UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ – UTFPR
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

JOSIEL AMARO DA SILVA

DESENVOLVIMENTO DE SISTEMAS COM PRODUTIVIDADE UTILIZANDO FERRAMENTAS VISUAIS DE PROGRAMAÇÃO

TRABALHO DE DIPLOMAÇÃO

MEDIANEIRA
2011

JOSIEL AMARO DA SILVA

DESENVOLVIMENTO DE SISTEMAS COM PRODUTIVIDADE UTILIZANDO FERRAMENTAS VISUAIS DE PROGRAMAÇÃO

Trabalho de Diplomação apresentado à disciplina de Trabalho de Diplomação, do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas – CSTADS – da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de Tecnólogo.

Orientador: Msc. Claudio Leones Bazzi

MEDIANEIRA

2011

Ministério da Educação

Universidade Tecnológica Federal do Paraná
Diretoria de Graduação e Educação Profissional
Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

TERMO DE APROVAÇÃO

Desenvolvimento de Sistemas com produtividade utilizando ferramentas visuais de programação

por

Josiel Amaro da Silva

O presente trabalho foi apresentado às..... hs do dia ..... como requisito parcial para a obtenção do título de Tecnólogo no Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, da Universidade Tecnológica Federal do Paraná, *Câmpus* Medianeira. O acadêmico foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho .....

_____	_____
Msc. Claudio Leones Bazzi UTFPR – *Campus* Medianeira	Marcela Turim Koschevic UTFPR – *Campus* Medianeira
_____	_____
Msc. Everton Coimbra de Araújo UTFPR – *Campus* Medianeira	Msc Juliano Rodrigo Lamb UTFPR – *Campus* Medianeira

AGRADECIMENTO

Agradeço primeiramente a Deus por não permitir que eu desistisse quando nada mais me dava motivos para continuar, a minha esposa Denise por estar do meu lado em todos os momentos, a minha família pelo incentivo e a todos os professores que soube compartilhar o conhecimento que possuíam com ética e profissionalismo.

RESUMO

SILVA, Josiel Amaro. **Desenvolvimento de Sistemas com produtividade, utilizando ferramentas visuais de programação.** Trabalho de conclusão de curso, Universidade Tecnológica Federal do Paraná. Medianeira, 2011.

Como melhora no desempenho da construção de *software*s, muitas empresas e desenvolvedores fazem uso de métodos como as ferramentas ágeis que padronizam e agilizam todo o processo de criação do *software*. Um dos maiores problemas no desenvolvimento de *software* está relacionando com a produtividade. Os negócios e a sociedade moderna estão exigindo cada vez mais sistemas, mais sofisticação, e tudo com maior rapidez. Um segundo aspecto do problema da produtividade é o tempo necessário para desenvolver um determinado sistema. Se o tempo médio de desenvolvimento de sistemas puder ser diminuído de três para um ano, o problema da demanda reprimida desaparecerá rapidamente. Várias pesquisas descobriram que cerca de um quarto de todos os projetos em grandes organizações de SIG (Sistemas de Informações Gerenciais) nunca são terminados (YOURDON, 1990). O presente trabalho consiste em desenvolver um *software* em linguagem *object pascal* para gestão de escolas de música integrando servidor local e servidor *web*.

Palavras-chaves: Produtividade, *Ferramentas ágeis, Delphi, Conexão banco local e WEB*

ABSTRACT

SILVA, Josiel Amaro. **Systems Development with productivity using visual programming tools**. Completion of course work, Federal Technological University of Paraná, Medianeira, 2011.

As improvement in building performance software, many companies and developers are using agile methods as tools that standardize and streamline the entire process of creating the software. One of the biggest problems in software development is related to productivity. Business and modern society seem to be demanding more and more systems, more sophisticated, and everything faster. A second aspect of the problem of productivity is the time required to develop a particular system. If the average time systems development can be reduced from three to one year, the problem of unmet demand will disappear quickly. Several studies have found that about one quarter of all projects in large organizations, MIS (Management Information Systems) are never completed (YOURDON, 1990). The present work was to develop a software object pascal language schools for managing music integrating local server and web server.

Keywords: Productivity Tools, agile, Delphi, Web site and database connection

LISTAS DE QUADROS

LISTAS DE FIGURAS

SUMÁRIO

1 INTRODUÇÃO

Em um clássico estudo sobre a demanda reprimida por sistemas de informações os pesquisadores Robert Alloway e Judith Quillard, da MIT *Sloan School*, descobriram que o *backlog* invisível de novos sistemas era 5,35 vezes maior que o visível. Isso indica que o problema da demanda reprimida é muito parecido com o proverbial *iceberg*: apenas uma pequena parte se mantém visível, com um enorme volume oculto sob a água. Isso, naturalmente, representa um problema importante para a empresa que prepara seu orçamento e seu planejamento com base apenas na demanda conhecida e visível por seus serviços. Um segundo aspecto do problema da produtividade é o tempo necessário para desenvolver um determinado sistema. Se o tempo médio de desenvolvimento de sistemas puder ser diminuído de três para um ano, o problema da demanda reprimida desaparecerá rapidamente. A questão é que os usuários estão geralmente preocupados não apenas com o problema global do *backlog*, mas também com o problema local da produtividade em relação a um determinado projeto. Eles receiam que, quando o novo sistema ficar pronto, as condições terão modificado tão drasticamente que os requisitos originais terão perdido a importância (YOURDON, 1990).

Um sistema novo muitas vezes está associado a uma oportunidade comercial percebida pelo usuário, e essa oportunidade frequentemente tem uma “janela”, um período de tempo após o qual a oportunidade comercial terá desaparecido e não mais haverá necessidade do novo sistema. Existe um terceiro motivo para a preocupação com a produtividade em algumas organizações: alguns projetos falham e são cancelados pela gerência antes mesmo de estarem prontos. Existem muitas razões para o fracasso de um projeto: problemas técnicos, problemas gerenciais, inexperiência da equipe, falta de tempo para ser feita uma adequada tarefa de análise e projeto de sistemas que normalmente é um problema da chefia, escasso envolvimento por parte da gerência ou dos usuários (YOURDON, 1990).

Além de ter uma boa equipe de desenvolvimento é necessário investir em bons *softwares* para a equipe de desenvolvedores. Caso contrário, nesse estágio do projeto se criará um “gargalo”. Empresas de médio e grande porte fazem uso de metodologias ágeis que padronizam e agilizam todo o processo de criação do *software*. Ferramentas visuais de programação como *Delphi* ou *NetBeans,* também

agilizam o processo de produção de uma aplicação, pois permite ao desenvolvedor maior agilidade ao disponibilizar recursos como um botão ou mesmo relatórios.

2 OJETIVO GERAL

Elaborar uma pesquisa sobre as principais ferramentas ágeis, sistemas com produtividade, ferramentas visuais de programação e desenvolver um sistema para escolas utilizando ferramentas de produtividade.

2.1 OBJETIVOS ESPECÍFICOS

- Desenvolver um referencial teórico sobre ferramentas ágeis;
- Desenvolver um referencial teórico sobre sistemas com produtividade;
- Desenvolver um referencial teórico sobre ferramentas visuais de programação;
- Desenvolvimento de um sistema para escolas utilizando ferramentas ágeis com produtividade.

3 JUSTIFICATIVA

Observou-se que para uma melhor gestão de estabelecimento de ensino musical é fundamental organizar e manter os dados de forma que torne organizado e permita manipulação dos mesmos como cruzamento de dados e pesquisas elaboradas. Agilizar o processo de matrícula, garantir a consistência das informações de entradas e saídas e organizar informações de clientes e fornecedores são alguns dos motivos para implementação de uma solução que atenda esse segmento.

Melhorar a eficiência do negócio possibilita que a empresa aumente sua competividade e sua lucratividade. Além disso, pode-se citar um melhoramento nas informações para tomadas de decisões, automatização das tarefas rotineiras, melhor controle interno das operações, melhor atendimento aos clientes, aumento da capacidade de reconhecer problemas, entre outros. A credibilidade também é acrescentada a empresa informatizada, já que os clientes sentirão que seus dados estarão mais seguros e organizados.

Para a construção de um sistema que forneça todas as funcionalidades para as necessidades apresentadas é necessário fazer uso de metodologias ágeis como XP ou SCRUM que fornecem técnicas para que os requisitos do sistema inicial não venham sofrer grandes modificações trazendo resultados indesejados na conclusão do sistema como, por exemplo, funcionalidades inúteis que não serão usadas pelo cliente. Fazer uso de ferramentas visuais trará maior produtividade ao projeto já que será simples e rápido associar um elemento de interface ao código da aplicação.

4 REFERENCIAL TEÓRICO

Segundo (YOURDON, 1990), um dos maiores problemas no desenvolvimento de *software* está relacionando com a produtividade. Os negócios e a sociedade moderna parecem estar exigindo cada vez mais sistemas, mais sofisticação, e tudo com maior rapidez. Como analista de sistema, deve-se perceber os dois aspectos mais importantes desse problema: um é a demanda reprimida por novos sistemas que precisam ser desenvolvidos, e o outro é o tempo necessário para a construção de cada um deles.

Em quase todas as organizações americanas onde existe um grupo central responsável pelo desenvolvimento de novos sistemas, existem serviços aguardando a vários anos para serem executados. Na realidade, muitas organizações têm um *backlog* de quatro a sete ou mais anos. O *backlog* se constitui de três diferentes tipos de sistemas:

- ***Backlog* visível**: são os novos sistemas solicitados oficialmente pelos usuários e que foram aceitos e tiveram suas verbas aprovadas pelas comissões apropriadas de gerência. Entretanto, os projetos não foram iniciados por falta dos recursos necessários tais como analistas de sistemas, programadores.

- ***Backlog* invisível:** são os novos sistemas que os usuários sabem que precisam, mas que não se dão ao trabalho de solicitar pelas vias “oficiais” porque ainda estão aguardando a prontificacão de projetos do *backlog* visível.

- ***Backlog* desconhecido:** são os novos sistemas que os usuários ainda não sabem que precisam, mas que emergirão logo que sejam terminados alguns dos sistemas dos *backlogs* visível e invisível.

4.1 METODOLOGIAS ÁGEIS– *SCRUM*

Segundo Fabio Aquita, em entrevista ao site da Info (INFO, 2011), a crise do *software* vem desde a década de 70. Desde os projetos militares da década de 60 todo mundo já sofria com os mesmo problemas. Sempre se vem fazendo do mesmo jeito desde então. Nos anos 90 diversos pensadores da área de engenharia de *software* se juntaram para tentar encontrar a metodologia correta para lidar com *software* e não chegaram a uma conclusão. Mas chegaram a um conjunto de valores importantes, que deram origem ao que chamamos de manifesto ágil. Segundo publicação no site *Manifest Argile* (MANIFESTO AGIL, 2011), existem alguns princípios que regem esse manifesto:

Garantir a satisfação do consumidor entregando rapidamente e continuamente *software*s funcionais;

- *Softwares* funcionais são entregues frequentemente (semanas, ao invés de meses);
- *Softwares* funcionais são a principal medida de progresso do projeto;
- Até mesmo mudanças tardias de escopo no projeto são bem-vindas.
- Cooperação constante entre pessoas que entendem do 'negócio' e desenvolvedores;
- Projetos surgem através de indivíduos motivados, entre os quais existe relação de confiança;
- Design do software deve prezar pela excelência técnica;
- Simplicidade;
- Rápida adaptação às mudanças;
- Indivíduos e interações mais do que processos e ferramentas;
- *Software* funcional mais do que documentação extensa;
- Colaboração com clientes mais do que negociação de contratos;
- Responder a mudanças mais do que seguir um plano.

Segundo o site *Improveit* (IMPROVEIT, 2011), *Scrum* é uma metodologia ágil para gestão e planejamento de projetos de *software*. No *Scrum*, os projetos são divididos em ciclos como na Figura 1, tipicamente mensais chamados de *Sprints*. O *Sprint* representa um *Time Box* dentro do qual um conjunto de atividades deve ser

executado. Metodologias ágeis de desenvolvimento de *software* são iterativas, ou seja, o trabalho é dividido em iterações, que são chamadas de *Sprints* no caso do *Scrum*. As funcionalidades a serem implementadas em um projeto são mantidas em uma lista que é conhecida como *Product Backlog, sendo considerada a prática responsável pela coleta dos requisitos (*SCHWABER & BEEDLE *2002).* No início de cada *Sprint*, faz-se um *Sprint Planning Meeting*, ou seja, uma reunião de planejamento na qual o *Product Owner* prioriza os itens do *Product Backlog* e a equipe seleciona as atividades que ela será capaz de implementar durante o *Sprint* que se inicia. As tarefas alocadas em um *Sprint* são transferidas do *Product Backlog* para o *Sprint Backlog*.

A cada dia de uma *Sprint*, a equipe faz uma breve reunião (normalmente de manhã), chamada *Daily Scrum*. Esta reunião é também chamada de *Stand Up Meeting* (reunião em pé), já que é de praxe que todos os membros a realizem de pé (RISING & JANOFF, 2000). O objetivo é disseminar conhecimento sobre o que foi feito no dia anterior, identificar impedimentos e priorizar o trabalho do dia que se inicia. Ao final de um *Sprint*, a equipe apresenta as funcionalidades implementadas em uma *Sprint Review Meeting*. Finalmente, faz-se uma *Sprint Retrospective* e a equipe parte para o planejamento do próximo *Sprint*. Assim reinicia-se o ciclo.

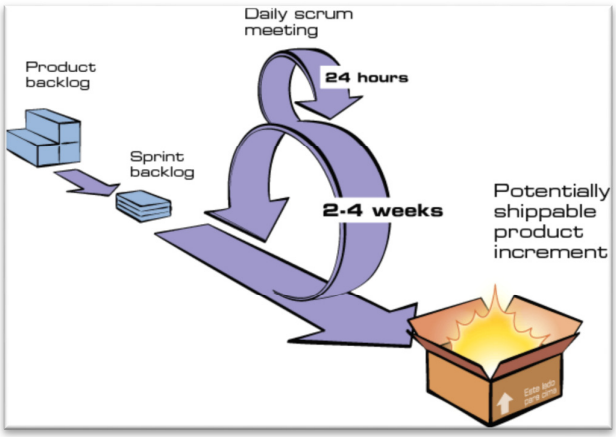


Figura 1 - Ciclos de um *SCRUM*
Fonte: IMPROVEITE 2011.

4.1.1 *Product Backlogs*

Product backlog é o coração do *Scrum.* Basicamente uma lista de requisitos, estórias, coisas que o cliente deseja, descritas utilizando a terminologia do cliente como no quadro 1. É chamado de estórias, ou algumas vezes apenas de itens do *backlog (*KNIBERG, 2007).

Estórias incluem os seguintes campos:

- **ID** – Uma identificação única, apenas um número com auto incremento. Isso é para evitar que se perca o controle sobre as estórias quando é mudado seus nomes.
- **Nome** – Um nome curto e descritivo para a estória. Por exemplo, "Ver o histórico de transações". Suficientemente claro para que os desenvolvedores e o *product owner* entendam sobre o que se está falando, e claro o bastante para distingui-la das demais estórias. Normalmente de 2 a 10 palavras.
- **Importância** – a pontuação de importância dessa estória para o *product owner*. Por exemplo, 10 ou 150, mais pontos = mais importante.
- **Estimativa inicial** – As estimativas iniciais da equipe sobre quanto tempo é necessário para implementar aquela estória, se comparada a outras estórias. A unidade é pontos por estória e geralmente corresponde mais ou menos a relação homem/dia ou seja quantas pessoas estão trabalhando para desenvolver aquele requisito.
- **Como demonstrar** – Uma descrição em alto nível de como a estória será demonstrada na apresentação do *Sprint*.
- **Notas** – quaisquer outras informações, esclarecimentos, referências a outras fontes de informação.

Itens de um product backlog					
ID	**Nome**	**Importância**	**Estimativa**	**Como demonstrar**	**Notas**
1	Depósito	30	5	Logar, abrir a página de depósito, depositar R$ 10,00, ir para a página do meu saldo e verificar que este aumentou em R$ 10,00.	Precisa de um diagrama UML de sequência. Não é necessário se preocupar com criptografia por enquanto.
2	Verificar seu próprio histórico de transações	10	8	Logar, clicar em "transações". Fazer um depósito. Voltar para transações, verificar se o novo depósito é listado.	Usar paginação para evitar consultas muito grandes ao banco de dados. Projetar de forma similar à página de visualização de usuários.

Quadro 1 – Itens de um product backlog
Fonte: KNIBERG, 2007.

O quadro 1 mostra como são planejadas as atividades a serem desenvolvidas em um projeto.

4.2 METODOLOGIAS ÁGEIS – XP(*Extreme Programin*)

Em artigo postado no site *Extreme Programming* (XP, 2011), Don Wells discorre que o primeiro projeto usando XP foi iniciado em 6 de março de 1996. É um dos vários Processos Ágeis existentes. Já foi provado ser muito bem sucedido em muitas empresas de todos os tamanhos e indústrias do mundo inteiro.

É bem sucedido porque sublinha a satisfação do cliente. Capacita os desenvolvedores a responder aos requisitos dos clientes, mesmo no final do ciclo de vida. Enfatiza o trabalho em equipe. Gerentes, clientes e desenvolvedores são todos parceiros em uma equipe colaborativa. Também implementa um ambiente simples, mas eficaz permitindo que as equipes tornem-se altamente produtivas. A equipe se auto organiza em torno do problema para resolvê-lo o mais eficientemente possível.

XP melhora um projeto de *software* em cinco formas essenciais: simplicidade, comunicação, *feedback*, respeito e coragem.

- **Simplicidade**: Faz somente o que for necessário e solicitado. Isto irá maximizar o valor criado para o investimento até o estágio de

desenvolvimento atual em que se encontrar o projeto. Dar pequenos passos com o objetivo de encontrar falhas durante o projeto. Criar projetos de interesse com prazos e custos razoáveis.

- **Comunicação**: Todo mundo faz parte da equipe e se comunicam diariamente. Trabalham-se juntos em tudo, desde os requisitos para o código.
- ***Feedback***: Apresenta como está o andamento do projeto para que se possa fazer as mudanças necessárias. Falar sobre o projeto e adaptar os processos a ele.
- **Respeito**: Todos se sentem respeitado sendo membro de uma equipe valorizada. Contribui com o projeto mesmo que seja apenas com motivação. Desenvolvedores devem respeitar a experiência dos clientes e vice-versa. A gestão é responsável e usa com autoridade a função que exerce.
- **Coragem**: Dizer a verdade sobre os progressos e estimativas. Não dar desculpas para o fracasso, porque o objetivo é sempre ter sucesso. Não temer nada porque ninguém trabalha sozinho. Adaptar-se às mudanças.

Programadores constantemente se comunicam com seus clientes e colegas para não se ter dúvidas sobre o que se está desenvolvendo. Podem obter *feedback* por meio de testes de *software* assim que comecem programar. Entregam o sistema para o cliente o mais cedo possível e com isso se necessário, realizar mudanças no projeto como sugerido. O aspecto mais positivo do XP em relação a outras metodologias, é a simplicidade de suas regras.

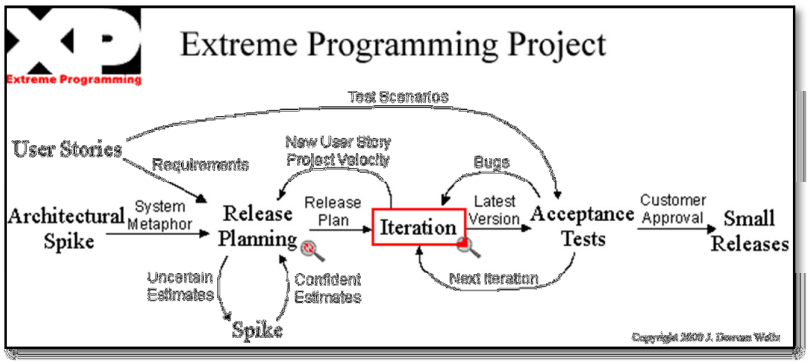


Figura 2 - Regras de trabalho usando XP
Fonte: XP(2011)

A Figura 2 mostra como as regras são trabalhadas em conjunto. Os clientes gostam de ser parceiros no processo de *software*. Os desenvolvedores podem participar ativamente independentemente do nível de experiência, e os gerentes se concentrar em comunicação e relacionamentos. Atividades improdutivas como aqueles que não estão definidas no *backlog* do projeto devem ser cortadas para reduzir custos e frustração de todos os envolvidos.

4.2.1 INICIANDO COM XP

Segundo o site *Extreme Programming* (XP, 2011), a maneira mais fácil para aplicar XP está no início de um novo projeto.

- Inicia-se coletando histórias de usuários e achando soluções para as mais difíceis. Usar no máximo 3 semanas.
- Em seguida, agendar uma reunião convidando os clientes, desenvolvedores e gerentes para criar uma agenda de reuniões semanais para apresentar e discutir sobre o projeto.
- Inicia-se o desenvolvimento iterativo com um planejamento de iteração e uma reunião.

Normalmente se procura uma nova metodologia como XP apenas depois que o projeto está com problemas de prazo ou custos. Neste caso, a melhor maneira é verificar na metodologia de *software* atual, se estiver utilizando algum, o motivo que está retardando o projeto e aplicar XP. Por exemplo, se vinte e cinco por cento dos requisitos desenvolvidos não estiver contribuindo com o projeto, então se reúne com os clientes e escrevem-se novas histórias de usuário.

Se estiver tendo problemas crônicos com a mudança de requisitos causando dificuldades em programar, então faça um planejamento de requisito simples e mais fácil.

Em primeiro lugar é necessário histórias do usuário. O estilo de desenvolvimento iterativo deve ser no tempo de planejamento de tarefas de programação. Se o maior problema é o número de erros na produção, em seguida, deve-se tentar automatizar testes de aceitação. Se o maior problema é a integração

de erros, deve-se tentar automatizar testes de unidade. Todos os testes devem passar em 100% de acertos. Se um ou dois desenvolvedores se tornar um gargalo, deve se fazer todas as alterações e tentar propriedade coletiva de código onde, todos os programadores da equipe podem corrigir erros, adicionar funcionalidades ou reconstruir o código. Pode-se continuar desta forma até que não haja problemas. Em seguida, basta dar início a prática. Ao solucionar um problema com XP, Inicialmente vai parecer fácil se estiver resolvendo um problema grande com um pouco de esforço extra. O segundo pode parecer muito mais fácil, mas em algum ponto entre usar algumas regras do XP ou todas as regras do XP ainda será necessário persistência para tudo funcionar bem.

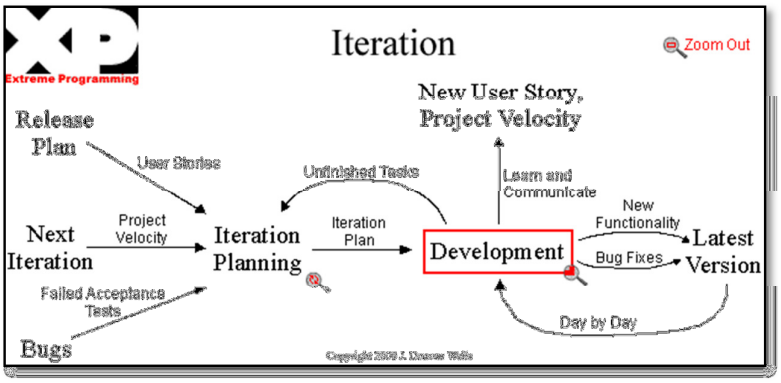


Figura 3 - Iteração em XP
Fonte: XP (2011)

A Figura 3 mostra uma reunião de planejamento de iteração que é usada para produzir plano de iteração de tarefas de programação. Cada iteração é de um a três semanas. Histórias de usuários são escolhidas para a iteração pelo cliente do plano de liberação na ordem de mais valioso para o cliente. Falhas de testes de aceitação, que são simulações realizadas com o sistema para verificar se estão de acordo com os requisitos devem ser anotados.

4.3 METODOLOGIAS ÁGEIS – *KANBAN*

Segundo o site Crisp (CRISP, 2011), *Kanban* significa muitas coisas. Literalmente, *Kanban* é uma palavra japonesa que significa "cartão visual". Na Toyota, *Kanban* é o termo usado para o sistema visual e sinalização física do sistema de produção da empresa. Como funciona:

- **Visualize o fluxo de trabalho** - Dividir o trabalho em partes, escrever cada item em um cartão e colocar na parede. Usa-se colunas nomeadas para ilustrar onde cada item está no fluxo de trabalho.
- **Limite de WIP (*work in progress*)** - atribuir limites explícitos para quantos itens pode estar em andamento em cada estado do *workflow.*
- **Medir o *lead time*** (tempo médio para completar um item, às vezes chamado de "tempo de ciclo") - aperfeiçoar o processo para fazer o *lead time* tão pequeno quanto possível.

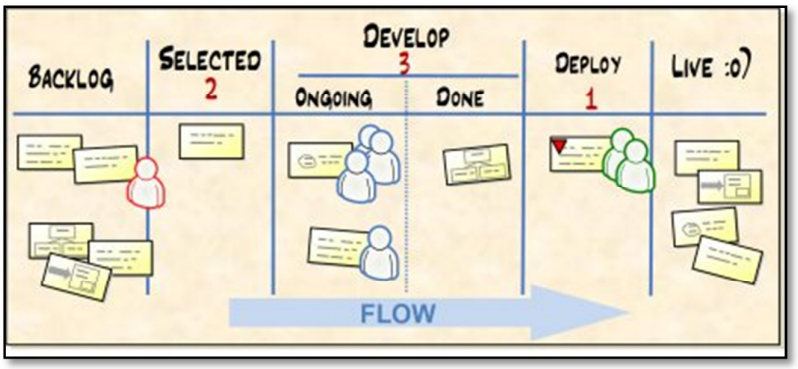


Figura 4 - Placa de Kanban
Fonte: CRISP(2011)

A Figura 4 é um exemplo de uma placa de *Kanban* simples, com limites de WIP em vermelho que são os números 2, 3 e 1.

4.3.1 Benefícios de um *Kanban*

- Os problemas conhecidos como “gargalo” tornam-se visíveis conforme se caminha o desenvolvimento do projeto, permitindo que o mesmo tenha participação de toda equipe não sobre caindo apenas a um participante.
- Fornece um caminho de evolução mais gradual dentro do modelo cascata para desenvolvimento ágil de *software*, ajudando assim as empresas que anteriormente tenham sido incapazes ou relutantes a experimentar métodos ágeis (KNINBER & SKARIN, 2009).
- Fornece uma maneira de usar metodologia ágil sem necessariamente ter que usar *sprints*. Útil para situações em que *sprints* não fazem muito sentido, como operações e equipes de apoio com uma elevada taxa de incerteza.
- Tende naturalmente a se espalhar por toda a organização de outros departamentos, como RH e vendas, aumentando assim a visibilidade de tudo o que está acontecendo na empresa.

4.3.2 Como usar *Kanban*

Kninber (KNINBER & SKARIN, 2009), descreve como utilizar o conceito de *Kanban* da seguinte forma.

Visualizar o fluxo de trabalho dividindo-o em partes e escrever cada item em um cartão e colocar na parede. Usar colunas nomeadas para ilustrar onde cada item está no fluxo de trabalho como na Figura 5. Limitar o trabalho em progresso WIP, associando limites explícitos para a quantidade de itens que podem estar em progresso em cada estado do fluxo de trabalho. Acompanhar o tempo de execução da tarefa (tempo médio para completar um item, algumas vezes chamado de “tempo de ciclo”). Aperfeiçoar o processo para tornar o tempo de execução o menor e mais previsível possível.

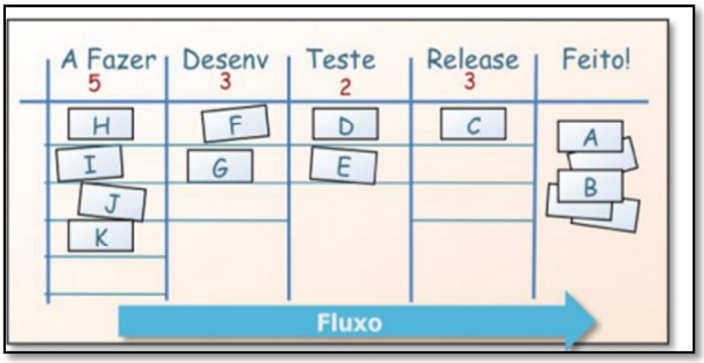


Figura 5 - Fluxo de um projeto em *Kanban*
Fonte: KNINBER & SKARIN (2009)

4.3.3 *Sprint* de um *Kanban*

Quando o *sprint* está pronto, a equipe faz uma revisão e demonstra as funcionalidades A, B, C, e D ao *product owner.* O *product owner*. Agora pode decidir se vai levar para produção da aplicação ou não.

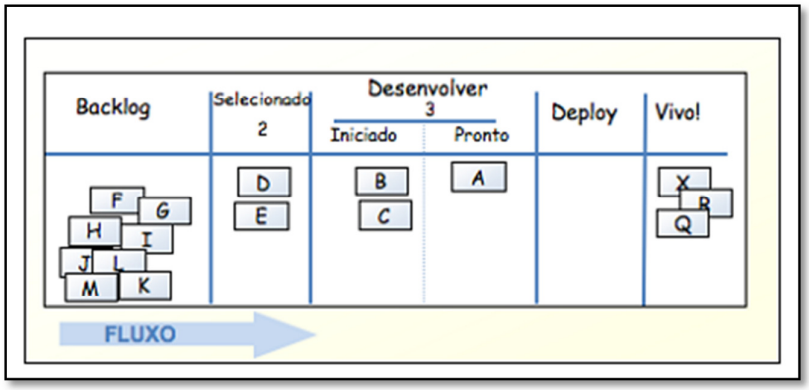


Figura 6 - Quadro de *Kanban*
Fonte: KNINBER & SKARIN (2009)

No exemplo da Figura 6, a coluna "*Backlog*" é, apenas, uma lista de coisas desejáveis, sem qualquer ordem especial. A coluna "Selecionado" contém os itens com maiores prioridades, com o limite *Kanban* igual a 2. Desta forma, deve conter apenas 2 itens com alta prioridade ao mesmo tempo. Sempre que a equipe estiver pronta para começar a trabalhar em um novo item, ela irá pegar o primeiro item da coluna "Selecionado". A qualquer momento o *Product Owner* poderá fazer mudanças nas colunas "*Backlog*" e "Selecionados", mas não poderá mudar as demais colunas. A coluna "Desenvolver" mostra o que está sendo desenvolvido no

momento, com limite *Kanban* de 3. Pode se dividir a coluna “Desenvolver” em duas sub-colunas “Iniciado” e “Pronto”, porém, não dará à equipe de produção a oportunidade de saber quais itens eles podem mover para a produção.

4.4 DIFERENÇAS ENTRE FERRAMENTAS

Segundo Kninberg (KNINBER & SKARIN, 2009), pode se comparar as ferramentas ágeis analisando quantas regras possuem. Prescritivo significa “mais regras a seguir” e adaptativo significa “menos regras a seguir”. Os dois extremos da escala acabam se tornando descartáveis. Metodologias ágeis como *Kanban* são chamadas de metodologias leves em inglês “*light weight*”, especificamente porque elas são menos prescritivos que os métodos tradicionais. O primeiro princípio do Manifesto ágil é “Indivíduos e Interações sobre processos e ferramentas”.

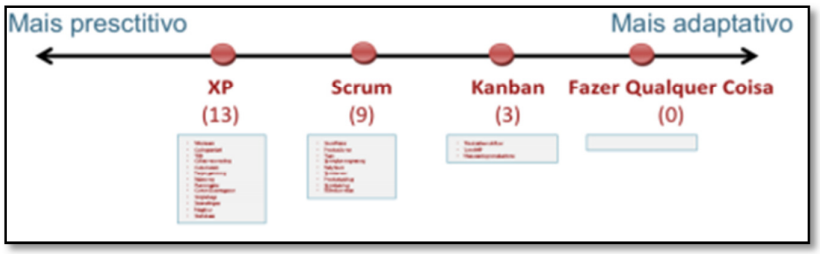


Figura 7 - Processos e ferramentas na escala prescritiva e adaptativa
Fonte: KNINBER & SKARIN (2009)

A Figura 7 apresenta as diferenças entre as ferramentas XP que é bastante prescritiva comparada ao *Scrum*. Ela inclui quase tudo do *Scrum* mais algumas boas práticas de engenharia bem específicas como desenvolvimento orientado a testes e programação em pares.

Scrum é menos prescritivo que XP, uma vez que ele não prevê nenhuma prática específica de engenharia. Porém, *Scrum* é mais prescritivo que *Kanban*, uma vez que prescreve coisas como iterações e equipes multifuncionais.

Kanban deixa quase tudo em aberto. As únicas restrições são: Visualizar o fluxo de trabalho e limitar as atividades em andamento.

4.5 FERRAMENTAS VISUAIS DE PROGRAMAÇÃO

Segundo Marcelo Leão (LEÃO, 2006), umas das primeiras ferramentas a adotar o conceito de RAD (*Rapid apllication Development*), que é uma maneira simples e rápida de associar um elemento de interface e o código da aplicação, foi o *Visual Basic for Windows* da *Microsoft.* No entanto havia algumas limitações como o desempenho dos aplicativos gerados pois não eram tão bons quanto as aplicações desenvolvidas em C++. Além disso, não suportava orientação a objeto. Foi então que a *Borland* lançou a *Borland Delphi* na versão 1.0. Uma ferramenta que alia a facilidade do Visual Basic ao poder da linguagem Pascal, cujo compilador apresenta o mesmo desempenho do *Borland* C++.

Segundo Fernando Enselmo (ENSELMO, 1997), enfatiza que Delphi está sendo usado por mais de 1.500 lugares como corporações e organizações, além de possuir dezenas de bibliotecas e ferramentas e Marcelo Leão (LEÃO,2006), que foi uma grande evolução a utilização de linguagens do tipo RAD, pois permite que o desenvolvedor tenha um grande rendimento ao desenvolver uma aplicação.

4.5.1 Liguagem Object Pascal

Segundo Fernando Enselmo (ENSELMO, 1997), *Object Pascal* é uma linguagem orientada a objetos não pura, mas híbrida por possuir características de programação não só visual, mas também escrita, para os programadores que já conhecem técnicas de estruturas de programação, como C, *Basic,* Pascal ou *xBASE* entre outras linguagens. Força a execução de passos lógicos tornando mais fácil o desenvolvimento no ambiente *Windows* de aplicações livres ou que utilizam banco de dados do tipo Cliente/Servidor. Trabalha com o uso de ponteiros para a alocação de memória e é totalmente compilável. Possibilita a criação e reutilização de objetos

e bibliotecas dinâmicas *(Dynamic Link Libraries* - DLL). Contém todo o conceito da orientação a objetos incluindo encapsulamento, herança e polimorfismo.

4.5.2 Compilador e IDE Delphi

Segundo artigo publicado no site *Plug Master* (PLUGMASTER, 2011), *Delphi* é um compilador e também uma IDE para o desenvolvimento de softwares. *Delphi* é largamente utilizado no desenvolvimento de aplicações *desktop* e aplicações multicamadas (cliente/servidor), compatível com os bancos de dados mais populares.

4.6 SISTEMAS COM PRODUTIVIDADE

A Produtividade é fundamental na determinação do custo de um projeto, servindo normalmente como base para comparações e estimativas. Muitas organizações utilizam a produtividade como critério na contratação de serviços de desenvolvimento e manutenção de sistemas, seja em licitações públicas, ou em solicitações de proposta. Não há unanimidade quanto à medida de tamanho de software a ser utilizada, sendo Pontos de Função o padrão para a medição do tamanho dos sistemas utilizada por praticamente todas as empresas brasileiras que efetuam algum tipo de medição sistemática. Outras medidas têm sido propostas e utilizadas, tais como as linhas de código *(SLOC – Source Lines of Code)* e diversas variações dos pontos de função (GOETHERT, 1992).

Nos projetos, a produtividade é utilizada na determinação do custo e preço, na avaliação do desempenho, na seleção de fornecedores e como base para estimativas em geral. Se uma organização conhecer sua produtividade em um dado contexto, poderá estimar, de forma simplificada, o custo de um projeto da seguinte forma (GARMUS, 2001): Estimar a quantidade de horas, multiplicando o tamanho estimado para o projeto em PF pela produtividade em H/PF. O resultado será o

esforço estimado para o projeto, medido em horas (GOETHERT, 1992). Multiplicar a quantidade de horas obtida no passo anterior pelo custo médio da hora, considerando a equipe. O resultado será a estimativa de custo do projeto. O preço poderá ser obtido acrescentando-se ao custo a margem de lucro pretendida pela organização. Como a produtividade varia bastante, frequentemente a mesma é calculada separadamente para cada plataforma computacional, ou conforme a principal linguagem de programação do projeto.

O cálculo da produtividade depende da determinação do tamanho do projeto e do respectivo esforço (quantidade de horas gasta). Idealmente, a medição dessas variáveis ocorrerá ao longo de todo o projeto, abrangendo todas as suas fases. Para a obtenção do tamanho do projeto em PF é indispensável a participação de pelo menos um especialista certificado pelo IFPUG (CFPS – *Certified Function Point Specialist)*. Isso garantirá o conhecimento e a uniformidade na aplicação das regras de contagem. Deverão ser medidas e contabilizadas, todas as alterações solicitadas e realizadas durante o projeto. Algumas organizações contabilizam, para efeito de pagamento, somente alterações solicitadas após o início da fase de construção do software. Todavia, não há um padrão universalmente aceito quanto a este ponto, que deve ser objeto de negociação entre cliente e fornecedor.

5 ESTUDO DE CASO

Em uma escola de música uma das formas usada para matrículas de alunos são as fichas tradicionais com escrita manual que, além de consumir tempo, traz enormes problemas de ilegibilidade. Quando se trata de anotar entradas e saídas entra o caderno com seus respectivos campos que além de não apresentar relatórios de forma dinâmica ainda deixa grandes margens de erros, sem falar do tempo que se gasta. Sem um sistema que auxilie na gerência haverá maior dificuldade nas tomadas de decisões, esforços repetitivos em tarefas rotineiras diminuindo a produtividade e a competitividade.

O presente trabalho consiste em desenvolver um *software* para gestão de escolas de música integrando servidor local e servidor *Web*. O *software* gerenciará todo processo de organização de uma escola como cadastro de clientes, funcionários, fornecedores, entradas, saídas, boletins, vendas de material e controle de mensalidades, trazendo grande agilidade a todo processo de matrícula além de garantir consistência das informações fornecidas pelo cliente dando mais uma opção de matricula online aos mesmos.

Além de o software ser apropriado para escolas de música será possível utilizá-lo em escolas de idiomas, escolas de informática, escolas de cursos preparatórios, cursos profissionalizantes, entre outros. Para desenvolvimento do trabalho será necessário conhecimento em linguagem *object pascal*, *SQL,* MySQL, PHP, comunicação entre servidor local e servidor *Web*, modelagem de banco, assim como na utilização dos *software*s específicos para cada linguagem. Também foi necessário conhecer esse segmento de negócio para que o sistema venha fornecer o resultado desejado utilizando-se de técnicas de análise.

Para desenvolvimento do projeto, após análise de requisitos onde se obteve as funcionalidades do sistema e seus objetivos, construiu-se os casos de uso. Na sequencia criou-se o DER (Diagrama de identidade e relacionamento) que permitiu a construção do banco de dados por meio do uso da linguagem SQL. Para desenvolvimento de *scripts* SQL. Para geração do banco fez se uso da ferramenta

SQuirreL SQL e para modelagem do banco a ferramenta WWW SQL *Designer* (Web). Como ferramenta de desenvolvimento usou-se *Borland Delphi na versão 6.*

5.1 ANÁLISE DE REQUISITOS

Manter usuário: Para utilizar o sistema o usuário deve ser incialmente cadastrado por um usuário Admin. (Quadro 2).

ID:	RF-01	**Nome:**	Manter usuários	**Usuário:**	Admin
Data:	02/10/2011	**Importância:**	Essencial	**Estimativa:**	
Descrição:	Cadastrar, alterar, excluir e consultar usuário				

Quadro 2 - Análise de requisitos, manter usuários.

Manter funcionário: Para controle de pagamentos os funcionários devem ser cadastrados no sistema por um usuário. (Quadro 3).

ID:	RF-02	**Nome:**	Manter funcionários	**Usuário:**	Usuário
Data:	02/10/2011	**Importância:**	Essencial	**Estimativa:**	
Descrição:	Cadastrar, alterar, excluir e consultar funcionários.				

Quadro 3 – Análise de requisitos, manter funcionários.

Manter clientes: Para o cliente efetuar uma compra de produtos ou matrícula na escola é necessário ser cadastrado no sistema por um usuário. (Quadro 4).

ID:	RF-03	**Nome:**	Manter clientes	**Usuário:**	Usuário
Data:	02/10/2011	**Importância:**	Essencial	**Estimativa:**	
Descrição:	Cadastrar, alterar, excluir e consultar clientes.				

Quadro 4 – Análise de requisitos, manter clientes.

Manter fornecedores: Para controle de contas a pagar é necessário informar um fornecedor que deve estar previamente cadastrado no sistema por um usuário. (Quadro 5).

ID:	RF-04	**Nome:**	Manter fornecedores	**Usuário:**	Usuário
Data:	02/10/2011	**Importância:**	Essencial	**Estimativa:**	
Descrição:	Cadastrar, alterar, excluir e consultar fornecedores.				

Quadro 5 – Análise de requisitos, manter fornecedores.

Manter contas a pagar: Para controle de saídas é necessário manter um cadastro do mesmo efetuado por um usuário do sistema. (Quadro 6).

ID:	RF-05	**Nome:**	Manter contas a pagar	**Usuário:**	Usuário
Data:	02/10/2011	**Importância:**	Essencial	**Estimativa:**	
Descrição:	Cadastrar, alterar, cancelar e consultar contas a pagar				

Quadro 6 – Análise de requisitos, contas a pagar.

Manter contas a receber: Para controle de contas a receber como mensalidades e produtos devem ser lançados no sistema por um usuário. (Quadro 7).

ID:	RF-06	**Nome:**	Manter contas a receber	**Usuário:**	Usuário
Data:	02/10/2011	**Importância:**	Essencial	**Estimativa:**	
Descrição:	Cadastrar, alterar, cancelar e consultar contas a receber				

Quadro 7 – Análise de requisitos, manter contas a receber.

Manter produtos: Ao efetuar uma venda os produtos da venda devem estar cadastrados no sistema por um usuário. (Quadro 8).

ID:	RF-07	**Nome:**	Manter produtos	**Usuário:**	Usuário
Data:	02/10/2011	I**mportância:**	Essencial	**Estimativa:**	
Descrição:	Cadastrar, alterar, excluir e consultar produtos.				

Quadro 8 – Análise de requisitos, manter produtos.

Manter cursos: Uma matrícula é realizada mediante alguns dados como o cliente e o curso que se pretende fazer que deve ser cadastrado por um usuário antecipadamente. (Quadro 9).

ID:	RF-08	**Nome:**	Manter cursos	**Usuário:**	Usuário
Data:	02/10/2011	I**mportância:**	Essencial	**Estimativa:**	
Descrição:	Cadastrar, alterar, excluir e consultar cursos				

Quadro 9 – Análise de requisitos, manter cursos.

Manter vendas: Toda venda efetuada deverá ser cadastrada no sistema por um usuário que informará o produto adquirido e o nome do cliente. (Quadro 10).

ID:	RF-09	**Nome:**	Manter vendas	**Usuário:**	Usuário
Data:	02/10/2011	I**mportância:**	Essencial	**Estimativa:**	
Descrição:	Cadastrar, alterar, cancelar e consultar vendas				

Quadro 10 – Análise de requisitos, manter vendas.

Manter lançamento no caixa: Para controle de entradas e saídas é necessário efetuar o lançamento no caixa que deve ser feito por um usuário do sistema. (Quadro 11).

ID:	RF-10	**Nome:**	Manter Lançamentos no caixa	**Usuário:**	Usuário
Data:	02/10/2011	I**mportância:**	Essencial	**Estimativa:**	
Descrição:	Cadastrar, alterar, cancelar e consultar lançamentos.				

Quadro 11 – Análise de requisitos, manter lançamentos no caixa.

Manter notas: Para possibilitar a impressão de boletins com suas notas as mesmas devem ser informadas no sistema por um usuário. (Quadro 12).

ID:	RF-11	**Nome:**	Manter notas	**Usuário:**	Usuário
Data:	02/10/2011	I**mportância:**	Importante	**Estimativa:**	
Descrição:	Cadastrar, alterar, cancelar e consultar notas no boletim				

Quadro 12 – Análise de requisitos, Admin cadastra nota.

Aluno realiza matrícula: Há duas formas de cadastro de cliente, uma por um usuário e a outra online pelo próprio cliente. (Quadro 13).

ID:	RF-12	**Nome:**	Aluno realiza matrícula	**Usuário:**	cliente
Data:	03/10/2011	I**mportância:**	Importante	**Estimativa:**	
Descrição:	Cliente realiza matrícula online				

Quadro 13 **– Análise de requisitos,** aluno realiza matrícula.

5.2 CASOS DE USO

Caso de uso 01:

Descrição: Cadastros que são feitos pelo cliente via *Web*.

Pré/Pós: Cliente realiza cadastro no sistema

Figura: 8

Caso de uso 02:

Descrição: Cadastros que são feitos pelo diretor .

Pré/Pós:Diretor realiza devidos cadastros no sistema

Figura: 9

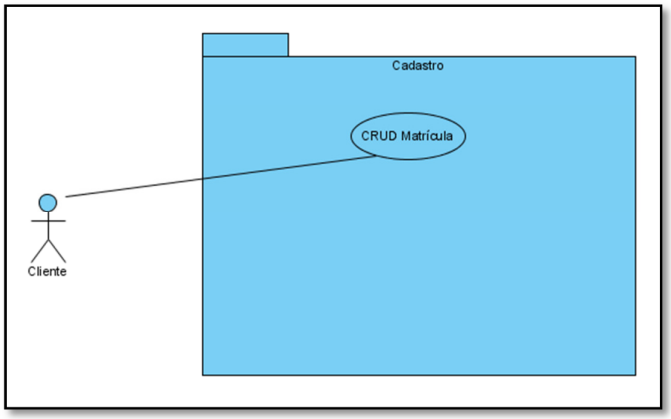


Figura 8 - Caso de uso do cadastro
Fonte: Autoria Própria

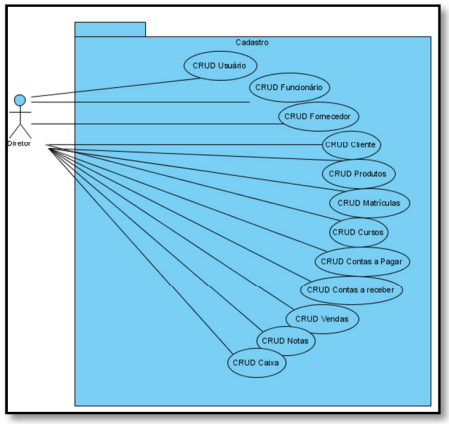


Figura 9 – Caso de uso do cadastro pelo diretor
Fonte: Autoria Própria

Caso de uso 03:

Descrição: Detalhamento do que é feito nos cadastros pelo diretor(inserir, excluir, alterar, consultar);

Pré/Pós: Cadastros de usuários só poderão ser feitos por administradores.

Fluxo(alternativas): Os cadastros só podem ser feitos pelas pessoas autorizadas e que possuem senha.

Figura: 1

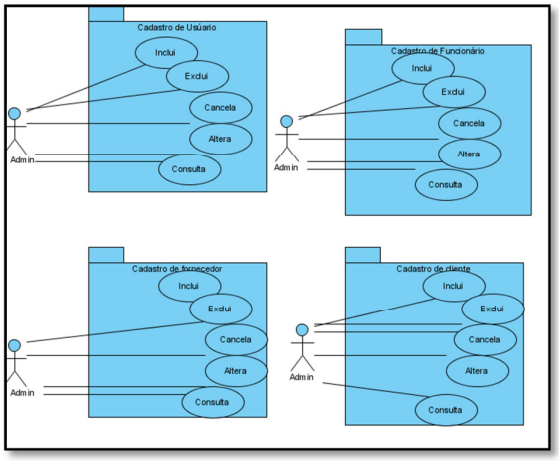


Figura 10 – Detalhamento do caso de uso feito pelo diretor
Fonte: Autoria Própria

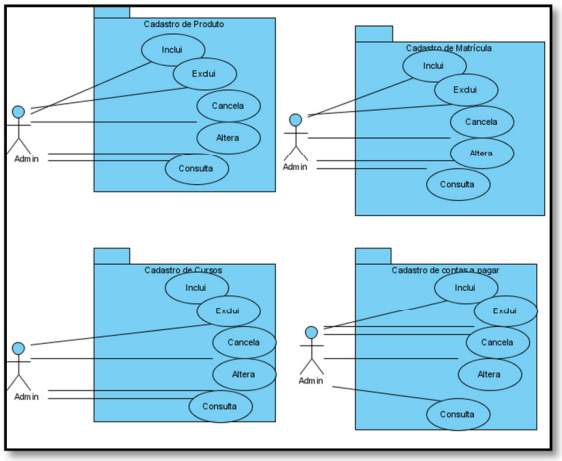


Figura 11 - Detalhamento do caso de uso feito pelo diretor
Fonte: Autoria Própria

Caso de uso 04:

Descrição: Cadastros que são feitos pelo funcionário.

Pré/Pós: Funcionário realiza devidos cadastros no sistema

Figura: 13

Caso de uso 05:

Descrição: Detalhamento do que é feito nos cadastros(inserir, excluir, alterar, consultar) pelo funcionário;

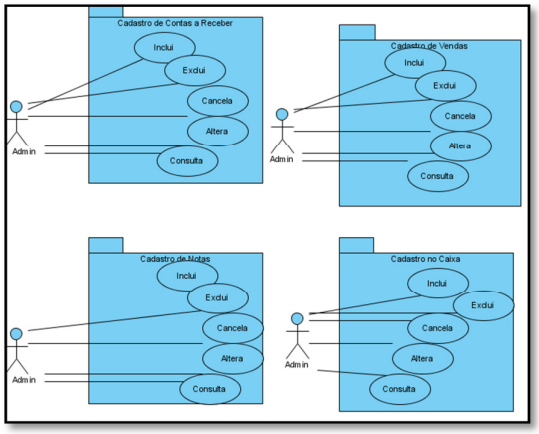


Figura 12 - Detalhamento do caso de uso feito pelo diretor
Fonte: Autoria Própria

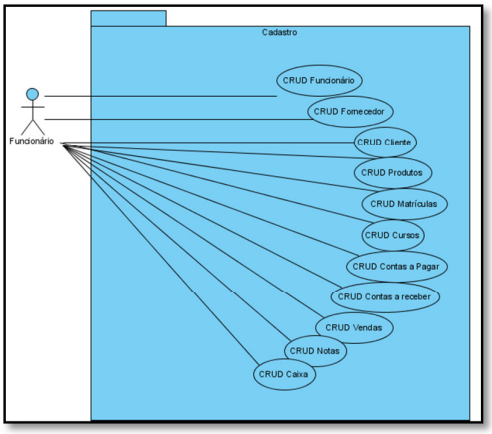


Figura 13 - Caso de uso do funcionário realizando cadastro
Fonte: Autoria Própria

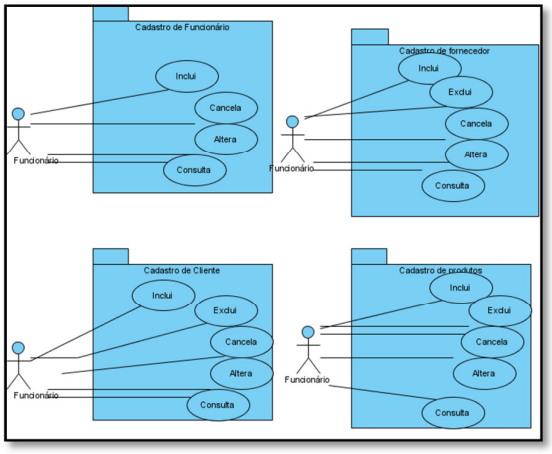


Figura 14 - Detalhamento do caso de uso de cadastro de clientes pelo funcionário
Fonte: Autoria Própria

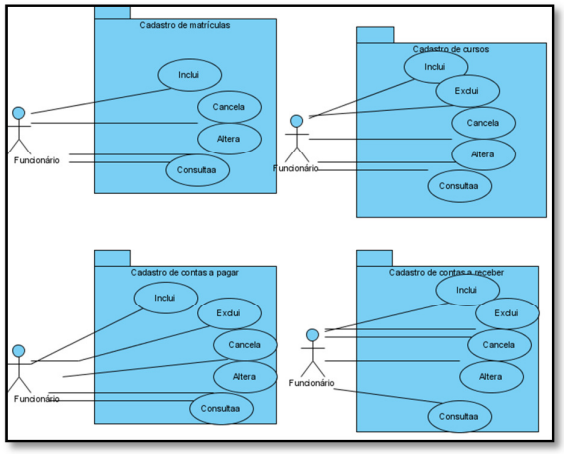


Figura 15 – Detalhamento do caso de uso de cadastro de clientes pelo funcionário
Fonte: Autoria Própria

5.3 DIAGRAMAS DE RELACIONAMENTO

O anexo 1 apresenta o diagrama de relacionamento descrito abaixo. A tabela TB_CONFIG armazena a opção do sistema de acessar o banco MySql configurado pelo administrador. TB_SITUACAO armazena situações de pagamentos como “pago” e não “pago” e a tabela TB_MENSAGEM guarda recados prévios como data de aniversários e datas comemorativas. **O relacionamento** entre TB_ESTADO e TB_CIDADE, mostra como são relacionados no banco os estados e as cidades.

Para efetuar a venda de um produto é necessário informar um item da TB_ITEMVENDA que por sua vez guarda os produtos do item em questão e para concluir, informar os dados de matrícula TB_MATRICULA ou caso não esteja matriculado os dados de inscrição TB_CLIENTE.

A matrícula é realizada mediante dados do cliente da TB_CLIENTE, curso pretendido da TB_CURSO, modalidade de preço da TB_MODALIDADEPRECO e da TB_TURMA. Para que os dados sejam inseridos no caixa é necessário informar se é entrada ou saída, histórico e código do cliente. O usuário é inserido

automaticamente e os dados são inseridos diretamente na tabela movimentação para consultas. Quando matriculado, o código é inserido automaticamente no caixa para possibilitar consulta de lançamentos pelos matriculados.

Os dados são inseridos no boletim mediante o código da matrícula, código do curso e da categoria que é o nível, período ou ano no qual o aluno está cursando. Em contas a receber, é cadastrado vendas que serão pagas a prazo através de boletos provenientes da tabela TB_VENDA onde deve-se informar o código do cliente ou da matrícula e caso o lançamento seja de uma venda de produtos deve-se informar código da mesma. Em conta a pagar, informa-se o código do fornecedor e os dados da movimentação são cadastrados automaticamente na tabela movimentos de contas a pagar.

5 APRESENTAÇÃO DO SISTEMA

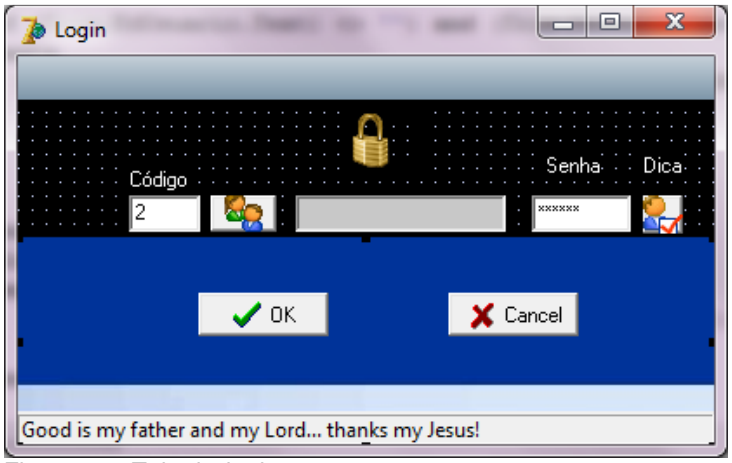


Figura 16 - Tela de *login*
Fonte: Autoria Própria

Para ter acesso as funcionalidades do sistema é necessário efetuar *login*, informando o usuário e senha conforme mostra a Figura 16. Caso se esqueça a senha poderá usar a opção dica. Se o usuário não estiver cadastrado, poderá solicitar ao usuário Administrador do sistema.

Figura 17 - Tela principal
Fonte: Autoria Própria

Todas as funcionalidades do sistema estão distribuídas por ordem de relevância na tela inicial conforme mostra a Figura 17. As mais comuns estão em botões de atalho.

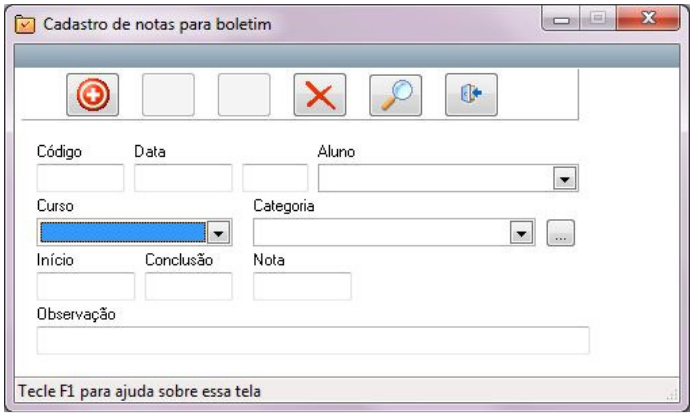


Figura 18 - Tela cadastra boletim
Fonte: Autoria Própria

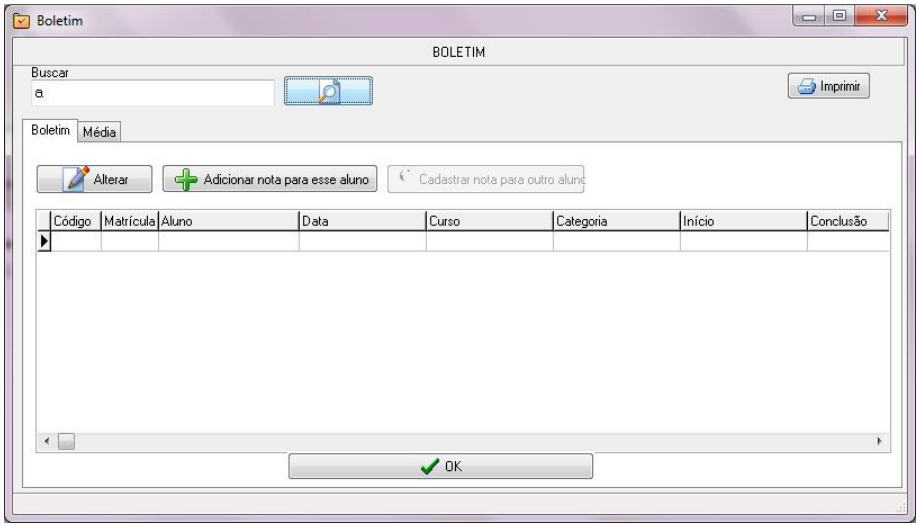


Figura 19 - Tela consulta boletim
Fonte: Autoria Própria

Na tela cadastra boletim, Figura 18, é possível cadastrar as notas do aluno informando o curso e a categoria da mesma. A categoria refere-se ao nível ou ano em que o aluno se realizou a avaliação. Na tela da Figura 19, é possível buscar as notas de cada aluno bem como a média das avaliações efetuadas até o momento.

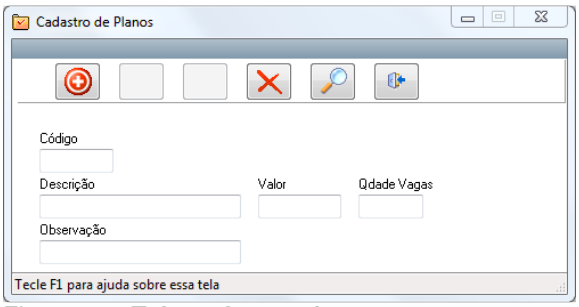


Figura 20 **- Tela cadastra planos**
Fonte: Autoria Própria

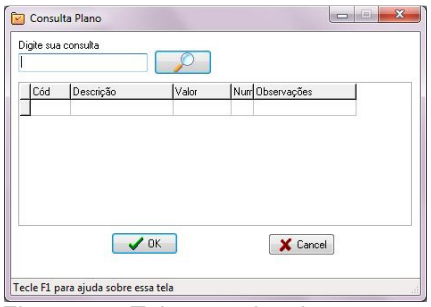


Figura 21 - Tela consulta planos
Fonte: Autoria Própria

Conforme mostra a Figura 20, o cadastro de planos é utilizado para cadastrar os valores dos planos de matrículas bem como a quantidade de alunos por cada plano. Dessa forma é possível alterar os valores das mensalidades de todos os alunos com facilidade. Na tela de consultas de planos, Figura 21, é possível consultar os planos assim como seus valores e quantidades de pessoas permitidas na turma.

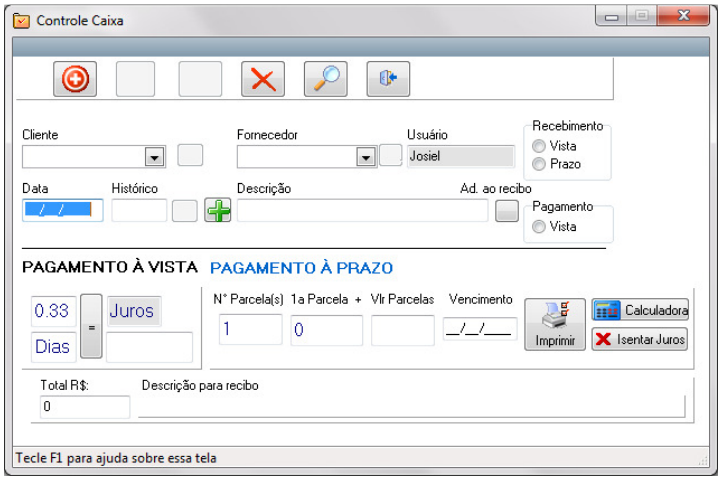


Figura 22 - Tela cadastra caixa
Fonte: Autoria Própria

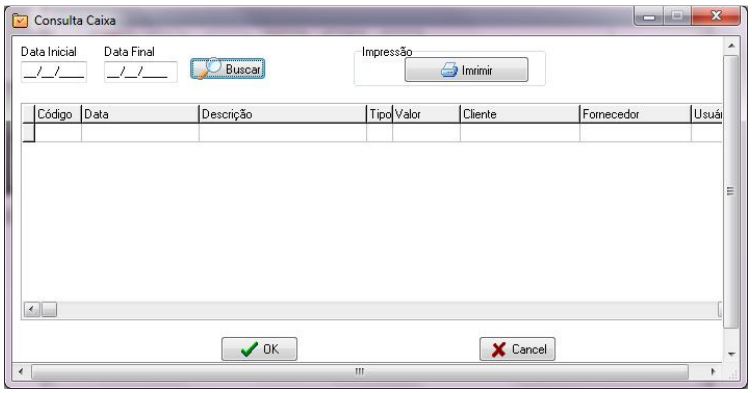


Figura 23 - Tela consulta caixa
Fonte: Autoria Própria

Para registro de entradas e saídas é usado o controle de caixa, conforme mostra a Figura 22, onde se escolhe o cliente caso seja uma entrada ou fornecedor caso seja saída. A entrada pode ser a vista ou parcelada com boleto. Ao lançar uma entrada o sistema verifica possíveis pendências do cliente. Após, é possível imprimir recibos para saídas ou boletos para as entradas. Para verificar dados de entradas e saídas usa-se a funcionalidade consulta caixa conforme mostra a Figura 23.

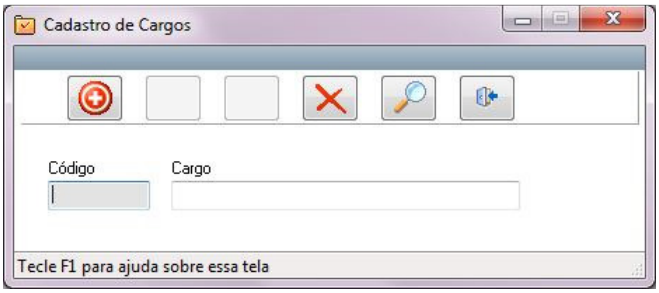


Figura 24 - Tela cadastra cargos
Fonte: Autoria Própria

Figura 25 - Tela consulta cargos
Fonte: Autoria Própria

Os funcionários são classificados por cargos como, gerentes, diretores, professores entre outros. Para isso, o cadastro é feito utilizando a tela “cadastra cargos” conforme mostra a Figura 24. Para busca dos cargos cadastrados dos funcionários pode-se usar o consulta cargos conforme mostra a Figura 25.

Figura 26 - Tela banco MySQL
Fonte: Autoria Própria

O Sistema possui uma funcionalidade para acessar cadastros em bancos MySQL em servidores *Web*, permitindo a importação de cadastro de clientes para o banco local conforme mostra a Figura 26. Essa funcionalidade é bastante útil quando a empresa possui um site com formulário de pré-matrícula.

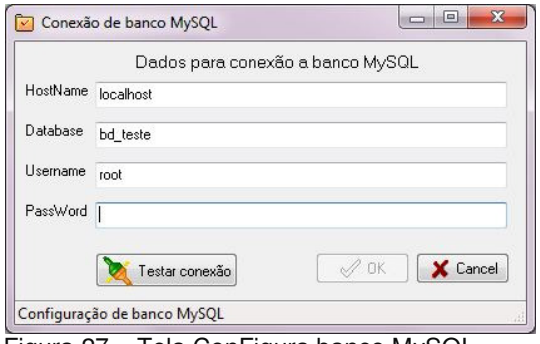


Figura 27 – Tela ConFigura banco MySQL
Fonte: Autoria Própria

Para acessar o banco de dados remoto é necessário configurá-lo informando endereço do *Host*, nome do banco, usuário e senha conforme mostra a Figura 27.

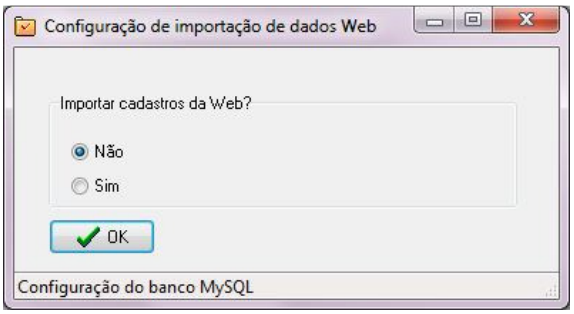


Figura 28 - Tela opção de acesso *Web*
Fonte: Autoria Própria

Caso a empresa não possuía formulário de pré-cadastro na *Web*, é possível desativar a essa conforme mostra a Figura 28.

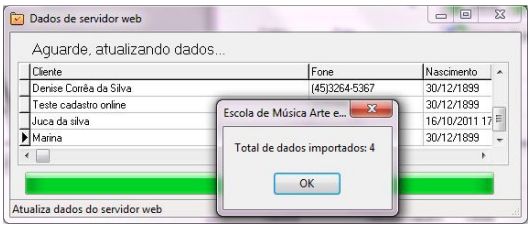


Figura 29 - Tela importa dados *Web*
Fonte: Autoria Própria

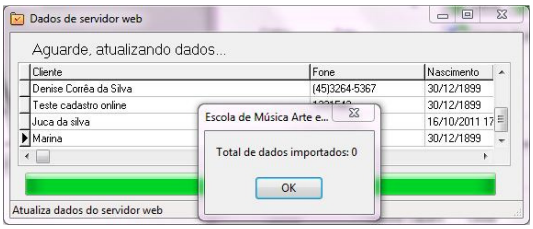


Figura 30 - Tela após importa dados *Web*
Fonte: Autoria Própria

O acesso ao banco e importação dos dados acontece no momento em que é iniciado um novo cadastro. O Sistema verifica inicialmente se os dados existentes no banco MySQL já existe no banco local INTERBASE e se caso existir, efetua-se o *download* dos dados do cliente e grava no banco conforme mostra a Figura 29 e a Figura 30. Após a importação, informa a quantidade de dados importados ao usuário do sistema.

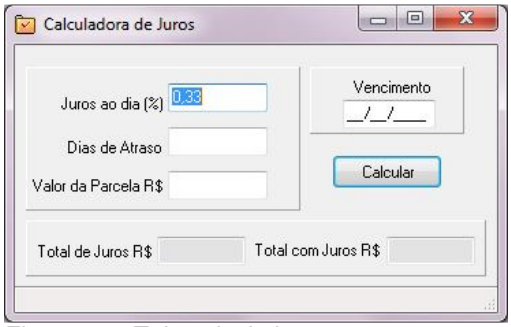


Figura 31 - Tela calcula juros
Fonte: Autoria Própria

Conforme mostra a Figura 31, é possível realizar cálculos rápidos de juros por atraso de pagamento utilizando a calculadora de juros no atalho adicionado ao caixa.

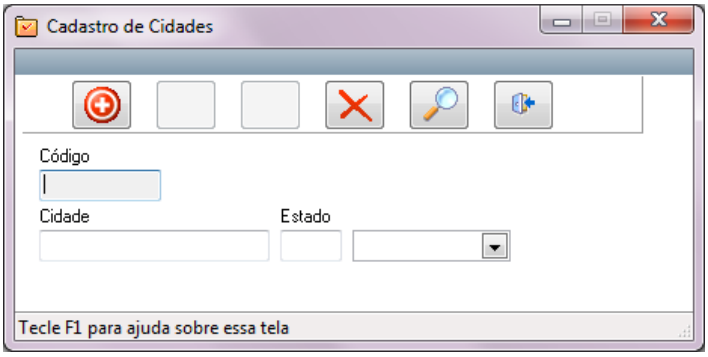


Figura 32 - Tela cadastra cidades
Fonte: Autoria Própria

Figura 33 - Tela consulta cidades
Fonte: Autoria Própria

Para cada estado previamente cadastrado, existem várias cidades. O cadastro das mesmas é efetuado após a escolha desse estado conforme mostra a Figura 32. Para consultar o cadastro das cidades e seus respectivos estados usa-se a funcionalidade consulta cidades conforme ilustrado na Figura 33.

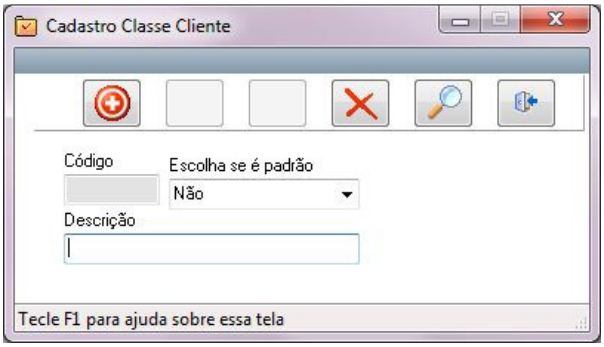


Figura 34 - Tela cadastra classe cliente
Fonte: Autoria Própria

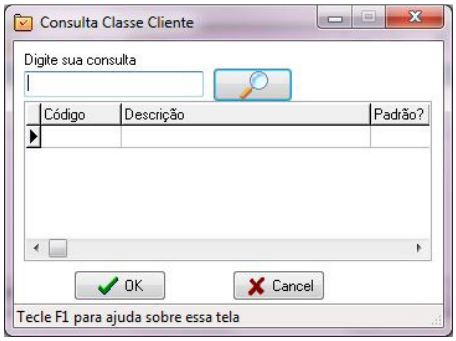


Figura 35 - Tela consulta classes
Fonte: Autoria Própria

Uma empresa pode dividir seus clientes em pessoa física ou jurídica, ou ainda, pode dar a classificação que se adeque ao seu ramo de negócio. Ao cadastrar é possível escolher se será uma categoria padrão ou não. Dessa forma agilizará a utilização dessa categoria durante uso do sistema como mostra a Figura

34. Para verificar as classificações dos clientes cadastrados pode se usar o consulta classes da Figura 35.

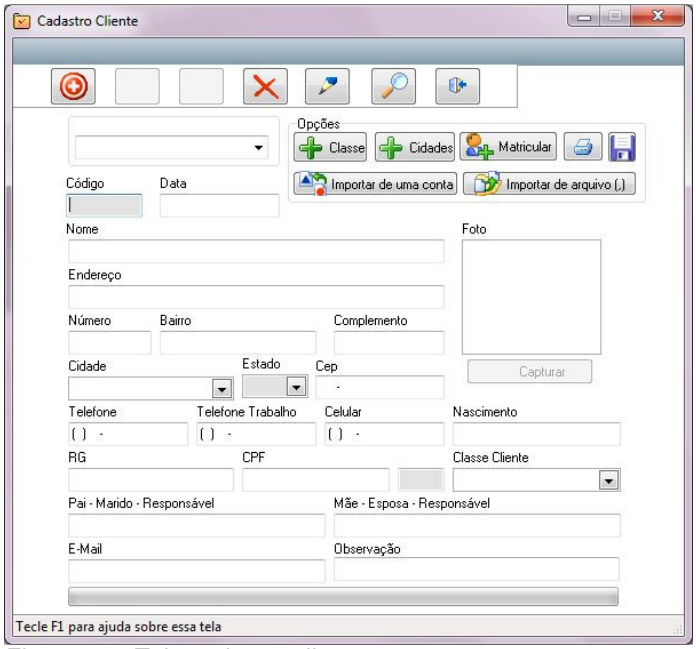


Figura 36 - Tela cadastra clientes
Fonte: Autoria Própria

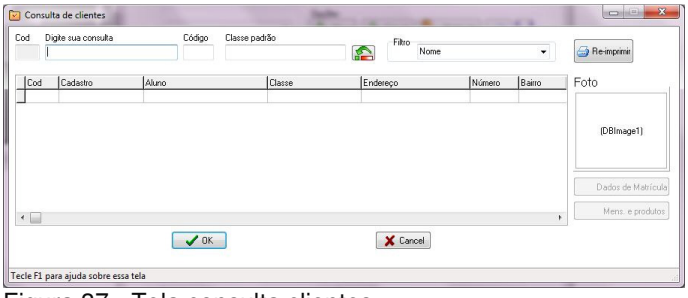


Figura 37 - Tela consulta clientes
Fonte: Autoria Própria

Para que seja possível efetuar uma venda de produtos, matrícula ou outra ação com um cliente é necessário cadastra-lo primeiro. Esse processo é feito de duas maneiras. Pelo próprio cliente na página de pré-cadastro online ou diretamente no sistema por um usuário. O sistema permite importar ou exportar um cliente para arquivo no formato TXT conforme ilustrado na Figura 36. Ao consultar um cliente, o sistema selecionará automaticamente a classe padrão. É possível efetuar a busca

usando o filtro que possui, nome, telefone, celular, pais e *e-mail* conforme ilustrado na Figura 37.

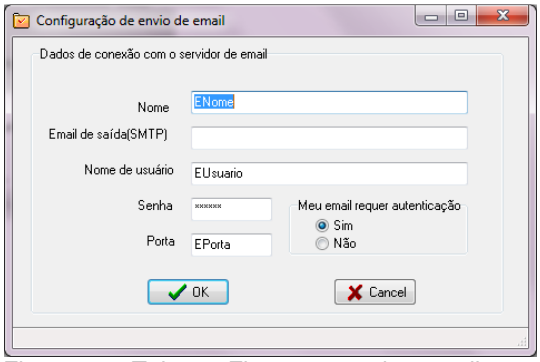


Figura 38 - Tela conFigura conta de *e-mail*
Fonte: Autoria Própria

Para envio de *e-mails* aos clientes deve-se primeiro configurar uma conta informando nome, *e-mail*, usuário, senha, porta e se deve autenticar ou não conforme ilustrado na Figura 38.

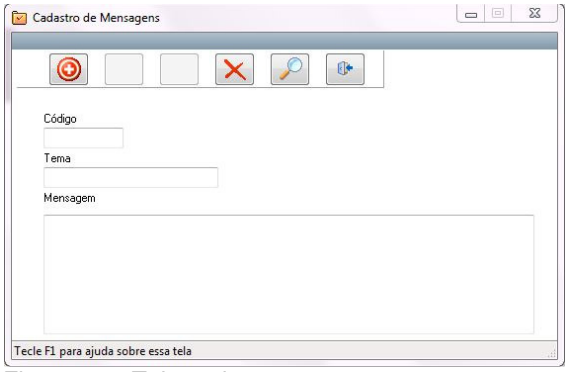


Figura 39 - Tela cadastra mensagens
Fonte: Autoria Própria

Mensagens que se repetem do tipo datas comemorativas podem ser cadastradas previamente antes de seu *e-mail* conforme ilustrado na Figura 39.

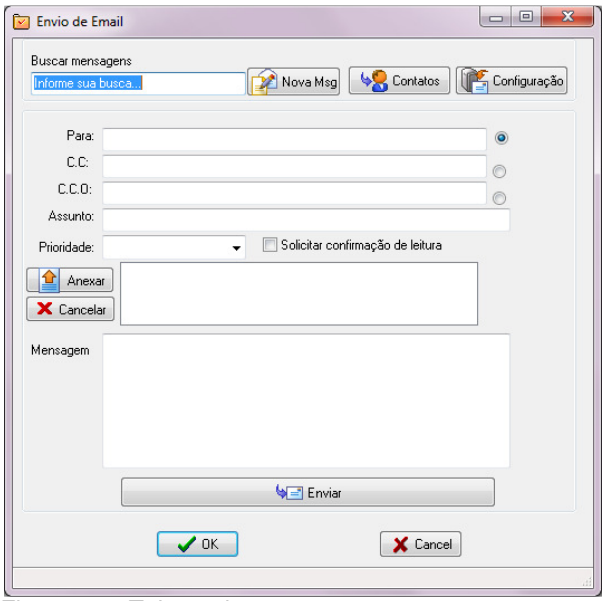


Figura 40 - Tela envia mensagem
Fonte: Autoria Própria

Para envio de mensagens aos clientes basta informar o *e-mail* do destinatário e a mensagem nos campos descritos conforme ilustrado na Figura 40. Se o *e-mail* for enviado, uma tela informará o usuário do sucesso da operação.

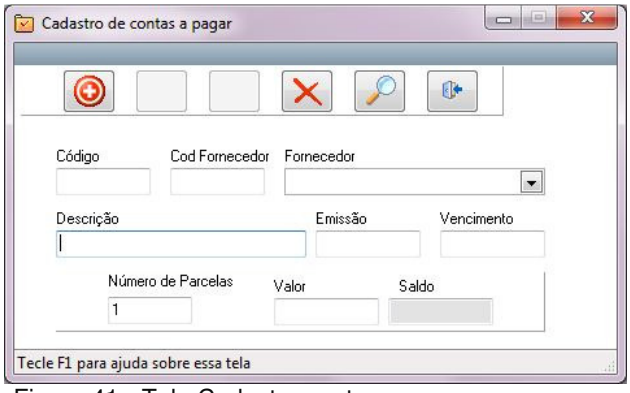


Figura 41 - Tela Cadastra contas a pagar
Fonte: Autoria Própria

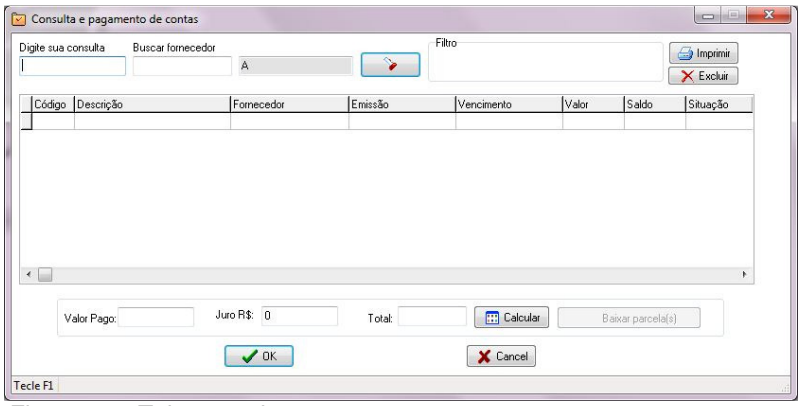


Figura 42 - Tela consulta contas a pagar
Fonte: Autoria Própria

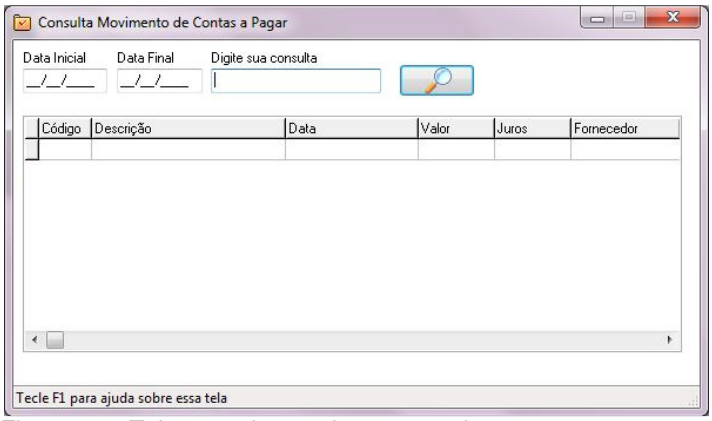


Figura 43 - Tela consulta movimentação de contas a pagar
Fonte: Autoria Própria

No controle de contas a pagar informa-se o fornecedor, descrição, data de emissão e vencimento, número de parcelas e valor conforme ilustrado na Figura 41 Na consulta por contas a pagar pode-se filtrar por fornecedores, informar juros efetuar lançamento no caixa conforme ilustrado na Figura 42. Para saber sobre as contas pagas, juros e datas, basta informar a consulta no campo especificado conforme ilustrado na Figura 43.

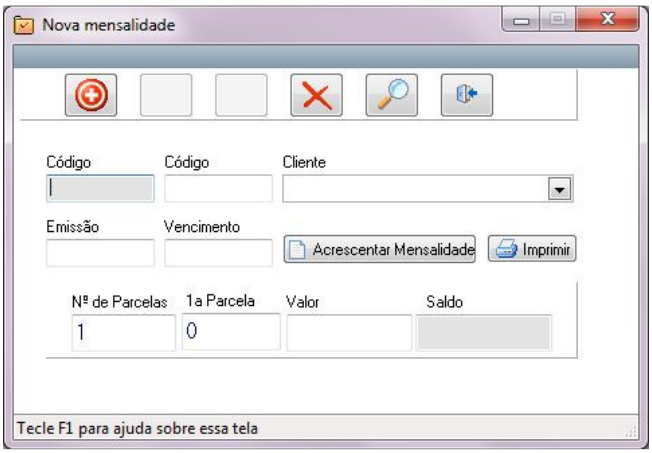


Figura 44 - Tela cadastra conta a receber
Fonte: Autoria Própria

O cadastro de recebimento de contas pode ser realizado de duas formas. Por parcelas ou direto para um dia desejado. Quando as mensalidades estão chegando ao fim basta clicar em acrescentar mensalidade para agilizar o processo de inserção e impressão das mesmas conforme ilustrado na Figura 44.

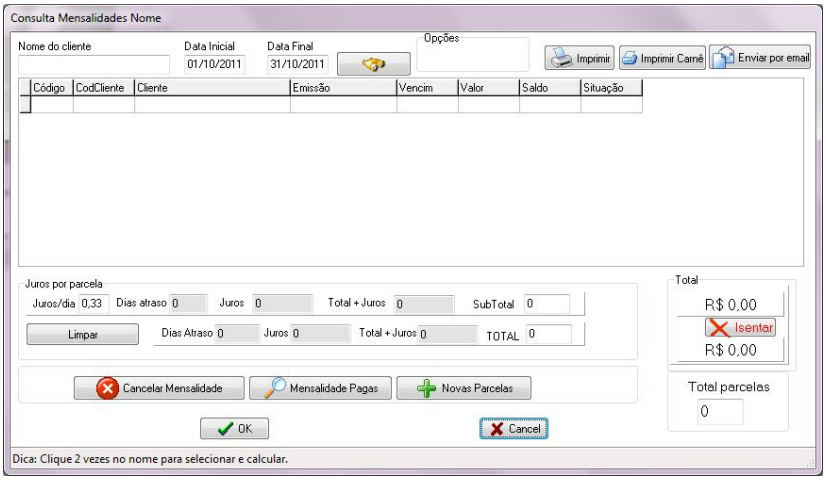


Figura 45 - Tela consulta contas a receber
Fonte: Autoria Própria

As mensalidades ou contas a receber podem ser encontradas de duas maneiras. Pelo nome ou pelo código do cliente. Depois de informado o nome é possível imprimir como lista, enviar como lembrete via *e-mail* do sistema, imprimir como carnê, cancelar ou com base na mesma adicionar novas conforme ilustrado na Figura 45.

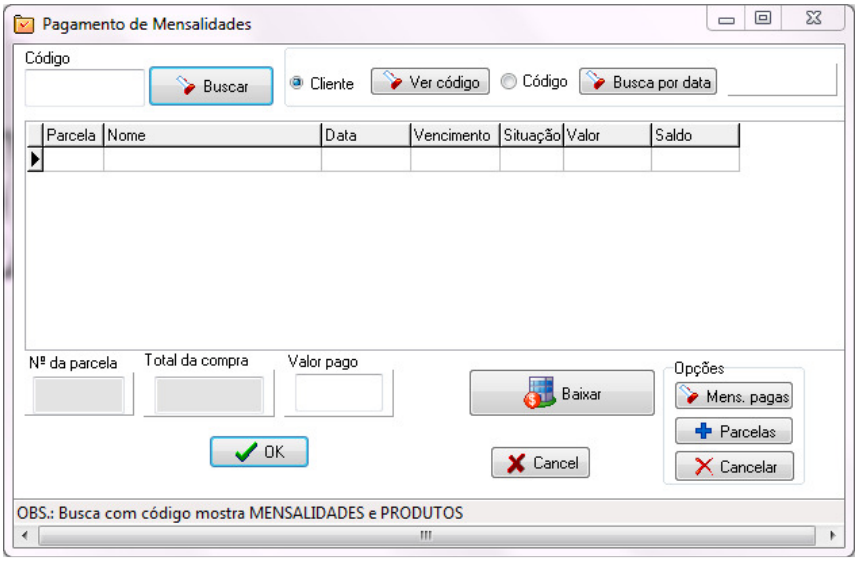


Figura 46 - Tela busca contas a receber por código
Fonte: Autoria Própria

Na busca por código quando encontradas, as mensalidades podem ser baixadas e seus valores de entrada adicionados ao caixa como pagamento à vista conforme ilustrado na Figura 46.

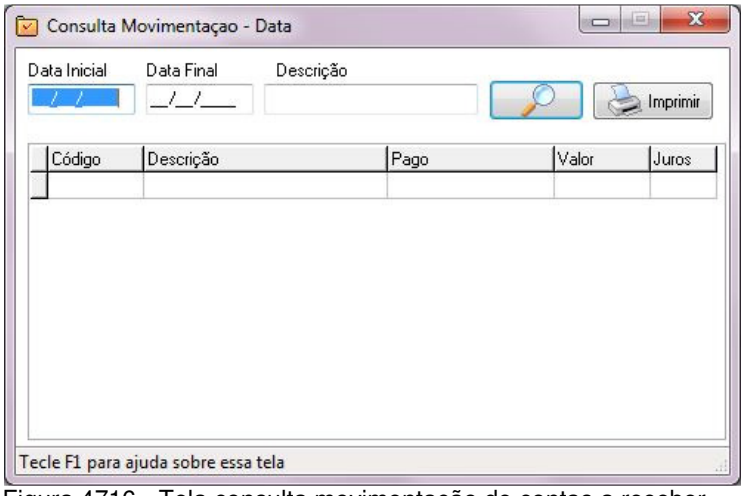


Figura 4716 - Tela consulta movimentação de contas a receber
Fonte: Autoria Própria

A movimentação apresenta quando as contas foram recebidas e se possui valores acrescidos de juros ou não conforme ilustrado na Figura 47.

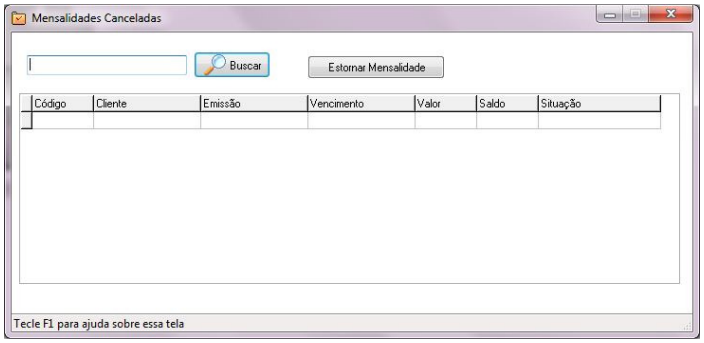


Figura 48 - Tela consulta mensalidades canceladas
Fonte: Autoria Própria

Mensalidades que foram canceladas podem ser estornadas ao sistema bastando apenas clicar no botão estornar mensalidades conforme ilustrado na Figura 48.

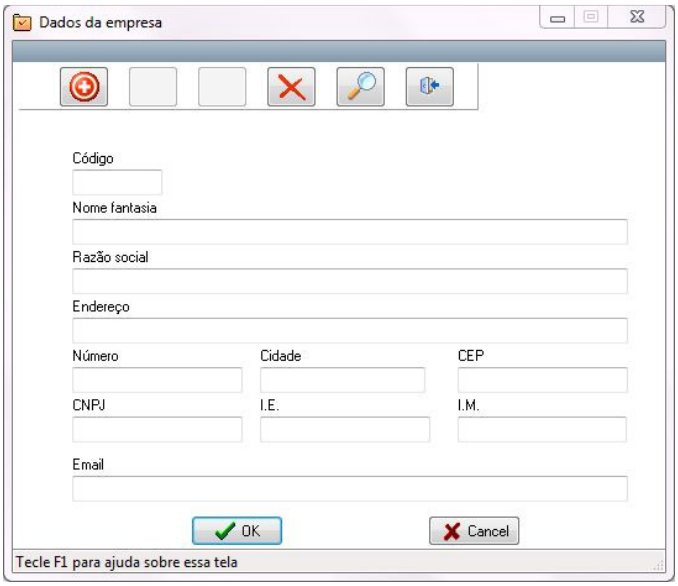


Figura 49 - Tela cadastra empresa
Fonte: Autoria Própria

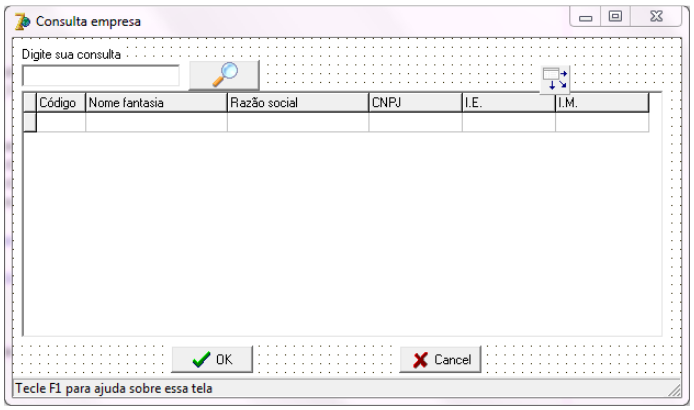


Figura 50 - Tela consulta empresa
Fonte: Autoria Própria

Para efeito de recibos os dados da empresa como nome e CNPJ são fundamentais. Por isso ao acessar o caixa é feito uma verificação se existe ou não dados cadastradas e o usuário é encaminhada para a tela de cadastro conforme ilustrado na Figura 49. Também é possível efetuar consultas dos dados da empresa conforme ilustrado na Figura 50.

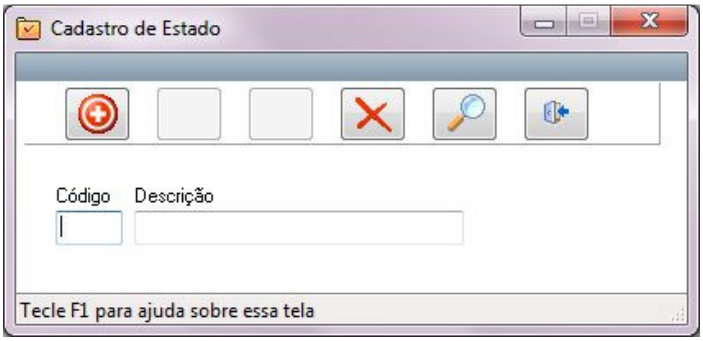


Figura 51 - Tela cadastra estado
Fonte: Autoria Própria

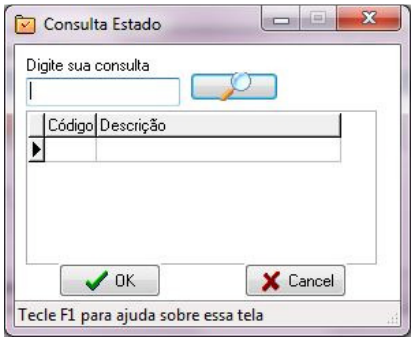


Figura 52 – Tela consulta estado
Fonte: Autoria Própria

No cadastro de estado conforme ilustrado na Figura 51, a chave primaria é a sigla do estado. Já que não pode existir duas iguais. A consulta apresenta o resultado dos estados cadastrados conforme ilustrado na Figura 52.

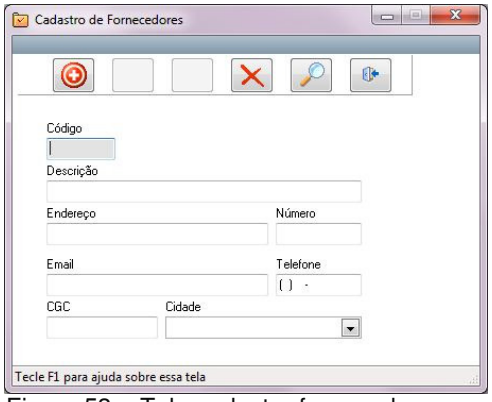


Figura 53 – Tela cadastra fornecedor
Fonte: Autoria Própria

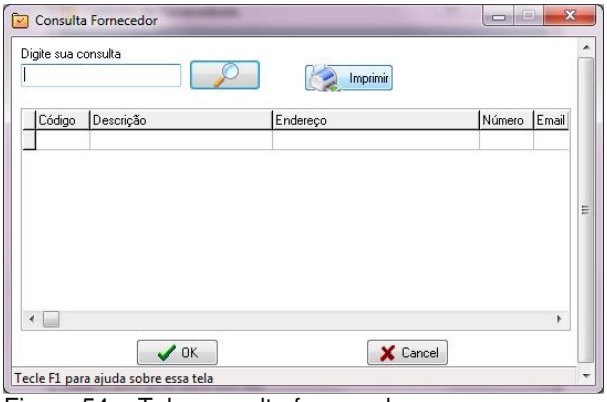


Figura 54 – Tela consulta fornecedor
Fonte: Autoria Própria

Para registro dos fornecedores é solicitado apenas os dados mais relevantes como contato, endereço e CNPJ conforme ilustrado na Figura 53. Para consulta dos fornecedores basta informar o nome da mesma para que o resultado possa ser mostrado conforme ilustrado na Figura 54.

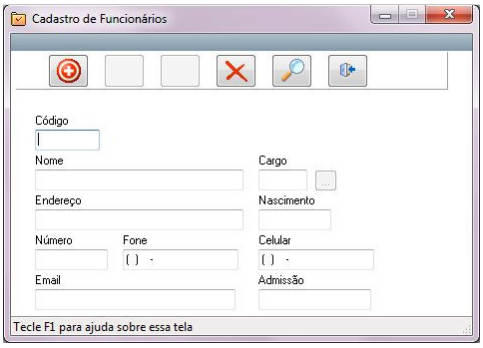


Figura 55 – Tela cadastra funcionário
Fonte: Autoria Própria

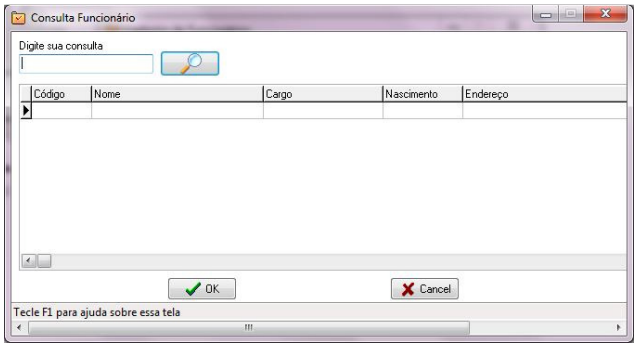


Figura 56 – Tela consulta funcionário
Fonte: Autoria Própria

Para que um funcionário seja identificado na empresa, em seu cadastro informa-se o nome, cargo que exercerá na empresa dados de endereço e data de admissão conforme ilustrado na Figura 55. Todos os dados cadastrados relacionados ao funcionário poderão ser encontrados em consulta funcionário conforme ilustrado na Figura 56.

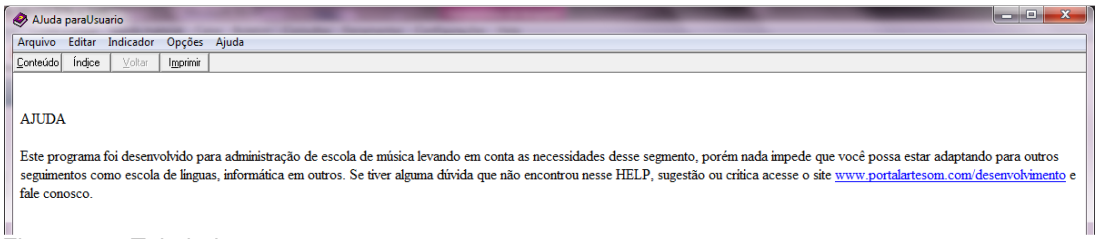


Figura 57 - Tela help
Fonte: Autoria Própria

O sistema fornece algumas dicas e informações ao usuário conforme ilustrado na Figura 57. Para isso em todas as telas do sistema existe F1 como atalho para acesso à mesma.

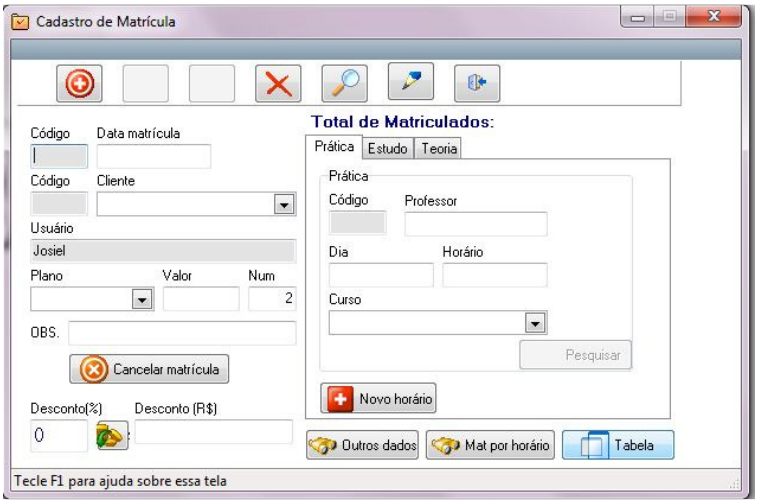


Figura 58 - Tela cadastra matrícula
Fonte: Autoria Própria

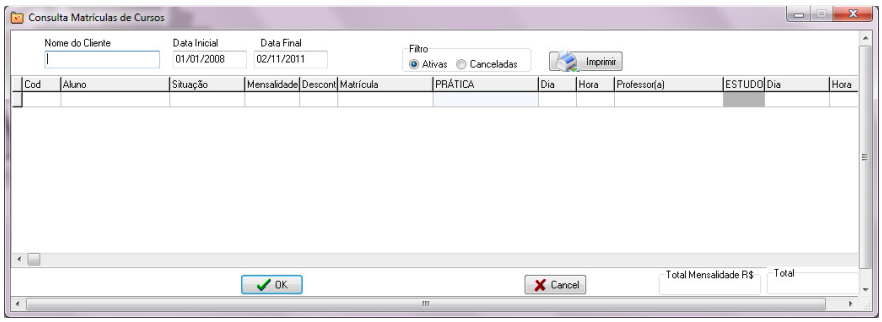


Figura 59 – Tela consulta matrícula
Fonte: Autoria Própria

Para que o cliente esteja matriculado é necessário registra-lo conforme ilustrado na Figura 58, informando seu nome, plano desejado, horário e curso. O Sistema oferece previamente três possibilidades de reserva de horário para o mesmo curso, podendo ser curso prática, apenas para estudo ou teoria. Para agilizar o processo de matrícula na mesma tela pode-se cancelar matrícula, adicionar novo horário e consultar outros dados do aluno. Para verificar as matrículas canceladas ou ativas pode se usar o consulta matrícula, conforme ilustrado na Figura 59 que também informa a quantidade de registro.

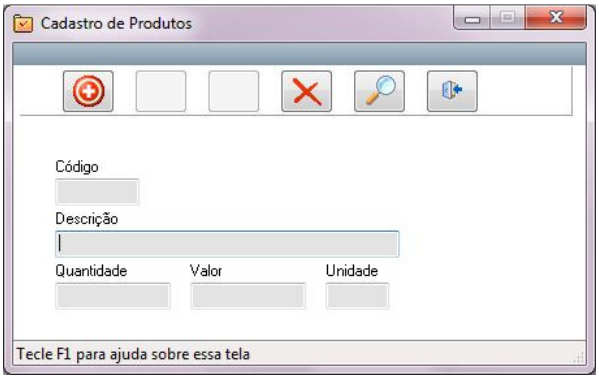


Figura 60 - Tela cadastra produtos
Fonte: Autoria Própria

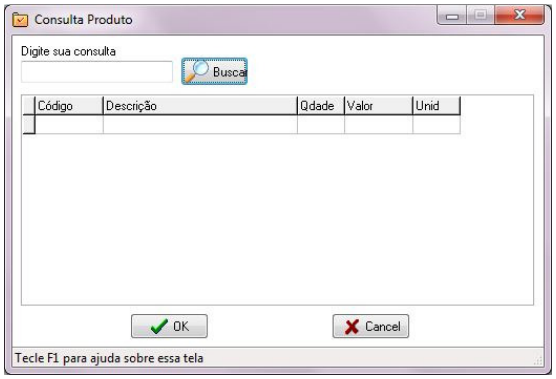


Figura 61 - Tela consulta produtos
Fonte: Autoria Própria

Os produtos oferecidos ao cliente podem ser cadastrados informando a descrição, quantidade, valor e a unidade de medida, ou seja, se é caixa ou unidades conforme ilustrado na Figura 60. Os produtos cadastrados podem ser consultados em consulta produtos conforme ilustrado na Figura 61.

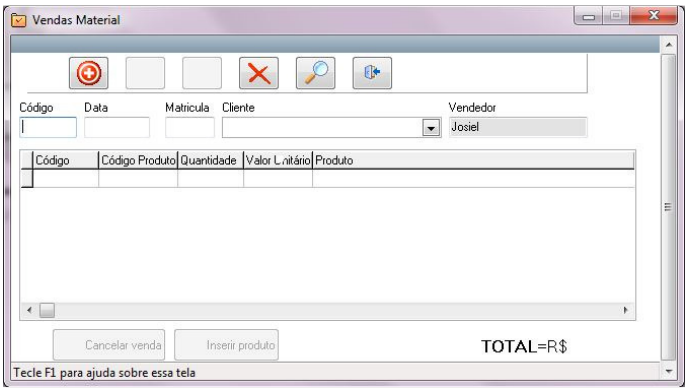


Figura 62 - Tela efetua venda de produt
Fonte: Autoria Própria

A venda de material é realizada após a escolha do produto no botão inserir produto conforme ilustrado na Figura 72 e depois informando o cliente. Também é possível cancelar a venda por um usuário administrador.

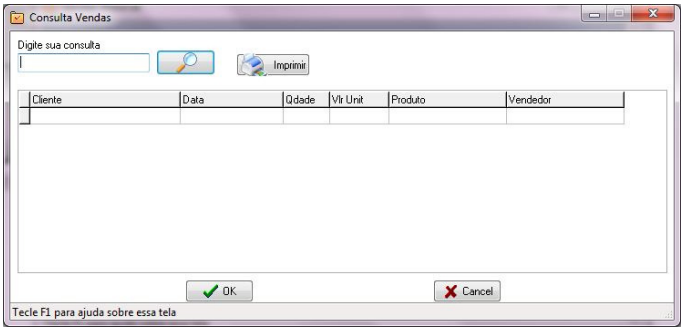


Figura 63 - Tela consulta vendas de produtos
Fonte: Autoria Própria

Par verificar a venda efetuada basta digitar o nome da consulta ou deixar em branco para apresentar todas as consultas conforme ilustrado na Figura 63.

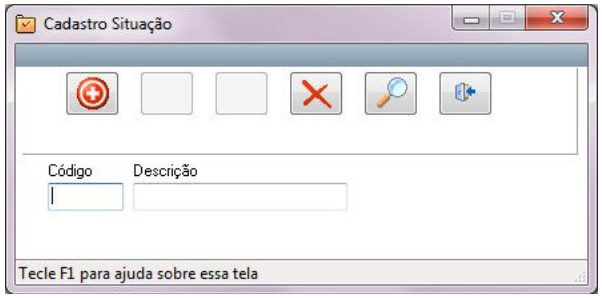


Figura 64 - Tela cadastra situação da venda

Fonte: Autoria Própria

Uma venda pode ter vários estados como pago e não pago. Para que essa opção esteja disponível deve-se primeiro registra-la no sistema conforme ilustrado na Figura 64.

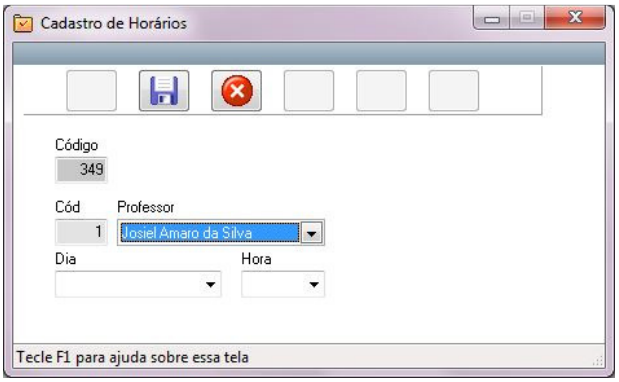


Figura 65 - Tela cadastra horários
Fonte: Autoria Própria

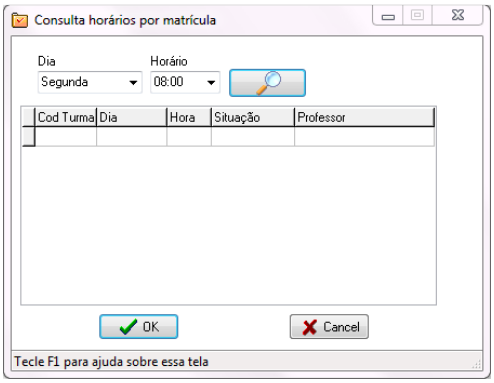


Figura 66 - Tela consulta horários
Fonte: Autoria Própria

Cada funcionário podem ter vários horários abertos turma conforme ilustrado na Figura 65. Os horários são fechados ou abertos automaticamente ou manualmente quando fecha a quantidade de alunos para aquela turma. Conforme ilustrado na Figura 66 é possível também realizar a consulta dos mesmos.

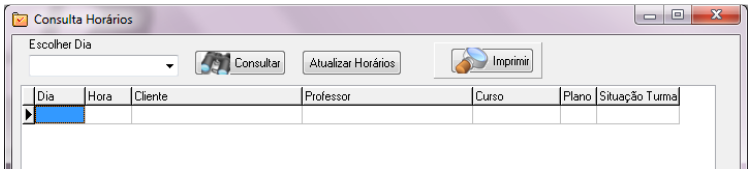


Figura 67 - Tela planilha de horários
Fonte: Autoria Própria

A planilha de horários, apresenta uma relação geral de horários de um determinado dia dos matriculados conforme ilustrado na Figura 67.

Figura 68 - Tela pré-visualização de impressão
Fonte: Autoria Própria

Todas as buscas efetuadas permite que sejam impressas. Para isso, usa-se a pré-visualização de impressão onde é possível escolher também a função salvar em TXT ou HTML conforme ilustrado na Figura 68.

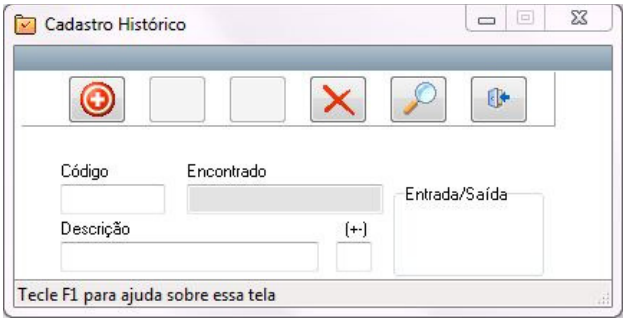


Figura 69 - Tela cadastra histórico do caixa
Fonte: Autoria Própria

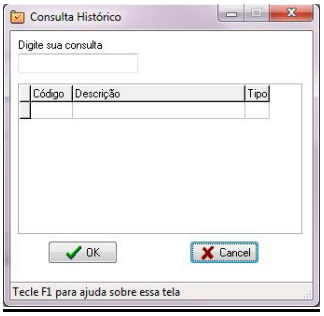


Figura 70 - Tela consulta histórico do caixa
Fonte: Autoria Própria

Um registro no caixa pode ser entrada (+) ou saída (-). No cadastrar histórico do caixa é possível cadastrar esses históricos que verifica possível duplicação caso o dado já exista conforme ilustrado na Figura 69. Para conferir ou consultar os dados pode-se usar o consulta histórico da caixa conforme ilustrado na Figura 70.

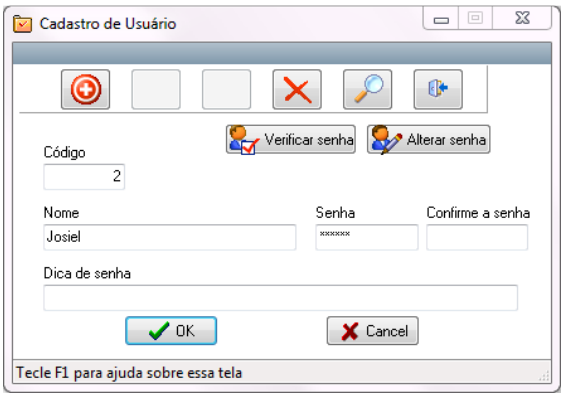


Figura 71 - Tela cadastra usuário
Fonte: Autoria Pópria

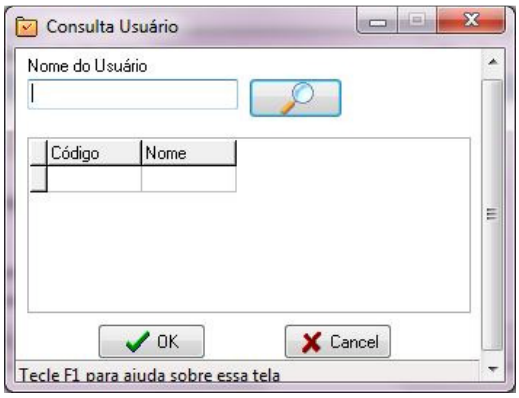


Figura 72 - Tela consulta usuário
Fonte: Autoria Própria

Usuários são todos os funcionários que administrarão o sistema. O Sistema já possui um usuário administrador (Admin), que terá permissão máxima no sistema como, adicionar ou alterar outros usuários, recuperar senhas e cancelar uma venda. Para ajudar no caso de esquecimento de uma senha o usuário poderá cadastrar uma dica de senha que poderá ser útil ao fazer *login*. Os eventos de caixa e vendas que envolvem movimentação financeira bem como as matrículas serão identificados automaticamente pelo usuário logado conforme ilustrado na Figura 71. A consulta mostra todos os usuários cadastrados conforme ilustrado na Figura 72.

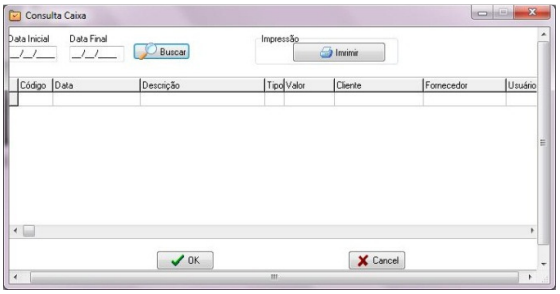


Figura 73 - Tela consulta caixa analítico
Fonte: Autoria Própria

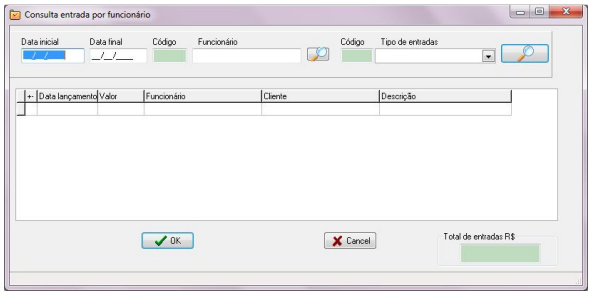


Figura 74 - Tela consulta caixa por funcionário
Fonte: Autoria Própria

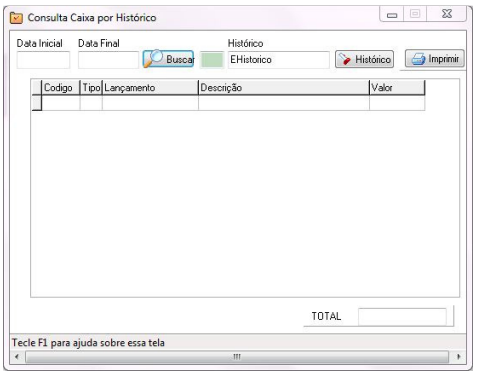


Figura 75 - Tela consulta caixa por histórico
Fonte: Autoria Própria

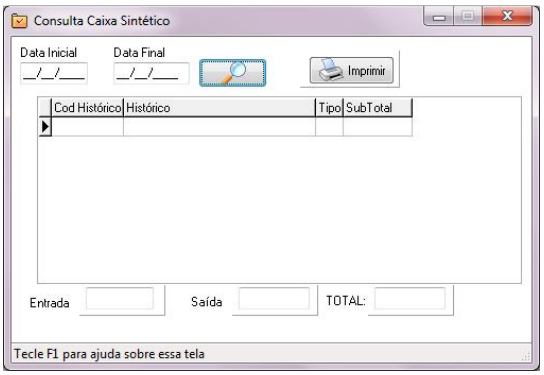


Figura 76 - Tela consulta caixa sintético
Fonte: Autoria Própria

Os registros do caixa poderão ser conferidos de várias maneiras. Por uma busca analítica que apresenta os dados mais detalhados conforme ilustrado na Figura 73, consulta de entradas por funcionários para efeito de pagamento conforme ilustrado na Figura 74, consulta por histórico conforme ilustrado na Figura 75 e consulta sintética onde apresenta os dados mais resumido com cálculo de entrada e saídas conforme ilustrado na Figura 76.

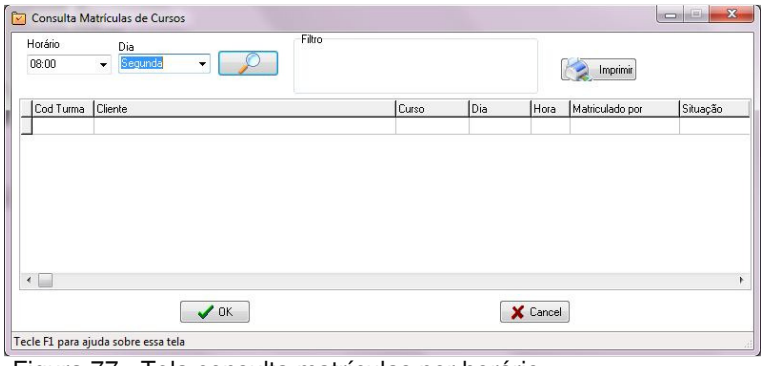


Figura 77 - Tela consulta matrículas por horário
Fonte: Autoria Própria

As matrículas poderão ser consultadas por horário, por dia ou combinando os ambos conforme ilustrado na Figura 77.

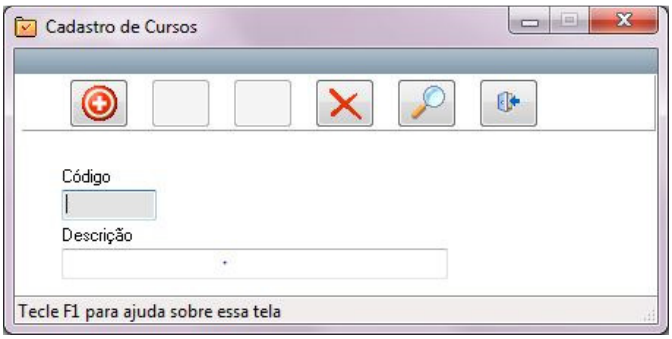


Figura 78 - Cadastra cursos
Fonte: Autoria Própria

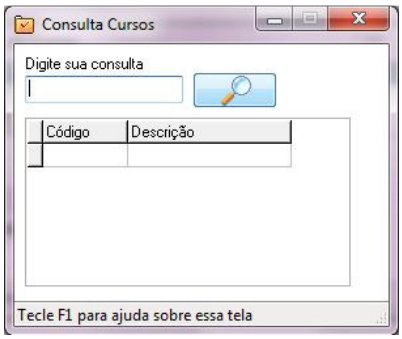


Figura 79 - Tela consulta cursos
Fonte: Autoria Própria

Um cliente pode se matricular em vários cursos que deve ser cadastrados antes do registro da matrícula conforme ilustrado na Figura 78. Após os dados podem ser consultados conforme ilustrado na Figura 79.

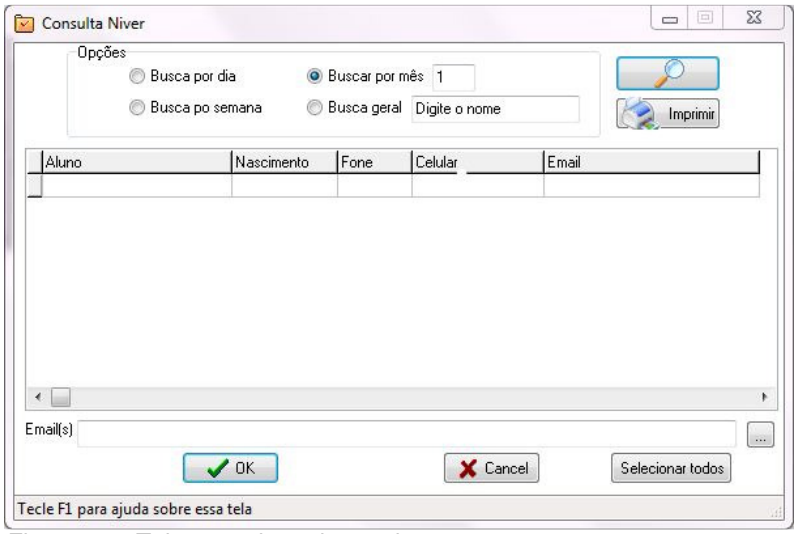


Figura 80 - Tela consulta aniversariantes
Fonte: Autoria Própria

Para envio de mensagens de aniversário, pode se consultar os aniversariantes do dia, da semana, do mês ou por nome conforme ilustrado na Figura 80. Se o cliente não possuir *e-mail* cadastrado o sistema informará o usuário**.**

6 CONCLUSÃO

Conclui-se, portanto, que desenvolver um *software* usando as métricas propostas pela análise de desenvolvimento como análise de requisitos, casos de uso e diagramas de relacionamento contribuiu para que o desenvolvimento fosse transparente e organizado, permitindo uma melhor visão do que foi desenvolvido. Como o sistema é rico em funcionalidades, muitas vezes detectar e corrigir problemas se tornou bastante complexo. Porém com o uso dos diagramas de relacionamento a complexidade foi amenizada.

Uma das dificuldades encontradas no desenvolvimento do projeto que possui vários requisitos para serem implementado, foi fazer uso das técnicas de ferramentas ágeis. Pois todas elas foram elaboradas e desenvolvidas para trabalho em equipe. No entanto, ter conhecimento das mesmas, fornecerá novas ferramentas para projetos que serão construídos com a participação de mais pessoas envolvidas.

Com relação a parte de produtividade da aplicação com uso de uma ferramenta visual foi de grande auxilio pois possibilitou que fosse implementado os requisitos com um prazo que não seria possível usando outras ferramentas.

7 TRABALHOS FUTUROS/CONTINUAÇÃO DO TRABALHO

O sistema será registrado em órgãos competentes e será distribuído com licença comercial. Com isso, novas versões com novas funcionalidades e correções serão desenvolvidas para dar qualidade e respaldo ao sistema.

8 REFERÊNCIAS BIBLIOGRÁFICAS

AGILE MANIFESTO. **Princípios por trás do manifesto ágil**. Disponível em: <http://www.agilemanifesto.org/iso/ptbr/principles.html/>. Acesso em: 8 de nov. 2011.

CRISP. ***Kanban***. Disponível em:< http://www.crisp.se/kanban > Acesso em: 10 de nov. 2011.

ENSELMO, Fernando. **Borland Delphi – Desvendado o Caminho das Pedras.** Copyright Fernando Antônio F. Anselmo 1997.

INFO. **Desenvolvimento ágil funciona?** Disponível em: <http://info.abril.com.br/noticias/ti/desenvolvimento-agil-funciona-30062009-3.shl>. Acesso em: 8 de nov. 2011.

KNIBER, H; SKARIN, M.***Kanban* e *Scrum*, obtendo o melhor de ambos**. C4Media Inc. 2009

KNIBERG, Henrik. ***Scrum* e XP direto das Trincheiras**. C4Media Inc. 2007

IMPROVEITE. ***Scrum***. Disponível em: <http://improveit.com.br/scrum/>. Acesso em: 8 de nov. 2011.

LEÃO, Marcelo. **Delphi 7 – Curso completo**. 1. ed. Axcel Books Ltd. 2003.

YOURDON, EDWARD. **Análise Estruturada Moderna.** 1. Ed. Campus. 1990.

XP.***Extreme Programming***.Disponível em: <http://www.extremeprogramming.org> Acesso em: 9 de nov. 2011.

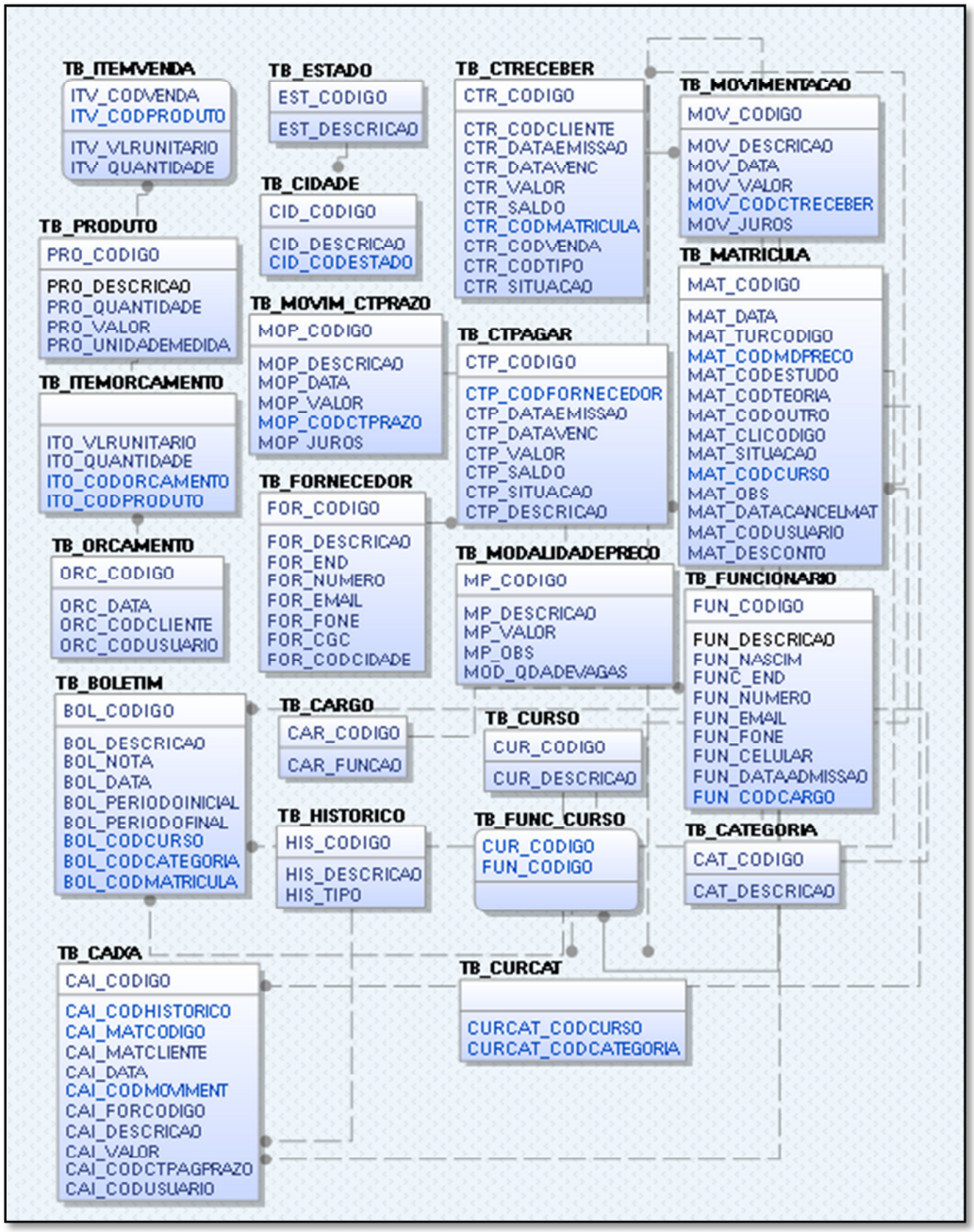
SCHWABER, K.; BEEDLE, M. **Agile Software Development with SCRUM.** Prentice Hall, 2002

BLUG MASTER, **Introdução a linguagem de programação Delphi,** Disponível em: <http://www.plugmasters.com.br/sys/materias/543/1/IntroduE7E3o-a-Linguagem-de-ProgramaE7E3o-Delphi>. Acesso em: 11 agosto. 2011.

GOETHERT, W. B. et al. **Software Effort & Schedule Measurement: A Framework for Counting Staff-hours and Reporting Schedule Informatio**n. CMU/SEI-92-TR-021. Software Engineering Institute, 1992.

GARMUS, D. & Herron, D. Function Point Analysis**. Measurement Practices for Successful Projects**. Addison-Wesley, 2001.

ANEXO


TB_ITEMVENDA
ITV_CODVENDA
ITV_CODPRODUTO
ITV_VLRUNITARIO
ITV_QUANTIDADE
TB_ESTADO
EST_CODIGO
EST_DESCRICAO
TB_CTRECEBER
CTR_CODIGO
CTR_CODCLIENTE
CTR_DATAEMISSAO
CTR_DATAVENC
CTR_VALOR
CTR_SALDO
CTR_CODMATRICULA
CTR_CODVENDA
CTR_CODTIPO
CTR_SITUACAO
TB_MOVIMENTACAO
MOV_CODIGO
MOV_DESCRICAO
MOV_DATA
MOV_VALOR
MOV_CODCTRECEBER
MOV_JUROS
TB_CIDADE
CID_CODIGO
CID_DESCRICAO
CID_CODESTADO
TB_PRODUTO
PRO_CODIGO
PRO_DESCRICAO
PRO_QUANTIDADE
PRO_VALOR
PRO_UNIDADEMEDIDA
TB_MOVIM_CTPRAZO
MOP_CODIGO
MOP_DESCRICAO
MOP_DATA
MOP_VALOR
MOP_CODCTPRAZO
MOP_JUROS
TB_CTPAGAR
CTP_CODIGO
CTP_CODFORNECEDOR
CTP_DATAEMISSAO
CTP_DATAVENC
CTP_VALOR
CTP_SALDO
CTP_SITUACAO
CTP_DESCRICAO
TB_MATRICULA
MAT_CODIGO
MAT_DATA
MAT_TURCODIGO
MAT_CODMDPRECO
MAT_CODESTUDO
MAT_CODTEORIA
MAT_CODOUTRO
MAT_CLICODIGO
MAT_SITUACAO
MAT_CODCURSO
MAT_OBS
MAT_DATACANCELMAT
MAT_CODUSUARIO
MAT_DESCONTO
TB_ITEMORCAMENTO
ITO_VLRUNITARIO
ITO_QUANTIDADE
ITO_CODORCAMENTO
ITO_CODPRODUTO
TB_FORNECEDOR
FOR_CODIGO
FOR_DESCRICAO
FOR_END
FOR_NUMERO
FOR_EMAIL
FOR_FONE
FOR_CGC
FOR_CODCIDADE
TB_MODALIDADEPRECO
MP_CODIGO
MP_DESCRICAO
MP_VALOR
MP_OBS
MOD_QDADEVAGAS
TB_FUNCIONARIO
FUN_CODIGO
FUN_DESCRICAO
FUN_NASCIM
FUNC_END
FUN_NUMERO
FUN_EMAIL
FUN_FONE
FUN_CELULAR
FUN_DATAADMISSAO
FUN_CODCARGO
TB_ORCAMENTO
ORC_CODIGO
ORC_DATA
ORC_CODCLIENTE
ORC_CODUSUARIO
TB_BOLETIM
BOL_CODIGO
BOL_DESCRICAO
BOL_NOTA
BOL_DATA
BOL_PERIODOINICIAL
BOL_PERIODOFINAL
BOL_CODCURSO
BOL_CODCATEGORIA
BOL_CODMATRICULA
TB_CARGO
CAR_CODIGO
CAR_FUNCAO
TB_CURSO
CUR_CODIGO
CUR_DESCRICAO
TB_HISTORICO
HIS_CODIGO
HIS_DESCRICAO
HIS_TIPO
TB_FUNC_CURSO
CUR_CODIGO
FUN_CODIGO
TB_CATEGORIA
CAT_CODIGO
CAT_DESCRICAO
TB_CAIXA
CAI_CODIGO
CAI_CODHISTORICO
CAI_MATCODIGO
CAI_MATCLIENTE
CAI_DATA
CAI_CODMOVIMENT
CAI_FORCODIGO
CAI_DESCRICAO
CAI_VALOR
CAI_CODCTPAGPRAZO
CAI_CODUSUARIO
TB_CURCAT
CURCAT_CODCURSO
CURCAT_CODCATEGORIA