



Universidade Estadual de Campinas



Centro Nacional de Processamento de Alto Desempenho - São Paulo

Apostila de Treinamento:
SAS Programação I

- Introdução ao SAS: Data Step e Proc Step -

Revisão: 2022

CONTEÚDO

1 – HISTÓRICO	pag.06
2 – CURIOSIDADES	pag.06
3 – DEFINIÇÃO BÁSICA	pag.07
4 – COMERCIALIZAÇÃO	pag.08
4.1 – Módulos SAS	pag.08
4.2 – Soluções Propostas	pag.09
5 – CONCEITOS BÁSICOS	pag.10
6 – AROUVOS SAS	pag.11
7 – ESTRUTURA DOS AROUVOS SAS	pag.12
8 – NOMECLATURA PARA AROUVOS E VARIÁVEIS	pag.13
Ex.1: Nomeclatura	pag.13
9 – SINTAXE DOS COMANDOS	pag.14
Ex.2: Sintaxe	pag.14
10 – EXECUÇÃO	pag.15
10.1 – Modo Interativo (Ambiente WINDOWS)	pag.15
10.2 – Modo Não Interativo (Ambiente de COMANDOS)	pag.20
10.3 – Programa SAS	pag.21
11 – LEITURA E GRAVAÇÃO DE DADOS	pag.22
11.1 – Introdução ao DATA STEP	
11.2 – Comando LIBNAME	pag.24
Ex.3: Declarações de Bibliotecas	pag.24
11.3 – Comando DATA	pag.25
Ex.4: Declaração de arquivo SAS para Gravação de Dados	pag.25
11.4 – Comando INFILE	pag.26
Ex.5: Declaração para leitura de arquivos textos sem formatação SAS	pag.26
11.5 – Comando INPUT	pag.27
11.5.1 – INPUT COLUNADO	pag.28
Ex.6: Leitura de dados com o INPUT colunado	pag.29
11.5.2 – INPUT FORMATADO	pag.30
11.5.3 – Descrição de um Campo Data	pag.31
Ex.7: Leitura de dados com o INPUT formatado	pag.32
Ex.8: Leitura de dados especiais com o INPUT formatado	pag.33
11.5.4 – Controle de Posição de Leitura do INPUT	pag.34
Ex.9: INPUT formatado com indicadores de coluna e de linha	pag.34
11.5.5 – INPUT LISTADO	pag.35
Ex.10: Leitura de dados com o INPUT listado	pag.36
11.5.6 – Comando INFORMAT	pag.37
Ex.11: Utilização do comando INFORMAT	pag.37
11.5.7 – Detalhes do Comando INPUT	pag.38
Ex.12: Todos os tipos de INPUT combinados	pag.39
1º LABORATÓRIO	pag.40
11.6 – Comando SET	pag.42
Ex.13: Declaração para leitura de arquivos com formatação SAS	pag.42
11.7 – Comando RUN	pag.43
12 – ESTRUTURA PROC STEP	pag.44
12.1 – Introdução	pag.44
12.1.1 – Comando VAR	pag.45
12.1.2 – Comando BY	pag.46
12.1.3 – Comando TITLE	pag.47
12.1.4 – Comando FOOTNOTE	pag.47
12.2 – Procedimento PRINT	pag.48
Ex.14:Procedimento PRINT 1	pag.49
Ex.15:Procedimento PRINT 2	pag.49
12.3 – Procedimento SORT	pag.50
Ex.16:Procedimento SORT	pag.51
13 – AROUVO DE MENSAGENS - "SAS Log"	pag.52
Ex.17: Exemplo de LOG sem problemas na execução 1	pag.52
Ex.18: Exemplo de LOG com problemas na execução 2	pag.53
Ex.19: Exemplo de LOG com problemas na execução 3	pag.54
2º LABORATÓRIO	pag.55

14 – ESTRUTURA DATA STEP – Comandos Básicos de Programação	pag.56
14.1 – Comando de Alocação (=)	pag.56
Ex.20: Comando de Alocação	pag.57
14.2 – Comando RETAIN	pag.58
Ex.21: Comando RETAIN	pag.58
14.3 – Comando IF-THEN/ELSE	pag.59
Ex.22: Comando IF (Somente Selecionar de Dados)	pag.60
Ex.23: Comando IF (Operadores de Comparação)	pag.61
Ex.24: Comando IF (Operadores Lógicos)	pag.62
Ex.25: Comando IF (Operadores Lógicos)	pag.62
Ex.26: Comando IF (Verdadeiro ou Falso)	pag.62
14.4 – Comando WHERE	pag.63
Ex.27: Comando WHERE	pag.64
14.5 – Comando DO/END	pag.65
Ex.28: Comando DO/END	pag.65
14.6 – Comando LENGTH	pag.66
Ex.29: Comando LENGTH	pag.66
14.7 – Comando DO/END Iterativo	pag.67
Ex.30: Comando DO/END Iterativo	pag.68
14.8 – Comando DO WHILE	pag.69
Ex.31: Comando DO WHILE	pag.69
14.9 – Comando DO UNTIL	pag.70
Ex.32: Comando DO UNTIL	pag.70
14.10 – Comandos DROP e KEEP	pag.71
Ex.33: Comandos DROP e KEEP	pag.72
14.11 – Opções de Arquivos SAS	pag.73
Ex.34: Opções: DROP= e KEEP=	pag.73
14.12 – Comando DELETE	pag.74
Ex.35: Comando DELETE	pag.74
14.13 – Comando OUTPUT	pag.75
Ex.36: Comando OUTPUT: Salvar Registros	pag.76
Ex.37: Comando OUTPUT: Direcionar e Salvar Registros	pag.76
3º LABORATÓRIO	pag.77
15 – FUNÇÕES DO SAS	pag.81
15.1 – Introdução	pag.81
15.2 – Funções de Tratamento de Caractere	pag.82
15.2.1 – Funções UPCASE, LOWCASE e PROPCASE	pag.82
15.2.2 – Função COMPRESS	pag.82
15.2.3 – Função CATX	pag.83
15.2.4 – Função COUNT	pag.83
Ex.38: Funções Caracteres: CATX, COMPRESS, PROPCASE e COUNT	pag.84
15.2.5 – Função SUBSTR= ou =SUBSTR	pag.85
15.2.6 – Função TRANWRD	pag.85
15.2.7 – Função FIND	pag.86
15.2.8 – Função SCAN	pag.87
15.2.9 – Função LENGTH	pag.87
Ex.39: Funções Caracteres: FIND e SCAN	pag.88
Ex.40: Funções Caracteres:TRANWRD, COMPRESS, SUBSTR e LENGTH	pag.89
15.3 – Funções Numéricas	pag.90
15.3.1 – Função INT	pag.90
15.3.2 – Função ROUND	pag.90
Ex.41: Funções Numéricas: INT e ROUND	pag.91
15.4 – Funções Estatísticas Descritivas	pag.92
15.4.1 – Função SUM	pag.92
15.4.2 – Função MEAN	pag.92
Ex.42: Funções Estatísticas: SUM e MEAN	pag.93
15.5 – Funções de Tratamento de Data e Hora	pag.94
15.5.1 – Função MDY	pag.94
15.5.2 – Funções: DAY, MONTH, YEAR	pag.94
15.5.3 – Função TODAY e DATE	pag.94

15.5.4 – Função HMS	pag.95
15.5.5 – Função TIME	pag.95
15.5.6 – Funções: HOUR, MINUTE, SECOND	pag.95
15.5.7 – Função INTCK	pag.96
Ex.43: Funções de Tratamento de Data: MDY, TODAY e INTCK	pag.97
4º LABORATÓRIO	pag.98
16 – COMANDOS DE CONFIGURAÇÃO	pag.99
16.1 – Comando OPTIONS	pag.99
Ex.44: Comando OPTIONS 1	pag.100
Ex.45: Comando OPTIONS 2	pag.100
16.2 – Comando LABEL	pag.101
Ex.46: Comando LABEL	pag.101
16.3 – Comando FORMAT	pag.102
Ex.47: Comando FORMAT	pag.103
16.4 – Comando ODS	pag.104
Ex.48: Comando ODS	pag.105
5º LABORATÓRIO	pag.106
17 – PROCEDIMENTOS ESTATÍSTICOS BÁSICOS	pag.107
17.1 – Procedimento MEANS	pag.107
Ex.49: Procedimento MEANS (Padrão)	pag.108
Ex.50: Procedimento MEANS (Com Opções)	pag.108
Ex.51: Procedimento MEANS (Comando CLASS)	pag.108
Ex.52: Procedimento MEANS (Comando BY)	pag.109
Ex.53: Procedimento MEANS (Comando CLASS2)	pag.109
Ex.54: Procedimento MEANS (Comando OUTPUT)	pag.110
17.2 – Procedimento FREQ	pag.111
Ex.55: Procedimento FREQ (Sem Cruzamento)	pag.112
Ex.56: Procedimento FREQ (Com Cruzamento)	pag.113
Ex.57: Procedimento FREQ (Cruzamento Listado)	pag.114
Ex.58: Procedimento FREQ (Opções do Comando TABLE)	pag.114
Ex.59: Procedimento FREQ (Opção OUT)	pag.115
17.3 - Procedimento TABULATE	pag.116
Ex.60: Procedimento TABULATE (Concatenação de Elementos)	pag.118
Ex.61: Procedimento TABULATE (Cruzamento de Elementos)	pag.118
Ex.62: Procedimento TABULATE (Três Dimensões 1)	pag.119
Ex.63: Procedimento TABULATE (Rotulação de Elementos: =)	pag.120
Ex.64: Procedimento TABULATE (Formatação de Elementos: *f=)	pag.121
Ex.65: Procedimento TABULATE (Elemento ALL e PCTN)	pag.122
Ex.66: Procedimento TABULATE (Três Dimensões 2)	pag.123
18 - RECURSO DE FORMATAÇÃO ESPECIAL	pag.124
18.1 - Procedimento FORMAT	pag.124
Ex.67: Procedimento FORMAT	pag.125
6º LABORATÓRIO	pag.126
19 - COMBINAÇÃO DE ARQUIVOS SAS	pag.128
19.1 - Concatenação de Arquivos	pag.128
Ex.68: Concatenação de Arquivos	pag.129
19.2 - Combinação Ordenada de Arquivos (Match-Merging)	pag.130
Ex.69: Combinação Ordenada de Arquivos	pag.131
19.3 – Interseção de Dados, opção de arquivos (IN=)	pag.132
Ex.70: Interseção de Dados	pag.133
7º LABORATÓRIO	pag.134
20 - DADOS MICROSOFT EXCEL	pag.135
20.1 - Modo Assistente: “Import Wizard” ou “Export Wizard”	pag.135
20.2 - Modo Programação: PROC IMPORT ou PROC EXPORT	pag.138
Ex.71: Procedimento IMPORT	pag.138
Ex.72: Procedimento EXPORT	pag.138
20.3 - Principais Problemas	pag.139
21 - REFERÊNCIAS	pag.140

Tipografia utilizada na apostila

Na apresentação de alguns comandos do SAS, foram utilizados símbolos gráficos que identificam, na sintaxe do comando, a característica de ser opcional ou obrigatório:

< *característica* > É **obrigatório** a informação no comando;

[*característica*] É **opcional** a informação no comando.

Exemplo: Utilização de procedimentos SAS

```
PROC <tipo> [opção1 opção2 opção3 . . . opção] ;  
    [comando1] ;  
    [comando2] ;  
    [WHERE <expressão lógica>] ;  
    . . .  
    [comandon] ;  
RUN ;
```

1 - HISTÓRICO

Década de 60;

North Caroline University;

Censo Agro-pecuário Norte-americano;

Conjunto de sub-rotinas (FORTRAN, IBM/370);

SAS Institute Inc. (1976);

2 - CURIOSIDADES

<http://www.sas.com>

<http://v9doc.sas.com>

SAS OnDemand for Academics – SAS Software in the cloud, for free;

https://www.sas.com/pt_br/software/on-demand-for-academics.html

Mais de 80.800 Instalações, em 150 países;

88 das 100 maiores empresas do mundo utilizam SAS.
(Fortune 500 Global List - 2021);

Receita Bruta em 2021 - \$3,1 bilhões;

2021 – mais de 11.764 funcionários;

SAS é a maior empresa de software do mundo de capital privado;
“SAS is the world's largest privately held software company”

No Brasil desde 1996, possui com aproximadamente 200 clientes;

O SAS adaptou o software para alguns idiomas: Chinês, Francês, Alemão, Hebreu, Italiano, Japonês, Russo, Espanhol, Polonês, Húngaro, Sueco, Coreano, etc.

3 - DEFINIÇÃO BÁSICA

O SAS é um software integrado para análise de dados, que consiste de vários produtos que permitem:

- Acesso e recuperação de dados em qualquer formato;
- Gerenciamento de arquivos;
- Análise estatística;
- Acesso a Banco de Dados (ORACLE, DB2, TERADATA, etc);
- Geração de gráficos (gif, jpg, bmp, etc);
- Geração de relatórios (html, pdf, ps, etc);
- Desenvolvimento de aplicativos;
- Soluções de negócios (Análise de Risco, “Data Mining”).

É um software de grande portabilidade, podendo operar em diversos ambientes computacionais:

- Mainframes IBM (CMS, MVS, OS, OS/390, Z/OS);
- Micros PC (DOS, Windows, OS/2);
- UNIX (Solaris, AIX, HP-UX, Irix, Linux).

4 - COMERCIALIZAÇÃO

4.1 - Módulos do SAS

<u>SAS/BASE</u>	Módulo básico, obrigatório em toda instalação;
<u>SAS/STAT</u>	Módulo estatístico;
<u>SAS/GRAPH</u>	Módulo gráfico (Histogramas, plots, ...);
<u>SAS/OR</u>	Módulo de análise e pesquisa operacional (Programação linear, Análise de Caminho Crítico);
<u>SAS/QC</u>	Módulo para análise de controle de qualidade;
<u>SAS/ETS</u>	Módulo de econometria (Séries Temporais, Modelagem de Equações Simultâneas);
<u>SAS/IML</u>	Módulo para análise e operação de matrizes;
<u>SAS/ACCESS</u>	Módulo para acesso aos diversos tipos de Banco de Dados;
<u>SAS/CONNECT</u>	Módulo para conexão entre ambientes operacionais heterogêneos;
<u>SAS/AF</u>	Módulo para desenvolvimento de aplicações;
<u>SAS/FSP</u>	Módulo para facilitar o acesso a arquivos com programação de telas.

4.2 – Soluções Propostas

[Analytics](#)

[Business Analytics](#)

[SAS® Cloud](#)

[Customer Intelligence](#)

[Fraud & Security Intelligence](#)

[Performance Management](#)

[Risk Management](#)

[Índice de Produtos de A-Z](#)

5 - CONCEITOS BÁSICOS

A funcionalidade do Sistema SAS foi construída em torno de quatro ideias básicas no tratamento de dados:

Acessar dados;
Administrar dados;
Analisar dados;
Apresentar dados.

- Estrutura de um programa SAS:

JOB SAS Arquivo com um programa SAS (É um conjunto de DATA Step's e PROC Step's);

DATA STEP Divisão lógica de um programa SAS, no qual se cria e altera um, ou vários arquivos SAS (Data Set's);

PROC STEP Divisão lógica de um programa SAS, no qual se analisa e manipula os dados contidos num arquivo SAS (Data Set's);

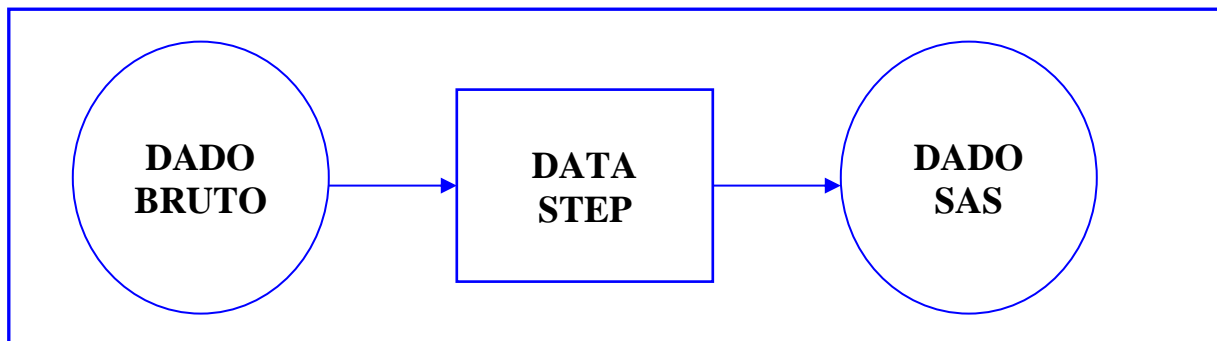
- Arquivo SAS:

DATA SET Arquivo de dados com estrutura SAS, contendo diversos registros (linhas de dados);

OBSERVAÇÃO Registro ou linha, de um arquivo de dados SAS.

6 - ARQUIVOS SAS

Todos os dados devem estar armazenados em arquivos com estrutura SAS (DATA Set's), para serem analisados pelos procedimentos do SAS (PROC's).



Os arquivos SAS são, a princípio, armazenados em uma área temporária definida pelo sistema SAS – biblioteca **WORK**, mas também podem ser armazenadas em áreas definidas pelo usuário (bibliotecas permanentes).

Em um único programa SAS (“Job SAS”), vários arquivos podem ser abertos e analisados.

Os arquivos SAS, depois de criados, podem ser analisados pelos procedimentos SAS.

7 - ESTRUTURA DOS ARQUIVOS SAS

O arquivo SAS é um conjunto de dados organizados num formato de tabela. Linhas divididas em colunas.

	NOME						SEXO	IDADE	ALTURA								
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	C	A	R	L	O	S			M		1	2		1	.	7	0
2	G	I	L	B	E	R	T	O	m		2	5		1	,	6	5
3	R	i	c	a	r	d	o		m		3	0		1	,	8	0
4	l	a	u	r	a				f		2	3		1	.	7	1
5	M	i	r	i	a	m			F		3	4		1	.	6	8
6	L	u	i	s					M		4	4		1	.	7	8
7	A	l	i	C	e				F		3	2		1	7	1	0
8	m	A	r	c	i	a			f		2	5		1	,	5	9

A combinação de colunas nas tabelas formam os campos de dados, que são chamadas de **variáveis**:

- As variáveis são formadas por uma ou mais colunas;
- Cada coluna representa um caractere alfanumérico;
- Cada variável recebe um nome;
- Para o SAS, só existem dois tipos de variáveis:

caractere (até 32.767 caracteres – 32.767 Bytes)

numérica ([Representação IEEE](#) – 8 Bytes)

OBS: Até 32.767 variáveis (Até o release SAS 9.1)

As linhas na tabela são chamadas de **observações**, ou registros, ou simplesmente, linhas. Não existe limite para o número de observações.

Um Arquivo SAS é dividido em duas partes:

Descritora

Contém a descrição do arquivo: Tamanho do arquivo, número de registros, nome, tipo, tamanho e formato de variáveis, etc;

Dados

Contém os dados do arquivo.

8 - NOMECLATURA PARA ARQUIVOS E VARIÁVEIS

Todos os nomes de arquivos e variáveis deverão seguir três regras básicas:

- Possuir de 1 a 32 caracteres;
- Começar com letra (A-Z ou a-z, é indiferente);
- Pode continuar com números, letras (maiúsculas ou minúsculas) ou somente o caractere especial “_” (grifo, sublinhar).

Variáveis no SAS, podem possuir até 6 atributos, sendo:

Obrigatórios: Nome (NAME), Tipo (TYPE), Tamanho (LENGTH);

Opcionais: Rótulo (LABEL), Formato de leitura (INFORMAT), Formato de impressão (FORMAT);

Ex.1: Nomeclatura

Nome do arquivo SAS que será gerado (Gravação)

```
DATA EXEMPLO1 ;  
  INFILE "C:\curso\sas\CADASTRO.TXT";  
  INPUT primeiro_nome $ 1-8 Sexo $ 10 Idade 13-14 ALTURA 16-19 ;  
RUN;
```

Nome da tabela com os dados que serão acessados (Leitura)

Identificação da posição (colunas) do dado será lido

Identificação do tipo de dado que será lido

Identificação do tipo de dado que será lido

9 - SÍNTAXE DOS COMANDOS

Todo comando SAS começa com uma **palavra-chave** de identificação (colorido em “azul” no ambiente “Windows” do SAS), e termina com o **ponto e vírgula (;)**.

- Os comandos podem começar e terminar em qualquer parte da linha;
- Um comando pode se estender por diversas linhas;
- Vários comandos podem ficar numa mesma linha.

OBS: Ou seja, o importante é a posição do “;” , que irá indicar a finalização do comando (É OBRIGATÓRIO!!!).

Ex.2: Sintaxe

```
DATA EXEMPLO1 ;  
  INFILE " C:\curso\sas\CADASTRO.TXT" ;  
  INPUT  NOME $ 1-8 SEXO $ 10  
         IDADE $ 13-14 ALTURA 16-19 ;  
  
RUN;  
  
PROC PRINT DATA=EXEMPLO1 ;  
RUN;  
  
PROC FREQ DATA=EXEMPLO1 ; TABLES IDADE*ALTURA ; RUN ;
```

10 - EXECUÇÃO

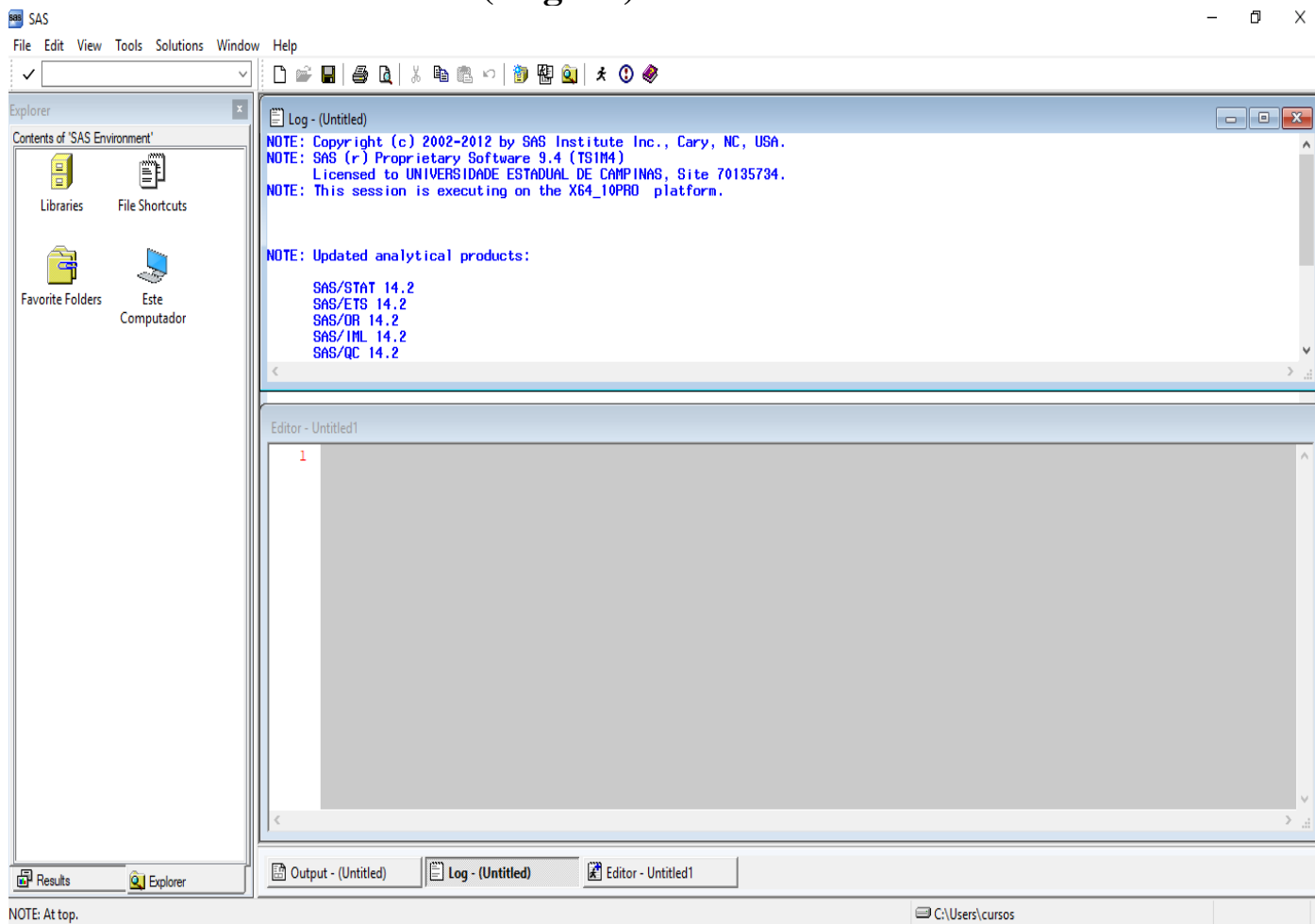
Um programa SAS pode ser executado de três maneiras diferentes:

- Modo interativo (Ambiente SAS Windows Clássico);
- Modo interativo (Ambiente SAS Enterprise Guide);
- Modo não interativo;
- Modo “batch”.

10.1 - Modo Interativo (Ambiente WINDOWS)

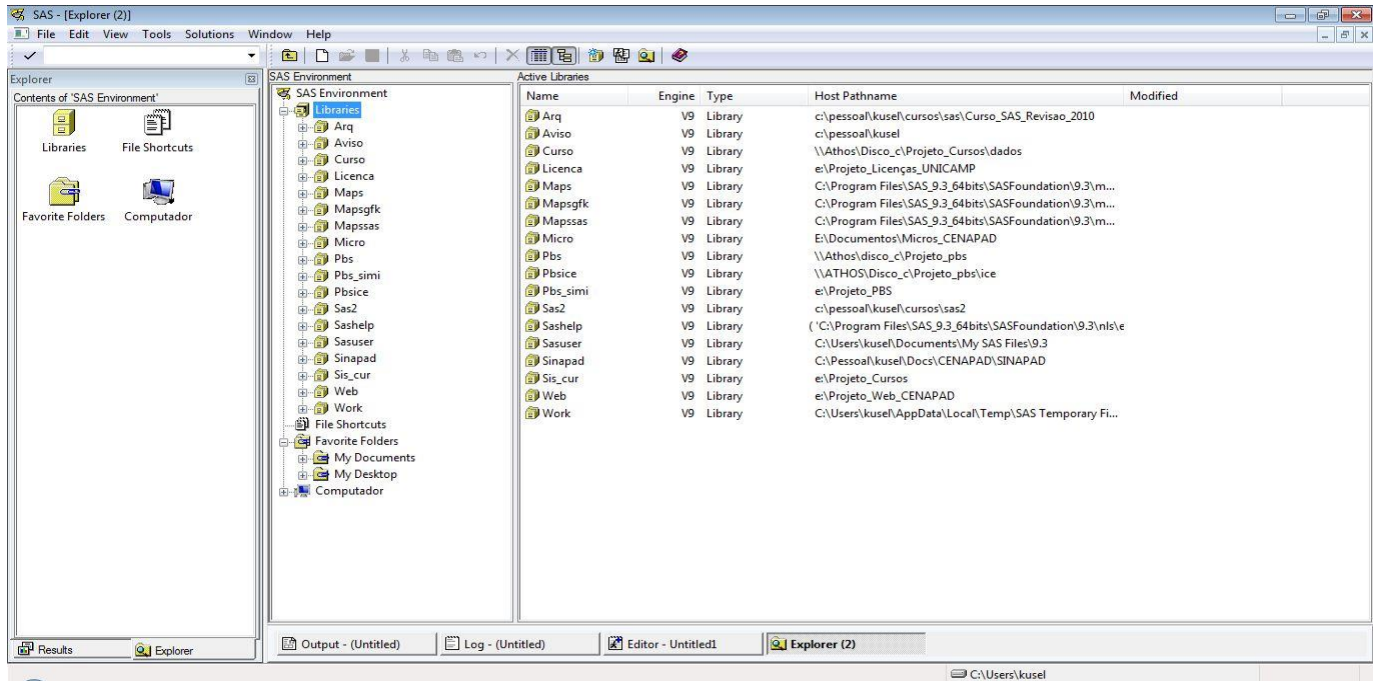
Localize o ícone do SAS na área de trabalho do “windows” ou, ative o ambiente SAS, seguindo o roteiro abaixo:

Iniciar → SAS → SAS 9.4(English)



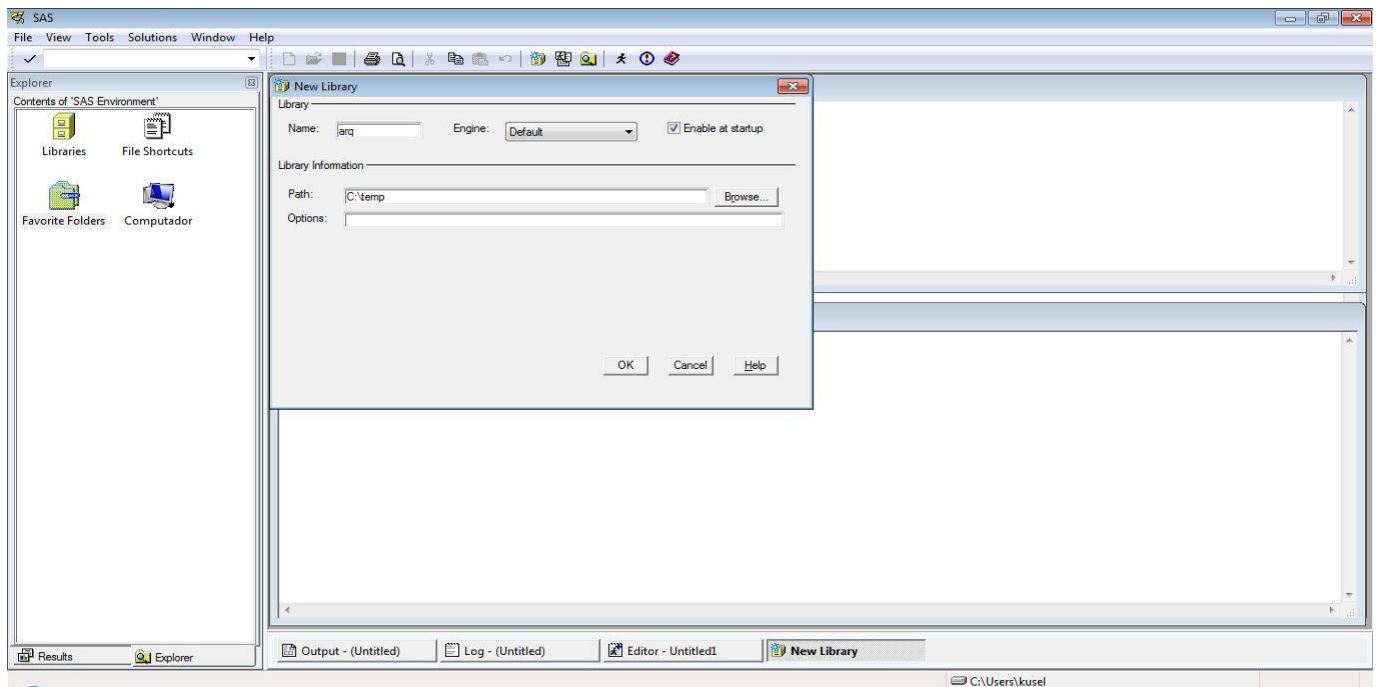
SAS Explorer – Libraries

View → Explorer



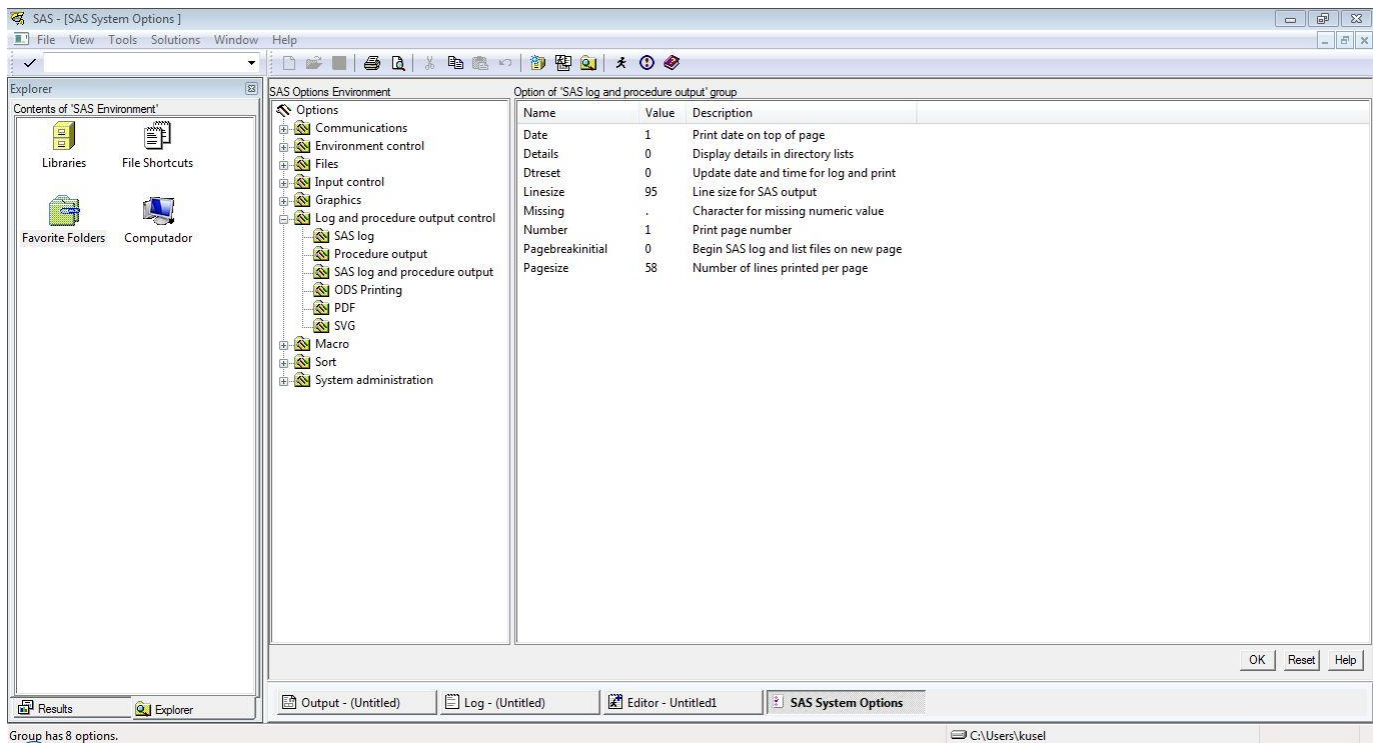
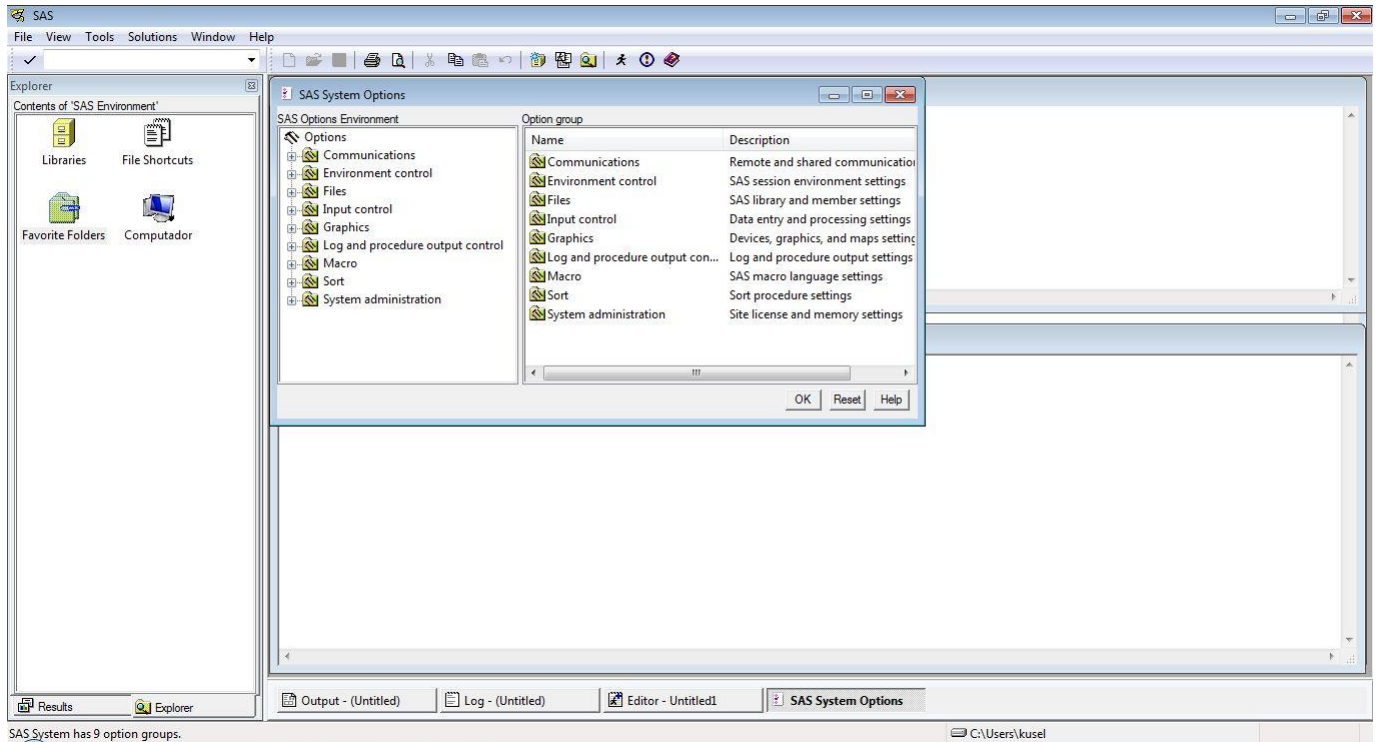
SAS Explorer – New Library

Tools → New library



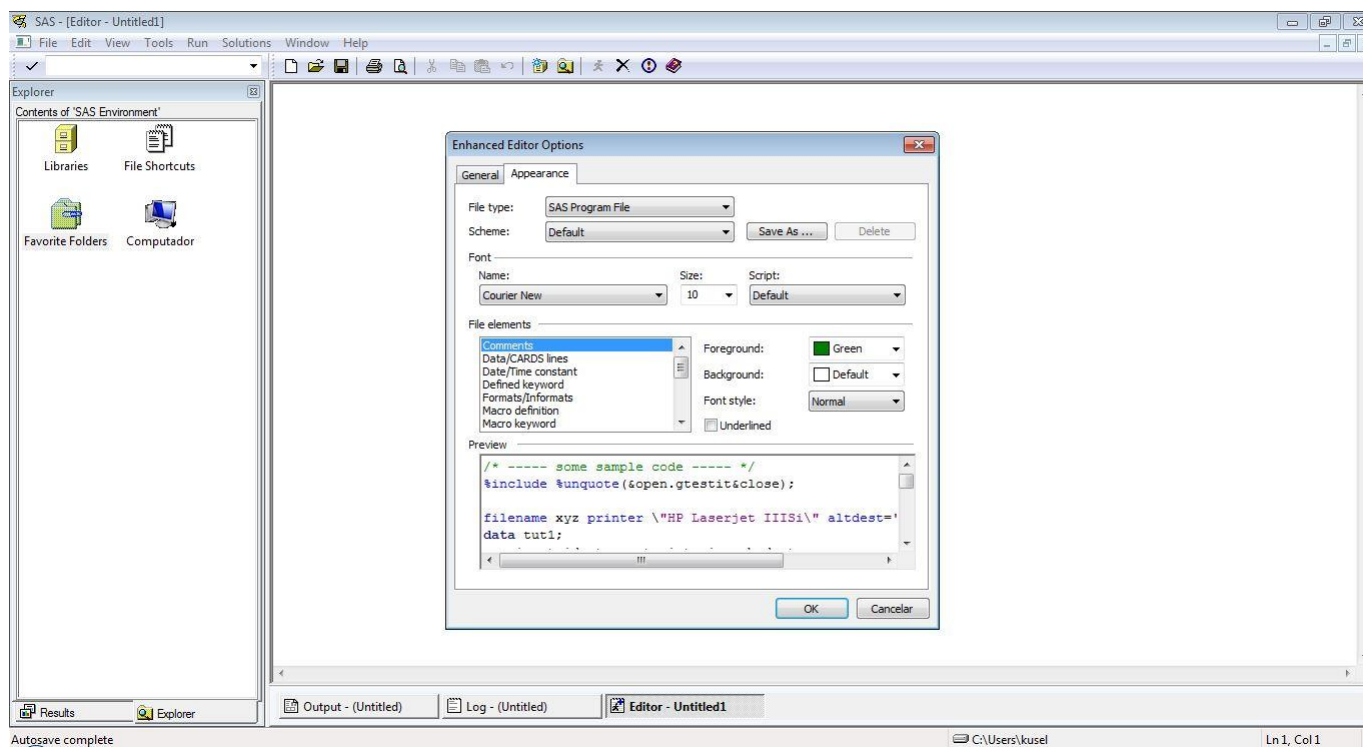
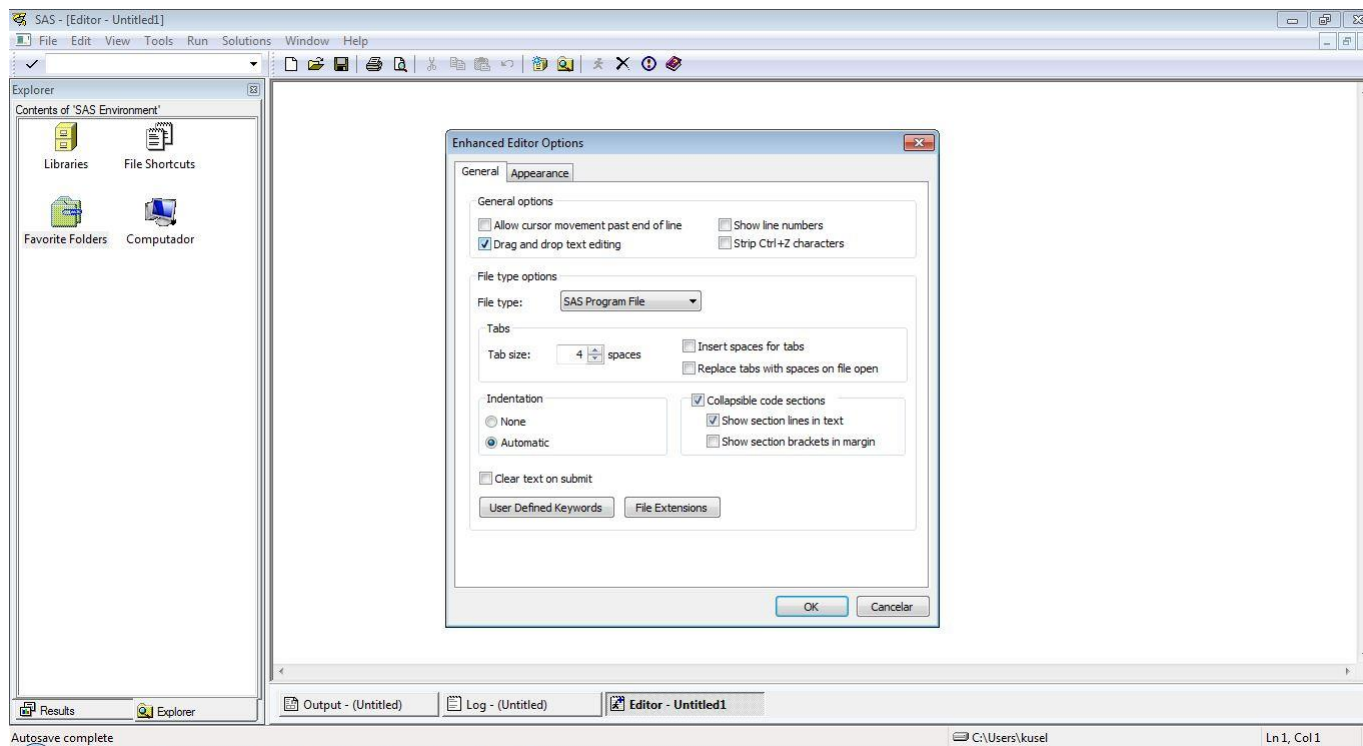
JANELA DE OPÇÕES DE CONFIGURAÇÃO

Tools → Options → System...



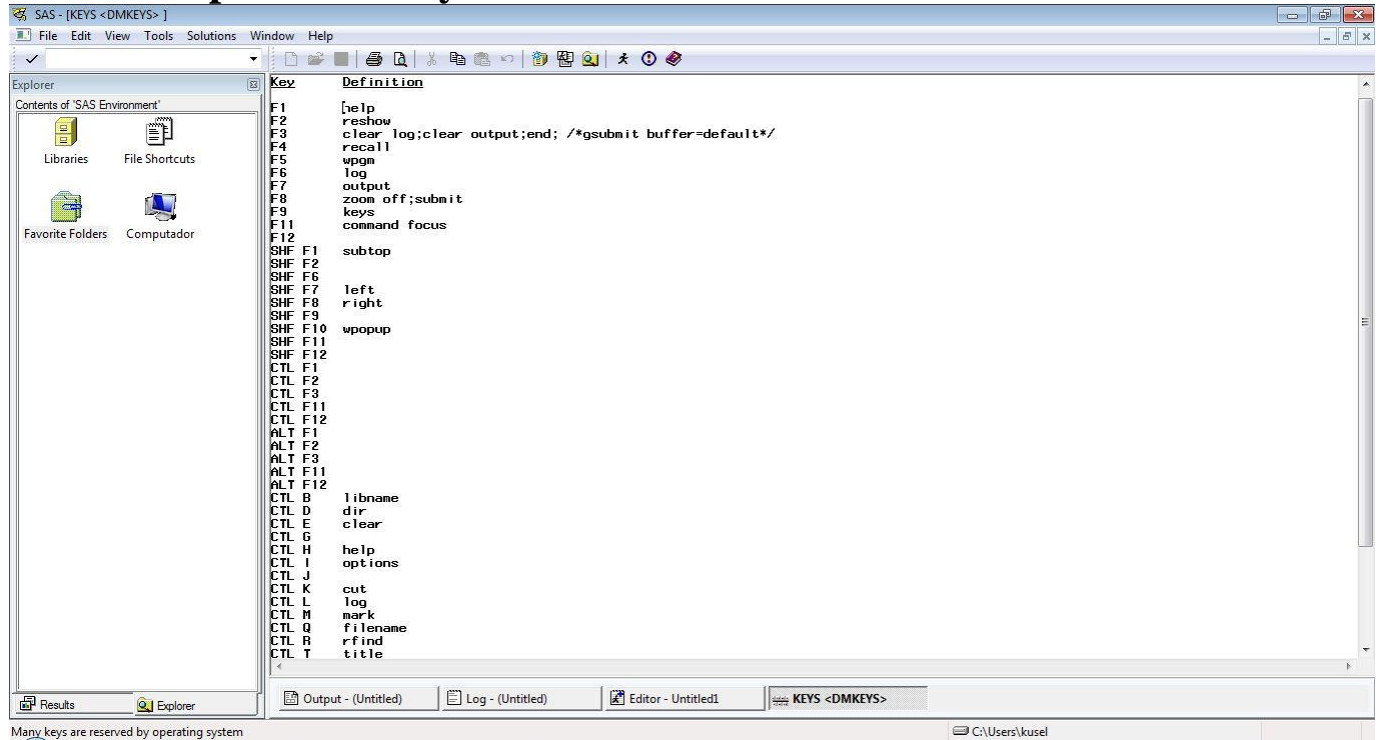
OPÇÕES DE CONFIGURAÇÃO DO EDITOR

Tools → Options → Enhanced Editor ...



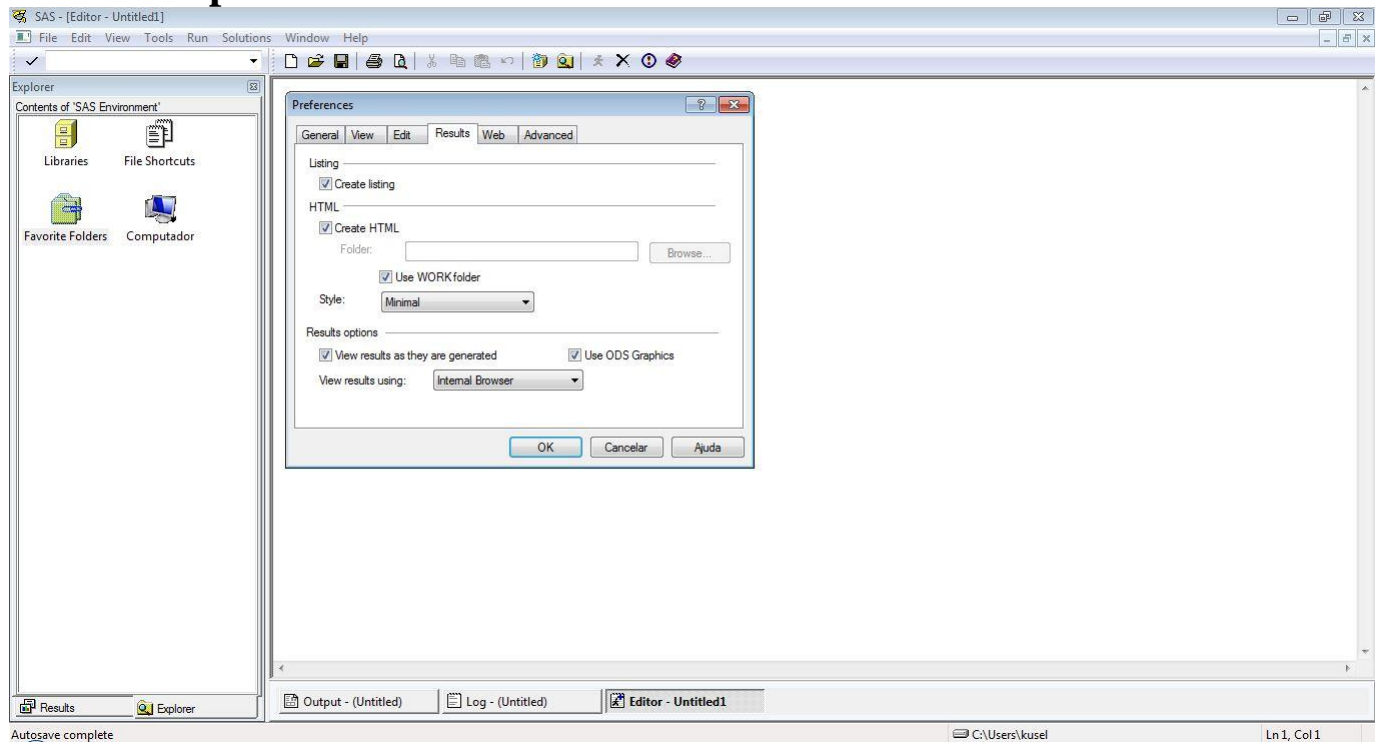
CONFIGURAÇÃO DE TECLADO

Tools → Options → Keys



RELATÓRIOS NO FORMATO TEXTO E/OU HTML

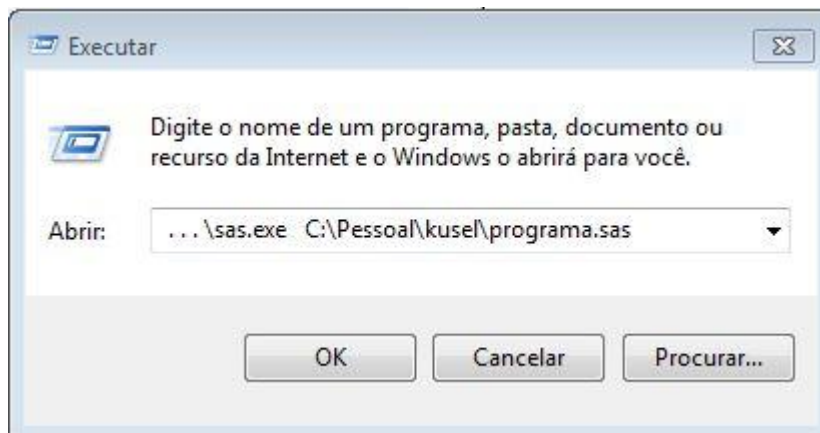
Tools → Options → Preferences ... → Results



10.2 - Modo Não Interativo (Ambiente de COMANDOS)

É uma maneira simples e rápida de execução de um programa SAS. O programa pode ser editado em qualquer editor, mas deverá ser e salvo em modo texto. Para executá-lo, basta ativar a janela de comandos do windows (“Executar ou cmd”):

Iniciar ==> Executar...



No diretório do executável SAS, serão gerados dois arquivos com o mesmo nome do programa:

<programa>.**lst** Possui os resultados do programa;

<programa>.**log** Possui mensagens da execução do programa.

OBS: Sempre analise o arquivo com extensão **log**. Se não for criado o arquivo com a extensão **lst**, com certeza existirão erros no programa, que poderão ser identificados no arquivo **log**.

10.3 – Programa SAS

Um programa SAS é uma combinação do uso de comandos das estruturas de programação básica do SAS e comandos especiais:

- **Comandos livres** - Utilizados em qualquer parte do programa;
- **Comandos genéricos** - Utilizados em DATA e PROC Step's;
- **Comandos da estrutura DATA Step** - Utilizados somente em DATA Step's;
- **Comandos da estrutura PROC Step** - Utilizados somente em PROC Step's;
- **Comandos de MACRO SAS**;
- **Comandos de Orientação ao Objeto SAS**.

A submissão de um programa SAS para execução, obedece a duas fases:

1ª) Compilação: Análise da sintaxe dos comandos, busca e definição de bibliotecas e arquivos, definição de variáveis,

2ª) Execução: Execução do programa, **por “Step”**, obedecendo a uma sequência lógica. Todos os step's serão executados. **Um “step” com erro, para de processar, e dá sequência ao próximo “step”**.

Os comandos de um programa SAS, se dividem em:

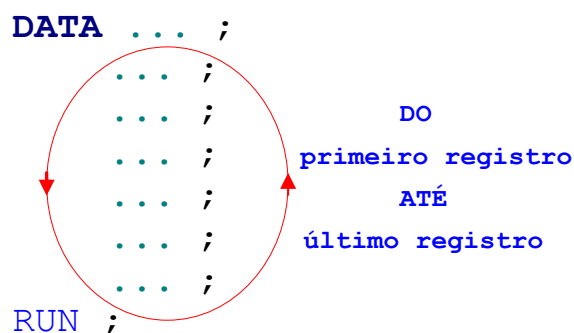
Comandos de Declaração Comandos cuja funcionalidade ocorre durante a fase de compilação, estabelecendo **uma única ação**;

Comandos de Execução Comandos cuja funcionalidade ocorre durante a fase de execução, permitindo e possibilitando que a sua ação ocorra **mais de uma vez**.

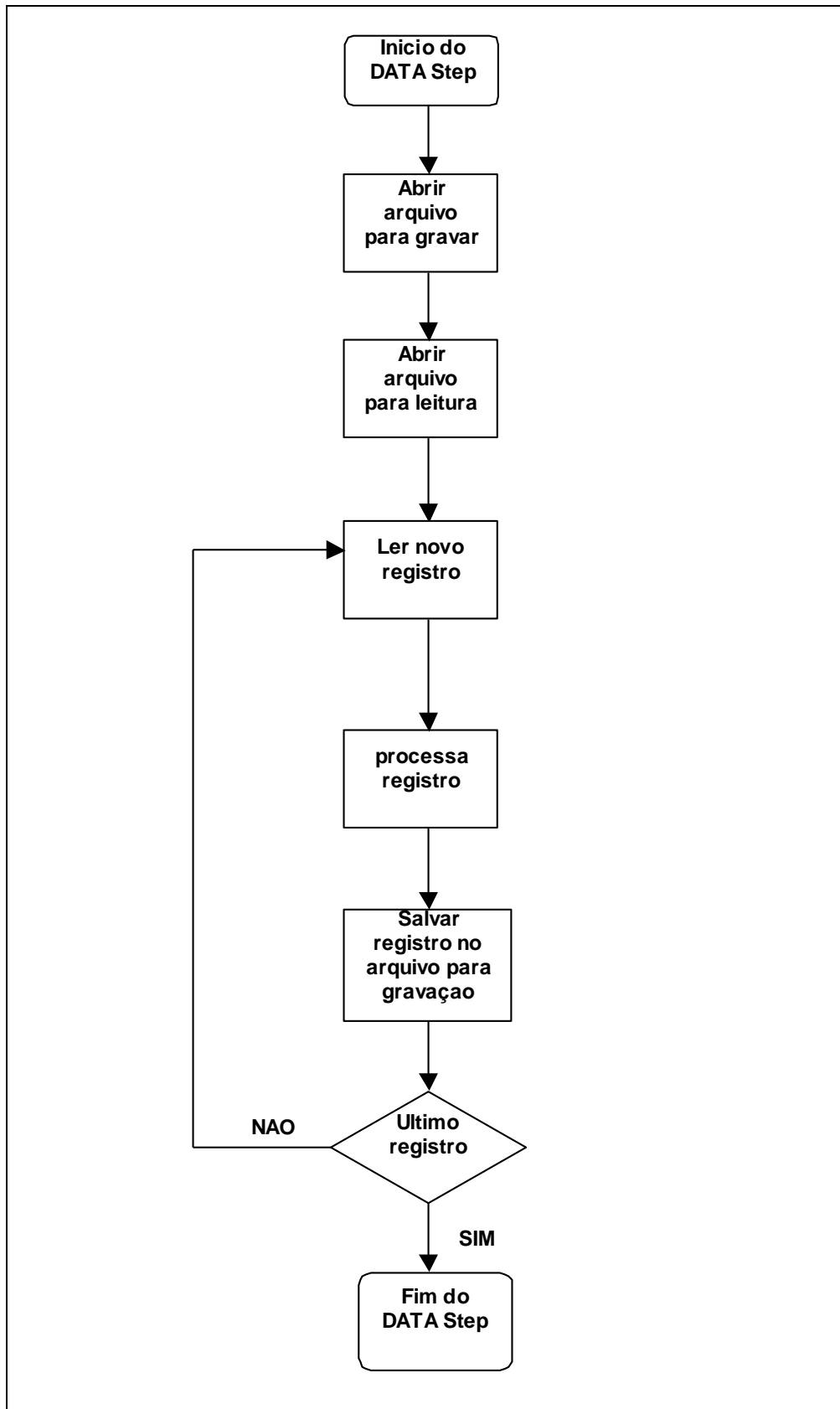
11 - LEITURA E GRAVAÇÃO DE DADOS

11.1 – Introdução ao DATA STEP

- É a seção de um programa SAS aonde se organizam e administram os dados, utilizando comandos de programação do modulo SAS/BASE e de acordo com uma lógica de processamento;
- A maioria dos comandos utilizados em uma estrutura DATA,Step, só existem e, só podem ser executados, nessa estrutura;
- Os comandos são posicionados em uma sequência lógica, ou seja, a execução de um determinado comando pode depender da execução do comando anterior (lógica de processamento);
- Os arquivos processados em um DATA Step, normalmente são processados de maneira sequencial, ou seja, a leitura dos dados é feita registro após registro;
- Quando se utiliza um DATA Step para ler um arquivo de dados, internamente ele representará um ciclo, um “loop”, que repetirá todos os comandos do DATA Step para cada leitura de um novo registro do arquivo, até encontrar o último registro, finalizando o DATA Step e seguindo para o próximo “step”;



Fluxo de Execução



11.2 - Comando LIBNAME

- **Tipo:** Comando de declaração;

- **Uso:** Livre;

- Comando que declara uma ou mais áreas (pasta, filesystem, diretório) ou locais aonde serão armazenados os arquivos SAS (Data Set's). Essas localizações são definidas como **bibliotecas** para o SAS e serão utilizadas para compor o nome dos arquivos.

- O nome de um arquivo SAS é composto por duas partes separadas por um “.”. A primeira parte é opcional, **o nome da biblioteca**, a segunda parte é obrigatória, **o nome do arquivo**.

[biblioteca.]<nome do arquivo>

- O comando LIBNAME, normalmente, é colocado no início de um programa SAS, sendo necessário executá-lo apenas uma única vez durante uma sessão SAS para a definição da biblioteca.

LIBNAME <biblioteca> "<diretório>" ;

biblioteca Palavra de no mínimo 1 e no máximo 8 caracteres.

diretório Nome de um diretório que **já exista** no seu ambiente.

OBS: O SAS define uma biblioteca, temporária, de nome **work**, que será utilizada automaticamente se não for especificada a primeira parte do nome do arquivo.

Ex.3: Declarações de bibliotecas

```
LIBNAME IN "C:\ALUNO" ;  
  
libname teste "d:\" ;  
  
LibName a 'G:\dados' ;  
  
libname _all_ list ; /* Limpa todas as declarações de bibliotecas */  
  
Libname _all_ clear ; /* Lista, no log, todas as bibliotecas disponíveis */
```


11.3 - Comando DATA

- **Tipo:** Comando de declaração;

- **Uso:** Data Step;

- Comando que **sempre** inicia uma estrutura DATA Step;

- Comando que declara e abre um, ou vários novos arquivos SAS (DATA Sets), **vazios**, somente para **gravação** de dados.

DATA [biblioteca.]<nome do arquivo> ... ;

Ex.4: Declarações de arquivos SAS para gravação de dados.

```
DATA "d:\paulo\dados\cadastro.sas7bdat" ;

DATA endereço ;

DATA ARQ1  arq2  ARQ3 ;

libname ENT "d:\paulo\dados" ;
data  ent.cadastro ;

DATA ;  /* O nome do arquivo será gerado automaticamente
        pelo SAS, na biblioteca WORK: DATA001, DATA002,
        ... */

DATA _NULL_ ; /*Não será gerado um novo arquivo SAS */
```

OBS: Não é obrigatório especificar a biblioteca no nome do arquivo (a primeira parte), neste caso, o SAS irá utilizar a biblioteca padrão, também chamada de biblioteca temporária (**WORK**).

11.4 - Comando INFILE

- **Tipo:** Comando de declaração e execução;

- **Uso:** Data Step;

- Comando que identifica e abre qualquer arquivo de dados, **somente para leitura**.

- Abre qualquer tipo de arquivo e interpreta todos os dados simplesmente como texto alfanumérico;

- Este comando deve ser posicionado após o comando DATA e necessita, sempre, e explicitamente, de um comando para leitura dos registros, o comando INPUT.

INFILE <"arquivo de dados"> [opções] ;

Opções:	LRECL=	Tamanho do registro;
	PAD	Completa de brancos os registros;
	MISSOVER	Previne o SAS de lê um novo registro para completar os dados de uma variável;
	DLM=	Especifica o delimitador entre os campos de dados de um arquivo delimitado;
	DSD	Determina que dois delimitadores juntos, representam um campo;

Ex.5: Declaração de leitura de arquivos textos sem formatação SAS

```
INFILE "C:\DADOS\ARQ1.TXT" ;  
INFILE "A:\CADASTRO.PRN" missover;  
infile 'C:\SAS\CAD.DAT' LRECL=3280 ;
```

```
DATA EXEMPLO ;  
  INFILE 'C:\TESTE\DADOS.TXT' lrecl=450 MISSOVER ;  
  ... ;  
  ... ;  
  ... ;  
RUN ;
```

11.5 - Comando INPUT

- **Tipo:** Comando de execução;

- **Uso:** Data Step;

- O comando INPUT serve para descrever ao SAS como estão armazenados e organizados os dados em arquivos que foram especificados pelo comando INFILE;

- Todo arquivo aberto pelo comando INFILE, **sempre** necessitará de um comando INPUT para ler o seu conteúdo;

- O comando INPUT pode ler um, ou vários registros, de maneira sequencial, registro após registro, até encontrar um registro especial, que identifica o fim de arquivo, “EOF” (End Of File), e irá atribuir nomes (**Variáveis**) para cada combinação de colunas de um registro;

- Junto com o comando INPUT, todos os comandos que vierem em seguida, serão executados repetidas vezes até o registro que indica o fim de arquivo, “EOF” (o ciclo do DATA Step, mencionado na pag.22);

- Três tipos básicos de comando INPUT:

- INPUT COLUNADO

- INPUT FORMATADO

- INPUT LISTADO

11.5.1 - INPUT COLUNADO

- Lê os campos de dados, atribuindo-lhes um nome (**NAME**), um tipo (**TYPE**), e especificando a sua posição inicial e final na linha de dados do arquivo, o tamanho (**LENGTH**). Os campos de dados no arquivo devem estar alinhados, fixos em colunas, para permitir a leitura correta dos dados.

INPUT <variável> [\$] <início>-<fim> [.decimal] ... ;

variável	Nome do campo de dados.
\$	Caractere opcional que indica que a variável só irá conter dados alfanuméricos. Sem o \$, significa que a variável será numérica.
início	Valor que indica a posição inicial da variável na linha de dados.
fim	Valor que indica a posição final da variável na linha de dados.
.decimal	Valor opcional, indica o número de posições decimais de uma variável numérica.

OBS: Durante a fase de compilação, através da análise do comando INPUT, o SAS irá definir o NOME, o TIPO e o TAMANHO (pelas informações de coluna inicial e final) das variáveis.

Ex.6:Leitura de dados com o INPUT COLUNADO

Arquivo: CADASTRO2.DAT

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
C	A	R	L	O	S			M		1	8		1	7	0	6	8	.	0
C	A	R	L	A				F		2	6		1	7	8	7	2	.	0
S	I	L	V	I	A			f		2	1		1	6	5	6	0	.	5
M	A	R	C	I	O			m		3	2		1	8	2	8	5	,	4

```
DATA FICHA ;
  INFILE "C:\curso\sas\CADASTRO2.DAT";
  input nome $ 1-8 sexo $ 9 idade 10-12 altura 13-16 .2 peso 17-20 ;
RUN;
proc print;run;
```

```

                                The SAS System                                11:33 Tuesday, June 3, 2014    1

Obs      NOME      SEXO      IDADE      ALTURA      PESO
  1     CARLOS      M         18         1.70         68.0
  2     CARLA      F         26         1.78         72.0
  3     SILVIA     f         21         1.65         60.5
  4     MARCIO     m         32         1.82         .
```

Não é permitido vírgulas em campos numéricos.

"missing value"

```
DATA FICHA2 ;
  INFILE "C:\curso\sas\CADASTRO2.dat";
  INPUT ALTURA 13-16 .2 IDADE 10-12 NOME $ 1-8
  PESO $ 17-20 SEXO $ 9 ;
RUN;
proc print;run;
```

Uma maneira de corrigir é definir a variável como alfanumérica (caractere).

```

                                The SAS System                                11:33 Tuesday, June 3, 2014    4

Obs      ALTURA      IDADE      NOME      PESO      SEXO
  1         1.70         18     CARLOS      68.0      M
  2         1.78         26     CARLA      72.0      F
  3         1.65         21     SILVIA     60.5      f
  4         1.82         32     MARCIO     85,4      m
```

11.5.2 - INPUT FORMATADO

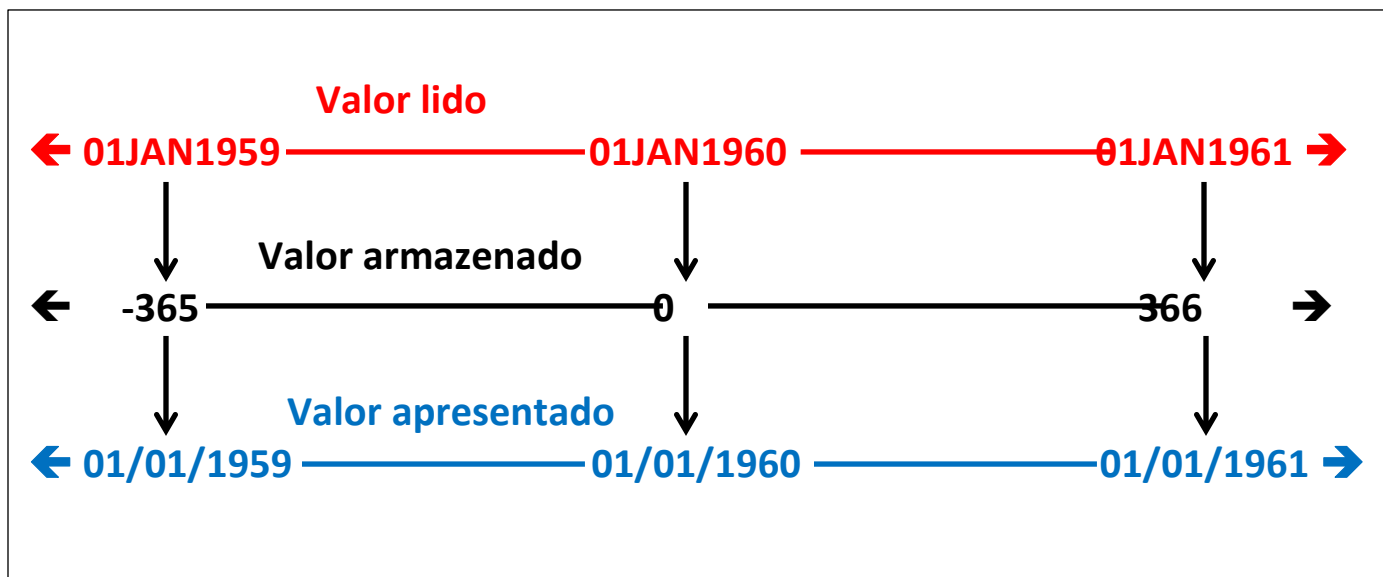
- Lê os campos de dados, atribuindo-lhes um nome (**NAME**), um tipo (**TYPE**), e especificando o **tamanho do campo (LENGTH)**, ou, um **formato de leitura (INFORMAT)** com o **tamanho do campo**, para interpretar dados de um campo;
- O formato de leitura é um recurso do SAS para entender e armazenar corretamente o dado que possui alguma característica especial;
- É o tipo de INPUT mais completo do SAS.

INPUT <variável> [\$] < w.d > ou [informatw.d] . . . ;

variável	Nome do campo de dados.
\$	Caractere opcional que indica que a variável só irá conter dados alfanuméricos. Sem o \$, significa que a variável será numérica.
w.	Valor que indica o tamanho (“width”) de um campo na linha de dados. O ponto no final é obrigatório.
w.d	Valor que indica o tamanho de um campo numérico com casas decimais.
<i>informatw.d</i>	Interpretação do campo. Nome de um formato especial de leitura dos dados: BINARYw.d Lê campo numérico no formato binário e o converte para decimal; COMMAXw.d Lê campo numérico com vírgula e a troca por ponto; Ew.d Lê campo numérico com notação científica; HEXw. Lê campo numérico em hexadecimal; DDMMYYw. Lê campo no formato padrão de datas dd/mm/yyyy.

11.5.3 - Descrição de um Campo Data

- Campos que representam datas necessitam que sejam identificados pelo SAS através de um formato de leitura de datas, e armazenados como numéricos;
- O valor numérico **armazenado** representa o **número de dias** em relação à data base do SAS (01/01/1960);



- É um campo que deve ser lido com um formato especial de leitura para identificar a formatação da data (**informat**) e para ser armazenado como numérico, possibilitando manipular a data em expressões e funções aritméticas (cálculos com datas);

Formatos de leitura de data mais utilizados:

DATE9. Lê dados no formato: **01JAN1960**

DDMMYY10. Lê dados no formato: **01/01/1960**

Formato padrão de uma constante data no SAS:

"ddmmyyyy"d → **"01JAN1960"d**

Ex.7: Leitura de dados com o INPUT FORMATADO

Arquivo: CADASTRO2.DAT

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
C	A	R	L	O	S			M		1	8		1	7	0	6	8	.	0
C	A	R	L	A				F		2	6		1	7	8	7	2	.	0
S	I	L	V	I	A			f		2	1		1	6	5	6	0	.	5
M	A	R	C	I	O			m		3	2		1	8	2	8	5	,	4

```
DATA FICHA ;
  INFILE "C:\curso\sas\CADASTRO2.dat";
  INPUT NOME $ 8. SEXO $ 1. IDADE 3. ALTURA 4.2 PESO 4. ;
RUN;
proc print;run;
```

The SAS System 11:33 Tuesday, June 3, 2014 9

Obs	NOME	SEXO	IDADE	ALTURA	PESO
1	CARLOS	M	18	1.70	68.0
2	CARLA	F	26	1.78	72.0
3	SILVIA	f	21	1.65	60.5
4	MARCIO	m	32	1.82	.

"missing value"

```
DATA FICHA2 ;
  INFILE "C:\curso\sas\CADASTRO2.dat";
  INPUT NOME $ 8. SEXO $ 1. IDADE 3.
        ALTURA 4.2 PESO commax4.1 ;
RUN;
proc print;run;
```

Leitura formatada para interpretar os dados numéricos com a vírgula.

The SAS System 11:33 Tuesday, June 3, 2014 10

Obs	NOME	SEXO	IDADE	ALTURA	PESO
1	CARLOS	M	18	1.70	68.0
2	CARLA	F	26	1.78	72.0
3	SILVIA	f	21	1.65	60.5
4	MARCIO	m	32	1.82	85.4

Ex.8: Leitura de dados especiais com o INPUT FORMATADO

Arquivo: NUMEROS2.PRN

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
2	3	4	.	5	6	7	.	5	5	5	-	1	0	1	5	-	0	8	-	1	9	6	2	6	6	8	3	,	5	8	1	0	%
3	3	3	.	4	9	9	.	1	0	0	-	1	4	2	5	/	1	1	/	1	9	8	2	8	8	4	5	,	4	5	1	5	%
9	8	6	.	3	3	3	.	1	1	2	-	2	2	0	5	/	0	6	/	1	9	8	5	5	6	3	2	,	5	5	1	2	%

```

DATA exemplo ;
  INFILE "C:\curso\sas\numeros2.prn";
  INPUT cpf commax14. aniversario ddmmyy10. salario commax7.2
        aumento percent3. ;
RUN;
PROC PRINT ; RUN ;

```

Datas armazenadas em número de dias desde 01/01/1960.

The SAS System 11:33 Tuesday, June 3, 2014 17

Obs	cpf	aniversario	salario	aumento
1	23456755510	957	6683.58	0.10
2	33349910014	8364	8845.45	0.15
3	98633311222	9287	5632.55	0.12

11.5.4 – Controle de Posição de Leitura do INPUT

Indicador de COLUNA (@)

@n Move a leitura para a coluna n ;

Indicador de LINHA (/)

/ Avança a leitura em **um** registro;

Ex.9: INPUT formatado com indicadores de coluna e de linha

Arquivo: POS.DAT

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1				P	A	U	L	O								1	3	3	5
2		A	N	L	I	S	T	A											
2				1	0	0					7	5	6	8	,	3	3		
1				J	U	L	I	A								2	3	4	2
2		M	E	D	I	C	A												
2				1	0	8					5	6	3	7	,	5	0		

```
DATA POSICAO ;  
  INFILE "C:\SAS\POS.DAT" ;  
  INPUT NOME $ 5-12 @17 MATRICULA $ 4. /      ← 1ª linha de dados  
        @3 FUNCAO $ 15. /                    ← 2ª linha de dados  
        NIVEL $ 5-7 @12 SALARIO commax8.2 ;  ← 3ª linha de dados  
RUN;  
PROC PRINT ; RUN ;
```

The SAS System

Obs	NOME	MATRICULA	FUNCAO	NIVEL	SALARIO
1	PAULO	1335	ANLISTA	100	7568.33
2	JULIA	2342	MEDICA	108	5637.50

11.5.5 - INPUT LISTADO

- Lê os campos de dados delimitados por um caractere especial, atribuindo-lhes um nome e tipo;
- O INPUT listado não permite a definição do tamanho do campo para cada variável. O SAS irá definir, para todas as variáveis, caracteres e numéricas, um tamanho padrão de **8 bytes**;
- O INPUT listado é normalmente utilizado com arquivos que não possuem uma formatação fixa em colunas, mas os campos são delimitados por um caractere especial;
- A opção DLM= do comando INFILE, especifica o caractere especial que existe como delimitador no arquivo texto informado. Se não for especificado, o caractere **branco** será o delimitador “default”;
- O comando INFORMAT (pag.37) auxilia na definição do tamanho e formato de leitura dos campos.

INPUT <variável> [\$] <variável> [\$] ... <variável> [\$] ;

variável Nome do campo de dados.

\$ Caractere opcional que indica que a variável só irá conter dados alfanuméricos. Sem o \$, significa que a variável será numérica.

Ex.10 – Leitura de dados com INPUT LISTADO

Arquivo: LAVOURA1.PRN

A	C	%	B	a	n	a	n	a		P	r	a	t	a	%	1	0	/	0	8	/	2	0	1	0	%	9	.	4	5	9	,	0	0				
S	P	%	L	a	r	a	n	j	a		S	e	l	e	t	a	%	2	5	/	0	5	/	2	0	1	0	%	2	.	0	5	7	,	0	0		
M	G	%	L	i	m	ã	o		G	a	l	e	g	o	%	1	5	/	0	3	/	2	0	0	8	%	2	8	.	3	7	1	,	0	0			
R	J	%	M	a	m	ã	o		P	a	p	a	y	a	%	2	3	/	0	5	/	2	0	0	8	%	2	.	1	9	8	,	0	0				

```
/* 1ª Tentativa: Uso padrão do INPUT LISTADO */
```

```
DATA LAVOURA;
  INFILE "c:\curso\sas\LAVOURA1.PRN";
  INPUT ESTADO $ PRODUTO $ D_COLHEITA VALOR;
RUN;
PROC PRINT;RUN;
```

The SAS System

Obs	ESTADO	PRODUTO	D_COLHEITA	VALOR
1	AC%Banan	Prata%10	.	.
2	MG%Limão	Galego%1	.	.

```
/* 2ª Tentativa: Indicando o delimitador e definindo todas a variáveis como caractere */
```

```
DATA LAVOURA;
  INFILE "c:\curso\sas\LAVOURA1.PRN" DLM="%";
  INPUT ESTADO $ PRODUTO $ D_COLHEITA $ VALOR $;
RUN;
PROC PRINT;RUN;
```

The SAS System

Obs	ESTADO	PRODUTO	D_COLHEITA	VALOR
1	AC	Banana P	10/08/20	9.459,00
2	SP	Laranja	25/05/20	2.057,00
3	MG	Limão Ga	15/03/20	28.371,0
4	RJ	Mamão Pa	23/05/20	2.198,00

11.5.6 – Comando INFORMAT

- **Tipo:** Comando de declaração;

- **Uso:** Data Step ou Proc Step;

- Normalmente é utilizado para especificar o formato de leitura de um campo, antes da leitura dos dados pelo comando INPUT LISTADO.

```
INFORMAT <variável1> <informatw.d>
          <variável2> <informatw.d>
          ...
          <variáveln> <informatw.d>;
```

Ex.11 – Utilização do comando INFORMAT

Arquivo: LAVOURA1.PRN

A	C	%	B	a	n	a	n	a		P	r	a	t	a	%	1	0	/	0	8	/	2	0	1	0	%	9	.	4	5	9	,	0	0				
S	P	%	L	a	r	a	n	j	a		S	e	l	e	t	a	%	2	5	/	0	5	/	2	0	1	0	%	2	.	0	5	7	,	0	0		
M	G	%	L	i	m	ã	o		G	a	l	e	g	o	%	1	5	/	0	3	/	2	0	0	8	%	2	8	.	3	7	1	,	0	0			
R	J	%	M	a	m	ã	o		P	a	p	a	y	a	%	2	3	/	0	5	/	2	0	0	8	%	2	.	1	9	8	,	0	0				

```
/* 3ª Tentativa: Utilizando o comando INFORMAT */
```

```
DATA LAVOURA;
  INFILE "c:\curso\sas\LAVOURA1.PRN" DLM=" ";
  INFORMAT ESTADO $2.
           PRODUTO $15.
           D_COLHEITA DDMMYY10.
           VALOR COMMAX12.2;
  INPUT ESTADO $ PRODUTO $ D_COLHEITA VALOR ;
RUN;
PROC PRINT; RUN;
```

The SAS System

Obs	ESTADO	PRODUTO	D_COLHEITA	VALOR
1	AC	Banana Prata	18484	9459
2	SP	Laranja Seleta	18407	2057
3	MG	Limão Galego	17606	28371
4	RJ	Mamão Papaya	17675	2198

11.5.7 - Detalhes do Comando INPUT

- Em campos numéricos, não são permitidos *caracteres brancos* entre os dígitos dos números;
- Os caracteres que indicam o sinal positivo ou negativo (+ -), o ponto decimal (.) e a notação científica para expoente (**E**), são permitidos em campos numéricos;
- Campos tipo caractere podem ter, no máximo, 32.767 caracteres;
- São permitidos *caracteres brancos*, em qualquer posição, em campos do tipo caractere;
- Campos em branco (sem informação) são considerados como "*missing value*" (valor perdido). O SAS irá armazenar o caractere **branco**, em campos caracteres, e o caractere **ponto**, em campos numéricos;
- Os campos de um registro de dados podem ser lidos em qualquer ordem.

```
INPUT PESO 17-20 NOME $ 1-8 IDADE 11-12 SEXO $ 9 ;
```

- Campos ou partes de um campo podem ser relidos.

```
INPUT NOME $ 1-8 PREFIXO $ 1-3 ENDERECO $ 10-40 APT $ 38-40 ;
```

- Dados caracteres serão armazenados alinhados pela esquerda, e os dados numéricos, alinhados pela direita;
- **Os tipos de INPUT, colunado, formatado e listado, podem ser combinados num único comando INPUT;**
- Detalhes de leituras com nomes e formatos repetitivos;

```
INPUT JAN 3. FEV 3. MAR 3. ABR 3. MAI 3. JUN 3. ;
```

```
INPUT (JAN FEV MAR ABR MAI JUN) (3.) ;
```

```
INPUT (MES1 MES2 MES3 MES4 MES5 MES6) (3.) ;
```

```
INPUT (MES1-MES6) (3.) ;
```

Ex.12 – Todos os tipos de INPUT combinados

Arquivo: informat.txt

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
J	o	A	O					1	2	3	4	5	6	7	8	9	0	1	1	2	/	1	0	/	7	0		1	e	4		8	.	5	6	7	,	4	8
S	a	N	D	r	a			8	9	0	5	6	7	8	4	3	2	5	0	1	-	0	4	-	6	5		2	e	5		9	.	7	8	6	,	5	5
C	a	R	L	o	s			2	3	4	5	4	3	6	7	8	9	2	1	1	.	0	8	.	7	4		3	e	2		6	.	0	4	5	,	1	2

```

DATA FORMATO;
  INFORMAT DATANIV DDMMYY8.
  SALARIO COMMAX8.2
  VALOR E3.;
  INFILE "c:\curso\sas\informat.txt";
  INPUT NOME $ 1-8 CPF $ 11. DATANIV VALOR SALARIO ;
RUN;
PROC PRINT;RUN;

```

The SAS System

Obs	DATANIV	SALARIO	VALOR	NOME	CPF
1	3937	8567.48	10000	Joao	12345678901
2	1917	9786.55	200000	Sandra	89056784325
3	5336	6045.12	300	Carlos	23454367892

1º LABORATÓRIO

1-Montar um programa SAS que leia um arquivo de dados externo (não SAS), utilizando o comando **INPUT**, **COLUNADO** ou **FORMATADO**, e gere um arquivo no formato SAS (Data Set), permanente.

1.1- Criar um arquivo SAS, **permanente**, com o nome **CADASTRO**, na pasta **c:\curso\sas** (Utilize os comandos **LIBNAME** e **DATA**);

1.2 - Abra o arquivo de dados externo de nome **CADASTRO.DAT**, também localizado na pasta **c:\curso\sas** (Utilize o comando **INFILE**). O Tamanho do registro de dados é de **130 bytes**;

1.3 - Leia os dados deste arquivo (Utilize o comando **INPUT**). Sabendo-se que o conteúdo do arquivo está dividido em diversos campos na seguinte ordem:

Descrição do Campo	Tipo do Campo	Tamanho do Campo	Decimais	Posição Inicial	Detalhe
NOME	Caractere	30		1	
SEXO	Caractere	1		31	
IDADE	Numérico	2		32	
PESO	Numérico	6	2	34	
ALTURA	Numérico	4	2	40	
ANIVERSARIO	Numérico	10		44	Campo formatado: Ex. 18/01/2005
ESTADO CIVIL	Caractere	1		54	
FILHOS	Numérico	2		55	
RG	Caractere	15		57	
CPF	Caractere	11		72	
EMPRESA	Caractere	15		83	
FUNCAO	Caractere	12		98	
ADMISSAO	Numérico	9		110	Campo formatado: Ex 18JAN2005
SALARIO	Numérico	12	2	119	Campo formatado: Ex. 3.548,25

1.4 - Releia parte do campo EMPRESA, na posição 89, defina uma nova variável que representará o TIPO DE EMPRESA, com 9 caracteres;

1.5 - Gere um relatório com os resultados da leitura do arquivo de dados, com os comandos abaixo:

PROC PRINT;RUN;

OBS: Verifique possíveis problemas que possam ter ocorridos na leitura dos dados. Salve os seus programas! Faça-o **sempre**, na pasta **c:\curso\sas**

2-Montar um programa SAS que leia um arquivo de dados externo (não SAS), utilizando o comando **INPUT LISTADO**, e gere um arquivo no formato SAS (Data Set), permanente.

2.1- Criar um arquivo SAS, **permanente**, com o nome **VAREJO**, na pasta **c:\curso\sas** (Utilize os comandos **LIBNAME** e **DATA**);

2.2 - Abra o arquivo de dados externo de nome **VAREJO.DAT**, também localizado na pasta **c:\curso\sas** (Utilize o comando **INFILE**). Esse arquivo não possui os dados alinhados, colunados, mas possui um delimitador (**#**) entre os campos;

2.3 - Leia os dados deste arquivo (Utilize o comando **INPUT**). Sabendo-se que o conteúdo do arquivo está dividido em diversos campos na seguinte ordem:

Descrição do Campo	Tipo do Campo	Tamanho do Campo	Decimais	Detalhe
CPF	Caractere	11		
DATA DA COMPRA	Numérico	10		Campo formatado: Ex. 18/01/2005
PRODUTO	Caractere	16		
PREÇO	Numérico	5	2	Campo formatado: Ex. 3.548,25

2.4 - Gere um relatório com os resultados da leitura do arquivo de dados, com os comandos abaixo:

PROC PRINT;RUN;

OBS: Verifique possíveis problemas que possam ter ocorridos na leitura dos dados. Salve os seus programas! Faça-o **sempre**, na pasta **c:\curso\sas**.

11.6 - Comando SET

- **Tipo:** Comando de declaração e execução;

- **Uso:** Data Step;

- **Durante a fase de compilação**, este comando abre um ou vários arquivos de dados, DATA Sets (**formato SAS**), **somente para leitura**, e que já existam em alguma biblioteca SAS. (Funciona como o comando INFILE)

- **Durante a fase de execução**, o comando lê os registros de maneira sequencial, um após o outro (Funciona como o comando INPUT);

- O comando deve ser posicionado, na maioria das vezes, no início de um Data Step, para permitir ao compilador ler a estrutura do arquivo especificado e **armazená-la na memória**.

SET [biblioteca.]<nome do arquivo> ... ;

Ex.13: Declaração para leitura de arquivos com formatação SAS

```
set "c:\curso\sas\treinamento";      ( Extensão de um arquivo sas: sas7bdat )  
  
set tab1 tab2 ;  
  
SET cadastro ;  
  
SET arq.dados ;
```

```
libname ab "c:\curso\sas";  
DATA ab.EXEMPLO1 ;  
    INFILE "C:\curso\sas\CADASTRO.TXT" ;  
    INPUT  NOME $ 1-8  SEXO $ 10  
           IDADE $ 13-14  ALTURA 16-19 ;  
  
RUN;  
PROC PRINT DATA=ab.EXEMPLO1 ; RUN;  
  
DATA ab.novo_exemplo ;  
    SET ab.exemplo1 ;  
    ... ;  
RUN ;
```

OBS: Não é obrigatório especificar a biblioteca, neste caso, o SAS irá utilizar a biblioteca padrão, também chamada de biblioteca temporária (**WORK**).

11.7 - Comando RUN

- **Tipo:** Comando de execução;

- **Uso:** Data Step e Proc Step;

- Comando que identifica o fim de uma estrutura Data Step e Proc Step e indica ao SAS que o “step” deve ser executado imediatamente;

- É apenas necessário no último “step”;

12 - Estrutura PROC Step

12.1 - Introdução

- É a seção de um programa SAS aonde se analisam os dados de um arquivo SAS, utilizando-se procedimentos específicos para cada tipo de análise.
- A maioria dos procedimentos SAS, geram relatórios técnicos;
- Todos os procedimentos iniciam com a palavra PROC seguida do nome específico do procedimento ou rotina desejada;
- Os comandos em uma PROC não seguem uma sequência lógica, ou seja, podem ser colocados em qualquer ordem dentro da PROC;

```
PROC <rotina> [opções] ;  
    [comando1] ;  
    [comando2] ;  
    ...  
    [comandon] ;  
RUN ;
```

rotina Palavra-chave que identifica o procedimento a ser executado.

opções Parâmetros opcionais de configuração para execução do procedimento.

comando Comandos de detalhamento na execução do procedimento.

RUN Comando para execução do procedimento SAS. Necessário no último procedimento do programa SAS.

12.1.1 - Comando VAR

- **Tipo:** Comando de declaração;

- **Uso:** Proc Step

- Comando comum à várias PROCS (PRINT, MEANS, SUMMARY, TABULATE, etc), mas que pode possuir características específicas da PROC utilizada; no geral, especifica uma **lista de variáveis** do arquivo SAS que serão processadas pelo procedimento.

VAR <*lista de variáveis*> ;

lista de variáveis

variável variável variável ... variável ;

x1-xn ; (variáveis numeradas)

varLXP--varB8T ; (todas as variáveis entre)

var: ; (todas as variáveis com o mesmo prefixo)

numeric ; (todas as variáveis numéricas)

character ; (todas as variáveis caracteres)

12.1.2 - Comando BY

- **Tipo:** Comando de declaração;

- **Uso:** Data Step e Proc Step;

- Especifica uma lista de variáveis que permitirão o processamento de grupos, determinados pelos valores das variáveis.

BY data step

- Sempre utilizado associado aos comandos: SET, MERGE, MODIFY e UPDATE, para controlar a leitura dos dados por agrupamento de valores.

BY [descending] <variável1> [descending] <variável2> ... ;

BY proc step

- Utilizado para permitir a análise dos dados por grupos, e gerar relatórios com os dados em ordem ascendente ou descendente.

BY [descending] <variável1> [descending] <variável2> ... ;

- Exemplo de variáveis que permitem o processamento por grupo: SEXO, EMPRESA, ESTADO, IDADE.

OBS: Muito importante! Este comando, seja utilizado em Data step ou em Proc step, **exige** que o arquivo que será processado, esteja **ordenado ou indexado** pelas variáveis especificadas, ou seja, antes do processamento, será necessário ordenar ou indexar os dados. Nesta apostila, veja o item sobre ordenação de dados – 12.3 - Procedimento SORT, pag. 44.

12.1.3 - Comando TITLE

- **Tipo:** Comando de declaração;

- **Uso:** Livre;

- Pode ser codificado em qualquer parte do programa. Especifica de 1 até 10 títulos para o relatório a ser gerado pelo procedimento.

```
TITLE "<cabeçalho>" ;  
TITLE1 "<cabeçalho>" ;  
...  
TITLE10 "<cabeçalho>" ;
```

OBS: Um título permanece ativo até que seja modificado, ou eliminado com o comando: **TITLE**;

12.1.4 - Comando FOOTNOTE

- **Tipo:** Comando de declaração;

- **Uso:** Livre;

- Pode ser codificado em qualquer parte do programa. Especifica de 1 até 10 rodapés para o relatório a ser gerado pelo procedimento.

```
FOOTNOTE "<rodapé>" ;  
FOOTNOTE1 "<rodapé>" ;  
...  
FOOTNOTE10 "<rodapé>" ;
```

OBS: Um rodapé permanece ativo até que seja modificado, ou eliminado com o comando: **FOOTNOTE**;

12.2 - Procedimento PRINT

- Procedimento do SAS que imprime, *na saída padrão*, os dados contidos nos arquivos SAS;
- A saída padrão é, a princípio, a janela OUTPUT em uma sessão SAS Windows.

```
PROC PRINT [opções] ;  
    [VAR <lista de variáveis> ;]  
    [BY <lista de variáveis> ;]  
    [SUM <lista de variáveis> ;]
```

opções	DATA=	Nome do arquivo SAS que será processado;
	DOUBLE	Espacejamento duplo;
	NOOBS	Suprime a coluna com o número de cada observação do arquivo;
	LABEL	Imprime uma <i>descrição</i> das variáveis. Descrição que pode ser definida na PROC step ou ter sido armazenada na estrutura do arquivo em um DATA step;
	BLANKLINE= <i>n</i>	Insere uma linha em branco a cada <i>n</i> linhas de dados;
VAR		Especifica uma lista de variáveis que serão impressas;
BY		Especifica a quebra ou agrupamento do relatório, por uma ou mais variáveis;
SUM		Especifica uma ou mais variáveis numéricas que serão totalizadas;

Ex.14: Procedimento PRINT 1

```
libname tre "c:\curso\sas";
title "Relatório de Funcionários" ;

PROC PRINT DATA=tre.cadastro noobs ;
var NOME FUNCAO ;
RUN ;
```

Relatório de Funcionários 11:33 Tuesday, September 12, 2017 1

nome	funcao
MARKO, PAULO	DESEMPREGADO
MOUA, MARCO	ANALISTA
SANTOS, PAULO	DESEMPREGADO
GUEDES, PAULO	PROGRAMADOR
SONTAS, MARCO	PROGRAMADOR
HORTA, PAULO	PROGRAMADOR
CERTO, CARLA	PROGRAMADOR
PISCO, ROSANE	PROGRAMADOR
MALA, ELIANE	DESEMPREGADO
APARECIDO, CARLA	PROGRAMADOR
YATAKA, ROSANE	PROGRAMADOR
MARUEL, ELIANE	PROGRAMADOR
MARQUES, CARLA	PROGRAMADOR
MILIA, CARLA	PROGRAMADOR

Ex.15: Procedimento PRINT 2

```
libname tre "c:\curso\sas";
title "Relatório de Funcionários" ;
footnote "FONTE: RH" ;

PROC PRINT DATA=tre.cadastro noobs ;
var NOME FUNCAO SALARIO ;
sum salario ;
RUN ;
```

Relatório de Funcionários 11:33 Tuesday, September 12, 2017 368

nome	funcao	salario
BENTES, LIGIA	PROGRAMADOR	1525.33
MOUA, CECILIA	DESEMPREGADO	.
MEREDITE, LIGIA	PROGRAMADOR	3004.89
PINTOTO, FRANCISCA	PROGRAMADOR	3125.17
MARKO, LICIA	PROGRAMADOR	3513.16
SONTAS, LAURA	DESEMPREGADO	.
MALA, LICIA	PROGRAMADOR	1865.04
HORTA, LAURA	PROGRAMADOR	2299.34
BENTES, LAURA	PROGRAMADOR	2555.46
MENDES, FLAVIA	PROGRAMADOR	3888.40
SAUL, EDUARDO	PROGRAMADOR	3759.84
BENTES, EDUARDO	PROGRAMADOR	3163.89
PISCO, FLAVIA	PROGRAMADOR	3349.55
MARKO, FLAVIA	PROGRAMADOR	4305.02
ANJOA, EDUARDO	PROGRAMADOR	3250.67
SUNAY, EDUARDO	PROGRAMADOR	1704.69
		=====
		1446180.53

FONTE: RH

12.3 - Procedimento SORT

- Procedimento do SAS que ordena os registros dos arquivos por uma ou mais variáveis, em ordem ascendente ou descendente;
- Não possui saída impressa;
- É possível gerar um novo arquivo ordenado.

PROC SORT [opções] ;
<BY/KEY> [descending] <lista de variáveis> ;

opções	DATA=	Nome do arquivo SAS para ser ordenado;
	OUT=	Nome do arquivo SAS, aonde serão armazenado os dados ordenados. Se não for utilizada essa opção, a ordenação será feita em cima do arquivo original;
	NODUPKEY	Opção que elimina os registros com chaves duplicadas;
	DUPOUT=	Nome para arquivo que irá armazenar os registros duplicados;
	SORTSIZE=	Especifica o limite para o uso de memória. Inicialmente em 256MBytes;
	BY ou KEY	Comando obrigatório , especifica uma ou mais variáveis que irão determinar a ordenação dos dados. Só pode existir um comando BY dentro de uma proc sort;

OBS: A ordenação sempre é feita na ordem ascendente. Para ordenar na ordem descendente é necessário colocar o parâmetro **descending**, antes da variável a ser ordenada no comando **BY**.

Ex.16: Procedimento SORT

```
title1 "Relatório de Funcionários" ;
title2 "por Função";
footnote;
PROC SORT DATA=TRE.CADASTRO OUT=ORDENA ;
      BY FUNCAO DESCENDING SALARIO;
RUN ;
PROC PRINT DATA=ORDENA NOOBS ;
      BY FUNCAO ;
      VAR NOME SALARIO ;
      SUM SALARIO ;
RUN ;
```

Relatório de Funcionários 1
por Função

```
----- funcao=ANALISTA -----
      nome                salario

      MOUA,MADALENA      14821.37
      MOUA,ELIANE        13694.19
      MOUA,MONICA        12568.82
      MOUA,MARCO         9988.53
      MOUA,PAULO         9548.36
      MOUA,LAURA        7402.21
      MOUA,CARLA         7221.58
      MOUA,TANIA         5418.33
      MOUA,LIGIA         5047.05
      -----
      funcao              85710.44

----- funcao=DIRETOR -----
      nome                salario

      MOUA,MARCELO       25377.28

----- funcao=GERENTE -----
      nome                salario

      MOUA,JOAO          20611.56
      MOUA,RENATO        20457.36
      MOUA,ROSANE        20336.22
      MOUA,MARIA         19303.66
      MOUA,LUIS          17414.14
      MOUA,LICIA         17260.69
      MOUA,MIRIAM        13075.91
      MOUA,MARCIO        10828.70
      -----
      funcao              139288.24

----- funcao=PROGRAMADOR -----
      nome                salario

      LUILA,CARLA        5178.65
      ...                ...
      ...                ...
      ...                ...
      GUEDES,EDUARDO     1507.53
      YATAKA,MONICA      1500.73
      -----
      funcao              1195804.57
      =====
                        1446180.53
```

13 - ARQUIVO DE MENSAGENS - "SAS Log"

- Ao se executar um programa SAS, é gerado um relatório de mensagens com **notificações e erros** da execução do programa (janela LOG).

- **Muito útil, deve ser sempre analisado com atenção !!!** Principalmente quando não aparecer o relatório com os resultados na janela OUTPUT, ou, quando esse relatório aparecer com problemas.

Ex.17: Exemplo de LOG sem problemas na execução 1

```
DATA LOG1 ;  
    INFILE "C:\curso\sas\DADOS.TXT" ;  
    INPUT NOME $1-8 SEXO $10 FUNCAO $12-23 SALARIO 25-31 ;  
RUN;  
PROC PRINT DATA=LOG1 NOOBS ;  
    VAR NOME FUNCAO ;  
RUN;
```

```
88 DATA LOG1 ;  
89     INFILE "C:\curso\sas\DADOS.TXT" ;  
90     INPUT NOME $1-8 SEXO $10 FUNCAO $12-23 SALARIO 25-31 ;  
91 RUN;
```

NOTE: The infile "C:\curso\sas\DADOS.TXT" is:
Filename= C:\curso\sas\DADOS.TXT,
RECFM=V,LRECL=256,File Size (bytes)=297,
Last Modified=06 de Agosto de 2010 15h34min0,
Create Time=16 de Junho de 2014 14h28min15

NOTE: 9 records were read from the infile
"C:\curso\sas\DADOS.TXT".
The minimum record length was 31.
The maximum record length was 33.

NOTE: The data set WORK.LOG1 has 9 observations and 4 variables.
NOTE: DATA statement used (Total process time):
real time 0.05 seconds
cpu time 0.01 seconds

```
92 PROC PRINT DATA=LOG1 NOOBS ;  
93     VAR NOME FUNCAO ;  
94 RUN;
```

NOTE: There were 9 observations read from the data set WORK.LOG1.
NOTE: PROCEDURE PRINT used (Total process time):
real time 0.01 seconds
cpu time 0.00 seconds

Esses números indicam coerência na leitura e gravação dos dados. Quando existir apenas um comando INPUT, sem avanço de linha e sem uma lógica de exclusão ou seleção de dados, os números, geralmente serão iguais.

Ex.18: Exemplo de LOG com problemas na execução 2

```
DATA LOG2 ;
  INFILE "C:\curso\sas\DADOS.TXT" ;
  INPUT NOME $1-8 SEXO $10 FUNÇÃO $12-23 SALARIO 25-31 ;
RUN;
PROC PRINT DATA=LOG2 NOOBS
  VAR NOME FUNÇÃO ;
RUN;
```

```
123 DATA LOG2 ;
124     INFILE "C:\curso\sas\DADOS.TXT" ;
125     INPUT NOME $1-8 SEXO $10 FUNÇÃO $12-23 SALARIO 25-31 ;
ERROR: The name FUNÇÃO is not a valid SAS name
```

Não é permitido caracteres especiais ou acentuados em nomes de arquivos e variáveis.

```
126 RUN;
```

```
NOTE: The SAS System stopped processing this step because of errors.
WARNING: The data set WORK.LOG2 may be incomplete. When this step was stopped there were 0
  observations and 3 variables.
WARNING: Data set WORK.LOG2 was not replaced because this step was stopped.
NOTE: DATA statement used (Total process time):
  real time          0.00 seconds
  cpu time           0.00 seconds
```

```
127 PROC PRINT DATA=LOG2 NOOBS
128     VAR NOME FUNÇÃO ;
    ---
    22
    202
```

Faltou finalizar com ; O comando PROC PRINT. O erro ocorre no próximo comando. O comando VAR foi tratado como uma opção do comando PROC PRINT; como essa opção não existe, ocorre o erro.

```
ERROR 22-322: Syntax error, expecting one of the following: ;, BLANKLINE, DATA, DOUBLE,
  HEADING, LABEL, N, NOOBS, OBS, ROUND, ROWS, SPLIT, STYLE, SUMLABEL, UNIFORM,
  WIDTH.
ERROR 202-322: The option or parameter is not recognized and will be ignored.
```

```
129 RUN;
```

```
NOTE: The SAS System stopped processing this step because of errors.
NOTE: PROCEDURE PRINT used (Total process time):
  real time          0.00 seconds
  cpu time           0.00 seconds
```

Ex.19: Exemplo de LOG com detalhes e problemas na execução 3

```
DATA LOG3 ;
  INFILE "C:\curso\sas\DADOS.TXT" ;
  INPUT NOME $1-8 SEXO $10 FUNCAO $12-23 SALARIO $25-33 ;
RUN;
PROC PRINT DATA=LOG3 NOOBS ;
  VAR NOME CARGO ;
  SUM SALARIO ;
RUN;
```

```
40 DATA LOG3 ;
41     INFILE "C:\curso\sas\DADOS.TXT" ;
42     INPUT NOME $1-8 SEXO $10 FUNCAO $12-23 SALARIO $25-33 ;
43 RUN;
```

```
NOTE: The infile "C:\curso\sas\DADOS.TXT" is:
      Filename=C:\curso\sas\DADOS.TXT,
      RECFM=V,LRECL=256,File Size (bytes)=297,
      Last Modified=06 de Agosto de 2010 15h34min0,
      Create Time=16 de Junho de 2014 14h28min15
```

Registros de tamanhos diferentes. Indica a necessidade de usar a opção MISSEVER no comando INFILE.

Este NOTE, indica que o SAS precisou ler um novo registro para completar a leitura de dados de uma variável.

Nove registros lidos

```
NOTE: 9 records were read from the infile
      "C:\curso\sas\DADOS.TXT".
      The minimum record length was 31.
      The maximum record length was 33.
```

```
NOTE: SAS went to a new line when INPUT statement reached past the end of a line.
```

```
NOTE: The data set WORK.LOG3 has 5 observations and 4 variables.
```

```
NOTE: DATA statement used (Total process time): Cinco registros gravados ???
      real time          0.02 seconds
      cpu time           0.01 seconds
```

```
44 PROC PRINT DATA=LOG3 NOOBS ;
45     VAR NOME CARGO ;
ERROR: Variable CARGO not found.
46     SUM SALARIO ;
ERROR: Variable SALARIO in list does not match type prescribed for this list.
47 RUN;
```

Não existe a variável CARGO no arquivo.

```
NOTE: The SAS System stopped processing this step because of errors.
```

```
NOTE: PROCEDURE PRINT used (Total process time):
      real time          0.04 seconds
      cpu time           0.01 seconds
```

A variável SALARIO foi originalmente criada no comando INPUT como sendo do tipo caractere. O comando SUM da PROC PRINT, exige que a variável seja do tipo numérica.

2º LABORATÓRIO

Manipular os procedimentos de impressão e ordenação.

Até o item 6, **utilize o arquivo SAS CADASTRO gerado no 1º laboratório.**

1 – Monte um programa SAS que gere um relatório:

- com as variáveis: **NOME, EMPRESA, FUNCAO e SALARIO;**
- com opções para: **espacejamento DUPLO** e para **não** aparecer à coluna com o número do registro (**OBS**);
(Utilize o procedimento **PRINT**)

2 – Monte um programa SAS que ordene os dados e gere um relatório.

- primeiro, um procedimento que ordene os dados por **EMPRESA** e salve os dados ordenados em um novo arquivo;
- segundo, um procedimento que gere um relatório com todos os dados, **agrupados** por **EMPRESA**;
(Utilize os procedimentos **SORT** e **PRINT**)

3 – Utilizando o arquivo ordenado do item anterior, monte um programa SAS:

- gere um relatório **agrupado** por **EMPRESA**;
- com as variáveis: **NOME, FUNCAO e SALARIO;**
- que possua um **somatório** da variável **SALARIO**;
- com uma linha em branco a cada 3 linhas de dados;
- com o **título** "Relatório de Salários por Empresa";
- coloque o **rodapé** "FONTE: Fundação Getúlio Vargas";
(Utilize o procedimento **PRINT**)

4 – Monte um programa SAS que gere um relatório:

- com as variáveis: **NOME, FUNCAO e SALARIO;**
- em ordem **ascendente** por **EMPRESA** e **descendente** por **SALARIO**.
(Utilize os procedimentos **SORT** e **PRINT**)

5 – Monte um programa SAS que gere um relatório:

- esteja ordenado por **EMPRESA, mas sem agrupamento**;
- com as variáveis: **NOME, EMPRESA e SALARIO;**
- tenha **espacejamento DUPLO**;
- **sem título e sem rodapé**;
- **com o somatório** da variável **SALARIO**;
(Utilize o procedimento **PRINT**)

6 – Monte um programa SAS que gere um relatório com apenas os funcionários que mais ganham por função, de cada empresa.

A - Primeiro passo:

- Execute um PROC SORT, por **EMPRESA, FUNCAO** e **ordem descendente** de **SALARIO**;
- Verifique os resultados com a PROC PRINT, com as variáveis: **NOME, EMPRESA, FUNCAO, SALARIO**;

B - Segundo passo:

- Execute um PROC SORT, **mas gerando um NOVO ARQUIVO temporário**, por **EMPRESA e FUNCAO**;
- Elimine as duplicatas de **EMPRESA e FUNCAO**, desse novo arquivo, através de um opção do PROC SORT;
- Verifique os resultados com a PROC PRINT, com as variáveis: **NOME, EMPRESA, FUNCAO, SALARIO**;
- Verifique se os resultados estão de acordo com o que foi solicitado no enunciado do exercício.

7 – Utilizando o arquivo SAS **VAREJO** criado no 1º laboratório, gere um relatório com as seguintes características:

- Esteja **ordenado** por **CPF e PRODUTO**. Crie um novo arquivo com os dados ordenados;
- **Agrupado** apenas pela variável **CPF**;
- Com as variáveis **PRODUTO e PRECO**;
- Com o **somatório** da variável **PRECO**;
- Sem apresentar a numeração de linhas.

8 – Utilizando o arquivo SAS **VAREJO** criado no 1º laboratório e, seguindo o mesmo raciocínio do exercício 6, gere um relatório com o **PRODUTO** de maior **PRECO**, por cada **CPF**.

14 – Estrutura DATA Step – Comandos Básicos de Programação

14.1 - Comando de Alocação (=)

- **Tipo:** Comando de execução;

- **Uso:** Data Step;

- Comando caractere utilizado para criar novas variáveis e editar as já existentes em um Data Step.

<variável> = <expressão> ;

variável Nome com no máximo 32 caracteres.
expressão Expressões do tipo:

- **Operações aritméticas simples:** + - * / **

x2=x ;	Move valor;
soma=x+y ;	Adição;
dif=x-y ;	Subtração;
dobro=x*2 ;	Multiplicação;
met=x/2 ;	Divisão;
cubo=x**3 ;	Potenciação;
y=-x ;	Mudança de sinal;

- **Constantes :**

N=0 ; Constante numérica; SEXO="F" ; Constante caractere;

- **Concatenação de caracteres :** ||

A="PAULO" "ANTUNES";	→	A="PAULOANTUNES"
B="PAULO" " " "ANTUNES";	→	B="PAULO ANTUNES"

- **Funções:**

S=SQRT(X) ; A=INT(X) ; Z=ABS(SQRT(X)-2) ;

- **Prioridade de execução em expressões aritméticas:**

1º- O operador com maior prioridade é executado primeiro;

2º- Se os operadores possuírem a mesma prioridade, será executado o que vier primeiro da esquerda para direita;

3º- Parêntesis dentro de parêntesis, o mais interno é executado primeiro;

1º	2º	3º	4º	5º
()	função	**	*/	+ -
←	←			←
Maior prioridade				Menor prioridade

A=X+Y+Z ; A=X+Y*Z ; A=X/(Y/Z) ;

Ex.20: Comando de Alocação

Arquivo: NUMEROS.TXT

01	02	03	04	05	06	07	08	09	10	11	12
9	4		5	6	5	0		1	0	5	0
9	5		6	2	8	0		1	1	4	0
9	6		8	4	5	0		2	4	5	0
9	7							1	5	0	0

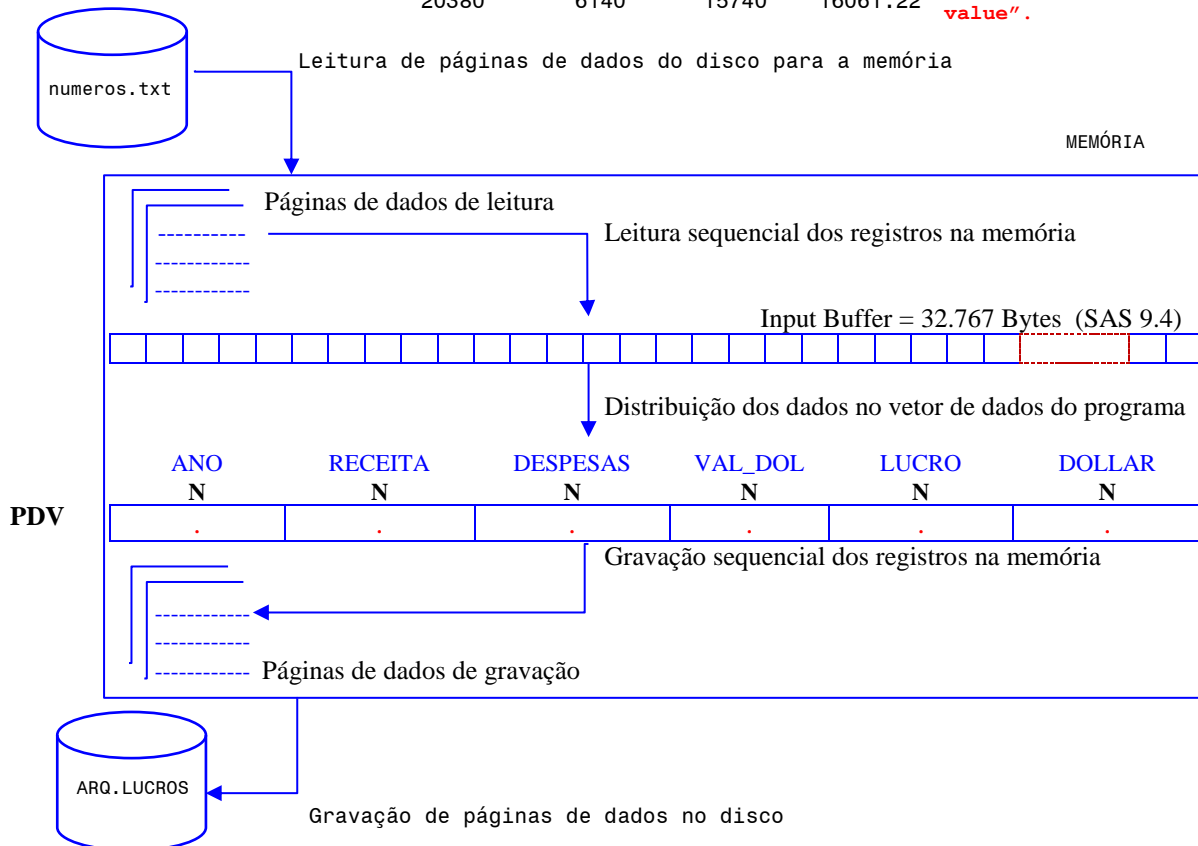
```
LIBNAME ARQ "C:\curso\sas" ;
DATA ARQ.LUCROS ;
  INFILE "C:\curso\sas\NUMEROS.TXT";
  INPUT ANO 1-2 RECEITA 4-7 DESPESAS 9-12;
  VAL_DOL = 0.98 ;
  ANO = ANO + 1900 ;
  LUCRO = RECEITA - DESPESAS ;
  DOLLAR = ( RECEITA - DESPESAS ) / VAL_DOL ;
RUN;

PROC PRINT DATA=ARQ.LUCROS NOOBS ;
  TITLE "Análise Contábil" ;
  VAR ANO RECEITA DESPESAS LUCRO DOLLAR ;
  SUM RECEITA DESPESAS LUCRO DOLLAR ;
RUN ;
```

Análise Contábil 11:17 Friday, June 27, 2014 2

ANO	RECEITA	DESPESAS	LUCRO	DOLLAR
1994	5650	1050	4600	4693.88
1995	6280	1140	5140	5244.90
1996	8450	2450	6000	6122.45
1997	.	1500	.	.
	=====	=====	=====	=====
	20380	6140	15740	16061.22

ATENÇÃO! Qualquer operação com "missing value" terá como resultado, "missing value".



14.2 - Comando RETAIN

- **Tipo:** Comando de declaração;

- **Uso:** Data Step;

- Normalmente o SAS inicia todas as variáveis **criadas** no DATA Step, pelo **comando INPUT** ou pelos **comandos de alocação**, com o valor **“missing”**, a cada iteração do DATA Step;

- O Comando RETAIN define uma variável com um valor inicial, e determina que ela não seja reinicializada a cada iteração do DATA Step;

- Deve ser posicionado sempre antes do comando que irá utilizar a variável definida;

RETAIN *<variável1> [valor inicial]*
 ... *...*
 <variáveln> [valor inicial];

Ex.21: Comando RETAIN

```
LIBNAME ARQ "C:\curso\sas" ;
DATA ARQ.LUCROS ;
  INFILE "C:\curso\sas\NUMEROS.TXT";
  RETAIN TOTAL 0;
  INPUT ANO 1-2 RECEITA 4-7 DESPESAS 9-12;
  ANO = ANO + 1900 ;
  LUCRO = RECEITA - DESPESAS ;
  TOTAL = TOTAL + LUCRO ;
  DOLLAR = LUCRO / 0.98 ;
RUN;

PROC PRINT DATA=ARQ.LUCROS NOOBS;
  TITLE "Análise Contábil" ;
  VAR RECEITA DESPESAS LUCRO DOLLAR TOTAL ;
  SUM RECEITA DESPESAS LUCRO DOLLAR ;
RUN ;
```

```

                                Analise Contábil                                11:17 Friday, June 27, 2014    3
RECEITA    DESPESAS    LUCRO    DOLLAR    TOTAL
    5650         1050         4600    4693.88    4600
    6280         1140         5140    5244.90    9740
    8450         2450         6000    6122.45    15740
    .             1500         .         .         .
=====
    20380         6140         15740    16061.22

```

14.3 - Comando IF-THEN/ELSE

- **Tipo:** Comando de Execução;

- **Uso:** Data Step;

- Comando que condiciona a execução de outro comando SAS, de acordo com uma **expressão lógica**;

- Se a expressão for **verdadeira** (ou resolver para um valor **diferente de 0**), será executado o comando que vier após o **THEN**. Se a expressão for **falsa** (ou resolver para o valor **igual a 0**), será executado o comando que vier após o **ELSE**. **Somente um comando** pode ser executado após o **THEN** e do **ELSE**.

```
IF <expressão>      [THEN] <comando> ;  
                        [ELSE] <comando> ;
```

expressão Combinação de variáveis com operadores de comparação e/ou operadores lógicos que determinam uma condição.

comando Comando do SAS que será executado de acordo com o resultado da condição.

Operadores de Comparação

GT	>	maior que
LT	<	menor que
EQ	=	igual a
LE	<=	menor ou igual a
GE	>=	maior ou igual a
NE	~=	não é igual (diferente)
NL		não é menor
NG		não é maior
IN		está no conjunto

Operadores Lógicos

AND	&	e, ambos
OR		ou, um ou outro
NOT	~	não, negação

Ex.22: Comando IF (Somente Selecionar de dados)

```
LIBNAME DAD "C:\curso\sas";
DATA DAD.DADOS86 ;
  INFILE "C:\curso\sas\CADASTRO.TXT" LRECL=50 MISSOEVER;
  INPUT NOME $8. @21 EMP $15. @37 ANO 4. ;
  IF ANO=1986 ;
```

```
RUN;
```

```
PROC PRINT;RUN;
```

Log

```
91 DATA DAD.DADOS86 ;
92     INFILE "A:\DADOS\CADASTRO.TXT" LRECL=50 MISSOEVER ;
93     INPUT NOME $8. @21 EMP $15. @37 ANO 4. ;
94     IF ANO=1986 ;
95 RUN;
```

NOTE: The infile "A:\DADOS\CADASTRO.TXT" is:
Filename=A:\DADOS\CADASTRO.TXT,
RECFM=V,LRECL=50,File Size (bytes)=22752,
Last Modified=01 de Julho de 2014 12h34min43,
Create Time=01 de Julho de 2014 12h28min44

NOTE: 550 records were read from the infile
"A:\DADOS\CADASTRO.TXT".

The minimum record length was 37. **Indicativo para uso da opção MISSOEVER**
The maximum record length was 40.

NOTE: The data set DAD.DADOS86 has 8 observations and 3 variables.

NOTE: DATA statement used (Total process time):

```
real time      0.02 seconds
cpu time       0.03 seconds
```

```
96
```

```
97 PROC PRINT;RUN;
```

NOTE: There were 8 observations read from the data set DAD.DADOS86.

NOTE: PROCEDURE PRINT used (Total process time):

```
real time      0.03 seconds
cpu time       0.00 seconds
```

The SAS System

08:46 Monday, June 30, 2014 2

Obs	NOME	EMP	ANO
1	JOAO	MALTA LTDA	1986
2	LUIS	ATLAS S.A.	1986
3	LICIA	ATLAS S.A.	1986
4	MIRIAM	MALTA LTDA	1986
5	MARIA	PARIS INSTITUTO	1986
6	RENATO	MALTA LTDA	1986
7	MARCIO	ATLAS S.A.	1986
8	ROSANE	MALTA LTDA	1986

Ex.23: Comando IF (Operadores de Comparação)

Arquivo: NOTAS.TXT

01	02	03	04	05	06	07	08	09	10	11
V	I	V	I	A	N	E		2	.	5
L	I	D	I	A				9	.	5
P	A	U	L	O				6	.	8
C	L	A	U	D	I	O		6	.	8
L	U	I	Z					4	.	9

```

DATA  NOTA ;
INFILE "C:\curso\sas\NOTAS.TXT" ;
INPUT  NOME $ 1-8  NOTA 9-11;
IF  NOTA < 3 THEN CONCEITO="E" ;
IF 3 <= NOTA < 5 THEN CONCEITO="D" ;
IF 5 <= NOTA < 7 THEN CONCEITO="C" ;
IF 7 <= NOTA < 9 THEN CONCEITO="B" ;
ELSE CONCEITO="A" ;

RUN;
PROC PRINT NOOBS ;
RUN;

```

The SAS System 12:12 Wednesday, July 2, 2014 1

NOME	NOTA	CONCEITO
VIVIANE	2.5	A
LIDIA	9.5	A
PAULO	6.8	A
CLAUDIO	6.8	A
LUIZ	4.9	A

O que está errado???

Todos os IFs são executados de forma independente, um após o outro.

Solução: IFs encadeados ou IFs embutidos ou IFs aninhados (“nested IFs”). Um IF dentro da lógica do IF anterior.

```

DATA  NOTA ;
INFILE "C:\curso\sas\NOTAS.TXT" ;
INPUT  NOME $ 1-8  NOTA 9-11;
IF  NOTA < 3 THEN CONCEITO="E" ;
ELSE IF 3 <= NOTA < 5 THEN CONCEITO="D" ;
ELSE IF 5 <= NOTA < 7 THEN CONCEITO="C" ;
ELSE IF 7 <= NOTA < 9 THEN CONCEITO="B" ;
ELSE CONCEITO="A" ;

RUN;
PROC PRINT NOOBS ;
RUN;

```

The SAS System 12:12 Wednesday, July 2, 2014 2

NOME	NOTA	CONCEITO
VIVIANE	2.5	E
LIDIA	9.5	A
PAULO	6.8	C
CLAUDIO	6.8	C
LUIZ	4.9	D

Ex.24: Comando IF (Operadores Lógicos)

Arquivo: CLASSE.TXT

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16
P	A	U	L	A			1		1	1		1	.	6	0
F	L	A	V	I	A		1		1	5		1	.	6	5
M	A	R	C	O	S		2		1	1		1	.	6	0
L	U	I	S				2		1	4		1	.	6	0

```
LIBNAME TESTE "C:\curso\sas" ;
DATA TESTE.CLASSE ;
  INFILE "C:\curso\sas\CLASSE.TXT";
  INPUT NOME $ 1-7 SEXO $ 8 IDADE 10-11 @13 ALTURA 4.2 ;
  IF (SEXO="1" AND ALTURA<1.55) OR (SEXO="1" AND IDADE<12)
    THEN CLASSE="100" ;
    ELSE IF (SEXO="2" AND ALTURA<1.55) OR (SEXO="2" AND IDADE<12)
      THEN CLASSE="200" ;
      ELSE CLASSE="300" ;
RUN;
PROC PRINT NOOBS ;
  VAR NOME CLASSE ;
RUN ;
```

The SAS System 12:41 Monday, January 26, 2015 2

NOME	CLASSE
PAULA	100
FLAVIA	300
MARCOS	200
LUIS	300

Ex.25: Comando IF (Operadores Lógicos)

```
DATA;
  ESTADO="SP";CIDADE="BAURU";
  IF ESTADO="RJ" OR ESTADO="SP" THEN REGIAO="SUDESTE" ;
  IF ESTADO="SP" AND CIDADE="CAMPINAS" THEN CEP="13000-000" ;
RUN;
PROC PRINT;RUN;
```

The SAS System 10:15 Thursday, September 14, 2017 2

Obs	ESTADO	CIDADE	REGIAO	CEP
1	SP	BAURU	SUDESTE	

Ex.26: Comando IF (Verdadeiro ou Falso)

```
data;
  if 100 then a="verdadeiro"; else a="falso";
  if 0 then b="verdadeiro"; else b="falso";
run;
proc print;run;
```

The SAS System 09:01 Thursday, August 17, 2017 9

Obs	a	b
1	verdadeiro	falso

14.4 - Comando WHERE

- **Tipo:** Comando de execução;

- **Uso:** Data Step, Proc Step e Opção de arquivos;

- Comando que possui como única ação, **selecionar registros**, se uma expressão for verdadeira;

- A expressão do comando WHERE só pode conter variáveis que já existam em arquivos SAS, ou seja, não podem ser utilizadas variáveis que estão sendo criadas no DATA Step, por comandos de atribuição (“=”), ou pelo comando INPUT;

- Além dos operadores de comparação e lógicos do comando IF, existem alguns operadores especiais no comando WHERE.

WHERE <expressão> ;

expressão Combinação de variáveis com operadores de comparação, operadores lógicos e/ou operadores especiais, que determinam uma condição. Se for verdadeira, o registro é selecionado, processado e salvo. Se for falso o registro é descartado.

Alguns operadores especiais:

BETWEEN-AND		entre um e outro
CONTAINS	?	contém

Ex.27: Comando WHERE

```
DATA TESTE;  
  SET ARQ.CADASTRO;  
  WHERE SALARIO BETWEEN 5000 AND 10000;
```

```
RUN;
```

```
PROC PRINT data=teste;  
  var nome salario;
```

```
RUN;
```

```
                                The SAS System                12:41 Monday, January 26, 2015  19
```

Obs	nome	salario
1	MOUA, PAULO	9548.36
2	MOUA, MARCO	9988.53
3	SONTAS, MARCO	5098.46
4	MEREDITE, JOAO	5023.91
5	MARUEL, MARCO	5157.09
6	ANJOA, MONICA	5066.22
7	LUILA, CARLA	5178.65
8	MEREDITE, LUIS	5123.16
9	MOUA, LIGIA	5047.05
10	MOUA, TANIA	5418.33
11	MOUA, CARLA	7221.58
12	MOUA, LAURA	7402.21

```
PROC PRINT DATA=ARQ.CADASTRO;  
  var nome;  
  WHERE NOME CONTAINS "SILVA";
```

```
RUN;
```

```
                                The SAS System                12:41 Monday, January 26, 2015  29
```

Obs	nome
17	YATAKA, SILVANA
21	SILVA, MADALENA
40	SILVA, CECILIA
44	SILVA, MARCO
197	SILVA, TANIA
213	SILVA, CARLA
278	SILVA, MARIA
283	SILVA, LICIA
376	SILVA, ROSANE
383	SILVA, MARCIO
399	SILVA, LAURA
411	SILVA, JOAO
423	SILVA, EDUARDO
440	SILVA, MONICA
450	SILVA, RENATO
453	SILVA, LIGIA
463	SILVA, MARCELO
468	SILVA, LUIS
498	SILVA, MIRIAM
502	SILVA, FRANCISCA
505	SILVA, FLAVIA
508	SILVA, ELIANE
532	SILVA, PAULO

14.5 - Comandos DO/END

- **Tipo:** Comando de execução;

- **Uso:** Data Step

- Comando que define a execução de um bloco de comandos do SAS;
- Todos os comandos existentes entre o comando DO e o comando END, deverão ser executados imediatamente;
- Comandos DO e END podem ser usados para executar um grupo de comandos quando se encontra uma condição, normalmente, para ampliar o comando IF-THEN/ELSE.

```
IF <expressão> THEN DO;  
    <comando1> ;  
    <comando2> ;  
    ...  
    <comandon> ;  
END ;
```

Ex.28: Comando DO/END

```
DATA TRABALHO ;  
  INFILE "C:\curso\sas\EMPRESA.DAT" ;  
  INPUT NOME $ 1-8 DEP_NO $ 10-12 COM 14-17 SALARIO 19-23 ;  
  IF DEP_NO="201" THEN DO ;  
    DEP="VENDAS" ;  
    SAL=COM+SALARIO ;  
  END ;  
  ELSE DO ;  
    DEP="ADMINISTRATIVO" ;  
    SAL=SALARIO ;  
  END ;  
RUN ;  
  
PROC PRINT NOOBS ;  
RUN ;
```

The SAS System 12:41 Monday, January 26, 2015 30

NOME	DEP_NO	COM	SALARIO	DEP	SAL
CARLOS	201	1500	1500	VENDAS	3000
ELIANE	101	.	1800	ADMINI	1800
ALVARO	101	.	3800	ADMINI	3800
FATIMA	201	500	1500	VENDAS	2000

14.6 - Comando LENGTH

- **Tipo:** Comando de declaração;

- **Uso:** Data Step;

- Comando que define uma variável, o tipo e o tamanho em bytes. Deve ser posicionado sempre antes da primeira utilização da variável no Data step.

```
LENGTH   variável [$] <tamanho>  
          . . .  
          variável [$] <tamanho> ;
```

variável Nome da variável que deseja definir;

\$ Caractere opcional, que indica que a variável será alfanumérica;

tamanho Tamanho, em bytes, da variável.

Ex.29: Comando LENGTH

```
DATA TRABALHO ;  
  INFILE "C:\curso\sas\EMPRESA.DAT" ;  
  INPUT NOME $ 1-8 DEP_NO $ 10-12 COM 14-17 SALARIO 19-23 ;  
  LENGTH DEP $ 14 ;  
  IF DEP_NO="201" THEN DO ;  
    DEP="VENDAS" ;  
    SAL=COM+SALARIO ;  
  END ;  
  ELSE DO ;  
    DEP="ADMINISTRATIVO" ;  
    SAL=SALARIO ;  
  END ;  
RUN ;  
PROC PRINT NOOBS ;  
RUN ;
```

The SAS System 12:41 Monday, January 26, 2015 31

NOME	DEP_NO	COM	SALARIO	dep	SAL
CARLOS	201	1500	1500	VENDAS	3000
ELIANE	101	.	1800	ADMINISTRATIVO	1800
ALVARO	101	.	3800	ADMINISTRATIVO	3800
FATIMA	201	500	1500	VENDAS	2000

14.7 - Comando DO/END Iterativo

- **Tipo:** Comando de execução;

- **Uso:** Data Step;

- Comando utilizado para repetir um grupo de comandos “n” vezes.

DO <índice> = <valor1>, <valor2>, ... , <valorn> ;

DO <índice> = <início> **TO** <fim> [**BY** <passo>]
[**WHILE/UNTIL** <(expressão)>];

índice	Variável auxiliar que armazena o valor da contagem do "loop".
início	Variável ou valor inicial do "loop".
fim	Variável ou valor final do "loop".
passo	Variável ou valor opcional que indica o incremento do "loop".
expressão	Campo opcional que representa uma combinação de variáveis que determinam uma condição para execução do "loop". A expressão deve vir entre parênteses.

Ex.30: Comando DO/END Iterativo

```
DO MES="JAN", "FEV", "MAR" ; (3 vezes) ;END;

DO I = 1 TO 120 ; (120 vezes) ;END;

DO I = 1 TO 120 BY 2 ; (60 vezes) ;END;

DO I = 120 TO 1 BY -1 ; (120 vezes) ;END;

DO I = 1 TO 10 WHILE (X < Y) ; (10 vezes ou enquanto X < Y) ;END;

DO I = 2 TO 20 BY 2 UNTIL ((X/3)>Y) ; (10 vezes ou até (X/3)>Y) ;END;

DO I = 10 TO 0 BY -1 WHILE (MES="JAN");(11 vezes ou enquanto o MES=JAN);END;
```

```
data teste;
  set arq.cadastro;
  where salario between 2000 and 2100;
  novosal=salario;
  do i=1 to 10 while (novosal<2200);
    novosal=novosal*1.05;
  end;
run;

proc print data=teste;
  var nome salario novosal i;
run;
```

```

                                The SAS System                                12:41 Monday, January 26, 2015 113
Obs      nome                    salario    novosal    (i)
1        SUNAY,FRANCISCA        2070.60    2282.84    3
2        MENDES,ELIANE          2071.18    2283.48    3
3        GUEDES,CARLA           2076.92    2289.80    3
4        LONAS,FLAVIA           2020.77    2227.90    3
5        HORTA,MONICA           2098.94    2203.89    2
6        MENDES,CARLA           2095.91    2200.71    2
7        APARECIDO,PAULO        2037.42    2246.26    3
8        BENTES,MARCO           2025.92    2233.58    3
9        BENTES,JOAO            2026.52    2234.24    3
10       YATAKA,PAULO           2056.53    2267.32    3
```

O valor de i, representa o número de vezes que o salário foi incrementado?

NÃO!

14.8 - Comando DO WHILE

- **Tipo:** Comando de execução;

- **Uso:** Data Step;

- Comando que repete um grupo de comandos **enquanto** uma expressão for verdadeira;

- A expressão é validada no **início** do “loop”. Se a expressão for falsa, os comandos dentro do “loop”, nunca serão executados;

- **ATENÇÃO!** Este tipo de “loop” pode incorrer em **ciclo infinito**, se os valores da expressão não se alterarem dentro do “loop”.

```
DO WHILE <(expressão)> ; <== ( Validação no início )  
    ...  
END;
```

expressão Combinação de variáveis que determina uma condição para a execução do "loop", **entre parênteses**.

Ex.31: Comando DO WHILE

```
DATA ;  
  X=0; N=0;  
  DO WHILE ( N LE 5 ) ; <== ( Validação no início. Executa 6 vezes )  
    X=X+N**2 ;  
    N=N+1 ;  
  END ;  
RUN;
```

O N, que valida a expressão, está sendo alterado no "loop".

```
PROC PRINT;  
RUN;
```

The SAS System

Obs	X	N
1	55	6

14.9 - Comando DO UNTIL

- **Tipo:** Comando de execução;

- **Uso:** Data step;

- Comando que repete um grupo de comandos **até** a expressão ser verdadeira;

- A expressão é validada no **fim** do “loop”. Se a expressão for falsa, os comandos dentro do “loop”, **serão executados**, pelo menos, **uma vez**.

- **ATENÇÃO!** Este tipo de “loop” pode incorrer em **ciclo infinito**, se os valores da expressão não se alterarem dentro do “loop”.

DO UNTIL <(expressão)> ;

...

END;<== (Validação no fim)

expressão Combinação de variáveis que determina uma condição para a execução do "loop", **entre parênteses**.

Ex.32: Comando DO UNTIL

```
DATA ;  
  X=0; N=0;  
  DO UNTIL (N GT 5) ;  
    X=X+N**2 ;  
    N=N+1 ;  
  END ; <== ( Validação no fim. Executa 6 vezes )  
RUN;  
  
PROC PRINT;RUN;
```

The SAS System

Obs	X	N
1	55	6

14.10 - Comandos DROP e KEEP

- **Tipo:** Comandos de declaração;

- **Uso:** Data step e Opção de arquivos;

- O comando DROP, determina que **algumas variáveis não sejam salvas** em um arquivo SAS;

- O comando KEEP, determina que **somente algumas variáveis sejam salvas** em um arquivo SAS;

- Pode ser posicionado em qualquer lugar de um Data Step;

- Esses dois comandos são exclusivos, ou se usa o comando DROP, ou se usa o comando KEEP, na lógica de um programa SAS.

DROP <variável1> <variável2> ... <variáveln>;

variável Nome de uma variável que não será salva em um arquivo SAS (Data Set).

KEEP <variável1> <variável2> ... <variáveln>;

variável Nome de uma variável que será salva em um arquivo SAS (Data Set).

Ex.33: Comandos DROP e KEEP

```
data dados;  
  infile "c:\curso\sas\dados.dat";  
  input nome $ 1-30 sexo $ 31 idade 32-33 peso 34-39 .2  
        altura 40-43 .2 ecivil $ 54 rg $ 57-71  
        cpf $ 72-82 funcao $ 98-109 ;
```

```
run;
```

```
proc contents short;run;
```

14:33 Monday, February 2, 2015 1

The CONTENTS Procedure

Alphabetic List of Variables for WORK.DADOS

altura cpf ecivil funcao idade nome peso rg sexo

```
DATA SUBARQ1 ;  
  SET DADOS ;  
  DROP PESO ALTURA ECIVIL ;
```

```
RUN;
```

```
proc contents short;run;
```

14:33 Monday, February 2, 2015 2

The CONTENTS Procedure

Alphabetic List of Variables for WORK.SUBARQ1

cpf funcao idade nome rg sexo

```
DATA SUBARQ2 ;  
  SET DADOS ;  
  KEEP NOME RG CPF ;
```

```
RUN;
```

```
proc contents short;run;
```

14:33 Monday, February 2, 2015 3

The CONTENTS Procedure

Alphabetic List of Variables for WORK.SUBARQ2

cpf nome rg

14.11 - Opções de Arquivos: DROP= e KEEP=

- Todo arquivo com estrutura SAS possui diversas opções de configuração, que podem ser especificados à direita do nome de cada arquivo, **entre parênteses**;
- São opções **do arquivo**, que tem sua ação vinculada ao comando no qual o arquivo está sendo especificado;
- Normalmente quando se faz referência a um arquivo em um programa SAS, a ação, ou será de **leitura dos dados**, ou será de **gravação dos dados**.

<arquivo SAS> [(<opção1> <opção2> ... <opçãoon>)] ;

Opções KEEP e DROP

<arquivo SAS> (**DROP**=<variável> <variável> ...) ;
<arquivo SAS> (**KEEP**=<variável> <variável> ...) ;

- Ação vinculada ao comando DATA (**gravação de dados**)

DROP Não grava
KEEP Somente grava

- Ação vinculada ao comando SET (**leitura de dados**)

DROP Não leia
KEEP Somente leia

Ex.34: Opções: DROP= e KEEP=

```
data dados;  
  infile "c:\curso\sas\dados.dat";  
  input nome $ 1-30 sexo $ 31 idade 32-33 peso 34-39 .2  
        altura 40-43 .2 ecivil $ 54 rg $ 57-71  
        cpf $ 72-82 funcao $ 98-109 ;  
run;  
  
DATA SUBARQ1 (DROP= PESO ALTURA ECIVIL) ;  
  SET DADOS ;  
RUN;  
  
DATA SUBARQ2 ;  
  SET DADOS (KEEP= NOME RG CPF) ;  
RUN;
```

14.12 - Comando DELETE

- **Tipo:** Comando de execução;

- **Uso:** Data Step;

- Comando que informa ao SAS que deve ser parado o processamento do registro atual e retornar imediatamente para o início do Data Step para uma nova iteração;

- O objetivo do comando é de não permitir a gravação do registro lido no novo arquivo SAS, mas permite trabalhar com os dados lidos;

- Normalmente esse comando é utilizado junto com o comando IF/THEN/ELSE, que irá determinar uma condição para execução do comando DELETE.

IF <expressão> THEN DELETE ;

Ex.35:Comando DELETE

Arquivo: PRODUTO.TXT

01	02	03	04	05	06	07	08	09
1	7		1	0		5	2	5
4	2		5	0		-	6	
3	1			5		1	0	0
2	4			0				3
5	1			1				5

```
DATA PRODUTO ;  
  INFILE "C:\curso\sas\PRODUTO.TXT";  
  INPUT DEP $ 1-2 UNIDADE 4-5 CUSTO 7-9 ;  
  IF UNIDADE <= 0 OR CUSTO <= 0 THEN DELETE ;  
  CUSTOUNI=CUSTO/UNIDADE ;
```

```
RUN;
```

```
PROC PRINT NOOBS ;
```

```
RUN ;
```

14:33 Monday, February 2, 2015 8

DEP	UNIDADE	CUSTO	CSTOUNI
17	10	525	52.5
31	5	100	20.0
51	1	5	5.0

14.13 - Comando OUTPUT

- **Tipo:** Comando de execução;

- **Uso:** Data Step;

- O SAS automaticamente codifica um comando OUTPUT (implícito), no final do Data step, para salvar os dados que foram processados.

```
DATA ... ;  
    ... ;  
    ... ;  
    ... ;  
    <output>  
RUN ;
```

- A codificação explícita de um comando OUTPUT, funciona de duas formas:

1 - Especifica ao SAS que o registro que está sendo processado deve ser gravado, **imediatamente**, e não, ao final do Data Step (OUTPUT implícito), em todos os arquivos especificados no comando DATA.

2 - Especifica para qual arquivo SAS deve ser gravado o registro que está sendo processado.

- Normalmente é utilizado junto com o comando IF-THEN/ELSE de acordo com uma condição;

ATENÇÃO! Ao se codificar um comando OUTPUT no programa, o SAS desabilita o **OUTPUT implícito** ao final do DATA Step. A ação de salvar o dado fica por conta da lógica do programa; da posição do comando OUTPUT.

OUTPUT [arquivo1 ... arquivon];

IF <expressão> THEN OUTPUT [arquivo1 ... arquivon];

expressão Combinação de variáveis e valores que determinam uma condição a ser testada.

arquivo Opcional, nome de um arquivo SAS, **especificado e criado pelo comando DATA.**

Ex.36: Comando OUTPUT: Salvar Registro

Arquivo: SAIDA.TXT

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18
R	O	S	A	N	A		5					0		1	5	0	0
P	A	U	L	O			1	0	1	.	1	0		1	0	0	0
M	A	R	I	A			1	0	1	.	1	5			8	0	0
C	L	A	U	D	I	A	1	0	1	.	2	0			7	5	0
C	A	R	L	O	S		8					0		1	8	0	0

```
LIBNAME K1 "C:\curso\sas" ;
DATA K1.VARIACAO ;
  INFILE "C:\curso\sas\SAIDA.TXT";
  INPUT NOME $ 1-7 NIVEL $ 8-9 VAL 10-13 SALARIO 14-18;
  SAL=0;
  IF NIVEL="10" THEN DO I=1 TO 2 ;
    SAL=SAL+SALARIO*VAL ;
    OUTPUT ;
  END ;
  DROP I VAL;
RUN;
PROC PRINT ; RUN ;
```

14:33 Monday, February 2, 2015 12

Obs	NOME	NIVEL	SALARIO	SAL
1	PAULO	10	1000	1100
2	PAULO	10	1000	2200
3	MARIA	10	800	920
4	MARIA	10	800	1840
5	CLAUDIA	10	750	900
6	CLAUDIA	10	750	1800

Ex.37: Comando OUTPUT: Direcionar e Salvar Registro

```
LIBNAME A "C:\curso\sas" ;
DATA A.HOMENS A.MULHERES ;
  SET A.CADASTRO ;
  IF SEXO="F" THEN OUTPUT A.MULHERES ;
  ELSE OUTPUT A.HOMENS ;
RUN;

PROC PRINT DATA=A.MULHERES ; VAR NOME ; RUN ;
PROC PRINT DATA=A.HOMENS ; VAR NOME ; RUN ;
```

3º LABORATÓRIO

Utilização dos comandos de programação em DATA Step

Em todos os programas, utilize o arquivo de dados CADAstro, gerado no 1º laboratório.

1 - Monte um programa que crie um novo arquivo SAS, somente com os dados das pessoas do **sexo feminino e casadas**. Gere um relatório com os dados desse arquivo. A variável que representa o **ESTADO CIVIL**, está codificada por números, que representam:

1 = solteiro

2 = casado

3 = separado.

Comandos possíveis: DATA, SET, IF/THEN/ELSE ou WHERE, PROC PRINT

2 - Monte um programa que crie um novo arquivo SAS. Utilizando o comando **IF**, analise a variável que representa o **ESTADO CIVIL** e crie três variáveis novas; cada uma, irá acumular o número de pessoas casadas, o número de pessoas solteiras e o número de pessoas separadas; Gere um relatório com os dados deste arquivo;

OBS: Variável acumuladora: **X=X+1**

O último registro processado terá o valor total da distribuição do **ESTADO CIVIL**.

Comandos possíveis: DATA, SET, IF/THEN/ELSE, = , PROC PRINT

3 - Monte um programa que crie um novo arquivo SAS:

- A partir da análise da variável **FILHOS**, crie uma nova variável (**DEP**) e altere a variável **SALARIO** :

Se possui filhos - **DEP = "COM DEPENDENTES"**
- **SALARIO** será o resultado da multiplicação da variável **SALARIO** por **1.1**;

Se não possui filhos - **DEP = "NÃO POSSUI DEPENDENTES"**

- Gere um relatório com os dados deste arquivo;

Comandos possíveis: DATA, SET, IF/THEN/ELSE, DO/END, =, , PROC PRINT

4 - Monte um programa que crie um novo arquivo SAS:

- Selecione as pessoas, **empregadas**, que ganham menos de **R\$ 3.000,00**;

- **Aumente** o valor do salário dessas pessoas em 10%, em até 12 vezes (relativos a 12 meses), ou, até o salário atingir um valor superior a R\$ 5.000,00.

- Para aumentar o valor do salário, utilize a equação: **SALARIO=SALARIO*1.1**

- **Salve** os valores incrementados, um a um;

- O arquivo só deverá conter as variáveis: **NOME, EMPRESA, FUNCAO** e **SALARIO** ;

- Gere um relatório com os dados deste arquivo;

Comandos possíveis: DATA, SET, IF/THEN/ELSE ou WHERE, DO iterativo/END com opção WHILE ou UNTIL, =, OUTPUT, DROP ou KEEP, PROC PRINT.

OBS: Utilize o comando **IF-THEN/ELSE** ou **WHERE** para selecionar as pessoas. Utilize o comando **DO/END Iterativo** para aumentar o salário. Utilize o comando **OUTPUT** para salvar os dados e utilize o comando **KEEP** ou **DROP** para determinar quais as variáveis que serão arquivadas.

5 - Monte um programa que crie três novos arquivos SAS, permanentes, no diretório c:\curso\sas : SA, LTDA e INSTITUTO :

- Elimine as pessoas que ocupam **FUNCAO** de "DIRETOR" e "GERENTE";
- Selecione os dados, de maneira que, cada arquivo só possua os dados referentes a cada **TIPO DE EMPRESA**.
- Gere um relatório para cada um dos arquivos;

Comandos possíveis: DATA, SET, IF/THEN/ELSE ou WHERE, OUTPUT, PROC PRINT

OBS: Utilize o comando **DATA**, para criar os arquivos. Utilize o comando **IF/THEN/ELSE**, para selecionar e salvar os dados. Utilize o comando **OUTPUT** para direcionar a gravação dos registros para os arquivos definidos no comando **DATA**.

6 - Monte um programa, com um único DATA Step, que crie sete novos arquivos SAS, permanentes, (PESSOAL , TRABALHO , FAMILIA, SOLTEIRO, CHEFE, SUB e DESEMPREGADOS) no diretório c:\curso\sas ;

- Cada arquivo deverá conter os seguintes dados:

PESSOAL	Somente as variáveis: NOME, SEXO, IDADE, PESO, ALTURA, ANIVERSARIO, ESTADO CIVIL, FILHOS, RG e CPF , e todos os registros;
TRABALHO	Somente as variáveis: NOME, RG, CPF, EMPRESA, FUNCAO, ADMISSAO, SALARIO e TIPO DE EMPRESA e pessoas que possuam emprego ;
FAMILIA	Todas as variáveis e somente as pessoas que possuam filhos;
SOLTEIRO	Todas as variáveis e somente as pessoas solteiras;
CHEFE	Todas as variáveis e somente as pessoas que ocupam cargo de chefia: GERENTE ou DIRETOR ;
SUB	Todas as variáveis e as pessoas que não ocupam cargos de chefia;
DESEMPREGADOS	Todas as variáveis e somente as pessoas desempregadas;

- Gere relatórios para ver o conteúdo de cada arquivo;

Comandos possíveis: DATA, SET, IF/THEN/ELSE, OUTPUT, PROC PRINT

7 – Exercício opcional.

Com os dados do arquivo **CADASTRO**, monte um programa SAS que gere um novo arquivo somente com as pessoas entre 20 e 30 anos, sendo que, cada pessoa selecionada receberá um bonus de R\$ 3.500,00 ao salário anual, até completar 30 anos. Crie uma variável nova (**BONUS**) com o valor total de bonus recebido por cada pessoa selecionada, e uma variável (**NUM**) com o número de vezes que a pessoa recebeu o bonus;

- Utilize o comando **WHERE** com um operador especial para selecionar os dados;
- Utilize o comando **DO WHILE** ou **UNTIL** para processar a variável com o valor total do bonus recebido;
- Não altere a variável **IDADE** do arquivo. Crie novas variáveis auxiliares:

IDADE_INICIAL que será igual a **IDADE**

IDADE_FINAL que será a **IDADE** incrementada

- Gere um relatório com as variáveis: **NOME, IDADE_INICIAL, IDADE_FINAL, SALARIO, BONUS , NUM;**

OBS: Atenção à lógica do **WHILE** ou **UNTIL**, para que o programa não entre em “loop”.

15 - FUNÇÕES DO SAS

15.1 - Introdução

- Toda função representa a execução de uma determinada rotina que irá retornar um determinado valor.
- Toda função SAS é recursiva. Função pode ser executada dentro de outra função, como parâmetro da função;
- As funções no SAS, normalmente, fazem parte de um comando de **alocação**, durante a criação ou alteração de uma variável no DATA Step, mas também podem ser utilizadas em expressões lógicas, em um comando **IF** ou **WHERE**.

<variável> = <função>([OF] *arg1, arg2, . . . , argn*) ;

variável	Nome até no máximo 32 caracteres, aonde será armazenado o valor retornado da execução da função.
função	Palavra-chave que identifica a função a ser processada.
arg	Parâmetros de entrada para processamento da função: variável, expressão aritmética, valor constante.
OF	Parâmetro opcional especial para as funções estatísticas , que indica o processamento num intervalo de argumentos.

Tipos de funções:

- Funções de Tratamento de Caractere;
- Funções de Tratamento de Data e Hora;
- Funções Matemáticas;
- Funções de Estatísticas Descritivas;
- Funções Trigonométricas e Hiperbólicas;
- Funções de Truncamento;
- Funções para Geração de Números Randômicos;
- Funções de Probabilidade;
- Funções Financeiras;
- Funções de Macro;

15.2 - Funções de Tratamento de Caractere

15.2.1 - Funções UPCASE , LOWCASE e PROPCASE

- Funções que convertem o conteúdo de uma variável ou expressão, para caracteres maiúsculos (UPCASE), minúsculos (LOWCASE) e apenas a primeira letra de todas as palavras em maiúsculo e o restante em minúsculo (PROPCASE).

<variável>=UPCASE (arg) ;
<variável>=LOWCASE (arg) ;
<variável>=PROPCASE (arg) ;

variável Nome até 32 caracteres que armazenará o resultado.

arg Variável ou expressão caractere, cujo conteúdo será convertido para letras maiúsculas, minúsculas ou misto.

15.2.2 - Função COMPRESS

- Função que retira caracteres, individualmente, de um argumento.

<variável>=COMPRESS(arg1 [,"arg2"]);

variável Nome até 32 caracteres que armazenará o resultado do processamento da função;

arg1 Primeiro argumento, determina a variável ou expressão caractere cujo conteúdo será compactado;

arg2 Segundo argumento, **opcional e entre aspas**, determina um ou mais caracteres que serão retirados, individualmente, do primeiro argumento.

OBS: Se o segundo argumento não for especificado, o padrão será retirar o caractere “branco” .

15.2.3 - Função CATX

- Função que concatena argumentos, retirando os espaços em branco à esquerda e a direita, e adiciona um delimitador entre os argumentos.

<variável>=CATX(dlim, arg1, arg2,...,argn);

variável Nome até 32 caracteres que armazenará o resultado, numérico, do processamento da função;

dlim Delimitador que será inserido entre cada argumento. Pode ser uma variável ou, um valor entre aspas;

argn Argumentos, determina as variáveis ou expressões caracteres cujos conteúdos serão concatenados;

OBS: O delimitador deverá vir informado entre aspas. Se a variável que armazenará o resultado não tiver sido definida, será então definida pela função, com **200 bytes**.

15.2.4 - Função COUNT

- Função que verifica o número de ocorrências de um valor, em uma variável ou expressão.

<variável>=COUNT(arg1,arg2,[“arg3”]);

variável Nome até 32 caracteres que armazenará o resultado, numérico, do processamento da função;

arg1 Variável ou expressão caractere cujo conteúdo será manipulado.

arg2 Variável ou, valor entre aspas que será contado;

arg3 **Opcional**, determina regras para a busca das ocorrências.

Ex. 38: Funções CATX, COMPRESS, PROPCASE e COUNT

```
data teste;
  rua="<Rua Dr. ANTÔNIO GUIMARÃES>";
  numero="<1260>";
  bloco="<BLOCO-A>";
  apt="<APT-101>";
  cidade="<CAMPINAS>";
  estado="<SÃO PAULO>";
  cep="<13000-000>";
  a=catx(", ", rua, numero, bloco, apt, cidade, estado, cep);
  b=compress(a, "<>");
  c=propcase(b, ", ");
  d=propcase(b, " , ");
  e=propcase(b, " , -");
  f=count(e, "-");
run;
proc print;run;
```

The SAS System 09:20 Monday, August 21, 2017 15

Obs	rua	numero	bloco	apt	cidade	estado	cep
1	<Rua Dr. ANTÔNIO GUIMARÃES>	<1260>	<BLOCO-A>	<APT-101>	<CAMPINAS>	<SÃO PAULO>	<13000-000>
Obs			a				
1	<Rua Dr. ANTÔNIO GUIMARÃES>,<1260>,<BLOCO-A>,<APT-101>,<CAMPINAS>,<SÃO PAULO>,<13000-000>						
Obs			b				
1	Rua Dr. ANTÔNIO GUIMARÃES,1260,BLOCO-A,APT-101,CAMPINAS,SÃO PAULO,13000-000						
Obs			c				
1	Rua dr. antônio guimarães,1260,Bloco-a,Apt-101,Campinas,São paulo,13000-000						
Obs			d				
1	Rua Dr. Antônio Guimarães,1260,Bloco-a,Apt-101,Campinas,São Paulo,13000-000						
Obs			e			f	
1	Rua Dr. Antônio Guimarães,1260,Bloco-A,Apt-101,Campinas,São Paulo,13000-000						3

15.2.5 - Função SUBSTR= ou =SUBSTR

- Função que efetua a manipulação (extrair ou substituir) do conteúdo de uma variável do tipo caractere.

<variável>=SUBSTR (arg1, arg2, [arg3]) ;
SUBSTR (arg1, arg2, [arg3])=<variável ou valor>;

- variável** Nome até 32 caracteres que armazenará o resultado.
- arg1** Variável ou expressão caractere cujo conteúdo será manipulado.
- arg2** Valor que indica a posição inicial do conteúdo da variável que será manipulado.
- arg3** **Opcional**, determina o número de caracteres do conteúdo da variável que se deseja manipular. Se não for especificado, a função pegará todos os caracteres do valor do segundo argumento até o final.

15.2.6 - Função TRANWRD

- Função que substitui todas as ocorrências de um valor por outro em uma variável ou expressão.

<variável>=TRANWRD(arg1, arg2 , arg3);

- variável** Nome até 32 caracteres que armazenará o resultado, numérico, do processamento da função;
- arg1** Variável ou expressão caractere cujo conteúdo será manipulado.
- arg2** Valor de uma variável ou, expressão entre aspas que será substituído;
- arg3** Valor de uma variável ou, expressão entre aspas que substituirá pelo valor do argumento 2;

15.2.7 - Função FIND

- Função que retorna um valor numérico que representa a primeira posição de um determinado caractere dentro do argumento.

<variável>=FIND(arg1, arg2[,arg3][,arg4]) ;

variável Nome até 32 caracteres que armazenará o resultado.

arg1 Variável ou expressão caractere cujo conteúdo será manipulado.

arg2 Caracteres, **entre aspas**, que se deseja localizar no arg1.

arg3 Caractere que configura um recurso de busca:

i ou I	Ignora se é maiúsculas ou minúsculas;
t ou T	Remove “brancos” à esquerda do arg1 e arg2.

arg4 Determina a posição inicial de busca no arg1:

1	Valor padrão. Inicia a busca na coluna 1 e caminha para a direita;
-n	Inicia a busca na coluna n e caminha para esquerda.

OBS: Se os caracteres especificados não forem encontrados, a função **FIND** retornará o valor **zero**.

15.2.8 - Função SCAN

- Função que retorna um valor de acordo com a sua posição relativa em relação a um ou vários delimitadores.

<variável>=SCAN(arg1, arg2[,"arg3"][,"arg4"]);

variável Nome até 32 caracteres que armazenará o resultado.

arg1 Variável ou expressão caractere cujo conteúdo será manipulado.

arg2 Valor numérico que indica a posição em relação a delimitadores.

arg3 Lista de delimitadores, entre aspas. Qualquer caractere pode ser um delimitador.

arg4 Adiciona regras para o uso de delimitadores e busca de valores.

OBS: O delimitador deverá vir informado entre aspas. Se nenhum delimitador for informado o caractere branco será o delimitador. Se a variável que armazenará o resultado não tiver sido definida, será então definida pela função com **200 bytes**.

15.2.9 - Função LENGTH

- Função que informa o tamanho, em bytes, do argumento.

<variável>=LENGTH(arg);

variável Nome até 32 caracteres que armazenará o resultado, numérico, do processamento da função;

arg Argumento, determina a variável ou expressão caractere cujo conteúdo será determinado o tamanho;

OBS: O Valor retornado pela função LENGTH, sempre será numérico.

Ex.39: Função: FIND e SCAN

```
DATA POSICAO;

    REGIAO="FRANÇA#ALEMANHA#ESPANHA-ITÁLIA";

    POS=FIND(REGIAO,"Espanha"); OUTPUT ;
    POS=FIND(REGIAO,"Espanha","i"); OUTPUT ;
    POS=FIND(REGIAO,"nha","I"); OUTPUT ;
    POS=FIND(REGIAO,"nha","I",-30);OUTPUT;
    POS=FIND(REGIAO,"I"); OUTPUT ;
    POS=FIND(REGIAO,"ING"); OUTPUT ;

    LOCAL=SCAN(REGIAO,1,"#");OUTPUT;
    LOCAL=SCAN(REGIAO,-1,"#"); OUTPUT;
    LOCAL=SCAN(REGIAO,3,"#"); OUTPUT;
    LOCAL=SCAN(REGIAO,2,"-"); OUTPUT;
    /* Selecionar apenas ESPANHA */
    LOCAL=SCAN(REGIAO,3,"#"); OUTPUT;
    LOCAL=SCAN(REGIAO,1,"-"); OUTPUT;
    LOCAL=SCAN(REGIAO,-2,"-"); OUTPUT;

    LOCAL=SCAN(REGIAO,3,"#-"); OUTPUT;
    LOCAL=SCAN(REGIAO,-2,"#-"); OUTPUT;

RUN;

PROC PRINT;
    var pos local;
RUN ;
```

The SAS System

09:20 Monday, August 21, 2017 23

Obs	POS	LOCAL
1	0	
2	17	
3	13	
4	21	
5	25	
6	0	
7	0	FRANÇA
8	0	ESPANHA-ITÁLIA
9	0	ESPANHA-ITÁLIA
10	0	ITÁLIA
11	0	ESPANHA-ITÁLIA
12	0	FRANÇA#ALEMANHA#ESPANHA
13	0	FRANÇA#ALEMANHA#ESPANHA
14	0	ESPANHA
15	0	ESPANHA

Ex.40: Funções Caracteres: TRANWRD,COMPRESS,SUBSTR,LENGTH

Arquivo: RG.TXT

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18
5	8	4	1	6	6	s	s	p	/	d	f						
2	3	4	s	s	p	/	s	p									
S	S	P	P	R	1	5	7	8	9	0	1						
2	2	3	9	7	4	8	8	s	s	p	/	r	s				
8	6	7	9	4	7	E	X	É	R	C	I	T	O				
1	2	3	4	5	6	S	S	P	/	G	O						
s	s	p	B	S	B	8	6	7	9	4	7						

```

DATA RGSSP (DROP=AUX) ERRO;
  INFILE "C:\curso\sas\RG.TXT";
  INPUT RG $ 18. ;
  /* 1°. Tratamento: Tudo em maiúsculo */
  RG=UPCASE (RG) ;
  /* 2°. Tratamento: Existe "/" separando SSP do estado ? */
  AUX= FIND (RG, "SSP/");
  IF AUX=0 THEN RG=TRANWRD (RG, "SSP", "SSP/");
  /* 3°. Tratamento: Existe a palavra SSP ? */
  AUX= FIND (RG, "SSP");
  IF AUX=0
    THEN OUTPUT ERRO;
  /* 4°. Tratamento: Aonde está localizado a palavra SSP ? */
  ELSE IF AUX=1
    THEN DO;
      RG_SSP=COMPRESS (RG, "0123456789") ;
      RG_NUM=SUBSTR (RG, LENGTH (RG_SSP)+1);
      OUTPUT RGSSP;
      END;
    ELSE DO;
      RG_NUM=SUBSTR (RG, 1, AUX-1) ;
      RG_SSP=SUBSTR (RG, AUX) ;
      OUTPUT RGSSP;
      END;
RUN;

PROC PRINT DATA=RGSSP NOOBS ;
VAR RG RG_NUM RG_SSP;
RUN ;

```

The SAS System

16:38 Tuesday, August 22, 2017 23

RG	RG_NUM	RG_SSP
584166SSP/DF	584166	SSP/DF
234SSP/SP	234	SSP/SP
SSP/PR1578901	1578901	SSP/PR
22397488SSP/RS	22397488	SSP/RS
123456SSP/GO	123456	SSP/GO
SSP/BSB867947	867947	SSP/BSB

15.3 - Funções de Numéricas

15.3.1 - Função INT

- Função que retorna com o valor inteiro de uma variável ou de uma expressão.

<variável> = INT (arg) ;

variável Nome até 32 caracteres que armazenará o resultado.

arg Variável ou expressão numérica, cujo conteúdo se deseja obter o valor inteiro (**sem decimais e sem arredondar**).

15.3.2 - Função ROUND

- Função que retorna um valor arredondado de uma variável ou expressão numérica.

<variável> = ROUND (arg1, [arg2]) ;

variável Nome até 32 caracteres que armazenará o resultado.

arg1 Variável ou expressão numérica cujo conteúdo se deseja arredondar.

arg2 **Opcional**, especifica o múltiplo do arredondamento e a posição numérica aonde será feita o arredondamento (**1** , **10** , . . . ou **0.1** , **0.01** , . . .). Se não for especificado, a função arredondará para um valor inteiro (**1**).

OBS: Analisando os valores da **esquerda para direita**, ao se determinar qual a **posição numérica** que será arredondada, **analisa-se a posição posterior** (à direita), de acordo com as regras de arredondamento no SAS:

0,00 até 4,99 (arredonda para “baixo”, mas não altera a **posição numérica**).
5,00 até 9,99 (arredonda para “cima” e adiciona +1 à **posição numérica**).

Ex.: 542,86 ≈ 540



542,86 ≈ 543



542,86 ≈ 542,9



Ex.41: Funções de Truncamento

Arquivo: arredondamento.txt

01	02	03	04	05	06	07
3	2	6	.	5	4	
1	9	7	7	.	0	9
5	5	8	.	8	8	
1	5	8	8	.	2	9

```
DATA DADOS ;  
  INFILE "C:\curso\sas\arredondamento.txt";  
  INPUT X 7.;  
  TRUNCA = INT ( X ) ;  
  INTEIRO = ROUND ( X , 1 ) ;  
  DEZENA = ROUND ( X , 10 ) ;  
  CENTENA = ROUND ( X , 100 ) ;  
  DECIMO = ROUND ( X , 0.1 ) ;  
  CENTES = ROUND ( X , 0.01 ) ;  
  MULT5=ROUND ( X , 5 ) ;  
RUN;  
  
PROC PRINT DOUBLE;  
RUN ;
```

The SAS System

Obs	X	TRUNCA	INTEIRO	DEZENA	CENTENA	DECIMO	CENTES	MULT5
1	326.54	326	327	330	300	326.5	326.54	325
2	1977.09	1977	1977	1980	2000	1977.1	1977.09	1975
3	558.88	558	559	560	600	558.9	558.88	560
4	1588.29	1588	1588	1590	1600	1588.3	1588.29	1590

15.4 - Funções Estatísticas Descritivas

15.4.1 - Função SUM

- Função que efetua um somatório de valores, não “*missing*”, entre variáveis numéricas, para cada registro de dados.

<variável> = SUM ([OF] *var1*, . . . *varn*) ;

Variável	Nome até 32 caracteres que armazenará o resultado.
<i>var1</i>, . . . <i>varn</i>	Nome das variáveis numéricas que se deseja efetuar um somatório.
OF	Opcional; identifica um intervalo de variáveis.

15.4.2 - Função MEAN

- Função que efetua uma média aritmética entre os valores, não “*missing*”, de variáveis numéricas, para cada registro de dados.

<variável> = MEAN ([OF] *var1*, . . . *varn*) ;

variável	Nome até 32 caracteres que armazenará o resultado.
<i>var1</i>, . . . <i>varn</i>	Nome das variáveis numéricas que se deseja efetuar uma média.
OF	Opcional; identifica um intervalo de variáveis.

Ex.42: Funções Estatísticas

Arquivo: TEMP.DAT

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
2	3	.	5	2	5	.	2	2	4	.	8	2	5	.	7	2	5	.	8	2	2	.	0
1	8	.	5	1	7	.	4	1	8	.	9	1	8	.	9	2	0	.	5	2	0	.	8
		1	0			1	2							1	5			1	8			1	2

```
LIBNAME T "C:\curso\sas" ;
DATA T.TEMP ;
  INFILE "C:\curso\sas\TEMP.DAT";
  INPUT (T1-T6) (4.) ;
  TEMPSOMA = SUM( OF T1-T6 ) ;
  TEMPMED = MEAN( OF T1-T6 ) ;
  /* ALTERNATIVA */
  TEMPSOMA2=T1+T2+T3+T4+T5+T6;
  TMPMED2=(T1+T2+T3+T4+T5+T6)/6;
RUN;

PROC PRINT DATA=T.TEMP DOUBLE NOOBS ;
RUN ;
```

```

                                The SAS System                15:16 Monday, February 9, 2015 52

T1      T2      T3      T4      T5      T6      TEMPSOMA  TEMPMED  TEMPSOMA2  TMPMED2
23.5    25.2    24.8    25.7    25.8    22.0    147      24.5000  147      24.5000
18.5    17.4    18.9    18.9    20.5    20.8    115      19.1667  115      19.1667
10.0    12.0      .      15.0    18.0    12.0    67      13.4000  .      .
```

15.5 - Funções de Tratamento de Data e Hora

15.5.1 - Função MDY

- Função que gera uma data no formato SAS, a partir de valores de dia, mês e ano.

<variável>=MDY(mm, dd, yyyy);

variável Nome até 32 caracteres que armazenará o resultado;

mm Valor numérico que representa o mês;

dd Valor numérico que representa o dia;

yyyy ou yy Valor numérico que representa o ano.

15.5.2 - Funções DAY, MONTH, YEAR

- Funções que extraem de uma data SAS, o dia, o mês e o ano respectivamente.

<variável>=DAY(data sas);

<variável>=MONTH(data sas);

<variável>=YEAR(data sas);

variável Nome até 32 caracteres que armazenará o resultado;

data Valor que representa uma data SAS.

15.5.3 - Funções TODAY ou DATE

- Funções que retornam a data atual do sistema, do ambiente de processamento. Não possui argumentos.

<variável>=TODAY();

<variável>=DATE();

15.5.4 – Função HMS

- Função que gera um valor de hora no formato SAS, a partir de valores de hora, minuto e segundo.

<variável>=HMS(hh, mm, ss);

variável Nome até 32 caracteres que armazenará o resultado;

hh Valor numérico que representa o mês;

mm Valor numérico que representa o dia;

ss Valor numérico que representa o ano.

15.5.5 – Função TIME

- Função que retorna a hora atual do ambiente de processamento. Não possui argumentos.

<variável>=TIME();

15.5.6 - Funções: hour, minute e second

- Funções que extraem de um valor tempo no formato SAS, a hora, minuto e segundo respectivamente.

<variável>=hour(tempo);

<variável>=minute(tempo);

<variável>=second(tempo);

variável Nome até 32 caracteres que armazenará o resultado do processamento da função;

tempo Valor que representa o tempo no formato SAS.

15.5.7 – Função INTCK

- Função que retorna um valor inteiro que representa o número de intervalos existentes entre duas datas;

<variável>=INTCK(“int”, dini , dfim [,”met”]);

variável Nome até 32 caracteres que armazenará o resultado do processamento da função;

int Tipo de intervalo: YEAR, MONTH, DAY, WEEK, TIME, etc,...;

dini Data de inicial;

dfim Data final;

met Método de contagem: CONTINUOUS ou DISCRETE

OBS: Método é opcional. Se não for especificado o padrão será DISCRETE.

Ex.43: Tratamento de DATA

Arquivo: ANIVERSARIO.TXT

1	2	3	4	5	6	7	8	9	10
1	0		1	1		1	9	7	6
2	5		0	4		2	0	0	0
1	4		0	8		1	9	3	4
0	8		0	1		1	9	6	7
0	3		0	3		1	9	5	5

```
DATA ANIVERSARIO;
  INFILE "C:\curso\sas\aniversario.txt";
  INPUT DIA 1-2 MES 4-5 ANO 7-10;
  ANIV=MDY (MES, DIA, ANO) ;
  DIA_SEMANA_NASCIMENTO=WEEKDAY (ANIV) ;
  /* ALTERNATIVA 1 */
  IDADE_EM_ANOS1=INT ((TODAY () -ANIV) /365.25) ;
  /* ALTERNATIVA 2 */
  IDADE_EM_ANOS2=INTCK ("YEAR", ANIV, TODAY ()) ;
  /* ALTERNATIVA 3 */
  IDADE_EM_ANOS3=INTCK ("YEAR", ANIV, TODAY (), "C") ;
  DIAS_ATE_2000="01jan2000"d - ANIV;
```

RUN;

```
TITLE1 "Relatório de Idades";
```

```
PROC PRINT;RUN;
```

Relatório de Idades					16:38 Tuesday, August 22, 2017 44				
Obs	DIA	MES	ANO	ANIV	DIA_SEMANA_ NASCIMENTO	IDADE_ EM_ANOS1	IDADE_ EM_ANOS2	IDADE_ EM_ANOS3	DIAS_ ATE_2000
1	10	11	1976	6158	4	40	41	40	8452
2	25	4	2000	14725	3	17	17	17	-115
3	14	8	1934	-9271	3	83	83	83	23881
4	8	1	1967	2564	1	50	50	50	12046
5	3	3	1955	-1765	5	62	62	62	16375

data;

```
dini='22jul2017'd;
```

```
dfim='01aug2017'd;
```

```
meses1=intck('month', dini, dfim);
```

```
meses2=intck('month', dini, dfim, 'c');
```

run;

```
proc print;run;
```

Relatório de Idades					16:38 Tuesday, August 22, 2017 45				
Obs	dini	dfim	meses1	meses2					
1	21022	21032	1	0					

4º LABORATÓRIO

Utilizar as funções do SAS.

1 - A partir dos dados do arquivo gerado no 1º laboratório, monte um programa que crie um novo arquivo SAS, **temporário**, somente com as pessoas que possuem emprego e com o sobrenome “Mendes” ou “Silva”. Defina novas variáveis de acordo com as especificações abaixo:

Variável	Descrição	Observação
PNome	Primeiro nome	Parte da variável NOME depois da vírgula. Apenas com a primeira letra em maiúsculo.
SNome	Sobrenome	Parte da variável NOME antes da vírgula. Apenas com a primeira letra em maiúsculo.
CP	Número principal do CPF	Os 9 primeiros dígitos da variável CPF .
DV	Dígitos verificadores	Os dois últimos dígitos da variável CPF .
CPFFMT	CPF Formatado.	Adicione o caractere “.” a cada três dígitos e o caractere “-“ para separar o DV
SALINT	Salário arredondado	Arredondar a variável SALARIO para um valor inteiro.
BONUS	Bônus sobre o salário	3,5% do valor da variável SALARIO arredondado, com duas casas decimais.
SALNOVO	Novo salário	Aumento de 7,58% sobre o valor da variável SALARIO arredondado, com duas casas decimais e múltiplo de 5.
T_EMP	Tempo de empresa	O tempo do funcionário na empresa, em anos, a partir da variável DATA ADMISSÃO .

Gere um relatório para verificar os resultados;

Possíveis funções: FIND, SCAN, SUBSTR, INT, ROUND, YEAR, TODAY, INTCK, PROPCASE

2 - A partir dos dados do arquivo do 1º laboratório, monte um programa que crie um arquivo SAS, **temporário**, somente com as pessoas que possuem emprego. Definir as novas variáveis de acordo com as especificações abaixo:

Variável	Descrição	Observação
SALFAM	Salário família	Para cada filho, acrescentar 5% do salário, ao salário: SALARIO*(1+(FILHOS*0.05))
SALCOM	Salário comissionado	Cargo de chefia (GERENTE ou DIRETOR), acrescentar 20% do salário, ao salário: SALARIO*1.2
SALTOTAL	Salário total	Soma do SALÁRIO , SALFAM e SALCOM
SALMEDIO	Salário Médio	Média do SALÁRIO , SALFAM e SALCOM

Possíveis funções: SUM e MEAN|.

16 - COMANDOS DE CONFIGURAÇÃO

16.1 - Comando OPTIONS

- **Tipo:** Comando de declaração;

- **Uso:** Livre;

- Comando de definição que altera e configura o ambiente SAS: opções de performances, de impressão, de visualização do ambiente SAS, de comunicação, etc. A opção sublinhada é o valor padrão.

OPTIONS opt1 opt2 ... optn ;

Algumas opções:

COMPRESS=NO Opção que especifica a compactação dos dados na geração de um arquivo SAS, Data set (**YES** ou **CHAR, BINARY, NO**); **Somente arquivos muito grandes!**

OBS=MAX Opção que especifica qual será o último registro a ser processado;

FIRSTOBS=1 Opção que especifica qual será o primeiro registro a ser processado;

DATE/NODATE Opção que determina a “impressão” ou não, da data nos relatórios;

NUMBER/NONUMBER Opção que determina a “impressão” ou não, do número da página;

PAGENO= Opção que especifica a numeração inicial das páginas de “OUTPUT”;

LINESIZE= Opção que especifica o tamanho da linha de “impressão” do relatório texto (número de colunas). Valores de **65 até 256**;

PAGESIZE= Opção que especifica o tamanho da página de “impressão” do relatório texto (número de linhas). Valor padrão de **24**;

REPLACE/NOREPLACE Opção que determina a sobreposição ou não, dos dados de um arquivo;

SORTSIZE= Opção que especifica a quantidade de memória que o procedimento SORT poderá utilizar (em bytes). No windows: **1 GBytes**;

MEMSIZE= Opção que especifica a quantidade de memória que o sistema SAS poderá utilizar para qualquer processamento. No windows: **2GBytes**;

OBS: Para verificar as opções padrões, execute o proc step:

PROC OPTIONS; RUN;

Ex.44: Comando OPTIONS 1

```
LIBNAME ARQ "C:\CURSO\SAS" ;
OPTIONS COMPRESS=YES OBS=100 PS=66 LS=132 NODATE ;
DATA ARQ.OPT ;
SET ARQ.VAREJO ;
<comando>
<comando>
      . . .
      . . .
      . . .
;
RUN;
PROC PRINT ;
RUN ;
```

Será gerado um arquivo **compactado ???** , com apenas **100 registros**; Será gerado um relatório texto na janela OUTPUT com **66 linhas** e **132 colunas** por página, e **sem a impressão da data** no cabeçalho.

Ex.45: Comando OPTIONS 2

```
LIBNAME ARQ "C:\CURSO\SAS" ;
OPTIONS SORTSIZE=750M COMPRESS=BINARY PS=60 LS=132 ;
PROC SORT DATA=ARQ.CADASTRO OUT=NOVOCAD; BY FUNCAO; RUN;
PROC PRINT DATA=NOVOCAD ; BY FUNCAO; RUN;
```

O procedimento SORT poderá utilizar até **750Mbytes** de memória para processar; o arquivo ordenado será **compactado no modo binário ???** , e será gerado um relatório com **60 linhas** e **132 colunas** por página, na tela **OUTPUT** da sessão Windows.

16.2 - Comando LABEL

- **Tipo:** Comando de declaração;
- **Uso:** Data Step e Proc Step;
- Comando de definição, utilizado para atribuir rótulos descritivos as variáveis;
- Utilizado para melhorar a aparência de um relatório.

DATA Step LABEL **permanente**, será armazenado na estrutura do arquivo;

PROC Step LABEL **temporário**, somente para a execução da PROC;

```
LABEL <variável> = "rótulo" ...  
      <variável> = "rótulo" ...  
      <variável> = "rótulo" ... ;
```

variável Nome da variável na qual será associado um rótulo descritivo.

rótulo Descrição, até 256 caracteres, que será associada a variável.

Ex.46: Comando LABEL

```
DATA ROTULO ;  
  INFILE "C:\curso\sas\ROTULO.DAT";  
  INPUT NOME $6. @8 ALT 4.1 @13 PES 4.1 ;  
  
  LABEL ALT = "ALTURA EM POLEGADAS"  
         PES = "PESO EM LIBRAS" ;  
  
RUN ;  
  
OPTIONS LS=65 NODATE NONUMBER ;  
TITLE ;  
PROC PRINT NOOBS SPLIT=" " ;  
RUN ;
```

NOME	ALTURA EM POLEGADAS	PESO EM LIBRAS
PAULO	59.0	99.5
MARCIO	57.3	83.0
SONIA	51.3	50.5

16.3 - Comando FORMAT

- **Tipo:** Comando de declaração;

- **Uso:** Data Step e Proc Step;

- Comando de definição que serve para formatar as variáveis e melhorar a aparência dos dados num relatório;

DATA Step **FORMAT permanente**, será armazenado na estrutura do arquivo;

PROC Step **FORMAT temporário**, somente para a execução da PROC;

```
FORMAT <variável> <formatow.d> ...  
          <variável> <formatow.d> ...  
          <variável> <formatow.d> ... ;
```

variável Nome da variável que será formatada.

formatow.d Nome especial do formato com o tamanho do campo (**w**) e, em casos de formatação numérica, o número de casas decimais (**d**). Os formatos podem ser criados pelo próprio usuário ou da [biblioteca de formatos do SAS](#):

COMMAXw.d Formata com pontos e vírgula;

DDMMYYw. Formata um campo data com “/”;

Ex.47: Comando FORMAT

```
LIBNAME ARQ "C:\curso\sas";

OPTIONS OBS=10 LS=100 NODATE NONUMBER;

TITLE;

PROC PRINT DATA=ARQ.CADASTRO SPLIT="*" ;
  VAR NOME ANIVERSARIO SALARIO;

  LABEL NOME="Sobrenome, Nome"
         ANIVERSARIO="Data de*Aniversário"
         SALARIO="Salário*Bruto";

  FORMAT ANIVERSARIO DDMYY10.
         SALARIO COMMAX12.2 ;

RUN;
```

Obs	Sobrenome, Nome	Data de Aniversário	Salário Bruto
1	MARKO, PAULO	18/10/1980	.
2	MOUA, MARCO	03/07/1969	9.988,53
3	SANTOS, PAULO	01/11/1986	.
4	GUEDES, PAULO	11/01/1977	3.828,77
5	SONTAS, MARCO	18/07/1979	5.098,46
6	HORTA, PAULO	02/08/1976	3.101,32
7	CERTO, CARLA	12/11/1979	3.972,30
8	PISCO, ROSANE	23/02/1977	3.045,33
9	MALA, ELIANE	02/02/1979	.

16.4 - Comando ODS

- **Tipo:** Comando de declaração;

- **Uso:** Livre;

- **Output Delivery System (ODS)**, é um recurso especial para a manipulação dos resultados gerados pelos procedimentos SAS, permitindo a escolha de padrões de impressão e visualização, geração de arquivos e geração de documentos.

ODS < padrão > < file= > [opções] ;

...

ODS < padrão > CLOSE ;

Padrões mais utilizados:

[HTML](#)

Gera os resultados no padrão “html”;

[PDF](#)

Gera os resultados no padrão “Adobe”;

[PS](#)

Gera os resultados no padrão “PostScript”;

[RTF](#)

Gera os resultados no padrão “Rich Text”;

file= Determina a localização e o nome do arquivo que será gerado com o padrão.

opções:

style= Somente para o padrão HTML: **default, statdoc, minimal, education, festival, journal, ocean, normal, sasweb, science, seaside, theme, . . . , etc**

OBS: Todo relatório criado dentro de uma estrutura ODS (ODS <padrão>/ODS CLOSE), sairá no padrão especificado. Cada padrão de visualização possui suas próprias opções;

Ex.48: Comando ODS

```
ODS HTML FILE="C:\temp\teste.html" STYLE=statdoc;

OPTIONS OBS=10;
PROC PRINT DATA=ARQ.CADASTRO SPLIT="*" ;
  VAR NOME ANIVERSARIO SALARIO;
  LABEL NOME="Sobrenome, Nome*"
        ANIVERSARIO="Data de Aniversário"
        SALARIO="Salário Bruto";
  FORMAT ANIVERSARIO DDMYY10.
         SALARIO COMMAX12.2 ;

RUN;

ODS HTML CLOSE;
```

The SAS System

Obs	Sobrenome, Nome	Data de Aniversário	Salário Bruto
1	MALA, ROSANE	17/04/1986	1.662,28
2	SILVA, CECILIA	29/07/1976	3.612,03
3	PINTOTO, TANIA	15/03/1977	3.133,36
4	MARQUES, LIGIA	13/10/1983	1.584,08
5	SERPA, RENATO	14/06/1977	3.317,27
6	PISCO, PAULO	27/06/1977	3.900,55
7	MARKO, MARCO	08/02/1984	2.139,16
8	GUEDES, CARLA	11/10/1985	2.076,92
9	YATAKA, CARLA	09/11/1986	1.534,09
10	MILIA, FLAVIA	16/07/1985	1.795,60

5º LABORATÓRIO

Configurar a aparência de relatórios

1 - Acrescente ao programa criado no 1º exercício do laboratório anterior, rótulos descritivos para todas as variáveis relacionadas abaixo, e formatos adequados para os valores numéricos. Defina essas características de maneira **permanente**. Gere um relatório, sem títulos, rodapés, datas e numeração de página, no formato HTML.

Rótulos:

pnome	Primeiro Nome
snome	Sobrenome
salint	Valor do Salário (INTEIRO)
bonus	Bonus (3,5% do Salário)
salnovo	Novo Salário (Aumento de 7,58%)
t_emp	Tempo do Funcionário na Empresa (Em anos)
dv	Dígito Verificador do CPF
cpf	CPF
salario	Salário Original
aniversario	Data de Aniversário
admissao	Data de Admissão

Formatos:

aniversario, admissao	Formato de datas no padrão dd/mm/yyyy
salario, salint, bonus, salnovo	Formato numérico com a vírgula separando as decimais

2 - Acrescente ao programa criado no 2º exercício do laboratório anterior, rótulos descritivos para todas as variáveis relacionadas abaixo, e formatos adequados para os valores numéricos. Defina essas características de maneira **temporária**, apenas para o relatório do exercício. Gere um relatório, sem títulos, rodapés e datas, no formato PDF, somente com as pessoas que possuem emprego.

Rótulos:

saltotal	Salário Total
salmedio	Salário Médio
nome	Nome
salfam	Salário Família
salcom	Salário com Comissão
filhos	Número de Filhos
salario	Salário
função	Função

OBS: Quebre os rótulos na posição que achar melhor, para o relatório não ficar muito extenso na largura.

Formatos:

salario, salfam, salcom, saltotal, salmedio	Formato numérico com a vírgula separando as casas decimais.
---	---

17 - PROCEDIMENTOS BÁSICOS

17.1 - Procedimento MEANS

- Procedimento que produz relatórios com estatísticas descritivas simples, somente para variáveis numéricas. O relatório gerado possui uma formatação padrão que, a princípio, possui apenas algumas estatísticas: Frequência (N), Média (MEAN), Desvio Padrão (STD), valor máximo (MAX) e o valor mínimo (MIN).

PROC MEANS [opções] [estatísticas];

CLASS <lista de variáveis> ;

VAR <lista de variáveis> ;

BY <lista de variáveis> ;

OUTPUT OUT=<nome do arquivo> [opções]

RUN;

opções: **DATA=** Nome do arquivo SAS para ser processado;
 MAXDEC= Especifica o número de casas decimais (máximo 8);

estatísticas: **mean sum max min median n mode std** etc

CLASS Especifica uma ou mais variáveis que possibilitam o agrupamento de dados. **OBS: Não é necessário executar um PROC SORT, antes;**

VAR Especifica as variáveis **numéricas** que serão analisadas pelo procedimento;

BY Especifica uma ou mais variáveis que possibilitam o agrupamento de dados. **OBS: É necessário executar um PROC SORT, antes;**

OUTPUT Comando para definição de um arquivo com os resultados;

OUT= Nome do arquivo de saída;
SUM= Nome das variáveis que irão armazenar a soma;
MEAN= Nome das variáveis que irão armazenar a média;
etc...

Ex.49: Procedimento MEANS (Padrão)

```
LIBNAME ARQ "C:\CURSO\SAS" ;
PROC MEANS DATA=ARQ.CADASTRO ;
RUN ;
```

The SAS System 08:55 Tuesday, February 24, 2015 1
The MEANS Procedure

Variable	N	Mean	Std Dev	Minimum	Maximum
idade	550	26.3618182	7.4751560	20.0000000	78.0000000
peso	550	79.3567273	12.6820018	41.9000000	111.5000000
altura	550	1.8381273	0.1027400	1.6000000	2.0600000
aniversario	550	7357.50	2737.25	-11481.00	9846.00
filhos	550	2.5200000	1.4531418	0	5.0000000
admissao	434	15682.62	1112.21	2345.00	16068.00
salario	434	3332.21	2693.34	1500.73	25377.28

Ex.50: Procedimento MEANS (Com Opções)

```
LIBNAME ARQ "C:\CURSO\SAS" ;
OPTIONS NODATE NONUMBER;
PROC MEANS DATA=ARQ.CADASTRO MAXDEC=2 MIN MAX MEAN MEDIAN SUM N ;
VAR IDADE PESO ALTURA SALARIO ;
RUN ;
```

The SAS System
The MEANS Procedure

Variable	Minimum	Maximum	Mean	Median	Sum	N
idade	20.00	78.00	26.36	25.00	14499.00	550
peso	41.90	111.50	79.36	78.50	43646.20	550
altura	1.60	2.06	1.84	1.84	1010.97	550
salario	1500.73	25377.28	3332.21	3080.41	1446180.53	434

Ex.51: Procedimento MEANS (Comando CLASS)

```
LIBNAME ARQ "G:\KUSEL\SAS" ;
OPTIONS NODATE NONUMBER;
PROC MEANS DATA=ARQ.MEDIA MAXDEC=2 N MEAN MAX MIN;
CLASS SEXO;
VAR SALARIO;
RUN ;
```

The SAS System

The MEANS Procedure

Analysis Variable : salario

sexo	N	Obs	N	Mean	Maximum	Minimum
F	350	271	3302.66	20336.22	1500.73	
M	200	163	3381.35	25377.28	1507.53	

Ex.52: Procedimiento MEANS (Comando BY)

```
LIBNAME ARQ "G:\KUSEL\SAS" ;
OPTIONS NODATE NONUMBER;
PROC SORT DATA=ARQ.MEDIA ; BY SEXO ;
PROC MEANS DATA=ARQ.MEDIA MAXDEC=2 N MEAN MAX MIN ;
  BY SEXO ;
  VAR SALARIO ;
RUN ;
```

The SAS System

----- sexo=F -----

The MEANS Procedure
Analysis Variable : salario

N	Mean	Maximum	Minimum
271	3302.66	20336.22	1500.73

----- sexo=M -----

Analysis Variable : salario

N	Mean	Maximum	Minimum
163	3381.35	25377.28	1507.53

Ex.53: Procedimiento MEANS (Comando CLASS 2)

```
LIBNAME ARQ "C:\CURSO\SAS" ;
OPTIONS NODATE NONUMBER ;
PROC MEANS DATA=ARQ.CADASTRO MAXDEC=2 SUM MEAN ;
  CLASS EMPRESA SEXO ;
  VAR SALARIO ;
RUN ;
```

The SAS System
The MEANS Procedure

Analysis Variable : salario

empresa	sexo	N Obs	Sum	Mean
ATLAS S.A.	F	58	179480.78	3094.50
	M	39	117757.21	3019.42
MALTA LTDA	F	101	377653.28	3739.14
	M	73	293684.98	4023.08
PARIS INSTITUTO	F	112	337886.16	3016.84
	M	51	139718.12	2739.57

Ex.54: Procedimiento MEANS (Comando OUTPUT)

```
LIBNAME ARQ "C:\CURSO\SAS" ;
OPTIONS NODATE NONUMBER ;
PROC MEANS DATA=ARQ.CADASTRO MAXDEC=2 SUM MIN MEAN MAX;
  CLASS EMPRESA SEXO ;
  VAR SALARIO ALTURA;
  OUTPUT OUT=SAIDA
         SUM=SOMA1 SOMA2
         MEAN=MEDIA1 MEDIA2;
RUN ;
```

The SAS System

The MEANS Procedure

empresa	sexo	N Obs	Variable	Sum	Minimum	Mean	Maximum
ATLAS S.A.	F	58	salario	179480.78	1518.96	3094.50	17260.69
			altura	105.86	1.62	1.83	2.04
	M	39	salario	117757.21	1507.53	3019.42	17414.14
			altura	73.71	1.70	1.89	2.05
MALTA LTDA	F	101	salario	377653.28	1512.03	3739.14	20336.22
			altura	182.87	1.61	1.81	2.05
	M	73	salario	293684.98	1510.24	4023.08	25377.28
			altura	139.75	1.72	1.91	2.06
PARIS INSTITUTO	F	112	salario	337886.16	1500.73	3016.84	19303.66
			altura	199.51	1.60	1.78	2.01
	M	51	salario	139718.12	1515.13	2739.57	9548.36
			altura	96.94	1.70	1.90	2.05

```
PROC PRINT DATA=SAIDA;
RUN;
```

The SAS System

Obs	empresa	sexo	_TYPE_	_FREQ_	SOMA1	SOMA2	MEDIA1	MEDIA2
1			0	434	1446180.53	798.64	3332.21	1.84018
2		F	1	271	895020.22	488.24	3302.66	1.80162
3		M	1	163	551160.31	310.40	3381.35	1.90429
4	ATLAS S.A.		2	97	297237.99	179.57	3064.31	1.85124
5	MALTA LTDA		2	174	671338.26	322.62	3858.27	1.85414
6	PARIS INSTITUTO		2	163	477604.28	296.45	2930.09	1.81871
7	ATLAS S.A.	F	3	58	179480.78	105.86	3094.50	1.82517
8	ATLAS S.A.	M	3	39	117757.21	73.71	3019.42	1.89000
9	MALTA LTDA	F	3	101	377653.28	182.87	3739.14	1.81059
10	MALTA LTDA	M	3	73	293684.98	139.75	4023.08	1.91438
11	PARIS INSTITUTO	F	3	112	337886.16	199.51	3016.84	1.78134
12	PARIS INSTITUTO	M	3	51	139718.12	96.94	2739.57	1.90078

17.2 - Procedimento FREQ

- Procedimento que produz tabelas simples e tabelas cruzadas, da frequência de uma ou várias variáveis numéricas ou caracteres.

```
PROC FREQ [opções1] ;  
           BY <lista de variáveis> ;  
           TABLES <variáveis> / [opções2] ;
```

opções1: **DATA=** Nome de um arquivo SAS que será processado;
NLEVELS Indica o número de valores distintos das variáveis que estão sendo analisadas;

BY Especifica uma ou mais variáveis que possibilitam o agrupamento de dados. **OBS: É necessário executar um PROC SORT, antes;**

TABLES Especifica as variáveis e a regra de formação da tabela de frequência. Utiliza-se o caractere “*” para cruzar as variáveis.
Ex.: (tables a*b y*x peso*altura sexo*peso*altura).

opções2:

LIST	Os resultados sairão num formato “listado”, ao invés de uma tabela cruzada;
MISSING	Inclui os valores “missing” na tabela;
NOCUM	Não gera as colunas de frequência e percentual acumulativo;
NOPERCENT	Não gera as colunas com percentuais.
NOFREQ	Não gera a coluna de frequência.
OUT=	Gera um arquivo com os resultados.

Ex.55: Procedimento FREQ (Sem Cruzamento)

```
LIBNAME ARQ "C:\CURSO\SAS" ;
PROC FREQ DATA=ARQ.CADASTRO NLEVELS COMPRESS;
    TABLES SEXO IDADE ;
RUN ;
```

The SAS System

The FREQ Procedure

Number of Variable Levels

Variable	Levels
sexo	2
idade	31

sexo	Frequency	Percent	Cumulative Frequency	Cumulative Percent
F	350	63.64	350	63.64
M	200	36.36	550	100.00

idade	Frequency	Percent	Cumulative Frequency	Cumulative Percent
20	39	7.09	39	7.09
21	53	9.64	92	16.73
22	41	7.45	133	24.18
23	49	8.91	182	33.09
24	47	8.55	229	41.64
25	51	9.27	280	50.91
26	45	8.18	325	59.09
27	62	11.27	387	70.36
28	53	9.64	440	80.00
29	44	8.00	484	88.00
30	44	8.00	528	96.00
35	1	0.18	529	96.18
36	1	0.18	530	96.36
37	1	0.18	531	96.55
38	1	0.18	532	96.73
41	1	0.18	533	96.91
45	2	0.36	535	97.27
49	1	0.18	536	97.45
50	1	0.18	537	97.64
54	1	0.18	538	97.82
57	1	0.18	539	98.00
58	1	0.18	540	98.18
61	1	0.18	541	98.36
67	2	0.36	543	98.73
68	1	0.18	544	98.91
69	1	0.18	545	99.09
71	1	0.18	546	99.27
73	1	0.18	547	99.45
74	1	0.18	548	99.64
77	1	0.18	549	99.82
78	1	0.18	550	100.00

Ex.56: Procedimento FREQ (Com Cruzamento)

```
LIBNAME ARQ "C:\CURSO\SAS" ;
PROC FREQ DATA=ARQ.CADASTRO ;
    TABLES SEXO*EMPRESA ;
RUN ;
```

The SAS System

The FREQ Procedure

Table of sexo by empresa

sexo		empresa					
Frequency	Percent	Row Pct	Col Pct	ATLAS S. A.	MALTA LT DA	PARIS IN STITUTO	Total
F	58	101	112	271			
	13.36	23.27	25.81	62.44			
	21.40	37.27	41.33				
	59.79	58.05	68.71				
M	39	73	51	163			
	8.99	16.82	11.75	37.56			
	23.93	44.79	31.29				
	40.21	41.95	31.29				
Total	97	174	163	434			
	22.35	40.09	37.56	100.00			

Frequency Missing = 116

Ex.57: Procedimento FREQ (Cruzamento Listado)

```
LIBNAME ARQ "C:\CURSO\SAS";
PROC FREQ DATA=ARQ.CADASTRO ;
    TABLES SEXO*EMPRESA / LIST;
RUN ;
```

The SAS System

The FREQ Procedure

sexo	empresa	Frequency	Percent	Cumulative Frequency	Cumulative Percent
F	ATLAS S.A.	58	13.36	58	13.36
F	MALTA LTDA	101	23.27	159	36.64
F	PARIS INSTITUTO	112	25.81	271	62.44
M	ATLAS S.A.	39	8.99	310	71.43
M	MALTA LTDA	73	16.82	383	88.25
M	PARIS INSTITUTO	51	11.75	434	100.00

Frequency Missing = 116

Ex.58: Procedimento FREQ (Opções do comando TABLE)

```
LIBNAME ARQ "C:\CURSO\SAS" ;
OPTIONS NODATE NUMBER PAGENO=1;
PROC FREQ DATA=ARQ.CADASTRO ;
    TABLE SEXO*EMPRESA / MISSING NOCUM NOPERCENT FORMAT=comma6.2;
RUN;
```

The SAS System

1

The FREQ Procedure

Table of sexo by empresa

sexo	empresa				Total
Frequency Row Pct Col Pct		ATLAS S.	MALTA LT	PARIS IN	
		A.	DA	STITUTO	
F		58,00	101,00	112,00	350,00
		16.57	28.86	32.00	
		59.79	58.05	68.71	
M		39,00	73,00	51,00	200,00
		18.50	36.50	25.50	
		40.21	41.95	31.29	
Total		97,00	174,00	163,00	550,00

Ex.59: Procedimento FREQ (Opção OUT)

```
LIBNAME ARQ "C:\CURSO\SAS" ;
OPTIONS NODATE NUMBER PAGENO=1;
PROC FREQ DATA=ARQ.CADASTRO ;
    TABLE SEXO*EMPRESA / MISSING OUT=SAIDA OUTPCT;
RUN;
```

The SAS System

1

The FREQ Procedure

Table of sexo by empresa

sexo		empresa			Total
Frequency	Percent	ATLAS S. A.	MALTA LT DA	PARIS IN STITUTO	
Row Pct	Col Pct				
F	79	58	101	112	350
	14.36	10.55	18.36	20.36	63.64
	22.57	16.57	28.86	32.00	
	68.10	59.79	58.05	68.71	
M	37	39	73	51	200
	6.73	7.09	13.27	9.27	36.36
	18.50	19.50	36.50	25.50	
	31.90	40.21	41.95	31.29	
Total	116	97	174	163	550
	21.09	17.64	31.64	29.64	100.00

```
PROC PRINT DATA=SAIDA;RUN;
```

The SAS System

2

Obs	sexo	empresa	COUNT	PERCENT	PCT_ROW	PCT_COL
1	F		79	14.3636	22.5714	68.1034
2	F	ATLAS S.A.	58	10.5455	16.5714	59.7938
3	F	MALTA LTDA	101	18.3636	28.8571	58.0460
4	F	PARIS INSTITUTO	112	20.3636	32.0000	68.7117
5	M		37	6.7273	18.5000	31.8966
6	M	ATLAS S.A.	39	7.0909	19.5000	40.2062
7	M	MALTA LTDA	73	13.2727	36.5000	41.9540
8	M	PARIS INSTITUTO	51	9.2727	25.5000	31.2883

17.3 - Procedimento TABULATE

- O procedimento TABULATE monta relatórios com estatísticas descritivas num formato tabular.

```
PROC TABULATE [opções1] ;  
    CLASS <variáveis> ;  
    VAR <variáveis> ;  
    BY <variáveis> ;  
    TABLE <página , linha , coluna> / [opções2] ;
```

opções1	DATA=	Nome do arquivo SAS;
	FORMAT=	Formata todos os valores de todas as células da tabela;
CLASS		Especifica uma ou mais variáveis, numéricas ou caracteres , que possibilitam o agrupamento de dados, e que serão utilizadas como variáveis classificatórias no comando TABLE. OBS: Não é necessário executar um PROC SORT, antes;
VAR		Especifica somente variáveis numéricas que irão participar de uma análise estatística, e que serão utilizadas no comando TABLE;
BY		Especifica uma ou mais variáveis que possibilitam o agrupamento de dados, com quebra da tabela. OBS: É necessário executar um PROC SORT, antes;
TABLE		Especifica a regra de formação da tabela:
	página	Especifica uma variável ou expressão que irá definir a regra de formação das páginas;
	linha	Especifica uma variável ou expressão que irá definir a regra de formação das linhas da tabela;
	coluna	Especifica uma variável ou expressão que irá definir a regra de formação das colunas da tabela;
	opções2	CONDENSE Especifica que seja gerado o máximo de tabelas possíveis em cada página de relatório;
		BOX= Especifica o texto que aparecerá na caixa vazia da tabela;
OBS:	Regras de formação:	elemento*elemento cruzamento elemento elemento concatenação (elemento elemento) agrupamento

Elementos: variáveis, funções estatísticas (**mean, sum, max, min, median, n, mode, std, pctn, pctsum**, etc), **ALL** (totalizador).

Todos os elementos de uma dimensão cruzam com os elementos de outra dimensão.

ATLAS S.A.

MALTA LTDA

PARIS INSTITUTO

Dimensão PÁGINA

Dimensão COLUNA

PARIS INSTITUTO

		N	%	Média de Salário
Sexo	Função			
F	GERENTE	1	0,18	19.303,66
	PROGRAMADOR	111	20,18	2.870,11
M	ANALISTA	1	0,18	9.548,36
	PROGRAMADOR	50	9,09	2.603,40
Total		163	29,64	2.930,09

Dimensão LINHA

Ex.60: Procedimento Tabulate (Concatenação de Elementos)

```

PROC TABULATE DATA=ARQ.CADASTRO ;
  CLASS SEXO ; → Variável de agrupamento.
  VAR IDADE PESO; → Variáveis de análise (para cálculo).
  TABLE SEXO (IDADE|PESO) * MEAN / BOX="TABELA 1";
RUN ;

```

Cruzamento entre elementos das dimensões.

Estatística que será utilizada para análise dos dados, **cruzando** com as variáveis de análise.

Concatenação dos resultados da análise estatística das variáveis de análise.

The SAS System

TABELA 1	idade	peso
	Mean	Mean
sexo		
F	26.33	76.48
M	26.42	84.40

Ex.61: Procedimento Tabulate (Cruzamento de Elementos)

```

PROC TABULATE DATA=ARQ.CADASTRO FORMAT=COMMAX8.2;
  CLASS SEXO EMPRESA ;
  VAR PESO;
  TABLE SEXO, EMPRESA*PESO*MEAN;
RUN ;

```

The SAS System

	empresa		
	ATLAS S.A.	MALTA LTDA	PARIS INSTITUTO
	peso	peso	peso
	Mean	Mean	Mean
sexo			
F	77,67	77,44	75,42
M	83,87	84,24	83,77

Ex.62: Procedimento Tabulate (Três Dimensões 1)

```
PROC TABULATE DATA=ARQ.CADASTRO ;  
  CLASS SEXO EMPRESA ;  
  VAR ALTURA;  
  TABLE SEXO          /* Dimensão Página */  
    ,  
    EMPRESA ALL       /* Dimensão Linha */  
    ,  
    ALTURA * MEAN    /* Dimensão Coluna */  
  / CONDENSE BOX=_PAGE_;  
RUN ;
```

The SAS System

sexo F	altura
	Mean
empresa	
ATLAS S.A.	1.83
MALTA LTDA	1.81
PARIS INSTITUTO	1.78
All	1.80

sexo M	altura
	Mean
empresa	
ATLAS S.A.	1.89
MALTA LTDA	1.91
PARIS INSTITUTO	1.90
All	1.90

Ex.63: Procedimento Tabulate (Rotulação de Elementos: =)

```

PROC TABULATE DATA=ARQ.CADASTRO ;
  CLASS filhos e_civil sexo;
  TABLE sexo*e_civil*filhos all="Total", n pctn;
RUN ;

```

→ Permite inserir um "label" (rótulo) para o elemento.

The SAS System

			N	PctN	
sexo	e_civil	filhos			
F	1	1	31.00	5.64	
		2	72.00	13.09	
		3	75.00	13.64	
		4	38.00	6.91	
	2	1	38.00	6.91	
		4	28.00	5.09	
		5	41.00	7.45	
	3	0	27.00	4.91	
	M	1	1	22.00	4.00
			2	40.00	7.27
3			48.00	8.73	
4			11.00	2.00	
2		1	24.00	4.36	
		4	25.00	4.55	
		5	13.00	2.36	
3		0	17.00	3.09	
Total			550.00	100.00	

Ex.64: Procedimento Tabulate (Formatação de Elementos: *f=)

```

TITLE1 "Relatório com a Média de Salário";
TITLE2 "e a Distribuição Percentual dos Funcionários por Empresa";
OPTIONS LS=130;
PROC TABULATE DATA=ARQ.CADASTRO MISSING;
  CLASS SEXO EMPRESA;
  VAR SALARIO ;
  TABLE SEXO="Sexo"*EMPRESA="Empresa"
    ALL="Total"
    ,
    N
    PCTN="%"
    SALARIO=""* (MEAN="Média de Salário" MAX="Salário Máximo") *f=commax12.2;
RUN ;

```

$$PCTN = \frac{\text{numerador}}{\text{denominador}} * 100$$

Permite formatar os dados de linhas e colunas, especificamente, para um ou mais elementos.

Relatório com a Média de Salário
e a Distribuição Percentual dos Funcionários por Empresa

		N	%	Média de Salário	Salário Máximo
Sexo	Empresa				
F		79.00	14.36	.	.
	ATLAS S.A.	58.00	10.55	3.094,50	17.260,69
	MALTA LTDA	101.00	18.36	3.739,14	20.336,22
	PARIS INSTITUTO	112.00	20.36	3.016,84	19.303,66
M		37.00	6.73	.	.
	ATLAS S.A.	39.00	7.09	3.019,42	17.414,14
	MALTA LTDA	73.00	13.27	4.023,08	25.377,28
	PARIS INSTITUTO	51.00	9.27	2.739,57	9.548,36
Total		550.00	100.00	3.332,21	25.377,28

Total Geral

denominador

PERGUNTA: 1 - Que tal gerar uma tabela com um **subtotal por sexo**...???
2 - E um **percentual** em relação ao subtotal por sexo...???

RESPOSTA: 1 - Será necessário acrescentar à regra do comando TABLE mais um elemento **ALL**, mas aonde ?
2 - Será necessário alterar o **denominador** da estatística **PCTN**, mas como ?

Ex.65: Procedimento Tabulate (Elementos ALL e PCTN)

```
ods html file="c:\curso\sas\teste.html" style=minimal;

TITLE1 "Relatório com a Média de Salário";
TITLE2 "e a Distribuição Percentual dos Funcionários por Empresa";

PROC TABULATE DATA=ARQ.CADASTRO MISSING;
  CLASS SEXO EMPRESA;
  VAR SALARIO ;
  TABLE SEXO="Sexo" * (EMPRESA="Empresa" ALL="Subtotal") ALL="Total"
  ,
  N*f=6.
  PCTN="%"*f=commax6.2
  PCTN < EMPRESA ALL > ="% Sexo"*f=commax6.2
  SALARIO=""* (MEAN="Média de Salário" MAX="Salário Máximo") *f=commax12.2;

RUN ;

ods html close;
```

A Utilização correta dos parêntesis permite o cruzamento com grupos de elementos e criar o subtotal.

O denominador é alterado quando se especifica elementos da tabela, colocados entre "<" e ">" na estatística PCTN. Neste caso são, exatamente, os mesmos elementos que criaram o subtotal.

Relatório com a Média de Salário
e a Distribuição Percentual dos Funcionários por Empresa

		N	%	% Sexo	Média de Salário	Salário Máximo
Sexo	Empresa	79	14,36	22,57	.	.
F	ATLAS S.A.	58	10,55	16,57	3.094,50	17.260,69
	MALTA LTDA	101	18,36	28,86	3.739,14	20.336,22
	PARIS INSTITUTO	112	20,36	32,00	3.016,84	19.303,66
	Subtotal	350	63,64	100,00	3.302,66	20.336,22
M	Empresa	37	6,73	18,50	.	.
	ATLAS S.A.	39	7,09	19,50	3.019,42	17.414,14
	MALTA LTDA	73	13,27	36,50	4.023,08	25.377,28
	PARIS INSTITUTO	51	9,27	25,50	2.739,57	9.548,36
	Subtotal	200	36,36	100,00	3.381,35	25.377,28
Total		550	100,00	100,00	3.332,21	25.377,28

Ex.66: Procedimento Tabulate (Três Dimensões)

```

OPTIONS NONUMBER LS=110;
PROC TABULATE DATA=ARQ.CADASTRO MISSING;
  CLASS SEXO EMPRESA FUNCAO;
  VAR SALARIO ;
  TABLE EMPRESA="", SEXO="Sexo"*FUNCAO="Função" ALL="Total" ,
    N*f=6. PCTN="%*f=commmax6.2 SALARIO="Salário"*MEAN="Média de Salário"*f=commmax12.2
    / CONDENSE;
RUN ;

```

The SAS System

		N	%	Salário
				Média de Salário
Sexo	Função			
F	DESEMPREGADO	79	14,36	.
M	DESEMPREGADO	37	6,73	.
Total		116	21,09	.

ATLAS S.A.

		N	%	Salário
				Média de Salário
Sexo	Função			
F	ANALISTA	4	0,73	6.272,29
	GERENTE	1	0,18	17.260,69
	PROGRAMADOR	53	9,64	2.587,38
M	GERENTE	2	0,36	14.121,42
	PROGRAMADOR	37	6,73	2.419,31
Total		97	17,64	3.064,31

MALTA LTDA

		N	%	Salário
				Média de Salário
Sexo	Função			
F	ANALISTA	3	0,55	13.694,79
	GERENTE	2	0,36	16.706,07
	PROGRAMADOR	96	17,45	3.157,88
M	ANALISTA	1	0,18	9.988,53
	DIRETOR	1	0,18	25.377,28
	GERENTE	2	0,36	20.534,46
	PROGRAMADOR	69	12,55	3.148,55
Total		174	31,64	3.858,27

PARIS INSTITUTO

		N	%	Salário
				Média de Salário
Sexo	Função			
F	GERENTE	1	0,18	19.303,66
	PROGRAMADOR	111	20,18	2.870,11
M	ANALISTA	1	0,18	9.548,36
	PROGRAMADOR	50	9,09	2.603,40
Total		163	29,64	2.930,09

18 - FORMATAÇÃO ESPECIAL

18.1 - Procedimento FORMAT

O procedimento FORMAT permite que o usuário crie os seus próprios formatos de leitura e impressão, para variáveis numéricas e caracteres.

```
PROC FORMAT ;  
    VALUE <nome> [( opções )  
                <intervalo> = "<descrição>" ;
```

VALUE Comando que define um formato diferente para os valores de uma variável numérica ou caractere.

nome -Nome do formato com até 32 caracteres;
-Nomes de formatos que serão aplicados em **variáveis caracteres, obrigatoriamente**, deverão iniciar com \$;
-Nomes de formato não podem **terminar com números**.

intervalo -Especifica um valor, ou intervalos de valores, que uma variável poderá assumir durante a formatação.
-Valores ou intervalos de valores de variáveis caracteres deverão vir entre **aspas**.
-Parâmetros especiais são utilizados para definir limites de intervalos:

LOW Menor valor encontrado;
HIGH Maior valor encontrado;
OTHER Qualquer outro valor não identificado;
< Não inclui o valor do intervalo.

descrição -Valor, no qual a variável formatada, assumirá no momento de sua utilização;
- No máximo **32.767** caracteres, entre **aspas**;

Ex.67: Procedimento FORMAT

```
OPTIONS PS=30 NODATE NONUMBER;

PROC FORMAT ;
  VALUE $EC  "1"="SOLTEIRO"
             "2"="CASADO"
             "3"="SEPARADO"
             OTHER="???"
             ;

  VALUE ALT  LOW-<1.50="Até 1,50"
             1.50-<1.70="De 1,50 até 1,70"
             1.70-HIGH="Mais de 1,70"
             ;

RUN;

PROC FREQ DATA=ARQ.CADASTRO ;
  FORMAT E_CIVIL $EC.
         ALTURA ALT. ;
  TABLE E_CIVIL ALTURA;
RUN ;
```

The SAS System

The FREQ Procedure

e_civil	Frequency	Percent	Cumulative Frequency	Cumulative Percent
SOLTEIRO	337	61.27	337	61.27
CASADO	169	30.73	506	92.00
SEPARADO	44	8.00	550	100.00

altura	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Até 1,50	6	1.09	6	1.09
De 1,50 até 1,70	38	6.91	44	8.00
Mais de 1,70	506	92.00	550	100.00

6º LABORATÓRIO

Utilização dos procedimentos básicos.

Utilizando o arquivo SAS gerado no 1º LABORATÓRIO:

- 1 - Gere um relatório com a **média**, o valor **máximo**, o valor **mínimo** e o **desvio padrão** das variáveis **PESO** e **ALTURA**, **classificados** por **SEXO** e com apresentação dos resultados com **duas casas decimais**.

- 2 - Gere um relatório e um arquivo apenas com a média e somatório da variável **SALARIO**, **classificados** por **EMPRESA**. Imprima os resultados arquivados também.

- 3 - Gere um relatório com as tabelas de **frequência** de **SEXO**, **IDADE**, **ESTADO CIVIL**, **FILHOS**, **EMPRESA** e **FUNCAO**.

- 4 - Gere um relatório com uma tabela de **frequência** de **SEXO versus IDADE**.

- 5 - Gere um relatório com duas tabelas de **frequência** de **ESTADO CIVIL versus FILHOS** e **SEXO versus EMPRESA** na mesma PROC.

- 6 - Utilizando o procedimento TABULATE, crie uma tabela que apresente a **média** das **variáveis IDADE** e **PESO**, **classificadas** pela variável **SEXO**.
 - A variável **SEXO**, deverá formar as **linhas**;
 - As variáveis **IDADE** e **PESO**, deverão formar as **colunas**;
 - Deverá ser utilizada a **opção MEAN** para formar as **colunas**;

- 7 - Utilizando o procedimento TABULATE, crie uma tabela que apresente a **média** e o **somatório** da **variável SALARIO**, **classificados** por **SEXO** e **EMPRESA**.
 - A expressão **SEXO versus EMPRESA** deverá formar as linhas;
 - A variável **SALARIO** deverá formar as colunas;
 - Deverão ser utilizadas as **opções MEAN** e **SUM** para formar as colunas
 - Tamanho da página de 55 linhas por 80 colunas;

- 8 - Acrescente a tabela do relatório anterior:
 - Um totalizador na linha, para criar um total geral. Adicionar o elemento **ALL**;
 - Um subtotal por **EMPRESA**. Adicionar mais um elemento **ALL**;
 - Rotule todos os elementos: variáveis, estatísticas e totalizadores;
 - Formate os resultados numéricos com pontos, vírgula e duas decimais;
 - Título: “Apresentação do Salário por Sexo e Empresa”;
 - Sem a data, e a paginação iniciando do 1;
 - Gere a tabela no padrão HTML e PDF.

9 - Criar um programa que gere **formatos especiais** para os valores das variáveis: **IDADE, PESO, ALTURA, ESTADO CIVIL** e **SALARIO** . Gere vários relatórios que utilize esses formatos.

- Criar descrições para intervalos da variável **IDADE**:

low-25 = "Até 25 anos"
26-50 = "26 a 50 anos"
51-HIGH = "Mais de 50 anos"

- Criar descrições para intervalos da variável **PESO**:

50-<70 = "50 a 69"
70-<80 = "70 a 79"
80-<100 = "80 a 99"

- Criar descrições para intervalos da variável **ALTURA**:

1.50-<1.66 = "1,50 a 1,65"
1.67-<1.80 = "1,66 a 1,79"
1.80-HIGH = "Mais de 1,80"

- Criar descrições para os valores da variável **ESTADO CIVIL**:

"1" = "SOLTEIRO"
"2" = "CASADO"
"3" = "SEPARADO"

- Criar descrições para intervalos da variável **SALARIO**:

do menor até 2000, não incluso = "Iniciante"
de 2000 até 4000, não incluso = "Junior"
de 4000 até 7000, não incluso = "Senior"
de 7000 até 10000, não incluso = "Gerente"
acima de 10000, não incluso = "Diretor"
missing="Desempregado";

10 - Gere um relatório com tabelas de **freqüência** de **IDADE, ALTURA, PESO** e **ESTADO CIVIL** com os dados **formatados** .

(Utilize o procedimento **FREQ** com o comando **FORMAT**)

11 - Gere um relatório com uma tabela de **freqüência** de **IDADE versus ESTADO CIVIL**, com os dados **formatado**.

(Utilize o procedimento **FREQ** com o comando **FORMAT**)

12 - Gere um relatório com as pessoas classificadas por **SEXO, EMPRESA** e **SALARIO**, com os dados de **SALARIO** formatados.

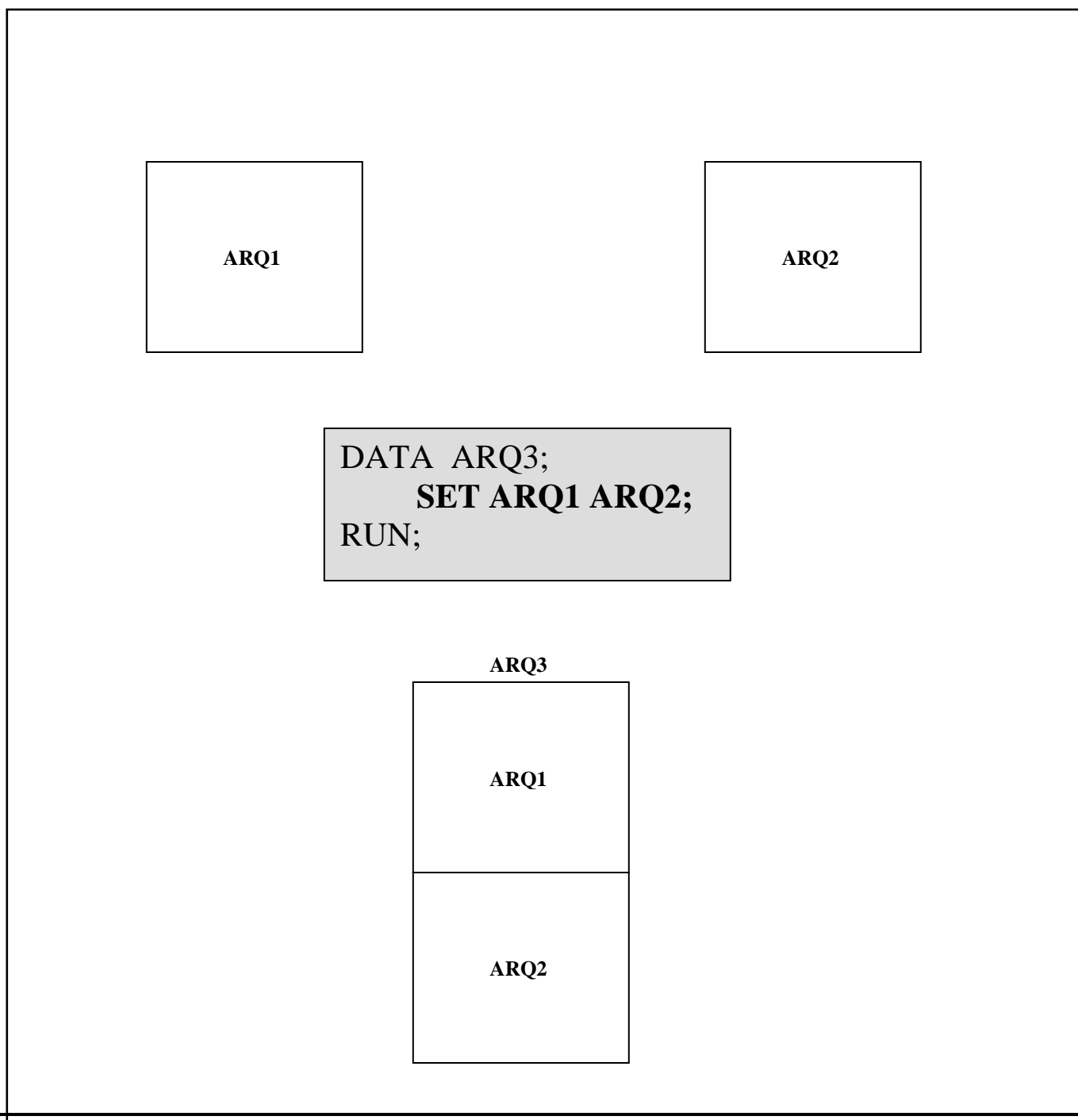
- Utilize o procedimento **TABULATE**;
- Comando **FORMAT**;

19 - COMBINAÇÃO DE ARQUIVOS SAS

19.1 - Concatenação de Arquivos

- Concatenação é a união sem critérios, de dois ou mais arquivos SAS, um após o outro, originando um único arquivo. Utiliza-se o comando SET dentro da estrutura DATA step.

```
SET <arquivo1> <arquivo2> ... <arquivon> ;
```



Ex.68: Concatenação de Arquivos

```
LIBNAME IN "C:\CURSO\SAS" ;
DATA IN.ARQ1 ;
  INFILE "C:\CURSO\SAS\ARQ1.TXT" ;
  INPUT NOME $ 1-6 @8 CPF $11. ;
RUN ;
```

ARQ1.TXT

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18
M	A	R	C	I	O		8	1	8	1	8	1	8	1	8	1	0
S	A	N	D	R	A		2	3	2	3	2	3	2	3	2	3	0
P	A	U	L	A			4	5	6	3	2	1	7	8	9	5	5
C	E	L	S	O			8	5	8	5	4	3	4	3	2	1	1
P	A	M	E	L	A		1	1	2	0	0	5	8	8	0	1	5

```
DATA IN.ARQ2 ;
  INFILE "C:\CURSO\SAS\ARQ2.TXT";
  INPUT CPF $ 1-11 SALARIO COMMAX8.2 ;
RUN ;
```

ARQ2.TXT

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20
8	1	8	1	8	1	8	1	8	1	0			8	7	6	5	,	5	5
2	3	2	3	2	3	2	3	2	3	0		1	0	5	6	7	,	4	0
4	5	6	3	2	1	7	8	9	5	5			6	4	3	2	,	1	0
8	5	8	5	4	3	4	3	2	1	1		1	1	6	7	8	,	8	9
3	5	8	0	9	3	2	2	1	7	8			7	8	9	0	,	4	0

```
DATA IN.CONCAT ;
  SET IN.ARQ1 IN.ARQ2 ;
RUN;
```

```
PROC PRINT DATA=IN.CONCAT ;
RUN ;
```

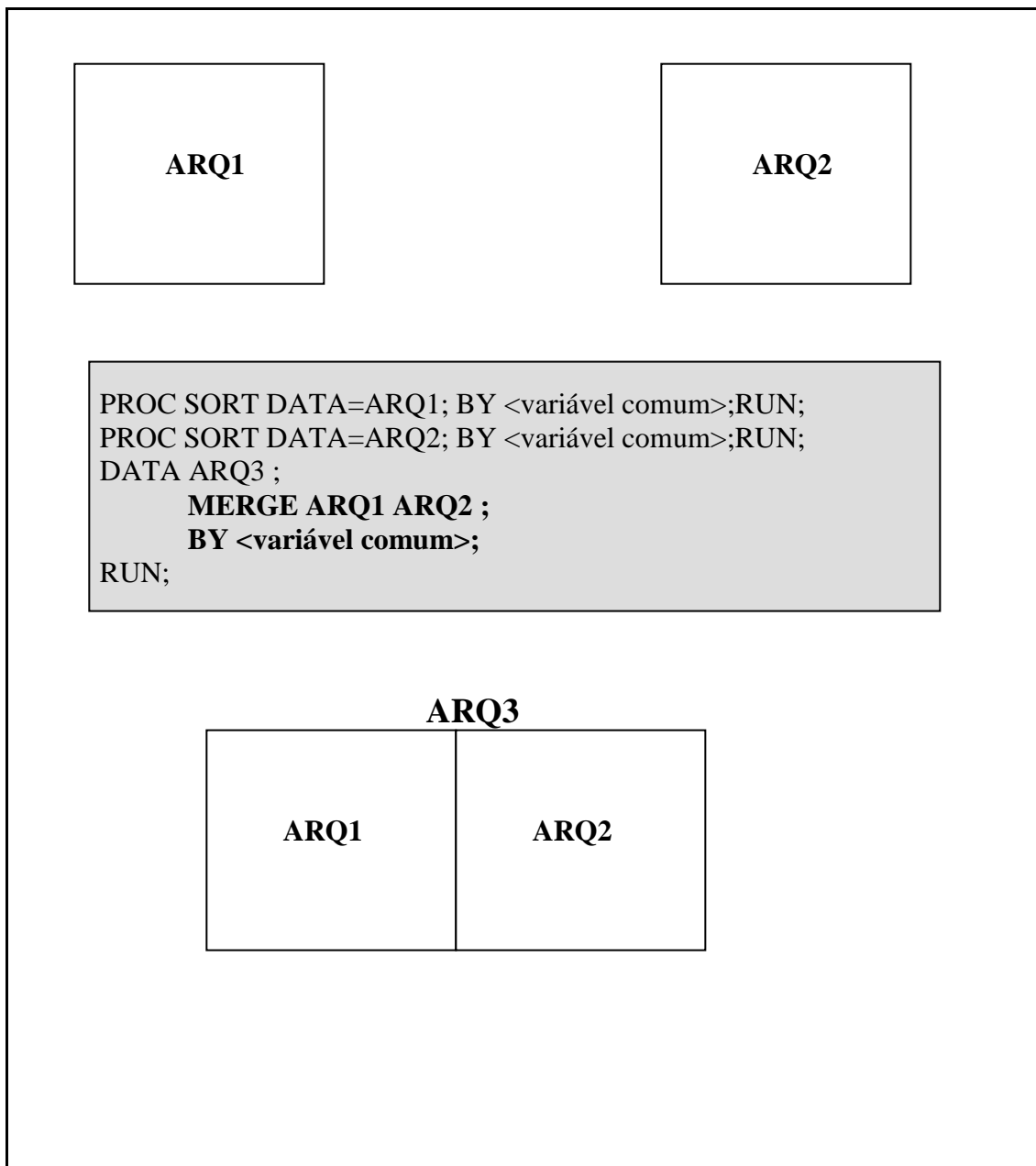
The SAS System

Obs	NOME	CPF	SALARIO
1	MARCIO	81818181810	.
2	SANDRA	23232323230	.
3	PAULA	45632178955	.
4	CELSO	85854343211	.
5	PAMELA	11200588015	.
6		81818181810	8765.5
7		35809322178	7890.4
8		23232323230	10567.4
9		45632178955	6432.1
10		85854343211	11678.8

19.2 - Combinação Ordenada de Arquivos (Match-Merging)

- Combinação ordenada é a união de dois ou mais arquivos SAS em um único arquivo, combinado e ordenado por uma ou mais variáveis comum. **Os arquivos deverão estar ordenados, antes do "Match-Merging", pelas variáveis em comum.**

```
MERGE <arquivo1> ... <arquivon> ;  
  BY <variável1 em comum> ... <variáveln em comum> ;
```



Ex.69: Combinação de Arquivos

```
LIBNAME IN "C:\CURSO\SAS" ;

PROC SORT DATA=IN.ARQ1 ; BY CPF ;
PROC SORT DATA=IN.ARQ2 ; BY CPF ;

DATA IN.ORD ;
  MERGE IN.ARQ1 IN.ARQ2 ;
  BY CPF ;

RUN;

PROC PRINT DATA=IN.ORD ; RUN ;
```

The SAS System

Obs	NOME	CPF	SALARIO
1	PAMELA	11200588015	.
2	SANDRA	23232323230	10567.4
3		35809322178	7890.4
4	PAULA	45632178955	6432.1
5	MARCIO	81818181810	8765.5
6	CELSO	85854343211	11678.8

Não combinou com
nenhum registro do
arquivo ARQ1

Não combinou com
nenhum registro do
arquivo ARQ2

Como excluir
esses registros do
resultado final ?

19.3 - Interseção de Dados, opção de arquivo (IN=)

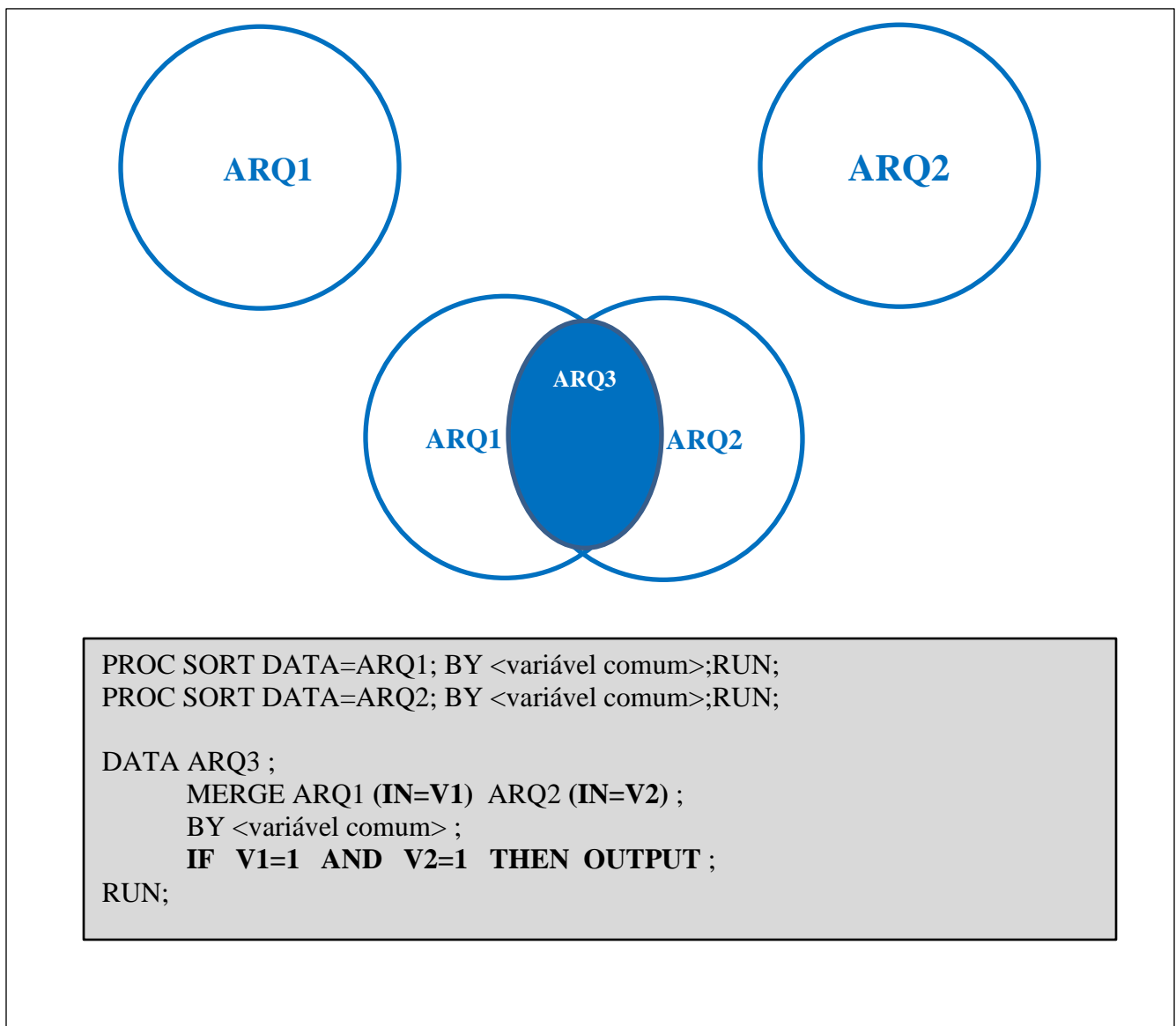
- Recurso utilizado na combinação de dois ou mais arquivos SAS, em um único arquivo, somente com os dados que combinarem de todos os arquivos, por uma ou mais variáveis comum. **Os arquivos deverão estar ordenados, antes do "Match-Merging", pelas variáveis em comum. Utiliza-se a opção IN= de Data Set.**

<arquivo sas> (IN=var) <arquivo sas> (IN=var) ...

var Variável numérica (“booleana”) que irá identificar:

- 0 O arquivo não contribuiu com dados (**falso**);
- 1 O arquivo contribuiu com dados (**verdadeiro**);

OBS: Variáveis “booleanas” **não são armazenadas** no arquivo.



Ex.70: Interseção de dados

```
LIBNAME IN "C:\CURSO\SAS" ;

PROC SORT DATA=IN.ARQ1 ; BY CPF ;
PROC SORT DATA=IN.ARQ2 ; BY CPF ;

DATA IN.ORD ;
  MERGE IN.ARQ1 (IN=V1) IN.ARQ2 (IN=V2);
  BY CPF ;
  IF V1=1 AND V2=1 THEN OUTPUT;
RUN;

PROC PRINT DATA=IN.ORD ; RUN ;
```

The SAS System

Obs	NOME	CPF	SALARIO
1	SANDRA	23232323230	10567.4
2	PAULA	45632178955	6432.1
3	MARCIO	81818181810	8765.5
4	CELSO	85854343211	11678.8

7º LABORATÓRIO

Utilização da combinação de arquivos SAS

Utilizando os arquivos SAS gerados no 3º LABORATÓRIO item 6, PESSOAL e TRABALHO:

1 - Monte um programa que gere um arquivo SAS (**CONCAT**) que será a concatenação dos dados dos arquivos: **PESSOAL** e **TRABALHO**. Gere um relatório com os dados do arquivo **CONCAT**, para visualizar o resultado.

(Utilize os comandos **DATA** e **SET**)

2 - Monte um programa que gere um arquivo SAS (**INTER**) que será a intercalação ordenada dos dados dos arquivos: **PESSOAL** e **TRABALHO** por uma variável comum **CPF**. Gere um relatório com os dados do arquivo **INTER**, para visualizar o resultado.

(Utilize os comandos **DATA**, **SET** e **BY**)

OBS: Lembre-se os arquivos devem estar ordenados pela variável comum.

3 - Monte um programa que gere um arquivo SAS (**JUSTA**) que será a combinação ordenada dos dados dos arquivos: **PESSOAL** e **TRABALHO** por uma variável comum **CPF**. Gere um relatório com os dados do arquivo **JUSTA** para visualizar os resultados.

(Utilize os comandos **DATA**, **MERGE** e **BY**)

OBS: Lembre-se, os arquivos devem estar ordenados pela variável comum.

4 – Monte um programa que crie um novo arquivo SAS que será o resultado da combinação do arquivo SAS, VAREJO, criado no 1º laboratório, com o arquivo SAS, INSTITUTO, criado no 3º laboratório, pela variável comum CPF e sem os desempregados. O arquivo final da combinação deverá possuir apenas os dados que combinam mutuamente. Gere um relatório.

(Utilize os comandos **DATA**, **MERGE** com a opção **IN=** e **BY**)

OBS: Lembre-se, os arquivos devem estar ordenados pela variável comum.

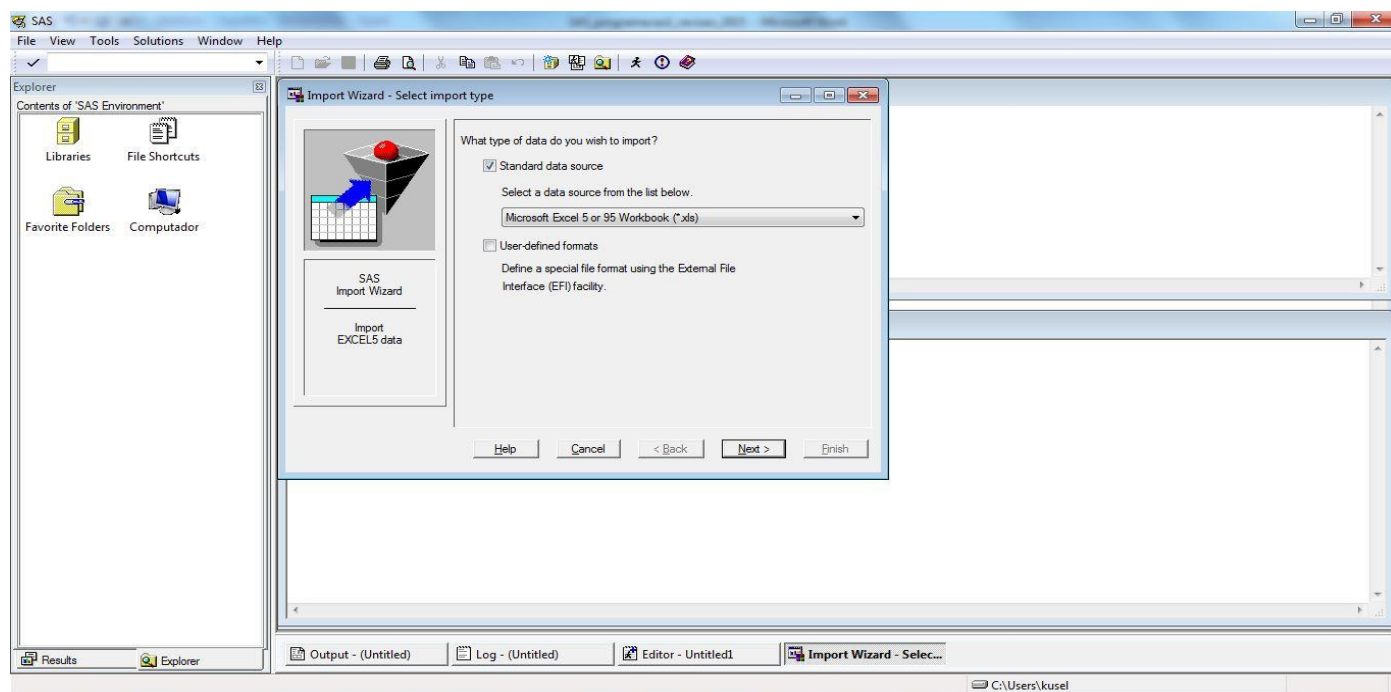
20 – Dados Microsoft EXCEL

O SAS possui meios para importar e exportar dados de/para o EXCEL da Microsoft. É necessário o licenciamento do produto “SAS/Access Interface to PC Files” para permitir ativar esse recurso.

20.1 - Modo Assistente: “Import Wizard” ou “Export Wizard”

A partir do menu principal do SAS:

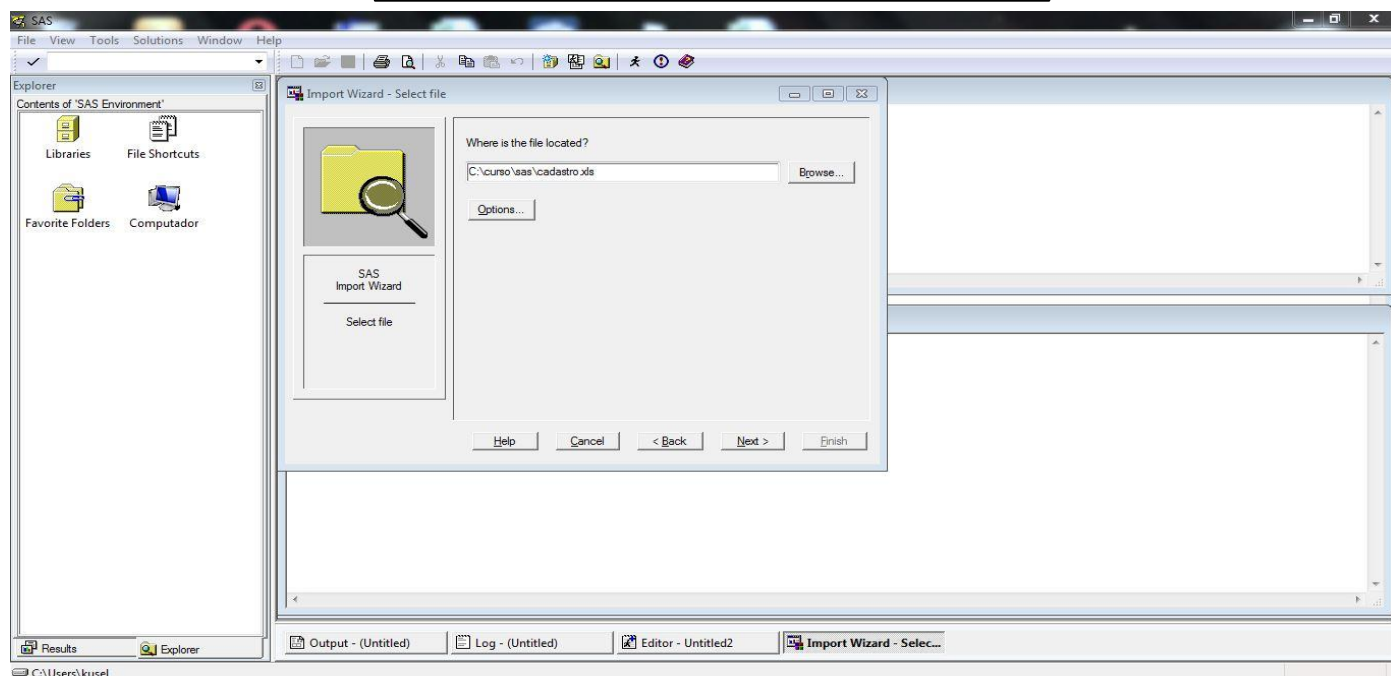
File → Import Data ... ou Export Data ...



Uma boa e genérica opção:

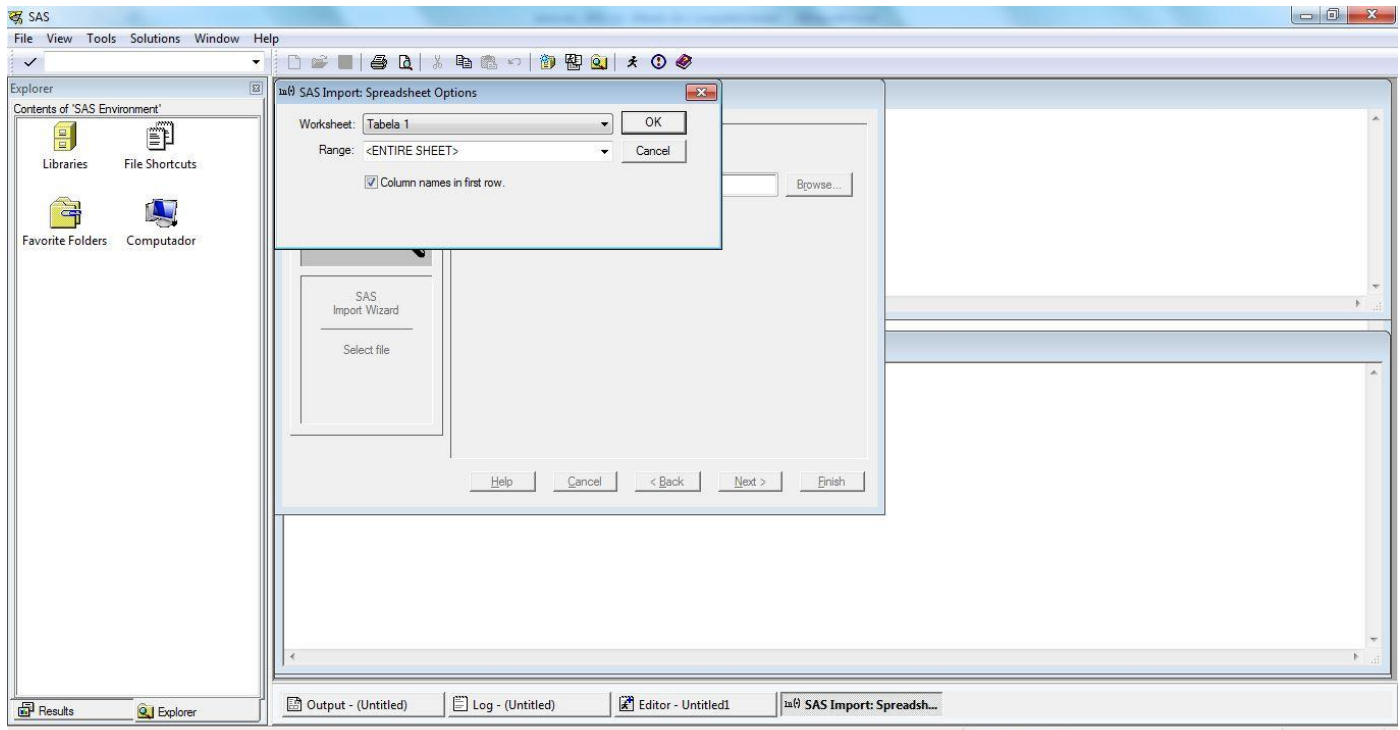
“Microsoft Excel 5 or 95 Workbook(*.xls)”

Next >



Localize e selecione o arquivo com extensão “.xls” →

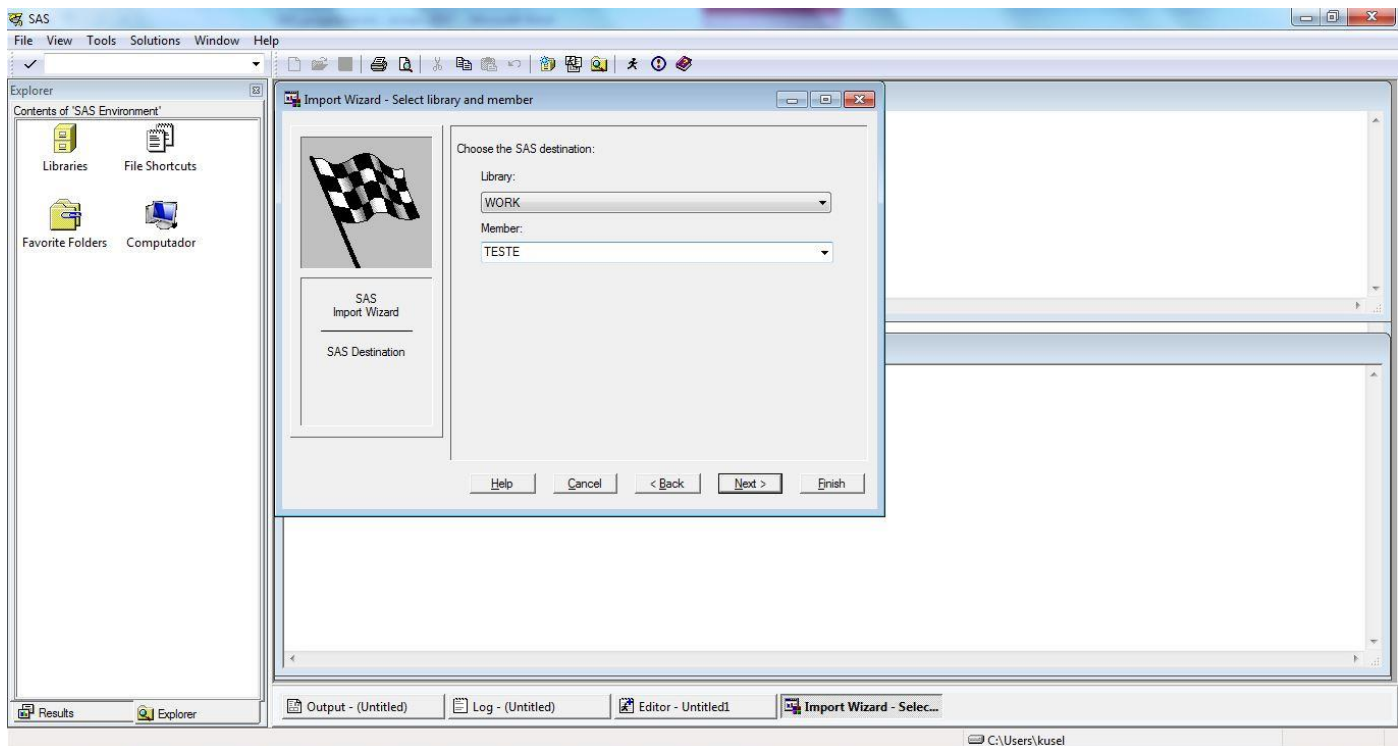
Options...



O preenchimento é automático nessa janela. Se for necessário, e existir mais de uma planilha no arquivo “.xls”, especificar a planilha e o intervalo de colunas para a importação. **Importante!** Assinalar que a primeira linha da planilha possui o nome dos campos.

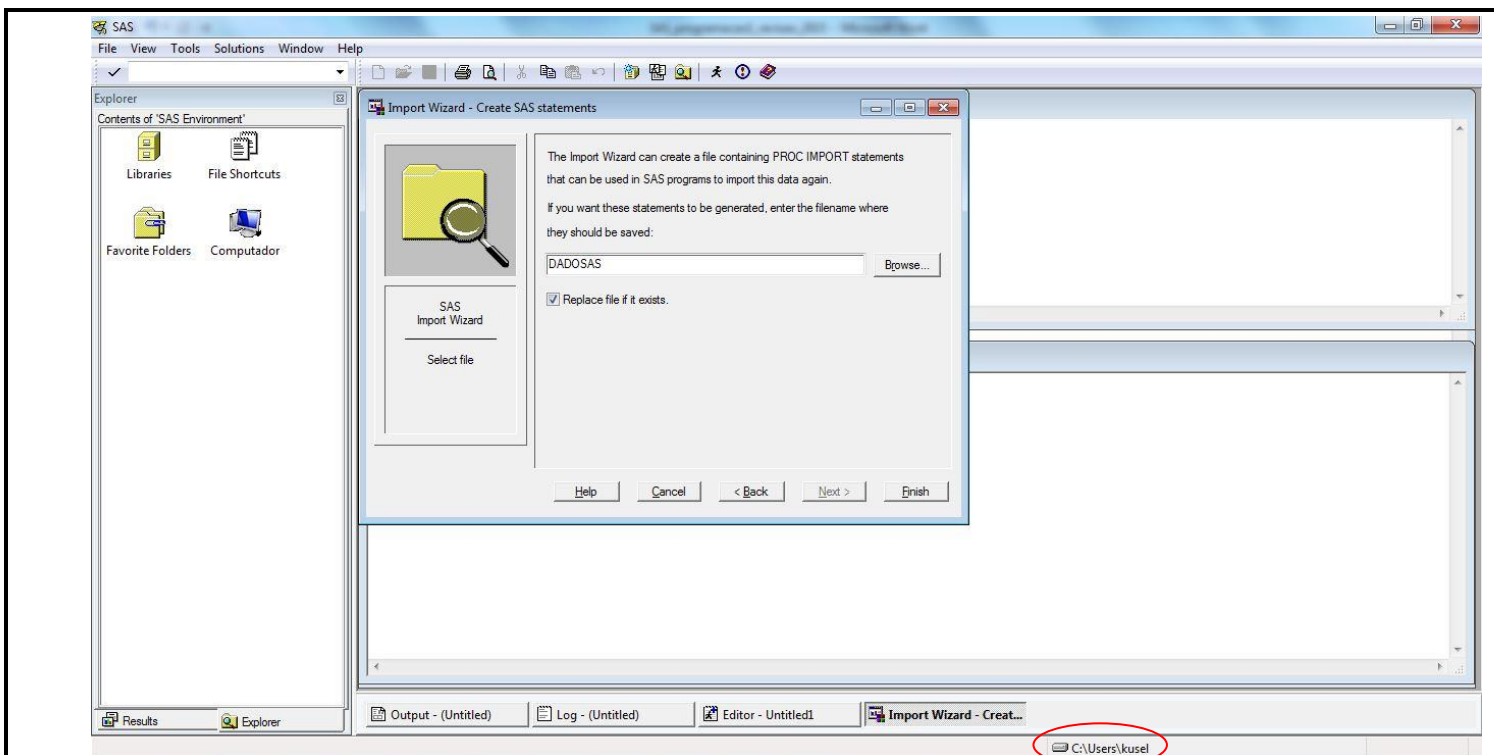
OK

Next >



Informar o nome do arquivo SAS que irá receber os dados importados do EXCEL; nome da biblioteca, nome do arquivo.

Next >



O modo assistente pode criar um programa SAS, que é exatamente a utilização do procedimento IMPORT. Muito útil para futuras execuções, bastando alterar o que for necessário.

Informe um nome qualquer para armazenar o programa e assinale, se desejar, para sobrepor o arquivo programa, caso já exista. O programa será salvo na pasta do usuário do Windows, indicado no rodapé da janela SAS. Se desejar outro local, basta indicá-lo junto com o nome do programa.

Finish

Verifique, na janela “log”, se a importação ocorreu com sucesso.

NOTE: WORK.TESTE data set was successfully created.

NOTE: The data set WORK.TESTE has 43 observations and 18 variables.

20.2 - Modo Programação: PROC IMPORT ou PROC EXPORT

Ex.71: Procedimento IMPORT

```
PROC IMPORT OUT= WORK.TESTE
            DATAFILE= "C:\curso\sas\cadastro.xls"
            DBMS=EXCEL5 REPLACE;
            SHEET="Sheet1";
            RANGE="<ENTIRE SHEET>";
            GETNAMES=YES;
RUN;
```

NOTE: WORK.TESTE data set was successfully created.
NOTE: The data set WORK.TESTE has 43 observations and 18 variables.

Ex.72: Procedimento EXPORT

```
LIBNAME ARQ "C:\curso\sas";
PROC EXPORT DATA= ARQ.CADASTRO
            OUTFILE= "C:\curso\sas\cad.xls"
            DBMS=EXCEL5 REPLACE;
RUN;
```

NOTE: Load completed. Examine statistics below.
NOTE: Inserted (550) obs into cad.xls.
NOTE: Rejected (0) insert attempts see the log for details.

NOTE: "C:\curso\sas\cad.xls" file was successfully created.

20.3 - Principais problemas

1 - Incompatibilidade de sistema de representação de dados (32bits ou 64bits)

Sistema Operacional	Software SAS	Software EXCEL	
32bits	32bits	32bits	OK
64bits	32bits	32bits	OK
64bits	64bits	32bits	ERRO
64bits	32bits	64bits	ERRO
64bits	64bits	64bits	OK

NOTE: WORK.TESTE data set was successfully created.

NOTE: The data set WORK.TESTE has 43 observations and 18 variables.

ERROR: Connect: Classe não registrada

ERROR: Error in the LIBNAME statement.

NOTE: Import Cancelled.

2 - Versões do Microsoft Excel.

O SAS importa/exporta alguns padrões/versões de arquivos EXCEL:

Microsoft Excel Workbook (*.xls *.xlsb *.xlsm *.xlsx)

Microsoft Excel Workbook on PC Files Server

Microsoft Excel 5 or 95 Workbook (*.xls)

Microsoft Excel 4 Spreadsheet (*.xls)

Atenção às mensagens de erro na janela de “log”, que podem indicar o problema com o padrão/versão escolhido e a versão original do arquivo EXCEL que está sendo importado/exportado.

ERROR: Database error. Examine fields and/or messages below.

The file specified is not an Excel 4, 5, or 7 (Excel 95) file.

ERROR: Database table must be entered first

ERROR: Database table must be entered first

ERROR: File WORK._IMEX_.DATA does not exist.

WARNING: The data set WORK.TTT may be incomplete. When this step was stopped there were 0 observations and 0 variables.

ERROR: Import unsuccessful. See SAS Log for details.

Para corrigir o problema, pode-se modificar o padrão/versão escolhido para importação/exportação ou, acessar o programa EXCEL, carregar a planilha e salvá-la com o padrão/versão correto para o SAS.

21 - REFERÊNCIAS

SAS® 9.4 Global Statements: Reference

SAS Institute Inc., 2021
Cary, NC, USA

Base SAS® 9.4 Procedures Guide, Seventh Edition

SAS Institute Inc., 2022
Cary, NC, USA

SAS Guide to TABULATE Processing, Second Edition

SAS Institute Inc., 1990
Cary, NC, USA
ISBN 1-55544-416-4

Step-by-Step Programming with Base SAS® Software

SAS Institute Inc., 2001
Cary, NC, USA
ISBN 978-1-58025-791-6

Curso de SAS/BÁSICO

Soft Consultoria