



Universidade Federal de Santa Catarina – UFSC
Departamento de Informática e Estatística – INE
Ciências da Computação
Dalton Silva Ribeiro Filho – 0123207-0

Raciocínio Baseado em Casos Aplicado ao Cálculo da Nutrição Parenteral

Orientador:

Prof. Dr. rer.nat. Aldo von Wangenheim

Co-orientadores:

Dra. Mônica Akemi Kurahayashi dos Reis

Msc. Daniel Duarte Abdala

Agradeço a todos os que me orientaram neste trabalho, em especial à Dra. Mônica, por sua contribuição fundamental e eterna paciência.

Índice

Lista de Abreviaturas	2
Lista de Tabelas	3
Lista de Figuras	4
Resumo	5
Abstract	6
1. Introdução	7
1.1. Objetivo Principal	7
1.2. Objetivos Específicos	7
1.3. Motivação	8
1.4. RBC	8
2. Metodologia	9
2.1. Ciclo RBC	9
2.1.1. Recuperação	9
2.1.2. Reutilização	13
2.1.3. Revisão e Retenção	16
2.2. Teste Retrospectivo	16
2.2.1. Procedimento	16
2.2.2. Análise do Erro	17
3. Conclusões	19
3.1. Trabalhos Futuros	19
4. Referências Bibliográficas	20
ANEXO I – Código Fonte	21
ANEXO II – Manual	36
ANEXO III – Teste Retrospectivo I	39
ANEXO IV – Teste Retrospectivo II	48
ANEXO V – Cronograma	58

Lista de Abreviaturas

RN – Recém-nascido

PLPF – Pedro Luiz de Paula Filho

HIJG – Hospital Infantil Joana de Gusmão

NP – Nutrição Parenteral

BD – Banco de Dados

RBC – Raciocínio Baseado em Casos

UFSC – Universidade Federal de Santa Catarina

VIG – Velocidade de infusão de glicose

VHT – Volume hídrico total

Lista de Tabelas

Tabela 1: Prescrições iniciais disponíveis por data	11
Tabela 2: Discrepâncias entre prescrições	13
Tabela 3: Sugestões de prescrição criadas a partir das prescrições da Tabela 2, em ordem decrescente de médias	14
Tabela 4: Erro nas sugestões de prescrição feitas a partir da base de dados de 1º de Novembro de 2005	17
Tabela 5: Erro percentual nas sugestões de prescrição feitas a partir da base de dados de 1º de Novembro de 2005	18

Lista de Figuras

Figura 1: Estrutura do caso	10
Figura 2: O baixo crescimento das prescrições iniciais válidas	11
Figura 3: Fórmula para o cálculo da similaridade entre casos	12
Figura 4: Definição de um caso (sem descritores de entrada) e de uma sugestão de prescrição	14
Figura 5: Fórmula para o cálculo do i -ésimo atributo de solução da n -ésima sugestão de prescrição	14
Figura 6: Exemplo de cálculo para o atributo de solução <i>VIG</i> de três sugestões de prescrição	14
Figura 7: Seleção do RN previamente cadastrado	15
Figura 8: Botão para acesso às sugestões	15
Figura 9: Seleção de uma das três sugestões	15
Figura 10: Aplicação da sugestão na tela de cálculo	15

Resumo

Este trabalho descreve um experimento de inteligência artificial no campo da neonatologia, onde raciocínio baseado em casos (RBC) foi usado para sugerir prescrições iniciais de nutrição parenteral (NP), onde, ao contrário de prescrições não iniciais, o médico não possui nenhuma precedência que o auxilie na atribuição de valores de ingredientes da nutrição parenteral.

Para o experimento foi utilizada uma base de dados do Hospital Infantil Joana de Gusmão, alimentada por um programa desenvolvido no mestrado de Pedro Luiz de Paula Filho. Um módulo RBC foi adicionado a este programa para permitir a sugestão de prescrições a partir da tela cálculo de NP, com o uso de alguns dos dados cadastrais do RN como descritores de entrada.

Uma métrica de similaridade foi definida baseada em quatro critérios sugeridos pela especialista consultada, que também sugeriu um algoritmo de adaptação de casos. Testes retrospectivos foram realizados, onde NPs originalmente prescritas foram comparadas às sugestões de prescrição do módulo RBC.

Abstract

This work describes an artificial intelligence experiment in which case based reasoning was used to suggest initial prescriptions of parenteral nutrition, where, as opposed to posterior prescriptions, the doctor doesn't have the benefit of precedence to aid him on the assignment of ingredient values of parenteral nutrition.

This experiment used a database from Hospital Infantil Joana de Gusmão, populated by a program developed during the Msc. of Pedro Luiz de Paula Filho. A CBR module was added to this program to allow suggestions of prescriptions from within the parenteral nutrition calculation screen, using some of the newborn's registration data as entry descriptors.

A similarity metric was defined based on four criteria suggested by a consulted specialist, who also suggested an algorithm for case adaptation. Retrospective tests were carried, where parenteral nutritions originally prescribed were compared against prescription suggestions from the CBR module.

1. Introdução

Em fevereiro de 2004 foi defendida a dissertação de mestrado de Pedro Luiz de Paula Filho, cujo produto principal foi um software para o cálculo de prescrições de nutrição parenteral [PFILHO2004]. Este software registra em um banco de dados todas as informações necessárias a aquele cálculo, como o peso do recém-nascido ao longo dos dias em que está internado, a quantidade de cada ingrediente usado na fabricação do seu soro, medicações utilizadas, entre outros.

Com estas informações disponíveis, tornou-se possível criar um sistema que sugere novas prescrições a partir da semelhança de dados de entrada com prescrições já existentes no BD. O mecanismo que permite o cálculo desta semelhança é uma técnica de inteligência artificial chamada Raciocínio Baseado em Casos, cujo uso no desenvolvimento do sistema será discutido posteriormente neste trabalho.

A utilidade maior desta funcionalidade é a sugestão de prescrições iniciais, pois ao contrário de prescrições posteriores, onde o médico cria a prescrição baseada em modificações da prescrição do dia anterior, não há nenhum precedente. Com RBC, é possível utilizar prescrições iniciais que já foram utilizadas em outros pacientes para sugerir prescrições iniciais, baseada na semelhança do paciente atual com pacientes registrados num BD.

1.1 Objetivo Principal

Estudar as implicações do uso de RBC na análise da NP, com foco nas prescrições iniciais.

1.2 Objetivos Específicos

- Definir atributos de similaridade entre prescrições de NP
- Definir um algoritmo para adaptação dos casos sugeridos pela máquina de inferência RBC
- Comparar as prescrições sugeridas com as NPs originalmente prescritas
- Analisar o erro das prescrições sugeridas

1.3 Motivação

Prescrições iniciais – que ao contrário de prescrições não iniciais, não possuem a vantagem da análise de prescrições anteriores – dependem da experiência do médico para sua efetivação. Uma ferramenta que sugira prescrições iniciais com base em prescrições passadas provê uma confiabilidade útil ao médico que deseja maior embasamento na criação de prescrições.

1.4 RBC

Raciocínio baseado em casos foi o núcleo deste trabalho. Graças ao RBC, foi possível sugerir prescrições sem o uso de qualquer regra de cálculo de NP. Tudo o que foi necessário foi uma base de dados, dados de entrada de um RN, uma métrica de similaridade e um algoritmo de adaptação de casos.

RBC é uma técnica de inteligência artificial que resolve problemas a partir da semelhança destes com problemas que já foram resolvidos [WANG2003]. RBC parte do pressuposto de que problemas semelhantes possuem soluções semelhantes [WANG2003].

Se RBC for concebido como um sistema de sugestão de prescrições, sua entrada será um RN e sua saída serão sugestões de prescrições para este RN. O que este sistema faz ao receber o RN de entrada é compara-lo com RNs que conhece para encontrar qual é o mais semelhante a este RN de entrada, segundo uma métrica de similaridade. De posse desta informação, ele seleciona a solução adotada para este problema e o retorna como saída do sistema, possivelmente com alguma adaptação [WANG2003]. Um sistema RBC se diferencia de um sistema baseado em regras pelo fato de a solução ser encontrada a partir deste cálculo de similaridade e possíveis adaptações, não a partir do uso de regras, embora soluções mistas já tenham sido utilizadas [GOLDING95].

2. Metodologia

Testar a aplicabilidade de RBC na sugestão de prescrições de NP implicou, durante a maior parte deste projeto, num grande esforço de assessoramento e correção do programa que gerava os dados que seriam utilizados para criar casos. Foram necessários pelo menos seis meses de entrevistas com a especialista consultada (neste projeto, a Dra. Mônica dos Reis) e correções no programa antes da apresentação de um protótipo para a profissional (cf. anexo V). As etapas deste projeto, desde o assessoramento até a adaptação dos casos sugeridos, correspondem às etapas do ciclo RBC [WANG2003].

2.1. Ciclo RBC

A primeira etapa do projeto – que no ciclo RBC, corresponde à fase de recuperação – teve como objetivo determinar quais atributos seriam usados no cálculo de similaridade entre casos, qual seria a estrutura de um caso, qual seria a métrica usada na comparação entre casos e quantos casos dentre os mais similares seriam retornados. A segunda etapa, correspondente à fase de reutilização no ciclo RBC, procurou saber se seria necessária alguma adaptação nos casos retornados como mais similares. A partir da terceira e quarta fases do ciclo RBC – respectivamente, revisão e retenção – não foi mais necessário o auxílio da especialista para o desenvolvimento da máquina de inferência RBC.

2.1.1. Recuperação

Na primeira entrevista com a Dra. Mônica, o objetivo era descobrir como seria um caso neste domínio de problema. Um caso prescinde da definição de um par problema-solução [WANG2003]. O *quê* seriam o problema e a solução já era conhecido – o *problema* é o RN, a *solução* é uma prescrição para este RN – faltava saber que atributos teriam este problema e esta solução.

Anterior à primeira entrevista com a Dra. Mônica, foi criado um protótipo inicial no qual o problema era um arranjo com todos os atributos numéricos registrados no BD para um dado RN; a solução para o problema era um arranjo com todos os atributos numéricos registrados no BD de uma prescrição calculada para um RN. Esta modelagem pré-assessoramento do caso desconhecia quais atributos do problema seriam determinantes para encontrar a solução e quais atributos deveriam existir na solução para que esta fosse

satisfatória. Utilizar somente os atributos necessários é de interesse do usuário do sistema RBC, pois excesso de atributos na modelagem do caso acarreta em maior demora no cálculo da similaridade.

Baseado neste interesse por um caso apropriadamente modelado, foi realizado um assessoramento [WANG2003] na primeira entrevista com a Dra. Mônica, ou seja, foram investigados quais seriam os fatores mais determinantes na similaridade entre um RN e outro. O assessoramento pode ser a tarefa mais complexa dentre as sub-tarefas da recuperação de um caso [WANG2003], portanto era estratégico evitar dirigir-se à especialista mediante perguntas que só fariam sentido para o implementador do sistema RBC. Investigar o que é mais determinante na heurística [FEFERBAUM] usada por neonatologistas para prescrever NPs iniciais equivale a descobrir o quê é mais determinante na similaridade entre um RN e outro, afinal, tal heurística é baseada inteiramente na avaliação do RN [FEFERBAUM].

Do assessoramento inicial foi estabelecido somente que o fator mais determinante no cálculo da NP seria o peso do RN. Não foram assessorados quais seriam os atributos da solução nesta primeira entrevista (cf. anexo V). Em assessoramentos posteriores (cf. anexo V), foram definidos mais três fatores de similaridade entre prescrições: *idade gestacional*, *dias de vida* e *dias de parenteral*; também foram definidos os atributos da solução: *aminoácidos*, *lipídios*, *sódio*, *potássio*, *cálcio*, *magnésio*, *zinco*, *fósforo*, *VIG* e *VHT*.

Problema	Solução		
Peso	Aminoácidos	Cálcio	VIG
Idade gestacional	Lipídios	Magnésio	
Dias de vida	Sódio	Zinco	VHT
Dias de parenteral	Potássio	Fósforo	

Figura 1: Estrutura do caso

Qualquer que fosse a modelagem do caso, ela teria que levar em consideração o BD usado pelo programa de NP [WANG2003], pois este BD guarda os dados mediante o qual os casos seriam criados. O programa de NP até então utilizado removia RNs que não estavam mais internados da tabela de RNs no BD. Somente suas prescrições permaneciam gravadas. Dados importantes, como idade gestacional e dias de vida, só estavam presentes na tabela de RNs no BD. Do ponto de vista da recuperação de casos, somente as soluções eram armazenadas, sem seus respectivos problemas.

Como se não houvesse falhas suficientes na versão Smalltalk do programa, este não fazia nenhuma crítica dos dados cadastrais do RN, de forma que vários dados em branco,

inválidos ou absurdos eram registrados no BD – com isto, grande parte dos dados cadastrados no BD eram inválidos para a pesquisa: em 24 de maio de 2005, só havia 77 prescrições válidas dentre as 407 prescrições registradas no BD até aquela data; dentre as 77 prescrições válidas, somente 11 eram prescrições iniciais. A tabela abaixo mostra o crescimento do número de prescrições iniciais válidas disponíveis para pesquisa:

Data da coleta	P	PV	PIV
24/05/05	407	77	11
03/06/05	442	88	11
13/06/05	454	95	11
14/07/05	497	153	17
19/08/05	542	184	19
01/11/05	655	254	24
16/11/05	692	270	26
16/01/06	825	354	30
27/03/06	932	430	32

Tabela 1: Prescrições iniciais disponíveis por data. P = Prescrições; PV = Prescrições válidas; PIV = Prescrições iniciais válidas.

Note que em 14 de Julho de 2005 alguns RNs foram artificialmente tornados válidos depois de ajustes manuais no BD (por exemplo: RNs registrados com peso superior a 1000Kg tiveram seus pesos divididos por 1000).

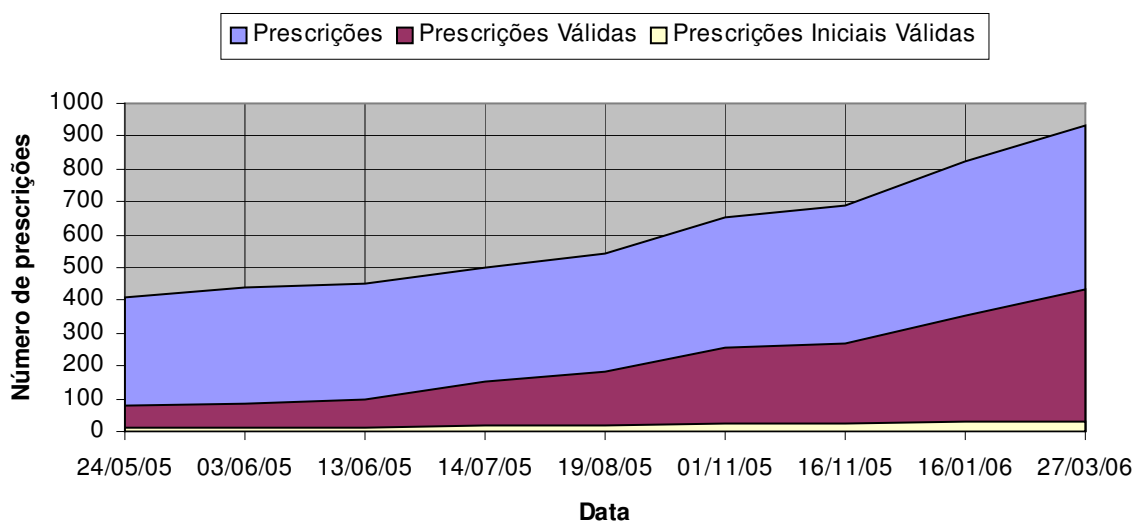


Figura 2: O baixo crescimento das prescrições iniciais válidas

Após a filtragem dos dados do BD, o programa de NP pode usar os dados válidos para criar objetos “Caso” e calcular a similaridade entre eles. Numa base de dados suficientemente grande, onde existisse uma quantidade razoável de casos para cada combinação possível de valores dos descritores de entrada (exceto o peso), o cálculo de similaridade poderia se restringir a calcular a diferença entre o peso do caso de entrada e o peso do caso armazenado, dado que combinações de peso, que é um número real, seriam difíceis. Casos cujos descritores de entrada (exceto o peso) fossem diferentes dos descritores de entrada do caso de entrada seriam imediatamente descartados na fase de filtragem de casos. Esta abordagem, embora cômoda, é inadequada: primeiro, porque desconsidera casos úteis à resolução do problema por diferenças mínimas – um caso armazenado cujo atributo *idade gestacional* fosse igual a 16 seria considerado completamente inútil para um caso de entrada cuja *idade gestacional* fosse igual a 17, mesmo que todos os outros atributos tivessem valores idênticos –; segundo, pois necessita de uma base de dados muito grande para ser funcional.

Portanto, foi utilizada uma métrica de similaridade que considera os quatro descritores de entrada obtidos na fase de assessoramento. Para o cálculo desta métrica foi utilizada a distância Euclidiana, por três razões: simplicidade de implementação; maior precisão em comparação à distância de Hamming; estrutura de caso simples: todos os descritores de entrada são números reais. Como o interesse maior desta pesquisa foi descobrir a aplicabilidade de RBC na sugestão de prescrições iniciais, o cálculo da distância foi ponderado para que o atributo *dias de parenteral* tivesse mais importância que os demais critérios, de forma que casos com os mesmos dias de parenteral fossem preponderantes dentre os casos mais similares encontrados pela máquina de inferência; assim, a importância de cada descritor de entrada foi: $W_{\text{peso}} = 1$; $W_{\text{ig}} = 1$; $W_{\text{dv}} = 1$ e $W_{\text{dp}} = 5$. O cálculo para a distância entre um caso de entrada *ce* e um caso armazenado *ca* é dado pela fórmula abaixo:

$$\text{Distância}(ce, ca) = \sqrt{W_{\text{peso}} \times \Delta_{\text{peso}}^2 + W_{\text{ig}} \times \Delta_{\text{ig}}^2 + W_{\text{dv}} \times \Delta_{\text{dv}}^2 + W_{\text{dp}} \times \Delta_{\text{dp}}^2}$$

Figura 3: Fórmula para o cálculo da similaridade entre casos, onde ig = idade gestacional; dv = dias de vida; dp = dias de parenteral.

O tipo de recuperação utilizada foi *seqüencial* [WANG2003]. Depois que um caso armazenado tinha sua distância ao caso de entrada calculado, aquele caso era armazenado em uma estrutura de dados ordenada em função daquela distância. No momento da seleção, bastava selecionar os *n* casos mais similares desta estrutura de dados. Como a base de dados

não tinha uma grande variedade de prescrições iniciais, um n alto implicaria em uma discrepância muito grande entre o caso mais similar e o n -ésimo mais similar; portanto, o n escolhido foi 3 por representar o ponto optimal entre o número e a discrepância entre sugestões.

RN teste

Dias de Vida	Idade Gestacional (DUM)	Dias de parenteral	Peso (Kg)
0	28	0	1.35

Prescrições em ordem decrescente de similaridade

VHT	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
80.0	4.0	2.0	0.5	3.0	0.0	0.0	4.0	0.25	0.0
80.0	5.0	3.0	0.0	3.0	400.0	0.0	2.0	0.25	0.0
90.0	6.0	2.5	1.0	3.0	400.0	0.0	2.0	0.25	0.0
80.0	6.0	3.0	1.0	3.0	400.0	2.0	4.0	0.25	0.0
90.0	5.0	3.0	1.0	3.0	0.0	0.0	4.0	0.25	0.0
90.0	6.0	3.0	1.0	3.0	400.0	2.0	4.0	0.25	0.0
80.0	6.0	3.0	1.0	3.0	400.0	0.0	2.0	0.25	0.0
80.0	7.0	2.0	0.0	4.0	0.0	1.0	2.0	0.25	0.0
70.0	4.0	2.0	0.5	0.0	400.0	0.0	4.0	0.25	0.0
100.0	7.0	3.0	2.0	3.0	400.0	2.0	2.0	0.25	0.0

Tabela 2: Discrepâncias entre prescrições. Pesquisa realizada no BD de 27 de Março de 2006 (cf. Tabela 1)

2.1.2. Reutilização

Em Agosto de 2005 foi apresentado à especialista consultada um protótipo no qual somente a fase de recuperação do ciclo RBC estava implementada. Neste protótipo, descritores de entrada e pesos correspondentes podiam ser modificados arbitrariamente (cf. anexo V). Após a definição dos destes itens, o protótipo mostrava todos os casos recuperados na interface, em ordem decrescente de similaridade com o caso de entrada. Após a apresentação, foi sugerido pela especialista que as prescrições sugeridas poderiam ser criadas a partir de médias dos valores das prescrições mais similares. Se somente uma sugestão fosse oferecida, bastaria fazer a média de todos os valores das prescrições, desde a prescrição mais similar até a n -ésima mais similar (n na versão final do programa de NP é três); mas para um número de sugestões maior que um, seria necessário que outras sugestões fossem resultado de

médias de um número menor de prescrições. Depois que n casos são selecionados na fase de recuperação, n sugestões de prescrição são criadas segundo a fórmula a seguir:

C_n = n-ésimo caso mais similar

c_{in} = i-ésimo atributo de solução do n-ésimo caso mais similar

S_n = n-ésima sugestão de prescrição

s_{in} = i-ésimo atributo de solução da n-ésima sugestão de prescrição

N = número de sugestões

$$C_n = \{c_{1n}, \dots, c_{in}\} \quad S_n = \{s_{1n}, \dots, s_{in}\}$$

Figura 4: Definição de um caso (sem descritores de entrada) e de uma sugestão de prescrição

$$s_{in} = \frac{\sum_{k=1}^{N-k+1} c_{ik}}{N-k+1}$$

Figura 5: Fórmula para o cálculo do i-ésimo atributo de solução da n-ésima sugestão de prescrição

Suponha que a *VIG* dos três casos mais similares são 4.0, 5.0 e 3.0, respectivamente. A *VIG* da primeira, segunda e terceira sugestão seria:

$$VIG_1 = \frac{4.0+5.0+3.0}{3-1+1} = 4.0 \quad VIG_2 = \frac{4.0+5.0}{3-2+1} = 4.5 \quad VIG_3 = \frac{4.0}{3-3+1} = 4.0$$

Figura 6: Exemplo de cálculo para o atributo de solução *VIG* de três sugestões de prescrição

VHT	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
83.29	4.89	2.43	0.66	2.98	181.11	0.36	3.24	0.25	0.00
80.29	4.59	2.33	0.51	2.68	181.11	0.16	3.44	0.25	0.00
81.40	4.92	2.33	0.46	3.12	136.66	0.27	3.21	0.25	0.00
81.40	4.79	2.46	0.58	3.00	186.66	0.15	3.21	0.25	0.00
82.83	4.80	2.46	0.58	3.00	186.66	0.43	3.5	0.25	0.00
82.83	4.63	2.46	0.58	3.00	120.0	0.1	3.5	0.25	0.00
80.83	4.83	2.46	0.58	3.00	200.0	0.5	3.5	0.25	0.00
83.33	4.83	2.33	0.58	3.00	200.0	0.0	3.0	0.25	0.00
80.00	4.50	2.5	0.25	3.00	200.0	0.0	3.0	0.25	0.00
80.00	4.00	2.0	0.5	3.00	0.0	0.0	4.0	0.25	0.00

Tabela 3: Sugestões de prescrição criadas a partir das prescrições da Tabela 2, em ordem decrescente de médias

A interface de sugestão de prescrições é disponibilizada para o usuário através de um botão que acessa uma janela onde as sugestões são mostradas. Este botão está contido na tela de cálculo do programa de NP, que por sua vez, só pode ser acessada mediante a seleção de um RN previamente cadastrado. É deste cadastro que a máquina de inferência obtém os dados para o cálculo de similaridade e para a sugestão de casos; desta forma, quando a tela de cálculo é acessada, a máquina de inferência já possui os dados necessários – tudo o que o usuário precisa para visualizar as sugestões é pressionar um botão. Se o usuário selecionar uma sugestão, os valores correspondentes da sugestão na tela de cálculo serão aplicados. Abaixo, capturas de tela desta interação na última versão do programa, em Java:



Figura 7: Seleção do RN previamente cadastrado

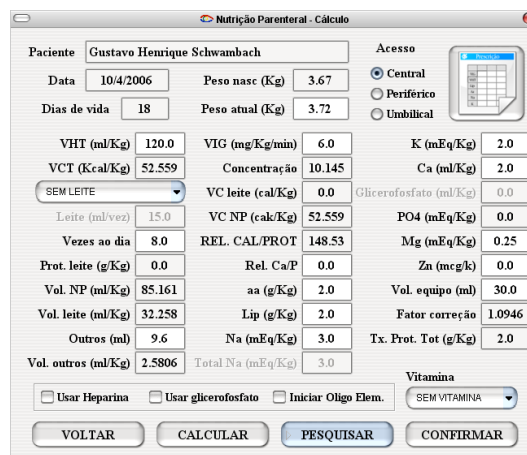


Figura 8: Botão para acesso às sugestões

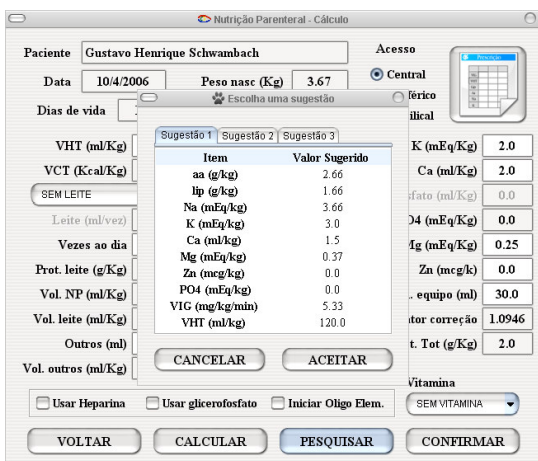


Figura 9: Seleção de uma das três sugestões

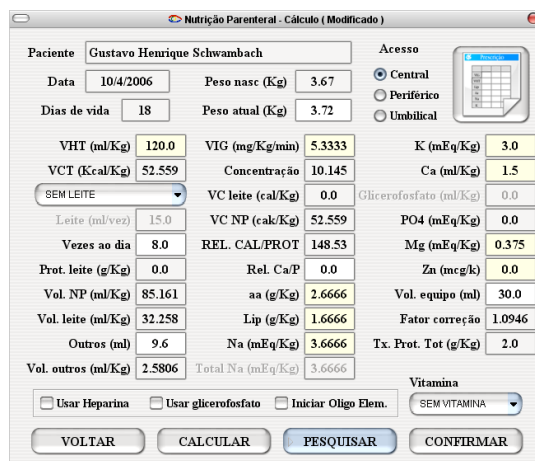


Figura 10: Aplicação da sugestão na tela de cálculo

2.1.3. Revisão e Retenção

A fase de revisão é executada inteiramente pelo usuário. Esta tarefa consiste em alterar os valores aplicados na tela de cálculo. Esta alteração é feita pela simples edição dos campos nos quais os valores da sugestão de prescrição foram aplicados. Depois que os valores são editados pelo usuário e o cálculo da prescrição já foi feito, procede a fase de retenção. Dada a estrutura simples do caso e as tabelas já existentes no BD, não foi necessário armazená-lo como um caso propriamente dito no BD. O BD contém duas tabelas cuja junção cria o caso na fase de recuperação, portanto, bastou gravar os dados da prescrição na tabela correspondente no BD; com exceção de *dias de vida*, dados correspondentes aos descritores de entrada fazem parte do cadastro do RN, cuja alteração só ocorre quando o usuário edita seu cadastro. O atributo *dias de vida* é obtido com a comparação da data da prescrição com a data de nascimento durante a fase de recuperação.

2.2. Teste Retrospectivo

O teste retrospectivo teve como objetivo verificar se o programa sugeriria prescrições plausíveis com o uso de prescrições de RNs já registrados no BD como dados de entrada. A semelhança entre as sugestões de prescrição e as prescrições efetivamente utilizadas é determinante no julgamento da plausibilidade desta implementação de RBC para a sugestão de prescrições iniciais. Os testes foram realizados com a base de dados em dois momentos: 1º de Novembro de 2005 e 27 de Março de 2006. Os resultados destes testes se encontram nos anexos III e IV, respectivamente.

2.2.1. Procedimento

1. Tomar uma prescrição inicial do BD;
2. Tomar os *dias de vida*, *idade gestacional* e *peso do RN* desta prescrição e considerar os descritores de entrada do sistema RBC;
3. Efetuar a fase de recuperação RBC, com três casos retornados e com a exclusão temporária do caso de entrada (de outra forma, o caso de entrada sempre seria retornado como o mais similar);
4. Efetuar a fase de reutilização RBC, com três sugestões de prescrição;
5. Comparar as sugestões com a prescrição de entrada.

2.2.2. Análise do Erro

Seria interessante se fosse possível utilizar um meio barato e objetivo de mensurar o erro nas sugestões de prescrição. A razão pelo qual este tipo de análise é inoportuna vem do conceito de erro numa prescrição de NP. Utilizar um g/kg de aminoácidos ao invés de três [FEFERBAUM] na primeira prescrição não é necessariamente um erro, mas uma diferença de política de prescrição: políticas mais ou menos conservadoras recomendam menos ou mais aminoácidos, respectivamente; por outro lado, uma dosagem de 20 g/kg de aminoácidos na primeira prescrição não é correta em nenhuma política de prescrição conhecida. Este tipo de erro nunca ocorrerá no sistema RBC deste projeto se este erro não ocorrer em nenhuma prescrição. Neste sistema, o valor de um atributo de solução de uma sugestão de prescrição está limitado pelo maior e menor valor do atributo correspondente num caso armazenado na base de casos. Este tipo de erro só é possível caso o programa de NP não emita alarmes no momento em que o médico insere um valor inválido de um item da prescrição durante o cálculo da NP – justamente a única fase do ciclo RBC deste sistema onde não há interferência da máquina de inferência.

Outro tipo de erro que pode ocorrer acontece pela má representação de um caso na base, pela má escolha dos pesos dos descritores de entrada, pelo número insuficiente destes descritores, ou ainda um mecanismo de adaptação não robusto o suficiente. Veja como exemplo, o caso do RN abaixo e as prescrições sugeridas pelo sistema; em seguida, compare com a prescrição original dada a este RN:

RN			
Dias de Vida	Idade Gestacional (DUM)	Dias de parenteral	Peso (Kg)
2	29	0	1.18

Prescrição original									
VHT	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
90.00	6.0	2.50	1.0	3.0	400.0	0.0	2.0	0.25	0

Prescrições sugeridas em ordem decrescente de médias									
VHT	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
96.67	6.67	3.00	2.00	3.00	400.0	2.00	2.67	0.25	0.00
105.00	7.00	3.00	2.50	3.00	400.0	2.00	2.00	0.25	0.00
100.00	7.00	3.00	2.00	3.00	400.0	2.00	2.00	0.25	0.00

Tabela 4: Erro nas sugestões de prescrição feitas a partir da base de dados de 1º de Novembro de 2005

Na tabela acima, segundo a especialista consultada, a justificativa para uma prescrição com K a 0 mEq/kg no terceiro dia de vida seria justificado por um nível alto de K no sangue do RN ou por insuficiência urinária. Não havia nenhum descritor de entrada para este problema específico, portanto, não foi dada preferência a este tipo de caso pela máquina de inferência – um caso com K a 2 mEq/kg seria considerado igualmente similar a um caso com K a 0 mEq/kg, desde que os dois tivessem os mesmos valores nos descritores de entrada.

Por causa das políticas de prescrição diferentes mencionadas anteriormente, não faz sentido utilizar erro percentual ou absoluto na análise do erro. Veja a tabela abaixo:

RN			
Dias de Vida	Idade Gestacional (DUM)	Dias de parenteral	Peso (Kg)
3	33	0	1.935

Prescrição original									
VHT	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
90.00	5.00	1.00	1.00	3.00	400.00	2.00	2.00	0.25	0.00

Prescrições sugeridas em ordem decrescente de médias									
VHT	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
110.00	7.67	2.33	2.33	3.00	400.00	2.00	2.00	0.25	0.00
105.00	8.50	2.50	2.50	3.00	400.00	2.00	2.00	0.25	0.00
100.00	8.00	2.00	2.00	3.00	400.00	2.00	2.00	0.25	0.00

Tabela 5: Erro percentual nas sugestões de prescrição feitas a partir da base de dados de 1º de Novembro de 2005

Note que o erro percentual nas sugestões de lipídios e aminoácidos é de no mínimo 100%, no entanto, segundo a especialista consultada, a sugestão não está errada: as sugestões de lipídios e aminoácidos são reflexo da nova política de prescrições adotada pelo HIJG.

Da mesma forma, não seria proveitoso utilizar erro absoluto como meio para análise do erro; para isto, seria necessário conhecer faixas de valores aceitáveis para cada valor de prescrição, para cada possível situação de um RN, para cada possível política de prescrição. Seria tão caro analisar o erro baseado no erro absoluto das sugestões, que se um programa fosse criado para esta análise, este poderia praticamente substituir o médico no momento da revisão de sugestões e inserção de valores no cálculo da prescrição, dado que o programa teria que conhecer a heurística de cada política de prescrição.

Portanto, para a análise do erro nas sugestões de prescrição deste sistema, o meio mais adequado é a análise do especialista. Em Novembro de 2005, uma análise das sugestões foi

feita pela especialista consultada; dos 24 conjuntos de 3 sugestões de prescrição, 3 conjuntos foram apontados como problemáticos pela profissional, contudo, estes conjuntos foram justamente os que apresentaram problema de má representação do caso na base ou de descritores de entrada insuficientes, cujas consequências já foram mencionadas anteriormente.

3. Conclusões

Durante a elaboração deste trabalho, foi notável a baixa quantidade de casos disponíveis para a pesquisa. Na última coleta de dados (cf. Tabela 1), somente 32 prescrições iniciais estavam disponíveis. Mesmo no teste retrospectivo baseado na base de dados de 1º de Novembro, onde a amostra era ainda menor (24 prescrições iniciais válidas), 87.5% dos RNs das prescrições correspondentes receberam sugestões plausíveis de prescrição de NP.

Dado que a sugestão de prescrições é baseada nos casos armazenados e não em regras independentes destes casos, o sistema de sugestões é adaptável a qualquer política de prescrição de NP. Esta mesma capacidade porém torna-se uma desvantagem caso o hospital decida mudar sua política de prescrições – neste caso, o sistema demorará tanto a se adaptar quanto for necessário para que as prescrições feitas com a nova política tornem-se suficientemente representativas na base de casos. Utilizar uma data inicial limite ou até mesmo usar a data da prescrição como um descritor de entrada pode ser uma solução, contanto que a taxa de inclusão de novos casos na base seja alta o bastante para garantir que haja casos suficientes dentro da faixa de tempo aceitável. Em outros sistemas, é necessário que o médico edite regras para se adaptar a novas políticas de prescrição [HORN]; no sistema desenvolvido neste projeto, o mecanismo que permite que as prescrições sugeridas se adaptem à política de prescrições do hospital é totalmente transparente para o usuário.

3.1 Trabalhos Futuros

Será fundamental refazer o teste retrospectivo quando aumentar o número de prescrições iniciais na base de casos do HIJG, que atualmente cresce a uma taxa de 80 cadastros por ano. A comparação dos testes retrospectivos futuros com testes retrospectivos anteriores dará uma idéia do aumento de segurança quando se utiliza uma base de casos onde há uma variedade maior de casos representados.

Também será interessante, ao se instalar o sistema de cálculo de NP em outros hospitais, comparar as diferenças entre sugestões de prescrição de NP entre estes. Um sistema web, com acesso a todas as bases de dados, poderá oferecer uma interface para um médico que queira estudar as diferenças de política de prescrição entre hospitais.

4. Referências Bibliográficas

FEFERBAUM, R.; FIGUEIREDO, A. D. **Necessidades nutricionais na nutrição parenteral prolongada em neonatologia.** Instituto de Metabolismo e Nutrição. Disponível em: < http://www.nutricaoclinica.com.br/index.php?option=com_content&task=view&id=25&Itemid=16# > Acessado em 12 de Abril de 2006.

GOLDING, A. R.; ROSENBLOOM, P. S. **Improving accuracy by combining rule-based and case-based reasoning.** Mitsubishi Electric Research Laboratories, 1995. Disponível em: < <http://www.merl.com/reports/docs/TR94-19a.pdf> > Acessado em 12 de Abril de 2006.

HORN, W, et al. **Development and evaluation of VIE-PNN, a knowledge-based system for calculating the parenteral nutrition of newborn infants.** Disponível em: < [http://wotan.liu.edu/docis/lib/cilt/rclis/dbl/arinme/\(2002\)24%253A3%253C217%253ADAEOVA%253E/www.ifs.tuwien.ac.at%252F~silvia%252Fpub%252Fpublications%252Fhor_aimj2002.pdf](http://wotan.liu.edu/docis/lib/cilt/rclis/dbl/arinme/(2002)24%253A3%253C217%253ADAEOVA%253E/www.ifs.tuwien.ac.at%252F~silvia%252Fpub%252Fpublications%252Fhor_aimj2002.pdf) > Acessado em 12 de Abril de 2006.

PAULA FILHO, P. L. **Modelo de Informatização de uma Unidade de Tratamento Intensivo Neonatal.** 2004. 88 f.. Dissertação (Mestrado em Ciências da Computação) – Curso de Ciências da Computação, Universidade Federal de Santa Catarina, Florianópolis, 2004.

WANGENHEIM, C. G. von; WANGENHEIM, A. **Raciocínio Baseado em Casos.** 1ª ed. Barueri: Manole, 2003. 293p.

ANEXO I – CÓDIGO FONTE

```
package teste.rbc;

import java.io.*;
import java.sql.*;
import java.util.*;
import modelo.rbc.*;
import modelo.bd.*;
import modelo.np.*;
import modelo.np.rbc.*;
import modelo.io.*;

/**
 * Realiza um teste retrospectivo com as prescrições de NP. O objetivo do teste
 * é determinar a diferença entre as prescrições sugeridas pelo programa e as
 * prescrições originais dadas a um RN.
 *
 * @author Dalton
 */
public class TesteRetrospectivo {

    // CONSTANTES *****

    /**
     * Tokens que serão substituídos pelos valores correspondentes no momento
     * da escrita do resultado do teste retrospectivo. Todos estes tokens estão
     * presentes no modelo em arquivo.
     */
    public static enum Token {NUMERO_CASO, DIAS_VIDA, IG, PESO, VIG, AA, LIP, NA,
        ZN, CA, K, MG, PO4}

    // PRIVADO *****

    /**
     * Cria um escritor baseado num modelo do teste retrospectivo em arquivo.
     * Ver documentação da classe EscritorModelo.
     *
     * @param original o caso com o qual todos os outros foram comparados
     * @param sugestoes as sugestões de prescrição
     * @param numeroCaso o número do caso, só para organização
     * @return EscritorModelo um escritor baseado num modelo do teste retrospectivo
     * @see {@link EscritorModelo}
     */
    private static final EscritorModelo getEscritor(CasoPrescricao original,
        ArrayList<CasoPrescricao> sugestoes, int numeroCaso) {

        EscritorModelo escritor = new EscritorModelo();

        // Número do caso, para organização
        escritor.adicioneValor(Token.NUMERO_CASO, numeroCaso);

        // Informações do RN original
        escritor.adicioneValor(Token.DIAS_VIDA, original.getDiasVida());
        escritor.adicioneValor(Token.IG, original.getIdadeGestacional());
        escritor.adicioneValor(Token.PESO, original.getPeso());

        // Valores originais da prescrição
        escritor.adicioneValor(Token.VIG, original.getVig());
        escritor.adicioneValor(Token.AA, original.getAminoAcidos());
        escritor.adicioneValor(Token.LIP, original.getLipidios());
        escritor.adicioneValor(Token.NA, original.getSodio());
        escritor.adicioneValor(Token.ZN, original.getZinco());
        escritor.adicioneValor(Token.CA, original.getCalcio());
        escritor.adicioneValor(Token.K, original.getPotassio());
        escritor.adicioneValor(Token.MG, original.getMagnesio());
        escritor.adicioneValor(Token.PO4, original.getFosforo());

        // Valores sugeridos de prescrição
        CasoPrescricao sugerido = null;

        for (int i = 0, c = 1; i < sugestoes.size(); i++, c++){
```

```

        sugerido = sugestoes.get(i);

        escritor.adicioneValor("VIG" + c, sugerido.getVig());
        escritor.adicioneValor("AA" + c, sugerido.getAminoAcidos());
        escritor.adicioneValor("LIP" + c, sugerido.getLipidios());
        escritor.adicioneValor("NA" + c, sugerido.getSodio());
        escritor.adicioneValor("ZN" + c, sugerido.getZinco());
        escritor.adicioneValor("CA" + c, sugerido.getCalcio());
        escritor.adicioneValor("K" + c, original.getPotassio());
        escritor.adicioneValor("MG" + c, sugerido.getMagnesio());
        escritor.adicioneValor("PO4" + c, sugerido.getFosforo());
    }

    return escritor;
}

/**
 * Obtém uma lista de casos de prescrição do BD utilizáveis para o teste
 * retrospectivo.
 *
 * @return uma lista de casos de prescrição do BD utilizáveis para o teste
 * retrospectivo
 * @throws Exception caso algum erro SQL ocorra ou não seja possível
 * conectar-se ao BD
 */
private ArrayList<CasoPrescricao> getCasosPrescricao() throws Exception {
    ArrayList<CasoPrescricao> resposta = new ArrayList<CasoPrescricao>();
    AcessoBD acessoBD = AcessoBD.getInstance();
    CasoPrescricao.inicieNormalizadores();
    ResultSet result = acessoBD.facaConsulta(getConsultaCasosPrescricao());
    result.first();
    do {
        resposta.add(new CasoPrescricao(result, result.getInt("newBornID")));
    } while (result.next());
    result.close();
    return resposta;
}

/**
 * Obtém o string com a consulta necessária para se obter todas as
 * prescrições iniciais de recém nascidos utilizáveis para o teste
 * retrospectivo
 *
 * @return o string para a consulta de todos os casos de prescrição
 * utilizáveis
 */
private String getConsultaCasosPrescricao(){
    StringBuffer consulta = new StringBuffer();
    consulta.append("SELECT newBorn.newBornID, newBorn.actualWeigth, ");
    consulta.append("(TO_DAYS(nppprescription.date) - ");
    consulta.append("TO_DAYS(newBorn.birthDate)) as 'diasVida',");
    consulta.append("pregnancyDurationDUM, prescriptionNumber,");
    consulta.append("vig, vht, aa, lipids, K, Ca, Na, Mg, Zn, PO4 ");
    consulta.append("FROM nppPrescription ");
    consulta.append("JOIN newBorn ON nppPrescription.newBornID = newBorn.newBornID ");
    consulta.append("WHERE newBorn.actualWeigth BETWEEN 0 AND 5 ");
    consulta.append("AND pregnancyDurationDUM > 0 ");
    consulta.append("AND prescriptionNumber = 0 ");
    return consulta.toString();
}

/**
 * Principal função da classe, faz o teste retrospectivo. Todos os casos
 * utilizáveis são obtidos, em seguida, cada caso é comparado com todos os
 * outros (exceto ele próprio) para determinar quais são os 3 casos mais
 * próximos deste. Em seguida, os 3 casos mais similares são enviados ao
 * escritor, de forma que ele obtenha o conteúdo do modelo do teste
 * retrospectivo em disco e substitua o valor dos tokens pelos valores das
 * sugestões de prescrição e também pelos valores da prescrição original.
 * Este conteúdo é acumulado num buffer de string, de forma que ao final do
 * método seja possível escrever este buffer num arquivo html para
 * visualizá-lo.
 *
 * @throws IOException caso algum erro de leitura \ escrita ocorra
 * @throws Exception caso algum problema no acesso ao BD ocorra
 */
private void facaTesteRetrospectivo() throws IOException, Exception {

```



```

// Todos os casos de prescrição utilizáveis para este teste
ArrayList<CasoPrescricao> casosPrescricao = getCasosPrescricao();
// Uma cópia desta lista, necessária para não ocorrerem erros de acesso concorrente
ArrayList<CasoPrescricao> casosCopiados =
    new ArrayList<CasoPrescricao>(casosPrescricao);
// O local do modelo em arquivo do teste retrospectivo
String modeloArquivo = "modeloRetrospectivo.htm";
// O número do caso, mostrado no modelo
int numeroCaso = 1;
EscritorModelo escritor = null;
ArrayList<CasoPrescricao> prescricoesSugeridas = null;
CalculoSimilaridade calculo = new CalculoSimilaridade(casosPrescricao.get(0));
StringBuffer bufferResposta = new StringBuffer("<html>\n");
for (CasoPrescricao caso : casosPrescricao){
    // Remove este caso da lista de casos
    casosCopiados.remove(caso);
    // Calcula as distâncias
    calculo.setNovoCalculo(caso, casosCopiados);
    // Pega as prescrições sugeridas
    prescricoesSugeridas = calculo.getCasosMaisSimilares(3);
    // Cria o escritor
    escritor = getEscritor(caso, prescricoesSugeridas, numeroCaso++);
    // Escreve o arquivo a partir do modelo
    bufferResposta.append(escritor.getConteudoArquivo(modeloArquivo));
    bufferResposta.append("\n<BR><BR>\n");
    // Re-adiciona o caso excluído
    casosCopiados.add(caso);
}
// Finaliza o html escrito
bufferResposta.append("</html>");
// Escreve o resultado do teste retrospectivo num arquivo HTML
escrevaStringBuffer("retrospectivo.html", bufferResposta);
}

/**
 * Escreve um buffer de string no disco.
 *
 * @param nomeArquivo o nome e local de destino do arquivo a ser gravado
 * @param buffer o buffer de string
 * @throws IOException caso ocorra algum problema de entrada \ saída
 */
private static void escrevaStringBuffer(String nomeArquivo, StringBuffer buffer)
    throws IOException {
    FileWriter escritor = new FileWriter(nomeArquivo);
    escritor.write(buffer.toString());
    escritor.close();
}

// MAIN *****

public static void main(String [] args){
    try {
        TesteRetrospectivo teste = new TesteRetrospectivo();
        teste.facaTesteRetrospectivo();
    }
    catch (Exception ex) {
        ex.printStackTrace();
    }
}

}

package modelo.io;

import java.io.*;
import java.util.TreeMap;
import java.util.ArrayList;

/**
 * Escreve um arquivo baseado num modelo que possui tokens. Os tokens devem
 * estar entre $$$. O valor de cada token deve ser passado a uma instância desta
 * classe antes de ela escrever o arquivo, ou o token ficará vazio no arquivo
 * escrito.
 *
 * @author Dalton
 */
public class EscritorModelo {

    // CONSTANTES *****

```

```

/** O caracter " como objeto String */
public static final String ASPAS = String.valueOf('"');

// INSTÂNCIA *****

/** Mapa que associa um token ao seu valor correspondente */
protected TreeMap<String, Object> mapaValores;

// CONSTRUTOR *****

/** Constrói um novo escritor de prescrições */
public EscritorModelo() {
    mapaValores = new TreeMap<String, Object>();
}

// ACESSO *****

/**
 * Insere o valor de um token para posterior criação do arquivo HTML
 *
 * @param token o token do arquivo
 * @param valor o valor do token
 */
public void adicioneValor(Object token, Object valor){
    mapaValores.put(token.toString(), valor);
}

/**
 * Escreve a prescrição no formato HTML
 *
 * @param localNomeModelo o caminho completo do modelo
 * @param localNomeDestino o caminho completo do destino
 *
 * @throws IOException caso não se encontre o modelo ou caso não se tenha
 * permissão para se escrever no caminho apontado
 */
public void escrevaArquivo(String localNomeModelo, String localNomeDestino)
    throws IOException {

    FileWriter escritor = new FileWriter(localNomeDestino);
    StringBuffer arquivo = getConteudoArquivo(localNomeModelo);
    escritor.write(arquivo.toString());
    escritor.close();
}

/**
 * Obtém o conteúdo de um arquivo modelo, já com seus tokens substituídos
 * pelos valores do mapa de valores desta classe
 *
 * @param localNomeModelo o caminho completo do modelo
 *
 * @return StringBuffer o buffer com o conteúdo do arquivo
 *
 * @throws IOException caso não se encontre o modelo ou caso não se tenha
 * permissão de leitura para se ler no caminho apontado
 */
public StringBuffer getConteudoArquivo(String localNomeModelo) throws IOException {
    File modelo = new File(localNomeModelo);
    Reader leitorArquivo = null;
    if (AcessoDisco.RODANDO_DE_JAR){
        InputStream stream = ClassLoader.getResourceAsStream(localNomeModelo);
        leitorArquivo = new InputStreamReader(stream);
    }
    else {
        leitorArquivo = new FileReader(modelo);
    }
    LineNumberReader leitor = new LineNumberReader(leitorArquivo);
    StringBuffer conteudo = new StringBuffer();
    String linha = leitor.readLine();
    while(linha != null){
        conteudo.append(processaLinha(linha));
        conteudo.append(AcessoDisco.NEWLINE);
        linha = leitor.readLine();
    }
    leitor.close();
    return conteudo;
}

```

```

}

/**
 * Substitui um token encontrado na linha por seu valor correspondente. Este
 * valor é extraído da estrutura de dados que mapeia um token ao seu valor.
 * Caso o valor não seja encontrado, o texto no modelo será deixado como
 * está.
 *
 * @param linha uma linha do arquivo
 *
 * @return a linha com os valores substituídos
 */
private String processeLinha(String linha){
    if (!linha.contains("$")){
        return linha;
    }

    // Último índice onde foi encontrado um cifrão
    int ultimoIndice = linha.indexOf('$');

    // Faz o parsing da linha
    while(ultimoIndice > 0){
        int comprimento = linha.indexOf('$',ultimoIndice+1) - ultimoIndice;
        String token = linha.substring(ultimoIndice+1,ultimoIndice+comprimento);
        Object valor = mapaValores.get(token);
        linha = linha.replace('$' + token + '$', valor != null ? valor.toString() : "");
        ultimoIndice = linha.indexOf('$',ultimoIndice+1);
    }

    return linha;
}

/**
 * Retorna uma faixa título cor azul claro com o título passado por parâmetro
 *
 * @param titulo o texto a ser escrito no HTML
 *
 * @return o string com o código HTML para ser usado na faixa título
 */
public static final String getFaixaTitulo(String titulo){
    StringBuffer resposta = new StringBuffer();
    resposta.append("<table width='100%' border='0' cellspacing='0' cellpadding='0'>");
    resposta.append("<tr><td bgcolor='#C0D0E2'>");
    resposta.append("<div align='center'><strong>");
    resposta.append(titulo);
    resposta.append("</strong></div></td>");
    resposta.append("</tr></table>");
    return resposta.toString();
}

/**
 * Retorna uma lista (de bolas) para ser usada em textos HTML
 *
 * @param lista a lista com os valores
 *
 * @return o código HTML para a lista de valores passados por parâmetro
 */
public static final String getListaBola(String... lista){
    StringBuffer resposta = new StringBuffer("<ul>");
    for (String s : lista){
        resposta.append("<li>");
        resposta.append(s);
        resposta.append("</li>");
    }
    resposta.append("</ul>");
    return resposta.toString();
}

/**
 * Retorna uma lista (de bolas) para ser usada em textos HTML
 *
 * @param lista a lista com os valores
 *
 * @return o código HTML para a lista de valores passados por parâmetro
 */
public static final String getListaBola(ArrayList<String> lista){
    StringBuffer resposta = new StringBuffer("<ul>");

```

```

        for (String s : lista){
            resposta.append("<li>");
            resposta.append(s);
            resposta.append("</li>");
        }
        resposta.append("</ul>");
        return resposta.toString();
    }
}

package modelo.np.rbc;

import java.sql.*;
import modelo.calc.*;
import modelo.np.bd.*;
import modelo.rbc.*;

/**
 * Uma prescrição representada como um caso.
 *
 * @author Dalton
 */
public class CasoPrescricao extends Caso <CasoPrescricao> {

    // INSTÂNCIA *****

    /** ID do RN no BD */
    protected int idRN;

    // INSTÂNCIA : CRITÉRIOS DE SIMILARIDADE *****

    /** Peso atual do paciente */
    protected double peso;

    /** Dias de vida do paciente */
    protected int diasVida;

    /** Idade gestacional do paciente */
    protected int idadeGestacional;

    /** Número de prescrições de nutrição parenteral que este paciente recebeu */
    protected int numParenterais;

    // INSTÂNCIA : PESO DOS CRITÉRIOS *****

    /** Importância do critério "peso" */
    protected static double pesoPeso = 1D;

    /** Importância do critério "dias de vida" */
    protected static double pesoDiasVida = 1D;

    /** Importância do critério "idade gestacional" */
    protected static double pesoIdadeGestacional = 1D;

    /** Importância do critério "número de parenterais" */
    protected static double pesoNumParenterais = 5D;

    // INSTÂNCIA : NORMALIZADORES DE DISTÂNCIA *****

    /** Normalizador de valores de peso */
    protected static Normal normalPeso;

    /** Normalizador de valores de dias de vida */
    protected static Normal normalDiasVida;

    /** Normalizador de valores de número de parenterais*/
    protected static Normal normalNumParenterais;

    /** Normalizador de valores de idade gestacional */
    protected static Normal normalIdadeGestacional;

    // INSTÂNCIA : ELEMENTOS DA PRESCRIÇÃO *****

    /** Velocidade de infusão de glicose */

```

```

protected double vig;

/** Volume hídrico total */
protected double vht;

/** Aminoácidos */
protected double aminoAcidos;

/** Lipídios */
protected double lipidios;

/** Sódio */
protected double sodio;

/** Potássio */
protected double potassio;

/** Cálcio */
protected double calcio;

/** Magnésio */
protected double magnesio;

/** Zinco */
protected double zinco;

/** Fósforo */
protected double fosforo;

// CONSTRUTOR *****

/**
 * Construtor usado quando se deseja criar um novo caso, geralmente o caso
 * principal com o qual outros casos serão comparados.
 *
 * @param peso peso atual do RN em Kg
 * @param diasVida dias de vida do RN
 * @param idadeGestacional idade gestacional do RN (DUM)
 * @param numParenterais número de parenterais que este RN já teve
 */
public CasoPrescricao(double peso, int diasVida, int idadeGestacional,
int numParenterais){

    verifiqueInicializacaoNormalizadores();
    this.peso = peso;
    this.diasVida = diasVida;
    this.idadeGestacional = idadeGestacional;
    this.numParenterais = numParenterais;
}

/**
 * Construtor usado quando se deseja criar um caso a partir de dados
 * existentes no BD.
 *
 * @param result um ResultSet do BD
 * @throws SQLException caso algum erro SQL ocorra
 */
public CasoPrescricao(ResultSet result) throws SQLException {
    verifiqueInicializacaoNormalizadores();
    vig = result.getDouble("vig");
    diasVida = result.getInt("diasVida");
    peso = result.getDouble("actualWeight");
    idadeGestacional = result.getInt("pregnancyDurationDUM");
    numParenterais = result.getInt("prescriptionNumber");
    vht = result.getDouble("vht");
    aminoAcidos = result.getDouble("aa");
    lipidios = result.getDouble("lipids");
    potassio = result.getDouble("K");
    sodio = result.getDouble("Na");
    calcio = result.getDouble("Ca");
    magnesio = result.getDouble("Mg");
    zinco = result.getDouble("Zn");
    fosforo = result.getDouble("PO4");
}

/**
 * Construtor usado quando se deseja criar um caso a partir de dados

```

```

* existentes no BD e se saber de quem é o caso.
*
* @param result um ResultSet do BD
* @param idRN o ID do RN
* @throws SQLException caso algum erro SQL ocorra
*/
public CasoPrescricao(ResultSet result, int idRN) throws SQLException {
    this(result);
    this.idRN = idRN;
}

/** Verifica se todos os normalizadores estão inicializados */
private void verifiqueInicializacaoNormalizadores(){
    if (normalPeso == null || normalDiasVida == null ||
        normalNumParenterais == null || normalIdadeGestacional == null){

        throw new RuntimeException("Você deve inicializar os normalizadores antes");
    }
}

/**
* Inicializa os normalizadores usados no cálculo da distância entre casos
*
* @throws SQLException caso algum erro SQL ocorra
* @throws ClassNotFoundException caso não seja encontrado o driver MySQL
*/
public static void inicieNormalizadores() throws SQLException, ClassNotFoundException {
    ProxyBD proxyBD = ProxyBD.getInstance();
    // Cria os normalizadores
    normalPeso = new Normal(proxyBD.getMinMaxPesoAtual(true),
        proxyBD.getMinMaxPesoAtual(false));
    normalDiasVida = new Normal(proxyBD.getMinMaxDiasVida(true),
        proxyBD.getMinMaxDiasVida(false));
    normalNumParenterais = new Normal(OD,
        proxyBD.getMinMaxNumParenterais(false));
    normalIdadeGestacional = new Normal(proxyBD.getMinMaxIdadeGestacional(true),
        proxyBD.getMinMaxIdadeGestacional(false));
}

// ACESSO *****

/**
* Obtém o string para consulta dos casos no BD
*
* @param rnExcluido o ID do RN que não deve estar incluso nos casos pesquisados
*
* @return o string para consulta dos casos no BD
*/
public static String getConsultaBD(int rnExcluido){
    return "SELECT nppPrescription.actualWeigth, pregnancyDurationDUM, "+
        " prescriptionNumber, (TO_DAYS(nppprescription.date) - "+
        "TO_DAYS(newBorn.birthDate)) as 'DiasVida', aa, lipids, Na, K, "+
        " Ca, Mg, Zn, VIG, VHT FROM nppPrescription JOIN newBorn ON "+
        "nppPrescription.newBornID = newBorn.newBornID WHERE "+
        "pregnancyDurationDUM > 0 AND nppPrescription.actualWeigth BETWEEN"+
        " 0.1 AND 5 AND newBorn.newBornID <> " +rnExcluido;
}

/**
* Obtém a distância entre esta prescrição e uma outra. O cálculo da
* distância é calculado a partir dos critérios peso, dias de vida, idade
* gestacional e número de parenterais
*
* @param caso a prescrição com o qual esta prescrição será comparada
* @return a distância entre este caso e o caso passado por parâmetro
*/
public double getDistancia(CasoPrescricao caso) {
    double distPeso = pesoPeso * Math.pow(
        normalPeso.getValorNormalizado(peso) -
        normalPeso.getValorNormalizado(caso.peso), 2D);
    double distDV = pesoDiasVida * Math.pow(
        normalDiasVida.getValorNormalizado(diasVida) -
        normalDiasVida.getValorNormalizado(caso.diasVida), 2D);
    double distNP = pesoNumParenterais * Math.pow(
        normalNumParenterais.getValorNormalizado(numParenterais) -
        normalNumParenterais.getValorNormalizado(caso.numParenterais), 2D);
    double distIG = pesoIdadeGestacional * Math.pow(

```

```

        normalIdadeGestacional.getValorNormalizado(idadeGestacional) -
        normalIdadeGestacional.getValorNormalizado(caso.idadeGestacional), 2D);

    return Math.sqrt(distPeso + distDV + distNP + distIG);
}

public double getAminoAcidos() {
    return aminoAcidos;
}

public double getCalcio() {
    return calcio;
}

public int getDiasVida() {
    return diasVida;
}

public int getIdadeGestacional() {
    return idadeGestacional;
}

public double getLipidios() {
    return lipidios;
}

public double getMagnesio() {
    return magnesio;
}

public int getNumParenterais() {
    return numParenterais;
}

public double getPeso() {
    return peso;
}

public static double getPesoDiasVida() {
    return pesoDiasVida;
}

public static double getPesoIdadeGestacional() {
    return pesoIdadeGestacional;
}

public static double getPesoNumParenterais() {
    return pesoNumParenterais;
}

public static double getPesoPeso() {
    return pesoPeso;
}

public static void setPesoDiasVida(double pesoDiasVida) {
    CasoPrescricao.pesoDiasVida = pesoDiasVida;
}

public static void setPesoPeso(double pesoPeso) {
    CasoPrescricao.pesoPeso = pesoPeso;
}

public static void setPesoIdadeGestacional(double pesoIdadeGestacional) {
    CasoPrescricao.pesoIdadeGestacional = pesoIdadeGestacional;
}

public static void setPesoNumParenterais(double pesoNumParenterais) {
    CasoPrescricao.pesoNumParenterais = pesoNumParenterais;
}

public void setAminoAcidos(double aminoAcidos) {
    this.aminoAcidos = aminoAcidos;
}

public void setCalcio(double calcio) {
    this.calcio = calcio;
}

```

```

public void setLipidios(double lipidios) {
    this.lipidios = lipidios;
}

public void setMagnesio(double magnesio) {
    this.magnesio = magnesio;
}

public void setPeso(double peso) {
    this.peso = peso;
}

public void setPotassio(double potassio) {
    this.potassio = potassio;
}

public void setSodio(double sodio) {
    this.sodio = sodio;
}

public void setVht(double vht) {
    this.vht = vht;
}

public void setVig(double vig) {
    this.vig = vig;
}

public void setZinco(double zinco) {
    this.zinco = zinco;
}

public void setFosforo(double fosforo) {
    this.fosforo = fosforo;
}

public double getPotassio() {
    return potassio;
}

public double getSodio() {
    return sodio;
}

public double getVht() {
    return vht;
}

public double getVig() {
    return vig;
}

public double getZinco() {
    return zinco;
}

public double getFosforo() {
    return fosforo;
}

public String toString() {
    return "Peso: " + peso + " DV: " + diasVida + " IG: " + idadeGestacional
        + " NP: " + numParenterais;
}
}

package modelo.rbc;

import java.util.Set;
import java.util.TreeMap;
import java.util.ArrayList;
import java.util.Iterator;

/**

```



```

* Efetua um cálculo de similaridade simples: compara a distância entre o caso
* principal e todos os casos que forem inseridos. Todos os objetos inseridos
* vão para uma estrutura de dados que os ordena segundo sua distância com o
* caso principal.
*
* @author Dalton
*/
public class CalculoSimilaridade <C extends Caso> {

    // INSTÂNCIA *****

    /** O caso com o qual todos os outros casos serão comparados */
    protected C casoPrincipal;

    /** Casos que foram comparados ao caso principal */
    protected TreeMap<Double, C> casos;

    // CONSTRUTOR *****

    /**
     * Cria um novo cálculo de similaridade a partir de um caso com o qual todos
     * os outros serão comparados.
     *
     * @param casoPrincipal o caso com o qual todos os outros casos serão comparados
     */
    public CalculoSimilaridade(C casoPrincipal){
        this.casoPrincipal = casoPrincipal;
        casos = new TreeMap<Double,C>();
    }

    // ACESSO *****

    /**
     * Adiciona um caso que será comparado ao caso principal e inserido numa
     * estrutura de dados ordenada. É neste método que é feito o cálculo da
     * distância entre casos.
     *
     * @param caso um caso a ser comparado com o caso principal
     */
    public void addCaso(C caso){
        casos.put(casoPrincipal.getDistancia(caso), caso);
    }

    /**
     * Obtém os <code>n</code> casos mais similares.
     *
     * @param n o número de casos mais similares a ser obtido. Pode ser menor
     * caso o número de casos seja menor que <code>n</code>
     *
     * @return os <code>n</code> casos mais similares.
     */
    public ArrayList<C> getCasosMaisSimilares(int n){
        ArrayList<C> resposta = new ArrayList<C>(n);
        Set<Double> chaves = casos.keySet();
        int max = Math.min(n, chaves.size());
        int cont = 0;
        Iterator<Double> iterador = chaves.iterator();
        while (cont < max){
            resposta.add(casos.get(iterador.next()));
            cont++;
        }
        return resposta;
    }

    /**
     * Reinicia o cálculo de similaridade com um novo caso principal e uma nova
     * lista de casos a comparar.
     *
     * @param casoPrincipal o caso principal com o qual todos os casos da lista
     * serão comparados
     * @param listaCasos a lista de casos com o qual o caso principal será
     * comparado
     */
    public void setNovoCalculo(C casoPrincipal, ArrayList<C> listaCasos){
        if (casoPrincipal != null && casos != null || casos.size() > 0){
            this.casoPrincipal = casoPrincipal;
            casos.clear();
        }
    }
}

```

```

        for (C caso : listaCasos){
            casos.put(casoPrincipal.getDistancia(caso), caso);
        }
    }
}

/** Imprime na tela todos os casos em ordem de similaridade */
public void listeCasosSimilares(){
    Set<Double> chaves = casos.keySet();
    for (double distancia : chaves){
        System.out.println("Distância: " + distancia + " Caso: " + casos.get(distancia));
    }
}
}

```

```
package modelo.rbc;
```

```

/**
 * Um caso. Todo caso deve saber calcular a distância entre ele e um outro caso.
 *
 *
 * @author Dalton
 */
public abstract class Caso <C extends Caso> {

    /**
     * Obtém a distância deste caso em relação a outro caso. A métrica a ser
     * utilizada é livre.
     *
     * @param caso o caso com o qual se fará a comparação
     *
     * @return a distância deste caso em relação ao caso passado por parâmetro
     */
    public abstract double getDistancia(C caso);
}

```

```
package modelo.rbc;
```

```

import java.util.ArrayList;

/**
 * Realiza uma adaptação num conjunto de casos
 *
 * @author Dalton
 */
public interface AdaptadorCasos <C extends Caso> {

    /**
     * Adapta um conjunto de casos
     *
     * @param casos o conjunto de casos
     */
    void adapteCasos(ArrayList<C> casos);
}

```

```
package modelo.np.rbc;
```

```

import java.util.*;
import modelo.rbc.*;

/**
 * Considere C o conjunto de casos de entrada e C' o conjunto de casos adaptado.
 * Considere c(i) um elemento do conjunto de casos de entrada, c'(i) um elemento
 * do conjunto de casos de saída e n o tamanho do conjunto C e C' (pois C' tem o
 * mesmo tamanho de C). Considere que o índice começa em 0 e vai até n-1.
 *
 * Um elemento c'(i) será igual a ( c(0) + ... + c(n-(i+1)) ) / (n-(i+1))
 */

```

```

*
* @author Dalton
*/
public class AdaptadorMedias implements AdaptadorCasos<CasoPrescricao> {

    /**
     * Adpta os casos de forma que o caso n seja a média de todos os casos
     * anteriores.
     *
     * @param casos a lista de casos a ser adaptada
     */
    public void adapteCasos(ArrayList<CasoPrescricao> casos) {
        for (int i = 1; i < casos.size(); i++){
            CasoPrescricao caso = casos.get(i);
            for (int j = 0; j < i; j++){
                CasoPrescricao outro = casos.get(j);
                caso.setAminoAcidos(caso.getAminoAcidos() + outro.getAminoAcidos());
                caso.setCalcio(caso.getCalcio() + outro.getCalcio());
                caso.setLipidios(caso.getLipidios() + outro.getLipidios());
                caso.setMagnesio(caso.getMagnesio() + outro.getMagnesio());
                caso.setPotassio(caso.getPotassio() + outro.getPotassio());
                caso.setSodio(caso.getSodio() + outro.getSodio());
                caso.setVht(caso.getVht() + outro.getVht());
                caso.setVig(caso.getVig() + outro.getVig());
                caso.setZinco(caso.getZinco() + outro.getZinco());
                caso.setFosforo(caso.getFosforo() + outro.getFosforo());
            }
            int soma = i + 1;
            caso.setAminoAcidos(caso.getAminoAcidos() / soma);
            caso.setCalcio(caso.getCalcio() / soma);
            caso.setLipidios(caso.getLipidios() / soma);
            caso.setMagnesio(caso.getMagnesio() / soma);
            caso.setPotassio(caso.getPotassio() / soma);
            caso.setSodio(caso.getSodio() / soma);
            caso.setVht(caso.getVht() / soma);
            caso.setVig(caso.getVig() / soma);
            caso.setZinco(caso.getZinco() / soma);
            caso.setFosforo(caso.getFosforo() / soma);
        }
        inverta(casos);
    }

    /**
     * Inverte a estrutura de dados de forma que o que sofreu menos médias seja
     * o último sugerido e o que sofreu mais médias seja o primeiro sugerido.
     *
     * @param casos a lista de casos
     */
    private void inverta(ArrayList<CasoPrescricao> casos){
        for (int i = 0; i < casos.size() / 2; i++){
            CasoPrescricao casoTrocado = casos.get(i);
            int posTroca = casos.size() - (i + 1);
            casos.set(i, casos.get(posTroca));
            casos.set(posTroca, casoTrocado);
        }
    }
}

package modelo.calc;

/**
 * Responsável por obter o valor normalizado de um atributo numérico
 *
 * @author Dalton
 */
public class Normal {

    // INSTÂNCIA *****

    /** Menor valor de um atributo num conjunto de dados */
    protected double menorValor;

    /** Maior valor de um atributo num conjunto de dados */
    protected double maiorValor;
}

```

```

/**
 * O inverso da diferença entre o maior e o menor valor deste atributo no
 * conjunto de dados. Pode ser expresso como 1 / (maiorValor - menorValor).
 */
protected double inversoDiferenca;

// CONSTRUTOR *****

/**
 * Cria um novo normalizador. Este construtor chama o método <code>
 * setEscala()</code> para inicializar variáveis internas.
 *
 * @param menorValor o menor valor deste atributo no conjunto de dados
 * @param maiorValor o maior valor deste atributo no conjunto de dados
 */
public Normal(double menorValor, double maiorValor){
    setEscala(menorValor, maiorValor);
}

// ACESSO *****

/**
 * O valor normalizado deste atributo numérico
 *
 * @param valor o atributo numérico a ser normalizado
 *
 * @return o valor normalizado deste atributo numérico
 */
public double getValorNormalizado(double valor){
    return inversoDiferenca * (valor - menorValor);
}

/**
 * O valor correspondente a um valor normalizado.
 *
 * @param normal o valor normalizado, de 0 a 1
 *
 * @return o valor correspondente a este valor normalizado
 */
public double getValorReal(double normal){
    return menorValor + (normal * (maiorValor - menorValor));
}

/**
 * Determina a escala deste normalizador. Este método é útil quando se
 * deseja usar novos valores para normalização sem inicializar um novo
 * normalizador.
 *
 * @param menorValor o menor valor deste atributo no conjunto de dados
 * @param maiorValor o maior valor deste atributo no conjunto de dados
 */
public void setEscala(double menorValor, double maiorValor){
    this.maiorValor = maiorValor;
    this.menorValor = menorValor;
    this.inversoDiferenca = 1D / (maiorValor - menorValor);

    // Se ambos forem iguais, faz com que todos os próximos resultados sejam
    // iguais a 0.5, supondo que todos serão iguais a menorValor
    if (maiorValor == menorValor){
        this.inversoDiferenca = 0.5;
        this.menorValor--;
    }
}

public double getMaiorValor() {
    return maiorValor;
}

public double getMenorValor() {
    return menorValor;
}
}

```

```

package modelo.np.rbc;

import java.util.ArrayList;
import java.sql.SQLException;
import modelo.rbc.CalculoSimilaridade;
import modelo.np.bd.ProxyBD;

/**
 * Sugere n prescrições a partir de critérios de similaridade pré-estabelecidos.
 *
 * @author Dalton
 */
public class SugestaoPrescricoes {

    /**
     * Retorna uma lista de sugestões de prescrição a partir de critérios de
     * similaridade pré-estabelecidos.
     *
     * @param diasVida dias de vida do RN
     * @param numParenterais número de parenterais do RN
     * @param idadeGestacional idade gestacional do RN
     * @param peso peso em Kg do RN
     * @param idRN ID do RN no BD, usado para evitar que o próprio RN seja usado
     * na comparação dos casos
     * @return uma lista com as sugestões de prescrição
     * @throws SQLException caso ocorra algum erro na consulta SQL
     * @throws ClassNotFoundException caso o driver MySQL não seja encontrado
     */
    public static final ArrayList<CasoPrescricao> sugiraPrescricoes(int diasVida,
        int numParenterais, int idadeGestacional, double peso, int idRN) throws
        SQLException, ClassNotFoundException{

        CasoPrescricao.inicieNormalizadores();

        // Cria o caso com o qual os outros casos serão comparados
        CasoPrescricao caso = new CasoPrescricao(peso, diasVida, idadeGestacional,
            numParenterais);

        // Obtém os casos que serão comparados com este
        ArrayList<CasoPrescricao> casos = ProxyBD.getInstance().getTodosCasosPrescricao(idRN);
        // Insere estes casos no ordenador
        CalculoSimilaridade calculo = new CalculoSimilaridade(caso);
        for (CasoPrescricao casoComparado : casos){
            calculo.addCaso(casoComparado);
        }
        // Obtém os 3 casos mais similares
        casos = calculo.getCasosMaisSimilares(3);
        // Adapta os casos
        new AdaptadorMedias().adapteCasos(casos);
        // Retorna as sugestões
        return casos;
    }
}

```

ANEXO II – MANUAL DO USUÁRIO

Este manual pressupõe que o usuário já cadastrou o RN e que sabe como acessar a tela de cálculo de NP.

Nutrição Parenteral - Cálculo

Paciente: RN de Teste 123

Data: 23/3/2006 Peso nasc (Kg): 1.25

Dias de vida: 0 Peso atual (Kg): 1.1

VHT (ml/Kg): 0.0 VIG (mg/Kg/min): 0.0 K (mEq/Kg): 0.0

VCT (Kcal/Kg): 0.0 Concentração: 0.0 Ca (ml/Kg): 0.0

SEM LEITE (dropdown) VC leite (cal/Kg): 0.0 Glicerofosfato (ml/Kg): 0.0

Leite (ml/vez): 0.0 VC NP (cal/Kg): 0.0 PO4 (mEq/Kg): 0.0

Vezes ao dia: 0.0 REL. CAL/PROT: 0.0 Mg (mEq/Kg): 0.0

Prot. leite (g/Kg): 0.0 Rel. Ca/P: 0.0 Zn (mcg/k): 0.0

Vol. NP (ml/Kg): 0.0 aa (g/Kg): 0.0 Vol. equipo (ml): 30.0

Vol. leite (ml/Kg): 0.0 Lip (g/Kg): 0.0 Fator correção: 0.0

Outros (ml): 0.0 Na (mEq/Kg): 0.0 Tx. Prot. Tot (g/Kg): 0.0

Vol. outros (ml/Kg): 0.0 Total Na (mEq/Kg): 0.0

Usar Heparina Usar glicerofosfato Iniciar Oligo Elem.

Vitamina: SEM VITAMINA (dropdown)

VOLTAR CALCULAR PESQUISAR CONFIRMAR

Figura A2.1 – Tela de cálculo da primeira prescrição de um RN

Ao acessar a tela de cálculo de NP, o botão PESQUISAR estará habilitado caso encontre pelo menos 70 prescrições iniciais no BD. Ao clicar no botão PESQUISAR, a seguinte tela será mostrada:

Escolha uma sugestão

Sugestão 1 Sugestão 2 Sugestão 3

Item	Valor Sugerido
aa (g/kg)	2.41
lip (g/kg)	0.83
Na (mEq/kg)	3.0
K (mEq/kg)	0.33
Ca (ml/kg)	3.0
Mg (mEq/kg)	0.25
Zn (mcg/kg)	266.66
PO4 (mEq/kg)	0.0
VIG (mg/kg/min)	5.33
VHT (ml/kg)	85.0

CANCELAR ACEITAR

Figura A2.2 – Tela de sugestões de prescrição

Note que são oferecidas 3 sugestões: a sugestão 1 é a média das três prescrições mais similares; a sugestão 2 é a média das duas prescrições mais similares; a sugestão 3 é a prescrição mais similar.

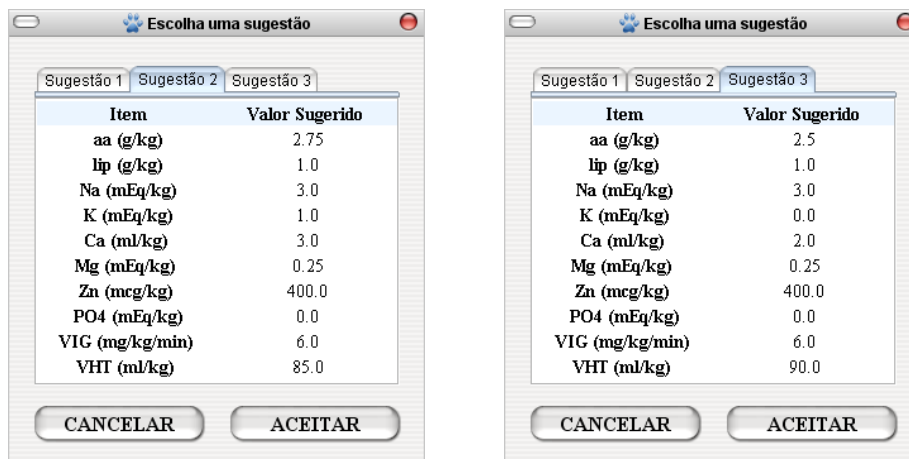


Figura A2.3 – Sugestões de prescrição (a primeira sugestão está na figura A2.2)

Caso nenhuma sugestão seja aceitável, o botão CANCELAR deve ser pressionado; caso alguma sugestão seja aceita, deve-se clicar na aba correspondente e em seguida pressionar o botão ACEITAR.

Se alguma sugestão for aceita e o botão ACEITAR for pressionado, a tela de cálculo de NP será assemelhará à tela mostrada a seguir:

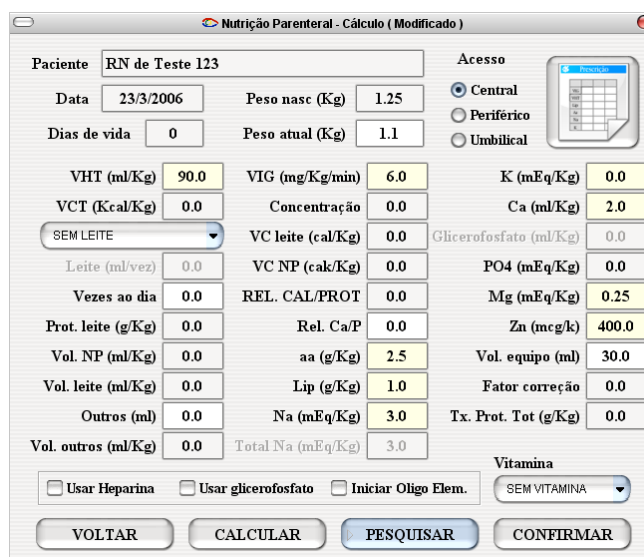


Figura A2.4 – Sugestão de prescrição aplicada à tela de cálculo

Em amarelo estão os campos cujos valores foram alterados de acordo com os valores da sugestão de prescrição. Esta alteração na cor dos campos também ocorre quando algum valor é alterado pelo usuário. Não há nenhum impedimento para alteração dos valores sugeridos para a prescrição.

Após esta etapa, o cálculo pode ser feito ao se pressionar o botão CALCULAR.

Nutrição Parenteral - Cálculo

Paciente	RN de Teste 123			Acesso	
Data	23/3/2006	Peso nasc (Kg)	1.25	<input checked="" type="radio"/> Central	
Dias de vida	0	Peso atual (Kg)	1.1	<input type="radio"/> Periférico	
				<input type="radio"/> Umbilical	
VHT (ml/Kg)	90.0	VIG (mg/Kg/min)	6.0	K (mEq/Kg)	0.0
VCT (Kcal/Kg)	49.559	Concentração	9.5999	Ca (ml/Kg)	2.0
SEM LEITE		VC leite (cal/Kg)	0.0	Glicerofosfato (ml/Kg)	0.0
Leite (ml/vez)	0.0	VC NP (cal/Kg)	49.559	PO4 (mEq/Kg)	0.0
Vezes ao dia	0.0	REL. CAL/PROT	105.49	Mg (mEq/Kg)	0.25
Prot. leite (g/Kg)	0.0	Rel. Ca/P	0.0	Zn (mcg/k)	400.0
Vol. NP (ml/Kg)	90.0	aa (g/Kg)	2.5	Vol. equipo (ml)	30.0
Vol. leite (ml/Kg)	0.0	Lip (g/Kg)	1.0	Fator correção	1.3030
Outros (ml)	0.0	Na (mEq/Kg)	3.0	Tx. Prot. Tot (g/Kg)	2.5
Vol. outros (ml/Kg)	0.0	Total Na (mEq/Kg)	3.0		
<input type="checkbox"/> Usar Heparina <input type="checkbox"/> Usar glicerofosfato <input type="checkbox"/> Iniciar Oligo Elem.			Vitamina		SEM VITAMINA
<input type="button" value="VOLTAR"/> <input type="button" value="CALCULAR"/> <input type="button" value="PESQUISAR"/> <input type="button" value="CONFIRMAR"/>					

Figura A2.4 – Tela de cálculo de NP depois que o cálculo é efetuado

Mesmo que existam muitos casos disponíveis e que a máquina de inferência retorne sugestões muito confiáveis, não há nada que possa ser feito para sugerir uma prescrição com segurança caso os dados cadastrais do RN estejam errados, pois a máquina sugerirá prescrições segundo o que foi cadastrado e não segundo a situação real do RN. No mínimo a data de nascimento e o peso atual do RN devem ser cadastrados corretamente.

ANEXO III – TESTE RETROSPECTIVO I

Teste realizado com a base de dados de 1º de Novembro de 2005 do Hospital Infantil Joana de Gusmão. Considere *peso* em Kg e *idade gestacional* pelo critério D.U.M.

RN Teste nº 1

Peso	Dias de Vida	Idade Gestacional
0.77	1	26

Prescrição Original

aa	VIG	VHT	Lipídios	Na	Zn	K	Ca	Mg	PO4
3.00	5.00	80.00	0.00	3.00	400.00	0.00	2.00	0.25	0.00

Prescrições Sugeridas

Médias	aa	VIG	VHT	Lipídios	Na	Zn	K	Ca	Mg	PO4
3	3.00	6.33	115.00	1.00	3.00	400.00	0.67	2.00	0.17	0.00
2	3.00	6.00	90.00	1.00	3.00	400.00	0.00	2.00	0.25	0.00
1	3.00	6.00	80.00	1.00	3.00	400.00	0.00	2.00	0.25	0.00

RN Teste nº 2

Peso	Dias de Vida	Idade Gestacional
2.5	2	35

Prescrição Original

aa	VIG	VHT	Lipídios	Na	Zn	K	Ca	Mg	PO4
1.00	5.00	80.00	1.00	3.00	400.00	0.00	2.00	0.25	0.00

Prescrições Sugeridas

Médias	aa	VIG	VHT	Lipídios	Na	Zn	K	Ca	Mg	PO4
3	2.00	5.33	83.33	1.67	3.00	400.00	1.33	2.67	0.17	0.00
2	2.00	5.00	85.00	1.50	3.00	400.00	2.00	2.00	0.25	0.00
1	2.00	4.00	80.00	1.00	3.00	400.00	2.00	2.00	0.25	0.00

RN Teste nº 3

Peso	Dias de Vida	Idade Gestacional
2.81	2	35

Prescrição Original

aa	VIG	VHT	Lipídios	Na	Zn	K	Ca	Mg	PO4
2.00	5.00	70.00	1.00	3.00	400.00	0.00	2.00	25.00	0.69

Prescrições Sugeridas

Médias	aa	VIG	VHT	Lipídios	Na	Zn	K	Ca	Mg	PO4
3	1.33	5.00	73.33	0.17	3.00	266.67	0.00	2.00	16.67	0.23
2	1.00	5.00	70.00	0.00	3.00	400.00	0.00	2.00	25.00	0.35
1	1.00	5.00	70.00	0.00	3.00	400.00	0.00	2.00	25.00	0.69

RN Teste nº 4

Peso	Dias de Vida	Idade Gestacional
1.18	3	29

Prescrição Original

aa	VIG	VHT	Lipídios	Na	Zn	K	Ca	Mg	PO4
2.50	6.00	90.00	1.00	3.00	400.00	0.00	2.00	0.25	0.00

Prescrições Sugeridas

Médias	aa	VIG	VHT	Lipídios	Na	Zn	K	Ca	Mg	PO4
3	3.00	6.67	96.67	2.00	3.00	400.00	2.00	2.67	0.25	0.00
2	3.00	7.00	105.00	2.50	3.00	400.00	2.00	2.00	0.25	0.00
1	3.00	7.00	100.00	2.00	3.00	400.00	2.00	2.00	0.25	0.00

RN Teste nº 5

Peso	Dias de Vida	Idade Gestacional
1.935	4	33

Prescrição Original

aa	VIG	VHT	Lipídios	Na	Zn	K	Ca	Mg	PO4
1.00	5.00	90.00	1.00	3.00	400.00	2.00	2.00	0.25	0.00

Prescrições Sugeridas

Médias	aa	VIG	VHT	Lipídios	Na	Zn	K	Ca	Mg	PO4
3	2.33	7.67	110.00	2.33	3.00	400.00	2.00	2.00	0.25	0.00

2	2.50	8.50	105.00	2.50	3.00	400.00	2.00	2.00	0.25	0.00
1	2.00	8.00	100.00	2.00	3.00	400.00	2.00	2.00	0.25	0.00

RN Teste nº 6

Peso	Dias de Vida	Idade Gestacional
3.33	3	36

Prescrição Original

aa	VIG	VHT	Lipídios	Na	Zn	K	Ca	Mg	PO4
1.00	6.00	80.00	1.00	3.00	400.00	2.00	2.00	0.25	0.00

Prescrições Sugeridas

Médias	aa	VIG	VHT	Lipídios	Na	Zn	K	Ca	Mg	PO4
3	2.33	7.67	93.33	2.33	3.00	400.00	2.00	2.00	0.25	0.00
2	2.00	7.00	90.00	2.00	3.00	400.00	2.00	2.00	0.25	0.00
1	2.00	7.00	90.00	2.00	3.00	400.00	2.00	2.00	0.25	0.00

RN Teste nº 7

Peso	Dias de Vida	Idade Gestacional
1.73	5	35

Prescrição Original

aa	VIG	VHT	Lipídios	Na	Zn	K	Ca	Mg	PO4
2.00	8.00	100.00	2.00	3.00	400.00	2.00	2.00	0.25	0.00

Prescrições Sugeridas

Médias	aa	VIG	VHT	Lipídios	Na	Zn	K	Ca	Mg	PO4
3	2.00	6.67	106.67	2.00	3.00	400.00	2.00	2.00	0.25	0.00
2	2.50	7.50	115.00	2.50	3.00	400.00	2.00	2.00	0.25	0.00
1	3.00	9.00	110.00	3.00	3.00	400.00	2.00	2.00	0.25	0.00

RN Teste nº 8

Peso	Dias de Vida	Idade Gestacional
0.88	3	29

Prescrição Original

aa	VIG	VHT	Lipídios	Na	Zn	K	Ca	Mg	PO4
3.00	6.00	80.00	1.00	3.00	400.00	2.00	4.00	0.25	0.00

Prescrições Sugeridas

Médias	aa	VIG	VHT	Lipídios	Na	Zn	K	Ca	Mg	PO4
3	3.00	6.33	96.67	1.50	3.00	400.00	2.00	3.33	0.25	0.00
2	3.00	6.00	95.00	1.25	3.00	400.00	2.00	4.00	0.25	0.00
1	3.00	6.00	90.00	1.00	3.00	400.00	2.00	4.00	0.25	0.00

RN Teste nº 9

Peso	Dias de Vida	Idade Gestacional
2.93	4	40

Prescrição Original

aa	VIG	VHT	Lipídios	Na	Zn	K	Ca	Mg	PO4
1.00	6.00	80.00	1.00	4.00	400.00	2.00	2.00	0.25	0.00

Prescrições Sugeridas

Médias	aa	VIG	VHT	Lipídios	Na	Zn	K	Ca	Mg	PO4
3	2.00	6.00	90.00	1.67	3.67	266.67	2.00	2.33	0.25	0.00
2	2.00	7.00	90.00	2.00	4.00	400.00	2.00	2.00	0.25	0.00
1	2.00	7.00	90.00	2.00	4.00	400.00	2.00	2.00	0.25	0.00

RN Teste nº 10

Peso	Dias de Vida	Idade Gestacional
0.7	7	27

Prescrição Original

aa	VIG	VHT	Lipídios	Na	Zn	K	Ca	Mg	PO4
3.00	7.00	165.00	1.00	3.00	400.00	2.00	2.00	0.00	0.00

Prescrições Sugeridas

Médias	aa	VIG	VHT	Lipídios	Na	Zn	K	Ca	Mg	PO4
3	3.00	6.00	106.67	1.00	3.00	400.00	0.67	2.00	0.17	0.00
2	3.00	6.00	120.00	1.00	3.00	400.00	1.00	2.00	0.13	0.00
1	3.00	7.00	160.00	2.00	3.00	400.00	2.00	2.00	0.00	0.00

RN Teste nº 11

Peso	Dias de Vida	Idade Gestacional
3.9	2	40

Prescrição Original

aa	VIG	VHT	Lipídios	Na	Zn	K	Ca	Mg	PO4
1.00	5.00	70.00	1.00	3.00	400.00	0.00	2.00	0.25	0.00

Prescrições Sugeridas

Médias	aa	VIG	VHT	Lipídios	Na	Zn	K	Ca	Mg	PO4
3	2.33	6.33	100.00	2.00	3.67	400.00	2.33	3.67	0.25	0.00
2	2.00	5.50	95.00	1.50	4.00	400.00	2.00	4.00	0.25	0.00
1	2.00	5.00	90.00	1.00	4.00	400.00	2.00	4.00	0.25	0.00

RN Teste n° 12

Peso	Dias de Vida	Idade Gestacional
1.36	2	32

Prescrição Original

aa	VIG	VHT	Lipídios	Na	Zn	K	Ca	Mg	PO4
2.00	6.00	80.00	0.00	3.00	400.00	0.00	6.00	0.25	0.00

Prescrições Sugeridas

Médias	aa	VIG	VHT	Lipídios	Na	Zn	K	Ca	Mg	PO4
3	2.33	4.50	80.00	0.83	2.00	400.00	0.67	4.00	0.25	0.00
2	2.00	4.25	80.00	0.75	1.50	400.00	1.00	4.00	0.25	0.00
1	2.00	4.50	90.00	1.00	3.00	400.00	2.00	4.00	0.25	0.00

RN Teste n° 13

Peso	Dias de Vida	Idade Gestacional
3.735	1	33

Prescrição Original

aa	VIG	VHT	Lipídios	Na	Zn	K	Ca	Mg	PO4
2.00	5.00	90.00	1.00	5.00	400.00	2.00	2.00	0.25	0.69

Prescrições Sugeridas

Médias	aa	VIG	VHT	Lipídios	Na	Zn	K	Ca	Mg	PO4
3	2.33	5.67	126.67	2.00	4.67	400.00	2.00	2.00	0.25	0.23
2	2.50	6.00	115.00	2.00	5.00	400.00	2.00	2.00	0.25	0.35
1	2.00	5.00	110.00	1.00	5.00	400.00	2.00	2.00	0.25	0.69

RN Teste nº 14

Peso	Dias de Vida	Idade Gestacional
1.0	29	27

Prescrição Original

aa	VIG	VHT	Lipídios	Na	Zn	K	Ca	Mg	PO4
2.00	10.00	120.00	1.00	6.00	400.00	3.00	2.00	0.25	0.00

Prescrições Sugeridas

Médias	aa	VIG	VHT	Lipídios	Na	Zn	K	Ca	Mg	PO4
3	2.67	9.00	120.00	1.33	6.67	400.00	3.67	2.00	0.25	0.00
2	2.50	10.00	130.00	1.00	7.00	400.00	4.00	2.00	0.25	0.00
1	2.00	10.00	130.00	1.00	7.00	400.00	4.00	2.00	0.25	0.00

RN Teste nº 15

Peso	Dias de Vida	Idade Gestacional
3.5	4	39

Prescrição Original

aa	VIG	VHT	Lipídios	Na	Zn	K	Ca	Mg	PO4
1.00	5.00	80.00	1.00	3.00	400.00	2.00	2.00	0.25	0.00

Prescrições Sugeridas

Médias	aa	VIG	VHT	Lipídios	Na	Zn	K	Ca	Mg	PO4
3	2.33	6.00	86.67	2.00	3.33	400.00	2.00	2.00	0.25	0.00
2	2.50	7.00	95.00	2.50	3.00	400.00	2.00	2.00	0.25	0.00
1	2.00	6.00	90.00	2.00	3.00	400.00	2.00	2.00	0.25	0.00

RN Teste nº 16

Peso	Dias de Vida	Idade Gestacional
1.04	2	27

Prescrição Original

aa	VIG	VHT	Lipídios	Na	Zn	K	Ca	Mg	PO4
2.00	4.00	80.00	0.50	3.00	0.00	0.00	4.00	0.25	0.00

Prescrições Sugeridas

Médias	aa	VIG	VHT	Lipídios	Na	Zn	K	Ca	Mg	PO4
3	3.00	6.00	100.00	1.83	3.00	0.00	0.67	3.33	0.25	0.00

2	3.00	5.50	95.00	1.50	3.00	0.00	0.00	4.00	0.25	0.00
1	3.00	5.00	90.00	1.00	3.00	0.00	0.00	4.00	0.25	0.00

RN Teste nº 17

Peso	Dias de Vida	Idade Gestacional
0.62	3	30

Prescrição Original

aa	VIG	VHT	Lipídios	Na	Zn	K	Ca	Mg	PO4
2.00	7.00	80.00	0.00	4.00	0.00	1.00	2.00	0.25	0.00

Prescrições Sugeridas

Médias	aa	VIG	VHT	Lipídios	Na	Zn	K	Ca	Mg	PO4
3	3.00	7.67	103.33	1.83	5.00	0.00	2.33	2.00	0.25	0.00
2	3.00	7.50	95.00	1.25	4.50	0.00	2.50	2.00	0.25	0.00
1	3.00	7.00	90.00	0.50	4.00	0.00	2.00	2.00	0.25	0.00

RN Teste nº 18

Peso	Dias de Vida	Idade Gestacional
2.64	4	36

Prescrição Original

aa	VIG	VHT	Lipídios	Na	Zn	K	Ca	Mg	PO4
2.00	6.00	80.00	2.00	3.00	400.00	0.00	4.00	0.00	0.00

Prescrições Sugeridas

Médias	aa	VIG	VHT	Lipídios	Na	Zn	K	Ca	Mg	PO4
3	1.67	4.67	80.00	0.83	3.00	266.67	0.67	2.00	0.17	0.00
2	1.50	5.00	80.00	0.75	3.00	200.00	0.00	2.00	0.13	0.00
1	2.00	5.00	80.00	0.50	3.00	0.00	0.00	2.00	0.00	0.00

RN Teste nº 19

Peso	Dias de Vida	Idade Gestacional
1.37	73	29

Prescrição Original

aa	VIG	VHT	Lipídios	Na	Zn	K	Ca	Mg	PO4
3.00	9.50	150.00	3.00	3.00	400.00	1.00	1.00	0.25	0.00

Prescrições Sugeridas

Médias	aa	VIG	VHT	Lipídios	Na	Zn	K	Ca	Mg	PO4
3	3.00	6.67	126.67	3.00	5.00	400.00	2.00	2.00	0.25	0.00
2	3.00	7.00	125.00	3.00	5.00	400.00	2.00	2.00	0.25	0.00
1	3.00	6.00	120.00	3.00	5.00	400.00	2.00	2.00	0.25	0.00

RN Teste n° 20

Peso	Dias de Vida	Idade Gestacional
1.985	3	37

Prescrição Original

aa	VIG	VHT	Lipídios	Na	Zn	K	Ca	Mg	PO4
2.00	6.00	80.00	1.00	3.00	400.00	1.00	2.00	0.25	0.00

Prescrições Sugeridas

Médias	aa	VIG	VHT	Lipídios	Na	Zn	K	Ca	Mg	PO4
3	3.00	7.33	100.00	2.33	3.00	400.00	2.00	2.00	0.25	0.00
2	3.00	7.00	95.00	2.00	3.00	400.00	2.00	2.00	0.25	0.00
1	3.00	8.00	90.00	2.00	3.00	400.00	2.00	2.00	0.25	0.00

RN Teste n° 21

Peso	Dias de Vida	Idade Gestacional
3.3	6	40

Prescrição Original

aa	VIG	VHT	Lipídios	Na	Zn	K	Ca	Mg	PO4
2.00	4.00	50.00	1.00	4.00	0.00	2.00	2.00	0.00	0.00

Prescrições Sugeridas

Médias	aa	VIG	VHT	Lipídios	Na	Zn	K	Ca	Mg	PO4
3	2.00	4.67	80.00	1.33	4.33	350.00	2.00	2.00	0.17	0.00
2	2.00	5.00	80.00	1.50	4.50	325.00	2.00	2.00	0.13	0.00
1	2.00	6.00	90.00	2.00	5.00	250.00	2.00	2.00	0.00	0.00

RN Teste n° 22

Peso	Dias de Vida	Idade Gestacional
3.0	2	37

Prescrição Original

aa	VIG	VHT	Lipídios	Na	Zn	K	Ca	Mg	PO4
2.00	4.00	70.00	1.00	3.00	0.00	0.00	2.00	0.25	0.00

Prescrições Sugeridas

Médias	aa	VIG	VHT	Lipídios	Na	Zn	K	Ca	Mg	PO4
3	3.00	5.33	88.33	1.33	3.67	400.00	2.00	3.33	0.25	0.00
2	3.00	5.50	85.00	1.50	3.50	400.00	2.00	3.00	0.25	0.00
1	3.00	5.00	90.00	1.00	4.00	400.00	2.00	4.00	0.25	0.00

RN Teste n° 23

Peso	Dias de Vida	Idade Gestacional
1.33	2	31

Prescrição Original

aa	VIG	VHT	Lipídios	Na	Zn	K	Ca	Mg	PO4
2.00	4.00	70.00	0.50	0.00	400.00	0.00	4.00	0.25	0.00

Prescrições Sugeridas

Médias	aa	VIG	VHT	Lipídios	Na	Zn	K	Ca	Mg	PO4
3	2.67	5.67	83.33	1.00	3.00	400.00	0.67	4.67	0.25	0.00
2	3.00	5.50	85.00	1.50	3.00	400.00	1.00	4.00	0.25	0.00
1	3.00	5.00	80.00	1.00	3.00	400.00	0.00	4.00	0.25	0.00

RN Teste n° 24

Peso	Dias de Vida	Idade Gestacional
3.0	4	37

Prescrição Original

aa	VIG	VHT	Lipídios	Na	Zn	K	Ca	Mg	PO4
3.00	5.00	90.00	1.00	4.00	400.00	2.00	4.00	0.25	0.00

Prescrições Sugeridas

Médias	aa	VIG	VHT	Lipídios	Na	Zn	K	Ca	Mg	PO4
3	2.67	5.00	81.67	1.33	3.33	266.67	1.33	2.67	0.25	0.00
2	2.50	5.00	75.00	1.50	3.00	200.00	1.00	2.00	0.25	0.00
1	2.00	4.00	70.00	1.00	3.00	0.00	0.00	2.00	0.25	0.00

ANEXO IV – TESTE RETROSPECTIVO II

Teste realizado com a base de dados de 27 de Março de 2006 do Hospital Infantil Joana de Gusmão. Considere *peso* em Kg e *idade gestacional* pelo critério D.U.M. Considere que as prescrições sugeridas estão em ordem decrescente de médias.

Caso #1

RN de entrada	Dias de Vida	Idade Gestacional	Dias de Parenteral	Peso
	2	29	0	1.1

Prescrições sugeridas	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	6.0	3.0	1.0	3.0	400.0	0.0	4.0	0.25	0.0
	7.0	2.0	0.0	4.0	0.0	0.0	2.0	0.25	0.0
Prescrição original	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	6.0	2.5	1.0	3.0	400.0	0.0	2.0	0.25	0.0

Caso #2

RN de entrada	Dias de Vida	Idade Gestacional	Dias de Parenteral	Peso
	3	33	0	1.935

Prescrições sugeridas	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	7.0	2.0	1.0	2.0	0.0	2.0	2.0	0.25	0.691176
	8.0	2.0	2.0	3.0	400.0	2.0	2.0	0.25	0.0
Prescrição original	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	6.0	2.0	0.0	3.0	400.0	2.0	6.0	0.25	0.0
Prescrição original	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	5.0	1.0	1.0	3.0	400.0	2.0	2.0	0.25	0.0

Caso #3

RN de entrada	Dias de Vida	Idade Gestacional	Dias de Parenteral	Peso
	1	35	0	2.5

Prescrições sugeridas	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	5.0	2.0	1.0	3.0	400.0	0.0	2.0	25.0	0.691176
	5.0	2.0	0.5	3.0	0.0	0.0	2.0	0.0	0.0
Prescrição original	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	6.0	1.0	1.0	3.0	400.0	0.0	2.0	0.25	0.0
Prescrição original	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	5.0	1.0	1.0	3.0	400.0	0.0	2.0	0.25	0.0

Caso #4

RN de entrada	Dias de Vida	Idade Gestacional	Dias de Parenteral	Peso
	2	36	0	3.5

Prescrições sugeridas	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	6.0	2.0	1.0	4.0	0.0	2.0	4.0	0.25	0.0
	5.0	2.0	0.5	3.0	0.0	2.0	2.0	0.0	0.0
	6.0	2.0	2.0	3.0	400.0	2.0	4.0	0.0	0.0
Prescrição original	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	6.0	1.0	1.0	3.0	400.0	2.0	2.0	0.25	0.0

Caso #5

RN de entrada	Dias de Vida	Idade Gestacional	Dias de Parenteral	Peso
	1	35	0	2.81

Prescrições sugeridas	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	5.0	1.0	1.0	3.0	400.0	0.0	2.0	0.25	0.0
	5.0	2.0	0.5	3.0	0.0	0.0	2.0	0.0	0.0
	6.0	1.0	1.0	3.0	400.0	0.0	2.0	0.25	0.0
Prescrição original	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	5.0	2.0	1.0	3.0	400.0	0.0	2.0	25.0	0.691176

Caso #6

RN de entrada	Dias de Vida	Idade Gestacional	Dias de Parenteral	Peso
	0	26	0	0.63

Prescrições sugeridas	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	4.0	2.0	0.5	3.0	0.0	0.0	4.0	0.25	0.0
	6.0	3.0	1.0	3.0	400.0	0.0	4.0	0.25	0.0
	6.0	2.5	1.0	3.0	400.0	0.0	2.0	0.25	0.0
Prescrição original	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	5.0	3.0	0.0	3.0	400.0	0.0	2.0	0.25	0.0

Caso #7

RN de entrada	Dias de Vida	Idade Gestacional	Dias de Parenteral	Peso
	4	35	0	1.76

Prescrições sugeridas	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	6.0	2.0	2.0	3.0	400.0	2.0	4.0	0.0	0.0
	5.0	1.0	1.0	3.0	400.0	2.0	2.0	0.25	0.0
	5.0	3.0	1.0	4.0	400.0	2.0	4.0	0.25	0.0
Prescrição original	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	8.0	2.0	2.0	3.0	400.0	2.0	2.0	0.25	0.0

Caso #8

RN de entrada	Dias de Vida	Idade Gestacional	Dias de Parenteral	Peso
	2	29	0	0.86

Prescrições sugeridas	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	6.0	2.5	1.0	3.0	400.0	2.0	2.0	0.25	0.0
	7.0	2.0	0.0	4.0	0.0	2.0	2.0	0.25	0.0
Prescrição original	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	6.0	3.0	1.0	3.0	400.0	2.0	4.0	0.25	0.0

Caso #9

RN de entrada	Dias de Vida	Idade Gestacional	Dias de Parenteral	Peso
	3	40	0	2.97

Prescrições sugeridas	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	6.0	2.0	1.0	5.0	0.0	2.0	4.0	0.25	0.0
	5.0	1.0	1.0	3.0	400.0	2.0	2.0	0.25	0.0
Prescrição original	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	6.0	1.0	1.0	4.0	400.0	2.0	2.0	0.25	0.0

Caso #10

RN de entrada	Dias de Vida	Idade Gestacional	Dias de Parenteral	Peso
	6	27	0	0.73

Prescrições sugeridas	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	6.0	3.0	1.0	3.0	400.0	2.0	4.0	0.25	0.0
	6.0	2.5	1.0	3.0	400.0	2.0	2.0	0.25	0.0
Prescrição original	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	7.0	3.0	1.0	3.0	400.0	2.0	2.0	0.0	0.0

Caso #11

RN de entrada	Dias de Vida	Idade Gestacional	Dias de Parenteral	Peso
	1	40	0	3.77

Prescrições sugeridas	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	6.0	2.0	1.0	5.0	0.0	0.0	4.0	0.25	0.0
	6.0	2.0	1.0	4.0	0.0	0.0	2.0	0.25	0.0
Prescrição original	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	5.0	1.0	1.0	3.0	400.0	0.0	2.0	0.25	0.0

Caso #12

RN de entrada	Dias de Vida	Idade Gestacional	Dias de Parenteral	Peso
	1	32	0	1.36

Prescrições sugeridas	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	4.0	2.0	0.5	0.0	400.0	0.0	4.0	0.25	0.0
	7.0	2.0	1.0	2.0	0.0	0.0	2.0	0.25	0.691176
	5.0	2.0	1.0	5.0	400.0	0.0	2.0	0.25	0.691176
Prescrição original	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	6.0	2.0	0.0	3.0	400.0	0.0	6.0	0.25	0.0

Caso #13

RN de entrada	Dias de Vida	Idade Gestacional	Dias de Parenteral	Peso
	0	33	0	2.1

Prescrições sugeridas	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	6.0	2.0	0.0	3.0	400.0	2.0	6.0	0.25	0.0
	7.0	2.0	1.0	2.0	0.0	2.0	2.0	0.25	0.691176
	5.0	1.0	1.0	3.0	400.0	2.0	2.0	0.25	0.0
Prescrição original	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	5.0	2.0	1.0	5.0	400.0	2.0	2.0	0.25	0.691176

Caso #14

RN de entrada	Dias de Vida	Idade Gestacional	Dias de Parenteral	Peso
	28	27	0	1.3

Prescrições sugeridas	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	7.0	3.0	1.0	3.0	400.0	3.0	2.0	0.0	0.0
	6.0	2.0	1.0	4.0	0.0	3.0	2.0	0.25	0.0
	8.0	2.0	2.0	3.0	400.0	3.0	2.0	0.25	0.0
Prescrição original	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	10.0	2.0	1.0	6.0	400.0	3.0	2.0	0.25	0.0

Caso #15

RN de entrada	Dias de Vida	Idade Gestacional	Dias de Parenteral	Peso
	3	39	0	2.8

Prescrições sugeridas	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	6.0	1.0	1.0	4.0	400.0	2.0	2.0	0.25	0.0
	4.0	2.0	1.0	3.0	0.0	2.0	2.0	0.25	0.0
	6.0	2.0	1.0	5.0	0.0	2.0	4.0	0.25	0.0
Prescrição original	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	5.0	1.0	1.0	3.0	400.0	2.0	2.0	0.25	0.0

Caso #16

RN de entrada	Dias de Vida	Idade Gestacional	Dias de Parenteral	Peso
	5	40	0	2.15

Prescrições sugeridas	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	4.0	2.0	1.0	4.0	0.0	2.0	2.0	0.0	0.0
	6.0	2.0	2.0	5.0	250.0	2.0	2.0	0.0	0.0
	4.0	2.0	1.0	4.0	400.0	2.0	2.0	0.25	0.0
Prescrição original	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	4.0	2.0	1.0	3.0	0.0	2.0	3.0	0.25	0.0

Caso #17

RN de entrada	Dias de Vida	Idade Gestacional	Dias de Parenteral	Peso
	1	27	0	0.9

Prescrições sugeridas	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	5.0	3.0	0.0	3.0	400.0	0.0	2.0	0.25	0.0
	6.0	3.0	1.0	3.0	400.0	0.0	4.0	0.25	0.0
	6.0	2.5	1.0	3.0	400.0	0.0	2.0	0.25	0.0
Prescrição original	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	4.0	2.0	0.5	3.0	0.0	0.0	4.0	0.25	0.0

Caso #18

RN de entrada	Dias de Vida	Idade Gestacional	Dias de Parenteral	Peso
	2	30	0	0.59

Prescrições sugeridas	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	6.0	3.0	1.0	3.0	400.0	1.0	4.0	0.25	0.0
	6.0	2.5	1.0	3.0	400.0	1.0	2.0	0.25	0.0
	4.0	2.0	0.5	0.0	400.0	1.0	4.0	0.25	0.0
Prescrição original	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	7.0	2.0	0.0	4.0	0.0	1.0	2.0	0.25	0.0

Caso #19

RN de entrada	Dias de Vida	Idade Gestacional	Dias de Parenteral	Peso
	5	40	0	3.37

Prescrições sugeridas	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	4.0	2.0	1.0	4.0	400.0	2.0	2.0	0.25	0.0
	4.0	2.0	1.0	4.0	0.0	2.0	2.0	0.0	0.0
	4.0	2.0	1.0	3.0	0.0	2.0	3.0	0.25	0.0
Prescrição original	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	6.0	2.0	2.0	5.0	250.0	2.0	2.0	0.0	0.0

Caso #20

RN de entrada	Dias de Vida	Idade Gestacional	Dias de Parenteral	Peso
	3	36	0	2.64

Prescrições sugeridas	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	5.0	3.0	1.0	4.0	400.0	0.0	4.0	0.25	0.0
	6.0	1.0	1.0	3.0	400.0	0.0	2.0	0.25	0.0
	6.0	2.0	1.0	4.0	0.0	0.0	4.0	0.25	0.0
Prescrição original	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	6.0	2.0	2.0	3.0	400.0	0.0	4.0	0.0	0.0

Caso #21

RN de entrada	Dias de Vida	Idade Gestacional	Dias de Parenteral	Peso
	72	29	0	2.8

Prescrições sugeridas	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	10.0	2.0	1.0	6.0	400.0	1.0	2.0	0.25	0.0
	6.0	2.0	1.0	4.0	0.0	1.0	2.0	0.25	0.0
	7.0	3.0	1.0	3.0	400.0	1.0	2.0	0.0	0.0
Prescrição original	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	9.5	3.0	3.0	3.0	400.0	1.0	1.0	0.25	0.0

Caso #22

RN de entrada	Dias de Vida	Idade Gestacional	Dias de Parenteral	Peso
	5	40	0	3.4

Prescrições sugeridas	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	6.0	2.0	2.0	5.0	250.0	2.0	2.0	0.0	0.0
	4.0	2.0	1.0	4.0	0.0	2.0	2.0	0.0	0.0
	4.0	2.0	1.0	3.0	0.0	2.0	3.0	0.25	0.0
Prescrição original	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	4.0	2.0	1.0	4.0	400.0	2.0	2.0	0.25	0.0

Caso #23

RN de entrada	Dias de Vida	Idade Gestacional	Dias de Parenteral	Peso
	2	37	0	2.13

Prescrições sugeridas	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	6.0	2.0	1.0	4.0	0.0	1.0	4.0	0.25	0.0
	4.0	2.0	1.0	3.0	0.0	1.0	2.0	0.25	0.0
	6.0	2.0	1.0	3.0	0.0	1.0	2.0	0.25	0.0
Prescrição original	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	6.0	2.0	1.0	3.0	400.0	1.0	2.0	0.25	0.0

Caso #24

RN de entrada	Dias de Vida	Idade Gestacional	Dias de Parenteral	Peso
	1	36	0	2.8

Prescrições sugeridas	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	5.0	2.0	1.0	3.0	400.0	0.0	2.0	25.0	0.691176
	4.0	2.0	1.0	3.0	0.0	0.0	2.0	0.25	0.0
Prescrição original	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	5.0	2.0	0.5	3.0	0.0	0.0	2.0	0.0	0.0

Caso #25

RN de entrada	Dias de Vida	Idade Gestacional	Dias de Parenteral	Peso
	5	40	0	3.3

Prescrições sugeridas	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	6.0	2.0	2.0	5.0	250.0	2.0	2.0	0.0	0.0
	4.0	2.0	1.0	4.0	400.0	2.0	2.0	0.25	0.0
Prescrição original	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	4.0	2.0	1.0	4.0	0.0	2.0	2.0	0.0	0.0

Caso #26

RN de entrada	Dias de Vida	Idade Gestacional	Dias de Parenteral	Peso
	1	37	0	2.54

Prescrições sugeridas	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	5.0	2.0	0.5	3.0	0.0	0.0	2.0	0.0	0.0
	6.0	2.0	1.0	3.0	400.0	0.0	2.0	0.25	0.0
Prescrição original	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	4.0	2.0	1.0	3.0	0.0	0.0	2.0	0.25	0.0

Caso #27

RN de entrada	Dias de Vida	Idade Gestacional	Dias de Parenteral	Peso
	1	31	0	1.33

Prescrições sugeridas	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	6.0	2.0	0.0	3.0	400.0	0.0	6.0	0.25	0.0
	7.0	2.0	1.0	2.0	0.0	0.0	2.0	0.25	0.691176
Prescrição original	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	4.0	2.0	0.5	0.0	400.0	0.0	4.0	0.25	0.0

Caso #28

RN de entrada	Dias de Vida	Idade Gestacional	Dias de Parenteral	Peso
	3	37	0	3.0

Prescrições sugeridas	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	6.0	2.0	1.0	4.0	0.0	2.0	4.0	0.25	0.0
	6.0	2.0	2.0	3.0	400.0	2.0	4.0	0.0	0.0
	4.0	2.0	1.0	3.0	0.0	2.0	2.0	0.25	0.0
Prescrição original	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	5.0	3.0	1.0	4.0	400.0	2.0	4.0	0.25	0.0

Caso #29

RN de entrada	Dias de Vida	Idade Gestacional	Dias de Parenteral	Peso
	2	32	0	2.0

Prescrições sugeridas	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	6.0	2.0	0.0	3.0	400.0	2.0	6.0	0.25	0.0
	5.0	1.0	1.0	3.0	400.0	2.0	2.0	0.25	0.0
	4.0	2.0	0.5	0.0	400.0	2.0	4.0	0.25	0.0
Prescrição original	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	7.0	2.0	1.0	2.0	0.0	2.0	2.0	0.25	0.691176

Caso #30

RN de entrada	Dias de Vida	Idade Gestacional	Dias de Parenteral	Peso
	2	38	0	2.86

Prescrições sugeridas	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	6.0	2.0	1.0	4.0	0.0	0.0	4.0	0.25	0.0
	4.0	2.0	1.0	3.0	0.0	0.0	2.0	0.25	0.0
	6.0	2.0	1.0	3.0	400.0	0.0	2.0	0.25	0.0
Prescrição original	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	6.0	2.0	1.0	3.0	0.0	0.0	2.0	0.25	0.0

Caso #31

RN de entrada	Dias de Vida	Idade Gestacional	Dias de Parenteral	Peso
	2	37	0	3.13

Prescrições sugeridas	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	6.0	2.0	1.0	3.0	400.0	2.0	2.0	0.25	0.0
	5.0	3.0	1.0	4.0	400.0	2.0	4.0	0.25	0.0
	6.0	2.0	1.0	3.0	0.0	2.0	2.0	0.25	0.0
Prescrição original	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	6.0	2.0	1.0	4.0	0.0	2.0	4.0	0.25	0.0

Caso #32

RN de entrada	Dias de Vida	Idade Gestacional	Dias de Parenteral	Peso
	2	40	0	2.83

Prescrições sugeridas	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	6.0	1.0	1.0	4.0	400.0	0.0	2.0	0.25	0.0
	5.0	1.0	1.0	3.0	400.0	0.0	2.0	0.25	0.0
Prescrição original	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	6.0	2.0	1.0	5.0	0.0	0.0	4.0	0.25	0.0

Caso #33

RN de entrada	Dias de Vida	Idade Gestacional	Dias de Parenteral	Peso
	2	39	0	4.0

Prescrições sugeridas	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	5.0	1.0	1.0	3.0	400.0	1.0	2.0	0.25	0.0
	4.0	2.0	1.0	3.0	0.0	1.0	2.0	0.25	0.0
Prescrição original	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	6.0	2.0	1.0	4.0	0.0	1.0	2.0	0.25	0.0

Caso #34

RN de entrada	Dias de Vida	Idade Gestacional	Dias de Parenteral	Peso
	7	40	0	3.35

Prescrições sugeridas	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	6.0	2.0	2.0	5.0	250.0	3.0	2.0	0.0	0.0
	4.0	2.0	1.0	4.0	0.0	3.0	2.0	0.0	0.0
Prescrição original	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	6.0	2.0	1.0	4.0	0.0	3.0	2.0	0.25	0.0

Caso #35

RN de entrada	Dias de Vida	Idade Gestacional	Dias de Parenteral	Peso
	3	38	0	3.55

Prescrições sugeridas	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	6.0	1.0	0.5	4.0	0.0	3.0	2.0	0.25	0.0
	5.0	3.0	1.0	4.0	400.0	3.0	4.0	0.25	0.0
Prescrição original	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	4.0	2.0	1.0	3.0	0.0	3.0	2.0	0.25	0.0

Caso #36

RN de entrada	Dias de Vida	Idade Gestacional	Dias de Parenteral	Peso
	4	38	0	3.025

Prescrições sugeridas	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	4.0	2.0	1.0	3.0	0.0	2.0	2.0	0.25	0.0
	5.0	3.0	1.0	4.0	400.0	2.0	4.0	0.25	0.0
Prescrição original	VIG	aa	Lipídios	Na	Zn	K	Ca	Mg	PO4
	6.0	1.0	0.5	4.0	0.0	2.0	2.0	0.25	0.0

ANEXO V – CRONOGRAMA

14/02/2005

Conclusão da construção de um protótipo em Smalltalk para cálculo de similaridade entre prescrições. O cálculo foi realizado considerando-se *todos* os atributos de uma prescrição como igualmente relevantes. A métrica usada para calcular a similaridade foi o Nearest Neighbor.

23/02/2005

Visita ao hospital Joana de Gusmão para conhecimento do ambiente de uso do programa de NP. Primeiro conversa com Dra. Mônica a respeito de critérios de similaridade entre prescrições apontou que o atributo mais relevante na comparação é o peso do RN. Nenhum atributo da sugestão de prescrição foi discutido.

Inclusão da coluna *prescriptionNumber* na tabela *npppPrescription* no BD do programa de NP. Esta inclusão foi necessária para facilitar a obtenção do número da prescrição de um RN. Alterações no código do programa NP necessárias para registrar tal índice no BD foram efetuadas.

25/02/2005

Foi constatado que o programa de NP permitia a remoção de registros de RNs do BD, pois havia referências na tabela de prescrições (*nppprescription*) a RNs inexistentes na tabela de RNs (*newborn*). Iniciou-se a alteração do programa de NP e do BD para que RNs pudessem somente ser removidos da lista de RNs internados mostrada no programa e não do registro de RNs no BD.

17/03/2005

Inclusão de novas colunas no BD: *bloodType* e *rhesus* na tabela *newborn*. Inclusão da coluna *inUTI*, também na tabela *newborn*, para que o programa NP só mostrasse RNs que ainda estivessem na UTI. Foram efetuadas alterações necessárias no programa NP para que ele

coletasse o tipo sanguíneo no momento de registro de um RN e também não removesse os RNs do BD.

14/04/2005

Conversa com Dra. Mônica apontou que a idade gestacional e dias de parenteral também são atributos relevantes no cálculo da similaridade. Criação de um protótipo para cálculo de similaridade entre prescrições baseado nos critérios até aqui considerados: *peso, idade gestacional e dias de parenteral*.

24/05/2005

Coleta de 407 prescrições (incluindo já coletadas) do BD para uso no protótipo para o cálculo da similaridade. Alteração do protótipo e testes.

03/06/2005

Conversa com Dra. Mônica a respeito dos critérios utilizados. Coleta de 442 prescrições (incluindo já coletadas) do BD para uso no protótipo para o cálculo da similaridade.

26/08/2005

Apresentação de um protótipo para Dra. Mônica e para o Prof. Aldo, onde os atributos *dias de vida, dias de parenteral, idade gestacional e peso* eram considerados os atributos determinantes no cálculo da similaridade de prescrições. O peso de cada atributo podia ser alterado via interface do protótipo. Aprovação dos resultados pela Dra. Mônica. Sugestão, por parte da Dra. Mônica, de uma forma de adaptação das prescrições sugeridas.

09/11/2005

Sugestão, por parte do Prof. Aldo, de um teste retrospectivo, onde a sugestão de prescrição inicial dada pelo protótipo para um dado RN seria comparada com a prescrição efetivamente utilizada com aquele RN.

12/11/2005

Integração da máquina de inferência RBC ao programa de NP. Elaboração do teste retrospectivo.

16/11/2005

Apresentação dos resultados do teste retrospectivo para Dra. Mônica. Resultados são tidos como satisfatórios: discrepâncias eventuais eram resultado de casos excepcionais, mal representados na base de casos; outras discrepâncias eram reflexo da nova política de prescrições iniciais do HIJG.

11/12/2005

Início da tradução do programa de NP para Java. O objetivo da tradução era a reconstrução do programa sem os erros da versão Smalltalk, além da criação de uma nova interface gráfica com o usuário e inclusão de novas funcionalidades.

16/01/2006

Fim da tradução do programa de NP para Java. Instalação da versão Java no HIJG. Fim do uso da versão Smalltalk.

13/03/2006

Coleta de uma cópia do BD usada no programa de NP no HIJG, com o objetivo de se realizar novo teste retrospectivo. Até a data, 917 prescrições foram cadastradas, dos quais 36 eram prescrições iniciais utilizáveis.