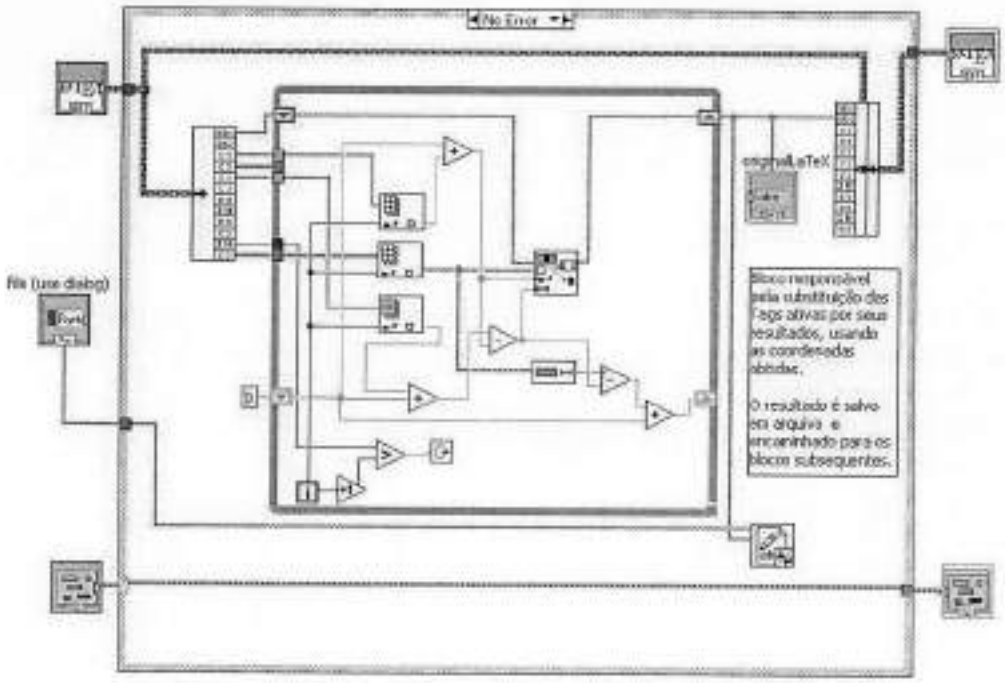


Figura 46: Anexo 17: LTreplaceData.vi

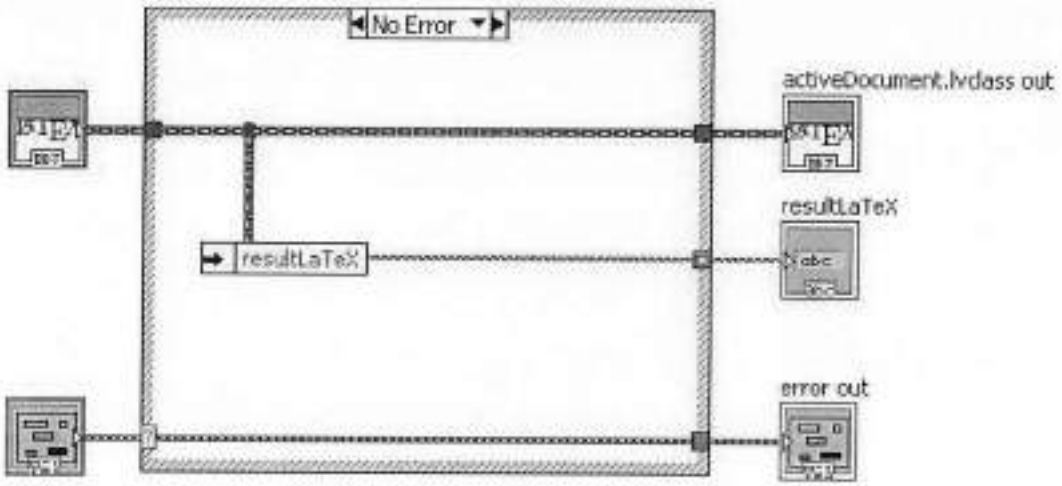


Figura 47: Anexo 18: LTreadOutput.vi

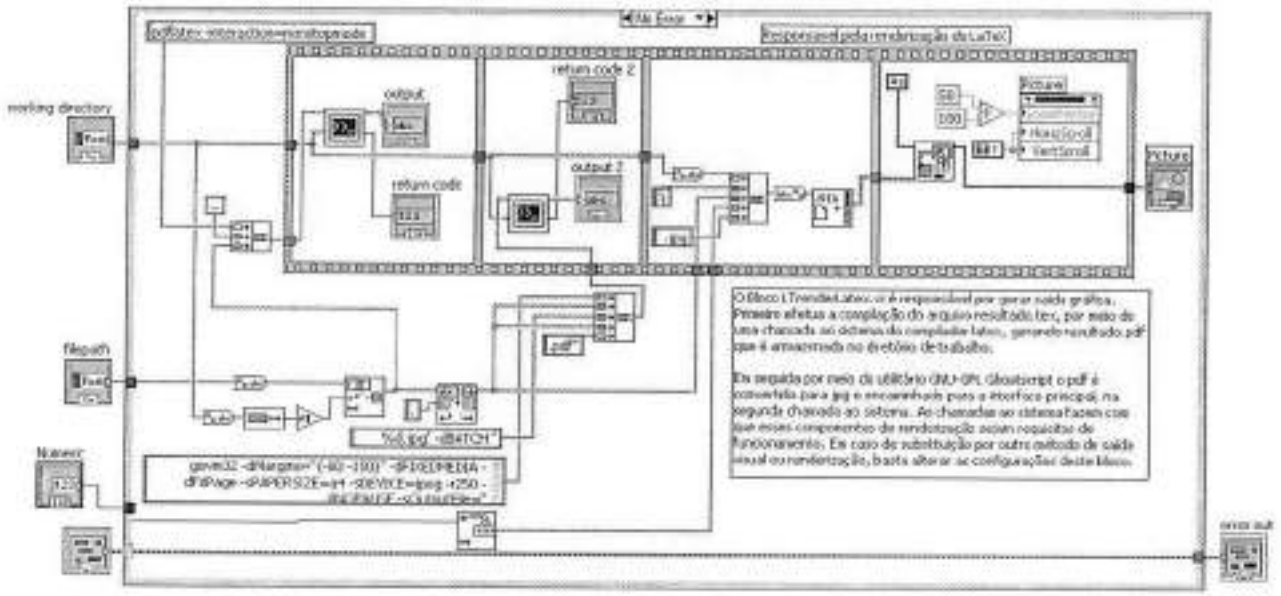


Figura 48: Anexo 19: LTrenderLatex.vi

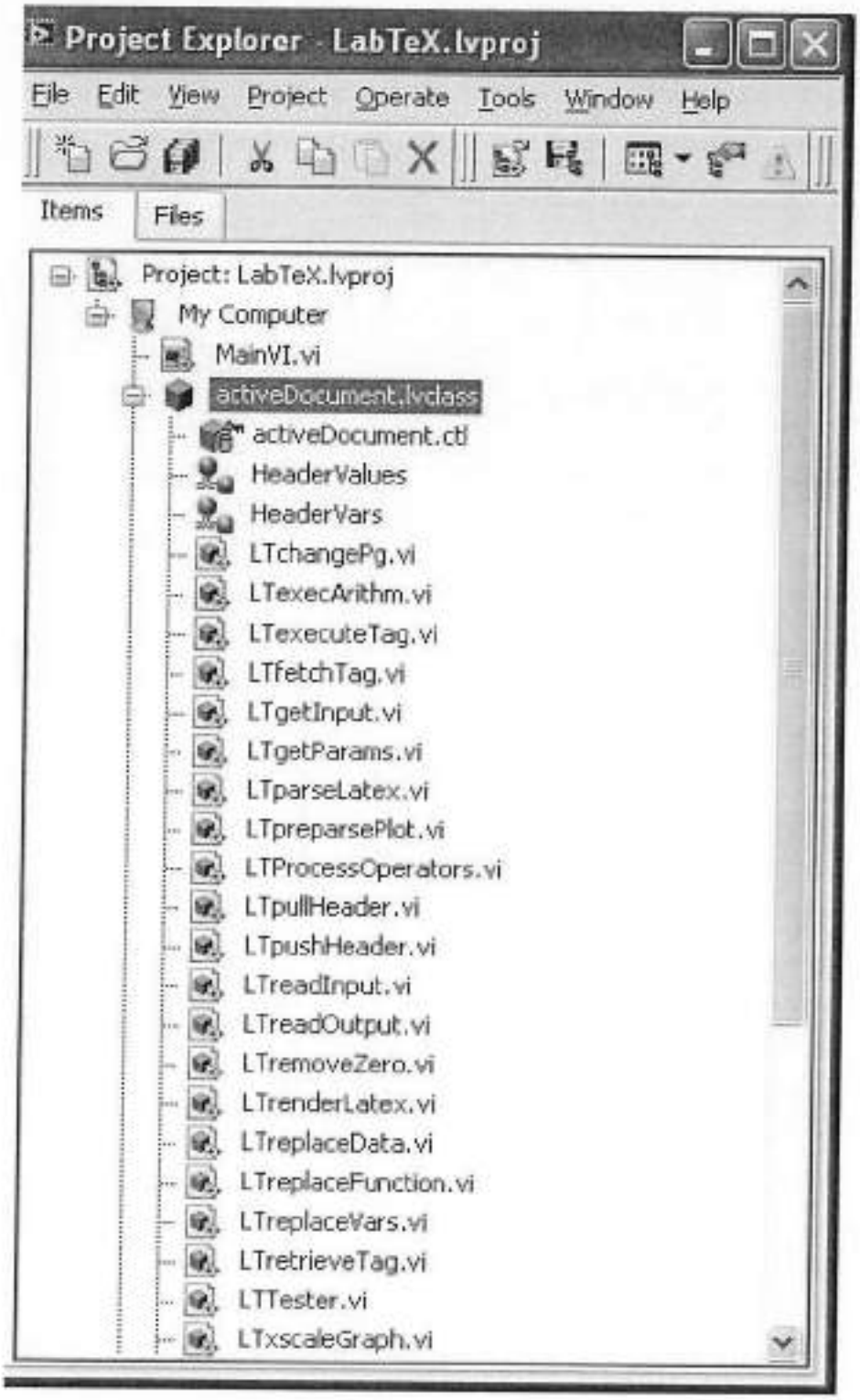


Figura 49: Anexo 20 - LabTeX.lvproj

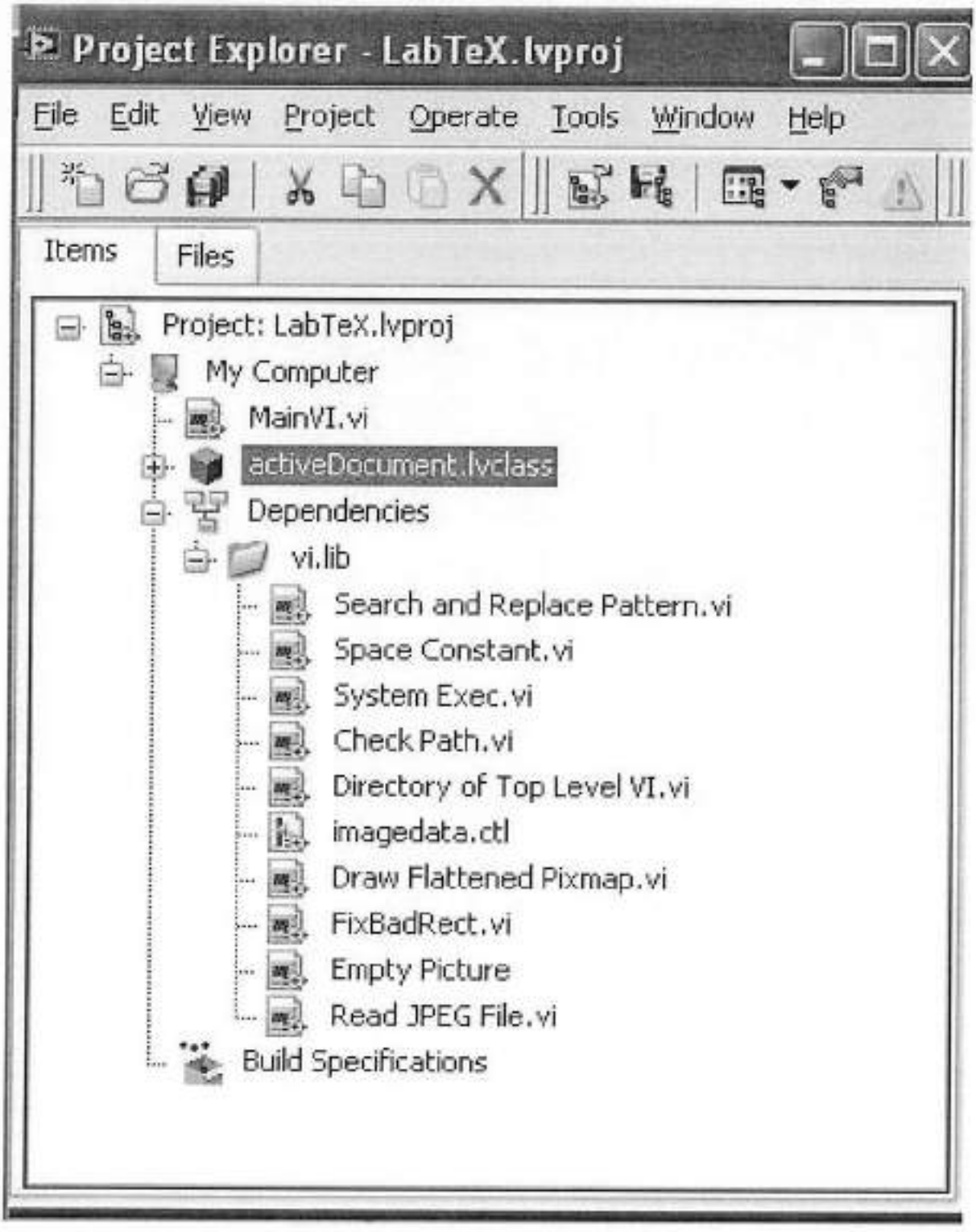


Figura 50: Anexo 20.b - LabTeX.lvproj - Dependencias

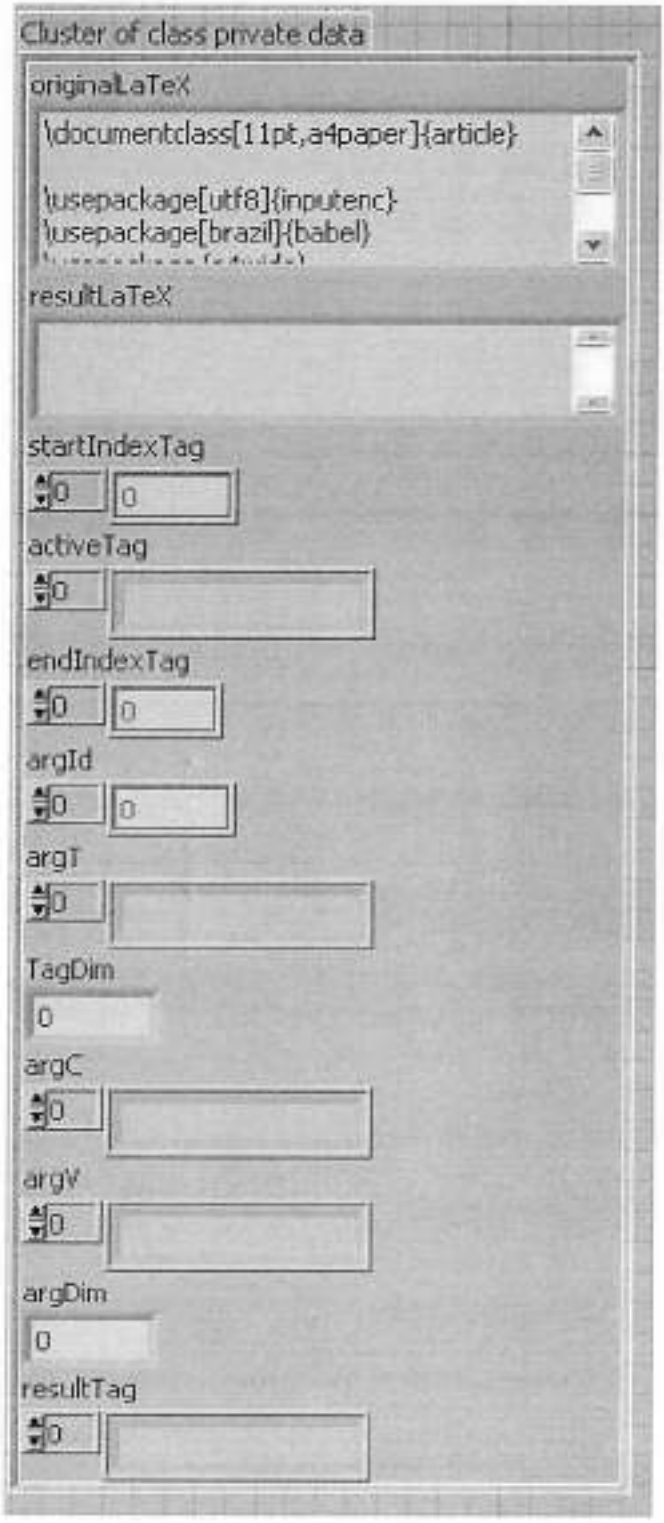


Figura 51: Anexo 21. - Classe activeDocument

Referências

[1] Paul Abrahams, Kathryn Hargreaves, and Karl Berry. *TEX for the Impatient.* URL: ftp://ftp.gwdg.de/pub/dante/info/impatient/book.pdf, 2003.

[2] David Bausum. *TEX Reference Manual.* O'Reilly Associates, Inc. URL: http://books.google.at/books?hl=en&id=lowdvusVjTwC&dq=tex&printsec=frontcover&source=web&ots=qX1xgHdwnh&sig=8CARQDtcNjo9rMBXyn3IpRRDgcU&sa=X&oi=book_result&resnum=8&ct=result.

[3] Claudio Beccari. *The curve2e package.* Comprehensive Archive TEX Network CTAN:macros/latex/contrib/curve2e/, 2005.

[4] Claudio Beccari. *Graphics in LATEX.* The PracTEX JournalAMS URL: http://www.tug.org/pracjourn/2007-1/beccari/beccari.pdf, 2007.

[5] CTAN. *UK List of TeX FAQ.* UK Comprehensive Archive TEX Network URL: http://www.tex.ac.uk/cgi-bin/texfaq2html?introduction=yes, 2009.

[6] Michael Downes. *Short Math Guide for LATEX.* American Mathematical Society. URL: ftp://ftp.ams.org/pub/tex/doc/amsmath/short-math-guide.pdf.

[7] Victor Eijkhout. *TeX by Topic - A Texnician´s Reference.* Adilson–Wesley URL: http://eijkhout.net/texbytopic/texbytopic.html.

[8] Simon Eveson. *An Introduction to Mathematical Document Production Using AMS-LaTeX.* URL: http://www-users.york.ac.uk/~spe1/texnotes07.pdf.

[9] George Gratzer. *More Math Into LATEX - 4th Edition, publisher=.*

[10] George Gratzer. *Math into LaTeX.* Comprehensive Archive TEX Network URL: http://www.ctan.org/tex-archive/info/mil/mil.pdf, 1996.

[11] Donald E. Knuth. *The TEX Book.* Adilson–Wesley, Reading, MA, third edition, 1986.

[12] João Kogler. *Programação com LabVIEW II- Videlicet.* http://jkogler.wordpress.com/2008/11/16/programacao-com-labview-2/, 2008.

[13] Conrad Kwok. *The eepic package.* Comprehensive Archive TEX Network CTAN:macros/latex/contrib/eepic/.

[14] Leslie Lamport. *BTEX, A document Preparation System.* Adilson–Wesley, Reading, MA, 1986.

[15] Leslie Lamport. *BTEX, A document Preparation System - 2nd. edition.* Adilson–Wesley, Reading, MA, 1994.

[16] Philipp Lehman. *The etoolbox package - An e-TeX toolbox for class and package authors. Version 1.7.* Comprehensive Archive TEX Network URL: `http://www.tex.ac.uk/tex-archive/macros/latex/contrib/etoolbox/etoolbox.pdf`, 2008.

[17] Sadao Massago. *LATEX via Exemplos.* URL: `http://www.dm.ufscar.br/~sadao/curso/latex/`, 2004.

[18] Scott Pakin. *The Comprehensive Latex Symbol List.* .

[19] Manuel Pégourié-Gonnard. *The ted package. v1.06.* Comprehensive Archive TEX Network URL: `http://www.tex.ac.uk/tex-archive/macros/latex/contrib/ted/ted.pdf`, 2008.

[20] Sunil Podar. *The epic package.* Comprehensive Archive TEX Network CTAN:macros/latex/contrib/epic/.

[21] Kristofer Høgsbro Rose and Ross Moore. *The XY-pic bundle.* Comprehensive Archive TEX Network CTAN:applications/Xy-pic/Xy-pic.html.

[22] Educatec Shop. *NI USB-6009 Funcionalidade Básica e custo.*

[23] J.H. Silverman. *Plain TEX Quick Reference Card.* Indian Ford Mason Ltd, GL19 3JB, UK URL: `http://refcards.com/docs/silvermanj/tex/tex-refcard-letter.pdf`, Math. Dept., Brown Univ., Providence, RI 02912 USA, 1998.

[24] LaTeX Project Site. *LATEX A document preparation system.* O'Reilly Associates, Inc. URL: `http://www.latex-project.org/`.

[25] Till Tantau. *The TikZ and PGF Packages - Manual for version 2.00, publisher=.*

[26] Tutorial Team. *Online Tutorials on LATEX.* Indian TEX User Group URL: `http://www.tug.org/tutorials/tugindia/`, 2000.

[27] UK-TUG. *UK-TUG Blog: Promoting the use of TeX in the United Kingdom.* UK TEX User Group URL: `http://uk.tug.org/`, 2009.

[28] Norman Walsh. *Making TEX Work.* O'Reilly Associates, Inc. URL: `http://makingtexwork.sourceforge.net/mtw/`, 1994.

[29] Michael Wichura. *The PiCTEX package.* Comprehensive Archive TEX Network CTAN:graphics/pictex/.

[30] Gerben Wierda. *The TeX showcase.* TEX User Group URL: `http://www.tug.org/texshowcase/`.

[31] Ghostscript *Ghostscript.* URL: `http://www.ghostscript.com/Ghostscript.html`.

[32] Gnuplot *Gnuplot.* URL: `http://www.gnuplot.info/`.