

ESTUDOS DE PERFIS DE DENSIDADE E TEMPERATURA ELETRÔNICA
ATRAVÉS DA ANÁLISE DE DADOS DE FOGUETES

Daniela Cristina Santana

Aluna da Universidade do Vale do Paraíba - Bolsa PIBIC / CNPq

Orientador: Dr. Mangalatayhl Ali Abdu, Pesquisador, Chefe da
Divisão de Aeronomia

A densidade e temperatura eletrônica da Ionosfera, N_e e T_e , são propriedades desse meio, cita-se ainda campo elétrico e campo magnético, que podem ser estudadas pela teoria magneto-iônica a qual relaciona as suas características com as características das ondas que se propagam nesse meio, tais como frequência da onda, vetor da onda, polarização e fase.

Nesta região a densidade de íons e elétrons é suficientemente alta de modo a interferir na propagação de ondas de rádio, sendo essa característica mais acentuada em 350 quilômetros de altitude como pode ser observada na figura 01 que é um perfil teórico obtido para a Campanha Guará pelo modelo IRI90. O modelo IRI90 é um modelo internacional de referência ionosférica empírico padrão baseado em observações experimentais com o objetivo de ser um modelo confiável e realista dessas propriedades. Na prática, para esse trabalho, essas propriedades foram estudadas através de análises sob os perfis de N_e preliminares obtidos por lançamentos de foguetes da Campanha Tapirei, realizada em 29 de abril de 1991, às 14:37hs(LT), sendo o apogeu em 441 quilômetros de altitude e 682 segundos de tempo total de dados coletados, da Campanha Guará realizada em 14 de outubro 1994, às 19:55hs(LT), sendo o apogeu em 956 quilômetros de altitude e 1053 segundos de tempo total de dados coletados e mais recentemente a Campanha Piri-açu cuja trajetória ainda não está disponível. De todas as campanhas, foram coletados dados digitalizados por carga úteis, que são mecanismos (sensores) desenvolvidos específicos sobre determinadas ocorrências físicas, no caso N_e e T_e . Esses dados foram processados passo a passo de forma semelhante em cada campanha devido ao próprio modo como foram coletados e implementados em linguagem C, devido as facilidades e as vantagens que a mesma oferece. Os resultados de todos os passos foram visualizados pelo aplicativo Grapher for Windows da Microsoft e o resultado final, o perfil da N_e e T_e preliminar de cada campanha, foram comparados com os respectivos perfis teóricos obtidos pelo modelo IRI90 pelo orientador.

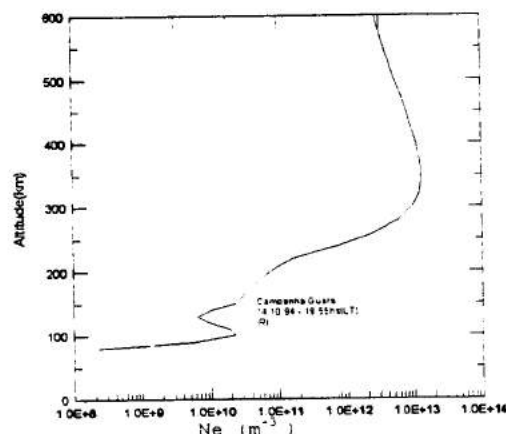


figura 01

**PIBIC/CNPq - PROGRAMA INTEGRADO DE BOLSAS
DE INICIAÇÃO CIENTÍFICA**

RELATÓRIO FINAL

**ESTUDOS DE Ne E Te ELETRÔNICA ATRAVÉS DA ANÁLISE
DE DADOS DE FOGUETES**

INPE
ÓRGÃO: DAE
ORIENTADOR: DR. POLINAYA MURALIKRISHNA
DR. MANGALATAHYL ALI ABDU

BOLSISTA: DANIELA CRISTINA SANTANA

- SÃO JOSÉ DOS CAMPOS -
- JUL./1996 -

INPE - INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

**PIBIC - PROGRAMA INSTITUCIONAL DE BOLSAS A INICIAÇÃO
CIENTÍFICA**

BOLSISTA - DANIELA CRISTINA SANTANA

RELATÓRIO FINAL

- JUL./1996 -

RESUMO

O relatório aqui apresentado tem a finalidade de expor o trabalho realizado durante o período de agosto/94 ao atual momento. Esse trabalho tem por objetivo principal obter mais perfis da Densidade Eletrônica Numérica da Ionosfera através dos dados coletados pela Campanha TAPIREÍ realizada em abril de 1989, pela Campanha GUARÁ realizada em outubro de 1994 e pela Campanha PIRI-AÇU realizada em dezembro de 1995, todas com o apoio do IAE/CTA - Instituto de Aeronáutica e Espaço / Centro Técnico Aeroespacial - situado em São José dos Campos/SP .

AGRADECIMENTOS

- ao Engenheiro **MÁRCIO NOGUEIRA BARBOSA** diretor do Instituto Nacional de Pesquisas Espaciais;

- ao Dr. **MANGALATAYHL ALI ABDU**, chefe do depto. de Aeronomia/CEA do Instituto Nacional de Pesquisas Espaciais;

- ao Dr **POLINAYA MURALIKRISHNA**, do depto. de Geofísica Espacial/CEA do Instituto Nacional de Pesquisas Espaciais;

- aos companheiros diários de trabalho **LÚCIA TERRA DE ALMEIDA LIMIRO**, **VALESCA DE OLIVEIRA BARBOSA**, **LETÍCIA TEXEIRA CONTTINI**, **FÁTIMA ALVES USIFATTI** e em especial a **JONAS DE SOUZA RODRIGUES** e **MARIA GORETI DOS SANTOS AQUINO** que sempre incentivaram e apoiaram esse trabalho;

- ao CNPq pelo apoio financeiro concedido através de bolsas.

SUMÁRIO

I- Introdução	08
I.1 - Ionosfera	08
I.2 - Irregularidades da Ionosfera	09
I.3 - A importância da Densidade Eletrônica	10
I.4 - Os equipamentos de Observações	10
I.4.1 - Carga útil	11
I.4.1.1 - Descrição dos Experimentos	12
II- A escolha da Linguagem C	14
III- Campanha TAPIREÍ	15
III.1 - Obtenção dos dados	15
III.2 - Processamento dos dados	15

IV- Campanha GUARÁ	18
IV.1 - Obtenção dos dados	18
IV.2 - Processamento dos dados	19
V- Campanha PIRI-AÇU	23
V.1 - Obtenção dos dados	23
VI- Conclusão	24
VI.1 - Perfil preliminar da Campanha TAPIREÍ MODO C4	24
VI.2 - Perfil preliminar da Campanha TAPIREÍ MODO CT	25
VI.3 - Perfil preliminar da Campanha GUARÁ MODO1	26
VI.4 - Perfil preliminar da Campanha GUARÁ MODO2	27
VI.5 - Trajetória do foguete da Campanha PIRI-AÇU	28
VII- Bibliografia	66

LISTA DE FIGURAS

fig. 01 - perfil teórico da Ne feito pelo modelo IRI90_____	08
fig. 02 - perfil teórico da Te feito pelo modelo IRI90_____	09
fig. 03 - perfil teórico da Campanha GUARÁ feito modelo IRI90____	18
fig. 04 - perfil em fase preliminar da Campanha GUARÁ modo1____	21
fig. 04.1 - perfil da linha base da Campanha GUARÁ modo1_____	21
fig. 05 - perfil em fase preliminar da Campanha GUARÁ modo2____	21
fig. 05.1- perfil da linha base da Campanha GUARÁ modo2_____	21
fig. 06 - perfil preliminar da Campanha Tapireí modo C4_____	25
fig. 07 - perfil preliminar da Campanha Tapireí modo Ct_____	26
fig. 08 - perfil preliminar da Campanha Guará modo1_____	27
fig. 09 - perfil preliminar da Campanha Guará modo2_____	28
fig. 10 - trajetória do foguete da Campanha Piri-açu_____	29

APÊNDICE

- i.1 - Listagens dos programas implementados
 - i.1.1 - Campanha Tapireí
 - i.1.2 - Campanha Guará
- i.2 - Distribuição dos módulos componentes da carga útil IONEX
- i.3 - Ilustração do diagrama de blocos simplificado da carga útil IONEX
- i.4 - Ilustração do diagrama de blocos do comutador PCM
- i.5 - Tabela de formatação dos dados da carga útil IONEX

I- INTRODUÇÃO

I.1 - IONOSFERA

A densidade eletrônica numérica da ionosfera, N_e , e a temperatura eletrônica, T_e , são duas das propriedades eletrônicas desse meio, cita-se ainda campo elétrico e campo magnético. Essas propriedades podem ser estudadas pela teoria magneto-iônica, a qual relaciona as suas características com as características das ondas que se propagam nesse meio tais como frequência da onda, vetor da onda, polarização e fase.

Nesta região a densidade de íons e elétrons é suficientemente alta de modo a interferir na propagação de ondas de rádio. É essa interferência na propagação de ondas de rádio que permite esse estudo da ionosfera através dessa teoria.

Na prática para esse trabalho, essas propriedades são estudada através de perfis obtidos pelo processamento de dados coletados por diversos equipamentos disponíveis (vide item I.4), apresentando-se semelhantemente como nas figuras abaixo que são perfis teóricos feitos pelo modelo IRI90:

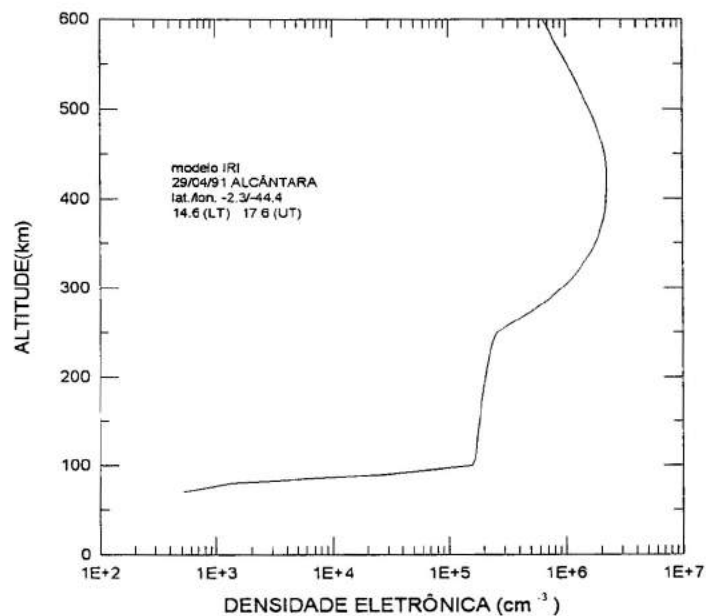


fig. 01 - exemplo de um perfil teórico da N_e feito pelo modelo IRI90.

Altitude(km)

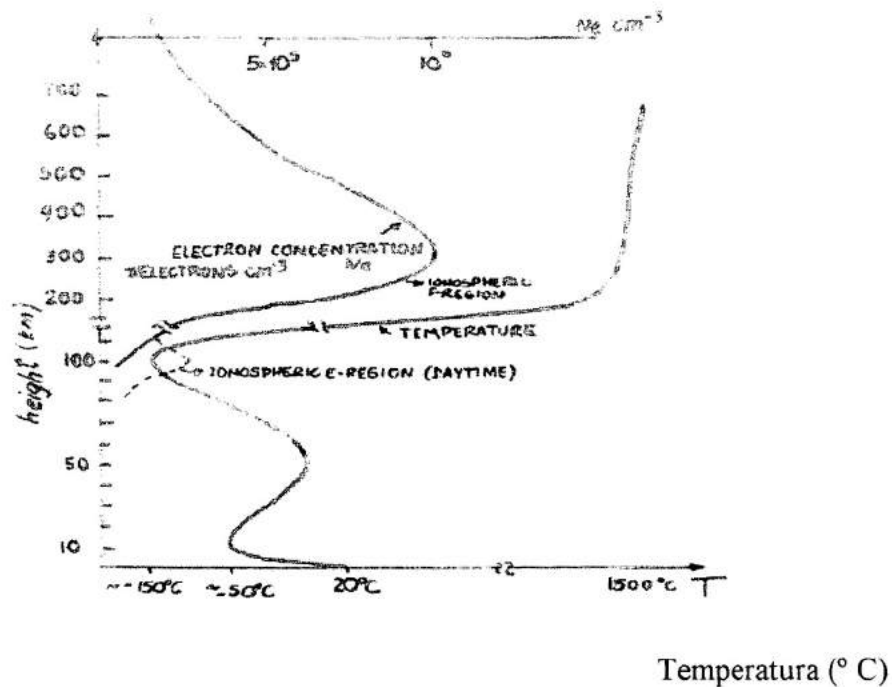


fig. 02 - exemplo de um perfil teórico da Te feito pelo modelo IRI90.

Uma das principais características da ionosfera que pode ser observada na fig. 01 é que com a elevação da altitude da Ionosfera, aproximadamente em 350km, é observado um aumento gradativo no número de elétrons livres. Acima desse ponto, nesse mesmo número, existe um decréscimo gradativo de elétrons livres. Isto se dá como uma consequência da absorção máxima das radiações solares nessa altitude. Já na fig. 02, pode ser visualizado o perfil da Te que diminui ao afastar-se da Terra (que atua como uma fonte), aumenta devido a camada de ozônio, continua a diminuir com a elevação da altitude e aumenta novamente também devido a absorção solar ser máxima nessa altitude de aproximadamente 350km.

O modelo IRI (Internacional Reference Ionospheric) é um modelo empírico padrão baseado em observações experimentais. Possui a finalidade de produzir uma representação padrão, confiável e realista das mais importantes propriedades ionosféricas, tais como densidade eletrônica, temperatura de íons e composição iônica (Bilitza, 1986). Esse modelo foi criado em 1960, pela URSI (Internacional Union of Radio Science) e o COSPAR (Committee on Space Research). Com o passar do tempo, já sofreu várias correções sempre com a intenção de aprimorar a representação média global dessas propriedades. A última versão do modelo que se dispõem aqui é de 1990, por isso o denominamos de IRI 90.

I.2 - IRREGULARIDADES DA IONOSFERA

Devido a sua complexidade e variabilidade, a ionosfera necessita ser monitorada continuamente através de intensa atividades de medidas e estudos de modo a fornecer um quadro claro da dinâmica dessa região. Dos fenômenos que ocorrem, pode-se citar o Spread-F e o Eletrojato Equatorial que são também conhecidos como irregularidades ionosféricas:

SPREAD-F: é também conhecido na literatura e no meio científico como “bolhas” ionosféricas. O que se sabe hoje, é que as “bolhas” ionosféricas apresentam concentração mais baixas de íons e se formam em baixas latitudes (principalmente na região do equador magnético) e interferem nas comunicações devido a brusca alteração no número de elétrons da região. Quando há um aumento da densidade, a “bolha” não deixa que o sinal de rádio emitido passe através do seu meio, ou seja, a frequência emitida de comunicação entrará na “bolha” e ficará circulando indefinidamente.

ELETROJATO EQUATORIAL: vem a ser um fluxo de corrente intensificado gerado pelo “impulso” dado aos íons pelos ventos neutros (movimento global das partículas não ionizadas) que corresponde a uma fina camada (≈ 1 km) com corrente de 100 A por cada 100 km na direção norte-sul. O fator de assimetria que possibilita o movimento de íons e elétrons em direções opostas (para gerar correntes) é o campo magnético da Terra.

I.3 - A IMPORTÂNCIA DA DENSIDADE ELETRÔNICA

Vários perfis da densidade eletrônica numérica da ionosfera já foram obtidos pela comunidade científica interessada em estudar os fenômenos ocorridos em seu meio e solucionar as interferências geradas. Como exemplo pode-se citar as interferências geradas nas comunicações via sinal de rádio, já que a sua capacidade de refletir sinais eletromagnéticos na faixa de alta frequência (2 a 30 Mhz) permite a comunicação entre dois pontos remotos na superfície da Terra. Um outro exemplo, é o sistema GPS (Global Positioning System), o qual permite via recepção de sinais de satélites em VHF ou UHF, a localização de determinado objeto no espaço ou na superfície terrestre dentro de uma certa resolução, que depende da recepção e do conhecimento da ionosfera. Esse sistema GPS é muito usado em navegação aérea, marítima e na localização de veículos de frotas.

I.4 - EQUIPAMENTOS DE OBSERVAÇÕES

Para monitorar continuamente a ionosfera é necessário descrever rapidamente alguns dos equipamentos disponíveis para a obtenção dos dados como:

IONOSSONDA: é um equipamento que transmite sinais de rádio-frequência, com uma varredura de frequência de 1 a 20Mhz, que são refletidos nas camadas ionosféricas e recebido de volta. O sinal recebido é processado e apresentado em forma de um gráfico denominado ionograma, que representa o perfil da densidade eletrônica das camadas ionosféricas.

DIGISSONDA: possui o mesmo princípio de funcionamento da ionossonda, mas é um equipamento digital mais moderno e preciso.

POLARÍMETRO: o equipamento recebe o sinal polarizado linearmente, que atravessa as camadas ionosféricas e é transmitido por um satélite geo-estacionário. A medida da fase e a amplitude desse sinal permite o estudo da variação integral da densidade eletrônica das camadas, bem como das perturbações (variações irregulares da densidade) que causam o espalhamento do sinal enviado pelo satélite.

CARGA ÚTIL: são sensores devidamente projetados acoplados em foguetes que quando expostos no Plasma Ionosférico medem os campos elétricos e a densidade eletrônica "in loco". Essa técnica será descrita mais detalhadamente adiante pois foi a utilizada nesse trabalho.

I.4.1 - CARGA ÚTIL

Apresentando uma melhor explicação sobre cargas úteis pode-se conceituar que são mecanismos (sensores) desenvolvidos para "colhetarem" dados específicos sobre determinadas ocorrências físicas (fenômenos que apresentam grande interesse para os pesquisadores), que tentam minimizar os problemas e melhorar os estudos da ionosfera e ainda que servem de suporte tecnológico para o desenvolvimento de satélites.

O funcionamento das cargas úteis se dá com seus mecanismos (sensores) acoplados aos foguetes que por sua vez possuem uma trajetória estabelecida com tempo determinado para obtenção dos dados basicamente através do seguinte esquema:

entrada de dados	→	medição	→	saída de dados
------------------	---	---------	---	----------------

A carga útil denominada IONEX foi desenvolvida pelos Grupos de Alta Atmosfera e Ionosfera do INPE, com o objetivo científico de estudar as irregularidades da ionosfera noturna equatorial além de permitir a realização de medidas de "housekeeping" e de monitorar o funcionamento dos dispositivos mecânicos para avaliação dos projetos realizados com vistas em futuros desenvolvimentos tecnológicos. Para isso foi composta por:

- uma Sonda Dupla de Campo Elétrico(CE);
- uma Sonda de Plasma Pulsada(PP);
- uma Sonda Capacitiva de Alta Frequência(HFC);
- um Fotômetro de Luminescência Atmosférica;
- um sistema de suprimento de energia;
- um comutador PCM e
- diversos sensores de temperatura e "housekeeping".

A estrutura mecânica das duas baias de instrumentação e a ogiva ejetável foram construídas pelo INPE, por meios de firmas externas, a partir dos desenhos fornecidos pelo IAE/CTA que receberam as modificações para atender às necessidades específicas dos experimentos.

I.4.1.1 - DESCRIÇÃO DOS EXPERIMENTOS

SONDA DUPLA DE CAMPO ELÉTRICO: o objetivo desse experimento é realizar medidas do campo elétrico ionosférico associado às irregularidades de "bolhas" de plasma ionosférico. São realizadas medidas de componentes CA e CC do campo elétrico entre a altura de 70km e o apogeu. Os sensores de CE estão montados nas extremidades de um par de hastes que serão desdobradas logo após a ejeção da ogiva. Será medida a diferença de potencial de plasma entre duas esferas sensoras separadas entre si por uma distância de três metros.

SONDA DE PLASMA PULSADA : a Sonda de Plasma Pulsada (SPP) é utilizada para medir a densidade, a temperatura e as irregularidades dos elétrons na ionosfera. O dispositivo funciona através da determinação da corrente entre o plasma e um sensor metálico de forma esférica, em função do potencial aplicado ao sensor. O sensor é uma esfera de cavidade com diâmetro de 55mm, revestido por uma camada de ouro e montado na extremidade de uma haste desdobrável.

SONDA CAPACITIVA DE ALTA FREQUÊNCIA : o objetivo desta sonda é medir a densidade eletrônica e determinar a frequência de colisão dos elétrons na ionosfera. A técnica consiste na determinação da capacitância de um sensor esférico imerso no plasma, que permite estudar a densidade eletrônica, uma vez que a capacitância é função da constante dielétrica do plasma, que depende da densidade(concentração) de elétrons.

O sensor - uma esfera e cavidade igual à da SPP - também é montado na extremidade de uma haste.

FOTÔMETRO DE LUMINESCÊNCIA ATMOSFÉRICA : a finalidade do fotômetro de oxigênio 6300 é medir a intensidade da emissão no comprimento de onda de 630,0nm (OI 6300) ao longo da trajetória do foguete. Esta medida permite determinar:

- (1) a altura e o perfil da camada de emissão;
- (2) a variação espacial da intensidade da emissão.

Esses dados complementam as medidas das densidades de íons e do campo elétrica, fornecendo a escala de grandeza da irregularidade da ionosfera durante o voo.

COMUTADOR PCM : é o módulo responsável pela aquisição, conversão e formatação dos dados. Os reguladores de tensão das baterias, o temporizador de dois canais (para comandar a ejeção da ogiva na altura de 60km e a calibração do fotômetro no apogeu) e o freqüencímetro do fotômetro também são montados dentro deste módulo.

Há dois tipos de comutador PCM(Pulse Coded Modulation):

- (1) - PCM/AM - modulado em amplitude
- (2) - PCM/FM - modulado em frequência

Como já dito é o módulo responsável pela aquisição, conversão e formatação dos dados, fornecendo o sinal serial codificado em NRZ-BI- ϕ -L , a uma taxa de 50000 bits por segundo ao modulador e transmissor. O módulo foi configurado para 32 canais analógicos e 10 palavras digitais de 10 bits. O diagrama de blocos do comutador PCM é apresentado no apêndice - item i.4, a formatação dos dados é apresentado no item i.5.

A carga útil IONEX é composta ainda por: uma baia de instrumentação, uma baia porta ogiva, um sistema de quatro hastes desdobráveis (das quais duas são telescópicas) que suportam os sensores ionosféricos, uma ogiva ejetável, uma baia de instrumentação (IAE/CTA) e uma baia porta-carga (IAE/CTA), que totalizam duas baias de instrumentação, uma baia porta-ogiva, uma baia porta-carga e ogiva ejetável(vide apêndice - item i.2 e i.3).

II- A ESCOLHA DA LINGUAGEM C

A escolha da linguagem C se deve as vantagens que a mesma oferece para esse tipo de processamento sequenciais, como por exemplo, ser uma linguagem de médio nível combinando elementos da linguagem de alto nível, visivelmente mais elaborada e compreensível, com a funcionabilidade de uma linguagem de baixo nível, acessando diretamente a máquina. Outro exemplo, é a “compartimentalização” do programa, ou seja, a habilidade de seccionar e mascarar o resto do programa de todas as informações necessárias para se realizar uma tarefa específica do própria programa. Ainda um outro exemplo, é a característica de ser uma linguagem estruturada permitindo a declaração de funções e procedimentos facilitando a manutenção periódica do programa por completo. Por essa particular característica a linguagem C se tornou mais aceita, a mais utilizada entre os programadores e a mais “portátil”, tornando possível adaptar um programa escrito para um tipo de máquina a outra máquina com hardware semelhante, assim um mesmo programa pode ser executado em todas as máquinas compatíveis.

III- CAMPANHA TAPIREÍ

III.1 - OBTENÇÃO DOS DADOS

Para se obter mais esse perfil foi utilizado dados coletados pela carga útil no lançamento do foguete do dia 29 de abril de 1991, às 17:37:41hs(U.T.) da base de lançamento brasileira em Alcântara(MA) sendo 682 segundos de tempo total de dados coletados, atingindo o apogeu em 441km de altitude.

Esses dados estavam disponíveis em duas formas dependendo do tempo em questão:

0.0s à 544.0s - disponíveis em fitas magnéticas fornecidas cordialmente pelo IAE/CTA - Instituto de Aeronáutica e Espaço/ Centro Técnico Aeroespacial - ambos situados em São José dos Campos;

544.0s à 682.0s - os dados foram lidos manualmente da carta gráfica fornecida pelo orientador.

III.2 - PROCESSAMENTO DOS DADOS

Após a obtenção dos dados de 0.0s à 682.0s, esses foram processados passo a passo por 06(seis) programas implementados com a explicação do orientador. No caso desta campanha, devido a disposição dos dados, esses foram processados de 02(duas) formas diferentes inicialmente: de 0.0s à 544.0s foram "filtrados" com a finalidade de se ter um valor máximo médio e um tempo médio referente a cada dado uniformizando-se com os dados de 544.0s à 682.0s. O termo "filtrados" foi usado porque os dados em discos estavam de 3 à 4 vezes repetidos para cada dado equivalente da carta gráfica. A disposição final dos dados ficou uniforme da seguinte maneira: os dados foram "divididos" em 05(cinco) grupos sendo respectivamente chamados de C1, C2, C3, C4 e CT. O dado C4 é essencial pois seria a "soma aritmética" dos dados anteriores ($C1 + C2 + C3$) enviado pela própria carga útil e o dado CT seria também a "soma aritmética" dos dados anteriores ($C1+C2+C3$) mas implementados manualmente dentro do programa chamado "tap01.c"(vide apêndice) . Portanto, teoricamente os valores dos dados de C4 e CT deveriam ser iguais, o que não ocorre na prática, sendo visivelmente detectado esse "erro" nos perfis finais da Ne obtidos isoladamente para ambos.

Após essa distinção entre os dados de C4 e CT, ignorou-se os dados relativos a C1, C2 e C3, como será descrito sequencialmente abaixo:

i- os arquivos de dados devidamente “descodificados”

como os dados coletados foram limitados de uma maneira a enviar um pulso no valor máximo analógico de 3.5volts, independente da altitude do foguete e, conseqüentemente, do tempo decorrido do lançamento do foguete, foi necessário então fazer suaves adaptações. Essas adaptações consistem em:

1 - se o valor do dado atual for 80% mais alto que o valor do dado anterior, soma-se uma constante “c” para C4 e uma constante “t” para CT;

2 - se o tempo equivalente desse dado atual for inferior ou igual a 50% do tempo total do experimento, significando que o foguete esteja na subida, incrementa-se a constante “c” de 3.5 (C4) e a constante “t” de 65535.0(CT);

- se o tempo equivalente desse dado atual for superior a 50% do tempo total do experimento, significando que o foguete esteja na descida, decrementa-se a constante “c” de 3.5(C4) e a constante “t” de 65535.0(CT). Desse modo, os dados foram “descodificados” totalmente, de 0.0s à 682.0s, através da implementação de um segundo programa chamado “tap02.c” (vide apêndice) para os dados de C4 e um terceiro programa chamado “tap03.c”(vide apêndice) para os dados de CT juntamente.

ii - o arquivo de dados relativos a altitude

como os arquivos gerados foram dispostos na forma primeira coluna relativa ao tempo e a segunda coluna relativa ao dado em toda a sua extensão, e o perfil da Ne é dado em frequência por altitude, ou frequência por altura, foi necessário então “gerar” o arquivo de alturas em função do único parâmetro em comum existente entre ambos: o próprio tempo. Através de um formulário fornecido pelo orientador que continha as alturas do foguete de 1 segundo em 1 segundo, gerou-se o arquivo com as alturas manualmente chamado “tapalt.dat “ pelo recurso do Norton Editor e implemtou-se a fórmula abaixo, também fornecida pelo orientador, em um programa chamado “tap04.c”(vide apêndice) para C4 e “tap05.c”(vide apêndice) para CT, que interpolava os tempos dos dados com os tempos das alturas, lembrando que o apogeu foi em 441km.

fórmula: $h = th_i + (((th_f - th_i) / (tdf - tdi)) * (td - tdi))$ onde

h: altura desejada;

thi: tempo inicial da altura disponível;

thf: tempo final da altura disponível;

tdi: tempo inicial da faixa do dado;

tdf: tempo final da faixa do dado;

td: tempo do dado atual.

iii - o arquivo de dados relativos as frequências com as alturas equivalentes aos dados que já estão “descodificados”, torna-se necessário ainda que se calcule as frequências relativas a esses dados através dos seguintes cálculos abaixo, também fornecidos pelo orientador por um programa chamado “tap06.c”(vide apêndice) para os dados C4 e “tap07.c” (vide apêndice) para os dados de CT que os implementa :

$$f01 = ((393214.0 + ((1023.0 * 2.591) / 3.5)) / 0.04);$$

$$f02 = ((393214.0 + ((1023.0 * 2.653) / 3.5)) / 0.04);$$

$$a = (f02 - f01) / (682.0 - 0.0905);$$

$$b = (f01 - (a * 0.0405));$$

$$f0 = ((a * \text{tempo do dado atual}) + b);$$

$$f = (\text{valor do dado atual} + 393214.0);$$

$$df = (f0 - f);$$

e, finalmente,

$$\text{frequência} = 0.242369 * df * f \quad \text{onde}$$

frequência: densidade eletrônica numérica da ionosfera (Ne).

obs: é necessário ressaltar que todos as constantes envolvidas no cálculo e, o próprio cálculo, foram fornecidos pelo orientador provindas de laboratórios.

IV- CAMPANHA GUARÁ

IV.1 - OBTENÇÃO DOS DADOS

A Campanha GUARÁ foi um trabalho de pesquisas científicas realizadas entre o Estados Unidos representados pela NASA - National Aeronautics Space Administrations - e outras entidades de pesquisas norte americanas, e o Brasil, representado pelo INPE - Instituto Nacional de Pesquisas Espaciais - e também outras instituições de pesquisas brasileiras, no período de 15 de agosto a 20 de outubro de 1994.

Foram ao todo 33 (trinta e três) lançamentos de foguetes para o estudo do Eletrojato Equatorial, Eletrodinâmica do Pôr do Sol e do Spread F.

Para esse trabalho foram utilizados os dados coletados pela carga útil do lançamento do foguete do dia 14 de outubro de 1994 às 19:55hs (L.T.) na base de lançamento brasileira em Alcântara(MA) sendo 1053s de tempo total de dados coletados, atingindo o apogeu em 956km de altitude.

Esses dados estavam disponíveis totalmente em 14(quatorze) arquivos, 14 (quatorze) discos de 3 ½", de 0.0s à 1053s fornecidas cordialmente pelos pesquisadores norte americanos, e apresenta um perfil teórico feito pelo modelo IRI (vide item I) abaixo:

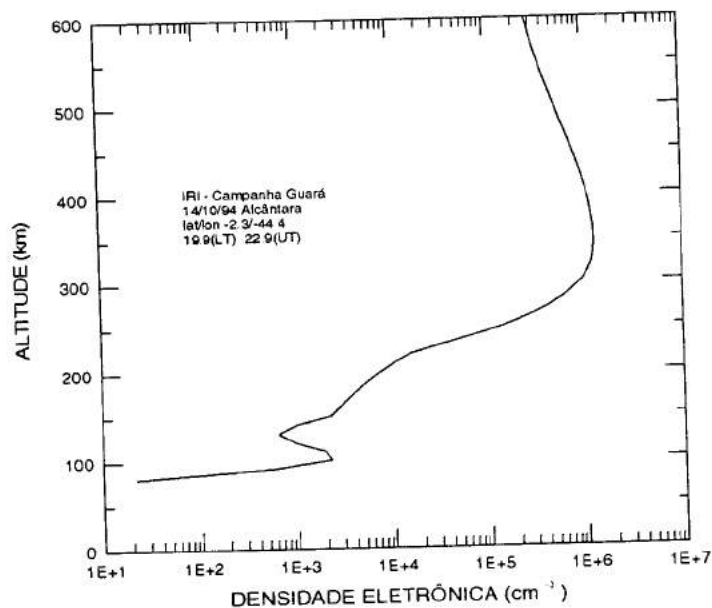


fig.03 - perfil teórico feito pelo modelo IRI90

IV.2 - PROCESSAMENTO DOS DADOS

Após a obtenção dos dados, esses foram processados por vários programas implementados passo a passo com a explicação do orientador sendo seu processamento semelhante em alguns pontos em relação ao processamento feito aos dados da Campanha Tapireí. Isto é necessário devido a diferença do mecanismo de coleta de dados projetados na carga útil de cada foguete.

Para o tempo total de dados disponíveis, 0.0s à 1053.0s, foi executado um primeiro programa chamado "gua01.c" (vide apêndice), com a finalidade de "filtrar" todos os arquivos para se ter um valor máximo médio e um tempo médio referente a cada dado.

Esse termo "filtrar" significa que foi feita uma média aritmética com todos os valores referentes aos dados e foi usado com a mesma conotação do item III.2. Então, continua ainda os 14 (quatorze) arquivos distintos, porém menores em tamanho.

O segundo programa, chamado "gua02.c", foi responsável por "dividir" os dados de todos os arquivos em HIGH FREQUENCY e LOW FREQUENCY, ou como foi denominado nesse trabalho, modo1 e modo2. Assim tem-se que obter um perfil da Ne para cada modo em questão.

Após essa distinção entre modo1 e modo2, uni-se os 14 arquivos de modo1 em um único arquivo e os 14 arquivos de modo2 também em um único arquivo, ficando assim com um arquivo para modo1 - HIGH FREQUENCY, e um arquivo para modo2 - LOW FREQUENCY. Isto foi feito através dos recursos do DOS, como na Campanha Tapireí, pela seguinte linha de comando:

```
c:> copy /b arq1.ext/b + arq2.ext/b +... arqsaida.ext
```

Os demais passos do processamento serão sequencialmente descritos abaixo:

i- os arquivos de dados devidamente "descodificados" como os dados coletados foram limitados de uma maneira a enviar um dado no valor máximo permitido, independente da altitude do foguete e, conseqüentemente, do tempo decorrido do seu lançamento, foi necessário então fazer uma "adaptação". Essa adaptação foi implementada no programa chamado "gua03.c" e consiste em se multiplicar o valor do dado pela constante 0.05539 fornecida pelo orientador tanto para o modo1 como para o modo2.

ii - o arquivo de dados relativos a altitude

esse é um dos pontos citados acima onde a criação desse arquivo teve o mesmo procedimento da Campanha Tapiref diferenciando apenas no formulário fornecido pelo orientador, ou seja, como os arquivos gerados foram dispostos na forma primeira coluna relativa ao tempo e a segunda coluna relativa ao dado em toda a sua extensão, e o perfil da Ne é dado em frequência por altitude, ou frequência por altura, foi necessário então "gerar" o arquivo de alturas em função do único parâmetro em comum existente entre ambos: o próprio tempo. Através de um formulário fornecido pelo orientador que continha as alturas do foguete de 10 segundos em 10 segundos, gerou-se o arquivo com as alturas manualmente chamado "guaalt.dat" pelo recurso do Norton Editor e implementou-se a fórmula abaixo, também fornecida pelo orientador, em um programa chamado "gua04.c" (vide apêndice) para modo1 e "gua05.c" (vide apêndice) para modo2, que interpolava os tempos dos dados com os tempos das alturas, lembrando que o apogeu foi em 956km.

fórmula: $h = th_i + (((th_f - th_i) / (td_f - tdi)) * (td - tdi))$ onde

h: altura desejada;
 th_i: tempo inicial da altura disponível;
 th_f: tempo final da altura disponível;
 tdi: tempo inicial da faixa do dado;
 tdf: tempo final da faixa do dado;
 td: tempo do dado atual.

iii - o arquivo de dados relativos a "linha base" definindo "linha base" pode-se dizer que são os dados essenciais responsáveis pelo contorno da curva do perfil da Ne. Para isso, foi implementado um programa para cada modo chamados de "gua06.c" (vide apêndice) para modo1 e "gua07.c" (vide apêndice) para modo2, que se diferenciavam no valor da divisão dos dados:

modo1 - high frequency - o valor da divisão é de 1.50e7 à 1.52e7;
 modo2 - low frequency - o valor da divisão é de 5.38e6 à 5.40e6.

Em seguida tem-se uma ilustração dos arquivos para modo1 e modo2, na forma dados por tempo, e mais explicitamente uma ilustração da "linha base", na mesma forma:

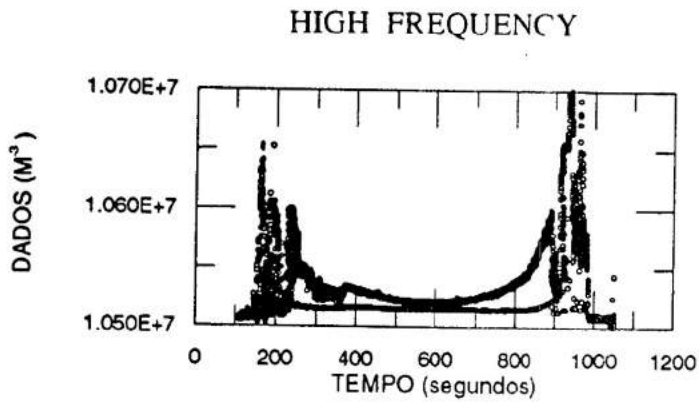


fig. 04

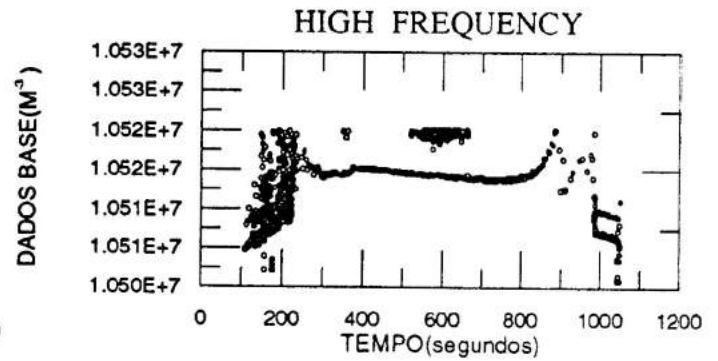


fig. 04.1

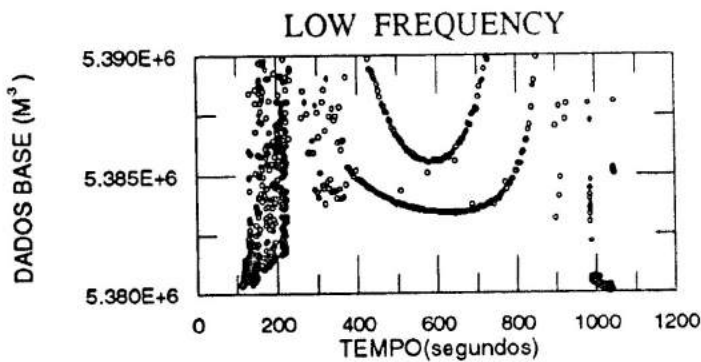


fig. 05

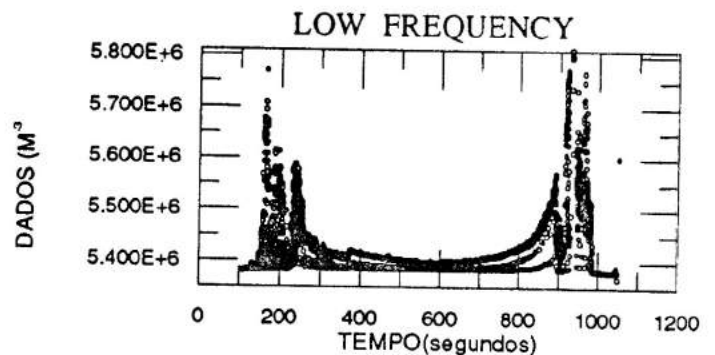


fig.05.1

iv - o arquivo de dados relativos as frequências com as alturas equivalentes aos dados que já estão “descodificados”, torna-se necessário ainda que se calcule as frequências relativas a esses dados através do cálculo abaixo, também fornecido pelo orientador, por um programa chamado “gua08.c”(vide apêndice) para modo1 e um programa chamado “gua09.c” (vide apêndice) para modo2 que implementa basicamente:

$$\text{frequência} = 2.0 * f * f * (f - f_0) * / f_0 * 81.0 * 0.5 * k \text{ onde}$$

frequência: densidade eletrônica numérica da ionosfera(Ne)

f0: dado relativo da “linha base”

f: dado atual

k: constante = 0.369 (modo1)

constante = 0.442 (modo2)

obs: é necessário ressaltar que todos as constantes envolvidas no cálculo e, o próprio cálculo, foram fornecidos pelo orientador.

v - o arquivo de dados relativos a eliminação do movimento de precessão do foguete

nesse perfil já calculado é necessário ainda que se elimine o movimento de precessão do foguete através da média corrida de 73(setenta e três) pontos. Média corrida é: tanto para o arquivo de frequências de modo1 como de modo2 o procedimento foi o mesmo. Calcula-se uma média aritmética para os 73(setenta e três) primeiros dados. Essa média aritmética é colocada na primeira posição. Após, calcula-se uma nova média aritmética com o 2° até o 74° dado e a coloca na segunda posição, uma nova média aritmética com o 3° até o 75° e a coloca na terceira posição e assim sucessivamente.

Essa lógica foi implementada num programa chamado "gua10.c"(vide apêndice) que foi executado tanto para o modo1 como para o modo2. E o número 73 pontos foi dado pelo orientador após ter plotado todos os gráficos relativos aos dados de magnitude e ter verificado a quantidade de pontos dentro de um período.

V- CAMPANHA PIRI-AÇU

V.1 - OBTENÇÃO DOS DADOS

A Campanha Piri-açu foi o lançamento de um único foguete no dia 18 de dezembro de 1995 às 21:17hs (L.T.) na base de lançamento brasileira em Alcântara(MA) sendo 757 segundos de tempo total de dados coletados atingindo o apogeu em aproximadamente 550km.

A trajetória está disponível em disco flexível de 3 1/2" em um único arquivo que foi diretamente plotada pelo aplicativo Grapher for windows e o resultado atingido foi o esperado.

Os dados ainda não estão a disposição ainda mas o seu processamento já está sendo explicado e analisado pelo orientador, e conseqüentemente irá para o próximo relatório.

VI- CONCLUSÃO

**VI.1 - PERFIL PRELIMINAR CAMPANHA TAPIREÍ
MODO C4**

**VI.2 - PERFIL PRELIMINAR CAMPANHA TAPIREÍ
MODO CT**

**VI.3 - PERFIL PRELIMINAR CAMPANHA GUARÁ
MODO1**

**VI.4 - PERFIL PRELIMINAR CAMPANHA GUARÁ
MODO2**

**VI.5 - TRAJETÓRIA DO FOGUETE DA CAMPANHA
PIRI-AÇU**

V.1 - PERFIL PRELIMINAR CAMPANHA TAPIREÍ MODO C4

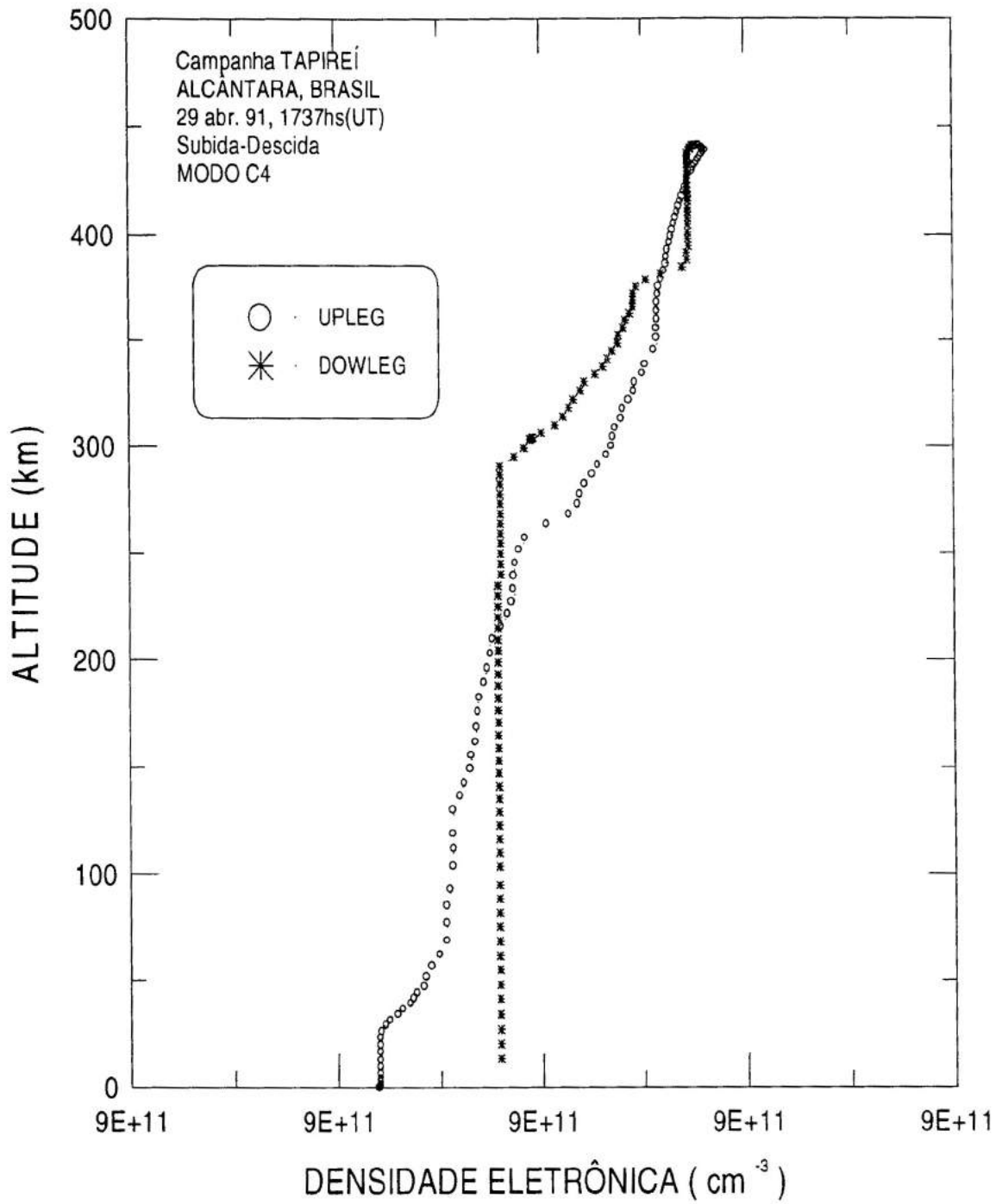


fig. 06

V.2 - PERFIL PRELIMINAR CAMPANHA TAPIREÍ MODO CT

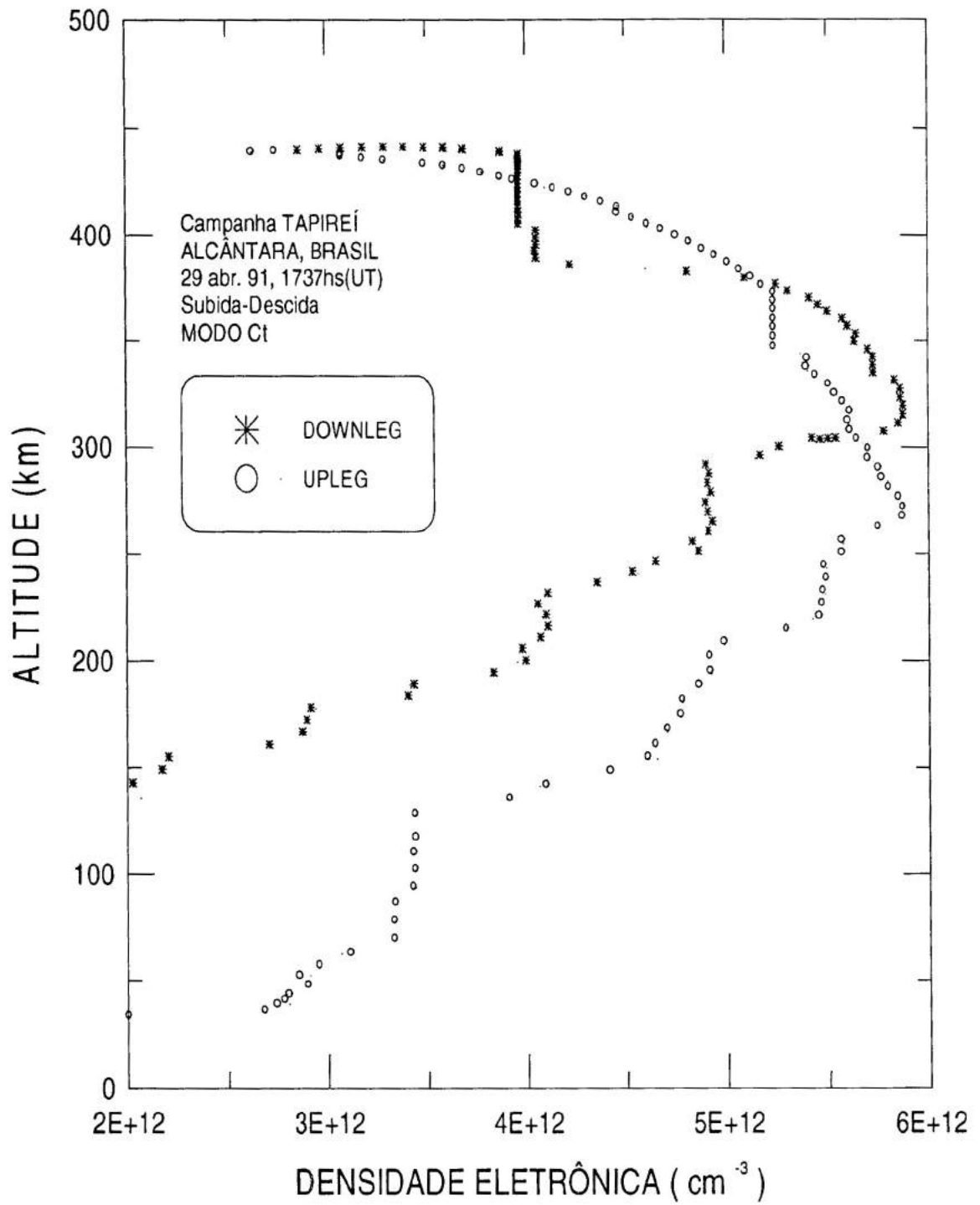


fig. 07

V.3 - PERFIL PRELIMINAR CAMPANHA GUARÁ MODO1

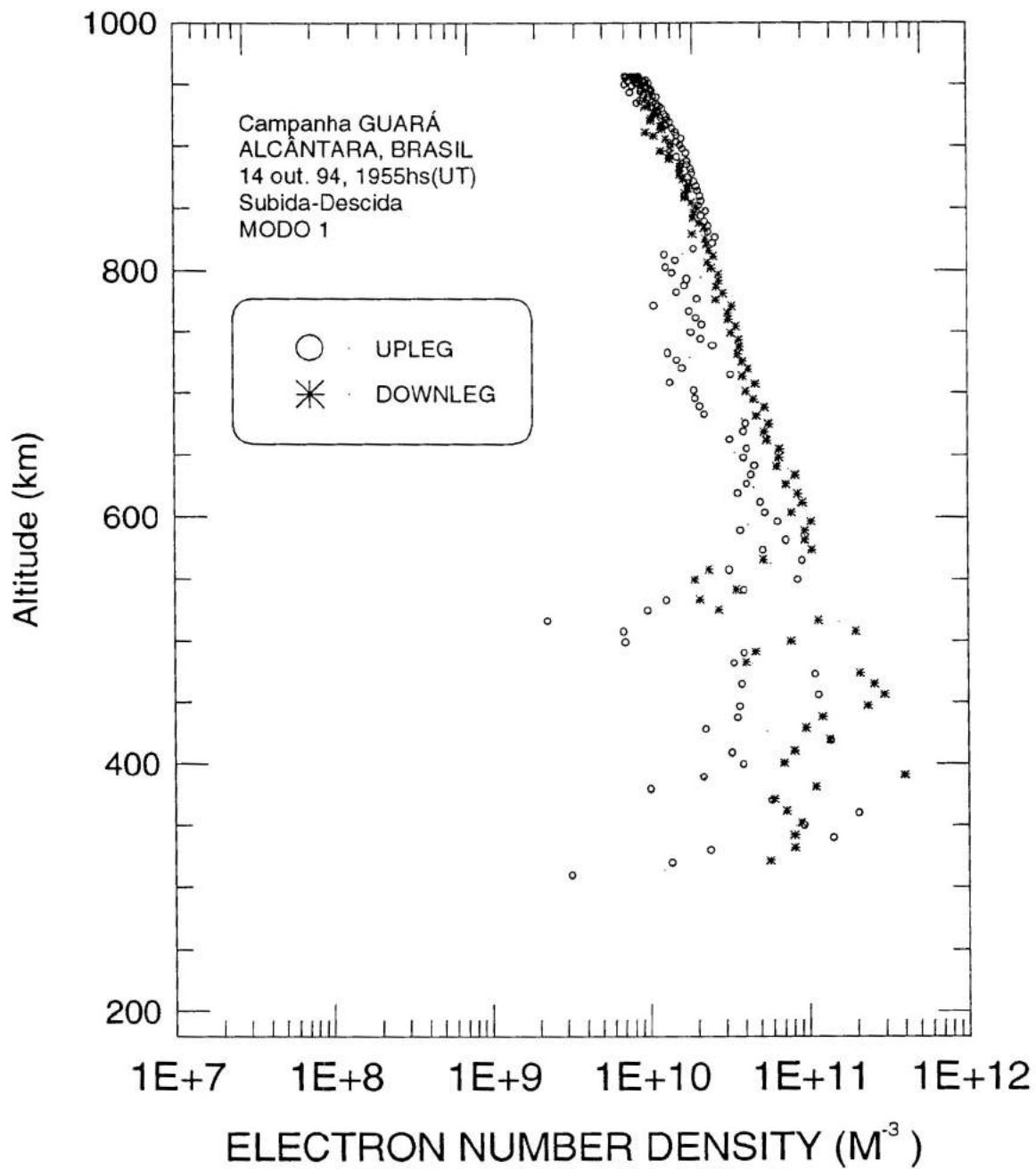


fig. 08

V.4 - PERFIL PRELIMINAR CAMPANHA GUARÁ MODO2

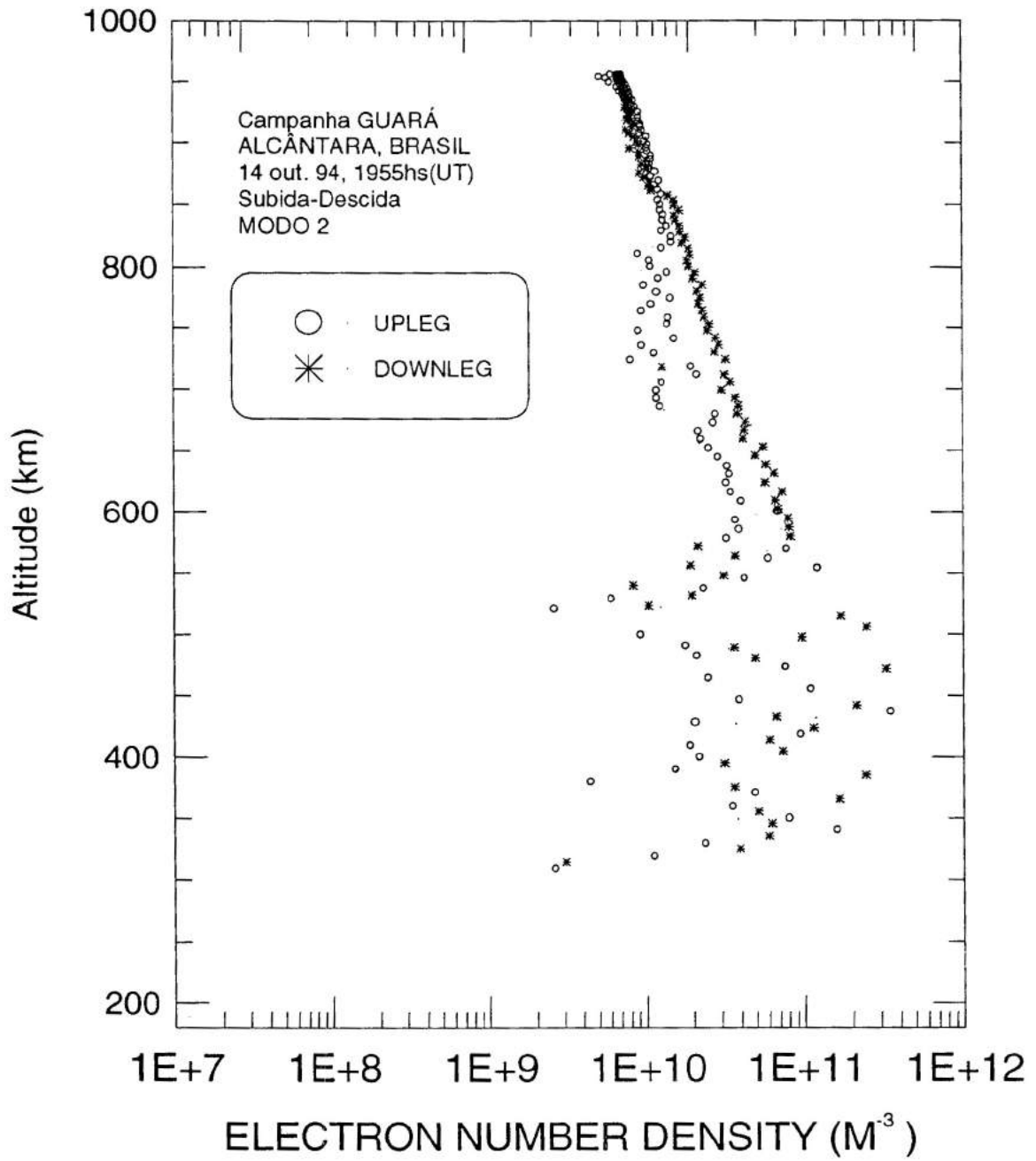


fig.09

V.5 - TRAJETÓRIA DO FOGUETE DA CAMPANHA PIRI-AÇU

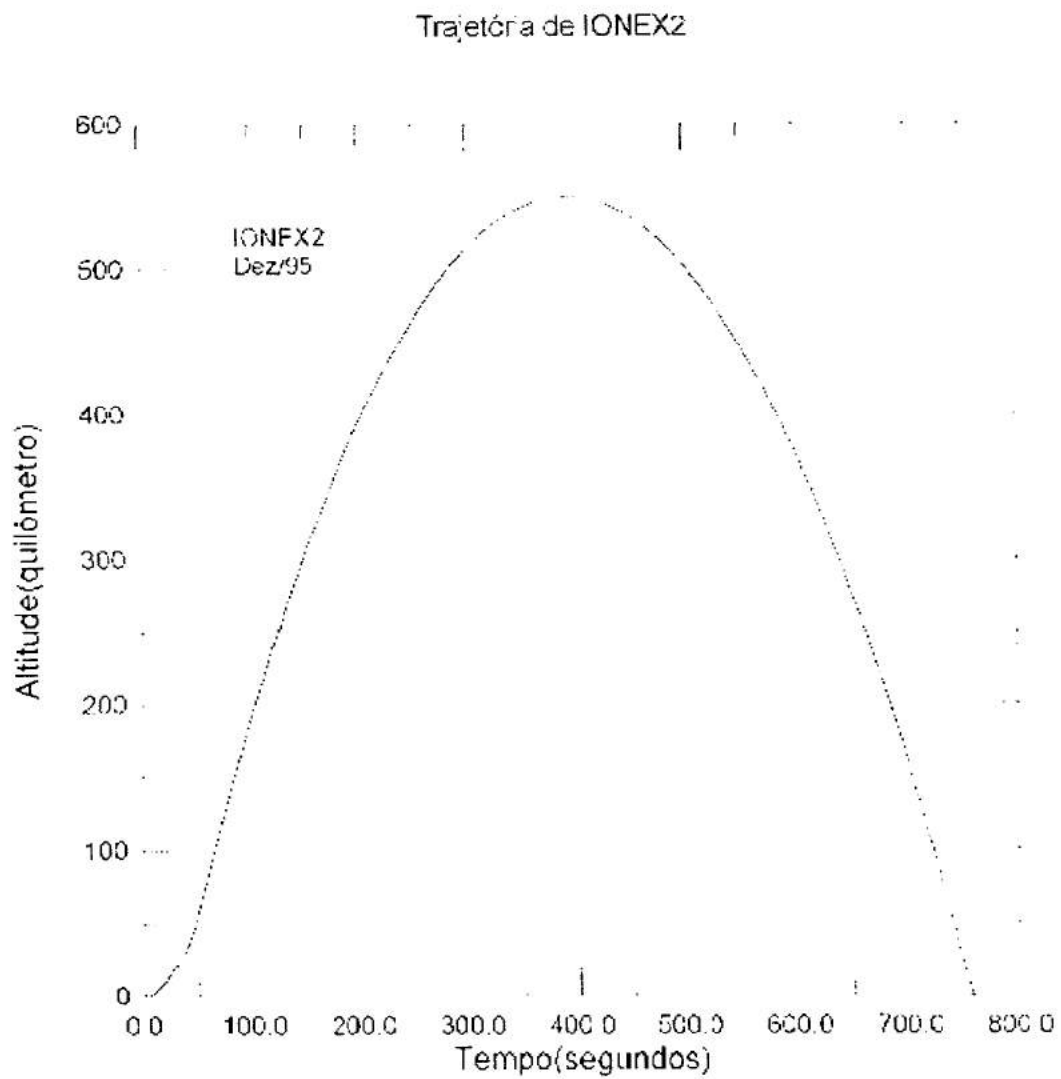


fig.10

APÊNDICE

i.1- LISTAGENS DOS PROGRAMAS IMPLEMENTADOS

i.1.1 - CAMPANHA TAPIREÍ

```

/* tap01.c - objetivo: uniformizar os dados de 0.0s à 682.0s gerando os valores de
c1,c2,c3,c4 e ct */

/* declaração das bibliotecas utilizadas */
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

/* declaração das variáveis utilizadas */
FILE *fpi,*fc1,*fc2,*fc3,*fc4,*fct;
int i = 0, ct = 2;
char x1[8], x2[8], x3[8];
float hr = 0.0 , hora = 0.0, valor = 0.0, c1= 0.0, c2 = 0.0, c3 = 0.0, c4 = 0.0, aux = 0.0;

/* rotina para abrir todos os arquivos utilizados */
abre_arq(){
    if ( ( fc1 = fopen ( "tap01c1.dat","w" ) ) == NULL ){
        printf ( "ARQ TAP01c1.DAT NÃO ABERTO" ); exit (1);
    }

    if ( ( fc2 = fopen ( "tap01c2.dat","w" ) ) == NULL ){
        printf ( "ARQ TAP01c2.DAT NÃO ABERTO" ); exit (1);
    }

    if ( ( fc3 = fopen ( "tap01c3.dat","w" ) ) == NULL ){
        printf ( "ARQ TAP01c3.DAT NÃO ABERTO" ); exit (1);
    }

    if ( ( fc4 = fopen ( "tap01c4.dat","w" ) ) == NULL ){
        printf ( "ARQ TAP01c4.DAT NÃO ABERTO" ); exit (1);
    }

    if ( ( fct = fopen ( "tap01ct.dat","w" ) ) == NULL ){
        printf ( "ARQ TAP01ct.DAT NÃO ABERTO" ); exit (1);
    }

    if ( ( fpi = fopen ( "raw120.dat","r" ) ) == NULL ){
        printf ( "ARQ RAW120.DAT NÃO ABERTO" ); exit (1);
    }
}

```

```

}

/* rotina para fechar todos os arquivos utilizados */
fecha_arq(){
    fclose(fc1); fclose(fc2); fclose(fc3);
    fclose(fc4); fclose(fct); fclose(fpi);
}

/* rotina para cálculo de c1, c2, c3, c4 e ct */
calc_arq(){
    float c = 0.0, vc = 0.0, h = 0.0;
    vc = aux / i; h = hr / i; aux = 0.0; hr = 0.0; i = 0; ++ct;
    if ( ct == 1 || ct == 2 ){
        if (vc >= 0.0 && vc <= 0.25) vc = 0.0;    if (vc > 0.25 && vc <= 0.75) vc = 0.5;
        if (vc > 0.75 && vc <= 1.25) vc = 1.0;    if (vc > 1.25 && vc <= 1.75) vc = 1.5;
        if (vc > 1.75 && vc <= 2.25) vc = 2.0;    if (vc > 2.25 && vc <= 2.75) vc = 2.5;
        if (vc > 2.75 && vc <= 3.25) vc = 3.0;    if (vc > 3.25 && vc <= 3.5) vc = 3.5;
    }
    if ( ct == 1 ){
        c1 = ( 8192.0 * 7.0 * vc ) / 3.50;    fprintf ( fc1, "%3.4f %2.3f\n", h, c1);
    }
    if ( ct == 2 ){
        c2 = (1024.0 * 7.0 * vc ) / 3.50;    hr = h;    fprintf ( fc2, "%3.4f %2.3f\n", h, c2);
    }
    if ( ct == 3 ){
        c3 = (1023.0 * vc ) / 3.50;    fprintf ( fc3, "%3.4f %2.3f\n", h, c3);
    }
    if ( ct == 4 ){
        c4 = (65535.0 * vc ) / 3.50;    fprintf ( fc4, "%3.4f %2.3f\n", h, c4);
        c = (c1+c2+c3);    fprintf ( fct, "%3.4f %2.3f\n", hr, c);
        c = 0.0; ct = 0; hr = 0.0;
    }
}

/* rotina para leitura de todos os arquivos utilizados */
le_arq(){
    while ( ( fscanf ( fpi, "%s %s %s", x1, x2, x3 ) ) != EOF){
        hora = atof ( x1);    if ( hora < 0.0 ) continue;

        if ( hora > 544.0){
            valor = atof(x3);
            aux = valor; hr = hora; i = 1;
            calc_arq(); continue;
        }
        if ( hora >= 0.0){
            valor = atof(x2);
            if ( valor < 0.052 && i > 0){

```



```
        calc_arq( ); continue;
    }
    if ( valor >= 0.052 ){
        i++;  aux += valor;  hr += hora;
    }
}
}
```

```
/* programa principal */
main(){
    clrscr();
    abre_arq();
    le_arq();
    fecha_arq();
}
```

```

/* tap02.c - objetivo: decodificar os dados de c4 (0.0s à 682.0s)*/

/* declaração das bibliotecas utilizadas */
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

/* declaração das variáveis globais */
FILE *fpo,*fpi;

/* rotina para abrir os arquivos utilizados */
abre_arq(){
    if ( ( fpo = fopen ("tap02.dat","w" ) ) == NULL ){
        printf ( "ARQ TAP02.DAT NÃO ABERTO"); exit (1);
    }
    if ( ( fpi = fopen ("tap01c4.dat","r" ) ) == NULL ){
        printf ( "ARQ TAP01c4.DAT NÃO ABERTO"); exit(1);
    }
}

/* rotina para fechar os arquivos utilizados */
fecha_arq( ){
    fclose(fpo); fclose(fpi);
}

le_arq( ){
    char x1[15],x2[15];
    float hora = 0.0, i = 0.0, aux = 0.0, dado = 0.0, ant = 0.0;
    fscanf(fpi,"%s %s",x1,x2); hora = atof(x1); ant = atof(x2);
    while( ( fscanf ( fpi,"%s %s",x1,x2 ) ) != EOF ) {
        hora = atof(x1); dado = atof(x2); aux = dado;
        if ( hora <= 341.0) if (dado < (0.2 * ant ) ) i += 3.50;
        if ( hora > 341.0) if (dado < (0.2 * ant ) ) i -= 3.50;
        ant = aux; dado += i; fprintf ( fpo,"%f %f\n",hora,dado );
    }
}

/* programa principal */
main(){
    clrscr();
    abre_arq();
    le_arq();
    fecha_arq();
}

```

```
/* tap03.c - objetivo: decodificar os dados de ct (0.0s à 682.0s)*/

/* declaração das bibliotecas utilizadas */
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

/* declaração das variáveis globais */
FILE *fpo,*fpi;

/* rotina para abrir os arquivos utilizados */
abre_arq(){
    if ( ( fpo = fopen ("tap03.dat","w" ) ) == NULL ){
        printf ("ARQ TAP03.DAT NÃO ABERTO" ); exit (1);
    }
    if ( ( fpi = fopen ("tap01ct.dat","r" ) ) == NULL ){
        printf ("ARQ TAP01ct.DAT NÃO ABERTO" ); exit (1);
    }
}

/* rotina para fechar os arquivos utilizados */
fecha_arq(){
    fclose(fpo); fclose(fpi);
}

le_arq(){
    char x1[15],x2[15];
    float hora = 0.0, i = 0.0, aux = 0.0, dado = 0.0, ant = 0.0;
    fscanf(fpi,"%s %s",x1,x2); hora = atof(x1); ant = atof(x2);
    while ( ( fscanf ( fpi,"%s %s",x1,x2 ) != EOF ){
        hora = atof(x1); dado=atof(x2); aux=dado;
        if ( hora <= 341.0 ) if ( dado < ( 0.2 * ant ) ) i += 65535.0;
        if ( hora > 341.0 ) if( dado < ( 0.2 * ant ) ) i - = 65535.0;
        ant = aux; dado += i; fprintf (fpo,"%f %f\n",hora,dado);
    }
}

/* programa principal */
main(){
    clrscr();
    abre_arq();
    le_arq();
    fecha_arq();
}
```

```
/* tap04.c - objetivo: gera o arquivo de dados x altura (0.0s à 682.0s) para c4 */

/* declaração das bibliotecas utilizadas */
#include <stdio.h>
#include <stdlib.h>

/* declaração das variáveis globais */
FILE *fpo,*fpi,*fh,*faux;
char x1[25],x2[25],x3[25],x4[25];
float tf = 0.0, t = 0.0;

abre_arq(){
    if ( ( fpo = fopen ( "tap04.dat","w" ) ) == NULL ){
        printf ( "ARQ TAP04.DAT NÃO ABERTO"); exit (1);
    }
    if ( ( fpi = fopen ( "tap02.dat","r" ) ) == NULL ){
        printf ( "ARQ TAP02.DAT NÃO ABERTO" ); exit (1);
    }
    if ( ( faux = fopen ( "tap02.dat","r" ) ) == NULL ){
        printf ( "ARQ TAP02.DAT NÃO ABERTO" ); exit (1);
    }
    if ( ( fh = fopen ( "tapalt.dat","r" ) ) == NULL ){
        printf ( "ARQ TAPALT.DAT NÃO ABERTO" ); exit (1);
    }
}

/* rotina para fechar os arquivos utilizados */
fecha_arq(){
    fclose(fpo);  fclose(fpi);
    fclose(faux); fclose(fh);
}

/* rotina para encontrar o tempo final */
enc_tf(){
    float h = 0.0;  t += 2.0;
    while(fscanf(faux,"%s %s",x1,x2) != EOF){
        h = atof(x1);
        if ( h < t ) continue;
        else tf = atof(x1);
        break;
    }
}
```

```
/* rotina para leitura do arquivo */
le_arq() {
float  zi = 0.0, zf = 0.0, ti = 0.0, d = 0.0, te = 0.0, alt = 0.0;
    fscanf ( fh,"%s %s",x4,x3);  zi = atof(x3);
    while ( ( fscanf ( fh,"%s %s",x4,x3 ) != EOF ) ){
        zf = atof(x3);
        fscanf ( fp,"%s %s",x4,x3);  ti = atof(x4);
        while ( ( fscanf ( fpi,"%s %s",x4,x3 ) != EOF ) ){
            te = atof(x4);  d = atof(x3);
            if ( te > tf ) enc_tf();
            if ( te < tf ){
                alt = zi + ( ( zf - zi ) / ( tf - ti ) ) * ( te - ti );
                fprintf ( fpo,"%f %f %f\n",alt,te,d);
            }
            else{ zi = zf;  break;
                }
        }
    }
}

/* programa principal */
main() {
    clrscr();
    abre_arq();
    le_arq();
    fecha_arq();
}
```

```
/* tap05.c - objetivo: gera o arq de dados x altura ( 0.0s à 682.0s) para ct */
```

```
/* declaração das bibliotecas utilizadas */
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
/* declaração das variáveis globais */
```

```
FILE *fpo,*fpi,*fh,*faux;
```

```
char x1[25],x2[25],x3[25],x4[25];
```

```
float tf = 0.0, t = 0.0;
```

```
abre_arq() {
```

```
    if ( ( fpo = fopen ( "tap05.dat","w" ) ) == NULL ){  
        printf ( "ARQ TAP05.DAT NÃO ABERTO" ); exit (1);  
    }
```

```
    if ( ( fpi = fopen ( "tap03.dat","r" ) ) == NULL ){  
        printf ( "ARQ TAP03.DAT NÃO ABERTO" ); exit (1);  
    }
```

```
    if ( ( faux = fopen ( "tap03.dat","r" ) ) == NULL ){  
        printf ( "ARQ TAP03.DAT NÃO ABERTO" ); exit (1);  
    }
```

```
    if ( ( fh = fopen ( "tapalt.dat","r" ) ) == NULL ){  
        printf ( "ARQ TAPALT.DAT NÃO ABERTO" ); exit (1);  
    }
```

```
}
```

```
/* rotina para fechar os arquivos utilizados */
```

```
fecha_arq() {
```

```
    fclose(fpo);  fclose(fpi);
```

```
    fclose(faux); fclose(fh);
```

```
}
```

```
/* rotina para encontrar o tempo final */
```

```
enc_tf() {
```

```
    float h = 0.0; t += 2.0;
```

```
    while ( fscanf ( faux,"%s %s",x1,x2 ) != EOF ){
```

```
        h=atof(x1);
```

```
        if ( h < t ) continue;
```

```
        else tf = atof(x1);
```

```
        break;
```

```
    }
```

```
}
```

```
/* rotina para leitura do arquivo */
le_arq(){
float  zi = 0.0, zf =0.0, ti = 0.0, d = 0.0, te = 0.0, alt = 0.0;
  fscanf (fh,"%s %s",x4,x3);  zi = atof(x3);
  while ( ( fscanf ( fh,"%s %s",x4,x3 ) != EOF ) ){
    zf = atof(x3);
    fscanf (fp,"%s %s",x4,x3);  ti = atof(x4);
    while((fscanf(fpi,"%s %s",x4,x3) != EOF)){
      te = atof(x4); d = atof(x3);
      if(te > tf) enc_tf();
      if(te < tf){
        alt = zi + ( ( ( zf - zi ) / ( tf - ti ) ) * ( te - ti ) );
        fprintf(fpo,"%f %f %f\n",alt,te,d);
      }
      else{ zi = zf;  break;
    }
  }
}

/* programa principal */
main(){
  clrscr();
  abre_arq();
  le_arq();
  fecha_arq();
}
```

```

/* tap06.c - objetivo: faz o cálculo da frequência para os dados de c4 */

/* declaração das bibliotecas utilizadas */
#include <stdio.h>
#include <stdlib.h>

/* declaração das variáveis globais */
FILE *fpo,*fpi;
char x1[25],x2[25],x3[25];
float alt = 0.0, d = 0.0, hora = 0.0, f0 = 0.0, f01 = 0.0, f02 = 0.0, a = 0.0, b = 0.0;
float f = 0.0, df = 0.0, n = 0.0, b1 = 0.0;

/* rotina para abrir os arquivos utilizados */
abre_arq( ){
    if ( ( fpo = fopen ("tap06.dat","w" ) ) == NULL ){
        printf ( "ARQ TAP06.DAT NÃO ABERTO" );    exit (1);
    }
    if ( ( fpi = fopen ("tap04.dat","r" ) ) == NULL ){
        printf ( "ARQ TAP04.DAT NÃO ABERTO" );    exit (1);
    }
}

/* rotina para fechar os arquivos utilizados */
fecha_arq( ){
    fclose(fpo); fclose(fpi);
}

/* rotina para leitura do arquivo => cálculo da frequência */
le_arq( ){
    f01 = ( ( 393214.0 + ( ( 1023 * 2.591 ) / 3.50 ) ) / 0.04 );
    f02 = ( ( 393214.0 + ( ( 1023 * 2.653 ) / 3.50 ) ) / 0.04 );
    a = ( f02 - f01 ) / ( 680.672485 - 0.0905 );
    b = ( f01 - ( a * 0.0405 ) );
    while ( ( fscanf ( fpi,"%s %s %s",x2,x3,x1) != EOF ) ){
        d = atof(x1); alt = atof(x2); hora = atof(x3);
        f0 = ( ( a * hora ) + b ); f = ( d + 393214.0 );
        df = ( f - f0 ); n = ( 0.242369 * df * f );
        fprintf (fpo,"%f %f %f\n",alt,hora,n);
        d = 0.0; f = 0.0; f0 = 0.0; df = 0.0; n = 0.0;
    }
}

```



```
/*programa principal */  
main( )  
  clrscr( );  
  abre_arq( );  
  le_arq( );  
  fecha_arq( );  
}
```

```

/* tap07.c - objetivo: faz o cálculo da frequência para os dados de ct */

/* declaração das bibliotecas utilizadas */
#include <stdio.h>
#include <stdlib.h>

/* declaração das variáveis globais */
FILE *fpo,*fpi;
char x1[25],x2[25],x3[25];
float alt = 0.0, d = 0.0, hora = 0.0, f0 = 0.0, f01 = 0.0, f02 = 0.0, a = 0.0, b = 0.0;
float f = 0.0, df = 0.0, n = 0.0, b1 = 0.0;

/* rotina para abrir os arquivos utilizados */
abre_arq(){
    if ( ( fpo = fopen ("tap07.dat","w" ) ) == NULL ){
        printf ( "ARQ TAP06.DAT NÃO ABERTO" );    exit (1);
    }

    if ( ( fpi = fopen ("tap05.dat","r" ) ) == NULL ){
        printf ( "ARQ TAP04.DAT NÃO ABERTO" );    exit (1);
    }
}

/* rotina para fechar os arquivos utilizados */
fecha_arq(){
    fclose(fpo); fclose(fpi);
}

/* rotina para leitura do arquivo => cálculo da frequência */
le_arq(){
    f01 = ( ( 393214.0 + ( ( 1023 * 2.591 ) / 3.50 ) ) / 0.04 );
    f02 = ( ( 393214.0 + ( ( 1023 * 2.653 ) / 3.50 ) ) / 0.04 );
    a = ( f02 - f01 ) / ( 680.672485 - 0.0905 );
    b = ( f01 - ( a * 0.0405 ) );
    while ( ( fscanf ( fpi,"%s %s %s",x2,x3,x1) != EOF ) ){
        d = atof(x1); alt = atof(x2); hora = atof(x3);
        f0 = ( ( a * hora ) + b ); f = ( d + 393214.0 );
        df = ( f - f0 ); n = ( 0.242369 * df * f );
        fprintf (fpo,"%f %f %f\n",alt,hora,n);
        d = 0.0; f = 0.0; f0 = 0.0; df = 0.0; n = 0.0;
    }
}

```

```
/* programa principal */  
main(){  
    clrscr();  
    abre_arq();  
    le_arq();  
    fecha_arq();  
}
```

VI.2 - CAMPANHA GUARÁ

```

/* gua01.c - objetivo: gerar arquivo de dados "filtrados" */

/* declaração das bibliotecas utilizadas */
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

/* declaração das variáveis globais */
FILE *fi,*fo;
char x1[15],x2[15],x3[15];
float lo=0.0, hi=0.0, t=0.0, h=0.0, l=0.0;

/* rotina para abrir os arquivos utilizados */
abre_arq( ){
    if ( ( fi = fopen ("arqent.dat","r" ) ) == NULL ){
        printf ("ARQ ENTRADA NÃO ABERTO" ); exit (1);
    }

    if ( fo = fopen ("arqsai.dat","w")==NULL){
        printf ("ARQ SAIDA NÃO ABERTO" ); exit (1);
    }
}

/* rotina para finalização dos arquivos utilizados */
fecha_arq( ){
    fclose(fi); fclose(fo);
}

/* rotina para leitura do arquivo => "filtra" os dados */
le_arq( ){
    while ( fscanf(fi,"%s %s %s",x1,x2,x3 ) != EOF ){
        t = atof(x3); hi = atof(x2); l = atof(x1);
        if ( ( l != lo ) && ( h != hi ) ){
            fprintf (fo,"%3.4f %3.4f %3.4f\n",t,hi,lo);
            l = lo; h = hi;
        }
    }
}

```

```
/* programa principal */  
main()  
{  
    clrscr();  
    abre_arq();  
    le_arq();  
    fecha_arq();  
}
```

(*) -> esse programa foi executado 14 vezes, uma vez para cada arquivo, sendo necessario para cada vez substituir as seguintes constantes pelos respectivos arquivos:

arqent	arqsai
dig01.dat	gua0101.dat
dig02.dat	gua0102.dat
dig03.dat	gua0103.dat
dig04.dat	gua0104.dat
dig05.dat	gua0105.dat
dig06.dat	gua0106.dat
dig07.dat	gua0107.dat
dig08.dat	gua0108.dat
dig09.dat	gua0109.dat
dig10.dat	gua0110.dat
dig11.dat	gua0111.dat
dig12.dat	gua0112.dat
dig13.dat	gua0113.dat
dig14.dat	gua0114.dat

```

/* gua02.c - objetivo: gera arquivo de dados "separados" em modo 1 e modo 2 */

/* declaração das bibliotecas utilizadas */
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

/* declaração das variáveis globais */
FILE *fi,*fo1,*fo2;
char x1[15],x2[15],x3[15];
int i = 0;
float lo = 0.0, hi = 0.0, t = 0.0, h = 0.0, l = 0.0, c = 0.0;

/* rotina para abrir os arquivos utilizados */
abre_arq(){
    if ( ( fi = fopen ("guara.dat","r" ) ) == NULL ){
        printf ( "ARQ DE DADOS ENTRADA NÃO ABERTO" ); exit (1);
    }

    if ( ( fo1 = fopen ("gua02m1.dat","w" ) ) == NULL ){
        printf ( "ARQ DE DADOS SAIDA MODO1 NÃO ABERTO" ); exit (1);
    }
    if ( ( fo2 = fopen("gua02m2.dat","w" ) ) == NULL ){
        printf ("ARQ DE DADOS SAIDA MODO2 NÃO ABERTO" ); exit (1);
    }
}

/* rotina para finalizar os arquivos utilizados */
fecha_arq(){
    fclose(fi); fclose(fo1); fclose(fo2);
}

/* rotina para leitura do arquivo e "separa-lo" em modo1 e modo2 */
leitura(){
    while ( ( fscanf (fi,"%s %s %s",x1,x2,x3) != EOF){
        lo = atof(x3); hi=atof(x2); t=atof(x1);
        if ((l != lo) || (h != hi)){
            l = lo; h = hi; i++;
            c = ( hi * 1024.0 ) + lo;
            if ( ( I % 2 ) == 0.0 ) fprintf (fo1,"%3.4f %3.4f\n",t,c);
            else fprintf (fo2,"%3.4f %3.4f\n",t,c);
        }
    }
}

```

```
/* programa principal */  
main(){  
    clrscr();  
    abre_arq( );  
    le_arq( );  
    fecha_arq( );  
}
```

```
/* gua03.c - objetivo:gera arquivo de dados adaptados pela cte. 0.05559 */
/* para modo1 e modo2 */

/* declaração das bibliotecas utilizadas */
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

/* declaração das variáveis globais */
FILE *fi,*fo, *fp, *fq;
char x1[15],x2[15];
float h = 0.0, t = 0.0, vl = 0.0;

/* rotina para abrir os arquivos utilizados */
abre_arq(){
    if ( ( fi = fopen ("gua02m1.dat","r" ) ) == NULL ){
        printf ("ARQ DE DADOS ENTRADA MODO1 NÇO ABERTO"); exit (1);
    }

    if ( ( fo = fopen ("gua03m1.dat","w" ) ) == NULL ){
        printf ("ARQ DE DADOS SAÖDA MODO1 NÇO ABERTO"); exit (1);
    }

    if ( ( fp = fopen ("gua02m2.dat","r" ) ) == NULL ){
        printf ("ARQ DE DADOS ENTRADA MODO2 NÇO ABERTO"); exit (1);
    }

    if ( ( fq = fopen ("gua03m2.dat","w" ) ) == NULL ){
        printf ("ARQ DE DADOS SAÖDA MODO2 NÇO ABERTO"); exit (1);
    }

}

/* rotina para finalizar os arquivos utilizados */
fecha_arq(){
    fclose(fi); fclose(fo); fclose(fq); fclose(fp);
}
```



```
/* rotina para leitura do arquivo => dados "adaptados" para modo1 e modo2 */
le_arq(){
    while ( ( fscanf (fi,"%s %s",x1,x2) ) != EOF ){
        h = atof(x2); t = atof(x1); vl = h / 0.05539;
        fprintf (fo,"%3.4f %e\n",t,vl);
    }
    while ( ( fscanf (fp,"%s %s",x1,x2) ) != EOF ){
        h = atof(x2); t = atof(x1); vl = h / 0.05539;
        fprintf (fq,"%3.4f %e\n",t,vl);
    }
}

/* programa principal */
main(){
    clrscr( );
    abre_arq( );
    le_arq( );
    getch( );
}
```

```

/* gua04.c - objetivo: gera arquivo de alturas para os dados de modo 1 */

/* declaração das bibliotecas utilizadas */
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

/* declaração das variáveis globais */
FILE *fi,*fo,*fih;
char x1[15],x2[15];
int i = 0;
float h = 138.9513, h1 = 0.0, h2 = 0.0, t2 = 0.0, t1 = 0.0, d1 = 0.0, d2 = 0.0;
float t = 0.0, d = 0.0, flag = 0.0;

/* rotina para abrir os arquivos utilizados */
abre_arq(){
    if ( ( fi = fopen ( "gua03m1.dat","r" ) ) == NULL ){
        printf ( "ARQ DE DADOS DE ENTRADA NÇO ABERTO" ); exit (1);
    }

    if ( ( fih = fopen ( "gualt.dat","r" ) ) == NULL ){
        printf ( "ARQ DE DADOS DE ALTURA NÇO ABERTO" ); exit (1);
    }

    if ( ( fõ = fopen ( "gua04.dat","w" ) ) == NULL ){
        printf ( "ARQ DE SAÏDA NÇO ABERTO" ); exit (1);
    }
}

/* rotina para finalizar os arquivos utilizados */
fecha_arq(){
    fclose(fi); fclose(fih); fclose(fo);
}

/* rotina para leitura do arquivo => calculo da altura */
le_arq(){
    fscanf (fih,"%s %s",x1,x2);
    h1 = atof(x1); t1 = atof(x2);
    while ( ( fscanf (fih,"%s %s",x1,x2) ) != EOF ){
        h2 = atof(x1); t2 = atof(x2);
        do{
            fscanf (fi,"%s %s",x1,x2);
            t = atof(x1); d = atof(x2);

```

```
        if ( t > 100.0 ) {
            h = h1 + ( ( ( h2 - h1 ) * ( t - t1 ) ) / ( t2 - t1 ) );
            fprintf ( fo, "%f %f %f\n", t, h, d );
            if ( h < 100.0 ) exit ( 0 );
        }
    } while ( t < t2 );
    t1 = t2; h1 = h2;
}
}

/* programa principal */
main ( ) {
    clrscr ( );
    abre_arq ( );
    le_arq ( );
    fecha_arq ( );
}
```

```

/* gua05.c - objetivo: gera arquivo de alturas para dados de modo 2 */

/* declaração das bibliotecas utilizadas */
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

/* declaração das variáveis globais */
FILE *fi,*fo,*fih;
char x1[15],x2[15];
int i = 0;
float h = 138.9513, h1 = 0.0, h2 = 0.0, t2 = 0.0, t1 = 0.0, d1 = 0.0, d2 = 0.0;
float t = 0.0, d = 0.0, flag = 0.0;

/* rotina para abrir os arquivos utilizados */
abre_arq(){
    if ( ( fi = fopen ("gua03m2.dat","r" ) ) == NULL ){
        printf ("ARQ DE DADOS DE ENTRADA MODO2 NÇO ABERTO"); exit (1);
    }

    if ( fih = fopen ("gualt.dat","r" ) ) == NULL ){
        printf ("ARQ DE DADOS DE ALTURA NÇO ABERTO" ); exit (1);
    }

    if ( ( fo = fopen ("gua05.dat","w" ) ) == NULL ){
        printf ("ARQ DE SAÖDA NÇO ABERTO"); exit (1);
    }
}

/* rotina para finalizar os arquivos utilizados */
fecha_arq(){
    fclose(fi); fclose(fih); fclose(fo);
}

/* para leitura do arquivo */
le_arq(){
    fscanf (fih,"%s %s",x1,x2);
    h1 = atof(x1); t1 = atof(x2);
    while ( ( fscanf(fih,"%s %s",x1,x2) ) != EOF ){
        h2 = atof(x1); t2 = atof(x2);
        do{
            fscanf (fi,"%s %s",x1,x2);
            t = atof(x1); d = atof(x2);

```

```
        if ( t > 100.0 ){
            h = h1 + ( ( h2 - h1 ) * ( t - t1 ) ) / ( t2 - t1 );
            fprintf (fo,"%f %f %e\n",t,h,d);
            if ( h < 100.0 ) exit (0);
        }
    }while ( t < t2 );
    t1 = t2; h1 = h2;
}
}

/* programa principal */
main(){
    clrscr( );
    abre_arq( );
    le_arq( );
    fecha_arq( );
}
```

```

/* gua06.c - objetivo: dados correspondente a "linha base" do arquivo para modo l */

/* declaração das bibliotecas utilizadas */
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

/* declaração das variáveis globais */
FILE *fi, *fo;
char x1[15],x2[15],x3[20];
float d = 0.0, t = 0.0, h = 0.0;

/* rotina para abrir os arquivos utilizados */
abre_arq( ){
    if ( ( fi = fopen ("gua04.dat","r" ) ) == NULL ){
        printf ("ARQ DE DADOS DE ENTRADA NÃO ABERTO"); exit (1);

        if ( ( fo = fopen ("gua06.dat","w" ) ) == NULL ){
            printf ("ARQ DE DADOS DE SAIDA NÃO ABERTO"); exit (1);
        }
    }
}

/* rotina para finalizar os arquivos utilizados */
fecha_arq( ){
    fclose(fi); fclose(fo);
}

/* rotina para leitura do arquivo */
le_arq( ){
    while ( ( fscanf( fi,"%s %s %s",x1,x2,x3 ) ) != EOF ){
        h = atof(x2); d = atof(x3); t = atof(x1);

        /* UPLEG - DOWNLEG */
        if ( h > 300.0 ){
            if ( d < 1.052E7 ) fprintf (fo,"%f %f %e\n",t,h,d);
        }
        if ( h <= 300.0 ){
            if ( d < 1.0506e7 ) fprintf (fo,"%f %f %e\n",t,h,d);
        }
    }
}

```

```
/* programa principal */  
main(){  
    clrscr( );  
    abre_arq( );  
    ler_arq( );  
    fec. a_arq( );  
}
```

```

/* gua07.c - objetivo: dados correspondentes a "linha base" do arquivo para modo2 */

/* declaração das bibliotecas utilizadas */
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

/* declaração das variáveis globais */
FILE *fi, *fo;
char x1[15],x2[15],x3[20];
float d = 0.0, t = 0.0, h = 0.0;

/* rotina para abrir os arquivos utilizados */
abre_arq(){
    if ( ( fi = fopen ("gua05.dat","r" ) ) == NULL ){
        printf ("ARQ DE DADOS DE ENTRADA MODO2 NÃO ABERTO" ); exit (1);
    }

    if ( ( fo = fopen ("gua07.dat","w" ) ) == NULL ){
        printf ("ARQ DE DADOS DE SAIDA NÃO ABERTO" ); exit (1);
    }
}

/* rotina para finalizar os arquivos utilizados */
fecha_arq(){
    fclose(fi); fclose(fo);
}

/* rotina para leitura do arquivo */

while ( scanf (fi,"%s %s %s",x1,x2,x3 ) != EOF ){
    t = atof(x2); d = atof(x3); h = atof(x1);

    /* UPLEG */
    if ( t < 570.10 ){
        if ( h > 500.0 ){
            if ( d >= 5.4E6 ) fprintf (fo,"%f %f %f\n",t,h,d);
        }
        if ( h <= 500.0 ) fprintf (fo,"%f %f %f\n",t,h,d);
    }

    /* DOWNLEG */
    if ( t > 570.10 ){

```



```
if ( h > 800.0 ){
    if ( d >= 5.4E6 ) fprintf (fo,"%f %f %f\n",t,h,d);
}
if ( ( h > 680.0 ) && ( h <= 800.0 ) ){
    if ( d >= 5.42E6 ) fprintf (fo,"%f %f %f\n",t,h,d);
}
if ( ( h > 640.0 ) && ( h <= 680.0 ) ){
    if ( d >= 5.44E6 ) fprintf (fo,"%f %f %f\n",t,h,d);
}
if ( ( h > 600.0 ) && ( h <= 640.0 ) ){
    if ( d >= 5.48E6 ) fprintf (fo,"%f %f %f\n",t,h,d);
}
if ( h <= 600.0 ){
    if ( d >= 5.38E6 ) fprintf (fo,"%f %f %f\n",t,h,d);
}
}
}

    \ principal */

    );
    abre_arq( );
    le_arq( );
    fecha_arq( );
}
```

```

/* gua08.c - objetivo: faz o cálculo da frequência (densidade eletrônica) */
/* do arquivo para modo l */

/* declaração das bibliotecas utilizadas */
#include <stdio.h>
#include <stdlib.h>

/* declaração das variáveis globais */
FILE *fpo,*fpi,*fbs;
char x1[15],x2[15],x3[15];
float h = 0.0, d = 0.0, tbi = 0.0, fbi = 0.0, t = 0.0 , tbf = 0.0 , fbf = 0.0;
float f = 0.0, df = 0.0, ne = 0.0, n1 = 0.0, n2 = 0.0;

/* rotina para abrir os arquivos utilizados */
abre_arq( ){
    if ( ( fpo = fopen ("gua08.dat","w" ) ) == NULL ){
        printf ("ARQ DE DADOS DE SAÏDA NÇO ABERTO"); exit (1);
    }

    if ( ( fbs = fopen ("gua06.dat","r" ) ) == NULL ){
        printf ("ARQ DE DADOS DE ENTRADA NÇO ABERTO"); exit (1);
    }

    if ( ( fpi = fopen ("gua04.dat","r" ) ) == NULL ){
        printf ("ARQ DE DADOS DE ENTRADA NÇO ABERTO"); exit (1);
    }
}

/* rotina para finalizar os arquivos utilizados */
{
    ); fclose(fpi); fclose(fbs);
}

/* rotina para leitura do arquivo => cálculo da frequência */
le_arq( ){
    fscanf (fbs,"%s %s %s",x1,x2,x3);
    tbi = atof(x1); fbi = atof(x3);
    while ( ( fscanf (fbs,"%s %s %s",x1,x2,x3) ) != EOF ){
        tbf = atof(x1); fbf = atof(x3);
        do{
            fscanf (fpi,"%s %s %s",x1,x2,x3);
            f = atof(x3); h = atof(x2); t = atof(x1);
            if ( t >= tbi ){

```

```
df = ( f - fbi ); n2 = ( fbi * 81.0 * 0.5 * 0.369 );
n1 = ( 2 * f * f * df ); ne = n1 / n2;
fprintf (fpo, "%f %f %e\n", t, h, ne);
    }
}while ( t < tbf );
tbi = tbf; fbi = fbf;
}
}

/* programa principal */
main( ){
    clrscr( );
    abre_arq( );
    le_arq( );
    fecha_arq( );
}
```

```

/* gua09.c - objetivo: faz o cálculo da frequência (densidade eletrônica) */
/* do arquivo para modo2 */

/* declaração das bibliotecas utilizadas */
#include <stdio.h>
#include <stdlib.h>

/* declaração das variáveis globais */
FILE *fpo,*fpi,*fbs;
char x1[15],x2[15],x3[15];
float h = 0.0, d = 0.0, tbi = 0.0, fbi = 0.0, t = 0.0 , tbf = 0.0 , fbf = 0.0;
float f = 0.0, df = 0.0, ne = 0.0, n1 = 0.0, n2 = 0.0;

/* rotina para abrir os arquivos utilizados */
abre_arq( ){
    if ( ( fpo = fopen ("gua09.dat","w" ) ) == NULL ){
        printf ("ARQ DE DADOS DE SAIDA NÃO ABERTO"); exit (1);
    }

    if ( ( fpi = fopen ("gua07.dat","r" ) ) == NULL ){
        printf ("ARQ DE DADOS DE ENTRADA NÃO ABERTO"); exit (1);
    }

    if ( ( fbs = fopen ("gua05.dat","r" ) ) == NULL ){
        printf ("ARQ DE DADOS DE ENTRADA NÃO ABERTO"); exit (1);
    }
}

/* rotina para finalizar os arquivos utilizados */
fecha_arq( ){
    fclose(fpo); fclose(fpi); fclose(fbs);
}

/* rotina para leitura do arquivo => cálculo da frequência */
le_arq( ){
    fscanf (fbs,"%s %s %s",x1,x2,x3);
    tbi = atof(x1); fbi = atof(x3);
    while ( ( fscanf (fbs,"%s %s %s",x1,x2,x3) ) != EOF ){
        tbf = atof(x1); fbf = atof(x3);
        do{
            fscanf (fpi,"%s %s %s",x1,x2,x3);
            f = atof(x3); h = atof(x2); t = atof(x1);

            if ( t >= tbi ){
                df = ( f - fbi ); n2 = ( fbi * 81.0 * 0.5 * 0.4420 );
                n1 = ( 2 * f * f * df ); ne = n1 / n2;
                fprintf (fpo,"%f %f %e\n",t,h,ne);
            }
        } while ( t >= tbi );
    }
}

```

```
    }  
  }while ( t < tbf );  
  tbi = tbf; tbi = tbf;  
  }  
}  
  
/* programa principal */  
main( ){  
  clrscr( );  
  abre_arq( );  
  le_arq( );  
  fecha_arq( );  
}
```

```

/* gua10.c - objetivo: elimina ruídos */

/* declaração das bibliotecas utilizadas */
#include <stdio.h>
#include <math.h>
#include <conio.h>

/* declaração das variáveis globais */
FILE *fpi, *fpo, *fp;
char x1[15], x2[15];
int i = 0;
float aux = 0.0, alt = 0.0, med = 0.0, valor = 0.0;

/* rotina para abrir os arquivos utilizados => fase1 */
abre_arq(){
    if ( ( fpi = fopen ("gua08.dat","r" ) ) == NULL ){
        printf ("ARQ DE DADOS DE ENTRADA NÃO ABERTO!!"); exit(1);
    }
    if ( ( fpo = fopen ("guamed15.dat","w" ) ) == NULL ){
        printf ("ARQ DE DADOS DA MÉDIA NÃO ABERTO!!"); exit(1);
    }
}

/* rotina para abrir os arquivos utilizados => fase2 */
abre_arq1(){
    if ( ( fpi = fopen ("guamed15.dat","r" ) ) == NULL ){
        printf ("ARQ DE DADOS DA MÉDIA NÃO ABERTO!!"); exit(1);
    }
    if ( ( fp = fopen ("gua08.dat","r" ) ) == NULL ){
        printf ("ARQ DE DADOS DE ENTRADA NÃO ABERTO!!"); exit(1);
    }
    if ( ( fpo = fopen ("gua10.dat","w" ) ) == NULL ){
        printf ("ARQ DE DADOS DE SAIDA NÃO ABERTO!!"); exit(1);
    }
}

/* rotina para leitura dos dados e cálculo da média de 15 pontos => fase1 */
calc_dds(){
    while (i <= 15){
        if (fscanf (fpi,"%s %s",x1,x2) != EOF ){
            alt = atof(x1); valor = atof(x2);

            aux+=valor; i++;
            if(i==5){
                i = 0; med = aux / 5; aux = 0.0;
                fprintf (fpo,"%f\n",med);
            }
        }
    }
}

```

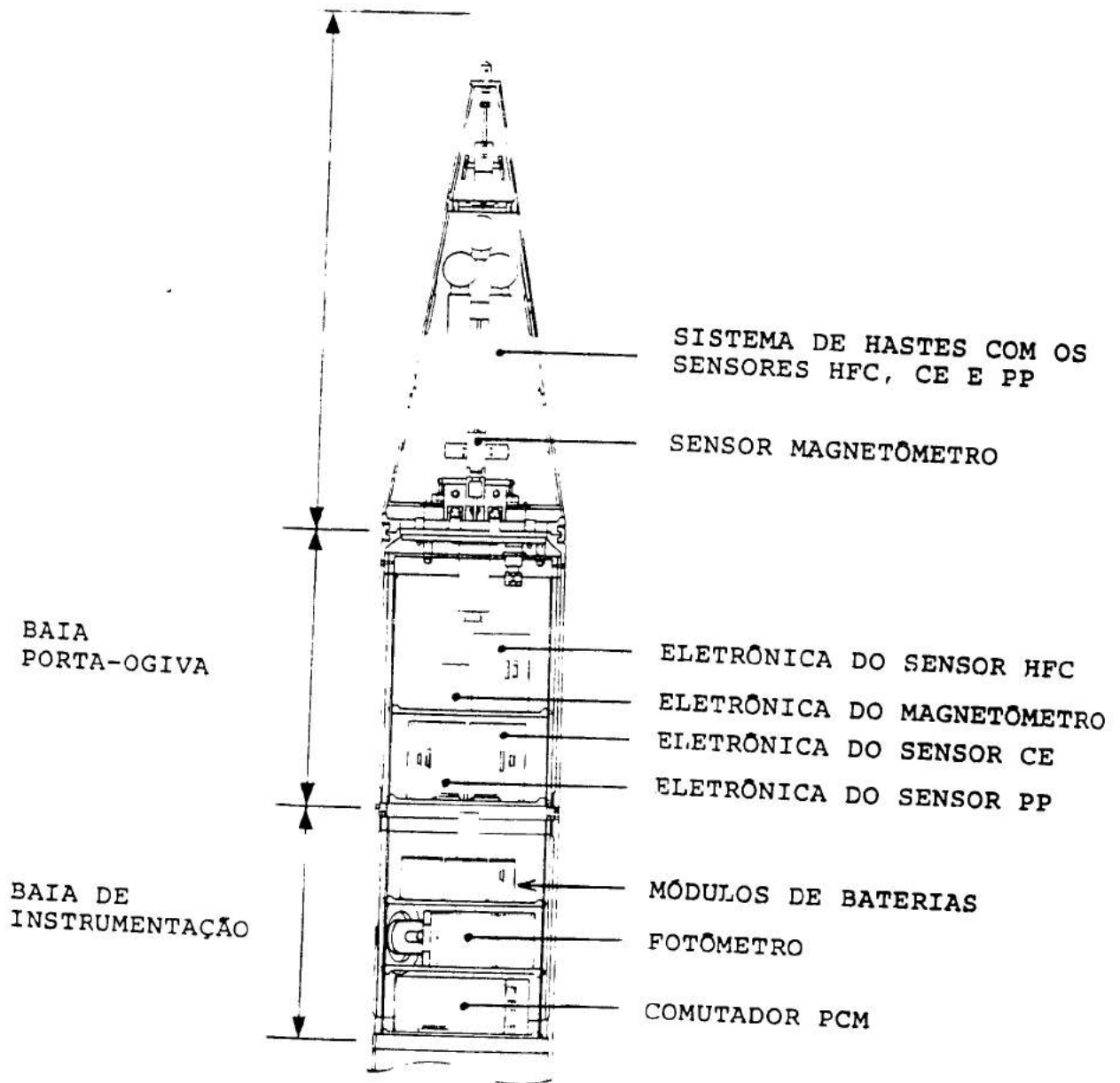
```
    }
  }
  else break;
}
}

/* rotina para leitura das médias e eliminação dos ruídos => fase2 */
calc_dds1(){
  while ( ( fscanf (fpi,"%s",x1) != EOF ){
    med = atof (x1);
    for ( i = 1; i <= 15; i++){
      if ( fscanf (fp,"%s %s",x1,x2) != EOF ){
        valor = atof (x2); alt = atof(x1);
        if (valor < 3*med && valor > med/3){
          fprintf (fpo,"%f %e\n",alt,valor);
        }
      }
    }
  }
}

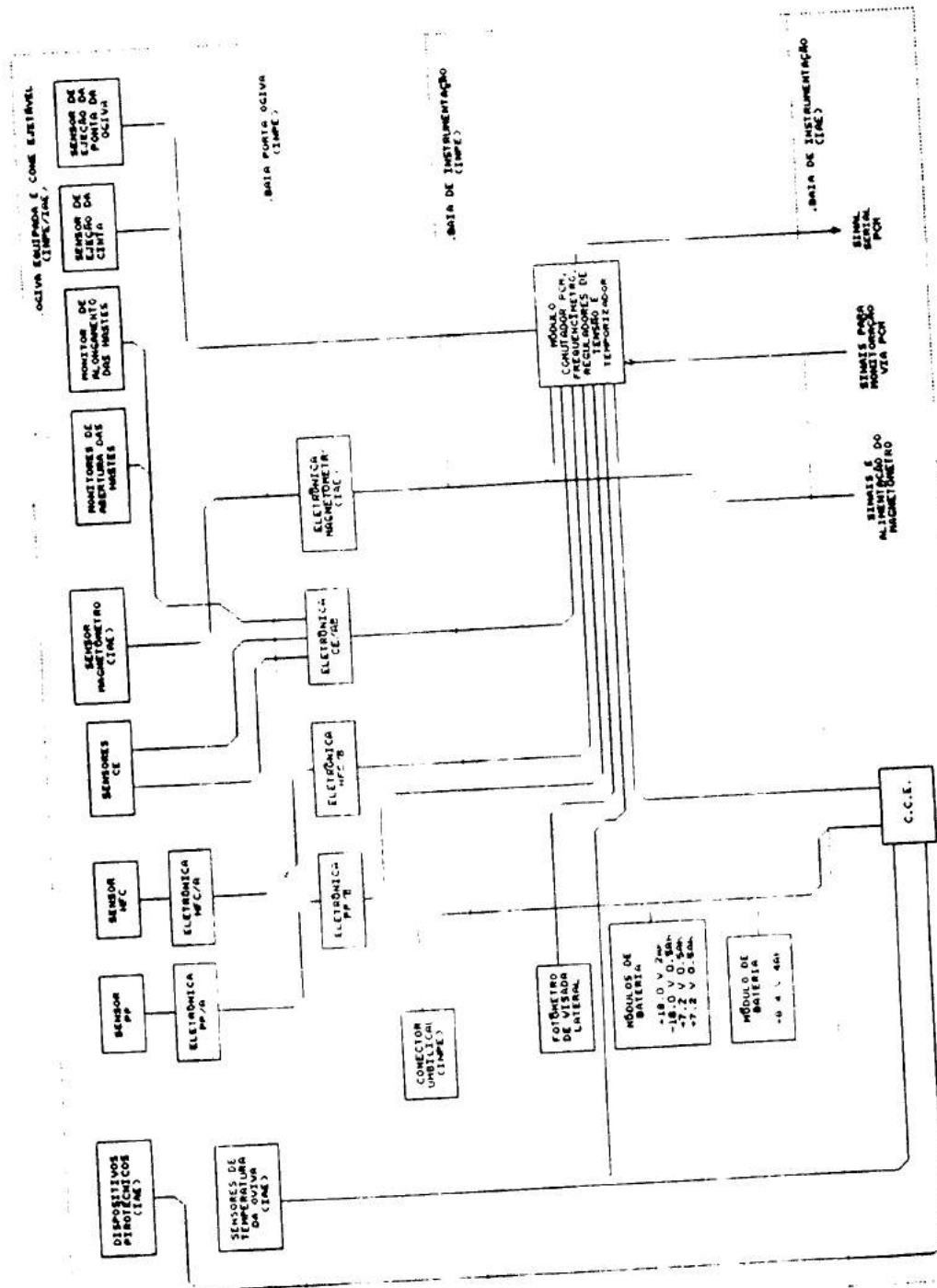
/* rotina para finalizar os arquivos utilizados */
fecha_arq(){
  fclose(fpi); fclose(fp); fclose(fpo);
}

/* programa principal */
main(){
  clrscr( );
  abre_arq( );
  calc_dds( );
  fecha_arq( );
  abre_arq1( );
  calc_dds1( );
  fecha_arq( );
}
```

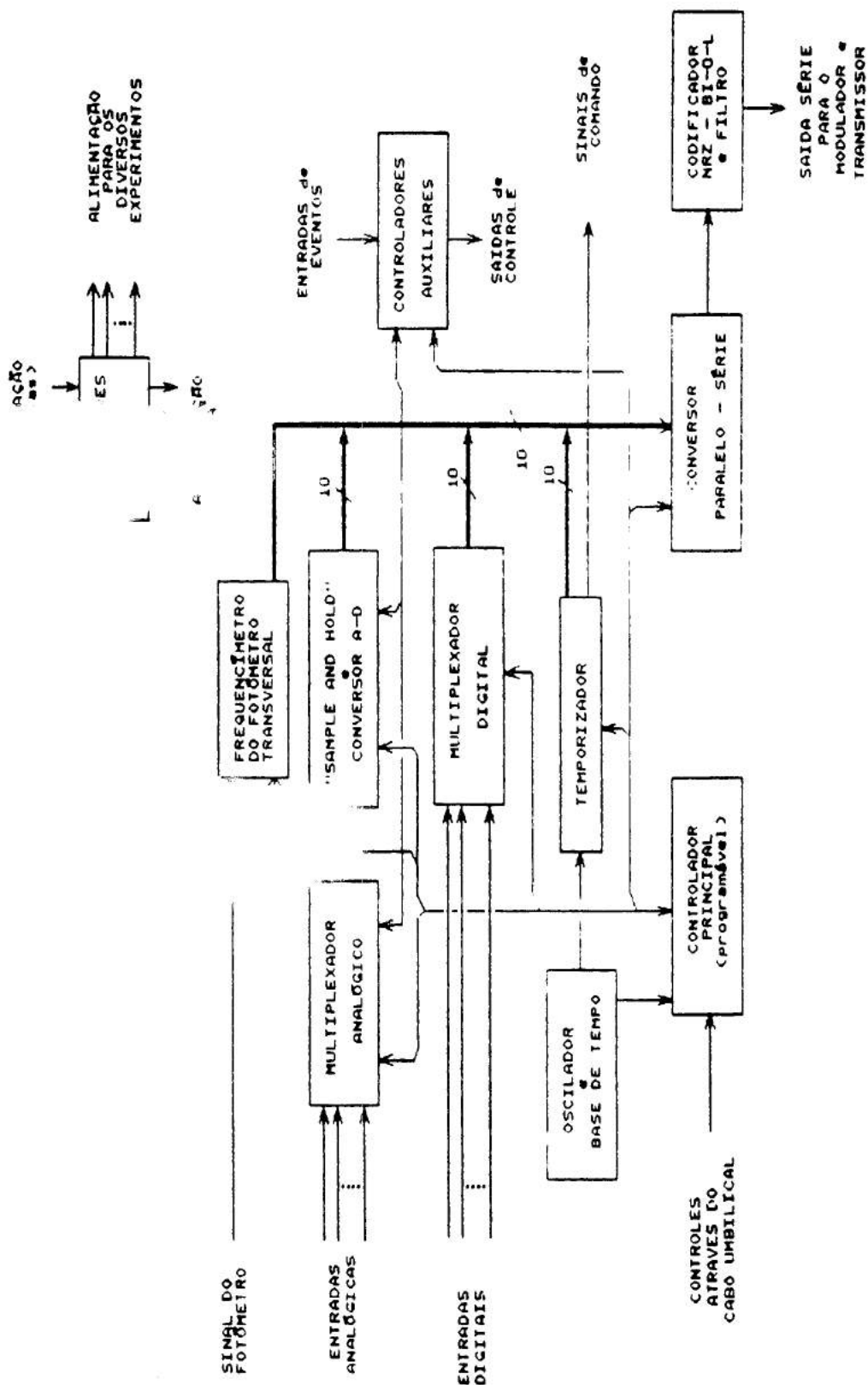
i.2- DISTRIBUIÇÃO DOS MÓDULOS COMPONENTES DA CARGA ÚTIL IONEX



i.3- ILUSTRAÇÃO DO DIAGRAMA EM BLOCOS SIMPLIFICADO DA CARGA ÚTIL IONEX



i.4- ILUSTRAÇÃO DO DIAGRAMA EM BLOCOS DO COMUTADOR PCM



VI - BIBLIOGRAFIA

Kirchhoff, V.W.J.H. - "Introdução à Geofísica Espacial". SP., Nova Stella. Editora da Universidade de São Paulo, Fapesp, 1991.

Souza, J.R. - "Estudo do Conteúdo Total Ionosférico". SJC., INPE, 1992. 70p. (INPE-5417-TDI/481)

Eras, A. - "Carga Útil IONEX - Operação Pacuruf". SJC., INPE, 1993.

Chryssafidis, M. - "Estudo Comparativo da Distribuição da Ne observada em baixas latitudes e predita pelo modelo IRI". SJC., INPE, 1992. (Dissertação de Mestrado)