

**UNIVERSIDADE FEDERAL DE SANTA CATARINA**  
**CENTRO TECNOLÓGICO**  
**DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA**  
**CURSO DE GRADUAÇÃO EM SISTEMAS DE INFORMAÇÃO**

**SISTEMA DE ACOMPANHAMENTO DA**  
**MERENDA ESCOLAR**

LEANDRO ZIMMER

Florianópolis  
2009

LEANDRO ZIMMER

# **SISTEMA DE ACOMPANHAMENTO DA MERENDA ESCOLAR**

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Sistemas de informação do Centro Tecnológico da Universidade Federal de Santa Catarina, como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informação.

Profº. José Eduardo De Lucca – Orientador

Florianópolis  
2009

LEANDRO ZIMMER

# **SISTEMA DE ACOMPANHAMENTO DA MERENDA ESCOLAR**

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Sistemas de Informação do Centro Tecnológico da Universidade Federal de Santa Catarina, como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: José Eduardo De Lucca

Banca Examinadora

Delson de Valois Santos

Renato Fileto

Florianópolis  
2009

**RESUMO**

Este trabalho visa criar uma ferramenta para auxiliar no acompanhamento da merenda escolar das prefeituras, através da criação de um módulo juntamente com o sistema i-educar, colaborando assim com a comunidade de software livre brasileira. O levantamento de requisitos para este trabalho foi realizado juntamente com o projeto Via Digital e a Prefeitura Municipal de Florianópolis.

Ele apresenta a fase de levantamento requisitos, análise, projeto e o módulo desenvolvido, utilizando HTML, PHP, e PostgreSQL.

**Palavras Chave:** Software Livre. PHP. Governo. Via Digital. I-educar. Sistema. PostgreSQL. Merenda Escolar. Alimentação Escolar.

# SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>8</b>
1.1 Apresentação.....	8
1.2 Objetivos.....	9
1.2.1 Objetivo Geral.....	9
1.2.2 Objetivos Específicos.....	9
1.3 Justificativa.....	9
1.4 Motivação.....	10
1.5 Organização Dos Capítulos.....	10
<b>2 MERENDA ESCOLAR.....</b>	<b>11</b>
2.1 Apresentação.....	11
2.2 Prefeitura Municipal De Florianópolis.....	12
<b>3 SISTEMA DE ACOMPANHAMENTO DA MERENDA ESCOLAR.....</b>	<b>18</b>
3.1 Definição Do Problema.....	18
3.2 Solução Proposta.....	18
3.2.1 Sistema Via Internet.....	18
3.2.2 Metodologia De Desenvolvimento.....	19
3.3 Software Livre.....	24
3.3.1 Definição.....	24
3.3.2 Via Digital.....	24
3.3.2 Sistema I-educar.....	25
3.4 Tecnologias.....	25
3.4.1 Servidor Apache.....	25
3.4.2 PHP.....	26
3.4.3 PostgreSQL.....	26
3.4.4 Interface.....	26
3.4.4.1 HTML.....	26
3.4.4.2 JavaScript.....	26
3.4.4.3 CSS.....	27
<b>4 DESENVOLVIMENTO.....</b>	<b>27</b>
4.1 Análise.....	27
4.2 Entrevistas.....	27
4.3 Requisitos.....	30
4.3.1 Requisitos Funcionais.....	30
4.3.2 Requisitos Não Funcionais.....	32
4.4 Diagramas De Casos De Uso.....	32
4.5 Diagramas De Classes.....	34
4.6 Diagramas De Banco De Dados.....	37
4.7 Projeto.....	39
4.7.1 Criação Da Base De Dados.....	39
4.7.2 Criação Das Classes.....	39
4.7.3 Criação Das Interfaces.....	39
4.7.4 Dificuldades Encontradas.....	40
<b>5 SISTEMA DESENVOLVIDO.....</b>	<b>40</b>
5.1 Funcionalidades E Telas.....	40
5.2 Implantação Do Módulo De Alimentação Escolar.....	43

<b>6 CONCLUSÃO.....</b>	<b>45</b>
<b>7 REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>46</b>
<b>8 ANEXOS.....</b>	<b>47</b>
8.1 Anexo A - <i>Requisitos Funcionais</i> .....	47
8.2 Anexo B - <i>Diagramas De Casos De Uso</i> .....	69
8.3 Anexo C - <i>Estrutura Do Banco De Dados</i> .....	85
8.4 Anexo D - <i>SQL Para Configuração Da Base De Dados</i> .....	91
8.5 Anexo E - <i>Artigo</i> .....	101
8.6 Anexo E - <i>Código Fontes</i> .....	111

## LISTA DE FIGURAS

FIG. 2.1 - PIRÂMIDE ALIMENTAR.....	14
FIG. 2.2 - ALIMENTOS NA PIRÂMIDE ALIMENTAR.....	16
FIG. 3.1 - HIERARQUIA DOS DIAGRAMAS UML.....	21
FIG. 3.2 - EXEMPLO DE CASO DE USO.....	22
FIG. 3.3 - CLASSE DO DIAGRAMA DE CLASSES.....	22
FIG. 3.4 - DIAGRAMA DE CLASSES - GENERALIZAÇÃO.....	22
FIG. 3.5 - DIAGRAMA DE CLASSES - ASSOCIAÇÃO.....	22
FIG. 3.6 - DIAGRAMA DE CLASSES - AGREGAÇÃO.....	23
FIG. 3.7 - DIAGRAMA DE CLASSES - COMPOSIÇÃO.....	23
FIG. 3.8 - DIAGRAMA DE BANCO DE DADOS.....	23
FIG. 4.1 - DIAGRAMA DE CLASSES PRODUTO FORNECEDOR.....	35
FIG. 4.2 - DIAGRAMA DE CLASSES ENVIO MENSAL.....	35
FIG. 4.3 - DIAGRAMAS DE CLASSES CARDÁPIOS.....	36
FIG. 4.4 - DIAGRAMA DE CLASSES IMC.....	36
FIG. 4.5 - DIAGRAMA DO BANCO DE DADOS.....	38
FIG. 5.1 - TELA DE PESQUISA DE PRODUTOS.....	42
FIG. 5.2 - TELA DE DETALHES DO PRODUTO.....	42
FIG. 5.3 - TELA DE INCLUSÃO E ALTERAÇÃO DE PRODUTOS.....	42
FIG. 5.4 - ESTRUTURA DE ARQUIVOS DO MÓDULO DE ALIMENTAÇÃO ESCOLAR.....	44

## LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
CGI	Common Gateway Interface
CSS	Cascading Style Sheet
GPL	General Public License
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IMC	Índice de Massa Corporal
PHP	Hypertext Preprocessor
PMF	Prefeitura Municipal de Florianópolis
PNAE	Programa Nacional de Alimentação Escolar
SC	Santa Catarina
SGBD	Sistema de Gerenciamento de Banco de Dados
SL	Software Livre
UML	Unified Modeling Language

# 1 INTRODUÇÃO

## 1.1 Apresentação

A gestão da informação nas empresas é fundamental para sua organização e bom funcionamento dos seus processos. Os softwares são uma ferramenta muito importante neste processo, eles auxiliam as pessoas há terem um maior controle das informações e também na tomada de decisões. O Brasil possui centenas de prefeituras, com as mais variadas necessidades de softwares para uma melhor gestão pública.

Em 2004 foi criado o Via Digital, um repositório de softwares livres e informações para a informatização de prefeituras. Uma das necessidades levantadas pelo projeto Via Digital é um sistema de acompanhamento da merenda nas instituições educacionais. Uma boa alimentação é essencial para os alunos estarem saudáveis e aptos a obter um bom aprendizado. O sistema que será proposto e implementado, será um módulo junto ao i-educar, que é um software livre para o controle de escolas e foi desenvolvido inicialmente pela prefeitura municipal de Itajaí-SC e disponibilizado como software livre através de uma parceria com a empresa Cobra Tecnologia.

O módulo desenvolvido será implantado na Prefeitura Municipal de Florianópolis/SC. A prefeitura conta com uma rede de mais de 60 escolas de educação infantil, fundamental e creches.

As premissas básicas deste trabalho estão definidas a seguir. Este trabalho está centrado no planejamento e desenvolvimento de um módulo junto ao i-educar parao acompanhamento da merenda escolar, com base na identificação dos processos relacionados a distribuição e custos dos alimentos e suas informações nutricionais e o cálculo do índice de massa corporal (IMC) dos alunos.

## **1.2 Objetivos**

### 1.2.1 Objetivo geral

Desenvolver um módulo junto ao software i-educar para o acompanhamento da merenda escolar das instituições de ensino das prefeituras.

### 1.2.2 Objetivos específicos

Primeiramente, estudar os conceitos e ferramentas envolvidas no desenvolvimento do sistema e colaborar com o projeto Via Digital e o i-educar.

Depois, levantar quais processos estão envolvidos no acompanhamento da merenda escolar em uma prefeitura, neste caso, estaremos realizando este levantamento na Prefeitura Municipal de Florianópolis. Em seguida, analisar os processos e criar a documentação necessária para o desenvolvimento do módulo no sistema.

Por fim, desenvolver o módulo proposto e disponibilizá-lo juntamente com o i-educar no portal de software público brasileiro.

## **1.3 Justificativa**

O mercado de software livre para prefeituras ainda carece de muitas ferramentas. O controle da merenda escolar, realizado pelas prefeituras em suas instituições de ensino, é um destes segmentos que necessitam de um sistema. Com base nesta necessidade, este trabalho visa planejar e desenvolver um módulo junto ao sistema i-educar para o acompanhamento da merenda escolar.

## 1.4 Motivação

Contribuir com a comunidade de software livre no desenvolvimento de sistemas que auxiliem na gestão pública das prefeituras. Visando assim, melhorar os serviços oferecidos para a população e um maior controle dos recursos públicos.

## 1.5 Organização dos capítulos

Este trabalho encontra-se organizado em seis capítulos. Os demais capítulos estão dispostos da seguinte forma. O capítulo 2 envolve os conceitos relacionados há merenda escolar no Brasil e detalhes de como funciona os processos relacionados a merenda escolar na Prefeitura Municipal de Florianópolis-SC. O capítulo 3 aborda o problema e a solução encontrada, bem como os conceitos e tecnologias envolvidas no desenvolvimento deste trabalho. O capítulo 4 descreve a fase de desenvolvimento, que inclui a fase de análise, definição de requisitos, criação de diagramas e o desenvolvimento do projeto e as dificuldades encontradas. O capítulo 5 demonstra o resultado final do sistema desenvolvido, descrevendo sutilmente as funcionalidades das telas e relatórios criados no sistema.

Finalmente, o capítulo 6 apresenta as conclusões e sugestões de trabalhos futuros.

## 2 MERENDA ESCOLAR

### 2.1 Apresentação

A merenda escolar é essencial para criar bons hábitos de alimentação e um dos seus principais objetivos é suprir, parcialmente, as necessidades nutricionais dos alunos. Uma alimentação adequada é fundamental para as crianças estarem aptas ao aprendizado, principalmente na infância que é o período que compõe a base de formação do ser humano (MASCARENHAS; SANTOS, 2006).

O Programa Nacional de Alimentação Escolar (PNAE) garante recursos financeiros, desde 1955, para a alimentação escolar da educação básica. O programa também define regras e métricas que as escolas devem cumprir. Os valores são repassados para as escolas dos estados, municípios e Distrito Federal. Os valores repassados atualmente pelo o programa são de R\$ 0,22 por aluno para cada dia letivo, para escolas indígenas e localizadas em comunidades de quilombos o valor é de R\$ 0,44, “*Em 2008, o PNAE investiu R\$ 1,49 bilhão para atender 34,6 milhões de alunos do ensino fundamental e da educação infantil.*”(FNDE, 2009), confira na tabela abaixo, tabela 2.1, a evolução dos recursos financeiros e alunos atendidos pelo programa . Os valores repassados devem ser incrementados pelos estados e municípios visando oferecer uma refeição de maior qualidade (FNDE, 2009).

Ano	Recursos financeiros (em milhões de R\$)	Alunos atendidos (em milhões)
1995	590,1	33,2
1996	454,1	30,5
1997	672,8	35,1
1998	785,3	35,3
1999	871,7	36,9
2000	901,7	37,1
2001	920,2	37,1
2002	848,6	36,9
2003	954,2	37,3
2004	1.025	37,8
2005	1.266	36,4
2006	1.500	36,3
2007	1.520	35,7
2008	1.490	34,6

Tabela 2.1 – Estatísticas do PNAE

Fonte: (FNDE/2009)

Os cardápios devem ser desenvolvidos por nutricionistas e suprir 30% das necessidades nutricionais dos alunos de escolas indígenas e localizadas em comunidades de quilombos e 15% das necessidades dos demais alunos. As refeições devem ser definidas respeitando os requisitos nutricionais e os hábitos alimentares e agrícolas de cada região (FNDE, 2009).

A seguir, iremos apresentar mais detalhes sobre a alimentação escolar e também informações sobre a rede municipal de escolas de Florianópolis e como funciona o trabalho para o fornecimento da merenda escolar.

## 2.2 Prefeitura Municipal de Florianópolis

A Prefeitura Municipal de Florianópolis (PMF) oferece merenda escolar para mais de 26.000 alunos, matriculados em suas mais de 90 unidades educativas (PMF, 2009a; PMF, 2009b). *“A alimentação escolar das unidades educativas, ligadas à Secretaria de Educação da Capital, é municipalizada, centralizada e com gestão própria da prefeitura.”* (PMF, 2009a).

Segundo a PMF (2009b), a estrutura de atendimento da rede municipal de ensino é composta por:

- *39 Creches, atendendo período integral e fornecendo 04 refeições por dia/criança;*
- *22 Núcleos de Educação Infantil, crianças de 2 a 6 anos, atendendo período integral e/ou meio período, fornecendo 02 e/ou 04 refeições por dia.*
- *12 Escolas Desdobradas e 25 Escolas Básicas, crianças de 1º a 8º série, fornecendo 01 refeição por dia/criança.*
- *53 Entidades Conveniadas – repasse de recursos financeiros e mais 10 Entidades que recebem recursos em espécie (gêneros alimentícios).*

A PMF visa oferecer uma alimentação de qualidade, que supra as necessidades nutritivas parciais e permita aos alunos desempenharem um bom rendimento escolar

e bons hábitos alimentares. O PNAE repassa aproximadamente 25% dos recursos financeiros aplicados na alimentação escolar do município, os outros 75% são aplicados pela prefeitura (PMF, 2009b).

Segundo a PMF (2009b), o PNAE pressupõe os seguintes requisitos:

- *Garantir a satisfação de 15% das necessidades nutricionais de crianças e adolescentes, através do aporte de 350 Kcal e 9g de proteínas;*
- *Introduzir hábitos alimentares saudáveis para crianças e adolescentes;*
- *Discutir e viabilizar a proposta de pedagogização da alimentação escolar.*

Os cardápios são definidos para cada escola e obedecem uma composição balanceada, seguindo a pirâmide alimentar (ver Fig. 2.1), com o objetivo de planejar uma refeição colorida e com alimentos dos grupos construtores, reguladores e energéticos (PMF, 2009b). Segundo a Dra. Ana Flávia Pinheiro (PINHEIRO, 2008), os construtores são alimentos conhecidos como plásticos e são responsáveis pela construção e formação do corpo (leites e derivados, leguminosas, carnes). Já os energéticos fornecem energia para todos os processos metabólicos e exercícios físicos (cereais, feculentos, gorduras). Os reguladores, por sua vez, regulam e controlam as funções do organismo, mantendo a saúde da pele e participando da formação dos tecidos (frutas, hortaliças, verduras, legumes). Os cardápios são definidos por nutricionistas, merendeiras, diretor e equipe pedagógica e visam obedecer as definições acima e as características de cada região.



Fig. 2.1 – Pirâmide Alimentar

Fonte: [http://www.pmf.sc.gov.br/educa/recursos\\_imagens/recursospiramide1.JPG](http://www.pmf.sc.gov.br/educa/recursos_imagens/recursospiramide1.JPG)

De acordo com a faixa etária dos alunos, e das necessidades nutritivas, são definidas as quantidades necessárias de cada alimento por aluno, também chamada de per capita de alimentos, os alimentos são divididos em menos perecíveis e mais perecíveis. Os alimentos menos perecíveis possuem um tempo maior para consumo e por este motivo são entregues mensalmente, já os alimentos mais perecíveis são entregues mensalmente. Na tabela abaixo (tabela 2.2) apresentamos a listagem dos dois tipos de alimentos que são fornecidos e na figura (fig. 2.2) você confere os alimentos agrupados de acordo com a pirâmide alimentar.

Menos Perecíveis	Mais Perecíveis
Amido de milho	Brócoli Chinês
Arroz	Caqui
Açúcar	Goiaba
Bolacha maisena	Músculo Moído
Bolacha salgada	Bife de Coxão Mole
Biscoito de Coco	Filé de peito de frango
Café	Lombo suíno
Colorau	Sobrecoxa de frango
Concentrado de tomate	Ovos
Doce de banana orgânico	Abacaxi

Farinha láctea	Banana branca
Farinha de mandioca	Laranja
Farinha de milho	Maçã
Farinha de trigo	Mamão
Fermento de bolo	Melancia
Feijão preto	Tangerina
Fósforo	Abóbora
Leite em pó	Alho
Leite de soja	Batata inglesa
Massa de sopa	Beterraba
Massa espaguete	Cenoura
Massa parafuso	Chuchu
Óleo de soja	Repolho
Pão de forma	Tomate
Mistura p/ Bolo, Chocolate	Couve-flor
Mistura p/ Bolo, Laranja	Pêssego
Mistura p/ Bolo, Chocolate, S/ Glúten	
Mistura p/ Bolo, Fubá	
Rosca de leite	
Sal	
Vinagre	
Achocolatado	
Aveia	
Fermento Biológico	
Fermento em Pó Químico	

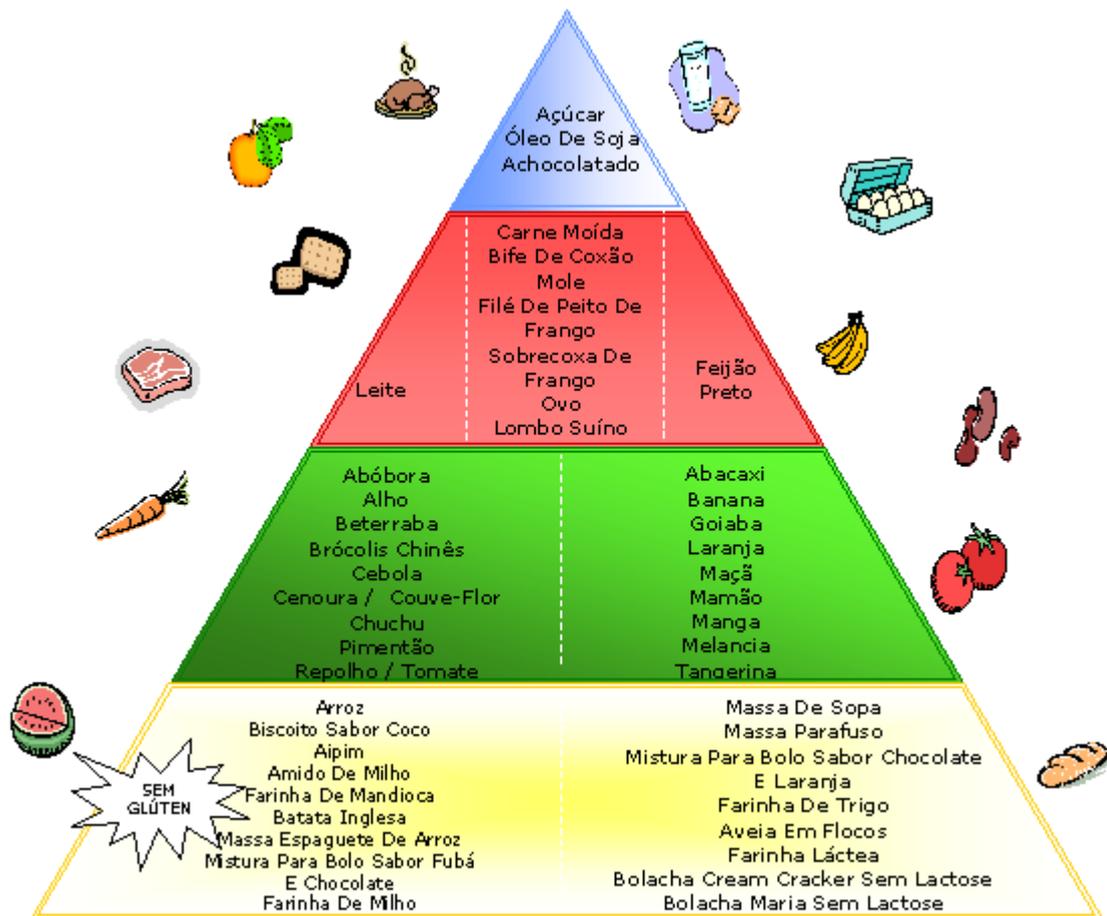


Fig. 2.2 – Alimentos na Pirâmide Alimentar

Fonte: [http://www.pmf.sc.gov.br/educa/recursos\\_imagens/piramidealimentar1.bmp](http://www.pmf.sc.gov.br/educa/recursos_imagens/piramidealimentar1.bmp)

Para a manutenção dos processos a PMF realiza ações nas unidades educativas, que são desenvolvidas pela coordenadoria e também executa as atividades que são desenvolvidas no órgão central. Nas unidades educativas, segundo a PMF(2009b) são realizadas as seguintes ações:

- *Acompanhamento da qualidade dos alimentos entregues, controle de estoque e da ficha de previsão de alimentos;*
- *Elaboração e reavaliação do cardápio junto com as merendeiras, diretores e equipe pedagógica;*
- *Acompanhamento durante a refeição da criança, diagnosticando aceitação e sobras;*

- *Orientação para merendeiras quanto aos cuidados higiênico-sanitários em relação ao serviço de alimentação, bem como à qualidade nutricional das refeições servidas;*
- *Participação em Paradas Pedagógicas junto à Unidade Escolar;*
- *Participação em reuniões de pais e funcionários.*

As atividades realizadas no órgão central, segundo a PMF(2009b), são:

- *Especificação dos Gêneros Alimentícios para licitação;*
- *Acompanhamento de todo o processo licitatório (compra, análise de amostras e distribuição de alimentos) e também dos contratos;*
- *Participação no Projeto de Pesquisa: “Sobrepeso e Obesidade e sua relação com o estilo de vida em escolares de 7 a 10 anos no município de Florianópolis, SC”;*
- *Elaboração, implantação e acompanhamento do projeto “Horta Escolar”;*
- *Especificação dos utensílios e equipamento de cozinha para licitação;*
- *Participação no Conselho Municipal de Alimentação Escolar;*
- *Capacitação semestral de todas as merendeiras, responsáveis pelo armazenamento, preparo e distribuição das refeições servidas às crianças.*
- *Fiscalização do cumprimento da Lei das Cantinas Escolares. (COMAE e Coordenadoria)*
- *Fiscalização e orientação do Controle de Estoque dos depósitos das Unidades escolares;*
- *Trabalho em conjunto com as nutricionistas da Empresa Terceirizada;*
- *Pesquisa de fornecedores para a introdução de novos alimentos na pauta da alimentação escolar, inclusive produtos orgânicos.*

- *Elaboração de per capita de acordo com a faixa etária dos alunos;*
- *Levantamento mensal do número de refeições servidas.*

Neste capítulo apresentamos detalhes sobre a alimentação escolar e modelo que é utilizado pela coordenadoria de alimentação escolar da prefeitura municipal de Florianópolis. No próximo capítulo iremos abordar questões como a necessidade apresentada pela coordenadoria, solução proposta e a metodologia utilizada.

### **3 SISTEMA DE ACOMPANHAMENTO DA MERENDA ESCOLAR**

#### **3.1 Definição do Problema**

Atualmente, a prefeitura de Florianópolis não consegue acompanhar de forma prática e rápida as quantidades de alimentos distribuídos na rede municipal. A falta de registro destas informações em uma base para análise histórica também é uma necessidade identificada pela prefeitura. Com o registro dos alimentos distribuídos e os valores nutricionais dos mesmos, seria possível realizar uma avaliação das calorias e proteínas que estão sendo consumidas pelos alunos e o custo das refeições.

Outro problema, é a falta de dados antropométricos dos alunos. Com dados como peso, medida e idade, seria possível realizar comparativos com as refeições servidas.

#### **3.2 Solução Proposta**

##### **3.2.1 Sistema via internet**

O solução proposta é um software livre, disponibilizado via internet, que permita o acesso de qualquer micro com um navegador Web e acesso a internet. O sistema irá permitir a inclusão das quantidades e valores nutricionais dos alimentos fornecidos para cada escola da rede municipal. O sistema também irá permitir a inclusão de alunos e dados como peso e altura. O sistema terá uma área de cadastros e uma área com relatórios com base nas informações cadastradas.

### 3.2.2 Metodologia de Desenvolvimento

O sistema que será desenvolvido é um módulo no i-educar e pode ser classificado como um sistema de pequeno porte. O desenvolvimento do sistema terá a participação de apenas um desenvolvedor que executa o papel de gerente do projeto e está em contato direto com o cliente. O tempo para desenvolvimento do sistema é curto. O cliente tem interesse em receber versões iniciais com partes das funcionalidades do sistema. Por estes e outros motivos, que serão apresentados abaixo, o modelo escolhido será baseado em metodologias ágeis (SILVA, 2008/1).

O processo de desenvolvimento de software ágil evoluiu a partir de 1990, como alternativa ao modelo cascata. “O processo originou-se da visão de que o modelo em cascata era burocrático, lento e contraditório a forma usual com que os engenheiros de software sempre realizaram trabalho com eficiência.” (Wikipédia, 2009a).

Esta metodologia visa diminuir os riscos no processo de desenvolvimento do sistema. Ele visa dividir as etapas do desenvolvimento em funcionalidades que gerem uma nova versão do sistema. O principal objetivo é reduzir os períodos destas etapas, chamados de iteração. Assim é possível lançar novas versões com maior frequência e o cliente pode validar e sugerir mudanças nas novas funcionalidades, evitando assim que grandes blocos do sistema possam ser desenvolvidos sem uma avaliação do cliente, aumentando o risco de retrabalho caso uma etapa não supra a necessidade esperada pelo cliente. (Wikipédia, 2009a)

Este trabalho não está centrado em uma metodologia ágil específica, ele irá abordar as características principais das metodologias ágeis. As principais características que serão adotadas por este trabalho, segundo (Wikipédia, 2009a), são:

- *Garantir a satisfação do consumidor entregando rapidamente e continuamente softwares funcionais;*
- *Softwares funcionais são entregues frequentemente (semanas, ao invés de meses);*
- *Softwares funcionais são a principal medida de progresso do projeto;*

- *Até mesmo mudanças tardias de escopo no projeto são bem-vindas.*
- *Cooperação constante entre pessoas que entendem do 'negócio' e desenvolvedores;*
- *Projetos surgem através de indivíduos motivados, e que deve existir uma relação de confiança.*
- *Design do software deve prezar pela excelência técnica;*
- *Simplicidade;*
- *Rápida adaptação às mudanças;*
- *Indivíduos e interações mais do que processos e ferramentas;*
- *Software funcional mais do que documentação extensa;*
- *Colaboração com clientes mais do que negociação de contratos;*
- *Responder a mudanças mais do que seguir um plano.*

Para desenvolvimento deste trabalho será feita uma análise do problema e depois entrevistas com as pessoas envolvidas com a área que utilizará o sistema junto ao cliente. Com base nas informações levantadas serão definidos os requisitos funcionais. *“O requisitos funcionais são aqueles que descrevem o comportamento do sistema, suas ações para cada entrada, ou seja, é aquilo que descreve o que tem que ser feito pelo sistema.”* (NUNES, 2007). Após esta definição serão levantados os requisitos não funcionais. *“Os requisitos não funcionais são aqueles que expressam como deve ser feito (não confundir requisitos não funcionais com design). Em geral se relacionam com padrões de qualidade como confiabilidade, performance, robustez, etc.”* (NUNES, 2007). Com os requisitos definidos a próxima etapa será a definição de uma visão completa do sistema e sua estrutura, para estas definições iremos utilizar três tipos de diagramas UML.

A Unified Modeling Language (UML) é uma linguagem e não uma metodologia de desenvolvimento. Ela não diz como um sistema deve ser feito, ela auxilia a visualização das funcionalidades e do sistema através de diagramas. *“Os objetivos da UML são: especificação, documentação, e estruturação para sub-*

visualização e maior visualização lógica de um total desenvolvimento de um sistema de informação. A UML é um modo de padronizar as formas de modelagem.” (Wikipédia, 2009b). A UML na versão 2.0 define 12 tipos de diagramas, confira na figura (Fig 3.1) apresentada abaixo.(Wikipédia, 2009b)

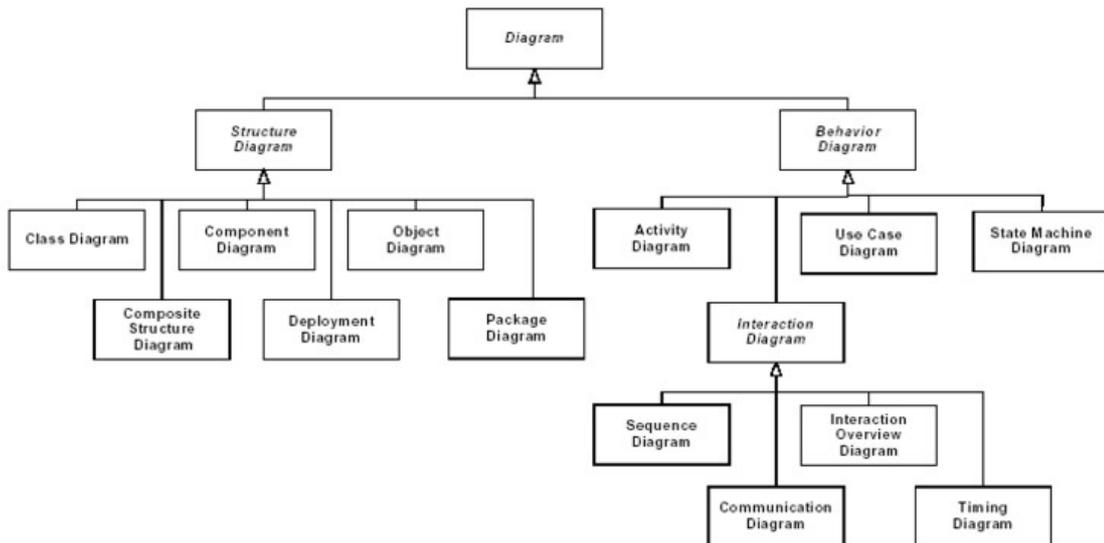


Fig. 3.1 – Hierarquia dos diagramas UML

Fonte: [http://pt.wikipedia.org/wiki/Ficheiro:Uml\\_hierarchie\\_des\\_diagrammes.png](http://pt.wikipedia.org/wiki/Ficheiro:Uml_hierarchie_des_diagrammes.png)

Neste trabalho iremos utilizar somente os diagramas de Casos de Uso (Use Case) e Classes (Class). Também iremos utilizar o diagrama de banco de dados, mas este não faz parte da UML. Abaixo apresentamos mais detalhes dos 3 tipos de diagramas que serão utilizados.

#### Diagramas de Casos de Uso

Os diagramas de casos de uso definem os principais recursos do sistema e suas interações com os usuários, chamados de atores. Atores podem ser um usuário ou computador e é o elemento externo que interage com o sistema (KDE, 2009). “Por ser um diagrama de fácil compreensão, é muito utilizado para exibir as funcionalidades para clientes, ou para ter uma visão geral de partes de um sistema.” (SILVA, 2008/1). Abaixo apresentamos um exemplo simples de dois atores interagindo com uma funcionalidade do sistema.

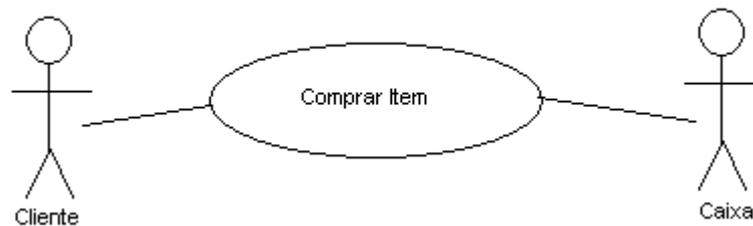


Fig. 3.2 – Exemplo de Caso de Uso

Fonte: <http://conteudo.imasters.com.br/2753/02.gif>

## Diagramas de Classes

A diagrama de classes apresenta uma visão da estrutura do sistema, através da descrição de suas classes e seus relacionamentos. Uma classe apresenta seu nome e seus atributos e métodos (ver Fig. 3.3). As classes podem apresentar associações de generalização (ver Fig. 3.4), associações (ver Fig. 3.5), agregação (ver Fig 3.6) e composição (ver Fig. 3.7). (KDE, 2009)

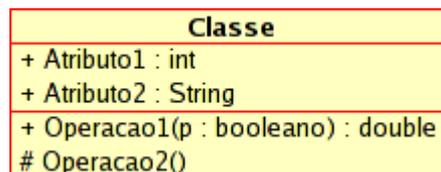


Fig. 3.3 – Classe do diagrama de classes

Fonte: [http://docs.kde.org/stable/pt\\_BR/kdesdk/umbrello/class.png](http://docs.kde.org/stable/pt_BR/kdesdk/umbrello/class.png)

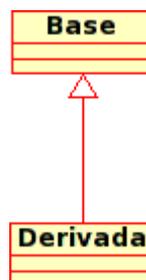


Fig. 3.4 – Diagrama de classes – Generalização

Fonte: [http://docs.kde.org/stable/pt\\_BR/kdesdk/umbrello/generalization.png](http://docs.kde.org/stable/pt_BR/kdesdk/umbrello/generalization.png)



Fig. 3.5 – Diagrama de classes – Associação

Fonte: [http://docs.kde.org/stable/pt\\_BR/kdesdk/umbrello/association.png](http://docs.kde.org/stable/pt_BR/kdesdk/umbrello/association.png)



Fig. 3.6 – Diagrama de classes – Agregação

Fonte: [http://docs.kde.org/stable/pt\\_BR/kdesdk/umbrello/aggregation.png](http://docs.kde.org/stable/pt_BR/kdesdk/umbrello/aggregation.png)



Fig. 3.7 – Diagrama de classes – Composição

Fonte: [http://docs.kde.org/stable/pt\\_BR/kdesdk/umbrello/composition.png](http://docs.kde.org/stable/pt_BR/kdesdk/umbrello/composition.png)

### Diagramas de Banco de Dados

Os diagramas de banco de dados mostram a estrutura da base de dados, suas tabelas, atributos, relacionamentos, chaves e outros. É muito útil para a visualização da estrutura de dados. (SILVA,2008/1)

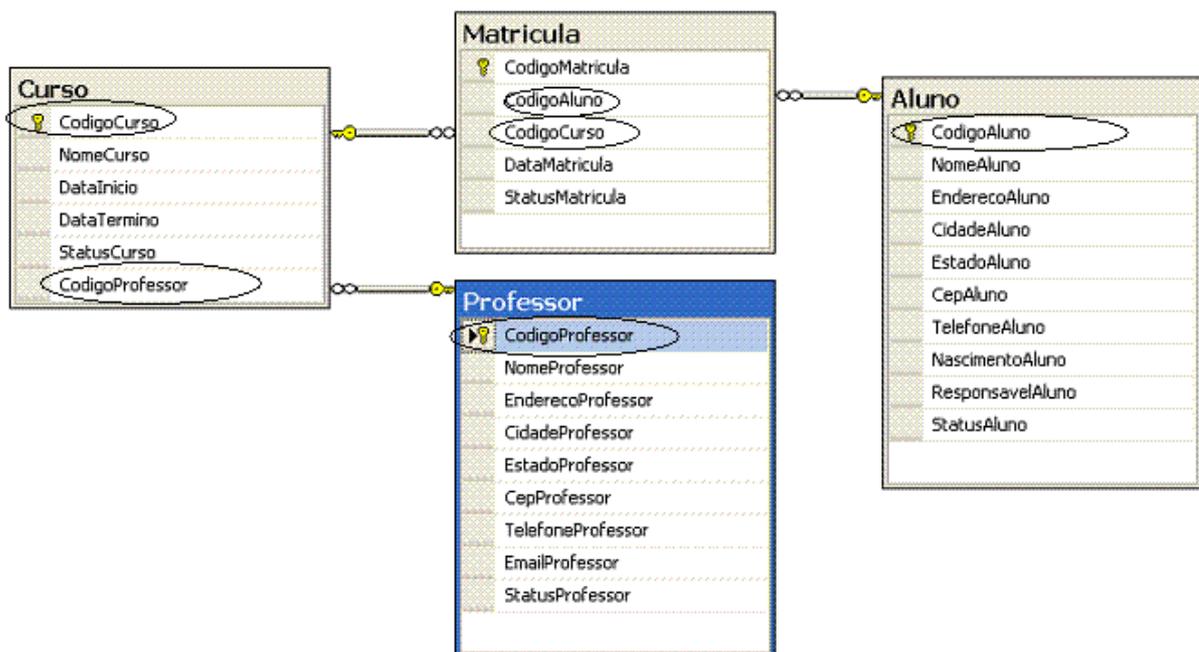


Fig. 3.8 – Diagrama de banco de dados

Fonte: <http://www.macoratti.net/vb5cbd1g.gif>

## 3.3 Software Livre

### 3.3.1 Definição

O Software Livre(SL), segundo a definição da Free Software Foundation, é qualquer programa que possa ser utilizado, copiado, alterado e distribuído sem nenhuma restrição. O SL possui alguns tipos de licenças que podem ser utilizadas, uma das principais é a GPL, onde define que o código fonte da aplicação deve ser disponibilizado em local onde possa ser acessado. O desenvolvimento baseado em SL visa acesso ao código, possibilitando que várias pessoas possam contribuir com a evolução do produto ou adaptá-los de acordo com suas necessidades (WIKIPEDIA, 2009e).

### 3.3.2 Via Digital

O Via Digital é um projeto que visa disponibilizar soluções de software livre para os municípios. O Via Digital surgiu a partir de uma pesquisa realizada em 2004, onde as prefeituras identificaram a necessidade da criação de um repositório único para obter soluções de software livre e a criação de um ambiente colaborativo para troca de experiências.

Abaixo apresentamos a definição do projeto segundo VIADIGITAL(2009):

*A proposta do projeto VIA DIGITAL é a implantação de um centro de referência virtual (portal) que permita o acesso e o desenvolvimento compartilhado de software livre para gestão municipal, propiciando a construção de um acervo público de soluções com foco na realidade das pequenas prefeituras. Tudo via Software Livre (SL).*

*Além disso, visando informatizar os órgãos municipais de forma sustentada e promover o desenvolvimento local, o portal servirá ainda como um ambiente virtual de capacitação técnica, troca de experiências, acesso a fontes idôneas de informação e criação de oportunidades de negócio a todos os participantes da cadeia de*

*produção do setor: técnicos e executivos da administração pública, profissionais de empresas de software e serviços relacionados, participantes das comunidades de SL e empreendedores.(VIADIGITAL, O Caminho inteligente para informatização pública, 2009)*

Este trabalho visa colaborar com o projeto Via Digital desenvolvendo um sistema para o setor de merenda escolar para as prefeituras, disponibilizando o sistema e as informações levantadas no repositório do projeto.

### 3.3.2 Sistema i-educar

O i-educar é um sistema de controle de escolas que foi desenvolvido inicialmente pela prefeitura municipal de Itajaí-SC e foi disponibilizado como software livre através de uma parceria com a empresa Cobra Tecnologia(SoftwarePúblico, 2009).

Os objetivos do i-educar segundo SoftwarePúblico(2009) são:

*Seu objetivo é o de centralizar todas as informações do sistema educacional municipal, diminuindo a necessidade de uso de papel, a duplicidade de documentos, o tempo de atendimento ao cidadão e racionalizando o trabalho do servidor público(SoftwarePúblico, Sobre i-educar, 2009).*

O i-educar utiliza as tecnologias apresentadas a seguir.

## 3.4 Tecnologias

### 3.4.1 Servidor Apache

O servidor APACHE é a principal tecnologia da Apache Software Foundation. Foi criado em 1995 por Rob McCool. O servidor é compatível com o protocolo HTTP versão 1.1 e é estruturado em módulos, sendo que o usuário pode escrever seus próprios módulos com a API do software. O servidor APACHE com PHP e

PostgreSQL é amplamente utilizado em planos de hospedagem de páginas WEB no mundo.

### 3.4.2 PHP

PHP é uma linguagem de programação muito utilizada para gerar páginas WEB dinâmicas. A linguagem surgiu por volta de 1994, como um pacote de programas CGI criados por Rasmus Lerdof e atualmente fornece suporte ao desenvolvimento orientado a objeto. O PHP fornece suporte a um grande número de base de dados e a diversos protocolos.

### 3.4.3 PostgreSQL

O PostgreSQL é um Sistema de Gerenciamento de Banco de Dados (SGBD) de código aberto, com mais de 15 de desenvolvimento ativo. Roda em todos os grandes sistemas operacionais, dentre eles podemos destacar o Linux, Windows, Mac OS/X, HP-UX, AIX, Solaris. Existem alguns sistemas com PostgreSQL que gerenciam mais de 4TB de dados (PostgreSQL, 2009).

### 3.4.4 Interface

#### 3.4.4.1 HTML

A Linguagem de Marcação de Hipertexto (HTML, do inglês HyperText Markup Language) é uma linguagem utilizada para produzir páginas WEB. O documento HTML pode ser interpretado por navegadores WEB e descreve a estrutura da página, que pode ser formada por tabelas, parágrafos, imagens, cabeçalhos contendo JavaScript e CSS embutidos.

#### 3.4.4.2 JavaScript

JavaScript se chamava inicialmente LiveScript e foi criado em 1995, pela Netscape, com o principal intuito de validar formulários e interagir com a página. O JavaScript é uma linguagem de programação interpretada pelo navegador Web.

### 3.4.4.3 CSS

CSS é um folha de estilo que define a apresentação de um documento HTML. Ao invés de definir o formato no documento HTML, o desenvolvedor cria um CSS e pode vincular com diversos documentos. Sempre que desejar alterar o formato de apresentação, basta alterar o documento CSS.

## 4 DESENVOLVIMENTO

### 4.1 Análise

O projeto do sistema foi realizado com base nos estudos realizados sobre tecnologias e merenda escolar, juntamente com as informações obtidas com o setor de alimentação escolar da prefeitura municipal de Florianópolis. Como o objetivo deste trabalho é incorporar o módulo de acompanhamento de merenda escolar ao i-educar, foi necessário uma análise da estrutura atual do sistema e seu funcionamento para iniciar a fase de desenvolvimento. A seguir apresentamos detalhes do levantamento de requisitos e da fase de projeto.

### 4.2 Entrevistas

Após estudar o setor de alimentação escolar foram definidos tópicos para levantar os requisitos para o desenvolvimento do sistema. A primeira entrevista foi realizada no dia 14 de maio de 2009, na coordenação de alimentação escolar da prefeitura e teve a participação do coordenador e de diversas pessoas envolvidas na manutenção do setor. A nutricionista , Renata Faust, responsável pelo setor, é o principal contato do projeto com a coordenadoria. Renata está contribuindo ativamente com o projeto. Abaixo apresentamos as questões que foram levantadas na entrevista, que foi dividida em dois tópicos:

#### **Estoque**

1. Existe algum controle de estoque atualmente nas escolas?

Resp.: As escolas enviam mensalmente um balanço do estoque.

2. Seria possível realizar este controle? Diariamente? Por refeição?

Resp.: Seria muito difícil cobrar este controle. Mensalmente já temos dificuldades.

3. As escolas poderiam realizar este controle informando entradas e saídas?

Resp.: Seria necessário uma pessoa para realizar esta tarefa. Atualmente seria inviável.

4. Todas as escolas possuem acesso a internet?

Resp.: Sim.

### **Cardápio**

1. Como funciona?

Resp.: São definidos pelas nutricionistas e especificados por refeição.

2. Os cardápios são padronizados?

Resp.: Não. Cada escola possui o seu cardápio.

3. Uma escola tem cardápios diferentes na mesma refeição?

Resp.: Não.

4. Quantidade de refeições servidas diariamente é importante?

Resp.: Sim. Mas é difícil saber quantas refeições são realmente servidas. O cálculo de refeições servidas é feito previamente com base no número de alunos matriculados.

5. O valor nutricional da refeição é importante? Como vocês fazem?

Resp.: É uma das informações mais importantes. As nutricionistas que definem os cardápios realizam este controle para oferecer as quantidades adequadas. Seria interessante uma forma de podermos acompanhar as quantidades nutricionais que estão sendo oferecidas de acordo com os alimentos enviados para as escolas.

6. Existem escolas com creches, educação infantil e ensino fundamental juntos?

Resp.: Não.

7. Controle de peso, idade, altura é importante?

Resp.: Atualmente estas medidas são obtidas somente de um pequeno grupo de alunos, mas seria interessante podermos registrar e acompanhar a evolução destas informações.

8. Todas as escolas tem nutricionistas?

Resp.: Sim. Temos uma nutricionista para cada 20 escolas.

9. Controle da validade dos alimentos?

Resp.: É realizado pelas merendeiras e acompanhado mensalmente pelas nutricionistas que visitam as escolas. Não temos problemas com alimentos vencidos, pois os alimentos não acumulam devido ao acompanhamento realizado pelas escolas e nutricionistas.

10. Existe algum acompanhamento da quantidade de alimentos enviados?

Resp.: Os alimentos são distribuídos pelas empresas vencedoras das licitações. Os mais perecíveis são distribuídos semanalmente e os menos perecíveis são mensalmente. Mensalmente é realizado um mapeamento na coordenadoria para definir a quantidade para cada escola. Este mapeamento é realizado de acordo com a quantidade e faixa etária dos alunos e com o per cápita mensal de cada alimento. Seria interessante poder acompanhar as quantidades enviadas para cada escola e poder comparar as quantidades consumidas por alunos de escolas diferentes.

Na entrevista acima já foram adicionadas as observações realizadas pela coordenadoria de alimentação escolar. As observações foram realizadas na entrevista e em diversos outros contatos realizados por e-mail e telefone.

No próximo tópico apresentamos os requisitos levantados durante as entrevistas.

## 4.3 Requisitos

Abaixo apresentamos uma descrição dos requisitos necessários para o sistema, de acordo com as necessidades levantadas juntamente com a coordenadoria de alimentação escolar do município.

Um sistema Web que permitirá o cadastro de usuários que poderão cadastrar novos usuários, escolas, alunos, produtos e suas referências nutricionais. Mensalmente os usuários irão cadastrar as quantidades de alimentos enviados para cada escola, deverá informar a quantidade por produto e a quantidade de vezes que ele será servido no mês. Juntamente com a quantidade de alimentos o usuário deverá informar o número de alunos. Com base nestas informações e detalhes do produto, serão calculados os valores nutricionais oferecidos mensalmente por aluno e serão gerados os históricos de distribuição de alimentos. Também será permitido a inclusão de medidas como peso e altura por aluno, com o objetivo de identificar o IMC dos alunos.

Acima apresentamos uma descrição superficial dos requisitos do sistema, nos próximos tópicos iremos apresentar uma visão mais completa dos requisitos funcionais e não funcionais.

### 4.3.1 Requisitos Funcionais

Abaixo apresentamos a lista dos requisitos funcionais do sistema, veja a descrição completa de cada requisito funcional no Anexo A.

[RF01] – Incluir Cardápio

[RF02] – Pesquisar Cardápio

[RF03] – Detalhar Cardápio

[RF04] – Alterar Cardápio

[RF05] – Incluir Nutricionista Escola

[RF06] – Pesquisar Nutricionista Escola

- [RF07] – Detalhar Nutricionista Escola
- [RF08] – Alterar Nutricionista Escola
- [RF09] – Remover Nutricionista Escola
- [RF10] – Incluir Produto
- [RF11] – Pesquisar Produto
- [RF12] – Detalhar Produto
- [RF13] – Alterar Produto
- [RF14] – Incluir Fornecedor
- [RF15] – Pesquisar Fornecedor
- [RF16] – Detalhar Fornecedor
- [RF17] – Alterar Fornecedor
- [RF18] – Incluir Produto Fornecedor
- [RF19] – Pesquisar Produto Fornecedor
- [RF20] – Detalhar Produto Fornecedor
- [RF21] – Alterar Produto Fornecedor
- [RF22] – Remover Produto Fornecedor
- [RF23] – Incluir Envio Mensal
- [RF24] – Pesquisar Envio Mensal
- [RF25] – Detalhar Envio Mensal
- [RF26] – Alterar Envio Mensal
- [RF27] – Remover Envio Mensal
- [RF28] – Listar IMC Aluno
- [RF29] – Incluir IMC

[RF30] – Pesquisar IMC

[RF31] – Detalhar IMC

[RF32] – Alterar IMC

[RF33] – Remover IMC

[RF34] – Relatório Resumo por Escola

[RF35] – Relatório Resumo por Mês

[RF36] – Relatório IMC por Escola

[RF37] – Relatório IMC por Mês

#### 4.3.2 Requisitos Não Funcionais

Os requisitos não funcionais estão descritos abaixo.

O acompanhamento de merenda escolar deve ser desenvolvido como um módulo do software livre i-educar, conseqüentemente deve ser um software livre em PHP e com base de dados PostgreSQL. O código fonte deve estar disponível na internet no portal de software livre público.

Deve ser um sistema Web que rode no Internet Explorer 6 ou superior e Firefox 3 ou superior.

Ter uma interface que carregue rapidamente, ou seja, com poucas informações. Sua usabilidade deve ser há mais simples possível.

Seu acesso será totalmente restrito, ou seja, somente usuários cadastrados terão acesso ao sistema.

#### 4.4 Diagramas de Casos de Uso

Os diagramas foram divididos por funcionalidades e cada diagrama possui seus casos de uso expandidos. Para visualizar todos os diagramas e casos de uso expandidos, ver o Anexo B.

Abaixo apresentamos a lista de diagramas e os casos de uso expandidos:

### **Manter Cardápio**

Diagrama que representa a manutenção de cadastro dos cardápios das escolas. Os casos expandidos são: Incluir Cardápio, Pesquisar Cardápio, Detalhar Cardápio e Alterar Usuário.

### **Manter Nutricionista Escola**

Diagrama que representa a manutenção dos nutricionistas de cada escola. Os casos expandidos são: Incluir Nutricionista Escola, Pesquisar Nutricionista Escola, Detalhar Nutricionista Escola, Alterar Nutricionista Escola e Remover Nutricionista Escola.

### **Manter Produto**

Diagrama que representa a manutenção dos produtos cadastrados no sistema. Os casos expandidos são: Incluir Produto, Pesquisar Produto, Detalhar Produto e Alterar Produto.

### **Manter Fornecedor**

Diagrama que representa a manutenção dos fornecedores de produtos. Os casos expandidos são: Incluir Fornecedor, Pesquisar Fornecedor, Detalhar Fornecedor e Alterar Fornecedor.

### **Manter Produto Fornecedor**

Diagrama que representa a manutenção dos produtos fornecidos por cada fornecedor. Os casos expandidos são: Incluir Produto Fornecedor, Pesquisar Produto Fornecedor, Detalhar Produto Fornecedor, Alterar Produto Fornecedor e Remover Produto Fornecedor.

### **Manter Envio Mensal**

Diagrama que representa a manutenção dos envios mensais de produtos para as escolas. Os casos expandidos são: Incluir Envio Mensal, Pesquisar Envio Mensal, Detalhar Envio Mensal, Alterar Envio Mensal e Remover Envio Mensal.

### **Manter IMC**

Diagrama que representa a manutenção dos IMCs cadastrados para os alunos. Os casos expandidos são: Pesquisar IMC Aluno, Incluir IMC, Pesquisar IMC, Detalhar IMC, Alterar IMC e Remover IMC.

### **Manter Relatórios**

Diagrama que representa a manutenção dos relatórios do sistema. Os casos expandidos são: Relatório Resumo por Escola, Relatório Resumo por Mês, Relatório IMC por Escola e Relatório IMC por Mês.

## **4.5 Diagramas de Classes**

Para a criação do módulo de acompanhamento da merenda escolar, junto ao sistema i-educar, foram criadas nove classes. Estas classes iniciam com clsAlimentacao, as outras classes que serão apresentadas nos diagramas já faziam parte do sistema i-educar e estão sendo apresentadas nos diagramas pois foram utilizadas na criação deste módulo. Os diagramas de classes foram agrupados por afinidade e serão apresentados abaixo em 4 diagramas.

O primeiro diagrama de classes apresentado abaixo (Fig. 4.1) apresenta a classe de produtos, grupo de produtos, fornecedores e os produtos por fornecedores.

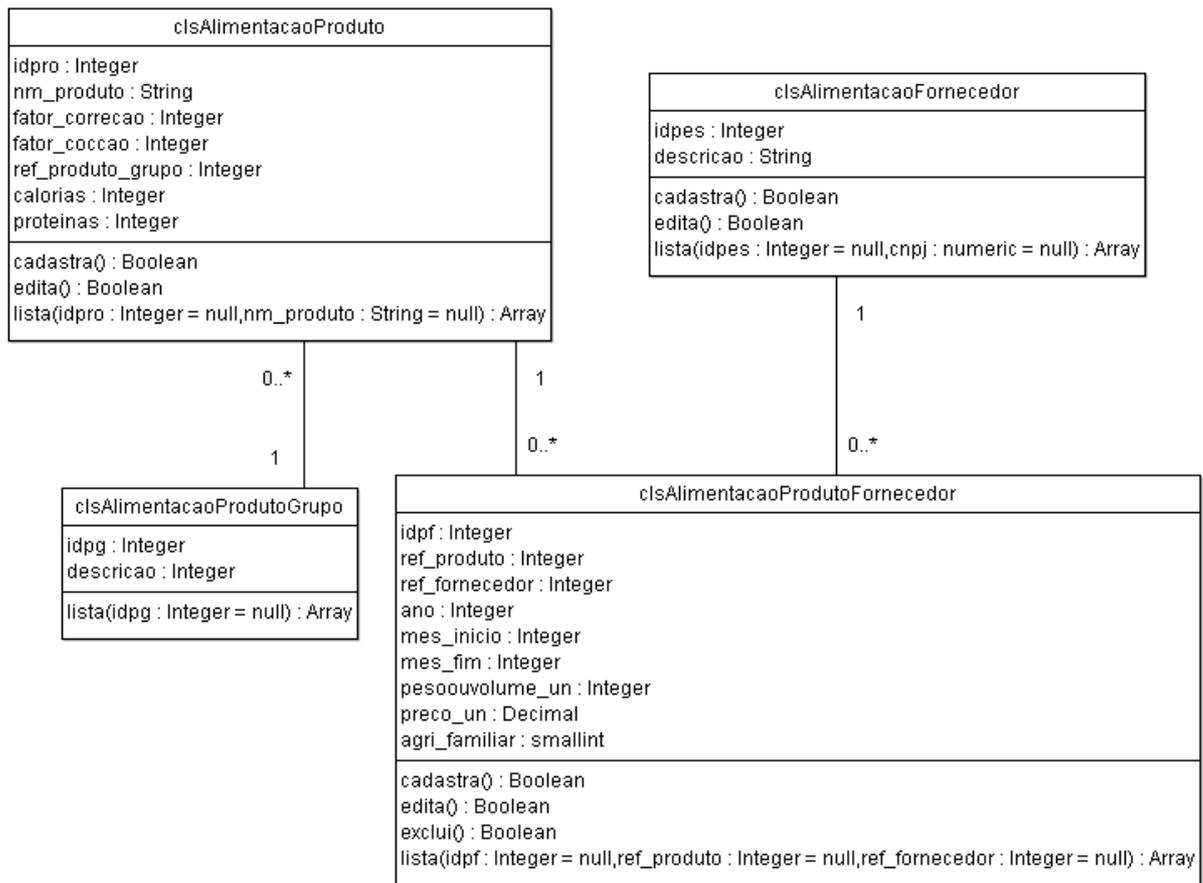


Fig. 4.1 – Diagrama de Classes Produto Fornecedor

No diagrama seguinte (Fig. 4.2) representa as classes relacionadas ao envio mensal de produtos para as escolas.

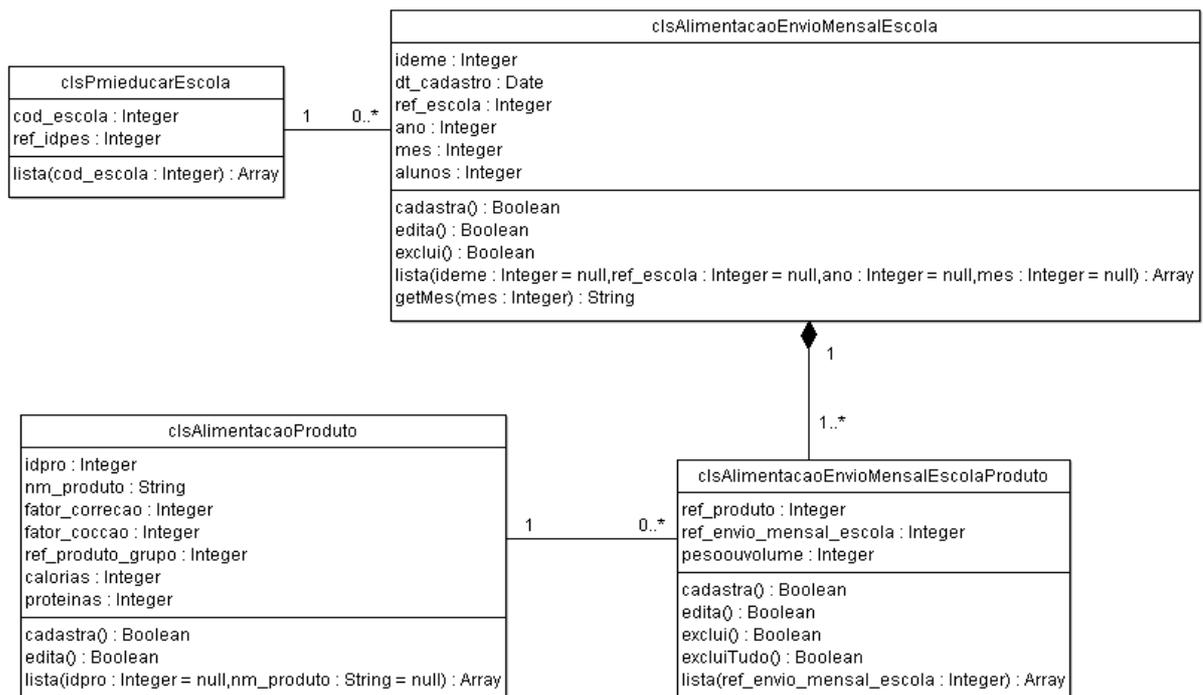


Fig. 4.2 – Diagrama de Classes Envio Mensal

O diagrama abaixo (Fig. 4.3) agrupa as classes relacionadas ao cadastro de cardápios das escolas.

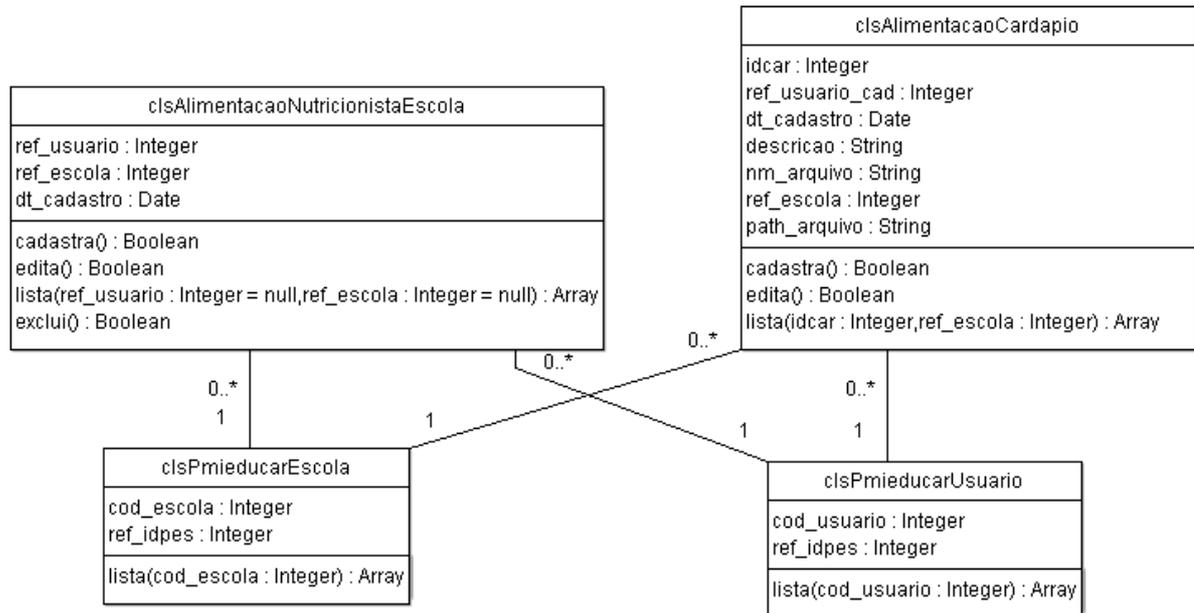


Fig. 4.3 – Diagramas de Classes Cardápios

O último diagrama de classes (Fig. 4.4), apresenta as classes relacionadas ao cadastro do IMC para os alunos.

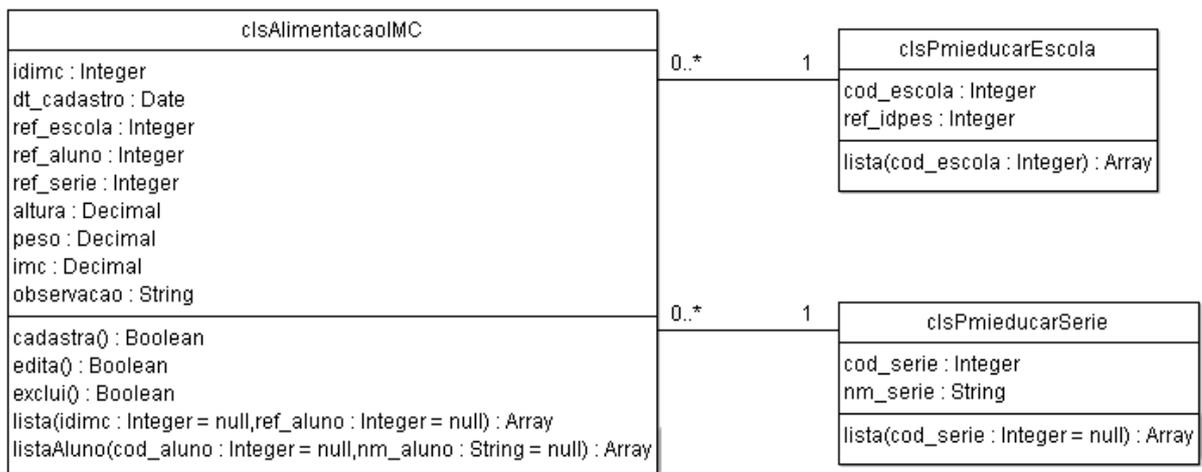


Fig. 4.4 – Diagrama de Classes IMC

## 4.6 Diagramas de Banco de Dados

O diagrama de banco de dados da figura abaixo (Fig. 4.5), apresenta todas as tabelas e atributos e seus relacionamentos. As tabelas precedidas de “alimentacao.” foram criadas como parte deste projeto dentro do i-educar. Já as outras tabelas já faziam parte do sistema i-educar, mas foram utilizadas na criação deste módulo dentro do i-educar. As tabelas, que já faziam parte do sistema, serão apresentadas somente com os atributos e relacionamentos que serão utilizados na criação deste módulo. A documentação detalhada da estrutura da base deste módulo está disponível no Anexo C.

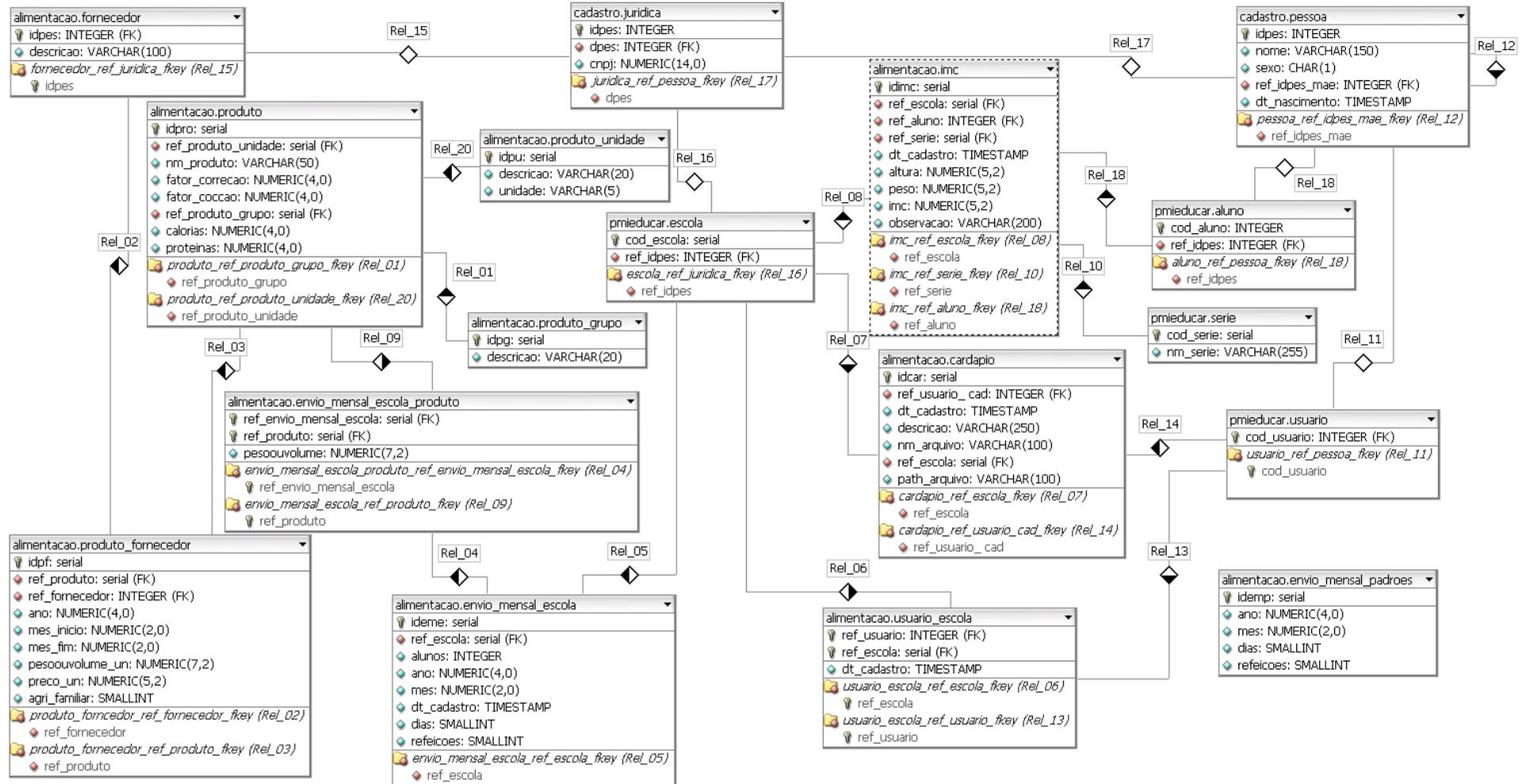


Fig. 4.5 – Diagrama do Banco de Dados

## 4.7 Projeto

Após o levantamento de requisitos e o planejamento do módulo de acompanhamento da merenda escolar, iniciamos o desenvolvimento deste módulo no sistema i-educar. Esta fase foi dividida em três etapas: criação da base de dados, criação das classes e criação das interfaces. A seguir apresentamos mais detalhes de cada uma delas.

### 4.7.1 Criação da Base de Dados

A criação da base de dados foi realizada facilmente com base nos scripts sql gerados a partir do diagrama de banco de dados. Como o PostgreSQL permite a criação de schemas para agrupamento de tabelas e funcionalidades. Como o sistema i-educar possui diversos schemas, criamos o schema “alimentacao” para agrupar as tabelas do módulo que estamos criando. As tabelas, chaves, constraints e incrementos foram criadas utilizando o pgAdmin III. A estrutura das tabelas criadas está disponível no anexo C, sendo que somente as tabelas do schema “alimentacao” foram criadas, pois as outras tabelas já faziam parte do sistema.

### 4.7.2 Criação das Classes

A criação das classes foi simples e rápida. Como você pode verificar através do diagrama de classes, todas as classes criadas apresentam métodos padrões, este padrão se deve ao fato das classes de interface interagirem sempre com os mesmos métodos. No entanto, outros métodos podem ser criados e foram necessários em algumas classes.

### 4.7.3 Criação das Interfaces

A criação da interface foi a etapa mais trabalhosa da criação do sistema. A interface dentro do i-educar é gerada por classes já existentes no sistema. A criação de formulários, listagens e classes é toda padronizada, gerando assim algumas limitações na criação das telas.

Todas as interfaces do sistemas herdam a classe `clsBase` que é responsável pela criação layout principal da página e também realiza o controle de permissões. As interfaces do i-educar foram definidas em tela de pesquisa ou listagem, tela de detalhes e tela de cadastro. Sempre que desejamos criar uma destas telas, devemos criar uma classe que herde características da sua classe pai, ou seja, para criar uma tela de listagem herdamos a classe `clsListagem`, tela de detalhes herdamos a `clsDetalhe` e cadastro a classe `clsCadastro`. Todas estas telas podem ser compostas de campos que são adicionados para pesquisa, cadastro ou somente exibição. Estes campos são gerados pela classe `clsCampos` e possuem todos os campos padrões do html e alguns padrões de exibição criamos para o sistema. As classes herdadas de `clsListagem` e `clsDetalhe` interagem com o método `lista` que estão disponíveis nas classes de negócios já criadas, a classe `clsCadastro` acessa os métodos `lista`, `cadastra`, `edita` e `exclui`.

Estas características de interface do sistema, permitem o desenvolvimento de novas funcionalidades sem a necessidade da criação de layout html, permitindo assim que programadores incluam novas telas ao sistema.

#### 4.7.4 Dificuldades Encontradas

A maior dificuldade encontrada foi a compreensão da estrutura de classes e base de dados do sistema, pois é bem vasta e não possui documentação. A maior dificuldade foi na criação da interface do sistema, pois foi necessário estudar todas as classes para saber quais campos de formulários existiam e como eles eram criados e seus parâmetros. Porém este modelo já criado permitiu a manutenção nos padrões de layout e funcionamento do sistema.

## 5 SISTEMA DESENVOLVIDO

### 5.1 Funcionalidades e Telas

O sistema desenvolvido é composto basicamente de 3 grupos de funcionalidades, que serão descritos abaixo:

### **Controle de cardápios**

Esta funcionalidade permite gerenciar os cardápios das escolas, através do cadastro do arquivo com o cardápio da escola. Pode ser cadastrado pelo administrador ou por um usuário nutricionista. As escolas que o usuário nutricionista tem permissão são definidas pelo administrador através da opção “Nutricionista Escolas” e o cadastro dos cardápios é realizado pela opção “Cardápio” disponíveis no menu “i-Educar Alimentação”.

### **Envio mensal de alimentos**

Esta funcionalidade tem o objetivo de registrar a quantidade de alimentos enviada para cada escola. Para registrar as quantidades enviadas, inicialmente é necessário realizar o cadastro dos produtos, opção “Produtos”, depois cadastrar a lista de fornecedores, opção “Fornecedores”, depois cadastrar quais produtos são fornecidos por cada fornecedor e suas informações como peso da embalagem e custo, através da opção “Produto Fornecedor”. Após definidas estas informações pode ser lançada as quantidades enviadas mensalmente para cada escola. Os relatórios “Resumo por Escola” e “Resumo por Mês” permitem acompanhar informações como: custo, custo por refeição, calorias diárias e proteínas diárias por aluno e custo com agricultura familiar.

### **Cadastro do IMC**

O cadastro do Índice de Massa Corporal é realizado através do cadastro do peso e altura do aluno. Esta inclusão deve ser realizado pela opção “IMC” e o acompanhamento pode ser realizado pelos relatórios “IMC por Escola” e “IMC por Série”.

Como já apresentamos anteriormente, o sistema i-educar possui uma estrutura de cadastrado organizada em tela de pesquisa, detalhes e inclusão e alteração. Como todas as telas do sistema seguem o mesmo padrão e visando apresentar como ficou as telas do módulo desenvolvido, iremos mostrar abaixo os 3 modelos de telas utilizando o cadastro de produtos como exemplo, verifique os requisitos funcionais para ver todas as telas e seus atributos.

**Filtros de busca**

Grupo

Produto - Listagem

Produto	Grupo	Fator Correção	Fator Cocção
Arroz	Menos Perecíveis	100	150
Bolacha rosquinha de chocolate	Menos Perecíveis	100	100
Cenoura	Mais Perecíveis	90	100
Feijão Preto	Menos Perecíveis	100	130
Tomate	Mais Perecíveis	80	100

Fig. 5.1 – Tela de pesquisa de produtos

**Produto - Detalhe**

Produto: Arroz

Fator Correção: 100

Fator Cocção: 150

Grupo: Menos Perecíveis

Calorias(Kcal)/100gr ou ml: 240

Proteínas (gramas)/100gr ou ml: 14

Fig. 5.2 – Tela de detalhes do produto

**Editar**

Produto \*

Fator Correção \*   
somente números

Fator Cocção \*   
somente números

Grupo \*

Calorias(Kcal) \*   
Quantidade há cada 100 gramas ou ml

Proteínas(gramas) \*   
Quantidade há cada 100 gramas ou ml

Fig. 5.3 – Tela de inclusão e alteração de produtos

## 5.2 Implantação do Módulo de Alimentação Escolar

Para incluir o módulo de acompanhamento de merenda escolar você já deve possuir o i-Educar instalado e configurado, mais informações sobre e como instalar o i-Educar acesse <http://svn.softwarepublico.gov.br/trac/ieducar/wiki>.

Após a instalação do i-Educar estar completa, você deve executar a configuração da base de dados através do SQL disponível no Anexo D. Atenção: neste script sql na última linha está a configuração do search\_path, ela deve estar igual ao search\_path atual, da sua instalação, mais a propriedade “,alimentacao”. Este SQL já inclui os menus e submenus no sistema.

Com a base de dados configurada e os menus já definidos, devemos apenas incluir as classes e arquivos necessários para o funcionamento do módulo de alimentação escolar, conforme a estrutura de arquivos da figura abaixo (Fig. 5.4). Com estas configurações realizadas, você necessita apenas das permissões para acessar este módulo pelo menu, para realizar as permissões é só acessar o cadastro de funcionários e selecionar o usuário que você deseja permitir o acesso.

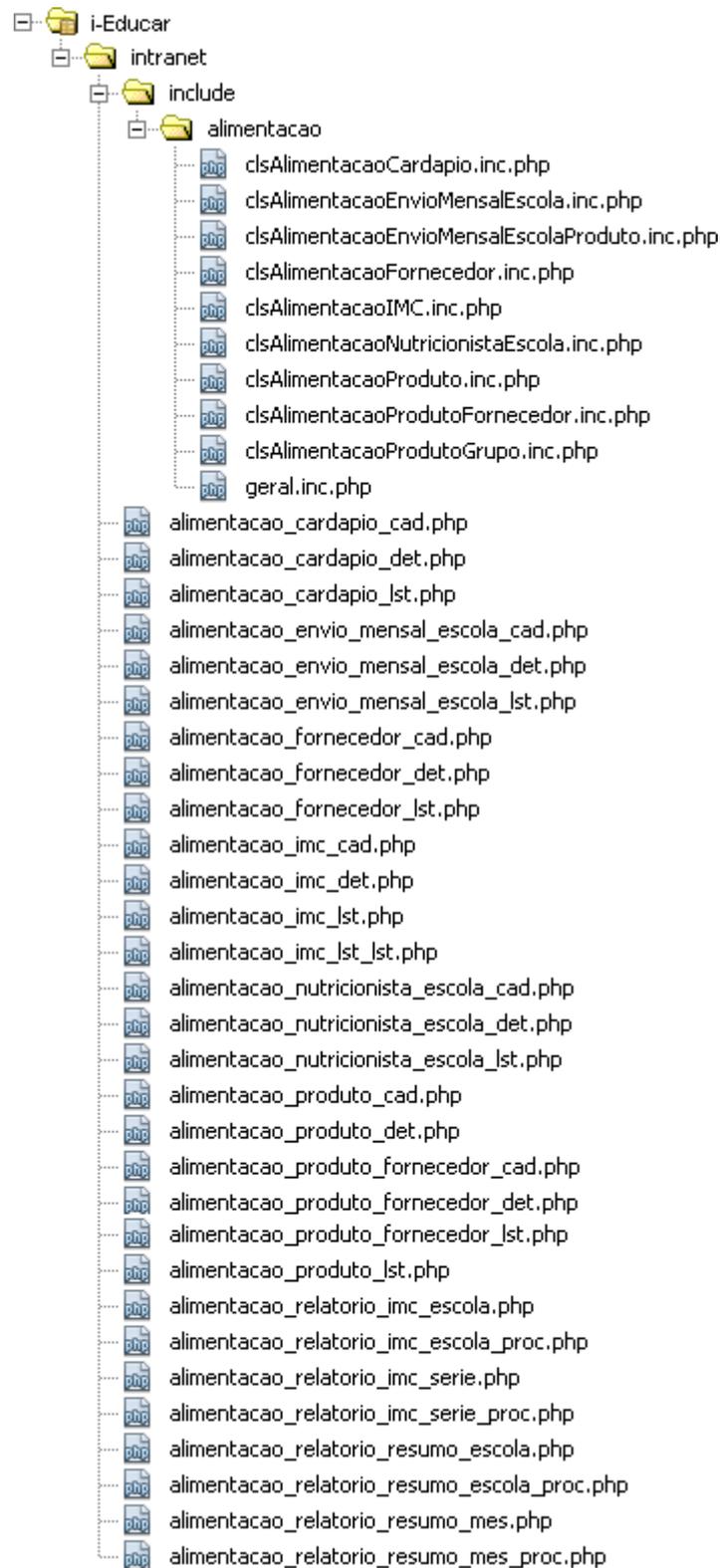


Fig. 5.4 – Estrutura de arquivos do módulo de alimentação escolar

## 6 CONCLUSÃO

Após a finalização do módulo, junto ao sistema i-educar, para acompanhamento da merenda escolar em instituições de ensino das prefeituras, podemos apresentar as seguintes conclusões:

O levantamento dos processos e requisitos, relacionados a alimentação escolar, foi realizado com sucesso na Prefeitura Municipal de Florianópolis e foi fundamental na obtenção das metas na fase de análise.

A criação deste trabalho, permitiu a utilização de muitas funcionalidades já existentes no sistema i-educar e colaborou para o aprimoramento deste software livre.

O módulo desenvolvido alcançou os objetivos planejados e está apto para ser utilizado via WEB nas redes de ensino municipais juntamente com o i-educar, auxiliando no controle produtos, fornecedores, na qualidade das refeições servidas, no acompanhamento dos custos e no índice de massa corporal dos alunos.

Os benefícios esperados são o controle das necessidades nutricionais diárias do alunos, que pode ser aferida através das calorias e proteínas diárias fornecidas por aluno. O acompanhamento do consumo mensal nas escolas, visando mudanças anormais no consumo mensal que podem ocorrer após alterações nos cardápios com pouca receptividade dos alunos. Controle dos custos e valores nutricionais dos cardápios, buscando assim o melhor custo benefício em relação a qualidade da merenda. E também o acompanhamento da massa corporal dos alunos, com o objetivo de traçar um paralelo com os valores nutricionais e alimentos fornecidos.

Com base no conhecimento adquirido, sugere-se como trabalho futuro a implantação e o acompanhamento dos benefícios esperados deste sistema em uma rede de ensino.

## 7 REFERÊNCIAS BIBLIOGRÁFICAS

FNDE. **Alimentação Escolar**, 2009. Disponível em:

<[http://www.fnde.gov.br/home/index.jsp?arquivo=alimentacao\\_escolar.html](http://www.fnde.gov.br/home/index.jsp?arquivo=alimentacao_escolar.html)>.

Acessado em 25 de junho de 2009

KDE. **Elementos UML - Diagrama de Caso de Uso**, 2009. Disponível em:

<[http://docs.kde.org/stable/pt\\_BR/kdesdk/umbrello/uml-elements.html](http://docs.kde.org/stable/pt_BR/kdesdk/umbrello/uml-elements.html)>. Acessado

em 24 de junho de 2009

MASCARENHAS Jean Márcia Oliveira; SANTOS, Juliana Cantalino dos.

AVALIAÇÃO DA COMPOSIÇÃO NUTRICIONAL DOS CARDÁPIOS E CUSTOS DA ALIMENTAÇÃO ESCOLAR DA REDE MUNICIPAL DE CONCEIÇÃO DO JACUÍPE/BA. **Sitientibus**, Feira de Santana, n.35, p.75-90, jul./dez. 2006.

SoftwarePúblico. **i-educar**, 2009. Disponível em:

<<http://svn.softwarepublico.gov.br/trac/ieducar/wiki/Sobre>>. Acessado em 02 de outubro de 2009.

PostgreSQL. **Sobre o PostgreSQL**, 2009. Disponível em:

<<http://www.postgresql.org.br/sobre>>. Acessado em 02 outubro de 2009.

NUNES, Bruno. **Análise de Requisitos: Funcionais x Não Funcionais**. Postado em Terça-feira, Maio 22, 2007. Disponível em:

<<http://maozinhadaweb.blogspot.com/2007/05/anlise-de-requisitos-funcionais-x-no.html>>. Acessado em 24 de junho de 2009.

PINHEIRO, Dra. Ana Flávia. **Portal do Coração - Alimentos energéticos , construtores e reguladores**, 2008. Disponível em:

<<http://portaldocoracao.uol.com.br/nutricao.php?id=1825>>. Acessado em 25 de junho de 2009.

PMF. **Alimentação escolar de Florianópolis é municipalizada**, 2009. Disponível em: <[http://www.pmf.sc.gov.br/portal/noticias/comunicacao/?acao=listar&id\\_noticia=5443](http://www.pmf.sc.gov.br/portal/noticias/comunicacao/?acao=listar&id_noticia=5443)>.

Acessado em 25 de junho de 2009

PMF. **Coordenadoria de alimentação escolar**, 2009. Disponível em: <[http://www.pmf.sc.gov.br/educa/recursos\\_alimentacao.htm](http://www.pmf.sc.gov.br/educa/recursos_alimentacao.htm)>. Acessado em 25 de junho de 2009

SILVA, Bruno Felipe da. **Uma aplicação JEE para votações acadêmicas**. Florianópolis: UFSC, 2008. 135f. Trabalho de conclusão de curso para bacharel em Sistemas de Informação, Universidade Federal de Santa Catarina, Florianópolis, 2008.

VIADIGITAL. **O Caminho inteligente para informatização pública**, 2009. Disponível em: <[http://www.viadigital.org.br/index.php?option=com\\_content&task=view&id=3&Itemid=20](http://www.viadigital.org.br/index.php?option=com_content&task=view&id=3&Itemid=20)>. Acessado em 25 de junho de 2009

WIKIPEDIA. **Desenvolvimento ágil de software**, 2009. Disponível em: <[http://pt.wikipedia.org/wiki/Desenvolvimento\\_%C3%A1gil\\_de\\_software#M.C3.A9todos\\_.C3.A1geis\\_e\\_o\\_gerenciamento\\_de\\_projetos](http://pt.wikipedia.org/wiki/Desenvolvimento_%C3%A1gil_de_software#M.C3.A9todos_.C3.A1geis_e_o_gerenciamento_de_projetos)>. Acessado em 24 de junho de 2009

WIKIPEDIA. **UML**, 2009. Disponível em: <<http://pt.wikipedia.org/wiki/UML>>. Acessado em 24 de junho de 2009

WIKIPEDIA. **MySQL**, 2009. Disponível em: <<http://pt.wikipedia.org/wiki/MySQL>>. Acessado em 26 de junho de 2009

WIKIPEDIA. **Software Livre**, 2009. Disponível em: <[http://pt.wikipedia.org/wiki/Software\\_livre](http://pt.wikipedia.org/wiki/Software_livre)>. Acessado em 26 de junho de 2009

## 8 ANEXOS

### 8.1 Anexo A - Requisitos Funcionais

Os requisitos podem ser classificados em necessário e desejável.

Os atores do sistema são: Administrador, Nutricionista.

**[RF01] – Incluir Cardápio**

Ator: Administrador, Nutricionista

Este requisito inicia quando o ator deseja incluir um novo cadastro.

Campos do formulário:

<b>Campo</b>	<b>Descrição</b>	<b>Obrigatório</b>
Descrição	Pequena descrição do cardápio	Sim
Arquivo	Documento contendo a planilha do cardápio.	Sim
Escola	Nome da escola. Campo de seleção. Somente serão mostradas as escolas que o usuário tem permissão de acesso.	Sim
Usuário	Usuário que está cadastrado. Campo oculto.	Sim
Data Cadastro	Data do cadastro. Campo oculto.	Sim

Classificação: Necessário

**[RF02] – Pesquisar Cardápios**

Ator: Administrador, Nutricionista

Este requisito inicia quando o ator deseja listar os cadastros.

Filtros:

<b>Campo</b>	<b>Descrição</b>	<b>Obrigatório</b>
Escola	Nome da escola. Usuário só pode ver as escolas com permissão.	Não

Retorno:

<b>Campo</b>	<b>Descrição</b>
--------------	------------------

Descrição	Pequena descrição do cardápio
Arquivo	Documento contendo a planilha do cardápio.
Escola	Nome da escola.
Usuário	Nome do usuário que realizou o cadastro.
Data Cadastro	Data do cadastro.

Classificação: Necessário

### **[RF03] – Detalhar Cardápio**

Ator: Administrador, Nutricionista

Este requisito inicia quando o ator deseja ver os dados de um cadastro. O ator deve selecionar o cadastro (através do requisito [RF03]).

Detalhes:

<b>Campo</b>	<b>Descrição</b>
Descrição	Pequena descrição do cardápio
Arquivo	Documento contendo a planilha do cardápio.
Escola	Nome da escola.
Usuário	Nome do usuário que realizou o cadastro.
Data Cadastro	Data do cadastro.

Classificação: Necessário

### **[RF04] – Alterar Cardápio**

Ator: Administrador e Nutricionista

Este requisito inicia quando o ator deseja alterar um cadastro. O ator deve clicar em editar (através do requisito [RF03]).

Campos do formulário:

<b>Campo</b>	<b>Descrição</b>	<b>Obrigatório</b>
Descrição	Pequena descrição do cardápio	Sim
Escola	Nome da escola. Campo de seleção. Somente serão mostradas as escolas que o usuário tem permissão de acesso.	Sim

Classificação: Necessário

### **[RF05]- Incluir Nutricionista Escola**

Ator: Administrador

Este requisito inicia quando o ator deseja incluir um novo cadastro.

Campos do formulário:

<b>Campo</b>	<b>Descrição</b>	<b>Obrigatório</b>
Escola	Nome da escola. Campo de seleção.	Sim
Nutricionista	Nome do usuário. Campo de seleção.	Sim

Classificação: Necessário.

### **[RF06]- Pesquisar Nutricionistas Escola**

Ator: Administrador

Este requisito inicia quando o ator deseja listar os cadastros.

Filtros:

<b>Campo</b>	<b>Descrição</b>	<b>Obrigatório</b>
Escola	Nome da escola. Campo de seleção.	Não
Nutricionista	Nome do usuário. Campo de seleção.	Não

Retorno:

<b>Campo</b>	<b>Descrição</b>
Escola	Nome da escola.
Nutricionista	Nome do usuário.
Data Cadastro	Data do cadastro.

Classificação: Necessário.

### **[RF07]- Detalhar Nutricionistas Escola**

Ator: Administrador

Este requisito inicia quando o ator deseja ver os dados de um cadastro. O ator deve selecionar o cadastro (através do requisito [RF06]).

Detalhes:

<b>Campo</b>	<b>Descrição</b>
Escola	Nome da escola.
Nutricionista	Nome do usuário.
Data Cadastro	Data do cadastro.

### **[RF08]- Alterar Nutricionista Escola**

Ator: Administrador

Este requisito inicia quando o ator deseja alterar o cadastro. O ator deve clicar em editar (através do requisito [RF07]).

Campos do formulário:

<b>Campo</b>	<b>Descrição</b>	<b>Obrigatório</b>
Escola	Nome da escola. Campo de seleção.	Sim
Nutricionista	Nome do usuário. Campo de seleção.	Sim

Classificação: Necessário.

### **[RF09]- Remover Nutricionista Escola**

Ator: Administrador

Este requisito inicia quando o ator deseja remover um cadastro. O ator deve clicar em excluir (através do requisito [RF08]).

Observação: O Ator deverá confirmar a remoção do cadastro.

Classificação: Necessário.

### **[RF10]- Incluir Produto**

Ator: Administrador

Este requisito inicia quando o ator deseja incluir um novo cadastro.

Campos do formulário:

<b>Campo</b>	<b>Descrição</b>	<b>Obrigatório</b>
Produto	Nome do produto	Sim
Fator Correção	Porcentagem que é aproveitada.	Sim
Fator Cocção	Porcentagem de ganho ou perda após cozimento.	Sim
Grupo	Menos Perecível, Mais Perecível ou Não Perecível. Campo de seleção.	Sim
Calorias	Calorias a cada 100gr/ml naturalmente.	Sim
Proteínas	Proteínas a cada 100gr/ml naturalmente.	Sim

Classificação: Necessário.

### **[RF11]- Pesquisar Produtos**

Ator: Administrador

Este requisito inicia quando o ator deseja listar os cadastros.

Filtros:

Campo	Descrição	Obrigatório
Grupo	Nome do grupo. Campo de seleção.	Sim

Retorno:

Campo	Descrição
Produto	Nome do produto
Grupo	Nome do grupo.
Fator Correção	Porcentagem.
Fator Cocção	Porcentagem.

Classificação: Necessário.

### [RF12]- Detalhar Produto

Ator: Administrador

Este requisito inicia quando o ator deseja ver os dados de um cadastro. O ator deve selecionar o cadastro (através do requisito [RF11]).

Detalhes:

Campo	Descrição
Produto	Nome do produto
Grupo	Nome do grupo.
Fator Correção	Porcentagem.
Fator Cocção	Porcentagem.
Calorias	Valor.
Proteínas	Valor.

Classificação: Necessário.

**[RF13]- Alterar Produto**

Ator: Administrador

Este requisito inicia quando o ator deseja alterar um cadastro. O ator deve selecionar um cadastro (através do requisito [RF10]).

Campos do formulário:

<b>Campo</b>	<b>Descrição</b>	<b>Obrigatório</b>
Produto	Nome do produto	Sim
Fator Correção	Porcentagem que é aproveitada.	Sim
Fator Cocção	Porcentagem de ganho ou perda após cozimento.	Sim
Grupo	Menos Perecível, Mais Perecível ou Não Perecível. Campo de seleção.	Sim
Calorias	Calorias a cada 100gr/ml naturalmente.	Sim
Proteínas	Proteínas a cada 100gr/ml naturalmente.	Sim

Classificação: Necessário.

**[RF14]- Incluir Fornecedor**

Ator: Administrador

Este requisito inicia quando o ator deseja incluir um novo cadastro.

Campos do formulário:

<b>Campo</b>	<b>Descrição</b>	<b>Obrigatório</b>
CNPJ	CNPJ do fornecedor. Já deve estar cadastro no sistema como pessoa jurídica.	Sim
Descrição	Uma pequena descrição.	Sim

Classificação: Necessário.

**[RF15]- Pesquisar Fornecedor**

Ator: Administrador

Este requisito inicia quando o ator deseja listar os cadastros.

Filtros:

Campo	Descrição	Obrigatório
CNPJ	CNPJ do fornecedor.	Não

Retorno:

Campo	Descrição
Fornecedor	Nome do fornecedor.
CNPJ	CNPJ do fornecedor.
Descrição	Descrição.

Classificação: Necessário.

### **[RF16]- Detalhar Fornecedor**

Ator: Administrador

Este requisito inicia quando o ator deseja ver os dados de um cadastro. O ator deve selecionar um cadastro (através do requisito [RF15]).

Detalhes:

Campo	Descrição
Fornecedor	Nome do fornecedor.
CNPJ	CNPJ do fornecedor.
Descrição	Descrição.

Classificação: Necessário.

### **[RF17]- Alterar Fornecedor**

Ator: Administrador

Este requisito inicia quando o ator deseja alterar um cadastro. O ator deve clicar em editar (através do requisito [RF16]).

Campos do formulário:

Campo	Descrição	Obrigatório
Descrição	Uma pequena descrição.	Sim

Classificação: Necessário.

### **[RF18] – Incluir Produto Fornecedor**

Ator: Administrador

Este requisito inicia quando o ator deseja incluir um novo cadastro.

Campos do formulário:

Campo	Descrição	Obrigatório
Produto	Nome do produto. Campo de seleção.	Sim
Fornecedor	Nome do fornecedor. Campo de seleção.	Sim
Ano	Ano do fornecimento.	Sim
Mês - Início	Mês de início do fornecimento.	Sim
Mês - Fim	Mês do fim do fornecimento.	Sim
Peso(Kg) ou Volume(L) Un.	Peso ou volume por unidade.	Sim
Preço Un.	Preço por unidade.	Sim
Agricultura Familiar	Sim ou Não. Produto de agricultura familiar. Campo de seleção.	Sim

Classificação: Necessário

### **[RF19] – Pesquisar Produto Fornecedor**

Ator: Administrador

Este requisito inicia quando o ator deseja listar os cadastros.

Filtros:

Campo	Descrição	Obrigatório
Produto	Nome do produto. Campo de seleção.	Não
Fornecedor	Nome do fornecedor. Campo de seleção.	Não

Retorno:

Campo	Descrição
Produto	Nome do produto
Fornecedor	Nome do fornecedor.
Ano	Ano do fornecimento.
Mês - Início	Mês de início do fornecimento.
Mês - Fim	Mês do fim do fornecimento.
Peso(Kg) ou Volume(L) Un.	Peso ou volume por unidade.
Preço Un.	Preço por unidade.

Classificação: Necessário

### **[RF20] – Detalhar Produto Fornecedor**

Ator: Administrador

Este requisito inicia quando o ator deseja ver os dados de um cadastro. O ator dele selecionar um cadastro (através do requisito [RF19]).

Detalhes:

<b>Campo</b>	<b>Descrição</b>
Produto	Nome do produto
Fornecedor	Nome do fornecedor.
Ano	Ano do fornecimento.
Mês - Início	Mês de início do fornecimento.
Mês - Fim	Mês do fim do fornecimento.
Peso(Kg) ou Volume(L) Un.	Peso ou volume por unidade.
Preço Un.	Preço por unidade.
Agricultura Familiar	Sim ou Não.

Classificação: Necessário

### **[RF21] – Alterar Produto Fornecedor**

Ator: Administrador

Este requisito inicia quando o ator deseja alterar um cadastro. O ator deve clicar em editar (através do requisito [RF20]).

Campos do formulário:

<b>Campo</b>	<b>Descrição</b>	<b>Obrigatório</b>
Produto	Nome do produto. Campo de seleção.	Sim
Fornecedor	Nome do fornecedor. Campo de seleção.	Sim
Ano	Ano do fornecimento.	Sim
Mês - Início	Mês de início do fornecimento.	Sim
Mês - Fim	Mês do fim do fornecimento.	Sim
Peso(Kg) ou Volume(L) Un.	Peso ou volume por unidade.	Sim
Preço Un.	Preço por unidade.	Sim
Agricultura Familiar	Sim ou Não. Produto de agricultura familiar.	Sim

	Campo de seleção.	
--	-------------------	--

Classificação: Necessário

### [RF22] – Remover Produto Fornecedor

Ator: Administrador

Este requisito inicia quando o ator deseja remover um cadastro. O ator deve clicar em excluir (através do requisito [RF21]).

Observação: O Ator deverá confirmar a remoção do cadastro.

Classificação: Necessário

### [RF23] – Incluir Envio Mensal

Ator: Administrador

Este requisito inicia quando o ator deseja incluir um novo cadastro.

Campos do formulário:

Campo	Descrição	Obrigatório
Escola	Nome da escola. Campo de seleção.	Sim
Ano	Ano do envio.	Sim
Mês	Mês do envio.	Sim
Alunos	Número de alunos em período integral. Escolas com meio período é realizada uma proporção.	Sim
Peso(Kg) ou Volume(L)	Cadastrado a quantidade enviada no mês para cada produto.	Sim
Data Cadastro	Data do cadastro. Campo oculto.	Sim
Dias	Quantidade de dias de aula no mês	Sim
Refeições	Quantidade de refeições diárias	Sim

Classificação: Necessário

### **[RF24] – Pesquisar Envio Mensal**

Ator: Administrador

Este requisito inicia quando o ator deseja listar os cadastros.

Filtros:

<b>Campo</b>	<b>Descrição</b>	<b>Obrigatório</b>
Escola	Nome da escola. Campo de seleção.	Não
Ano	Ano. Campo de seleção.	Não
Mês	Mês. Campo de seleção.	Não

Retorno:

<b>Campo</b>	<b>Descrição</b>
Escola	Nome da escola.
Ano	Ano do envio.
Mês	Mês do envio.
Alunos	Número de alunos em período integral. Escolas com meio período é realizada uma proporção.
Data Cadastro	Data do cadastro.

Classificação: Necessário

### **[RF25] – Detalhar Envio Mensal**

Ator: Administrador

Este requisito inicia quando o ator deseja ver os dados de um cadastro. O ator deve selecionar o cadastro (através do requisito [RF24]).

Detalhes:

<b>Campo</b>	<b>Descrição</b>
Escola	Nome da escola.
Ano	Ano do envio.
Mês	Mês do envio.
Alunos	Número de alunos em período integral. Escolas com meio período é realizada uma proporção.
Data Cadastro	Data do cadastro.
Peso(Kg) ou Volume(L)	Quantidade enviada no mês para cada produto.
Dias	Quantidade de dias de aula no mês
Refeições	Quantidade de refeições diárias

Classificação: Necessário

### **[RF26] – Alterar Envio Mensal**

Ator: Administrador

Este requisito inicia quando o ator deseja alterar um cadastro. O ator deve clicar em editar (através do requisito [RF25]).

Campos do formulário:

<b>Campo</b>	<b>Descrição</b>	<b>Obrigatório</b>
Escola	Nome da escola. Campo de seleção.	Sim
Ano	Ano do envio.	Sim
Mês	Mês do envio.	Sim
Alunos	Número de alunos em período integral. Escolas com meio período é realizada uma proporção.	Sim
Peso(Kg) ou Volume(L)	Cadastrado a quantidade enviada no mês para cada produto.	Sim
Data Cadastro	Data do cadastro. Campo oculto.	Sim

Dias	Quantidade de dias de aula no mês	Sim
Refeições	Quantidade de refeições diárias	Sim

Classificação: Necessário

### **[RF27] – Remover Envio Mensal**

Ator: Administrador

Este requisito inicia quando o ator deseja remover um cadastro. O ator deve clicar em excluir (através do requisito [RF26]).

Observação: O Ator deverá confirmar a remoção do cadastro.

Classificação: Necessário

### **[RF28] – Pesquisar IMC Alunos**

Ator: Administrador e Nutricionista

Este requisito inicia quando o ator deseja listar os alunos cadastrados.

Filtros:

<b>Campo</b>	<b>Descrição</b>	<b>Obrigatório</b>
Nome	Nome do aluno.	Sim

Retorno:

<b>Campo</b>	<b>Descrição</b>
Aluno	Nome do aluno.
Nome da mãe	Nome da mãe.
Data de Nascimento	Data de nascimento do aluno.
Sexo	Sexo do aluno.

Classificação: Necessário

### [RF29] – Incluir IMC

Ator: Administrador e Nutricionista

Este requisito inicia quando o ator deseja incluir um novo cadastro. O ator deve selecionar um aluno (através do requisito [RF28]), e clicar em novo.

Campos do formulário:

Campo	Descrição	Obrigatório
Aluno	Nome do aluno. Campo passado como parâmetro.	Sim
Escola	Nome da escola. Campo de seleção.	Sim
Série	Série do aluno atualmente.	Sim
Altura(Mt)	Altura do aluno em metros.	Sim
Peso(kg)	Peso do aluno em quilogramas.	Sim
IMC	Campo calculado com base na altura e peso. Cálculo: $\text{Peso} / \text{altura}^2$ .	Sim
Data Cadastro	Data do cadastro. Campo oculto.	Sim
Observações	Observações	Não

Classificação: Necessário

### [RF30] – Pesquisar IMC

Ator: Administrador e Nutricionista

Este requisito inicia quando o ator deseja listar os cadastros.

Filtros:

Campo	Descrição	Obrigatório
-------	-----------	-------------

Nome	Nome do aluno. Passado como parâmetro.	Sim
------	--	-----

Retorno:

Campo	Descrição
Escola	Nome da escola.
Série	Série do aluno atualmente.
Altura(Mt)	Altura do aluno em metros.
Peso(kg)	Peso do aluno em quilogramas.
IMC	IMC
Data Cadastro	Data do cadastro.

Classificação: Necessário

### [RF31] – Detalhar IMC

Ator: Administrador e Nutricionista

Este requisito inicia quando o ator deseja ver os dados do cadastro. O ator deve selecionar o cadastro (através do requisito [RF32]).

Detalhes:

Campo	Descrição
Aluno	Nome do aluno.
Escola	Nome da escola.
Série	Série do aluno atualmente.
Altura(Mt)	Altura do aluno em metros.
Peso(kg)	Peso do aluno em quilogramas.
IMC	IMC
Data Cadastro	Data do cadastro.

Observações	Observações
-------------	-------------

Classificação: Necessário

### [RF32] – Alterar IMC

Ator: Administrador e Nutricionista

Este requisito inicia quando o ator deseja alterar um cadastro. O ator deve clicar em editar (através do requisito [RF31]).

Campos do formulário:

Campo	Descrição	Obrigatório
Aluno	Nome do aluno. Campo passado como parâmetro.	Sim
Escola	Nome da escola. Campo de seleção.	Sim
Série	Série do aluno atualmente.	Sim
Altura(Mt)	Altura do aluno em metros.	Sim
Peso(kg)	Peso do aluno em quilogramas.	Sim
IMC	Campo calculado com base na altura e peso. Cálculo: $\text{Peso} / \text{altura}^2$ .	Sim
Data Cadastro	Data do cadastro. Campo oculto.	Sim
Observações	Observações	Não

Classificação: Necessário

### [RF33] – Remover IMC

Ator: Administrador e Nutricionista

Este requisito inicia quando o ator deseja remover um cadastro. O ator deve clicar em excluir (através do requisito [RF32]).

Observação: O Ator deverá confirmar a remoção do cadastro.

Classificação: Necessário

### [RF34] – Relatório Resumo por Escolas

Ator: Administrador

Este requisito inicia quando o ator deseja gerar um relatório com um resumo por escolas das informações nutricionais e custos das refeições.

Filtros:

Campo	Descrição	Obrigatório
Ano	Ano. Campo de seleção.	Sim
Mês	Nome do mês. Campo de seleção.	Não

Retorno:

Campo	Descrição
Escola	Nome da escola.
Custo total	Valor em reais do custo total para a escola.
Custo por refeição	Custo por refeição.
Calorias diárias/aluno	Calorias diárias em Kcal por aluno.
Proteínas diárias/aluno	Quantidade de proteínas em gramas por aluno.
Agric. Familiar	Custo total com produtos de agricultura familiar e a proporção ao total.

Classificação: Necessário

### [RF35] – Relatório Resumo por Mês

Ator: Administrador

Este requisito inicia quando o ator deseja gerar um relatório com um resumo por mês das informações nutricionais e custos das refeições.

Filtros:

<b>Campo</b>	<b>Descrição</b>	<b>Obrigatório</b>
Ano	Ano. Campo de seleção.	Sim
Escola	Nome da escola. Campo de seleção.	Não

Retorno:

<b>Campo</b>	<b>Descrição</b>
Mês	Nome da mês.
Custo total	Valor em reais do custo total para a escola.
Custo por refeição	Custo por refeição.
Calorias diárias/aluno	Calorias diárias em Kcal por aluno.
Proteínas diárias/aluno	Quantidade de proteínas em gramas por aluno.
Agric. Familiar	Custo total com produtos de agricultura familiar e a proporção ao total.

Classificação: Necessário

### **[RF36] – Relatório IMC por Escolas**

Ator: Administrador

Este requisito inicia quando o ator deseja gerar um relatório com o IMC médio por escolas.

Filtros:

<b>Campo</b>	<b>Descrição</b>	<b>Obrigatório</b>
Ano	Ano. Campo de seleção.	Sim
Série	Nome da série. Campo de seleção.	Não
Sexo	Sexo. Campo de seleção.	Não

Retorno:

<b>Campo</b>	<b>Descrição</b>
Escola	Nome da escola.
IMC	Valor do IMC.

Classificação: Necessário

### **[RF37] – Relatório IMC por Série**

Ator: Administrador

Este requisito inicia quando o ator deseja gerar um relatório com o IMC médio por série.

Filtros:

<b>Campo</b>	<b>Descrição</b>	<b>Obrigatório</b>
Ano	Ano. Campo de seleção.	Sim
Escola	Nome da escola. Campo de seleção.	Não
Sexo	Sexo. Campo de seleção.	Não

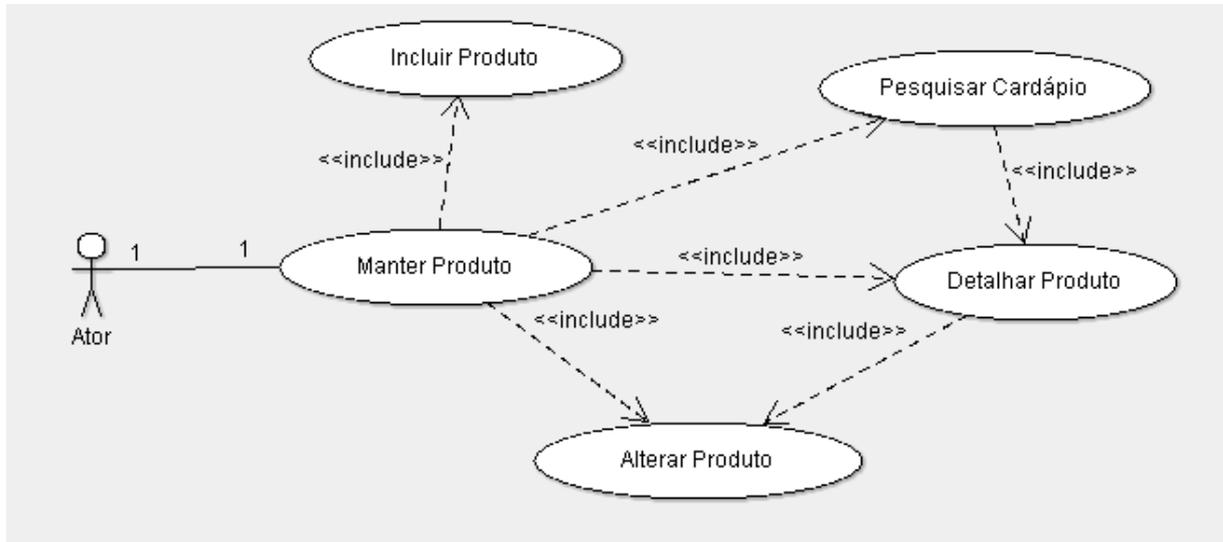
Retorno:

<b>Campo</b>	<b>Descrição</b>
Série	Nome da série.
IMC	Valor do IMC.

Classificação: Necessário

## 8.2 Anexo B – Diagramas de Casos de Uso

### Manter Cardápio



### Incluir Cardápio

Visão geral: Este caso de uso tem a finalidade de incluir um novo cardápio.

Ator: Administrador e Nutricionista

Ação do ator	Resposta do sistema
1. O ator deseja incluir um cardápio	Mostra a tela de inclusão de cardápio
2. O ator informa os dados do cardápio	O sistema valida e grava os dados
Tratamento de exceções	
1.1 O ator é do tipo nutricionista	O sistema só mostra as escolas que o nutricionista tem permissão
2.1 O ator verifica que faltaram dados obrigatórios	O sistema não permite o cadastro

### Pesquisar Cardápio

Visão geral: Este caso de uso tem a finalidade de pesquisar os cardápios cadastros.

Ator: Administrador e Nutricionista

Ação do ator	Resposta do sistema
1. O ator deseja pesquisar um cardápio	Mostra a tela de pesquisa de cardápio

2. O ator pode informar alguns filtros e pesquisar	O sistema realiza a pesquisa
<b>Tratamento de exceções</b>	
2.1 O ator verifica que não retornaram dados	O sistema retorna uma mensagem de aviso

### Detalhar Cardápio

Visão geral: Este caso de uso tem a finalidade de mostrar os dados de um cardápio cadastrado.

Ator: Administrador e Nutricionista

Ação do ator	Resposta do sistema
1. O ator realiza o caso de uso "Listar Cardápio" e seleciona um cardápio cadastrado	O sistema exibe os dados do cardápio selecionado

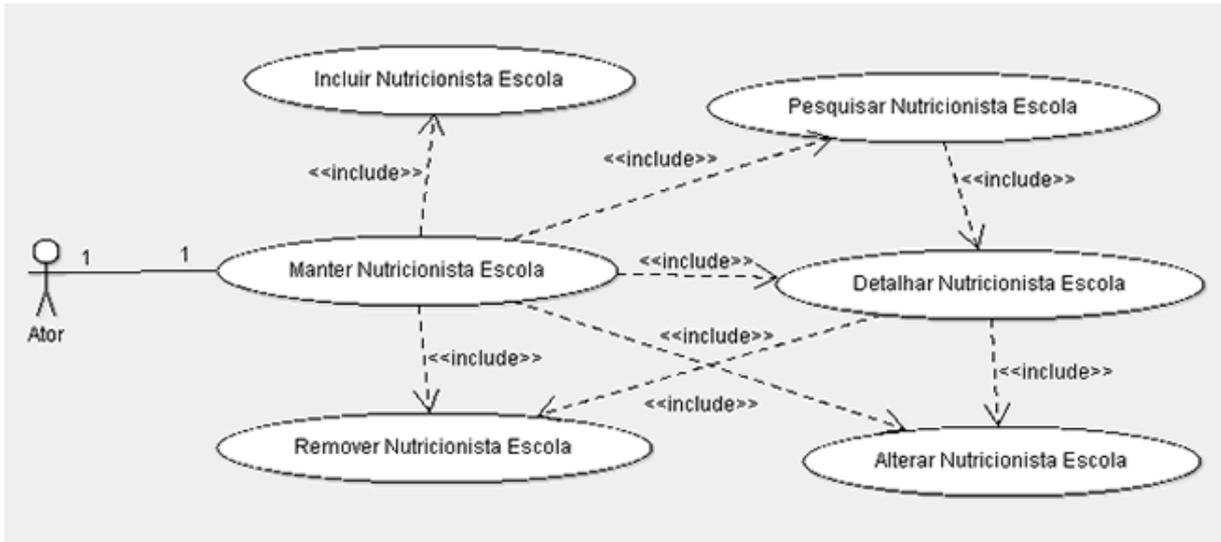
### Alterar Cardápio

Visão geral: Este caso de uso tem a finalidade alterar um cardápio cadastrado.

Ator: Administrador e Nutricionista

Ação do ator	Resposta do sistema
1. O ator realiza o caso de uso "Detalhar Cardápio" e clica em editar	Mostra a tela de alteração de cardápio com base no cadastro selecionado. Não permite alterar o arquivo.
2. O ator informa os dados que deseja alterar	O sistema valida e grava os dados
<b>Tratamento de exceções</b>	
1.1 O ator é do tipo nutricionista	O sistema só mostra as escolas que o nutricionista tem permissão
2.1 O ator verifica que faltaram dados obrigatórios	O sistema não permite o cadastro

### Manter Nutricionista Escola



### Incluir Nutricionista Escola

Visão geral: Este caso de uso tem a finalidade de incluir um nutricionista para uma escola.

Ator: Administrador

Ação do ator	Resposta do sistema
1. O ator deseja incluir um nutricionista	Mostra a tela de inclusão de nutricionista
2. O ator informa os dados	O sistema valida e grava os dados
Tratamento de exceções	
2.1 O ator verifica que o usuário selecionado já está cadastrado como nutricionista da escola informada	O sistema não permite o cadastro

### Pesquisar Nutricionista Escola

Visão geral: Este caso de uso tem a finalidade de pesquisar os nutricionistas cadastros.

Ator: Administrador

Ação do ator	Resposta do sistema
1. O ator deseja pesquisar um nutricionista	Mostra a tela de pesquisa de nutricionista
2. O ator pode informar alguns filtros e pesquisar	O sistema realiza a pesquisa
Tratamento de exceções	
2.1 O ator verifica que não retornaram dados	O sistema retorna uma mensagem de aviso

### Detalhar Nutricionista Escola

Visão geral: Este caso de uso tem a finalidade de mostrar os dados de um nutricionista cadastrado.

Ator: Administrador

Ação do ator	Resposta do sistema
1. O ator realiza o caso de uso "Listar Nutricionista Escola" e seleciona um nutricionista cadastrado	O sistema exibe os dados do nutricionista selecionado

### Alterar Nutricionista Escola

Visão geral: Este caso de uso tem a finalidade alterar um nutricionista cadastrado.

Ator: Administrador

Ação do ator	Resposta do sistema
1. O ator realiza o caso de uso "Detalhar Nutricionista Escola" e clica em editar	Mostra a tela de alteração de nutricionista com base no cadastro selecionado.
2. O ator informa os dados	O sistema valida e grava os dados
Tratamento de exceções	
2.1 O ator verifica que o usuário selecionado já está cadastrado como nutricionista da escola informada	O sistema não permite o cadastro

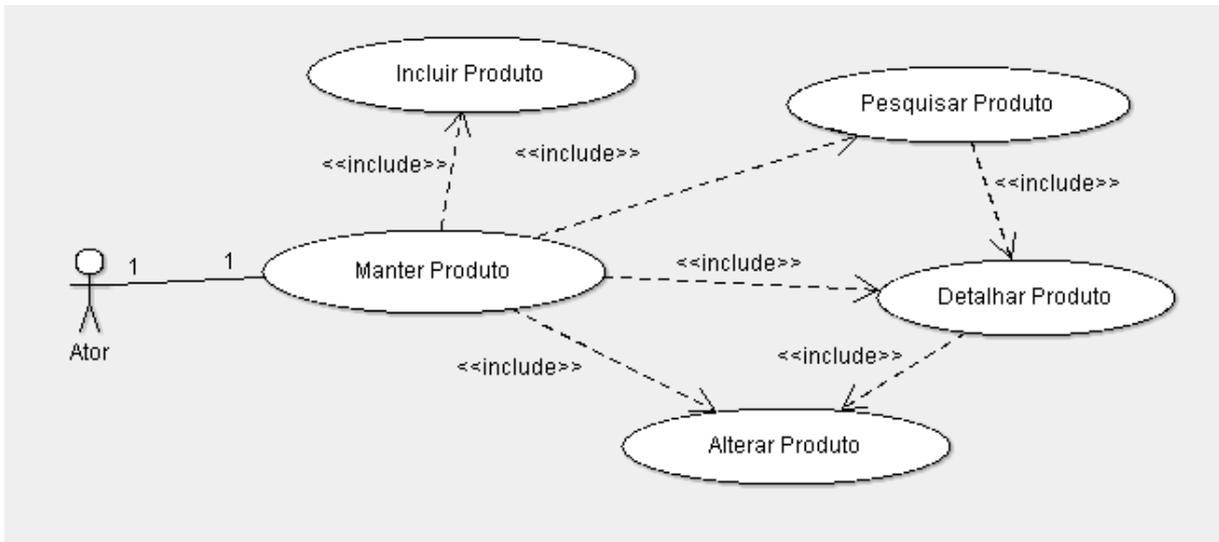
### Remover Nutricionista Escola

Visão geral: Este caso de uso tem a finalidade de remover um nutricionista cadastrado.

Ator: Administrador

Ação do ator	Resposta do sistema
1. O ator realiza o caso de uso "Alterar Nutricionista Escola" e clica em excluir	Mostra a confirmação
2. O ator confirma a exclusão	O sistema realiza a exclusão

## Manter Produto



### Incluir Produto

Visão geral: Este caso de uso tem a finalidade de incluir um novo produto.

Ator: Administrador

Ação do ator	Resposta do sistema
1. O ator deseja incluir um produto	Mostra a tela de inclusão de produto
2. O ator informa os dados do produto	O sistema valida e grava os dados
Tratamento de exceções	
2.1 O ator verifica que faltaram dados obrigatórios	O sistema não permite o cadastro

### Pesquisar Produto

Visão geral: Este caso de uso tem a finalidade de pesquisar os produtos cadastrados.

Ator: Administrador

Ação do ator	Resposta do sistema
1. O ator deseja pesquisar um produto	Mostra a tela de pesquisa de produto
2. O ator pode informar alguns filtros e pesquisar	O sistema realiza a pesquisa
Tratamento de exceções	
2.1 O ator verifica que não retornaram dados	O sistema retorna uma mensagem de aviso

### Detalhar Produto

Visão geral: Este caso de uso tem a finalidade de mostrar os dados de um produto cadastrado.

Ator: Administrador

Ação do ator	Resposta do sistema
1. O ator realiza o caso de uso "Listar Produto" e seleciona um produto cadastrado	O sistema exibe os dados do produto selecionado

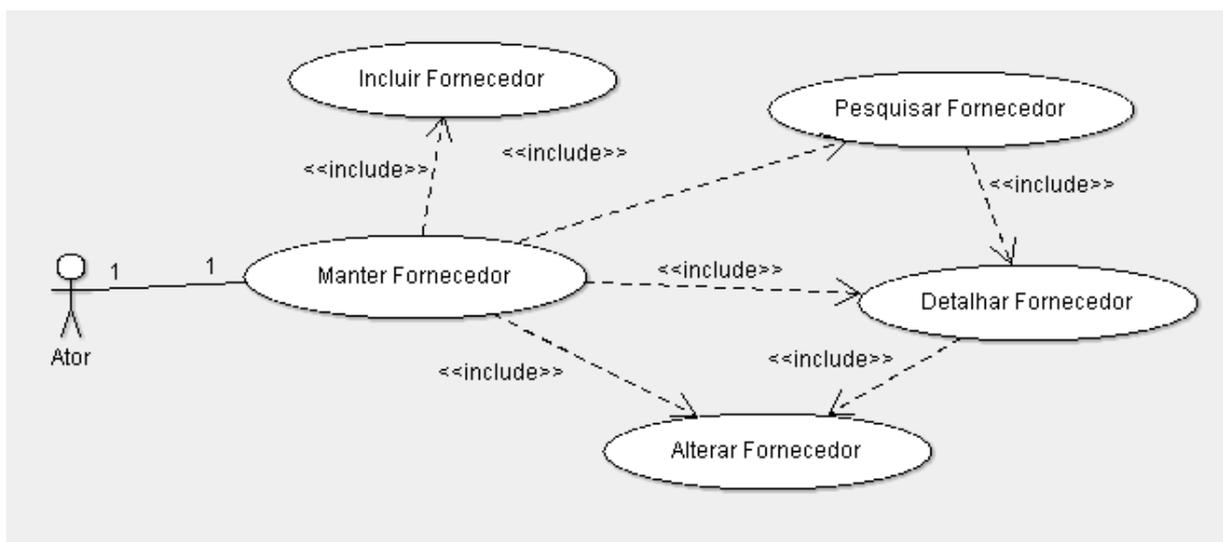
### Alterar Produto

Visão geral: Este caso de uso tem a finalidade alterar um produto cadastrado.

Ator: Administrador

Ação do ator	Resposta do sistema
1. O ator realiza o caso de uso "Detalhar Produto" e clica em editar	Mostra a tela de alteração de produto com base no cadastro selecionado.
2. O ator informa os dados que deseja alterar	O sistema valida e grava os dados
Tratamento de exceções	
2.1 O ator verifica que faltaram dados obrigatórios	O sistema não permite o cadastro

### Manter Fornecedor



### Incluir Fornecedor

Visão geral: Este caso de uso tem a finalidade de incluir um novo fornecedor.

Ator: Administrador

Ação do ator	Resposta do sistema
1. O ator deseja incluir um fornecedor	Mostra a tela de inclusão de fornecedor
2. O ator informa os dados do fornecedor	O sistema valida e grava os dados
Tratamento de exceções	
2.1 O ator verifica que faltaram dados obrigatórios	O sistema não permite o cadastro

### Pesquisar Fornecedor

Visão geral: Este caso de uso tem a finalidade de pesquisar os fornecedores cadastrados.

Ator: Administrador

Ação do ator	Resposta do sistema
1. O ator deseja pesquisar um fornecedor	Mostra a tela de pesquisa de fornecedor
2. O ator pode informar alguns filtros e pesquisar	O sistema realiza a pesquisa
Tratamento de exceções	
2.1 O ator verifica que não retornaram dados	O sistema retorna uma mensagem de aviso

### Detalhar Fornecedor

Visão geral: Este caso de uso tem a finalidade de mostrar os dados de um fornecedor cadastrado.

Ator: Administrador

Ação do ator	Resposta do sistema
1. O ator realiza o caso de uso "Listar Fornecedor" e seleciona um fornecedor cadastrado	O sistema exibe os dados do fornecedor selecionado

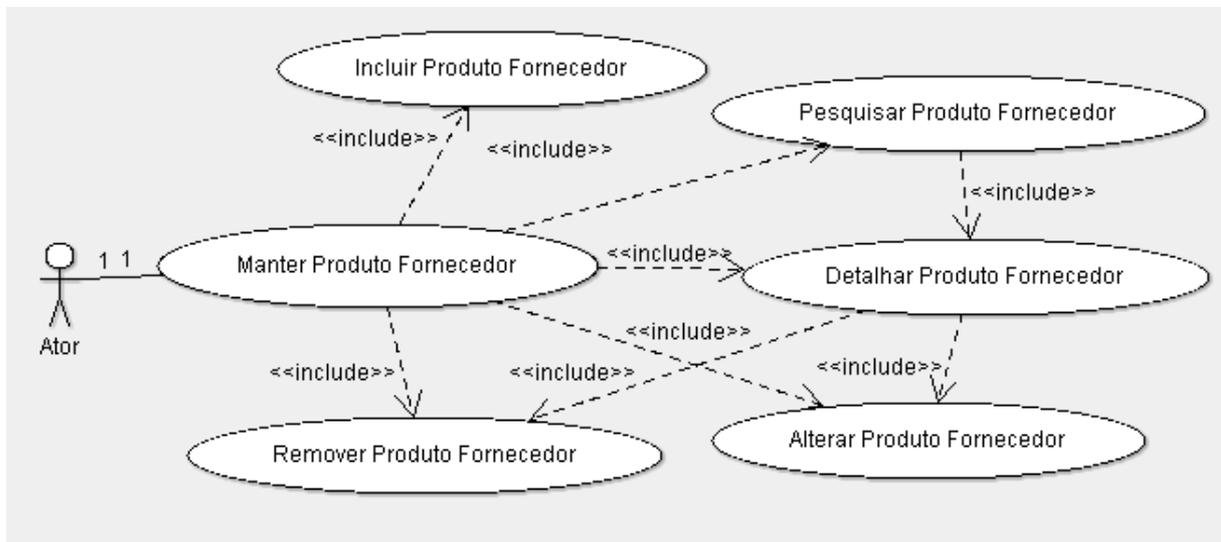
### Alterar Fornecedor

Visão geral: Este caso de uso tem a finalidade de alterar um fornecedor cadastrado.

Ator: Administrador

Ação do ator	Resposta do sistema
1. O ator realiza o caso de uso "Detalhar Fornecedor" e clica em editar	Mostra a tela de alteração de fornecedor com base no cadastro selecionado.
2. O ator informa os dados que deseja alterar	O sistema valida e grava os dados
Tratamento de exceções	
2.1 O ator verifica que faltaram dados obrigatórios	O sistema não permite o cadastro

### Manter Produto Fornecedor



### Incluir Produto Fornecedor

Visão geral: Este caso de uso tem a finalidade de incluir os fornecedores de cada produto.

Ator: Administrador

Ação do ator	Resposta do sistema
1. O ator deseja incluir um fornecedor por produto	Mostra a tela de inclusão de fornecedor por produto
2. O ator informa os dados do fornecedor para o produto	O sistema valida e grava os dados
Tratamento de exceções	
2.1 O ator verifica que faltaram dados obrigatórios	O sistema não permite o cadastro

### Pesquisar Produto Fornecedor

Visão geral: Este caso de uso tem a finalidade de pesquisar os fornecedores dos produtos cadastrados.

Ator: Administrador

Ação do ator	Resposta do sistema
1. O ator deseja pesquisar um fornecedor por produto	Mostra a tela de pesquisa de fornecedor por produto
2. O ator pode informar alguns filtros e pesquisar	O sistema realiza a pesquisa
Tratamento de exceções	
2.1 O ator verifica que não retornaram dados	O sistema retorna uma mensagem de aviso

### Detalhar Produto Fornecedor

Visão geral: Este caso de uso tem a finalidade de mostrar os dados de um fornecedor para um produto cadastrado.

Ator: Administrador

Ação do ator	Resposta do sistema
1. O ator realiza o caso de uso "Listar Produto Fornecedor" e seleciona um fornecedor por produto cadastrado	O sistema exibe os dados do fornecedor para o produto selecionado

### Alterar Produto Fornecedor

Visão geral: Este caso de uso tem a finalidade alterar um fornecedor por produto cadastrado.

Ator: Administrador

Ação do ator	Resposta do sistema
1. O ator realiza o caso de uso "Detalhar Produto Fornecedor" e clica em editar	Mostra a tela de alteração de fornecedor por produto com base no cadastro selecionado.
2. O ator informa os dados que deseja alterar	O sistema valida e grava os dados
Tratamento de exceções	
2.1 O ator verifica que faltaram dados obrigatórios	O sistema não permite o cadastro

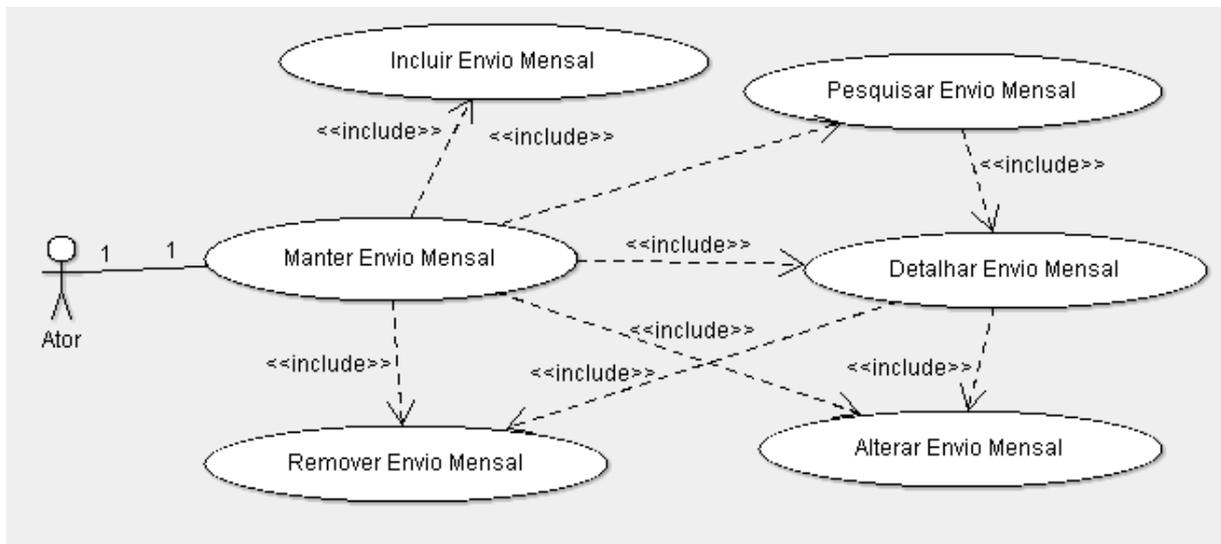
### Remover Produto Fornecedor

Visão geral: Este caso de uso tem a finalidade de remover um fornecedor para um produto cadastrado.

Ator: Administrador

Ação do ator	Resposta do sistema
1. O ator realiza o caso de uso "Alterar Produto Fornecedor" e clica em excluir	Mostra a confirmação
2. O ator confirma a exclusão	O sistema realiza a exclusão

### Manter Envio Mensal



### Incluir Envio Mensal

Visão geral: Este caso de uso tem a finalidade de incluir um envio mensal de produtos para um escola.

Ator: Administrador

Ação do ator	Resposta do sistema
1. O ator deseja incluir um envio mensal	Mostra a tela de inclusão de envio mensal
2. O ator informa os dados do envio mensal	O sistema valida e grava os dados
Tratamento de exceções	
2.1 O ator verifica que faltaram dados obrigatórios	O sistema não permite o cadastro

### Pesquisar Envio Mensal

Visão geral: Este caso de uso tem a finalidade de pesquisar os envio mensais cadastrados.

Ator: Administrador

Ação do ator	Resposta do sistema
1. O ator deseja pesquisar um envio mensal	Mostra a tela de pesquisa de envio mensal
2. O ator pode informar alguns filtros e pesquisar	O sistema realiza a pesquisa
Tratamento de exceções	
2.1 O ator verifica que não retornaram dados	O sistema retorna uma mensagem de aviso

### Detalhar Envio Mensal

Visão geral: Este caso de uso tem a finalidade de mostrar os dados de um envio mensal cadastrado.

Ator: Administrador

Ação do ator	Resposta do sistema
1. O ator realiza o caso de uso "Listar Envio Mensal" e seleciona um envio mensal cadastrado	O sistema exibe os dados do envio mensal selecionado

### Alterar Envio Mensal

Visão geral: Este caso de uso tem a finalidade alterar um envio mensal cadastrado.

Ator: Administrador

Ação do ator	Resposta do sistema
1. O ator realiza o caso de uso "Detalhar Envio Mensal" e clica em editar	Mostra a tela de alteração de envio mensal com base no cadastro selecionado.
2. O ator informa os dados que deseja alterar	O sistema valida e grava os dados
Tratamento de exceções	
2.1 O ator verifica que faltaram dados obrigatórios	O sistema não permite o cadastro

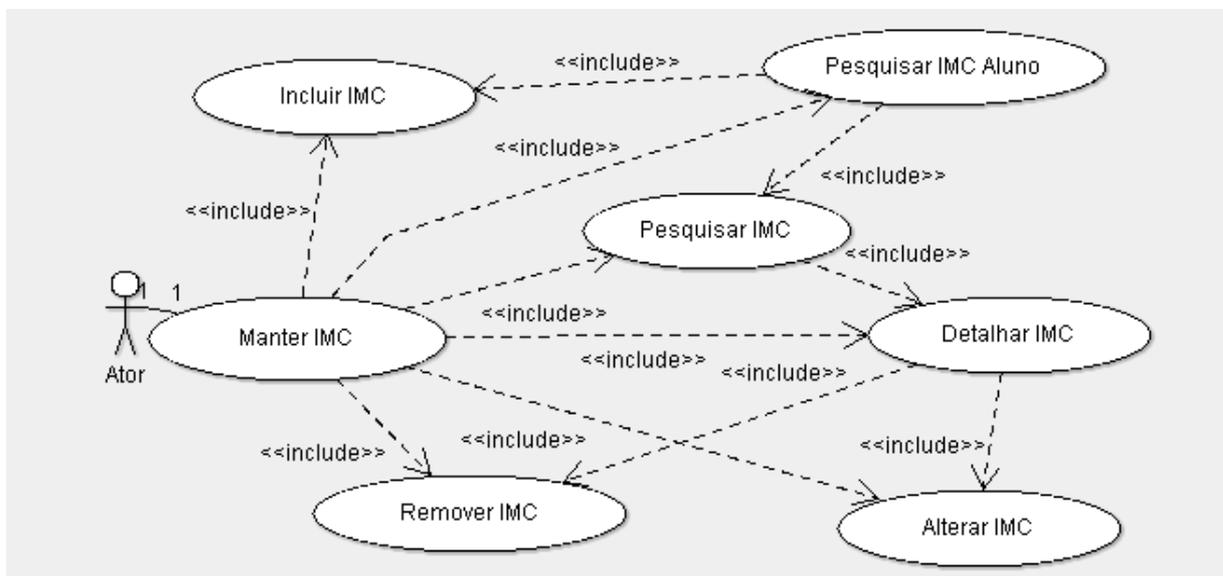
### Remover Envio Mensal

Visão geral: Este caso de uso tem a finalidade de remover um envio mensal cadastrado.

Ator: Administrador

Ação do ator	Resposta do sistema
1. O ator realiza o caso de uso "Alterar Envio Mensal" e clica em excluir	Mostra a confirmação
2. O ator confirma a exclusão	O sistema realiza a exclusão

### Manter IMC



### Pesquisar IMC Aluno

Visão geral: Este caso de uso tem a finalidade listar os alunos cadastrados para cadastro do IMC.

Ator: Administrador

Ação do ator	Resposta do sistema
1. O ator deseja pesquisar um aluno	Mostra a tela de pesquisa de aluno
2. O ator tem que informar o nome do aluno e pesquisar	O sistema realiza a pesquisa
Tratamento de exceções	
2.1 O ator verifica que não retornaram dados	O sistema retorna uma mensagem de aviso

### Incluir IMC

Visão geral: Este caso de uso tem a finalidade de incluir os dados para o cálculo do IMC para um aluno.

Ator: Administrador

Ação do ator	Resposta do sistema
1. O ator realiza o caso de uso "Listar IMC Aluno" e seleciona um aluno cadastrado	Mostra a tela de inclusão dos dados
2. O ator informa os dados	O sistema valida e grava os dados
Tratamento de exceções	
2.1 O ator verifica que faltaram dados obrigatórios	O sistema não permite o cadastro

### Pesquisar IMC

Visão geral: Este caso de uso tem a finalidade de os dados de IMC cadastrados para um aluno.

Ator: Administrador

Ação do ator	Resposta do sistema
1. O ator realiza o caso de uso "Listar IMC Aluno" e seleciona um aluno cadastrado	Mostra a tela com os IMC's já cadastrados
Tratamento de exceções	
11 O ator verifica que não retornaram dados	O sistema retorna uma mensagem de aviso

### Detalhar IMC

Visão geral: Este caso de uso tem a finalidade de mostrar os dados de um IMC cadastrado.

Ator: Administrador

Ação do ator	Resposta do sistema
1. O ator realiza o caso de uso "Listar IMC" e seleciona um IMC cadastrado	O sistema exibe os dados do IMC selecionado

### Alterar IMC

Visão geral: Este caso de uso tem a finalidade de alterar um IMC cadastrado.

Ator: Administrador

Ação do ator	Resposta do sistema
1. O ator realiza o caso de uso "Detalhar IMC" e clica em editar	Mostra a tela de alteração de IMC com base no cadastro selecionado.
2. O ator informa os dados que deseja alterar	O sistema valida e grava os dados
Tratamento de exceções	
2.1 O ator verifica que faltaram dados obrigatórios	O sistema não permite o cadastro

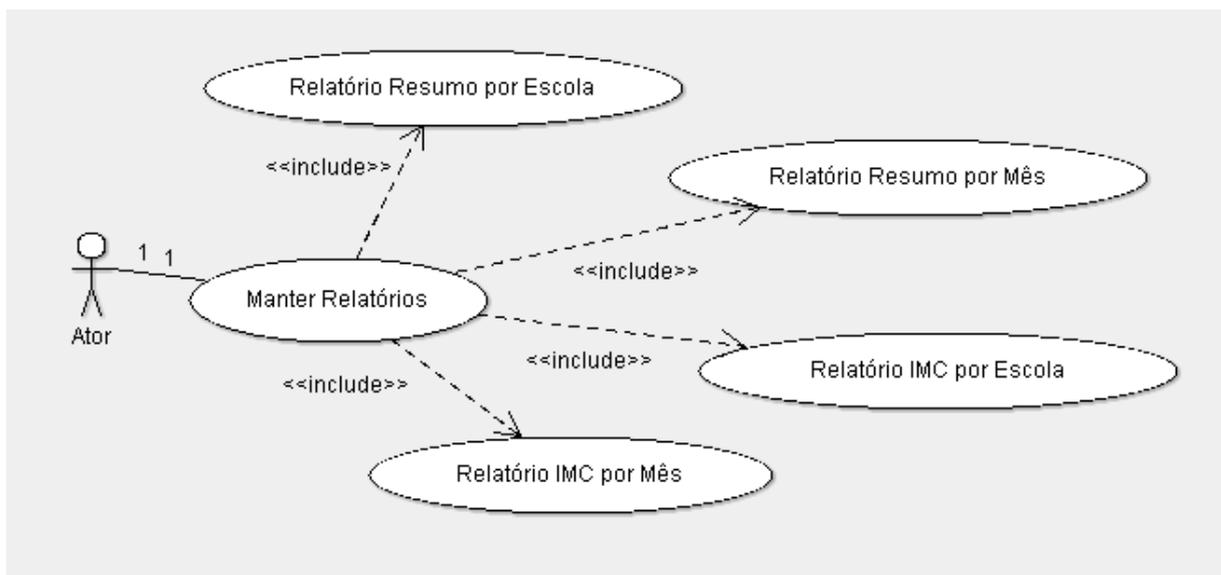
## Remover IMC

Visão geral: Este caso de uso tem a finalidade de remover um IMC cadastrado.

Ator: Administrador

Ação do ator	Resposta do sistema
1. O ator realiza o caso de uso "Alterar IMC" e clica em excluir	Mostra a confirmação
2. O ator confirma a exclusão	O sistema realiza a exclusão

## Manter Relatórios



## Relatório Resumo por Escola

Visão geral: Este caso de uso tem a finalidade de gerar um relatório com um resumo por

escolas das informações nutricionais e custos das refeições.

Ator: Administrador

Ação do ator	Resposta do sistema
1. O ator deseja gerar um relatório	Mostra a tela de filtros
2. O ator define os filtros e pesquisa	O sistema realiza a pesquisa e gera o relatório
Tratamento de exceções	
2.1 O ator verifica que não retornaram dados	O sistema retorna uma mensagem de aviso e não gera o relatório

### Relatório Resumo por Mês

Visão geral: Este caso de uso tem a finalidade de gerar um relatório com um resumo por mês das informações nutricionais e custos das refeições.

Ator: Administrador

Ação do ator	Resposta do sistema
1. O ator deseja gerar um relatório	Mostra a tela de filtros
2. O ator define os filtros e pesquisa	O sistema realiza a pesquisa e gera o relatório
Tratamento de exceções	
2.1 O ator verifica que não retornaram dados	O sistema retorna uma mensagem de aviso e não gera o relatório

### Relatório IMC por Escola

Visão geral: Este caso de uso tem a finalidade de gerar um relatório a média do IMC por escola.

Ator: Administrador

Ação do ator	Resposta do sistema
1. O ator deseja gerar um relatório	Mostra a tela de filtros
2. O ator define os filtros e pesquisa	O sistema realiza a pesquisa e gera o relatório
Tratamento de exceções	
2.1 O ator verifica que não retornaram dados	O sistema retorna uma mensagem de aviso e

	não gera o relatório
--	----------------------

### Relatório IMC por Série

Visão geral: Este caso de uso tem a finalidade gerar um relatório a média do IMC por série.

Ator: Administrador

Ação do ator	Resposta do sistema
1. O ator deseja gerar um relatório	Mostra a tela de filtros
2. O ator define os filtros e pesquisa	O sistema realiza a pesquisa e gera o relatório
Tratamento de exceções	
2.1 O ator verifica que não retornaram dados	O sistema retorna uma mensagem de aviso e não gera o relatório

## 8.3 Anexo C – Estrutura do Banco de Dados

<b>alimentacao.cardapio</b>						
ColumnName	DataType	PrimaryKey	NotNull	Flags	Comment	AutoInc
<b>idcar</b>	<b>serial</b>	PK	NN		Identificar do cardápio	AI
ref_usuario_cad	INTEGER		NN	UNSIGNED	Usuário que cadastrou	
dt_cadastro	TIMESTAMP		NN		Data do cadastro	
descricao	VARCHAR(250)				Descritivo	
nm_arquivo	VARCHAR(100)		NN		Nome do arquivo anexado com o cardápio	
ref_escola	serial		NN		Escola que pertence o cardápio	
path_arquivo	VARCHAR(100)		NN		Caminho do arquivo no servidor	
IndexName		IndexType			Columns	
PRIMARY		PRIMARY			idcar	
cardapio_ref_escola_fkey (Rel_07)		Index			ref_escola	
cardapio_ref_usuario_cad_fkey (Rel_14)		Index			ref_usuario_cad	

<b>alimentacao.envio_mensal_escola</b>						
ColumnName	DataType	PrimaryKey	NotNull	Flags	Comment	AutoInc
<b>ideme</b>	<b>serial</b>	PK	NN		Identificar do envio mensal	AI
ref_escola	serial		NN		Escola	
alunos	INTEGER		NN	UNSIGNED	Quantidade de alunos no mês	
ano	NUMERIC(4,0)		NN		Ano do envio	

mes	NUMERIC(2,0)	NN		Mês do envio
dt_cadastro	TIMESTAMP	NN		Data do cadastro
dias	SMALLINT	NN	UNSIGNED	Quantidade de dias de aula no mês
refeicoes	SMALLINT	NN	UNSIGNED	Quantidade de refeições diárias
IndexName		IndexType		Columns
PRIMARY		PRIMARY		ideme
envio_mensal_escola_ref_escola_fk ey (Rel_05)		Index		ref_escola

**alimentacao.envio\_mensal\_escola\_produto**

ColumnName	DataType	PrimaryKey	NotNull	Flags	Comment	AutoInc
<b>ref_envio_mensal_escola</b>	<b>serial</b>	PK	NN		Envio mensal	AI
<b>ref_produto</b>	<b>serial</b>	PK	NN		Produto	
pesoouvolume	NUMERIC(7,2)				Peso ou volume enviado	
IndexName		IndexType		Columns		
PRIMARY		PRIMARY		ref_envio_mensal_escola ref_produto		
envio_mensal_escola_produto_ref_ envio_mensal_escola_fkey (Rel_04)		Index		ref_envio_mensal_escola		
envio_mensal_escola_ref_produto_f key (Rel_09)		Index		ref_produto		

**alimentacao.envio\_mensal\_padroes**

ColumnName	DataType	PrimaryKey	NotNull	Flags	Comment	AutoInc
<b>idemp</b>	<b>serial</b>	PK	NN		Identificar	
ano	NUMERIC(4,		NN		Ano de referência	

mes	NUMERIC(2,0)	NN		Mês de referência
dias	SMALLINT	NN	UNSIGNED	Dias de aula no mês
refeicoes	SMALLINT	NN	UNSIGNED	Quantidade de refeições diárias
<b>IndexName</b>				
<b>IndexType</b>		<b>Columns</b>		
PRIMARY		PRIMARY idemp		

**alimentacao.fornecedor**

ColumnName	DataType	PrimaryKey	NotNull	Flags	Comment	AutoInc
idpes	INTEGER	PK	NN	UNSIGNED	Identificador do fornecedor	AI
descricao	VARCHAR(100)				Simple descrição	
<b>IndexName</b>						
<b>IndexType</b>		<b>Columns</b>				
PRIMARY		PRIMARY idpes				
fornecedor_ref_juridica_fkey (Rel_15)		Index				idpes

**alimentacao.imc**

ColumnName	DataType	PrimaryKey	NotNull	Flags	Comment	AutoInc
idimc	serial	PK	NN		Identificador	AI
ref_escola	serial		NN		Escola	
ref_aluno	INTEGER		NN	UNSIGNED	Aluno	
ref_serie	serial		NN		Série	
dt_cadastro	TIMESTAMP		NN		Data do cadastro	
altura	NUMERIC(5,2)		NN		Altura	

peso	NUMERIC(5,2)	NN	Peso
imc	NUMERIC(5,2)	NN	Índice de Massa Corporal. Peso dividido pela altura ao quadrado.
observacao	VARCHAR(200)		Simple descrição
IndexName		IndexType	Columns
PRIMARY		PRIMARY	idimc
imc_ref_escola_fkey (Rel_08)		Index	ref_escola
imc_ref_serie_fkey (Rel_10)		Index	ref_serie
imc_ref_aluno_fkey (Rel_18)		Index	ref_aluno

**alimentacao.produto**

ColumnName	DataType	PrimaryKey	NotNull	Flags	Comment	AutoInc
<b>idpro</b>	<b>serial</b>	PK	NN		Identificar do produto	AI
ref_produto_unidade	serial		NN		Unidade de medida	
nm_produto	VARCHAR(50)		NN		Nome do produto	
fator_correcao	NUMERIC(4,0)		NN		Quantidade aproveitável.	
fator_coccao	NUMERIC(4,0)		NN		Peso após cozimento	
ref_produto_grupo	serial		NN		Grupo do produto	
calorias	NUMERIC(4,0)		NN		Calorias a cada 100g	
proteinas	NUMERIC(4,0)		NN		Proteínas a cada 100g	
IndexName		IndexType	Columns			
PRIMARY		PRIMARY	idpro			
produto_ref_produto_grupo_fkey (Rel_01)		Index	ref_produto_grupo			
produto_ref_produto_unidade_fkey		Index	ref_produto_unidade			

(Rel\_20)

**alimentacao.produto\_fornecedor**

ColumnName	DataType	PrimaryKey	NotNull	Flags	Comment	AutoInc
<b>idpf</b>	<b>serial</b>	PK	NN		Identificador	AI
ref_produto	serial		NN		Produto	
ref_fornecedor	INTEGER		NN	UNSIGNED	Fornecedor	
ano	NUMERIC(4,0)		NN		Ano do fornecimento	
mes_inicio	NUMERIC(2,0)		NN		Mês do inicio do fornecimento	
mes_fim	NUMERIC(2,0)		NN		Mês do fim do fornecimento	
pesoouvolume_un	NUMERIC(7,2)				Pesou ou volume por unidade	
preco_un	NUMERIC(5,2)				Preço por unidade	
agri_familiar	SMALLINT		NN	UNSIGNED	Agricultura familiar	

IndexName	IndexType	Columns
PRIMARY	PRIMARY	idpf
produto_fornecedor_ref_fornecedor_fk (Rel_02)	Index	ref_fornecedor
produto_fornecedor_ref_produto_fk (Rel_03)	Index	ref_produto

**alimentacao.produto\_grupo**

ColumnName	DataType	PrimaryKey	NotNull	Flags	Comment	AutoInc
<b>idpg</b>	<b>serial</b>	PK	NN		Identificador do grupo	AI
descricao	VARCHAR(20)				Nome do grupo	

IndexName	IndexType	Columns
-----------	-----------	---------

PRIMARY	PRIMARY	idpg
---------	---------	------

**alimentacao.produto\_unidade**

ColumnName	DataType	PrimaryKey	NotNull	Flags	Comment	AutoInc
<b>idpu</b>	<b>serial</b>	PK	NN		Identificador da unidade	
descricao	VARCHAR(20)		NN		Simple descrição	
unidade	VARCHAR(5)		NN		Unidade de medida	
IndexName	IndexType		Columns			
PRIMARY	PRIMARY		idpu			

**alimentacao.usuario\_escola**

ColumnName	DataType	PrimaryKey	NotNull	Flags	Comment	AutoInc
<b>ref_usuario</b>	<b>INTEGER</b>	PK	NN	UNSIGNED	Usuário que será nutricionista da escola	AI
<b>ref_escola</b>	<b>serial</b>	PK	NN		Escola	
dt_cadastro	TIMESTAMP		NN		Data do cadastro	
IndexName	IndexType		Columns			
PRIMARY	PRIMARY		ref_usuario ref_escola			
usuario_escola_ref_escola_fkey (Rel_06)	Index		ref_escola			
usuario_escola_ref_usuario_fkey (Rel_13)	Index		ref_usuario			

## 8.4 Anexo D – SQL para Configuração da Base de Dados

```
-- Schema: "alimentacao"
-- DROP SCHEMA alimentacao;
CREATE SCHEMA alimentacao
  AUTHORIZATION ieducaruser;
GRANT ALL ON SCHEMA alimentacao TO ieducaruser;

-- Sequence: alimentacao.cardapio_idcar_seq
-- DROP SEQUENCE alimentacao.cardapio_idcar_seq;
CREATE SEQUENCE alimentacao.cardapio_idcar_seq
  INCREMENT 1
  MINVALUE 1
  MAXVALUE 9223372036854775807
  START 1
  CACHE 1;
ALTER TABLE alimentacao.cardapio_idcar_seq OWNER TO ieducaruser;

-- Sequence: alimentacao.envio_mensal_padroes_idemp_seq
-- DROP SEQUENCE alimentacao.envio_mensal_padroes_idemp_seq;
CREATE SEQUENCE alimentacao.envio_mensal_padroes_idemp_seq
  INCREMENT 1
  MINVALUE 1
  MAXVALUE 9223372036854775807
  START 9
  CACHE 1;
ALTER TABLE alimentacao.envio_mensal_padroes_idemp_seq OWNER TO ieducaruser;

-- Sequence: alimentacao.envio_mensal_escola_ideme_seq
-- DROP SEQUENCE alimentacao.envio_mensal_escola_ideme_seq;
CREATE SEQUENCE alimentacao.envio_mensal_escola_ideme_seq
  INCREMENT 1
  MINVALUE 1
  MAXVALUE 9223372036854775807
```

```
START 9
CACHE 1;
ALTER TABLE alimentacao.envio_mensal_escola_ideme_seq OWNER TO ieducaruser;

-- Sequence: alimentacao.imc_idimc_seq
-- DROP SEQUENCE alimentacao.imc_idimc_seq;
CREATE SEQUENCE alimentacao.imc_idimc_seq
INCREMENT 1
MINVALUE 1
MAXVALUE 9223372036854775807
START 8
CACHE 1;
ALTER TABLE alimentacao.imc_idimc_seq OWNER TO ieducaruser;

-- Sequence: alimentacao.produto_fornecedor_idpf_seq
-- DROP SEQUENCE alimentacao.produto_fornecedor_idpf_seq;
CREATE SEQUENCE alimentacao.produto_fornecedor_idpf_seq
INCREMENT 1
MINVALUE 1
MAXVALUE 9223372036854775807
START 7
CACHE 1;
ALTER TABLE alimentacao.produto_fornecedor_idpf_seq OWNER TO ieducaruser;

-- Sequence: alimentacao.produto_grupo_idpg_seq
-- DROP SEQUENCE alimentacao.produto_grupo_idpg_seq;
CREATE SEQUENCE alimentacao.produto_grupo_idpg_seq
INCREMENT 1
MINVALUE 1
MAXVALUE 9223372036854775807
START 3
CACHE 1;
ALTER TABLE alimentacao.produto_grupo_idpg_seq OWNER TO ieducaruser;
```

```

-- Sequence: alimentacao.produto_unidade_idpu_seq
-- DROP SEQUENCE alimentacao.produto_unidade_idpu_seq;
CREATE SEQUENCE alimentacao.produto_unidade_idpu_seq
INCREMENT 1
MINVALUE 1
MAXVALUE 9223372036854775807
START 3
CACHE 1;
ALTER TABLE alimentacao.produto_unidade_idpu_seq OWNER TO ieducaruser;
-- Sequence: alimentacao.produto_idpro_seq
-- DROP SEQUENCE alimentacao.produto_idpro_seq;
CREATE SEQUENCE alimentacao.produto_idpro_seq
INCREMENT 1
MINVALUE 1
MAXVALUE 9223372036854775807
START 1
CACHE 1;
ALTER TABLE alimentacao.produto_idpro_seq OWNER TO ieducaruser;

-- Table: alimentacao.usuario_escola
-- DROP TABLE alimentacao.usuario_escola;
CREATE TABLE alimentacao.usuario_escola
(
  ref_usuario integer NOT NULL,
  ref_escola integer NOT NULL,
  dt_cadastro timestamp without time zone NOT NULL,
  CONSTRAINT pk_usuario_escola PRIMARY KEY (ref_usuario, ref_escola),
  CONSTRAINT usuario_escola_ref_escola_fkey FOREIGN KEY (ref_escola)
    REFERENCES pmieducar.escola (cod_escola) MATCH SIMPLE
    ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT usuario_escola_ref_usuario_fkey FOREIGN KEY (ref_usuario)
    REFERENCES pmieducar.usuario (cod_usuario) MATCH SIMPLE
    ON UPDATE RESTRICT ON DELETE RESTRICT
)

```

```

)
WITH OIDS;
ALTER TABLE alimentacao.usuario_escola OWNER TO ieducaruser;
GRANT ALL ON TABLE alimentacao.usuario_escola TO ieducaruser;

-- Table: alimentacao.cardapio
-- DROP TABLE alimentacao.cardapio;
CREATE TABLE alimentacao.cardapio
(
  idcar integer NOT NULL DEFAULT nextval('cardapio_idcar_seq'::regclass),
  dt_cadastro timestamp without time zone NOT NULL,
  descricao character varying(250),
  nm_arquivo character varying(100) NOT NULL,
  ref_usuario_cad integer NOT NULL,
  ref_escola integer NOT NULL,
  path_arquivo character varying(100),
  CONSTRAINT pk_cardapio PRIMARY KEY (idcar),
  CONSTRAINT cardapio_ref_escola_fkey FOREIGN KEY (ref_escola)
    REFERENCES pmieducar.escola (cod_escola) MATCH SIMPLE
    ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT cardapio_ref_usuario_cad_fkey FOREIGN KEY (ref_usuario_cad)
    REFERENCES pmieducar.usuario (cod_usuario) MATCH SIMPLE
    ON UPDATE RESTRICT ON DELETE RESTRICT
)
WITH OIDS;
ALTER TABLE alimentacao.cardapio OWNER TO ieducaruser;
GRANT ALL ON TABLE alimentacao.cardapio TO ieducaruser;

-- Table: alimentacao.produto_grupo
-- DROP TABLE alimentacao.produto_grupo;
CREATE TABLE alimentacao.produto_grupo
(
  idpg integer NOT NULL DEFAULT nextval('produto_grupo_idpg_seq'::regclass),

```

```

descricao character varying(20) NOT NULL,
CONSTRAINT pk_produto_grupo PRIMARY KEY (idpg)
)
WITH OIDS;
ALTER TABLE alimentacao.produto_grupo OWNER TO ieducaruser;
GRANT ALL ON TABLE alimentacao.produto_grupo TO ieducaruser;
-- Table: alimentacao.produto_unidade
-- DROP TABLE alimentacao.produto_unidade;
CREATE TABLE alimentacao.produto_unidade
(
idpu integer NOT NULL DEFAULT nextval('produto_unidade_idpu_seq'::regclass),
descricao character varying(20) NOT NULL,
unidade character varying(5) NOT NULL,
CONSTRAINT pk_produto_unidade PRIMARY KEY (idpu)
)
WITH OIDS;
ALTER TABLE alimentacao.produto_unidade OWNER TO ieducaruser;
GRANT ALL ON TABLE alimentacao.produto_unidade TO ieducaruser;
-- Table: alimentacao.produto
-- DROP TABLE alimentacao.produto;
CREATE TABLE alimentacao.produto
(
idpro integer NOT NULL DEFAULT nextval('produto_idpro_seq'::regclass),
nm_produto character varying(50) NOT NULL,
fator_correcao numeric(4) NOT NULL DEFAULT 0,
fator_coccao numeric(4) NOT NULL DEFAULT 0,
ref_produto_grupo integer NOT NULL,
ref_produto_unidade integer NOT NULL,
calorias numeric(4) NOT NULL DEFAULT 0,
proteinas numeric(4) NOT NULL DEFAULT 0,
CONSTRAINT pk_produto PRIMARY KEY (idpro),
CONSTRAINT produto_ref_produto_grupo_fkey FOREIGN KEY (ref_produto_grupo)
REFERENCES alimentacao.produto_grupo (idpg) MATCH SIMPLE

```

```

        ON UPDATE RESTRICT ON DELETE RESTRICT,
CONSTRAINT produto_ref_produto_unidade_fkey FOREIGN KEY (ref_produto_unidade)
    REFERENCES alimentacao.produto_unidade (idpu) MATCH SIMPLE
    ON UPDATE RESTRICT ON DELETE RESTRICT
)
WITH OIDS;
ALTER TABLE alimentacao.produto OWNER TO ieducaruser;
GRANT ALL ON TABLE alimentacao.produto TO ieducaruser;

-- Table: alimentacao.fornecedor
-- DROP TABLE alimentacao.fornecedor;
CREATE TABLE alimentacao.fornecedor
(
    idpes integer NOT NULL,
    descricao character varying(100),
    CONSTRAINT pk_fornecedor PRIMARY KEY (idpes),
    CONSTRAINT fornecedor_ref_juridica_fkey FOREIGN KEY (idpes)
        REFERENCES cadastro.juridica (idpes) MATCH SIMPLE
        ON UPDATE RESTRICT ON DELETE RESTRICT
)
WITH OIDS;
ALTER TABLE alimentacao.fornecedor OWNER TO ieducaruser;
GRANT ALL ON TABLE alimentacao.fornecedor TO ieducaruser;

-- Table: alimentacao.produto_fornecedor
-- DROP TABLE alimentacao.produto_fornecedor;
CREATE TABLE alimentacao.produto_fornecedor
(
    idpf integer NOT NULL DEFAULT nextval('produto_fornecedor_idpf_seq'::regclass),
    ref_produto integer NOT NULL,
    ref_fornecedor integer NOT NULL,
    ano numeric(4) NOT NULL,
    mes_inicio numeric(2) NOT NULL,

```

```

mes_fim numeric(2) NOT NULL,
pesoouvolume_un numeric(7,2),
preco_un numeric(5,2),
agri_familiar smallint NOT NULL DEFAULT 0,
CONSTRAINT pk_produto_fornecedor PRIMARY KEY (idpf),
CONSTRAINT produto_fornecedor_ref_fornecedor_fkey FOREIGN KEY (ref_fornecedor)
    REFERENCES alimentacao.fornecedor (idpes) MATCH SIMPLE
    ON UPDATE RESTRICT ON DELETE RESTRICT,
CONSTRAINT produto_fornecedor_ref_produto_fkey FOREIGN KEY (ref_produto)
    REFERENCES alimentacao.produto (idpro) MATCH SIMPLE
    ON UPDATE RESTRICT ON DELETE RESTRICT
)
WITH OIDS;
ALTER TABLE alimentacao.produto_fornecedor OWNER TO ieducaruser;
GRANT ALL ON TABLE alimentacao.produto_fornecedor TO ieducaruser;

-- Table: alimentacao.envio_mensal_padroes
-- DROP TABLE alimentacao.envio_mensal_padroes;

CREATE TABLE alimentacao.envio_mensal_padroes
(
    idemp integer NOT NULL DEFAULT nextval('envio_mensal_padroes_idemp_seq'::regclass),
    ano numeric(4) NOT NULL,
    mes numeric(2) NOT NULL,
    dias smallint NOT NULL,
    refeicoes smallint NOT NULL,
    CONSTRAINT pk_envio_mensal_padroes_idemp PRIMARY KEY (idemp)
)
WITH OIDS;
ALTER TABLE alimentacao.envio_mensal_padroes OWNER TO ieducaruser;
GRANT ALL ON TABLE alimentacao.envio_mensal_padroes TO ieducaruser;

-- Table: alimentacao.envio_mensal_escola

```

```

-- DROP TABLE alimentacao.envio_mensal_escola;
CREATE TABLE alimentacao.envio_mensal_escola
(
  ideme integer NOT NULL DEFAULT nextval('envio_mensal_escola_ideme_seq'::regclass),
  ref_escola integer NOT NULL,
  alunos integer NOT NULL DEFAULT 0,
  ano numeric(4) NOT NULL,
  mes numeric(2) NOT NULL,
  dias smallint NOT NULL,
  refeicoes smallint NOT NULL,
  dt_cadastro timestamp without time zone NOT NULL,
  CONSTRAINT pk_envio_mensal_escola_ideme PRIMARY KEY (ideme),
  CONSTRAINT envio_mensal_escola_ref_escola_fkey FOREIGN KEY (ref_escola)
    REFERENCES pmieducar.escola (cod_escola) MATCH SIMPLE
    ON UPDATE RESTRICT ON DELETE RESTRICT
)
WITH OIDS;
ALTER TABLE alimentacao.envio_mensal_escola OWNER TO ieducaruser;
GRANT ALL ON TABLE alimentacao.envio_mensal_escola TO ieducaruser;

-- Table: alimentacao.envio_mensal_escola_produto
-- DROP TABLE alimentacao.envio_mensal_escola_produto;
CREATE TABLE alimentacao.envio_mensal_escola_produto
(
  ref_envio_mensal_escola integer NOT NULL,
  ref_produto integer NOT NULL,
  pesoovolume numeric(7,2) NOT NULL DEFAULT 0,
  CONSTRAINT pk_envio_mensal_escola_produto PRIMARY KEY (ref_envio_mensal_escola,
ref_produto),
  CONSTRAINT envio_mensal_escola_produto_ref_envio_mensal_escola_fkey FOREIGN KEY
(ref_envio_mensal_escola)
    REFERENCES alimentacao.envio_mensal_escola (ideme) MATCH SIMPLE
    ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT envio_mensal_escola_ref_produto_fkey FOREIGN KEY (ref_produto)

```

```

REFERENCES alimentacao.produto (idpro) MATCH SIMPLE
ON UPDATE RESTRICT ON DELETE RESTRICT
)
WITH OIDS;
ALTER TABLE alimentacao.envio_mensal_escola_produto OWNER TO ieducaruser;
GRANT ALL ON TABLE alimentacao.envio_mensal_escola_produto TO ieducaruser;

-- Table: alimentacao.imc
-- DROP TABLE alimentacao.imc;
CREATE TABLE alimentacao.imc
(
idimc integer NOT NULL DEFAULT nextval('imc_idimc_seq'::regclass),
ref_escola integer NOT NULL,
ref_aluno integer NOT NULL,
ref_serie integer NOT NULL,
dt_cadastro timestamp without time zone NOT NULL,
altura numeric(5,2) NOT NULL,
peso numeric(5,2) NOT NULL,
imc numeric(5,2) NOT NULL,
observacao character varying(200),
CONSTRAINT pk_imc PRIMARY KEY (idimc),
CONSTRAINT imc_ref_aluno_fkey FOREIGN KEY (ref_aluno)
REFERENCES pmieducar.aluno (cod_aluno) MATCH SIMPLE
ON UPDATE RESTRICT ON DELETE RESTRICT,
CONSTRAINT imc_ref_escola_fkey FOREIGN KEY (ref_escola)
REFERENCES pmieducar.escola (cod_escola) MATCH SIMPLE
ON UPDATE RESTRICT ON DELETE RESTRICT,
CONSTRAINT imc_ref_serie FOREIGN KEY (ref_serie)
REFERENCES pmieducar.serie (cod_serie) MATCH SIMPLE
ON UPDATE RESTRICT ON DELETE RESTRICT
)
WITH OIDS;
ALTER TABLE alimentacao.imc OWNER TO ieducaruser;

```

```
GRANT ALL ON TABLE alimentacao.imc TO ieducaruser;
```

```
-- Inclusão dos menus
```

```
INSERT INTO PORTAL.MENU_MENU(cod_menu_menu,nm_menu) VALUES(1000,'i-Educar - Alimentação')
```

```
INSERT INTO PORTAL.MENU_SUBMENU( cod_menu_submenu, ref_cod_menu_menu, cod_sistema, nm_submenu, arquivo, nivel) VALUES(10000,1000,2, 'Cardápio', 'alimentacao_cardapio_lst.php',3)
```

```
INSERT INTO PORTAL.MENU_SUBMENU( cod_menu_submenu, ref_cod_menu_menu, cod_sistema, nm_submenu, arquivo, nivel) VALUES(10001,1000,2, 'Nutricionista Escolas', 'alimentacao_nutricionista_escola_lst.php',3)
```

```
INSERT INTO PORTAL.MENU_SUBMENU( cod_menu_submenu, ref_cod_menu_menu, cod_sistema, nm_submenu, arquivo, nivel) VALUES(10002,1000,2, 'Produtos', 'alimentacao_produto_lst.php',3)
```

```
INSERT INTO PORTAL.MENU_SUBMENU( cod_menu_submenu, ref_cod_menu_menu, cod_sistema, nm_submenu, arquivo, nivel) VALUES(10003,1000,2, 'Fornecedores', 'alimentacao_fornecedor_lst.php',3)
```

```
INSERT INTO PORTAL.MENU_SUBMENU( cod_menu_submenu, ref_cod_menu_menu, cod_sistema, nm_submenu, arquivo, nivel) VALUES(10004,1000,2, 'Produto Fornecedor', 'alimentacao_produto_fornecedor_lst.php',3)
```

```
INSERT INTO PORTAL.MENU_SUBMENU( cod_menu_submenu, ref_cod_menu_menu, cod_sistema, nm_submenu, arquivo, nivel) VALUES(10005,1000,2, 'Envio Mensal Escola', 'alimentacao_envio_mensal_escola_lst.php',3)
```

```
INSERT INTO PORTAL.MENU_SUBMENU( cod_menu_submenu, ref_cod_menu_menu, cod_sistema, nm_submenu, arquivo, nivel) VALUES(10006,1000,2, 'IMC', 'alimentacao_imc_lst.php',3)
```

```
INSERT INTO PORTAL.MENU_SUBMENU( cod_menu_submenu, ref_cod_menu_menu, cod_sistema, nm_submenu, arquivo, nivel) VALUES(10007,1000,2, 'Relatório - IMC por Escola', 'alimentacao_relatorio_imc_escola.php',3)
```

```
INSERT INTO PORTAL.MENU_SUBMENU( cod_menu_submenu, ref_cod_menu_menu, cod_sistema, nm_submenu, arquivo, nivel) VALUES(10008,1000,2, 'Relatório - IMC por Série', 'alimentacao_relatorio_imc_serie.php',3)
```

```
INSERT INTO PORTAL.MENU_SUBMENU( cod_menu_submenu, ref_cod_menu_menu, cod_sistema, nm_submenu, arquivo, nivel) VALUES(10009,1000,2, 'Relatório - Resumo por Escola', 'alimentacao_relatorio_resumo_escola.php',3)
```

```
INSERT INTO PORTAL.MENU_SUBMENU( cod_menu_submenu, ref_cod_menu_menu, cod_sistema, nm_submenu, arquivo, nivel) VALUES(10010,1000,2, 'Relatório - Resumo por Mês', 'alimentacao_relatorio_resumo_mes.php',3)
```

```
INSERT INTO PORTAL.MENU_SUBMENU( cod_menu_submenu, ref_cod_menu_menu, cod_sistema, nm_submenu, arquivo, nivel) VALUES(10011,1000,2, 'Envio Mensal Padrões', 'alimentacao_envio_mensal_padroes_lst.php',3)
```

```
INSERT INTO ALIMENTACAO.PRODUTO_GRUPO(descricao) VALUES('Menos Perecíveis')
```

```
INSERT INTO ALIMENTACAO.PRODUTO_GRUPO(descricao) VALUES('Mais Perecíveis')
```

```
INSERT INTO ALIMENTACAO.PRODUTO_GRUPO(descricao) VALUES('Não Perecível')
```

```
INSERT INTO ALIMENTACAO.PRODUTO_UNIDADE(descricao,unidade) VALUES('Kg -
Quilograma','Kg')
```

```
INSERT INTO ALIMENTACAO.PRODUTO_UNIDADE(descricao,unidade) VALUES('L - Litros','L')
```

```
ALTER DATABASE ieducardb SET search_path TO "$user", public, portal, cadastro, acesso,
alimentos, consistenciacao, historico, pmiacoes, pmicontrolesis, pmidrh, pmieducar, pmiotopic,
urbano, alimentacao;
```

## 8.5 Anexo E – Artigo

# Sistema de Acompanhamento da Merenda Escolar

**Leandro Zimmer**

Departamento de Informática e Estatística – Universidade Federal de Santa Catarina (UFSC)

Caixa Postal 475 – 88040-900 – Florianópolis – Santa Catarina – Brasil

[zimmer@inf.ufsc.br](mailto:zimmer@inf.ufsc.br)

**Abstract.** *This work aims at to create a tool to assist in the accompaniment of pertaining to school feeding of the city halls, through the creation of a module together with the system i-educar, thus collaborating with the Brazilian community of free software. The survey of requirements for this work was carried through together with the project Via Digital and the City hall of Florianópolis. It presents the phase of survey requisite, analysis, project and the developed module, using HTML, PHP, and PostgreSQL.*

**Resumo.** *Este trabalho visa criar uma ferramenta para auxiliar no acompanhamento da merenda escolar das prefeituras, através da criação de um módulo juntamente com o sistema i-educar, colaborando assim com a comunidade de software livre brasileira. O levantamento de requisitos para este trabalho foi realizado juntamente com o projeto Via Digital e a Prefeitura Municipal de Florianópolis. Ele apresenta a fase de levantamento requisitos, análise, projeto e o módulo desenvolvido, utilizando HTML, PHP, e PostgreSQL.*

## 1. Introdução

A gestão da informação nas empresas é fundamental para sua organização e bom funcionamento dos seus processos. Os softwares são uma ferramenta muito importante neste processo, eles auxiliam as pessoas há terem um maior controle das informações e também na tomada de decisões. O Brasil possui centenas de prefeituras, com as mais variadas necessidades de softwares para uma melhor gestão pública.

Em 2004 foi criado o Via Digital, um repositório de softwares livres e informações para a informatização de prefeituras. Uma das necessidades levantadas pelo projeto Via Digital é um sistema de acompanhamento da merenda nas instituições educacionais. Uma boa alimentação é essencial para os alunos estarem saudáveis e aptos a obter um bom aprendizado. O sistema que será proposto e implementado, será um módulo junto ao i-educar, que é um software livre para o controle de escolas e foi desenvolvido inicialmente pela prefeitura municipal de Itajaí-SC e disponibilizado como software livre através de uma parceria com a empresa Cobra Tecnologia.

As premissas básicas deste trabalho estão definidas a seguir. Este trabalho está centrado no planejamento e desenvolvimento de um módulo junto ao i-educar para o acompanhamento da merenda escolar, com base na identificação dos processos relacionados a distribuição e custos dos alimentos e suas informações nutricionais e o cálculo do índice de massa corporal (IMC) dos alunos. A identificação dos processos será realizada em parceria com o setor de alimentação escolar da Prefeitura Municipal de Florianópolis-SC.

## 2. Merenda Escolar

A merenda escolar é essencial para criar bons hábitos de alimentação e um dos seus principais objetivos é suprir, parcialmente, as necessidades nutricionais dos alunos. Uma alimentação adequada é fundamental para as crianças estarem aptas ao aprendizado, principalmente na infância que é o período que compõe a base de formação do ser humano (MASCARENHAS; SANTOS, 2006).

O Programa Nacional de Alimentação Escolar (PNAE) garante recursos financeiros, desde 1955, para a alimentação escolar da educação básica. O programa também define regras e métricas que as escolas devem cumprir. Os cardápios devem ser desenvolvidos por nutricionistas e suprir 30% das necessidades nutricionais dos alunos de escolas indígenas e localizadas em comunidades de quilombos e 15% das necessidades dos demais alunos. As refeições devem ser definidas respeitando os requisitos nutricionais e os hábitos alimentares e agrícolas de cada região (FNDE, 2009).

A Prefeitura Municipal de Florianópolis (PMF) oferece merenda escolar para mais de 26.000 alunos, matriculados em suas mais de 90 unidades educativas (PMF, 2009a; PMF, 2009b). “A alimentação escolar das unidades educativas, ligadas à Secretaria de Educação da Capital, é municipalizada, centralizada e com gestão própria da prefeitura.” (PMF, 2009a). Os cardápios são definidos para cada escola e obedecem uma composição balanceada, seguindo a pirâmide alimentar (ver Fig. 2.1)



**Fig. 2.1 – Pirâmide Alimentar**

Fonte: [http://www.pmf.sc.gov.br/educa/recursos\\_imagem/recursospiramide1.JPG](http://www.pmf.sc.gov.br/educa/recursos_imagem/recursospiramide1.JPG)

De acordo com a faixa etária dos alunos, e das necessidades nutritivas, são definidas as quantidades necessárias de cada alimento por aluno, também chamada de per capita de alimentos, os alimentos são divididos em menos perecíveis e mais perecíveis. Os alimentos menos perecíveis possuem um tempo maior para consumo e por este motivo são entregues mensalmente, já os alimentos mais perecíveis são entregues mensalmente.

### 3. Sistema de Acompanhamento da Merenda Escolar

Atualmente, a prefeitura de Florianópolis não consegue acompanhar de forma prática e rápida as quantidades de alimentos distribuídos na rede municipal. A falta de registro destas informações em uma base para análise histórica também é uma necessidade identificada pela prefeitura. Com o registro dos alimentos distribuídos e os valores nutricionais dos mesmos, seria possível realizar uma avaliação das calorias e proteínas que estão sendo consumidas pelos alunos e o custo das refeições. Outro problema, é a falta de dados antropométricos dos alunos. Com dados como peso, medida e idade, seria possível realizar comparativos com as refeições servidas.

A solução proposta é um software livre, disponibilizado via internet, que permita o acesso de qualquer micro com um navegador Web e acesso a internet. O sistema irá permitir a inclusão das quantidades e valores nutricionais dos alimentos fornecidos para cada escola da rede municipal. O sistema também irá permitir a inclusão de alunos e dados como peso e altura. O sistema terá uma área de cadastros e uma área com relatórios com base nas informações cadastradas.

O sistema que será desenvolvido é um módulo no i-educar e pode ser classificado como um sistema de pequeno porte. O desenvolvimento do sistema terá a participação de apenas um desenvolvedor que executa o papel de gerente do projeto e está em contato direto com o cliente. O tempo para desenvolvimento do sistema é curto. O cliente tem interesse em receber versões iniciais com partes das funcionalidades do sistema. Por estes e outros

motivos, que serão apresentados abaixo, o modelo escolhido será baseado em metodologias ágeis (SILVA, 2008/1).

Esta metodologia visa diminuir os riscos no processo de desenvolvimento do sistema. Ela visa dividir as etapas do desenvolvimento em funcionalidades que gerem uma nova versão do sistema. O principal objetivo é reduzir os períodos destas etapas, chamados de iteração. Assim é possível lançar novas versões com maior frequência e o cliente pode validar e sugerir mudanças nas novas funcionalidades, evitando assim que grandes blocos do sistema possam ser desenvolvidos sem uma avaliação do cliente, aumentando o risco de retrabalho caso uma etapa não supra a necessidade esperada pelo cliente. (Wikipédia, 2009a)

As tecnologias adotadas para o desenvolvimento do sistema foram o banco de dados PostgreSQL, servidor Apache e a linguagem de desenvolvimento PHP, tendo em vista que o sistema i-educar já foi desenvolvido com estas tecnologias.

## **4. Desenvolvimento**

### **4.1 Análise**

O projeto do sistema foi realizado com base nos estudos realizados sobre tecnologias e merenda escolar, juntamente com as informações obtidas com o setor de alimentação escolar da prefeitura municipal de Florianópolis. Como o objetivo deste trabalho é incorporar o módulo de acompanhamento de merenda escolar ao i-educar, foi necessário uma análise da estrutura atual do sistema e seu funcionamento para iniciar a fase de desenvolvimento. A seguir apresentamos detalhes do levantamento de requisitos e da fase de projeto.

### **4.2 Entrevistas**

Após estudar o setor de alimentação escolar foram definidos tópicos para levantar os requisitos para o desenvolvimento do sistema. A primeira entrevista foi realizada no dia 14 de maio de 2009, na coordenação de alimentação escolar da prefeitura e teve a participação do coordenador e de diversas pessoas envolvidas na manutenção do setor. Os tópicos abordados nas entrevistas foram referentes ao estoque e aos cardápios, com base nas entrevistas foram definidos os requisitos.

### **4.3 Requisitos**

Abaixo apresentamos uma descrição dos requisitos necessários para o sistema, de acordo com as necessidades levantadas juntamente com a coordenação de alimentação escolar do município.

Um sistema Web que permitirá o cadastro de usuários que poderão cadastrar novos usuários, escolas, alunos, produtos e suas referências nutricionais. Mensalmente os usuários irão cadastrar as quantidades de alimentos enviados para cada escola, deverá informar a quantidade por produto e a quantidade de vezes que ele será servido no mês. Juntamente com a quantidade de alimentos o usuário deverá informar o número de alunos. Com base nestas informações e detalhes do produto, serão calculados os valores nutricionais oferecidos mensalmente por aluno e serão gerados os históricos de distribuição de alimentos. Também será permitido a inclusão de medidas como peso e altura por aluno, com o objetivo de identificar o IMC dos alunos.

#### **4.4 Diagramas de Casos de Uso**

Os diagramas do sistema foram agrupados por funcionalidades, sendo que cada diagrama possui seus casos de uso expandidos. Os diagramas de casos de uso criados foram:

##### ***Manter Cardápio***

Diagrama que representa a manutenção de cadastro dos cardápios das escolas. Os casos expandidos são: Incluir Cardápio, Pesquisar Cardápio, Detalhar Cardápio e Alterar Usuário.

##### ***Manter Nutricionista Escola***

Diagrama que representa a manutenção dos nutricionistas de cada escola. Os casos expandidos são: Incluir Nutricionista Escola, Pesquisar Nutricionista Escola, Detalhar Nutricionista Escola, Alterar Nutricionista Escola e Remover Nutricionista Escola.

##### ***Manter Produto***

Diagrama que representa a manutenção dos produtos cadastrados no sistema. Os casos expandidos são: Incluir Produto, Pesquisar Produto, Detalhar Produto e Alterar Produto.

##### ***Manter Fornecedor***

Diagrama que representa a manutenção dos fornecedores de produtos. Os casos expandidos são: Incluir Fornecedor, Pesquisar Fornecedor, Detalhar Fornecedor e Alterar Fornecedor.

##### ***Manter Produto Fornecedor***

Diagrama que representa a manutenção dos produtos fornecidos por cada fornecedor. Os casos expandidos são: Incluir Produto Fornecedor, Pesquisar Produto Fornecedor, Detalhar Produto Fornecedor, Alterar Produto Fornecedor e Remover Produto Fornecedor.

### ***Manter Envio Mensal***

Diagrama que representa a manutenção dos envios mensais de produtos para as escolas. Os casos expandidos são: Incluir Envio Mensal, Pesquisar Envio Mensal, Detalhar Envio Mensal, Alterar Envio Mensal e Remover Envio Mensal.

### ***Manter IMC***

Diagrama que representa a manutenção dos IMCs cadastrados para os alunos. Os casos expandidos são: Pesquisar IMC Aluno, Incluir IMC, Pesquisar IMC, Detalhar IMC, Alterar IMC e Remover IMC.

### ***Manter Relatórios***

Diagrama que representa a manutenção dos relatórios do sistema. Os casos expandidos são: Relatório Resumo por Escola, Relatório Resumo por Mês, Relatório IMC por Escola e Relatório IMC por Mês.

## **4.5 Diagramas de Classes**

Para a criação do módulo de acompanhamento da merenda escolar, junto ao sistema i-educar, foram criadas nove classes. Estas classes deste módulo iniciam com `clsAlimentacao`, as demais classes já faziam parte do sistema i-educar. Os diagramas de classes foram agrupados por afinidade e foram divididos em 4 grupos.

O primeiro diagrama de classes apresenta as classes de produtos, grupo de produtos, fornecedores e os produtos por fornecedores. O segundo representa as classes relacionadas ao envio mensal de produtos para as escolas. O terceiro é o diagrama de classes relacionadas ao cadastro de cardápios das escolas. O último diagrama de classes , apresenta as classes relacionadas ao cadastro do IMC para os alunos.

## **4.6 Diagramas de Banco de Dados**

O diagrama de banco de dados, apresenta todas as tabelas e atributos e seus relacionamentos. As tabelas precedidas de “alimentacao.” foram criadas como parte deste módulo dentro do i-educar. Já as outras tabelas já faziam parte do sistema i-educar, mas foram utilizadas na criação deste módulo dentro do i-educar. As tabelas, que já faziam parte do

sistema, foram apresentadas somente com os atributos e relacionamentos que foram utilizados neste trabalho.

## 4.7 Projeto

Após o levantamento de requisitos e o planejamento do módulo de acompanhamento da merenda escolar, iniciamos o desenvolvimento deste módulo no sistema i-educar. Esta fase foi dividida em três etapas: criação da base de dados, criação das classes e criação das interfaces.

A criação da base de dados foi realizada facilmente com base nos scripts sql gerados a partir do diagrama de banco de dados. Como o PostgreSQL permite a criação de schemas para agrupamento de tabelas e funcionalidades. Como o sistema i-educar possui diversos schemas, criamos o schema “alimentacao” para agrupar as tabelas do módulo que estamos criando.

A criação das classes foi simples e rápida. Todas as classes criadas apresentam métodos padrões, este padrão se deve ao fato das classes de interface interagirem sempre com os mesmos métodos. No entanto, outros métodos podem ser criados e foram necessários em algumas classes.

Todas as interfaces do sistemas herdam a classe clsBase que é responsável pela criação layout principal da página e também realiza o controle de permissões. As interfaces do i-educar foram definidas em tela de pesquisa ou listagem, tela de detalhes e tela de cadastro. Sempre que desejamos criar uma destas telas, devemos criar uma classe que herde características da sua classe pai, ou seja, para criar uma tela de listagem herdamos a classe clsListagem, tela de detalhes herdamos a clsDetalhe e cadastro a classe clsCadastro. Todas estas telas podem ser compostas de campos que são adicionados para pesquisa, cadastro ou somente exibição. Estes campos são gerados pela classe clsCampos e possuem todos os campos padrões do html e alguns padrões de exibição criamos para o sistema. As classes herdadas de clsListagem e clsDetalhe interagem com o método lista que estão disponíveis nas classes de negócios já criadas, a classe clsCadastro acessa os métodos lista, cadastra, edita e exclui.

A maior dificuldade encontrada, durante o desenvolvimento, foi a compreensão da estrutura de classes e base de dados do sistema, pois é bem vasta e não possui documentação. A maior dificuldade foi na criação da interface do sistema, pois foi necessário estudar todas as classes para saber quais campos de formulários existiam e como eles eram criados e seus parâmetros. Porém este modelo já criado permitiu a manutenção nos padrões de layout e funcionamento do sistema.

## 5. Sistema Desenvolvido

O sistema desenvolvido é composto basicamente de 3 grupos de funcionalidades, que serão descritos abaixo:

## **Controle de cardápios**

Esta funcionalidade permite gerenciar os cardápios das escolas, através do cadastro do arquivo com o cardápio da escola. Pode ser cadastrado pelo administrador ou por um usuário nutricionista. As escolas que o usuário nutricionista tem permissão são definidas pelo administrador através da opção “Nutricionista Escolas” e o cadastro dos cardápios é realizado pela opção “Cardápio” disponíveis no menu “i-Educar Alimentação”.

## **Envio mensal de alimentos**

Esta funcionalidade tem o objetivo de registrar a quantidade de alimentos enviada para cada escola. Para registrar as quantidades enviadas, inicialmente é necessário realizar o cadastro dos produtos, opção “Produtos”, depois cadastrar a lista de fornecedores, opção “Fornecedores”, depois cadastrar quais produtos são fornecidos por cada fornecedor e suas informações como peso da embalagem e custo, através da opção “Produto Fornecedor”. Após definidas estas informações pode ser lançada as quantidades enviadas mensalmente para cada escola. Os relatórios “Resumo por Escola” e “Resumo por Mês” permitem acompanhar informações como: custo, custo por refeição, calorias diárias e proteínas diárias por aluno e custo com agricultura familiar.

## **Cadastro do IMC**

O cadastro do Índice de Massa Corporal é realizado através do cadastro do peso e altura do aluno. Esta inclusão deve ser realizado pela opção “IMC” e o acompanhamento pode ser realizado pelos relatórios “IMC por Escola” e “IMC por Série”.

Como já apresentamos anteriormente, o sistema i-educar possui uma estrutura de cadastrado organizada em tela de pesquisa, detalhes e inclusão e alteração. Como todas as telas do sistema seguem o mesmo padrão e visando apresentar como ficou as telas do módulo desenvolvido, iremos mostrar abaixo os 3 modelos de telas utilizando o cadastro de produtos como exemplo, verifique os requisitos funcionais para ver todas as telas e seus atributos.

**Filtros de busca**

Grupo

Produto - Listagem

Produto	Grupo	Fator Correção	Fator Cocção
Arroz	Menos Perecíveis	100	150
Bolacha rosquinha de chocolate	Menos Perecíveis	100	100
Cenoura	Mais Perecíveis	90	100
Feijão Preto	Menos Perecíveis	100	130
Tomate	Mais Perecíveis	80	100

Fig. 5.1 – Tela de pesquisa de produtos

**Produto - Detalhe**

Produto: Arroz

Fator Correção: 100

Fator Cocção: 150

Grupo: Menos Perecíveis

Calorias(Kcal)/100gr ou ml: 240

Proteínas (gramas)/100gr ou ml: 14

Fig. 5.2 – Tela de detalhes do produto

**Editar**

Produto \*

Fator Correção \*   
somente números

Fator Cocção \*   
somente números

Grupo \*

Calorias(Kcal) \*   
Quantidade há cada 100 gramas ou ml

Proteínas(gramas) \*   
Quantidade há cada 100 gramas ou ml

Fig. 5.3 – Tela de inclusão e alteração de produtos

## 6. Conclusão

Após a finalização do módulo, junto ao sistema i-educar, para acompanhamento da merenda escolar em instituições de ensino das prefeituras, podemos apresentar as seguintes conclusões:

O levantamento dos processos e requisitos, relacionados a alimentação escolar, foi realizado com sucesso na Prefeitura Municipal de Florianópolis e foi fundamental na obtenção das metas na fase de análise.

A criação deste trabalho, permitiu a utilização de muitas funcionalidades já existentes no sistema i-educar e colaborou para o aprimoramento deste software livre.

O módulo desenvolvido alcançou os objetivos planejados e está apto para ser utilizado via WEB nas redes de ensino municipais juntamente com o i-educar, auxiliando no controle produtos, fornecedores, na qualidade das refeições servidas, no acompanhamento dos custos e no índice de massa corporal dos alunos.

Os benefícios esperados são o controle das necessidades nutricionais diárias do alunos, que pode ser aferida através das calorias e proteínas diárias fornecidas por aluno. O acompanhamento do consumo mensal nas escolas, visando mudanças anormais no consumo mensal que podem ocorrer após alterações nos cardápios com pouca receptividade dos alunos. Controle dos custos e valores nutricionais dos cardápios, buscando assim o melhor custo benefício em relação a qualidade da merenda. E também o acompanhamento da massa corporal dos alunos, com o objetivo de traçar um paralelo com os valores nutricionais e alimentos fornecidos.

Com base no conhecimento adquirido, sugere-se como trabalho futuro a implantação e o acompanhamento dos benefícios esperados deste sistema em uma rede de ensino.

## Referências

FNDE. **Alimentação Escolar**, 2009. Disponível em: <[http://www.fnde.gov.br/home/index.jsp?arquivo=alimentacao\\_escolar.html](http://www.fnde.gov.br/home/index.jsp?arquivo=alimentacao_escolar.html)>. Acessado em 25 de junho de 2009

MASCARENHAS Jean Márcia Oliveira; SANTOS, Juliana Cantalino dos. AVALIAÇÃO DA COMPOSIÇÃO NUTRICIONAL DOS CARDÁPIOS E CUSTOS DA ALIMENTAÇÃO ESCOLAR DA REDE MUNICIPAL DE CONCEIÇÃO DO JACUÍPE/BA. **Sitientibus**, Feira de Santana, n.35, p.75-90, jul./dez. 2006.

PMF. **Alimentação escolar de Florianópolis é municipalizada**, 2009. Disponível em: <[http://www.pmf.sc.gov.br/portal/noticias/comunicacao/?acao=listar&id\\_noticia=5443](http://www.pmf.sc.gov.br/portal/noticias/comunicacao/?acao=listar&id_noticia=5443)>. Acessado em 25 de junho de 2009





```

var $idcar;

var $descricao;

var $nm_arquivo;

var $ref_escola;

var $ref_usuario_cad;

var $dt_cadastro;

function Inicializar()
{
    $retorno = "Novo";

    @session_start();

    $this->peessoa_logada = $_SESSION['id_pessoa'];

    @session_write_close();

    $this->idcar=$_GET["idcar"];

    $obj_permissoes = new clsPermissoes();

    $obj_permissoes->permissao_cadastra( 10000, $this->peessoa_logada,3,
"alimentacao_cardapio_lst.php" );

    if( is_numeric( $this->idcar ) )
    {
        $obj_cardapio = new clsAlimentacaoCardapio();

        $lst = $obj_cardapio->lista($this->idcar,null);

        if (is_array($lst))
        {
            $registro = array_shift($lst);

            if( $registro )
            {
                foreach( $registro AS $campo => $val ) // passa todos os
valores obtidos no registro para atributos do objeto

```

```

        $this->$campo = $val;

        /** verificao de permissao para exclusao
        //$this->fexcluir = $obj_permissoes-
        >permissao_excluir(572,$this->peessoa_logada,7);
        /**

        $retorno = "Editar";
    }else{
        header( "Location: alimentacao_cardapio_lst.php" );
        die();
    }
}

    $this->url_cancelar = ($retorno == "Editar") ? "alimentacao_cardapio_det.php?
idcar={$registro["idcar"]}": "alimentacao_cardapio_lst.php";

    $this->nome_url_cancelar = "Cancelar";
    return $retorno;
}

function Gerar()
{
    // primary keys
    $this->campoOculto( "idcar", $this->idcar );

    // text
    $this->campoTexto( "descricao", "Descrição", $this->descricao, 30, 255, true );

    if($this->idcar > 0)
    {
        $this->campoRotulo("nm_arquivo","Arquivo",{ $this->nm_arquivo});
    }
    else
    {

```

```

        $this->campoArquivo("nm_arquivo", "Arquivo", $this->nm_arquivo, "50");
    }

    $opcoes = array();
    $obj_escola = new clsPmieducarEscola();
    $lista = $obj_escola->lista();

    if( is_array( $lista ) && count( $lista ) )
    {
        foreach ( $lista AS $registro )
        {
            $opcoes[$registro["cod_escola"]] = $registro["nome"];
        }
    }

    $this->campoLista( "ref_escola", "Escola", $opcoes, $this->ref_escola, "", false, "", "", "", true );

    // $this->campoMemo( "desc_funcao", "Descri&ccedil;&atilde;o Func&atilde;o", $this->desc_funcao, 60, 5, false );

    // data

}

function Novo()
{
    @session_start();
    $this->pessoa_logada = $_SESSION['id_pessoa'];
    @session_write_close();

    $caminho = "arquivos/cardapios/";

    if ( !empty($_FILES['nm_arquivo']['name']) )
    {

```

```

$caminho .= date("Y-m-d")."-";
list($usec, $sec) = explode(" ", microtime());
$caminho .= substr(md5("{$usec}{$sec}"), 0, 8);
while (file_exists("{$caminho}"))
{
    $caminho = $caminho . "a";
}
$ext = end(explode(".",$_FILES['nm_arquivo']['name']));
$caminho .= ".$ext;
move_uploaded_file($_FILES['nm_arquivo']['tmp_name'], "{$caminho}");
}
else
{
    $this->mensagem = "Cadastro não realizado.<br>O arquivo é obrigatório.";
    echo "<!--\nErro ao cadastrar clsAlimentacaoEscolar\nArquivo obrigatório-->";
    return false;
}

$obj = new clsAlimentacaoCardapio();
$obj->idcar = $this->idcar;
$obj->dt_cadastro = "NOW()";
$obj->ref_usuario_cad = $this->pessoa_logada;
$obj->ref_escola = $this->ref_escola;
$obj->descricao = $this->descricao;
$obj->nm_arquivo = $_FILES['nm_arquivo']['name'];
$obj->path_arquivo = $caminho;
$cadastrou = $obj->cadastra();
if( $cadastrou )
{
    $this->mensagem .= "Cadastro efetuado com sucesso.<br>";
    header( "Location: alimentacao_cardapio_1st.php" );
    die();
    return true;
}

```

```

    }

    $this->mensagem = "Cadastro não realizado.<br>";
    echo "<!--\nErro ao cadastrar clsAlimentacaoEscolar\n-->";
    return false;
}

function Editar()
{
    @session_start();
    $this->peessoa_logada = $_SESSION['id_pessoa'];
    @session_write_close();

    $obj = new clsAlimentacaoCardapio();
    $obj->idcar = $this->idcar;
    $obj->ref_escola = $this->ref_escola;
    $obj->descricao = $this->descricao;
    $cadastrou = $obj->edita();
    if( $cadastrou )
    {
        $this->mensagem .= "Cadastro editado com sucesso.<br>";
        header( "Location: alimentacao_cardapio_det.php?idcar={$this->idcar}" );
        die();
        return true;
    }

    $this->mensagem = "Cadastro não editado.<br>";
    echo "<!--\nErro ao cadastrar clsAlimentacaoEscolar\n-->";
    return false;
}

```

```

function Excluir()
{
    @session_start();
    $this->peessoa_logada = $_SESSION['id_peessoa'];
    @session_write_close();

    return false;
}
}

```

```
// cria uma extensao da classe base
```

```
$pagina = new clsIndexBase();
```

```
// cria o conteudo
```

```
$miolo = new indice();
```

```
// adiciona o conteudo na clsBase
```

```
$pagina->addForm( $miolo );
```

```
// gera o html
```

```
$pagina->MakeAll();
```

```
?>
```

```
alimentacao_cardapio_det.php
```

```
<?php
```

```
/* *****
```

```
*
```

```
*
```

```
* @author Prefeitura Municipal de Itajaí
```

```
*
```

```
* @updated 29/03/2007
```

```
*
```

```
* Pacote: i-PLB Software Público Livre e Brasileiro
```

```
*
```

```
*
```

```
*
```

```
* Copyright (C) 2006 PMI - Prefeitura Municipal de Itajaí
```

```
*
```

```
*
```

```
ctima@itajai.sc.gov.br
```

```
*
```



```

    }
}

class indice extends clsDetalhe
{
    /**
     * Titulo no topo da pagina
     *
     * @var int
     */
    var $titulo;

    var $idcar;

    function Gerar()
    {
        @session_start();
        $this->peessoa_logada = $_SESSION['id_pessoa'];
        session_write_close();

        $this->titulo = "Cardápio - Detalhe";
        $this->addBanner( "imagens/nvp_top_intranet.jpg", "imagens/nvp_vert_intranet.jpg",
        "Intranet" );

        $this->idcar=$_GET["idcar"];

        $obj_cardapio = new clsAlimentacaoCardapio();
        $lista = $obj_cardapio->lista($this->idcar,null);
        $registro = $lista[0];

        if( ! $registro )
        {
            header( "location: alimentacao_cardapio_lst.php" );

```

```

        die();
    }

    $this->addDetalhe( array( "Descrição", $registro["descricao"] ) );

    $this->addDetalhe( array( "Arquivo", "<a target=\"blank\"
href=\"{$registro["path_arquivo"]}\">{$registro["nm_arquivo"]}</a>" ) );

    $this->addDetalhe( array( "Data Cadastro", date('d/m/Y',strtotime($registro["dt_cadastro"])) ) );

    $obj_ref_cod_escola = new clsPmieducarEscola( $registro["ref_escola"] );
    $det_ref_cod_escola = $obj_ref_cod_escola->detalhe();
    $nm_escola = $det_ref_cod_escola["nome"];

    $obj_pessoa = new clsPessoa_( $registro["ref_usuario_cad"] );
    $det_pessoa = $obj_pessoa->detalhe();
    $nm_pessoa = $det_pessoa["nome"];

    $this->addDetalhe( array( "Usuário", $nm_pessoa ) );
    $this->addDetalhe( array( "Escola", $nm_escola ) );

    $obj_permissoes = new clsPermissoes();
    if( $obj_permissoes->permissao_cadastra( 10000, $this->pessoa_logada, 3 ) )
    {
        $this->url_novo = "alimentacao_cardapio_cad.php";
        if ( $registro["ref_usuario_cad"] == $this->pessoa_logada )
        {
            $this->url_editar = "alimentacao_cardapio_cad.php?idcar={$this->idcar}";
        }
    }
}

```

```

        $this->url_cancelar = "alimentacao_cardapio_1st.php";
        $this->largura = "100%";
    }
}

```

```

// cria uma extensao da classe base
$pagina = new clsIndexBase();
// cria o conteudo
$miolo = new indice();
// adiciona o conteudo na clsBase
$pagina->addForm( $miolo );
// gera o html
$pagina->MakeAll();
?>

```

alimentacao\_cardapio\_1st.php

```
<?php
```

```
/* *****
```

```

*

```

```

*

```

```

*   @author Prefeitura Municipal de Itajaí
*

```

```

*   @updated 29/03/2007
*

```

```

*   Pacote: i-PLB Software Público Livre e Brasileiro
*

```

```

*

```

```

*

```

```

*

```

```

*   Copyright (C) 2006   PMI - Prefeitura Municipal de Itajaí
*

```

```

*

```

```

*

```

```

*   ctima@itajai.sc.gov.br
*

```

```

*

```

```

*

```

```

*   Este programa é software livre, você pode redistribuí-lo e/ou
*

```

```

*   modificá-lo sob os termos da Licença Pública Geral GNU, conforme
*

```



```
{  
  
    /**  
     * Referencia pega da session para o idpes do usuario atual  
     *  
     * @var int  
     */  
    var $pessoa_logada;  
  
    /**  
     * Titulo no topo da pagina  
     *  
     * @var int  
     */  
    var $titulo;  
  
    /**  
     * Quantidade de registros a ser apresentada em cada pagina  
     *  
     * @var int  
     */  
    var $limite;  
  
    /**  
     * Inicio dos registros a serem exibidos (limit)  
     *  
     * @var int  
     */  
    var $offset;  
  
    var $ref_escola;  
  
    function Gerar()  
    {
```

```

@session_start();
$this->peessoa_logada = $_SESSION['id_pessoa'];
session_write_close();

$this->titulo = "Cardápio - Listagem";

foreach( $_GET AS $var => $val ) // passa todos os valores obtidos no GET para
atributos do objeto
    $this->$var = ( $val === "" ) ? null: $val;

$this->addBanner( "imagens/nvp_top_intranet.jpg", "imagens/nvp_vert_intranet.jpg",
"Intranet" );

$obj_permissao = new clsPermissoes();

$lista_busca = array(
    "Descrição",
    "Arquivo",
    "Data Cadastro",
    "Usuário",
    "Escola"
);

$this->addCabecalhos($lista_busca);

$opcoes = array();
$obj_escola = new clsPmieducarEscola();
$lista = $obj_escola->lista();

$opcoes["0"] = "Todas";
if( is_array( $lista ) && count( $lista ) )
{
    foreach ( $lista AS $registro )
    {

```

```

        $opcoes[$registro["cod_escola"]] = $registro["nome"];
    }

}

$this->campoLista( "ref_escola", "Escola", $opcoes, $this-
>ref_escola, "", false, "", "", "", false );

$obj_cardapio = new clsAlimentacaoCardapio();
$filtro_escola = null;
if($this->ref_escola > 0)
{
    $filtro_escola = $this->ref_escola;
}
$lista = $obj_cardapio->lista(null,$filtro_escola);

if( is_array( $lista ) && count( $lista ) )
{
    foreach ( $lista AS $registro )
    {
        $obj_ref_cod_escola = new
clsPmieducarEscola( $registro["ref_escola"] );
        $det_ref_cod_escola = $obj_ref_cod_escola->detalhe();
        $nm_escola = $det_ref_cod_escola["nome"];

        $obj_pessoa = new clsPessoa_( $registro["ref_usuario_cad"] );
        $det_pessoa = $obj_pessoa->detalhe();
        $nm_pessoa = $det_pessoa["nome"];

        $lista_busca = array();
        $lista_busca[] = "<a href='\"alimentacao_cardapio_det.php?
idcar={$registro[\"idcar\"]}'\">{$registro[\"descricao\"]}</a>";
        $lista_busca[] = "<a target='\"blank\"
href='\"{$registro[\"path_arquivo\"]}'\">{$registro[\"nm_arquivo\"]}</a>";
    }
}

```

```

        $lista_busca[] = "<a href=\"alimentacao_cardapio_det.php?
idcar={$registro[\"idcar\"]}\">.date('d/m/Y',strtotime($registro[\"dt_cadastro\"])).\"</a>";

        $lista_busca[] = "<a href=\"alimentacao_cardapio_det.php?
idcar={$registro[\"idcar\"]}\">{$nm_pessoa}</a>";

        $lista_busca[] = "<a href=\"alimentacao_cardapio_det.php?
idcar={$registro[\"idcar\"]}\">{$nm_escola}</a>";

        $this->addLinhas($lista_busca);

    }

}

if( $obj_permissao->permissao_cadastra( 10000, $this->pessoa_logada, 3 ) )
{
    $this->acao = "go(\"alimentacao_cardapio_cad.php\")";
    $this->nome_acao = "Novo";
}

$this->largura = "100%";
}
}

// cria uma extensao da classe base
$pagina = new clsIndexBase();

// cria o conteudo
$miolo = new indice();

// adiciona o conteudo na clsBase
$pagina->addForm( $miolo );

// gera o html
$pagina->MakeAll();

?>

alimentacao_envio_mensal_escola_cad.php

<?php
/* *****

```



```
require_once( "include/pmieducar/geral.inc.php" );
require_once( "include/alimentacao/geral.inc.php" );

class clsIndexBase extends clsBase
{
    function Formular()
    {
        $this->SetTitulo( "{$this->_instituicao} i-Educar - Envio Mensal Escola " );
        $this->processoAp = "10005";
    }
}

class indice extends clsCadastro
{
    /**
     * Referencia pega da session para o idpes do usuario atual
     *
     * @var int
     */
    var $pessoa_logada;

    var $ideme;
    var $ref_escola;
    var $ano;
    var $mes;
    var $alunos;
    var $dt_cadastro;
    var $pesoovolume;
    var $dias;
    var $refeicoes;
    var $javascript;
    var $sacaodropdonw;
```

```

function Inicializar()
{
    $retorno = "Novo";
    @session_start();
    $this-> pessoa_logada = $_SESSION['id_pessoa'];
    @session_write_close();

    $this->ideme=$_GET["ideme"];

    $obj_permissoes = new clsPermissoes();
    $obj_permissoes->permissao_cadastra( 10005, $this->pessoa_logada,3,
"alimentacao_envio_mensal_escola_lst.php" );

    if( is_numeric( $this->ideme ) )
    {
        $obj_envio = new clsAlimentacaoEnvioMensalEscola();
        $lst = $obj_envio->lista($this->ideme);

        if (is_array($lst))
        {
            $registro = array_shift($lst);
            if( $registro )
            {
                foreach( $registro AS $campo => $val ) // passa todos os
valores obtidos no registro para atributos do objeto
                $this->$campo = $val;

                /** verificao de permissao para exclusao
                $this->fexcluir = $obj_permissoes-
>permissao_excluir(10005,$this->pessoa_logada,3);
                /**

                $retorno = "Editar";

```

```

        }else{
            header( "Location:
alimentacao_envio_mensal_escola_lst.php" );
            die();
        }
    }
}
else
{
    $obj_envio_mensal_padroes = new clsAlimentacaoEnvioMensalPadroes();
    $lista = $obj_envio_mensal_padroes->lista();
    if( is_array( $lista ) && count( $lista ) )
    {
        $this->acaodropdonw = "trocaAnoMes()";
        $this->javascript = '<script language="javascript"> var envioPadroes =
new Array();';
        foreach ( $lista AS $registro )
        {
            $this->javascript .= 'envioPadroes['.$registro["ano"].
$registro["mes"].']=new Array( "'.$registro["dias"].'", "'.$registro["refeicoes"].'");';
        }
        $this->javascript .= '</script>';
    }
}

$this->url_cancelar = ($retorno == "Editar" ?
"alimentacao_envio_mensal_escola_det.php?ideme={$registro["ideme"]}":
"alimentacao_envio_mensal_escola_lst.php";

$this->nome_url_cancelar = "Cancelar";

return $retorno;
}

function Gerar()
{
    // primary keys
    $this->campoOculto( "ideme", $this->ideme );
}

```

```

    $opcoes = array();
    $obj_escola = new clsPmieducarEscola();
    $lista = $obj_escola->lista();

    if( is_array( $lista ) && count( $lista ) )
    {
        foreach ( $lista AS $registro )
        {
            $opcoes[$registro["cod_escola"]] = $registro["nome"];
        }
    }

    $this->campoLista( "ref_escola", "Escola", $opcoes, $this->ref_escola, "", false, "", "", "", true );

    $opcoes = array();
    for ($i = 2008; $i <= date("Y");$i++)
    {
        $opcoes[$i] = $i;
    }

    $this->campoLista( "ano", "Ano", $opcoes, $this->ano, $this->acaodropdonw, false, "", "", "", true );

    $obj_envio = new clsAlimentacaoEnvioMensalEscola();

    $this->campoLista( "mes", "Mês", $obj_envio->getArrayMes(), $this->mes, $this->acaodropdonw, false, "", "", "", true );

    $this->campoNumero( "alunos", "Alunos", $this->alunos, 30, 255, true );

    $this->campoNumero( "dias", "Dias", $this->dias, 30, 255, true );

    $this->campoNumero( "refeicoes", "Refeições por dia", $this->refeicoes, 30, 255,
true );

```

```

$Obj = new clsAlimentacaoEnvioMensalEscolaProduto();
$filtro_envio = 0;
if($this->ideme!="")
{
    $filtro_envio = $this->ideme;
}
$registros = $Obj->lista($filtro_envio);

$teste = "<table>";

$teste .= "<tr><td class='formmtdt'><span
class='form'><b>Produto</b></span></td><td class='formmtdt'><span
class='form'><b>Quantidade</b></span></td></tr>";

if( $registros )
{
    foreach ( $registros AS $campo )
    {
        $prod_cad = "0";
        if ( $campo["ref_envio_mensal_escola"] != "" )
        {
            $prod_cad = "1";
        }

        $campo["pesoouvolumen"] =
number_format($campo["pesoouvolumen"],2,"","");

        $teste .= "<tr><td class='formmtdt'><span class='form'>".
$campo["nm_produto"]."</span></td><td><input size='6' maxlength='6' onkeypress='return
aceitaNumeros(this)' onKeyup='formataMonetario(this, event);' class='obrigatorio' type='text'
name='produto[".$campo["idpro"]."]' value='".$campo["pesoouvolumen"]."'><input type='hidden'
name='produto_cad[".$campo["idpro"]."]' value='".$prod_cad."'> ".$campo["unidade"]."</td></tr>";

    }
}

$teste .= "</table>";

$this->campoRotulo("produto","Quantidade por produto",$teste);

```

```

}

function Novo()
{
    @session_start();
    $this->peessoa_logada = $_SESSION['id_pessoa'];
    @session_write_close();

    $obj = new clsAlimentacaoEnvioMensalEscola();

    $lst = $obj->lista(null,$this->ref_escola,$this->ano,$this->mes);
    if (is_array($lst))
    {
        $registro = array_shift($lst);
        if( $registro )
        {
            $this->mensagem = "Cadastro não realizado.<br>Envio já
cadastrado para a escola e período informado.";
            echo "<!--\nErro ao cadastrar
clsAlimentacaoEnvioMensalEscola\n-->";
            return false;
        }
    }

    $obj->ideme = $this->ideme;
    $obj->dt_cadastro = "NOW()";
    $obj->ref_escola = $this->ref_escola;
    $obj->ano = $this->ano;
    $obj->mes = $this->mes;

```

```

$obj->dias = $this->dias;
$obj->refeicoes = $this->refeicoes;
$obj->alunos = $this->alunos;
$cadastrou = $obj->cadastra();
if( $cadastrou )
{
    $obj = new clsAlimentacaoEnvioMensalEscolaProduto();

    if( is_array( $this->produto ) && count( $this->produto ) )
    {
        foreach ( $this->produto AS $campo => $val )
        {
            if($val != "")
            {
                $obj->ref_envio_mensal_escola = $cadastrou;
                $obj->ref_produto = $campo;
                $obj->pesoouvolume = $val;
                $obj->cadastra();
            }
        }
    }

    $this->mensagem .= "Cadastro efetuado com sucesso.<br>";
    header( "Location: alimentacao_envio_mensal_escola_lst.php" );
    die();
    return true;
}

$this->mensagem = "Cadastro não realizado.<br>";
echo "<!--\nErro ao cadastrar clsAlimentacaoEnvioMensalEscola\n-->";
return false;
}

```

```

function Editar()
{
    @session_start();
    $this->peessoa_logada = $_SESSION['id_pessoa'];
    @session_write_close();

    $obj = new clsAlimentacaoEnvioMensalEscola();

    $lst = $obj->lista(null,$this->ref_escola,$this->ano,$this->mes);
    if (is_array($lst))
        {
            $registro = array_shift($lst);
            if( $registro )
                {
                    if($registro["ideme"]!=$this->ideme)
                        {
                            $this->mensagem = "Cadastro não
realizado.<br>Envio já cadastrado para a escola e período informado.";
                            echo "<!--\nErro ao cadastrar
clsAlimentacaoEnvioMensalEscola\n-->";
                            return false;
                        }
                }
        }

    $obj = new clsAlimentacaoEnvioMensalEscola();
    $obj->ideme = $this->ideme;
    $obj->ref_escola = $this->ref_escola;
    $obj->ano = $this->ano;
    $obj->mes = $this->mes;
    $obj->dias = $this->dias;

```

```
$obj->refeicoes = $this->refeicoes;
$obj->alunos = $this->alunos;
$cadastrou = $obj->edita();
if( $cadastrou )
{
    $obj = new clsAlimentacaoEnvioMensalEscolaProduto();

    if( is_array( $this->produto ) && count( $this->produto ) )
    {
        foreach ( $this->produto AS $campo => $val )
        {
            $obj->ref_envio_mensal_escola = $this->ideme;
            $obj->ref_produto = $campo;
            if($val != "")
            {
                $obj->pesoouvolume = str_replace(",",".", $val);
                if($this->produto_cad[$campo]=="1")
                {
                    $obj->edita();
                }
                else
                {
                    $obj->cadastra();
                }
            }
            else
            {
                $obj->exclui();
            }
        }
    }
}
```

```

        $this->mensagem .= "Cadastro editado com sucesso.<br>";
        header( "Location: alimentacao_envio_mensal_escola_det.php?ideme={$this-
>ideme}" );

        die();

        return true;
    }

    $this->mensagem = "Cadastro não editado.<br>";
    echo "<!--\nErro ao cadastrar clsAlimentacaoEnvioMensalEscola\n-->";
    return false;

}

function Excluir()
{
    @session_start();
    $this->peessoa_logada = $_SESSION['id_pessoa'];
    @session_write_close();

    $obj = new clsAlimentacaoEnvioMensalEscolaProduto();
    $obj->ref_envio_mensal_escola = $this->ideme;
    $obj->excluiTudo();
    $obj = new clsAlimentacaoEnvioMensalEscola();
    $obj->ideme = $this->ideme;
    $obj->exclui();

    header( "Location: alimentacao_envio_mensal_escola_lst.php" );
    die();
    return true;
}
}

```

```

// cria uma extensao da classe base
$pagina = new clsIndexBase();
// cria o conteudo
$miolo = new indice();
// adiciona o conteudo na clsBase
$pagina->addForm( $miolo );
// gera o html
$pagina->MakeAll();

?>
<script language="javascript">
function trocaAnoMes()
{
    var ano = document.getElementById("ano").value;
    var mes = document.getElementById("mes").value;
    if(envioPadroes[ano+""+mes])
    {
        document.getElementById("dias").value = envioPadroes[ano+""+mes][0];
        document.getElementById("refeicoes").value = envioPadroes[ano+""+mes][1];
    }
}

function aceitaNumeros(){

    if (document.all) // Internet Explorer
        var tecla = event.keyCode;
    else if(document.layers) // Netscape
        var tecla = e.which;

    if ((tecla > 47 && tecla < 58)) // numeros de 0 a 9
        return true;
}

```

```

else {
    if (tecla != 8) // backspace
        //event.keyCode = 0;
        return false;
    else
        return true;
}
}
</script>

```

```

<?
echo $miolo->javascript;
?>

```

alimentacao\_envio\_mensal\_escola\_det.php

```

<?php
/* *****
*
*
*   @author Prefeitura Municipal de Itajaí
*
*   @updated 29/03/2007
*
* Pacote: i-PLB Software Público Livre e Brasileiro
*
*
*   Copyright (C) 2006   PMI - Prefeitura Municipal de Itajaí
*
*                               ctima@itajai.sc.gov.br
*
*
*
*   Este programa é software livre, você pode redistribuí-lo e/ou
*   modificá-lo sob os termos da Licença Pública Geral GNU, conforme
*   publicada pela Free Software Foundation, tanto a versão 2 da

```

```

*   Licença como (a seu critério) qualquer versão mais nova. *
*
*
*   Este programa é distribuído na expectativa de ser útil, mas SEM *
*   QUALQUER GARANTIA. Sem mesmo a garantia implícita de COMERCIALI- *
*   ZAÇÃO ou de ADEQUAÇÃO A QUALQUER PROPÓSITO EM PARTICULAR. Con-
*
*   sulte a Licença Pública Geral GNU para obter mais detalhes. *
*
*
*   Você deve ter recebido uma cópia da Licença Pública Geral GNU *
*   junto com este programa. Se não, escreva para a Free Software *
*   Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA *
*   02111-1307, USA. *
*
*
*
*****/

require_once ("include/clsBase.inc.php");
require_once ("include/clsDetalhe.inc.php");
require_once ("include/clsBanco.inc.php");
require_once ("include/pmieducar/geral.inc.php" );
require_once ("include/alimentacao/geral.inc.php" );

class clsIndexBase extends clsBase
{
    function Formular()
    {
        $this->SetTitulo( "{$this->_instituicao} i-Educar - Envio Mensal Escola" );
        $this->processoAp = "10005";
    }
}

class indice extends clsDetalhe
{

```

```

/**
 * Titulo no topo da pagina
 *
 * @var int
 */
var $titulo;

var $ideme;

function Gerar()
{
    @session_start();
    $this->peessoa_logada = $_SESSION['id_pessoa'];
    session_write_close();

    $this->titulo = "Envio Mensal Escola - Detalhe";
    $this->addBanner( "imagens/nvp_top_intranet.jpg", "imagens/nvp_vert_intranet.jpg",
    "Intranet" );

    $this->ideme=$_GET["ideme"];

    $obj_envio = new clsAlimentacaoEnvioMensalEscola();
    $lista = $obj_envio->lista($this->ideme);
    $registro = $lista[0];

    if( ! $registro )
    {
        header( "location: alimentacao_envio_mensal_escola_lst.php" );
        die();
    }

    $obj_ref_cod_escola = new clsPmieducarEscola( $registro["ref_escola"] );

```

```

$det_ref_cod_escola = $obj_ref_cod_escola->detalhe();
$nm_escola = $det_ref_cod_escola["nome"];

$this->addDetalhe( array( "Escola", $nm_escola ) );

$this->addDetalhe( array( "Ano", $registro["ano"] ) );

$this->addDetalhe( array( "Mês", $obj_envio->getMes($registro["mes"]) ) );

$this->addDetalhe( array( "Alunos", $registro["alunos"] ) );

$this->addDetalhe( array( "Data Cadastro",
date('d/m/Y',strtotime($registro["dt_cadastro"])) ) );

$this->addDetalhe( array( "Dias", $registro["dias"] ) );

$this->addDetalhe( array( "Refeições por dia", $registro["refeicoes"] ) );

$obj = new clsAlimentacaoEnvioMensalEscolaProduto();
$registros = $obj->lista($this->ideme);

$teste = "<table cellpadding='3'>";

$teste .= "<tr><td class='formmtd'><span
class='form'><b>Produto</b></span></td><td class='formmtd'><span
class='form'><b>Quantidade</b></span></td></tr>";

if( $registros )
{
    foreach ( $registros AS $campo )
    {
        $prod_cad = "0";
        if ( $campo["ref_envio_mensal_escola"] != "" )
        {
            $prod_cad = "1";

```

```

    }

    $campo["pesoouvolume"] =
number_format($campo["pesoouvolume"],2,".", "");

    $teste .= "<tr><td class='formmtdtd'><span class='form'>".
$campo["nm_produto"]."</span></td><td class='formmtdtd'><span class='form'>".
$campo["pesoouvolume"]." ".$campo["unidade"]."</span></td></tr>";

    }
}
$teste .= "</table>";

$this->addDetalhe( array("Quantidade por produto",$teste));

$obj_permissoes = new clsPermissoes();
if( $obj_permissoes->permissao_cadastra( 10005, $this->peessoa_logada, 3 ) )
{
    $this->url_novo = "alimentacao_envio_mensal_escola_cad.php";
    $this->url_editar = "alimentacao_envio_mensal_escola_cad.php?ideme={$this->ideme}";
}

$this->url_cancelar = "alimentacao_envio_mensal_escola_lst.php";
$this->largura = "100%";
}
}

// cria uma extensao da classe base
$pagina = new clsIndexBase();

// cria o conteudo
$miolo = new indice();

// adiciona o conteudo na clsBase
$pagina->addForm( $miolo );

// gera o html
$pagina->MakeAll();

```



```

*      02111-1307, USA.
*
*
*
*****/

require_once ("include/clsBase.inc.php");
require_once ("include/clsListagem.inc.php");
require_once ("include/clsBanco.inc.php");
require_once ("include/pmieducar/geral.inc.php" );
require_once ("include/alimentacao/geral.inc.php" );

class clsIndexBase extends clsBase
{
    function Formular()
    {
        {
            $this->SetTitulo( "{$this->_instituicao} i-Educar - Envio Mensal Escola " );
            $this->processoAp = "10005";
        }
    }
}

class indice extends clsListagem
{
    /**
     * Referencia pega da session para o idpes do usuario atual
     *
     * @var int
     */
    var $pessoa_logada;

    /**
     * Titulo no topo da pagina
     *
     * @var int
     */

```

```

var $titulo;

/**
 * Quantidade de registros a ser apresentada em cada pagina
 *
 * @var int
 */
var $limite;

/**
 * Inicio dos registros a serem exibidos (limit)
 *
 * @var int
 */
var $offset;

var $ref_escola;
var $ano;
var $mes;

function Gerar()
{
    @session_start();
    $this->peessoa_logada = $_SESSION['id_pessoa'];
    session_write_close();

    $this->titulo = "Envio Mensal Escola - Listagem";

    foreach( $_GET AS $var => $val ) // passa todos os valores obtidos no GET para
atributos do objeto
        $this->$var = ( $val === "" ) ? null: $val;

    $this->addBanner( "imagens/nvp_top_intranet.jpg", "imagens/nvp_vert_intranet.jpg",
"Intranet" );

```

```

$obj_permissao = new clsPermissoes();

$lista_busca = array(
    "Escola",
    "Ano",
    "Mês",
    "Alunos",
    "Data Cadastro"
);

$this->addCabecalhos($lista_busca);

$opcoes = array();
$obj_escola = new clsPmieducarEscola();
$lista = $obj_escola->lista();

$opcoes["0"] = "Todas";
if( is_array( $lista ) && count( $lista ) )
{
    foreach ( $lista AS $registro )
    {
        $opcoes[$registro["cod_escola"]] = $registro["nome"];
    }
}

$this->campoLista( "ref_escola", "Escola", $opcoes, $this-
>ref_escola, "", false, "", "", "", false );

$opcoes = array();
$opcoes[0] = "Todos";
for ($i = 2008; $i <= date("Y");$i++)

```

```
{
    $opcoes[$i] = $i;
}
$this->campoLista( "ano", "Ano", $opcoes, $this->ano,"",false,"","",false );

$obj_envio = new clsAlimentacaoEnvioMensalEscola();

$opcoes = array();
$opcoes = $obj_envio->getArrayMes();
$opcoes[0] = "Todos";

$this->campoLista( "mes", "Mês", $opcoes, $this->mes,"",false,"","",false );

$filtro_escola = null;
if($this->ref_escola > 0)
{
    $filtro_escola = $this->ref_escola;
}
$filtro_ano = null;
if($this->ano > 0)
{
    $filtro_ano = $this->ano;
}
$filtro_mes = null;
if($this->mes > 0)
{
    $filtro_mes = $this->mes;
}
$lista = $obj_envio->lista(null,$filtro_escola,$filtro_ano,$filtro_mes);

if( is_array( $lista ) && count( $lista ) )
```

```

    {
        foreach ( $lista AS $registro )
        {
            $obj_ref_cod_escola = new
clsPmieducarEscola( $registro["ref_escola"] );

            $det_ref_cod_escola = $obj_ref_cod_escola->detalhe();

            $nm_escola = $det_ref_cod_escola["nome"];

            $lista_busca = array();

            $lista_busca[] = "<a
href=\"alimentacao_envio_mensal_escola_det.php?ideme={$registro["ideme"]}\">{$nm_escola}</a>";

            $lista_busca[] = "<a
href=\"alimentacao_envio_mensal_escola_det.php?
ideme={$registro["ideme"]}\">{$registro["ano"]}</a>";

            $lista_busca[] = "<a
href=\"alimentacao_envio_mensal_escola_det.php?ideme={$registro["ideme"]}\">".Obj_envio-
>getMes($registro["mes"])."</a>";

            $lista_busca[] = "<a
href=\"alimentacao_envio_mensal_escola_det.php?
ideme={$registro["ideme"]}\">{$registro["alunos"]}</a>";

            $lista_busca[] = "<a
href=\"alimentacao_envio_mensal_escola_det.php?
ideme={$registro["ideme"]}\">".date('d/m/Y',strtotime($registro["dt_cadastro"]))."</a>";

            $this->addLinhas($lista_busca);
        }
    }

    if( $obj_permissao->permissao_cadastra( 10005, $this->pessoa_logada, 3 ) )
    {
        $this->acao = "go(\"alimentacao_envio_mensal_escola_cad.php\")";
        $this->nome_acao = "Novo";
    }

    $this->largura = "100%";
}

```





```
*/  
  
var $pessoa_logada;  
  
var $idemp;  
var $ano;  
var $mes;  
var $dias;  
var $refeicoes;  
  
function Inicializar()  
{  
    $retorno = "Novo";  
    @session_start();  
    $this->pessoa_logada = $_SESSION['id_pessoa'];  
    @session_write_close();  
  
    $this->idemp=$_GET["idemp"];  
  
    $obj_permissoes = new clsPermissoes();  
    $obj_permissoes->permissao_cadastra( 10011, $this->pessoa_logada,3,  
"alimentacao_envio_mensal_padroes_lst.php" );  
  
    if( is_numeric( $this->idemp ) )  
    {  
        $obj_envio_mensal_padroes = new clsAlimentacaoEnvioMensalPadroes();  
        $lst = $obj_envio_mensal_padroes->lista($this->idemp);  
  
        if( is_array($lst) )  
        {  
            $registro = array_shift($lst);  
            if( $registro )  
            {
```

foreach( \$registro AS \$campo => \$val ) // passa todos os valores obtidos no registro para atributos do objeto

```
$this->$campo = $val;
```

```
/** verificao de permissao para exclusao
```

```
$this->fexcluir = $obj_permissoes->permissao_excluir(10011,$this->peessoa_logada,3);
```

```
/**
```

```
$retorno = "Editar";
```

```
}else{
```

```
header( "Location: alimentacao_envio_mensal_padroes_lst.php" );
```

```
die();
```

```
}
```

```
}
```

```
}
```

```
$this->url_cancelar = ($retorno == "Editar") ?  
"alimentacao_envio_mensal_padroes_det.php?idemp={$registro["idemp"]}" :  
"alimentacao_envio_mensal_padroes_lst.php";
```

```
$this->nome_url_cancelar = "Cancelar";
```

```
return $retorno;
```

```
}
```

```
function Gerar()
```

```
{
```

```
// primary keys
```

```
$this->campoOculto( "idemp", $this->idemp );
```

```
$opcoes = array();
```

```
for ($i = 2008; $i <= date("Y");$i++)
```

```
{
```

```
$opcoes[$i] = $i;
```

```
}
```

```

$this->campoLista( "ano", "Ano", $opcoes, $this->ano, "", false, "", "", "", true );

$obj_envio_mensal_padroes = new clsAlimentacaoEnvioMensalPadroes();

$this->campoLista( "mes", "Mês", $obj_envio_mensal_padroes->getArrayMes(), $this-
>mes, "", false, "", "", "", true );

$this->campoNumero( "dias", "Dias", $this->dias, 30, 255, true );

$this->campoNumero( "refeicoes", "Refeições por dia", $this->refeicoes, 30, 255,
true );

}

function Novo()
{
    @session_start();
    $this->pessoa_logada = $_SESSION['id_pessoa'];
    @session_write_close();

    $obj = new clsAlimentacaoEnvioMensalPadroes();
    $obj->ano = $this->ano;
    $obj->mes = $this->mes;
    $obj->dias = $this->dias;
    $obj->refeicoes = $this->refeicoes;
    $cadastrou = $obj->cadastra();
    if( $cadastrou )
    {
        $this->mensagem .= "Cadastro efetuado com sucesso.<br>";
        header( "Location: alimentacao_envio_mensal_padroes_lst.php" );
        die();
    }
}

```

```

        return true;
    }

    $this->mensagem = "Cadastro não realizado.<br>";
    echo "<!--\nErro ao cadastrar clsAlimentacaoEnvioMensalPadroes\n-->";
    return false;
}

function Editar()
{
    @session_start();
    $this-> pessoa_logada = $_SESSION['id_pessoa'];
    @session_write_close();

    $obj = new clsAlimentacaoEnvioMensalPadroes();
    $obj->idemp = $this->idemp;
    $obj->ano = $this->ano;
    $obj->mes = $this->mes;
    $obj->dias = $this->dias;
    $obj->refeicoes = $this->refeicoes;
    $cadastrou = $obj->edita();
    if( $cadastrou )
    {
        $this->mensagem .= "Cadastro editado com sucesso.<br>";
        header( "Location: alimentacao_envio_mensal_padroes_det.php?idemp={$this->idemp}" );
        die();
        return true;
    }

    $this->mensagem = "Cadastro não editado.<br>";
    echo "<!--\nErro ao cadastrar clsAlimentacaoEnvioMensalPadroes\n-->";
    return false;
}

```

```

    }

function Excluir()
{
    @session_start();
    $this->peessoa_logada = $_SESSION['id_pessoa'];
    @session_write_close();

    $obj = new clsAlimentacaoEnvioMensalPadroes();
    $obj->idemp= $this->idemp;
    $excluiu = $obj->exclui();
    if( $excluiu )
    {
        $this->mensagem .= "Cadastro excluído com sucesso.<br>";
        header( "Location: alimentacao_envio_mensal_padroes_lst.php" );
        die();
        return true;
    }

    $this->mensagem = "Cadastro não excluído.<br>";
    echo "<!--\nErro ao cadastrar clsAlimentacaoEnvioMensalPadroes\n-->";
    return false;

    return false;
}
}

// cria uma extensao da classe base
$pagina = new clsIndexBase();

// cria o conteudo
$miolo = new indice();

```

```
// adiciona o conteudo na clsBase
$pagina->addForm( $miolo );
// gera o html
$pagina->MakeAll();
?>
```

alimentacao\_envio\_mensal\_padroes\_det.php

```
<?php
```

```
/* **** */
```

```
*
```

```
*
```

```
* @author Prefeitura Municipal de Itajaí
```

```
*
```

```
* @updated 29/03/2007
```

```
*
```

```
* Pacote: i-PLB Software Público Livre e Brasileiro
```

```
*
```

```
*
```

```
*
```

```
* Copyright (C) 2006 PMI - Prefeitura Municipal de Itajaí
```

```
*
```

```
*
```

```
ctima@itajai.sc.gov.br
```

```
*
```

```
*
```

```
*
```

```
* Este programa é software livre, você pode redistribuí-lo e/ou *
```

```
* modificá-lo sob os termos da Licença Pública Geral GNU, conforme *
```

```
* publicada pela Free Software Foundation, tanto a versão 2 da *
```

```
* Licença como (a seu critério) qualquer versão mais nova. *
```

```
*
```

```
*
```

```
* Este programa é distribuído na expectativa de ser útil, mas SEM *
```

```
* QUALQUER GARANTIA. Sem mesmo a garantia implícita de COMERCIALI- *
```

```
* ZAÇÃO ou de ADEQUAÇÃO A QUALQUER PROPÓSITO EM PARTICULAR. Con- *
```

```
*
```

```
* sulte a Licença Pública Geral GNU para obter mais detalhes. *
```

```
*
```

```
*
```



```

function Gerar()
{
    @session_start();
    $this->peessoa_logada = $_SESSION['id_pessoa'];
    session_write_close();

    $this->titulo = "Envio Mensal Padrões - Detalhe";
    $this->addBanner( "imagens/nvp_top_intranet.jpg", "imagens/nvp_vert_intranet.jpg",
    "Intranet" );

    $this->idemp=$_GET["idemp"];

    $obj_envio_mensal_padroes = new clsAlimentacaoEnvioMensalPadroes();
    $lista = $obj_envio_mensal_padroes->lista($this->idemp);
    $registro = $lista[0];

    if( ! $registro )
    {
        header( "location: alimentacao_envio_mensal_padroes_lst.php" );
        die();
    }

    $this->addDetalhe( array( "Ano", $registro["ano"] ) );

    $this->addDetalhe( array( "Mês", $obj_envio_mensal_padroes-
    >getMes($registro["mes"])) );

    $this->addDetalhe( array( "Dias", $registro["dias"] ) );

    $this->addDetalhe( array( "Refeições por dia", $registro["refeicoes"] ) );

    $obj_permissoes = new clsPermissoes();
    if( $obj_permissoes->permissao_cadastra( 10011, $this->peessoa_logada, 3 ) )
    {

```

```

        $this->url_novo = "alimentacao_envio_mensal_padroes_cad.php";
        $this->url_editar = "alimentacao_envio_mensal_padroes_cad.php?idemp={$this->idemp}";

    }

    $this->url_cancelar = "alimentacao_envio_mensal_padroes_lst.php";
    $this->largura = "100%";
}
}

```

```
// cria uma extensao da classe base
```

```
$pagina = new clsIndexBase();
```

```
// cria o conteudo
```

```
$miolo = new indice();
```

```
// adiciona o conteudo na clsBase
```

```
$pagina->addForm( $miolo );
```

```
// gera o html
```

```
$pagina->MakeAll();
```

```
?>
```

```
alimentacao_envio_mensal_padroes_lst.php
```

```
<?php
```

```
/* *****
```

```
*

```

```
*

```

```
*   @author Prefeitura Municipal de Itajaí

```

```
*

```

```
*   @updated 29/03/2007

```

```
*

```

```
*   Pacote: i-PLB Software Público Livre e Brasileiro

```

```
*

```

```
*

```

```
*

```

```
*   Copyright (C) 2006   PMI - Prefeitura Municipal de Itajaí

```

```
*

```



```
        $this->processoAp = "10011";
    }
}

class indice extends clsListagem
{
    /**
     * Referencia pega da session para o idpes do usuario atual
     *
     * @var int
     */
    var $pessoa_logada;

    /**
     * Titulo no topo da pagina
     *
     * @var int
     */
    var $titulo;

    /**
     * Quantidade de registros a ser apresentada em cada pagina
     *
     * @var int
     */
    var $limite;

    /**
     * Inicio dos registros a serem exibidos (limit)
     *
     * @var int
     */
    var $offset;
```

```

var $ano;

var $mes;

function Gerar()
{
    @session_start();
    $this->peessoa_logada = $_SESSION['id_pessoa'];
    session_write_close();

    $this->titulo = "Envio Mensal Padrões - Listagem";

    foreach( $_GET AS $var => $val ) // passa todos os valores obtidos no GET para
atributos do objeto
        $this->$var = ( $val === "" ) ? null: $val;

    $this->addBanner( "imagens/nvp_top_intranet.jpg", "imagens/nvp_vert_intranet.jpg",
    "Intranet" );

    $obj_permissao = new clsPermissoes();

    $lista_busca = array(
        "Ano",
        "Mês",
        "Dias",
        "Refeições por dia"
    );

    $this->addCabecalhos($lista_busca);

    $opcoes = array();
    $opcoes[0] = "Todos";
    for ($i = 2008; $i <= date("Y");$i++)

```

```
{
    $opcoes[$i] = $i;
}
$this->campoLista( "ano", "Ano", $opcoes, $this->ano,"",false,"","",false );

$obj_envio = new clsAlimentacaoEnvioMensalEscola();

$opcoes = array();
$opcoes = $obj_envio->getArrayMes();
$opcoes[0] = "Todos";

$this->campoLista( "mes", "Mês", $opcoes, $this->mes,"",false,"","",false );

$obj_envio_mensal_padroes = new clsAlimentacaoEnvioMensalPadroes();
$filtro_ano = null;
if($this->ano > 0)
{
    $filtro_ano = $this->ano;
}
$filtro_mes = null;
if($this->mes > 0)
{
    $filtro_mes = $this->mes;
}
$lista = $obj_envio_mensal_padroes->lista(null,$filtro_ano,$filtro_mes);

if( is_array( $lista ) && count( $lista ) )
{
    foreach ( $lista AS $registro )
    {
```

```

        $lista_busca = array();

        $lista_busca[] = "<a
href=\"alimentacao_envio_mensal_padroes_det.php?
idemp={\$registro[\"idemp\"]}\">{\$registro[\"ano\"]}</a>";

        $lista_busca[] = "<a
href=\"alimentacao_envio_mensal_padroes_det.php?
idemp={\$registro[\"idemp\"]}\">{\$obj_envio_mensal_padroes->getMes(\$registro[\"mes\"])}</a>";

        $lista_busca[] = "<a
href=\"alimentacao_envio_mensal_padroes_det.php?
idemp={\$registro[\"idemp\"]}\">{\$registro[\"dias\"]}</a>";

        $lista_busca[] = "<a
href=\"alimentacao_envio_mensal_padroes_det.php?
idemp={\$registro[\"idemp\"]}\">{\$registro[\"refeicoes\"]}</a>";

        $this->addLinhas($lista_busca);

    }

}

if( $obj_permissao->permissao_cadastra( 10011, $this->peessoa_logada, 3 ) )
{
    $this->acao = "go(\"alimentacao_envio_mensal_padroes_cad.php\")";
    $this->nome_acao = "Novo";
}

$this->largura = "100%";
}
}

// cria uma extensao da classe base
$pagina = new clsIndexBase();

// cria o conteudo
$miolo = new indice();

// adiciona o conteudo na clsBase
$pagina->addForm( $miolo );

// gera o html
$pagina->MakeAll();

?>

```



```

*****/

require_once ("include/clsBase.inc.php");
require_once ("include/clsCadastro.inc.php");
require_once ("include/clsBanco.inc.php");
require_once ("include/pmieducar/geral.inc.php" );
require_once ("include/alimentacao/geral.inc.php" );

class clsIndexBase extends clsBase
{
    function Formular()
    {
        $this->SetTitulo( "{$this->_instituicao} i-Educar - Fornecedor " );
        $this->processoAp = "10003";
    }
}

class indice extends clsCadastro
{
    /**
     * Referencia pega da session para o idpes do usuario atual
     *
     * @var int
     */
    var $pessoa_logada;

    var $idpes;
    var $descricao;
    var $cnpj;

    function Inicializar()
    {
        $retorno = "Novo";
        @session_start();
    }
}

```

```

$this->peessoa_logada = $_SESSION['id_pessoa'];
@session_write_close();

$this->idpes=$_GET["idpes"];

$obj_permissoes = new clsPermissoes();
$obj_permissoes->permissao_cadastra( 10003, $this->peessoa_logada,3,
"alimentacao_fornecedor_lst.php" );

if( is_numeric( $this->idpes ) )
{
    $obj_fornecedor = new clsAlimentacaoFornecedor();
    $lst = $obj_fornecedor->lista($this->idpes);

    if (is_array($lst))
    {
        $registro = array_shift($lst);
        if( $registro )
        {
            foreach( $registro AS $campo => $val ) // passa todos os
valores obtidos no registro para atributos do objeto
                $this->$campo = $val;

            /** verificao de permissao para exclusao
            //$this->fexcluir = $obj_permissoes-
->permissao_excluir(572,$this->peessoa_logada,7);
            /**

            $retorno = "Editar";
        }else{
            header( "Location: alimentacao_fornecedor_lst.php" );
            die();
        }
    }
}

```

```

        }
    }
    $this->url_cancelar = ($retorno == "Editar") ? "alimentacao_fornecedor_det.php?idpes={$registro["idpes"]}" : "alimentacao_produto_lst.php";
    $this->nome_url_cancelar = "Cancelar";
    return $retorno;
}

function Gerar()
{
    // primary keys
    $this->campoOculto( "idpes", $this->idpes );

    if($this->idpes > 0)
    {
        $this->campoRotulo("cnpj","Fornecedor","{$this->cnpj}");
    }
    else
    {
        $parametros = new clsParametrosPesquisas();
        $parametros->setSubmit( 0 );
        $parametros->setPessoa( 'J' );
        $parametros->setPessoaNovo( "S" );
        $parametros->setPessoaCPF("N");
        $parametros->setPessoaTela('window');
        $parametros->setCodSistema(13);
        $parametros->adicionaCampoTexto( "cnpj", "cnpj" );
        $this->campoCnpjPesq( "cnpj", "CNPJ", $this->cnpj,
"pesquisa_pessoa_lst.php", $parametros->serializaCampos(), true );
    }

    // text
    $this->campoTexto( "descricao", "Descrição", $this->descricao, 30, 255, true );
}

```

```
}

function Novo()
{
    @session_start();
    $this->peessoa_logada = $_SESSION['id_pessoa'];
    @session_write_close();

    $obj = new clsAlimentacaoFornecedor();

    $pCnpj = null;
    $pCnpj = $this->cnpj;
    $pCnpj = str_replace("-", "", str_replace("/", "", str_replace(".", "", $pCnpj)));
    if(!is_numeric($pCnpj))
    {
        $pCnpj = null;
    }
    if (!is_numeric($pCnpj))
    {
        $this->mensagem = "Cadastro não realizado.<br>CNPJ inválido.";
        return false;
    }

    $lst = $obj->lista(null,$pCnpj);
    if (is_array($lst))
    {
        $registro = array_shift($lst);
        if( $registro )
        {
```

```

        $this->mensagem = "Cadastro não realizado.<br>Fornecedor já
cadastrado.";
        return false;
    }
}

$obj_juridica = new clsJuridica(null,$pCnpj);
$det = $obj_juridica->detalhe();
if(!is_array($det))
{
    $this->mensagem = "Cadastro não realizado.<br>CNPJ não está cadastrado
como empresa.";
    return false;
}

$obj->idpes = $det["idpes"];
$obj->descricao = $this->descricao;
$cadastrou = $obj->cadastra();
if( $cadastrou )
{
    $this->mensagem .= "Cadastro efetuado com sucesso.<br>";
    header( "Location: alimentacao_fornecedor_1st.php" );
    die();
    return true;
}

$this->mensagem = "Cadastro não realizado.<br>";
echo "<!--\nErro ao cadastrar clsAlimentacaoProduto\n-->";
return false;
}

function Editar()
{

```

```

        @session_start();
        $this->peessoa_logada = $_SESSION['id_pessoa'];
        @session_write_close();

        $obj = new clsAlimentacaoFornecedor();
        $obj->idpes = $this->idpes;
        $obj->descricao = $this->descricao;
        $cadastrou = $obj->edita();
        if( $cadastrou )
        {
            $this->mensagem .= "Cadastro editado com sucesso.<br>";
            header( "Location: alimentacao_fornecedor_det.php?idpes={$this->idpes}" );
            die();
            return true;
        }

        $this->mensagem = "Cadastro não editado.<br>";
        echo "<!--\nErro ao cadastrar clsAlimentacaoProduto\n-->";
        return false;
    }

function Excluir()
{
    @session_start();
    $this->peessoa_logada = $_SESSION['id_pessoa'];
    @session_write_close();

    return false;
}
}

```





```

var $titulo;

var $idpes;

function Gerar()
{
    @session_start();
    $this->peessoa_logada = $_SESSION['id_pessoa'];
    session_write_close();

    $this->titulo = "Fornecedor - Detalhe";
    $this->addBanner( "imagens/nvp_top_intranet.jpg", "imagens/nvp_vert_intranet.jpg",
    "Intranet" );

    $this->idpes=$_GET["idpes"];

    $obj_fornecedor = new clsAlimentacaoFornecedor();
    $lista = $obj_fornecedor->lista($this->idpes,null);
    $registro = $lista[0];

    if( ! $registro )
    {
        header( "location: alimentacao_fornecedor_1st.php" );
        die();
    }

    $this->addDetalhe( array( "Fornecedor", $registro["fantasia"] ) );

    $this->addDetalhe( array( "CNPJ", $registro["cnpj"] ) );

    $this->addDetalhe( array( "Descrição", $registro["descricao"] ) );

```

```

$obj_permissoes = new clsPermissoes();
if( $obj_permissoes->permissao_cadastra( 10003, $this-> pessoa_logada, 3 ) )
{
    $this->url_novo = "alimentacao_fornecedor_cad.php";
    $this->url_editar = "alimentacao_fornecedor_cad.php?idpes={$this->idpes}";
}

    $this->url_cancelar = "alimentacao_fornecedor_lst.php";
    $this->largura = "100%";
}
}

```

```
// cria uma extensao da classe base
```

```
$pagina = new clsIndexBase();
```

```
// cria o conteudo
```

```
$miolo = new indice();
```

```
// adiciona o conteudo na clsBase
```

```
$pagina->addForm( $miolo );
```

```
// gera o html
```

```
$pagina->MakeAll();
```

```
?>
```

```
alimentacao_fornecedor_lst.php
```

```
<?php
```

```
/* *****
```

```
*

```

```
*

```

```
* @author Prefeitura Municipal de Itajaí

```

```
*

```

```
* @updated 29/03/2007

```

```
*

```

```
* Pacote: i-PLB Software Público Livre e Brasileiro

```

```
*

```

```
*

```

```

*      Copyright (C) 2006      PMI - Prefeitura Municipal de Itajaí      *
*
*                               ctima@itajai.sc.gov.br
*
*                               *
*
*      Este programa é software livre, você pode redistribuí-lo e/ou *
*      modificá-lo sob os termos da Licença Pública Geral GNU, conforme *
*      publicada pela Free Software Foundation, tanto a versão 2 da *
*      Licença como (a seu critério) qualquer versão mais nova. *
*
*                               *
*
*      Este programa é distribuído na expectativa de ser útil, mas SEM *
*      QUALQUER GARANTIA. Sem mesmo a garantia implícita de COMERCIALI- *
*      ZAÇÃO ou de ADEQUAÇÃO A QUALQUER PROPÓSITO EM PARTICULAR. Con- *
*
*      sulte a Licença Pública Geral GNU para obter mais detalhes. *
*
*                               *
*
*      Você deve ter recebido uma cópia da Licença Pública Geral GNU *
*      junto com este programa. Se não, escreva para a Free Software *
*      Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA *
*      02111-1307, USA.
*
*                               *

```

```

*****/

```

```

require_once ("include/clsBase.inc.php");
require_once ("include/clsListagem.inc.php");
require_once ("include/clsBanco.inc.php");
require_once ("include/pmieducar/geral.inc.php" );
require_once ("include/alimentacao/geral.inc.php" );

```

```

class clsIndexBase extends clsBase

```

```

{

```

```

    function Formular()

```

```

    {

```

```
        $this->SetTitulo( "{$this->_instituicao} i-Educar - Fornecedor " );  
        $this->processoAp = "10003";  
    }  
}
```

```
class indice extends clsListagem
```

```
{  
    /**  
     * Referencia pega da session para o idpes do usuario atual  
     *  
     * @var int  
     */  
    var $pessoa_logada;  
  
    /**  
     * Titulo no topo da pagina  
     *  
     * @var int  
     */  
    var $titulo;  
  
    /**  
     * Quantidade de registros a ser apresentada em cada pagina  
     *  
     * @var int  
     */  
    var $limite;  
  
    /**  
     * Inicio dos registros a serem exibidos (limit)  
     *  
     * @var int  
     */  
}
```

```

var $offset;

var $cnpj;

function Gerar()
{
    @session_start();
    $this->peessoa_logada = $_SESSION['id_pessoa'];
    session_write_close();

    $this->titulo = "Fornecedor - Listagem";

    foreach( $_GET AS $var => $val ) // passa todos os valores obtidos no GET para
atributos do objeto
        $this->$var = ( $val === "" ) ? null: $val;

    $this->addBanner( "imagens/nvp_top_intranet.jpg", "imagens/nvp_vert_intranet.jpg",
"Intranet" );

    $obj_permissao = new clsPermissoes();

    $lista_busca = array(
        "Fornecedor",
        "CNPJ",
        "Descrição"
    );

    $this->addCabecalhos($lista_busca);

    $parametros = new clsParametrosPesquisas();
    $parametros->setSubmit( 1 );
    $parametros->setPessoa( 'J' );
    //$parametros->setPessoaCampo('sem_cnpj');
    $parametros->setPessoaNovo( "S" );

```

```

$parametros->setPessoaCPF("N");
$parametros->setPessoaTela('window');
//$this->campoOculto( "sem_cnpj", "" );
$parametros->setCodSistema(13);
$parametros->adicionaCampoTexto( "cnpj", "cnpj" );

$this->campoCnpjPesq( "cnpj", "CNPJ", $this->cnpj, "pesquisa_pessoa_lst.php",
$parametros->serializaCampos(), true );

```

```

$obj_fornecedor = new clsAlimentacaoFornecedor();
$filtro_cnpj = null;
if($this->cnpj != "")
{
    $filtro_cnpj = str_replace("/", "", str_replace("-", "", str_replace(".", "", $this-
>cnpj)));

    if ( ! is_numeric(filtro_cnpj))
    {
        $filtro_cnpj = null;
    }
}

$lista = $obj_fornecedor->lista(null, $filtro_cnpj);

if( is_array( $lista ) && count( $lista ) )
{
    foreach ( $lista AS $registro )
    {
        $lista_busca = array();

        $lista_busca[] = "<a href=\"alimentacao_fornecedor_det.php?
idpes={$registro[\"idpes\"]}\">{$registro[\"fantasia\"]}</a>";

        $lista_busca[] = "<a href=\"alimentacao_fornecedor_det.php?
idpes={$registro[\"idpes\"]}\">{$registro[\"cnpj\"]}</a>";

        $lista_busca[] = "<a href=\"alimentacao_fornecedor_det.php?
idpes={$registro[\"idpes\"]}\">{$registro[\"descricao\"]}</a>";

        $this->addLinhas($lista_busca);
    }
}

```

```

    }

    if( $obj_permissao->permissao_cadastra( 10003, $this->peessoa_logada, 3 ) )
    {
        $this->acao = "go(\"alimentacao_fornecedor_cad.php\")";
        $this->nome_acao = "Novo";
    }

    $this->largura = "100%";
}
}

// cria uma extensao da classe base
$pagina = new clsIndexBase();

// cria o conteudo
$miolo = new indice();

// adiciona o conteudo na clsBase
$pagina->addForm( $miolo );

// gera o html
$pagina->MakeAll();

?>

```

alimentacao\_imc\_cad.php

```
<?php
```

```
/* ****
```

```
*

```

```
*

```

```
* @author Prefeitura Municipal de Itaja?

```

```
* @updated 29/03/2007

```

```
* Pacote: i-PLB Software P?blico Livre e Brasileiro

```

```
*

```

```
*

```

```
* Copyright (C) 2006 PMI - Prefeitura Municipal de Itaja?

```

```
*

```

```

*                               ctima@itajai.sc.gov.br
*
*
*                               *
*
* Este programa ? software livre, voc? pode redistribu?-lo e/ou *
* modific?-lo sob os termos da Licen?a P?blica Geral GNU, conforme *
* publicada pela Free Software Foundation, tanto a vers?o 2 da *
* Licen?a como (a seu crit?rio) qualquer vers?o mais nova. *
*
*                               *
*
* Este programa ? distribu?do na expectativa de ser ?til, mas SEM *
* QUALQUER GARANTIA. Sem mesmo a garantia impl?cita de COMERCIALI- *
* ZA??O ou de ADEQUA??O A QUALQUER PROP?SITO EM PARTICULAR. Con-
*
* sulte a Licen?a P?blica Geral GNU para obter mais detalhes. *
*
*                               *
*
* Voc? deve ter recebido uma c?pia da Licen?a P?blica Geral GNU *
* junto com este programa. Se n?o, escreva para a Free Software *
* Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA *
* 02111-1307, USA.
*
*                               *
*
* *****/

```

```

require_once ("include/clsBase.inc.php");
require_once ("include/clsCadastro.inc.php");
require_once ("include/clsBanco.inc.php");
require_once ("include/pmieducar/geral.inc.php" );
require_once ("include/alimentacao/geral.inc.php" );

```

```

class clsIndexBase extends clsBase
{
    function Formular()
    {
        $this->SetTitulo( "{$this->_instituicao} i-Educar - IMC " );
    }
}

```

```

        $this->processoAp = "10006";
    }
}

class indice extends clsCadastro
{
    /**
     * Referencia pega da session para o idpes do usuario atual
     *
     * @var int
     */
    var $pessoa_logada;

    var $idimc;
    var $ref_aluno;
    var $ref_serie;
    var $ref_escola;
    var $altura;
    var $peso;
    var $imc;
    var $dt_cadastro;
    var $observacao;

    function Inicializar()
    {
        $retorno = "Novo";
        @session_start();
        $this->pessoa_logada = $_SESSION['id_pessoa'];
        @session_write_close();

        $this->idimc=$_GET["idimc"];
        $this->ref_aluno=$_GET["ref_aluno"];
    }
}

```

```

        $obj_permissoes = new clsPermissoes();

        $obj_permissoes->permissao_cadastra( 10006, $this-> pessoa_logada,3,
"alimentacao_imc_lst_lst.php" );

        if( is_numeric( $this->idimc ) )
        {

            $obj_imc = new clsAlimentacaoIMC();
            $lst = $obj_imc->lista($this->idimc);

            if (is_array($lst))
            {

                $registro = array_shift($lst);
                if( $registro )
                {

                    foreach( $registro AS $campo => $val ) // passa todos os
valores obtidos no registro para atributos do objeto
                        $this->$campo = $val;

                    /** verificao de permissao para exclusao
                        $this->fexcluir = $obj_permissoes-
>permissao_excluir(10006,$this->pessoa_logada,3);
                        /**

                    $retorno = "Editar";
                }else{
                    header( "Location: alimentacao_imc_lst_lst.php" );
                    die();
                }
            }
        }

        $this->url_cancelar = ($retorno == "Editar") ? "alimentacao_imc_det.php?idimc={$this-
>idimc}&ref_aluno={$this->ref_aluno}" : "alimentacao_cardapio_lst_lst.php?ref_aluno={$this-
>ref_aluno}";

        $this->nome_url_cancelar = "Cancelar";

```

```

        return $retorno;
    }

function Gerar()
{
    // primary keys
    $this->campoOculto( "idimc", $this->idimc );

    $obj_imc = new clsAlimentacaoIMC();

    $lista = $obj_imc->listaAluno($this->ref_aluno);
    if(is_array($lista))
    {
        $nm_aluno = $lista[0]["nome"];
    }
    $this->campoOculto( "ref_aluno", $this->ref_aluno );
    $this->campoRotulo("nm_aluno","Aluno","{$nm_aluno}");

    $opcoes = array();
    $obj_escola = new clsPmieducarEscola();
    $lista = $obj_escola->lista();

    if( is_array( $lista ) && count( $lista ) )
    {
        foreach ( $lista AS $registro )
        {
            $opcoes[$registro["cod_escola"]] = $registro["nome"];
        }
    }

    $this->campoLista( "ref_escola", "Escola", $opcoes, $this->ref_escola, "", false, "", "", "", true );
}

```

```

    $opcoes = array();
    $obj_serie = new clsPmieducarSerie();
    $lista = $obj_serie->lista();

    if( is_array( $lista ) && count( $lista ) )
    {
        foreach ( $lista AS $registro )
        {
            $opcoes[$registro["cod_serie"]] = $registro["nm_serie"];
        }
    }

    $this->campoLista( "ref_serie", "Série", $opcoes, $this->ref_serie,"",false,"", "",true );

    $this->campoMonetario( "altura", "Altura(Mt).", $this->altura, 30, 255, true );

    $this->campoMonetario( "peso", "Peso(Kg).", $this->peso, 30, 255, true );

    $this->campoMemo( "observacao", "Observações", $this->observacao, 70, 3, false );

}

function Novo()
{
    @session_start();
    $this->pessoa_logada = $_SESSION["id_pessoa"];
    @session_write_close();

    $caminho = "arquivos/cardapios/";

    $this->peso = str_replace(",",".",$this->peso);
    $this->altura = str_replace(",",".",$this->altura);

```

```

$obj = new clsAlimentacaoIMC();
$obj->dt_cadastro = "NOW()";
$obj->ref_aluno = $this->ref_aluno;
$obj->ref_escola = $this->ref_escola;
$obj->ref_serie = $this->ref_serie;
$obj->altura = $this->altura;
$obj->peso = $this->peso;
$obj->observacao = $this->observacao;
$obj->imc = number_format(($this->peso / ($this->altura * $this->altura)),2,".", "");
$cadastrou = $obj->cadastra();
if( $cadastrou )
{
    $this->mensagem .= "Cadastro efetuado com sucesso.<br>";
    header( "Location: alimentacao_imc_lst_lst.php?ref_aluno={$this->ref_aluno}"
);
    die();
    return true;
}

$this->mensagem = "Cadastro não realizado.<br>";
echo "<!--\nErro ao cadastrar clsAlimentacaoIMC\n-->";
return false;
}

function Editar()
{
    @session_start();
    $this-> pessoa_logada = $_SESSION['id_pessoa'];
    @session_write_close();

    $obj = new clsAlimentacaoIMC();
    $obj->idimc = $this->idimc;
    $this->peso = str_replace(",",".", $this->peso);

```

```

$this->altura = str_replace(",",".", $this->altura);
$obj->ref_escola = $this->ref_escola;
$obj->ref_serie = $this->ref_serie;
$obj->altura = $this->altura;
$obj->peso = $this->peso;
$obj->observacao = $this->observacao;
$obj->imc = number_format(($this->peso / ($this->altura * $this->altura)),2,".", "");
$cadastrou = $obj->edita();
if( $cadastrou )
{
    $this->mensagem .= "Cadastro editado com sucesso.<br>";
    header( "Location: alimentacao_imc_det.php?idimc={$this->idimc}&ref_aluno={$ref_aluno}" );
    die();
    return true;
}

$this->mensagem = "Cadastro não editado.<br>";
echo "<!--\nErro ao cadastrar clsAlimentacaoIMC\n-->";
return false;

}

function Excluir()
{
    @session_start();
    $this->peessoa_logada = $_SESSION['id_pessoa'];
    @session_write_close();

    $obj = new clsAlimentacaoIMC();
    $obj->idimc = $this->idimc;
    $obj->exclui();
}

```





```

class indice extends clsDetalhe
{
    /**
     * Titulo no topo da pagina
     *
     * @var int
     */
    var $titulo;

    var $idimc;
    var $ref_aluno;

    function Gerar()
    {
        @session_start();
        $this->peessoa_logada = $_SESSION['id_pessoa'];
        session_write_close();

        $this->titulo = "IMC - Detalhe";
        $this->addBanner( "imagens/nvp_top_intranet.jpg", "imagens/nvp_vert_intranet.jpg",
        "Intranet" );

        $this->idimc=$_GET["idimc"];
        $this->ref_aluno=$_GET["ref_aluno"];

        $obj_imc = new clsAlimentacaoIMC();
        $lista = $obj_imc->lista($this->idimc);
        $registro = $lista[0];

        if( ! $registro )
        {
            header( "location: alimentacao_imc_lst.php" );

```

```

        die();
    }

    $lista = $obj_imc->listaAluno($registro["ref_aluno"]);
    if(is_array($lista))
    {
        $nm_aluno = $lista[0]["nome"];
        $this->ref_aluno = $registro["ref_aluno"];
    }

    $this->addDetalhe( array( "Aluno", $nm_aluno ) );

    $obj_ref_cod_escola = new clsPmieducarEscola( $registro["ref_escola"] );
    $det_ref_cod_escola = $obj_ref_cod_escola->detalhe();
    $nm_escola = $det_ref_cod_escola["nome"];

    $this->addDetalhe( array( "Escola", $nm_escola ) );

    $obj_serie = new clsPmieducarSerie( $registro["ref_serie"] );
    $det_serie = $obj_serie->detalhe();
    $nm_serie = $det_serie["nm_serie"];

    $this->addDetalhe( array( "Série", $nm_serie ) );
    $this->addDetalhe( array( "Data Cadastro", date('d/m/Y',strtotime($registro["dt_cadastro"])) ) );

    $this->addDetalhe( array( "Altura", $registro["altura"] ) );
    $this->addDetalhe( array( "Peso", $registro["peso"] ) );
    $this->addDetalhe( array( "IMC", $registro["imc"] ) );
    $this->addDetalhe( array( "Observações", $registro["observacao"] ) );

    $obj_permissoes = new clsPermissoes();

```

```

        if( $obj_permissoes->permissao_cadastra( 10006, $this->peessoa_logada, 3 ) )
        {
            $this->url_novo = "alimentacao_imc_cad.php?ref_aluno={$this->ref_aluno}";
            $this->url_editar = "alimentacao_imc_cad.php?idimc={$this->idimc}";
        }

        $this->url_cancelar = "alimentacao_imc_lst_lst.php?ref_aluno={$this->ref_aluno}";
        $this->largura = "100%";
    }
}

```

```
// cria uma extensao da classe base
```

```
$pagina = new clsIndexBase();
```

```
// cria o conteudo
```

```
$miolo = new indice();
```

```
// adiciona o conteudo na clsBase
```

```
$pagina->addForm( $miolo );
```

```
// gera o html
```

```
$pagina->MakeAll();
```

```
?>
```

```
alimentacao_imc_lst.php
```

```
<?php
```

```
/* *****
```

```
 *
```

```
 *
```

```
 *   @author Prefeitura Municipal de Itajaí
```

```
 *
```

```
 *   @updated 29/03/2007
```

```
 *
```

```
 * Pacote: i-PLB Software Público Livre e Brasileiro
```

```
 *
```

```
 *
```

```
 *
```

```
 *   Copyright (C) 2006   PMI - Prefeitura Municipal de Itajaí
```

```
 *
```



```
        $this->processoAp = "10006";
    }
}

class indice extends clsListagem
{
    /**
     * Referencia pega da session para o idpes do usuario atual
     *
     * @var int
     */
    var $pessoa_logada;

    /**
     * Titulo no topo da pagina
     *
     * @var int
     */
    var $titulo;

    /**
     * Quantidade de registros a ser apresentada em cada pagina
     *
     * @var int
     */
    var $limite;

    /**
     * Inicio dos registros a serem exibidos (limit)
     *
     * @var int
     */
    var $offset;
```

```

var $nm_aluno;

function Gerar()
{
    @session_start();
    $this->peessoa_logada = $_SESSION['id_pessoa'];
    session_write_close();

    $this->titulo = "IMC - Listagem de alunos";

    foreach( $_GET AS $var => $val ) // passa todos os valores obtidos no GET para
atributos do objeto
        $this->$var = ( $val === "" ) ? null: $val;

    $this->addBanner( "imagens/nvp_top_intranet.jpg", "imagens/nvp_vert_intranet.jpg",
    "Intranet" );

    $obj_permissao = new clsPermissoes();

    $lista_busca = array(
        "Aluno",
        "Nome da mãe",
        "Data de Nascimento",
        "Sexo"
    );

    $this->addCabecalhos($lista_busca);

    $this->campoTexto( "nm_aluno", "Nome do aluno", $this->nm_aluno, 30, 255, false );

    $obj_cardapio = new clsAlimentacaoIMC();
    $filtro_aluno = null;
    if($this->nm_aluno != "")

```

```

{
    $filtro_aluno = $this->nm_aluno;
}

if($filtro_aluno!="")
{
    $lista = $obj_cardapio->listaAluno(null,$filtro_aluno);
    if( is_array( $lista ) && count( $lista ) )
    {
        foreach ( $lista AS $registro )
        {
            $lista_busca = array();
            $lista_busca[] = "<a href=\"alimentacao_imc_lst_lst.php?
ref_aluno={$registro[\"cod_aluno\"]}\">{$registro[\"nome\"]}</a>";
            $lista_busca[] = "<a href=\"alimentacao_imc_lst_lst.php?
ref_aluno={$registro[\"cod_aluno\"]}\">{$registro[\"nm_mae\"]}</a>";
            $lista_busca[] = "<a href=\"alimentacao_imc_lst_lst.php?
ref_aluno={$registro[\"cod_aluno\"]}\">.date('d/m/Y',strtotime($registro[\"data_nasc\"])).\"</a>";
            $lista_busca[] = "<a href=\"alimentacao_imc_lst_lst.php?
ref_aluno={$registro[\"cod_aluno\"]}\">{$registro[\"sexo\"]}</a>";
            $this->addLinhas($lista_busca);
        }
    }
}

/*
if( $obj_permissao->permissao_cadastra( 10006, $this->pessoa_logada, 3 ) )
{
    $this->acao = "go(\"alimentacao_imc_cad.php\")";
    $this->nome_acao = "Novo";
}*/

$this->largura = "100%";
}
}

```

```

// cria uma extensao da classe base
$pagina = new clsIndexBase();

// cria o conteudo
$miolo = new indice();

// adiciona o conteudo na clsBase
$pagina->addForm( $miolo );

// gera o html
$pagina->MakeAll();

?>

```

alimentacao\_nutricionista\_escola\_cad.php

```
<?php
```

```
/* *****
```

```
*

```

```
*

```

```
* @author Prefeitura Municipal de Itaja?

```

```
*

```

```
* @updated 29/03/2007

```

```
*

```

```
* Pacote: i-PLB Software P?blico Livre e Brasileiro

```

```
*

```

```
*

```

```
*

```

```
* Copyright (C) 2006 PMI - Prefeitura Municipal de Itaja?

```

```
*

```

```
*

```

```
ctima@itajai.sc.gov.br

```

```
*

```

```
*

```

```
*

```

```
* Este programa ? software livre, voc? pode redistribu?-lo e/ou

```

```
* modific?-lo sob os termos da Licen?a P?blica Geral GNU, conforme

```

```
*

```

```
* publicada pela Free Software Foundation, tanto a vers?o 2 da

```

```
*

```

```
* Licen?a como (a seu crit?rio) qualquer vers?o mais nova.

```

```
*

```

```
*

```

```
* Este programa ? distribu?do na expectativa de ser ?til, mas SEM

```

```
*

```

```
* QUALQUER GARANTIA. Sem mesmo a garantia impl?cita de COMERCIALI-

```

```
*

```



```

var $pessoa_logada;

var $ref_escola;
var $ref_usuario;
var $dt_cadastro;

function Inicializar()
{
    $retorno = "Novo";
    @session_start();
    $this->pessoa_logada = $_SESSION['id_pessoa'];
    @session_write_close();

    $this->ref_usuario=$_GET["ref_usuario"];
    $this->ref_escola=$_GET["ref_escola"];

    $obj_permissoes = new clsPermissoes();
    $obj_permissoes->permissao_cadastra( 10001, $this->pessoa_logada,3,
"alimentacao_nutricionista_escola_lst.php" );

    if( is_numeric( $this->ref_usuario ) && is_numeric( $this->ref_escola ) )
    {
        $obj_nutricionista_escola = new clsAlimentacaoNutricionistaEscola();
        $lst = $obj_nutricionista_escola->lista($this->ref_usuario,$this->ref_escola);

        if (is_array($lst))
        {
            $registro = array_shift($lst);
            if( $registro )
            {
                foreach( $registro AS $campo => $val ) // passa todos os
valores obtidos no registro para atributos do objeto

```

```

        $this->$campo = $val;

        /** verificao de permissao para exclusao
        $this->fexcluir = $obj_permissoes-
>permissao_excluir(10001,$this->peessoa_logada,3);
        /**

        $retorno = "Editar";

        }else{
            header( "Location:
alimentacao_nutricionista_escola_lst.php" );
            die();
        }
    }
}

    $this->url_cancelar = ($retorno == "Editar") ?
"alimentacao_nutricionista_escola_det.php?
ref_escola={$registro["ref_escola"]}&ref_usuario={$registro["ref_usuario"]}":
"alimentacao_nutricionista_escola_lst.php";

    $this->nome_url_cancelar = "Cancelar";

    return $retorno;
}

function Gerar()
{
    $opcoes = array();
    $obj_escola = new clsPmieducarEscola();
    $lista = $obj_escola->lista();

    if( is_array( $lista ) && count( $lista ) )
    {
        foreach ( $lista AS $registro )
        {
            $opcoes[$registro["cod_escola"]] = $registro["nome"];
        }
    }
}

```

```

    }

    $this->campoLista( "ref_escola", "Escola", $opcoes, $this-
>ref_escola,"",false,"", "", "", true );

    $opcoes = array();
    $obj_usuario = new clsPmieducarUsuario();
    $lista = $obj_usuario->lista(null,null,null,null,null,7);

    if( is_array( $lista ) && count( $lista ) )
    {
        foreach ( $lista AS $registro )
        {
            $obj_pessoa = new clsPessoa_($registro["cod_usuario"]);
            $det_pessoa = $obj_pessoa->detalhe();
            $nm_pessoa = $det_pessoa["nome"];
            $opcoes[$registro["cod_usuario"]] = $nm_pessoa;
        }
    }

    $this->campoLista( "ref_usuario", "Nutricionista", $opcoes, $this-
>ref_usuario,"",false,"", "", "", true );

}

function Novo()
{
    @session_start();
    $this->pessoa_logada = $_SESSION["id_pessoa"];
    @session_write_close();

    if(false)
    {

```

```

        $this->mensagem = "Cadastro não realizado.<br>Dados já cadastrados.";
        echo "<!--\nErro ao cadastrar clsAlimentacaoNutricionistaEscola\Dados já
cadastrados-->";

        return false;
    }

    $obj = new clsAlimentacaoNutricionistaEscola();
    $obj->dt_cadastro = "NOW()";
    $obj->ref_usuario= $this->ref_usuario;
    $obj->ref_escola = $this->ref_escola;
    $cadastrou = $obj->cadastra();
    if( $cadastrou )
    {
        $this->mensagem .= "Cadastro efetuado com sucesso.<br>";
        header( "Location: alimentacao_nutricionista_escola_lst.php" );
        die();
        return true;
    }

    $this->mensagem = "Cadastro não realizado.<br>";
    echo "<!--\nErro ao cadastrar clsAlimentacaoNutricionistaEscola\n-->";
    return false;
}

function Editar()
{
    @session_start();
    $this->pessoa_logada = $_SESSION['id_pessoa'];
    @session_write_close();

    $this->mensagem = "Cadastro não editado.<br>Este cadastro não pode ser editado.";
    echo "<!--\nErro ao cadastrar clsAlimentacaoNutricionistaEscola\n-->";
    return false;
}

```

```

    }

function Excluir()
{
    @session_start();
    $this->peessoa_logada = $_SESSION['id_pessoa'];
    @session_write_close();

    $obj = new clsAlimentacaoNutricionistaEscola();
    $obj->ref_usuario= $this->ref_usuario;
    $obj->ref_escola = $this->ref_escola;
    $cadastrou = $obj->exclui();

    $this->mensagem = "Cadastro excluído com sucesso.<br>";
    header( "Location: alimentacao_nutricionista_escola_lst.php" );
    die();
    return true;
}
}

// cria uma extensao da classe base
$pagina = new clsIndexBase();
// cria o conteudo
$miolo = new indice();
// adiciona o conteudo na clsBase
$pagina->addForm( $miolo );
// gera o html
$pagina->MakeAll();
?>
alimentacao_nutricionista_escola_det.php
<?php

```

/\* \*\*\*\* \*/

\*

\*

\* @author Prefeitura Municipal de Itajaí

\* @updated 29/03/2007

\* Pacote: i-PLB Software Público Livre e Brasileiro

\*

\*

\* Copyright (C) 2006 PMI - Prefeitura Municipal de Itajaí

\*

ctima@itajai.sc.gov.br

\*

\*

\* Este programa é software livre, você pode redistribuí-lo e/ou \*  
\* modificá-lo sob os termos da Licença Pública Geral GNU, conforme \*  
\* publicada pela Free Software Foundation, tanto a versão 2 da \*  
\* Licença como (a seu critério) qualquer versão mais nova. \*

\*

\*

\* Este programa é distribuído na expectativa de ser útil, mas SEM \*  
\* QUALQUER GARANTIA. Sem mesmo a garantia implícita de COMERCIALI- \*  
\* ZAÇÃO ou de ADEQUAÇÃO A QUALQUER PROPÓSITO EM PARTICULAR. Con- \*  
\* sulte a Licença Pública Geral GNU para obter mais detalhes. \*

\*

\*

\* Você deve ter recebido uma cópia da Licença Pública Geral GNU \*  
\* junto com este programa. Se não, escreva para a Free Software \*  
\* Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA \*  
\* 02111-1307, USA. \*

\*

\*

\*\*\*\* \*/

```
require_once ("include/clsBase.inc.php");
```

```
require_once ("include/clsDetalhe.inc.php");
```

```
require_once ("include/clsBanco.inc.php");
require_once ("include/pmieducar/geral.inc.php" );
require_once ("include/alimentacao/geral.inc.php" );

class clsIndexBase extends clsBase
{
    function Formular()
    {
        $this->SetTitulo( "{$this->_instituicao} i-Educar - Nutricionista Escola" );
        $this->processoAp = "10001";
    }
}

class indice extends clsDetalhe
{
    /**
     * Titulo no topo da pagina
     *
     * @var int
     */
    var $titulo;

    var $ref_usuario;
    var $ref_escola;

    function Gerar()
    {
        @session_start();
        $this->pessoa_logada = $_SESSION['id_pessoa'];
        session_write_close();

        $this->titulo = "Nutricionista Escola - Detalhe";
    }
}
```

```
"Intranet" );
$this->addBanner( "imagens/nvp_top_intranet.jpg", "imagens/nvp_vert_intranet.jpg",
```

```
$this->ref_usuario=$_GET["ref_usuario"];
$this->ref_escola=$_GET["ref_escola"];

```

```
$obj_nutricionista_escola = new clsAlimentacaoNutricionistaEscola();
$lista = $obj_nutricionista_escola->lista($this->ref_usuario,$this->ref_escola);
$registro = $lista[0];

```

```
if( ! $registro )
{
    header( "location: alimentacao_nutricionista_escola_lst.php" );
    die();
}

```

```
$obj_ref_cod_escola = new clsPmieducarEscola( $registro["ref_escola"] );
$det_ref_cod_escola = $obj_ref_cod_escola->detalhe();
$nm_escola = $det_ref_cod_escola["nome"];

```

```
$obj_pessoa = new clsPessoa_( $registro["ref_usuario"] );
$det_pessoa = $obj_pessoa->detalhe();
$nm_pessoa = $det_pessoa["nome"];

```

```
$this->addDetalhe( array( "Nutricionista", $nm_pessoa ) );
$this->addDetalhe( array( "Escola", $nm_escola ) );

```

```
$this->addDetalhe( array( "Data Cadastro", date('d/m/Y',strtotime($registro["dt_cadastro"])) ) );

```

```
$obj_permissoes = new clsPermissoes();
if( $obj_permissoes->permissao_cadastra( 10001, $this->pessoa_logada, 3 ) )

```

```

        {
            $this->url_novo = "alimentacao_nutricionista_escola_cad.php";
            $this->url_editar = "alimentacao_nutricionista_escola_cad.php?
ref_usuario={$this->ref_usuario}&ref_escola={$this->ref_escola}";
        }

        $this->url_cancelar = "alimentacao_nutricionista_escola_lst.php";
        $this->largura = "100%";
    }
}

```

```
// cria uma extensao da classe base
```

```
$pagina = new clsIndexBase();
```

```
// cria o conteudo
```

```
$miolo = new indice();
```

```
// adiciona o conteudo na clsBase
```

```
$pagina->addForm( $miolo );
```

```
// gera o html
```

```
$pagina->MakeAll();
```

```
?>
```

```
alimentacao_nutricionista_escola_lst.php
```

```
<?php
```

```
/* *****
```

```
*
```

```
*
```

```
* @author Prefeitura Municipal de Itajaí
```

```
*
```

```
* @updated 29/03/2007
```

```
*
```

```
* Pacote: i-PLB Software Público Livre e Brasileiro
```

```
*
```

```
*
```

```
*
```

```
* Copyright (C) 2006 PMI - Prefeitura Municipal de Itajaí
```

```
*
```

```

*
*                               ctima@itajai.sc.gov.br
*
*
*
*                               *
*
* Este programa é software livre, você pode redistribuí-lo e/ou *
* modificá-lo sob os termos da Licença Pública Geral GNU, conforme *
* publicada pela Free Software Foundation, tanto a versão 2 da *
* Licença como (a seu critério) qualquer versão mais nova. *
*
*                               *
*
* Este programa é distribuído na expectativa de ser útil, mas SEM *
* QUALQUER GARANTIA. Sem mesmo a garantia implícita de COMERCIALI- *
* ZAÇÃO ou de ADEQUAÇÃO A QUALQUER PROPÓSITO EM PARTICULAR. Con- *
* sulte a Licença Pública Geral GNU para obter mais detalhes. *
*
*                               *
*
* Você deve ter recebido uma cópia da Licença Pública Geral GNU *
* junto com este programa. Se não, escreva para a Free Software *
* Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA *
* 02111-1307, USA. *
*
*                               *
*
* *****/
require_once ("include/clsBase.inc.php");
require_once ("include/clsListagem.inc.php");
require_once ("include/clsBanco.inc.php");
require_once ("include/pmieducar/geral.inc.php" );
require_once ("include/alimentacao/geral.inc.php" );

class clsIndexBase extends clsBase
{
    function Formular()
    {
        $this->SetTitulo( "{$this->_instituicao} i-Educar - Nutricionista Escola " );
    }
}

```

```
        $this->processoAp = "10001";
    }
}

class indice extends clsListagem
{
    /**
     * Referencia pega da session para o idpes do usuario atual
     *
     * @var int
     */
    var $pessoa_logada;

    /**
     * Titulo no topo da pagina
     *
     * @var int
     */
    var $titulo;

    /**
     * Quantidade de registros a ser apresentada em cada pagina
     *
     * @var int
     */
    var $limite;

    /**
     * Inicio dos registros a serem exibidos (limit)
     *
     * @var int
     */
    var $offset;
```

```

var $ref_escola;
var $ref_usuario;

function Gerar()
{
    @session_start();
    $this->peessoa_logada = $_SESSION['id_pessoa'];
    session_write_close();

    $this->titulo = "Nutricionista Escola - Listagem";

    foreach( $_GET AS $var => $val ) // passa todos os valores obtidos no GET para
atributos do objeto
        $this->$var = ( $val === "" ) ? null: $val;

    $this->addBanner( "imagens/nvp_top_intranet.jpg", "imagens/nvp_vert_intranet.jpg",
    "Intranet" );

    $obj_permissao = new clsPermissoes();

    $lista_busca = array(
        "Nutricionista",
        "Escola",
        "Data Cadastro"
    );

    $this->addCabecalhos($lista_busca);

    $opcoes = array();
    $obj_escola = new clsPmieducarEscola();
    $lista = $obj_escola->lista();

    $opcoes["0"] = "Todas";

```

```

if( is_array( $lista ) && count( $lista ) )
{
    foreach ( $lista AS $registro )
    {
        $opcoes[$registro["cod_escola"]] = $registro["nome"];
    }
}

$this->campoLista( "ref_escola", "Escola", $opcoes, $this-
>ref_escola, "", false, "", "", "", false );

$opcoes = array();
$obj_usuario = new clsPmieducarUsuario();
$lista = $obj_usuario->lista(null,null,null,null,null,7);

$opcoes["0"] = "Todos";
if( is_array( $lista ) && count( $lista ) )
{
    foreach ( $lista AS $registro )
    {
        $obj_pessoa = new clsPessoa_( $registro["cod_usuario"] );
        $det_pessoa = $obj_pessoa->detalhe();
        $nm_pessoa = $det_pessoa["nome"];
        $opcoes[$registro["cod_usuario"]] = $nm_pessoa;
    }
}

$this->campoLista( "ref_usuario", "Nutricionista", $opcoes, $this-
>ref_usuario, "", false, "", "", "", false );

```

```

$obj_nutricionista_escola = new clsAlimentacaoNutricionistaEscola();
$filtro_escola = null;
if($this->ref_escola > 0)
{
    $filtro_escola = $this->ref_escola;
}
$filtro_usuario = null;
if($this->ref_usuario > 0)
{
    $filtro_usuario = $this->ref_usuario;
}
$lista = $obj_nutricionista_escola->lista($filtro_usuario,$filtro_escola);

if( is_array( $lista ) && count( $lista ) )
{
    foreach ( $lista AS $registro )
    {
        $obj_ref_cod_escola = new
clsPmieducarEscola( $registro["ref_escola"] );

        $det_ref_cod_escola = $obj_ref_cod_escola->detalhe();

        $nm_escola = $det_ref_cod_escola["nome"];

        $obj_pessoa = new clsPessoa_( $registro["ref_usuario"] );
        $det_pessoa = $obj_pessoa->detalhe();

        $nm_pessoa = $det_pessoa["nome"];

        $lista_busca = array();

        $lista_busca[] = "<a href=\"alimentacao_nutricionista_escola_det.php?
ref_usuario={$registro["ref_usuario"]}&ref_escola={$registro["ref_escola"]}\">{$nm_pessoa}</a>";

        $lista_busca[] = "<a href=\"alimentacao_nutricionista_escola_det.php?
ref_usuario={$registro["ref_usuario"]}&ref_escola={$registro["ref_escola"]}\">{$nm_escola}</a>";

        $lista_busca[] = "<a href=\"alimentacao_nutricionista_escola_det.php?
ref_usuario={$registro["ref_usuario"]}&ref_escola={$registro["ref_escola"]}\">.date('d/m/Y',strtotime($r
registro["dt_cadastro"])).\"</a>";

```

```

        $this->addLinhas($lista_busca);
    }

}

if( $obj_permissao->permissao_cadastra( 10001, $this->peessoa_logada, 3 ) )
{
    $this->acao = "go(\"alimentacao_nutricionista_escola_cad.php\")";
    $this->nome_acao = "Novo";
}

$this->largura = "100%";
}
}

// cria uma extensao da classe base
$pagina = new clsIndexBase();

// cria o conteudo
$miolo = new indice();

// adiciona o conteudo na clsBase
$pagina->addForm( $miolo );

// gera o html
$pagina->MakeAll();

?>
alimentacao_produto_cad.php

<?php
/* *****
*
*
*   @author Prefeitura Municipal de Itaja?
*
*   @updated 29/03/2007
*
* Pacote: i-PLB Software P?blico Livre e Brasileiro
*

```



```

function Formular()
{
    $this->SetTitulo( "{$this->_instituicao} i-Educar - Produto " );
    $this->processoAp = "10002";
}
}

```

```

class indice extends clsCadastro

```

```

{
    /**
     * Referencia pega da session para o idpes do usuario atual
     *
     * @var int
     */
    var $pessoa_logada;

    var $idpro;
    var $nm_produto;
    var $fator_correcao;
    var $fator_coccao;
    var $ref_produto_grupo;
    var $ref_produto_unidade;
    var $calorias;
    var $proteinas;

    function Inicializar()
    {
        $retorno = "Novo";
        @session_start();
        $this->pessoa_logada = $_SESSION['id_pessoa'];
        @session_write_close();
    }
}

```

```

$this->idpro=$_GET["idpro"];

$obj_permissoes = new clsPermissoes();

$obj_permissoes->permissao_cadastra( 10002, $this->peessoa_logada,3,
"alimentacao_produto_lst.php" );

if( is_numeric( $this->idpro ) )
{
    $obj_produto = new clsAlimentacaoProduto();
    $lst = $obj_produto->lista($this->idpro);

    if( is_array($lst) )
    {
        $registro = array_shift($lst);
        if( $registro )
        {
            foreach( $registro AS $campo => $val ) // passa todos os
valores obtidos no registro para atributos do objeto
                $this->$campo = $val;

            /** verificao de permissao para exclusao
            //$this->fexcluir = $obj_permissoes-
->permissao_excluir(572,$this->peessoa_logada,7);
            /**

            $retorno = "Editar";
        }else{
            header( "Location: alimentacao_produto_lst.php" );
            die();
        }
    }
}

$this->url_cancelar = ($retorno == "Editar") ? "alimentacao_produto_det.php?
idpro={$registro["idpro"]}" : "alimentacao_produto_lst.php";

$this->nome_url_cancelar = "Cancelar";

```

```

        return $retorno;
    }

    function Gerar()
    {
        // primary keys
        $this->campoOculto( "idpro", $this->idpro );

        // text
        $this->campoTexto( "nm_produto", "Produto", $this->nm_produto, 30, 255, true );

        $this->campoNumero( "fator_correcao", "Fator Correção", $this->fator_correcao, 30,
255, true );

        $this->campoNumero( "fator_coccao", "Fator Cocção", $this->fator_coccao, 30, 255,
true );

        $opcoes = array();
        $obj_produto_grupo = new clsAlimentacaoProdutoGrupo();
        $lista = $obj_produto_grupo->lista();

        if( is_array( $lista ) && count( $lista ) )
        {
            foreach ( $lista AS $registro )
            {
                $opcoes[$registro["idpg"]] = $registro["descricao"];
            }
        }

        $this->campoLista( "ref_produto_grupo", "Grupo", $opcoes, $this-
>ref_produto_grupo,"",false,"", "", "", true );

        $opcoes = array();
        $obj_produto_unidade = new clsAlimentacaoProdutoUnidade();

```

```

$lista = $obj_produto_unidade->lista();

if( is_array( $lista ) && count( $lista ) )
{
    foreach ( $lista AS $registro )
    {
        $opcoes[$registro["idpu"]] = $registro["descricao"];
    }
}

$this->campoLista( "ref_produto_unidade", "Unidade de medida", $opcoes, $this-
>ref_produto_unidade,"",false,"", "", "", true );

$this->campoNumero( "calorias", "Calorias(Kcal)", $this->calorias, 30, 255, true,
"Quantidade há cada 100 gramas ou ml" );

$this->campoNumero( "proteinas", "Proteínas(gramas)", $this->proteinas, 30, 255,
true, "Quantidade há cada 100 gramas ou ml" );

}

function Novo()
{
    @session_start();
    $this->pessoa_logada = $_SESSION['id_pessoa'];
    @session_write_close();

    $obj = new clsAlimentacaoProduto();
    $obj->ref_produto_grupo = $this->ref_produto_grupo;
    $obj->ref_produto_unidade = $this->ref_produto_unidade;
    $obj->nm_produto = $this->nm_produto;
    $obj->fator_correcao = $this->fator_correcao;

```

```

$obj->fator_coccao = $this->fator_coccao;
$obj->calorias = $this->calorias;
$obj->proteinas = $this->proteinas;
$cadastrou = $obj->cadastra();
if( $cadastrou )
{
    $this->mensagem .= "Cadastro efetuado com sucesso.<br>";
    header( "Location: alimentacao_produto_lst.php" );
    die();
    return true;
}

$this->mensagem = "Cadastro não realizado.<br>";
echo "<!--\nErro ao cadastrar clsAlimentacaoProduto\n-->";
return false;
}

function Editar()
{
    @session_start();
    $this->peessoa_logada = $_SESSION['id_pessoa'];
    @session_write_close();

    $obj = new clsAlimentacaoProduto();
    $obj->idpro = $this->idpro;
    $obj->ref_produto_grupo = $this->ref_produto_grupo;
    $obj->ref_produto_unidade = $this->ref_produto_unidade;
    $obj->nm_produto = $this->nm_produto;
    $obj->fator_correcao = $this->fator_correcao;
    $obj->fator_coccao = $this->fator_coccao;
    $obj->calorias = $this->calorias;
    $obj->proteinas = $this->proteinas;
    $cadastrou = $obj->edita();
}

```

```

        if( $cadastrou )
        {
            $this->mensagem .= "Cadastro editado com sucesso.<br>";
            header( "Location: alimentacao_produto_det.php?idpro={$this->idpro}" );
            die();
            return true;
        }

        $this->mensagem = "Cadastro não editado.<br>";
        echo "<!--\nErro ao cadastrar clsAlimentacaoProduto\n-->";
        return false;
    }

    function Excluir()
    {
        @session_start();
        $this->pessoa_logada = $_SESSION['id_pessoa'];
        @session_write_close();

        return false;
    }
}

// cria uma extensao da classe base
$pagina = new clsIndexBase();
// cria o conteudo
$miolo = new indice();
// adiciona o conteudo na clsBase
$pagina->addForm( $miolo );
// gera o html
$pagina->MakeAll();

```



```

*      02111-1307, USA.
*
*
*
*****/

require_once ("include/clsBase.inc.php");
require_once ("include/clsDetalhe.inc.php");
require_once ("include/clsBanco.inc.php");
require_once ("include/pmieducar/geral.inc.php" );
require_once ("include/alimentacao/geral.inc.php" );

class clsIndexBase extends clsBase
{
    function Formular()
    {
        $this->SetTitulo( "{$this->_instituicao} i-Educar - Produto" );
        $this->processoAp = "10002";
    }
}

class indice extends clsDetalhe
{
    /**
     * Titulo no topo da pagina
     *
     * @var int
     */
    var $titulo;

    var $idpro;

    function Gerar()
    {
        @session_start();
    }
}

```

```

$this->pessoa_logada = $_SESSION['id_pessoa'];
session_write_close();

$this->titulo = "Produto - Detalhe";
$this->addBanner( "imagens/nvp_top_intranet.jpg", "imagens/nvp_vert_intranet.jpg",
    "Intranet" );

$this->idpro=$_GET["idpro"];

$obj_produto = new clsAlimentacaoProduto();
$lista = $obj_produto->lista($this->idpro);
$registro = $lista[0];

if( ! $registro )
{
    header( "location: alimentacao_produto_1st.php" );
    die();
}

$this->addDetalhe( array( "Produto", $registro["nm_produto"] ) );

$this->addDetalhe( array( "Fator Correção", $registro["fator_correcao"] ) );

$this->addDetalhe( array( "Fator Cocção", $registro["fator_coccao"] ) );

$obj_ref_grupo = new clsAlimentacaoProdutoGrupo();
$det_ref_grupo = $obj_ref_grupo->lista($registro["ref_produto_grupo"]);
$nm_grupo = $det_ref_grupo[0]["descricao"];

$this->addDetalhe( array( "Grupo", $nm_grupo ) );

$obj_ref_unidade = new clsAlimentacaoProdutoUnidade();

```

```

$det_ref_unidade = $obj_ref_unidade->lista($registro["ref_produto_unidade"]);
$nm_unidade = $det_ref_unidade[0]["descricao"];

$this->addDetalhe( array( "Unidade de medida", $nm_unidade) );

$this->addDetalhe( array( "Calorias(Kcal)/100g ou ml", $registro["calorias"]) );

$this->addDetalhe( array( "Proteínas(gramas)/100g ou ml", $registro["proteinas"]) );

$obj_permissoes = new clsPermissoes();
if( $obj_permissoes->permissao_cadastra( 10002, $this-> pessoa_logada, 3 ) )
{
    $this->url_novo = "alimentacao_produto_cad.php";
    $this->url_editar = "alimentacao_produto_cad.php?idpro={$this->idpro}";

}

$this->url_cancelar = "alimentacao_produto_lst.php";
$this->largura = "100%";
}
}

// cria uma extensao da classe base
$pagina = new clsIndexBase();
// cria o conteudo
$miolo = new indice();
// adiciona o conteudo na clsBase
$pagina->addForm( $miolo );
// gera o html
$pagina->MakeAll();
?>
alimentacao_produto_fornecedor_cad.php

```

<?php

/\* \*\*\*\* \*/

```

*
*
*   @author Prefeitura Municipal de Itaja?
*
*   @updated 29/03/2007
*
* Pacote: i-PLB Software P?blico Livre e Brasileiro
*
*
*   Copyright (C) 2006   PMI - Prefeitura Municipal de Itaja?
*
*                               ctima@itajai.sc.gov.br
*
*
*
* Este programa ? software livre, voc? pode redistribu?-lo e/ou
* modific?-lo sob os termos da Licen?a P?blica Geral GNU, conforme
* publicada pela Free Software Foundation, tanto a vers?o 2 da
* Licen?a como (a seu crit?rio) qualquer vers?o mais nova.
*
*
* Este programa ? distribu?do na expectativa de ser ?til, mas SEM
* QUALQUER GARANTIA. Sem mesmo a garantia impl?cita de COMERCIALI-
* ZA??O ou de ADEQUA??O A QUALQUER PROP?SITO EM PARTICULAR. Con-
*
* sulte a Licen?a P?blica Geral GNU para obter mais detalhes.
*
*
*
* Voc? deve ter recebido uma c?pia da Licen?a P?blica Geral GNU
* junto com este programa. Se n?o, escreva para a Free Software
* Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA
* 02111-1307, USA.
*
*
*

```

\*\*\*\* \*/

require\_once ("include/clsBase.inc.php");

```

require_once ("include/clsCadastro.inc.php");
require_once ("include/clsBanco.inc.php");
require_once( "include/pmieducar/geral.inc.php" );
require_once( "include/alimentacao/geral.inc.php" );

```

```

class clsIndexBase extends clsBase

```

```

{
    function Formular()
    {
        $this->SetTitulo( "{$this->_instituicao} i-Educar - Produto Fornecedor " );
        $this->processoAp = "10004";
    }
}

```

```

class indice extends clsCadastro

```

```

{
    /**
     * Referencia pega da session para o idpes do usuario atual
     *
     * @var int
     */
    var $pessoa_logada;

    var $idpf;
    var $ref_produto;
    var $ref_fornecedor;
    var $ano;
    var $mes_inicio;
    var $mes_fim;
    var $agri_familiar;
    var $pesoouvolume_un;
    var $preco_un;

```

```

function Inicializar()
{
    $retorno = "Novo";
    @session_start();
    $this-> pessoa_logada = $_SESSION['id_pessoa'];
    @session_write_close();

    $this->idpf=$_GET["idpf"];

    $obj_permissoes = new clsPermissoes();
    $obj_permissoes->permissao_cadastra( 10004, $this->pessoa_logada,3,
"alimentacao_produto_fornecedor_lst.php" );

    if( is_numeric( $this->idpf ) )
    {
        $obj_produto_fornecedor = new clsAlimentacaoProdutoFornecedor();
        $lst = $obj_produto_fornecedor->lista($this->idpf);

        if (is_array($lst))
        {
            $registro = array_shift($lst);
            if( $registro )
            {
                foreach( $registro AS $campo => $val ) // passa todos os
valores obtidos no registro para atributos do objeto
                {
                    $this->$campo = $val;

                    /** verificao de permissao para exclusao
                    $this->fexcluir = $obj_permissoes-
>permissao_excluir(10004,$this->pessoa_logada,3);
                    /**

                    $retorno = "Editar";

```

```

        }else{
            header( "Location:
alimentacao_produto_fornecedor_lst.php" );
            die();
        }
    }
}

$this->url_cancelar = ($retorno == "Editar" ?
"alimentacao_produto_fornecedor_det.php?idpf={$registro["idpf"]}" :
"alimentacao_produto_fornecedor_lst.php");

$this->nome_url_cancelar = "Cancelar";
return $retorno;
}

function Gerar()
{
    // primary keys
    $this->campoOculto( "idpf", $this->idpf );

    $opcoes = array();
    $obj_produto = new clsAlimentacaoProduto();
    $lista = $obj_produto->lista();

    if( is_array( $lista ) && count( $lista ) )
    {
        foreach ( $lista AS $registro )
        {
            $opcoes[$registro["idpro"]] = $registro["nm_produto"];
        }
    }

    $this->campoLista( "ref_produto", "Produto", $opcoes, $this-
>ref_produto, "",false,"", "", "",true );

    $opcoes = array();

```

```

$obj_fornecedor = new clsAlimentacaoFornecedor();
$lista = $obj_fornecedor->lista();

if( is_array( $lista ) && count( $lista ) )
{
    foreach ( $lista AS $registro )
    {
        $opcoes[$registro["idpes"]] = $registro["fantasia"];
    }
}

$this->campoLista( "ref_fornecedor", "Fornecedor", $opcoes, $this->ref_fornecedor,"",false,"","",true );

$opcoes = array();
for ($i = 2008; $i <= date("Y");$i++)
{
    $opcoes[$i] = $i;
}
$this->campoLista( "ano", "Ano", $opcoes, $this->ano,"",false,"","",true );

$obj_produto_fornecedor = new clsAlimentacaoProdutoFornecedor();

$this->campoLista( "mes_inicio", "Mês - Início", $obj_produto_fornecedor->getArrayMes(), $this->mes_inicio,"",false,"","",true );

$this->campoLista( "mes_fim", "Mês - Fim", $obj_produto_fornecedor->getArrayMes(), $this->mes_fim,"",false,"","",true );

$this->campoMonetario( "pesoovolume_un", "Peso(Kg) ou Volume(L) por Un.", number_format($this->pesoovolume_un,2,"",), 30, 255, true );

$this->campoMonetario( "preco_un", "Preço Un.",number_format($this->preco_un,2,"",),10,16,true);

```

```

        $this->campoLista( "agri_familiar", "Agricultura Familiar", array("Não","Sim"), $this-
>agri_familiar,"",false,"", "", "",true );

```

```

    }

```

```

function Novo()

```

```

{

```

```

    @session_start();

```

```

    $this->pessoa_logada = $_SESSION['id_pessoa'];

```

```

    @session_write_close();

```

```

    $obj = new clsAlimentacaoProdutoFornecedor();

```

```

    $obj->ref_produto = $this->ref_produto;

```

```

    $obj->ref_fornecedor = $this->ref_fornecedor;

```

```

    $obj->ano = $this->ano;

```

```

    $obj->mes_inicio = $this->mes_inicio;

```

```

    $obj->mes_fim = $this->mes_fim;

```

```

    $obj->pesoouvolume_un = str_replace(",",".",$this->pesoouvolume_un);

```

```

    $obj->preco_un = str_replace(",",".",$this->preco_un);

```

```

    $obj->agri_familiar = $this->agri_familiar;

```

```

    $cadastrou = $obj->cadastra();

```

```

    if( $cadastrou )

```

```

    {

```

```

        $this->mensagem .= "Cadastro efetuado com sucesso.<br>";

```

```

        header( "Location: alimentacao_produto_fornecedor_lst.php" );

```

```

        die();

```

```

        return true;

```

```

    }

```

```

    $this->mensagem = "Cadastro não realizado.<br>";

```

```

    echo "<!--\nErro ao cadastrar clsAlimentacaoProduto\n-->";

```

```

    return false;

```

```

}

function Editar()
{
    @session_start();
    $this->peessoa_logada = $_SESSION['id_pessoa'];
    @session_write_close();

    $obj = new clsAlimentacaoProdutoFornecedor();
    $obj->idpf = $this->idpf;
    $obj->ref_produto = $this->ref_produto;
    $obj->ref_fornecedor = $this->ref_fornecedor;
    $obj->ano = $this->ano;
    $obj->mes_inicio = $this->mes_inicio;
    $obj->mes_fim = $this->mes_fim;
    $obj->pesoovolume_un = str_replace(",", ".", $this->pesoovolume_un);
    $obj->preco_un = str_replace(",", ".", $this->preco_un);
    $obj->agri_familiar = $this->agri_familiar;          $cadastrou = $obj->edita();
    if( $cadastrou )
    {
        $this->mensagem .= "Cadastro editado com sucesso.<br>";
        header( "Location: alimentacao_produto_fornecedor_det.php?idpf={$this-
>idpf}" );
        die();
        return true;
    }

    $this->mensagem = "Cadastro não editado.<br>";
    echo "<!--\nErro ao cadastrar clsAlimentacaoProdutoFornecedor\n-->";
    return false;
}

```

```

function Excluir()
{
    @session_start();
    $this->peessoa_logada = $_SESSION['id_pessoa'];
    @session_write_close();

    $obj = new clsAlimentacaoProdutoFornecedor();
    $obj->idpf = $this->idpf;
    $excluiu = $obj->exclui();
    if( $excluiu )
    {
        $this->mensagem .= "Cadastro excluído com sucesso.<br>";
        header( "Location: alimentacao_produto_fornecedor_lst.php" );
        die();
        return true;
    }

    $this->mensagem = "Cadastro não excluído.<br>";
    echo "<!--\nErro ao cadastrar clsAlimentacaoProdutoFornecedor\n-->";
    return false;

    return false;
}
}

// cria uma extensao da classe base
$pagina = new clsIndexBase();
// cria o conteudo
$miolo = new indice();
// adiciona o conteudo na clsBase
$pagina->addForm( $miolo );
// gera o html

```

```
$pagina->MakeAll();
```

```
?>
```

```
alimentacao_produto_fornecedor_det.php
```

```
<?php
```

```
/* ****
```

```
*
*
```

```

*
```

```
* @author Prefeitura Municipal de Itajaí
```

```
*
```

```
* @updated 29/03/2007
```

```
*
```

```
* Pacote: i-PLB Software Público Livre e Brasileiro
```

```
*
```

```
*
```

```

*
```

```
* Copyright (C) 2006 PMI - Prefeitura Municipal de Itajaí
```

```
*
```

```
*
```

```
ctima@itajai.sc.gov.br
```

```
*
```

```
*
```

```

*
```

```
* Este programa é software livre, você pode redistribuí-lo e/ou *
```

```
* modificá-lo sob os termos da Licença Pública Geral GNU, conforme *
```

```
* publicada pela Free Software Foundation, tanto a versão 2 da *
```

```
* Licença como (a seu critério) qualquer versão mais nova. *
```

```
*
```

```

*
```

```
* Este programa é distribuído na expectativa de ser útil, mas SEM *
```

```
* QUALQUER GARANTIA. Sem mesmo a garantia implícita de COMERCIALI- *
```

```
* ZAÇÃO ou de ADEQUAÇÃO A QUALQUER PROPÓSITO EM PARTICULAR. Con- *
```

```
*
```

```
* sulte a Licença Pública Geral GNU para obter mais detalhes. *
```

```
*
```

```

*
```

```
* Você deve ter recebido uma cópia da Licença Pública Geral GNU *
```

```
* junto com este programa. Se não, escreva para a Free Software *
```

```
* Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA *
```

```
* 02111-1307, USA.
```

```
*
```

```

*
*
*****/

require_once ("include/clsBase.inc.php");
require_once ("include/clsDetalhe.inc.php");
require_once ("include/clsBanco.inc.php");
require_once ("include/pmieducar/geral.inc.php" );
require_once ("include/alimentacao/geral.inc.php" );

class clsIndexBase extends clsBase
{
    function Formular()
    {
        $this->SetTitulo( "{$this->_instituicao} i-Educar - Produto Fornecedor" );
        $this->processoAp = "10004";
    }
}

class indice extends clsDetalhe
{
    /**
     * Titulo no topo da pagina
     *
     * @var int
     */
    var $titulo;

    var $idpf;

    function Gerar()
    {
        @session_start();
        $this->pessoa_logada = $_SESSION['id_pessoa'];
    }
}

```

```

session_write_close();

$this->titulo = "Produto Fornecedor - Detalhe";
$this->addBanner( "imagens/nvp_top_intranet.jpg", "imagens/nvp_vert_intranet.jpg",
"Intranet" );

$this->idpf=$_GET["idpf"];

$obj_produto_fornecedor = new clsAlimentacaoProdutoFornecedor();
$lista = $obj_produto_fornecedor->lista($this->idpf);
$registro = $lista[0];

if( ! $registro )
{
    header( "location: alimentacao_produto_fornecedor_lst.php" );
    die();
}

$this->addDetalhe( array( "Produto", $registro["nm_produto"] ) );

$this->addDetalhe( array( "Fornecedor", $registro["fantasia"] ) );

$this->addDetalhe( array( "Ano", $registro["ano"] ) );

$this->addDetalhe( array( "Mês Início", $obj_produto_fornecedor-
>getMes($registro["mes_inicio"]) ) );

$this->addDetalhe( array( "Mês Fim", $obj_produto_fornecedor-
>getMes($registro["mes_fim"]) ) );

$this->addDetalhe( array( "Quantidade por Un.", $registro["pesoouvolume_un"]." ".
$registro["unidade"] ) );

$this->addDetalhe( array( "Preço Un.", "R$".str_replace(".",",",
$registro["preco_un"]) ) );

```

```

        $this->addDetalhe( array( "Agricultura Familiar", $obj_produto_fornecedor-
>getAgri($registro["agri_familiar"])) );

```

```

        $obj_permissoes = new clsPermissoes();
        if( $obj_permissoes->permissao_cadastra( 10004, $this-> Pessoa_logada, 3 ) )
        {
            $this->url_novo = "alimentacao_produto_fornecedor_cad.php";
            $this->url_editar = "alimentacao_produto_fornecedor_cad.php?idpf={$this->idpf}";
        }

        $this->url_cancelar = "alimentacao_produto_fornecedor_lst.php";
        $this->largura = "100%";
    }
}

```

```

// cria uma extensao da classe base

```

```

$pagina = new clsIndexBase();

```

```

// cria o conteudo

```

```

$miolo = new indice();

```

```

// adiciona o conteudo na clsBase

```

```

$pagina->addForm( $miolo );

```

```

// gera o html

```

```

$pagina->MakeAll();

```

```

?>

```

```

alimentacao_produto_fornecedor_lst.php

```

```

<?php

```

```

/* *****

```

```

    *

```

```

    *

```

\* @author Prefeitura Municipal de Itajaí

\* @updated 29/03/2007

\* Pacote: i-PLB Software Público Livre e Brasileiro

\*

\*

\* Copyright (C) 2006 PMI - Prefeitura Municipal de Itajaí

\* ctima@itajai.sc.gov.br

\*

\*

\*

\* Este programa é software livre, você pode redistribuí-lo e/ou \*  
\* modificá-lo sob os termos da Licença Pública Geral GNU, conforme \*  
\* publicada pela Free Software Foundation, tanto a versão 2 da \*  
\* Licença como (a seu critério) qualquer versão mais nova. \*

\*

\*

\* Este programa é distribuído na expectativa de ser útil, mas SEM \*  
\* QUALQUER GARANTIA. Sem mesmo a garantia implícita de COMERCIALI- \*  
\* ZAÇÃO ou de ADEQUAÇÃO A QUALQUER PROPÓSITO EM PARTICULAR. Con- \*  
\* sulte a Licença Pública Geral GNU para obter mais detalhes. \*

\*

\*

\* Você deve ter recebido uma cópia da Licença Pública Geral GNU \*  
\* junto com este programa. Se não, escreva para a Free Software \*  
\* Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA \*  
\* 02111-1307, USA. \*

\*

\*

\*\*\*\*\*/

```
require_once ("include/clsBase.inc.php");
require_once ("include/clsListagem.inc.php");
require_once ("include/clsBanco.inc.php");
require_once ("include/pmieducar/geral.inc.php" );
```

```
require_once( "include/alimentacao/geral.inc.php" );
```

```
class clsIndexBase extends clsBase
```

```
{  
    function Formular()  
    {  
        $this->SetTitulo( "{$this->_instituicao} i-Educar - Produto Fornecedor " );  
        $this->processoAp = "10004";  
    }  
}
```

```
class indice extends clsListagem
```

```
{  
    /**  
     * Referencia pega da session para o idpes do usuario atual  
     *  
     * @var int  
     */  
    var $pessoa_logada;  
  
    /**  
     * Titulo no topo da pagina  
     *  
     * @var int  
     */  
    var $titulo;  
  
    /**  
     * Quantidade de registros a ser apresentada em cada pagina  
     *  
     * @var int  
     */  
    var $limite;
```

```

/**
 * Inicio dos registros a serem exibidos (limit)
 *
 * @var int
 */
var $offset;

var $ref_produto;
var $ref_fornecedor;

function Gerar()
{
    @session_start();
    $this->peessoa_logada = $_SESSION['id_peessoa'];
    session_write_close();

    $this->titulo = "Produto Fornecedor - Listagem";

    foreach( $_GET AS $var => $val ) // passa todos os valores obtidos no GET para
    atributos do objeto
        $this->$var = ( $val === "" ) ? null: $val;

    $this->addBanner( "imagens/nvp_top_intranet.jpg", "imagens/nvp_vert_intranet.jpg",
    "Intranet" );

    $obj_permissao = new clsPermissoes();

    $lista_busca = array(
        "Produto",
        "Fornecedor",
        "Ano",
        "Mês Inicio",
        "Mês Fim",

```

```

        "Quantidade por Un.",
        "Preço Un."
    );

    $this->addCabecalhos($lista_busca);

    $opcoes = array();
    $obj_produto = new clsAlimentacaoProduto();
    $lista = $obj_produto->lista();

    $opcoes["0"] = "Todos";
    if( is_array( $lista ) && count( $lista ) )
    {
        foreach ( $lista AS $registro )
        {
            $opcoes[$registro["idpro"]] = $registro["nm_produto"];
        }
    }

    $this->campoLista( "ref_produto", "Produto", $opcoes, $this->
    >ref_produto, "",false,"", "", "",false );

    $opcoes = array();
    $obj_fornecedor = new clsAlimentacaoFornecedor();
    $lista = $obj_fornecedor->lista();

    $opcoes["0"] = "Todos";
    if( is_array( $lista ) && count( $lista ) )
    {
        foreach ( $lista AS $registro )
        {
            $opcoes[$registro["idpes"]] = $registro["fantasia"];
        }
    }

```

```

    }

}

$this->campoLista( "ref_fornecedor", "Fornecedor", $opcoes, $this-
>ref_fornecedor,"",false,"","",false );

$obj_produto_fornecedor = new clsAlimentacaoProdutoFornecedor();
$filtro_produto = null;
if($this->ref_produto > 0)
{
    $filtro_produto = $this->ref_produto;
}
$filtro_fornecedor = null;
if($this->ref_fornecedor > 0)
{
    $filtro_fornecedor = $this->ref_fornecedor;
}
$lista = $obj_produto_fornecedor->lista(null,$filtro_produto,$filtro_fornecedor);

if( is_array( $lista ) && count( $lista ) )
{
    foreach ( $lista AS $registro )
    {

        $lista_busca = array();

        $lista_busca[] = "<a href=\"alimentacao_produto_fornecedor_det.php?
idpf={$registro[\"idpf\"]}\">{$registro[\"nm_produto\"]}</a>";

        $lista_busca[] = "<a href=\"alimentacao_produto_fornecedor_det.php?
idpf={$registro[\"idpf\"]}\">{$registro[\"fantasia\"]}</a>";

        $lista_busca[] = "<a href=\"alimentacao_produto_fornecedor_det.php?
idpf={$registro[\"idpf\"]}\">{$registro[\"ano\"]}</a>";

        $lista_busca[] = "<a href=\"alimentacao_produto_fornecedor_det.php?
idpf={$registro[\"idpf\"]}\">{$obj_produto_fornecedor->getMes($registro[\"mes_inicio\"])}</a>";

```



\* @author Prefeitura Municipal de Itajaí

\* @updated 29/03/2007 \*

\* Pacote: i-PLB Software Público Livre e Brasileiro \*

\* \*

\* Copyright (C) 2006 PMI - Prefeitura Municipal de Itajaí \*

\* ctima@itajai.sc.gov.br

\* \*

\* Este programa é software livre, você pode redistribuí-lo e/ou \*  
\* modificá-lo sob os termos da Licença Pública Geral GNU, conforme \*  
\* publicada pela Free Software Foundation, tanto a versão 2 da \*  
\* Licença como (a seu critério) qualquer versão mais nova. \*

\* \*

\* Este programa é distribuído na expectativa de ser útil, mas SEM \*  
\* QUALQUER GARANTIA. Sem mesmo a garantia implícita de COMERCIALI- \*  
\* ZAÇÃO ou de ADEQUAÇÃO A QUALQUER PROPÓSITO EM PARTICULAR. Con- \*  
\* sulte a Licença Pública Geral GNU para obter mais detalhes. \*

\* \*

\* Você deve ter recebido uma cópia da Licença Pública Geral GNU \*  
\* junto com este programa. Se não, escreva para a Free Software \*  
\* Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA \*  
\* 02111-1307, USA. \*

\* \*

\*\*\*\*\*/

```
require_once ("include/clsBase.inc.php");
require_once ("include/clsListagem.inc.php");
require_once ("include/clsBanco.inc.php");
require_once ("include/pmieducar/geral.inc.php" );
```

```
require_once( "include/alimentacao/geral.inc.php" );
```

```
class clsIndexBase extends clsBase
```

```
{  
    function Formular()  
    {  
        $this->SetTitulo( "{$this->_instituicao} i-Educar - Produto " );  
        $this->processoAp = "10002";  
    }  
}
```

```
class indice extends clsListagem
```

```
{  
    /**  
     * Referencia pega da session para o idpes do usuario atual  
     *  
     * @var int  
     */  
    var $pessoa_logada;  
  
    /**  
     * Titulo no topo da pagina  
     *  
     * @var int  
     */  
    var $titulo;  
  
    /**  
     * Quantidade de registros a ser apresentada em cada pagina  
     *  
     * @var int  
     */  
    var $limite;
```

```

/**
 * Inicio dos registros a serem exibidos (limit)
 *
 * @var int
 */
var $offset;

var $ref_produto_grupo;
var $nm_produto;

function Gerar()
{
    @session_start();
    $this->peessoa_logada = $_SESSION['id_pessoa'];
    session_write_close();

    $this->titulo = "Produto - Listagem";

    foreach( $_GET AS $var => $val ) // passa todos os valores obtidos no GET para
atributos do objeto
        $this->$var = ( $val === "" ) ? null: $val;

    $this->addBanner( "imagens/nvp_top_intranet.jpg", "imagens/nvp_vert_intranet.jpg",
    "Intranet" );

    $obj_permissao = new clsPermissoes();

    $lista_busca = array(
        "Produto",
        "Grupo",
        "Fator Correção",
        "Fator Cocção"
    );
}

```

```

$this->addCabecalhos($lista_busca);

$opcoes = array();
$obj_produto_grupo = new clsAlimentacaoProdutoGrupo();
$lista = $obj_produto_grupo->lista();

$opcoes["0"] = "Todos";
if( is_array( $lista ) && count( $lista ) )
{
    foreach ( $lista AS $registro )
    {
        $opcoes[$registro["idpg"]] = $registro["descricao"];
    }
}

$this->campoLista( "ref_produto_grupo", "Grupo", $opcoes, $this-
>ref_produto_grupo,"",false,"", "", "", false );

$obj_cardapio = new clsAlimentacaoProduto();
$filtro_grupo = null;
if($this->ref_produto_grupo > 0)
{
    $filtro_grupo = $this->ref_produto_grupo;
}
$lista = $obj_cardapio->lista($filtro_grupo,$this->nm_produto);

if( is_array( $lista ) && count( $lista ) )
{
    foreach ( $lista AS $registro )
    {
        $obj_ref_grupo = new clsAlimentacaoProdutoGrupo();
    }
}

```

```

        $det_ref_grupo = $obj_ref_grupo-
>lista($registro["ref_produto_grupo"]);

        $nm_grupo = $det_ref_grupo[0]["descricao"];

        $lista_busca = array();

        $lista_busca[] = "<a href=\"alimentacao_produto_det.php?
idpro={$registro["idpro"]}\">{$registro["nm_produto"]}</a>";

        $lista_busca[] = "<a href=\"alimentacao_produto_det.php?
idpro={$registro["idpro"]}\">{$nm_grupo}</a>";

        $lista_busca[] = "<a href=\"alimentacao_produto_det.php?
idpro={$registro["idpro"]}\">{$registro["fator_correcao"]}</a>";

        $lista_busca[] = "<a href=\"alimentacao_produto_det.php?
idpro={$registro["idpro"]}\">{$registro["fator_coccao"]}</a>";

        $this->addLinhas($lista_busca);
    }

}

if( $obj_permissao->permissao_cadastra( 10002, $this->peessoa_logada, 3 ) )
{
    $this->acao = "go(\"alimentacao_produto_cad.php\")";
    $this->nome_acao = "Novo";
}

$this->largura = "100%";
}
}

// cria uma extensao da classe base
$pagina = new clsIndexBase();

// cria o conteudo
$miolo = new indice();

// adiciona o conteudo na clsBase
$pagina->addForm( $miolo );

// gera o html

```

```
$pagina->MakeAll();
```

```
?>
```

```
alimentacao_relatorio_imc_escola_proc.ph
```

```
<?php
```

```
/* *****
```

```
*
*
```

```

*
```

```
* @author Prefeitura Municipal de Itajaí
```

```

*
```

```
* @updated 29/03/2007
```

```

*
```

```
* Pacote: i-PLB Software Público Livre e Brasileiro
```

```

*
```

```

*
```

```

*
```

```
* Copyright (C) 2006 PMI - Prefeitura Municipal de Itajaí
```

```

*
```

```
• ctima@itajai.sc.gov.br
```

```

*
```

```

*
```

```

*
```

```

*
```

```
* Este programa é software livre, você pode redistribuí-lo e/ou
```

```
* modificá-lo sob os termos da Licença Pública Geral GNU, conforme
```

```

*
```

```
* publicada pela Free Software Foundation, tanto a versão 2 da
```

```

*
```

```
* Licença como (a seu critério) qualquer versão mais nova.
```

```

*
```

```

*
```

```

*
```

```
* Este programa é distribuído na expectativa de ser útil, mas SEM
```

```

*
```

```
* QUALQUER GARANTIA. Sem mesmo a garantia implícita de COMERCIALI-
```

```

*
```

```
* ZAÍ ou de ADEQUAÍ A QUALQUER PROPRIEDADE PARTICULAR. Con-
```

```
* sulte a Licença Pública Geral GNU para obter mais detalhes.
```

```

*
```

```

*
```

```

*
```

```
* Você deve ter recebido uma cópia da Licença Pública Geral GNU
```

```

*
```

```
* junto com este programa. Se não, escreva para a Free Software
```

```

*
```

```
* Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA
```

```

*
```

```

*      02111-1307, USA.
*
*
*
*****/

require_once ("include/clsBase.inc.php");
require_once ("include/clsCadastro.inc.php");
require_once ("include/clsBanco.inc.php");
require_once ("include/pmieducar/geral.inc.php" );
require_once ("include/clsPDF.inc.php");

class clsIndexBase extends clsBase
{
    function Formular()
    {
        $this->SetTitulo( "{$this->_instituicao} i-Educar - Relatório - IMC por Escolas" );
        $this->processoAp = "10007";
    }
}

class indice extends clsCadastro
{
    /**
     * Referencia pega da session para o idpes do usuario atual
     *
     * @var int
     */
    var $pessoa_logada;

    var $sexo;
    var $ref_serie;

```

```
var $ano;

var $sequencial;

var $pdf;

var $pagina_atual = 1;
var $total_paginas = 1;

var $page_y = 135;

var $get_link = false;

var $total;

function Inicializar()
{
    $retorno = "Novo";
    @session_start();
    $this->peessoa_logada = $_SESSION['id_peessoa'];
    @session_write_close();

    $obj_permissoes = new clsPermissoes();

    return $retorno;
}

function Gerar()
{
    $obj_permissoes = new clsPermissoes();
    $nivel_usuario = $obj_permissoes->nivel_acesso($this->peessoa_logada);

    if($_POST){
        foreach ($_POST as $key => $value) {
```

```

        $this->$key = $value;

    }
}

//Campos de pesquisa

$opcoes = array();
for ($i = 2008; $i <= date("Y");$i++)
{
    $opcoes[$i] = $i;
}
$this->campoLista( "ano", "Ano", $opcoes, $this->ano,"",false,"", "", "",false );

$opcoes = array();
$obj_serie = new clsPmieducarSerie();
$lista = $obj_serie->lista();

$opcoes[""] = "Todas";
if( is_array( $lista ) && count( $lista ) )
{
    foreach ( $lista AS $registro )
    {
        $opcoes[$registro["cod_serie"]] = $registro["nm_serie"];
    }
}

$this->campoLista( "ref_serie", "Série", $opcoes, $this->ref_serie,"",false,"", "", "",false );

$this->campoLista( "sexo", "Sexo",
array("ambos"=>"Ambos","M"=>"Masculino","F"=>"Feminino"), $this->sexo,"",false,"", "", "",false );

```

```
$this->url_cancelar = "alimentacao_relatorio_imc_escola.php";
$this->nome_url_cancelar = "Cancelar";

$this->acao_enviar = 'acao2()';
$this->acao_executa_submit = false;

    }

}

// cria uma extensao da classe base
$pagina = new clsIndexBase();
// cria o conteudo
$miolo = new indice();
// adiciona o conteudo na clsBase
$pagina->addForm( $miolo );
// gera o html
$pagina->MakeAll();

?>

<script>

function acao2()
{

    if(!acao())
        return;
```

```

showExpansivelImprimir(400, 200,"",[], "Boletim");

document.formcadastro.target = 'miolo_'+(DOM_divs.length-1);

document.getElementById( 'btn_enviar' ).disabled =false;

document.formcadastro.submit();

}

document.getElementById('formcadastro').action ='alimentacao_relatorio_imc_escola_proc.php';

</script>

alimentacao_relatorio_imc_serie.php

<?php
/* *****
*
*
*   @author Prefeitura Municipal de Itajaí
*
*   @updated 29/03/2007
*
* Pacote: i-PLB Software Público Livre e Brasileiro
*
*
*   Copyright (C) 2006   PMI - Prefeitura Municipal de Itajaí
*
*                               ctima@itajai.sc.gov.br
*
*
*
*   Este programa é software livre, você pode redistribuí-lo e/ou
*   modificá-lo sob os termos da Licença Pública Geral GNU, conforme
*   publicada pela Free Software Foundation, tanto a versão 2 da

```



```
{
```

```
/**  
 * Referencia pega da session para o idpes do usuario atual  
 *  
 * @var int  
 */  
var $pessoa_logada;  
  
var $sexo;  
var $ref_escola;  
var $ano;  
  
var $sequencial;  
var $pdf;  
var $pagina_atual = 1;  
var $total_paginas = 1;  
  
var $page_y = 135;  
  
var $get_link = false;  
  
var $total;  
  
function Inicializar()  
{  
    $retorno = "Novo";  
    @session_start();  
    $this->pessoa_logada = $_SESSION['id_pessoa'];  
    @session_write_close();  
}
```

```

$obj_permissoes = new clsPermissoes();

return $retorno;
}

function Gerar()
{

$obj_permissoes = new clsPermissoes();
$nivel_usuario = $obj_permissoes->nivel_acesso($this->peessoa_logada);

if($_POST){
    foreach ($_POST as $key => $value) {
        $this->$key = $value;
    }
}

//Campos de pesquisa

$opcoes = array();
for ($i = 2008; $i <= date("Y");$i++)
{
    $opcoes[$i] = $i;
}
$this->campoLista( "ano", "Ano", $opcoes, $this->ano,"",false,"","",false );

$opcoes = array();
$obj_escola = new clsPmieducarEscola();
$lista = $obj_escola->lista();

$opcoes[""] = "Todas";

```

```

if( is_array( $lista ) && count( $lista ) )
{
    foreach ( $lista AS $registro )
    {
        $opcoes[$registro["cod_escola"]] = $registro["nome"];
    }
}

$this->campoLista( "ref_escola", "Escola", $opcoes, $this-
>ref_escola, "", false, "", "", "", false );

$this->campoLista( "sexo", "Sexo",
array("ambos"=>"Ambos", "M"=>"Masculino", "F"=>"Feminino"), $this->sexo, "", false, "", "", "", false );

$this->url_cancelar = "alimentacao_relatorio_imc_serie.php";
$this->nome_url_cancelar = "Cancelar";

$this->acao_enviar = 'acao2()';
$this->acao_executa_submit = false;

}

}

// cria uma extensao da classe base
$pagina = new clsIndexBase();
// cria o conteudo
$miolo = new indice();
// adiciona o conteudo na clsBase
$pagina->addForm( $miolo );
// gera o html

```

```
$pagina->MakeAll();
```

```
?>
```

```
<script>
```

```
function acao2()
```

```
{
```

```
    if(!acao())
```

```
        return;
```

```
    showExpansivellmpimir(400, 200,"",[], "Boletim");
```

```
    document.formcadastro.target = 'miolo_'+(DOM_divs.length-1);
```

```
    document.getElementById( 'btn_enviar' ).disabled =false;
```

```
    document.formcadastro.submit();
```

```
}
```

```
document.getElementById('formcadastro').action ='alimentacao_relatorio_imc_serie_proc.php';
```

```
</script>
```

```
alimentacao_relatorio_imc_serie_proc.php
```

```
<?php
```

```
    /* *****
```

```
    *
```

```
    *
```

```
    *      @author Prefeitura Municipal de Itajaí
```

```
    *
```

\* @updated 29/03/2007

\* Pacote: i-PLB Software Público Livre e Brasileiro

\*

\*

\* Copyright (C) 2006 PMI - Prefeitura Municipal de Itajaí

\*

ctima@itajai.sc.gov.br

\*

\*

\*

\* Este programa é software livre, você pode redistribuí-lo e/ou \*  
\* modificá-lo sob os termos da Licença Pública Geral GNU, conforme \*  
\* publicada pela Free Software Foundation, tanto a versão 2 da \*  
\* Licença como (a seu critério) qualquer versão mais nova. \*

\*

\*

\* Este programa é distribuído na expectativa de ser útil, mas SEM \*  
\* QUALQUER GARANTIA. Sem mesmo a garantia implícita de COMERCIALI- \*  
\* ZAÇÃO ou de ADEQUAÇÃO A QUALQUER PROPÓSITO EM PARTICULAR. Con- \*  
\* sulte a Licença Pública Geral GNU para obter mais detalhes. \*

\*

\*

\* Você deve ter recebido uma cópia da Licença Pública Geral GNU \*  
\* junto com este programa. Se não, escreva para a Free Software \*  
\* Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA \*  
\* 02111-1307, USA. \*

\*

\*

\*\*\*\*\*/

```
require_once ("include/clsBase.inc.php");
require_once ("include/clsCadastro.inc.php");
require_once ("include/clsBanco.inc.php");
require_once ("include/pmieducar/geral.inc.php" );
require_once ("include/clsPDF.inc.php");
```

```

class clsIndexBase extends clsBase
{
    function Formular()
    {
        $this->SetTitulo( "{$this->_instituicao} i-Educar - Relatório - IMC por Série" );
        $this->processoAp = "10008";
        $this->renderMenu = false;
        $this->renderMenuSuspendo = false;
    }
}

```

```

class indice extends clsCadastro
{

    /**
     * Referencia pega da session para o idpes do usuario atual
     *
     * @var int
     */
    var $pessoa_logada;

    var $ano;
    var $sexo;
    var $ref_escola;

    var $nm_instituicao;

    var $pdf;

    var $page_y = 139;

```

```
var $get_link;
```

```
var $campo_assinatura;
```

```
var $total = 0;
```

```
function renderHTML()
```

```
{
```

```
    if($_POST){
```

```
        foreach ($_POST as $key => $value) {
```

```
            $this->$key = $value;
```

```
        }
```

```
    }
```

```
    $fonte = 'arial';
```

```
    $corTexto = '#000000';
```

```
    $filtro_anoescola = "";
```

```
    $filtro_sexo = "";
```

```
    if (is_numeric($this->ref_escola))
```

```
        $filtro_anoescola .= " AND a.ref_escola = {$this->ref_escola} ";
```

```
    if (is_numeric($this->ano))
```

```
        $filtro_anoescola .= " AND date_part('year', a.dt_cadastro) = {$this->ano} ";
```

```
    if ($this->sexo!="ambos")
```

```
        $filtro_sexo .= " AND c.sexo = '{$this->sexo}' ";
```

```
    $sql = "select ref_serie, nm_serie, (sum(imc)/count(1)) as imc from alimentacao.imc a
```

```
        join
```

```
        pmieducar.serie s
```

```
        on s.cod_serie=a.ref_serie {$filtro->anoescola}
```

```
        join
```

```
        (
```

```
        select cod_aluno, sexo from pmieducar.aluno a join cadastro.fisica f
```

```

        on a.ref_idpes=f.idpes
        ) c
        on c.cod_aluno=a.ref_aluno {$filtro->sexo}
        group by ref_serie, nm_serie
        order by nm_serie
        ";
$db = new clsBanco();
$db->Consulta($sql);

$this->total = $db->Num_Linhas();
if ($this->total > 0)
{

        $this->nm_instituicao = "Prefeitura X";
        $this->pdf = new clsPDF("IMC por Série - {$this->ano}", "IMC por Série", "A4",
"", false, false);

        $this->pdf->largura = 842.0;
        $this->pdf->altura = 595.0;
        $this->page_y = 125;
        $this->pdf->OpenFile();
        $this->addCabecalho();
        $esquerda = 30;
        $altura = 130 + 18 * 2;
        $direita = 782;
        $tam_texto = 9;
        $altura = 130;
        $altura_escrita = 3;
        $ref_cod_escola_aux = null;
        while ( $db->ProximoRegistro() )
        {

                list($ref_serie,$serie, $imc) = $db->Tupla();

                $esquerda_aux = $esquerda = 30;

```

```

$this->pdf->linha_relativa($esquerda, $altura+=18, 0, 18);
$this->pdf->linha_relativa($esquerda, $altura, $direita, 0);
$this->pdf->escreve_relativo($serie, $esquerda + 3, $altura +
$altura_escrita, 600, 30, $fonte, $tam_texto, $corTexto);
$this->pdf->linha_relativa($esquerda += 650, $altura, 0, 18);

$this->pdf->escreve_relativo(number_format($imc,2,"", ""), $esquerda
+ 1, $altura + $altura_escrita, 132, 30, $fonte, $tam_texto, $corTexto, 'center');
$this->pdf->linha_relativa($esquerda += 132, $altura, 0, 18);

$this->pdf->linha_relativa($esquerda_aux, $altura + 18, $direita, 0);
if ($altura > $this->pdf->altura - 50)
{
    $this->pdf->ClosePage();
    $this->pdf->OpenPage();
    $this->addCabecalho();
    $esquerda = 30;
    $altura = 130 + 18*2;
    $direita = 782;
    $tam_texto = 9;
    $altura = 130;

    $altura_escrita = 5;
}
}
$this->pdf->CloseFile();
$this->get_link = $this->pdf->GetLink();

echo "<script>window.onload=function()
{parent.EscondeDiv('LoadImprimir');window.location='download.php?filename=".$this-
>get_link."'}</script>";

echo "<html><center>Se o download não iniciar automaticamente <br /><a
target='_blank' href="" . $this->get_link . "" style='font-size: 16px; color: #000000; text-decoration:
underline;'>clique aqui!</a><br><br>";

```

`<span style='font-size: 10px;'>Para visualizar os arquivos PDF, é necessário instalar o Adobe Acrobat Reader.<br>`

`Clique na Imagem para Baixar o instalador<br><br>`

`<a href='\"http://www.adobe.com.br/products/acrobat/readstep2.html\"' target='\"new\\\"><br><img src='\"imagens/acrobat.gif\"' width='\"88\\\" height='\"31\\\" border='\"0\\\"></a>`

`</span>`

`</center>";`

`}`

`else`

`{`

`echo '<script>`

`alert("Nenhum registro encontrado.");`

`window.parent.fechaExpansivel('div_dinamico_'+`

`(window.parent.DOM_divs.length-1));`

`</script>';`

`return true;`

`}`

`}`

`function addCabecalho()`

`{`

`// variavel que controla a altura atual das caixas`

`$altura = 30;`

`$fonte = 'arial';`

`$corTexto = '#000000';`

`// cabecalho`

`$this->pdf->quadrado_relativo( 30, $altura, 782, 85 );`

`$this->pdf->InsertJpng( "gif", "c:/paginas/ieducar/intranet/imagens/brasao.gif", 50, 95, 0.30 );`

`//titulo principal`

```

        $this->pdf->escreve_relativo( "PREFEITURA X", 30, 30, 782, 80, $fonte, 18,
$corTexto, 'center' );

        $this->pdf->escreve_relativo( date("d/m/Y"), 745, 30, 100, 80, $fonte, 12, $corTexto,
'left' );

        //dados escola

        $this->pdf->escreve_relativo( "Instituição: $this->nm_instituicao", 120, 52, 300, 80,
$fonte, 9, $corTexto, 'left' );

        if($this->ref_escola=="")
        {
            $escola = "todas";
        }
        else
        {
            $obj_ref_cod_escola = new clsPmieducarEscola( $this->ref_escola );
            $det_ref_cod_escola = $obj_ref_cod_escola->detalhe();
            $escola = $det_ref_cod_escola["nome"];
        }

        $this->pdf->escreve_relativo( "IMC por Série - Ano: {$this->ano} / Escola: {$escola} /
Sexo: {$this->sexo}", 30, 78, 782, 80, $fonte, 12, $corTexto, 'center' );

        $this->pdf->escreve_relativo( "Total de Séries: {$this->total}", 30, 95, 782, 80, $fonte,
10, $corTexto, 'center' );

        $this->pdf->linha_relativa(30, $altura += 100, 782, 0);

        $esquerda = 30;
        $altura = 30;
        $direita = 782;
        $tam_texto = 10;
        $altura = 130;

        $this->pdf->linha_relativa($esquerda, $altura, 0, 18);

```

```

        $this->pdf->escreve_relativo("Série", $esquerda + 3, $altura+3, 600, 30, $fonte,
    $tam_texto);

```

```

        $this->pdf->linha_relativa($esquerda += 650, $altura, 0, 18);

```

```

        $this->pdf->escreve_relativo("IMC", $esquerda + 1, $altura+3, 130, 30, $fonte,
    $tam_texto, $corTexto, 'center');

```

```

        $this->pdf->linha_relativa($esquerda += 132, $altura, 0, 18);

```

```

        $this->page_y +=19;

```

```

    }

```

```

function Editar()

```

```

{

```

```

    return false;

```

```

}

```

```

function Excluir()

```

```

{

```

```

    return false;

```

```

}

```

```

}

```

```

// cria uma extensao da classe base

```

```

$pagina = new clsIndexBase();

```

```

// cria o conteudo

```

```

$miolo = new indice();

```

```

// adiciona o conteudo na clsBase

```

```

$pagina->addForm( $miolo );

```

```

// gera o html

```

```

$pagina->MakeAll();

```

?>

alimentacao\_relatorio\_imc\_escola.php

<?php

/\* \*\*\*\*

\*

\*

\* @author Prefeitura Municipal de Itajaí

\*

\* @updated 29/03/2007

\*

\* Pacote: i-PLB Software Público Livre e Brasileiro

\*

\*

\*

\* Copyright (C) 2006 PMI - Prefeitura Municipal de Itajaí

\*

\*

ctima@itajai.sc.gov.br

\*

\*

\*

\* Este programa é software livre, você pode redistribuí-lo e/ou

\*

\* modificá-lo sob os termos da Licença Pública Geral GNU, conforme

\*

\* publicada pela Free Software Foundation, tanto a versão 2 da

\*

\* Licença como (a seu critério) qualquer versão mais nova.

\*

\*

\*

\* Este programa é distribuído na expectativa de ser útil, mas SEM

\*

\* QUALQUER GARANTIA. Sem mesmo a garantia implícita de COMERCIALI-

\*

\* ZAÍO ou de ADEQUAÇÃO A QUALQUER PROPÓSITO EM PARTICULAR. Con-

\* sulte a Licença Pública Geral GNU para obter mais detalhes.

\*

\*

\*

\* Você deve ter recebido uma cópia da Licença Pública Geral GNU

\*

\* junto com este programa. Se não, escreva para a Free Software

\*

\* Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA

\*

\* 02111-1307, USA.

\*

```

*
*
*****/

require_once ("include/clsBase.inc.php");
require_once ("include/clsCadastro.inc.php");
require_once ("include/clsBanco.inc.php");
require_once ("include/pmieducar/geral.inc.php" );
require_once ("include/clsPDF.inc.php");
require_once ("include/alimentacao/geral.inc.php" );

class clsIndexBase extends clsBase
{
    function Formular()
    {
        $this->SetTitulo( "{$this->_instituicao} i-Educar - Relatório - Resumo por Escola" );
        $this->processoAp = "10009";
    }
}

class indice extends clsCadastro
{
    /**
     * Referencia pega da session para o idpes do usuario atual
     *
     * @var int
     */
    var $pessoa_logada;

    var $mes;
    var $ano;

```

```
var $sequencial;
var $pdf;
var $pagina_atual = 1;
var $total_paginas = 1;

var $page_y = 135;

var $get_link = false;

var $total;

function Inicializar()
{
    $retorno = "Novo";
    @session_start();
    $this->pessoa_logada = $_SESSION['id_pessoa'];
    @session_write_close();

    $obj_permissoes = new clsPermissoes();

    return $retorno;
}

function Gerar()
{
    $obj_permissoes = new clsPermissoes();
    $nivel_usuario = $obj_permissoes->nivel_acesso($this->pessoa_logada);

    if($_POST){
        foreach ($_POST as $key => $value) {
            $this->$key = $value;
        }
    }
}
```

```
        }
    }

    //Campos de pesquisa

    $opcoes = array();
    for ($i = 2008; $i <= date("Y");$i++)
    {
        $opcoes[$i] = $i;
    }
    $this->campoLista( "ano", "Ano", $opcoes, $this->ano, "", false, "", "", "", false );

    $obj_envio = new clsAlimentacaoEnvioMensalEscola();

    $opcoes = array();
    $opcoes = $obj_envio->getArrayMes();
    $opcoes[""] = "Todos";

    $this->campoLista( "mes", "Mês", $opcoes, $this->mes, "", false, "", "", "", false );

    $this->url_cancelar = "alimentacao_relatorio_resumo_escola.php";
    $this->nome_url_cancelar = "Cancelar";

    $this->acao_enviar = 'acao2()';
    $this->acao_executa_submit = false;

}

}
```

```

// cria uma extensao da classe base
$pagina = new clsIndexBase();
// cria o conteudo
$miolo = new indice();
// adiciona o conteudo na clsBase
$pagina->addForm( $miolo );
// gera o html
$pagina->MakeAll();

```

```
?>
```

```
<script>
```

```
function acao2()
{
```

```

    if(!acao())
        return;

```

```

    showExpansivelImprimir(400, 200,"", "Boletim");

```

```

    document.formcadastro.target = 'miolo_'+(DOM_divs.length-1);

```

```

    document.getElementById( 'btn_enviar' ).disabled =false;

```

```

    document.formcadastro.submit();

```

```
}
```

```

document.getElementById('formcadastro').action ='alimentacao_relatorio_resumo_escola_proc.php';

```

```
</script>
```

alimentacao\_relatorio\_imc\_escola\_proc.php

<?php

/\* \*\*\*\* \*/

\*

\*

\* @author Prefeitura Municipal de Itajaí

\*

\* @updated 29/03/2007

\*

\* Pacote: i-PLB Software Público Livre e Brasileiro

\*

\*

\*

\* Copyright (C) 2006 PMI - Prefeitura Municipal de Itajaí

\*

\*

ctima@itajai.sc.gov.br

\*

\*

\*

\* Este programa é software livre, você pode redistribuí-lo e/ou \*

\* modificá-lo sob os termos da Licença Pública Geral GNU, conforme \*

\* publicada pela Free Software Foundation, tanto a versão 2 da \*

\* Licença como (a seu critério) qualquer versão mais nova. \*

\*

\*

\* Este programa é distribuído na expectativa de ser útil, mas SEM \*

\* QUALQUER GARANTIA. Sem mesmo a garantia implícita de COMERCIALI- \*

\* ZAÇÃO ou de ADEQUAÇÃO A QUALQUER PROPÓSITO EM PARTICULAR. Con- \*

\*

\* sulte a Licença Pública Geral GNU para obter mais detalhes. \*

\*

\*

\* Você deve ter recebido uma cópia da Licença Pública Geral GNU \*

\* junto com este programa. Se não, escreva para a Free Software \*

\* Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA \*

\* 02111-1307, USA.

\*

\*

\*

\*\*\*\* \*/

```
require_once ("include/clsBase.inc.php");
require_once ("include/clsCadastro.inc.php");
require_once ("include/clsBanco.inc.php");
require_once ("include/pmieducar/geral.inc.php" );
require_once ("include/clsPDF.inc.php");
require_once ("include/alimentacao/geral.inc.php" );

class clsIndexBase extends clsBase
{
    function Formular()
    {
        $this->SetTitulo( "{$this->_instituicao} i-Educar - Relatório - Resumo por escola" );
        $this->processoAp = "10009";
        $this->renderMenu = false;
        $this->renderMenuSuspenso = false;
    }
}

class indice extends clsCadastro
{
    /**
     * Referencia pega da session para o idpes do usuario atual
     *
     * @var int
     */
    var $pessoa_logada;

    var $ano;
    var $mes;
```

```
var $nm_instituicao;
```

```
var $pdf;
```

```
var $page_y = 139;
```

```
var $get_link;
```

```
var $campo_assinatura;
```

```
var $total = 0;
```

```
function renderHTML()
```

```
{
```

```
    if($_POST){
```

```
        foreach ($_POST as $key => $value) {
```

```
            $this->$key = $value;
```

```
        }
```

```
    }
```

```
    $fonte = 'arial';
```

```
    $corTexto = '#000000';
```

```
    $filtro = "";
```

```
    if (is_numeric($this->ano))
```

```
        $filtro .= " AND date_part('year', eme.dt_cadastro) = {$this->ano} ";
```

```
    if (is_numeric($this->mes))
```

```
        $filtro .= " AND eme.mes = '{$this->mes}' ";
```

```
    $sql = "select
```

```
        sum(custo) as custototal,
```

```
        (sum(custo)/sum(refeicoes)) as custoporrefeicao,
```

```
        (sum(calorias)/sum(alunosdias)) as caloriasdiarias,
```

```

(sum(proteinas)/sum(alunosdias)) as proteinasdiarias,
(sum(custo_familiar)) as custo_agrifamiliar,
ref_escola,
escola
from
(
select
sum((emep.pesoouvolumepf.pesoouvolumepf.preco_un) as
custo,
sum(((emep.pesoouvolumepf.pesoouvolumepf.preco_un)/
(p.fator_correcao/100))/100)*p.calorias) as calorias,
sum(((emep.pesoouvolumepf.pesoouvolumepf.preco_un)/
(p.fator_correcao/100))/100)*p.proteinas) as proteinas,
eme.ref_escola,
eme.mes,
(eme.alunos*eme.dias*eme.refeicoes) as refeicoes,
(eme.alunos*eme.dias) as alunosdias,
sum(case when agri_familiar = 1 then
((emep.pesoouvolumepf.pesoouvolumepf.preco_un) else 0 end) as custo_familiar
from alimentacao.envio_mensal_escola eme join
alimentacao.envio_mensal_escola_produto emep
on eme.ideme=emep.ref_envio_mensal_escola {$filtro}
join alimentacao.produto p on emep.ref_produto=p.idpro
left join alimentacao.produto_fornecedor pf
on emep.ref_produto=pf.ref_produto and eme.mes between
pf.mes_inicio and pf.mes_fim
group by eme.ref_escola, eme.ano, eme.mes, eme.alunos, eme.dias,
eme.refeicoes
) tb1
join
(
select cod_escola, (case when nome is null then nm_escola else
nome end) as escola from pmieducar.escola e left join pmieducar.escola_complemento ec
on e.cod_escola=ec.ref_cod_escola
left join cadastro.pessoa p
on p.idpes=e.ref_idpes

```

```

        )
        tb2
        on tb1.ref_escola=tb2.cod_escola
        group by ref_escola, escola
        order by escola
    ";
    $db = new clsBanco();
    $db->Consulta($sql);

    $this->total = $db->Num_Linhas();
    if ($this->total > 0)
    {

        $this->nm_instituicao = "Prefeitura X";
        $this->pdf = new clsPDF("Resumo por Escola - {$this->ano}", "Resumo por
Escola", "A4", "", false, false);
        $this->pdf->largura = 842.0;
        $this->pdf->altura = 595.0;
        $this->page_y = 125;
        $this->pdf->OpenFile();
        $this->addCabecalho();
        $esquerda = 30;
        $altura = 130 + 18 * 2;
        $direita = 782;
        $tam_texto = 9;
        $altura = 130;
        $altura_escrita = 3;
        $ref_cod_escola_aux = null;
        while ( $db->ProximoRegistro() )
        {
            list($custototal,$custoporrefeicao,$caloriasdiarias,$proteinasdiarias,
$custo_agrifamiliar,$ref_escola,$escola) = $db->Tupla();

            $esquerda_aux = $esquerda = 30;

```

```

$this->pdf->linha_relativa($esquerda, $altura+=18, 0, 18);

$this->pdf->linha_relativa($esquerda, $altura, $direita, 0);

    $this->pdf->escreve_relativo($escola, $esquerda + 3, $altura +
    $altura_escrita, 200, 30, $fonte, $tam_texto, $corTexto);

    $this->pdf->linha_relativa($esquerda += 210, $altura, 0, 18);

    $this->pdf->escreve_relativo("R$
    ".number_format($custototal,2,"","."), $esquerda + 1, $altura + $altura_escrita, 100, 30, $fonte,
    $tam_texto, $corTexto, 'center');

    $this->pdf->linha_relativa($esquerda += 102, $altura, 0, 18);

    $this->pdf->escreve_relativo("R$
    ".number_format($custoporrefeicao,2,"","."), $esquerda + 1, $altura + $altura_escrita, 115, 30, $fonte,
    $tam_texto, $corTexto, 'center');

    $this->pdf->linha_relativa($esquerda += 117, $altura, 0, 18);

    $this->pdf-
    >escreve_relativo(number_format($caloriasdiarias,2,"",".") " Kcal", $esquerda + 1, $altura +
    $altura_escrita, 115, 30, $fonte, $tam_texto, $corTexto, 'center');

    $this->pdf->linha_relativa($esquerda += 118, $altura, 0, 18);

    $this->pdf-
    >escreve_relativo(number_format($proteinasdiarias,2,"",".") " gramas", $esquerda + 1, $altura +
    $altura_escrita, 115, 30, $fonte, $tam_texto, $corTexto, 'center');

    $this->pdf->linha_relativa($esquerda += 117, $altura, 0, 18);

    $this->pdf->escreve_relativo("R$
    ".number_format($custo_agrifamiliar,2,"",".") (" ".number_format((( $custo_agrifamiliar/
    $custototal)*100),1,"",".") "%)", $esquerda + 1, $altura + $altura_escrita, 115, 30, $fonte, $tam_texto,
    $corTexto, 'center');

    $this->pdf->linha_relativa($esquerda += 118, $altura, 0, 18);

    $this->pdf->linha_relativa($esquerda_aux, $altura + 18, $direita, 0);

    if ($altura > $this->pdf->altura - 50)
    {

        $this->pdf->ClosePage();

        $this->pdf->OpenPage();
    }

```

```

        $this->addCabecalho();

        $esquerda = 30;

        $altura = 130 + 18*2;

        $direita = 782;

        $tam_texto = 9;

        $altura = 130;

        $altura_escrita = 5;

    }

}

$this->pdf->CloseFile();

$this->get_link = $this->pdf->GetLink();

        echo "<script>>window.onload=function()
{parent.EscondeDiv('LoadImprimir');window.location='download.php?filename=".$this-
>get_link."'}/</script>";

        echo "<html><center>Se o download não iniciar automaticamente <br /><a
target='_blank' href='\" . $this->get_link . \"' style='font-size: 16px; color: #000000; text-decoration:
underline;'>clique aqui!</a><br><br>

        <span style='font-size: 10px;'>Para visualizar os arquivos
PDF, é necessário instalar o Adobe Acrobat Reader.<br>

        Clique na Imagem para Baixar o instalador<br><br>

        <a
href=\"http://www.adobe.com.br/products/acrobat/readstep2.html\" target=\"new\"><br><img
src=\"imagens/acrobat.gif\" width=\"88\" height=\"31\" border=\"0\"></a>

        </span>

        </center>";

    }

    else
    {

        echo '<script>

                alert("Nenhum registro encontrado.");

                window.parent.fechaExpansivel('\div_dinamico_' +
(window.parent.DOM_divs.length-1));

```

```

        </script>;
        return true;
    }
}

function addCabecalho()
{
    // variavel que controla a altura atual das caixas
    $altura = 30;
    $fonte = 'arial';
    $corTexto = '#000000';

    // cabecalho
    $this->pdf->quadrado_relativo( 30, $altura, 782, 85 );
    $this->pdf->InsertJpng( "gif", "c:/paginas/educar/intranet/imagens/brasao.gif", 50, 95,
0.30 );

    //titulo principal
    $this->pdf->escreve_relativo( "PREFEITURA X", 30, 30, 782, 80, $fonte, 18,
$corTexto, 'center' );
    $this->pdf->escreve_relativo( date("d/m/Y"), 745, 30, 100, 80, $fonte, 12, $corTexto,
'left' );

    //dados escola
    $this->pdf->escreve_relativo( "Instituição: $this->nm_instituicao", 120, 52, 300, 80,
$fonte, 9, $corTexto, 'left' );

    if($this->mes=="")
    {
        $mes = "todos";
    }
    else
    {
        $obj_envio = new clsAlimentacaoEnvioMensalEscola();
        $mes = $obj_envio->getMes($this->mes);
    }
}

```

```
}
```

```
$this->pdf->escreve_relativo( "Resumo por Escola - Ano: {$this->ano} / Mês: {$mes} ",
30, 78, 782, 80, $fonte, 12, $corTexto, 'center' );
```

```
$this->pdf->escreve_relativo( "Total de Escolas: {$this->total}", 30, 95, 782, 80,
$fonte, 10, $corTexto, 'center' );
```

```
$this->pdf->linha_relativa(30, $altura += 100, 782, 0);
```

```
$esquerda = 30;
```

```
$altura = 30;
```

```
$direita = 782;
```

```
$tam_texto = 10;
```

```
$altura = 130;
```

```
$this->pdf->linha_relativa($esquerda, $altura, 0, 18);
```

```
$this->pdf->escreve_relativo("Escola", $esquerda + 3, $altura+3, 200, 30, $fonte,
$tam_texto);
```

```
$this->pdf->linha_relativa($esquerda += 210, $altura, 0, 18);
```

```
$this->pdf->escreve_relativo("Custo Total", $esquerda + 1, $altura+3, 100, 30, $fonte,
$tam_texto, $corTexto, 'center');
```

```
$this->pdf->linha_relativa($esquerda += 102, $altura, 0, 18);
```

```
$this->pdf->escreve_relativo("Custo por Refeição", $esquerda + 1, $altura+3, 115, 30,
$fonte, $tam_texto, $corTexto, 'center');
```

```
$this->pdf->linha_relativa($esquerda += 117, $altura, 0, 18);
```

```
$this->pdf->escreve_relativo("Calorias Diárias/Aluno", $esquerda + 1, $altura+3, 115,
30, $fonte, $tam_texto, $corTexto, 'center');
```

```
$this->pdf->linha_relativa($esquerda += 118, $altura, 0, 18);
```

```
$this->pdf->escreve_relativo("Proteínas Diárias/Aluno", $esquerda + 1, $altura+3,
115, 30, $fonte, $tam_texto, $corTexto, 'center');
```

```
$this->pdf->linha_relativa($esquerda += 117, $altura, 0, 18);
```

```
$this->pdf->escreve_relativo("Agric. Familiar", $esquerda + 1, $altura+3, 115, 30,  
$fonte, $tam_texto, $corTexto, 'center');
```

```
$this->pdf->linha_relativa($esquerda += 118, $altura, 0, 18);
```

```
$this->page_y +=19;
```

```
}
```

```
function Editar()
```

```
{
```

```
    return false;
```

```
}
```

```
function Excluir()
```

```
{
```

```
    return false;
```

```
}
```

```
}
```

```
// cria uma extensao da classe base
```

```
$pagina = new clsIndexBase();
```

```
// cria o conteudo
```

```
$miolo = new indice();
```

```
// adiciona o conteudo na clsBase
```

```
$pagina->addForm( $miolo );
```

```
// gera o html
```

```
$pagina->MakeAll();
```

```
?>
```