

Desenvolvimento *WEB* de Sistema de Movimentação Financeira Para Empresas de Pequeno Porte

Jean Wilderson Bodot¹, Willian Ricardo Fialka Agner²

^{1,2}, Tecnologia em Análise e Desenvolvimento de Sistemas
Faculdade Guairacá.
Guarapuava – PR – Brasil

wildersonbodot@hotmail.com, will.agner@hotmail.com

Resumo. Diante da necessidade de aprofundar os conhecimentos nas áreas que envolvem o desenvolvimento de um Sistema Web, o presente trabalho busca apresentar a criação de um software e sua aplicabilidade no auxílio financeiro de empresas de pequeno porte. Sendo assim, amenizando a dificuldade na aquisição de tais ferramentas para administração financeira, relacionados aos custos de sistemas mais complexos de controle de entradas, saídas e fluxo de caixa. Dessa maneira suprir a necessidade de instrumentos que auxiliem tais empresas a alcançar os objetivos almejados numa realidade onde os softwares deixam de ser instrumentos somente de empresas e empreendedores de grande porte. Demonstrando também as vantagens da utilização de ferramentas de desenvolvimento open source como uma opção de redução de custos.

Abstract. Faced with the need to deepen the knowledge in areas that involve the development of a Web system, this paper aims to present the creation of software and its application in financial assistance to small businesses. Therefore, easing the difficulty in acquiring such tools for financial management regarding the costs of more complex systems of control inputs, outputs and cash flow. Thus to meet the need of tools that help such companies to achieve the goals set in a reality where software tools are no longer just business entrepreneurs and large. Also demonstrating the advantages of using open source development tools as an option for reducing costs.

1. Introdução

O uso de um *software* como ferramenta de auxílio para automatização de processos que antes eram executados exclusivamente pelo homem deixou de ser somente para empresas e empreendedores de grande porte. Cada vez mais buscando aprimorar seus rendimentos, as empresas procuram por tecnologia para obter melhores resultados financeiros.

Na atualidade, para uma empresa sobreviver no mercado ela necessita de agilidade para reagir as constantes variações e tomadas de decisões diante de tamanha competitividade. Sendo esta uma tarefa complicada e dificultosa, pois a apresentação detalhada dos resultados financeiros é um trabalho que demanda muito tempo e dedicação, efetuando cálculos e comparações para obtenção de resultados, pois afinal o que está sendo julgado é a vida financeira da empresa.

Com uma ferramenta específica que torne este processo ágil, diante da complexidade para captação desses resultados, onde é necessário que sejam feitas buscas setorizando contas, sejam receitas, sejam despesas e que perde-se muito tempo na execução desta tarefa e os resultados acabam sendo desatualizados, é o que justifica a necessidade de estudos como este. O tempo desperdiçado no levantamento dados poderia estar sendo utilizados em ações que realmente trouxessem efeito positivo, pois afinal, nesse mercado competitivo, tempo é dinheiro.

Neste sentido, o presente trabalho apresenta um *software WEB*, com ferramentas *Open Source*, como uma solução com custo reduzido para desenvolvimento *WEB*, com o uso de ferramentas de fácil aprendizado, propondo uma facilitação na obtenção dos resultados de receitas e despesas, e facilitar o paradigma do fluxo de caixa, tornando esses resultados de fácil entendimento para empreendedores, gerentes, operadores financeiros, que poderão fazer uso dos resultados apresentados.

Na pesquisa e desenvolvimento do projeto foram utilizadas as seguintes ferramentas: *HTML*, *CSS*, *PHP*, *SGBD MySQL*, *MySQL Workbench*, *NetBeans*, *XAMPP*, *Astah Community*, com ciclo de vida incremental além da utilização do padrão de arquitetura de software MVC, que proporciona um melhor estruturação das pastas do sistema.

2. Fundamentação Teórica

Para o desenvolvimento de um *software* de diagnósticos financeiros de uma empresa, não existe a necessidade de ser especialista em todas as funções organizacionais em consultoria financeira, mas devemos sê-lo para estabelecer um relacionamento que ajude o cliente na tomada de decisões, a função do desenvolvedor é criar diagnósticos que orientem e auxiliem o cliente, segundo Schein (1972, *apud* Teixeira, 2012).

O objetivo é estabelecer uma maneira de captura dos resultados obtidos em um determinado período e expô-los aos clientes de maneira rápida, precisa e hábil.

Considerando a análise de Teixeira (2012), o desenvolvimento de ferramentas que possam auxiliar na análise dos dados financeiros, não está diretamente ligado a conhecimentos nas áreas contábeis ou administrativas, o conceito adotado está direcionado à facilitação de obtenção e apresentação dos dados, para facilidade de interpretação pelos consultores, diretores ou gerentes.

A competitividade do mercado com a facilidade de acesso e as novas tecnologias, faz com que os empresários busquem maneiras para permanecerem no mercado, mas o que realmente determina essa permanência é o conhecimento de todas as informações financeiras da empresa. Para obter dados realmente precisos e confiáveis, esse resultado deve ser apresentado de forma imparcial e sem envolver interesses. Bastos *et al*(2011) comenta sobre a importância da gestão financeira para uma empresa por tratar da maneira como se analisa as finanças e decide nos investimentos e quanto isso refletirá no resultado.

Cada vez mais o empresário ou o empreendedor, está interessado em melhorar o seu desempenho econômico, muitas vezes desconhecem da existência de ferramentas que possam auxiliá-los no conhecimento do que realmente está acontecendo na área financeira, ou de como fazê-lo. A falta de conhecimento na área financeira também é

uma das dificuldades encontradas, é função do analista financeiro é desenvolver uma solução para tais dificuldades. Borges & Rodrigues (2007) definem que o Demonstrativo de Fluxo de Caixa (DFC) é uma das ferramentas que demonstra de forma detalhada os recebimentos e pagamentos de uma empresa, o que possibilita ao empresário responder a famosa e velha questão! **De onde veio e onde foi aplicado o dinheiro da empresa?** [grifo nosso]

Um levantamento relacionado ao assunto e observações dos autores, propõe o uso do DFC como uma das maneiras de auxiliar a empresa, de uma forma fácil e com o menor custo possível, porém que atenda as expectativas junto à equipe de consultoria, com a colocação de dados que realmente sejam de utilidade para os fins de consultoria. Mendes (2009) comenta sobre o uso do DFC como uma ferramenta de visão na tomada de decisão importante para o gestor definir ações como relação ao caixa, e destaca também a importância para investidores, credores e outros usuários possam avaliar a capacidade de gerar fluxos positivos, de honrar seus compromissos e empréstimos, a liquidez e flexibilidade financeira, além das ações e as reações através das tomadas de decisão com investimentos e financiamentos.

Existem dois métodos de apresentação do DFC das atividades operacionais: o direto e o indireto (figura 1). Hoji (2010) descreve sobre a DFC pelo método direto, como uma melhor forma de visualização e compreensão das operações financeiras e atividades operacionais, já o método indireto mostra quais foram as alterações no giro (Ativo Circulante e Passivo Circulante) que provocaram aumento ou diminuição no Caixa, porém sem explicar diretamente as entradas e saídas de dinheiro. Como o objetivo é demonstrar os resultados de forma hábil, optou-se pelo DFC no método direto.

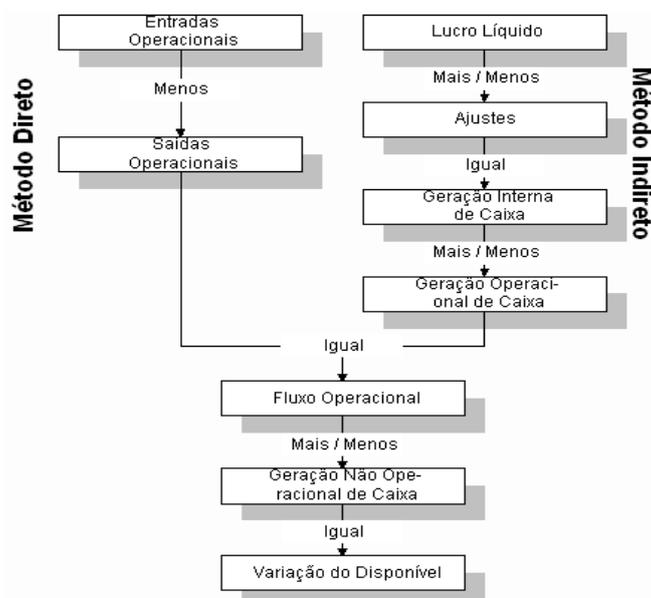


Figura 1. Fonte: (Sá, 1998, pg.36)

A precisão do DFC proporciona uma visão antecipada aos fatos e tendências, também no processo de planejamento da empresa, podendo se acompanhar diariamente o desempenho com tempo para aplicar medidas corretivas. Ferreira (2003) descreve a

importância contábil do DFC, além de não haver necessidade de um conhecimento contábil para o entendimento das informações, a apresentação clara e segura possibilita o auxílio na tomada de decisões, e sua aplicabilidade demonstra a forma com que a empresa emprega os seus recursos.

2.1. Ferramentas Utilizadas

Este capítulo descreve as tecnologias empregadas no desenvolvimento do *software* e do conceito *WEB 2.0* e também das ferramentas utilizadas no desenvolvimento do *software*: *HTML*, *CSS*, *PHP*, *SGBD MySQL*, *MySQL Workbench*, *Netbeans*, *XAMPP*, *Astah Community*.

As necessidades de usabilidade do *software* influenciaram na escolha da tecnologia para o desenvolvimento do projeto: o acesso pela *Internet* para consulta de informações possibilitadas pelo desenvolvimento *WEB*; a disponibilização das informações sem a necessidade de instalação de *software* no computador.

A variedade de equipamentos eletrônicos disponibilizados com acesso a *Internet* faz com que o desenvolvimento *WEB* aumente a cada dia, sendo mais um atrativo para o desenvolvimento, Pressman (2011, p.29), inicia seu livro com essa afirmativa,

“A visão da velha escola de *software* – você compra, é seu proprietário e é o responsável pelo seu gerenciamento – está chegando ao fim. Atualmente com a *WEB 2.0* e a computação pervasiva, que vem surgindo com força, estamos por ver uma geração completamente diferente de *software*. Ele será distribuído via *Internet* e parecerá estar residente nos dispositivos do computador de cada usuário... Porém, estará residente em um servidor bem distante”.

A visão de Pressman (2011) demonstra que o desenvolvimento *WEB* trás algumas vantagens com relação ao desenvolvimento dos *softwares Desktop*. Vantagens que se iniciam nas constantes alterações e atualizações que ocorrem em decorrência do grande crescimento e aparecimento de novas tecnologias, até por questões de manutenção, caso haja necessidade, uma vez alocado em um servidor, as atualizações e correções podem ser feitas sem a interação ou intervenção do usuário, mas que indiretamente acabam por refletir melhorias para dezenas, centenas ou milhares de usuários, dependendo do alcance do *software*.

O aumento de demanda por tecnologia reflete diretamente na necessidade de se desenvolver de maneira rápida, fácil e mais barata possível, porém sem perder qualidade no produto desenvolvido, e que para chegar a esse resultado não necessite de supercomputadores que demandem muita tecnologia, ou seja, qualquer pessoa com um pouco de conhecimento possa programar em casa mesmo com o seu computador pessoal, conforme Pressman (2011).

Alguns princípios devem ser observados para a programação *WEB*, estes princípios envolvem a estruturação, a padronização e o uso das ferramentas de validação e integração de tecnologias, a lista abaixo fornece um bom entendimento do que se busca em desenvolvimento *WEB* como resume Loudon (2010).

London (2010) destaca ainda que para aplicações *WEB* é necessário componentes modulares, reutilizáveis, sustentáveis e confiáveis, além da importância da orientação a objetos em *JavaScript*.

Seguindo o modelo dos princípios de desenvolvimento de Loudon (2010), optou-se pela utilização das seguintes ferramentas de desenvolvimento *WEB*: *HTML*, *CSS*, *JavaScript*, *PHP*, e *MySQL*.

As aplicações *WEB* ocorrem por meio de um navegador, ou *Browser* onde o usuário irá digitar a *Universal Resource Location* (URL), que será o endereço da página que se deseja acessar para localização dos arquivos ou informações solicitados, o usuário envia uma requisição ao servidor, sendo gerada uma conexão, havendo autorização do servidor, o usuário passa a receber e carregar todo o conteúdo referente a solicitação efetuada, essa conexão é do tipo *Transmission Control Protocol /Internet Protocol* (TCP/IP), um protocolo para que dois equipamentos consigam comunicação pela *internet*, o formato de leitura destas informações é por Linguagem de Marcação de Hipertexto, ou *HyperText Markup Language* (*HTML*), segundo Bruner e Scheibe (2007).

O *HTML*, que mesmo sendo uma linguagem de marcação muito utilizada na criação de *sites*, passou a ser utilizada por muitos programadores como um padrão de desenvolvimento, pois além de ser de fácil manutenção, não necessita de programas ou *softwares* específicos para desenvolvimento, bastando apenas ter um computador, um *browser* e um editor de textos, como o bloco de notas, por exemplo para usuários que utilizam o Windows como sistema operacional, como destaca Quierelli (2012), além da vantagem de ser de fácil entendimento e em constantes atualizações. Fowler (2003) destaca sua utilização também torna a possibilidade de manutenção do código com uma maior facilidade sem alterações no layout e com um código mais enxuto, uma vez que se altere somente um objeto essa alteração refletirá em todas as telas.

Sua função é determinar a marcação e estruturação de apresentação do conteúdo das páginas, além da possibilidade de interligação e a criação de sequências de páginas, basicamente se divide em quatro conceitos fundamentais: elemento, etiqueta, atributo e valor. A função do elemento é criar uma estrutura (parágrafos, listas, tabelas, imagens, etc.) que irão compor o *HTML*. As etiquetas servem para determinar ou delimitar o início e o fim de determinado elemento. O atributo são as características do elemento, que são configurados com um determinado valor. Segundo Pimentel (2013), como é demonstrado sua estruturação na figura 2.

```
<HTML>
<HEAD>
<TITLE>A minha primeira página</TITLE>
</HEAD>
<BODY>
</BODY>
</HTML>
```

Figura 2. Exemplo de estrutura *HTML* básico (Pimentel 2013).

Como o *HTML* foi criado para fornecer as informações ao usuário, não cabe a ele fornecer informações de elementos, isso é uma função das folhas de estilo composta por camadas, ou *Cascading Style Sheets* (*CSS*) é também a sua maior finalidade segundo Silva (2012), fazendo com que as *CSS* assumam a responsabilidade pela formatação de

cores de fonte, tamanhos de texto, aspectos visuais de um documento, estilos de textos disposição e conteúdos e imagens como destaca Quierelli (2012). O uso de CSS possibilita a retirada das marcações fazendo com que toda e qualquer declaração seja função de apresentação.

Quierelli (2012) destaca ainda que para produção de conteúdos profissionais é fundamental o uso de CSS, aliados as possibilidades de tecnologias do *Personal Home Page (PHP)* originalmente *Hypertext Preprocessor*, que fica responsabilizado pela inserção de conteúdos dinâmicos e do acesso ao banco de dados.

O uso do *PHP* proporciona várias vantagens, torna um programa ou uma página de fácil manutenção, ele trabalha incorporado ao *HTML* e sua utilização reduz a quantidade a de linhas de comando para executar as funções, conforme a afirmação de Spyer (2007, p.100), “A mesma função exige 13 linhas em ASP e quatro PHP”.

Mais uma das vantagens da utilização do *PHP* está na constante manutenção e aperfeiçoamentos pelos seus desenvolvedores, e por se tratar de uma ferramenta *Open Source*, a existência de fóruns na *internet* permite o programador esclarecer dúvidas e solucionar problemas que muitas vezes já foram enfrentados por outros desenvolvedores, a utilização do código fonte aberto cada vez mais agrega adeptos ao seu uso, que apesar de não ser uma solução para todos os problemas de desenvolvimento, mas consegue agregar muito pois afinal é feito por desenvolvedores *WEB* para desenvolvedores *WEB*, como destaca Bruner & Scheibe (2007).

Além de ser uma ferramenta de simples aprendizagem, oferece vários recursos a programadores profissionais, outra característica e talvez a mais forte e significativa seja o suporte a vários tipos de banco de dados, o que torna um atrativo maior para o desenvolvimento, além da possibilidade de portabilidade com banco de dados já existentes, apenas alterando algumas linhas de comando segundo seu *site* oficial (Comunidade PHP.net, 2013).

O *PHP* também é a tecnologia utilizada por uma grande quantidade de servidores *WEB* sendo encontrado na grande parte das vezes trabalhando com o sistema de gerenciamento de banco de dados (SGBD) *Structured Query Language (SQL)* sendo um dos mais populares o *MySQL*, devido a sua portabilidade, compatibilidade, desempenho e estabilidade, como definem Bruner & Scheibe (2007), a compatibilidade e relacionamento entre *PHP* e *MySQL* é tanta que Jobstraibizer (2010, pg.21) criou uma seção em seu livro com o título “*MySQL* e *PHP*: uma história de amor”, onde cita uma pesquisa onde 90% da preferência dos participantes é pelo *MySQL*, conforme vemos na tabela abaixo:

Tabela 1. Larga Vantagem para o MySQL (Jobstraibizer 2010)

SGBD pesquisados	Resultados
Preferem o <i>MySQL</i>	90%
Preferem o <i>Postgre SQL</i>	4%
Preferem o <i>Oracle</i>	2%
Preferem Outros	4%

O *MySQL* também é de fácil portabilidade para os vários dos principais sistemas operacionais utilizados, entre os suportados atualmente como *Windows*, *Linux*, *FreeBSD* (e outros sistemas BSD), *Solaris* e até mesmo o *Mac OS*. Sua popularidade é tanta que estimasse mais de 10 milhões de usuários no mundo fazem uso do *MySQL*, dentre eles alguns usuários estão *NASA*, *Friendster*, Banco Bradesco, *Dataprev*, *HP*, *Nokia*, *Sony*, *Lufthansa*, *U.S Army*, *US. Federal Reserve Bank*, *Associated Press*, *Alcatel*, *Slashdot*, *Cisco Systems*, *Google CanaVialis S.A.* e outros, conforme pesquisas no *site* (Oficina da Net 2013).

Para executarmos a aplicação temos a necessidade de um servidor *WEB*, a forma mais prática de instalação é a utilização do pacote de componentes *XAMPP*, entre suas vantagens estão fácil instalação possui o servidor necessário além do *Apache*, *MySQL*, *PHP* e *Perl*, e ainda funcionar em diversos sistemas operacionais como *Windows*, *Linux* e *Mac OS* segundo seu *site* oficial (*XAMPP*, Apache Friends 2013).

Para modelagem do banco de dados e criação do diagrama de entidade e relacionamento (DER) conforme apêndice E, e manipulação de tabelas do banco de dados, foi utilizada a ferramenta *MySQL Workbench* versão 5.2 CE, que além de ser um *software* livre desenvolvido pela empresa *Oracle Corporations*, sua utilização permite o desenho e a criação dos diagramas que representam a visualização e uma melhor compreensão do banco de dados, também possui a vantagem de gerar o código *SQL* para criação do banco de dados no SGDB, ele também permite a possibilidade de verificação de modelagem de banco de dados já existentes, e ainda possibilita a sincronização caso haja a necessidade de alterações nas tabelas já existentes no banco, segundo Rosa (2011).

No desenvolvimento do código, foi utilizada a ferramenta *NetBeans IDE* versão 7.3.1 (*Integrated Development Environment* - Ambiente Integrado para Desenvolvimento) que, “é um *software* gratuito, e também possui recursos poderosos para tornar o desenvolvimento mais ágil e prático”, segundo Rosa (2011, p.42), outra qualidade está no uso da função auto completar o que acaba reduzindo o tempo gasto no desenvolvimento do código, torna possível a visualização das camadas de desenvolvimento, além de auxiliar na “capacitação inicial ao desenvolvedor que não possuía prática no desenvolvimento”, complementa Oliveira (2008).

As aplicações com as ferramentas independem da utilização de um *Framework* específico, utilizando-se do paradigma de Programação Orientado a Objeto (POO), onde o reaproveitamento de várias classes, promovido pela organização da arquitetura, pois todas as ferramentas utilizadas proporcionam a POO, segundo Bruner & Scheibe (2007), Loudon (2010, p.17) complementa, “Você verá que a maioria das técnicas é de aplicação relativamente simples e que nenhuma depende da utilização de *frameworks* específicos”. Essas afirmações fizeram com que durante o desenvolvimento um *Framework* fosse sendo moldando conforme eram encontradas necessidades no projeto.

Ainda segundo Dall’Oglio (2009) com relação ao paradigma POO, a intenção é programar com uma relação mais próxima possível do mundo real, se estruturando e tratando de objetos que já conhecemos, essas comparações proporcionam um melhor entendimento da estrutura do sistema, os conceitos são: o uso de classes, objetos, atributos, métodos, mensagem, herança, associação, encapsulamento, abstração,

polimorfismo, interface e pacotes. O objetivo desse conceito é criar um conjunto de classes que irão conter objetos, cada objeto terá seus atributos, que serão acessados através de métodos, a herança é o mecanismo que tornará uma classe estendível a outras classes, dessa maneira disponibilizando esses objetos e efetuando o reaproveitamento de códigos.

Para entendimento do sistema utilizando os padrões de POO, é necessário a implantação dos padrões da linguagem unificada de modelagem, ou *Unified Modeling Language (UML)*, fazendo com que cada elemento seja entendido pelo desenvolvedor, independente da ferramenta utilizada, ou da linguagem utilizada para o desenvolvimento, com a análise dos diagramas, qualquer programador consiga ter entendimento do que está propondo a modelagem por meio de gráficos, dessa maneira expondo as necessidades do sistema antes mesmo de começarmos o desenvolvimento segundo Bezerra (2007), porém para o desenvolvimento a necessidade de primeiramente compreendê-lo, para aplicar as ferramentas que mais se adequa ao processo, economizando assim tempo e pessoal no desenvolvimento, como define Pressman (2011). Para a aplicação e desenvolvimento de *UML*, foi utilizada a ferramenta *Astah Community*, que além de ser gratuita é uma das mais utilizadas pela capacidade de trabalhar com vários formatos de diagramas ainda de forma intuitiva e complexa ao mesmo tempo segundo Lahoud *et al* (2011).

Para o início do desenvolvimento do projeto, foi optado pelo ciclo de vida incremental, que combina processos lineares e paralelos, onde os incrementos funcionam como protótipos do sistema, a avaliação pode ser feita pelo cliente com a entrega de partes do sistema, o que permite solucionar falhas logo no início do projeto, além de novos requisitos podem ser incorporados no sistema, priorizando o que será desenvolvido primeiro onde cada incremento é independente, diminuído os riscos de fracasso do sistema, pois as funções são testadas antes da entrega final do software, o ciclo de vida incremental tem muita importância num desenvolvimento que não disponibiliza de muito pessoal e o prazo de entrega curto, possibilitando a liberação parcial do projeto sem causar um atraso excessivo no prazo final como descreve Pressman (2011) demonstrado abaixo (figura 3).

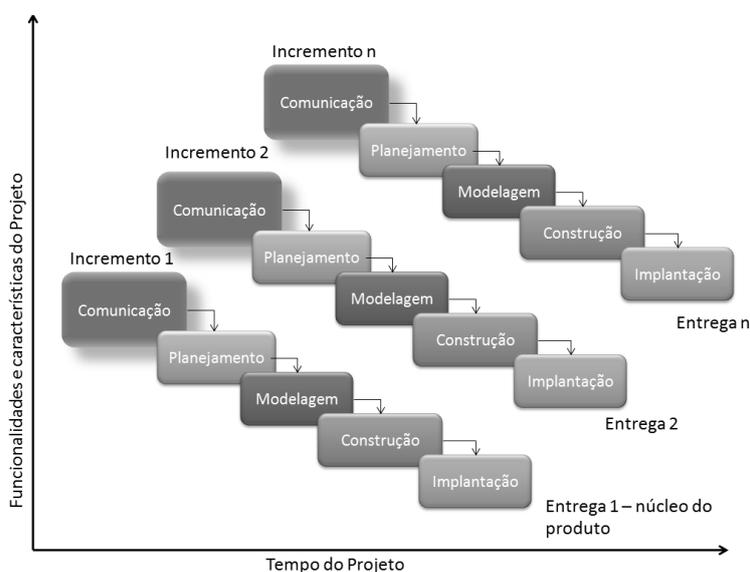


Figura 3. Ciclo de vida Incremental (Pressman, 2010).

3. Desenvolvimento

Na primeira fase do projeto de desenvolvimento foi efetuado um *Briefing* preliminar com a coleta de informações pelos métodos de pesquisa e observação, um questionário foi desenvolvido e aplicado com relação às necessidades e dificuldades do cliente, e um levantamento das ferramentas que seriam utilizadas, além da análise e observação do das ferramentas utilizadas pelo cliente, demonstrado no apêndice A.

Na segunda etapa do projeto, foi efetuado a análise dos requisitos e pesquisas e levantamentos sobre o tema proposto pelo cliente, foram observados alguns softwares existentes no mercado e disponíveis na *internet*, efetuados teste quanto às possibilidades e funções de alguns deles até se definirem as necessidades, optado pelo desenvolvimento de um modelo DFC, que é o que melhor se adequa as necessidades do cliente.

Com os requisitos do sistema definidos, iniciou-se a fase de modelagem de *UML* e especificações do sistema e elaboração dos diagramas para o desenvolvimento, utilizando-se da ferramenta *Astah Community*, foram criados os diagramas de caso de uso, diagrama de classes, e sequência, dispostos nos apêndices B, C e D.

Foi definido o DER e modelagem do SGBD, utilizando a ferramenta *MySQL Workbench*. Dentro do ciclo de vida incremental a ferramenta demonstrou a sua capacidade de manipulação e sincronização com o banco de dados em alterações que foram necessárias no decorrer do desenvolvimento na estruturação de tabelas e relacionamentos, o DER está disponível no apêndice E.

Na próxima etapa foi o desenvolvimento de layout de tela, considerando os conceitos de Interação Humano-Computados (IHC), que trata da avaliação e implementação de sistemas computacionais interativos para uso humano, ou seja, a maneira e disposição das ferramentas para utilização do homem, tornando essa interação o mais agradável possível, Carvalho (1994).

Foi desenvolvida a prototipagem inicial das telas, expondo as diferentes atividades executadas além de alguns aspectos de *design* físico, foram gerados esboços a mão em papel como alternativas de disposição de telas, sub-telas e *menus*. A presença do cliente e usuários do sistema possibilitou uma melhor avaliação estrutural e posicionamento dos campos na tela, demonstrados nos Apêndices de G, H, I, J e K.

Com o desenvolvimento das etapas anteriores do projeto, deu início ao modelo estrutural do projeto e a arquitetura do *software*, o padrão aplicado no desenvolvimento é o *Model View Controller* (MVC), que basicamente é dividido em três camadas e cada camada fica responsável por uma ação dentro do sistema, conforme Apêndice F. A *Model* (modelo) fica responsável pela persistência com o banco de dados através dos métodos *create, read, update and delete* (CRUD), ficando com a função de criar, ler, atualizar e excluir dados do banco. A *View* (visão) é a camada responsável pela interação direta do usuário com o sistema, através da *URL* o usuário faz a solicitação do conteúdo da página, esta solicitação é direcionada então para a camada *Controller* (controlador) que analisa a entrada e o tipo de informação solicitada, definindo qual a ação a ser tomada e o tipo de saída de dados, solicitando a camada *Model* o retorno da CRUD retorna ao usuário por meio da *View* os dados dispostos em *HTML*, dividindo

assim as aplicações sem a necessidade do usuário saber como ocorre essa interação de dados.

O próximo passo foi a construção do arquivo *Index.php*, que será o responsável pelo acesso direto ao sistema, e o desenvolvimento do *Template* para o carregamento da camada *View* e instância da classe *App*, que é a responsável pelas solicitações e requisições as camadas *model e controller*.

A classe *App* fica responsável por iniciar as seções carregando os atributos conforme as solicitações pela *URL*, os atributos foram definidos como: módulo, ação e chave e filha. Retornando essa solicitação através da classe *HTML*.

Outra classe que deve ser citada é a *Menu*, que é a responsável pela autenticação ou controle de usuário, através da classe faz uma requisição a *Model* de usuário, que vai definir a quais menus e quais os tipos de dados os usuários terão acesso.

Para agilizar o desenvolvimento, foi utilizado o *Bootstrap*, que é basicamente um pacote de *JavaScript e CSS* reduzindo o tempo de desenvolvimento destas ferramentas, além de auxiliar no *layout* de apresentação para o usuário, e seguindo a arquitetura MVC, foi devidamente alocada dentro da pasta *public/css* e *public/javascript*.

Na nomenclatura das classes foi utilizada a primeira letra da arquitetura do MVC a que determina a qual módulo ela pertence + o nome da classe + o sufixo *.class* + formato do arquivo, por exemplo *cPagar.class.php* e *mPagar.class.php* onde a letra “c” indica que um *controller* da classe pagar em PHP, e a letra “m” indica que um *model* da classe pagar em PHP, nas classe *view* por se tratarem de arquivos *HTML*, foi utilizado somente o nome da função que ela exerce + o formato, como exemplo *cadFavorecido.html*, que é a responsável pela visualização da tela de cadastro de favorecido, ou *favorecidoShowLinha.html* que é a responsável pela exibição do conteúdo das linhas que serão exibidas na tabela do cadastro de favorecidos.

Com a possibilidade de divisão dos módulos do ciclo de vida incremental, depois de determinado o padrão de arquitetura, as classes, quais a funções de cada classe, foi desenvolvido o *Wireframe* da tela inicial de cada *menu*, sendo elabora uma visita ao cliente para demonstração da prototipagem inicial, para a aprovação do cliente e possíveis alterações no contexto e tipo de dados, bem como apresentação desses dados na tela, já com persistência em banco de dados, conforme apêndices L e M.

Aprovado o primeiro protótipo, foi solicitada apenas que fosse acrescentado ao sistema no *menu* de lançamentos, tanto de contas a pagar como contas a receber, um demonstrativo de contas que apresentasse um acumulado de contas vencidas, vencendo no dia, e a vencer.

Para acrescentar essa função foi criada uma tela *modal*, que é chamada através de um botão, onde retornam uma pesquisa ao banco conforme descrita acima, foi acrescentado na *model* de cada uma das classes a função de soma das contas ascendente, do dia atual e descentes, que foi facilmente implementada sem a necessidade de grades alterações no sistema devido a sua arquitetura MVC.

Na tela principal (*Home*) foi proposto uma a visualização de um resumo de entradas, saídas, e um comparativo gráfico em barras que pudesse apresentar durante o

mês consultado o pico de entradas e saídas do mês, para observar quais os dias de maior fluxo de entradas e saídas.

Foram propostos os relatórios de: contas a pagar e contas a receber apresentando: o número do documento, a data de vencimento, o valor, favorecido e *status* se estão pagos ou a pagar. Bem como os relatórios de usuários do sistema, favorecidos, que neste caso podem ser fornecedores ou credores, que utilizam a mesma tabela no banco de dados.

Seguindo o ciclo de vida incremental e depois de efetuadas as atualizações propostas pelo cliente, o projeto partiu para última etapa onde foi colocado em uso para testes no sistema pelo cliente, durante um período de cinco dias, onde foram efetuados testes de persistência com o banco de dados, e usabilidade, que atendeu as expectativas que foram propostas sendo aprovado pelos usuários do sistema.

4. Resultados

Durante o desenvolvimento do presente trabalho, pôde-se observar a importância do demonstrativo de fluxo de caixa, como forma de avaliar o potencial financeiro de uma empresa, proporcionando a possibilidade de planejar ações futuras com o auxílio da análise dos resultados atingidos durante um determinado período. Também foi possível perceber que através desta ferramenta é possível demonstrar a magnitude da empresa honrar e cumprir seus compromissos, que podem ser comprovados com o histórico de todas das movimentações financeiras.

A utilização de ferramentas *open source*, demonstrou a possibilidade de desenvolvimento de atividades sem que haja a necessidade de investimentos em *software* ou aplicações pagas. O uso das chamadas ferramentas livres, possibilita que qualquer pessoa com uma base de aprendizado consiga desenvolver controles independente da sua dimensão, além da disponibilização de auxílio nas mais variadas formas, desde a criação das modelagens até a sua integralização. Graças a vários *sites* e fóruns relacionados ao assunto que acabam tornando o uso de ferramentas deste porte cada vez mais popularizadas.

Dentro do ciclo de vida incremental ficaram muito claras as vantagens da arquitetura MVC, bem como do paradigma da orientação a objeto, onde nas modificações que foram aplicadas no decorrer do ciclo apenas houve a necessidade de aumentar algumas linhas de código nas classes já existentes, sem que fosse preciso reestruturar todo o sistema, ou criar novas classes, uma vez que os objetos já estavam criados apenas lhes foram atribuídos algumas novas características e condições.

O uso do *MySQL Workbench* também apresentou suas vantagens, pois as alterações na estrutura do banco de dados, foram feitas de maneira muito rápida e sem demandar a necessidade da criação de códigos *SQL*, a importância da sincronização e confiabilidade passada pela ferramenta superou as expectativas, além de proporcionar a criação do diagrama de estrutura e relacionamento entre as tabelas, exibindo o gráfico de fácil entendimento, reduzindo também o tempo entre o projeto e o desenvolvimento do código.

A escolha do desenvolvimento de uma aplicação *WEB* para o sistema, do mesmo modo evidenciou o seu benefício por estar acessível em qualquer computador ou

dispositivo com acesso a um navegador *WEB* dentro da *Local Area Network* (LAN), além da alternativa de disponibilização e acesso pela *internet* e sem a necessidade de instalação o que viabiliza a manutenção.

O incremento de um *Framework* igualmente pode acrescentar vantagens ao trabalho, pois evitou o uso de códigos e funcionalidades que não estariam presentes nessa concepção, tornando a estrutura simples e funcional.

Todas as ferramentas utilizadas neste projeto, apesar de não utilizado na totalidade os seus recursos, dentro do objetivo que foi proposto, possibilitaram a redução no tempo para a execução da tarefa de resgate dos dados financeiros além de tornar o desenvolvimento com custos reduzidos, simples, funcional e acima de tudo qualidade e profissionalismo.

5. Considerações Finais

Dentro do objetivo proposto, desenvolver um *software* capaz de cadastrar a movimentação financeira da empresa em um banco de dados, apresentando os resultados financeiros para auxiliar na tomada de decisões financeiras de fácil interpretação e com custo baixo que auxilie no acompanhamento do movimento financeiro considerando a empresa como de médio porte com potencial de crescimento e monitorando principalmente as despesas, o demonstrativo de fluxo de caixa conseguiu atender a necessidade proposta no projeto.

Seu uso contribuirá para que empresas possam entender o paradigma da movimentação financeira, respondendo a pergunta de onde veio e para onde foi o dinheiro da empresa, possibilitando tomadas de decisões baseadas em resultados precisos, visando melhores resultados.

Em breve, o objetivo é estender as funções do *software* disponibilizando consulta individuais por contas e também criando possibilidades de comparativos estatísticos. Também investir em portabilidade com base de dados de softwares com módulos financeiros já existentes, possibilitando ser agregado em sistemas já existentes.

Referencias Bibliográficas

APACHE FRIENDS – XAMPP – Agosto de 2013. Disponível em http://www.apachefriends.org/pt_br/xampp.HTML. Acessado em 18/10/2013.

BASTOS, Kátia Henrique; QUINTANA, Alexandre Costa; JACQUES, Flávia Verônica Silva. **Análise da demonstração dos fluxos de caixa da empresa Wilson Sons Limited e subsidiárias por meio de indicadores** – Universidade de Santa Cruz do Sul, SC julho de 2011. Disponível em http://www.unisc.br/portal/upload/com_arquivo/tc20_328554648071.pdf acessado em 11/10/2013.

BEZERRA, Eduardo – **Princípios de Análise e Projeto de Sistemas com UML** - 2007 – Elsevier Editora Ltda. Rio de Janeiro – RJ

BORGES, António; RODRIGUES, Azevedo; RODRIGUES, Rogério; RODRIGUES, José – **As Novas Demonstrações Financeiras**. Lisboa, Novembro 2007: Áreas Editora. ISBN 978 – 989 – 8058-195.

- BRUNER, Rodrigo; SCHEIBE, Willian Andrews. **Portal Multimídia com WEB 2.0**. Sociedade Educacional de Santa Catarina – SOCIESC – Instituto Superior Tupy – IST. Joinville – Santa Catarina, 2007 – Disponível em <http://www.cpgmne.ufpr.br/redeticbr/index.php?option=com_phocadownload&view=category&download=75:2007-2-portal-multimidia-com-a-WEB-2.0-rodrigo-bruner-e-willian-andrews-scheibe&id=15:gesto-de-conhecimento&Itemid=89>. Acessado em 08/10/2013.
- CARVALHO, J.O.F. **Referenciais para projetistas e usuários de interfaces de computadores destinadas aos deficientes visuais**. 1994. Dissertação (Mestrado em Engenharia Elétrica) – Faculdade de Engenharia Elétrica, Universidade Estadual de Campinas, Campinas, 1994. Disponível em: <<http://docentes.puc-campinas.edu.br/ceatec/oscar>>.
- COMUNIDADE PHP.NET – **Introdução ao PHP** – Disponível em <http://www.php.net/manual/pt_BR/intro-whatism.php>. Acessado em 15/06/2013.
- DALL’OGLIO, Pablo. **PHP - Programando Orientado a Objetos**. Editora Novatec – 2007 – São Paulo – SP.
- FERREIRA, Neide de Souza – **A importância da Gestão do Fluxo de Caixa no Processo Decisório das Empresas**. Universidade Federal da Paraíba – Centro de Ciências Sociais Aplicadas. 2003. Disponível em <[http://bis.sebrae.com.br/GestorRepositorio/ARQUIVOS_CHRONUS/bds/bds.nsf/E5D4B978975DD98A03256FAC00740E9E/\\$File/NT000307D2.pdf](http://bis.sebrae.com.br/GestorRepositorio/ARQUIVOS_CHRONUS/bds/bds.nsf/E5D4B978975DD98A03256FAC00740E9E/$File/NT000307D2.pdf)>. Acessado em 11/10/2013.
- FOWLER, Martin. **Padrões de arquitetura de aplicações corporativas**. Artmed Editora S/A. – 2003. Porto Alegre – RS
- HOJI, Masakazu. **Administração Financeira e Orçamentária: matemática financeira aplicada, estratégias financeiras, orçamento empresarial**. 8. ed. São Paulo: Atlas, 2010.
- JOBSTRAIBIZER, Flavia – **Criação de Banco de Dados com MySQL** – Universo dos Livros Editora – São Paulo – SP – 2010 disponível em <books.google.com.br/books?isbn=8578731115>. Acessado em 13/10/2013.
- LAHOUD, Yesmin Marie S; PEREIRA, Gislane C.; JUNIOR, José Francisco de A.; DELFINO, Sérgio R.; ORLENDINI, Guilherme – **Jogo Educativo para a Disciplina de Geografia** - Faculdades Integradas de Ourinhos (FIO) – Ourinhos – SP – 2011. Disponível em <<http://revistas.unoeste.br/revistas/ojs/index.php/ce/article/view/718/926>>. Acessado em 20/11/2013.
- LOUDON, Kyle – **Desenvolvimento de Grandes Aplicações WEB** – Editora NOVATEC, 2010.
- MENDES, Nélida - **A importância da demonstração do fluxo de caixa e o paradigma da utilidade** – Portal do Conhecimento do Cabo Verde, Maio de 2009. Disponível em <<http://hdl.handle.net/10961/986>>. Acessado em 08/10/2013.

Oficina da Net – **MySQL – o que é?** 2013. Disponível em <[http://www.oficinadanet.com.br/artigo/2227/MySQL - o que e](http://www.oficinadanet.com.br/artigo/2227/MySQL_-_o_que_e)>. Acessado em 13/10/2013.

OLIVEIRA, Cícero David Leite – **Desenvolvimento de componentes *WEB Services* disponibilizados por meio de *WEB Servers***. Instituto Tecnológico de Aeronáutica – São José dos Campos – SP Outubro de 2008 – Disponível em <<http://www.bibl.ita.br/xivencita/COMP01.pdf>>. Acessado em 20/10/2013.

PIMENTEL, Mariano Gomes – **Construção de Páginas *WEB* com *HTML* – Nível Básico** NCE/UFRJ - Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais. 2013. Disponível em <<http://www.nce.ufrj.br/ginape/cursoHTML/>>. Acessado em 20/10/2013.

PRESSMAN, Roger S. **Engenharia de Software**, Sexta Edição. Editora McGrawHill: Porto Alegre, 2010.

PRESSMAN, Roger S. – **Engenharia de Software – Uma Abordagem Profissional**. Editora Bookman, Sétima Edição. 2011 – Disponível em <<books.google.com.br/books?isbn=8580550440>>. Acessado em 12/10/2013.

QUIERELLI, Davi Antônio – **Criando sites com *HTML* – CSS – PHP. Construindo um projeto iniciante**. Edição do Autor. 1ª Ed. 2012 – Disponível em <<books.google.com.br/books?id=TDZJ2AFV108C>>. Acessado em 13/10/2013.

ROSA, Douglas Eduardo – **Sistema computacional para catalogação e apoio à identificação de unidades de dispersão**. Universidade Federal de Pelotas - Pelotas – RS - 2011 – Disponível em <http://inf.ufpel.edu.br/nopcc/lib/exe/fetch.php?media=monografias:2010:2010-mono-douglas_rosa.pdf>. Acessado em 15/10/2013.

SÁ, Carlos Alexandre de - **Gerenciamento do fluxo de caixa**. Apostila, São Paulo: Top Eventos, 1998.

SILVA, Maurício Samy - **CSS3: desenvolva aplicações *WEB* profissionais com uso dos poderosos recursos de estilização das CSS3**; [tradução Rafael Zanolli]. - Novatec Editora - São Paulo – SP – 2012.

SPYER Juliano – **Conectado: o que a internet fez com você e o que você pode fazer com ela** / Jorge Zahar Editora – Rio de Janeiro – RJ – 2007 Disponível em <<books.google.com.br/books?isbn=8537801917>>. Acessado em 13/10/2013.

TEIXEIRA, Gilmar Claret; COELHO, Rosani - **Referenciais de consultoria SEBRAE / consultores conteudistas**. Brasília - DF SEBRAE, 2012.

APÊNDICE A – Breafing Preliminar

Nome da empresa

Zeus Excelência em Segurança Eletrônica.

Nome do projeto ou campanha

Software Web para auxílio nas movimentações financeiras

Data de realização:

15/07/2013

Tarefa (o que deve ser realizado nesta atividade)

Levantamento de requisitos, questionário e observação.

Objetivos do projeto: (o que você quer alcançar)

Desenvolver um software capaz de cadastrar a movimentação financeira da empresa em um banco de dados, apresentando os resultados financeiros para auxiliar na tomada de decisões financeiras de fácil interpretação e com custo baixo que auxilie no acompanhamento do movimento financeiro considerando a empresa como de médio porte com potencial de crescimento e monitorando principalmente as despesas

Contexto do projeto: (detalhes relevantes do contexto)

O cliente utiliza-se planilhas eletrônicas para executar o levantamento dos dados referentes a fluxo de entrada e saída do caixa.

Os dados são lançados manualmente nas planilhas, não possui um banco de dados.

Estratégia

Desenvolver um banco de dados.

Controle de acesso por senha.

Exibição de gráfico.

Controle de contas a pagar e receber. (Fluxo de Caixa)

Estratégia do projeto: Vantagem competitiva circunstancial:

Desenvolver com software Web devido ao custo para o desenvolvimento, e a possibilidade de acesso pela rede LAN e Internet.

Suporte:

Referencia de sistema o site www.pyme.com.br apresentado pelo cliente.

Cópia da planilha de levantamento utilizada pelo cliente para do desenvolvimento do controle.

Obrigações:

Desenvolver a modelagem das telas para aprovação.

Entrega dos requisitos descritos no campo Estratégia.

Datas chave:

Previsão de entrega de Prototipagem : 02 de Setembro de 2013

Data do briefing:

15/07/2013

Conceitos criativos apresentados pela agência:

Desenvolvimento da tela inicial com exibição de gráfico demonstrando os picos de entradas e saídas de caixa

Participaram da elaboração deste documento:

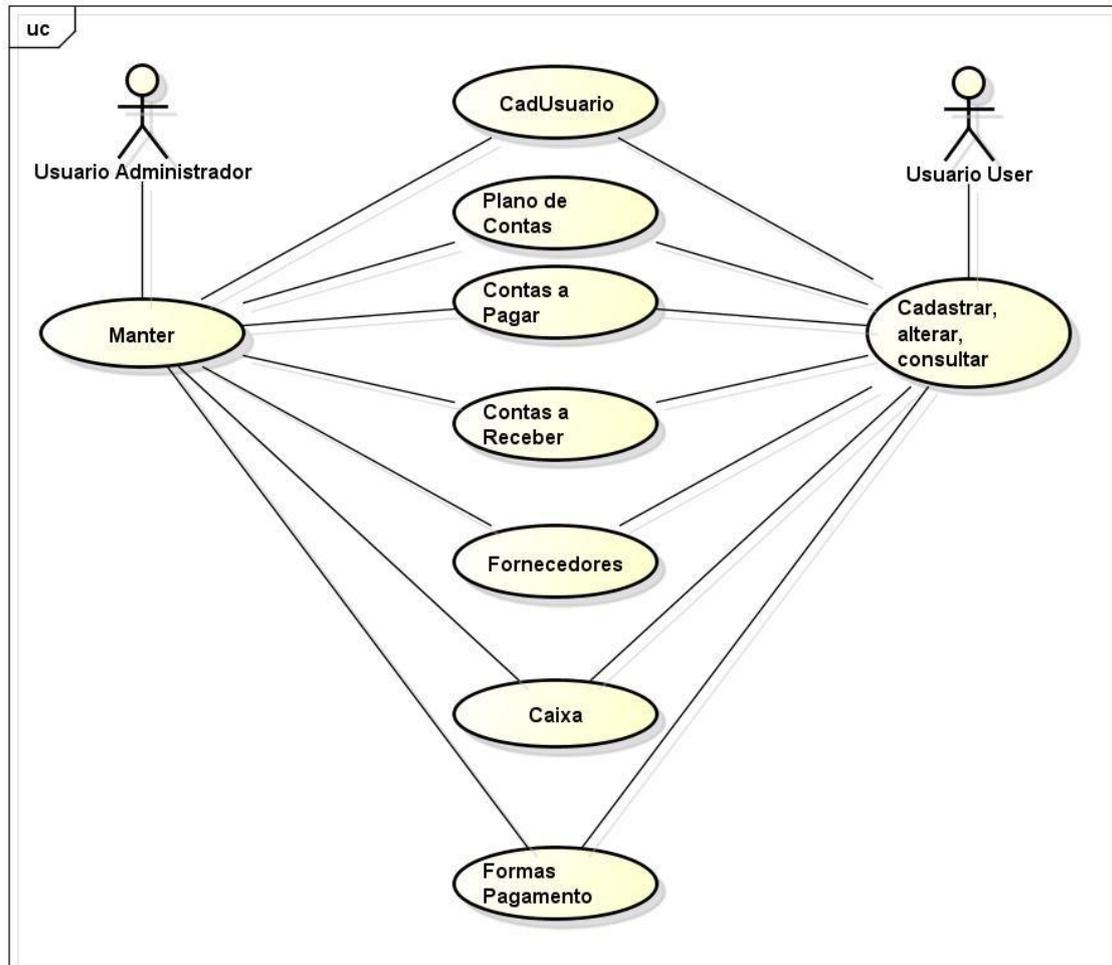
Jean Wilderson Bodot

Técnico responsável pelo desenvolvimento

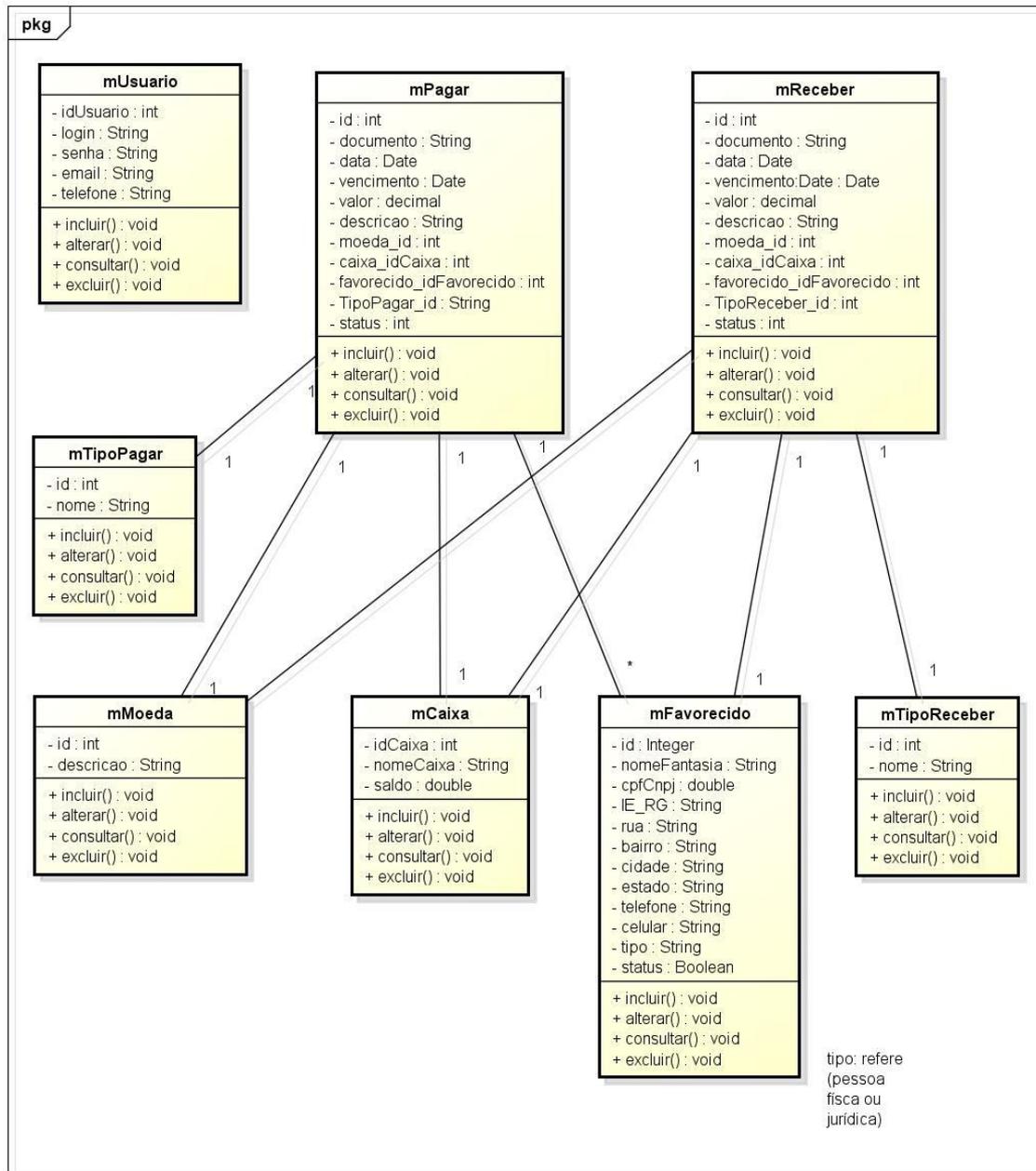
Ricardo Folador – Sócio Proprietário da Empresa.

Guarapuava, 15 de Julho de 2013.

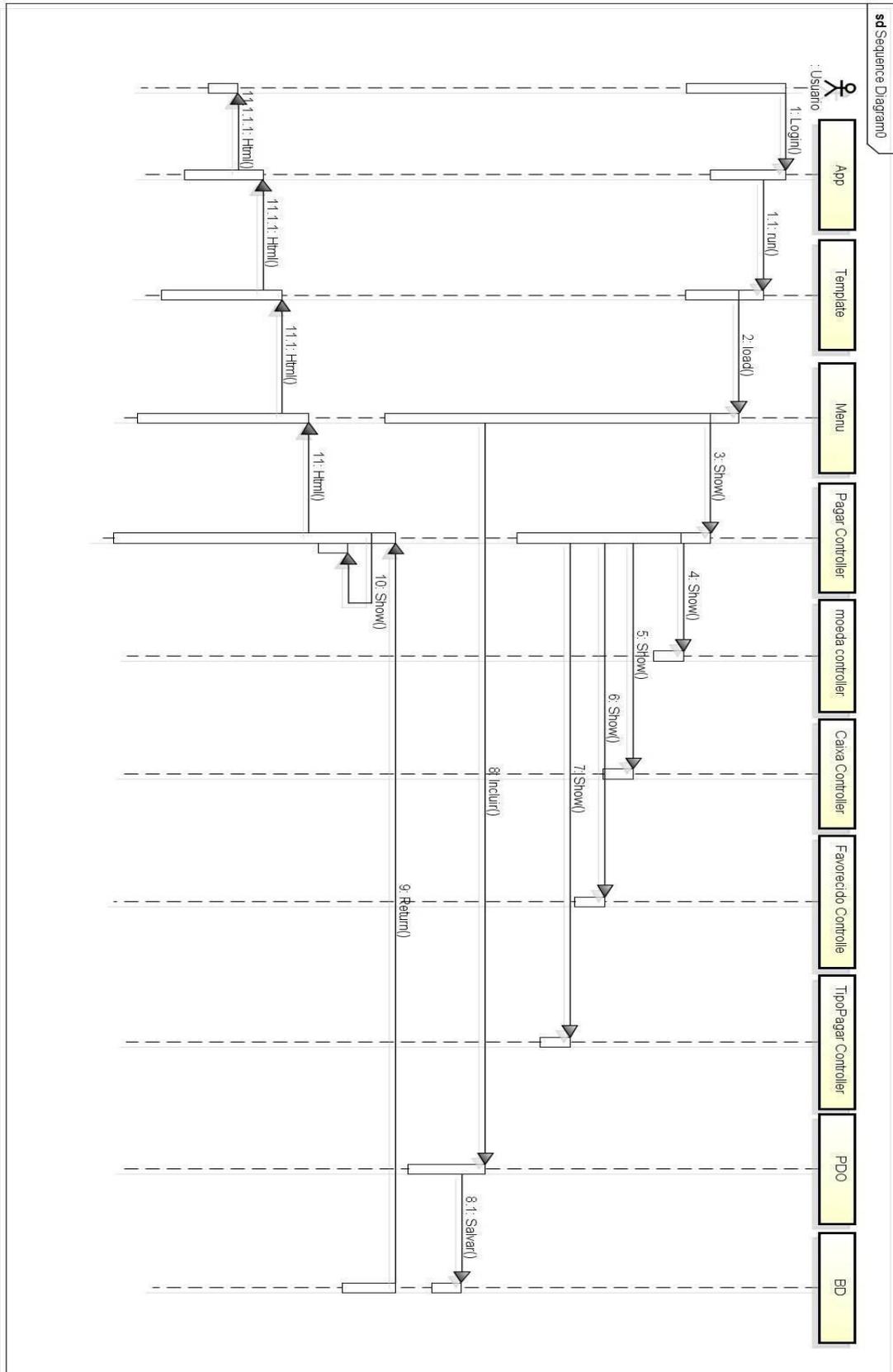
APÊNDICE B – Diagrama de caso de uso



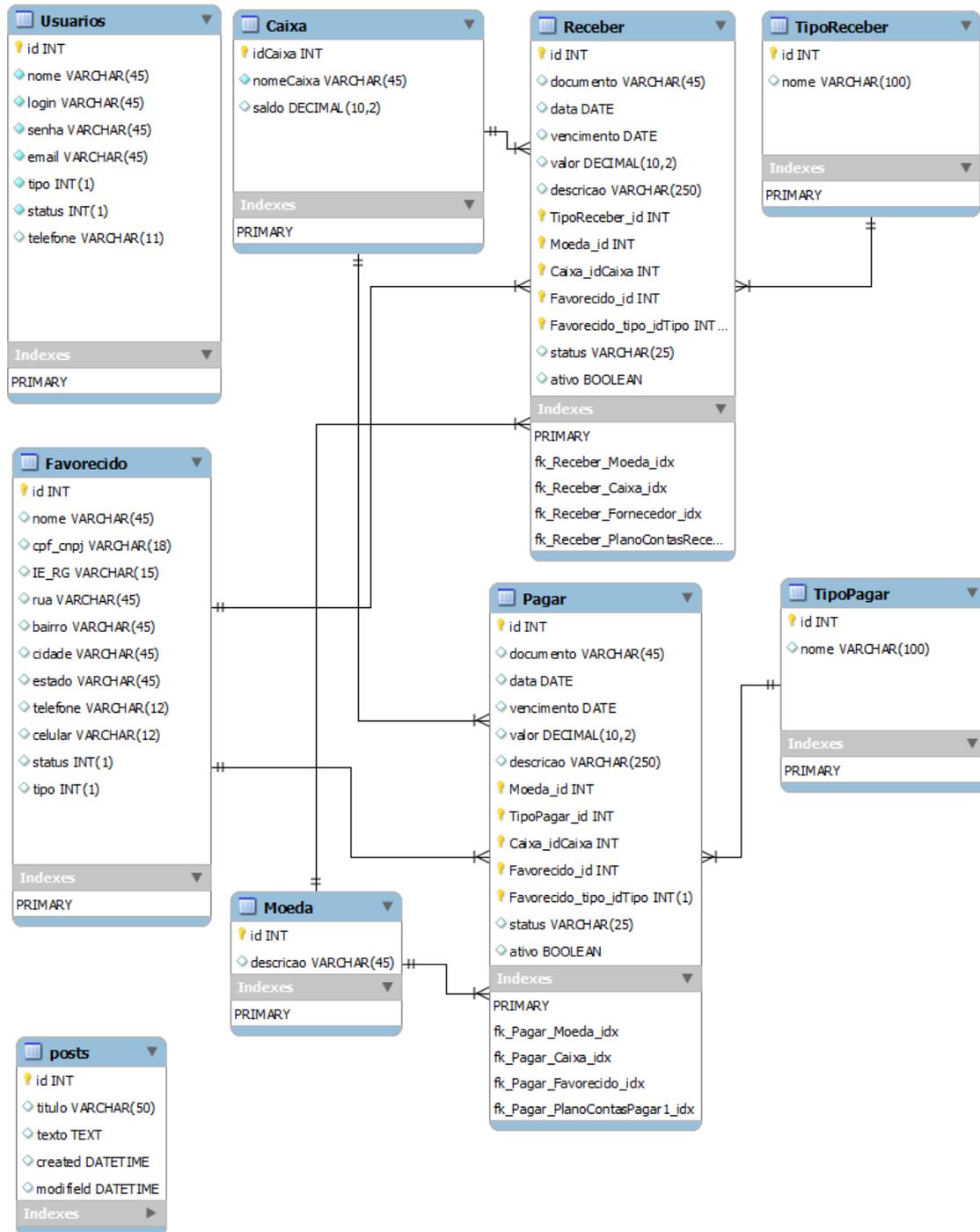
APÊNDICE C – Diagrama de classes



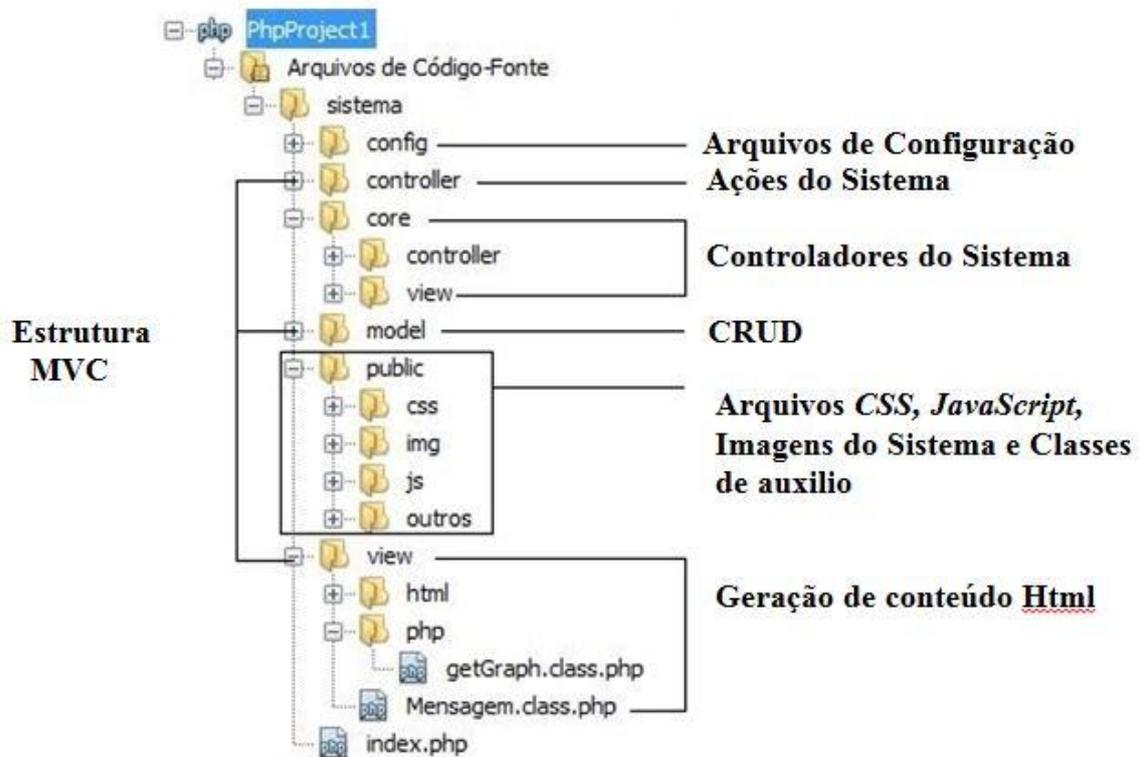
APÊNDICE D – Diagrama de seqüência de cadastro de contas a pagar



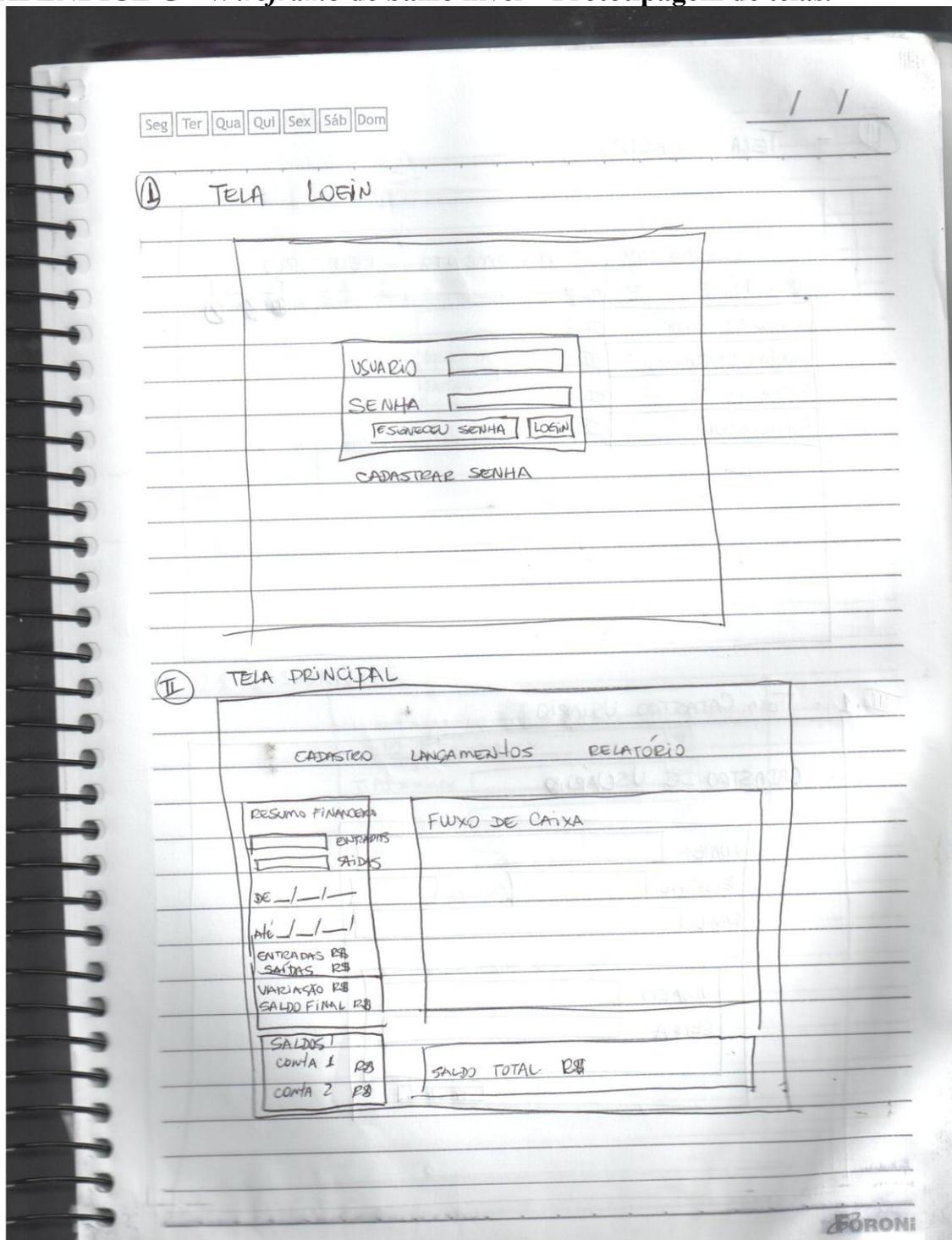
APÊNDICE E – DER



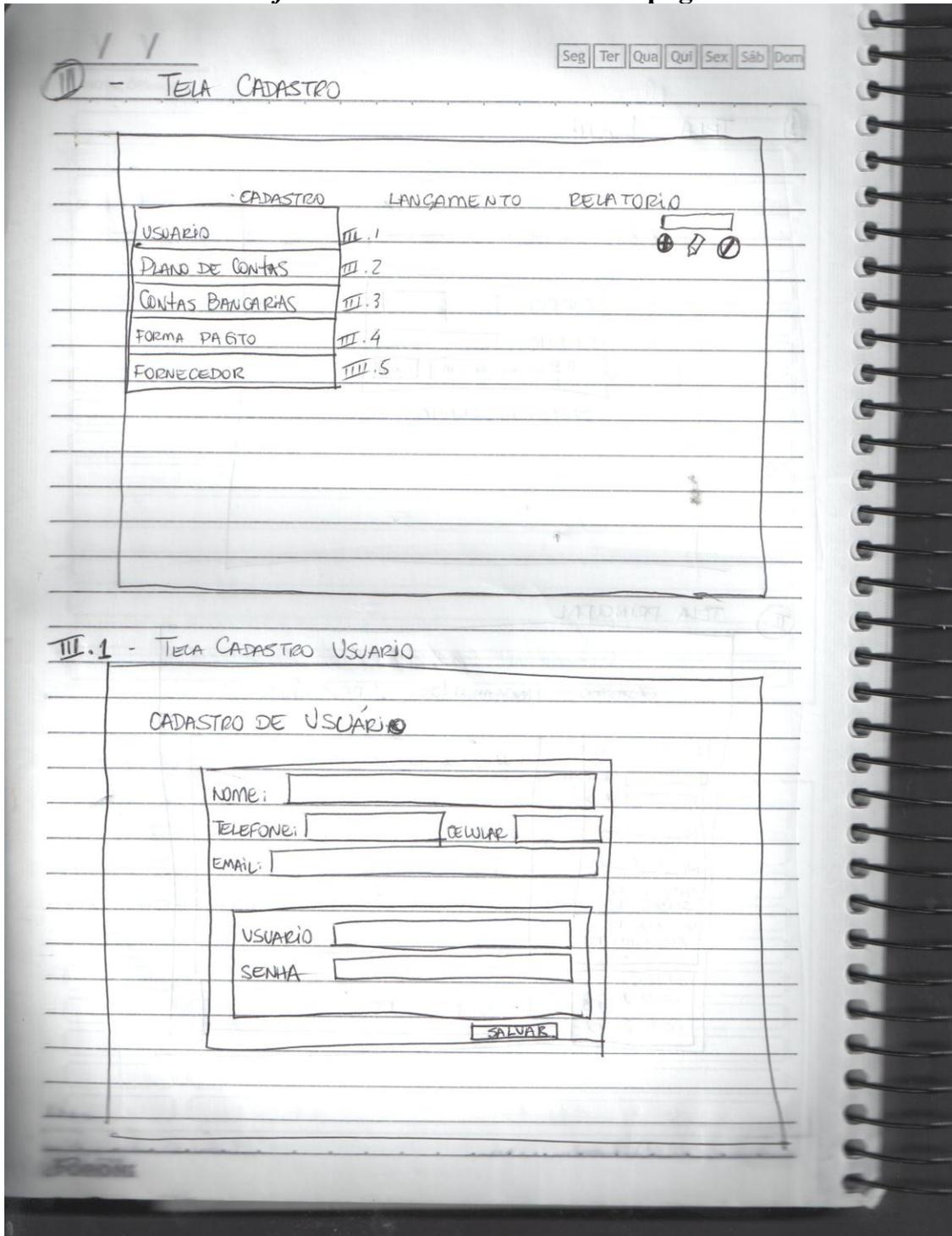
APÊNDICE F – Padrão MVC – Estrutura de Pastas do Sistema



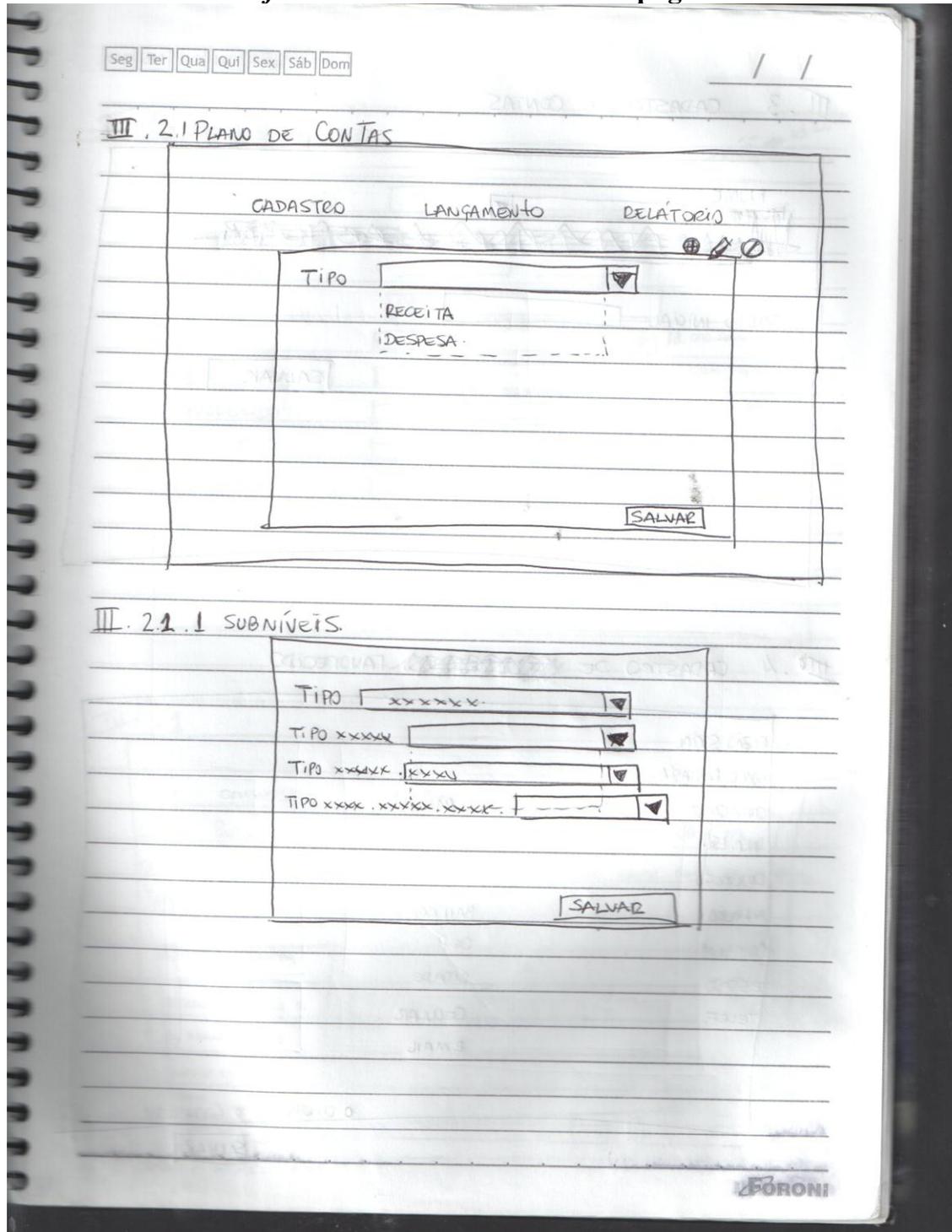
APÊNDICE G – Wireframe de baixo nível – Prototipagem de telas.



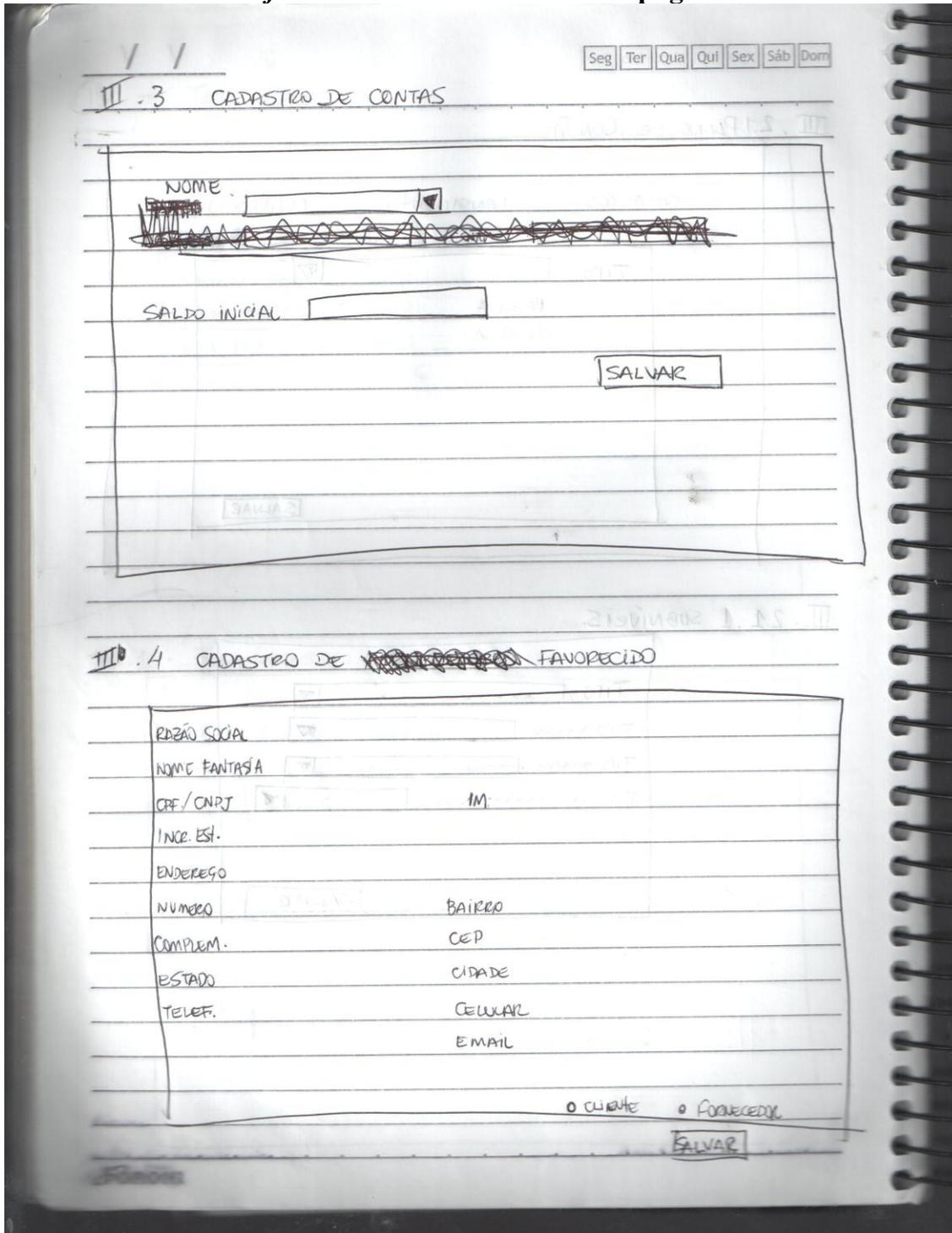
APÊNDICE H – Wireframe de baixo nível – Prototipagem de telas.



APÊNDICE I – Wireframe de baixo nível – Prototipagem de telas.



APÊNDICE J – Wireframe de baixo nível – Prototipagem de telas.



APÊNDICE K – Wireframe de baixo nível – Prototipagem de telas.

Seg Ter Qua Qui Sex Sáb Dom

/ /

IV - 1.1 LANÇAMENTOS

if id = id Lcto

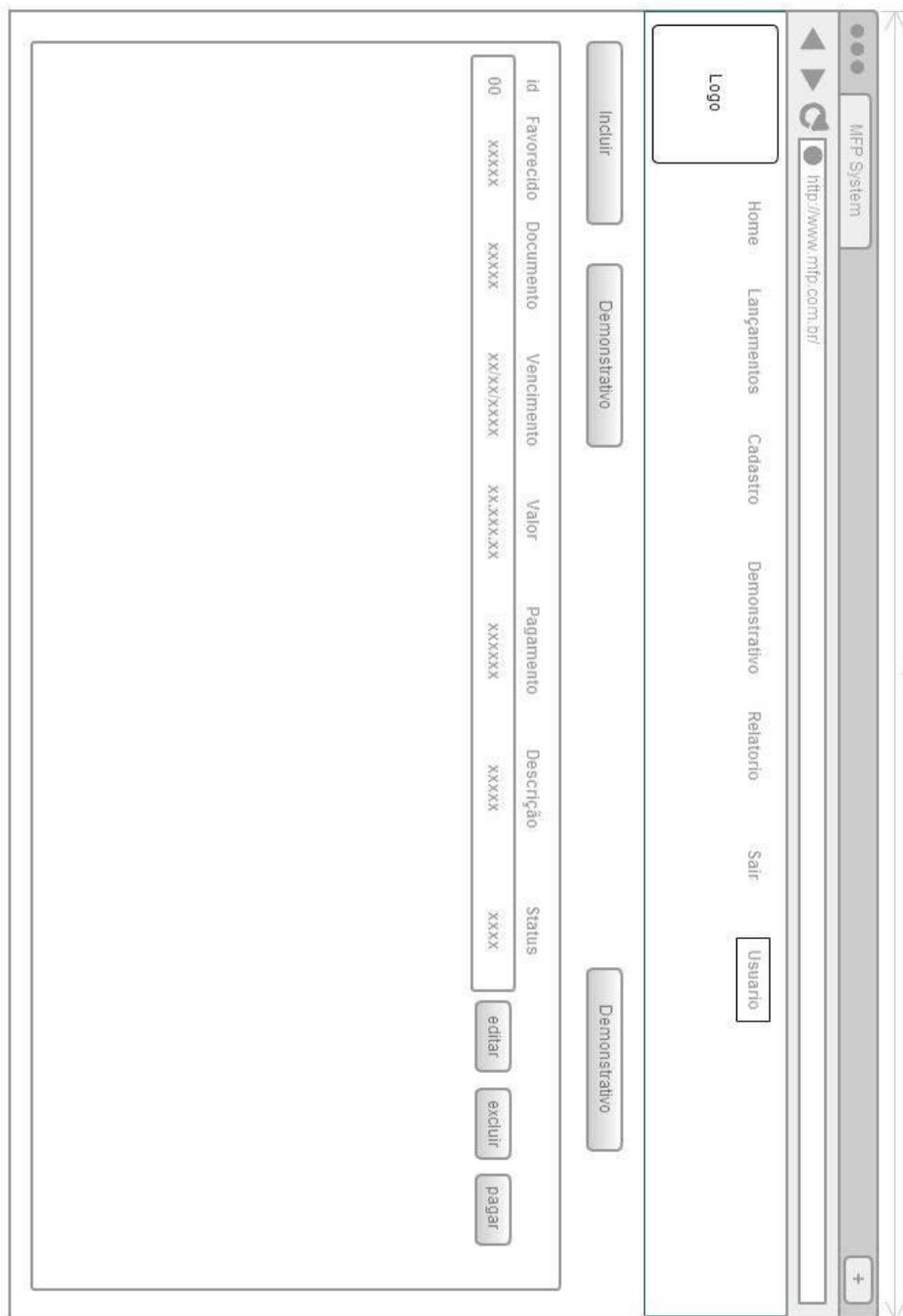
Nº DOCUMENTO	<input type="text"/>	id ID AUT. - INCE.
DATA DO DOCUMENTO	<input type="text"/>	
VENCIMENTO	<input type="text"/>	
VALOR	<input type="text" value="R\$"/>	
CONTA	<input type="text"/>	id. cad. conta Tipo Conta
FORMA PAGAMENTO	<input type="text"/>	id Form. PGTO
FAVORECIDO	<input type="text"/>	id. Favorecido
CONDICÃO	<input type="text"/>	id. COND.
DESCRIÇÃO	<input type="text"/>	STRING:

IV - 1

CASASCAK	LANÇAMENTOS	AGENDA
CONTAS PAGAR	[ID da conta] ⊕ / ⊖	
CONTAS RECEBER	VOTO. PAGAMENTO EMISSÃO DESDE FAVOR. CONTA. VALOR €	
TRANSFERÊNCIAS		
RESUMO DA CONTA		
ATRASADO R\$ 9		
A VENCER R\$		
VENCE HOJE R\$		
PAGO		
TOTAL		

→ + CONDIÇÃO + Nº DOG. CAIXA

APÊNDICE L – Wireframe menus do sistema e tabela de consulta de movimento



APÊNDICE M – Wireframe de cadastro de movimento

