



República Federativa do Brasil
Ministério do Desenvolvimento, Indústria
e do Comércio Exterior
Instituto Nacional da Propriedade Industrial.

(21) **PI0619245-9 A2**

(22) Data de Depósito: 17/11/2006
(43) Data da Publicação: 20/09/2011
(RPI 2124)



(51) *Int.Cl.:*
G06F 3/02
G06F 3/048

(54) **Título:** TECLADO VIRTUAL INTELIGENTE

(30) **Prioridade Unionista:** 15/12/2005 US 11/304.240

(73) **Titular(es):** MICROSOFT CORPORATION

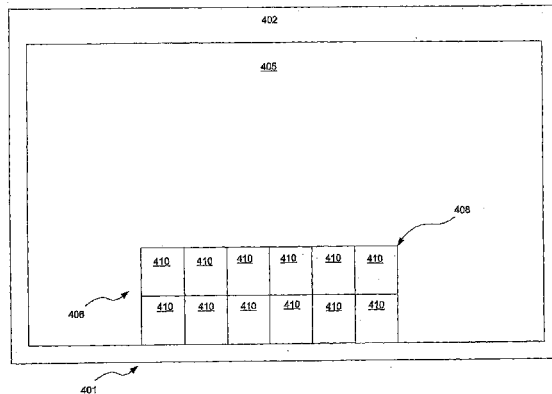
(72) **Inventor(es):** Krishna V. Lotipalli

(74) **Procurador(es):** NELLIE ANNE DANIEL SHORES

(86) **Pedido Internacional:** PCT US2006044741 de
17/11/2006

(87) **Publicação Internacional:** WO 2007/070223 de
21/06/2007

(57) **Resumo:** TECLADO VIRTUAL INTELIGENTE. São descritas métodos para uso de um teclado virtual. É provido um teclado virtual possuindo uma ou mais teclas virtual com um símbolo apresentado nas uma ou mais teclas virtual. Um modificador pode ser aplicado a um dos símbolos apresentados sobre uma das teclas virtual através da seleção da tecla virtual, seguida pelo provimento de uma entrada direcional. Em uma modalidade, diferentes entradas direcionais são associadas a diferentes modificadores de forma a que diferentes símbolos sejam alimentados, dependendo da entrada direcional. Em uma modalidade, um usuário pode provar a entrada de direção por meio de um toque e arraste direcional de um seletor através de uma tela sensível ao toque. Em uma modalidade, uma pluralidade de símbolos associados a uma pluralidade de teclas modificadoras são tornados visíveis após a tecla virtual ser selecionada, de forma a que o usuário possa determinar visualmente qual entrada direcional irá prover o símbolo desejado.





PI0619245-9

"TECLADO VIRTUAL INTELIGENTE"

Embasamento

Os computadores portáteis, tais como palms, laptops, PDAs e PCs de bolso, apresentam alguns problemas de projeto desafiadores. Por um lado, é desejável que eles sejam de baixo peso e fácil portabilidade, sendo de uso simples; por outro lado, é desejável que eles propiciem uma longa vida para a bateria, tenham grande poder de processamento e apresentem um visor de grande porte. Tais exigências conflitantes de projeto tornam difícil a produção de um computador portátil ideal.

Para minimizar o peso e volume, um recurso tem sido o de usar um teclado virtual ou sensível ao toque no visor para a entrada ou alimentação de dados. Um método para o provimento de um teclado virtual consiste em prover um visor com um recurso de sensibilidade ao toque. Uma parte do visor, que pode consistir de uma fração variando de 1 a $1/N$, em que N é um número inteiro positivo, pode ser configurada para apresentar símbolos, tais como os caracteres de um alfabeto, em teclas virtuais em uma disposição que se assemelha a um teclado. Ao pressionar o local apropriado no visor o símbolo em questão é selecionado. Pela seleção repetida de símbolos, um indivíduo pode alimentar dados de forma conhecida.

No entanto, como poderá ser apreciado, existe uma quantidade limitada de espaço disponível no visor de um dispositivo portátil. O espaço disponível fica ainda mais reduzido se uma parte do visor deve ser usada para apresentar as

seleções do usuário. Ademais, a redução do tamanho dos símbolos abaixo de um certo ponto é impraticável devido à dificuldade em se distinguir visualmente a diferença entre os símbolos. Portanto, pode ser difícil apresentar todos os
5 símbolos que poderiam, de outra forma, ser desejáveis a serem mostrados. Tal problema é adicionalmente exacerbado com certos idiomas que incluem símbolos que podem ou não incluir um acento.

Uma solução consiste em prover teclas modificadoras, tais como a tecla "shift". Ao selecionar a tecla modificadora apropriada, as escolhas disponíveis para cada tecla virtual pode se modificar. No entanto, isto requer a adição de seleções. O uso de uma tecla modificadora em um teclado "hard" normal tipicamente não constitui um problema, pois
15 ambas as mãos podem ser usadas simultaneamente. Todavia, um teclado virtual é tipicamente operado através do uso do mesmo dedo ou dispositivo apontador para a seleção das teclas virtuais subsequentes. Dessa forma, demandar que o usuário primeiramente selecione uma tecla modificadora e a seguir
20 selecione a tecla virtual desejada pode reduzir de forma indesejável a velocidade de alimentação de dados.

Resumo

Um teclado virtual, possuindo um certo número de teclas virtuais, pode ser provido em um visor / visor sensível ao toque e cada tecla virtual pode incluir um símbolo. O
25 usuário pode selecionar uma das teclas virtuais pressionando um seletor na tela sensível ao toque na posição apropriada. O usuário pode selecionar uma versão modificada do símbolo

selecionado inicialmente "arrastando" o seletor em uma direção alinhada com um modificador. A seleção pode ser finalizada ou alimentada elevando-se o seletor da tela sensível ao toque. Para prover um "feedback" para o usuário, os símbolos
5 potenciais associados ao símbolo selecionado inicialmente podem ser apresentados em torno da seleção inicial de forma a que o arraste do seletor na direção do símbolo desejado irá permitir ao usuário selecionar tal símbolo.

O presente resumo é provido para introduzir uma
10 seleção de conceitos de uma forma simplificada, os quais serão adicionalmente descritos mais adiante na Descrição Detalhada. O presente resumo não se destina a identificar características ou recursos chave ou essenciais da matéria objeto reivindicada, nem se destina a limitar o escopo da matéria
15 objeto reivindicada.

Breve Descrição dos Desenhos

A presente invenção esta ilustrada, como exemplo e não limitação, nas figuras anexas, nas quais referências numéricas semelhantes indicam elementos similares e nas quais:

20 A Figura 1a ilustra um diagrama esquemático de um ambiente de computação digital de uso geral, no qual certos aspectos da presente invenção podem ser implementados.

As Figuras 1b a 1m apresentam um ambiente de computação de uso geral que dá suporte a um ou mais aspectos da
25 presente invenção.

A Figura 2 ilustra um diagrama esquemático de um ambiente de computação do tipo palm que dá suporte a um ou mais aspectos da presente invenção.

A Figura 3 ilustra um diagrama esquemático de um ambiente de computação do tipo portátil em que podem ser implementados um ou mais aspectos da presente invenção.

5 A Figura 4 ilustra uma modalidade de um teclado virtual em um visor no qual um ou mais aspectos da presente invenção podem ser implementados.

A Figura 5 ilustra uma tecla virtual apresentando um símbolo no qual um ou mais aspectos da presente invenção podem ser implementados.

10 As Figuras 6a e 6b ilustram modalidades de um conjunto de símbolos provido ao longo de um perímetro de um primeiro símbolo de acordo com um ou mais aspectos da presente invenção.

15 As Figuras 7 e 8 ilustram métodos para a seleção de símbolos associados a um primeiro símbolo de acordo com um ou mais aspectos da presente invenção.

A Figura 9 ilustra um método para alimentação de um símbolo de acordo com um ou mais aspectos da presente invenção.

20 Descrição Detalhada

O uso de um teclado virtual apresenta certas vantagens. Como exemplo, podem ser providas APIs para permitir que outros aplicativos controlem e monitorem o uso do visor na tela. Além disso, a cor e aparência de um teclado virtual
25 podem ser facilmente customizadas. Ademais, um teclado virtual similar pode ser usado para uma diversidade de aplicativos e hardware com modificações simples do tamanho no visor das teclas virtuais e do tamanho da fonte usada para os

símbolos. Outras vantagens ficarão claras após a leitura da descrição que se segue.

Ambiente Geral de Computação

A Figura 1a ilustra um exemplo de um ambiente de
5 um sistema de computação 100 adequado no qual pode ser im-
plementada a invenção. O ambiente de sistema de computação
100 constitui apenas um exemplo de um ambiente de computação
adequado e não se destina a sugerir qualquer limitação quan-
to ao escopo de utilização ou de funcionalidade da invenção.
10 O ambiente de computação 100 também não deve ser interpreta-
do como possuindo qualquer dependência ou exigência relacio-
nadas a qualquer um dos, ou combinação dos, componentes i-
lustrados no ambiente operacional 100 exemplar.

A invenção pode ser operada com vários outros am-
15 bientes ou configurações de sistemas operacionais de uso ge-
ral ou de propósito específico. Os exemplos de sistemas, am-
bientes e/ou configurações de computação que podem ser ade-
quados para uso com a invenção incluem, porém não ficam li-
mitados a, computadores pessoais, computadores servidores,
20 dispositivos de mão / palms ou do tipo laptop, sistemas de
múltiplos processadores, sistemas baseados em microprocessa-
dores, conjuntos do tipo set top, componentes eletrônicos
programáveis, PCs em rede, minicomputadores, computadores
mainframe, ambientes de computação distribuídos que incluem
25 quaisquer dos sistemas ou dispositivos acima e similares.

A invenção pode ser descrita no contexto geral de
instruções para execução por computadores, tais como módulos
de programas, sendo executadas por um computador. De um modo

geral, os módulos de programas incluem rotinas, programas, objetos, componentes, estruturas de dados, etc., que efetuam tarefas específicas ou implementam tipos de dados abstratos específicos. A invenção pode também ser praticada em ambientes de computação distribuídos, em que as tarefas são efetuadas por dispositivos de processamento remotos que estão ligados através de uma rede de comunicações. Em um ambiente de computação distribuído, os módulos de programas podem estar localizados em meios de armazenamento de computadores locais e remotos, incluindo dispositivos de armazenamento de memória.

Fazendo referência à Figura 1a, um sistema exemplar para a implementação da invenção inclui um dispositivo de computação de uso geral na forma de um computador 110. Os componentes do computador 110 podem incluir, porém não ficam limitados a, uma unidade de processamento 120, uma memória de sistema 130 e um barramento de sistema 121 que acopla vários componentes do sistema, incluindo a memória do sistema, à unidade de processamento 120. O barramento do sistema 121 pode ser qualquer um dentre vários tipos de estruturas de barramento, incluindo um barramento de memória ou um controlador de memória, um barramento de periféricos e um barramento local, usando qualquer uma dentre uma variedade de estruturas de barramento. Como exemplo, mas não limitação, tais estruturas incluem o barramento ISA (Industry Standard Architecture), o barramento de micro canal (MCA), o barramento ISA ampliado (EISA), o barramento local VESA (Video Eletronics Standards Association) e o barramento PCI (Peri-

pheral Component Interconnect) também conhecido como barramento mezanino.

O computador 110 inclui tipicamente uma diversidade de meios para leitura por computador. Os meios para leitura por computador podem consistir de quaisquer meios disponíveis que possam ser acessados pelo computador 110 e incluem meios voláteis e não voláteis, meios removíveis e não removíveis. Como exemplo, mas não limitação, os meios para leitura por computador podem incluir meios de armazenamento em computador e meios de comunicação. Os meios de armazenamento em computador incluem meios voláteis e não voláteis, e removíveis e não removíveis, implementados em qualquer método ou tecnologia para armazenamento de informações, tais como instruções para leitura por computador, estruturas de dados, módulos de programa, ou outros dados. Os meios para armazenamento em computador incluem, porém não ficam limitados a, RAM, ROM, EEPROM, memória flash, ou outras tecnologias de memória, CD-ROM, discos versáteis digitais (DVD), ou outro armazenamento em discos ópticos, cassetes magnéticos, fita magnética, armazenamento em disco magnético ou outros dispositivos de armazenamento magnéticos, ou qualquer outro meio que possa ser utilizado para armazenar as informações desejadas e que possam ser acessadas pelo computador 110. Os meios de comunicação incorporam tipicamente instruções para leitura por computador, estruturas de dados, módulos de programas, ou outros dados em um sinal de dados modulado, tais como uma onda portadora, ou outro mecanismo de transporte e incluem quaisquer meios para transporte de informações. O

termo "sinal de dados modulado" significa um sinal que tem uma ou mais de suas características ajustadas ou modificadas de maneira a codificar informações no sinal. Como exemplo, mas não limitação, os meios de comunicação incluem meios ca-
5 beados, tais como uma rede de cabos, ou uma conexão direta por cabos, e meios sem fio ou "wireless" tais como meios acústicos, de RF, infravermelho e outros meios sem fio. Combinações dos acima mencionados também devem ser incluídas no escopo de meios para leitura por computador.

10 A memória do sistema 120 inclui meios de armazenamento em computador na forma de memória volátil e/ou não volátil, tal como a memória apenas para leitura (ROM) 131 e memória de acesso aleatório (RAM) 132. Um sistema básico de alimentação / saída (BIOS) 133, contendo as rotinas básicas
15 que auxiliam a transferência de informações entre elementos no interior do computador 110, tais como durante a inicialização, ficam tipicamente armazenadas na ROM 131. A RAM 132 contém tipicamente dados e/ou módulos de programas que estão imediatamente acessíveis e/ou estão sendo presentemente ope-
20 rados pela unidade de processamento 134, programas aplicativos 135, outros módulos de programas 136 e dados de programas 137.

O computador 110 pode também incluir outros meios de armazenamento em computador removíveis / não removíveis,
25 voláteis / não voláteis. Apenas como exemplo, a Figura 1a ilustra um drive de disco rígido 141 que lê ou grava meios magnéticos não removíveis não voláteis, um drive de disco magnético 151 que lê ou grava um disco magnético removível

não volátil 152 e um drive de disco óptico 155 que lê ou grava um disco óptico removível não volátil 156, tal como um CD-ROM ou outro meio óptico. Outros meios de armazenamento em computador removíveis / não removíveis, voláteis / não voláteis que podem ser usados no ambiente operacional exemplar incluem, porém não ficam limitados a, cassetes de fita magnética, placas de memória flash, DVDs, fita de vídeo digital, RAM de estado sólido, ROM de estado sólido e similares. O drive de disco rígido 141 está tipicamente conectado ao barramento do sistema 121 através de uma interface de memória não removível, tal como a interface 140, enquanto o drive de disco magnético 151 e o drive de disco óptico 155 estão tipicamente conectados ao barramento do sistema 121 por meio de uma interface de memória removível, tal como a interface 150.

Os drives e seus meios para armazenamento em computador associados acima descritos e ilustrados na Figura 1a propiciam o armazenamento de instruções para leitura por computador, estruturas de dados, módulos de programas e outros dados para o computador 110. Na Figura 1a, por exemplo, o drive de disco rígido 141 está ilustrado como armazenando o sistema operacional 144, programas aplicativos 145, outros módulos de programas 146 e dados de programas 147. Note-se que tais componentes podem ser os mesmos ou diferentes dos sistema operacional 134, programas aplicativos 135, outros módulos de programas 136 e dados de programas 137. O sistema operacional 144, os programas aplicativos 145, outros módulos de programas 146 e dados de programas 147 recebem aqui

referências numéricas diferentes para ilustrar que, no mínimo, eles constituem cópias diferentes. Um usuário pode alimentar comandos e informações ao computador 20 através de dispositivos de alimentação / entrada tais como um teclado 162 e um dispositivo apontador 161, comumente denominado como um "mouse", "trackball", ou "touch pad". Outros dispositivos de alimentação (não são mostrados) podem incluir um microfone, um "joystick", um "game pad", uma antena parabólica, um scanner, ou similares. Estes e outros dispositivos de alimentação estão amiúde conectados à unidade de processamento 120 através de uma interface de alimentação de usuário 160 que está acoplada ao estação base do sistema, mas que pode estar conectada através de outras estruturas de barramento ou interface, tais como uma porta paralela, uma porta para jogos, ou uma conexão do tipo barramento serial universal (USB). Um monitor 191, ou outro tipo de dispositivo de visor, está também conectado ao barramento de sistema 121 através de uma interface, tal como uma interface de vídeo 190. Além do monitor, os computadores podem também incluir outros dispositivos de saída periféricos, tais como alto falantes 197 e uma impressora 196, que podem estar conectados através de uma interface de saída de periféricos 195.

O computador 110 pode operar em um ambiente em rede usando conexões lógicas para um ou mais computadores remotos, tal como um computador remoto 180. O computador remoto 180 pode ser um computador pessoal, um servidor, um roteador, um PC em rede, um dispositivo par (peer) ou outro nodo

de rede comum, e inclui tipicamente vários ou todos os elementos acima descritos com relação ao computador 110, apesar de apenas um dispositivo de armazenamento de memória 181 ter sido ilustrado na Figura 1a. As conexões lógicas apresentadas na Figura 1a incluem uma rede de área local (LAN) 171 e uma rede de área ampla (WAN) 173, porém podem também incluir outras redes. Tais ambientes de trabalho em rede ou "networking" são comuns em escritórios, redes de computadores de empresas, intranets e na Internet.

Quando usado em um ambiente de trabalho em rede LAN, o computador 110 é conectado à LAN 171 através de um adaptador ou interface de rede 170. Quando usado em um ambiente de trabalho em rede WAN, o computador 110 inclui tipicamente um modem 172 ou outros dispositivos para estabelecer comunicações através de uma WAN 173, tal como a Internet. O modem 172, que pode ser interno ou externo, pode estar conectado ao barramento do sistema 121 através da interface de alimentação de usuário 160, ou outro mecanismo apropriado. Em um ambiente de trabalho em rede, os módulos de programas apresentados com relação ao computador 110, ou partes do mesmo, podem ser armazenados no dispositivo de armazenamento de memória remoto. Como exemplo, mas não limitação, a Figura 1a ilustra programas aplicativos remotos 185 como residindo no dispositivo de memória 181. Deve ser notado que as conexões de rede apresentadas são exemplares e que podem ser usados outros dispositivos para estabelecimento de um link de comunicação entre os computadores.

Ambiente de Programação

Uma interface de programação (ou, mais simplesmente, interface) pode ser considerada como qualquer mecanismo, processo, protocolo, para permitir que um ou mais segmentos de código se comuniquem com, ou acessem, a funcionalidade
5 provida por um ou mais outros segmentos de código. Alternativamente, uma interface de programação pode ser considerada como um ou mais mecanismos, métodos, chamadas de função, módulos, objetos, etc., de um componente de um sistema capaz de acoplamento de comunicação com um ou mais mecanismos, métodos,
10 todos, chamadas de função, módulos, etc., de outros componentes. O termo "segmento de código" na frase anterior inclui uma ou mais instruções ou linhas de código e inclui, por exemplo, módulos de código, objetos, sub-rotinas, funções e assim por diante, independentemente da terminologia
15 aplicada ou de se os segmentos de código estão compilados separadamente, ou se os segmentos de código são providos como fonte, intermediário, ou código objeto, se os segmentos de código são utilizados em um sistema ou processo runtime, ou se eles estão localizados na mesma ou em diferentes máquinas,
20 ou se a funcionalidade representada pelos segmentos de código é implementada integralmente em software, integralmente em hardware, ou em uma combinação de hardware e software.

Teoricamente, uma interface de programação pode
25 ser considerada de um modo geral tal como apresentado na Figura 1b ou 1c. A Figura 1b ilustra uma interface, a interface 1, como um conduto através do qual se comunicam um primeiro e um segundo segmentos de código. A Figura 1c ilustra

uma interface como compreendendo os objetos de interface I1 e I2 (os quais podem ou não fazer parte dos primeiro e segundo segmentos de código), que permitem aos primeiro e segundo segmentos de código de um sistema se comunicar através de um meio M. Na vista da Figura 1c, pode-se considerar que os objetos I1 e I2 mais o meio M constituem a interface. Apesar de as Figuras 1b e 1c apresentarem fluxo bidirecional e interfaces em cada ponta do fluxo, certas implementações podem apresentar fluxo de informações em apenas uma direção (ou nenhum fluxo de informações, tal como descrito mais adiante), ou podem possuir apenas um objeto de orifício em uma ponta. Como exemplo, mas não limitação, termos tais como interface de programação de aplicativo (API), ponto de entrada, método, função, sub-rotina, chamada a procedimento remoto e interface modelo de objeto componente (COM), estão englobados pela definição de interface de programação.

Aspectos de tal interface de programação podem incluir o método pelo qual o primeiro segmento de código transmite informações (em que "informações" é utilizado em seu significado mais amplo e inclui dados, comandos, requisições, etc.) para o segundo segmento de código; o método pelo qual o segundo segmento de código recebe as informações; e a estrutura, seqüência, sintaxe, organização, esquema, timing e conteúdo das informações. Com referência a isto, o próprio meio de transporte subjacente pode ser irrelevante para a operação da interface, seja o meio cabeado ou wireless, ou uma combinação de ambos, contanto que as informações sejam transportadas da forma definida pela interface.

Em certos casos, as informações podem não ser passadas em uma ou ambas as direções no sentido convencional, dado que a transferência de informações pode ocorrer através de outro mecanismo (por exemplo, informações colocadas em um buffer, 5 arquivo, etc., separadas do fluxo de informações entre os segmentos de código), ou não existente, tal como quando um segmento de código simplesmente acessa a funcionalidade efetuada por um segundo segmento de código. Qualquer um ou a totalidade de tais aspectos pode ser importante em uma dada 10 situação, dependendo por exemplo de se os segmentos de código fazem parte de um sistema em uma configuração acoplada de forma livre ou acoplada de forma restrita, devendo tal listagem portanto ser considerada como ilustrativa e não limitante.

15 Tal noção de uma interface de programação é conhecida pelos técnicos na área e fica clara através da descrição detalhada acima da invenção. No entanto, existem outras formas para implementação de uma interface de programação e, a menos de exclusão expressa, tais outras formas são consideradas como englobadas pelas reivindicações apresentadas ao 20 final do presente relatório descritivo. Tais outras formas podem aparentar ser mais sofisticadas ou complexas do que a vista simplificada das Figuras 1b e 1c, elas porém desempenham de qualquer forma uma função similar para obter o mesmo 25 resultado geral. Serão agora brevemente descritas algumas modalidades alternativas ilustrativas de uma interface de programação.

A. Fatorização

Uma comunicação de um segmento de código para outro pode ser efetuada de forma indireta desmembrando-se a comunicação em múltiplas comunicações individuais. Tal está esquematicamente representado nas Figuras 1d e 1e. Como mos-
5 trado, algumas interfaces podem ser descritas em termos de conjuntos divisíveis de funcionalidades. Dessa forma, a funcionalidade de interface das Figuras 1b e 1c pode ser fatorizada de modo a obter o mesmo resultado, da mesma forma que pode-se prover matematicamente 24, ou 2 vezes 2 vezes 3 ve-
10 zes 2. Assim sendo, e tal como ilustrado na Figura 1d, a função provida pela interface 1 pode ser subdividida de modo a converter as comunicações da interface em múltiplas interfaces, interface 1a, interface 1b, interface 1c, etc., obtendo-se o mesmo resultado. Tal como ilustrado na Figura 1e,
15 a função provida pela interface I1 pode ser subdividida em múltiplas interfaces I1a, I1b, I1c, etc., obtendo-se o mesmo resultado. De forma similar, a interface I2 do segundo segmento de código, que recebe informações provenientes do primeiro segmento de código pode ser fatorizada em múltiplas
20 interfaces I2a, I2b, I2c, etc. Quando da fatorização, o número de interfaces incluídas com o primeiro segmento de código não necessita estar de acordo com o número de interfaces incluídas com o segundo segmento de código. Em qualquer dos casos das Figuras 1d e 1e, o espírito funcional das in-
25 terfaces I1 e I2 permanece o mesmo que nas Figuras 1b e 1c respectivamente. A fatorização de interfaces pode também seguir propriedades associativas, comutativas e outras propriedades matemáticas de tal forma que a fatorização seja de

difícil reconhecimento. Como exemplo, a ordem das operações pode não ser importante e, conseqüentemente, uma função efetuada por uma interface pode ser efetuada bem antes de chegar à interface por outra peça de código ou interface, ou
 5 efetuada por um componente separado do sistema. Além disso, os técnicos na área de programação poderão apreciar que existe uma diversidade de formas para efetuar diferentes chamadas de função que obtêm o mesmo resultado.

B. Redefinição

10 Em alguns casos, pode ser possível ignorar, adicionar, ou redefinir certos aspectos (por exemplo, parâmetros) de uma interface de programação, obtendo-se porém o resultado desejado. Tal está ilustrado nas Figuras 1f e 1g. Como exemplo, presumindo-se que a interface 1 da Figura 1b inclui
 15 uma chamada de função quadrado (alimentação, precisão, saída), uma chamada que inclui três parâmetros: alimentação / entrada, precisão e saída, e que é emitida a partir do primeiro segmento de código para o segundo segmento de código. Caso o parâmetro intermediário, precisão, não seja relevante
 20 em uma dada situação, tal como mostrado na Figura 1f, ele poderia ser ignorado, ou mesmo substituído por um parâmetro sem significado (em tal situação). Pode-se também adicionar um parâmetro adicional sem significado. Em qualquer dos casos, a funcionalidade de elevação ao quadrado pode ser obti-
 25 da, contanto que a saída seja devolvida após a alimentação ser elevada ao quadrado pelo segundo segmento de código. A precisão pode muito bem ser um parâmetro significativo para uma porção a jusante ou outra porção do sistema de computa-

ção. No entanto, uma vez que seja reconhecido que a precisão não é necessária para o propósito restrito de cálculo do quadrado, ela pode ser substituída ou ignorada. Como exemplo, em lugar de passar um valor de precisão válido, um valor sem significado, tal como uma data de nascimento, poderia ser passado sem afetar adversamente o resultado. De forma similar, e tal como mostrado na Figura 1g, a interface I1 é substituída pela interface I1', redefinida para ignorar ou adicionar parâmetros à interface. A interface I2 pode ser definida de forma similar como a interface I2', redefinida de modo a ignorar parâmetros desnecessários, ou parâmetros que podem ser processados em outro local. O ponto neste caso é o de que em alguns casos uma interface de programação pode incluir aspectos, tais como parâmetros, que não são necessários para algum propósito, podendo portanto ser ignorados ou redefinidos, ou processados em outro local para outros propósitos.

C. Codificação Online

Pode também ser viável fundir parte ou a totalidade da funcionalidade de dois módulos de código separados de tal forma que a "interface" entre eles mude de forma. Como exemplo, a funcionalidade das Figuras 1b e 1c pode ser convertida à funcionalidade das Figuras 1h e 1i, respectivamente. Na Figura 1h, os primeiro e segundo segmentos de código anteriores da Figura 1b são fundidos em um módulo contendo ambos. Em tal caso, os segmentos de código podem ainda estar em comunicação entre si, porém a interface pode estar adaptada para uma forma mais adequada ao módulo único. Dessa

forma, por exemplo, declarações formais de chamada e retorno podem não mais ser necessárias, porém o processamento ou respostas de acordo com a interface 1 podem ainda estar em execução. De forma similar, apresentada na Figura 1i, parte
 5 (ou a totalidade) da interface I2 da Figura 1c pode ser colocada em linha na interface 1i para formar a interface I1". Tal como ilustrado, a interface I2 está dividida em I2a e I2b e a porção de interface I2a foi codificada em linha com a interface I1 para formar a interface I1". Como exemplo
 10 concreto, considere-se que a interface I1 da Figura 1c efetue uma chamada de função de elevação ao quadrado (alimentação, saída), que é recebida pela interface I2, que, após processamento do valor passado com a alimentação (para elevá-lo ao quadrado) pelo segundo segmento de código, devolve
 15 o resultado elevado ao quadrado com a saída. Em tal caso, o processamento efetuado pelo segundo segmento de código (elevar ao quadrado a alimentação) pode ser efetuado pelo primeiro segmento de código sem uma chamada à interface.

D. Divórcio

20 Uma comunicação proveniente de um segmento de código para outro pode ser efetuada indiretamente através da partição da comunicação em múltiplas comunicações individuais. Tal está esquematicamente representado nas Figuras 1j e 1k. Tal como mostrado na Figura 1j, uma ou mais peças de
 25 middleware (as interface de divórcio, uma vez que elas divorciam funcionalidades e/ou funções de interface da interface original) são providas para converter as comunicações na primeira interface, a interface 1, para conformá-las a

uma interface diferente, em tal caso as interfaces 2a, 2b e 2c. Tal poderia ser feito por meio de, digamos, um sistema operacional de acordo com um protocolo da interface 1, porém a seguir o sistema operacional é modificado para uso de uma interface diferente, em tal caso as interfaces 2a, 2b e 2c. A questão é a de que a interface original usada pelo segundo segmento de código é modificada de tal forma que ela não é mais compatível com a interface usada pelo primeiro segmento de código e, portanto, é usado um intermediário para tornar as interfaces antiga e nova compatíveis. De forma similar, tal como mostrado na Figura 1k, um terceiro segmento de código pode ser introduzido com a interface de divórcio DI1 para receber as comunicações provenientes da interface I1 e com a interface de divórcio DI2 para transmitir a funcionalidade de interface para, por exemplo, as interfaces I2a e I2b, re-projetadas para funcionar com a DI2, porém para prover o mesmo resultado funcional. De forma similar, a DI1 e a DI2 podem trabalhar em conjunto para traduzir a funcionalidade das interfaces I1 e I2 da Figura 1c para um novo sistema operacional, provendo porém o mesmo, ou um similar, resultado funcional.

E. Reescrita

Mais outra variante possível consiste em reescrever dinamicamente o código para substituir a funcionalidade de interface por algo diferente, mas que atinge o mesmo resultado geral. Como exemplo, pode existir um sistema no qual um segmento de código apresentado em uma linguagem intermediária (por exemplo, Microsoft IL, Java ByteCode, etc.) é

provida para um compilador JIT (Just-in-Time) ou interprete em um ambiente de execução (tal como aquele provido pelo framework .Net, o ambiente runtime Java, ou outros ambientes similares do tipo runtime). O compilador JIT pode ser escrito de forma a converter dinamicamente as comunicações provenientes do primeiro segmento de código para o segundo segmento de código, isto é, para conforma-los com uma interface diferente, tal como possa ser requerida pelo segundo segmento de código (seja o original ou um segundo segmento de código diferente). Tal está representado nas Figuras 1l e 1m. Como pode ser visto na Figura 1l, tal método é similar ao cenário de Divórcio acima descrito. Ele poderia ser efetuado, por exemplo, onde uma base instalada de aplicativos está projetada para se comunicar com um sistema operacional de acordo com um protocolo de interface, mas a seguir o sistema operacional é modificado para usar uma interface diferente. O compilador JIT poderia ser usado para conformar as comunicações "em percurso" dos aplicativos base instalados para a nova interface do sistema operacional. Tal como apresentado na Figura 1m, tal método de reescrita dinâmica das interfaces pode ser aplicado para também fatorizar dinamicamente, ou de outra forma alterar, as interfaces.

Deve também ser notado que os cenários acima descritos para obtenção do mesmo, ou similar, resultado como uma interface através de modalidades alternativas podem também ser combinados de várias formas, seja em série e/ou em paralelo, ou com outro código interveniente. Dessa forma, as modalidades alternativas acima apresentadas não são mutua-

mente excludentes e podem ser misturadas, emparelhadas e combinadas para produzir os mesmos, ou equivalentes, cenários em comparação aos cenários genéricos apresentados nas Figuras 1b e 1c. Deve também ser notado que, como na maioria das estruturas de programação, existem outras formas similares para se alcançar uma funcionalidade igual ou similar de uma interface que não podem ser aqui descritas, mas que são, de qualquer forma, representadas pelo espírito e escopo da invenção, isto é, deve ser notado que ela é pelo menos parcialmente a funcionalidade representada por, e os resultados vantajosos possibilitados por, uma interface que se subordina ao valor de uma interface.

Ambientes de Computação Portáteis

A Figura 2 ilustra um exemplo de um PC portátil / palm 201 que pode ser usado de acordo com vários aspectos da presente invenção. Quaisquer ou a totalidade dos recursos, características, subsistemas e funções no sistema da Figura 1a podem ser incluídos no computador da Figura 2. O PC portátil 201 inclui uma superfície de visor 202 de grande porte, por exemplo um visor de painel plano de digitalização, de preferência uma tela de visor de cristal líquido (LCD), na qual é apresentada uma pluralidade de janelas 203. Usando um estilete 204, o qual constitui um exemplo de um seletor, um usuário pode selecionar, realçar e/ou escrever sobre a superfície de visor de digitação 202. Os exemplos de superfícies de visor de digitação 202 adequados incluem digitalizadores de caneta eletromagnéticos, tais como os digitalizadores de caneta Mutoh ou Wacom. Outros tipos de digitaliza-

dores de caneta, por exemplo, digitalizadores ópticos, também podem ser usados. O PC portátil 201 interprete gestos efetuados pelo uso do estilete 204 para manipular dados, alimentar texto, criar desenhos e/ou executar tarefas de aplicativos de computador convencionais, tais como planilhas, programas de processamento de texto e similares.

O estilete 204 pode estar equipado com um ou mais botões ou teclas, ou outros recursos, para ampliar sua capacidade de seleção. Em uma modalidade, o estilete 204 poderia ser implementado na forma de um "lápis" ou "caneta", no qual uma extremidade constitui uma porção de escrita e a outra extremidade constitui uma ponta de "apagador" e que, quando passada pelo visor, indica que partes do visor devem ser apagadas. Adicionalmente, o próprio dedo do usuário poderia ser o estilete 204, sendo usado para selecionar ou indicar partes da imagem apresentada em um visor sensível ao toque ou sensível à proximidade. Conseqüentemente, o termo "dispositivo de alimentação / entrada de usuário", tal como é aqui utilizado, tenciona possuir uma ampla definição e engloba muitas variações de dispositivos de alimentação bem conhecidos, tais como o estilete 204. A região 205 apresenta uma região de feedback ou região de contato que permite ao usuário determinar onde o estilete 204 entrou em contato com a superfície do visor 202. Dessa forma, um estilete 204 pode ser usado como um seletor eficaz em certos aspectos da presente invenção.

Em várias modalidades, o sistema provê uma plataforma de tinta na forma de um conjunto de serviços COM (com-

ponent object model) que um aplicativo pode usar para capturar, manipular e armazenar tinta. Um serviço permite a um aplicativo ler e gravar tinta usando as representações descritas de tinta. A plataforma de tinta pode também incluir
5 uma linguagem mark-up incluindo uma linguagem como a XML (extensible markup language). Além disso, o sistema pode usar DCOM como outra implementação. Outras implementações adicionais podem ser usadas, incluindo o modelo de programação Win32 e o modelo de programação .Net da Microsoft Corporation.
10 ration.

A Figura 3 apresenta componentes funcionais do dispositivo de computação portátil ("H/PC") 320. Tal como mostrado, o H/PC 320 inclui um processador 360, uma memória 362, um visor 328 e um teclado 332. A memória 362 pode incluir tanto memória volátil (por exemplo, RAM) como memória
15 não volátil (por exemplo, ROM, placas PCMCIA, etc.). Um sistema operacional 364 pode estar residente na memória 362 e pode ser executado no processador 360. Em uma modalidade, o H/PC 320 inclui um sistema operacional, tal como o sistema operacional Windows® CE da Microsoft Corporation, ou outro
20 sistema operacional.

Um ou mais programas aplicativos 366 podem ser carregados para a memória 362 e rodar no sistema operacional 364. Os exemplos de aplicativos incluem programas de e-mail,
25 programas de programação, programas de PIM (gerenciamento de informações pessoais), programas de processamento de texto, programas de planilhas, programas de navegação na Internet, jogos e outros aplicativos conhecidos. O H/PC 320 pode tam-

bém incluir um gerenciador de notificações 368 carregado na memória 362, o qual pode também ser executado no processador 360. O gerenciador de notificações 368 pode lidar com requisições de notificação provenientes dos aplicativos 366.

5 O H/PC 320 inclui um suprimento de energia 370, o qual pode ser implementado na forma de uma ou mais baterias, ou alguma outra fonte de energia, tal como um capacitor ou célula de combustível, ou similares. O suprimento de energia 370 pode incluir também uma fonte de energia externa que se
10 sobre põe ou recarrega as baterias embutidas, tal como um adaptador de CA ou um suporte de armazenamento energizado.

 Como mostrado, o H/PC 320 inclui três tipos de mecanismos de notificação externos: um LED 340, um dispositivo de vibração 372 e um gerador de áudio 374. Tais dispositivos
15 podem estar diretamente acoplados ao suprimento de energia 370 de forma a que, quando ativados, eles possam permanecer ligados por um tempo ditado pelo mecanismo de notificação, mesmo que o processador do H/PC 320 360 e outros componentes possam ser desligados para reduzir o consumo de energia. Em
20 uma modalidade, o LED 340 pode permanecer ligado até que o usuário aja. As versões atuais do dispositivo de vibração 372 e do gerador de áudio 374 usam muita energia para as baterias de H/PC atuais e portanto eles podem ser configurados para serem desligados quando o restante do sistema for des-
25 ligado ou após um intervalo finito após a ativação.

Aplicativos

Fazendo referência à Figura 4, é ali apresentada uma modalidade de um computador portátil 401. O computador

portátil, que pode ser configurado tal como acima descrito, inclui um estojo 402 no qual é provido um visor 405. como mostrado, é provido um teclado virtual 406, incluindo uma disposição 408 de teclas virtuais 410. deve ser notado que
5 outras disposições 408 também são possíveis.

Caso desejado, a disposição 408 do teclado virtual 406 poderia ser controlada por um aplicativo através de uma API, tal como acima descrito. Em uma modalidade, um parâmetro poderia ser usado para ajustar a disposição 408 das teclas virtuais 410. O parâmetro poderia incluir um valor padrão que poderia ser modificado de forma a ajustar a disposição 408. Dessa forma, diferentes aplicativos poderiam ajustar a disposição 408 das teclas virtuais 410 conforme desejado. Em uma modalidade, o teclado virtual 406 poderia incluir um conjunto de disposições 408 que poderiam ser selecionadas pelo aplicativo, cada disposição posicionando as teclas virtuais 410 em diferentes locais predeterminados. Em outra modalidade, a disposição 408 poderia ser completamente customizável de forma a que os aplicativos pudessem posicio-
10
15
20 nar as teclas virtuais 410 individuais conforme desejado.

Para permitir aos indivíduos alimentar dados ao dispositivo de computação, cada tecla virtuais 410 pode ser associada a um símbolo, de forma a que o caractere "a", tal como usado na língua inglesa. Em uma modalidade, apresentada
25 na Figura 5, a tecla virtual 410 pode incluir um símbolo 505 posicionado aproximadamente próximo ao centro da tecla virtual 410. Uma vez selecionada a tecla virtual 410, o que pode ser feito pela compressão de um seletor sobre uma super-

fície de um visor sensível ao toque, o visor associado à tecla virtual pode ser ajustado. Como será notado, caso a tecla virtual seja grande o suficiente, mais de um símbolo poderá ser mostrado sobre a tecla virtual e a seleção da tecla virtual permite ao usuário então selecionar quais dos símbolos sendo apresentados se deseja escolher.

Em uma modalidade, apresentada na Figura 6a, a seleção da tecla virtual 410 leva à apresentação de símbolos adicionais 505. Deve ser notado que a seleção pode consistir da compressão de um seletor sobre o visor sensível ao toque na posição alinhada com a tecla virtual desejada. Em uma modalidade, podem ser apresentados os símbolos 505a a 505h. O símbolo 505 também pode continuar sendo apresentado. Os símbolos 505a a 505h representam o símbolo 505 em conjunto com diferentes teclas modificadoras. Dessa forma, o símbolo 505a está associado ao símbolo 505 e a uma primeira tecla modificadora.

A Figura 6b ilustra uma modalidade alternativa. O símbolo 505 consiste de uma representação de um "5". Ao prover uma alimentação de direção, outro número, tal como 1 a 4 ou 6 a 9, pode ser selecionado. Como será notado, tal modalidade permitiria que uma única tecla virtual propicie todos os algarismos exceto zero. Um possível método para prover o zero seria requerer que o usuário efetue um arraste de direção primeiramente em uma direção e a seguir na direção oposta. Dessa forma, como será notado, movimentos compostos poderiam estar associados a teclas modificadoras adicionais caso desejado. Alternativamente, um ou mais dos símbolos as-

sociados aos vários modificadores poderiam se modificar após um certo período de retardo. Dessa forma, em uma modalidade, caso o toque fosse mantido sem prover uma alimentação direcional, o símbolo 505 poderia se modificar de um "5" para um "0" após um certo período de tempo predeterminado.

Em uma modalidade, um usuário pode selecionar uma tecla virtual 505 e os vários símbolos podem se tornar imediatamente visíveis. Em uma modalidade alternativa, os vários símbolos podem aparecer somente se o usuário efetua um contato prolongado com a tecla virtual 505. como poderá ser notado, o período de contato com a tecla virtual 505 necessário para levar ao aparecimento dos símbolos adicionais pode ser ajustado através de uma API, tal como acima descrito.

Em uma modalidade, certos símbolos representando as modificações mais comuns do símbolo 505 podem aparecer em primeiro lugar. Dessa forma, pode ser provida uma resposta escalonada. Além disso, certas posições podem ser populadas com um certo símbolo antes de outras posições. Como será notado, são possíveis muitas variações e, em uma modalidade, tais variações podem ser ajustadas através de uma API. Dessa forma, por exemplo, mas sem limitação, o símbolo 505a poderia aparecer em primeiro lugar, a seguir o símbolo 505e e assim por diante. Como poderá ser notado, caso o símbolo 505 não possuir modificações possíveis em número suficiente, o número de símbolos 505a a 505e que podem ser apresentados pode ser adequadamente reduzido. Ademais, certos símbolos podem ser designados para eventos definidos pelo usuário e certas ações customizáveis pelo aplicativo, pelo usuário, ou

por ambos.

Em uma modalidade, o primeiro símbolo apresentado sobre a tecla virtual e os símbolos associados ao símbolo inicialmente apresentado podem ser determinados em resposta a um aplicativo. Dessa forma, certos símbolos apresentados inicialmente poderiam ter um maior número de símbolos associados, enquanto outros símbolos poderia ter um número menor de símbolos associados. Em uma modalidade, um primeiro símbolo poderia ter um único segundo símbolo representativo de um primeiro símbolo sujeito a um modificador a ele associado. Em tal modalidade, um contato por um tempo maior do que o período programado, o qual pode ser qualquer período de tempo maior do que zero, poderia levar ao aparecimento do segundo símbolo. Para aumentar o número de opções associadas a tal símbolo, um usuário poderia associar símbolos adicionais ao primeiro símbolo. Em uma modalidade, o usuário poderia selecionar a posição dos símbolos adicionais, seja pelo deslocamento de símbolos existentes, ou pela escolha de uma posição que não estava sendo associada no momento com o primeiro símbolo e uma alimentação direcional. Em uma modalidade alternativa, a posição seria selecionada automaticamente pelo aplicativo de acordo com um algoritmo. Um de tais algoritmos poderia consistir de primeiramente popular as posições representadas como os símbolos 505a, 505c, 505e e 505g na Figura 6, em um sentido horário. Como poderá ser notado, as variações de tais algoritmos são numerosas e poderiam incluir a análise estatística das ações do usuário e poderiam também incluir alimentação proveniente de estudos de utili-

zação.

Para selecionar um símbolo, o usuário poderia mover o seletor, o qual poderia ser um estilete, um dedo, ou algum outro dispositivo adequado para prover alimentação direcional, na direção do símbolo. Como mostrado na Figura 6, 5 oito direções de movimento após a seleção inicial poderiam prover oito símbolos diferentes relacionados ao primeiro símbolo 505. No entanto, caso o usuário não provesse uma alimentação direcional (ou caso a alimentação direcional estivesse abaixo de um limite predeterminado), então o primeiro 10 símbolo 505 seria escolhido. Deve ser notado que apesar de serem ilustrados oito símbolos diferentes, poderiam ser apresentados símbolos adicionais. No entanto, é provável que um indivíduo teria dificuldade para prover consistentemente 15 um movimento preciso em mais do que oito direções; portanto, algumas etapas adicionais poderiam ser úteis caso mais de oito modificadores devam ser associados a uma única tecla virtual.

Em uma modalidade, os símbolos adicionais poderiam 20 ser apresentados ao longo de um perímetro externamente ao limite da tecla virtual 505, podendo estar espaçados. Para auxiliar o usuário com a seleção do símbolo desejado, o movimento na direção do símbolo causaria uma mudança gráfica, tal como uma mudança de cor ou gradiente do símbolo potencialmente selecionado. O usuário poderia então mover em uma 25 direção diferente para levar um símbolo diferente ao longo do perímetro a ser realçado. Como poderá ser notado, o número de símbolos associados, o tamanho do perímetro, o tipo de

feedback e a sensibilidade constituem apenas alguns dos parâmetros que poderiam ser variados de tecla virtual a tecla virtual e de aplicativo a aplicativo, tais parâmetros podendo ser controlados pelos aplicativos através de uma API.

5 A Figura 7 ilustra um método para seleção de um símbolo de acordo com um aspecto da presente invenção. Tal como ilustrado, na etapa 710 o usuário seleciona um primeiro símbolo. Em uma modalidade, a seleção envolve a pressão sobre a superfície do visor sensível ao toque com um seletor, 10 estando o toque alinhado com uma tecla virtual desejada que apresenta um símbolo apropriado.

 A seguir, na etapa 715, um conjunto de símbolos associados ao primeiro símbolo se torna visível. Em uma modalidade, o conjunto contém um símbolo. Em uma modalidade 15 alternativa, o conjunto contém oito símbolos. Como poderá ser notado, as variações no número de símbolos providos no conjunto podem variar como desejado. Em uma modalidade, os símbolos podem ser tornar gradualmente mais visíveis, ou podem aparecer de alguma outra forma, tal como deslizando a 20 partir da direita, deslizando a partir do centro, ou simplesmente aparecendo repentinamente. Como poderá ser notado, tal controle gráfico sobre como os símbolos são populados e tornados visíveis pode ser variado e pode ser controlado através de uma API. Um fading ou esmaecimento gradual pode 25 ser útil, pois ele permitiria ao usuário receber um lembrete quanto a quais símbolos estão associados ao símbolo inicialmente selecionado, sem causar uma confusão indesejável na tela.

Na etapa 720, o usuário provê a alimentação direcional para selecionar um símbolo a partir do conjunto de símbolos associados ao símbolo inicialmente selecionado. A alimentação direcional pode ser provida pelo movimento do seletor na direção do símbolo desejado. Em uma modalidade, a alimentação direcional pode ser provida deslizando o seletor ao longo da superfície de um visor sensível ao toque em uma direção alinhada com um símbolo no conjunto de símbolos. Tal deslizamento pode ser designado como arraste direcional do seletor. Como foi acima mencionado, a alimentação direcional pode levar o símbolo que está associado à alimentação direcional específica a ficar realçado. Em uma modalidade, o símbolo realçado poderia mudar de tamanho, poderia mudar de cor ou contraste, ou poderia prover algum outro tipo de feedback gráfico, incluindo animação. Em uma modalidade, múltiplos símbolos podem ser providos no conjunto. Em uma modalidade com múltiplos símbolos providos no conjunto, a alimentação direcional inicial provida pelo usuário irá escolher um dos símbolos no conjunto. Em uma modalidade alternativa com múltiplos símbolos no conjunto, o usuário pode movimentar livremente em diferentes direções de forma a realçar e escolher diferentes símbolos que não o símbolo que teria sido escolhido com base na alimentação de direção inicialmente provida.

25 A seguir, na etapa 725, após o símbolo desejado ser realçado, o usuário finaliza a seleção e leva o símbolo realçado a ser transmitido (por exemplo, entra ou alimenta o símbolo desejado no aplicativo). Em uma modalidade, a fina-

lização da seleção do símbolo poderia ser efetuada pela remoção do seletor da superfície do visor. Como poderá ser notado, outros métodos, tal como um acionamento de um dispositivo de alimentação, poderiam ser também usados para finalizar a seleção do símbolo realçado. Em uma modalidade, um acionamento de um botão ou tecla poderia causar a seleção finalizada e a transmissão do símbolo correntemente realçado. Tal método poderia ser útil para um dispositivo portátil em uma situação em que o usuário estiver passando por uma quantidade significativa de vibração ou que esteja incapaz de manter o seletor firme por alguma outra razão, tal como um estado doentio.

A Figura 8 ilustra outra modalidade de um método para seleção de um símbolo. Como pode ser notado, uma diferença entre os métodos apresentados na Figura 7 e na Figura 8 consiste da omissão da etapa de os símbolos associados ao primeiro símbolo se tornarem visíveis. Dessa forma, em uma modalidade, os símbolos associados ao primeiro símbolo não ficarão visíveis, mas podem ainda ser selecionados através da seleção e alimentação direcional apropriadas providas pelo usuário. Em uma modalidade alternativa, a seleção de um símbolo não levará os símbolos associados a se tornarem visíveis a menos que ocorra um período prolongado de retardo (por exemplo, o usuário continua a pressionar o visor sensível ao toque por um período de tempo estendido). Em outra modalidade alternativa, os símbolos associados não se tornarão visíveis a menos que o usuário propicie uma alimentação secundária. Como pode ser notado, o fato de os símbolos as-

sociados se tornarem visíveis pode ser controlado por um aplicativo e poderia ser ajustado de forma a ser controlado pelo usuário através da comutação liga / desliga do recurso em um aplicativo.

5 Uma vantagem do método apresentado na Figura 8 é a de que o visor não fica congestionado com símbolos adicionais e, dependendo do tipo de visor, pode ser necessária menos energia, uma vez que não ocorre mudança do visor. Além disso, tal método é menos complexo do ponto de vista gráfico. No entanto, como pode ser notado, tal método requer que o usuário conheça quais símbolos estão associados ao primeiro símbolo selecionado, juntamente com a direção de movimento necessária para selecionar o símbolo associado desejado. Para certos idiomas, em que existem vários símbolos secundários associados ao primeiro símbolo, uma configuração de não
10 apresentação dos símbolos secundários pode ser um pouco mais difícil. Portanto, tal configuração pode ser melhor adequada para usuários capacitados ou do ponto de vista da energia.

 Como poderá ser notado pela descrição acima, quando existem múltiplos símbolos associados a cada símbolo, se
20 torna mais difícil prover informações suficientes para o usuário sem encher indesejável o visor. Uma solução potencial que pode ser usada de acordo com vários aspectos da presente invenção consiste em associar uma tecla modificadora específica a uma direção particular. Como exemplo, se as direções
25 possíveis associadas a um valor em graus variando entre 0 e 359 graus (pois 360 graus seria equivalente a zero), o movimento ao longo de um vetor em 0 grau poderia estar associado

à tecla "shift". Deve ser notado que a direção de 0 grau depende de um ponto de referência e portanto pode ser ajustada conforme desejado. Fazendo novamente referência à Figura 6, 0 grau poderia estar alinhado com a localização do símbolo 505a. Dessa forma, em uma modalidade, o posicionar um selector sobre um símbolo e a seguir movimenta-lo na direção de zero grau (por exemplo, para cima) seria equivalente a pressionar uma tecla virtual de "selecionar" e a seguir pressionar o símbolo. Como pode ser notado, o pressionar um símbolo e a seguir prover um arraste direcional é mais eficiente e pode ser efetuado mais rapidamente do que primeiramente selecionar uma tecla virtual de shift e a seguir selecionar a tecla desejada. Apesar de serem possíveis inúmeras variações, pode ser mais fácil requerer que o usuário aponte para regiões separadas por pelo menos 45 graus, de forma a que haja separação suficiente entre as seleções.

Em uma modalidade, "para cima" (ou 0 grau) pode estar associado à tecla modificadora "shift", "para baixo" (ou 180 graus) com a tecla modificadora "control", para a direita (ou 270 graus) com a tecla modificadora "alt", e para a esquerda (ou 90 graus) com uma tecla modificadora de "função". Como pode ser notado, podem ser providas variações de tais e a direção associada a um modificador específico pode ser customizada pelo usuário. Além disso, certas modificações do símbolo, tal como um sobrescrito ou subscrito, poderiam também ser associadas a um arraste direcional.

Deve ser notado que certas teclas virtuais podem não possuir variações suficientes possíveis para popular to-

das as direções associadas a um símbolo apresentado na tecla virtual. Como exemplo, pode não existir nada associado à modificadora "alt" e com o símbolo "m" em inglês. Apesar de tal poder ser específico por aplicativo e ajustável através de uma API, tal como acima descrito, em uma modalidade, um primeiro símbolo pode normalmente possuir somente um modificador a ele associado, tal como, porém não limitado a, um modificador shift. Como exemplo, em uma modalidade, um teclado destinado ao uso para datilografia em inglês poderia associar uma direção para cima com uma versão maiúscula das 26 letras do alfabeto, e qualquer outro movimento poderia ser ignorado. Dessa forma, um "a" maiúsculo poderia ser alimentado através da seleção da tecla virtual "a" e a seguir pelo provimento de um arraste direcional na direção ou vetor de zero grau. Em uma modalidade alternativa, as letras maiúsculas poderiam ser o tipo normal de símbolo alimentado e as letras minúsculas poderiam ser alimentadas através da seleção de uma letra maiúscula seguida por arraste direcional ao longo de um vetor de 180 graus.

Independentemente do número de símbolos associados a um símbolo apresentado em uma tecla virtual, pode ser útil permitir a um usuário associar símbolos adicionais ao primeiro símbolo, tal como foi acima mencionado. Dessa forma, em uma modalidade, o usuário poderia designar um símbolo para uma tecla virtual e uma direção de movimento. Em tal modalidade, o símbolo "\$" poderia ser associado com a seleção da tecla virtual que apresenta o símbolo "s" e um movimento para baixo. O usuário poderia mesmo importar símbolos adi-

cionais caso desejado. Dessa forma, como pode ser notado, a criatividade do usuário constitui o fator limitante principal para o número de variações possíveis.

Em uma modalidade, uma série de ações também poderia ser designada para uma seleção e direção de movimento particulares. Como exemplo, uma macro para tornar negrito símbolos futuros poderia ser associada à seleção do símbolo "b" ("bold") e um movimento para a direita. Para "desligar" o recurso negrito, a mesma operação poderia ser usada, ou o usuário poderia selecionar "b" e um movimento para a esquerda. Dessa forma, como pode ser notado, existem inúmeras variações, algumas das quais poderiam ser customizadas de acordo com o aplicativo. Deve ser notado que macros mais complexas também poderiam ser associadas à seleção de uma tecla virtual e uma alimentação direcional.

Deve ser notado que certos idiomas limitam a alimentação com base no que foi previamente alimentado. Além disso, certos idiomas formam combinações de letras. Como exemplo, a seleção de um "n" no Kana japonês poderia prover ao usuário a escolha de na, ni, nu e no. Dessa forma, após a entrada ou seleção de um símbolo inicial, o usuário só seria capaz de alimentar certos símbolos que correspondem ao contexto do símbolo previamente alimentado ou selecionado. Dessa forma, em uma modalidade, os símbolos associados a um símbolo particular podem se ajustar de acordo com regras gramaticais de escrita e símbolos alimentados previamente.

A Figura 9 ilustra uma modalidade de tal método. Primeiramente, na etapa 910, o usuário alimenta um primeiro

símbolo. A alimentação de símbolos pode ser provida pelo uso de um seletor, o qual pode ser um estilete, para tocar e liberar uma tecla virtual que é provida em um teclado virtual em um visor de tela de toque. A seguir, na etapa 915, o usuário seleciona uma tecla virtual que apresenta um segundo símbolo, estando o segundo símbolo relacionado ou desejado ser usado com o primeiro símbolo. Em uma modalidade, a seleção é efetuada pela manutenção da pressão na tecla virtual com o seletor. Como o segundo símbolo pode ser associado a um certo número de variações possíveis, na etapa 920 um conjunto de símbolos relacionados a pelo menos algumas das variações possíveis no segundo símbolo é apresentado em torno do segundo símbolo. Em uma modalidade, todos os símbolos que estão associados ao primeiro símbolo e ao segundo símbolo são apresentados em torno do segundo símbolo de uma maneira desejada.

Na etapa 925, o usuário provê alimentação direcional. A medida que os símbolos são providos em torno do segundo símbolo, o movimento para longe do ponto inicial de seleção irá ser tipicamente em direção a um dos símbolos no conjunto. Em uma modalidade, a alimentação direcional pode ser provida por arraste do seletor através do visor sensível ao toque em direção a um dos símbolos no conjunto após pressionar inicialmente a tecla virtual que apresentava o segundo símbolo. Na etapa 930, o usuário susta o processo de seleção e alimentação de direção e a seleção do símbolo escolhido dentre o conjunto de símbolos é finalizada (por exemplo, alimentada). Em um aplicativo de processamento de tex-

to, a alimentação poderia resultar em o símbolo escolhido ser apresentado ao lado do símbolo alimentado anteriormente. Em uma modalidade, o usuário pode finalizar a seleção levantando o seletor da tela sensível ao toque após arrastar o
5 seletor na direção do símbolo desejado. A finalização da seleção pode também ser designada como alimentação ou entrada ou transmissão do símbolo escolhido. Em uma modalidade, como foi acima descrito, o visor pode realçar o símbolo correntemente escolhido de forma a prover feedback para o usuário
10 com referência a qual símbolo será alimentado caso o usuário eleve o seletor do visor neste momento.

Deve ser notado que apesar de os aplicativos podem controlar os recursos descritos através de uma API, um sistema operacional pode também controlar os recursos caso
15 desejado. No entanto, níveis potencialmente mais elevados de flexibilidade são providos caso os recursos possam ser controlados por um aplicativo através de uma API.

A presente invenção foi descrita em termos de modalidades preferidas e exemplares da mesma. Apesar de a matéria objeto ter sido descrita em uma linguagem específica para os recursos estruturais e/ou atos metodológicos, deve ficar claro que a matéria objeto definida nas reivindicações
20 anexas não fica necessariamente limitada aos recursos ou atos específicos acima descritos. Ao contrário, os recursos ou atos específicos acima descritos são descritos como formas exemplares de implementação das reivindicações. Várias outras modalidades, modificações e variações inseridas no escopo e espírito das reivindicações anexas irão ocorrer aos
25

técnicos na área após uma leitura da presente descrição.

REIVINDICAÇÕES

1.Método para alimentação de dados para um aplicativo rodando em um computador com um teclado virtual em uma tela sensível ao toque, **CARACTERIZADO** pelo fato de que compreende:

(a)receber um toque seletor e um arraste direcional seletor na tela sensível ao toque;

(b)determinar se o toque seletor está associado a uma primeira tecla virtual apresentando um primeiro símbolo e se o arraste direcional seletor está associado a uma primeira tecla modificadora;

(c)selecionar um segundo símbolo, o segundo símbolo estando associado ao primeiro símbolo e à primeira tecla modificadora; e

(d)transmitir o segundo símbolo para o aplicativo.

2.Método, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que quatro teclas modificadoras estão associadas ao primeiro símbolo e em que cada modificadora está associada a uma direção diferente selecionada a partir de um conjunto de direções consistindo de 0 grau, 90 graus, 180 graus e 270 graus, estando o arraste direcional seletor direcionado ao longo da direção de 0 grau.

3.Método, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que o determinar a tecla virtual associada ao toque seletor em (b) compreende:

i.determinar se o seletor está tocando o visor sensível ao toque em uma primeira posição; e

ii.determinar se a primeira tecla virtual está as-

sociada à primeira posição no visor sensível ao toque.

4.Método, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que o receber o arraste seletor direcional em (a) compreende:

5 i.determinar a primeira posição em que o seletor tocou a tela sensível ao toque;

ii. determinar uma posição final em que o seletor para de tocar a tela sensível ao toque; e

10 iii. determinar um vetor associado à mudança de posição entre a primeira posição e a posição final.

5.Método, de acordo com a reivindicação 4, **CARACTERIZADO** pelo fato de que o determinar em (b) que a primeira tecla modificadora está associada ao arraste direcional seletor compreende:

15 i. determinar um conjunto de direções associado à primeira tecla virtual;

ii. determinar uma primeira direção a partir do conjunto de direções com o qual o vetor do arraste seletor direcional está mais proximamente alinhado; e

20 iii. determinar se a primeira tecla modificadora está associada à primeira direção.

6.Método, de acordo com a reivindicação 5, **CARACTERIZADO** pelo fato de que o receber compreende também:

25 iv.realçar o segundo símbolo associado à primeira tecla modificadora.

7.Método, de acordo com a reivindicação 1, **CARACTERIZADO** pelo fato de que compreende também:

(e)após receber o toque, porém antes que o arraste

direcional seja recebido, posicionar visivelmente o segundo símbolo ao longo de um vetor associado à primeira tecla modificadora.

8.Meio para leitura por computador, **CARACTERIZADO**
5 pelo fato de que inclui instruções para leitura por computador para efetuar as etapas de:

(a)receber uma primeira seleção associada a uma primeira tecla virtual possuindo um primeiro símbolo;

(b)determinar um conjunto de símbolos associados à
10 primeira tecla virtual;

(c)receber um arraste direcional associado à primeira seleção; e

(d)comunicar um segundo símbolo proveniente do conjunto de símbolos, estando o segundo símbolo associado à
15 alimentação direcional e ao primeiro símbolo.

9.Meio para leitura por computador, de acordo com a reivindicação 8, **CARACTERIZADO** pelo fato de que compreende também instruções para leitura por computador para efetuar as etapas de:

20 (e)receber uma notificação de que um seletor tocou uma tecla sensível ao toque por um período predeterminado; e

(f)apresentar o conjunto de símbolos em torno do primeiro símbolo.

10.Meio para leitura por computador, de acordo com
25 a reivindicação 9, **CARACTERIZADO** pelo fato de que compreende também instruções para leitura por computador para efetuar a etapa de:

(g)após receber o arraste de direção, realçar o

segundo símbolo associado à alimentação direcional.

11.Meio para leitura por computador, de acordo com a reivindicação 9, **CARACTERIZADO** pelo fato de que o apresentar em (f) compreende:

5 i. determinar um perímetro para posicionar o conjunto de símbolos em torno do primeiro símbolo; e

ii. prover instruções para posicionar cada símbolo no conjunto de símbolos no perímetro em uma posição apropriada.

10 12.Meio para leitura por computador, de acordo com a reivindicação 11, **CARACTERIZADO** pelo fato de que o prover as instruções em (ii) compreende:

(1)determinar o conjunto de símbolos associados ao primeiro símbolo;

15 (2)determinar as direções associadas à pelo menos uma respectiva tecla modificadora associada ao conjunto de símbolos; e

(3)prover instruções para posicionar cada um dos símbolos no conjunto de símbolos correspondente à direção da
20 pelo menos uma tecla modificadora respectiva.

13.Meio para leitura por computador, de acordo com a reivindicação 8, **CARACTERIZADO** pelo fato de que compreende também instruções para leitura por computador para efetuar a etapa de:

25 (e)prover instruções para apresentar o conjunto de símbolos em torno do primeiro símbolo, cada um dos símbolos no conjunto estando posicionado em uma direção correspondente a uma tecla modificadora associada ao respectivo símbolo.

14. Meio para leitura por computador, de acordo com a reivindicação 8, **CARACTERIZADO** pelo fato de que o receber em (c) compreende:

i. determinar se um seletor acionou uma tela sensível ao toque em uma primeira posição; e

ii. determinar se o seletor deslizou através da tela sensível ao toque para uma segunda posição.

15. Método para alimentação de dados usando um conjunto de teclas virtual no visor de um sistema de computador possuindo uma interface gráfica de usuário que inclui um visor e um dispositivo de seleção da interface de usuário, **CARACTERIZADO** pelo fato de que inclui as etapas de:

(a) receber uma seleção de uma primeira tecla virtual proveniente do conjunto de teclas virtuais, a primeira tecla virtual incluindo um primeiro símbolo;

(b) receber uma alimentação de direção provida pelo dispositivo de seleção da interface de usuário; e

(c) inserir um segundo símbolo associado à alimentação direcional e ao primeiro símbolo.

16. Método, de acordo com a reivindicação 15, **CARACTERIZADO** pelo fato de que compreende a etapa de:

(d) apresentar o segundo símbolo associado ao primeiro símbolo antes de receber a alimentação direcional.

17. Método, de acordo com a reivindicação 16, **CARACTERIZADO** pelo fato de que o apresentar na etapa (d) compreende:

i. aguardar por um período apreciável de tempo após receber a seleção; e

ii. gerar uma representação gráfica do segundo

símbolo associado ao primeiro símbolo no visor na posição apropriada.

18.Método, de acordo com a reivindicação 17,
CARACTERIZADO pelo fato de que a etapa (ii) compreende:

5 (1)após receber a seleção da primeira tecla virtual, determinar que o segundo símbolo está associado à tecla virtual;

 (2)determinar a alimentação direcional associada ao segundo símbolo; e

10 (3)apresentar a representação gráfica do segundo símbolo no visor, estando o segundo símbolo posicionado de acordo com a alimentação de direção associada ao segundo símbolo.

19.Método, de acordo com a reivindicação 15,
15 **CARACTERIZADO** pelo fato de que o receber em (b) compreende:

 i.notar uma posição inicial de um seletor em uma tela sensível ao toque;

 ii.notar uma posição final do seletor na tela sensível ao toque; e

20 iii.determinar uma direção em que o seletor se movimentou com relação a uma referência predeterminada.

20.Método, de acordo com a reivindicação 15,
CARACTERIZADO pelo fato de que compreende também a etapa de:

 (d)prover o conjunto de teclas virtuais em um visor em uma disposição customizável, em que um aplicativo pode selecionar a partir de uma dentre pelo menos duas disposições diferentes.

25

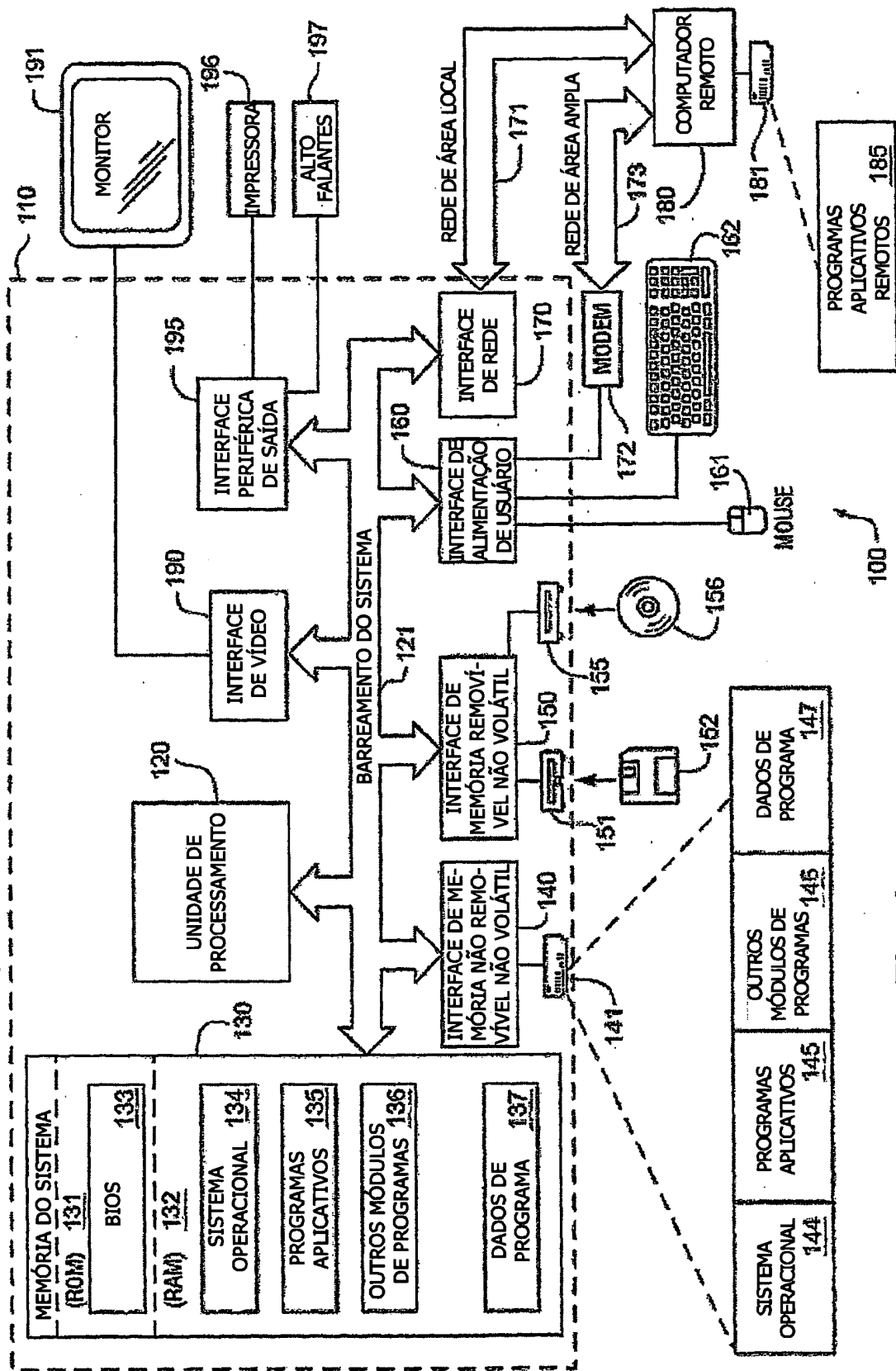


Fig. 1a

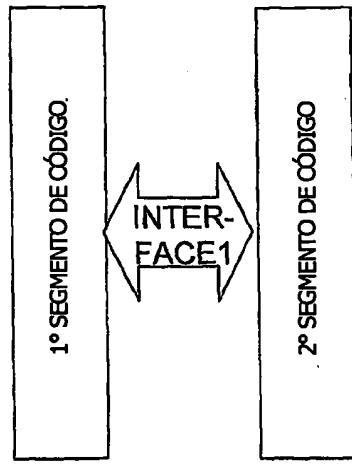


Fig. 1b

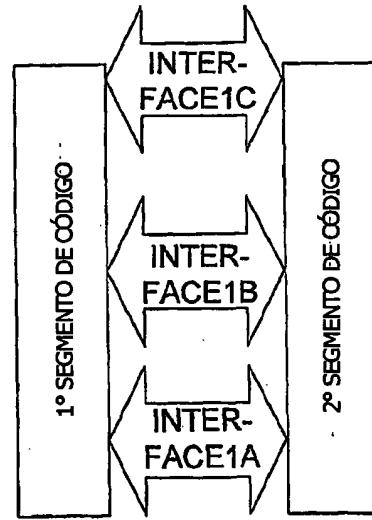


Fig. 1d

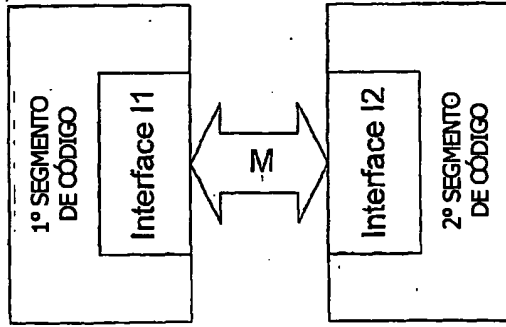


Fig. 1c

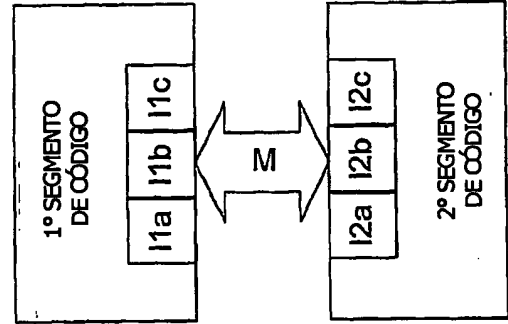
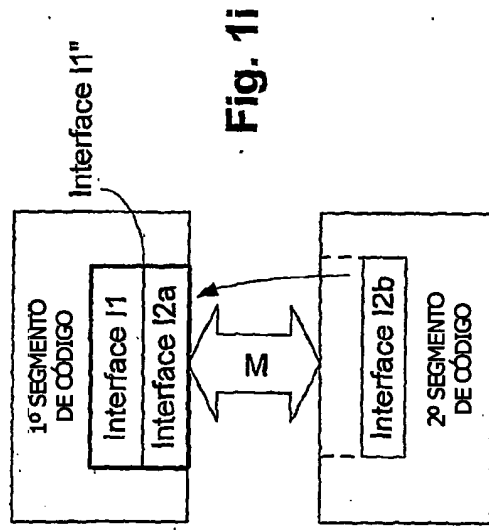
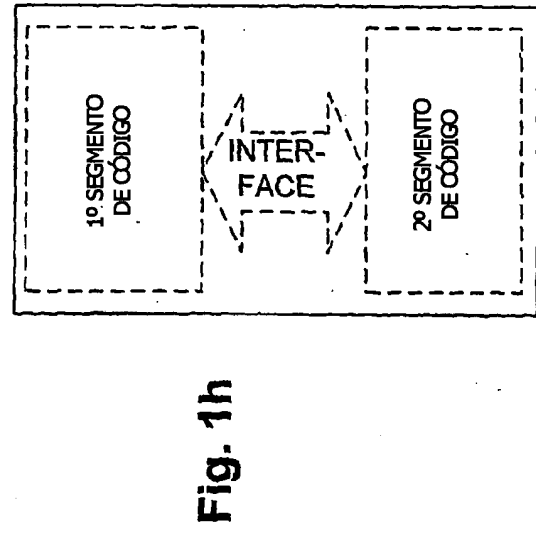
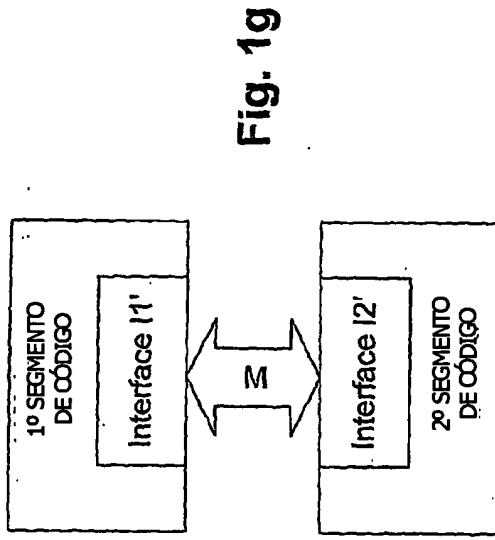
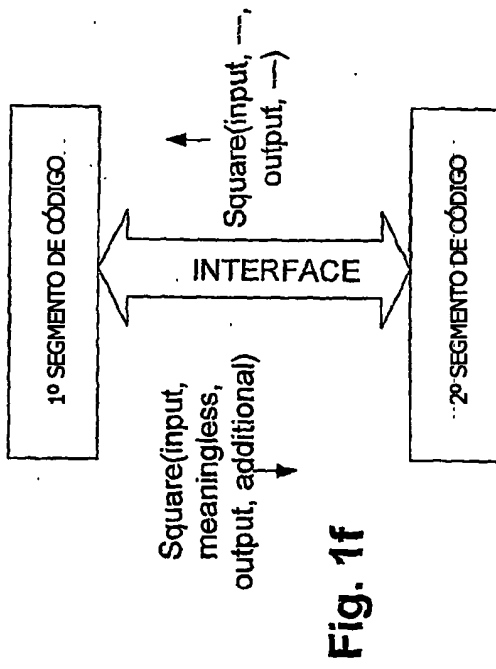
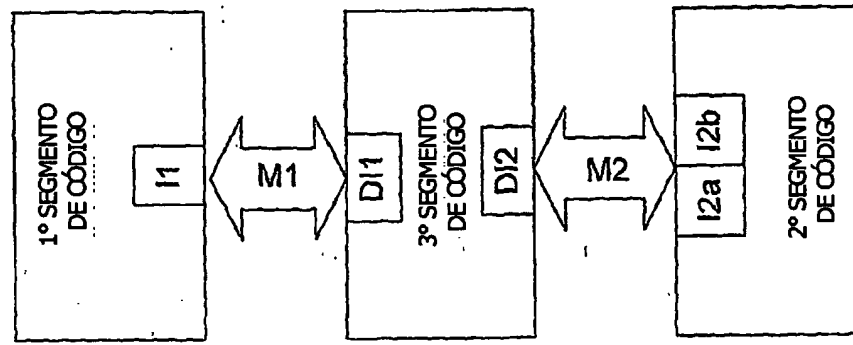
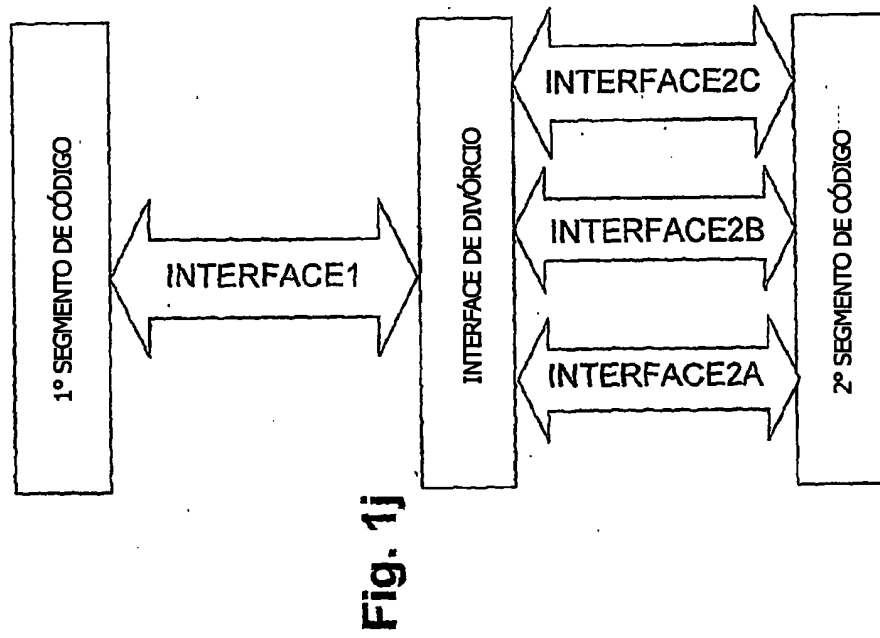
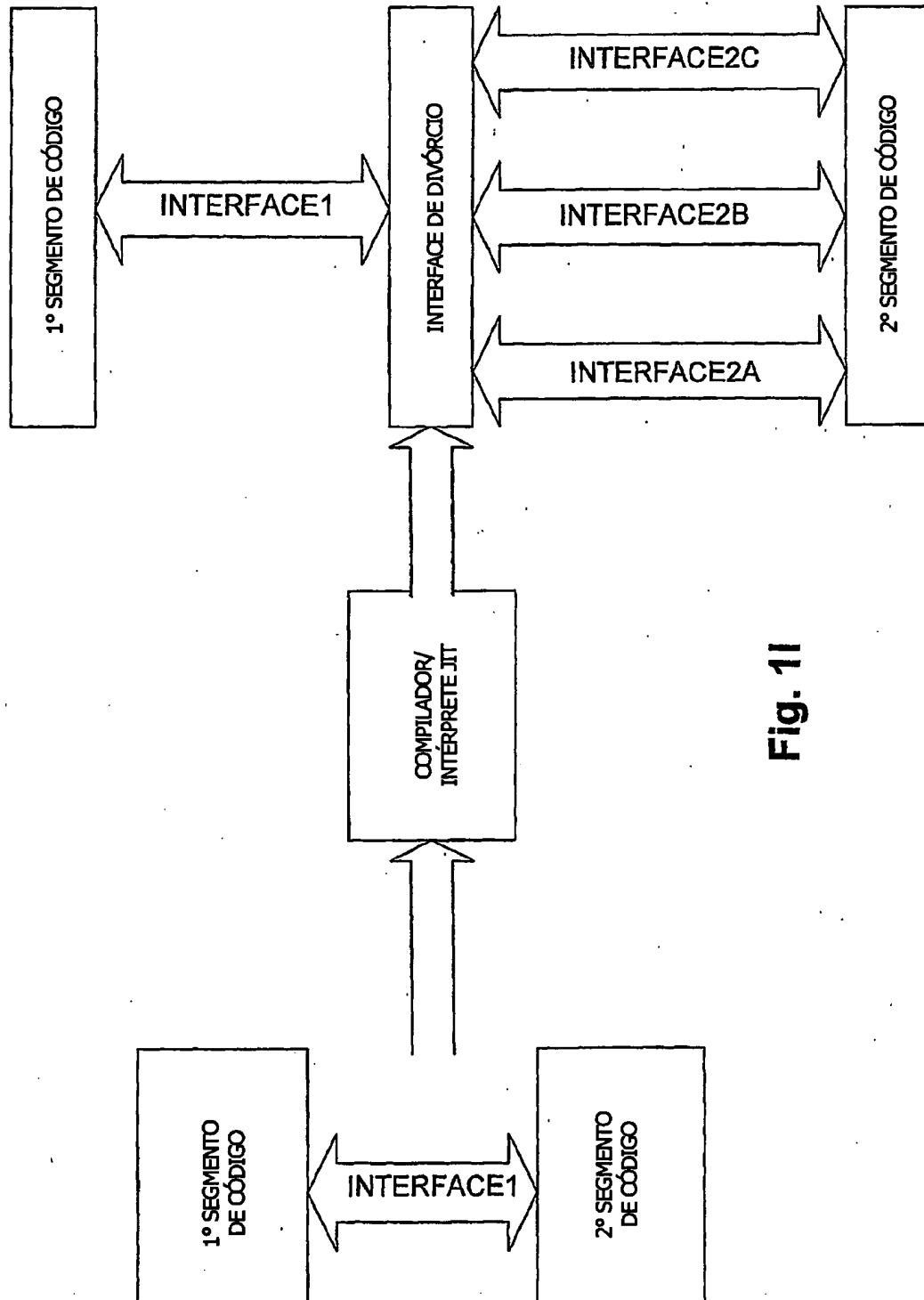
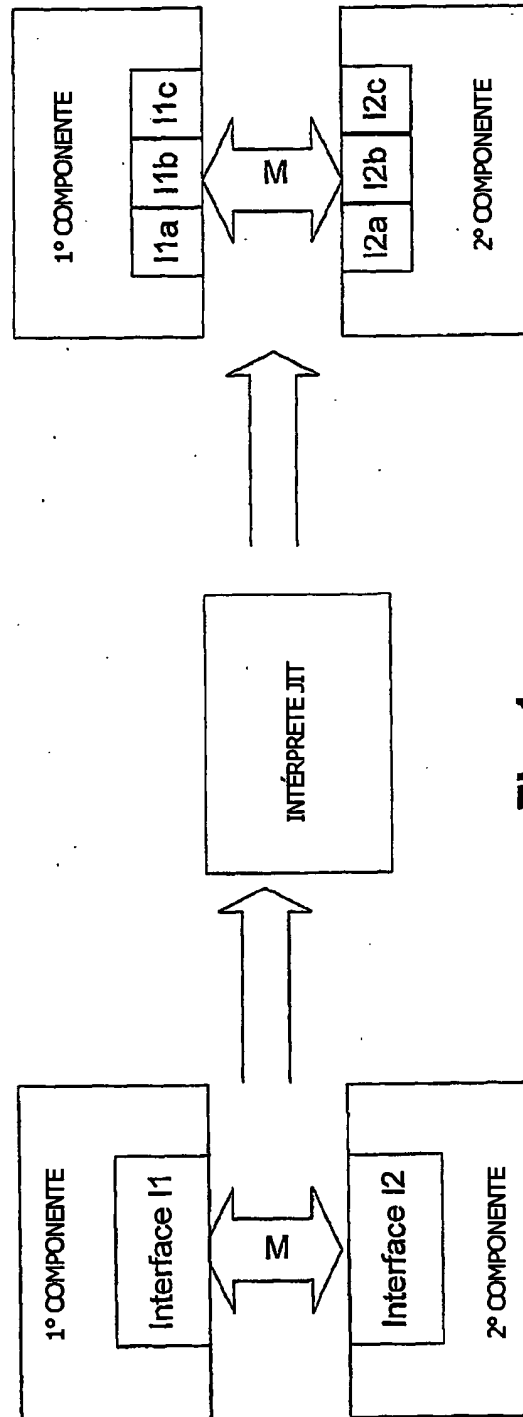


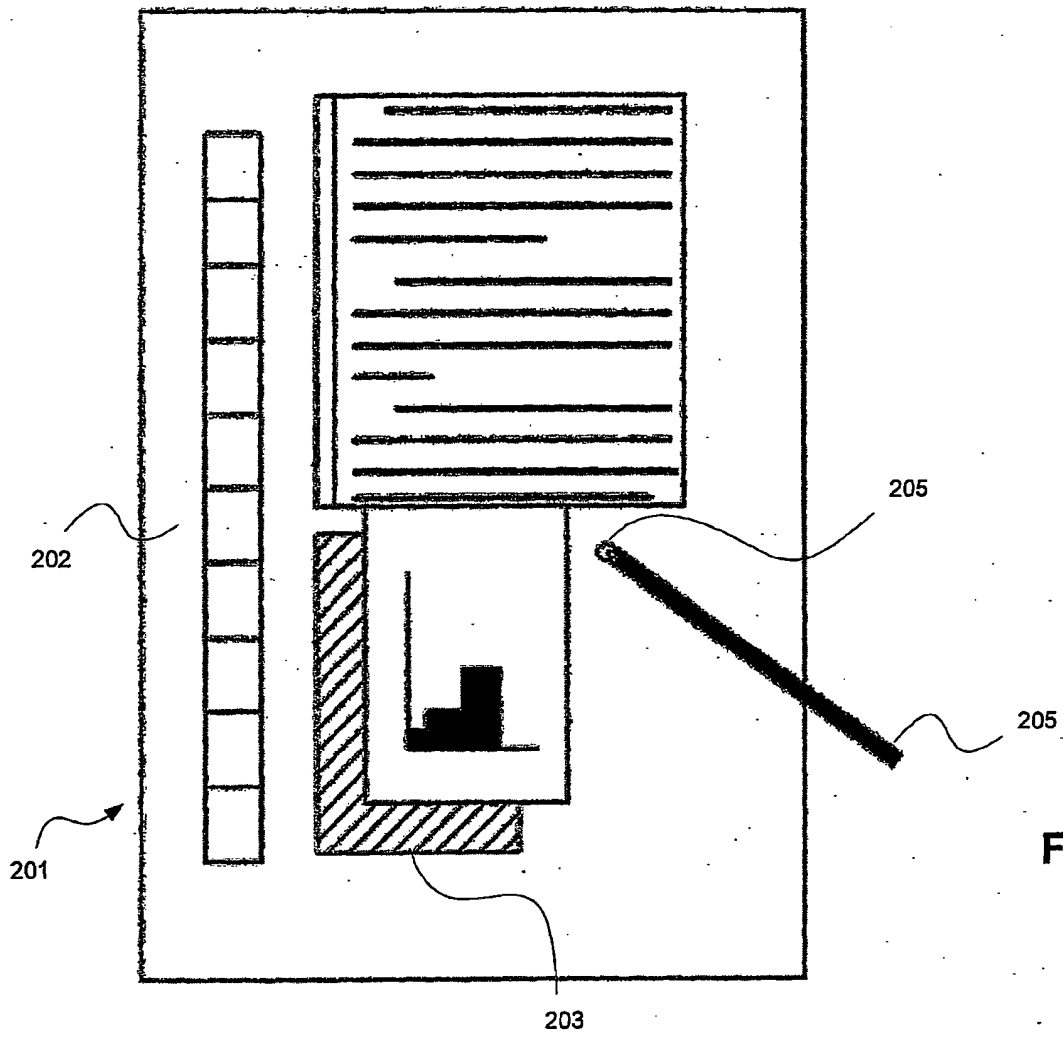
Fig. 1e





**Fig. 1I**

**Fig. 1m**



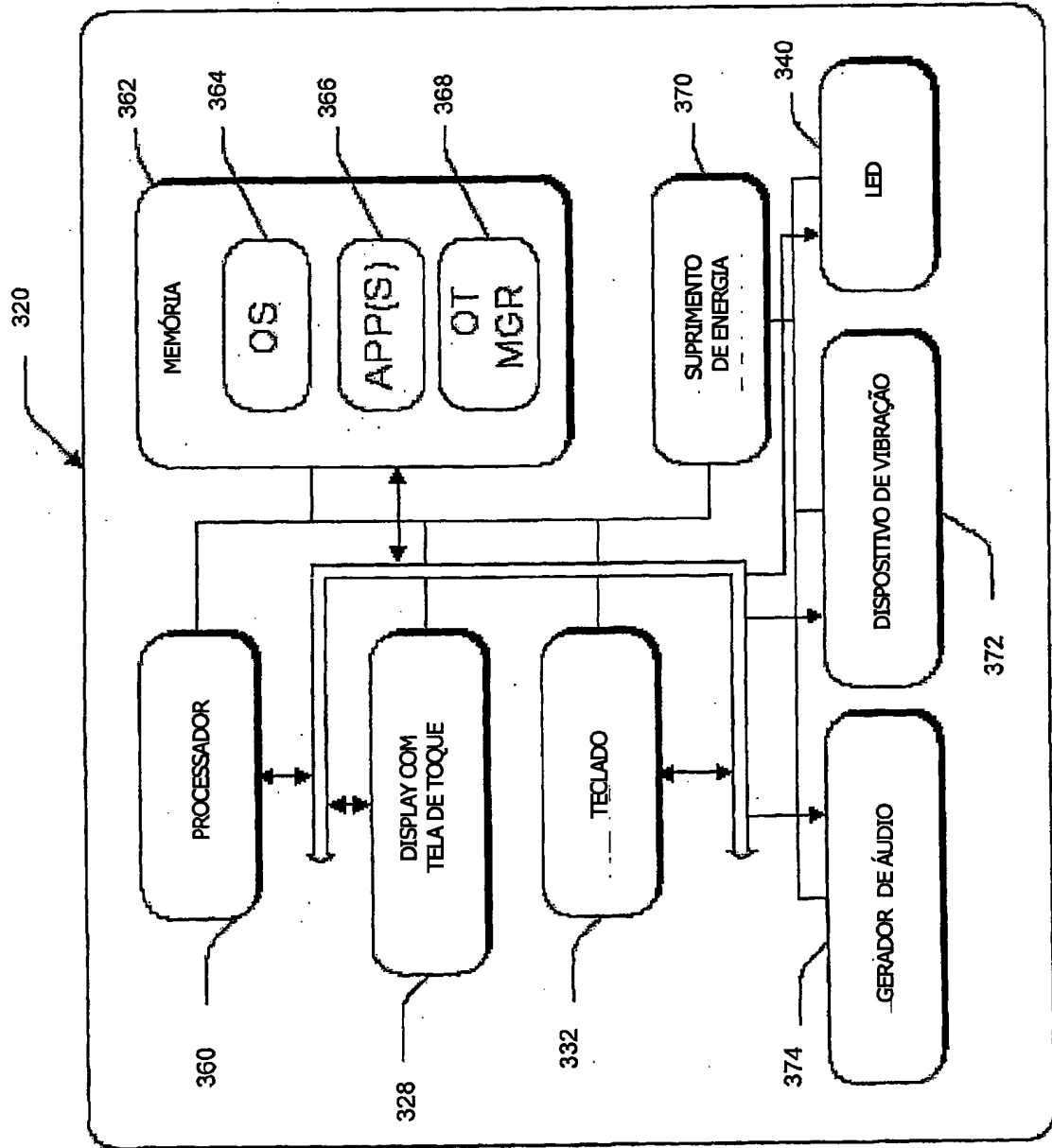


Fig. 3

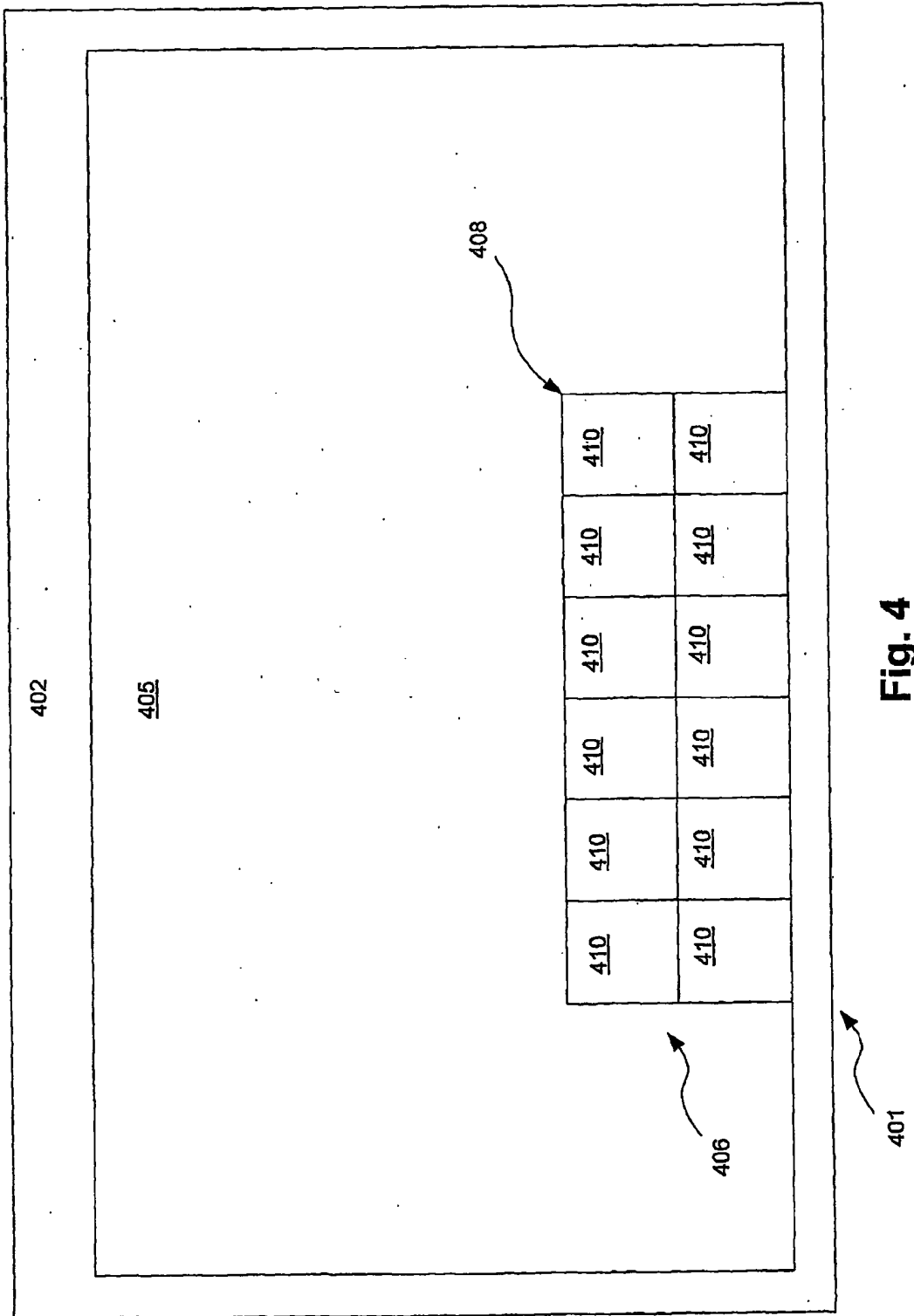


Fig. 4

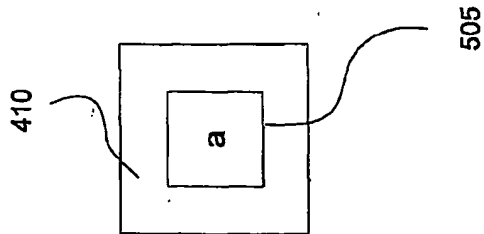


Fig. 5

<u>505b</u> À	<u>505a</u> A	<u>505h</u> Ā
<u>505c</u> à	<u>505</u> a	<u>505g</u> Ā
<u>505d</u> â	<u>505e</u> ā	<u>505f</u> ä

Fig. 6a

<u>505b</u> 1	<u>505a</u> 2	<u>505h</u> 3
<u>505c</u> 4	<u>505</u> 5	<u>505g</u> 6
<u>505d</u> 7	<u>505e</u> 8	<u>505f</u> 9

Fig. 6b

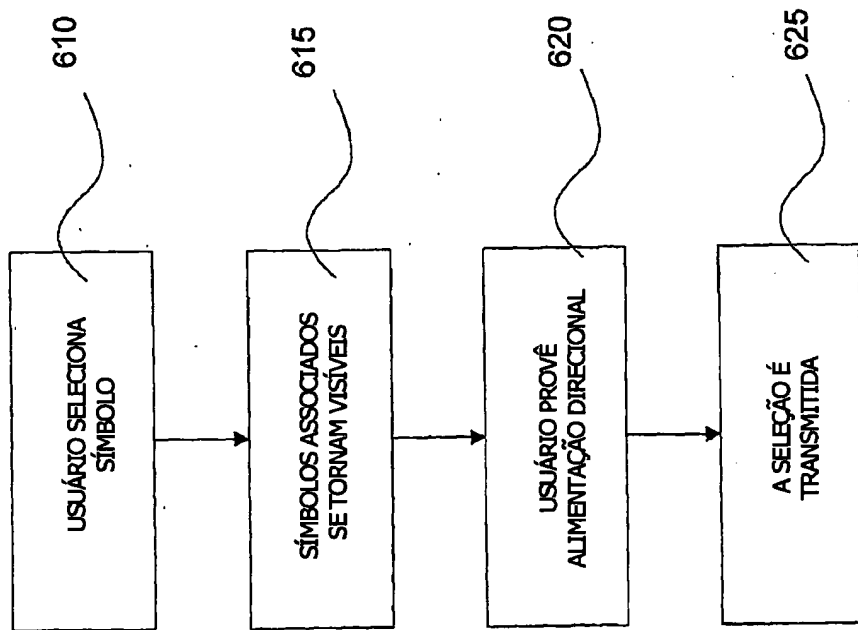


Fig. 7

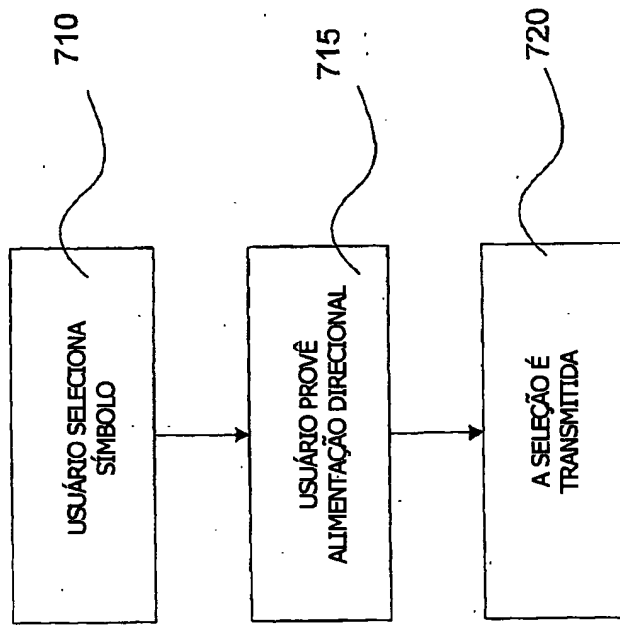
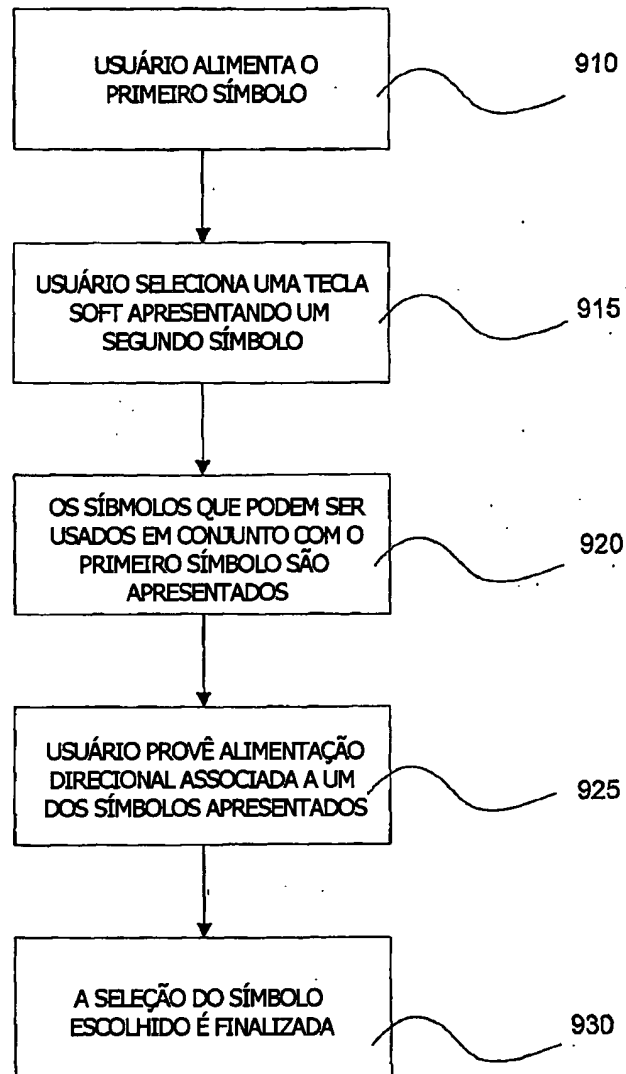


Fig. 8

**Fig. 9**

P10619245-9

RESUMO

"TECLADO VIRTUAL INTELIGENTE"

São descritos métodos para uso de um teclado virtual. É provido um teclado virtual possuindo uma ou mais teclas virtual com um símbolo apresentado nas uma ou mais teclas virtual. Um modificador pode ser aplicado a um dos símbolos apresentados sobre uma das teclas virtual através da seleção da tecla virtual, seguida pelo provimento de uma entrada direcional. Em uma modalidade, diferentes entradas direcionais são associadas a diferentes modificadores de forma a que diferentes símbolos sejam alimentados, dependendo da entrada direcional. Em uma modalidade, um usuário pode provar a entrada de direção por meio de um toque e arraste direcional de um seletor através de uma tela sensível ao toque. Em uma modalidade, uma pluralidade de símbolos associados a uma pluralidade de teclas modificadoras são tornados visíveis após a tecla virtual ser selecionada, de forma a que o usuário possa determinar visualmente qual entrada direcional irá prover o símbolo desejado.