

## PADRONIZAÇÃO EM COMPUTAÇÃO GRÁFICA

Antonio Tadeu Mendes Beraldo  
ITAUTEC Informática S.A.  
Caixa Postal 15156 - S.P. - Brasil

Gilberto José da Cunha  
ITAUTEC Informática S.A.  
Caixa Postal 15156 - S.P. - Brasil

### Resumo

É apresentada uma breve retrospectiva da evolução da tecnologia e aplicações na área de computação gráfica, destacando-se as motivações e iniciativas no sentido de se estabelecer padrões para o setor. É caracterizada a arquitetura de um sistema gráfico e as interfaces entre os diversos módulos, e as áreas de padronização. São apresentados os principais trabalhos de padronização em andamento, descrevendo-se sinteticamente cada um, identificando-se as correspondentes normas em elaboração ou já aprovadas e as respectivas entidades patrocinadoras.

Standards for Computer Graphics.

### Abstract

The evolution of Computer Graphics technology and applications is discussed briefly as well as the efforts on establishing standards for this purpose. The architecture of a typical computer graphics system is presented at modules level, interfaces and corresponding standard item. Basic standards are described, their specification, status, sponsorship entities and official documents issued.

### 1. HISTÓRICO DA PADRONIZAÇÃO EM COMPUTAÇÃO GRÁFICA

O interesse por Computação Gráfica surgiu praticamente junto com a própria computação. Os primeiros periféricos gráficos foram construídos ainda na década de 50:

- . Terminais desde 1951
- . Plotters desde 1953
- . Light pens desde 1958
- . Terminais coloridos desde 1962
- . Tablets desde 1964

Naturalmente tais equipamentos eram bastante rudimentares e seu uso extremamente restrito, dadas as limitações de tecnologia da época que os tornava muito caros, além de que o software com que se podia contar era restrito aos casos experimentais a que se destinavam tais equipamentos.

Os produtos para Computação Gráfica somente passaram a ser economicamente atraentes para o mercado a partir dos fins da década de 50 e início da década de 60, destacando-se nesta época os plotters de pena Calcomp e um pouco mais tarde os terminais 'storage' da Tektronix.

Estas duas famílias de produtos deram origem aos primeiros padrões 'de facto' em Computação Gráfica:

- . HCBS - Host Computer Basic Software, da Calcomp - conjunto de rotinas chamadas pelo programa de aplicação permitindo a geração de desenhos em plotter.
- . TCS - Terminal Control System, da Tektronix - conjunto de rotinas, também chamadas diretamente pelo programa de aplicação, permitindo a captura de dados gráficos do terminal, bem como a sua exibição na tela.

Naturalmente, no decorrer do tempo, surgiram outros fornecedores de equipamentos gráficos que ou aceitavam comandos do tipo HCBS ou TCS ou então tinham rotinas próprias de acionamento.

Com o aumento do nível tecnológico dos computadores, eles se tornaram mais potentes e mais econômicos, permitindo software mais sofisticado, sendo particularmente enfatizado o interesse no enriquecimento da interação homem-máquina, através de sistemas 'on-line' e com uma variedade de periféricos que agrupados em um mesmo local para efeitos ergonômicos, passaram a ser chamados de "work-station".

Neste contexto, surgem no início da década de 70, os chamados sistemas 'turn-key', cujo propósito era ser uma solução completa para problemas em áreas específicas, baseados

em uma integração íntima entre hardware e software, cada um dos quais com sua arquitetura própria, que era considerada o "segredo da casa". Isto deixava o usuário limitado às facilidades oferecidas pelo sistema tanto no que se refere ao software como aos equipamentos integrados pelo fabricante na sua arquitetura.

Com a grande difusão da informática, nos meados da década de 70, surge uma preocupação maior com a integração de soluções, o que consequentemente implica na integração de software e equipamentos, bem como o compartilhamento de um mesmo equipamento para resolver diferentes problemas.

Evidentemente compatibilizar todo esse universo de coisas não se restringe mais apenas a software de acionamento de periféricos gráficos, mas envolve a padronização de interfaces físicas, protocolos de comunicação, sistemas de gerenciamento de tarefas, software básico de acionamento de dispositivos e modelos de organização e apresentação de dados.

Esses assuntos são analisados em seus diversos aspectos dentro da ISO, através do subcomitê TC97/SC21 - Open System Interconnection. A parte especificamente relacionada com Computação Gráfica é tratada no Grupo de Trabalho WG2 do SC21. Nas diversas entidades internacionais existem estruturas especificamente voltadas para o assunto, destacando-se particularmente o X3H3, que é o Comitê Técnico para padronização Computação Gráfica da ANSI.

Um dos objetos de maior interesse de padronização em Computação Gráfica tem sido um núcleo gráfico que permita a comunicação do programa de aplicação com uma variedade de equipamentos, deixando-o independente dos dispositivos periféricos, o que possibilita facilidade de programação, portabilidade do software, com substanciais ganhos financeiros e de produtividade.

Em 1974, com a criação do GSPC (Graphics Standards Planning Committee) pelo SIGGRAPH da Association for Computer Machinery americana e a realização em 1976 de um "Workshop" sobre metodologia em computação gráfica em Seillac (França), patrocinado pelo IFIP, o esforço de padronização passou ser tratado a nível internacional e de maneira objetiva.

A especificação de um núcleo gráfico de base foi definida como o principal objetivo do "workshop" em Seillac, tendo recebido as atenções de vários grupos de pesquisadores de diversos países.

Dos resultados alcançados por esses grupos, destacam-se os de dois deles:

- O GSPC, com o desenvolvimento de duas versões da especificação do sistema CORE, uma em 1979, formando a última delas um núcleo gráfico com funções tridimensionais.
- O grupo do Instituto Alemão de Padronização (DIN), com a produção de várias versões do GKS (Graphical Kernel System), sis-

tema bidimensional com um conjunto de funções mais restrito do que o do CORE. E me nos polêmico.

O GKS e o CORE foram as principais propostas de padronização analisadas pelo ISO como candidatas a padrão gráfico internacional, que após várias reuniões decidiu-se, em 1979, a concentrar esforços no GKS.

O GKS tornou-se norma ISO (IS 7942) em agosto de 1985 e norma ANSI (X3.124-1985) em outubro de 1985, porém ao longo do processo de ser objeto de trabalho das entidades internacionais de padronização, os quais passaremos a descrever sucintamente, a seguir.

## 2. ARQUITETURA E INTERFACES DE UM SISTEMA GRÁFICO

### 2.a - Caracterização da Arquitetura

Os trabalhos de padronização na área de Computação Gráfica levaram a uma representação da arquitetura dos sistemas gráficos com forme apresentado na figura 1 onde estão representados os blocos funcionais e as interfaces correspondentes.

O bloco identificado como "Programa de Aplicação" concentra todas as rotinas específicas da aplicação, sendo o módulo encarregado de resolver os problemas específicos do usuário.

O módulo "Núcleo Gráfico" independente de dispositivo é o responsável por todo o processamento gráfico básico, oferecendo funções padronizadas pelas quais o programa de aplicação tem acesso a todos os dispositivos gráficos.

A "Interface" entre o programa de aplicação e o núcleo gráfico, representa a idéia de "binding", ou tradutor de linguagens, sendo o módulo que permite o acesso, por exemplo, de um programa escrito em FORTRAN às rotinas de um Núcleo Gráfico escrito em PASCAL.

A "Interface Gráfica Virtual" é o módulo básico de acesso aos periféricos gráficos.

O Núcleo Gráfico "conversa" diretamente com a Interface Gráfica Virtual, que se encarrega de traduzir os comandos e funções disponíveis no Núcleo para as linguagens específicas de cada dispositivo gráfico.

Isso é feito através de "drivers" que inclusive podem efetuar a simulação de funções existentes no Núcleo e não existentes no dispositivo. Note-se que para os dispositivos que forem fabricados com linguagem de comunicação definida pela Interface Gráfica Virtual, não haverá a necessidade da existência de um "driver".

Outro módulo componente da arquitetura é o "Metafile", que especifica a sintaxe e a semântica da definição de figuras independentemente do dispositivo. Compreende o codificador, responsável pela interpretação daqueles dados.

Finalmente, a estrutura apresenta um " Banco de Dados", que deverá conter informações

a nível de aplicação (especialmente CAD), foram matadas de acordo com especificações padronizadas de forma a permitir troca de informações entre sistemas a nível de aplicação.

## 2.b - Propostas de Padrões

### - Núcleo Gráfico Independente de Dispositivo

A padronização de núcleo gráfico independente de dispositivo foi o primeiro assunto a ser objeto de trabalhos os quais tiveram início na primeira metade da década de 70.

Nesta área existe um padrão oficial a nível de ISO/ANSI que é o GKS-Graphical Kernel System (IS 7942/X3.124-1985). Esta norma define funções gráficas básicas em duas dimensões estando em fase de normatização sua extensão para 3 dimensões conhecida como GKS 3D.

Além do GKS 3D, está também em fase de trabalhos de normatização uma outra especificação de núcleo gráfico voltada para aplicações mais complexas, cobrindo também a estruturação hierárquica dos dados gráficos PHIGS - Programmer's Hierarchical Interactive Graphics System.

### - GKS : Graphical Kernel System

O objetivo do GKS é definir uma interface portátil e independente de "hardware" para aplicações gráficas bidimensionais com entrada interativa.

As funções do GKS são definidas de forma independente do tipo de dispositivo, aplicação ou linguagem de programação.

A interface com o programa de aplicação, definida pelo GKS, refere-se ao modelo de sintaxe que deve ser utilizado para fazer referência às funções componentes do núcleo gráfico. Essa interface padroniza também a sequência de chamadas das funções do núcleo.

A interface com o dispositivo refere-se ao protocolo utilizado para a comunicação entre as funções dependentes de dispositivo e as independentes. Ela define um protocolo para o "driver" que é consistente para todos os dispositivos gráficos.

Assim, a proposta GKS provê duas características principais :

- . Padronização do ponto de vista do programador de aplicação;
- . Independência de dispositivo, no que diz respeito aos periféricos gráficos .

Essas características trazem vantagens econômicas aos usuários em relação aos custos de :

- . Desenvolvimento de "software" de aplicação;
- . Treinamento de programadores;
- . Adaptação a novos tipos de periféricos; etc..

As aplicações baseadas em GKS podem ir desde sistemas com exigências simples de saída até sistemas de grande interatividade, sem pre bidimensionais.

O GKS provê funções para tratamento de :

- . Saída gráfica : dispõe de seis tipos de primitivos cujos atributos podem ser definidos individualmente ou em grupo, em cada estação de trabalho.
- . Entrada gráfica : seis classes de entrada, utilizáveis sob três modos diferentes (por pedido, amostragem ou fila de eventos).
- . Estações de Trabalho : Trata cada estação de trabalho e seus atributos separadamente.
- . Transformações : Permitem controle de transformações de coordenadas (de um sistema de coordenadas dependente da aplicação para um sistema virtual normalizado e deste para as coordenadas do dispositivo gráfico).
- . Segmentação : Permitem o agrupamento de primitivos e sua identificação.
- . Metafile : Possibilita o armazenamento e transporte de informação gráfica a longo prazo.
- . Controle : Funções de abertura do núcleo, das estações de trabalho, adiamento e regeneração, etc..

O GKS pode se comunicar com quaisquer dispositivos através de "drivers" adequados, inclusive com dispositivos padrão CGI e com o CGM, explicados mais abaixo.

Existem diversas implementações comerciais do padrão GKS e aplicativos baseados em núcleos padrão GKS, sendo sua difusão bastante ampla a nível internacional, destacando-se, por exemplo, sua adoção como IFIPS-Federal Information Processing Standard nos USA, com base na sua aprovação como norma ANSI.

### - GKS 3D : Graphical Kernel Systems - 3D

Cobre as funções do GKS-standard que é bidimensional, estendidas para o contexto de 3 dimensões, com total compatibilidade.

Abrange naturalmente conceitos específicos de 3 dimensões, como por exemplo projeções, eliminação de linhas escondidas, de forma a permitir manipulações de modelos tridimensionais.

Além disso implementa o conceito de estação de trabalho tridimensional, para entrada de dados e interação.

### - PHIGS : Programmer's Hierarchical Interactive Graphics Standard

O seu objetivo é satisfazer de modo eficiente necessidades não atendidas, em parte ou totalmente, pela definição atual do GKS :

- . Alto grau de interatividade
- . Estruturação hierárquica dos dados gráficos
- . Possibilidade de modificação em tempo real dos dados gráficos
- . Adaptação à ambientes distribuídos
- . Dados gráficos em duas e três dimensões

Algumas áreas de aplicação são :

- . Computer Aided Engineering
  - Projetos de circuito impresso, integrado, VLSI
  - Sistemas desenhadores (drafting systems)
  - Projeto mecânico tridimensional
- . Controle
  - Monitoramento e controle de processos
  - Controle de tráfego
  - Sistemas de gerenciamento
- . Simulação
  - Processos
  - Robótica
  - Tráfego
  - Meteorologia
- . Científico
  - Modelagem de moléculas
  - Análise de movimento

O projeto do PHIGS tenta integrar as seguintes áreas funcionais: primitivos de saída, atributos, transformações, organização da estrutura de desenho, estações de trabalho e entrada.

O seu conjunto de funções tem recebido muita influência do GKS e do CORE, sendo que toda a sua parte bidimensional deverá buscar total compatibilidade com as correspondentes funções do GKS.

#### - Interface Gráfica Virtual

A padronização em termos de interface gráfica virtual está restrita a uma única proposta que está em processo de normatização tanto na ANSI quanto na ISO que é:

#### - CGI : Computer Graphics Interface

A proposta de padrão CGI define uma interface bidimensional independente de "hardware" entre os níveis mais altos de um pacote gráfico e o "driver" de dispositivo específico de entrada ou saída.

O CGI suporta três classes de dispositivos virtuais:

- . Dispositivos de saída.
- . Dispositivos de entrada.
- . Dispositivos de entrada e saída.

Implementações específicas do CGI geralmente serão limitadas às capacidades de entrada e saída necessárias para fazer eficientemente a interface com "drivers" para dispositivos físicos específicos. Já que estas capacidades e necessidades podem variar bastante, o padrão CGI define um conjunto básico de elementos de entrada e saída e conjuntos de opções que podem ser acrescentados aos elementos básicos.

Os elementos no CGI podem ser agrupados em:

- . Controle: Para configuração e inicializações.
- . Primitivos: Um repertório estendido de primitivos em relação ao GKS.

. Atributos: Definidos em grupos ou individualmente, como no GKS.

. Entrada: O CGI suporta cinco classes de entrada (das seis do GKS) e quatro modos de entrada (os três suportados pelo GKS e mais um modo de entrada que exige interação sincronizada entre o programa de aplicação e o operador).

#### - Especificação para Armazenamento de Dados Gráficos

Neste caso também existe somente uma proposta de norma em fase de padronização, devendo ser a próxima norma a ser aprovada tanto em termos de ISO quanto de ANSI, que é o CGM.

#### - CGM: Computer Graphics Metafile

O padrão proposto do CGM define uma estrutura de dados independente de "hardware" para armazenamento e transferência de informação gráfica.

Um "metafile" CGM consiste de uma lista de elementos escrita por um gerador CGM ao tempo em que o arquivo é criado. O arquivo é convertido por um interpretador CGM que passa os elementos para uma forma que pode ser processada pelo sistema receptor.

O "metafile" CGM define:

- . Elementos de controle
- . Primitivos
- . Atributos

A terminologia e estrutura geral do padrão CGM proposto aproximam-se bastante das definidas para o GKS. Foram feitos esforços para fazer o CGM compatível também às informações gráficas geradas pelo sistema CORE, mas o CGM é claramente um membro da família de padrões formada pelo GKS, PHIGS, CGM e CGI.

O CGM é destinado, de fato, para ser um padrão conjugado ao CGI, servindo como fonte de informação para imagens a serem exibidas ou exemplo, um espaço de coordenadas virtuais para posicionar os primitivos que compõem uma imagem.

#### - Formatação para Troca de Dados entre Sistemas Aplicativos

A troca de dados entre sistemas aplicativos envolvendo computação gráfica - Sistemas CAD tem sido objeto de estudos essencialmente na ANSI, que inclusive possui norma aprovada a respeito, contudo, o assunto ainda é objeto de polêmicas, haja visto que a referida norma já está na sua terceira versão.

Na verdade existem trabalhos no sentido de evoluir a norma IGES para uma norma mais abrangente e mais voltada para Automação Industrial: Product Data Exchange Standard-PDES.

Além desses trabalhos, existem outros na mesma direção em andamento na ISO, com propostas de diversos países. Nós restringiremos contudo a uma breve descrição do IGES.

- IGES : Initial Graphics Exchange Specification

O IGES define formatos padrão de arquivos para armazenamento e transmissão de informação do banco de dados de um sistema CAD para o banco de dados de outro sistema CAD (que pode ser diferente do anterior) para a geração de desenhos e outros tipos de representação gráfica.

É importante notar que a saída gráfica pode ser apenas uma parte subjacente da aplicação do IGES. O propósito principal de uma transferência de dados via IGES pode ser, por exemplo, a geração - por um sistema CAM - de códigos de comando numérico para fabricação de uma peça definida em um sistema CAD. A formatação dos dados produzidos pelo sistema de projeto pode ser diferente do necessário como entrada para a geração dos códigos de comando numérico.

Como interface entre sistemas, o IGES é implementado por dois módulos de "software": um pré-processador que converte dados do sistema origem em entidades IGES e um pós-processador que traduz arquivos IGES em uma forma que pode ser entendida pelo sistema destino.

Como o PHIGS, o padrão IGES cria um banco de dados hierárquico. Montagens podem ser quebradas em submontagens e estas, em peças, muitas das quais podem ser comuns a outras montagens ou submontagens.

O pós-processador IGES pode fazer acesso aleatoriamente às entidades referenciadas por outras entidades à medida em que atravessa a estrutura hierárquica.

As entidades IGES podem ser divididas em três categorias:

- . Geometria : primitivos gráficos,
- . Anotação : parte primitivos, parte atributos,
- . Estrutura e definição : entidades de controle.

As entidades de geometria e definição são dadas em espaços de definição bidimensionais ou tridimensionais locais à aplicação, sendo então transformados para um espaço modelo do IGES.

Dois formatos de arquivos IGES foram padronizados : um formato de 80 colunas por registro e um formato binário, mais recente, que é uma sequência de "bits", reduzindo o espaço necessário de armazenamento a 50% do anterior.

No contexto dos demais padrões de computação gráfica, o IGES aproxima-se às funções de um "metafile" de captura de figuras no sentido em que contém todas as informações necessárias para gerar desenhos completos, não importando o tipo de sistema usado para criar a definição do objeto projetado, nem o tipo de sistema que converterá estes dados em imagens.

- Interface para Videotexto

A padronização de interfaces para a troca

de informações de sistemas videotexto tem merecido especial interesse, tendo sido objeto de uma especificação em 1979 pelo governo Canadense e mais tarde pelos USA gerando uma outra especificação, por iniciativa industrial, que é o NAPLPS.

- MPLPS : North American Presentation Level Protocol Syntax

Idealizado especificamente para dispositivos gráficos tipo "raster", ele padroniza um método de compressão da informação para transmissão e armazenamento.

Assume a existência de inteligência local no dispositivo de exibição transmitindo, então, instruções com as quais o receptor pode gerar os elementos de desenho.

O seu método de exibição de dados gráficos é baseado num método de codificação gráfica chamado "alfageométrico". Assim, as instruções de descrição do desenho são enviadas ao terminal receptor, interpretadas pelo microprocessador local e o desenho é gerado.

As instruções descrevem formas geométricas compostas de pontos, linhas, arcos, áreas e polígonos.

O MPLPS foi desenvolvido pelo governo canadense em conjunto com a AT&T e outras companhias de comunicação como base para aplicações de teletexto e videotexto, sendo uma extensão ao Canadian Telidon Videotex System.

### 3. SITUAÇÃO ATUAL E TENDÊNCIAS

O estado atual e tendências das normas de Computação Gráfica pode ser descrito com base na Tabela I, onde estão apresentados o estágio de desenvolvimento de cada norma na ISO, que é a entidade internacional de normatização mais abrangente.

Para melhor compreensão da Tabela I é conveniente conhecer algumas informações sobre o que representa cada fase do processo de normatização. O processo parte de uma proposta de trabalho e vai evoluindo para fases em grau crescente de consenso, até se tornar norma oficial:

FASE I - New Work Item - NWI

- . Início de novo projeto. Deve ter um draft de documento.
- . Circula consulta por 3 meses para que os membros decidam se aceitam o novo item, se o suportam e se forneceram recursos.
- . A produção do documento base leva de 5 a 8 meses.
- . X3H3 forneceu os db para : CGM, CGI e PHIGS.
- . Não está representada na Tabela I.

FASE II - Working Draft - WD

- . A partir do db são preparados os WD para circular pelos membros por período em geral de 3 meses.
- . Existem reuniões dos WG cada 9/10 meses.
- . Este estágio leva de 6 a 18 meses dependendo do db, consenso, etc..

TABELA - I

NORMA	WD	DP	DIG	IS
GKS	1980	1981	1982	1985
GKS-3D		3/1986	6/1986	8/1987
CGM	1983	1984	8/1986	1/1987
CGI	10/1985	9/1986	12/1988	12/1989
PHIGS	9/1985	9/1986	12/1987	12/1988
Teste de Conformidade	9/1986		9/1987	
LANGUAGE BINDINGS:				
=====				
GKS LB-FORTRAN			10/1985	10/1986
GKS LB-PASCAL			9/1985	9/1986
GKS LB-ADA		11/1985	6/1986	9/1987
GKS LB-C	8/1985		3/1986	
GKS-3D LB-FORTRAN	7/1985	3/1986	2/1987	2/1988
GKS-3D LB-PASCAL	11/1985	4/1986	2/1987	2/1988
GKS-3D LB-ADA	9/1986	4/1987	10/1987	9/1988
PHIGS LB-FORTRAN	9/1985	9/1986	12/1987	12/1988
PHIGS LB-PASCAL	12/1986	9/1987	9/1988	9/1989
PHIGS LB-ADA	9/1985	9/1986	12/1987	12/1988

FASE III - Draft Proposal - DP

- Após o WD estar bem completo o SC21 pode registrá-lo como DP.
- Pode ser via reunião do SC21 ou votação em 3 meses dos membros.
- São feitas circulações e reuniões para comentários.
- Podem existir vários ciclos de 3 meses de votação.
- O WG2 deseja que este estágio não leve mais do que 12/14 meses.

FASE IV - Draft International Standard

- Após o consenso técnico do DP e sua escrita em formato ISO, ele é circulado pela ISO por 6 meses entre os membros.
- Este estágio leva de 9 a 12 meses após o texto ser recebido pela ISO.
- Podem surgir críticas que modifiquem muito o documento e gere um novo ciclo de DIS.

FASE V - International Standard

Após consenso o texto deve ser aprovado no conselho da ISO e ser publicado.

Observações :

- O IGES e NAPLPS não aparecem na tabela, pois são tratados somente no contexto da ANSI.
- Alguns dados não comparecem, pois não foi possível obtê-los.
- Pode haver diferenças de ordem de alguns meses para a mesma data, entre dados de uma fonte e outra devido a revisões e/ou por se referirem a votações adicionais na mesma fase.

A ANSI tem um processo semelhante de normatização e a evolução das normas nessa entidade tem seguido um cronograma mais ou menos semelhante ao da ISO.

Pode-se deduzir que em termos de núcleo gráfico, a única norma oficial e "de facto" existente é o GKS, pois foi aprovada na ISO em 10/1985, ocasião em que foi também aprovada na ANSI.

A operacionalização do GKS e sua tendência de evolução pode ser vista pela previsão de normatização de sua extensão: GKS-3D e pelo cronograma de trabalho previsto para os 'bindings' para diversas linguagens.

A próxima norma a ser aprovada deverá ser o CGM, tanto na ISO quanto na ANSI.

O PHIGS vem merecendo atenção crescente, porém há polêmica quanto à sua especificação, especialmente no que se refere a garantia de sua compatibilidade com o GKS.

O IGES é uma norma em evolução existindo discussões específicas em diversos países, de vendo evoluir para uma formulação mais ampla.

Todos estes aspectos deverão ser compatibilizados a médio prazo bem como outros relacionados dentro do modelo OSI - Open System Intercommunication.

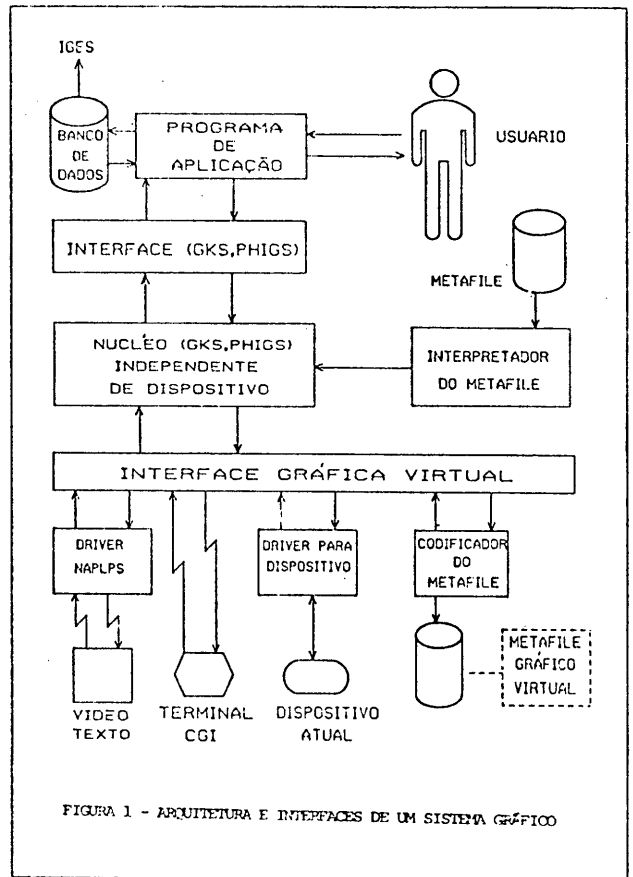


FIGURA 1 - ARQUITETURA E INTERFACES DE UM SISTEMA GRÁFICO

4. REFERÊNCIAS

- 4.a - Computer Graphics and Applications - Agosto/86 e Outubro/85
- 4.b - Documentos de referência ISO - IS7942 (GKS); DP8805 (GKS3D); DP8651 (GKS LB-Fortran); TC97/N=447 (PHIGS); TC97/N561 (CGI)
- 4.c - Padronização em Computação Gráfica: Propostas e Tendências - Trabalho apresentado no Congresso Nacional de Informática 1985 - Gilberto J. Cunha, Fernando T. Barreiros e André L. Battaglia
- 4.d - The Managers Guide to CAD/CAM Standards for Integration - CAD/CAM Alert - 1986