

# PUC

---

Série: Monografias em Ciências da Computação

Nº 13/78

NORMAS DE PROGRAMAÇÃO E DOCUMENTAÇÃO:

VERSÃO 1

por

P.L. STEINBRUCH

RUBENS N. MELO

Departamento de Informática

---

Pontifícia Universidade Católica do Rio de Janeiro  
Rua Marquês de São Vicente, 225 – CEP-22453  
Rio de Janeiro – Brasil

Séries: Monografias em Ciência da Computação

Nº 13/78

Editor da Série: Michael F. Challis

Junho, 1978

M3891 m2

SETOR DE DOCUMENTAÇÃO E INFORMAÇÃO	
CÓDIGO / REGISTRO	DATA
4435	15/08/79
DEPT.º DE INFORMÁTICA	

NORMAS DE PROGRAMAÇÃO E DOCUMENTAÇÃO:

versão 1\*

por

P.L. STEINBRUCH  
RUBENS N. MELO

---

\* Este relatório faz parte do projeto "Ferramentas para o projeto e construção de sistemas de informação apoiados em bancos de dados", coordenado pelo Prof. Rubens N. Melo e parcialmente financiado pelo contrato TC 2222.0060/77 do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

## RESUMO

O presente trabalho tem por finalidade estabelecer normas para as atividades, de programação e documentação, dos membros do projeto "Ferramentas para o projeto e construção de sistemas de informação apoiados em bancos de dados".

A linguagem mais utilizada para o desenvolvimento de programas, neste projeto, é o FORTRAN IV e por esta razão muitas das normas presentes neste manual são baseadas neste fato.

No entanto, no caso de utilizar-se uma outra linguagem de programação, este manual pode ser utilizado fazendo-se pequenas alterações nas normas que estão diretamente vinculadas à linguagem.

Inicialmente são apresentados alguns conceitos pertinentes a programação estruturada e a modular.

A seguir são apresentadas normas para a documentação de atividades e de programas.

No apêndice encontra-se o "lay-out" dos formulários mais utilizados.

## PALAVRAS-CHAVE

DOCUMENTAÇÃO, FORMULÁRIOS, "TOP-DOWN", "BOTTOM-UP", MANUAIS.

## ABSTRACT

This work establishes guides to be used in the programming and documentation activities of the project: "Tools for the Design and Implementation of Information Systems Supported by Data Bases".

The programming language mainly utilized in this project is FORTRAN therefore most guides in this manual are Fortran-oriented. Nevertheless it is claimed that the manual may be useful with minor adaptations even if some other language is used.

First some concepts regarding structured programming and modular programming are introduced. Then guides for the documentation of activities and programs are presented. Finally in the appendices the lay out of the most used forms are showed.

## KEY-WORDS

Documentation, FORMS, TOP-DOWN, BOTTOM-UP, MANUALS.

## S U M Á R I O

1. NORMAS DE PROGRAMAÇÃO	
1.1. Introdução .....	1-1
1.2. Estruturas de Controle .....	1-1
1.3. Composição de Instruções .....	1-2
1.4. Desenvolvimento de Programas .....	1-3
1.5. Implementação "Top-Down" .....	1-4
1.6. Implementação "Bottom-Up" .....	1-5
1.7. Considerações Complementares .....	1-5
2. NORMAS DE DOCUMENTAÇÃO	
2.1. Documentação de Sistemas .....	2-1
2.2. Documentação de Atividades .....	2-1
2.2.1. Relatório Semanal .....	2-1
2.2.2. Relatório Quinzenal .....	2-2
2.2.3. Relatório Mensal .....	2-3
2.2.4. Relatório Trimestral .....	2-5
2.2.5. Relatório Anual .....	2-6
2.3. Documentação de Programas .....	2-8
2.3.1. Documentação Interna .....	2-8
2.3.1.1. Identificação do Programa .....	2-8
2.3.1.2. Identificação de Instruções .....	2-10
2.3.1.3. Identificação de Variáveis .....	2-11
2.3.1.4. Comentários .....	2-12
2.3.2. Documentação Externa .....	2-13
2.3.2.1. Manual de Especificação .....	2-13
2.3.2.2. Manual de Implementação .....	2-17
2.3.2.3. Manual de Utilização .....	2-19
APÊNDICE - FORMULÁRIOS	
1. Relatório Semanal (FORM# 1) .....	A-1
2. Relatório Quinzenal (FORM# 2) .....	A-2
3. Relatório Mensal (FORM# 3) .....	A-3
4. Relatório Trimestral (FORM# 4/1 e FORM# 4/2) .....	A-4
5. Relatório Anual (FORM# 5/1 e FORM# 5/2) .....	A-6

## 1. NORMAS DE PROGRAMAÇÃO

### 1.1. Introdução

Com a finalidade de padronizar não só a documentação dos programas, mas também a maneira de escrevê-los, apresentamos a seguir algumas metodologias e técnicas que deverão ser utilizadas pelas equipes de programação.

O objetivo final é obter programas modulares, cujos módulos foram codificados de uma forma estruturada. Para tanto, nos itens que se seguem veremos alguns conceitos básicos de programação estruturada e programação modular.

### 1.2. Estruturas de Controle

Em uma máquina com um só processador, ao executar-se um programa, executa-se a primeira instrução, depois a segunda e assim por diante.

Qualquer "instrução" que permita que se executem instruções de uma forma não-sequencial, chamaremos de estrutura de controle.

As estruturas de controle podem ser divididas em:

- i) desvio incondicional
- ii) desvios condicionais
- iii) repetições

Um desvio incondicional é algo do tipo GO TO label.

Um desvio condicional pode ser um CASE, um IF-THEN-ELSE, um GO TO computado ou controlado, etc.

Uma repetição é um FOR, um WHILE, um REPEAT ou um PERFORM.

Preferimos entender a instrução do tipo CALL como uma instrução especial que poderia ser substituída por um conjunto de instruções e não como

sendo uma estrutura de controle.

### 1.3. Composição de Instruções

As instruções em um programa podem ser compostas (arranjadas) de quatro maneiras:

- por sequenciação
- por seleção
- por iteração
- por encadeamento

Sequenciação: Dizemos que duas (ou mais) instruções foram compostas por sequenciação se:

- i) uma instrução foi escrita (codificada) imediatamente após a outra, e,
- ii) a execução de uma instrução ocorre imediatamente após a execução da outra.

É interessante notar que para compor instruções por sequenciação não se necessita de nenhuma estrutura de controle.

Seleção: Dizemos que duas instruções foram compostas por seleção se a execução de qualquer uma delas depende do teste de uma mesma condição. Neste caso a estrutura de controle utilizada é uma estrutura do tipo desvio condicional.

Iteração: Dizemos que duas (ou mais) instruções foram compostas por iteração se estas instruções estão sujeitas a uma estrutura de controle do tipo repetição.

Encadeamento: Duas instruções estão compostas por encadeamento quando:

- i) uma instrução não foi escrita (codificada) imediatamente após a outra, e,
- ii) a execução de uma ocorre imediatamente após a execução da outra, e,
- iii) o fato da execução de uma ocorrer imediatamente após a execução da outra se deve a um desvio incondicional que foi composto por sequenciação com a primeira.

#### 1.4. Desenvolvimento de Programas

Por razões que já foram exaustivamente discutidas na literatura, desejamos que os programas produzidos, pelas equipes de programação, sejam modulares.

Não iremos descer a muitos detalhes neste ítem e para maiores esclarecimentos o leitor é convidado a consultar a referência [12].

Do ponto de vista operacional, diremos, resumidamente, que um módulo é um programa, ou, um subprograma, ou ainda uma combinação destes. Por subprograma entendemos uma subrotina ou uma função subprograma.

Então, o que pretendemos em última instância, é a obtenção de programas que sejam por sua vez combinação de outros programas ou subprogramas.

Graficamente podemos representar esta situação por uma estrutura hierarquica do tipo árvore, onde a raiz seria o módulo maior que chama todos os outros, isto é, os de nível mais baixo. As folhas seriam os módulos mais elementares e suas funções são normalmente as mais primitivas (em relação ao programa como um todo) possíveis.

Para reduzir consideravelmente a complexidade de um módulo, não será permitida a construção de um módulo que possua mais do que cinco (5) estruturas de controle do tipo desvio condicional.



Recomenda-se que os desvios incondicionais não sejam utilizados.

Uma maneira prática de se obter programas modulares com código estruturado é a seguinte:

- i) divide-se o problema em sub-problemas que por sua vez poderão ser subdivididos.
- ii) uma vez tendo sido atingido um nível adequado de detalhe (subproblemas que quando transformados em programas (subprogramas) não contenham mais do que 5 estruturas de controle do tipo desvio condicional) passa-se à codificação utilizando-se para compor instruções, seleção, seleção e repetição.

A medida que os módulos vão sendo obtidos, eles poderão ir sendo testados segundo os critérios que apresentaremos nos dois itens a seguir.

#### 1.5. Implementação "TOP-DOWN"

Os módulos dos níveis mais altos vão sendo codificados, i.e., todos os módulos de nível mais alto devem estar codificados para que se comece a codificação de um módulo subordinado a qualquer um dos anteriores. Quando todos os módulos de um mesmo nível estiverem codificados, é feito um teste. Quando se testa um conjunto de módulos, usa-se na prática considerar os módulos subordinados como DUMMY, i.e., módulos que posteriormente serão codificados e testados.

As vantagens do enfoque Top-Down são a resolução a priori das interfaces e facilidade no teste dos módulos.

Seja B um módulo subordinado ao módulo A. Usando este enfoque, o módulo A vai ser codificado e testado primeiro que B. Desta forma quando B for codificado, os parâmetros que A deve passar para B já estarão bem definidos e no teste do módulo B já poderemos chamá-lo através do módulo A.

## 1.6. Implementação "BOTTOM-UP"

No caso "Bottom-Up" a implementação é feita dos módulos de nível mais baixo para os de nível mais alto.

Assim que um módulo é codificado, para testá-lo o programador deve criar um módulo DRIVER que forneça os parâmetros necessários para a execução do módulo a ser testado. Por exemplo, seja o módulo B um submódulo de A. Assim que B é codificado, devemos testá-lo. Devido ao fato de que o módulo A não foi codificado ainda, a execução de B só é possível com a criação de um módulo DRIVER que simula a chamada de B pelo módulo A. É importante observar que no momento que o módulo A é codificado e testado, o módulo DRIVER, que substitui A, se torna desnecessário e é eliminado.

Normalmente a abordagem "Bottom-Up" é utilizada quando os módulos de nível inferior são mais simples de serem codificados do que os de nível superior ou quando alguns dos módulos de nível inferior já existem.

## 1.7. Considerações Complementares

### 1.7.1. FORTRAN Estruturado

Está a disposição das equipes de programação um preprocessador de FORTRAN<sup>4</sup> que aceita as seguintes estruturas de controle:

- desvios condicionais: IF-THEN-ELSE, CASE
- repetições: FOR, WHILE, REPEAT e PERFORM

O preprocessador permite, ainda, a definição de macros.

Os interessados em utilizar estas ferramentas deverão consultar a referência [14] para se inteirarem dos detalhes sintáticos da mesma.

## 2. NORMAS DE DOCUMENTAÇÃO

### 2.1. Documentação de Sistemas

As normas relativas à documentação de sistemas estão sendo elaboradas e serão acrescentadas em uma próxima versão deste manual.

### 2.2. Documentação de Atividades

As atividades dos participantes deste projeto serão documentadas utilizando-se cinco formulários básicos, a saber:

- i) Relatório de Atividades Semanais (FORM# 1)
- ii) Relatório de Atividades Quinzenais (FORM# 2)
- iii) Relatório de Atividades Mensais (FORM# 3)
- iv) Relatório de Atividades Trimestrais (FORM# 4/1 e FORM# 4/2)
- v) Relatório de Atividades Anuais (FORM# 5/1 e FORM# 5/2).

A seguir será descrito como deve ser preenchido cada\* um destes formulários.

#### 2.2.1. Relatório de Atividades Semanais (FORM# 1)

O relatório de atividades semanais é identificado pela referência: FORM# 1, que se encontra na sua extremidade direita.

O cabeçalho deve ser preenchido observando-se o seguinte:

- i) O número do relatório deverá ser deixado em branco.
- ii) O campo reservado ao nome do relator, e a data devem ser preenchidos quando do início da confecção do relatório.

---

\* Muitas vezes ocorrerá de repetir-se na descrição de preenchimento de um formulário, trechos idênticos a outros vistos anteriormente na descrição de preenchimento de um outro formulário.

Isto se deve ao fato de querer-se ter a descrição completa de uso de um formulário sem que o leitor seja obrigado a procurar referências feitas a outros trechos do manual.

- iii) Se uma folha não for suficiente para conter o relatório, para continuação deve ser usado o formulário FORM# 5/2.
- iv) O campo "FOLHA DE" deverá ser preenchido após a confecção do relatório e para tanto procede-se da seguinte forma:
  - i) Conta-se quantas folhas compõe o relatório;
  - ii) Supondo que o relatório possua n folhas, na primeira teremos: FOLHA 1 DE n.  
Na segunda, FOLHA 2 DE n; na enésima, FOLHA n DE n.

O campo reservado ao relatório propriamente dito deve ser preenchido, utilizando-se redação em forma livre, relatando-se o que foi feito na última semana.

#### 2.2.2. Relatório de Atividades Quinzenais (FORM# 2)

O relatório de atividades quinzenais é identificado pela referência: FORM# 2, que se encontra na sua extremidade direita.

A finalidade maior deste relatório é apresentar um "retrato" do que foi feito nas duas últimas semanas e o que se pretende fazer nas próximas duas.

Este relatório é, por natureza, de caráter sumário.

Só os objetivos mais gerais devem ser mencionados nele.

O cabeçalho deve ser preenchido observando-se o seguinte:

- i) O número do relatório deverá ser deixado em branco.
- ii) O campo reservado ao nome do relator, e, a data devem ser preenchidos quando do início da confecção do relatório.

O campo destinado ao relatório propriamente dito é dividido em quatro partes:

- i) Principais atividades da última quinzena: como o próprio nome diz, neste campo serão listadas as principais atividades desenvolvidas na última quinzena.
- ii) Progressos em relação à penúltima quinzena: neste campo serão listados os progressos feitos em relação à penúltima quinzena, isto é, qual foi a contribuição incremental realizada na última quinzena em termos do projeto como um todo?
- iii) Objetivos alcançados em relação aos listados na penúltima quinzena: conforme o título deverão ser listados os objetivos alcançados em relação aos listados na penúltima quinzena.
- iv) Objetivos a serem atingidos na próxima quinzena: deverão ser listados os objetivos que se espera que sejam atingidos na próxima quinzena.

Ao final o relatório deverá ser assinado, pelo relator, na linha acima da palavra "ASSINATURA".

### 2.2.3. Relatório de Atividades Mensais (FORM# 3)

O relatório de atividades mensais é identificado pela referência: FORM# 3, que se encontra na sua extremidade direita.

O cabeçalho deve ser preenchido observando-se o seguinte:

- i) O número do relatório deverá ser deixado em branco.
- ii) O campo reservado ao mes deve ser preenchido com o mes sobre o qual se faz o relatório.

iii) O campo reservado ao nome do relator, e, a data devem ser preenchido quando do início da confecção do relatório.

iv) O campo "FOLHA DE" deverá ser preenchido após a confecção do relatório e para tanto procede-se da seguinte forma:

i) conta-se quantas folhas compõe o relatório.

ii) supondo que o relatório possua n folhas, na primeira teremos FOLHA 1 DE n.

na segunda, FOLHA 2 DE n; na enésima FOLHA n DE n.

O campo reservado ao relatório propriamente dito deve ser preenchido, utilizando-se redação em forma livre, relatando-se que foi feito no mes em questão e ressaltando-se:

i) atividades realizadas;

ii) objetivos atingidos;

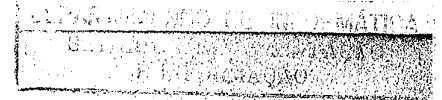
iii) objetivos previstos e não atingidos;

iv) dificuldades encontradas na consecução dos objetivos;

v) previsão para realização das atividades que concorrem para a consecução dos objetivos que não foram atingidos;

vi) previsão das implicações em função do descrito em iii e v.

O relatório mensal deve ser o mais detalhado possível e para continuação de sua folha-de-rosto (FORM# 3) deverá ser utilizado o formulário identificado pela referência FORM# 5-2.



#### 2.2.4. Relatório de Atividades Trimestrais (FORM# 4/1 e FORM# 4/2)

O relatório de atividades trimestrais é composto por dois formulários: FORM# 4/1 e FORM# 4/2.

Este relatório é, por definição, um relatório conciso cuja finalidade de maior é apresentar sucintamente os progressos feitos nos últimos três meses com o fim de apresentar uma imagem - a mais real possível - do estado atual do projeto.

Instruções para preenchimento do formulário FORM# 4/1:

O cabeçalho deve ser preenchido observando-se o seguinte:

- i) O número do relatório deverá ser deixado em branco.
- ii) O campo reservado ao nome do relator e a data devem ser preenchidos quando do início da confecção do relatório.
- iii) O campo "FOLHA DE" deverá ser preenchido após a confecção do relatório e para tanto procede-se da seguinte maneira:
  - i) Conta-se quantas folhas compõe o relatório;
  - ii) Supondo que o relatório possua n folhas, na primeira teremos FOLHA 1 DE n.  
Na segunda, FOLHA 2 DE n.  
Na enésima, FOLHA n DE n.

O campo reservado ao relatório propriamente dito deve ser preenchido da seguinte maneira:

- i) Depois das palavras MES DE deverá ser escrito o nome do mês sobre o qual irá ser feito o relatório.

- ii) Se o espaço reservado não for suficiente, procede-se como no item anterior com relação a próxima ocorrência do "MES DE" e utiliza-se o novo espaço a disposição. Este procedimento pode ser feito até que a folha termine. Para continuação usa-se o FORM# 4/2.
- iii) Ao concluir o relatório do primeiro mes do trimestre, inicia-se este mesmo procedimento para o segundo mes e depois para o terceiro. Em resumo, o relatório trimestral poderá ser somente uma folha do tipo de FORM# 4/1 ou uma folha do tipo FORM# 4/1 e tantas quantas se necessite do tipo do FORM# 4/2. Para confecção do texto propriamente dito deve ser usada redação na forma livre.

Instruções para o preenchimento do formulário FORM# 4/2:

O FORM# 4/2 é um formulário utilizado para o relatório trimestral de atividades no caso de ser necessário mais do que uma folha para o mesmo.

O seu preenchimento é equivalente ao descrito para o formulário FORM# 4/1 no que diz respeito aos itens que lhes são comuns.

#### 2.2.5. Relatório de Atividades Anuais (FORM# 5/1 e FORM# 5/2)

O relatório anual de atividades é composto por dois tipos de formulários: FORM# 5/1 e FORM# 5/2.

Este relatório é um documento que relata todas as atividades desenvolvidas por cada participante do projeto ao longo de um período de um ano. Devido a isto deve conter de forma detalhada e precisa cada atividade desenvolvida pelo participante, juntamente com todos os detalhes pertinentes a esta atividade. Não devem deixar de ser mencionadas as dificuldades encontradas no desenvolver de cada atividade, bem como as atitudes tomadas para solucionar problemas sem solução prevista.



Instruções para o preenchimento do formulário FORM# 5/1:

O cabeçalho deve ser preenchido observando-se o seguinte:

- i) O número do relatório deverá ser deixado em branco.
- ii) O campo reservado ao nome do relator, e a data devem ser preenchidos quando do início da confecção do relatório.
- iii) O campo "FOLHA DE" deverá ser preenchido após a confecção do relatório e para tanto procede-se da seguinte forma:
  - i) Conta-se quantas folhas compõe o relatório;
  - ii) Supondo que o relatório possua n folhas, na primeira teremos FOLHA 1 DE n. Na segunda, FOLHA 2 DE n; na enésima, FOLHA n DE n.

O campo reservado ao relatório propriamente dito deve ser preenchido, utilizando-se redação em forma livre.

Quando for conveniente, poderão ser usados, também, tabelas e gráficos.

Instruções para preenchimento do formulário FORM# 5/2.

O formulário FORM# 5/2 é um formulário utilizado para continuação dos relatórios semanais, mensais e anuais. Quanto ao texto a ser escrito no mesmo, deve-se observar as diretrizes utilizadas para o relatório ao qual ele está sendo usado como continuação.

## 2.3. Documentação de Programas

### 2.3.1. Documentação Interna

#### 2.3.1.1. Identificação do Programa

A identificação do programa será feita baseada nos seguintes itens:

- 2.3.1.1.1. nome do programa
- 2.3.1.1.2. código do programa
- 2.3.1.1.3. descrição do programa
- 2.3.1.1.4. sub-programas necessários
- 2.3.1.1.5. "linguagem de codificação"
- 2.3.1.1.6. "Requisitos":
  - EQUIPAMENTOS
  - SISTEMA OPERACIONAL
  - MEMÓRIA ALOCADA
  - TEMPO DE EXECUÇÃO
- 2.3.1.1.7. variáveis do programa
- 2.3.1.1.8. manuais
- 2.3.1.1.9. data da codificação
- 2.3.1.1.10. programador(es)

que serão dispostos conforme o "lay-out" apresentado na Fig. I.

#### 2.3.1.1.1. nome do programa

O nome do programa deverá ser composto por, no máximo sete caracteres, começando por uma letra, e deverá ser escolhido de tal forma que seja auto-explicativo.

Será sob este nome que o programa será catalogado na biblioteca de programas-fonte.

```

*****
* NOME: XXXXXX                                CODIGO: BDXX
*****
* DESCRIÇÃO DO PROGRAMA:
* _____
* _____
* _____
*****
* SUBPROGRAMAS:
* _____
* _____
* _____
*****
* LINGUAGEM DE CODIFICAÇÃO: NOME DA LINGUAGEM
*****
* REQUISITOS:
*   EQUIPAMENTOS:
*   _____
*   _____
*   _____
*
*   SISTEMA OPERACIONAL:   XXXXX
*   MEMORIA ALOCADA:       XXXXX
*   TEMPO DE EXECUÇÃO:     _____
*****
* VARIÁVEIS:
*   INTEIRAS
*       IXXXXX - _____
*       IXXXXX - _____
*       IXXXXX - _____
*
*   REAIS
*       RXXXXX - _____
*       RXXXXX - _____
*       RXXXXX - _____
*
*   LOGICAS
*       LXXXXX - _____
*       LXXXXX - _____
*       LXXXXX - _____
*****
* MANUAIS:
* _____
* _____
* _____
*****
* DATA DA CODIFICAÇÃO:
*       XX/MES/XX
*****
* PROGRAMADOR(ES): _____
*****

```

FIGURA 1

#### 2.3.1.1.2. Código do programa

O código do programa será composto de quatro caracteres da seguinte maneira:

- i) os dois primeiros serão sempre a letra B seguida da D, isto é, BD.
- ii) os dois seguintes serão um número que será atribuído a cada novo programa.

#### 2.3.1.1.3. Descrição do programa

A descrição do programa é um pequeno texto que explica o que o programa faz. Ela deve ser sucinta e completa.

#### 2.3.1.1.4. Sub-programas necessários

Em cada programa deverá aparecer os nomes dos sub-programas que ele necessita. Esses nomes aparecerão sob este título conforme a FIG.I.

#### 2.3.1.1.5. Linguagem de Codificação

Deverá ser especificada a linguagem de codificação utilizada bem como deverão ser feitas referências aos manuais básicos desta linguagem.

#### 2.3.1.1.6. Requisitos

O equipamento utilizado deve ser especificado em termos de fabricante, modelo e capacidade de memória.

O sistema operacional deve ser especificado pela sigla completa fornecida pelo fabricante.

Deverá ainda ser especificada a memória alocada pelo programa e o tempo de execução do mesmo.

#### 2.3.1.1.7. Variáveis do programa

Todas as variáveis que aparecem em um programa, devem ser identificadas no início do mesmo. (Conforme FIG.I).

#### 2.3.1.1.8. Manuais

Em cada programa deverá ser feita uma referência aos manuais que podem ser consultados para melhor esclarecer o que faz o programa, como ele foi implementado e como ele deve ser usado.

#### 2.3.1.1.9. Data da codificação

Em cada programa deverá constar a data de sua codificação para fins de atualização ou manutenção.

#### 2.3.1.1.10. Programador

Finalmente, cada programa deverá trazer o nome do seu autor, isto é, de quem o codificou.

#### 2.3.1.2. Identificação de Instruções

Será codificado somente um comando por linha. (cartão)

As colunas de 73-80 devem estar identificadas e enumeradas (IDEMNUME) de tal forma que se saiba a que programa pertence uma determinada instrução, e, qual sua localização na sequência de instruções.

Esta identificação será feita seguindo o seguinte critério:

- i) Nas colunas de 73-76 será codificado o código do programa\*
- ii) Nas colunas 77-80 será codificado um número que identificará a posição do comando na sequência do programa.

### 2.3.1.3. Identificação de Variáveis

As variáveis de um determinado programa, terão seus nomes formados da seguinte maneira:

- i) a primeira letra, do nome da variável, será uma letra que indique o tipo desta variável (conforme FIG.II).
- ii) os restantes cinco caracteres do nome dessa variável deverão formar um nome que seja adequado e significativo à variável.

TIPO	LETRA
REAL	R
INTEIRO	I
DUPLA PRECISÃO	D
CARACTER	S
HEXADECIMAL	Z
LÓGICO	L
COMPLEXO	C
BINÁRIO	B

FIG. II

---

\* ver Identificação do Programa

2.3.1.4. Comentários

Os comentários em um programa deverão ser feitos da seguinte maneira:

- i) primeiro deverá vir um cartão em branco.
- ii) depois deverão vir tantas linhas (cartões) de comentário quanto se necessite.
- iii) finalmente deverá vir mais um cartão em branco.

Exemplo:

```
C
C OS COMENTARIOS FEITOS EM
C UM PROGRAMA DEVERÃO SER
C PRECEDIDOS POR UM CARTÃO
C EM BRANCO E SEGUIDOS
C DE UM CARTÃO EM BRANCO
C
```

OBS.: O conjunto de comentários de um programa deverá ser suficiente para definir "precisamente" o algoritmo que se deseja implementar com esse programa.

### 2.3.2. Documentação Externa

#### 2.3.2.1. Manual de Especificação

NOME DO PROGRAMA:

CÓDIGO:

##### 1. ENTRADAS DO PROGRAMA

Todas as entradas do programa deverão ser descritas. Os formulários FORM# 6 e FORM# 7 poderão ser utilizados na medida em que ajudem a esclarecer a situação. Quando for o caso, referências a outros manuais deverão ser feitas.

##### 2. SAÍDAS DO PROGRAMA

Todas as saídas do programa deverão ser descritas. Os formulários FORM# 6, FORM# 7 e FORM# 8 poderão ser utilizados na medida em que ajudem a esclarecer a situação. Quando for o caso, referências a outros manuais deverão ser feitas.

##### 3. PRINCIPAIS FUNÇÕES DO PROGRAMA

Deve ser feita uma breve descrição das funções a serem realizados pelo programa. A seguir, deve ser feita uma descrição detalhada a respeito do interrelacionamento destas funções.

##### 4. FORMA DE COMUNICAÇÃO ENTRE O PROGRAMA E OUTROS PROGRAMAS

Neste item devem ser listados todos os programas (sub-programas) que são chamados por este programa, e, todos os programas que chamam ou interagem com este programa.

Deve também ser especificado como os programas em questão se comunicam (passagem de parâmetros, áreas globais, etc.).



## 5. REGRAS, DECISÕES LÓGICAS E ESTRUTURAS DE DADOS

Neste item é especificado de forma integral e detalhada o algoritmo que o programa deverá implementar. Devem ser especificadas as condições iniciais do problema, as condições de contorno, o domínio dos dados, as restrições do problema e a descrição detalhada das estruturas de dados a serem utilizadas em cada etapa do problema.

O desenvolvimento do algoritmo por refinamentos sucessivos é recomendado.

Se for o caso, podem ser utilizadas tabelas de decisão (FORM# 9).

## 6. CRITÉRIOS DE VALIDAÇÃO E EDIÇÃO

Deverão ser listados os critérios para validação e edição segundo o processamento.

## 7. AÇÕES A SEREM TOMADAS EM SITUAÇÕES DE ERRO

Neste item devem ser especificadas todas as ações a serem tomadas em cada situação possível de erro.

As mensagens de erro devem ser padronizadas para um mesmo programa.

Deverão ser especificados, também, os meios em que serão emitidas mensagens (impressora, console, terminal, etc.), e, que mensagens serão publicadas em que meios.

NOTA: EXISTEM PELO MENOS TRES FASES DISTINTAS NESTE ITEM:

- i) uma referência cruzada de que ações são tomadas em cada situação de erro.
- ii) a definição da padronização das mensagens.

iii) a relação das mensagens a serem impressas, separadas por dispositivo.

## 8. TABELAS E FÓRMULAS ESPECIAIS

Todas as tabelas adicionais e fórmulas especiais pertinentes ao programa ou a sua compreensão serão aqui apresentadas.

## 9. REFERÊNCIAS

Dependendo da dificuldade do problema, poderão ser úteis referências à fontes que contenham esclarecimentos a respeito de modelos utilizados ou teorias associadas ao modo como o problema foi interpretado pelo especificador.

## 10. PLANO DE TESTE DO PROGRAMA

### 10.1. Organização do teste

#### 10.1.1. Objetivos de teste do programa

Aqui deverá ser mencionado o grau de confiabilidade a ser atingido

#### 10.1.2. Alocação de responsabilidades para a condução do teste.

Deverá ser determinado quem são os responsáveis pelo projeto dos arquivos e consistência dos dados de entrada.

#### 10.1.3. Referência à confiabilidade dos sub-programas utilizados pelo programa.

### 10.2. Critérios de validade

#### 10.2.1. Controle

##### 10.2.1.1. Tolerâncias e erros permissíveis

##### 10.2.1.2. "Check-points" para recuperação de erros

##### 10.2.1.3. Requisitos de pré-processamento

##### 10.2.1.4. Critérios para aceitação de arquivos

##### 10.2.1.5. Critérios para aceitação de entradas

#### 10.2.2. Processamento

##### 10.2.2.1. Combinações não-inválidas de entradas e de informações contidas em arquivos

10.2.2.2. Códigos e combinações de códigos não-inválidos

10.2.2.3. Parâmetros não-inválidos

10.2.3. Saída

10.2.3.1. Comprimento e formato das mensagens de erro

10.2.3.2. Título e códigos das mensagens

10.3. Programa de teste

O programa de teste do programa é especificado utilizando-se o FORM# 10.

Neste item tem-se a sequência de operações necessárias para um teste completo do programa.

Para a aplicação do FORM# 10, cada fase do programa de teste deverá receber um código de identificação.

10.4. Casos de teste

Cada caso de teste deverá ser explicado em termos da organização e conteúdo do arquivo e entradas utilizadas.

As saídas esperadas para cada teste devem ser expressas através de amostras dos resultados previstos.

2.3.2.2. Manual de Implementação

NOME DO PROGRAMA:

CÓDIGO:

Linguagem Utilizada: nome da linguagem de programação utilizada

## 1. ENTRADAS DO PROGRAMA

Todas as entradas do programa deverão ser descritas. Os formulários FORM# 6 e FORM# 7 poderão ser utilizados na medida em que ajudem a esclarecer a situação. Quando for o caso, referências a outros manuais deverão ser feitas.

## 2. SAÍDAS DO PROGRAMA

Todas as saídas do programa deverão ser descritas. Os formulários FORM# 6, FORM# 7 e FORM# 8 poderão ser utilizados na medida em que ajudem a esclarecer a situação. Quando for o caso, referências a outros manuais deverão ser feitas.

## 3. PRINCIPAIS FUNÇÕES DO PROGRAMA

Deve ser feita uma breve descrição das funções a serem realizadas pelo programa. A seguir, deve ser feita uma descrição detalhada a respeito do interrelacionamento destas funções.

## 5. IMPLEMENTAÇÃO DAS ESTRUTURAS DE DADOS

Descrever as operações que serão aplicadas a cada estrutura, com a explicação do processamento de cada uma, justificando, assim, o uso da estrutura; comentar a natureza de cada variável do programa, especificando se a mesma é de entrada, saída, ou de comunicação entre programas.

## 6. RESTRIÇÕES DE IMPLEMENTAÇÃO

Esta seção inclui, por exemplo, o número máximo de posições de um vetor, dada sua declaração inicial.

7. MÉTODOS DE RESOLVER CALCULOS, FUNÇÃO E/OU PROCEDURES

Apresentar como foi codificado cada método, justificando como este se ajusta à codificação.

8. EQUIPAMENTO NECESSÁRIO

Descrição dos dispositivos necessários à implementação do programa.

9. IMPLEMENTAÇÃO DO ALGORITMO

Desenvolvimento do problema por refinamentos sucessivos até a sua completa codificação (programa).

### 2.3.2.3. Manual de Utilização

#### Parte A: Introdução

##### 1) Background geral de informações

- compreende um resumo dos conhecimentos necessários ao entendimento do programa e seus objetivos.

##### 2) Objetivo do programa

- colocar de maneira clara e concisa o objetivo da aplicação do programa.

##### 3) Que problemas podem ser resolvidos pelo programa

- descrição geral dos problemas para os quais o programa em questão é aplicável; não é necessário detalhar a descrição.

##### 4) Descrição do uso "padrão" do programa

- esta seção especifica a maneira objetiva e simples de se encarar o programa.

##### 5) Descrição de todas as maneiras de usar o programa, informalmente

- descrever de forma sucinta os usos "periféricos" (em relação ao uso "padrão") do programa.

##### 6) Listar e explicar as funções e procedures que o programa usa

- explicar brevemente, dizendo em cada caso se a função ou procedure é obrigatória ou opcional.

##### 7) J C L

- apresentar as opções do JCL necessário à execução do programa.

8) Restrições e limitações

- relacionar as limitações e restrições do programa.

Parte B:

1) Informação completa de como usar o programa

- esta seção completa o conhecimento inicialmente adquirido com os itens 1 a 5 da PARTE A; inclui uma descrição das instruções com explicações e comentários que se fizerem necessários.

2) Especificar os dados de entrada (cada tipo):

2.1) objetivo

2.2) ilustração do formato

2.3) exemplo

2.4) campos e subcampos → descrição dos objetivos de cada um

2.5) tamanho → se for variável, indicar a gama de variação

2.6) categoria do conteúdo → binário, hexadecimal ou outra categoria.

3) Os passos gerais do processamento

- como cada "step" altera os dados de entrada; se o processamento varia de acordo com os dados de entrada, explicar como varia; indicar como o usuário especifica qualquer processamento opcional; se as opções são requisitadas por instruções de controle, discutir cada instrução e apresentar qualquer parâmetro possível de ser incluído na mesma.

4) Interpretação das saídas (cada tipo):

4.1) objetivo

4.2) ilustração do formato

4.3) exemplo

- 4.4) explicar o significado da informação
- 4.5) sugerir possíveis usos para a informação

5) Arquivos e recursos usados

- descrição sucinta, fazendo referência a outros manuais onde o usuário poderá encontrar os detalhes necessários ao conhecimento total do assunto.

6) Tempo de processamento e espaço estimados como função dos dados

- apresentar uma "média" de tempo e quantidade de memória utilizada em relação à entrada do programa.

7) Mensagens de erro

- se não houver um manual de mensagens de erro, é necessário listá-las e fornecer explicação adicional necessária; caso contrário pode-se fazer uma referência ao manual.

8) Condição de erro

- indicar as saídas anormais do programa.

9) Correção de erro

- procedimentos para verificar a validade da saída; inclusive, apontando possíveis erros nos dados de entrada.

10) Modificações possíveis e extensões

11) Exemplos de utilização



APÊNDICE

FORM# 1 .....	A-1
FORM# 2 .....	A-2
FORM# 3 .....	A-3
FORM# 4/1 .....	A-4
FORM# 4/2 .....	A-5
FORM# 5/1 .....	A-6
FORM# 5/2 .....	A-7
FORM# 6 .....	A-8
FORM# 7 .....	A-9
FORM# 8 .....	A-10
FORM# 9 .....	A-11
FORM# 10 .....	A-12

RELATÓRIO SEMANAL

Nº

NOME:

DATA / /

FOLHA DE

[Empty rectangular box for report content]

RELATÓRIO QUINZENAL

Nº

NOME:

DATA: / /

PRINCIPAIS ATIVIDADES DA ÚLTIMA QUINZENA: \_\_\_\_\_

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

PROGRESSOS EM RELAÇÃO A PENÚLTIMA QUINZENA: \_\_\_\_\_

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

OBJETIVOS ALCANÇADOS EM RELAÇÃO AOS LISTADOS NA PENÚLTIMA QUINZENA: \_\_\_\_\_

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

OBJETIVOS A SEREM ATINGIDOS NA PRÓXIMA QUINZENA: \_\_\_\_\_

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

ASSINATURA

FORM# 2

RELATÓRIO MENSAL

Nº

MES:

NOME:

DATA: / /

FOLHA: DE

[Empty rectangular box for report content]

RELATÓRIO TRIMESTRAL

Nº

NOME:

DATA: / /

FOLHA: DE

MES DE

MES DE

MES DE

MES DE

FOLHA DE

A large, empty rectangular box with a black border, intended for data entry or drawing.

MES DE

A large, empty rectangular box with a black border, intended for data entry or drawing.

MES DE

A large, empty rectangular box with a black border, intended for data entry or drawing.

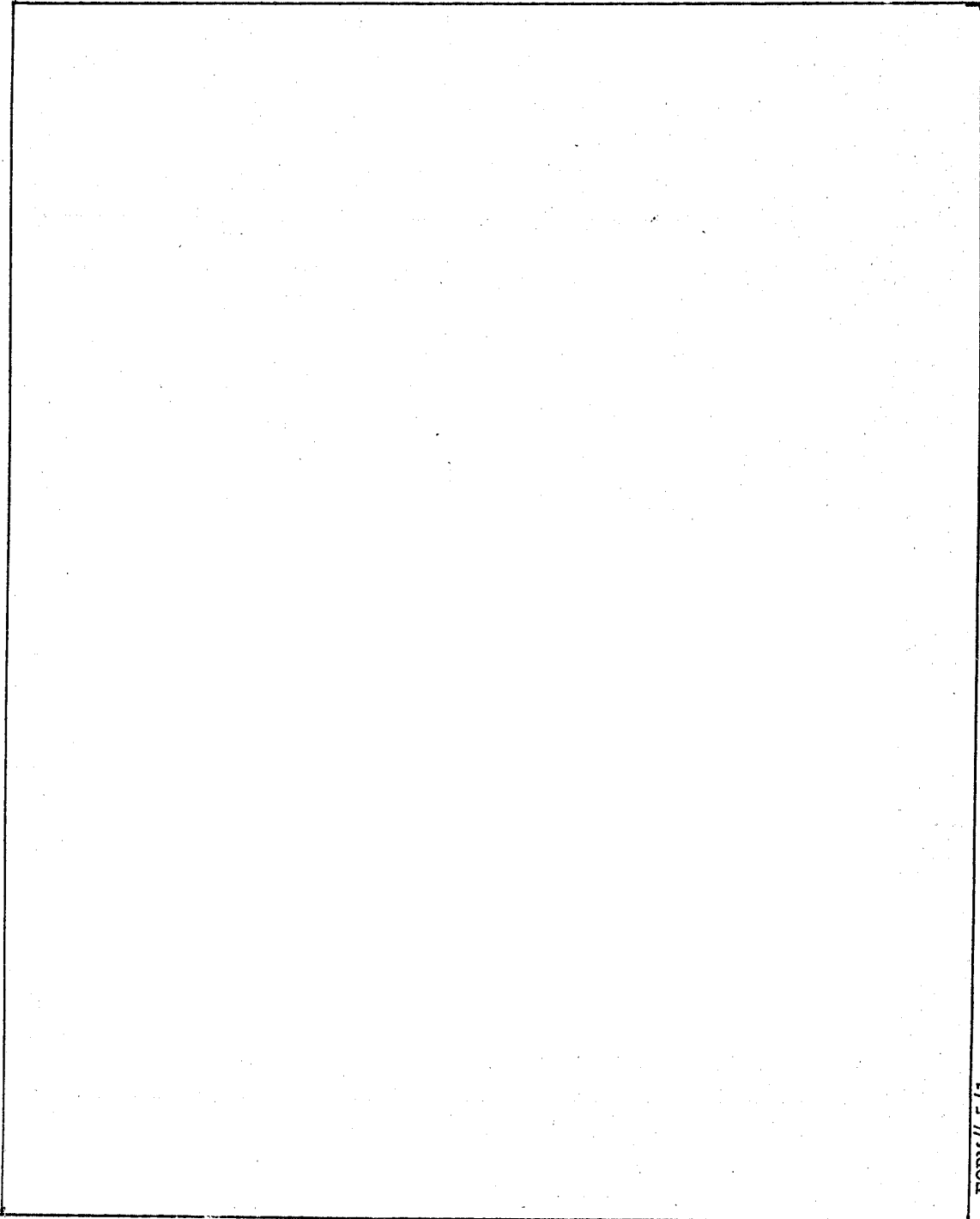
RELATÓRIO ANUAL

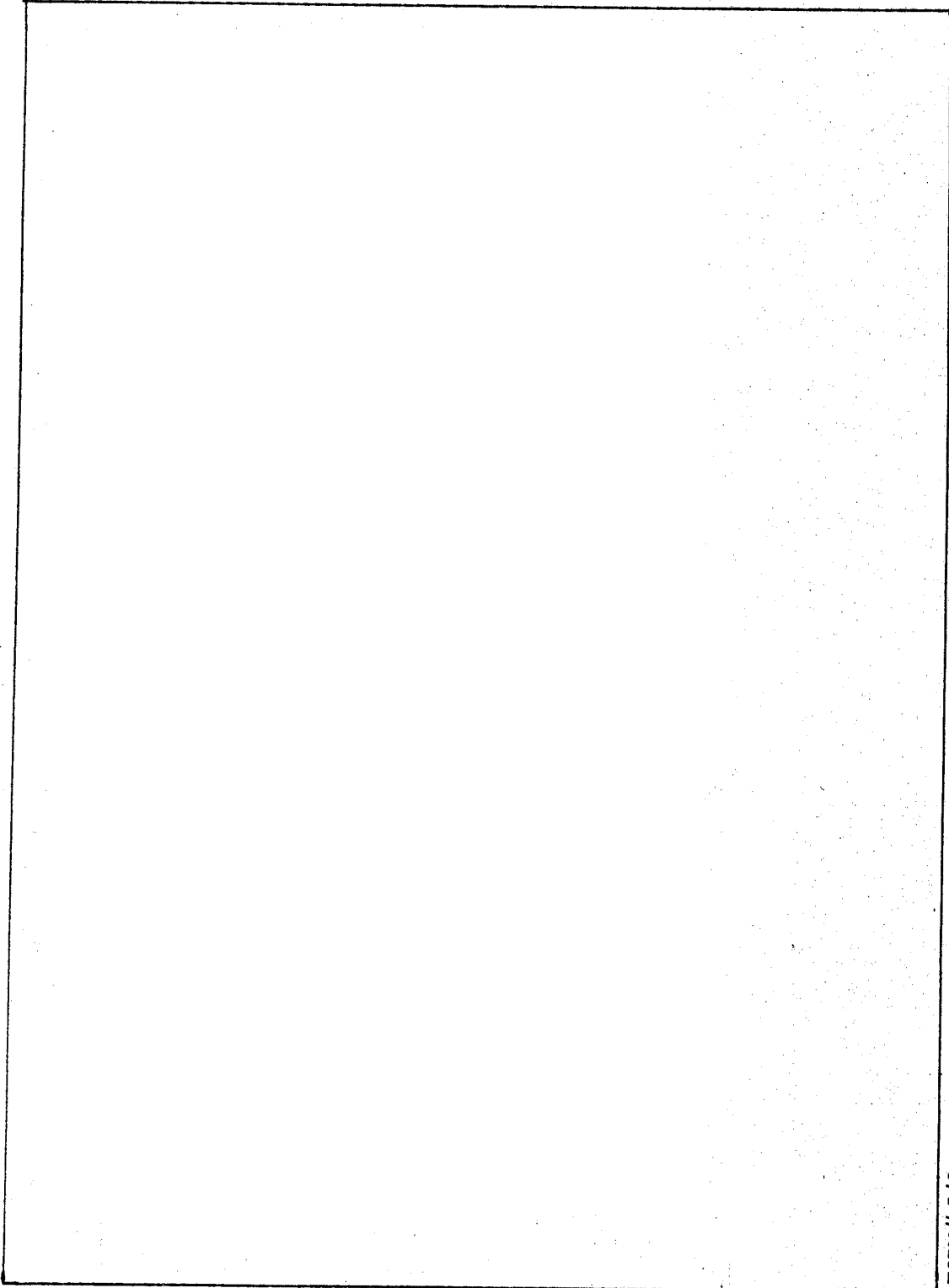
Nº

NOME:

DATA / /

FOLHA DE









# IBM System/360 Record Layout Worksheet

**BYTES:**  
 8 - double word  
 4 - word  
 2 - halfword  
 1 - 1/2-packet-decimal digit

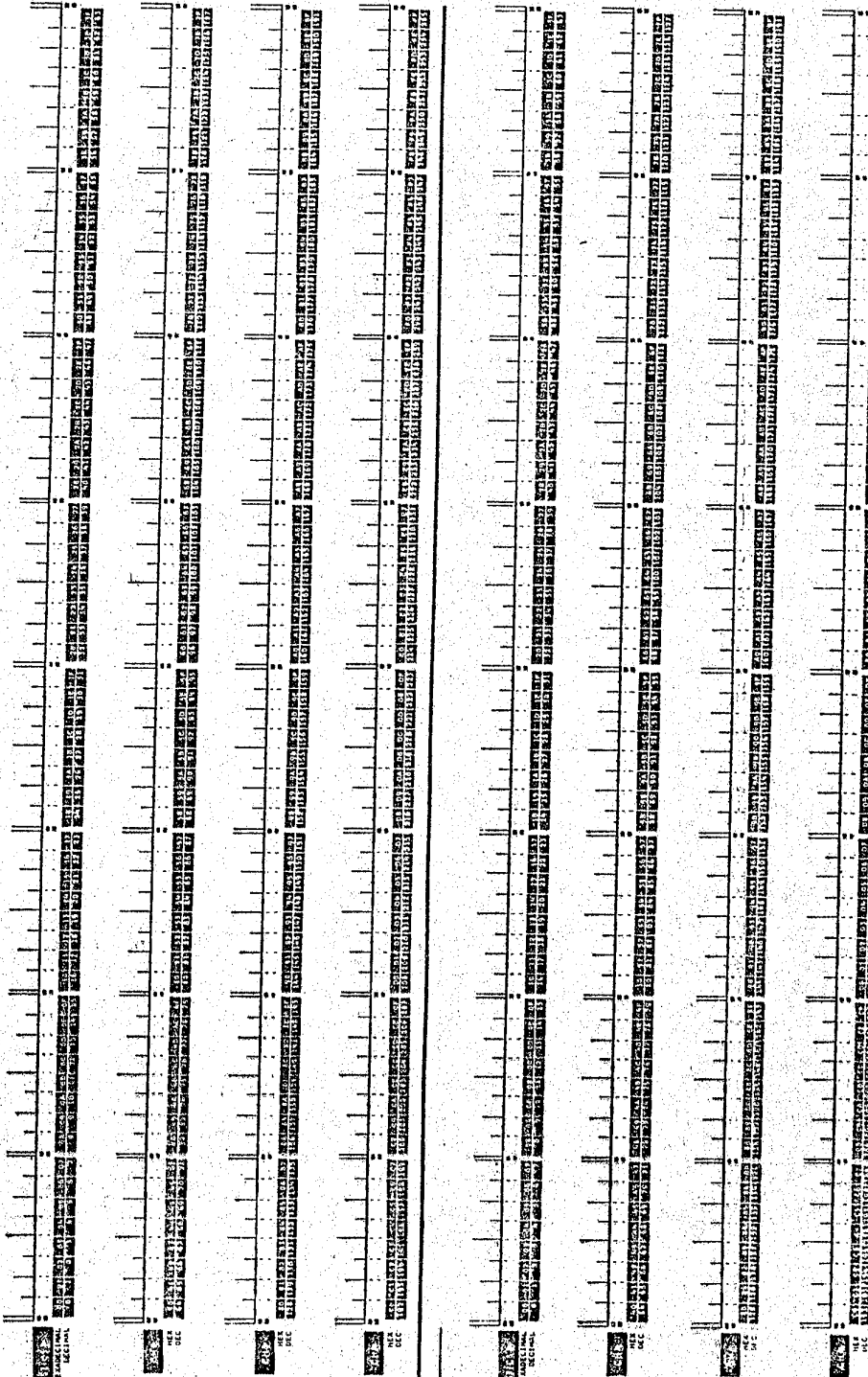
INTERNATIONAL BUSINESS MACHINES CORPORATION

X30-1711-0 U/M 025  
 Printed in U.S.A.

Record Name \_\_\_\_\_

APPLICATION \_\_\_\_\_

Page \_\_\_\_\_ of \_\_\_\_\_  
 Date \_\_\_\_\_



HEX	DEC
000	256
100	512
200	768
300	1024
400	1280
500	1536
600	1792
700	2048
800	2304
900	2560
A00	2816
B00	3072
C00	3328
D00	3584
E00	3840
F00	4096

HEX	DEC
000	256
100	512
200	768
300	1024
400	1280
500	1536
600	1792
700	2048
800	2304
900	2560
A00	2816
B00	3072
C00	3328
D00	3584
E00	3840
F00	4096

**CHARACTERISTIC CODES**

- C - character, 8-bit code
- K - hexadecimal, 4-bit code
- B - binary
- F - fixed-point, full word
- H - fixed-point, halfword
- P - packed decimal
- Z - zoned decimal
- G - floating-point, full word
- D - floating-point, double word
- A - address value, full word
- Y - address value, halfword
- V - address, extended symbol
- S - address, base displacement


 PONTIFÍCA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO RIO DATACENTRO		GABARITO DE IMPRESSÃO											
PROGRAMA		EMPREGADO											
	0	1	2	3	4	5	6	7	8	9	10	11	12
1	A	2	3	4	5	6	7	8	9	10	11	12	
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													
41													
42													
43													
44													
45													
46													
47													
48													
49													
50													

TABELA DE DECISÃO

PROGRAMA:  
FOLHA DE

PREPARADA POR: \_\_\_\_\_

AÇÃO OU CONDIÇÃO	REGRA												ELSE	
	1	2	3	4	5	6	7	8	9	10	11	12		

PLANO DE TESTE

NOME DO PROGRAMA:

FASE	TESTE	ENTRADAS REQUERIDAS	ARQUIVOS REQUERIDOS	RESULTADOS

4. BIBLIOGRAFIA

- [1] LONDON, K.R - Documentation Standards, Petrocelli Books, New York, 1974.
- [2] van DUIN, J.- Documentation Manual, Auerbach, 1972.
- [3] CLARK, F.J, et alli - Business Systems and Data Processing Procedures  
Prentice-Hall, 1972.
- [4] WALSH, D.A. - A guide for software documentation, McGraw-Hill, 1969.
- [5] GLEIM, G.A. - Electronic data processing systems and procedures, Prentice-Hall, 1971.
- [6] IBM - HIPO - A design aid and documentation technique, GC20-1851.
- [7] TEICHROEW, D. - Computer aided documentation overview, Ann Arbor, Univ.of Michigan, 1975.
- [8] GOOS, G. - Documentation. Munich, Technical Univ. of Munich, 1972.
- [9] MIKELSON, M. e WLADAWSKY, I. - On the formal documentation of programs.  
Yorktown Hights, IBM Cj. Watson Research Center, 1976.
- [10] LLOYD, L. e GERBRANDT, L. - UBC documentation: a guide to documentation produced by the UBC computing center. Vancouver, Univ. of British Columbia, 1974.
- [11] LUCENA, C.J. - Manual para o projeto e documentação de sistemas de programação, Rio de Janeiro, PUC-RJ, 1976.
- [12] STEINBRUCH, P.L. e FRESNEDA, P.S.V. - Novos aspectos em modularização de programas, Revista Brasileira de Tecnologia, vol.7, 1976.
- [13] OLIVEIRA, A.M. de e SOARES, O.A. - Sugestões de documentação e normas para a programação, Rio de Janeiro, IBGE, 1975.

- [14] MELLO, R.N. e STEINBRUCH, P.L. - FORT-S: Um preprocessor para FOR-  
TRAN-IV estruturado - Rio de Janeiro - PUC-RJ, a ser publi-  
cado.
- [15] KUEHNE, R.S. et alli - Manual of Computer documentation standards -  
Prentice-Hall, 1972.