

ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO

TRABALHO DE FORMATURA

ESTIMATIVA DE PRAZOS E ORÇAMENTO DE PROGRAMAS

AUTOR - PAULINO G. FRANCISCHINI

ORIENTADOR - JOSÉ WOILER

## A G R A D E C I M E N T O

---

- . Ao Prof. José Woiler pela valiosa orientação e acompanhamento dado ao trabalho.
- . Ao Sr. Roberto Ferraro Alves, pelo apoio oferecido e pela oportunidade de estágio.
- . Ao Sr. Nilo Mario Optiz, Supervisor do Departamento de Assessoria em Assuntos Especiais, pela experiência a mim transmitida.
- . À Sra. Denize Cini da Costa, pelo excelente trabalho de datilografia.

A minha família

## S U M A R I O

A necessidade deste trabalho nasce com a constante elevação dos custos de pessoal especializado nos Centros de Processamento de Dados, ocupando o lugar a que antes pertencia aos custos com equipamentos.

Apresentamos a aplicação de um método de estimativa de prazos em programação visando atingir dois objetivos: minimizar os riscos em orçamento de programas e sistemas para clientes e avaliar a produtividade dos programadores nas suas atividades de desenvolvimento e manutenção de programas.

Utilizando-se de variáveis objetivas e minimizando as avaliações subjetivas, os dados são tratados em bases estatísticas chegando a resultados de alta confiabilidade, ao contrário dos métodos que apresentam somente variáveis e avaliações subjetivas.

## Í N D I C E

---

Fls.

Sumário	1
1 - A Empresa	5
1.1 - As Empresas Lojicred	5
1.2 - Histórico da Lojicred Processamento de Dados	9
2 - Estrutura do Trabalho	
2.1 - Necessidade do Trabalho	10
2.2 - Definição do Objetivo	11
2.3 - Critérios Utilizados	13
2.3.1 - Fases Produtivas	13
2.3.2 - Fases Não Produtivas	16
2.3.3 - Tipos de Programas	17
2.3.4 - Variáveis Utilizadas	25
2.3.5 - Variáveis Subjetivas	35
3 - Fundamentos Teóricos	
3.1 - Regressão Linear Múltipla	39
3.2 - Correlação Linear Múltipla	41
3.3 - Correlação Parcial	41
3.4 - Análise de Melhoria	42
3.5 - Inclusão de Variáveis pelo processo STEPWISE	42
3.6 - Intervalo de Confiança	43
4 - Análise de Trabalhos Semelhantes	
4.1 - Critérios para Estimativa de Tempos	45
4.1.1 - Linhas de Codificação	45
4.1.2 - Pesos para Funções	46
4.1.3 - Curvas Aproximadas	48
4.2 - Critérios de Produtividade	
4.2.1 - Linhas de Codificação	51
4.2.2 - Tempo Real x Tempo Estimado	51

<b>5 - Sistema Atual de Controle de Atividades</b>	
<b>5.1 - Fluxo de Informações</b>	<b>53</b>
<b>5.2 - Atividades no Departamento de Análise e Programação</b>	<b>55</b>
<b>5.2.1 - Analistas de Sistemas e O &amp; M</b>	<b>55</b>
<b>5.2.2 - Analistas de Software</b>	<b>57</b>
<b>5.2.3 - Documentador de Sistemas</b>	<b>57</b>
<b>5.2.4 - Programadores</b>	<b>57</b>
<b>5.3 - Apontamento de horas</b>	<b>57</b>
<b>5.4 - Relatórios</b>	<b>61</b>
<b>5.5 - Crítica do Sistema Atual</b>	<b>62</b>
<b>5.5.1 - Distorções no Apontamento das horas</b>	<b>63</b>
<b>5.5.2 - Falhas nos Relatórios</b>	
<b>5.5.3 - Falhas na Análise dos Relatórios</b>	<b>66</b>
 <b>6 - O Método de Estimativa de Tempos e Orçamento</b>	
<b>6.1 - Estudo Preliminar</b>	<b>68</b>
<b>6.2 - Levantamento de Dados</b>	<b>72</b>
<b>6.2.1 - Pesquisa de Tempos de Desenvolvimento e Manutenção de Programas</b>	<b>72</b>
<b>6.2.2 - Categorização de Programas</b>	<b>74</b>
<b>6.2.3 - Tabelamento de Variáveis Objetivas</b>	<b>77</b>
<b>6.2.4 - Avaliação das Variáveis Subjetivas</b>	<b>85</b>
<b>6.2.4.1 - Qualificação Individual do Programador</b>	<b>85</b>
<b>6.2.4.2 - Lógica do Programa</b>	<b>87</b>
<b>6.2.4.3 - Complexidade da Alteração</b>	<b>87</b>
<b>6.3 - Equações e Tabelas</b>	<b>89</b>
<b>6.3.1 - Desenvolvimento</b>	<b>89</b>
<b>6.3.2 - Manutenção</b>	<b>91</b>
<b>6.3.3 - Tolerância</b>	<b>92</b>
<b>6.4 - Aspecto Orçamentário</b>	<b>93</b>
<b>6.5 - Avaliação de Produtividade</b>	<b>99</b>
 <b>7 - Análise de Resultados</b>	
<b>7.1 - Desenvolvimento de Programa</b>	<b>101</b>
<b>7.1.1 - Variáveis Objetivas</b>	<b>102</b>
<b>7.1.2 - Tipos de Programas</b>	<b>103</b>

7.2 - Manutenção de Programas	111
7.2.1 - Localização da Manutenção	111
7.2.2 - Classes de Dificuldade	111
7.3 - Aplicações	
7.3.1 - Desenvolvimento de Programas	113
7.3.2 - Manutenção de Programas	118
8 - Conclusões	120
9 - Bibliografria	122
Anexos	
A.1 - Relatórios do Sistema Atual de Controle de Atividades	123
A.2 - Tabela de Correlações	127
A.3 - Tabelas finais da Regressão	135
A.4 - Cálculo de Orçamento	143

## 1 - A Empresa

### 1.1- As Empresas Lojicred

As empresas Lojicred iniciaram suas atividades em 1964 com fundação da LOJISTA S.A., hoje denominada LOJICRED FINANCEIRA S.A., que capta recursos de poupança, através do aceite de letras de câmbio emitidas, e o repasse na forma de financiamentos de crédito.

Pela própria dinâmica do setor financeiro, outras duas empresas vieram para assessorar a financeira, responsabilizando-se exclusivamente pelas formas de captação e repasse: LOJICRED PROMOTORA DE VENDAS LTDA, hoje LOJICRED SERVIÇOS S.A. que tem sob sua responsabilidade os compromissos por aval, cobrança, abertura de crédito e controle de conta corrente dos crediaristas e LOJIVAL S.A., hoje LOJICRED DISTRIBUIDORA DE TÍTULOS E VALORES MOBILIÁRIOS S.A., operando no mercado de capitais, para a colocação das letras de câmbio da financeira.

Para atuar na administração, compra e venda de valores cotados na Bolsa de Valores, mercado de Balcão e no "overnight", foi criada a LOJIMEC S.A., depois denominada de LOJICRED CORRETORA DE CÂMBIO E TÍTULOS S.A. completando o quadro das empresas que atuam no setor financeiro. Uma quinta empresa, LOJIPLAN CONTROLE E SISTEMAS LTDA., hoje LOJICRED PROCESSAMENTO DE DADOS S.A., assessora as demais empresas do Grupo e à terceiros atuando como "bureau" de processamento de dados.

### 1.2- Histórico da Lojicred Processamento de Dados

A Lojicred Processamento de Dados Ltda, iniciou suas atividades em janeiro de 1972, atuando com um CPD das Empresas Lojicred sendo sua denominação na época Lojiplan Controles e Sistemas Ltda, e os sistemas iniciais desenvolvidos foram os de Controle de Crediário e de Controle dos Lojistas Filados.

No primeiro semestre de 1977 já com uma série de sistemas padrões desenvolvidos e tendo atendido as necessidades básicas das empresas do grupo, a empresa, utilizando-se das

vantagens da economia da escala, intensificou sua presença no mercado e estendeu seus produtos padrões a um número maior de clientes, atingindo de forma mais concreta sua posição de "Bureau".

Atualmente a empresa atende as Empresas Lojicred principalmente nas áreas de planejamento e atualização dos processos de controle e desenvolve em paralelo, sistemas voltados para os demais clientes ativos ou potenciais, buscando uma maior autonomia econômico-financeira.

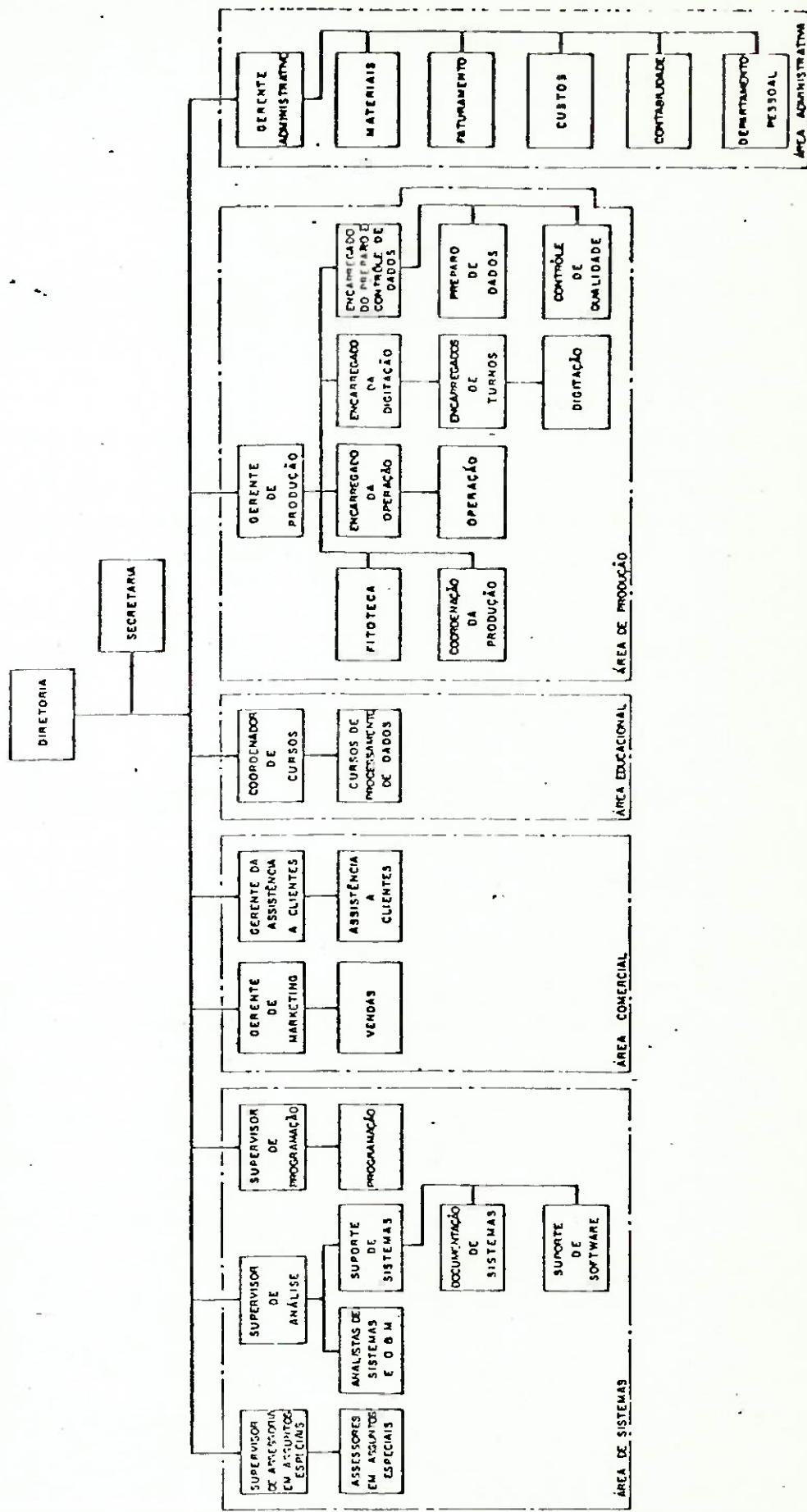
Com a experiência adquirida e os recursos disponíveis, a empresa resolveu atuar também na área do ensino, montando cursos profissionalizantes voltados para a área de processamento de dados, iniciando com cursos de programação COBOL e incrementando em seguida cursos de operação e digitação.

A configuração atual dos equipamentos é a seguinte:

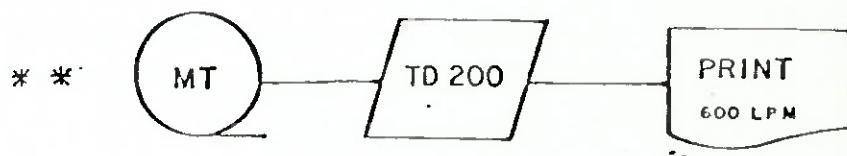
QUANTIDADE	EQUIPAMENTO	FORNECEDOR	TIPO
1	CPU com 240 KB	IBM	3145
1	Console	IBM	3215
1	Console de vídeo	IBM	3278
3	Disco 87,3 MB	IBM	2319
1	Disco 58,2 MB	IBM	2318
1	Leitora/Perfuradora	IBM	2540
1	Impressora	IBM	1403
5	Fitas 200 KB/s	IBM	3420
13	Terminais de vídeo	IBM	3278
10	Unidades de Diskette	IBM	3742
1	Conversora Diskette	IBM	3747
1	Unidade de Gravação/conversão	COBRA	TD 200
1	Unidade de fita	COBRA	MT
1	Impressora	COBRA	PRINT
4	Unidades de Gravação	COBRA	TD 100

As páginas seguintes mostram o Organograma da Empresa e a configuração atual do equipamento.

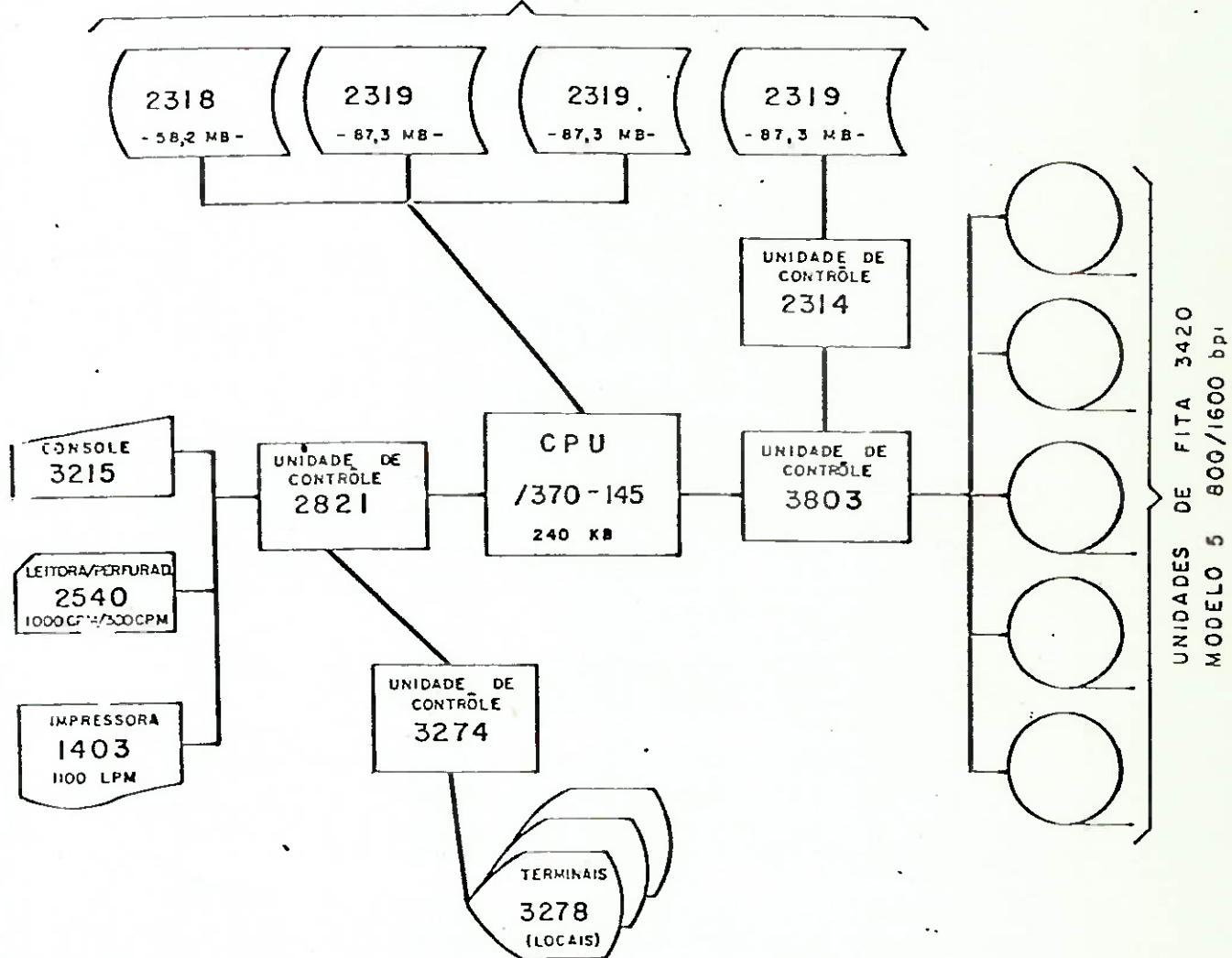
**ORGANOGRAMA DA EMPRESA / CPD**



CONFIGURAÇÃO - ATUAL



UNIDADES DE DISCO → II. SPINDLES = 320,1 MB



TERMINAIS - QUANTIDADE TOTAL = 14

### 1.3- O Estágio

O estágio foi desenvolvido no Departamento de Assessoria em Assuntos Especiais, inicialmente para desenvolver um sistema que viria resolver um problema específico da empresa. Por não ser exequível no tempo exigido para o desenvolvimento do Trabalho de Formatura, foi abandonado logo nos levantamentos iniciais de dados.

Todo o apoio por parte da Gerência e da Diretoria Técnica foi dado para encontrar-se novo tema, sendo escolhido o que agora apresentamos, como fruto de um estudo para encontrar um método seguro de orçar programas e impor metas ao departamento de Programação para o cumprimento da carteira de solicitações.

Pelas próprias características do trabalho, o desenvolvimento do estágio consistiu na sua maior parte, em um levantamento de todos os dados históricos que estavam disponíveis, tanto em desenvolvimento quanto em manutenção de programas. Afastando o perigo de uma má documentação dos manuais, a pesquisa das variáveis objetivas foi feita nas pastas de documentação dos programas para garantir uma boa qualidade de dados. Os tempos de desenvolvimento foram obtidos a partir das Folhas de Serviços Executados somando os totais parciais mês a mês, referentes a um programa específico. As descrições das manutenções analisadas foram obtidas nas Folhas de Alteração de Programas, documento emitido pelo Departamento de Análise para o Departamento de Programação. Os tempos de manutenção seguem o mesmo esquema do desenvolvimento.

Para que a análise estatística tivesse significância, foram pesquisados quase 150 programas, mais de duas centenas de alterações, bibliografia especializada e contando sempre com a experiência dos Gerentes de Departamento. Cabe aqui, um agradecimento a todos aqueles que direta ou indiretamente colaboraram na realização deste trabalho.

## 2 - Estrutura do Trabalho

### 2.1- Necessidade do Trabalho

Se verificarmos todos os programas em desenvolvimento e em manutenção de qualquer departamento de Programação, veremos neles uma característica em comum: estão sempre atrasados.

Com isto, a carteira de trabalho deste departamento é enorme e sistematicamente os cronogramas não são cumpridos.

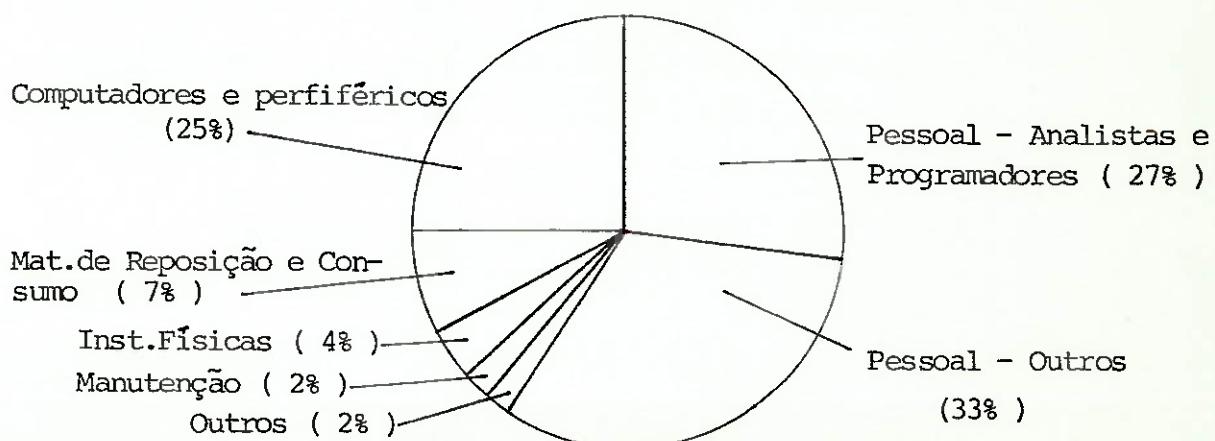
No caso de firmas prestadoras de serviços em computação, este problema é grave porque primeiro não há meios de fazer um orçamento para clientes, correndo um sério risco de prejuízo se o tempo for mal estimado, e de perder o cliente se a margem de segurança for alta demais; segundo é que um atraso no atendimento cria uma péssima imagem da empresa e constantes reclamações para o Departamento de Assistência a Clientes; terceiro, que para suprir estas deficiências, são obrigados a contratar um maior número de programadores, elevando assim os custos da Programação.

Segundo dados da SUCESU-SP (Sociedade dos Usuários de Computadores e Equipamentos Subsidiários de São Paulo), os salários e encargos pagos a analistas e programadores são responsáveis por 46% do total de despesas de pessoal, distribuídos em:

- 50% Análise
- 50% Programação

Em termos gerais, os custos em processamento de dados no Brasil, distribuem-se conforme o gráfico abaixo:

Fonte: SUCESU



A preocupação em medir a produtividade de um CPD, começou a menos de 10 anos podendo ser facilmente verificado pelo número de publicações que se fizeram nesta área antes e depois de 1970. Isto se deve ao fato de que os custos de desenvolvimento se software eram muito baixos em relação aos de produção de hardware. Durante a última década o avanço tecnológico nos computadores, permitiram uma queda no preço de venda de equipamentos, a tal ponto de hoje apresentar apenas 25% dos despêndios do CPD.

Segundo Feldmann, apesar destes artigos e livros serem de excelente nível, ainda não conseguem suprir algumas dificuldades tais como a de definição das tarefas em análise e programação, a de se conhecer a complexidade das tarefas e medir sua importância. Para um sistema automatizado de 10 anos, os custos de desenvolvimento e manutenção estão porcentualmente lotados:

Desenvolvimento	25 a 40%
Manutenção	60 a 75%

A relação do custo da manutenção com o custo total de desenvolvimento pode ser analisada como:

Primeiro ano	23%
Primeiros 4 anos	85%

Estes dados justificam este estudo abrangendo as duas atividades da programação. As atividades no departamento de análise são variadas, sendo complexa a definição de tarefas-padrão para aplicarmos um método arcálogo, porém, é uma porta aberta para estudos posteriores, já que é igualmente responsável nos custos do CPD.

Em média, um analista recebe salários mais altos do que um programador com o mesmo tempo de experiência, e é comum que um programador que se destaca pela sua qualificação pessoal seja admitido no departamento de Análise. O que equipara os custos em termo de departamento é pelo maior número de programadores que analistas.

## 2.2- Definição do Objetivo

O objetivo principal é o desenvolvimento de um método seguro de estimar tempos em programação, derivando deste outros dois colaterais: permitir o orçamento de desenvolvimento e

manutenção de programas, e avaliar a produtividade dos programadores.

Isto vem atender à necessidade do CPD de se organizar para atender às solicitações que lhe são encaminhadas, permitindo conhecer o que se pode exigir de cada programador individualmente, repreendendo ou premiando sua performance e ao mesmo tempo dimensiona o número de programadores necessários para atender o volume de trabalho solicitado ao departamento.

Conforme discutiremos no capítulo 4, uma das diretrizes tomadas desde o início do trabalho foi a fuga às estimativas subjetivas, uma vez que já existem diversos artigos e trabalhos sob este enfoque, cujos parâmetros e pesos carecem de justificativa. Porém durante o desenvolvimento deste trabalho, ficou evidente a impossibilidade de eliminarmos totalmente estas estimativas subjetivas e procurarmos minimizá-las tanto no número quanto na dificuldade em avaliá-las, colocando padrões de dificuldades.

O método aqui apresentado foi baseado na tese de Mestrado "Como estimar prazos certos em programação" de Paulo Roberto Feldmann, FGV, no entanto, uma nota importante que ele mesmo chama a atenção é a seguinte "O sucesso na aplicação da presente metodologia é função diretamente proporcional do fato de o CPD adotar a filosofia de produzir programas pequenos e com poucas funções cada". Na maioria dos Centros de Processamento de Dados, esta regra não é observada e o que encontramos com frequência são programas com várias funções, o que dificulta muito a correlação entre as variáveis independentes e o tempo estimado da atividade.

O enfoque dado na manutenção de programas é diferente do desenvolvimento. Enquanto que este último usa quase que somente variáveis objetivas, o segundo não guarda correlação com nenhuma variável que possa ser obtida a priori, tendo que ater-nos à padronização das alterações e analisar dados históricos.

No próximo ítem analisa-se os critérios utilizados no levantamento de dados e no capítulo seguinte o resumo teórico dos procedimentos estatísticos. Os demais critérios que são encontrados em outros métodos são confrontados para justificar a escolha. O capítulo 5 traz a crítica do sistema atual de controle de atividades utilizado pelo CPD; e o desenvol-

desenvolvimento do sistema manual de estimativa e orçamento aparece no capítulo 6.

A análise de resultados obtidos na aplicação do método proposto evidenciará sua necessidade de ser implantado.

Os objetivos propostos vem atender à solicitação do departamento de Programação da Copicred Processamento de Dados ao qual foi proposto um sistema mecanizado tendo os seguintes objetivos:

- a) - Gerar relatórios de previsão e orçamento de desenvolvimento de programas na Programação, utilizando variáveis objetivas.
- b) - Gerar relatórios de previsão e orçamento de manejo de programas na Programação, utilizando variáveis objetivas e minimizando as subjetivas.
- c) - Gerar relatórios de acompanhamento de eficiência de programadores.
- d) - Gerar relatórios de histórico de manutenção mês a mês nos últimos 2 anos.

### 2.3- Critérios Utilizados

Uma das maiores dificuldade para a estimativa do tempo de programação é o isolamento dos fatores que influem na confecção e manutenção de programas. Vários foram os métodos propostos para solucionar este problema. Brandon aponta os fatores ambientais intrínsicos de um CPD, afetando o tempo que o programador efetivamente produz (1).

#### 2.3.1- Fases Produtivas

Durante esta fase distingue cinco categorias de atividade: estudo, codificação, depuração, testes e documentação.

Estudo: Começa com o recebimento da pasta do programa do Departamento de Análise, contendo a definição do problema, arquivos, registros, códigos, etc. No decorrer desta fase, entra em contacto com o analista responsável para tirar possíveis dúvidas a respeito do progra-

(1) Project Control Standard, BRANDON, Dick

ma. Esta fase termina com a confecção do diagrama de blocos.

Nas manutenções de programas, o programador recebe a "Folha de Alteração de Programas", ao mesmo tempo, requisita do arquivo, a "passa de programa", contudo a folha de operações lay-out de registros, etc. Esta fase termina na localização dos acertos que devem ser feitos conforme a requisição do Departamento de Análise.

Codificação: O produto da fase anterior deve ser agora codificado para tornar operacional para o computador, ou seja descrever a mesma sequência numa linguagem que o computador possa entender. O nível da linguagem utilizada influirá no tempo de codificação, mas como a linguagem COBOL é utilizada na quase totalidade dos programas que encontramos nos CPD, simplesmente desprezamos a influência desta variável considerando apenas esta linguagem como utilizada.

Em manutenção de programas, esta fase varia muito dependendo do grau de dificuldade da manutenção. Na categoria de manutenção simples, pode-se mudar apenas 2 cartões como é o caso de "estouro de acumuladores" ou "Renames do Sort" o outro extremo se verifica na categoria de manutenção difícil, onde muitas vezes é necessário refazer a quase totalidade do programa.

Depuração: Durante a fase de codificação, ocorrem erros em instruções, dimensionamentos e formatos que são normais em desenvolvimento de programas e mais raros em manutenções simples, mas cabe ao programador gastar um certo tempo para que as instruções fiquem intelegráveis para a máquina. O tempo gasto nesta fase é grande para programadores junior e "trainee", mas decresce rapidamente com a experiência adquirida até

estabilizar quando atinge a categoria de Senior (diz-se que o programador está "tarembado").

Testes: Esta talvez seja a fase mais importante do desenvolvimento e manutenção de programas pois é responsabilidade do programador testar o programa com todas as casuísticas possíveis que possam ocorrer na operação, antes de liberá-lo. A medida que se verifique as combinações de IF's, detecte erros de preenchimento de formulários, maior será a probabilidade de prever as variações que podem ocorrer nos dados de entrada, acarretando cancelamento do programa já em operação, e muitas vezes, as consequências são graves. O analista que definiu o programa também se encarrega de testá-lo, juntamente com outros programas do mesmo sistema, também com o mesmo objetivo.

Em manutenção esta fase é a que ocupa mais tempo em todo processo. Dado que na maioria das vezes as manutenções requerem poucos cartões (salvo casos extremos), o tempo de teste é significativamente maior conforme veremos mais adiante. Isto se explica pelo fato de que, ao alterar, ou incluir alguns poucos cartões, geralmente modifica a entrada para alguma(s) outra(s) subrotina(s) dentro do programa que poderão não aceitar a modificação introduzida.

Ocorre que estas alterações se alastram por todo o programa que muitas vezes compensa mais fazer um programa novo do que tentar modificar o antigo.

Documentação: A documentação geralmente apresenta um esquema único de apresentação. Aqui se coloca os objetivos, tabelas, procedimentos, cálculos, mensagens pela consola, em resumo, tudo o que foi feito no programa para que possa tornar mais

fácil sua manutenção.

No caso de manutenções esta fase não existe ou é mínima pois as modificações já vem documentadas do Departamento de Análise através da Folha de Alteração de Programas.

Quando a manutenção modifica profundamente um programa, alterando o nº de arquivos, ou registros de entrada ou saída, inclusão de chaves UPSIS, acompanha um outro documento chamado "Alteração de Manuais" (AM).

#### 2.3.2- Fases não Produtivas

As fases não produtivas estão diretamente relacionadas com os fatores ambientais podendo ser divididas em:

Necessidades Pessoais: aqui engloba tanto as necessidades fisiológicas quanto psicológicas; O grau de concentração com que trabalha um programador requer um período de descanso para evitar a fadiga excessiva. Se o programador atinge esta situação, devido à urgência na liberação de sistemas, que o faz trabalhar em ritmo intenso durante vários dias (e noites), sua produtividade cai verticalmente.

Espera: As restrições de Hardware ou de equipamentos tais como perfuradora, influem diretamente no tempo perdido pelo programador à espera dos testes ou em filas para utilização de equipamento.

Relacionamento Pessoal: O tamanho da equipe influí negativamente na produtividade, porque naturalmente

ocorre conversas paralelas ao desenvolvimento das atividades, e como este serviço requer o máximo de concentração, o rendimento é abaixo do normal (1).

Uma das formas de se contornar o problema é dividindo a equipe em vários ambientes confortáveis e silenciosos - (2). Ao considerarmos os Centros de Processamento de Dados de pequeno e médio porte cuja principal função é atender um determinado grupo, não terá recursos suficientes para atender a estas exigências, logo podemos considerar esta variável, simplificando nosso modelo e adaptando convenientemente a produtividade como veremos mais adiante.

(1) The Mythical Man - Month: Essays on Software Engineering, BROOKS, Fred.

(2) Como estimar prazos certos em programação, FELDMANN, Paulo Roberto.

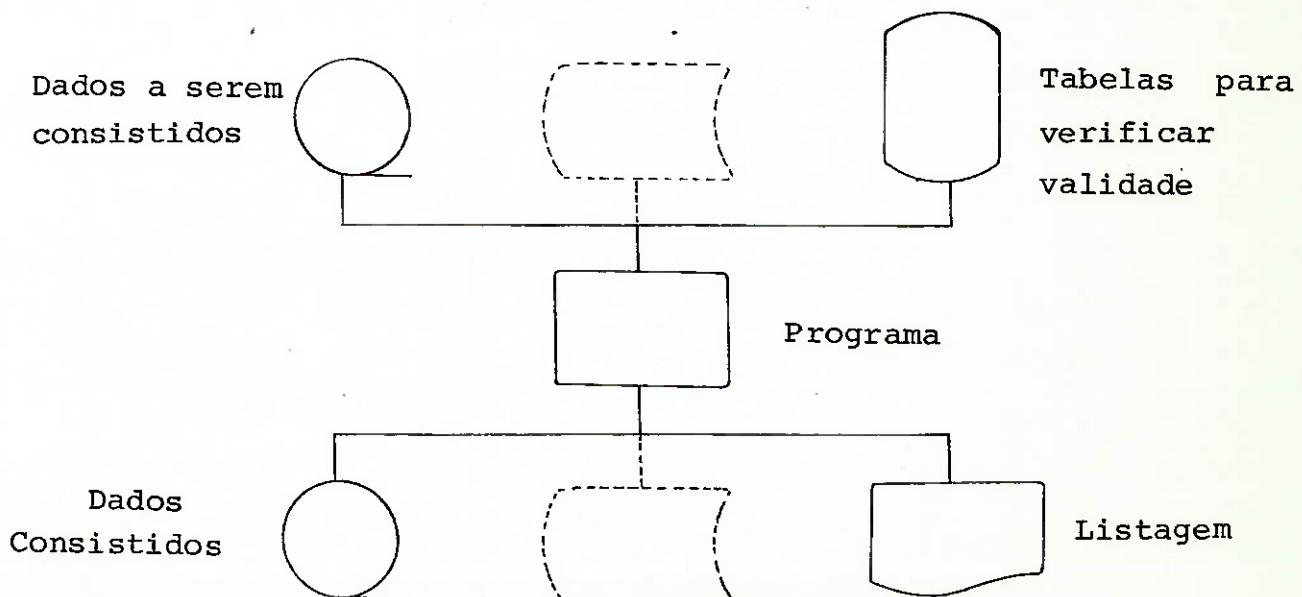
#### 2.3.3- Tipos de Programas

Os sistemas podem conter basicamente seis tipos de programas. Consistência, Formatação, Organização, Cálculo, Atualização e Listagem. Embora esta divisão já esteja consagrada em processamento de dados, muitas vezes é difícil enquadrar um programa em uma destas categorias, e os limites entre eles não são muito precisos. Daí a necessidade de se padronizar estes tipos de programas para reduzir a subjetividade na categorização.

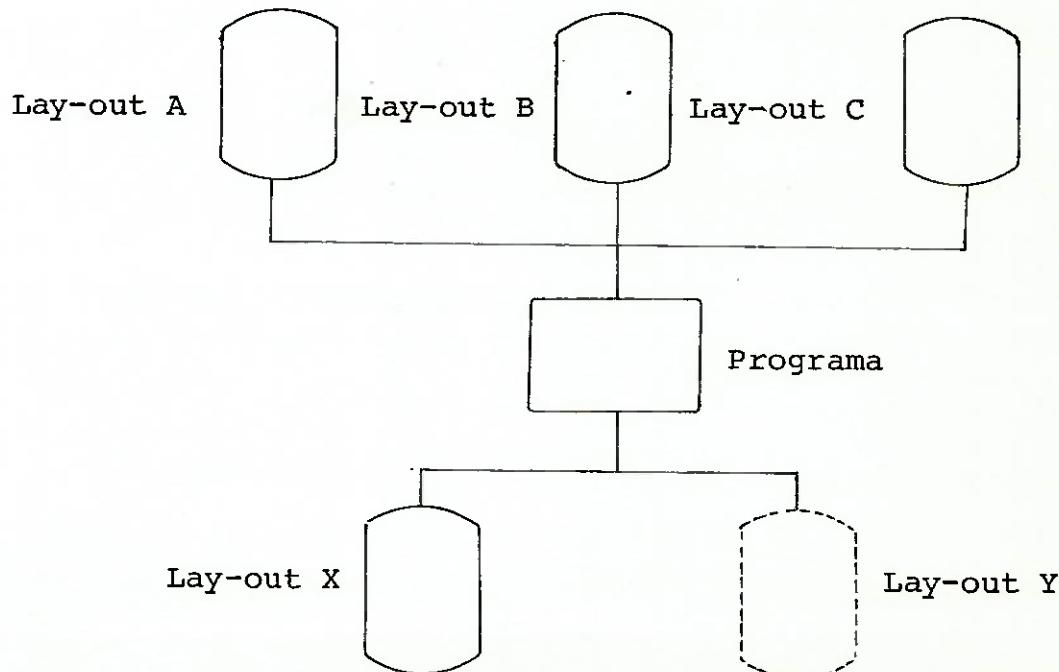
Consistência: Encontram-se qualmente na entrada do

sistema, e são responsáveis pela validade no sistema. Esta validade de campos significa verificar se estão corretamente preenchidos (uma consistência clássica é verificar se os campos são mínimos ou alfanuméricos conforme o conteúdo pré-estabelecido do campo). Fechamento de lotes é outro tipo, verificar dígito de controle, pesquisa em tabelas, etc. Embora a dificuldade destes programas seja variável, existem poucos "tipos de consistência" - cujas rotinas são fixas facilitando muito o trabalho de elaboração da lógica do programa. Isto é o que influui na eficiência do programa "é o número de arquivos de entrada e de saída" (1). Dentro desses arquivos temos que considerar os vários lay-outs de registros que devem ser consistidos em diferentes subrotinas. Fica evidente que quanto maior o número de layouts, maior o número de subrotinas e portanto o tempo de confecção.

As manutenções deste tipo de programa são bastante simples, qualmente modifica-se ligeiramente uma subrotina para aceitar ou rejeitar um tipo de Registro. Outras vezes modifica-se o registro de entrada ou de saída para modificar, incluir ou retirar um campo.



Formatação: Há poucos programas formadores em um sistema. Os mais comuns são aqueles encontrados nas interfaces de sistemas diferentes. Um exemplo típico é um sistema para cálculo do custo de um departamento ou de uma empresa, que necessariamente uma das entradas é o cadastro da folha de pagamento; para que este arquivo possa ser utilizado dentro do sistema, seu lay out deve ser modificado, combinando com outros arquivos saindo um arquivo em um formato único. Muitas vezes estes programas aparecem no meio sistema para satisfazer uma condição no processamento, tais como garantizar o registro para que possa ser lido no programa seguinte sendo os exemplos mais comuns as expansões e compactações de registro. No caso de empresas prestadoras de serviço, ao atender a solicitação de algum cliente, é necessário criar um sub-sistema dentro do sistema aparecendo uma interface. Podem também aparecer depois dos programas de consistência. Neste tipo apresentam-se os maiores problemas porque é comum a formatação estar incluída no programa de consistência apresentando dificuldades na sua estimativa.



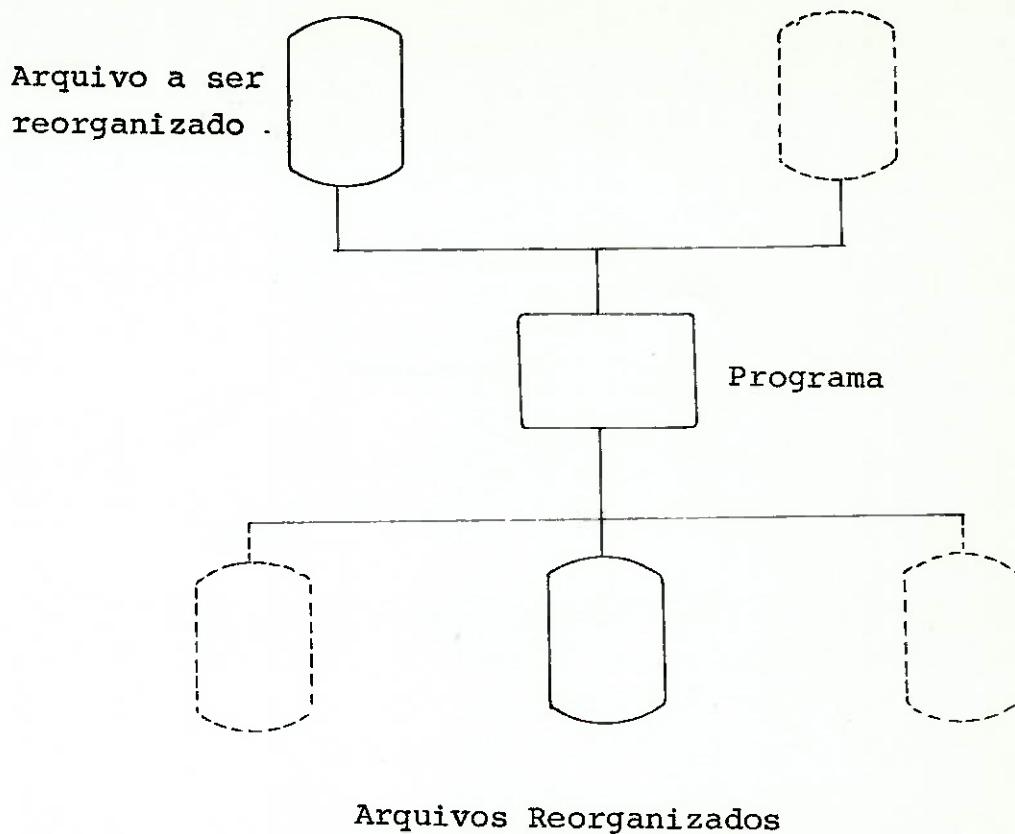
Organização: Os programas organizadores na verdade apresentam três tipos: os organizadores propriamente ditos, compactadores e separadores de arquivos. O primeiro tipo geralmente usa um programa utilitário, tipo Sort ou Merge, sendo estes de fácil confecção, sendo seu tempo relativamente curto. Outras vezes será preciso conectar o programa dada a impossibilidade de se usar um produto pronto, mas não traz grandes problemas para o programador porque acontece o mesmo que nos programas de consistência ou seja do programa não varia muito.

Os programas separadores localizam imediatamente antes dos atualizadores e calculadores porque são responsáveis em enviar registros específicos para cada programa. Também são de fácil confecção, predominando os comandos "IF's".

Caracteriza-se por possuir um arquivo de entrada, uma tabela de consulta que pode ser uma entrada em disco, e vários arquivos de saída.

Muitas vezes estes arquivos apresentam o mesmo lay-out de saída, mas com códigos de movimento diferentes.

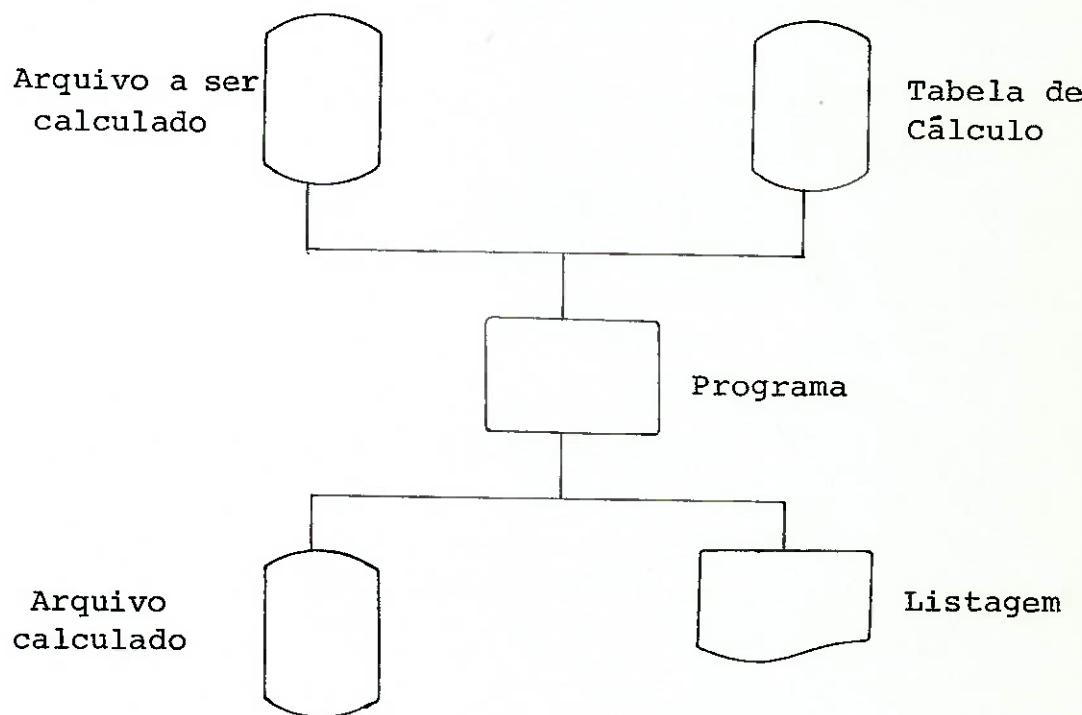
Os compactadores na verdade, são apenas uma subrotina a mais de acumulação para registro de mesmo código. Assim poderemos notar que o grau de dificuldade do programa varia muito pouco dependendo apenas do número de registros e arquivos manipulados.



As manutenções neste tipo de programa acontecem principalmente na introdução ou retirada de um código de registro ou modificação de lay-out's. Estes programas são de fácil manutenção porque as modificações estão bem localizadas.

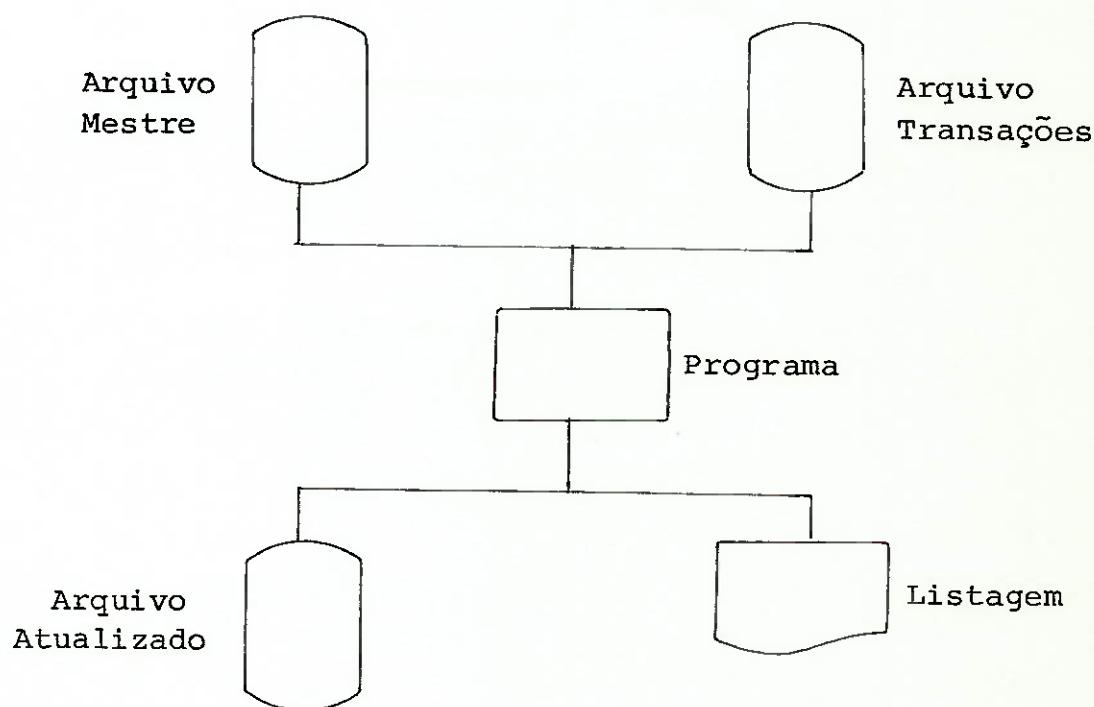
Cálculo: Este talvez seja o tipo de programa mais difícil de ser categorizado. Em primeiro lugar pela multiplicidade de funções que pode desempenhar; os cálculos podem ter um grau de complexidade muito grande, com consultas à tabelas em memória ou disco, cálculo de índices para análise da gerência, manipulando um grande número de dados, ou pode fazer cálculos simples de parcelas de pagamento. Outra dificuldade é encontrar programas calculadores "puros". Geralmente é uma enorme subrotina de programas atualizados, o que dificulta muito sua categorização; outras vezes o

encontramos gerando apenas relatórios de saída, o que poderia ser confundido com os listadores. Para solucionar este problema, podemos caracterizar este tipo de programa quando o cálculo é a principal função dentro do programa; ou seja, pela experiência dos chefes de secção entrevistados, notamos que esta categorização, embora subjetiva, é comum a vários; visto que os cálculos, principalmente se forem complexos, são logo enfatizados pelo analista.



Atualização: Estes programas geram relatórios, mas são visivelmente rotinas secundárias, ou seja, a impressão apenas tem a função de informar o que foi feito na atualização. Opera-se com 2 tipos de arquivos de entrada; o cadastro (ou mestre) a ser atualizado e os de transações que pode ser mais de um; na saída temos o cadastro atualizado, uma listagem quando houver

e outros arquivos com movimentos gerados a partir da atualização, para outros programas.



As manutenções nestes programas, é muito demorada, pela própria dificuldade em confeccioná-lo. O programador tem que recorrer aos diagramas de blocos para localizar a manutenção, sendo que se sua lógica for muito complicada é preferível refazer grande parte do programa. As vezes chegamos a casos extremos como é o caso de um programa, que no departamento é chamado de "coração do crediário". A maior dificuldade em sua manutenção é que está escrito em linguagem Assembler, mas é o principal programa do Sistema Crediário, do qual depende o maior volume de negócios das empresas LOJICRED. Se a manutenção for relativamente simples ,

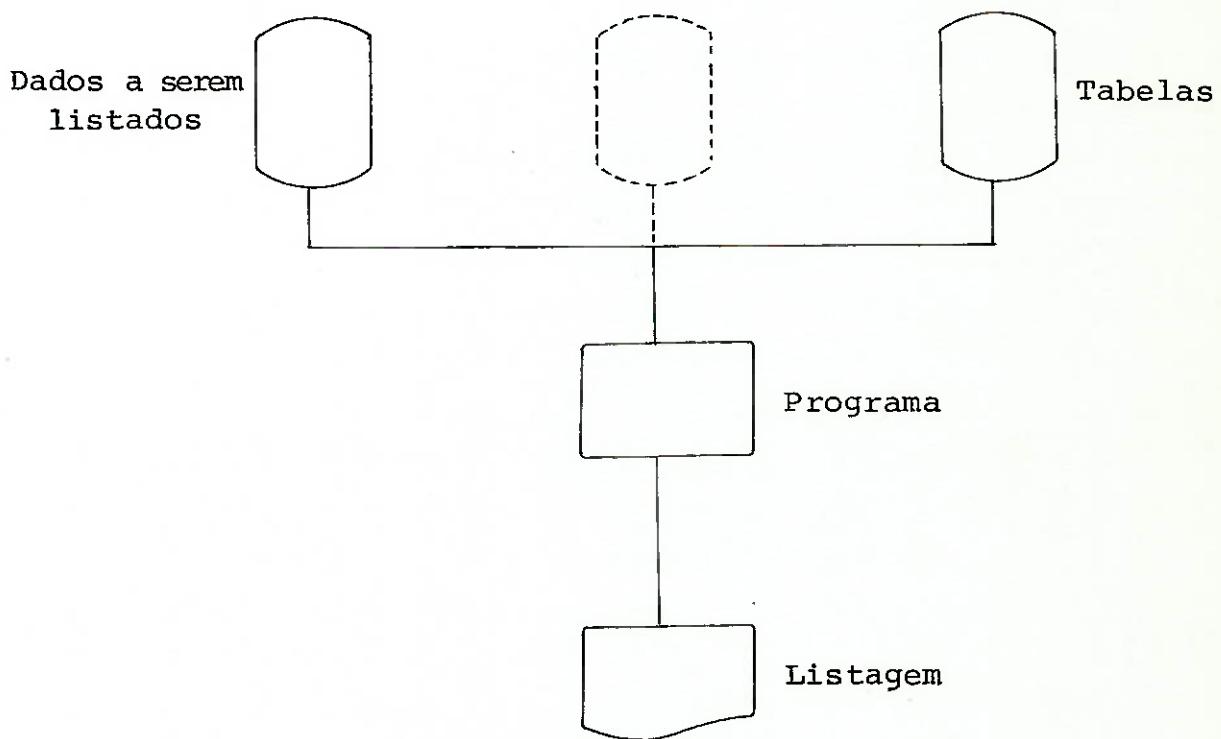
qualquer programador poderá fazê-la, mas se houver uma mudança na lógica do programa, sómente programadores capacitados e conucedores do sistema farão a alteração.

Listagem: Nos CPD's onde não é usada a programação estruturada, os programas listadores nunca desempenham a função única de imprimir. O que pudemos observar é uma tentativa de "economia" de programas, dividindo-o em 2 partes: cálculo/organização e listagem. Logicamente o grau de dificuldade de cada programa varia muito em função da 1a. parte e pouco em relação à segunda. Uma das formas de contornar o problema, é categorizá-lo por apresentar como saída, apenas relatórios e não arquivos. Mas mesmo assim a dificuldade continua, porque vários programas que tinham alguma outra função, atendiam a esta condição, e o tempo de confecção e manutenção era ainda muito maior do que outros, com as mesmas características de entrada e saída mas que apenas geravam relatórios e os cálculos eram feitos em outro programa.

A solução encontrada foi uma sub-divisão destes programas em listadores simples e listadores complexos. Isto permitiu que contornássemos a dificuldade mas ao mesmo tempo introduzimos uma variável subjetiva, que é difícil de padronizar.

Listadores simples: São programas cuja rotina principal é a de impressão, cujos dados de entrada já estão organizados. Pode vir alguma outra subrotina simples ou que use um pacote "general purpose".

Listadores complexos: A rotina principal é de impressão, mas apresenta outras subrotinas para organizar os arquivos de entrada (tais como acumulação de parciais) ou cálculos que não envolvam mais de uma tabela.



#### 2.3.4- Variáveis Objetivas

Ao apresentarmos os tipos de programas, enfatizamos o problema de que quanto maior o número de arquivos manipulados no programa, maior a complexidade em confeccioná-lo, e para sua manutenção, requer-se um tipo de estudo maior devido à sua lógica interna. Brandon (1) e Szweda (2) concordam neste ponto, pois "mais complexo será o balanceamento dos arquivos" (2) Feldmann (3), acrescenta que também depende do número de lay-outs diferentes nos arquivos, porque cada

lay-out diferente deverá gerar uma subrotina para processá-lo. Para que este método pudesse ser aplicado também a programas que apresentassem um meio termo entre programação estruturada e programação "faz tudo", foi necessário adicionar variáveis que aperfeiçoasse o modelo.

A justificativa de cada variável utilizada é dada a seguir.

Arquivos de entrada: A única variável que interessa ao falarmos em arquivos é seu número, não importando se estão gravados em fita ou em disco, concordando com Feldmann (3), que despreza esta diferenciação. Na maioria dos métodos analisados, dá-se muita importância a este fato, inclusive diferenciando se está organizado de modo sequencial, index-sequencial ou diretamente (1).

Arquivos de saída: Os mesmos comentários do ítem anterior, apenas ressaltando que os relatórios foram considerados à parte por tratar-se de rotinas específicas e um tratamento nos registros diferente e a complexidade desta subrotina convém que esteja desligada (independente) das sub rotinas de gravação de arquivos.

- (1) Project Control Standards, Brandon, Dick
- (2) Information Processing Management, Szweda, Ralph
- (3) Como estimar prazos certos em programação, Feldmann

Relatórios: Trata-se do número de lay-outs diferentes de relatório, muitas vezes variando apenas a organização, a existência e o número de quebras. Qualquer uma das três variáveis pode ser obtida pela folha de operações do programa.

Arquivo de Guia: A existência ou não deste arquivo pode influir em certos tipos de programas como veremos a seguir. É muito comum encontrarmos em firmas prestadoras de serviço, dado que o programa é utilizado para vários clientes, sendo apenas o nome da firma e outros dados sejam alterados. Outras vezes pode vir com uma tabela de consulta a que tem que acessar frequentemente. Neste caso a correlação é evidente. Esta variável foi somada aos arquivos de entrada, sendo utilizada ou não conforme o tipo de programa.

Registros de entrada: Falamos aqui de nº de lay-out's diferentes de entrada, porque é evidente que para lay-out's diferentes, haverá tratamento diferente cuja complexidade varia diretamente com o número destes. Se o mesmo lay-out estiver contido em 2 arquivos de entrada diferentes, será contado como único. Do mesmo modo, se há vários registros de entrada, mas o programa despreza alguns e processa outros o número será os registros de entrada processados (1).

Registros de saída: Os registros contidos em arquivos e saída em cartão, disco ou fita, seguem o mesmo tratamento dos de entrada. Muitas vezes (como no caso de atualização) o registro de saída tem o mesmo lay-out de de entrada, mas será contado nas duas variáveis.

- (1) Como estimar prazos certos em programação,  
Feldmann, Paulo Roberto.

Linhas em Relatório: Aproveitando o fato da maioria dos programas gerarem relatórios uma maneira de medir sua complexidade é pelo número de lay-outs de linha que imprime. Este critério deve ser bem analisado para que se possa estimá-lo com precisão.

Relatórios com formato padrão:- Os lay-outs de linha computados são: Cabeçalho específico do relatório (exclui-se o que contém o nome da firma, nº da página, data, pois são impressos com uma subrotina já desenvolvida e comum a todos os programas). Os cabeçalhos do tipo:

Linha 1	Pagamentos	Pagamentos	Pagamentos
Linha 2	Atrasados	Atrasados	Atrasados
Linha 3	Mês 1	Mês 2	Mês 3

serão contados como apenas 1 linha.

Linha de Totais: Também será computado como um registro único, mesmo que por razões de espaço tenha que ser colocada em 2 linhas:

Totais	XX.XXX	XX.XXX	XX.XXX
	XX.XXX	XX.XXX	XX.XX

Totais de quebra serão computados separadamente.

Linha 1	TOTAL LOJA	XXX.XXX
Linha 2	TOTAL CIDADE	XXX.XXX
Linha 3	TOTAL ESTADO	XXX.XXX

Nº de linhas de totais = 3.

Rodapés: Não serão computados pelo mesmo motivo do cabeçalho de firmas.

Linhas de Relatório: propriamente dito. O critério é que se forem produzidas dentro de uma mesma subrotina, e por tanto o mesmo endereçamento para leitura e impressão, serão tratados como registro único - geralmente coincidirá com o mesmo lay-out de saída, podendo adotar este critério. A dificuldade surge quando há mais de uma linha para cada registro de entrada.

Linha 1 Chapa XXXX FGTS a recolher XXXX Dias Trab. XXX  
Linha 2 Parcelas XX.XX XX.XX XX.XX XX.XX XX.XX

No exemplo a linha 1, pode ser obtida através do cadastro de funcionários, enquanto que a linha 2 vem de uma rotina de cálculo, ou seja 2 procedimentos distintos; neste caso o número de linhas computadas será 2.

Nos programas que geram vários relatórios em que apenas diferenciam-se pelo nível de quebra as linhas comuns não contadas como uma única, exceção feita aos totais que serão computados separadamente.

Formulários Especiais: O mesmo critério se aplica neste tipo de relatório, porque ocorre que dados obtidos em um único endereçamento no cadastro, estejam dispostos em uma região do formulário, como é o exemplo de emissão de envelopes.

Registros do tipo:

PEDRO FOGAGNOLLI  
Al. Santos, 272, Ap. 206  
São Paulo  
01108  
CC. 10814227/08

Serão computados como um único registro.

Para as linhas numéricas diferentes, como é o caso de envelopes de pagamento, cada cálculo de desconto é feito em rotina diferente: IAPAS, Reuniões, Sindicatos, sendo computados separadamente, se não pudessem ser enquadrados numa mesma categoria de cálculo por exemplo:

Entrada	XX.XXX
Parcela 1	XX.XXX
Parcela 2	XX.XXX

Nestes 3 cálculos, muda-se apenas a taxa pelo que é multiplicada o valor total, podendo considerar como 1 registro apenas.

Linhos de Totais: Mesmos comentários feitos para os formulários padrões.

Chaves UPSIS: As chaves UPSIS são usadas quando queremos ter para um mesmo programa uma ou mais opções de entrada ou saída. Opções de entrada, seria, por exemplo, o processamento de um código de movimento numa subrotina específica, com a propriedade de que esta subrotina possa estar ligada ou desligada conforme'

o desejo do usuário, que escolherá a situação perfurando 1 ou 0 no guia correspondente. Se a opção escolhida for "ligada" o registro entrará para ser processado, caso contrário será desprezado. Do mesmo modo que exemplificamos com tipos de registro, as chaves UPSIS permitem escolher o nº de arquivos de entrada, ou a meio em que estão acondicionadas, ou códigos de movimentos, etc.

Opções de saída são frequentemente encontradas para comandar o número de arquivos que devem ser gravados de acordo com os códigos de movimento. Por exemplo:

UPSI ligada

ARQ 1 Contem mov 1, mov 2, mov 3

UPSI desligada

ARQ 1 Contem mov 1

ARQ 2 Contem mov 2

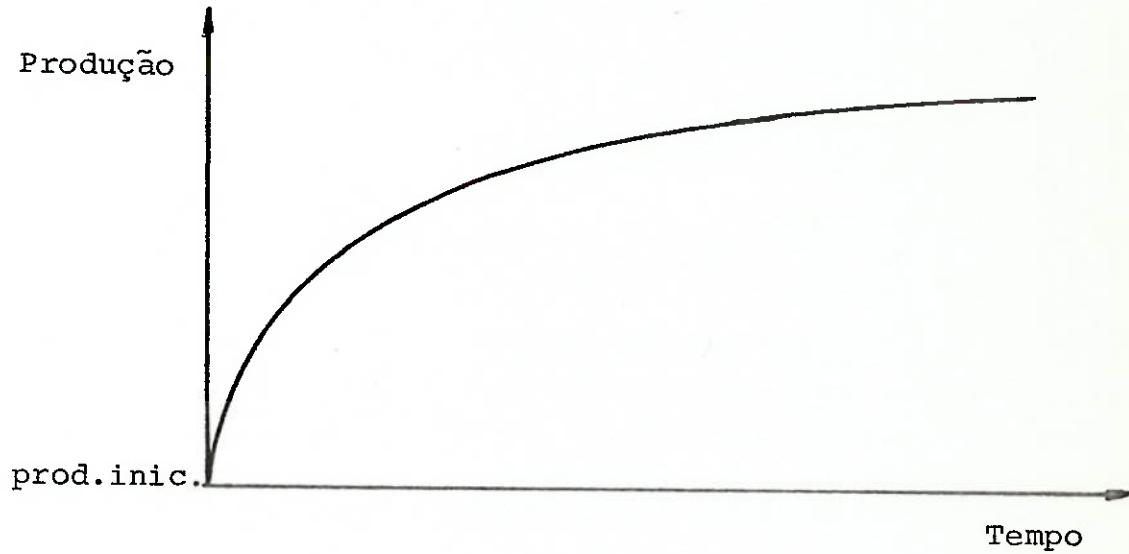
ARQ 3 Contem mov 3

Opções de processamento ocorrem quando se quer variar parâmetros dentro do programa, independentes de entrada ou saída. Por exemplo: Cálculos com ou sem considerar IOF. A correlação positiva entre a complexidade e o número de chaves UPSIS num programa é evidente, porém corre o risco de ser redundante, ou seja, se o número de relatórios de saída estiver dependendo da chave UPSI, estaremos contando 2 vezes a mesma variável.

Códigos de Movimento: Aqui analisaremos apenas os códigos de movimentos processados, não contando os desprezados

pelo processamento. Haverá subrotinas específicas para manipulá-los do mesmo modo que nos tipos de lay-out's de registros, obtendo mera correlação positiva entre esta variável e o tempo de confecção e manutenção de programas.

Rítmo do Programador: Os anos de experiência em programação e a qualidade pessoal do programador influem no tempo de confecção e manutenção de um programa. Em tarefas fabris. é comum ouvir-se falar em curva de aprendizado, na qual correlaciona-se a produção obtida pelo funcionário com o tempo, obtendo-se uma equação do tipo:



Este é um parâmetro muito importante de ser obtida, pois com ele teremos o tempo normal que o programa deveria ser feito. Há que explicar alguns conceitos introduzidos aqui.

Uma caracterização de programadores que encontramos nos CPD'S

é sua divisão entre trainee, junior, pleno e senior, de acordo com o tempo de experiência em programação.

- . Trainee - curso recem concluido: é admitido na empresa por seis meses; para experiência.
- . Junior - ao ser efetivado, depois do tempo de experiência, passa a categoria Junior até completar 2 a 2 anos e meio em programação.
- . Pleno - nesta categoria enquadraram-se os programadores de 2 anos e meio a 4 anos de experiência.
- . Senior - Experiência superior a 4 anos.

Além do tempo de experiência, uma variável que influí muito é a capacidade pessoal do programador, que não deixa de ser uma estimativa subjetiva feita pelo chefe do departamento de Programação. Para simplificar dividimos em três categorias.

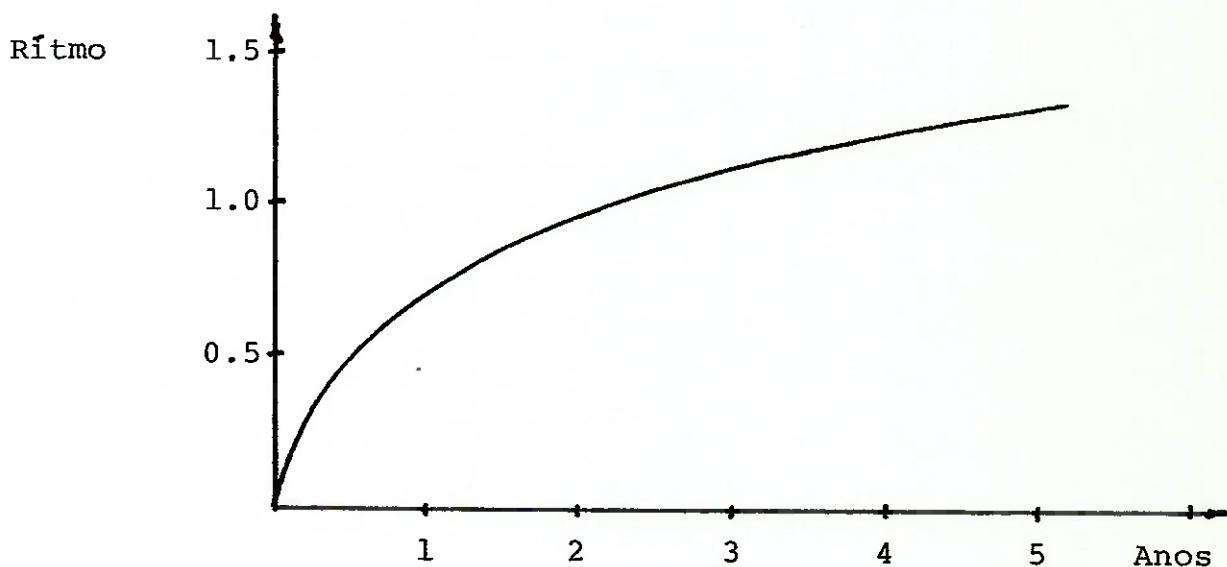
Bom , Regular , Ruim

Dada a categoria em que pode estar enquadrado um programador, temos que estabelecer um Padrão para a estimativa de tempos, pois é evidente que um mesmo programa será feito mais rapidamente por um senior regular do que por um junior regular, de modo que esta diferença de velocidades deve ser medida objetivamente. O programador "padrão" escolhido foi o pleno regular, cuja produção num dia normal de trabalho será confrontado com os demais ou seja, se um programador pleno regular desenvolveu certo programa em X horas, quantas horas levariam um programador júnior e um senior para desenvolverem o mesmo programa? O coeficiente que nos dará esta correção de tempo é chamado de ritmo do programador, que é função da sua experiência e qualificação individual Szweda (1) Feldmann (2),

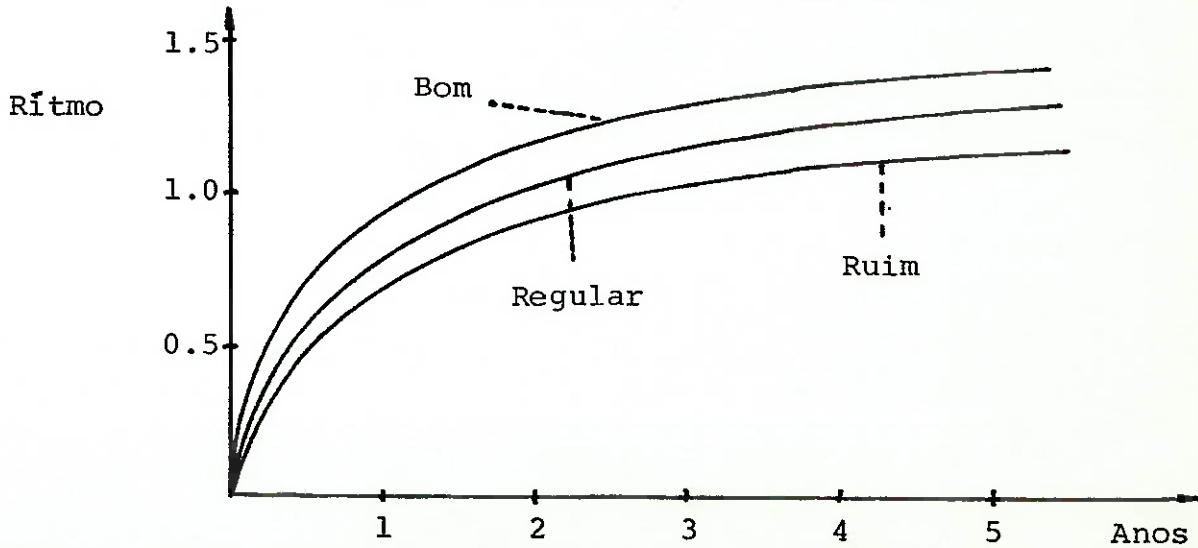
abordaram este tema, porém o primeiro usa critérios subjetivos para avaliação contando muito com a experiência do chefe do departamento; Feldmann chega a ter mais objetividade no seu método de avaliação. Com experimentos feitos dando o mesmo programa para ser desenvolvido por programadores das três categorias, junior, pleno e senior e admitindo o ritmo do programador pleno igual a 1, chegou a seguinte conclusão:

- júnior      0,4 a 0,8
- Senior      1,1 a 1,5

Aproveitando-se destes dados podemos construir um gráfico relacionando o tempo de experiência com o fator de ritmo.



Associando a este gráfico, a qualidade individual do programador, podemos construir três curvas com as quais enquadraremos nossos programadores



As curvas obtidas estão sujeitas a muitos erros pois não houve tempo nem oportunidade de chegarmos a uma regressão razoável por falta de dados, apenas analisamos programas parecidos feitos por programadores de mesmo tempo e experiência; porém a idéia foi objetivarmos mais as estimativas do ritmo do programador.

#### 2.3.5- Variáveis Subjetivas

A tarefa de manutenção ocupa 60% do tempo produtivo de um CPD, sendo dividida em 2 classes:

##### Manutenção Corretiva:

- quando ocorrem situações em que o programa não previu ao ser feito, quer por erro do programador, quer por erro da análise, acarretando o aborto do processamento. Cabe ao programador modificar algumas subrotinas para que possa aceitar esta nova situação.

##### Manutenção Adaptativa:

- Nos CPD's bem organizados este tipo representa aproximadamente 70% do total das manutenções. É solicitada quando o cálculo de um imposto muda de estrutura, ou para acrescentar linhas no relatório a pedido de um cliente, etc. Na Lojicred Processamento de Dados, o DAC (Departamento de Assistência a Clientes) é responsável pelo atendimento ao cliente e requisição da alteração ao DAP (Departamento de Análise e Programação).

Tanto num como no outro tipo de manutenção, o tempo varia desde alguns minutos até muitos dias de trabalho, quando é preciso refazer grande parte do programa.

Dada a dificuldade de obtermos variáveis independentes para correlacionar com o tempo de manutenção, optou-se pela categorização destes e trabalhar com médias históricas corrigidas pelo fator de ritmo do programador e pela produtividade obtida por amostragem de trabalho.

Definimos seis graus de dificuldade de manutenção:

Classe 1 ou Facílima: Envolve alteração de alguns cartões de programa, levando mais tempo no estudo e localização da manutenção.

Classe 2 ou Fácil: Não envolve alteração na lógica do programa, apenas alteração de alguma subrotina ou acrescentar algum procedimento. A manutenção é bem localizada e o tempo de estudo é curto.

Classe 3 ou Média: Envolve alteração na lógica de alguma subrotina, necessitando que se faça um diagrama de blocos antes da codificação. Geralmente trata-se de uma manutenção adaptativa para um novo cliente. Tempo de estudo da alteração é pequeno e a lógica do programa é clara.

Classe 4 ou Dificultada: A alteração tem o mesmo grau de dificuldade que o anterior, apenas diferenciando pela lógica do programa que torna difícil a localização da manutenção.

Classe 5 ou Difícil: Envolve alteração total de subrotina e modificação de outras, inclusão de UPSIS, são tipicamente manutenções adaptativas.

Classe 6 ou Reprogramação: São raros em programação e por falta de dados não foram estudados. Quando a alteração é profunda na lógica das subrotinas e estas por sua vez apresentam lógicas complexas, o analista opta pela confecção de novo programa. Podem originar-se por um erro da Análise na definição do programa ou

por falta de claridade na solicitação do cliente.

Além do nível de dificuldade, uma variável que influencia a localização da manutenção dentro do programa. Pode-se estar na subrotina de:

- Leitura
- Consistência
- Organização
- Atualização
- Cálculo
- Gravação
- Impressão
- Procedimentos Diversos.

Uma solicitação frequente nos Departamentos de Análise e Programação são pequenos programas que resolvam uma necessidade imediata de um cliente, corrijam campos em cadastro, preparem dados para entrar em um programa, etc., que podem ou não serem incluídos no sistema. Exemplos: listagem dos clientes com mais de 50 anos, total de depósitos em bancos da Zona Norte, Cálculo do aumento de salário, etc. Como o tempo de confecção é menor do que os programas normais, foi incluída como manutenção embora se trate de um programa novo. Na tabela é denominada de

• Programas Paralelos

Para efeito de cruzamento, foram usadas as classes de 1 a 4, na ordem crescente de dificuldade. A tabela fica com o seguinte aspecto.

DIFÍCULDADE SUBROTINA		1	2	3	4	5
LEITURA	—X—					
	S					
CONSISTÊNCIA	—X—					
	S					
ORGANIZAÇÃO	—X—					
	S					
ATUALIZAÇÃO	—X—					
	S					
CÁLCULO	—X—					
	S					
GRAVAÇÃO	—X—					
	S					
IMPRESSÃO	—X—					
	S					
PROCESSAMENTOS DIVERSOS	—X—					
	S					
PROGRAMAS PARALELOS	—X—					
	S					

As manutenções estudadas não abrangem todas as variações que podem ser solicitadas ao DAP, mas sua maioria. Na tentativa de resolver o problema quando surgir uma dúvida na classificação, foi criado um item de Processamentos Diversos para orientar o chefe de Programação na falta de uma estimativa melhor.

### 3 - Fundamentos Teóricos

No capítulo anterior vimos como cada uma das variáveis influencia va no tempo de manutenção e confecção de programas. Agora o que buscamos é um meio de saber quais são as variáveis que realmente influenciam em cada tipo de programa e em que proporção. As variáveis independentes utilizadas são todas objetivamente mensuradas' ou seja para cada caso pode-se determinar exatamente o valor da variável, salvo os casos de manutenções curtas que são estudadas' mais adiante. Deste modo poderemos determinar um hiper-plano a n dimensões que melhor explica os pontos dados.

Para isto devemos apresentar dois conceitos

#### 3.1- Regressão Linear Múltipla

Pelo conjunto dos pontos dados das n variáveis, podemos ajustar infinitos hiper-planos para tentar explicar como cada uma delas influenciam no tempo de confecção, mas apenas um hiper-plano será o que melhor resultado dará, ou seja, o erro que cometerá ao ajustar-se aos pontos dados será o menor possível.

Sejam os vários programas i, apresentados pelas variáveis  $x_{1i}, x_{2i}, \dots, x_{ni}$  e pelo tempo de confecção e manutenção corrigidos  $y_i$ . Ajustemos um hiper-plano.

$$\hat{y} = a + b_1x_1 + b_2x_2 + \dots + b_nx_n$$

O que procuramos é que  $(y_i - \hat{y}_i)$ , ou seja a diferença entre o valor estimado  $\hat{y}$ , e o valor real  $y_i$  seja o menor possível. Para que isto se verifique em todos os pontos de amostra, temos em primeiro lugar que escolher um estimador, que é a quantidade calculada em função dos elementos da amostra que será usado no processo de estimação do parâmetro desejado (1), que nos leva ao procedimento dos mínimos quadrados. No caso estudado temos que

$$\begin{aligned} & \min \sum_{i=1}^k (y_i - \hat{y}_i)^2 = \\ & = \min \sum_{i=1}^k (y_i - a - b_1x_{1i} - b_2x_{2i} - \dots - b_nx_{ni})^2 \end{aligned}$$

considerando k programas

A condição de mínimos será satisfeita se as derivadas parciais dos coeficientes  $a$ ,  $b_1$ ,  $b_2 \dots$ ,  $b_n$  forem igualadas a zero ou seja

$$\frac{\partial}{\partial a} \sum_{i=1}^k (y_i - a - b_1 x_{1i} - b_2 x_{2i} - \dots - b_n x_{ni})^2 = 0$$

$$\frac{\partial}{\partial b_1} \sum_{i=1}^k (y_i - a - b_1 x_{1i} - b_2 x_{2i} - \dots - b_n x_{ni})^2 = 0$$

$$\vdots$$

$$\frac{\partial}{\partial b_n} \sum_{i=1}^k (y_i - a - b_1 x_{1i} - b_2 x_{2i} - \dots - b_n x_{ni})^2 = 0$$

vem:

$$-2 \sum_{i=1}^k (y_i - a - b_1 x_{1i} - b_2 x_{2i} - \dots - b_n x_{ni}) = 0$$

$$-2 \sum_{i=1}^k (y_i - a - b_1 x_{1i} - b_2 x_{2i} - \dots - b_n x_{ni}) x_{1i} = 0$$

$$-2 \sum_{i=1}^k (y_i - a - b_1 x_{1i} - b_2 x_{2i} - \dots - b_n x_{ni}) x_{ni} = 0$$

(1) Estatística, Costa Neto, Pedro Luis de Oliveira; fornecendo o seguinte sistema a n equações a n incógnitas.

$$\sum_{i=1}^k x_{1i} y_i = a \sum_{i=1}^k x_{1i} + b_1 \sum_{i=1}^k x_{1i} x_{ni} + \dots + b_n \sum_{i=1}^k x_{1i} x_{ni}$$

$$\sum_{i=1}^k x_{2i} y_i = a \sum_{i=1}^k x_{2i} + b_1 \sum_{i=1}^k x_{1i} x_{2i} + \dots + b_n \sum_{i=1}^k x_{2i} x_{ni}$$

$$\sum_{i=1}^k x_{ni} y_i = a \sum_{i=1}^k x_{ni} + b_1 \sum_{i=1}^k x_{1i} x_{ni} + \dots + b_n \sum_{i=1}^k x_{ni}^2$$

No estudo, envolveu 8 incógnitas de modo que chegariam a 7 sistemas de equações 9 x 9, tornando-se impraticável fazê-los manualmente. Para tanto, foi usado o procedimento REGRESSION do SPSS (Statistical Package for the Social Sciences).

### 3.2- Correlação Linear Múltipla

A tendência de variação entre a variável dependente e as independentes, pode ser medido por um número  $R$ , definido por:

$$R = \sqrt{\frac{b_1 s_{iy} + b_2 s_{2y} + \dots + b_n s_{ny}}{s_{yy}}}$$

onde:  $s_{ty} = \sum_{i=1}^k (x_{ti} - \bar{x})(y_i - \bar{y})$

$$s_{yy} = \sum_{i=1}^k (y_i - \bar{y})^2$$

que varia entre 0 e 1, não considerando os valores negativos de  $R$ , pois é perfeitamente possível termos a variável dependente positivamente correlacionada com algumas das variáveis independentes, e negativamente correlacionada com outras. Da mesma forma,  $R^2$ , indica a parcela da variação total de  $y$ , explicada pelo hiper-plano da regressão (1), ou seja, o "resíduo" não explicado pela regressão pode ser medido através de

$$s_{mr}^2 = \frac{s_{yy} - \sum_{i=1}^n b_i s_{iy}}{k - n - 1}$$

### 3.3- Correlação Parcial

Ocorre algumas vezes, que duas variáveis que não apresentam relação entre si, apareçam numa matriz de correlação com um coeficiente bastante alto. Isto se dá pelo fato das duas, embora não influam uma na outra, guardam íntima relação com uma terceira variável comum a ambas. Com este conceito, podemos estudar qual a correlação entre as duas primeiras, deixando a terceira constante, ou quantas outras forem necessárias e assim, o coeficiente de correlação parcial entre  $x$ ,  $y$  com respeito a  $z$ , será denotada por  $r_{x,y,z}$ , definido por

$$r_{x,y,z} = \frac{r_{xy} - r_{xz} r_{yz}}{\sqrt{(1 - r_{xz}^2)(1 - r_{yz}^2)}}$$

### 3.4- Análise de Melhoria

No caso de regressão à duas variáveis, a análise de melhoria mostra se um polinômio de grau mais alto melhora significativamente a correlação. Para testar a melhoria no ajuste, monta-se o quadro de Análise de Variância. No quadro abaixo estuda-se o caso de parábola e reta.

(1) Estatística, Costa Neto, Pedro Luis de Oliveira

<u>Fonte de Variação</u>	<u>Soma de Quadrados</u>	<u>Graus de Liberdade</u>	<u>Quadrado Médio</u>	<u>F</u>	<u>F <math>\alpha</math></u>
Melhoria de ajuste	$\sum_{i=1}^k (\hat{y}_{pi} - \hat{y}_i)^2$	1	$\bar{Q}$	$F = \frac{\bar{Q}}{S_p^2}$	$F_{1, k-3}$ ,
Residual s/ a parábola	$\sum_{i=1}^k (y_i - \hat{y}_{pi})^2$	$k-3$	$S_p^2 = \frac{\bar{Q}}{n-3}$		
Residual sobre a reta	$S_{yy} - b^2 S_{xx}$	$k-2$			

Transcrito de ESTATÍSTICA, Costa Neto, Pedro Luis de Oliveira  
onde

$\hat{y}_{pi}$  = é a estimativa dada pela curva parabólica no ponto i

$\hat{y}_i$  = é a estimativa dada pela reta no ponto i

$\alpha$  = nível de significância do teste ou a probabilidade de decidirmos rejeitar uma proposição que na realidade seja verdadeira.

k = nº de programas

Para o caso de multicorrelação, analisamos o fato de que se a entrada de uma nova variável, melhora significativamente a correlação, cujo procedimento é análogo ao mostrado no quadro anterior, podendo as duas serem consideradas com um problema único, bastando considerar  $x_1 = x$ ,  $x_2 = x^2$ , etc.

### 3.5- Inclusão de variáveis pelo processo STEPWISE

No ítem anterior, dissemos que a inclusão de uma variável na regressão pode trazer ou não melhorias significativas ao

ajuste, porém surge a questão de qual a melhor ordem inclu -  
são para que isto se verifique. A resposta vem através de um  
procedimento chamado de "Stepewise" que inclui em primeiro  
lugar as variáveis independentes mais correlacionada com a  
dependente. O processo continua através de análise de corre -  
lações parciais entre a variável que irá entrar na equação e  
as demais já incluídas para modificar os coeficientes ou se  
alguma delas deva ser retirada da equação.

Isto permite que a equação de regressão não tenha variáveis  
que não contribuam significativamente para a melhoria de cor -  
elações. Este fato será importante ressaltar para que os re -  
sultados obtidos tenham significância estatística.

### 3.6- Intervalo de confiança

A equação do hiper-plano obtida é uma estimativa à qual está  
associado um erro na determinação de um ponto. Assim torna-se  
necessário que determinemos um hiper-espaço em torno deste  
hiper-plano no qual com certo nível de significância podemos  
dizer que  $\hat{y}$  está naquela região. A este hiper-espaço, denomi -  
namos intervalo de confiança de  $y$ , que será uma estimativa  
do coeficiente de segurança que usaremos em orçamento.

A amplitude do intervalo de confiança é dada pela expressão

$$A = t_{\alpha/2, k - (1+n)} \cdot \sqrt{C \cdot s_e^2}$$

onde

$$t_{\alpha/2, k - (1+n)} = \text{Distribuição de Student}$$

$$s_e^2 = \frac{\sum y^2 - b_1 \sum x_1 y - \dots - b_n \sum x_n y}{k - n - 1}$$

$$C = \frac{1}{n} + (c_{11} x_1^{*2} + c_{12} x_1^{*} x_2^{*} + \dots + c_{1n} x_1^{*} x_n^{*})$$

$$+ (c_{21} x_1^{*} x_2^{*} + c_{22} x_1^{*2} + \dots + c_{2n} x_2^{*} x_n^{*})$$

$$+ \dots$$

$$+ c_{n1} x_n^* x_1^* + c_{n2} x_2^* + \dots + c_{nn} x_n^{*2} )$$

$k = n^{\text{a}}$  de programas

$n = n^{\text{a}}$  de variáveis

Os coeficientes  $C_{ij}$  são obtidos de dois modos equivalentes:

$$\begin{bmatrix} c_{11} & c_{12} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \dots & c_{nn} \end{bmatrix} \begin{bmatrix} \sum x_1^2 & \sum x_1 x_2 & \dots & \sum x_1 x_n \\ \sum x_2 x_1 & \sum x_2^2 & \dots & \sum x_2 x_n \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \sum x_n x_1 & \sum x_n x_2 & \dots & \sum x_n^2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 10 \\ 0 & 0 & 0 & \dots & 01 \end{bmatrix}$$

ou

$$\begin{bmatrix} c_{11} & c_{12} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \dots & c_{nn} \end{bmatrix} \cdot \begin{bmatrix} \sum x_1 y \\ \sum x_2 y \\ \vdots \\ \sum x_n y \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

A amplitude variará ao variarmos  $x_1^*$ ,  $x_2^*$ , ...,  $x_n^*$ , podendo provar que atingirá o menor valor para  $x = (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$

#### 4 - Análise de Trabalhos Semelhantes

A preocupação de orçar bem os programas desenvolvidos e suas manutenções, surgiram quando os custos dos departamentos de Análise e Programação começaram a pesar muito nos orçamentos dos CPD's desviando a atenção dos diretores que antes centravam suas atenções no hardware. Vários foram os métodos propostos com esta finalidade e neste capítulo procura-se analisar os mais importantes.

##### 4.1- Critérios para estimativa de tempos

###### 4.1.1- Linhas de Codificação

Este método baseia-se na análise de dados históricos de programas desenvolvidos e em manutenção, cuja única variável é as linhas de codificação. No fundo, trata-se de uma estimativa para estimar uma estimativa, ou seja, ao receber a pasta do programa, o programador baseado na sua experiência e na dificuldade do problema, estima um número de linhas de codificação para cada sub-divisão do programa ou sub-rotina. Com a somatória "ponderada" ou seja, associando a cada sub-rotina um coeficiente de dificuldade, chega-se ao número de linhas de codificação que será contrastada com uma tabela de médias e obtido o tempo.

Temos nesta exposição três pontos de crítica:

- A estimativa, ainda que grosseira do nº de linhas de codificação para cada sub-rotina, deve ser feita após o estudo de programa, que em muitos casos é significativo. Contorna-se este ponto fazendo com que o próprio analista estime o tempo.
- Todas as estimativas são subjetivas, sem padrões que possam guiar o analista ou programador. Alguns que utilizam este método, constroem tabelas relacionando função de sub-rotina com nº de linhas de codificações, mas nunca são confiáveis pois trata-se de uma tabela específica para o CPD.
- O que é linha de codificação? Uma má definição desta variável pode fazer com que as estimativas sejam completamente equivocadas. Para fugir à estas medidas de produtividade, o programador dispõe de diversos recursos, tais como cartões de comentários

em COBOL, definição de campos de trabalho adicionais e que não são utilizados, IF's equivalentes, etc.

#### 4.1.2- Pesos para funções

Esta é a técnica mais utilizada para estimativa de tempos em programação. Este método foi introduzido por Szweda (1) porém cai em subjetivismo ou seja deve contar com a experiência do programador para avaliar o grau de dificuldade do programa.

Em geral tem quatro fases de avaliação

- 1- Características de Input/Output
- 2- Principais funções processadas
- 3- Competência do Programador
- 4- Conhecimento da Tarefa

Para cada fase da avaliação, tem-se várias atividades ou processamentos que o programa exige às quais associa-se um peso dependendo de sua dificuldade de confeccioná-lo (por confecção entende-se o conjunto de atividades de estudo, codificação, depuração, teste e documentação).

Como exemplo de subdivisão de características de Input/output estão:

- nº de arquivos de entrada com até 5 lay-out's de registros diferentes.
- meio em que estão acondicionados
  - disco
  - fita
- modo de acesso
  - sequencial
  - direto
  - index-sequential

(1) Information Processing Management, Szweda, Ralph.

Este método, devido à sua facilidade de aplicação obteve tanta aceitação nos CPD's, porém encontra-se alguns pontos falhos.

- como as estimativas são subjetivas está sujeita a

erros de avaliação e consequentemente a introduzir erros na estimativa de tempo.

- Os coeficientes são obtidos por aproximações sucedivas.
- Quanto maior o número de coeficientes, mais difícil é seu balanceamento. Há trabalhos em que o número de coeficientes é de várias dezenas e devido à dificuldade de obtê-los por aproximações sucessivas, sua confiabilidade diminui a medida que seu número aumenta.
- Os conceitos de simples, médio, complexo e muito complexo é muito variável de programador para programador e não há meios de padronizá-los satisfatoriamente, devido a variedade de programas, tornando a faixa de avaliação muito ampla.
- Nenhum dos artigos pesquisados trazem o modo pelo qual foi obtido o coeficiente de competência do programador.
- Alguns trabalhos, no objetivo de diminuir o nº de pesos, agrupam os itens das fases (Input/Output, - etc) em grandes grupos, perdendo a sensibilidade aos vários tipos de programas, ou seja, um registro vai receber tratamento diferente em um programa calculador do que em um programa listador. Enquanto que um lê e imprime, o outro o coloca numa rotina de cálculo podendo ser bastante complexa. Vê-se que o peso para o mesmo registro em um e outro caso deve ser diferente.

Ao longo destes dez anos, este método sofreu várias modificações, entre as quais uma que foi aproveitada no novo método.

- Dividindo-se os programas em vários tipos consegu-se que os pesos sejam específicos para cada categoria de processamento.
- Introduzir padrões de complexidade para guiar as avaliações

Exemplo de estimativa por peso:

Fase / Atividade	Peso
Disco imput com registro em formato simples	1
Fita imput com registros de três tipos	3
Relatório	2
<b>Sub - Total</b>	<b>6</b>
Organização de arquivos simples	4
Atualização de cadastro complexa	6
Cálculos médios	2
<b>Sub - Total</b>	<b>12</b>
<b>Total</b>	<b>18</b>

Competência do programador	1,00
Conhecimento da tarefa	1,25
<b>Total</b>	<b>2,25</b>

Estimativa do tempo  $18 \times 2,25 = 40,5$  homens/dia

#### 4.1.3- Curvas aproximadas

O método das curvas aproximadas é uma variação do método de atribuição de pontos e pesos. Como nos demais métodos, suas estimativas são subjetivas embora já exista um avanço no sentido de objetivá-las.

A determinação do peso do programa segue as mesmas regras do método anterior. Por exemplo:

Coeficiente de atualização

. não tem cálculo                    CA=0

- . cálculo simples CA=1
- . cálculo médio CA=2
- . cálculo complexo CA=3

Os pontos serão dados pela somatória

$$\text{Pontos} = 3 \sum_{TE=1}^4 (\text{TE} \times \text{nº Arq entrada}) + \\ 2 \sum_{TS=1}^4 (\text{TS} \times \text{nº Arq saída} + 3(\text{TR.nº relatório} + \\ + 5 (\text{CP} + \text{CA} + \text{AT} + \text{CC} + \text{CQ}$$

onde

- . TE = nº arquivos de entrada
  - . 0 a 2 registros diferentes ou variáveis TE = 1
  - . 3 a 5 registros diferentes ou variáveis TE = 2
  - . 6 a 11 registros diferentes ou variáveis TE = 2
  - . + de 11 registros diferentes ou variáveis TE = 4
- . TS = nº arquivo de saída
- . TR = tipo de relatório
- . CP = coeficiente de processamento
- . etc

A partir do peso, enquadra-se em uma das seguintes categorias:

Programa Simples	Programa Médio	Programa Complexo	Programas Especiais
Peso 10, 20, 30	Peso 40, 50, 60	Peso 70, 80, 90	Peso 100

Para diminuir a subjetividade das avaliações, determina padrões para cada um dos coeficientes. Introduz uma divisão em várias fases da programação:

- F1 - Estudos e diagrama de blocos
- F2 - Codificação

F3 - Montagem, arquivo de teste e pré-cálculo

F4 - Testes e documentações

Mas nas suas estimativas de tempo, torna uma hipótese básica: Podemos exprimir qualquer capacidade em função da capacidade básica (regra de três simples). Por capacidade no caso, entende-se o nível individual do programador, e guarda entre as várias categorias uma relação linear. Esta hipótese não é aceita conforme discutimos no capítulo 2.

Este método introduz inovações em relação ao de atribuição de pontos que são usadas neste trabalho, tais como a padronização das avaliações subjetivas e a divisão das fases da programação, porém assume hipóteses que carecem de justificativa para sua aplicação tais como as "regras de três simples".

Uma equação típica apresentada é

$$PR = \frac{P}{W} \cdot \frac{23 - 3NB}{23 - 3N}$$

onde

PR = Prazo

P = Peso

W = Coeficiente de tempo

N = Nível

NB = Nível base para que prazo=peso/coef.  
tempo

Pelos parâmetros contidos na equação, podemos notar que são complexos de serem estimados e são resultado de outras equações aproximativas. Se este erro for acumulando-se a cada estimativa, o erro final será grande.

#### 4.2- Critérios de Produtividade

Medir de maneira segura a produtividade da Programação é um dos objetivos a que nos propuzemos alcançar com a implantação deste método. Existem basicamente dois índices para medir este valor, embora posteriormente se introduzisse várias alterações. São eles:

#### 4.2.1- Linhas de Codificação

O método de estimar o número de linhas de codificação a priori está sujeito a várias críticas e não deram bons resultados; no entanto medir a produtividade com este índice é um critério muito utilizado em CPD e vem sendo criticado e defendido em vários artigos de revistas especializadas.

É obtido pelo quociente entre

$$\frac{\text{Nº de linhas de Codificação}}{\text{Homens / Dia}}$$

Alguns pontos de crítica estão associados a este critério (1):

- A taxa é menor quanto mais complexo for o programa
  - O que é linha de codificação?
- (1) Como estimar prazos certos em programação,  
Feldmann, Paulo Roberto.
- Muitas atividades dos programadores não tem relação nenhuma com linhas de codificação.
  - O critério penaliza linguagens de alto nível cujos programas são normalmente grandes

#### 4.2.2- Tempo real x tempo estimado

Este critério não tinha boa aceitação pela dificuldade de se determinar com relativa segurança, o tempo estimado de confecção do programa. Quando o método dos pesos para funções começou a apresentar resultados razoáveis, este índice passou a ser utilizado.

É obtido pelo quociente

$$\frac{\text{Tempo Real}}{\text{Tempo Estimado}}$$

Com a utilização de avaliações subjetivas, para valores muito distantes de 1, torna-se difícil saber se a avaliação estava errada ou o programador trabalhou pouco ou bem demais, isto equivale a dizer que havia segurança nas estimativas.

O desenvolvimento deste método, as avaliações em desenvolvimento de programas são objetivas e de alta confiabilidade nos resultados. Para o caso de manutenções em que os critérios subjetivos são predominantes a faixa de variação para cada categoria de manutenção é pequena e amortiza o efeito. Para o caso de programação onde o tempo de programação é maior e sujeito a mais erros, usa-se os mesmos critérios para desenvolvimento.

## 5 - Sistema Atual de Controle de Atividades

### 5.1- Fluxo de Informações

Apresentamos o fluxo simplificado de informação, para uma solicitação de alteração ou desenvolvimento de um programa.

#### Legenda

CI      Comunicação Interna

CE      Comunicação Externa (Carta ou Pedido)

SOL     Solicitação

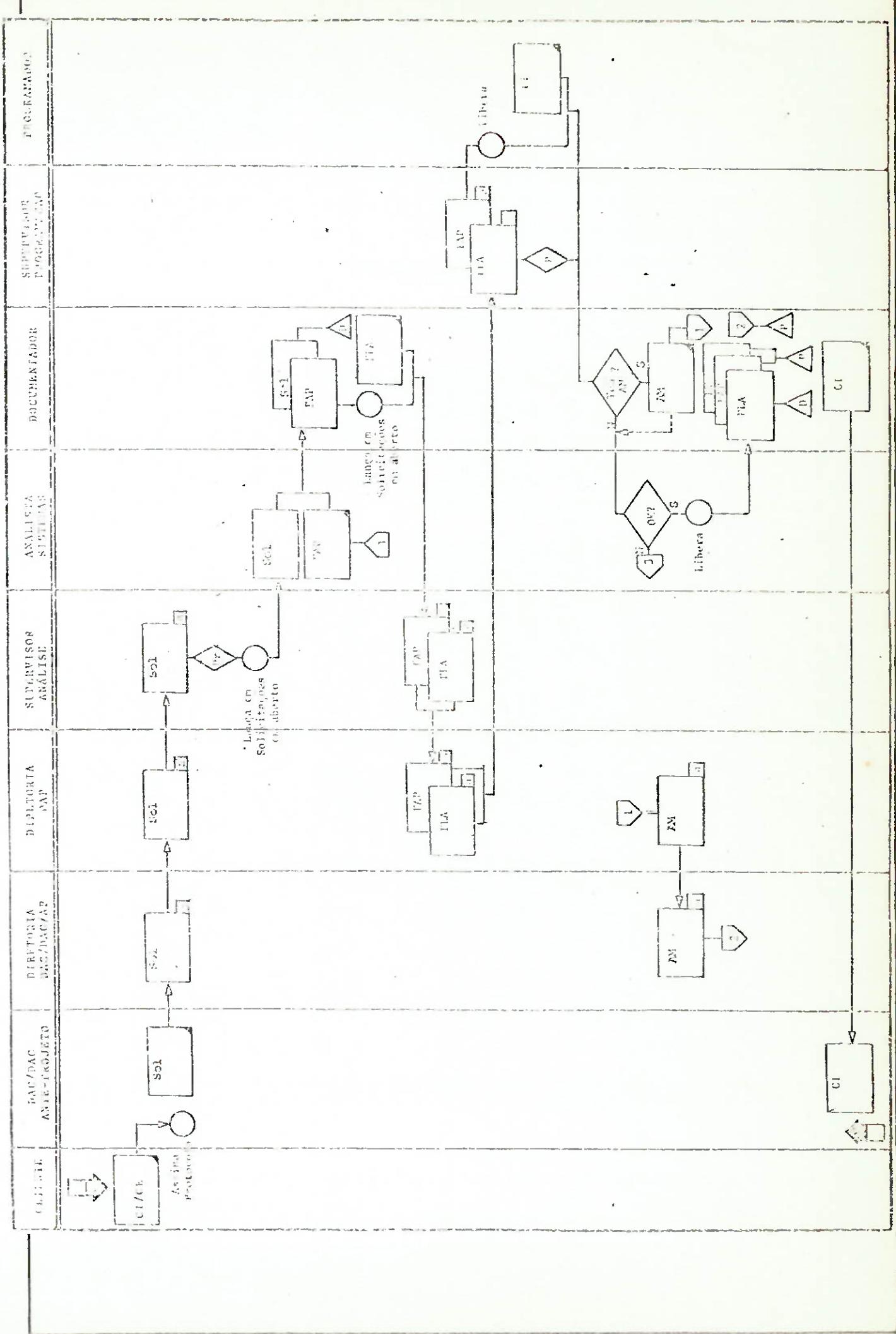
FAP     Folha de Alteração de Programa

FLA     Folha de Alteração para Análise

LT      Listagem de teste

AM      Alteração de Manual

PR      Prioridade



## 5.2- Atividades no Departamento de Análise e Programação

As tarefas desenvolvidas pelos analistas e programadores dentro do Departamento influem no tempo de desenvolvimento e manutenção de programas, ou seja, quanto maior a divisão de tarefas, menor o tempo gasto desde a análise até a liberação. Com o objetivo de dar critérios de comparação, enuméramos as atividades destes dois departamentos da Lojicred Processamento de Dados.

### 5.2.1- Analistas de Sistemas e O & M

- Efetuar os levantamento iniciais para a elaboração de um sistema ou de rotinas de O & M.
- Elaborar
  - Ante-projeto de sistemas
  - Ante-projeto de rotinas de trabalho
  - Orçamento
  - Projeto Piloto
  - Cronograma de atividades
  - Definição de programas
  - Material para testes do sistema
- Testar e liberar os sistemas
- Acompanhar e orientar a confecção dos Manuais de Operação, Digitação, Expedição, Vendas, Sistema e Procedimentos.
- Apresentar o sistema (em reunião) aos setores envolvidos e entregar os respectivos manuais.
- Treinar cada setor envolvido no sistema e implantar, realçando as respectivas necessidades e responsabilidades.
- Executar um paralelo de novo sistema mecanizado com o sistema vigente, providenciando a correção das irregularidades observadas.
- Acompanhar a implantação a ser feita pelo Departamento de Assistência a Clientes no caso de Sistemas.
- Implantar as rotinas de trabalho, ou novos controles nos casos de tarefas de O & M.
- Estudar a viabilidade das solicitações de alterações de Sistemas.

- Definir as alterações de programas e orientar as alterações correspondentes no manual.
- Testar e liberar as alterações e correções solicitadas, entregando os formulários comunicadores das modificações aos setores envolvidos.
- Acompanhar a implantação das alterações ou correções.

NÍVEIS DE ANÁLISE		LIDER DE PROJETO	ANALISTA SENIOR	ANALISTA PLENO	ANALISTA JUNIOR	ANALISTA TRAINEE
I n s t r u ç a o	Universitária completa (matemática, administração, economia, engenharia)	x	x			
E x p e r i ê n c i a	Universitária cursando (matemática, administração, economia, engenharia)			x	x	x
	Levantamento de Dados	x	x	x	x	
	Racionalização de Rotinas e formulários	x	x	x		Não
	Documentação	x	x	x	x	é
	Estrutura Sintética	x	x	x		exigida
	Controles Básicos (Cronogramas, Orçamentos, etc.)	x	x	x		experi-
	Preparação e Execução de Testes	x	x	x	x	ência
	Preparação e Execução de Paralelos	x	x	x		
	Chefia e Liderança	x				

### 5.2.2- Analista de Software

- Desenvolver subrotinas de apoio à programação e operação.
- Implantar "pacotes" de Software.
- Efetuar pesquisas sobre novas técnicas e participar de cursos na área de Software.

	NÍVEIS DE ANÁLISE	ANALISTA SENIOR	ANALISTA PLENO	ANALISTA JUNIOR
I n s t r u c a o	Universitária Completa ou cursando (matemática, economia, administração, Engenharia).	x	x	x
E x p e r i é n c i a	Software	x	x	Não é exigida experiência
E x p e r i é n c i a	Linguagem Assembler ou Cobol	x	x	
	Programação	x	x	
	No Computador Utilizado	x		

### 5.2.3- Documentador de Sistemas

- Descrever o objetivo, conteúdo e destino dos relatórios e documentos impressos pelos sistemas.
- Descrever o objetivo, modo de preenchimento e fechamento de lotes de documentos e planilhas de entrada dos sistemas.
- Desenhar os lay-out's definitivos dos arquivos dos sistemas.
- Elaborar as folhas de operação dos programas e o fluxograma do sistema.
- Elaborar os índices e coletar os anexos para montagem dos manuais.

- Organizar a elaboração de cópias reprográficas para a montagem dos manuais em duplicata.

#### 5.2.4- Programadores

- . Estruturar uma lógica (em diagrama de bloco) com base na definição do programa.
- . Codificar a lógica obtida e enviar a Digitação.
- . Montar o programa fonte e cadastrá-lo em arquivo de fontes catalogando-o em seguida para testes.
- . Constituir uma massa de testes para cada programa.
- . Testar e liberar os programas
- . Efetuar alterações em programas
- . Documentar o programa.

NÍVEIS DE PROGRAMADOR		Programador Senior	Programador Pleno	Programador Junior
Instrução	2º Grau Completo	x		
Experiência	2º Grau Cursando		x	x
	Linguagem Cobol	x	x	Não
	Linguagem Assembler	x	x	é
	Lógica	x	x	exigida
	JCL, DOS/VS, ETC	x	x	experiência

#### 5.3- Apontamento de Horas

O formulário de apontamento de horas utilizado pela empresa é uma entrada para um sistema automático de apuração destas horas para dar um retorno aos chefes de departamento, sobre a performance dos seus funcionários e do volume de solicitações atendidas naquele período.



Processamento de Dados Ltda.

CONTROLE DE SERVIÇOS EXECUTADOS

(NOMI): Paulino G. Francischini

DATA	2	0	1	0	8	0
CAT	0	2				
CHINA	0	0	6	5	8	
SEES	2	2	0			
Hill	0	1				
CT	0	1				

0. N.	DP	PRIMA/SIST	HORAS	SIT	Q.DIA
0	0	0	0	0	0
0	0	0	0	0	0

21	0   0   8   3   0	0   2	S   1   3   0   0	0   0   3   0	0   3   0	38	40
26	28	34					

0	5	0	3	7	1	1	2
0	5	0	3	7	1	1	2

CARICATORIA

OPEN ACCESS

卷之三

01 - PROJETOS  
02 - ANALISANDO DOCUMENTOS  
03 - ACERBIOS

**S I T U A Ç Ã O**  
01 - INTERROMPIDO  
02 - CODIFICAÇÃO  
03 - TESTE  
04 - ENTREGUE

Cann 80:111-118

卷之三

Os funcionários o preenchem diariamente e o entrega à secretaria no final do expediente ou na manhã do dia seguinte. Torna-se oportuno descrever aqui o dia normal de trabalho do programador propriamente dito.

O departamento de Programação da Lojicred Processamento de Dados, não possui departamento exclusivo de desenvolvimento de programas; todos os programadores dedicam-se a manutenção e desenvolvimento conforme as prioridades de cada Ordem de Serviço, de modo que associado às restrições de hardware para a execução de testes, o programador planeja o seu dia para atender à todos serviços que tem em mãos.

No começo do seu dia de trabalho, examina os testes que enviou para executar no final da tarde do dia anterior, sendo que um programador tenha em média quatro programas nesta fase.

A análise dos testes ocupa grande parte do tempo, cerca de 60%, devendo o programador fazer as devidas correções para novo teste ou entregar a pasta do programa para o chefe do departamento. Os testes são executados durante o intervalo de refeições, retornando no início da tarde.

No restante do período o programador dedica-se à codificação, depuração, documentação ou estudo de novos programas, completando seu dia de trabalho. Nota-se aqui que o tempo que dedica ao desenvolvimento é variável, dependendo do que ocorrer na fase de análise de testes, que muitas vezes possuem uma prioridade maior. Assim a unidade de tempo utilizada foi Hora e não Dia, conforme sugeria a maioria dos métodos pesquisados.

Os campos utilizados pelo sistema de controle de serviços executados são:

CT - Código de Cartão (01 ou 09)

DEP - Departamento a que pertence o funcionário

SETOR - Setor a que pertence o funcionário

CHAPA - Nº de chapa do funcionário

CAT - Categoria

01 - Programador

02 - Analista e Líder de Projeto

DATA - Data da execução do serviço

O.S. - nº da Ordem de Serviço

00801 - Diversos

00802 - Pesquisa Técnica

00803 - Manutenção de Programas

00804 - Alteração de Borderô

00805 - Sistema Padrão de Financeira

00806 - Sistema Crediário

Etc.

O.P. - Operação desenvolvida pelo funcionário no Programa /Sistema.

01 - Programação

02 - Análise/Coordenação/Documentador

03 - Acertos

PROGRA/SIST - nº de programa/sistema que está sendo desenvolvido ou em manutenção.

SIT - Situação que se encontra o programa

01 - Interrompido

02 - Codificação

03 - Teste

04 - Entregue

TESTES - totais transcritos dos testes executados

QUANT - quantidade de testes feitos no dia para o programa/- sistema

MIN - total de minutos gastos nos testes executados

FOL - total de folhas gastas nos testes executados

TOTAL - Total de horas e Testes do dia.

Pode-se colocar linhas adicionais se o número de programas manipulados for maior que 4.

#### 5.4- Relatórios

O processamento destes dados de entrada consiste apenas na consistência, classificação, totalização e listagem. Os relatórios variam de acordo com o nível de quebra que se deseja nas totalizações.

• Relatório Analítico

Totaliza por

DEP	CHAPA	PROGR/SIST	O.S.
-----	-------	------------	------

Apresenta praticamente uma listagem classificada dos dados de entrada e totalizada segundo os níveis descritos acima. Sua finalidade é mostrar de maneira detalhada o volume de trabalho executado no mês.

No anexo 1, encontra-se uma listagem do relatório analítico.

• Relatórios de Totais por O.S.

O programa lista um relatório de totais por Ordens de Serviços, classificadas pelo número de cadastramento. A finalidade deste relatório é auxiliar na avaliação dos custos e margem de lucro, sendo que uma via é enviado ao Departamento Administrativo.

Os níveis de detalhe de setor, departamento, chapa, e programa/sistema são suprimidos conforme vemos na listagem deste relatório contida no anexo 1.

• Relatórios de Totais por Chapa

Totaliza por

SETOR	DEP	CHAPA
-------	-----	-------

Este relatório suprime a O.S. em que o funcionário esteve trabalhando, que consistirá em um ponto de crítica neste relatório, ou seja, é uma listagem do número de horas que o programador deveria ter trabalhado.

É classificado por ordem crescente de chapa dentro de um mesmo departamento, e por ordem crescente do nº do departamento dentro de um setor. A listagem deste relatório está no anexo 1.

5.5- Crítica do Sistema Atual

Até agora dedicamo-nos em apresentar o sistema de controle'

de serviços executados sem a preocupação de apontar suas faihas. A análise crítica concentra-se em três pontos principais: dados de entrada, relatórios de saída e as decorrências dos dois primeiros.

Os pontos discutidos nos itens a seguir se relacionam ao Departamento de Programação.

#### 5.5.1- Distorções no apontamento das horas

Discutimos no capítulo 2 as fases produtivas e não produtivas das atividades em programação. Isto quer dizer que necessariamente uma parcela das horas trabalhadas não poderão estar alocados em O.S. produtiva. Ao observarmos a listagem do relatório analítico vê-se que a máxima precisão que se tem no apontamento é de meia hora e sempre arredondados "para mais". A não observância destes detalhes pode introduzir erros consideráveis no cálculo do custo da Ordem de Serviço visto que o erro é acumulativo a cada arredondamento.

As causas destas distorções são basicamente três:

- O programador encobre as horas ociosas.
- O preenchimento do formulário é difícil para o programador recordar a distribuição das horas gastas em cada programa.
- Os relatórios não são devidamente analisados pelos chefes de departamento e muitos erros de preenchimento de formulário não são apontados. Na falta de um retorno para o programador, incentivando a preencher o formulário corretamente, acaba descuidando por completo desta tarefa.

Há uma O.S. denominada DIVERSOS cuja finalidade é alocar períodos em que o funcionário não trabalhou por motivos extraordinários ou cuidou de outros encargos designado pelo chefe de departamento.

Sendo o programa apenas um organizador de dados de entrada, não possui cadastro e nem há possibilidade de fechamento para verificar a validade dos dados de entrada. Isto quer dizer que um programa pode estar

com a ordem de serviço errada, ou algum número trocado, o programa o aceitará e causará erro nas acumulações. A simples análise da listagem não acusa a presença deste erro e o único meio de consistir é através de comparação dos programas e O.S. atendidas naquele mês durante o processamento. A verificação manual é impraticável.

### 5.5.2- Falhas nos Relatórios

Analisamos aqui as deficiências dos relatórios em termos de lay-out, de informações desnecessárias e omitidas.

#### . Relatório Analítico

P	SET	CHAPA	CAT	C.S.	P	PROGRAMA	TEMPO	GASTO	TESTES		ENTREGA		SITUAÇÃO	
									SISTEMA I	HORAS	VALOR	QTE.	MINUTOS	VALOR
01	020	00424	02	00610	01	049003	8,0	1.912,00	1	4	41,66	40	4	4,60
01	020	00424	02	00610	01	049003	5,0	1.912,00	1	4	41,66	60	6	6,90
01	020	00424	02	00610	01	049003	0,5	119,50	2	6	02,60	35	4	4,03

A linha de maior detalhe do relatório analítico é reproduzida acima e vemos que informações como código de cartão, setor, departamento, nº de chapa e função são impressas em todas as linhas quando esta informação é necessária apenas na identificação de cada quebra.

O campo DATA que informaria o dia do mês em que o funcionário trabalhou naquele programa é omitido, - obrigando ao chefe da Programação a ter um apontamento paralelo de dias trabalhados. Basear-se no número de linhas registradas para um mesmo programa pode fornecer informações erradas porque dentro de um mesmo dia , o programa pode ter sido codificado e testado, gerando uma linha para cada atividade.

Para o campo SITUAÇÃO, há quatro opções possíveis.

- 1 - Interrompido
- 2 - Codificação
- 3 - Teste
- 4 - Entregue

No capítulo 2, as tarefas desenvolvidas pelo programador são:

- Estudo
- Codificação
- Depuração
- Teste
- Documentação

Comparando as duas classificações de tarefas ou situações, vemos que apenas duas estão previstas no sistema atual, desprezando as demais e introduzindo erros na descrição das situações.

Não há critério para uso dos campos 01 e 04. No caso de campo 01, deveria ser anotado sem que os campos Tempo Gasto e Testes Efetuados estivessem preenchidos para afirmar a efetiva interrupção do programa e receber outro cuja prioridade é 00 ou URGENTE. Se a interrupção for de um dia de trabalho, esta situação deve ou não ser apontada? O que pudemos observar é que os funcionários não recorrem ao manual para preencher corretamente o formulário.

As situações 02 e 03 são evidentes e são utilizadas pelo programador mesmo com as limitações apontadas anteriormente.

Entregue deveria ser encontrada na listagem quando o programador liberasse o programa porém:

- Os testes podem ter sido rejeitados pela Análise e voltado para a programação.
- Pode ocorrer que haja mais de uma manutenção num mesmo programa dentro do mês.
- O programador pode preencher por engano o campo ou mesmo propositadamente.

Assim sabermos quantas vezes o programa sofreu alteração é muito difícil e na maioria dos casos este item é omitido nos apontamentos de horas.

- Relatórios de totais por chapa.

Visto que o funcionário pode alterar a distribuição das horas sem que seja notado, este relatório transforma-se na listagem das horas em que o programador

deveria ter trabalhado, ou seja é desnecessário e não tem utilidade dentro da empresa.

Para o caso de horistas ou "free-lance" talvez pudesse ser útil, mas como o número destes funcionários é reduzido, não justifica seu processamento.

. Relatório de totais por Programa/Sistema

Um relatório de grande utilidade para os chefes dos departamentos de Programação e Análise, é o de totais por Programa/Sistema, sem o nível de detalhe que tem relatório analítico.

Seria totalizado apenas por setor, omitindo as quebras por O.S., Departamento e Chapa. Assim contornaríamos os problemas de erros em designação de O.S., seria mais fácil detectar um erro de preenchimento do número de Programa e mesmo que haja duas manutenções em um mesmo mês alocadas em programadores diferentes, elas apareceriam juntas, facilitando uma visão de conjunto do desempenho do departamento.

#### 5.5.3- Falhas na Análise dos Relatórios

Se os dados de entrada podem estar alterados, verifica-se um ditado em Processamento de Dados "lixo entra, lixo sai".

O primeiro ponto de crítica na análise de relatórios não depende deles, mas sim do seu conteúdo, que pode estar fora da realidade devido ao apontamento mal feito; ou seja se houver possibilidade de análise, levará a decisões equívocas.

. Relatório Analítico

- Com a omissão do campo DATA no relatório, o número de dias trabalhados naquele programa é aproximadamente o número de linhas. Com a inclusão desse campo, a retirada dos campos supérfluos apontados, e devida ordenação por data, forneceria ao chefe da programação dados para análise.

- O relatório apenas traz o total do mês de processamento e não o que foi gasto no(s) mês(es) anterior(es), assim uma visão global é difícil de ser obtida.
- Se houver erros no preenchimento dos formulários horas de um mesmo programa aparecerá em vários totais, dependendo do erro; por exemplo: troca de O.S., nº de programa errado, ou seja, para obter-se o total real deverá somar vários totais parciais, se os erros puderem ser identificados.
- Não existe nenhuma informação sobre a complexidade da alteração ou desenvolvimento de modo que não há termos para comparação.

- Relatórios de Totais por O.S.

Devido aos erros no preenchimento de formulário, as informações contidas neste relatório não são confiáveis, tendo que apelar para processos manuais de informações de custo.

- Relatório de Totais por Chapa

Sem o nível de detalhe das O.S. trabalhadas durante o mês, este relatório traz pouca informação para os chefes de departamento e pode ser considerado como relatório supérfluo.

## 6 - O Método de Estimativa de Tempos e Orçamento

Depois de discutirmos os vários métodos para estimação de tempos e criticar o sistema atual de controle de serviços executados, passamos ao desenvolvimento do projeto de orçamento de programas em desenvolvimento e em manutenção, constituindo este capítulo, o principal tópico do trabalho.

Procuramos mostrar as várias fases do processo de forma breve e objetiva, visto que nos capítulos anteriores foram apresentadas as justificativas e hipóteses assumidas aqui.

### 6.1- Estudo Preliminar

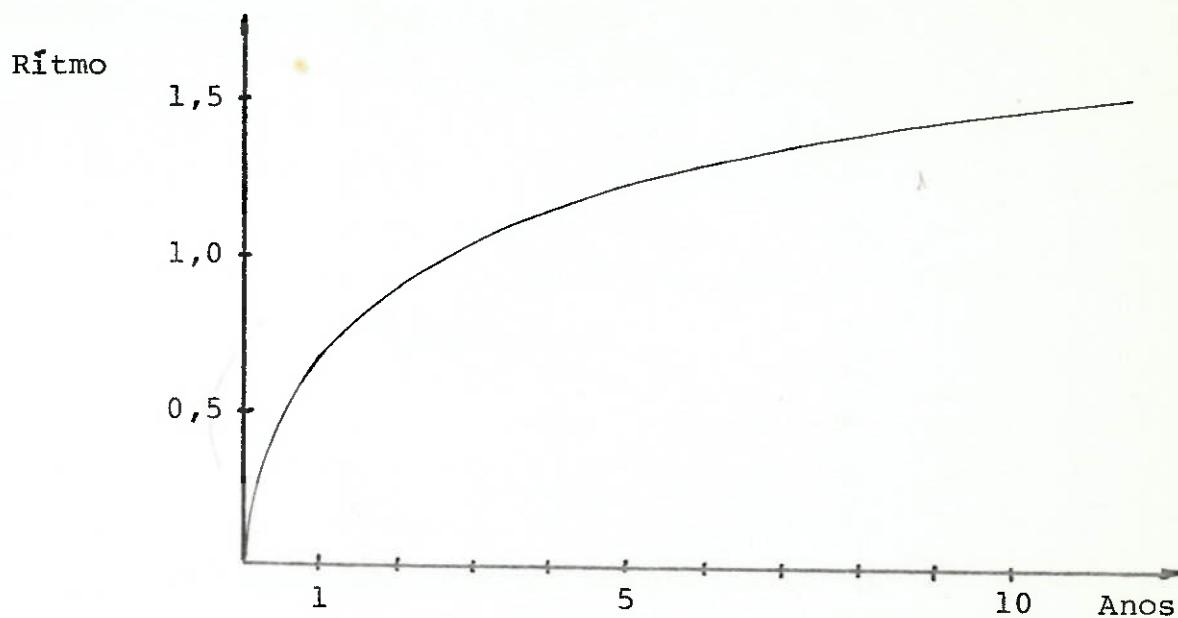
Nem todos os pontos criticados nos métodos alternativos foram resolvidos pela aplicação deste, principalmente no que diz respeito ao ritmo do programador. Nenhum dos trabalhos apresentados justificavam a presença de valores para cada categoria. Em um experimento relatado por Feldmann (1), exposto no capítulo 2 chega aos seguintes valores:

Júnior	0,4	a	0,8
Pleno	0,9	a	1,0
Senior	1,1	a	1,5

Vemos que ainda assim a faixa é grande pois para um programa em que um programador pleno leva 100 horas para fazer, um júnior leva de 125 a 250 horas.

Com estes valores, os melhores que pode-se encontrar na bibliografia, constroem-se um gráfico em função da experiência do programador.

(1) Como estimar prazos certos em programação, Feldmann Paulo Roberto.



A equação será da forma

$$y = a x^b \quad \text{ou}$$

$$RT = a N^b$$

onde

$RT$  = Ritmo do programador

$N$  = número de anos de experiência em programação.

linearizando

$$\ln RT = \ln a + b \ln N$$

$N$	$\ln N$	$RT$	$\ln RT$
0,5	- 0,693	0,4	- 0,916
1,0	0	0,6	- 0,510
1,5	0,405	0,8	- 0,223
2,0	0,693	0,9	- 0,105
3,0	1,098	1,0	0
4,0	1,389	1,15	0,139
5,0	1,609	1,3	0,262
7,0	1,946	1,4	0,336
9,0	2,197	1,45	0,371
10,0	2,302	1,5	0,405

obtemos

$$RT = 0,611 \quad N^{0,4273}$$

Para facilidade de consulta, construiu-se a tabela

N	RT*
0,5	0,40
1,0	0,61
1,5	0,78
2,0	0,82
2,5	0,90
3,0	0,97
3,5	1,04
4,0	1,10
4,5	1,16
5,0	1,21
5,5	1,26
6,0	1,31
7,0	1,40
8,0	1,38
9,0	1,56
10,0	1,58

Esta é a primeira tentativa de resolver o problema e nos tópicos seguintes estudaremos algumas melhorias.

Foram frustradas todas as tentativas de usar o mesmo método de estimação de tempo para desenvolvimento de programas para manutenção. Não houve outra solução do que partirmos para as avaliações subjetivas ou seja contar com a experiência do chefe da programação para avaliá-los. Seguindo o exemplo de trabalhos anteriores procura-se introduzir padrões de dificuldades para facilitar a avaliação, e a aplicação de qualquer outro método mais complexo tais como peso para funções, o volume de trabalho para o encarregado seria grande e o tempo de cálculo maior que a correção do programa propriamente dito.

Com o levantamento de dados obtemos um histórico do

desempenho daquela equipe de programação, mas como saber se o esforço para desenvolver e corrigir programas, estava adequado aos padrões normais de produtividade? A bibliografia existente recomenda considerar as tolerâncias durante o dia de trabalho como

- . Pessoais ] 15% em média
- . Fadiga ]
- . Espera → amostragem

Estes valores foram obtidos na Lojicred Processamentos de Dados com amostragem tipo PRODUZINDO e NÃO PRODUZINDO. Como a equipe de programadores é pequena e o espaço não é grande, é fácil saber em que condição estava o programador. Esta amostragem está sujeita a algumas críticas, tais como:

- . Há períodos do dia em que o trabalho é mais intenso, principalmente na hora anterior ao envio dos testes do período para processamento.
- . Um programador pode estar sentado na mesa mas "simulando" trabalho, e numa observação rápida, não se percebe.
- . Este método inibe os programadores e é muito frequente de mostrarem que não aceitam "serem vigiados" com frequentes queixas ao chefe do departamento.

A condição de ESPERA foi suprimida por causa da dinâmica de trabalho adotada, ou seja se a perfuradora estiver ocupada, poderá cuidar de outro teste ou do programa mais recente. Nos casos em que estão ociosos, foi assumido que estão recompondo da fadiga.

Temos então:

Condição	Nº de observações	Porcent.
Produzindo	99	79%
Não Produzindo	27	21%
Total	126	100

O número de observações necessárias é calculado pela formula

$$N \geq \left( \frac{z_{\alpha \%}}{er} \right)^2 \left( \frac{1 - p}{p} \right)$$

Nível de significância = 95%

erro relativo er = 10%

$$N \geq \left( \frac{1,96}{0,10} \right)^2 \left( \frac{1 - 0,79}{0,79} \right) = 102$$

## 6.2- Levantamento de Dados

O método escolhido é em si, simples de ser aplicado pois utiliza pacotes já prontos para fazer a maioria dos cálculos estatísticos. O maior problema encontra-se na fase de levantamento de dados, pois para obtermos resultados confiáveis o número de amostras deve ser grande. Devido ao fato de que o tempo disponível ser relativamente certo, o nº de amostras foi pequeno conforme analisaremos no capítulo 7, mas o suficiente para analisarmos seus resultados com segurança razoável.

### 6.2.1- Pesquisa de tempos de desenvolvimento e manutenção de programas

Para a análise de regressão foram levantados perto de 150 programas nos três anos anteriores. Devido ao fato dos formulários serem mal preenchidos, o número de horas de alguns, principalmente os mais antigos eram nitidamente falsas devido à disparidade com relação a média.

Destes, selecionou-se 98 programas sendo que a maioria pudemos acompanhar seu desenvolvimento e garantir a validade dos dados.

Como dissemos anteriormente, o volume de manutenções em um DAP, supera o de desenvolvimento e não foi difícil reunir uma amostra grande para nossos estudos; seu número estava por volta de 460 manutenções, porém por questões de tempo, e pelas mesmas razões apontadas quanto ao preenchimento dos formulários, seu número

caiu para 250 manutenções.

A distribuição dos programas em desenvolvimento é

T i p o	Nº
Organizadores	20
Consistência	08
Atualizadores	15
Calculadores	15
Formatadores	14
Listador Simples	16
Listador com Operação	10
T o t a l	98

Para manutenção, a distribuição é

MANUTENÇÃO ROTIAS	1	2	3	4	5
Leitura	3	4	4	0	6
Consistência	0	0	5	0	6
Organizadores	5	8	4	5	8
Cálculo	7	6	3	6	6
Atualização	0	7	5	3	7
Gravação	4	6	4	0	8
Impressão	4	6	4	0	5
Processamentos Diversos	5	7	8	0	8
Programas Paralelos	3	5	5	6	

### 6.2.2- Categorização de Programas

Se um DAP não se orienta para a aplicação de programação estruturada torna-se difícil classificar um programa pelas categorias que foram apresentadas.

No levantamento de dados ocorriam situações em que uma decisão tinha que ser tomada para classificar um programa e se esta mesma situação se apresentava mais vezes a análise e decisão era colocada como guia de categorização de programas, como apresenta o capítulo 2.

O problema de dificuldade de funções era de fácil solução, porque na definição do programa notava-se uma parte principal e outra secundária, para preparar a entrada dados para a principal, como é o caso de organização ou saída de dados como no caso de formatação ou listagem. Nota-se que nos dois casos, se o programa antes de ser processado na rotina principal precisa de um preparo de dados, o nº de registros de entrada ou de movimentos será grande e permitirá que o programa apresentado possa ser estimado com relativa segurança do mesmo modo ocorre com uma rotina secundária de saída. No entanto, como trabalha-se com estaística, não deixará de existir programas que não seguirão as considerações feitas aqui, o que não deve surpreender.

Segundo Feldmann, a duração média destes programas em termos de dias para programação modular, distribui-se pelas várias categorias de programas segundo a tabela abaixo:

Categoría	Nº de Programas	Duração Média (*)
1 - Atualizadores	17	35,8
2 - Consistência	18	27,1
3 - Listadores	27	20,5
4 - Organizadores	25	12,5
5 - Formatadores	22	10,6
T o t a l	109	20,1

(\*) Já corrigida pelo fator de ritmo ou seja, ajustados pela experiência do programador. Unidade: dias.

Na programação que podemos classificar como "semi-modular" a situação altera-se um pouco. Por exemplo:

- . Alguns DAP's adotam o sistema de formatar os registros dentro do programa de consistência, e outros o separam como acontece na Lojicred Processamento de Dados.
- . Os programas calculadores quase sempre tem um relatório de saída, pois assim "economiza" um programa listador. Com este histórico que pode não necessariamente ocorrer, "vicia" a amostra e prejudica a estimativa de um programa que não apresente esta saída.
- . A grande dificuldade em categorizar programas listadores gerou a necessidade de dividí-lo em duas subcategorias, porque era patente que embora os programas apresentarem sómente relatório de saída, não gravando nenhum arquivo, apresentavam duas distribuições distintas. Este acréscimo no tempo de

desenvolvimento era devido ao fato de apresentarem algumas subrotinas preparatórias para a listagem sendo uma parte considerável do programa que não poderia ser desprezada. A estes listadores sem subrotinas consideráveis ou pequenas, demos o nome de Listador Simples enquanto que ao outro tipo denominamos de Listadores Complexos ou Listadores Com Operação.

Até aqui falamos em desenvolvimento de programas, mas as manutenções também apresentam casos para apreciação. A classificação apresentada no capítulo 2 evidentemente não abrange todos tipos de manutenções que podem ser solicitadas ao DAP, porém a grande maioria está classificada.

- A categorização é mais difícil de ser feita no caso de manutenção do que no desenvolvimento, porque nestes últimos, as funções já estão consagradas em Processamento, enquanto que as alterações nos programas assumem características tão diversas que torna difícil sua categorização. Para resolver isto, introduziu-se uma categoria de Processamentos Diversos para que o chefe da Programação, na falta de outro recurso melhor, usar este.
- É comum que em uma mesma solicitação tenha que fazer várias modificações e mais de uma subrotina. Como são praticamente independentes, como já foi comentado, o tempo total será dado pela soma das duas médias, com a segurança dos desvios padrões respectivos pelo fator de ritmo dos programadores.
- A dificuldade que o programador encontra ao estudar a lógica do programa, para fazer a alteração, não é fácil de ser prevista pela simples leitura da solicitação de alterações enviada pelo Departamento de Análise. Porém é uma variável importante de ser conhecida porque pode mudar substancialmente a dificuldade de manutenção. Nos CPD's os programas "enrolados" são conhecidos pelo chefe de Programação, o que facilita sua avaliação.

- Os programas paralelos são mais ou menos difíceis de serem feitos devido à função que processa (cálculo, alteração, listagem, etc), porém por falta de dados esta classificação foi omitida.

#### 6.2.3- Tabelamento das Variáveis Objetivas

As variáveis testadas foram:

- V01 - nº de arquivos de entrada sem arquivo de guia
- V02 - nº de arquivos de entrada com arquivo de guia
- V03 - nº de arquivos de saída
- V04 - nº relatórios
- V05 - nº de registros de entrada
- V06 - nº de registros de saída
- V07 - nº de chaves UPSIS
- V08 - nº de códigos de movimento
- V09 - Tempo de desenvolvimento corrigido.

- As variáveis V01 e V02, não podem aparecer em uma mesma equação, mas somente aquela que apresentar melhor resultado para a regressão.
- Se duas ou mais variáveis forem fortemente correlacionadas, o procedimento STEPWISE se encarregará de retirá-la da equação.
- V09, na equação de regressão é denominada Variável Dependente e as demais, Variáveis Independentes.

A finalidade do levantamento de dados, é obter as tabelas colocadas nas páginas seguintes, conforme a sequência de categorias de programas

- Atualizadores
- Calculadores
- Consistências
- Organizadores
- Formatadores
- Listadores Simples
- Listadores com Operação

## Atualizadores

PROGR.	V01	V02	V03	V04	V05	V06	V07	V08	V09
1	2	3	4	1	9	10	3	9	283
2	2	3	2	1	9	10	2	5	209
3	2	3	2	1	7	9	4	6	280
4	2	3	2	2	3	9	1	3	95
5	2	2	1	3	4	9	0	1	87
6	3	3	1	3	4	18	2	1	127
7	2	2	3	1	10	12	2	1	250
8	2	2	1	1	2	2	0	1	65
9	2	2	2	1	1	2	0	2	60
10	2	3	1	1	2	3	2	2	100
11	2	3	1	0	2	1	2	4	77
12	2	2	1	1	3	3	0	2	75
13	2	3	2	1	9	12	1	10	309
14	2	2	2	1	3	4	0	2	95
15	3	3	1	1	4	5	4	9	231

## Calculadores

PROGR.	V01	V02	V03	V04	V05	V06	V07	V08	V09
1	1	2	1	0	7	3	1	0	120
2	1	1	0	2	2	5	3	0	130
3	1	2	1	0	4	10	0	0	160
4	1	2	0	1	3	4	0	0	140
5	1	2	1	0	3	1	2	0	80
6	1	1	0	1	2	8	6	0	190
7	1	1	1	1	4	3	0	0	80
8	1	1	0	1	5	2	0	0	91
9	1	1	0	1	6	2	0	0	92
10	1	2	0	1	2	3	1	0	46
11	1	2	3	4	3	10	2	0	170
12	1	2	0	1	1	7	1	0	70
13	1	1	2	1	2	5	0	0	45
14	1	1	0	1	2	3	0	0	44
15	1	2	0	1	2	7	2	0	98

## Consistênciā

PROGR.	V01	V02	V03	V04	V05	V06	V07	V08	V09
1	1	2	2	1	6	8	3	7	180
2	1	2	1	1	3	6	0	13	264
3	1	1	1	1	2	13	0	10	186
4	2	2	1	1	1	7	0	7	60
5	1	2	1	1	2	3	1	0	40
6	2	3	1	1	5	14	0	0	264
7	1	2	1	1	3	8	0	0	95
8	1	1	1	1	3	5	0	0	56

## Organizadores

PROGR.	V01	V02	V03	V04	V05	V06	V07	V08	V09
1	2	3	2	1	2	3	0	0	52
2	1	2	6	0	3	5	0	0	88
3	2	3	2	1	2	2	1	0	47
4	1	2	5	0	4	5	0	1	90
5	1	2	1	0	1	2	0	0	41
6	1	1	3	0	1	1	0	0	33
7	2	2	2	1	1	2	0	0	43
8	1	2	2	0	3	3	0	0	18
9	1	2	3	1	3	4	0	0	45
10	1	1	1	0	2	2	0	0	13
11	2	3	3	0	4	3	0	0	38
12	2	3	3	0	2	2	1	0	42
13	2	2	1	3	4	6	0	0	120
14	1	2	1	1	1	9	1	0	86
15	2	3	2	1	2	19	1	3	120
16	2	3	2	3	2	9	2	3	133
17	1	2	3	2	1	18	1	5	170
18	2	3	1	3	1	12	0	3	170
19	3	3	1	1	3	2	10	0	110
20	2	3	1	1	2	3	3	0	15

## Formatadores

PROGR.	V01	V02	V03	V04	V05	V06	V07	V08	V09
1	4	5	1	0	5	1	0	0	75
2	2	3	1	0	2	3	1	0	21
3	2	3	2	1	2	2	0	0	30
4	1	1	1	0	3	2	0	0	35
5	3	3	0	1	3	12	0	0	42
6	1	1	1	0	1	2	0	0	18
7	1	2	2	1	2	6	2	0	64
8	1	1	2	0	1	2	4	0	30
9	2	2	2	1	1	2	0	0	44
10	1	1	3	0	3	3	0	0	38
11	1	2	3	1	3	4	0	0	55
12	2	3	3	0	4	3	0	0	60
13	2	3	2	0	5	5	1	0	79
14	1	2	1	0	2	3	0	0	21

## Listadores Simples

PROGR.	V01	V02	V03	V04	V05	V06	V07	V08	V09
1	1	2	0	1	1	5	1	0	55
2	1	2	0	1	4	3	0	0	47
3	1	2	0	2	4	12	0	0	60
4	1	1	0	1	1	6	1	0	36
5	1	1	0	1	3	3	1	0	35
6	1	1	0	1	1	3	0	0	45
7	1	2	0	2	1	7	1	0	42
8	1	2	0	1	1	2	4	0	44
9	1	1	0	2	3	10	0	0	44
10	1	2	0	1	2	3	0	0	32
11	1	1	0	1	2	2	0	0	41
12	1	2	0	1	3	5	0	0	30
13	1	2	0	1	1	2	3	0	50
14	1	2	0	2	3	8	3	0	71
15	1	2	0	2	3	8	3	0	72
16	1	3	0	1	3	4	1	0	45

## Listadores Com Operação

PROGR.	V01	V02	V03	V04	V05	V06	V07	V08	V09
1	1	2	0	2	1	9	2	0	97
2	1	2	0	1	5	3	2	0	84
3	1	2	0	1	1	4	0		55
4	1	2	0	3	5	7	0	0	112
5	2	3	0	1	3	8	0	0	98
6	1	2	0	2	3	12	0	0	94
7	1	2	0	1	4	3	0	0	60
8	2	2	0	3	1	12	8	0	110
9	1	2	0	1	3	6	0	0	60
10	1	1	0	1	3	6	0	0	60

#### 6.2.4- Avaliação das Variáveis Subjetivas

A impossibilidade de desenvolver um modelo que utilizasse variáveis objetivas para estimar o tempo de manutenção, ocasionou a busca de um método fácil para guiar o chefe da Programação, ao calcular o volume de solicitações de manutenção que poderá ser atendido no mês.

A manutenção do programa depende de três variáveis principais, nas quais engloba-se todas as variações que se encontra na prática. Com isto, perde-se níveis de detalhes que explicariam os pontos fora dos limites estimados.

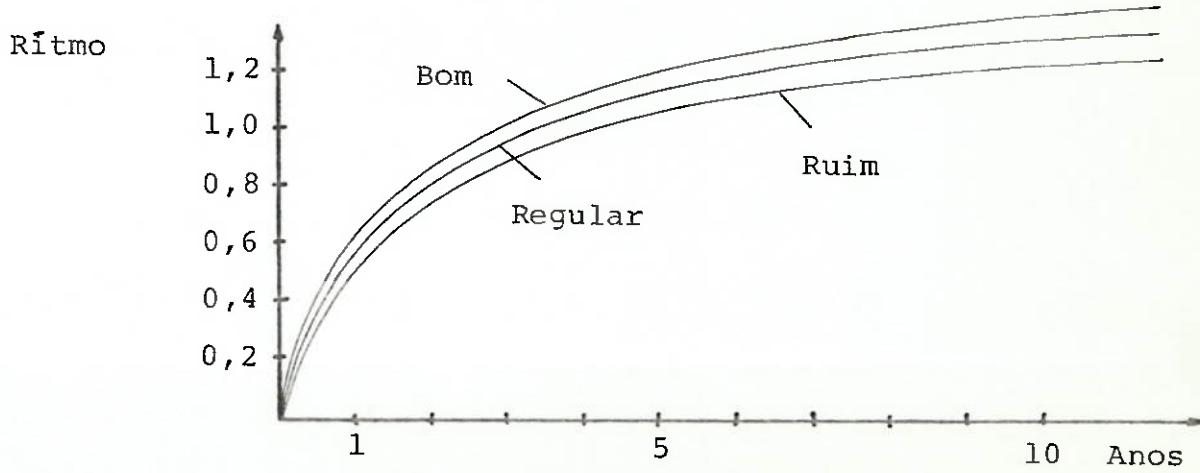
##### 6.2.4.1- Qualificação Individual do Programador

Dissemos no capítulo 2, que o fator de ritmo varia com a experiência e com a qualificação individual do programador. Do mesmo modo que ocorre nas curvas da produtividade no aprendizado, podemos definir três categorias de programadores, conforme discutimos no referido capítulo.

Individualmente um programador pode ser classificado como:

- Bom
- Regular
- Ruim

Reflete-se no gráfico fator de ritmo x experiência segundo as curvas:



Plotando-se pontos conforme as experiências feitas e explicadas no capítulo 2, temos as seguintes curvas de regressão.

. Bom

$$RT = 0,704 \text{ N} \quad 0,3847$$

$$R^2 = 0,98$$

. Regular

$$RT = 0,611 \text{ N} \quad 0,4273$$

$$R^2 = 0,96$$

. Ruim

$$RT = 0,492 \text{ N} \quad 0,4927$$

$$R^2 = 0,97$$

Para facilitar a consulta construiu-se a tabela comparativa

<u>EXPERIENCIA</u>	<u>QUALIFICAÇÃO</u>	RUIM	REGULAR	BOM
0,5		0,35	0,40	0,54
1,0		0,50	0,61	0,70
1,5		0,60	0,78	0,82
2,0		0,70	0,82	0,92
2,5		0,79	0,90	1,00
3,0		0,85	0,97	1,07
3,5		0,91	1,04	1,14
4,0		0,97	1,10	1,20
4,5		1,03	1,16	1,25
5,0		1,09	1,21	1,31
5,5		1,14	1,26	1,36
6,0		1,19	1,31	1,40
7,0		1,28	1,40	1,49
8,0		1,37	1,48	1,57
9,0		1,45	1,56	1,60
10,0		1,53	1,58	1,62

A classificação em uma das três categorias apresentadas dependerá do chefe da Programação, devendo variar de um DAP para outro.

#### 6.2.4.2- Lógica do Programa

Este é o principal fator que determina, que um programa sofra alteração classe 6 ou reprogramação. Influi principalmente no tempo de estudo que o programador gasta para localizar a alteração, e quanto mais difícil a alteração, maior o tempo de estudo.

- Se a estrutura estiver bem esquematizada, será fácil encontrar a subrotina contida na solicitação e o tempo de estudo será pequeno.
- Todos os processamentos que se referem a um tipo de registro ou código de movimento devem convergir para um só lugar no programa ou ter indicadores que apontem outros processamentos. É muito comum ao alterar-se o lay-out de registro, alterar todas as rotinas que o processam.
- A falta de documentação dificulta a localização da alteração. Se uma manutenção for feita e não registram nos formulários apropriados, em uma manutenção posterior haverá incompatibilidades na solicitação ou perdas de teste sem localizar o erro.

Esta variável foi incluída como variante da complexidade da alteração, como veremos a seguir.

#### 6.2.4.3- Complexidade da alteração

Esta variável é a que possui maior peso na determinação do tempo de manutenção. Tem a desvantagem de ser subjetiva e o modo que se encontrou para minimizar as variações na avaliação foi classificá-las em seis categorias

e padronizá-las. O capítulo 2 analisa detalhadamente estas categorias e embora perca em qualidade na avaliação simplifica muito os cálculos e consultas a tabelas.

- . Classe 1 ou Facílima
- . Classe 2 ou Fácil
- . Classe 3 ou Média
- . Classe 4 ou Dificultada
- . Classe 5 ou Difícil
- . Classe 6 ou Reprogramação

Na manutenção classe 4, o tempo de estudo da alteração ganha papel relevante, e incluída dentro da complexidade da alteração. Assim, não é necessária uma terceira dimensão na matriz de médias em manutenção.

### 6.3- Equações e Tabelas

#### 6.3.1- Desenvolvimento de Programas

##### • CALCULADORES

$$Y = 3,1 V02 + 15,4 V05 + 10,0 V06 + \\ + 14,6 V07 - 16,3$$

$$R^2 = 0,77$$

##### • ATUALIZADORES

$$Y = 6,6 V01 + 14,6 V05 + 2,2 V06 + \\ + 12,7 V07 + 10,6 V08 - 4,9$$

$$R^2 = 0,94$$

##### • CONSISTÊNCIA

$$Y = 26,68 V01 + 4,1 V05 + 10,8 V06 + \\ + 12,6 V08 - 62,1$$

$$R^2 = 0,95$$

##### • ORGANIZADORES

$$Y = 11,6 V03 + 26,9 V04 + 4,8 V06 + \\ + 5,5 V08 - 10,0$$

$$R^2 = 0,87$$

##### • FORMATADORES

$$Y = 0,9 V02 + 3,4 V03 + 14,5 V04 \\ + 13,6 V05 + 5,5 V07 - 8,6$$

$$R^2 = 0,81$$

. LISTADORES COM OPERAÇÃO

$$\begin{aligned}Y &= 1,2 V04 + 7,7 V05 + 3,8 V06 + \\&+ 4,9 V07 + 19,02\end{aligned}$$

$$R^2 = 0,84$$

. LISTADORES SIMPLES

$$\begin{aligned}Y &= 14,4 V04 + 1,5 V05 + 0,5 V06 \\&+ 4,4 V07 + 18,1\end{aligned}$$

$$R^2 = 0,80$$

6.3.2- Manutenção

		1	2	3	4	5
LEITURA	$\bar{x}$	1,0	3,0	5,0	7,0*	12,0
	S	0,4	0,5	0,7		2,3
CONSISTÊNCIA	$\bar{x}$	1,0*	6,0*	13,0	15,0*	22,0
	S			2,1		3,8
ORGANIZAÇÃO	$\bar{x}$	1,0	6,0	11,0	15,0	21,0
	S	0,4	1,3	1,1	2,4	5,2
CÁLCULO	$\bar{x}$	2,5	8,0	16,0	18,0	25,0
	S	0,6	1,0	0,8	2,8	4,4
ATUALIZAÇÃO	$\bar{x}$	2,0*	5,0	12,0	17,0	20,0
	S		1,2	1,4	1,4	4,5
GRAVAÇÃO	$\bar{x}$	0,5	2,0	5,0	6,0*	10,0
	S	0,4	0,6	0,8		1,6
IMPRESSÃO	$\bar{x}$	1,0	4,5	7,0	9,0*	16,0
	S	0,4	0,7	0,8		2,3
PROCESSAMENTOS	$\bar{x}$	1,5	5,0	9,5	11,0*	18,0
DIVERSOS	S	0,5	1,8	3,0		4,9
PROGRAMAS	$\bar{x}$	2,5	5,0	11,5	20,0	
PARALELOS	S	0,4	1,4	2,2	5,4	

\* Valores estimados, dados que  $n = 0$

$$S = \frac{1}{C_2} \sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \quad \text{onde}$$

$\frac{1}{C_2}$  = Coeficiente de correção de vício p/amostras pequenas.

Intervalo

$$\left( \begin{array}{c} \text{Tempo} \\ \text{Real} \end{array} \right) = \left[ \left( \begin{array}{c} \text{Tempo} \\ \text{Médio} \end{array} \right) + KS_\alpha \times \left( \begin{array}{c} \text{Desvio} \\ \text{Padrão} \end{array} \right) \right] \times \frac{1}{RT}$$

Tomando

$$KS_\alpha = \frac{t_{n-1, \alpha/2}}{\sqrt{n}}$$

onde

n	KS ( $\alpha=5\%$ )	KS ( $\alpha=10\%$ )
2	8,98	4,45
3	2,48	1,69
4	1,59	1,17
5	1,24	0,95
6	1,05	0,82
7	0,92	0,73
8	0,84	0,67
9	0,77	0,62
10	0,72	0,58

### 6.3.3- Tolerância

Segundo Barnes (1)

$$\left( \begin{array}{c} \text{Tempo} \\ \text{Padrão} \end{array} \right) = \left( \begin{array}{c} \text{Tempo} \\ \text{Normal} \end{array} \right) + \text{Tolerâncias} \times \left( \begin{array}{c} \text{Tempo} \\ \text{Normal} \end{array} \right)$$

O tempo normal, é o tempo corrigido de desenvolvimento de programas obtido no levantamento de dados. Tempo Real e Tolerâncias devem ser estimadas a partir das amostragens de trabalho. O nível de 79% de tempo produtivo foi considerado satisfatório em comparação com a bibliografia, de modo que não fizemos nenhuma correção nas equações. Se a porcentagem de tempo Não Produzindo for elevada, a tolerância deverá ser negativa para corrigir o tempo real.

- (1) Estudo de Movimentos e de Tempos: Projeto e Medida do Trabalho, Barnes, Ralph

#### 6.4- Aspecto Orçamentário

Objetivo de rentabilidade da firma

Como "bureau" de processamento de dados, a empresa tem dois objetivos:

- atender às solicitações das empresas LOJICRED
- atender às solicitações dos clientes cadastrados.

A partir destes objetivos, a diretoria pode tomar três posições ao orçar um sistema ou programa:

- Se o programa/sistema for específico para o cliente solicitante, a fatura terá uma parcela de lucro normal.
- Se o programa/sistema for específico para alguma das empresas LOJICRED será faturado pelo custo do desenvolvimento.
- Se o programa/sistema solicitado pelo cliente ou por uma das empresas LOJICRED puder ser aplicado a outros clientes, a diretoria decide investir no sistema, podendo faturar abaixo do custo.

Como a empresa já possui a maioria dos sistemas tradicionais no mercado (Folha de Pagamento, Crediário, Ativo Fixo, etc), a maioria das solicitações são específicas para cada cliente. Enumeramos algumas a seguir.

- Programas impressores em formulários especiais para a empresa solicitante.
- O cliente adquire um sistema já existente, mas a sequência de numeração de clientes não é aceita pelo sistema, portanto deveria ser feitos programas adicionais.
- As alterações que o cliente solicita podem ser tão profundas que é necessário montar sub-sistemas periféricos. Com isto, alguns sistemas contam com mais de 200 programas.
- Devido ao custo de manter os departamentos de Análise e Programação, alguns clientes optam por desenvolver'

seus sistemas em "bureaux", e utilizar equipamento próprio.

Feitas estas considerações notamos que a rentabilidade da LOJICRED Processamento de Dados baseia-se nos serviços prestados a clientes, em termos de faturamento, e no grupo LOJI em termo de redução de gastos em processamento. O atendimento dos dois objetivos competem nos recursos disponíveis nos Departamentos de Análise, Programação e Operação, assim se qualquer solicitação atendida das empresas LOJICRED é faturada pelo custo, o orçamento para terceiros necessita ser muito bem calculado para evitar que a fonte de renda se transforme em sorvedouro de recursos. Por outro lado, a configuração atual do equipamento utilizado necessita um volume de trabalho maior do que o Grupo pode absorver, isto quer dizer que os clientes não necessários para viabilizar o investimento fixo, gerando uma necessidade de apresentar preços competitivos no mercado, visto que o número de concorrentes é considerável.

Em resumo, em torno do valor estimado, temos que considerar uma região que maximize a função lucro.

O dimensionamento desta região exigiria um estudo profundo nos custos e benefícios para a empresa, o que não é objetivo principal deste trabalho. Dado isto, tivemos que procurar um estimador para a segurança do orçamento e competitividade no mercado, e dos vários testes feitos com a aplicação nos programas já entregues, optou-se pelo orçamento no limite superior da Região de confiança para  $\alpha = 20\%$

No capítulo 3, apresentamos a fórmula para cálculo da região, e com o auxílio do método PACITTI (1), para cálculo de matrizes, obtivemos os coeficientes  $C_{ij}$ . O anexo 4 traz os resultados obtidos, que transcrevemos nas páginas seguintes já inseridos na fórmula de cálculo.

(1) FORTRAN - Monitor, Pacitti, Tercio

. ATUALIZADORES

	V01	V05	V06	V07	V08
V01	0,205	- 0,152	0,078	- 0,035	- 0,022
V05	0,258	- 0,201	- 0,076	- 0,063	0,037
V06	- 0,160	0,111	0,025	0,058	- 0,013
V07	0,148	0,021	- 0,568	- 0,014	0,126
V08	- 0,200	0,127	0,229	0,004	0,061

. CALCULADORES

	V02	V05	V06	V07
V02	- 0,309	0,270	0,117	- 0,029
V05	0,185	- 0,092	0,031	- 0,046
V06	- 0,035	- 0,015	+ 0,058	+ 0,021
V07	0,099	- 0,040	- 0,169	0,076

. FORMATADORES

	V02	V03	V04	V05	V07
V02	- 2,840	2,190	- 0,214	0,204	0,036
V03	- 5,550	4,090	- 0,371	0,505	0,185
V04	6,100	- 4,580	0,402	- 0,233	- 0,162
V05	3,710	- 2,810	0,248	- 0,301	- 0,059
V07	9,150	- 6,700	0,770	- 1,020	- 0,410

. LISTADORES COM OPERAÇÃO

	V04	V05	V06	V07
V04	- 1,116	- 0,398	1,880	1,203
V05	0,174	0,402	0,680	0,190
V06	0,369	0,448	- 0,490	- 0,104
V07	1,419	0,008	- 1,602	- 0,982

• LISTADORES SIMPLES

	V04	V05	V06	V07
V04	- 0,325	0,273	- 0,122	0,204
V05	- 0,095	0,114	- 0,050	- 0,039
V06	0,136	- 0,114	0,025	- 0,011
V07	- 0,157	- 0,023	0,226	- 0,024

• ORGANIZADORES

	V01	V04	V06	V08
V03	0,066	- 0,091	0,038	0,023
V04	0,265	- 0,199	0,016	0,024
V05	- 0,059	0,087	- 0,030	- 0,032
V08	- 0,084	- 0,100	0,085	0,132

• CONSISTÊNCIA

	V01	V05	V06	V08
V01	0,296	0,926	0,753	- 2,450
V05	- 0,078	- 0,126	- 0,388	0,732
V06	0,022	- 0,143	- 0,077	0,308
V08	- 0,026	0,043	0,284	- 0,316

### Alocação de Custos

Os custos e despesas de um DAP Padrão podem alocar-se em:

- . Variáveis
  - . Mão-de-Obra
    - . Salários
    - . Encargos Sociais
    - . Seguros
    - . Bonificações
    - . Outros
  - . Equipamento - Testes
- . Fixos
  - . Equipamento - Perfuração/Digitação
  - . Aluguel
  - . Gerência
  - . Luz
  - . Água
  - . Telefone
  - . Material de Consumo

### Rateio dos Custos

- . Variáveis
  - . Mão-de-Obra - São rateados em 176 horas ou 22 dias trabalhados. Como se trata de orçamento as variações de mês para mês influirão pouco.
  - . Equipamento - Teste - Na bibliografia (1), a relação do custo de utilização do computador com o custo total de desenvolvimento é de 20%. No levantamento feito para avaliar a relação custo equipamento/custo total, encontrou-se o valor de 10%. Por razões de segurança na estimativa adotaremos o maior valor.

(1) Engenharia de Software, Aplicações e Recomendações  
Teixeira, Sergio Roberto Pinto.

. Fixos

- Equipamento - Perfuração/Digitação - o aluguel desse equipamento é rateado em

80% Programação

20% Análise

Critério: Amostragem de tempo de utilização.

- Aluguel - Pela relação de área ocupada pela Programação no imóvel.

- Gerência - Itens de mão-de-obra

- Luz - Rateio pela potência instalada

- Água - Rateio pelo nº de funcionários

- Telefone - Aluguel de telefones instalados mais despesas extras

- Materiais de Consumo - específico para cada departamento.

- Cartões

- Folhas de impressão

- Material de escritório

- Copia.

Os custos indiretos serão rateados por

nº de programadores x 176

assim

$$\left( \begin{array}{c} \text{Custo} \\ \text{Programação} \end{array} \right) = \left( \begin{array}{c} \text{Horas} \\ \text{Estimadas} \end{array} \right) \times \left[ \left( \begin{array}{c} \text{Mão de} \\ \text{Obra} \\ \text{Horária} \end{array} \right) + \left( \frac{\text{Custo Fixo}}{\text{nº progr.} \times 176} \right) + \left( \begin{array}{c} \text{Custo} \\ \text{Equip} \end{array} \right) \right]$$

como

$$\text{CUSTO EQUIPAMENTO} = 0,2 \text{ CUSTO TOTAL}$$

Pelos dados apresentados no capítulo 2, a relação de custo entre Análise e Programação é de 50% para cada uma, logo, hipótese razoável é que

$$\left( \begin{array}{c} \text{CUSTO} \\ \text{ANÁLISE} \end{array} \right) \approx \left( \begin{array}{c} \text{CUSTO} \\ \text{PROGRAMAÇÃO} \end{array} \right)$$

E

$$\begin{pmatrix} \text{CUSTO} \\ \text{TOTAL} \end{pmatrix} = \begin{pmatrix} \text{CUSTO} \\ \text{PROGRAM.} \end{pmatrix} + \begin{pmatrix} \text{CUSTO} \\ \text{ANÁLISE} \end{pmatrix} + \begin{pmatrix} \text{CUSTO} \\ \text{EQUIPAMENTO} \end{pmatrix}$$

VEM

$$\begin{pmatrix} \text{CUSTO} \\ \text{TOTAL} \end{pmatrix} = \left[ \begin{pmatrix} \text{CUSTO} \\ \text{PROGRAM.} \end{pmatrix} + \begin{pmatrix} \text{CUSTO} \\ \text{ANÁLISE} \end{pmatrix} \right] \times 1,25$$

$$\begin{pmatrix} \text{CUSTO} \\ \text{TOTAL} \end{pmatrix} \approx 2,5 \times \begin{pmatrix} \text{CUSTO} \\ \text{PROGRAMAÇÃO} \end{pmatrix}$$

O custo da Análise já está praticamente definido quando a definição do programa é entregue ao Departamento, faltando apenas acrescentar o tempo gasto com teste dentro do sistema, que é responsabilidade do analista.

#### 6.5- Avaliação da Produtividade

O quociente entre o tempo real de desenvolvimento e o tempo estimado corrigido só passou a ser usado quando o tempo estimado foi determinado com razoável confiabilidade. Até então, o índice que melhor refletia a produtividade era

Nº de linhas de Codificação

Homens/Dia

que foi criticado no capítulo 4

Aplicação

Programador: Júnior

Experiência: 1 ano

Qualificação: Regular

Tempo Desenvolvimento Estimado RT = 1 : 100 hs

Tempo Real de Desenvolvimento : 155

Pela tabela RT = 0,61

$$\text{Tempo Estimado Corrigido} \quad 100 \times \frac{1}{0,61} = 164 \text{ hs}$$

$$\varphi = \frac{\text{Tempo Real}}{\text{Tempo Estimado Corrig.}} = \frac{155}{164} = 0,94$$

No caso de manutenção, os valores da tabela refletem o histórico corrigido da empresa pelo fator de ritmo do programador. A análise é feita com o enquadramento ou não do tempo real da manutenção dentro da faixa de valores obtidos na matriz.

### Aplicação

Programador : Pleno

Experiência : 3 anos

Qualificação: Regular

### MANUTENÇÃO

SUBROTINA	CLASSE	$\bar{X}$	S	N
Cálculo	3	16,0	1,5	3

$$\alpha = 0,10$$

$$\begin{aligned}\bar{X}_{\max} &= \bar{X} + KS \alpha * S = \\ &= 16,0 + 1,69 \times 0,8 = 17,5\end{aligned}$$

$$\bar{X}_{\text{corr}} = \frac{\bar{X}_{\max}}{RT} = \frac{17,5}{1,04} = 16,8$$

$$\text{Tempo real} = 14,5$$

Dentro do limite estabelecido

## 7 - Análise de Resultados

### 7.1- Desenvolvimento de Programas

O núcleo deste trabalho, resume-se na obtenção das equações de regressão apresentadas em 6.4.1 e na tabela de categorias de manutenção obtida em 6.4.2. A análise dos primeiros, consiste no teste de hipóteses.

$$H_0 : \beta_i = 0$$

$$H_1 : \beta_i \neq 0$$

A disposição para análise de variância é a seguinte:

Fonte	Soma de Quadrados	Graus Liberdade	Quadrado Médio	F
Variação				
Devido à regressão	$\sum_{i=1}^n b_i S_{iy}$	n	$\frac{\sum_{i=1}^n b_i S_{iy}}{n} = Q$	$F = \frac{Q}{S_m^2}$
Residual	$SQ = S_{yy} - \sum_{i=1}^n b_i S_{iy}$	$K-n-1$	$S_m^2 = \frac{SQ}{K-n-1}$	
Total	$S_{yy}$	$K-1$		

onde

k - nº de programas da amostra

n - nº de variáveis na equação

Para rejeitar  $H_0$  :

$$F \geq F_{n1, n2, \alpha}$$

onde

$n_1$  - graus de liberdade da variação devido à regressão

$n_2$  - graus de liberdade da variação residual

$\alpha$  - nível de significância

### 7.1.1- Variáveis Objetivas

As variáveis V01 (nº de arquivos de entrada sem arquivo guia), V02 (nº arquivos de entrada com arquivo guia), V03 (nº arquivo de saída), V04 (nº de relatórios), assumem valores dentro de uma estreita faixa (de 0 a 5). Isto impede que tenha uma correlação acen tuada com a variável dependente como podemos observar nas matrizes de correlação do Anexo 2.

Para analisarmos a correlação  $r_{ij}$ , basta selecionar a coluna  $i$  e a linha  $j$ , ou vice-versa, dado que a matriz é simétrica. A variável V09 (tempo corrigido) é a variável dependente da equação de regressão, de modo que para saber-se os  $C_i$ , deve-se observar a coluna 09.

O campo SUBFILE fornece o tipo de programa a que a tabela se refere, segundo os códigos:

- . ATUAL Atualizadores
- . CALC Calculadores
- . ORGAN Organizadores
- . FORM Formatadores
- . CONST Consistência
- . LISTS Listadores Simples
- . LISTC Listadores Com Operação

A análise de multicolinearidade que ocorre quando duas ou mais variáveis independentes possuem alta correlação entre si, é feita automaticamente pelo procedimento STEPWISE. A multicolinearidade deve ser eliminada pois introduz viés nos resultados a serem obtidos, e para entendermos a sequência de inclusão das variáveis na equação, nos serviremos do Anexo 2 (tabela de correlações) e Anexo 3 (tabela final da regressão); neste último, a coluna VARIABLE, mostra a sequência de variáveis, MULTIPLE R, a correlação múltipla da equação com as variáveis incluídas até a linha correspondente; R SQUARE, coluna anterior elevada ao quadrado; B, coeficiente da variável na equação de regressão.

O procedimento consiste em incluir em primeiro lugar a variável independente que possua correlação mais alta com a dependente (como exemplo, analisaremos o sub-arquivo ATUAL). Na tabela de correlação, o maior valor que encontramos na coluna 09 é 0,88, correspondendo à variável V05, sendo esta a primeira variável a entrar para a equação. A segunda maior correlação com V09 é 0,84, correspondendo a V08; Para verificar se V08 não é suficientemente explicado por V05, analisa-se a correlação  $R_{5,8}$ , e encontramos 0,62 ( $R^2_{5,8} = 0,38$ ).

Sendo o coeficiente baixo, de modo a evitar a multicolinearidade, V08 é incluída na equação com R SQUARE = 0,92. O próximo valor mais alto na coluna V09 é 0,74, correspondendo a variável V06; Analisando sua correlação com V05, encontramos 0,87, de modo que pode ser suprimida da equação por ser suficientemente explicada por V05, mas com a diminuição no acréscimo em R SQUARE, poderá vir a ser incluída se não houver outra opção melhor. O próximo valor mais alto é 0,68 correspondendo a V07. Analisando sua correlação com as variáveis incluídas temos 0,43 e 0,61, justificando sua inclusão. Como vemos, a cada variável incluída, o procedimento torna-se mais complexo, sendo difícil ser feito manualmente, portanto, recorrendo ao auxílio do computador.

#### 7.1.2- Tipos de Programas

Para cada tipo de programa, foi feita a análise de variância de acordo com a entrada das variáveis, verificando se com o nível de significância  $H_0$  pode ser rejeitada. Para cada variável incorporada à equação, os resultados da análise de variância são apresentadas sinteticamente na tabela.

O nível de significância  $\alpha$ , foi adotado observando-se o seguinte critério

$$n \leq 10, \quad \alpha = 0,05$$

$$n > 10, \quad \alpha = 0,025$$

## ATUALIZADORES

VARIÁVEL INCLUÍDA	G L REGRESSÃO	G L RESIDUAL	F CALCULADO	F CRÍTICO	H <sub>0</sub>
V05	1	13	47,12	6,45	rejeitada
V08	2	12	67,85	5,10	rejeitada
V07	3	11	63,19	4,63	rejeitada
V06	4	10	45,32	4,47	rejeitada
V01	5	9	32,91	4,48	rejeitada

$$\alpha = 0,025$$

Amostra = 15

s<sub>e</sub> = 26,02

## Análise de Variância Final

Fonte de Variação	Soma Quadrados	Graus Liberdade	Quadrado Médio	F
Regressão	111.447	5	22.289	32,91
Residual	6.094	9	677	
Total	117.511	14		

$$F_{5,9, 0,025} = 4,48$$

## CONSISTÊNCIA

VARIÁVEL INCLUÍDA	G L REGRESSÃO	G L RESIDUAL	F CÁLCULO	F CRÍTICO	H <sub>0</sub>
V06	1	6	4,22	5,99	não rejeitada
V08	2	5	29,40	5,79	rejeitada
V01	3	4	20,52	6,59	rejeitada
V05	4	3	16,50	9,12	rejeitada
-	-	-	-	-	-

$$\alpha = 0,05$$

Amostra : 8

$$s_e = 29,45$$

## Análise de Variância Final

Fonte de Variação	Soma Quadrados	Graus Liberdade	Quadrado Médio	F
Regressão	57.268	4	14.317	16,50
Residual	2.602	3	867	
Total	59.870	7		

$$F_{4,3, \ 0,05} = 9,12$$

## CALCULADORES

VARIÁVEL INCLUÍDA	G L REGRESSÃO	G L RESIDUAL	F CALCULADO	F CRÍTICO	H <sub>0</sub>
V06	1	13	8,20	6,45	rejeitada
V05	2	12	7,40	5,10	rejeitada
V07	3	11	12,21	4,63	rejeitada
V04	4	10	8,37	4,47	rejeitada
V02	5	9	6,23	4,48	rejeitada

$$\alpha = 0,025$$

Amostra 15

s<sub>e</sub> 26,35

Análise de Variância Final

Fonte de Variação	Soma Quadrados	Graus Liberdade	Quadrado Médio	F
Regressão	23.447	5	4.689	6,23
Residual	6.769	9	752	
Total	30.216	14		

$$F_{5,9,0,025} = 4,48$$

## ORGANIZADORES

VARIÁVEL INCLUÍDA	G L REGRESSÃO	G L RESIDUAL	F CALCULADO	F CRÍTICO	H <sub>0</sub>
V06	1	18	28,75	5,98	rejeitada
V04	2	17	27,39	4,62	rejeitada
V03	3	16	23,40	4,69	rejeitada
V07	4	15	25,49	3,80	rejeitada
-	-	-	-	-	-

$$\alpha = 0,025$$

Amostra 20

s<sub>e</sub> 19,93

## Análise de Variância Final

Fonte de Variação	Soma Quadrados	Graus Liberdade	Quadrado Médio	F
Regressão	40.518	4	10.129	25,49
Residual	5.960	15	379	
Total	46.478	19		

$$F_{4,15,0,025} = 3,80$$

## FORMATADOR

VARIÁVEL INCLUÍDA	G L REGRESSÃO	G L RESIDUAL	F CALCULADO	F CRÍTICO	H <sub>0</sub>
V05	1	12	17,10	6,55	rejeitada
V04	2	11	12,81	5,26	rejeitada
V07	3	10	13,34	4,83	rejeitada
V03	4	9	9,73	4,72	rejeitada
V02	5	8	6,97	4,82	rejeitada

$$\alpha = 0,025$$

Amostra 14

s<sub>e</sub> 11,02

## Análise de Variância Final

Fonte de Variação	Soma Quadrados	Graus Liberdade	Quadrado Médio	F
Regressão	4.236	5	877	6,97
Residual	972	8	121	
Total	5.208	13		

$$F_{5,8,0,025} = 4,82$$

## LISTADORES SIMPLES

Variável Incluída	G.L. Regressão	G.L. Residual	F Calculado	F Crítico	H <sub>0</sub>
V04	1	14	35,69	6,30	rejeitada
V07	2	13	23,17	4,97	rejeitada
V05	3	12	15,76	4,47	rejeitada
V06	4	11	11,07	4,28	rejeitada
-	-	-	-	-	-

$$\alpha = 0,025$$

Amostra = 16

s<sub>e</sub> = 6,81

## Análise de Variância Final

Fonte de Variação	Soma Quadrados	Graus Liberdade	Quadrado Médio	F
Regressão	2.057	4	514	11,07
Residual	510	11	46	
Total	2.567	15		

$$F_{4,11,0,025} = 4,28$$

## LISTADORES COM OPERAÇÃO

VARIÁVEL INCLUÍDA	G L REGRESSÃO	G L RESIDUAL	F CALCULADO	F CRÍTICO	H <sub>0</sub>
V04	1	8	6,87	5,32	rejeitada
V07	2	7	4,75	4,74	rejeitada
V05	3	6	4,68	4,76	não rejeitada
V06	4	5	6,53	5,19	rejeitada
-	-	-	-	-	-

$$\alpha = 0,05$$

Amostra 10

s<sub>e</sub> 12,84

## Análise de Variância Final

Fonte de Variação	Soma Quadrados	Graus Liberdade	Quadrado Médio	F
Regressão	3.609	4	902	6,53
Residual	690	5	138	
Total	4.299	9		

$$F_{4,5,0,05} = 5,19$$

. Os listadores com operação possuem uma variação muito grande no tempo de confecção do programa, de corrente do fato das rotinas (exceto a de listagem) serem muito variadas quanto à sua função.

## 7.2- Manutenção de Programas

### 7.2.1- Localização da Manutenção

A definição da sub-rotina em que deverá ser feita a alteração não apresenta problemas maiores.

Todas as alterações testadas puderam enquadrar-se em uma das linhas da tabela, mas como poderemos verificar, a divisão das sub-rotinas fez com que o número de alterações analisadas para cada uma delas ficou reduzido, prejudicando a confiabilidade do resultado.

Com um levantamento de Programação, até a liberação para Análise, fornecerá uma tabela mais apurada.

### 7.2.2- Classes de Dificuldade

A categorização da classe de dificuldade de uma manutenção, é uma tarefa desempenhada pelo chefe da Programação. Embora subjetiva, foi a melhor solução encontrada, dado que a correlação com variáveis objetivas não deu resultados satisfatórios. Conforme dissemos no capítulo 4, este tipo de avaliação pode ser orientado com estabelecimento de padrões, e para cada categoria, a faixa de variação deve ser pequena.

Para que estas duas condições fossem satisfatórias ou seja, para cada categoria, o padrão de avaliação pudesse ser simples e a faixa de variação fosse pequena, o número de classes de manutenção foi relativamente alto, prejudicando ainda mais a confiabilidade dos resultados obtidos.

À margem destas observações, construímos uma tabela de intervalos de confiança para as médias obtidas no capítulo 6.

$$\alpha = 0,05$$

$$eo = t_{n-1}; 0,025 \cdot \frac{s_x}{\sqrt{n}}$$

DIFÍCULDADE SUB-ROTEIRO	1	2	3	4	5
Leitura	0,5	0,7	1,5	2,0*	3,5
Consistência	0,5*	1,5*	2,5	3,0*	4,0
Organizador	0,5	1,0	1,5	3,0	4,5
Cálculo	0,5	1,0	2,0	3,0	4,5
Atualização	0,5*	1,0	1,5	3,0	4,0
Gravação	0,5	0,5	1,0	2,0*	2,0
Impressão	0,5	0,5	1,0	2,0*	2,5
Processamentos Diversos	0,5	1,5	2,5	3,0*	4,0
Programas Paralelos	0,7	1,5	2,5	5,5	/ / / /

\* Valor estimado dado que  $n = 0$

Quando a alteração se dá em duas sub-rotinas o tempo estimado será a soma das médias da respectiva classe de dificuldade ou seja

$$\bar{x}_c = \bar{x}_A + \bar{x}_B$$

O desvio padrão da soma é dado por

$$s_c = \sqrt{\frac{s_A^2}{n_A} + \frac{s_B^2}{n_B}}$$

dado que consideramos a manutenção nas diversas subrotinas como Variáveis Aleatórias Independentes.

### 7.3- Aplicações

#### 7.3.1- Desenvolvimento de Programas

Apresentamos alguns orçamentos de programas novos, com o objetivo de testar a validade das equações obtidas e aplicá-las à prática. Podemos notar, nos exemplos, que pode ocorrer tanto super estimação, quanto sub-estimação, e os efeitos negativos de um e outro ou seja, perda do cliente ou prejuízo no faturamento devem ser minimizados. O orçamento pelo limite superior do intervalo de confiança com  $\alpha/2 = 0,10$  foi uma das formas que gerou melhores resultados.

O custo da análise foi orçado com base na hipótese na igualdade de custos entre os dois departamentos, conforme discutimos no capítulo 2, hipótese esta possível de crítica. Os custos e margens de lucro apresentados são fictícios.

PROGRAMA : 2412

TIPO : Calculador

OBJETIVO : Distribuição e acumulação de movimentos

V01	Arq. Entrada	= 2
V02	Arq. Entrada + guia	= 2
V03	Arq. Saída	= 1
V04	Relatórios	= 0
V05	Reg. Entrada	= 3
V06	Reg. Saída	= 3
V07	Chaves UPSI	= 0
V08	Movimentos	= 2

PROGRAMADOR : 380

EXPERIÊNCIA : 2 anos

QUALIFICAÇÃO: Bom

### Equação

$$Y = 3,1 V02 + 15,4 V05 + 10,0 V06 + 14,6 V07 - 16,3$$

$$Y = 3,1 \times 2 + 15,4 \times 3 + 10,0 \times 2 + 14,6 \times 0 - 16,3$$

$$Y = 6,2 + 46,2 + 30,0 + 14,6 - 16,3$$

$$Y = 66,1$$

$$Y_{corr} = \frac{66,1}{0,92} = 73,4$$

Intervalo de Confiança = 9,2 horas

$$Y^* = Y_{corr} + 9,2 = 82,6$$

Custo/hora Funcionário 380 = 356,00

### Custo Programação

$$1,25 \times 82,6 \times 358,00 = 36.963,00$$

$$\text{Custo Análise} = 35.000,00$$

$$\text{Total} \qquad \qquad \qquad 71.963,00$$

$$\text{Lucro } (x 0,15) \qquad \qquad \qquad 10.794,00$$

$$\qquad \qquad \qquad 82.758,00$$

Horas Reais = 93,0 horas

PROGRAMA : 3340

TIPO : Listador com operação

OBJETIVO : Listar carnês e resumo de operações

V01	Arq. Entrada	=	2
V02	Arq. Entrada + guia	=	2
V03	Arq. Saída	=	0
V04	Relatórios	=	3
V05	Reg. Entrada	=	2
V06	Reg. Saída	=	12
V07	Chaves UPSI	=	1
V08	Movimentos	=	0

PROGRAMADOR : 5194

EXPERIÊNCIA : 1 ano e 4 meses

QUALIFICAÇÃO: Regular

$$Y = 1,2 V04 + 7,7 \times V05 + 3,8 V06 \\ + 4,9 V07 + 19,02$$

$$Y = 1,2 \times 3 + 7,7 \times 2 + 3,8 \times 12 \\ + 4,0 \times 1 + 19,02$$

$$Y = 87,52$$

$$Y_{corr} = \frac{87,5}{0,73} = 120,0$$

Intervalo de Confiança = 7,2 horas

$$Y^k = Y_{corr} + 7,2 = 127,2 \text{ horas}$$

Custo/hora Funcionário 5192 = 332,00

## Custo Programação

1,25 x 127,2 x 332,00 = 52.788,00

Custo Análise = 55.000,00

Total 107.788,00

Lucro ( x 0,15 ) 16.168,00

123.956,00

Tempo Real = 121,0 horas.

PROGRAMA : 3321

TIPO : Formatador

OBJETIVO : Conversão de lay-out para compatibilizar  
com P 3202

V01 Arq.Entrada = 1

V02 Arq.Entrada + guia = 1

V03 Arq.Saída = 1

V04 Relatório = 0

V05 Reg.Entrada = 3

V06 Reg.Saída = 3

V07 Chaves UPSI = 0

V08 Movimento = 4

PROGRAMADOR : 5194

EXPERIÊNCIA : 1 ano e 6 meses

QUALIFICAÇÃO : Regular

$$Y = 0,9 V02 + 3,4 V03 + 14,5 V04 + 13,0 V05 \\ + 5,5 V07 - 8,6$$

$$Y = 0,9 \times 1 + 3,4 \times 1 + 14,5 \times 0 + 13,0 \times 3 \\ + 5,5 \times 0 - 8,6$$

$$Y = 34,7$$

$$Y_{corr} = \frac{y}{0,78} = 44,5$$

Intervalo de confiança = 3,4 horas

$$Y^* = Y_{corr} + 3,4 = 47,9$$

Custo/hora funcionário 5194 = 332,00

Custo Programação

$$1,25 \times 47,9 \times 332,00 = 19.878,00$$

$$\text{Custo Análise} = 25.000,00$$

$$\text{Total} = 44.878,00$$

$$\text{Lucro (x 0,15)} = 6.731,00$$

$$51.610,00$$

Tempo Real = 39,5 horas

### 7.3.2- Manutenção de Programas

A principal finalidade do uso da tabela, não é orçar a manutenção, mas sim de organizar a carteira de solicitação do Departamento de Programação, de modo que apresentamos o orçamento da manutenção, e cumprimento ou não do prazo previsto.

Para o orçamento, segue-se o mesmo critério para desenvolvimento de programas.

Se em um mesmo programa, a manutenção envolve mais de uma sub-rotina, temos que

$$\bar{x}_T = \sum_{i=1}^k \bar{x}_i$$

$$s_T = \sum_{i=1}^k \frac{s_i^2}{n_i}$$

$$e_0 = t \sum_{i=1}^k n_i - k, \alpha/2 \cdot s_T$$

onde

$k$  - nº de sub-rotinas

$n$  - amostra

- Manutenção de 1 sub-rotina no 2410

PROGRAMA : 2410

PROGRAMADOR: Júnior

EXPERIÊNCIA: 1 ano

QUALIFICAÇÃO: Bom

Sub-rotina	Classe	$\bar{x}$	$e_0$	$\bar{x}_{max}$
Gravação	2	2,0	0,5	2,5

$$\frac{\bar{x}_{max}}{RT} = 3,5 \text{ horas}$$

Custo/hora = 320,00

Custo Programação = 320, 3,5 = 1.120,00

Custo Análise = 1.500,00

Custo total 2.620,00

$\bar{x}$  real = 4,0 hs.

. Manutenção de três sub-rotinas no 3355

PROGRAMA : 3355

PROGRAMADOR: Pleno

EXPERIÊNCIA: 3,5 anos

QUALIFICAÇÃO: Regular

Sub-rotinas	Classe	$\bar{x}$	S	n
Atualização	2	5,0	1,2	7
Cálculo	2	8,0	1,0	6
Impressão	3	7,0	0,8	4

$$\bar{x} = 6,0 + 7,0 + 8,0 = 20,0$$

$$e_0 = t_{17-3, 0,025} \cdot \sqrt{\frac{1,2^2}{7} + \frac{1,0}{6} + \frac{0,8^2}{4}}$$

$$= 2,14 \times 0,54 = 1,2$$

$$\bar{x}_{\max} = 21,2 \text{ horas}$$

$$\bar{x}_{\max \text{ corrigido}} = \frac{21,2}{1,04} = 20,4$$

Custo/hora = 370,00

Custo Programação = 20,4 x 370,00 = 7.548,00

Custo Análise = 11.000,00

18.548,00

$\bar{x}$  real = 18,5 hs.

## 8 - Conclusões

Em termos de estimativa de tempos em programação, o método apresentado, tanto de desenvolvimento quanto de manutenção do programas, fornecem resultados bons, permitindo à Empresa melhorar o controle de solicitações ao D.A.P.

Duas observações cabem serem feitas aqui. A primeira, com respeito ao desenvolvimento de programas, é a grande dependência do programa orçado ter ou não relatórios. Podemos notar pelos coeficientes da equação V04, que assume valores altos e por consequência apresenta uma grande sensibilidade no resultado, além disto, o número de registros de saída, pela ausência de relatório, também diminui. A causa básica é o fato de amostra ter sido pequena e "viciada" neste aspecto.

Note-se que conceitualmente os coeficientes devem ser positivos, pois a cada aumento em uma das variáveis, a dificuldade do desenvolvimento deverá aumentar, portanto aceitaremos apenas os coeficientes positivos da regressão.

A segunda observação refere-se à tabela de manutenções. A confiabilidade dos resultados ficou prejudicada pela amostra ter sido pequena, ou seja, o mesmo ponto de crítica para o desenvolvimento, porém foi o melhor conjunto de dados, tanto em um caso como em outro, que pudemos reunir no tempo disponível.

Quanto ao aspecto orçamentário, no que diz respeito ao Departamento de Programação, ficou satisfatoriamente resolvido. O problema surge na estimativa do custo no Departamento de Análise para cada programa. A hipótese adotada de igualdade de custos nos dois departamentos é criticada pela maioria dos Gerentes, mas o resultado apresentado nos primeiros capítulos, foi confirmado pelo levantamento de dados na Lojicred Processamento de Dados, ou seja os custos totais dos dois departamentos dividem-se em

53 % Análise

47 % Programação

Lógicamente há variações nesta proporção de programa para programa, e uma aproximação pela média é bastante grosseira. Este ponto fica em aberto para dimensões, e o seu desenvolvimento é tema

para um outro trabalho. Por conveniência, no programa de cálculo do custo do desenvolvimento deixamos este valor opcional entre a adoção do mesmo custo no Departamento de Programação e um outro valor estimado pela Diretoria.

A avaliação da produtividade dos programadores cumpriu seu objetivo ao fornecer dados para o Gerente de Programação sobre o cumprimento ou não do cronograma estipulado, tanto para desenvolvimento de programas quanto para manutenção, conforme os critérios de avaliação apresentados. Pôde-se assim, indicar parâmetros sobre o desempenho do funcionário e discutir sobre premiação e penalidades.

9 - Bibliografia

FELDMANN, Paulo Roberto, Como Estimar Prazos Certos em Programação, São Paulo, Compucenter.

SZWEDA, Ralph, Information Processing Management, Princeton, Averbach Publishers Ind.

BRANDON, Dick H., Management Standards for Data Processing, New York, Van Nostran Reinhold Company.

BROOKS, Fred, The Mythical Man-Month: Essays on Software Engineering, Mass., Addison Wesley

TEIXEIRA, Sergio Roberto Pinto, Engenharia de Software: Aplicações e Recomendações, São Paulo, Edgard Blücher.

BARNES, Ralph M., Estudo de Movimentos e de Tempos: Projeto e Medida do Trabalho, São Paulo, Edgard Blücher

COSTA NETO, Pedro Luis de O., Estatística, São Paulo, Edgard Blücher

DUNN, Olive J., CLARCK, Virginia A., Applied Statistics, Analysis of Variance and Regression, U.S.A., John Wiley.

Anexo 1 - Relatórios do Sistema Atual de  
Controle de atividades

A.1.1 Relatório Analítico

A.1.2 Relatório de Totais por Chapa

A.1.3 Relatório de Totais por Chapa



RELATÓRIO SERV-EXECUT. P/CHAPA LOJIPLAN CONTROLES & SISTMAS LTDA.							FOLHA NO. 01		
DATA 30/04/79									
CONTROLE DE SERVIÇOS EXECUTADOS									
*	DP SET	CHAPA	CAT	O.S.	DP	PROGRAMA	TEMPO		
						SISTEMA I	GASTO - I		
						HORAS	ESTIMADO S		
						VALOR I QTDÉ.	ESTIMADO S		
						MINUTOS	ESTIMADO S		
						VALOR	ESTIMADO S		
						FOLHAS	ESTIMADO S		
						VALOR	ESTIMADO S		
						I COD.	ESTIMADO S		
						DESCRIÇÃO			
01 010 00091	TOTAL	CHAPA	160,0	15.200,00	36	186	1.967,02	1.015	116,78
01 010 00383	TOTAL	CHAPA	160,0	15.200,00	28	147	1.538,40	650	74,77
01 010 00385	TOTAL	CHAPA	160,0	15.200,00	25	69	722,17	491	56,52
01 010 00390	TOTAL	CHAPA	160,0	15.200,00	8	28	293,06	302	34,74
01 010 00448	TOTAL	CHAPA	160,0	15.200,00	11	72	753,56	437	50,26
01 010 00478	TOTAL	CHAPA	160,0	15.200,00	15	36	376,79	655	75,37
01 010 00480	TOTAL	CHAPA	160,0	15.200,00	33	136	1.423,40	1.038	119,42
01 020 00003	TOTAL	CHAPA	160,0	30.240,00	11	60	627,96	229	26,35
01 020 00072	TOTAL	CHAPA	160,0	38.240,00	1	33	345,30	230	26,45
01 020 00100	TOTAL	CHAPA	160,0	38.240,00	43	353	3.694,49	3.238	372,47
01 020 00146	TOTAL	CHAPA	159,0	38.001,00	10	77	805,86	1.277	146,87
01 020 00424	TOTAL	CHAPA	160,0	38.240,00	10	68	711,69	714	82,14
01 020 00443	TOTAL	CHAPA	125,7	30.052,30	7	14	1.407,53	475	54,63

\*\* LOJIPLAN CONTROLES &amp; SISTEMAS LTDA \*\*\*

RELATÓRIO SERV-EXECUT. P/LIJPLAN SERVIÇO		FOLHA N.º. 02
CONTROLE DE SERVIÇOS EXECUTAÇÕES		DATA 30/04/79
<b>* DP SET_UHAPA CAT_D-S. GP_PFLGRNAL TEMPO GASTO I TESTES E RETRATADO OS SISTEMA'S VALOR I SITUAÇÃO VALOR I QIDE. MINUTOS VALOR FOLHAS DESCRIÇÃO</b>		
<b>* HORA'S HORAS</b>		
01 010	00625	
TOTAL O. SERV.	22,0	2.090,30
01 020	00601	
TOTAL O. SERV.	47,5	11.352,50
01 020	00602	
TOTAL O. SERV.	38,9	9.297,10
01 020	00604	
TOTAL O. SERV.	346,4	82.769,60
01 020	00605	
TOTAL O. SERV.	43,5	10.396,50
01 020	00608	
TOTAL O. SERV.	36,0	8.604,00
01 020	00609	
TOTAL O. SERV.	8,3	1.983,70
01 020	00610	
TOTAL O. SERV.	293,2	70.074,80
01 020	00612	
TOTAL O. SERV.	36,0	8.604,00
01 020	00621	
TOTAL O. SERV.	10,3	2.461,70
01 020	00825	
TOTAL O. SERV.	58,0	13.862,00
01 020	00826	
<b>* LJIPLAN CONTROLES E SISTEMAS UTA</b>		

Anexo 2 - Tabela de Correlações

A.2.1 Atualizadores

A.2.2 Calculadores

A.2.3 Organizadores

A.2.4 Formatadores

A.2.5 Consistência

A.2.6 Listadores Simples

A.2.7 Listadores Com Operação

10/30/80 PAGE 8

FILE PRGRM CREATION DATE = 10/30/80  
SUBFILE ATUAL > VARIAVEIS INDEPENDENTES

## CORRELATION COEFFICIENTS

A VALUE OF 99.00000 IS PRINTED  
IF A COEFFICIENT CANNOT BE COMPUTED.

	V01	V02	V03	V04	V05	V06	V07	V08	V09
V01	1.000000	0.32026	-0.33691	0.37272	-0.10483	-0.01049	0.42308	0.12029	0.09925
V02	0.32026	1.00000	0.06376	-0.07053	0.26368	0.27667	0.72058	0.59917	0.46364
V03	-0.33691	0.06376	1.00000	-0.19562	0.68348	0.57237	0.23737	0.44303	0.60019
V04	0.37272	-0.07053	-0.19562	1.00000	-0.06349	0.36051	-0.1907	-0.37658	0.17605
V05	-0.10483	0.26368	0.68348	-0.06349	1.00000	0.86868	0.43569	0.62350	0.88532
V06	-0.01049	0.27667	0.57237	0.36051	0.86868	1.00000	0.30429	0.44771	0.74495
V07	0.42306	0.72058	0.23737	-0.19907	0.43569	0.30429	1.00000	0.61970	0.67638
V08	0.12029	0.59917	0.44303	-0.37658	0.62350	0.44771	0.61970	1.00000	0.83923
V09	0.09925	0.46364	0.60019	-0.17605	0.88532	0.74495	0.67638	0.83923	1.00000

11/03/60

PAGE 2

FILE PRGRM CREATION DATE = 11/03/60  
SUBFILE CALC

) VARIAVEIS INDEPENDENTES

## CORRELATION COEFFICIENTS

A VALUE OF 99.00000 IS PRINTED  
IF A COEFFICIENT CANNOT BE COMPUTED.

	V01	V02	V03	V04	V05	V06	V07	V08	V09
V01	1.00000	99.00000	99.00000	99.00000	99.00000	99.00000	99.00000	99.00000	99.00000
V02	99.00000	1.00000	0.18235	0.07675	-0.04886	0.28937	-0.05011	99.00000	0.16117
V03	99.00000	0.18235	1.00000	0.44087	0.10163	0.35719	-0.08529	99.00000	0.17972
V04	99.00000	-0.07675	0.4087	1.00000	-0.27124	0.41345	0.21539	99.00000	0.28196
V05	99.00000	-0.04886	0.10163	-0.27124	1.00000	-0.43669	-0.34529	99.00000	0.15730
V06	99.00000	0.28937	0.35719	0.41345	0.35669	1.00000	0.36269	99.00000	0.62268
V07	99.00000	-0.05011	-0.08529	0.21539	-0.34529	0.36289	1.00000	99.00000	0.55218
V08	99.00000	99.00000	99.00000	99.00000	99.00000	99.00000	1.00000	99.00000	1.00000
V09	99.00000	0.16117	0.17972	0.28196	0.15730	0.62268	0.55218	99.00000	1.00000

PAGE 27

10/30/80

FILE PRGRM (CREATION DATE = 10/30/80 ) VARIATEIS INDEPENDENTES  
 SUBFILE ORGAN

## CORRELATION COEFFICIENTS

A VALUE OF 99.00000 IS PRINTED  
 IF A COEFFICIENT CANNOT BE COMPUTED.

	V01	V02	V03	V04	V05	V06	V07	V08	V09
V01	1.00000	0.76066	-0.38482	0.38540	0.41326	-0.03676	0.61980	0.05550	0.20563
V02	0.76066	1.00000	-0.15729	0.32502	0.11844	0.19072	0.38001	0.23508	0.26190
V03	-0.38482	-0.15729	1.00000	-0.39281	0.36324	-0.00729	-0.28721	-0.06656	0.02056
V04	0.38540	0.32502	-0.39281	1.00000	-0.13286	0.51973	0.11035	0.62444	0.73644
V05	0.43326	0.11844	-0.36324	-0.13286	1.00000	-0.23069	0.08776	-0.37716	-0.08140
V06	-0.03676	0.19072	-0.00729	0.51973	-0.23069	1.00000	-0.06602	0.87010	0.78424
V07	0.61980	0.38001	-0.628721	0.11035	0.06776	-0.06602	1.00000	-0.01462	0.18836
V08	0.05550	0.23508	-0.66556	0.62444	-0.37716	0.87010	-0.01462	1.00000	0.79443
V09	0.20563	0.26190	0.02056	0.73644	-0.08140	0.78424	0.18836	0.79443	1.00000

10/30/80 PAGE 69

FILE PRGRM CREATION DATE = 10/30/80  
SUBFILE FORM

) VARIAVEIS INDEPENDENTES

## CORRELATION COEFFICIENTS

A VALUE OF 99.00000 IS PRINTED  
IF A COEFFICIENT CANNOT BE COMPUTED.

	V01	V02	V03	V04	V05	V06	V07	V08	V09
V01	1.00000	0.89758	-0.29995	0.07255	0.53901	0.19418	-0.27001	99.00000	0.44513
V02	0.89758	1.00000	-0.18054	0.07763	0.62828	0.09415	-0.25003	99.00000	0.54381
V03	-0.29995	-0.18054	1.00000	0.20672	0.13646	0.12450	0.05919	99.00000	0.34577
V04	0.07255	0.07763	0.20672	1.00000	-0.25634	0.43866	-0.11452	99.00000	0.12697
V05	0.53991	0.62828	0.13846	-0.25634	1.00000	0.14663	-0.30539	99.00000	0.76659
V06	0.19418	0.09415	-0.12450	0.43866	0.14663	1.00000	-0.03632	99.00000	0.14979
V07	-0.27001	-0.25003	0.05919	-0.11452	-0.30539	-0.03832	1.00000	99.00000	-0.00569
V08	99.00000	99.00000	99.00000	99.00000	99.00000	99.00000	1.00000	99.00000	1.00000
V09	0.44513	0.54381	0.34577	0.12697	0.76659	0.14979	-0.00569	99.00000	1.00000

PAGE 53  
10/30/80FILE PRGRM CREATION DATE = 10/30/80      ) VARIATEIS INDEPENDENTES  
SUBFILE CONST

## CORRELATION COEFFICIENTS

A VALUE OF 99.00000 IS PRINTED  
IF A COEFFICIENT CANNOT BE COMPUTED.

	V01	V02	V03	V04	V05	V06	V07	V08	V09
V01	1.00000	0.07881	C*07881	99.00000	0.21591	0.60168	0.10426	*0.24635	0.39800
V02	0.07881	1.00000	-0.14286	99.00000	-0.10690	-0.43830	-0.18898	-0.01941	-0.36318
V03	0.07881	-0.14286	1.00000	99.00000	0.00000	0.13251	0.94491	0.71356	C*16111
V04	99.00000	99.00000	99.00000	1.00000	99.00000	99.00000	99.00000	99.00000	99.00000
V05	0.23591	-0.10694	0.00000	99.00000	1.00000	0.55682	-0.17678	0.14529	0.63674
V06	0.60168	-0.43830	0.13251	99.00000	0.55682	1.00000	-0.01348	-0.19809	0.64268
V07	0.10426	-0.18698	0.94491	99.00000	-0.17678	-0.01348	1.00000	0.10273	0.01084
V08	-0.24635	-0.01941	0.21356	99.00000	0.14529	-0.19809	0.10273	1.00000	0.57173
V09	0.39800	-0.36318	0.16111	99.00000	0.63674	0.64268	0.01084	0.57173	1.00000

11/03/80

PAGE 7

FILE PRGRM (CREATION DATE = 11/03/80 ) VARIAVEIS INDEPENDENTES  
 SUBFILE LISTS

## CORRELATION COEFFICIENTS

A VALUE OF 99.00000 IS PRINTED  
 IF A COEFFICIENT CANNOT BE COMPUTED.

	V01	V02	V03	V04	V05	V06	V07	V08	V09
V01	1.00000	99.00000	99.00000	99.00000	99.00000	99.00000	99.00000	99.00000	99.00000
V02	99.00000	1.00000	99.00000	0.16364	0.15467	0.11704	0.17201	99.00000	0.19558
V03	99.00000	99.00000	1.00000	99.00000	99.00000	99.00000	99.00000	99.00000	99.00000
V04	99.00000	0.16364	99.00000	1.00000	0.34028	0.84215	0.48327	99.00000	0.84752
V05	99.00000	0.15467	99.00000	0.34028	1.00000	0.46142	-0.12542	99.00000	0.31918
V06	99.00000	0.11704	99.00000	0.84215	0.46142	1.00000	0.24050	99.00000	0.71131
V07	99.00000	0.17201	99.00000	0.48327	0.12542	0.24050	1.00000	99.00000	0.62876
V08	99.00000	99.00000	99.00000	99.00000	99.00000	99.00000	1.00000	99.00000	1.00000
V09	99.00000	0.19558	99.00000	0.84752	0.31918	0.71131	0.62876	99.00000	1.00000

FILE PRGRM CREATION DATE = 10/30/80 ) VARIATEIS INDEPENDENTES  
SUBFILE LISTC

CORRELATION COEFFICIENTS

A VALUE OF 99.00000 IS PRINTED  
IF A COEFFICIENT CANNOT BE COMPUTED.

	V01	V02	V03	V04	V05	V06	V07	V08	V09
V01	1.00000	0.72786	99.00000	-0.07538	-0.18699	0.03711	0.030552	99.00000	0.16139
V02	0.72786	1.00000	99.00000	-0.13927	0.00000	-0.04572	-0.02352	99.00000	0.01164
V03	99.00000	99.00000	1.00000	99.00000	99.00000	99.00000	99.00000	99.00000	99.00000
V04	-0.07538	-0.13927	99.00000	1.00000	-0.15504	0.70008	0.53831	99.00000	0.67995
V05	-0.18699	0.00000	99.00000	-0.15504	1.00000	-0.45803	-0.36657	99.00000	0.11066
V06	0.03711	-0.04572	99.00000	0.70008	-0.45803	1.00000	0.46253	99.00000	0.62090
V07	0.030552	-0.02352	99.00000	0.53831	-0.36657	0.46253	1.00000	99.00000	0.64928
V08	99.00000	99.00000	99.00000	99.00000	99.00000	99.00000	1.00000	99.00000	99.00000
V09	0.01164	0.01164	0.67995	0.11066	0.62090	0.64928	99.00000	1.00000	1.00000

Anexo 3 - Tabelas Finais da Regressão

A.3.1 Atualizadores

A.3.2 Calculadores

A.3.3 Organizadores

A.3.4 Formatadores

A.3.5 Consistência

A.3.6 Listadores Simples

A.3.7 Listadores Com Operação

```
FILE PROGRAM CREATION DATE = 10/20/00 , 10:11AM  
SUBFILE ATUAL  
***** MULTIPLE REGRESSION *****  
VARIABLE LIST 1  
REGRESSION LIST 2
```

18/39/80 PAGE 16

DEPENDENT VARIABLE

SUMMARY TABLE

VARIABLE	MULTIPLE R	R SQUARE	RSC CHANGE	SIMPLE R	B	BETA
V05	0.88532	0.78379	0.78379	0.88532	14.63989	0.49504
V08	0.95852	0.91876	0.13497	0.83923	10.61470	0.36442
V07	0.97219	0.94516	0.02640	0.67638	12.74265	0.19573
V06	0.97351	0.94773	0.00257	0.74495	2.68451	0.09225
	0.97373	0.94815	0.00042	0.09925	6.61679	0.02541
					-4.90820	

### CONSTANT



10/30/00 PAGE 34

10/30/00

PROGRAM (CREATION DATE = 10/30/00 ) VARIAVEIS INDEPENDENTES

REGRESSION LIST

DEPENDENT VARIABLE 89

## SUMMARY TABLE

VARIABLE	MULTIPLE R	R SQUARE	RSQ CHANGE	SIMPLE R	B	BETA
V06	0.78424	0.61504	0.61504	0.78424	4.75204	0.50592
V04	0.87361	0.76320	0.14816	0.73644	26.93653	0.57189
V03	0.90247	0.91445	0.05125	0.02056	11.57147	0.32092
	0.93368	0.87176	0.05731	0.18836	5.46238	0.25087
					-9.99929	

CONSTANT)

10/30/80 PAGE 73

FILE PROGRAM CREATION DATE = 10/30/80 ) VARIOUS INDEPENDENTES  
SUBFILE FORM )  
DEPENDENT VARIABLE.. v09  
\* \* \* \* \* MULTIPLE REGRESSION \* \* \* \* \*  
VARIABLE LIST 1  
REGRESSION LIST 1

## SUMMARY TABLE

VARIABLE	MULTIPLE R	R SQUARE	RSQ CHANGE	SIMPLE R	B	BETA
V05	0.76659	0.58766	0.58766	0.76659	1.358910	0.90719
V04	0.83646	0.69966	0.41199	0.12697	14.65810	0.36412
V07	0.89452	0.80016	0.10050	-0.04669	54.68715	0.31740
V03	0.90130	0.81234	0.01218	0.34577	3.37057	0.13501
V02	0.90187	0.81337	0.00103	0.54381	0.86685	0.04931
(CONSTANT.)					-6.57058	

10/30/80 . PAGE 56

SUMMARY TABLE

VARIABLE	MULTIPLE R	R SQUARE	RSE CHANGE	SIMPLE R	B	BETA
V06	0.64268	0.41304	0.64268	0.57751	10.77865	0.57751
V08	0.96002	0.92165	0.50861	0.57173	12.56942	0.70715
V01	0.96902	0.93900	0.01736	0.39800	26.68284	0.16490
V05	0.97802	0.95653	0.01753	0.63674	4.013039	0.16880
					-6.208237	

PAGE 10  
11/03/89

#### SUMMARY TABLE

BETA	E	H	
MULTIPLE R	R SQUARE	RSD CHANGE	SIMPLE R
0.84752	0.71830	0.84752	1.441566
0.88373	0.78097	0.62876	4.44585
0.89312	0.79766	0.01668	1.50360
0.89506	0.80113	0.00347	0.51754
			18.09218

CONSTANT

PAGE 21

FILE PRGRM (CREATION DATE = 10/30/00 ) VARIAVEIS INDEPENDENTES  
SUBFILE LISTC.

\*\*\*\*\* MULTIPLE REGRESSION \*\*\*\*\*  
\*\*\*\*\* VARIABLE LIST 1 \*\*\*\*\*  
\*\*\*\*\* REGRESSION LIST 1 \*\*\*\*\*

INDEPENDENT VESTABILISATION 809

#### SUMMARY TABLE

VARIABLE	MULTIPLF R	R SQUARE	RSG CHANGE	SIMPLE R	B	BETA
V004	0.467995	0.46234	0.46234	0.67995	1.24203	0.04792
V007	0.75849	0.57531	0.1297	0.64928	4.95361	0.56580
V005	0.83709	0.70071	0.1540	0.11066	7.72477	0.60067
V006	0.91621	0.83943	0.13872	0.62090	3.83372	0.60078
					19.02946	

Anexo 4 - Cálculo de Orçamentos

PROGRAMA 7777  
CATEGORIA ATL2  
CODIGO 6 V A N & A V E T I S

V01	ARC. ENTRADA	2
V02	ARC.ENTRADA+GUIA	2
V03	ARC. SAIDA	2
V04	FELATORIO	1
V05	PEG. ENTRADA	2
V06	PEG SAIDA	3
V07	CHAVES UPSI	4
V08	MOVIMENTOS	3

PRUGRAMADUM 4.5  
EXPERIENCIA 1.7  
QUALIFICACAO 2

PRAZO PAGAR	100.00
PRAZO CORRIDO	131.00
INTERVALO SEGURANCA	25.00
PRAZO P/ CANCELAR	150.00
CUSTO HORA FUNC	415= 336.00

RECLAMANTES  
 CUSTO PRODUCAO 65092.00  
 CUSTO ANALISE -----  
 CUSTO TOTAL 125092.00  
 LUCRO (X 0.15) 18763.00  
 FATURAMENTO 143855.00

## PROGRAMA ?7777

## CATEGORIA URGEZ

CODIGO	V A N A V I T I S	
V01	ARC. ENTRADA	1
V02	ARC. ENTRADA+GUIA	2
V03	ARC. SAIDA	4
V04	FELATÓRIO	6
V05	FIC. ENTRADA	4
V06	REC. SAIDA	4
V07	CHAVES UPSI	2
V08	MOVIMENTOS	6

## PROGRAMADOR 7.0

## EXPERIENCIA 2.0

## QUALIFICACAO 1

PRAZO PADRAO	75.0
PRAZO CONFIABIL	82.0
INTERVALO SEGURANCA	12.0
PRAZO P/ CALCULO	95.0
CUSTO HORA FUNC /16	372.00

## URGAMENTO

CUSTO PROGRAMACAO	44263.27
CUSTO ANALISE	-----
CUSTO TOTAL	68526.54
LUCRO (X 0.25)	13278.00
FATURAMENTO	16804.54

PROJETO 77777	
CATEGORIA	VALOR
CÓDIGO	3
VARIAVEIS	
V01	ARC. ENTRADA
V02	ARC. ENTRADA+SAIDA
V03	ARC. SAIDA
V04	RELATORIO
V05	REG. ENTRADA
V06	REG. SAIDA
V07	CHAVES UPS!
V08	MOVIMENTOS
	PROGRAMADAS 200
	EXPERIENCIA 1,5
	QUALIFICACAO 2
PRAZO PAUKAU	9500
PRAZO CORRIGIR	15100
INTERVALO SEGURANCA	1440
PRAZO P/ CALCULO	15104
CUSTO HORA FUNC	562,00
	354,00
VALORES	
CUSTO PROGRAMACAO	66228,93
CUSTO ANALISE	20000,00
CUSTO TOTAL	66228,93
LUCRO (x 0,15)	67434,00
FATURAMENTO	633602,93