Utilização da tecnologia JavaServer Faces para desenvolvimento de um sistema de controle nutricional da alimentação escolar

Gustavo Krauczuk[1], Rodrigo Souza D'Ávila[1], Elena Mariele Bini[1]

[1] Tecnologia em Análise e Desenvolvimento de Sistemas
Faculdade Guairacá – Guarapuava – Paraná

gustavo0893@gmail.com, rsdavila@gmail.com, elena_bini@yahoo.com.br

Abstract. *In this article, presents the performed steps for the development of a web system that have by objective help the nutritionist active in menu elaboration for academic feeding, reducing the time spent with this work and confiability increasing of this process. For development of this system, was employed the JavaServer Faces technology, together with Hibernate and MySQL database. The system use brought benefits in the menu elaboration process and calculating nutrient.*

Resumo. *Nesse artigo, apresentam-se as etapas realizadas para o desenvolvimento de um sistema web que tem por objetivo auxiliar o nutricionista que atua na elaboração de cardápios para a alimentação escolar, reduzindo o tempo gasto com este trabalho e aumentando a confiabilidade do processo. Para o desenvolvimento deste sistema empregou-se a tecnologia JavaServer Faces, juntamente com o Hibernate e o banco de dados MySQL. A utilização do sistema trouxe benefícios no processo de elaboração de cardápios e de cálculos de nutrientes.*

Introdução

O nutricionista é um profissional da saúde que atua buscando a promoção, manutenção e recuperação da saúde através da educação alimentar. Em especial os nutricionistas que atuam nas Secretarias Municipais de Educação, estes tem o papel de elaborar cardápios destinados aos alunos das instituições municipais de ensino, visando garantir-lhes uma alimentação de qualidade. Estes devem planejá-los de forma a suprir as necessidades nutricionais dos alunos, levando em consideração fatores como a idade e os alimentos mais adequados para eles.

Partindo dessa necessidade desenvolveu-se o sistema SCAE (Sistema de Controle da Alimentação Escolar), apresentado nesse trabalho, tendo como objetivo auxiliar o nutricionista da Secretaria Municipal de Educação de Pitanga, Paraná, na tarefa de elaboração dos cardápios destinados às instituições públicas de ensino, diminuindo o tempo gasto com este trabalho e aumentando a confiabilidade do processo. Com a utilização do *software* desenvolvido, o nutricionista pode elaborar cardápios de forma mais rápida e confiável, permitindo-o saber quais alimentos estão sendo consumidos pelos alunos e se estes atendem às suas necessidades nutricionais. Ainda o SCAE possibilita a geração de cardápios impressos para serem destinados às instituições de ensino, contribuindo, assim, para um maior conhecimento do que está sendo consumido nas escolas da rede pública.

O *software* foi desenvolvido em linguagem Java utilizando o framework *JavaServer Faces*. Para tal desenvolvimento também foram empregadas as ferramentas Astah para a elaboração de diagramas propostos pela UML e o Netbeans 7.3 para programação. Para persistência e recuperação de dados foi utilizado o banco de dados MySQL, juntamente com o SGBD *MySQL Workbench* e o *framework* Hibernate.

Fundamentação Teórica

Conforme Oliveira e Mendes (2008), durante a infância a alimentação da criança precisa ser planejada de tal forma que não haja risco de faltar qualquer nutriente, já que a alimentação adequada é essencial para garantir o crescimento e o desenvolvimento da criança. O planejamento de cardápios tem por objetivo elaborar refeições que atendam requisitos como hábitos alimentares, características nutricionais dos alunos, qualidade higiênico-sanitária e estejam adequados aos mercados de abastecimento e à capacidade de produção local. Nesse contexto, conforme Abreu et. al. (2011), o nutricionista é o profissional mais capacitado para esse fim.

Segundo Novelletto (2005), no cotidiano de um nutricionista o planejamento de cardápios é um trabalho constante. Logo, a criação de um *software* capaz de auxiliar o profissional na elaboração de cardápios torna essa tarefa mais rápida e mais confiável. Com a utilização desse *software*, o nutricionista poderá elaborar cardápios com maior precisão e qualidade, levando em consideração as recomendações nutricionais adequadas para cada faixa etária e de cada instituição de ensino. Ainda, o sistema possibilita a geração de relatórios que poderão ser encaminhados às instituições de ensino.

O desenvolvimento do sistema em plataforma web justifica-se por sua utilização não depender de investimentos em infra-estrutura (*hardware* ou *software* específico), bastando apenas um navegador instalado, podendo ser acessado de qualquer lugar que possua conexão com a internet e reduzindo tempo de manutenção, já que atualizações e reparos serão vistos e sentidos por todos os usuários.

A tecnologia WEB empregada para o desenvolvimento do software foi o *JavaServer Faces* (JSF). O JSF é, conforme o site oficial da Oracle (2012), "um framework de componentes de interface de usuário do lado servidor para aplicações web baseadas em tecnologia Java", ou seja, permite a elaboração de interfaces de usuário em formulários, ligando-os a objetos Java e fazendo a separação entre lógica e regras de negócio. Esta ferramenta contém uma *Application Programming Interface* (API) para representar componentes de interface, manipulação de eventos, validação e conversão de dados e definição de navegação das páginas. Essa linguagem foi desenvolvida pela *Sun Microsystem* e utilizada para desenvolver aplicativos para diversos ambientes e dispositivos. Segundo o site oficial do Java (2013), é uma tecnologia que capacita programas da mais alta qualidade, além de se relacionar perfeitamente com banco de dados MySQL, sendo aceitável por qualquer sistema operacional, reduzindo significativamente o custo e o tempo de desenvolvimento.

A linguagem Java segue o paradigma de programação orientada a objetos. Segundo Bezerra (2007), nesse paradigma, objetos são "coisas" do mundo real, podendo defini-lo como uma instância de uma classe. Para ele, uma classe representa uma abstração das características mais importantes do mundo real, sendo assim um modelo para a criação de um objeto ou de um grupo de objetos que contém atributos e

serviços em comum. Conforme Deitel (2005, p.21), “com a tecnologia de objeto, os programadores podem construir grande parte do software que será necessária combinando partes intercambiáveis e padronizadas chamadas de classes”.

A ferramenta Netbeans IDE foi empregada para o desenvolvimento do sistema. Segundo o site oficial do Netbeans (2012), ela é um ambiente de desenvolvimento de código aberto, multiplataforma e escrito totalmente em Java, capaz de escrever, compilar, depurar e executar programas, fornecendo uma base concreta para a criação de projetos e módulos, contendo várias bibliotecas, módulos e API’s, oferecendo recursos como editor de código, visualizador de classes integrado e *plugins* para *Unified Modeling Language* (UML), permitindo, que o desenvolvedor crie fácil e rapidamente aplicações Java para *desktop*, aparelhos móveis e web, oferecendo suporte para as novas tecnologias Java e banco de dados MySQL. Essa integração e suporte para o banco de dados MySQL é importante, já que o mesmo foi selecionado como o gerenciador de banco de dados a ser empregado no desenvolvimento do *software* apresentado.

O MySQL é um servidor de banco de dados *Structured Query Language* (SQL), composto, conforme Oliveira (2002), por um conjunto de comandos para a criação e manipulação desse banco, além da possibilidade de inclusão, alteração, atualização e pesquisa dos dados contidos em suas tabelas, onde a tabela, por sua vez, pode ser compreendida como um conjunto de linhas e colunas, colunas estas que qualificam cada linha com informações que são relacionadas ao objeto, tornando possível armazenar dados em uma ou várias tabelas.

A escolha do banco de dados MySQL deve-se ao fato de que ele é, conforme o site oficial do SEER (2013), um servidor multiusuário e multithreaded, ou seja, permite acesso concorrente de usuários, oferecendo velocidade, robustez e facilidade de uso, rodando em praticamente todos os sistemas operacionais, inclusive Linux. Segundo Ferrari (2007) sua finalidade é acessar dados, independentemente do tipo de hardware ou de software que se está utilizando.

Para o gerenciamento do banco de dados foi empregada a ferramenta *MySQL Workbench*. Conforme o site oficial do MySQL (2012), ele é uma ferramenta visual unificada para arquitetos de banco de dados, desenvolvedores e *Database Administrators* (DBA), oferecendo modelagem de banco de dados, desenvolvimento de SQL e ferramentas de administração abrangentes para a configuração do servidor e administração do usuário, sendo disponível em Windows, Linux e MAC OS, permitindo que o DBA, desenvolvedor ou arquiteto de banco de dados projete, modele, gere e gerencie o banco de dados visualmente, incluindo várias ferramentas que o desenvolvedor precisa para a criação de modelos Entidade Relacionamento (ER) complexos e engenharia reversa, além de ferramentas visuais que facilitam a criação, execução e otimização de consultas SQL.

No quesito persistência foi empregado nesse desenvolvimento o *Java Database Connectivity* (JDBC). O JDBC é, de acordo com o site oficial da Oracle (2012), uma API escrita em linguagem Java para execução de instruções e manipulação de dados SQL para qualquer tipo de banco de dados relacional, onde cada tipo de banco de dados possui seu *driver* JDBC, disponível em qualquer plataforma Java, sendo dispensável de instalação. Ainda Oracle (2012) diz que ele permite ao programador criar uma conexão com qualquer base de dados tabular, enviar instruções SQL e manipular dados, sendo fácil de implementar e com um custo de manutenção baixo, sendo ideal para

desenvolvimento de aplicações corporativas, já que esconde a complexidade de muitas tarefas de acesso aos dados, fazendo uma parte do "trabalho pesado", tornando o desenvolvimento de uma aplicação fácil e econômica.

Visando ainda minimizar o tempo de desenvolvimento e de recuperação de dados, foi utilizado o Hibernate, que é

> um Objeto/Relacional de Mapeamento de ferramentas nos meios Java. O termo Objeto/Relacional de Mapeamento (ORM) refere-se à técnica de mapeamento de dados, representada desde o objeto modelo aos dados relacionais modelo com um esquema baseado na SQL. (KING, 2011, p.11)

Ainda segundo King (2011), o Hibernate possibilita a criação de classes persistentes utilizando o Java convencional, classes estas que são definidas em documentos de mapeamento que são compilados na inicialização do programa, proporcionando facilidades na consulta e recuperação dos dados, aliviando o desenvolvedor em até 95% o tempo de programação relacionado à persistência.

Para a geração de relatórios foi empregado o iReport. De acordo com a Jaspersoft Corporation (2011), o iReport é um programa *Open Source* multiplataforma capaz de criar visualmente relatórios para aplicações Java. Segundo ela, o iReport compila um arquivo JRXML, arquivo esse que contém todas as informações básicas sobre o *layout* do relatório, e o salva como um arquivo executável binário que será utilizado para a geração do relatório, integrando o *template* do relatório com as informações obtidas da base de dados. Ainda o iReport permite ao desenvolvedor desenhar relatórios, economizando tempo de desenvolvimento e gerando relatórios em XML, PDF, HTML, DOCX, ODT e outros formatos.

Para se obter uma melhor perspectiva do sistema, foi empregado o uso da *Unified Modeling Language* (UML). Conforme Tacla (2012), a UML é uma linguagem composta de elementos gráficos que permite a modelagem de sistemas seguindo o paradigma da orientação a objetos.

Dentro do conceito de UML uma das representações existentes é o diagrama de caso de uso (DCU). Ainda Tacla (2013, p. 16) afirma que o "diagrama de casos de uso é uma ferramenta de comunicação entre clientes, usuários e desenvolvedores para discutirem e definirem as funcionalidades que devem ser realizadas no sistema". Dentro disso, os elementos que compõe o diagrama de casos de uso são, de acordo com Bezerra (2007), os casos de uso (que especificam uma sequência completa de interações entre o sistema e os agentes externos a ele), os atores (que são quaisquer elementos externos ao sistema e que irão interagir com o mesmo) e os relacionamentos existentes entre eles. Ainda para Bezerra (2007), este diagrama corresponde a uma visão externa de alto nível do sistema, representando graficamente os atores, casos de uso e os relacionamentos existentes entre esses elementos, devendo possuir uma linguagem simples e entendível tanto pelo usuário como pelo desenvolvedor.

Em relação ao diagrama de classes, Guedes (2011) cita que esse diagrama define as classes utilizadas no sistema, com seus respectivos atributos e métodos, bem como as relações existentes entre elas e as informações trocadas entre si. Dentro disso, cada classe é representada por um retângulo contendo, no máximo, três compartimentos, sendo que o primeiro compartimento refere-se ao nome da classe, o segundo aos atributos da classe e o terceiro aos seus métodos (operações).

As associações presentes no diagrama de classes representam a existência de relacionamentos entre objetos. Bezerra (2007) afirma que esse elemento, representado no diagrama de classes por uma linha ligando as classes, representa os relacionamentos criados entre os objetos durante a execução do programa.

A UML também oferece o diagrama de sequência. O diagrama de sequência mostra como ocorre o fluxo de eventos por meio dos casos de uso na ordem temporal em que acontecem. Para Lima (2012), esse diagrama mostra como devem ser executadas as ações dentro do sistema, evidenciando quais objetos devem ser criados para implementar as funcionalidades descritas nos casos de uso, bem como definir as classes que devem existir no diagrama de classes, apresentando as mensagens trocadas pelos objetos entre as diferentes camadas da aplicação. Ainda Lima (2012) afirma que cada elemento desse diagrama possui uma representação gráfica, os quais são: as linhas de vida, compostas por duas partes: a cabeça e a cauda, onde a cabeça contém a identificação do objeto, e a cauda, representada por uma linha tracejada, da qual partem as mensagens para o interior do sistema; as mensagens, mostradas no diagrama por uma seta ligando uma linha de vida a outra, representam as mensagens trocadas pelos objetos e o tipo de mensagem correspondente; ocorrências de execução, representadas graficamente como blocos retangulares sobre a linha de vida do objeto, onde o topo do retângulo coincide com o recebimento de uma mensagem, e parte de baixo corresponde ao término de uma operação realizada pelo objeto.

No desenvolvimento do *software* também foi adotado o padrão *Model View Controller* (MVC). Segundo Lamim (2009), MVC é um padrão de arquitetura de *software*, onde é introduzido o componente *Controller* entre o *Model* e o *View*, fazendo a separação das tarefas de acesso aos dados e lógica de negócio, lógica de apresentação e de interação com o usuário. O sistema é dividido em camadas e então aplica-se o MVC, que diz como os componentes da aplicação se interagem, baseando-se no princípio que o *Controller* envia solicitações ao *Model* e a *View* observa o *Model*.

Para Tacla (2012), o *Model* representa um domínio específico da informação em que o sistema opera e diz como os dados devem ser acessados e modificados consistentemente, o *View* faz a interação com usuário, apresentando as diversas visões de dados do negócio, não se preocupando como eles foram obtidos e o *Controller* faz a intermediação das duas camadas, controlando o fluxo de apresentação das informações.

Buscando-se a qualidade no desenvolvimento do *software* apresentado neste trabalho, optou-se pelo ciclo de desenvolvimento em espiral. O modelo espiral é, segundo Pressman (2011), um modelo de processo de software evolucionário que envolve aspectos do modelo de prototipação e do modelo em cascata, possibilitando o desenvolvimento de versões do *software* cada vez mais completas. O modelo espiral

> [...] é um gerador de *modelos de processos* dirigidos a riscos, e é utilizado para guiar a engenharia de sistemas intensivos de software, que ocorrem de forma concorrente e tem múltiplos envolvidos. Possui duas características principais que o distinguem. A primeira consiste em uma abordagem *cíclica* voltada para ampliar, incrementalmente, o grau de definição e a implementação de um sistema, enquanto diminui o grau de risco do mesmo. A segunda, característica consiste em uma série de pontos âncora de controle para assegurar o comprometimento de interessados quanto à busca de soluções para um sistema que seja mutuamente satisfatórias e praticáveis. (BOEHM, 1988, *apud* PRESSMAN, 2011, p. 65)

Segundo Pressman (2011), empregando-se o modelo espiral, o *software* é desenvolvido em uma série de versões evolutivas, onde a primeira versão pode ser entendida como um modelo, sendo produzidas versões mais completas a cada iteração posterior. Esse modelo utiliza a prototipagem como estratégia para a redução de riscos, tornando possível a aplicação de prototipação em qualquer estágio da evolução do *software*, mantendo uma abordagem em etapas característica do modelo cascata, reduzindo riscos antes de se tornarem problemas potenciais.

A usabilidade também foi considerada no desenvolvimento do *software* apresentado. Cybis (2003), afirma que a usabilidade é a capacidade que um sistema oferece ao usuário de realizar tarefas de maneira eficaz, eficiente e agradável, pois, segundo ele, o desenvolvimento de sistemas com boa usabilidade causa impacto na eficiência, eficácia e na produtividade da empresa, pois fará com que o usuário atinja seus objetivos com menos esforço e mais satisfação.

Bastien e Scapin (1993, *apud* Cybis, 2003) desenvolveram oito critérios ergonômicos que propõem boas maneiras de se criar uma interface com usabilidade. São elas: condução, carga de trabalho, controle explícito, adaptabilidade, gestão de erros, homogeneidade, significado dos códigos e denominações e compatibilidade. Ainda Bastien e Scapin (1993, *apud* Cybis, 2003), citam que avaliações de usabilidade de uma dada interface mostraram que seus critérios proporcionam um aumento da satisfação e do desempenho do usuário com relação a tal interface. Com base nisso, o *software* desenvolvido buscou considerar alguns critérios mencionados. São eles: homogeneidade, compatibilidade e controle explícito.

Na seção etapas do desenvolvimento do trabalho os passos realizados serão descritos em detalhes.

Etapas do Desenvolvimento do Trabalho

Seguindo o ciclo de vida em espiral, o desenvolvimento do *software* iniciou-se com a fase de planejamento. Nessa fase, os requisitos foram levantados através de diálogos com o usuário, observando seu trabalho no dia-a-dia. Os requisitos levantados foram: cadastro de instituições de ensino, com seus dados básicos, como nome, endereço, diretor, telefone e per capita dos alimentos recomendadas para a instituição, bem como as categorias de ensino e as quantidades de alunos pertencentes a ela; cadastro de alimentos com suas informações, como descrição, unidade de medida e categoria, bem como a descrição e quantidade de seus nutrientes; possibilidade de elaboração de cardápios que levassem em consideração a quantidade de alunos de cada instituição e a recomendação nutricional adequada; e a geração de relatórios das instituições cadastradas e dos cardápios elaborados. Assim, após o levantamento de requisitos, foram determinados os prazos e restrições.

A partir disso, com a utilização da ferramenta Astah, foram elaborados os diagramas de casos de uso, o diagrama de classes e o diagrama de sequência. Os diagramas propostos pela UML encontram-se nos apêndices A, B, C, D e E. Em seguida, o diagrama de classes foi analisado e iniciada a modelagem do banco de dados com a utilização da ferramenta *MySQL Workbench*. A modelagem lógica do banco de dados encontra-se no apêndice F.

Após isso, na fase de análise de riscos, foi feito um levantamento para identificar os possíveis problemas que poderiam atingir o *software.* Nessa etapa foram escolhidas

alternativas e caminhos que trariam maiores chances de sucesso, dentro dos prazos e custos estabelecidos na etapa anterior. São elas: a escolha da plataforma WEB para desenvolvimento, eliminando o risco de incompatibilidade de *hardware* do usuário; e a escolha de tecnologias gratuitas, eliminando o risco de haverem custos de desenvolvimento.

Na próxima fase iniciou-se o desenvolvimento do *software*, seguindo o padrão MVC. Partiu-se, então, para a criação do banco de dados, o qual foi feito com o Hibernate, utilizando as *Annotations*. As *Annotations* são marcações feitas nas classes Java pertencentes ao pacote *Model*, para que o Hibernate identifique quais atributos serão persistidos no banco de dados. As tabelas criadas com as *Annotations* foram: tbinstituicao, tbalimento, tbnutriente, tbcategoriaensino, tbnutrientesbase, tbalimentoscardapio, tbalimentosinstituicao e tbcardapio.

Depois de criadas as classes de persistência do pacote *Model*, iniciou-se a criação dos *ManagedBeans*. Os *ManagedBeans* são classes que recebem informações das páginas XHTML, processam as informações e retornam ao usuário. Essas classes pertencem ao pacote *Controller*, pois fazem a intermediação entre as solicitações do usuário e a base de dados.

Feito isso, iniciou-se o desenvolvimento das telas do sistema. Estas telas encontram-se no pacote *View*. Foi desenvolvido, primeiramente, um *template* para as telas do sistema. O *template* é um arquivo XHTML de modelo padrão, modelo esse composto por cabeçalho, aba esquerda, centro, menu e rodapé. Esse padrão é seguido por todas as telas do sistema, visando atingir uma homogeneidade no sistema.

Depois de desenvolvido o *template*, partiu-se para a elaboração das telas do sistema, as quais foram desenvolvidas com o uso do Netbeans e integrando o *PrimeFaces*, que é um *framework* de componentes visuais para aplicações WEB.

As telas do sistema foram desenvolvidas seguindo o critério de homogeneidade e compatibilidade, mantendo uma estabilidade entre os elementos das telas e fornecendo informações de fácil compreensão. As telas também fornecem *feedback* ao usuário sobre as ações feitas por ele. Na figura 1 pode-se visualizar o padrão seguido pelas telas.

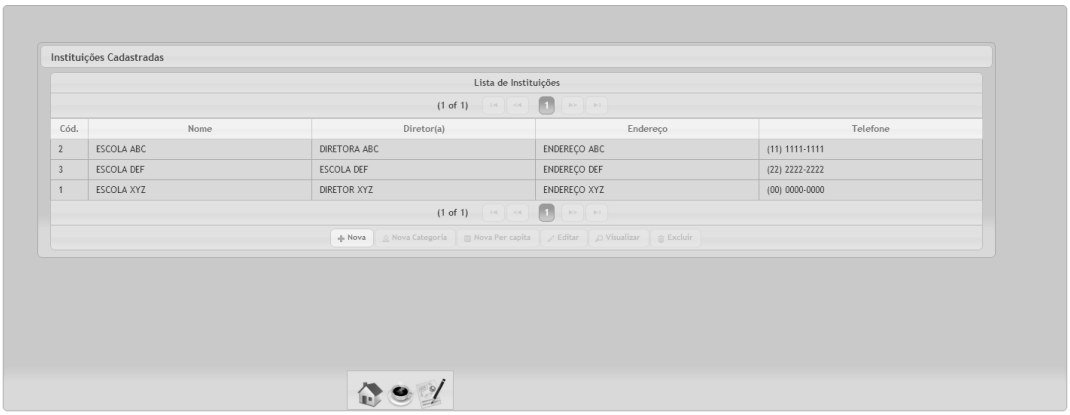


Figura 1. Tela de instituições. Em todas as telas do sistema segue-se o mesmo padrão e formato, respeitando o conceito de homogeneidade.

Após o desenvolvimento do sistema, foram feitos testes e validações do *software*. Foram realizados testes de caixa preta, fornecendo dados de entrada e comparando as saídas do sistema com resultados previamente obtidos. Também foi realizado o teste de aceitação com o usuário para verificar se o sistema atendia às suas necessidades. O processo de testes e validações foi registrado através de documentos onde o usuário respondia se o *software* atendeu ou não às suas necessidades. Tais documentos encontram-se nos apêndices G, H e I.

O sistema foi então implantado e colocado à disposição do usuário para que o mesmo o utilizasse. A partir da resposta do usuário é que se verifica se a versão do *software* satisfez ou está satisfazendo as necessidades requeridas para que se possa partir para a próxima versão e iniciar um novo ciclo. As versões do *software* desenvolvidas foram todas acompanhadas pelo usuário e registradas em fichas, as quais possuem pareceres e observações do usuário de cada versão desenvolvida. Essas fichas encontram-se nos apêndices J, L e M.

Resultados

Com a utilização, pelo nutricionista responsável pela elaboração dos cardápios da alimentação escolar, do *software* desenvolvido, observou-se uma redução considerável no tempo gasto com a elaboração de tais cardápios, com os cálculos dos nutrientes e com a geração dos relatórios.

O *software* também trouxe contribuições no que diz respeito à transparência do processo de elaboração dos cardápios, uma vez que os relatórios destes cardápios podem ser enviados às instituições de ensino e colocados em local visível aos alunos, pais e professores.

Outro benefício obtido com a utilização do *software* é que, analisando os relatórios gerados pelo sistema, o nutricionista pode observar a quantidade de cada alimento enviado às instituições de ensino e, assim, estimar a quantidade necessária desses alimentos para um próximo período de tempo, garantindo assim a suficiência da alimentação nas instituições.

Utilizando os diagramas da UML, foi possível ter uma melhor perspectiva do sistema antes da programação em si. Estes diagramas foram muito importantes para se definir a estrutura e as funcionalidades do sistema, além de deixar claro ao usuário quais seriam tais funcionalidades. Com isso obteve-se, também, uma redução no tempo gasto com a modelagem lógica do sistema.

A modelagem lógica também se fez importante no desenvolvimento deste trabalho. Com ela, foi possível definir quais seriam as classes persistentes do sistema e a melhor forma de relacionamento dos dados para acompanhar as regras de negócio.

O uso do Hibernate trouxe facilidades ao desenvolvimento do trabalho. Uma dessas facilidades é a redução da complexidade das operações de inserção, busca, alteração e exclusão de dados, o que acabou reduzindo o tempo de desenvolvimento das classes responsáveis pela manipulação de dados. Com isso, pôde-se destinar mais tempo ao estudo das regras de negócio.

Com a utilização da linguagem de programação Java, seguida do paradigma da orientação a objetos, obteve-se uma redução considerável no tempo de desenvolvimento

do sistema, uma vez que há reaproveitamento de código já escrito e a possibilidade de utilização de métodos prontos.

A tecnologia *JavaServer Faces* trouxe vantagens como rapidez do sistema, independência de plataforma do usuário e interface atraente, além de facilidade de manutenção. Isso faz com que o sistema possa ser acessado de qualquer lugar e qualquer computador.

A realização deste trabalho proporcionou um vasto ganho de conhecimento. A utilização da linguagem Java exigiu o estudo e o conhecimento dos métodos *toString()*, *toObject()* e *hashCode()*, utilizados para conversões de objetos no sistema.

A utilização dos *frameworks JavaServer Faces* e *Primefaces* demandaram muito tempo de estudo e pesquisa, já que, até então, não eram tecnologias familiares e requeriam boa compreensão da sua estrutura e do funcionamento dos seus componentes. O estudo da estrutura do JSF contribuiu, também, para um melhor entendimento do padrão MVC.

Considerações Finais

A elaboração de cardápios adequados é uma tarefa complexa, pois demanda tempo, no que diz respeito aos cálculos necessários com nutrientes e quantidades de alimentos. O *software* apresentado nesse trabalho foi desenvolvido para auxiliar o nutricionista responsável pela elaboração dos cardápios da rede pública de ensino a desempenhar esta tarefa de forma rápida e precisa. Com isso, notaram-se benefícios trazidos pela utilização do sistema nesse processo, tais como redução do tempo de elaboração dos cardápios e precisão nos cálculos dos nutrientes.

Como trabalho futuro, será desenvolvida uma nova versão do *software* em que serão consideradas apenas per capitas de alimentos por instituição, que são as quantidades de alimento necessárias para cada faixa etária e para cada instituição específica, e não mais per capitas gerais, já que podem haver casos em que determinadas quantidades de alimentos podem ser insuficientes ou excessivas para certas instituições.

Referências

Abreu, E. S., Spinelli, M. G. N., Pinto, A. M. S. (2011) "Gestão de unidades de alimentação e nutrição: um modo de fazer". São Paulo - SP.

Bezerra, E. (2007) "Princípios de análise e projeto de sistemas com UML". Rio de Janeiro - RJ.

Cybis, W. A. (2003) "Engenharia da Usabilidade: uma abordagem ergonômica". Florianópolis - SC.

Deitel, H. M. & Deitel, P. J. (2005) "Java como programar". São Paulo - SP.

Ferrari, F. A. (2007) "Crie banco de dados em MySQL: desvende o poder desta poderosa ferramenta". São Paulo - SP.

Guedes, G. T. A. (2011) "UML 2.0: uma abordagem prática". São Paulo - SP.

IBM. (2011) "Introdução à programação em Java, parte 1: fundamentos da linguagem Java". <http://www.ibm.com/developerworks/br/java/tutorials/j-introtojava1/section2.html>. Último acesso: Maio, 2013.

Jaspersoft Corporation (2011) "iReport Ultimate Guide".

Java. (2013) "O que é Java e por que é necessária". <http://www.java.com/pt_BR/download/faq/whatis_java.xml>. Último acesso: Maio, 2013.

King, G. et al. (2011) "Documentação de referência do Hibernate".

Lamim, J. (2009) "MVC: o padrão de arquitetura de software". <http://www.oficinadanet.com.br/artigo/1687/mvc>. Último acesso: Novembro, 2012.

Lima, A. S. (2012) "UML 2.3: do requisito à solução". São Paulo - SP.

Macedo, A. (2012) "Relatórios em Java - JasperReports e iReport". <http://www.k19.com.br/artigos/relatorios-em-java-jasperreports-e-irepor>. Último acesso: Novembro, 2012.

MySQL. (2012) "MySQL Workbench 5.2". <https://www.mysql.com/products/workbench/>. Último acesso: Novembro, 2012.

Netbeans. (2012) "Netbeans IDE Features". <http://netbeans.org/features/index.html>. Último acesso: Outubro, 2012.

Novelletto, D. L. A. (2005) "A importância do planejamento de cardápios para as unidades de alimentação e nutrição considerando as condições de trabalho no processo produtivo: um estudo de caso". In: Revista de Nutrição. Campinas – SP.

Oliveira, C. H. P. (2002) "SQL: curso prático". São Paulo - SP.

Oliveira, J. F., Mendes, R. C. D. (2008) "Avaliação da qualidade nutricional do cardápio do Centro de Educação Infantil (CEI) do município de Douradina – MS". Douradina - MS.

Oracle. (2012) "JDBC Overview". <http://www.oracle.com/technetwork/java/overview-141217.html#5>. Último acesso: Novembro, 2012.

Pressman, R. S. (2011) "Engenharia de Software: uma abordagem profissional". Porto Alegre - RS.

Primefaces (2013) "Primefaces user's guide". < http://www.primefaces.org/documentation.html>. Último acesso: Abril, 2013.

SEER (2013) "O que é MySQL?". < http://seer.ibict.br/index.php?option=com_content&task=view&id=237&Itemid=74>. Último acesso: Junho, 2013.

Sociedade Brasileira de Diabetes (2013). "O nutricionista e seu papel na sociedade". <http://www.diabetes.org.br/perguntas-e-respostas/142>. Último acesso: Maio, 2013.

Tacla, C. A. (2012) "Design Patterns parte 3: padrão MVC". <http://www.dainf.ct.utfpr.edu.br/~tacla/DesignPatterns/0030-JavaDP-MVC.pdf>. Último acesso: Outubro, 2012.

Tacla, C. A. (2013) "Análise e projeto OO & UML 2.0". < http://sistemas.riopomba.ifsudestemg.edu.br/dcc/materiais/865043995_AN%C3%81LISE%20E%20PROJETO%20OO%20E%20UML.pdf>. Último acesso: Junho, 2013.

Apêndices

Apêndice A. Diagrama de casos de uso.

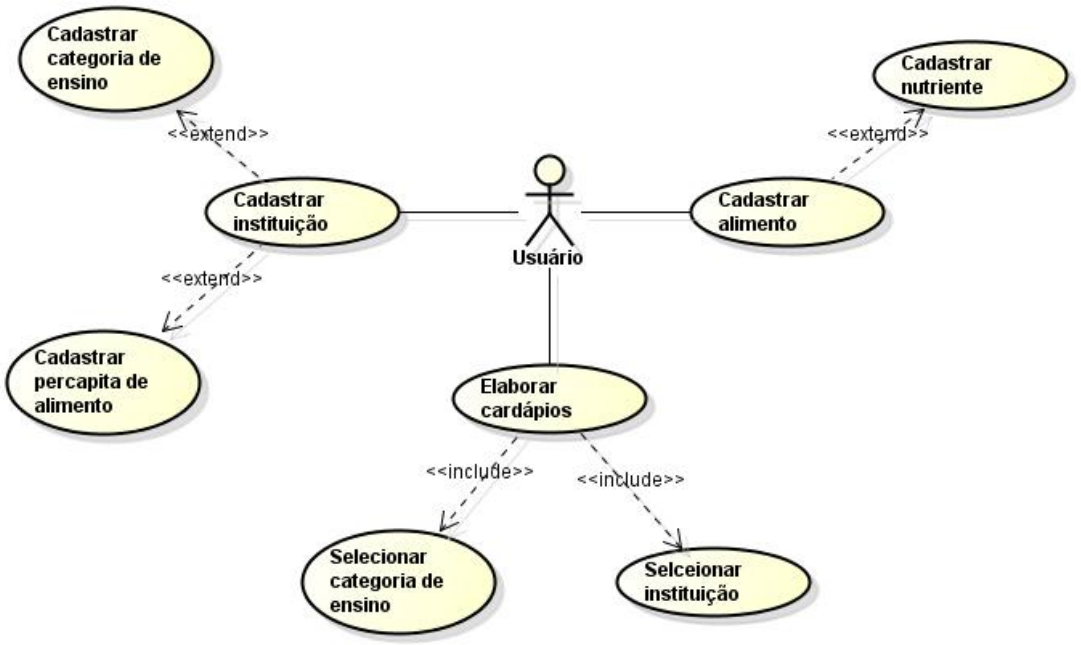


Diagrama de casos de uso

Apêndice B. Diagrama de classes.

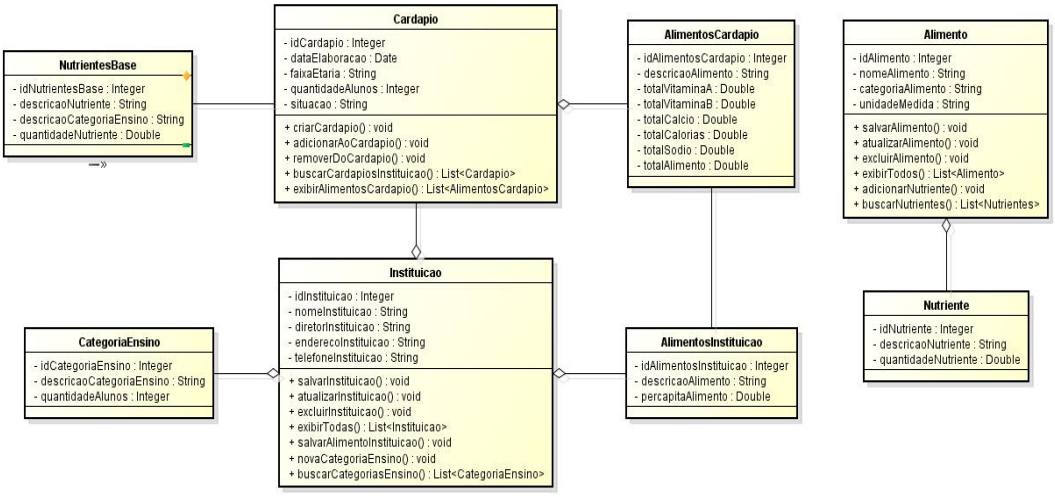


Diagrama de Classes

Apêndice C. Diagrama de sequência do caso de uso "Cadastrar instituição"

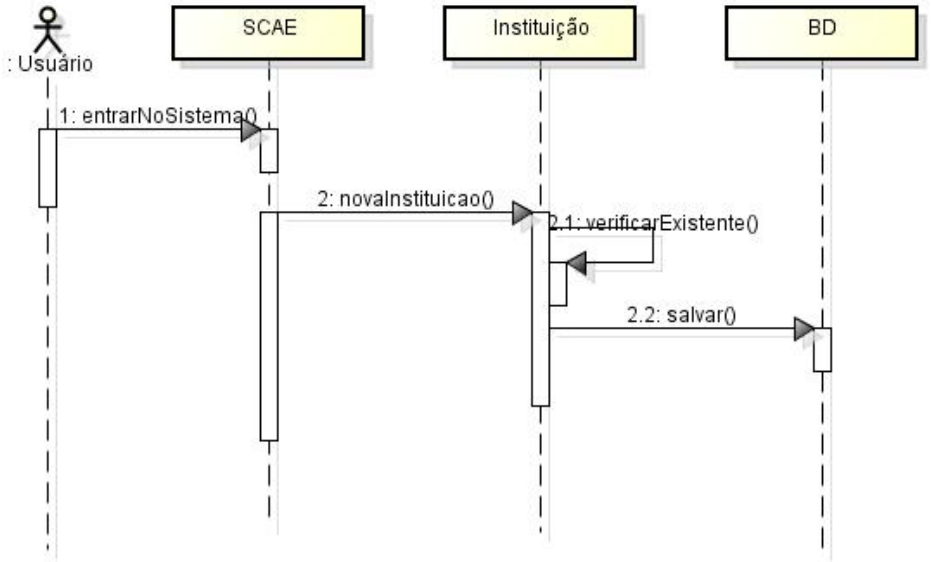


Diagrama de sequencia do caso de uso "Cadastrar instituição"

Apêndice D. Diagrama de sequência do caso de uso "Cadastrar alimento"

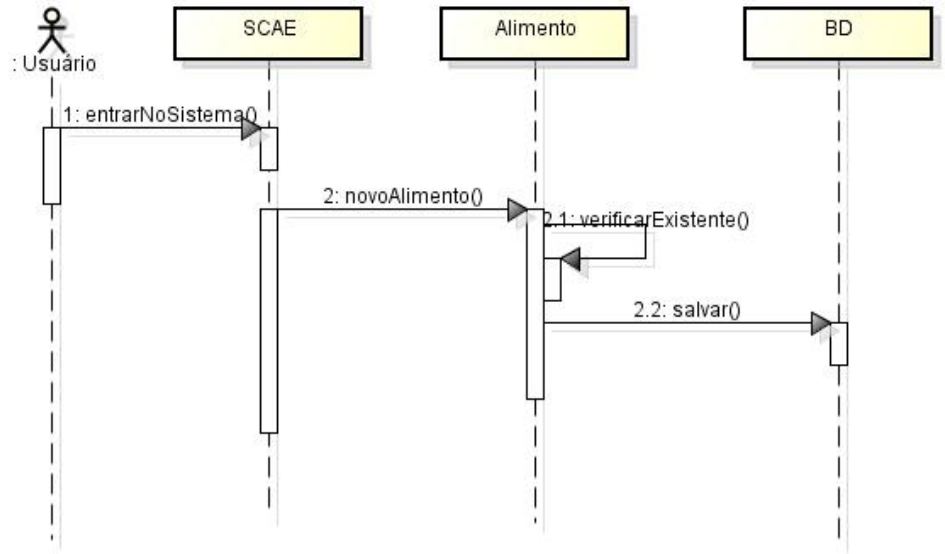


Diagrama de sequência do caso de uso "Cadastrar alimento"

Apêndice E. Diagrama de sequência do caso de uso "Elaborar cardápios"

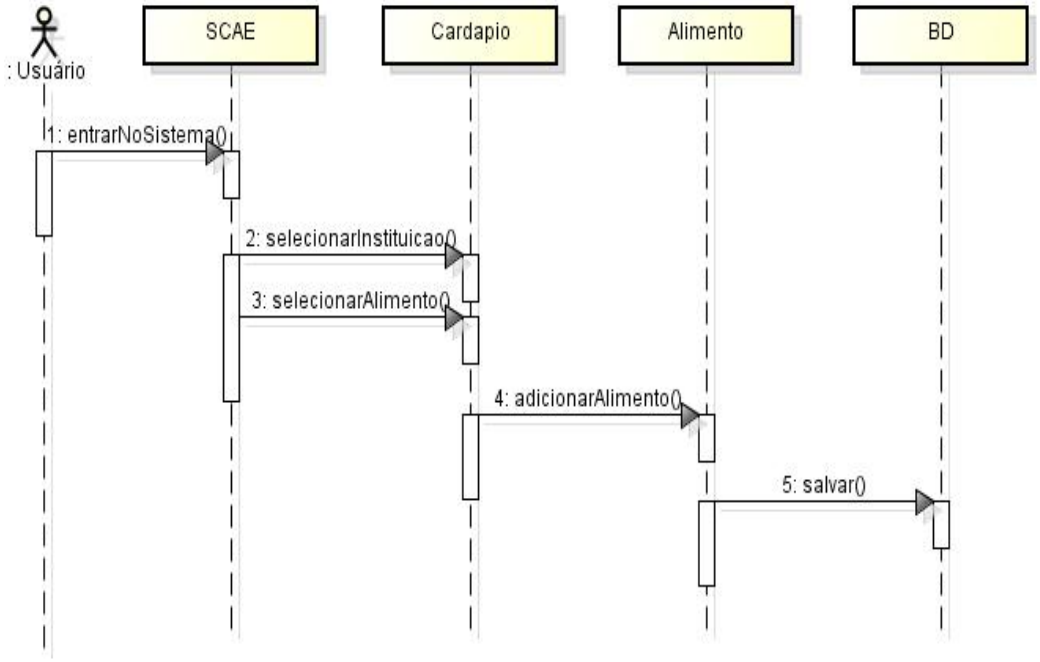
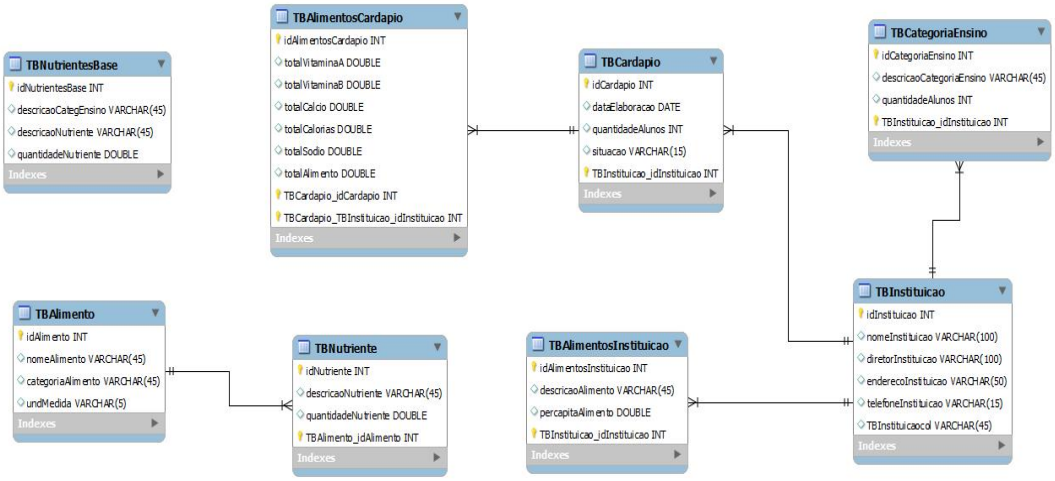


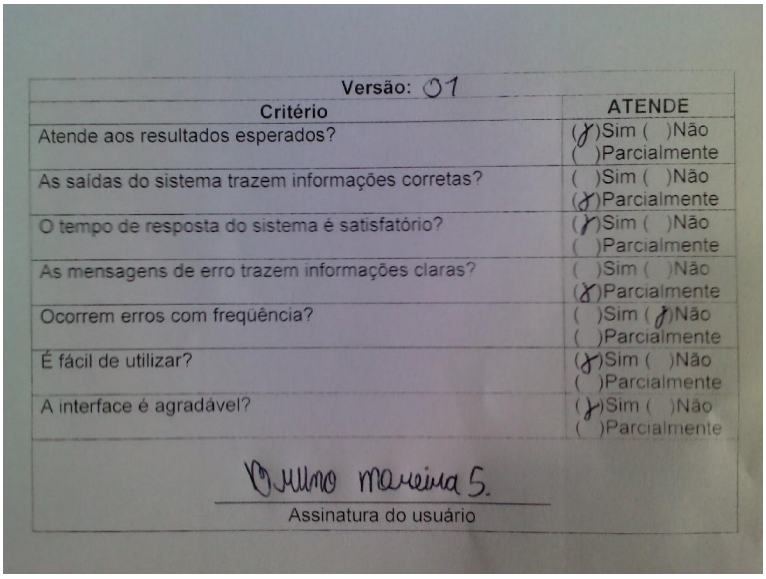
Diagrama de sequência do caso de uso "Elaborar cardápios"

Apêndice F. Modelagem lógica do sistema



Modelagem Lógica

Apêndice G. Ficha de teste da versão 1 do sistema.



Versão: 01

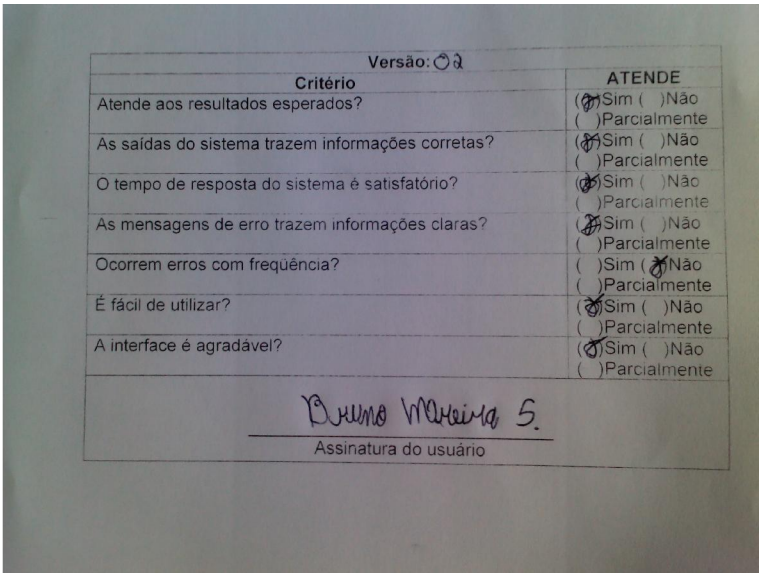
Critério	ATENDE
Atende aos resultados esperados?	(X)Sim ( )Não ( )Parcialmente
As saídas do sistema trazem informações corretas?	( )Sim ( )Não (X)Parcialmente
O tempo de resposta do sistema é satisfatório?	(X)Sim ( )Não ( )Parcialmente
As mensagens de erro trazem informações claras?	( )Sim ( )Não (X)Parcialmente
Ocorrem erros com freqüência?	( )Sim (X)Não ( )Parcialmente
É fácil de utilizar?	(X)Sim ( )Não ( )Parcialmente
A interface é agradável?	(X)Sim ( )Não ( )Parcialmente

Bruno Moreira S.

Assinatura do usuário

Ficha de teste da versão 1 do sistema, de 04/03/2013

Apêndice H. Ficha de teste da versão 2 do sistema.



Versão: 02

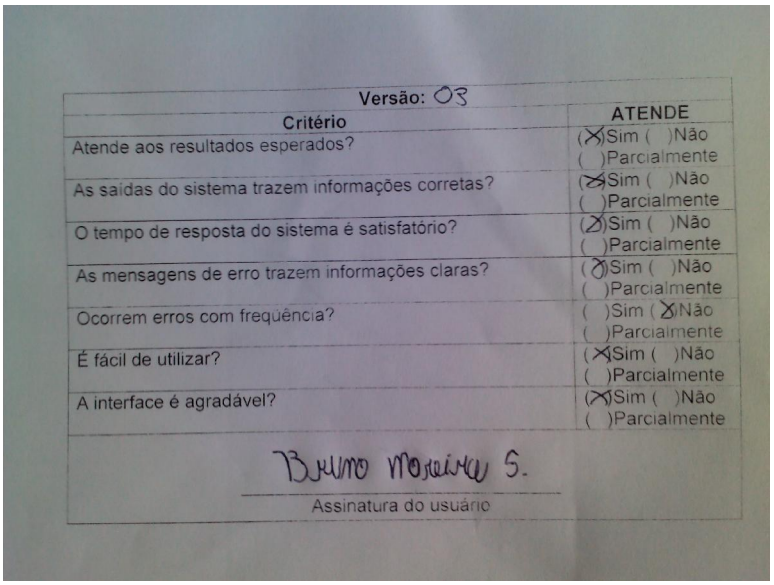
Critério	ATENDE
Atende aos resultados esperados?	(X)Sim ( )Não ( )Parcialmente
As saídas do sistema trazem informações corretas?	(X)Sim ( )Não ( )Parcialmente
O tempo de resposta do sistema é satisfatório?	(X)Sim ( )Não ( )Parcialmente
As mensagens de erro trazem informações claras?	(X)Sim ( )Não ( )Parcialmente
Ocorrem erros com freqüência?	( )Sim (X)Não ( )Parcialmente
É fácil de utilizar?	(X)Sim ( )Não ( )Parcialmente
A interface é agradável?	(X)Sim ( )Não ( )Parcialmente

Bruno Moreira S.

Assinatura do usuário

Ficha de teste da versão 2 do sistema, de 07/04/2013

Apêndice I. Ficha de teste da versão 3 do sistema.



Versão: 03

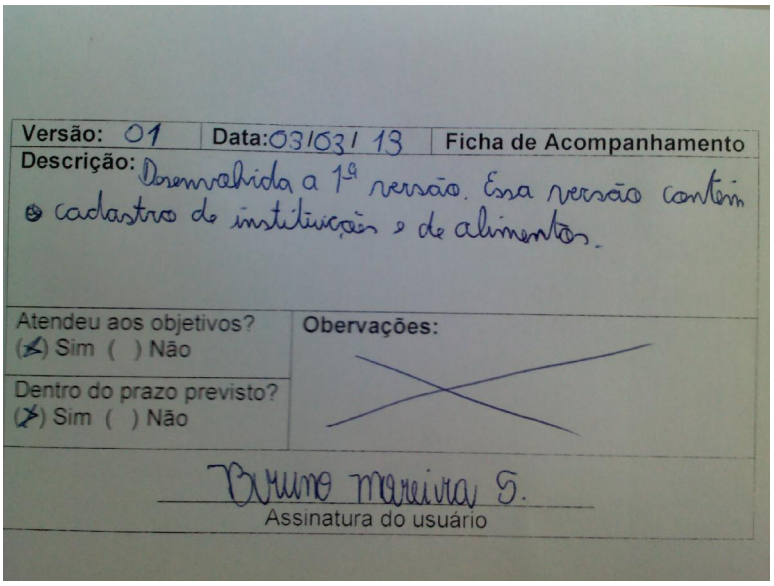
Critério	ATENDE
Atende aos resultados esperados?	(X)Sim ( )Não ( )Parcialmente
As saídas do sistema trazem informações corretas?	(X)Sim ( )Não ( )Parcialmente
O tempo de resposta do sistema é satisfatório?	(X)Sim ( )Não ( )Parcialmente
As mensagens de erro trazem informações claras?	(X)Sim ( )Não ( )Parcialmente
Ocorrem erros com frequência?	( )Sim (X)Não ( )Parcialmente
É fácil de utilizar?	(X)Sim ( )Não ( )Parcialmente
A interface é agradável?	(X)Sim ( )Não ( )Parcialmente

Bruno Moreira S.

Assinatura do usuário

Ficha de teste da versão 3 do sistema, de 23/05/2013

Apêndice J. Ficha de avaliação da versão 1 do sistema.



Versão: 01 | Data: 03/03/13 | Ficha de Acompanhamento

Descrição: Desenvolvida a 1ª versão. Essa versão contém o cadastro de instituições e de alimentos.

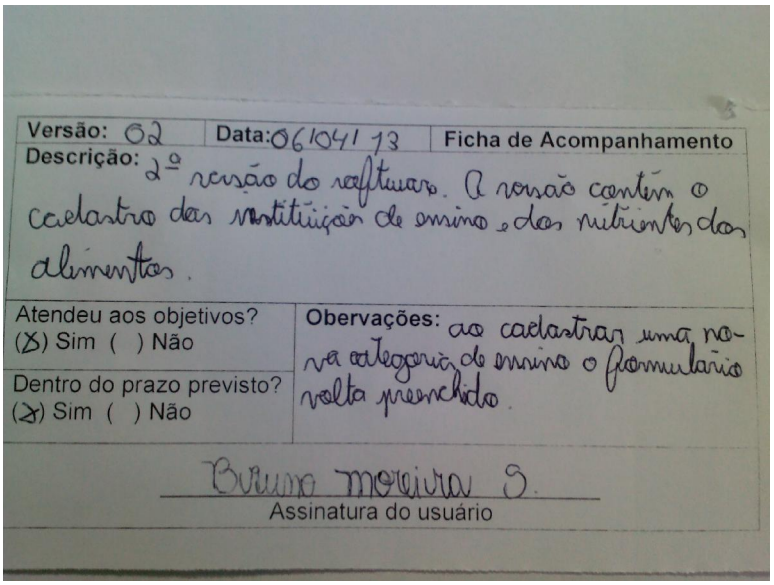
Atendeu aos objetivos?	Obervações:
(X) Sim ( ) Não	
Dentro do prazo previsto? (X) Sim ( ) Não	

Bruno Moreira S.

Assinatura do usuário

Ficha de avaliação da versão 1 do sistema

Apêndice L. Ficha de avaliação da versão 2 do sistema.



Versão: 02 | Data: 06/04/13 | Ficha de Acompanhamento

Descrição: 2ª versão do software. A versão contém o cadastro das instituições de ensino e dos nutrientes dos alimentos.

Atendeu aos objetivos? (X) Sim ( ) Não

Dentro do prazo previsto? (X) Sim ( ) Não

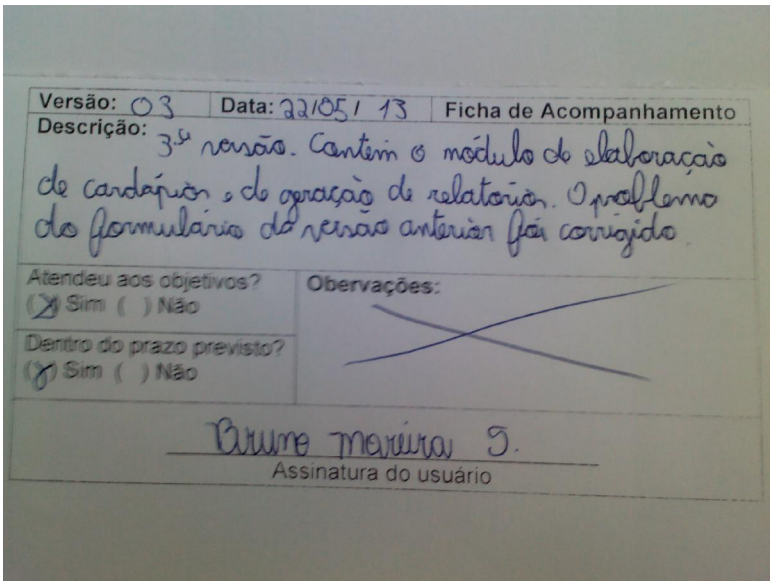
Obervações: ao cadastrar uma nova categoria de ensino o formulário volta preenchido.

Bruno Moreira S.

Assinatura do usuário

Ficha de avaliação da versão 2 do sistema

Apêndice M. Ficha de avaliação da versão 3 do sistema.



Versão: 03 | Data: 22/05/13 | Ficha de Acompanhamento

Descrição: 3ª versão. Contém o módulo de elaboração de cardápios e de geração de relatórios. O problema do formulário da versão anterior foi corrigido.

Atendeu aos objetivos? (X) Sim ( ) Não

Dentro do prazo previsto? (X) Sim ( ) Não

Obervações:

Bruno Moreira S.

Assinatura do usuário

Ficha de avaliação da versão 3 do sistema