

# **LPP – Linguagem de Projeto de Programação: proposta de padronização da estrutura sintática de uma linguagem de projeto de programação a ser definida para a área de desenvolvimento de software para países com idioma português**

**JOSÉ AUGUSTO NAVARRO GARCIA MANZANO<sup>1</sup>**

## **RESUMO**

Este texto apresenta uma proposta de modelo textual para a representação da linha de raciocínio lógico do conceito de programação de computadores em uma linguagem de projeto de programa em português.

**Palavras-chave:** Linguagem de programação de computadores, programação de computadores, pseudocódigo, linguagem de projeto de programação, português estruturado, algoritmo descritivo, programação estruturada de computadores, lógica de programação de computadores.

## **ABSTRACT**

This paper presents a text model representing the computer logical reasoning of programming concept in a language design program in portuguese.

**Key-words:** Programming language, computer logical, computation

---

<sup>1</sup> Professor na área de Tecnologia da Informação (TI) leciona as disciplinas de Algoritmos, Lógica de Programação, Estrutura de Dados, Linguagens de Programação (PASCAL, C, C++, JavaScript, entre outras), Tópicos Avançados de Processamento de Dados, Engenharia de Software, Arquitetura de Computadores, Engenharia da Informação. Atua profissionalmente nas instituições de ensino superior: Faculdade Cantareira (São Paulo – SP) e Faculdades Interação Americana (São Bernardo do Campo – SP). Possui mestrado em Administração de Empresas e é autor de vários títulos na área de TI publicados pela Editora Érica Ltda.

## 1. INTRODUÇÃO

Este artigo tem por objetivo discutir e propor um formato padronizado para a elaboração da documentação escrita da linha de raciocínio lógico a ser definida e utilizada como representação genérica e textual do código de programação de computadores dentro de um padrão em língua portuguesa, por meio do uso de uma Linguagem de Projeto de Programação Portuguesa (LPP), de maneira que possa tal código textual de documentação ser facilmente traduzido para uma linguagem formal de programação de computadores.

O tema deste artigo não é um assunto inédito, pois a primeira publicação registrada a este respeito ocorreu no ano de 1975 em um artigo intitulado “PDL - A Tool for Software Design”, escrito pelos pesquisadores Stephen H. Caine e E. Kent Gordon, que o apresentaram em uma conferência nacional sobre computação no Estado da Califórnia, nos Estados Unidos da América. Neste artigo os autores apresentam uma forma de escrita que objetiva representar, de forma clara, as ações a serem executadas em um computador sem levar em consideração o uso de qualquer condição que sugira o uso de uma linguagem formal de programação de computadores. A sigla PDL apresentada por Caine e Gordon significa *Program Design Language*, sendo então sugerido neste artigo como correlato o termo LPP (Linguagem de Projeto de Programação) como sua mais pura tradução. Não se trata de apresentar uma tradução da proposta PDL para LPP, mas sim o de formatar e propor critérios que permitam o uso de uma nomenclatura portuguesa mais equilibrada e homogênea em relação à forma

que vem sendo utilizada por vários profissionais e autores por estar num estágio bastante confuso e desorganizado. Deseja-se com este trabalho sugerir algumas regras de escrita que vão mais além das orientações apresentadas por Caine e Gordon, pois se pretende aliar junto à técnica de PDL a definição e uso de um formato de pseudocódigo único, no sentido de evitar a multiplicidade de dialetos atualmente encontrados.

O conceito de pseudocódigo caracteriza-se por ser uma técnica de codificação de programação que por meio de uma linguagem quase computacional consiga representar na forma escrita do idioma local a ação lógica a ser executada por um programa de computador. É óbvio que a proposta de LPP apresentada neste artigo possui características herdadas da técnica de PDL e da técnica de pseudocódigo e por este motivo será mantido nesta proposta o mesmo rigor de representação do código de programa que existe em uma linguagem formal de programação de computadores.

O fator que motivou e incentivou a elaboração desta proposta de trabalho decorre do fato de vários profissionais da área de desenvolvimento de programas de computadores (*software*) principalmente no Brasil ao fazerem (quando fazem) uso das técnicas (PDL e/ou pseudocódigo), não o fazem segundo uma regra ou forma única, cada qual, quando a usa faz de uma forma particular e diferente um do outro, ocasionando dificuldades no uso desta técnica. Assim sendo, espera-se com este artigo fornecer uma contribuição de subsídios e orientações para a elaboração padronizada e formal de uma linguagem escrita

que represente de forma textual a lógica de programação de computadores a ser utilizada como elemento de documentação de código de programa.

## 2. FUNDAMENTAÇÃO CONCEITUAL

Com o objetivo de tornar mais claro o entendimento sobre os conceitos **PDL** (*Program Design Language*) e **pseudocódigo** faz-se a seguir uma rápida descrição das principais características de cada técnica. É pertinente ressaltar que não é objetivo deste estudo explanar detalhadamente estes conceitos, mas apenas situar o leitor neste contexto.

O período anterior ao ano de 1975 foi marcado por um aumento nos custos de produção e desenvolvimento de programas de computadores e pela queda na qualidade e na confiabilidade destes programas. A partir de 1971 foram sendo desenvolvidas algumas técnicas de desenvolvimento que visavam facilitar o trabalho de criação, implementação e manutenção de programas e sistemas, entre estas técnicas se consolidou a técnica de *programação estruturada* e a técnica *top-down* (técnica de desenvolvimento que estabelece na fase de criação em pensar num sistema computacional de cima para baixo), apesar de existir desde o início da década de 1960 estudos e propostas de programação de computadores com orientação a objetos (PRESSMAN, 2005).

Muitas das técnicas desenvolvidas a partir de 1971 e mesmo antes deste período influenciaram o desenvolvimento da técnica de documentação: PDL,

a qual proporcionou, segundo seus autores, um aumento de produtividade no desenvolvimento de programas de computadores. Relata-se que em projetos complexos se obteve um aumento na taxa de produção, muito acima do ciclo normal de desenvolvimento. É apontado também que houve um ganho de 60 a 65 linhas de código de programa por dia/homem e uma redução de 0.25 horas no tempo de processamento efetuado por um computador. De acordo com Caine e Gordon (1976) estas taxas de produção usando a técnica PDL foram realizadas aproximadamente na metade do tempo quando comparada com o uso de apenas técnicas estruturadas e de 4 a 6 vezes melhor quando comparada com técnicas clássicas de desenvolvimento.

O conceito de pseudocódigo caracteriza-se por ser uma linguagem de alto nível (linguagem semelhante a forma escrita humana) de programação de computadores não formal (linguagem que normalmente não possui sua implementação real em um computador). Um pseudocódigo é uma estrutura de escrita que possui uma sintaxe particular (seguindo um estilo computacional), podendo esta estrutura ser representada em qualquer idioma falado no planeta. Não há definido para a representação do pseudocódigo uma forma sintática padronizada, embora normalmente se utilize a influência de uma linguagem formal de programação de computadores, tais como: ALGOL, PASCAL, C, C++, entre outras.

O objetivo primordial de um pseudocódigo é o de representar textualmente e de forma genérica a linha de pensamento lógico e o fluxo de execução de um programa de computador. O pseudocódigo é em sua essência

uma forma escrita de se estabelecer um algoritmo (conjunto de regras e operações bem definidas e ordenadas, destinadas à solução de um problema, ou de uma classe de problemas, em um número finito de etapas – dicionário Aurélio) que represente de forma clara a ação da programação de um computador por meio de convenções estruturais.

O uso do pseudocódigo permite omitir os rigores de seqüências de dados ou detalhes de uma determinada linguagem formal de programação de computadores, uma vez que muitas vezes a linguagem formal de programação de computadores possui características particulares a seu funcionamento (WIKIPEDIA, 2006). O pseudocódigo representa de forma direta a ação a ser executada pelo programa que ainda será codificado em uma linguagem formal de programação de alto nível. A idéia de uso do pseudocódigo sugere a definição da programação de um computador independentemente deste computador existir concretamente, uma vez que o trabalho de programação se baseia no conceito de abstração. No entanto, o conceito de abstração é normalmente deixado de lado quando se trata de literaturas especializadas nesta área, a maior parte das obras publicadas versa normalmente o tema sobre o uso dos computadores em si, mas não versam o fato de como se deve fazer para entender como esses computadores realmente funcionam, e uma forma de aprender a entender o nível de abstração que os computadores fazem uso é programá-los estando longe deles (TENÓRIO, 2001). Primeiramente desenvolvendo-se os seus diagramas de blocos de acordo com a norma ISO 5807:1985 (E) e em seguida de preferência escrevendo o código do programa antes em pseudocódigo.

Aparentemente pode até parecer que a definição e a implementação da etapa de escrever primeiro o código do programa em pseudocódigo para depois codificá-lo em uma linguagem de programação formal seja demorada e que atrase o tempo de projeto, mas ela garante legibilidade, eficiência e ganho de tempo na conclusão do projeto, como apontado por Caine e Gordon.

### 3 . LINGUAGEM DE PROJETO DE PROGRAMAÇÃO – LPP

Uma das questões discutidas neste artigo é o de se definir um formato textual padronizado e homogêneo para a definição e utilização dos comandos de pseudocódigo português a ser utilizado na documentação do código de um programa de computador. Além da preocupação em relação às palavras de comandos a serem definidas, existe a questão relacionada a forma de nomeação da técnica em português, pois são vários os nomes sugeridos, tais como: *portugol* (GUIMARÃES & LAGES, 1994), *linguagem de descrição* (VELOSO, et. al., 1996), *linguagem universal de programação* (TERADA, 1991), *linguagem de leitura simples* (SZWARCFITER & MARKENZON, 1994), e outras tantas formas encontradas e definidas no sentido de tentar representar e indicar a mesma idéia e o mesmo conceito.

É no sentido de homogeneizar e ter-se uma forma padronizada que este artigo foca sua discussão, uma vez que se propõe um formato único a ser

considerado oficialmente e desta forma poder ser usada por todas as pessoas relacionadas ao processo de desenvolvimento de *software* de uma mesma maneira. Sugere-se neste artigo a definição da estrutura de documentação e representação textual da linha de raciocínio computacional utilizando-se uma **Linguagem de Projeto de Programação** (LPP) em português levando-se em consideração e influência de sua forma análoga em inglês **Programa Design Language** (PDL).

O fato de cada autor ou profissional definir uma forma particular de escrever a estrutura funcional de um código computacional e também de definir um nome de identificação não caracteriza nenhuma espécie de erro, mas acaba por criar uma série de dialetos que se tornam confuso e criam interpretações e discussões desnecessárias sobre qual forma é ou não válida. Por esta razão é que se sugere uma forma mais homogênea. A seguir encontra-se a definição da tabela 1 com as palavras de comando (palavras-chave ou palavras reservadas) de pseudocódigo usadas pelos autores pesquisados e suas correlações com as linguagens formais de programação PASCAL<sup>2</sup> e STRUCTURE BASIC<sup>3</sup> e a proposta da Linguagem de Projeto de Programação:

---

<sup>2</sup> Em relação a linguagem PASCAL estão sendo considerados os seus principais comandos.

<sup>3</sup> Em relação a linguagem STRUCTURE BASIC está sendo considerado o ambiente QuickBASIC.



**Tabela 1 – Comparativo das Formas de Sintaxe**

GUL	VEL	TER	SZM	PAS	BAS	LPP
ATÉ	ATÉ	ATÉ		TO	TO	ATÉ
ATÉ	ATÉ	FIM-REPITA		UNTIL	LOOP	ATÉ QUE
CARACTER	CAR			STRING	STRING	CARACTERE
				CASE	SELECT	CASO
VETOR/MATRIZ	VET			ARRAY	DIM	CONJUNTO
DE	DE	←		:=	=	DE
ENQUANTO	ENQUANTO	ENQTO	ENQUANTO	WHILE	WHILE	ENQUANTO
ENTÃO	ENTÃO	ENTÃO	ENTÃO	THEN	THEN	ENTÃO
IMPRIMA ()	ESCREVA ()	PARE-COM-SAÍDA ()		WRITE ()	PRINT	ESCREVA
FAÇA	FAÇA	FAÇA	FAÇA	DO		FAÇA
FIM	FIM	PARE		END		FIM
				END	END SELECT	FIM CASO
FIM ENQUANTO		FIM-ENQTO		;	WEND	FIM ENQUANTO
FIM PARA		FIM-PARA	PARE	;	NEXT	FIM PARA
FIM REGISTRO				END	END TYPE	FIM REGISTRO
FIM SE		FIM-SE		;	END IF	FIM SE
INÍCIO	INÍCIO			BEGIN		INÍCIO
INTEIRO	INT			INTEGER	AS INTEGER	INTEIRO
LEIA ()	LEIA ()			READ ()	INPUT	LEIA
LÓGICO	LOG			BOOLEAN		LÓGICO
PARA	PARA	PARA	PARA	FOR	FOR	PARA
PASSO	INCR				STEP	PASSO
PROCEDIMENTO	PROC		PROCEDIMENTO	PROCEDURE	SUB	PROCEDIMENTO
			ALGORITMO	PROGRAM		PROGRAMA
REAL	REAL			REAL	AS SINGLE	REAL
REGISTRO	REG			RECORD		REGISTRO
REPITA	REPITA	REPITA		REPEAT	DO	REPITA
SE	SE	SE	SE	IF	IF	SE
				OF	CASE	SEJA
SENÃO	SENÃO	SENÃO	SENÃO	ELSE	ELSE	SENÃO
TIPO	TIPO			TYPE	TYPE	TIPO
	VAR			VAR	DIM	VAR

Para uma melhor interpretação da tabela considere o nome de cada coluna segundo a seguinte legenda: GUL – Guimarães e Lages; VEL – Veloso, et. al.; TER – Terada; SZM – Szwarcfiter & Markenzon; PAS – Código Pascal; BAS – Código em Structure BASIC e LPP – Linguagem de Projeto de Programação.

A indicação das linguagens formais de programação PASCAL e STRUCTURE BASIC foram aqui consideradas para ajudar a demonstrar o motivo da definição de uma regra de codificação mais genérica e geral para a forma LPP em relação ao formato utilizado pelos demais autores. Observe que a LPP engloba um conjunto maior de instruções em relação ao conjunto de instruções das linguagens PASCAL e STRUCTURE BASIC e também do conjunto proposto no trabalho particular de cada um dos autores apontados.

Os campos da tabela que se encontram grafados em branco em qualquer uma das formas apresentadas mostram a ausência de comandos referentes às ações indicadas pelos autores de cada obra consultada. Quanto aos campos relacionados às linguagens formais de programação PASCAL e STRUCTURE BASIC que estão em branco, estes indicam a ausência da ação na linguagem em si. Esta ocorrência é normalmente encontrada na multiplicidade de linguagens de programação existentes, para dar uma idéia existem mais de 2.500 linguagens de programação de computadores, como pode ser constatado nos *sites* da Internet (acessíveis até setembro de 2006):

- *Computer Languages History* (<http://www.levenez.com/lang/>);
- *The Language List* (<http://people.ku.edu/~nkinners/LangList/Extras/langlist.htm>).

Outro ponto a ser observado é que fica evidente a falta de conformidade entre os autores estudados. Nota-se que cada autor utiliza uma nomenclatura particular, às vezes considera alguns aspectos da ação de programação de um computador e outras vezes não considera absolutamente nada.

Além da questão de uso de um conjunto de palavras reservadas, há também a definição de uso de operadores aritméticos utilizados nas operações matemática e os operadores lógicos e relacionais utilizados nas operações lógicas

que necessitam serem estabelecidas segundo um determinado padrão. Assim sendo, segue-se a apresentação da tabela 2 com as comparações entre as várias definições dos operadores aritméticos, lógicos e relacionais:

**Tabela 2 – Paralelo entre Operadores Aritméticos, Lógicos e Relacionais**

GUL	VEL	TER	SZM	PAS	BAS	LPP
+	+	+		+	+	+
-	-	-		-	-	-
*	*	*		*	*	*
/	/	/		/	/	/
DIV	DIV			DIV		DIV
←	←	←	:=	:=	=	←
**					^	↑
E	&			AND	AND	.E.
OU	∨			OR	OR	.OU.
NÃO	¬			NOT	NOT	.NÃO.
>	>	>	>	>	>	>
<	<	<	<	<	<	<
≥	≥	≥	≥	>=	>=	>=
≤	≤	≤	≤	<=	<=	<=
=	=	=	=	=	=	=
≠	≠	≠	≠	<>	<>	<>

Definida assim a proposta de estrutura de codificação a ser utilizada pela LPP cabe apresentar a relação dos comandos propostos (palavras reservadas) e sua classificação sintática dentro da esfera da língua portuguesa, de acordo com a ação a ser executada por cada um dos comandos.

LPP	Classificação Sintática
ATÉ	Preposição
ATÉ QUE	Conjunção (de acordo com seu equivalente inglês - <i>until</i> )
CARACTERE	Substantivo Masculino
CASO	Substantivo Masculino
CONJUNTO	Adjetivo
DE	Preposição
ENQUANTO	Conjunção
ENTÃO	Advérbio
ESCREVA	Verbo (Imperativo Afirmativo)
FAÇA	Verbo (Imperativo Afirmativo)
FIM	Substantivo Masculino
FIM CASO	Substantivo Masculino com Substantivo Masculino
FIM ENQUANTO	Substantivo Masculino com Conjunção
FIM PARA	Substantivo Masculino com Preposição
FIM REGISTRO	Substantivo Masculino com Substantivo Masculino
FIM SE	Substantivo Masculino com Conjunção
FUNÇÃO	Substantivo Feminino
INÍCIO	Substantivo Masculino
INTEIRO	Adjetivo
LEIA	Verbo (Imperativo Afirmativo)
LÓGICO	Adjetivo
PARA	Preposição
PASSO	Substantivo Masculino
PROCEDIMENTO	Substantivo Masculino
PROGRAMA	Substantivo Masculino
REAL	Substantivo Masculino
REGISTRO	Substantivo Masculino
REPITA	Verbo (Imperativo Afirmativo)
SE	Conjunção
SEJA	Interjeição
SENÃO	Conjunção
TIPO	Substantivo Masculino
VAR (variável)	Substantivo Feminino

Uma linguagem de programação de computadores de alto nível (mesmo que seja uma linguagem de documentação como é a LPP) é formada por preposições, conjunções, substantivos, adjetivos, advérbios, verbos e interjeições e devido a estas características recebe o nome de linguagem de programação, que é o conjunto de instruções e regras de composição e encadeamento, pelo

qual se expressam ações que são executadas por um computador. Assim sendo, passa-se a ter um instrumento de comunicação.

## CONCLUSÃO

Este artigo não pode ser por sua natureza conclusivo, uma vez que abre a discussão em apresentar uma proposta de documentação que seja oficializada para uma representação mais formal e homogênea da forma escrita e da documentação textual da linha de raciocínio computacional.

Na análise aqui apresentada sugere-se uma forma mais homogênea de formato escrito que seja útil para os profissionais de desenvolvimento de software e também para as pessoas que estão se inserindo neste mercado de trabalho. A LPP pode ser utilizada também como um elemento de apoio pedagógico no ensino da lógica de programação de computador como também ocorre com a PDL.

Acredita-se que a proposta ora feita pode vir a auxiliar a definição de critérios computacionais mais organizados que permitam a facilidade de uso das linguagens formais de programação de computadores, sua documentação, facilidade na manutenção de sistemas e melhor estrutura no sentido de poder auxiliar auditorias internas e externas no processo de desenvolvimento de sistemas e da aplicação dos conceitos de *engenharia de software*.

## REFERÊNCIAS BIBLIOGRÁFICAS

CAINE, S. H.; GORDON, E. K. PDL: A Tool for Software Design. In PROCEEDINGS OF THE 1975 NATIONAL COMPUTING CONFERENCE, 1975, Anaheim, CA. Montvale, NJ: AFIPS Press, 1975, 271-276.

GUIMARÃES, A. de M.; LAGES, A. de C. **Algoritmos e Estruturas de Dados**. 18. ed. Rio de Janeiro: Livros Técnicos e Científicos, 1994, 216 p.

PRESSMANN, R. S. **Software Engineering**: a practitioner's approach, 6. ed. Ohio: McGraw-Hill, Inc., 2005, 880 p.

SZWARCFITER, J. L.; MARKENZON, L. **Estruturas de Dados e seus Algoritmos**. Rio de Janeiro: Livros Técnicos e Científicos, 1994, 320 p.

TENÓRIO, R. Computadores de Papel: **Máquinas Abstratas para o Ensino Concreto**. 2. ed. São Paulo: Cortez Editora, 2001, 117 p. (Coleção Questões da Nossa Época, 80).

TERADA, R. **Desenvolvimento de Algoritmos e Estruturas de Dados**. São Paulo: Makron Books, 1991, 255 p.

VELOSO, P. et. al. **Estruturas de Dados**. Rio de Janeiro: Campus, 1996, 228 p.

WIKIPEDIA. **Pseudocode**. Disponível em:

<<http://en.wikipedia.org/wiki/Pseudocodes>>. Acesso em: 13 set. 2006.