

Desenvolvimento de Sistema Web para Sorveteria

Misael Alan Somer¹, Sergio Ribeiro², Regiane Orlovski³

^{1,2,3} Tecnologia em Análise e Desenvolvimento de Sistemas – Faculdade Guairacá
CEP 85010-000 – Guarapuava – PR – Brasil

¹misael_alan_somer@hotmail.com.br, ²professor@sergioribeiro.com.br,
³regianeorlovski@hotmail.com

Abstract. *This paper aims to describe and present the influence of those processes performed to create a Web System for Ice Cream Shop. For the development of the software was used programming language PHP object-oriented, with the aid of the tool NetBeans IDE and the management system database MySQL. The Lifecycle addressed to software development was cascaded with feedback that allows overlap between phases and feedback corrections, obtained as a result of a web as requested by owner.*

Resumo. Este trabalho tem como objetivo descrever e apresentar a influência nos processos realizados para a criação de um Sistema Web para Sorveteria. Para o desenvolvimento do software foi utilizada a linguagem de programação PHP orientada a objetos, com o auxílio da ferramenta *NetBeans IDE* e o sistema de gerenciamento de banco de dados *MySQL*. O ciclo de vida abordado ao desenvolvimento do software foi cascata com realimentação que permite superposição entre fases e a realimentação de correções, obteve como resultado um sistema Web conforme requisitados pelo proprietário.

Introdução

As empresas vêm investindo na tecnologia para facilitar nos processos administrativos para que ocorra a redução de custos e assim poder oferecer um produto/serviço de qualidade. O problema que as empresas encontram é achar um software que atenda todas as funções da mesma, elas até encontram softwares cheios de recursos que muitas vezes não necessitam de todas as suas funções prestadas pelo sistema e ainda assim para implementar ou manter o mesmo tem o custo muito alto.

Assim levando as empresas a buscarem softwares específicos com uma forma de saída, o que seria um software de uso simples onde atenda as funções que o cliente deseja, com isso deve se levar em conta todos esses fatores, onde busca se projetar um sistema que levem em conta as pessoas e a forma que interagem com o sistema.

Para o desenvolvimento de um sistema Web em PHP (*Hipertext Pre Processor*) Orientado a Objetos, onde atenderá as funções básicas de: cadastro de clientes, de produto, de venda, controle de estoque e relatórios, é necessário uma serie de ferramentas para auxílio no desenvolvimento do software como: XAMPP 1.8.1 como servidor Web, IDE (*Integrated Development Environment*) *NetBeans 7.2.1* para desenvolvimento da programação, *Mysql Workbench 5.2* para gerenciamento de banco de dados, o *DBDesigner 4* para a modelagem lógica do banco de dados e o *Astah Professional*, utilizado para construção dos diagramas de UML (*Unified Modeling*

Language - Linguagem de Modelagem Unificada), com o auxílio dessas ferramentas vai ser desenvolvido um sistema web para sorveteria.

Este trabalho tem por objetivo apresentar os processos e etapas realizadas para o desenvolvimento de um software Web em PHP orientado a objetos, que atenderá as seguintes funções: de realizar cadastro de clientes, produto, fornecedor, realização de venda e movimentação de estoque.

Fundamentação Teórica

O computador em nível das tecnologias de informação é a grande revolução das últimas décadas, pois vem permitindo uma comunicação global e o seu uso cada vez maior vem revolucionando a maneira de comunicar e informar. As novas descobertas científicas e tecnológicas estão hoje integradas no mundo doméstico, do trabalho, da saúde, da segurança, dos transportes, informação e comunicação. Atualmente todos os setores empresariais de grande e médio porte estão sendo transformados pelo seu impacto, mais ainda há setores de micro e pequeno porte que não sofreram grandes influencia pela tecnologia.

Nos dias atuais os computadores exercem um forte impacto sobre as empresas, pois as mesmas não conseguem sobreviver à concorrência do mercado sem a tecnologia, inovação de serviços prestados e produtos oferecidos. A tecnologia faz parte da evolução das empresas, desde a Revolução Industrial. Carvalho (1997) comenta que a partir da Revolução Industrial os conhecimentos tecnológicos e a estrutura social foram modificados de forma acelerada, porém foi a partir da segunda metade do século XX que a humanidade mais acumulou conhecimentos e mais acelerou o processo de transformações sociais na qual resultou em um profundo impacto no processo produtivo, atingindo tanto os níveis econômicos e sociais, como o contínuo processo de evolução e inovação tecnológica do século atual, ressaltando que sem tecnologia as empresas se tornam decadentes.

A tecnologia da informação (TI) ajuda as empresas em seus processos deixando-os mais ágeis, outra vantagem é o crescente fluxo de informação que tem se tornado um dos bens mais valiosos de uma empresa, e o acesso a essa informação mais fácil. A tecnologia da informação também deixa mais acessível um sistema de proteção e de informações mais eficiente.

Segundo Bezerra (2007) para o desenvolvimento de um sistema de informação é necessário realizar tarefas complexas. Já Dennis (2011) ressalta que para o desenvolvimento de um sistema é fundamental uma coleta de requisitos, onde os mesmos possam capacitar os recursos da empresa que o sistema de informações terá de incluir, como a habilidade de coletar dados ou receberem informações de estoque a medida que os pedidos ocorrem e as vendas forem feitas.

Os requisitos são as características básicas necessárias para definir os critérios de aceitação de um projeto, Paula Filho (2011) comenta que a engenharia de software tem por objetivo colocar nos produtos as características especiais que são os requisitos, outras características podem aparecer acidentalmente, mas os produtos não devem ser desenhados para incluí-las, já que, normalmente todas as características extras significam um custo adicional de desenho ou de fabricação.

Se uma vez identificadas e definidas as necessidades mais importantes da empresa na coleta de requisitos, e que possa identificar as solicitações e o valor agregado pelo sistema, Dennis (2011) ressalta que é nesta hora que deve iniciar formalmente o projeto, na maioria das vezes o início do projeto começa com a modelagem da UML.

A UML é uma linguagem visual para modelar sistemas orientados a objetos. Bezerra (2007) comenta que a UML é uma linguagem que define elementos gráficos que podem ser utilizados na modelagem de sistemas, esses elementos permitem representar os conceitos do paradigma da orientação a objetos, por meio dos elementos gráficos definidos nesta linguagem pode-se construir diagramas que representam diversas perspectivas de um sistema.

De acordo com Bezerra (2007), cada elemento gráfico da UML possui uma sintaxe e uma semântica, a sintaxe de um elemento corresponde à forma predeterminada de desenhar o elemento, e a semântica define o que significa o elemento e com que objetivo ele deve ser utilizado, além disso, conforme descrito, tanto a sintaxe quanto a semântica são elementos da UML.

O desenvolvimento de um sistema envolve diversas fases, como a coleta de requisitos, análise de requisitos, modelagem de UML, desenvolvimento, testes e implementação, o encadeamento específico dessas fases para a construção do sistema dá-se o nome de Modelo de Ciclo de Vida. O modelo de ciclo de vida em questão seria a cascata é também chamado de clássico ou linear, Bezerra (2007) o ciclo de vida cascata se caracteriza por possuir uma tendência na progressão sequencial entre uma fase e a seguinte, podendo haver uma retroalimentação de uma fase para a fase anterior.

Segundo Paula Filho (2011), na prática é sempre necessário permitir que, em fases posteriores, haja revisão e alteração de resultados das fases anteriores, uma das variantes que permite superposição entre fases e a realimentação de correções é um modelo mais realista, com realimentação entre as fases, chamado de Cascata com realimentação que pode ser melhor visualizado a seguir (Figura, 1).

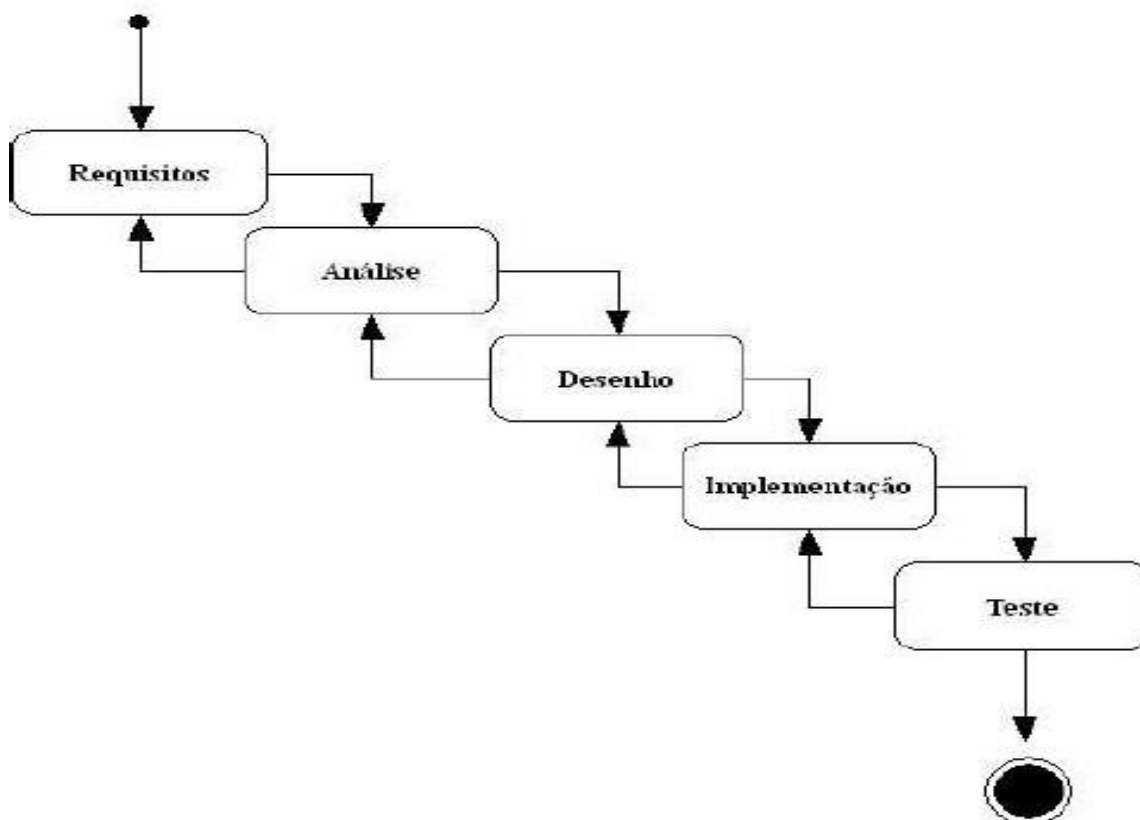


Figura 1. Modelo de ciclo de vida em Cascata com realimentação. Paula Filho (2011).

O ciclo de vida em cascata é adequado a projetos de pequena duração, tais como os projetos do *Personal Software Process* (PSP). De acordo com Paula Filho (2011) é conveniente também para miniprocessos, que são subprocessos bem-delimitados executados dentro de um processo maior, atividades como resolução de problemas, resolução de defeitos, alteração de requisitos, aquisições, manutenção e inovação são tratadas como miniprocessos com ciclo de vida em cascata.

Como referido anteriormente as empresas buscam softwares que atendam suas necessidades, sendo assim, este estudo propõem o entendimento da linguagem PHP orientada a objetos e o desenvolvimento de uma aplicação Web voltada ao controle das funções básicas de uma sorveteria.

De acordo com Dall'Oglio (2009), o *Hipertext Pre Processor* (PHP) é uma das linguagens Web mais utilizadas no mundo de hoje, pois sua facilidade em criar aplicações dinâmicas e funções flexíveis permite desde a criação de simples portais até complexas aplicações de negócio.

O desenvolvimento de um sistema que seja ágil, é primordial para oferecer aos seus clientes um atendimento mais eficiente, com isso dando melhor satisfação aos clientes. O PHP de acordo com Batista Neto (2010) atualmente situa-se dentro destes cenários de programação como uma das mais versáteis e poderosas ferramentas de desenvolvimento, facilitando a vida do desenvolvedor e reduzindo a quantidade de conhecimentos avançados que são necessários para a realização deste tipo de trabalho.

Segundo Batista Neto (2010), o PHP é uma linguagem de programação completa que herdou características de outras linguagens como C, Java e Perl, sendo que sua familiaridade com alguma destas linguagens torna-se o seu aprendizado extremamente fácil.

O PHP é uma linguagem que permite criar sites web dinâmicos, possibilitando uma interação com o usuário por meio de formulários, parâmetros da *Uniform Resource Locator* - Localizador-Padrão de Recursos (URL) e links. Xavier (2008), relata uma diferença entre o PHP e o JavaScript é que o código PHP é executado no servidor, sendo enviado para o cliente apenas o *HyperText Markup Language* -Linguagem de Marcação de Hipertexto (HTML) puro, com isso é possível interagir com banco de dados e aplicações existentes no servidor, sem a necessidade do cliente ter acesso direto do código fonte.

Para a linguagem de programação PHP conforme Dall'Oglio (2009) relata que com o uso da orientação a objetos juntamente com o emprego de boas práticas de programação, onde possibilita manter um ritmo sustentável no desenvolvimento de aplicações Web.

A orientação a objetos é um padrão que representa toda uma forma para construção de um sistema. Ao trabalhar com a orientação a objetos é fundamental entender o conceito de classes e objetos. Dall'Oglio (2009) refere-se que classe é uma estrutura que define um tipo de dados, podendo conter atributos (variáveis) e também funções (métodos) para manipular esses atributos. Já um objeto contém exatamente a mesma estrutura e as propriedades de uma classe, no entanto sua estrutura é dinâmica, seus atributos podem mudar de valor durante a execução do programa e declarar diversos objetos oriundos de uma mesma classe.

Segundo Xavier (2008), com a programação orientada a objetos, pode-se programar de uma maneira em que modela-se os processos de programação levando a chegar próximo a realidade, onde trata se cada componente de um programa com um objeto com suas características e funcionalidades.

Também para compor e auxiliar a aplicação será utilizado o HTML que conforme Costa (2007), é uma linguagem padrão utilizada para o acesso e exibição de páginas Web. As linhas de código HTML são interpretadas pelo browser que mostra o resultado final ao utilizador. Genericamente, a linguagem HTML é constituída de textos e de códigos especiais denominados marcas ou tags.

De acordo com Costa (2007), o HTML foi desenvolvido em 1992 por Tim Berners Lee e Robert Cailau no CERN, correspondendo a uma implementação de *Standard Generalized Markup Language* (SGML), originalmente o HTML definia estritamente a estrutura lógica de um documento, e não a sua aparência física, mas com a pressão dos utilizadores, as versões posteriores do HTML foram forçadas a fornecer cada vez mais controles da aparência do documento.

Tendo o HTML para mostrar as telas de um sistemas, é necessário ter uma Folha de Estilo em Cascata para deixar os mesmos com uma boa aparência, para Somera (2006) o *Cascading Style Sheet* (CSS) é uma ferramenta utilizada para a construção da aparência de páginas para Web, que permite o uso de uma técnica diferente da

convencional (HTML puro), possibilitando uma considerável redução no tempo de trabalho, se tornou uma necessidade para quem deseja ser um bom *Webdeveloper* (desenvolvedor de páginas para Internet) e para quem quiser criar qualquer projeto para Web.

Segundo Silva (2008), a finalidade do CSS é gerenciar todas as funções de apresentação de um documento, e esta é sua finalidade maior, entende-se como unidade básica a menor porção de código capaz de produzir um efeito de estilização.

O Bootstrap é uma biblioteca de CSS que trabalha exatamente o que necessita para o projeto, segundo Cochran (2012), o Twitter Bootstrap oferece uma série de estilos acessíveis para atender as necessidades de muitos, que ocorrem frequentemente em diversas circunstâncias, incluindo algumas opções de estilo que pode não ter considerado antes, mas que pode ser útil neste momento.

Também é necessário o uso de outras bibliotecas com jQuery, que de acordo com Silva (2010), é uma biblioteca JavaScript criada por John Resig e disponibilizada como software livre e aberto, ou seja, de emprego e uso regido segundo uma licença com regras estabelecidas pelo MIT (*Massachusetts Institute of Technology*) ou pelo GPL – GNU (*General Public License*). A biblioteca faz coisas como HTML, documento transversal e manipulação, seja de eventos, animação e Ajax muito mais simples com uma API fácil de usar, que funciona por meio de uma multidão de browsers. Com uma combinação de versatilidade e capacidade de expansão, jQuery mudou a maneira que milhões de pessoas escrevem JavaScript.

De acordo com Flanagan (2002), o JavaScript é uma linguagem de programação leve, interpretada e com recursos de orientação a objetos, essa versão de JavaScript do lado cliente permite que o conteúdo executável seja incluído e, nas páginas Web – significa que um Web Site não precisa mais de HTML estático, mas pode incluir programas que interagem com o usuário, controlar o método, o navegador e assim criar conteúdo HTML dinamicamente.

O JavaScript consegue controlar o comportamento e o conteúdo de um navegador, mas não consegue desenhar gráficos ou desenhar operações de rede, Flanagan (2002) comenta que o JavaScript do lado do cliente consegue interagir com *applets* Java embutidos em uma página Web e os controlar; nesse sentido, JavaScript pode realmente empregar Java em scripts.

Para trabalhar com a questão de desenhar e montar tabelas na comunicação do HTML com o JavaScript utiliza-se o AJAX, que de acordo com Moroni (2007), o objetivo do AJAX é evitar que a cada solicitação do servidor à página precise ser completamente recarregada. Para isso, utiliza o objeto XMLHttpRequest por meio do JavaScript para intermediar todas as solicitações ao servidor.

Tendo esclarecido e definido as questões de linguagem de programação PHP orientado a objetos e os meios que proporcionam uma interação com outras linguagens que iram proporcionar uma aplicação melhor definida a um sistema com maiores recursos em seus meios, então define o *design pattern* ou padrão de projeto, para uma

maior organização em suas classes e métodos abordados durante o desenvolvimento do software e diante disso podendo dar início a parte de programação do sistema.

Dall'Oglio (2009), fala que *Model View Controller (MVC)* é o *design pattern* mais conhecido de todos e utilizado para diferentes tipos de aplicação. Pois seus conceitos remontam a plataforma Smaltalk na década de 1970. Basicamente uma aplicação que segue o *pattern Model View Controller* é dividida em três camadas. As letras que compõem o nome deste *pattern* representam cada um desses aspectos.

Segundo Melo e Nascimento (2007), o MVC é o conceito que aborda a codificação de sistemas em camadas explicadas, e a figura 2 a seguir mostra isso. Vale ressaltar, entretanto, que este padrão não demonstra diretamente a estrutura de classes que precisa codificar e estruturar, o conceito do MVC é baseado no relacionamento do tripé *Model, View e Controller*. Sendo que tanto *Controller* como *View* depende de *Model*, já que ambas as classes requisitam informações desta mesma.

A separação da aplicação nesses três aspectos traz uma série de vantagens ao desenvolvedor, permitindo que reutilize um mesmo objeto de modelo em diversas visualizações diferentes. Dall'Oglio (2009) relata que as três camadas são distintas, porém interagem umas com as outras da maneira mais conhecida, é a interação entre *view* e *controller*. Um exemplo para compreender melhor o assunto seria: um sistema MVC clássico terá uma classe *Controller* para cada *View* existente, mas esta abordagem não é a única existente podendo haver do *view* para *model*, e *model* para *controller*.

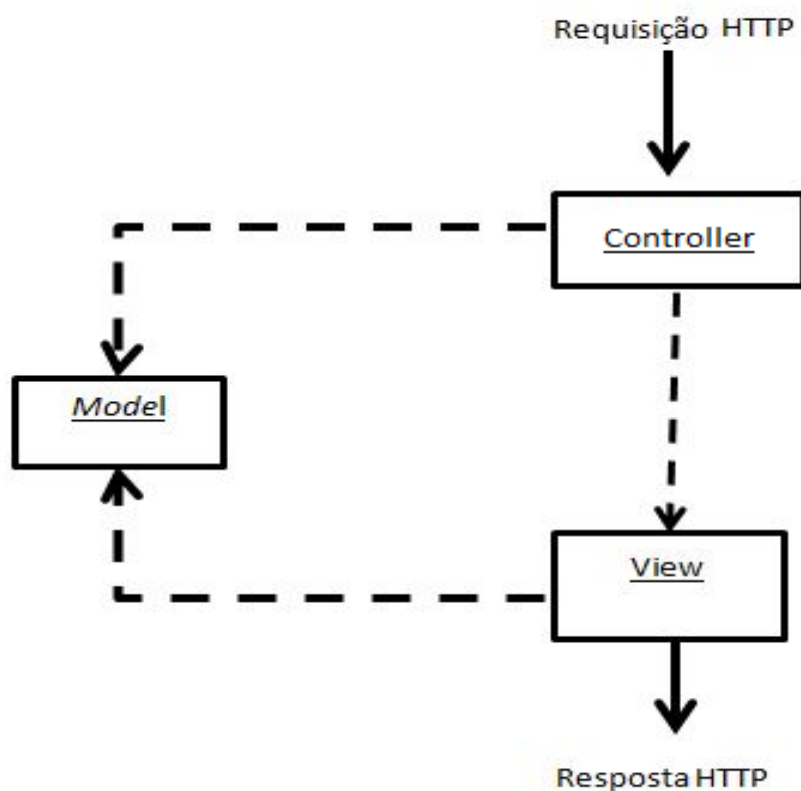


Figura 2. Modelo Clássico em 3 Camadas MVC. Melo e Nascimento (2007).

Por isso, Melo e Nascimento (2007) ressaltam que cada uma das classes do conceito *Model View Controller* tem uma finalidade: significa que um sistema que adota o conceito padrão de projeto pode ter várias classes que fazem o papel de *Model*, outras do *Controller* e pelo menos uma do *View*. Assim utilizando o MVC, se separa a sua aplicação em três distintos modos, onde podem ter mais que três, mais a principais camadas da aplicação são:

- A Modelo (*Model*) comunicação, fluxo de dados e lógica de negócio.
- A Visualização (*View*) trata dados para visualização do usuário.
- A Controle (*Control*) direciona o fluxo da aplicação fazendo a ponte entre a *View* e o *Model*.

Também não esquecendo, de uma parte tão importante quanto as partes descritas anteriores do sistema, precisa-se de um bom banco de dados, que é de suma importância para o software, pois é onde toda informação coletada e gerada pelo software fica armazenada permitindo a sua gravação com segurança, podendo retornar suas informações em si guardadas de uma forma clara e precisa.

Um banco de dados, é uma coleção de dados ou informações relacionadas entre si, Guimarães (2010), refere-se que dados representam aspectos do mundo real com significado próprio e que desejamos armazenar para uso futuro, ao se projetar um banco de dados tem-se em mente um conjunto de aplicações que primordialmente se deseja fazer sobre dados, que determinam o uso principal para elaborar o banco de dados.

Segundo Heuser (2009), um modelo conceitual é uma descrição do banco de dados de forma independente de implementação em um Sistema de Gerenciamento de Banco de Dados (SGBD). O modelo conceitual registra que dados podem aparecer no banco de dados, mas não registra como estes dados estão armazenados no nível de SGBD.

Structured Query Language (SQL) segundo Guimarães (2010), é uma linguagem de definição e de manipulação de dados relacionais, desenvolvida nos laboratórios da IBM nos anos 70. Atualmente Oliveira (2011), relata que SQL pode ser considerada um padrão para manipulação de dados em banco de dados.

De acordo com FERRARI (2007), SQL não serve para criar rotinas de procedimentos a serem executados pelo computador, mais serve para informar quais dados queremos manipular. De fato, a finalidade do SQL é acessar dados, independentemente do tipo de hardware ou software que esta sendo usado.

Etapas do Desenvolvimento do Projeto

Para que o desenvolvimento do Sistema Web em PHP Orientado a Objetos para sorveteria pudesse ter início, ocorreu primeiramente o levantamento de requisitos juntamente ao cliente, onde nesta etapa realizou-se uma entrevista com o proprietário da empresa, o mesmo solicitou as funcionalidades básicas do software conforme representado no apêndice de levantamento de requisitos “A”.

No processo de coleta foi definido juntamente com o proprietário os seguintes requisitos: tela de cadastro de cliente, tela de produto, categoria, fornecedores, movimentação de estoque e tela para operação de venda. Após a realização da coleta de requisitos concluída, houve a necessidade da escolha de outro fator que é determinante no desenvolvimento do software, a escolha do ciclo de vida do projeto. O ciclo de vida escolhido para o desenvolvimento do projeto do sistema web para sorveteria foi o cascata com realimentação.

A elaboração dos desenhos que compõem o ciclo de vida foi realizada logo após a análise de requisitos, gerando alguns diagramas conforme encontrado no apêndice “B”, “C” “D” e “E” para estudo do plano de negócio, que ajudara no desenvolvimento e criação do projeto.

Mas durante esses processos teve outros fatores importantes, que levaram a escolha da linguagem de programação, definida que seria o PHP Orientado a Objetos, cabe a escolha da ferramenta para auxílio no desenvolvimento do software que foi a IDE NetBeans 7.3, uma ferramenta de código aberto, gratuita e também pode ser utilizada em vários sistemas operacionais tais com Windows, Linux e Mac, também podendo desenvolver com essas ferramentas sistemas com outras linguagens de programação tanto com aplicação *Web, Desktop ou Mobile*.

Para armazenamento dos dados foi adotado o banco de dados MySQL, para trabalhar com o banco de dados utilizou-se o phpMyAdmin e o Mysql Workbench 5.2 que ajudou na análise do banco de dados, junto com essas escolhas foi definido o servidor web que será executado diretamente numa máquina local, para isso foi definido o XAMPP que possui um pacote de aplicações livres como Apache, PHP, MySQL e phpMyAdmin.

Na elaboração da documentação do software como modelagem da UML o software escolhido foi Astah Professional, na elaboração da modelagem lógica do banco de dados utilizou-se o Mysql Workbench, conforme resultados representados no apêndice de diagramas de UML “B”, “C”, “D” e “E”.

Levando em conta os resultados obtidos durante a análise de requisitos, seguiu-se para a etapa de desenho determinada pelo ciclo de vida, primeiramente obteve-se os diagramas de caso de uso, depois os diagramas de classes, de sequência e por fim o de modelagem lógica do banco de dados que pode ser encontrado no apêndice conforme referido no parágrafo anterior.

Obtendo os diagramas ocorreu à necessidade de determinar uma implementação de padrão de projeto (*design Pattern*) a ser seguido, neste caso foi determinado o padrão MVC que auxiliou na organização dos códigos e dando uma facilidade na hora de desenvolver e na busca de códigos para a manutenção se necessário.

Então para início do desenvolvimento do sistema utilizamos a NetBeans IDE para criar um novo projeto Web em PHP com o nome de “websis”, dentro desse projeto cria-se a estrutura do projeto com as seguintes pastas: *config*, *controller*, *model*, *view*, *public* e *plugin*. Sendo que cada uma das pastas tem sua função, a pasta *config* tem a função de configurações como exemplo os dados de conexão com o banco de dados, a *controller* tem as classes principal, de conexão com banco e seção, a *model* são as

classes que trabalharam com o *view* e também em que está concentrada grande parte da programação, a *view* se trata da visão, no caso todas as telas do sistema em HTML e algumas classes para interagir com o *controller* e *model*, na pasta *public* ficarão todos os arquivos que o usuário tem acesso para gerar e montar juntamente com a *view* as telas, como o CSS e JavaScript, na pasta *plugin* vai estar as bibliotecas que neste caso seriam a biblioteca FPDF que é uma função para gerar arquivos em pdf, que no sistema será utilizada para geração de relatórios.

Durante o desenvolvimento do software foi possível adquirir conhecimentos novos na forma de programar com PHP orientado a objetos, como a maneira de conseguir separar o PHP do HTML criando uma classe para ler o código HTML como na figura 3.

```
Class Html{

    public static function load( $arquivo_html ){
        if(file_exists($arquivo_html)):
            return file_get_contents($arquivo_html);
        else:
            return 'O arquivo "'.$arquivo_html.'" não foi encontrado !';
        endif;
    }
}
```

Figura 3. Classe para ler arquivos da extensão Html.

A classe a cima tem a função de verificar se o arquivo que vai ser requerido durante sua instância existente, caso seja existente vai retornar os arquivos presentes no arquivo HTML, caso contrário vai retornar um erro informando que o arquivo requerido não foi encontrado. Para chamar essa classe, instancia um novo objeto, tendo o objeto chama-se o método da classe load e passa-se seu parâmetro conforme a figura 4 abaixo representa.

```
$html_fornecedor = new Html();
$html = $html_fornecedor->load('view/fornecedor-show.html');
```

Figura 4. Instanciando um novo objeto.

Também houve a oportunidade de conhecer bibliotecas como o Bootstrap, que seria um CSS com vários estilos, mais primeiramente teve que se estudar qual era sua necessidade para poder utilizar o mesmo, verificando na documentação encontrada no site, que mostra do mais básico ao mais avançado separando por categorias diferentes. Outra biblioteca que auxiliou na criação dos relatórios foi a FPDF, é uma biblioteca

bem eficiente para gerar documentos em pdf, seu código de fácil entendimento e de interação com projetos.

O processo de desenvolvimento trouxe o resultado de um software Web em PHP orientados a objetos, onde o software atente as funções requeridas primeiramente pela empresa na coleta de requisitos, pois passa pelo processo de implementação, onde ainda a empresa encontra-se em adaptação do local e aquisição de novos hardwares.

Após e durante essa etapa de programação surgiram uma serie de imprevistos, como o ciclo de vida escolhido, parte para realização de outra parte do ciclo que é a implementação, ocorrendo um imprevisto no desenho ou um detalhe que ficou confuso o ciclo de vida nos permite voltar diretamente para etapa de desenho neste caso, não havendo a necessidade de fazer todo o processo do ciclo por completo para depois retornar a tarefa onde ocorreu o erro. Outro imprevisto ocorreu durante a programação percebe-se uma necessidade de criação de um *trigger* (gatilho), que está associado com uma tabela que executa a função especificada pela *trigger* no banco de dados que trabalhara toda vez que for inserida, deletada ou atualizada a tabela especifica na criação da mesma.

Durante os testes realizados não foi possível obter um resultado significativo, pois a empresa está no processo de inserção de dados, como eram poucos os dados no sistema, o software atendeu todos os requisitos que a ele foi determinado.

Passado todo o processo de desenvolvimento do software ou programação, iniciou-se a etapa de testes, em que foram realizadas todas as etapas necessárias do processo que o software apresenta. Havendo necessidade de correção ou apresentação de uma falha durante o desenvolvimento do software, o ciclo de vida permite voltar diretamente na etapa de desenvolvimento anterior. Tendo realizado todos os testes, correções e abordado todos os processos presentes no ciclo de desenvolvimento, inicia-se a parte de implantação do sistema na empresa.

Resultados

Por meio do desenvolvimento do sistema Web para Sorveteria, foi possível aprimorar todos os conhecimentos adquiridos durante o decorrer do curso e também aperfeiçoar o conhecimento em várias partes de programação com PHP trabalhando a orientação a objetos, com questões de JavaScript onde aborda o aprendizado do AJAX e seus recursos, em relação a banco de dados a *trigger* e sua real importância para organizar os dados no banco.

Em relação da UML ajudou a entender os processos do software, onde identificado suas funções no diagrama caso de uso, entender como o usuário realiza a função por meio do diagrama de sequência na execução de uma tarefa, a modelagem lógica do banco de dados e o diagrama de classes.

O processo de desenvolvimento resultou em um software Web em PHP orientados a objetos que estão representados no apêndices de Telas do Sistema: “H”, “I”, “J”, “K”, “L”, “M” e “N”, onde o sistema atente as funções requeridas primeiramente pela empresa na coleta de requisitos, que foi de fundamental importância

para o planejamento dentro do ciclo de vida abordado no processo de criação do software.

Considerações Finais

Com a realização deste trabalho foi possível colocar em pratica os conceitos adquiridos, e a realização do processo de desenvolvimento de um software. A importância de fazer um planejamento com um ciclo de vida e fatores que compõem suas etapas como um levantamento de requisitos, a montagem de um cronograma bem definindo para a conclusão dos mesmos.

Também durante algumas etapas apareceram vários resultados muito deles positivos, por exemplo: como as telas presentes no apêndice “K”, “L”, “M” e “N”, na implantação do software na empresa onde realmente será testado pelos usuários podendo obter resultados satisfatórios, ou podendo aparecer novas necessidades de modificações e até geração de novos módulos.

Futuramente pretende-se analisar os resultados obtidos e descritos pela empresa onde o sistema está sendo implantado, onde dará oportunidade de continuidade ao desenvolvimento do software podendo-se criar funções novas como a implantação de leitor de código de barras.

Referências Bibliográficas

- BEZERRA, Eduardo. (2007) “Princípios de Análise e Projeto de Sistemas com UML”. Editora: Elsevier. Rio de Janeiro – RJ.
- BRUNO, Odemir M; ESTROZI, Leandro F.; BATISTA NETO, João E. S. (2010) “Programando para a Internet com PHP”. Editora Brasport, Rio de Janeiro – RJ.
- CARBONI, Irenice de Fátima. (2003) “Lógica de Programação”. Editora: Pioneira Thomson Learning, São Paulo – SP.
- CARVALHO, M. G. (1997) “Tecnologia, Desenvolvimento Social e Educação Tecnológica. Revista Educação & Tecnologia”. Curitiba: Centro Federal de Educação Tecnológica do Paraná, julho de 1997, semestral, p.70-87
- COCHRAN, David. (2012) “Twitter Bootstrap Web Development How-To”. Editora: Packt Publishing Ltd. Livery Place 35 Livery Street Birmingham B3 2PB, UK (Reino Unido).
- COSTA, Carlos J. (2007) “Desenvolvimento para Web”. Editora Lusocredito, Ltda. Pr. Mota Veiga – Lisboa. Portugal.
- DALL'OGGIO, Pablo. (2009) “PHP Programando com Orientação a Objetos”. Editora: Novatec. São Paulo – SP.
- DENNIS, Alan. WIXOM, Barbara Haley. (2011) “Análise e Projeto de Sistemas: Segunda Edição”. Editora: LTC.
- FERRARI, Fabrício Augusto. (2007) “Crie banco de dados em MySQL”. Editora: Digerati Books. São Paulo – SP.

- FLANAGAN, David. (2002) “JavaScript: o guia definitivo”. Editora: Artmed. Porto Alegre – RS.
- GUIMARÃES, Célio Cardoso. (2010) “Fundamentos de banco de dados: Modelagem, projeto e linguagem SQL”. Editora: Unicamp. Campinas – SP.
- HEUSER, Carlos Alberto. (2009). “Projeto de Banco de Dados”. Editora: Bookman. Porto Alegre – RS.
- MELO, Ana Cristina. (2010) “Desenvolvendo aplicações com UML 2.2: do conceitual à implementação”. 3ª edição. Editora Brasport, Rio de Janeiro – RJ.
- MELO, Alexandre Altair de. e NASCIMENTO, Mauricio G. F. (2007) “PHP Profissional: Aprenda a desenvolver sistemas profissionais orientados a objetos com padrões de projeto”. Editora: Novatec. São Paulo – SP.
- MORONI, Herbert. (2007) “Criação de sites em AJAX Tudo sobre a linguagem que revolucionou a Internet”. Editora: Digerati Books. São Paulo – SP.
- NIEDERAUER, Juliano. (2007) “Web interativa com Ajax e PHP”. Editora: Novatec. São Paulo – SP.
- OLIVEIRA, Celso Henrique Poderoso. (2011) “SQL: Curso Prático”. Editora: Novatec. São Paulo – SP.
- PAULA FILHO, Wilson de Páuda. (2011) “Engenharia de Software: Fundamentos, Métodos e Padrões”. Editora: LTC. Rio de Janeiro – RJ.
- SILVA, Maurício Samy. (2008) “Construindo sites com CSS e (X)HTML: Sites controlados por folhas de estilo em cascata”. Editora: Novatec. São Paulo – SP.
- SILVA, Mauricio Samy, (2010) “jQuery: a biblioteca do programador JavaScript”. Editora Novatec, São Paulo – SP.
- SOMERA, Guilherme. (2006) “Treinamento prático em CSS”. Editora: Digerati Books, São Paulo – SP.
- XAVIER, Fabrício S. V. (2008) “PHP – do Básico à Orientação a Objetos”. Editora: Ciência Moderna. Rio de Janeiro – RJ.

Apêndices

Levantamento de requisitos referentes ao Sistema Web para Sorveteria

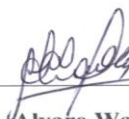
- Levantamento de Requisitos
 - Apêndice A: Relatório de Levantamento de requisitos

Levantamento de requisitos


Requisitos

1. Função para usuário cadastrar cliente no sistema;
2. Permitir ao usuário cadastrar produtos e categorias;
3. Função para usuário cadastrar fornecedores;
4. Função para usuário movimentar estoque;
5. Função para usuário realizar operação de venda;
6. Relatórios;

Data: 02/11/12.



Alvaro Waechter
Proprietário

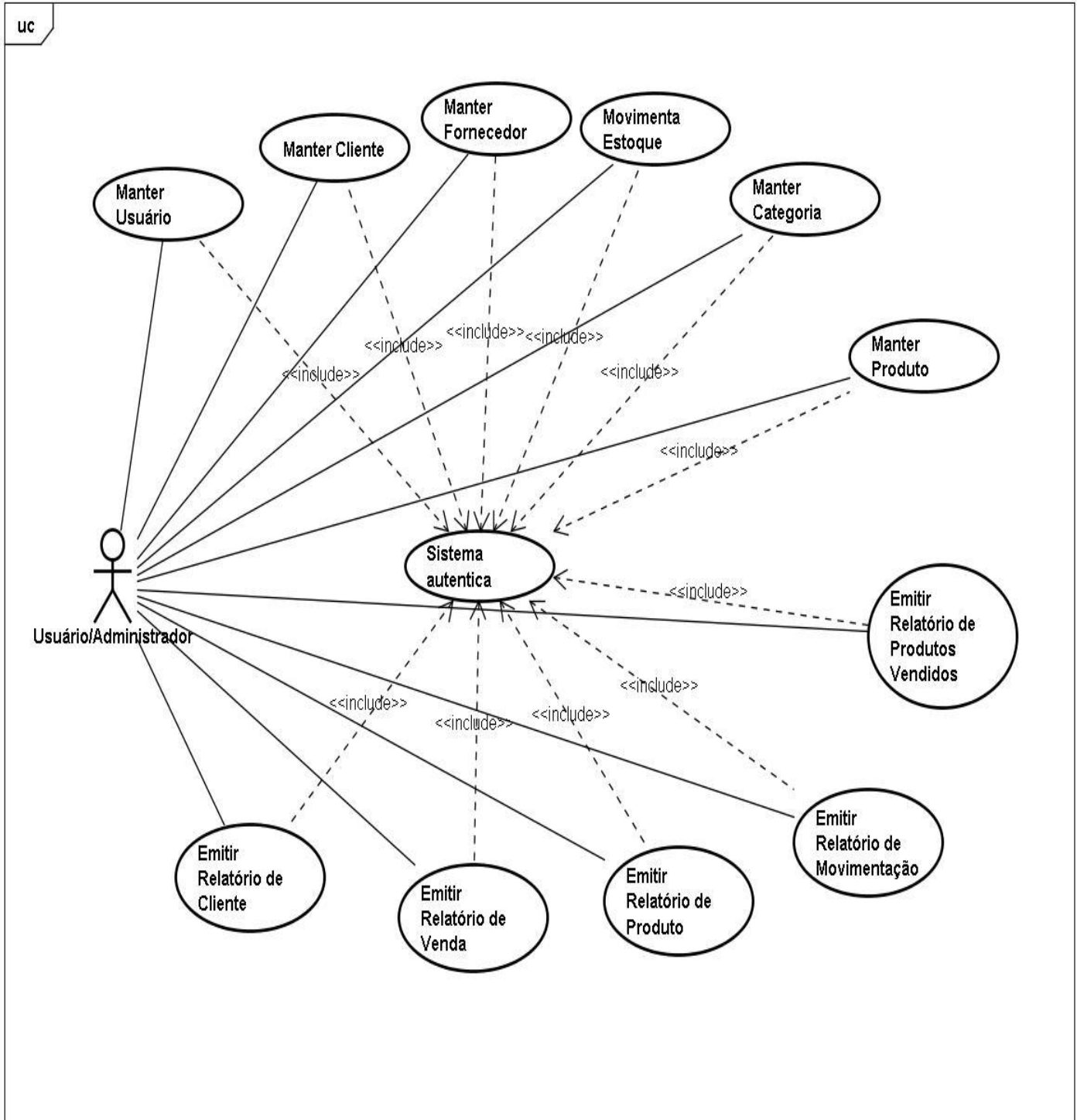


Misael Alan Sömer
Acadêmico de Analista de Sistemas

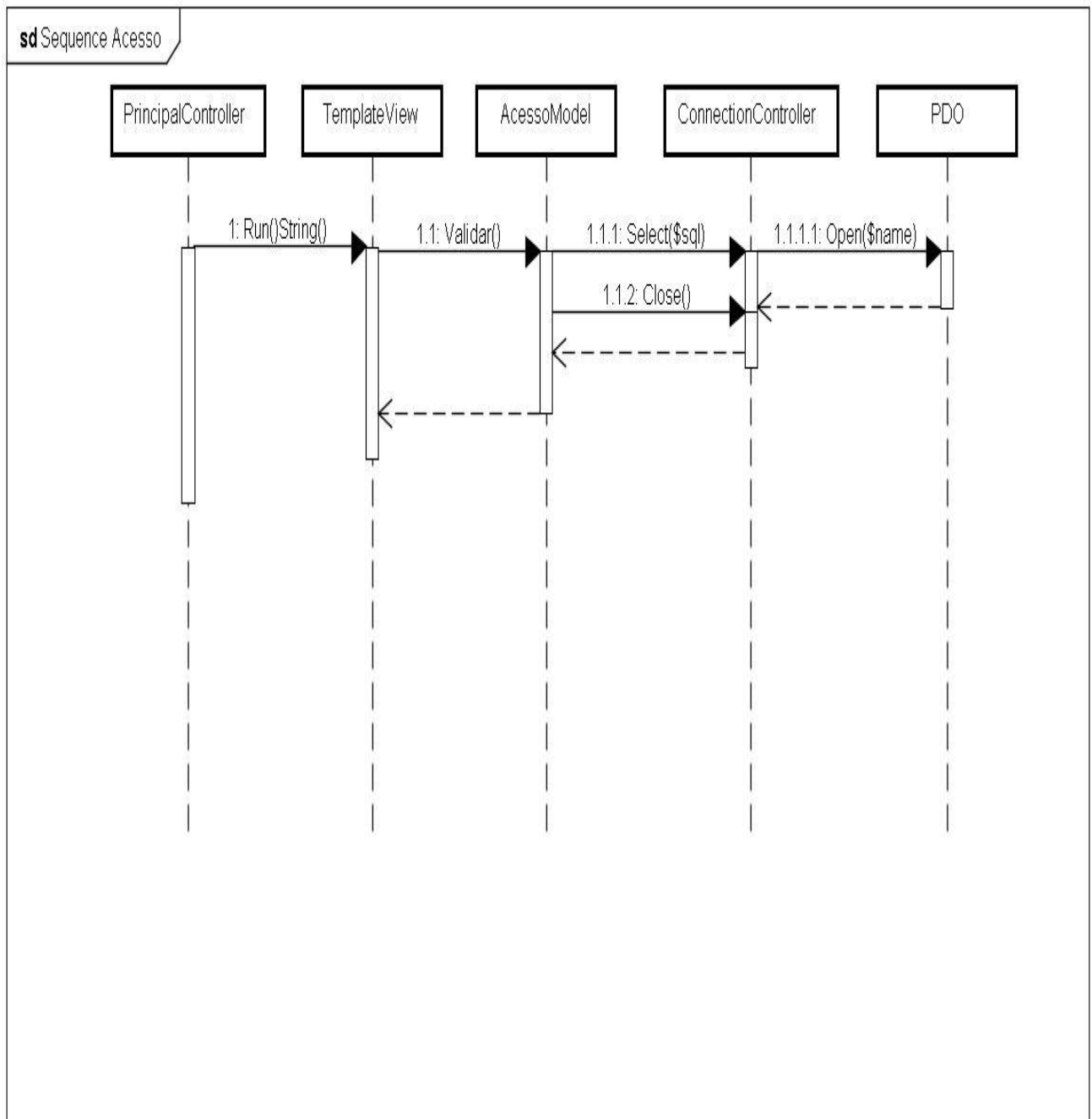
Guarapuava - PR

Diagramas UML referentes ao Sistema Web para Sorveteria

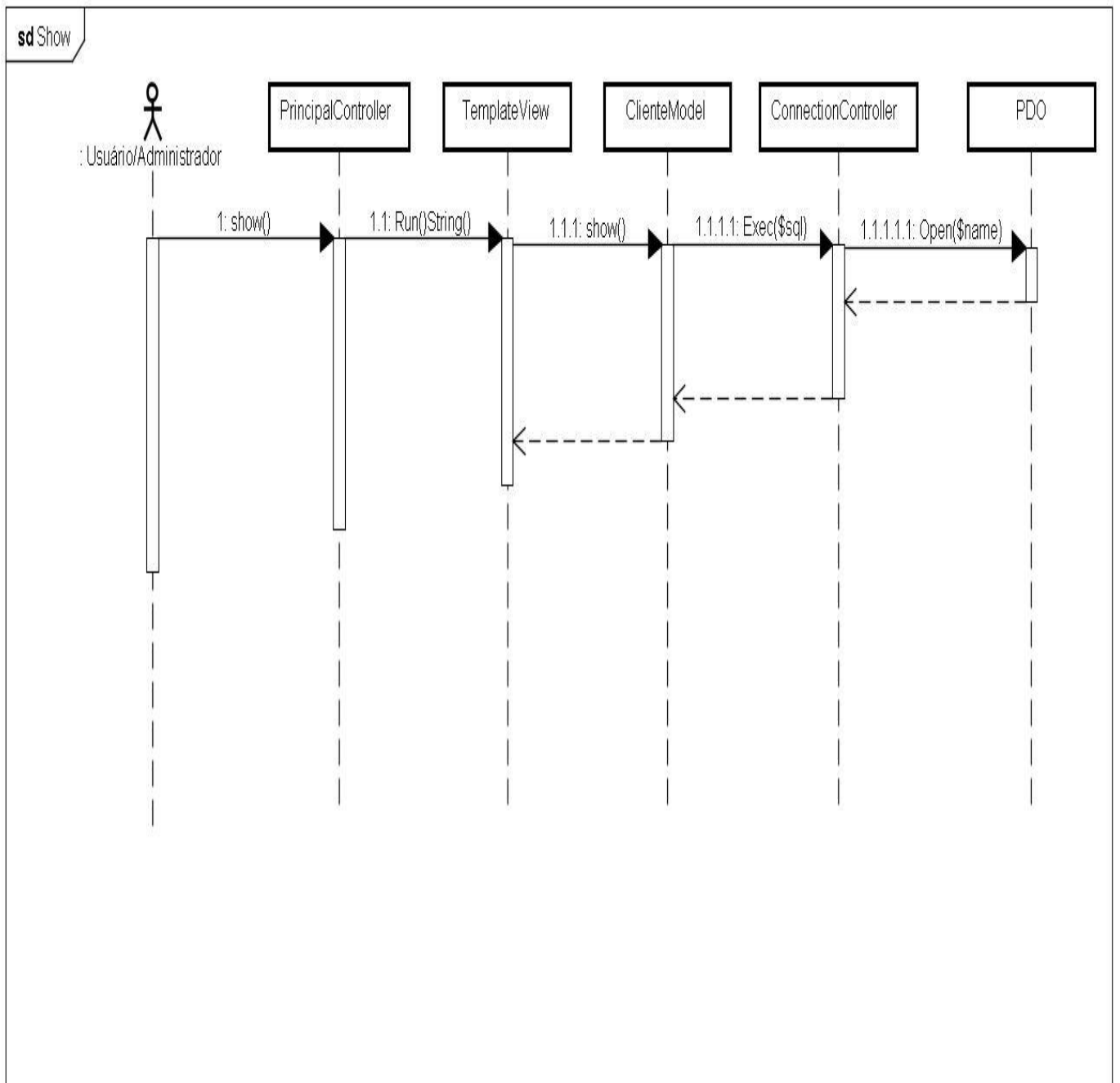
- Diagramas de Caso de Usos
 - Apêndice B: Diagrama Caso de Uso.



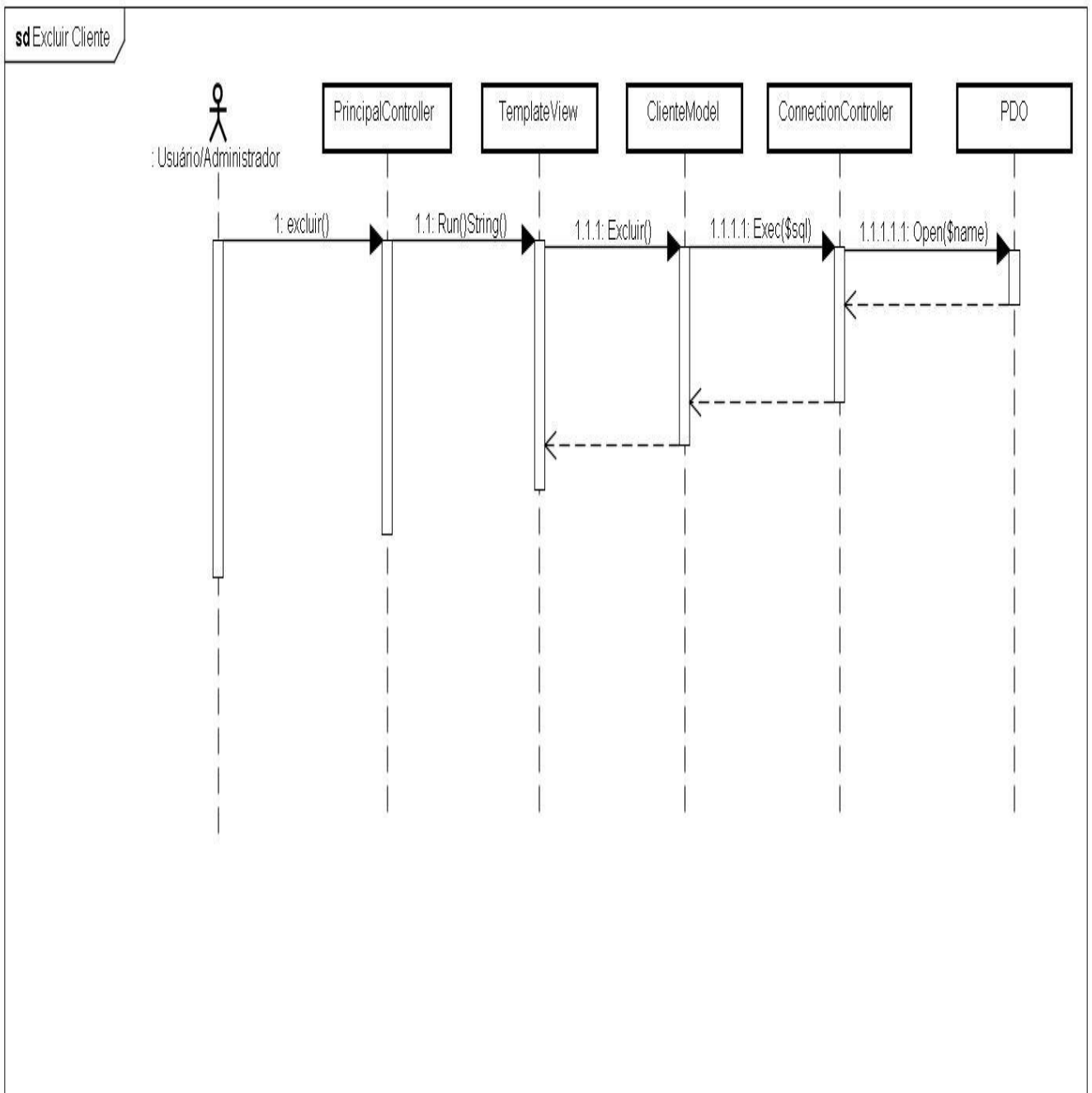
○ Apêndice C: Diagrama de Sequência Acesso.



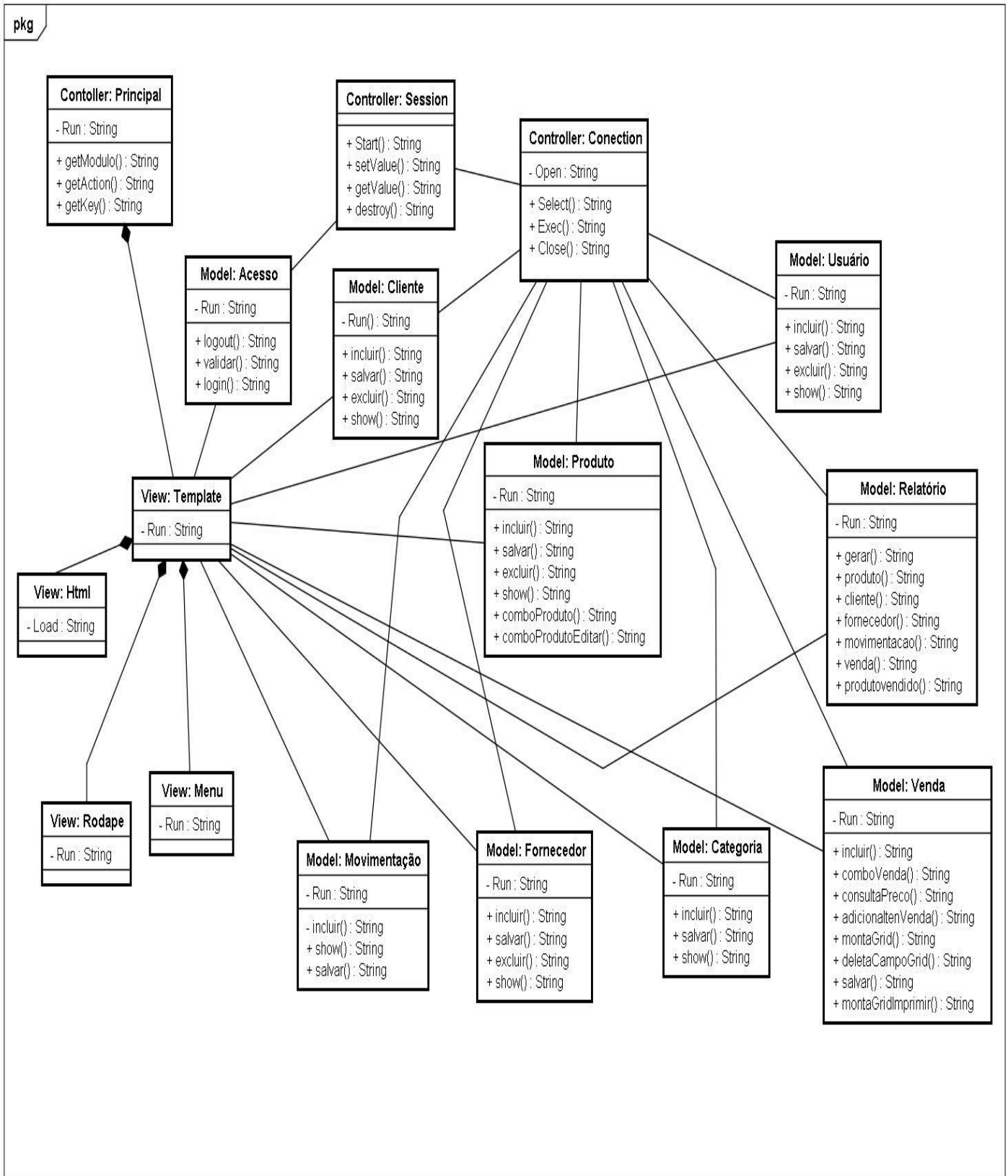
○ Apêndice D: Diagrama de sequência Show



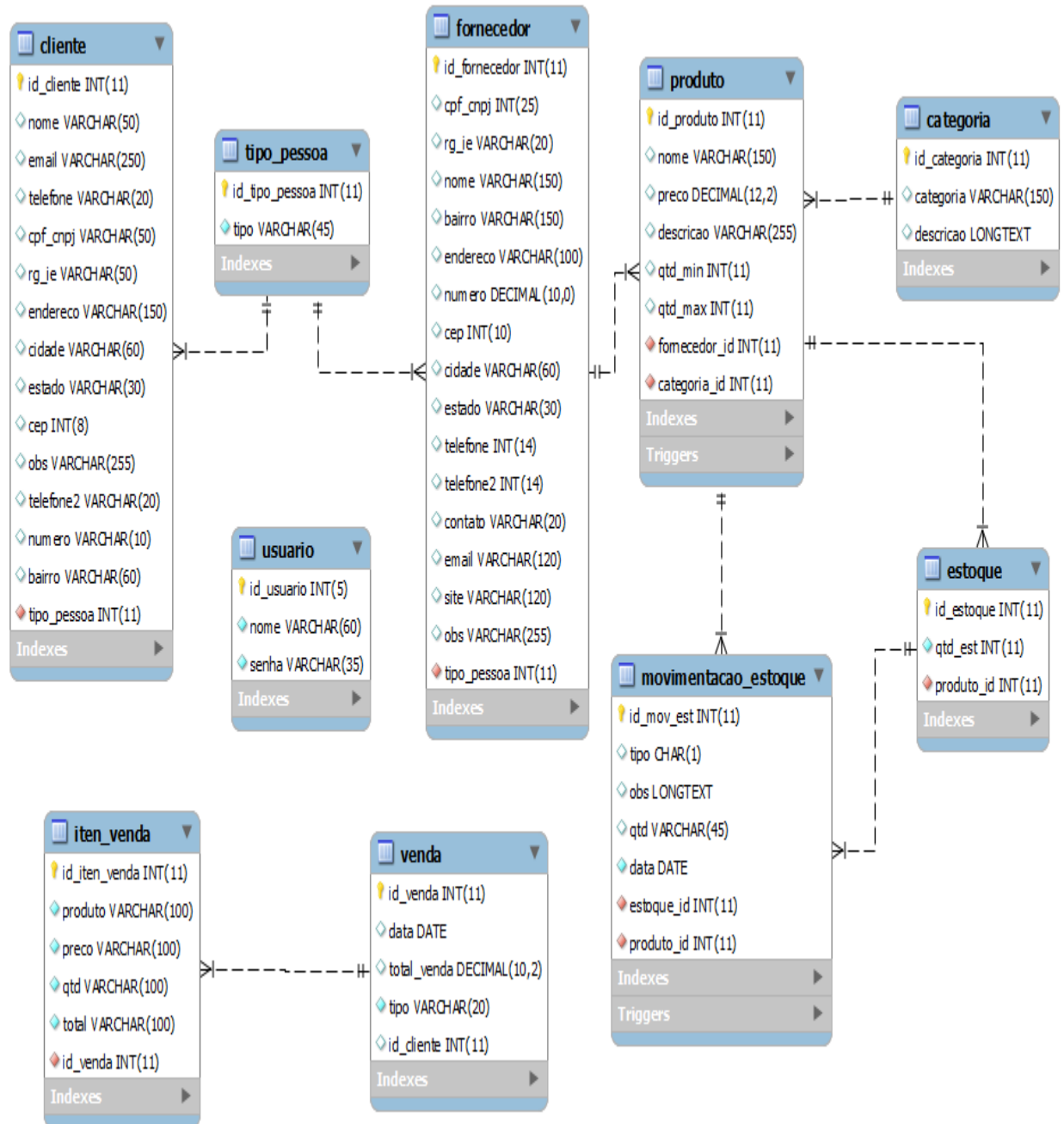
○ Apêndice E: Diagrama de Sequência Excluir Cliente.



○ Apêndice F: Diagrama de Classe

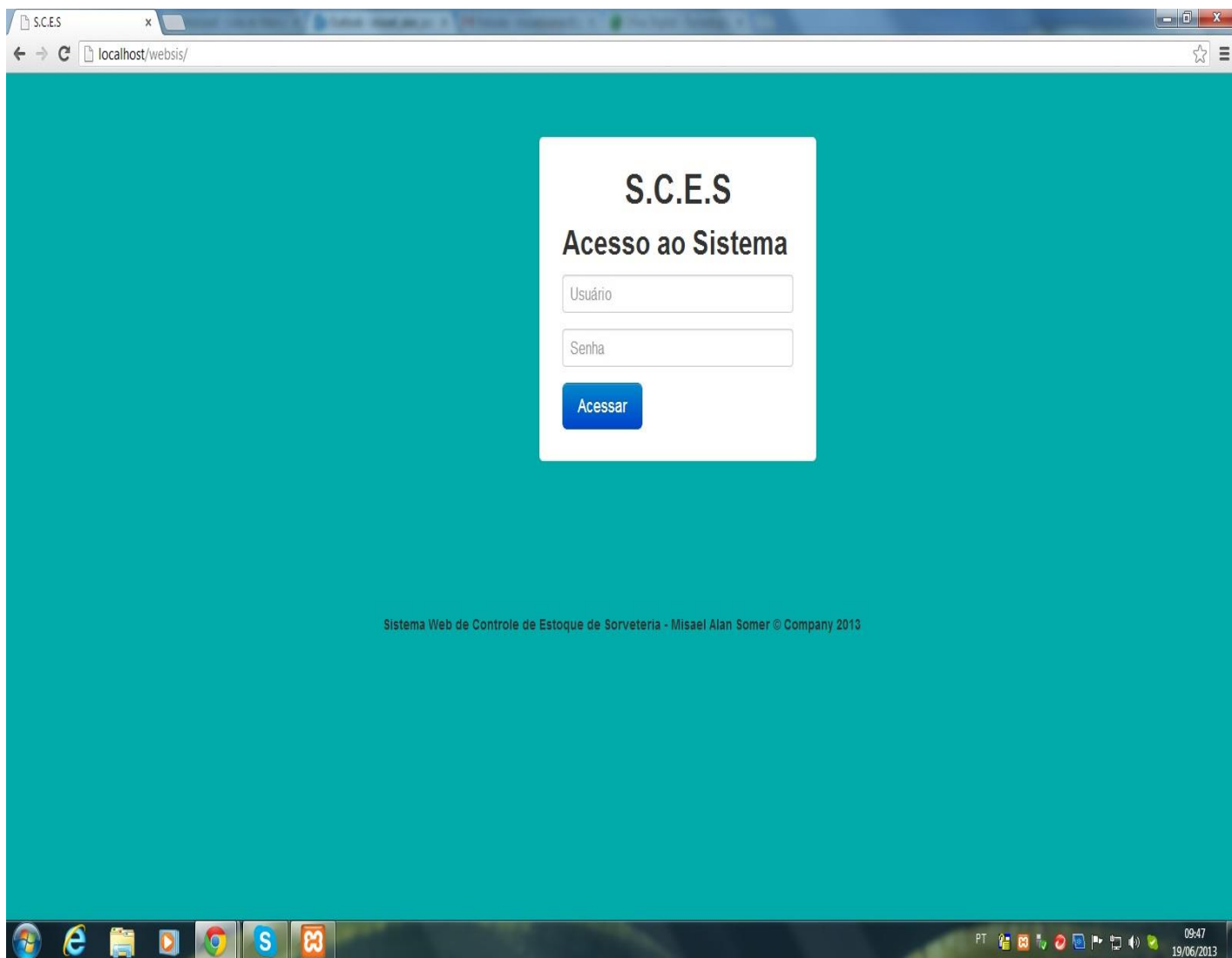


○ Apêndice G: Diagrama de Modelagem de Banco de Dados.

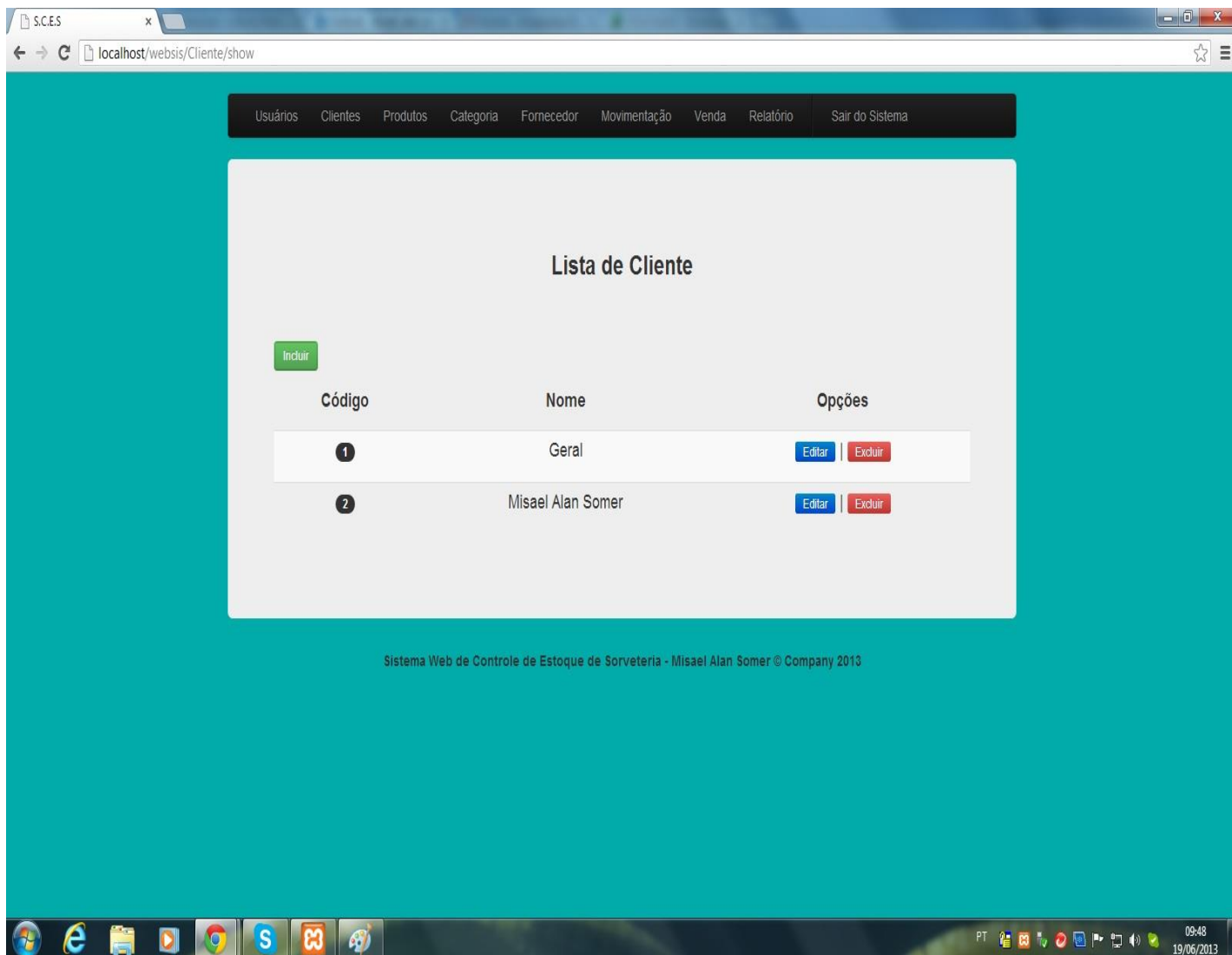


Imagens referentes ao Sistema Web para Sorveteria

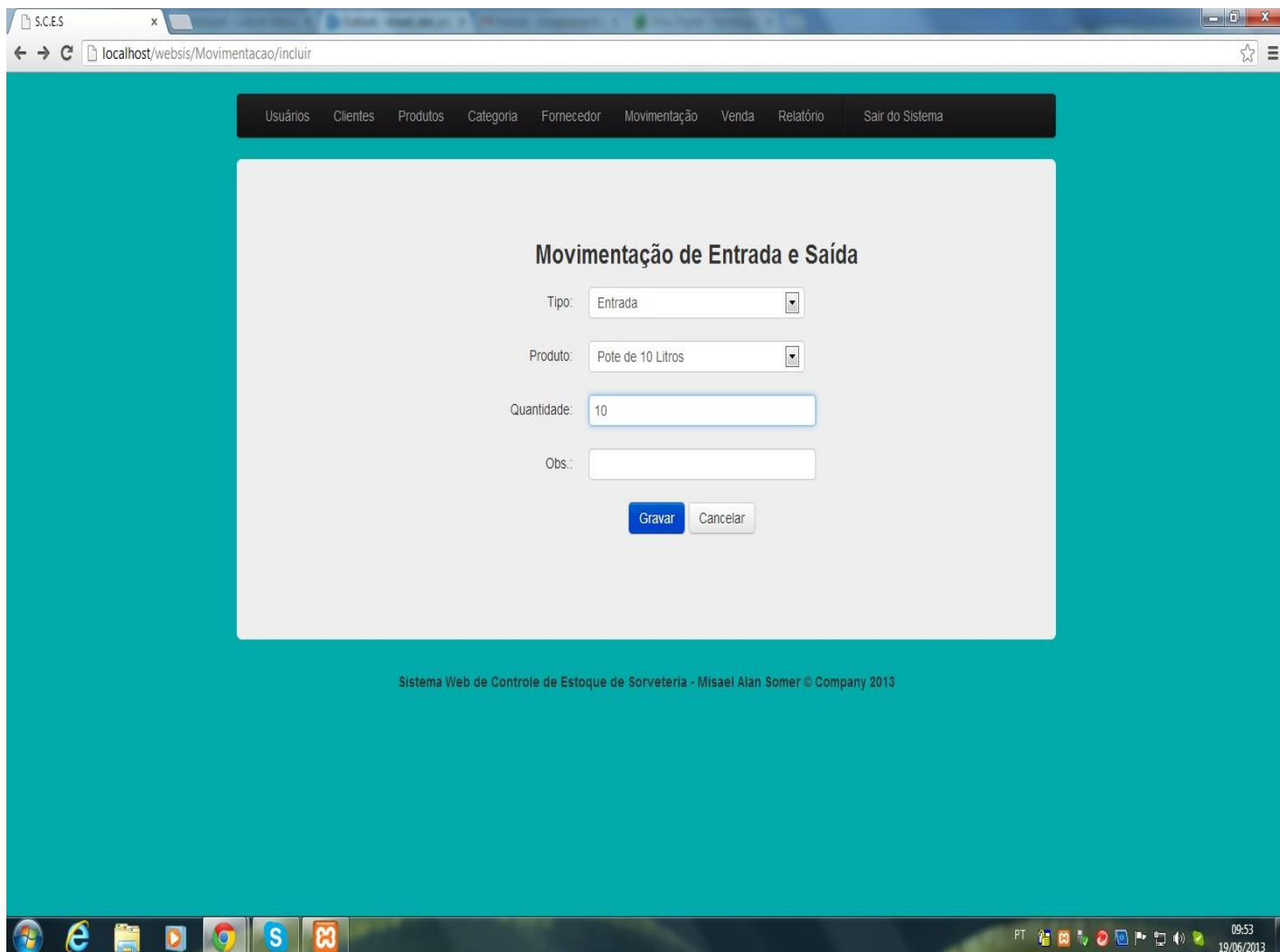
- Telas do Sistema.
 - Apêndice H: Tela de acesso ao Sistema.



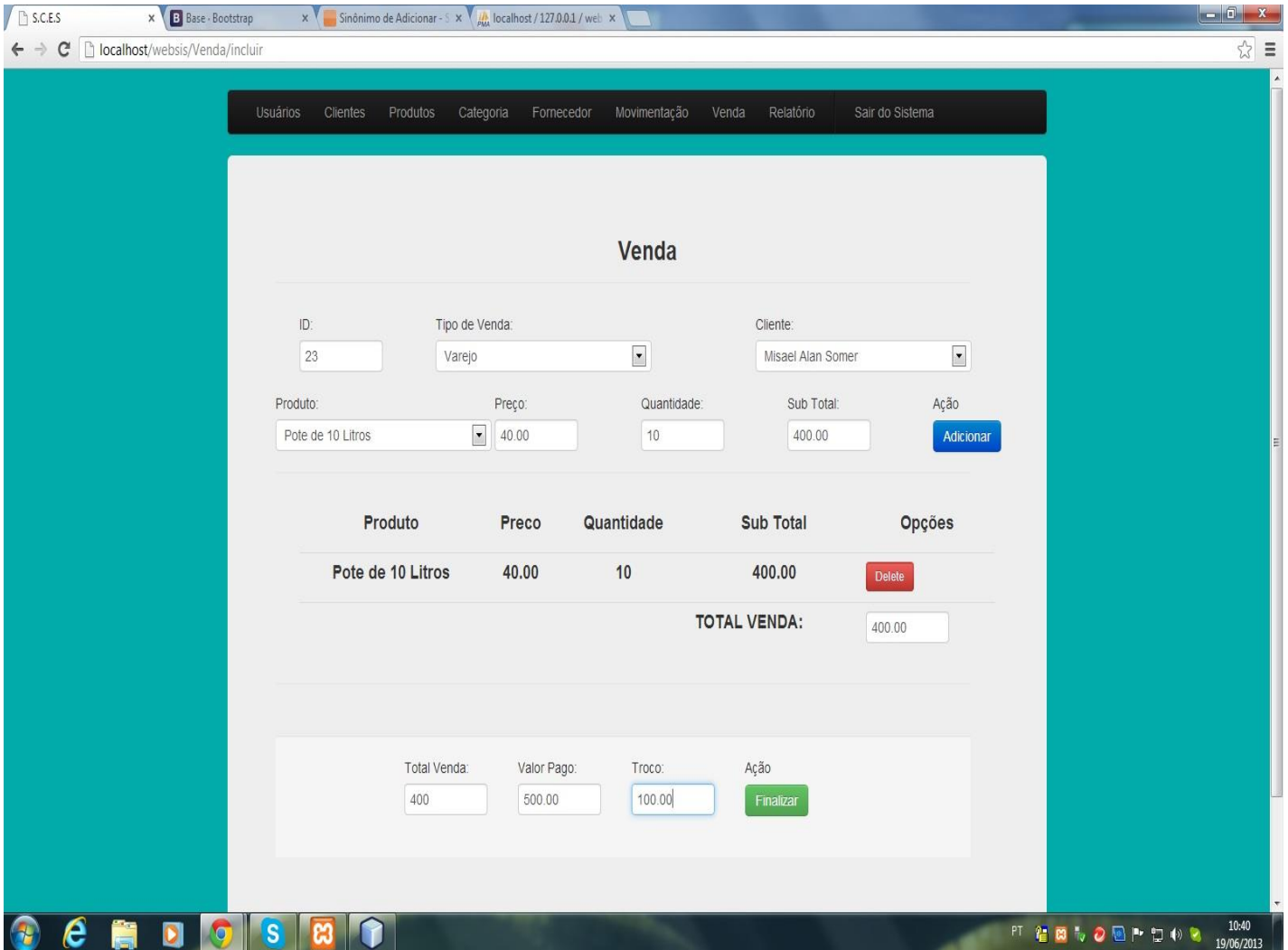
- Apêndice I: Tela lista de cliente.



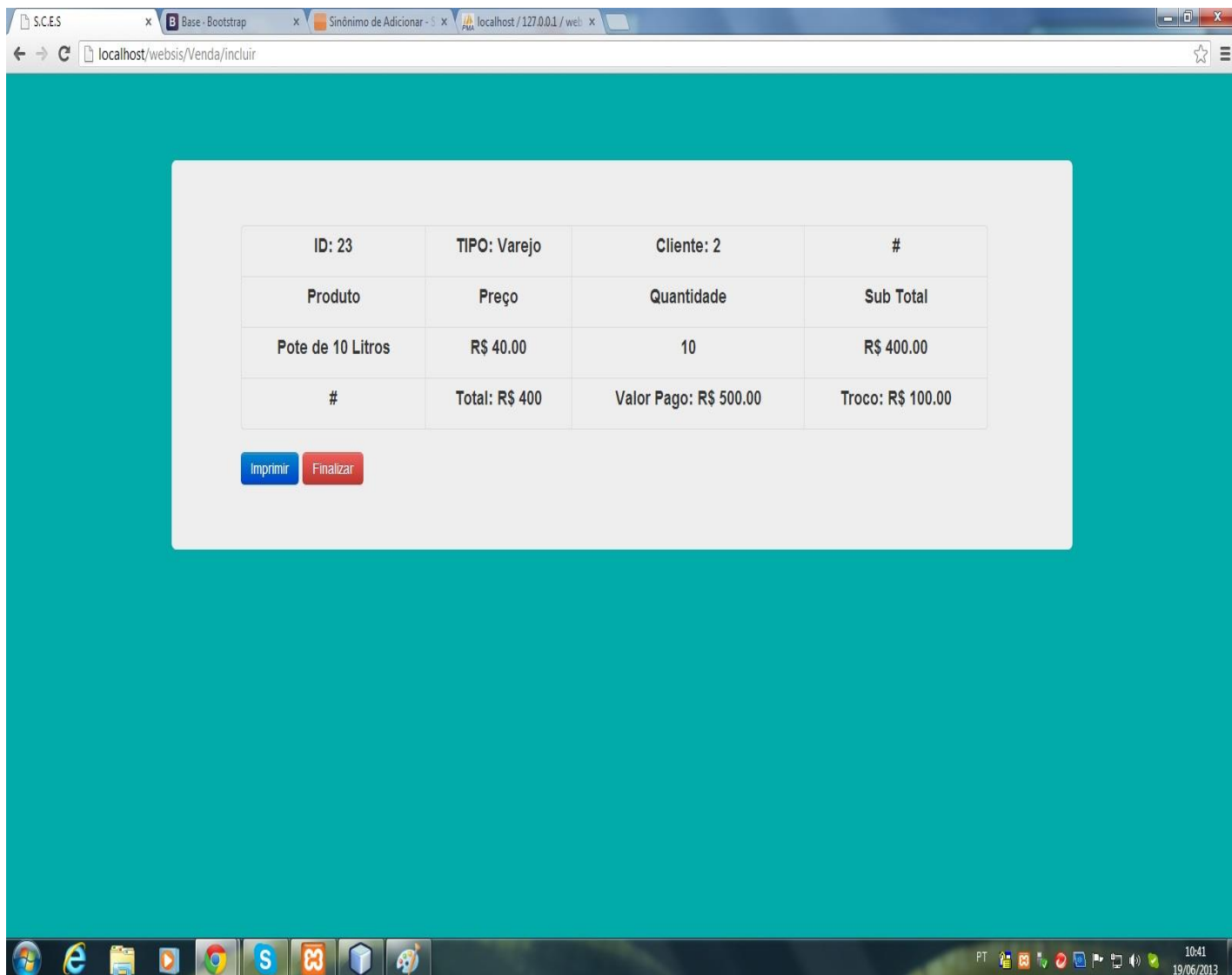
- Apêndice J: Tela de Movimentação de Estoque.



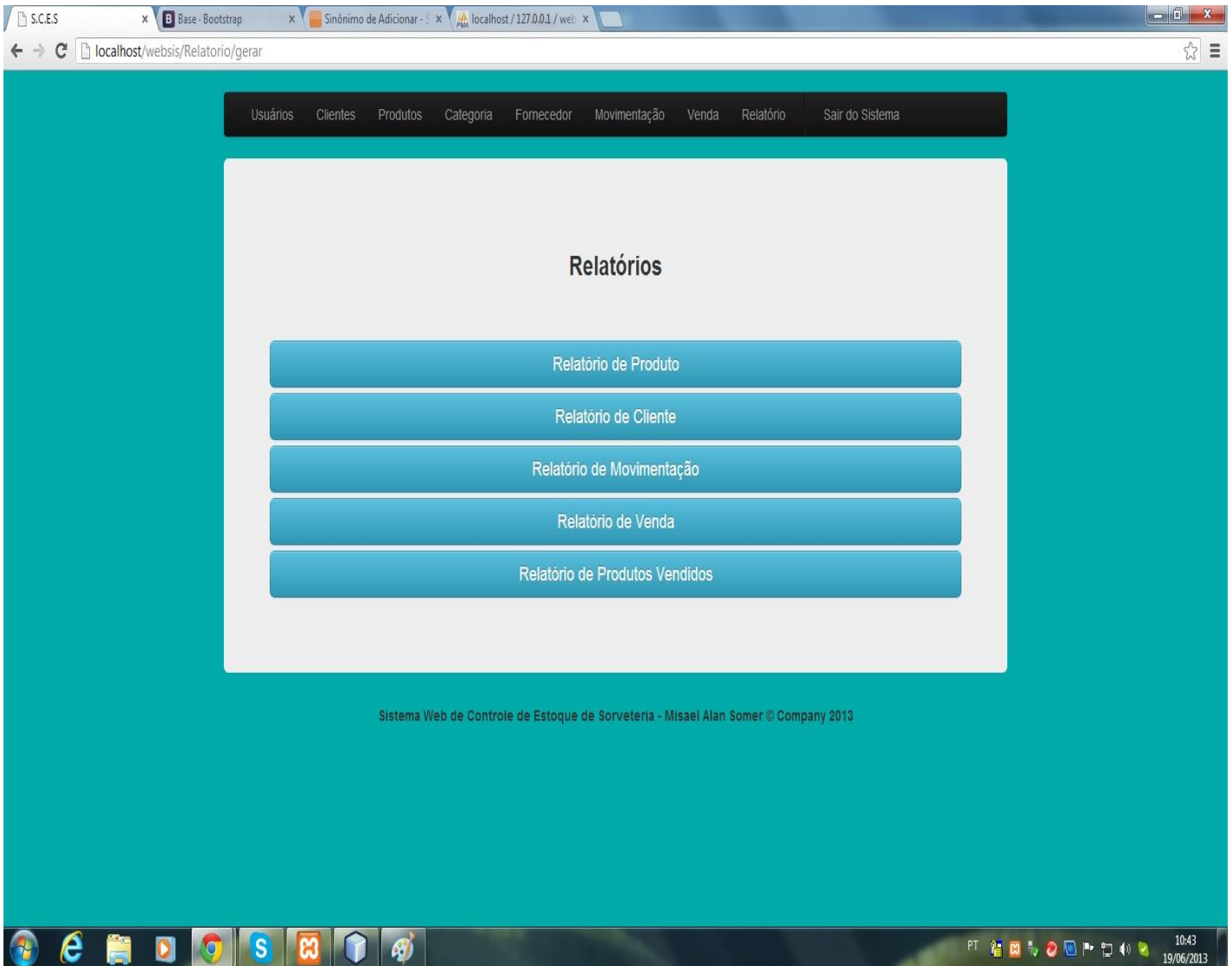
○ Apêndice K: Tela de Venda.



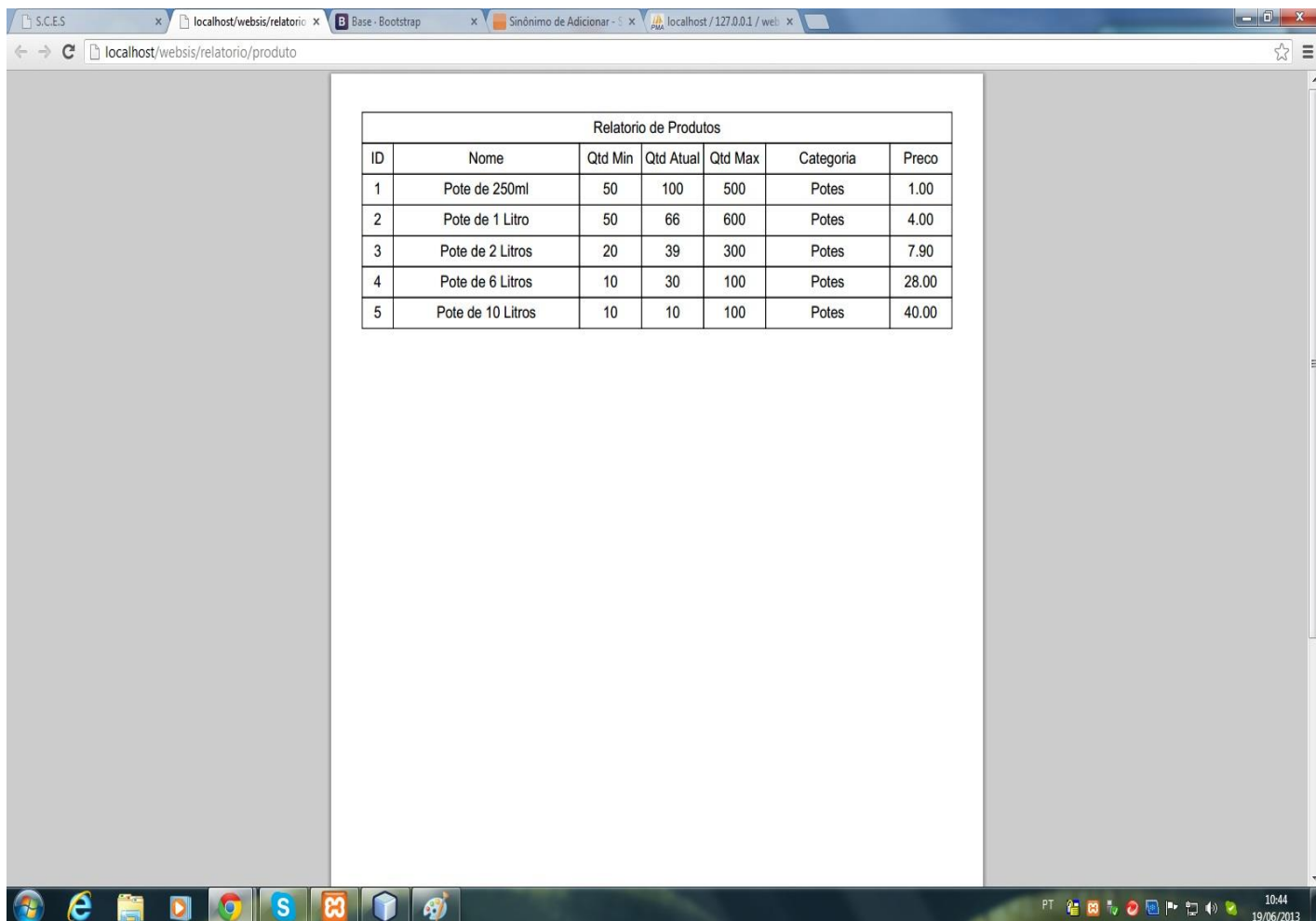
- Apêndice L: Tela de comprovante da venda realizada para imprimir.



○ Apêndice M: Tela de Relatórios.



- Apêndice N: Tela com Relatório de Produto.



The screenshot shows a web browser window with the address bar displaying 'localhost/websis/relatorio/produto'. The main content area contains a table titled 'Relatorio de Produtos'. The table has seven columns: ID, Nome, Qtd Min, Qtd Atual, Qtd Max, Categoria, and Preco. It lists five product entries with their respective quantities and prices.

Relatorio de Produtos						
ID	Nome	Qtd Min	Qtd Atual	Qtd Max	Categoria	Preco
1	Pote de 250ml	50	100	500	Potes	1.00
2	Pote de 1 Litro	50	66	600	Potes	4.00
3	Pote de 2 Litros	20	39	300	Potes	7.90
4	Pote de 6 Litros	10	30	100	Potes	28.00
5	Pote de 10 Litros	10	10	100	Potes	40.00