

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
COORDENAÇÃO DO CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E
DESENVOLVIMENTO DE SISTEMAS

IVAN DE JESUS RIBEIRO
JOSÉ HENRIQUE REBUSKI

WEBFRAMING: UM GERENCIADOR DE ORÇAMENTO EM
CONSTRUÇÃO CIVIL

TRABALHO DE CONCLUSÃO DE CURSO

PONTA GROSSA

2015

IVAN DE JESUS RIBEIRO
JOSÉ HENRIQUE REBUSKI

**WEBFRAMING: UM GERENCIADOR DE ORÇAMENTO EM
CONSTRUÇÃO CIVIL**

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, da Coordenação de Análise e Desenvolvimento de Sistemas (COADS), da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. MSc. Luiz Rafael Schmitke

PONTA GROSSA

2015



TERMO DE APROVAÇÃO

WEBFRAMING: UM GERENCIADOR DE ORÇAMENTO EM CONSTRUÇÃO CIVIL

por

**IVAN DE JESUS RIBEIRO
JOSÉ HENRIQUE REBUSKI**

Este Trabalho de Conclusão de Curso (TCC) foi apresentado em 06 de novembro de 2015 como requisito parcial para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

MSc. Luiz Rafael Schmitke
Prof.(a) Orientador(a)

MSc. Denis Lucas Silva
Membro titular

MSc. Vinicius Camargo Andrade
Membro titular

- O Termo de Aprovação assinado encontra-se na Coordenação do Curso -

RESUMO

RIBEIRO, Ivan. Rebuski, José. **WEBFRAMING: Um Gerenciador De Orçamento Em Construção Civil**. 2015. 57 f. Trabalho de Conclusão de Curso Tecnologia em Análise e Desenvolvimento de Sistemas - Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2015.

Atualmente a tecnologia é ferramenta essencial para o gerenciamento de empresas em qualquer ramo, não utilizar é perder dinheiro. Pensando nisso a UpRise, uma empresa de desenvolvimento de projetos de construção civil, percebeu a necessidade de uma ferramenta para a otimização de seus processos de gerenciamento de orçamento de construções. A empresa utilizou alguns sistemas de gerenciamento onde não atendia suas necessidades, dentre eles, pode-se citar o Next Construção Civil, tendo vários módulos e com algumas limitações. Então chegamos a decisão de desenvolver um sistema simplificado e direto na necessidade da empresa, com isso surgiu o projeto Web Framing, para a gestão e organização do projeto adotou-se a metodologia de gerenciamento SCRUM.

Palavras-chave: Web Framing. Metodologia Ágil. SCRUM. Gestão de Projetos.

ABSTRACT

RIBEIRO, Ivan. Rebuski, José. **WEBFRAMING: A Budget Manager In Construction**. 2015. 57 f. Trabalho de Conclusão de Curso Tecnologia em Análise e Desenvolvimento de Sistemas - Federal Technology University - Paraná. Ponta Grossa, 2015.

Currently the technology is an essential tool for business management in any branch, do not use is to lose money. With this in mind the uprise, a civil construction projects development company, realized the need for a tool for optimizing their construction budget management processes. The company used some management systems which did not meet their needs, among which we can mention the Next Construction, with various modules and with some limitations. Then we come to decision to develop a simplified and direct system in need of company, with it came the Web Framing project for the management and project organization adopted the SCRUM management methodology.

Keywords: Web Framing. Agile methodology. SCRUM. Project management.

LISTA DE ILUSTRAÇÕES

Figura 1 - Relacionamento entre classes	11
Figura 2 - Instanciando um objeto	12
Figura 3 - Classes com os atributos encapsulados	12
Figura 4 - Polimorfismo	13
Figura 5 - Diagrama representando o conceito de Perry e Wolf.....	14
Figura 6 - Estrutura Model, View e Controller.....	17
Figura 7 - Comportamento dinâmico do MVC.	18
Figura 8 - Ciclo do Scrum.....	21
Figura 9 - Módulos Next Construção Civil	28
Figura 10 - Interface de Duplicata do sistema Next.....	28
Figura 11 - Interface de Consulta Contratos.....	29
Figura 12 - Interface de Consulta Contratos.....	29
Figura 13 - Interface de Insumos.....	30
Figura 14 - Interface de Conciliação Bancaria.....	31
Figura 15 - Interface de Dados de Obras	31
Figura 16 - Diagrama de Caso de Uso	34
Figura 17 - Modelo Relacional Web Framing	38
Figura 18 - Estrutura de pastas do sistema WebFraming	40
Figura 19 - Interface de Login Web Framing	41
Figura 20 - Interface Principal do sistema Web Framing.....	42
Figura 21 - Interface de Cadastro de Insumo	43
Figura 22 - Interface de Gerenciamento de Insumos	44
Figura 23 - Interface de Cadastrar Composição.....	45
Figura 24 - Interface de Gerenciar Composições.....	46
Figura 25 - Interface de Cadastrar Obra	47
Figura 26 - Interface de Gerenciar Obras.....	48
Figura 27 - Interface de Cadastrar Orçamento.....	49
Figura 28 - Interface de Gerenciar Orçamento.....	50
Figura 29 - Interface de Gerar Curva ABC	51

LISTA DE QUADROS

Quadro 1 - Valores da XP	20
Quadro 2 - <i>Sprints</i>	32
Quadro 3 - <i>Sprints</i> adicionais	32
Quadro 4 - Tarefas mais relevantes de cada <i>Sprint</i>	33
Quadro 5 - Descrição caso de uso Logar no Sistema	35
Quadro 6 - Descrição caso de uso Cadastrar Insumo.....	36
Quadro 7 - Descrição caso de uso Cadastrar Composição	36
Quadro 8 - Descrição caso de uso Cadastrar Obra	37
Quadro 9 - Descrição caso de uso Cadastrar Orçamento.....	37
Quadro 10 - Principais tabelas do sistema WebFraming	39

LISTA DE ABREVIATURAS E SIGLAS

OO	Orientação a Objetos
MVC	Model-View-Controller
C3	Chrysler Comprehensive Compensation
PHP	Personal Home Page
SGBD	Sistemas de Gerenciamento de Banco de Dados

SUMÁRIO

1 INTRODUÇÃO	9
1.1 OBJETIVOS	9
1.1.1 Objetivo Geral	9
1.1.2 Objetivos Específicos	10
2 FUNDAMENTAÇÃO TEÓRICA	11
2.1 ORIENTAÇÃO A OBJETOS	11
2.2 ARQUITETURA DE SOFTWARE	13
2.3 PADRÕES DE PROJETO DE SOFTWARE	15
2.3.1 Padrão MVC	16
2.4 METODOLOGIAS ÁGEIS	18
2.4.1 Extreme Programming	19
2.4.2 SCRUM	20
2.5 FRAMEWORKS DE DESENVOLVIMENTO DE SOFTWARE	22
2.5.1 CodeIgniter	23
2.5.2 Zend	24
2.5.3 Symfony	25
2.6 SISTEMAS DE GERENCIAMENTO DE BANCO DE DADOS RELACIONAIS	26
2.7 ESTUDO DE SISTEMAS SIMILARES	27
2.7.1 Next Construção Civil	27
2.7.2 Controle de Obra	29
3 DESENVOLVIMENTO	32
3.1 GERENCIAMENTO DO PROJETO	32
3.2 DIAGRAMAS	33
3.2.1 Caso de uso	33
3.2.2 Modelo Relacional	38
3.3 IMPLEMENTAÇÃO DO SOFTWARE	39
4 RESULTADOS	41
5 CONCLUSÃO	52
5.1 TRABALHOS FUTUROS	52
REFERÊNCIAS	53
APÊNDICE A - Descrição dos Casos de Uso do Sistema WebFraming	56
APÊNDICE B - Modelo Relacional do Sistema WebFraming	62

1 INTRODUÇÃO

Atualmente a tecnologia da informação tem facilitado à execução de atividades e processos nas mais diversas áreas. Com o avanço da tecnologia, surgem novas necessidades, solucionar problemas ou facilitar processos, vivemos conectados em uma grande rede de informações, buscando ferramentas simples e objetivas para melhorar nossas tarefas.

Visando aperfeiçoar o gerenciamento dos seus projetos, a UP Rise, uma empresa de desenvolvimento de projetos de construção civil, decidiu implantar um sistema informatizado para a gestão de suas obras e orçamentos, a fim de conseguir maximizar seus lucros com o controle dos insumos que possuem maior valor para a obra, por meio da Curva ABC. Conseguindo assim obter uma hierarquia dos insumos, gerando a prioridade para negociação e compra dos mesmos.

Tendo em vista que os *softwares* disponíveis no mercado possuíam grande complexidade, a empresa propôs o desenvolvimento de um sistema restrito unicamente as suas necessidades.

Buscando atender a necessidade da empresa e otimizar seu processo no gerenciamento de orçamento de construção civil, foi proposto um projeto *web* aplicando a metodologia de gerenciamento SCRUM para a gestão do projeto, a cumprimento dos prazos, no desenvolvimento utilizou o *framework Web CodeIgniter* para linguagem PHP com o banco de dados MySQL para o armazenamento.

1.1 OBJETIVOS

Neste capítulo será abordado o objetivo geral do trabalho, bem como os objetivos específicos.

1.1.1 Objetivo Geral

Implementar um sistema *Web* para gestão de orçamento na construção civil aplicando padrões de projeto de *software* e a metodologia ágil SCRUM.

1.1.2 Objetivos Específicos

- Procurar e analisar sistemas similares a fim de levantar requisitos inerentes a sistemas de controle de orçamento na construção civil;
- Verificar os padrões de projeto de *software* que podem ser utilizados para desenvolvimento de aplicações *Web*;
- Aplicar a metodologia SCRUM na gestão do projeto de *software*;
- Aplicar o uso de um *framework Web* PHP em conjunto com um banco de dados relacional.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo aborda os conceitos utilizados no desenvolvimento do sistema WebFraming. A seção 2.1 apresenta orientação a objetos. A seção 2.2 apresenta arquitetura de software. A seção 2.3 discorre sobre padrões de projeto de software. A seção 2.4 aborda as metodologias ágeis para o desenvolvimento de software. A seção 2.5 apresenta frameworks de desenvolvimento de software. A seção 2.6 apresenta sistemas de gerenciamento de banco de dados relacional. E por fim na seção 2.7 o estudo dos sistemas similares.

2.1 ORIENTAÇÃO A OBJETOS

Como definido por GAMMA et al. (2000), programas orientados a objetos (OO) são um conjunto de classes que definem os objetos presentes. Uma classe comporta os dados e os procedimentos que operam sobre os dados. Os procedimentos são normalmente chamados de métodos ou operações. Um objeto executa um método quando recebe um pedido (ou mensagem) de um cliente. As requisições de métodos para obter um objeto são restritas, o estado interno de um objeto está encapsulado, e não pode ser acessado diretamente.

Através do relacionamento de herança, como mostra a Figura 1, as instâncias dos objetos formam uma hierarquia de classes (VINCENZI, 2004).

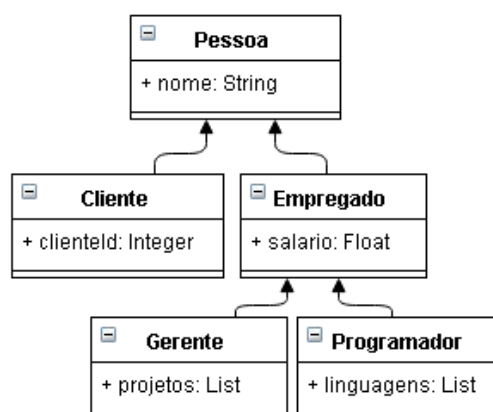


Figura 1 - Relacionamento entre classes
Fonte: Autoria própria.

Uma classe é um modelo ou *template* estático utilizado para abstrair as características intrínsecas de um determinado grupo de objetos semelhantes, tendo

seus dados armazenados em atributos e as suas ações e operações definidas como métodos (DALL'OGGIO, 2007). A instanciação de um objeto, exibido na Figura 2, é realizada em tempo de execução, sendo assim, toda vez que um novo objeto é criado, são copiados os dados definidos na classe, tendo o estado e comportamento encapsulados (VINCENZI, 2004).

```

$empregado = new Empregado();
$empregado->nome = 'João';
$empregado->salario = 3100;

```

Figura 2 - Instanciando um objeto
Fonte: Autoria própria.

O encapsulamento é um mecanismo utilizado para prover proteção de acesso aos atributos e métodos de um objeto por meio da indicação do nível de visibilidade do membro (DALL'OGGIO, 2007), garantindo assim a ocultação da informação onde a interface e a implementação da classe estão separadas sintaticamente, pois apenas os métodos pertencentes ao objeto podem acessar os dados encapsulados.

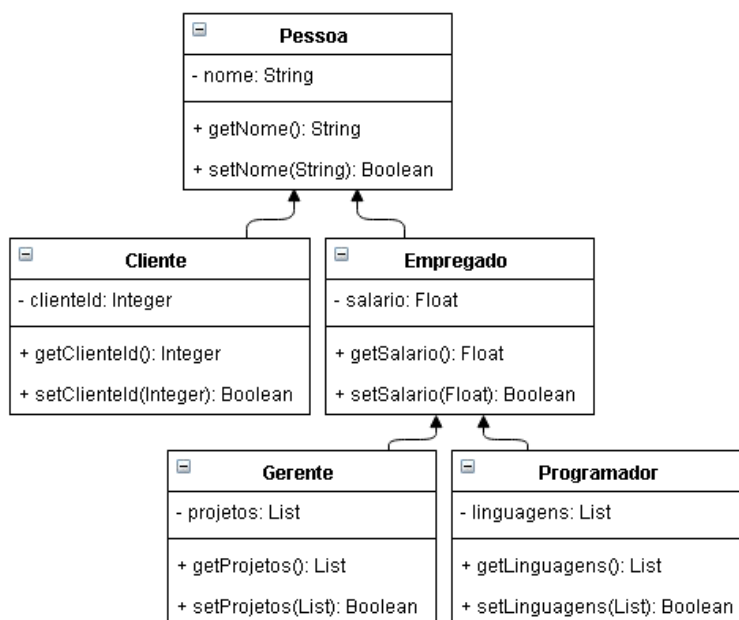


Figura 3 - Classes com os atributos encapsulados
Fonte: Autoria própria.

Outra característica de projetos OO é o polimorfismo, o qual remete a capacidade de assumir muitas formas. Na orientação a objeto o polimorfismo é a capacidade de classes derivadas de uma determinada classe-base tenham métodos

com a mesma assinatura, como mostra a Figura 4, mas trazendo resultados diferentes, devido ao processamento ser individual em cada classe filha (VINCENZI, 2004).

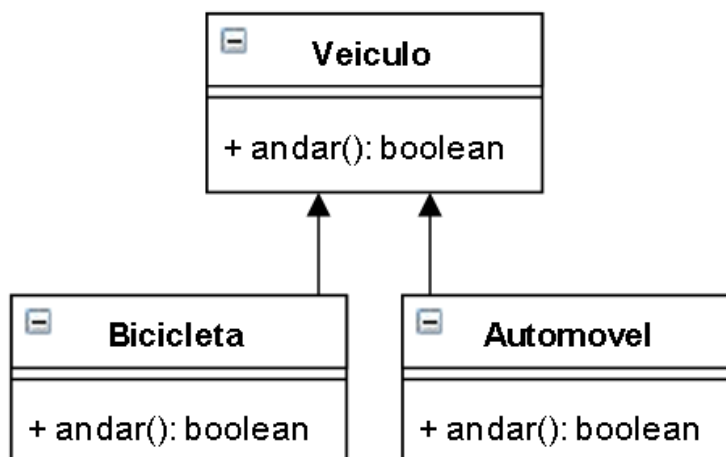


Figura 4 - Polimorfismo
Fonte: Autoria própria.

2.2 ARQUITETURA DE SOFTWARE

Perry e Wolf definem a arquitetura de *software* como um conjunto de elementos arquiteturais que possuem alguma forma de organização. Os elementos, bem como sua organização, são definidos por decisões tomadas para atender objetivos e restrições. Três tipos de elementos arquiteturais podem ser destacados (PERRY; WOLF, 1992):

Elementos de processamento: usam ou transformam informação;

Elementos de dados: contem a informação a ser usada e transformada;

Elementos de conexão: ligam elementos de qualquer tipo entre si.

A organização tem como função determinar as relações entre os elementos que compõem a arquitetura do *software*. Essas relações possuem propriedades utilizadas para definir como os elementos devem interagir para que os objetivos do sistema sejam atendidos. No processo de seleção de alternativas, essas relações devem ser ponderadas a fim de indicar sua relevância.

A Figura 5 ilustra os elementos arquiteturais conceituado por Perry e Wolf (1992), onde os elementos de processamento através da lógica do negócio

manipulam os elementos de dados, utilizando os elementos de conexão para acessá-los.

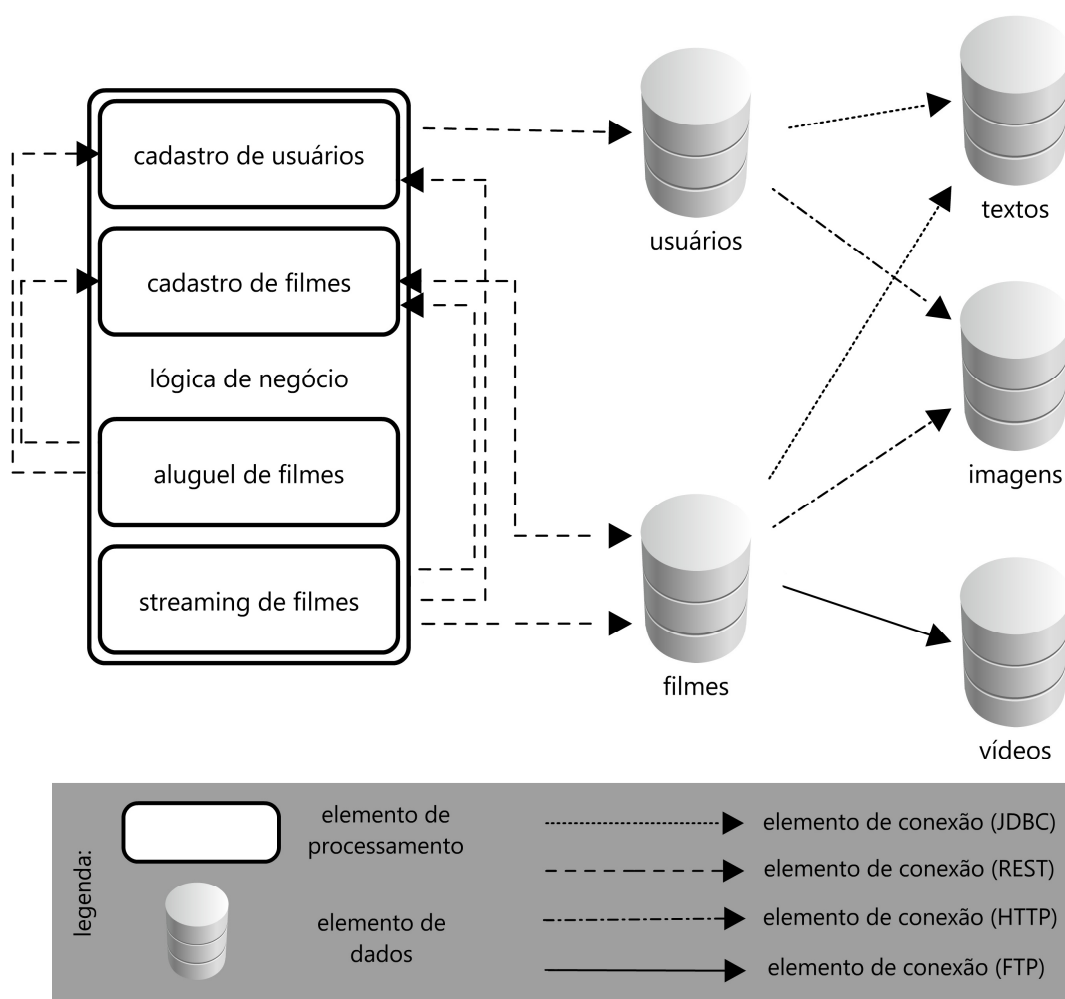


Figura 5 - Diagrama representando o conceito de Perry e Wolf
Fonte: Germoglio (2008).

Para Garlan e Shaw (1994), a necessidade de se adotar arquitetura de *software* no desenvolvimento de um sistema se torna necessária a partir do momento em que há um crescimento tanto no tamanho, quanto na complexidade do sistema a ser desenvolvido, ficando mais explícito os motivos de se aplicar a arquitetura de *software* no projeto de sistemas. Sendo assim, a arquitetura de *software* não se limita apenas a definição dos algoritmos e estruturas de dados, mas também envolve decisões sobre as estruturas do sistema, a estrutura global de controle, os protocolos de comunicação, sincronização e acesso aos dados, atribuição de funcionalidade ou distribuição física dos elementos do sistema, entre outros atributos de qualidade, que poderão impactar no comportamento do sistema em termos de escala e desempenho.

Através da definição de arquitetura de *software* de Garlan e Shaw, observa-se três aspectos importantes, são eles:

- Aplicar os conhecimentos da arquitetura de *software* quando trabalhar com grandes sistemas.
- Separar as tarefas entre o design detalhado, sendo algoritmos e estruturas de dados, e design arquitetural, sendo os elementos e organização do sistema em relação à estrutura do sistema, controle, comunicação ou implantação.
- Para alcançar escalabilidade ou desempenho no sistema, o processo de design da arquitetura deve se preocupar com atributos de qualidade.

2.3 PADRÕES DE PROJETO DE SOFTWARE

Os estudos sobre padrões de projeto tiveram início com o arquiteto Christopher Alexander, com a publicação do livro “*A pattern language*” em 1977. Neste livro, são abordados mais de 250 padrões construtivos, onde cada um desses padrões era como um pequeno manual arquitetônico, para essas repetições arquiteturais deu-se o nome de *pattern*. A motivação para criação desse catálogo era que, segundo o autor, os métodos tradicionais de arquitetura não atendiam corretamente às necessidades da sociedade, seu objetivo era melhorar a qualidade de vida das pessoas (Jandl, 2003).

“Um padrão descreve um problema que ocorre inúmeras vezes em determinado contexto, e descreve ainda a solução para esse problema, de modo que essa solução possa ser utilizada sistematicamente em distintas situações.” (ALEXANDER et al., 1977)

Os padrões de projetos de *software* tiveram origem no final da década de 80, quando Ward Cunningham e Kent Beck criaram um conjunto de padrões em Smalltalk, para serem utilizados no desenvolvimento de interfaces. Paralelamente, Jim Coplien desenvolveu um catálogo de padrões em C++, chamado idiomas. Nesse mesmo período, Erich Gamma reconhecia a importância de se agrupar as estruturas de projetos que se repetiam com frequência, enquanto trabalhava em sua tese de doutorado sobre desenvolvimento de *software* orientado a objeto.

Segundo Pressman (2011) padrões de projeto de *software* pode também considerar um conjunto de influências de projeto, descrevendo os requisitos não

funcionais, facilitando a manutenção e a portabilidade associadas ao *software* ao qual o padrão deve ser aplicado. Em essência, as influências de projeto descrevem o ambiente e as condições que devem existir para tornar o padrão de projeto aplicável.

Segundo Pressman (2011), padrões de projetos podem ser usados durante todo o projeto de software.

Tais padrões podem atender a um elemento específico do projeto como um conjunto de componentes a fim de resolver algum determinado problema do sistema, ou relacionamentos entre componentes ou ainda os mecanismos para efetuar a comunicação entre componentes.

“Cada padrão tem uma característica diferente para ajudar em algum lugar onde se precisa de mais flexibilidade ou precisa encapsular uma abstração ou de se fazer um código menos casado.” (GAMMA et al., 2000)

2.3.1 Padrão MVC

A abordagem *Model View Controller* (MVC) consiste basicamente na separação do sistema em três tipos de objetos, ou camadas como mostrado por DALL'OGGIO (2007). *Model* é o objeto de aplicação, ou seja, é a camada responsável por fazer a persistência dos dados e ainda a regra de negócio do sistema. A *View* é a apresentação da interface, é nesta camada em que se localizam a tela onde o usuário poderá fazer as ações. Por fim o *Controller* define a maneira como a interface do usuário reage às iterações destes, invocando o *Model* a fim de manipular os dados.

Antes da utilização do padrão MVC, os projetos de interface para o usuário tendiam a agrupar esses objetos, então, com a adoção do padrão MVC, passou-se a separar as interfaces da regra de negócio a fim de aumentar a flexibilidade e a reutilização (BUSCHMANN et al. 1996).

Cada *View* tem um componente *Controller* associado, conforme ilustra a Figura 6. *Controller* recebe a entrada de mensagens, geralmente como eventos que codificam o movimento ou o click do mouse, ou ainda a entrada do teclado. Os eventos são convertidos em chamadas para serviços do *Model* ou pela *View*. O usuário interage com o sistema unicamente através da *View*.

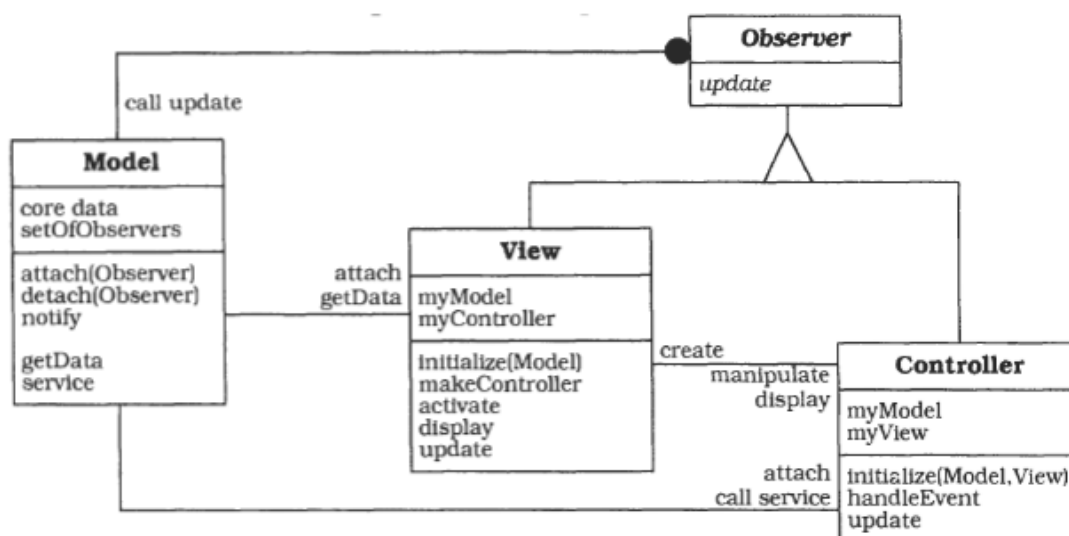


Figura 6 - Estrutura Model, View e Controller
 Fonte: Buschmann et al. (1996)

A separação em *Model*, *View* e *Controller* permite múltiplas visões do mesmo *Model*. Se o usuário alterar o *Model* através do *Controller* de uma *View*, todas as outras *Views* dependentes sobre estes dados devem refletir as alterações. Por isso, o *Model* notifica todas as *Views* sempre suas alterações de dados. A *View*, por sua vez, obtém novos dados do *Model* para exibir novas informações.

Como ilustrado na Figura 7 o comportamento dinâmico do MVC. Para simplificar apenas um par *View-Controller* é mostrado nos diagramas. Como interação do usuário que resultando em alterações no *Model* acionando o mecanismo *change-propagation*:

- O *Controller* aceita entrada do usuário no seu procedimento de tratamento de eventos, interpreta o evento, e ativa um procedimento do modelo de serviço.
- O *Model* executa o serviço solicitado. Isso resulta em uma mudança para os seus dados internos.
- O *Model* notifica todas as *Views* e *Controller* registrados com o mecanismo *change-propagation*, chamando os seus procedimentos de atualização.
- Cada *View* solicita os dados alterados a partir do *Model* e exibindo novamente na tela. Cada *Controller* registrado recupera dados do *Model* para habilitar ou desabilitar certas funções de usuário. Por exemplo,

permitindo a entrada de menu para salvar dados pode ser uma consequência de alterações aos dados do *Model*.

- O *Controller* original recupera o controle e retorna de seu procedimento *event handling*.

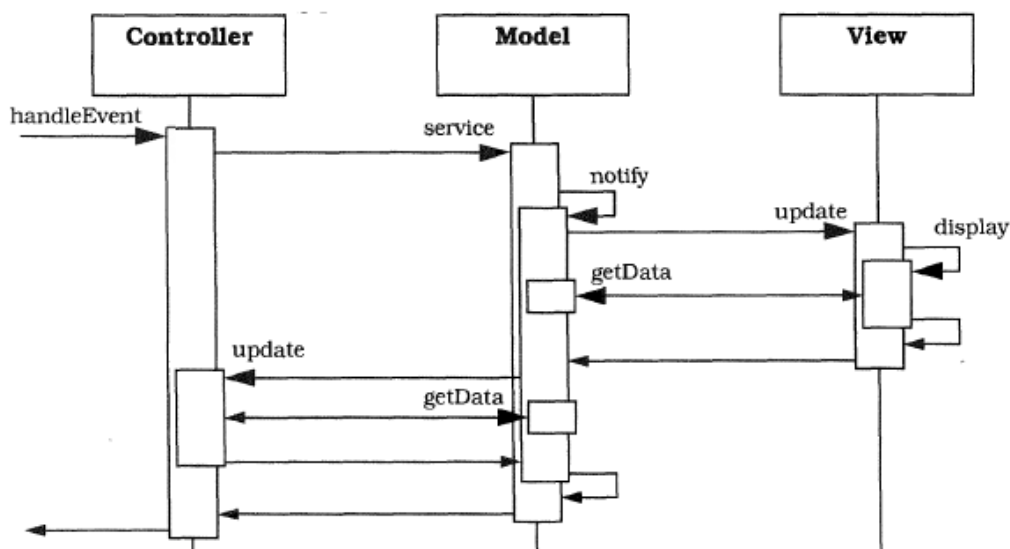


Figura 7 - Comportamento dinâmico do MVC.
Fonte: Buschmann et al. (1996)

2.4 METODOLOGIAS ÁGEIS

O termo “Metodologias Ágeis para o desenvolvimento de software” difundiu-se em 2001 quando dezessete especialistas no desenvolvimento de *software*, posteriormente conhecida como a “Aliança Ágil”, assinaram o Manifesto para o desenvolvimento ágil de *software* (PRESSMAN, 2011).

Por meio do Manifesto Para o Desenvolvimento Ágil de Software passou-se a valorizar:

- **Indivíduos e Interações** em vez de processos e ferramentas;
- **Software Funcionando** em vez de documentação abrangente;
- **Colaboração de cliente** em vez de negociação de contratos;
- **Resposta a Modificações** em vez de seguir um plano;

O “Manifesto Ágil” (2001) mostra que processos, ferramentas, documentação, negociação de contratos e planejamento tem importância secundária quando

comparado com os indivíduos e iterações, *software* funcionando, colaboração do cliente e resposta rápidas a mudanças e alterações. Esse conceito abrange como pequenas e medias organizações trabalham e executam seus processos.

2.4.1 Extreme Programming

No projeto *Chrysler Comprehensive Compensation (C3)*, em 1996, Kent Beck propôs e aplicou a metodologia Extreme Programming (XP) pela primeira vez, finalizando um trabalho em pouco mais de um ano, após vários anos usando metodologias tradicionais sem sucesso (MOREIRA, 2011).

A XP é uma metodologia ágil para equipes pequenas e médias que desenvolvem *software* baseado em requisitos vagos e que se modificam rapidamente (BECK, 2004). No Quadro 1 Beck (2004) define um conjunto de cinco valores que embasam o trabalho realizado utilizando o XP.

Valores	Descrição
Comunicação	<ul style="list-style-type: none"> - Colaboração estreita da equipe; - Estabelecimento de metáforas;
Simplicidade	<ul style="list-style-type: none"> - Projetar para as necessidades imediatas; - Facilidade de implementação;
Feedback	<ul style="list-style-type: none"> - Principais fontes: Software implementado, Cliente e equipe de desenvolvimento; - levantamento de novas necessidades sugeridas no decorrer do projeto;
Coragem	<ul style="list-style-type: none"> - Disciplina para aprende a projetar para “Hoje”; - Reconhecimento de que as necessidades futuras podem mudar drasticamente;
Respeito	<ul style="list-style-type: none"> - Respeito entre todos os membros envolvidos no projeto e com o próprio

	<i>software</i> ; - Entrega com sucesso de incrementos de <i>software</i> aumenta o respeito da equipe pelo processo SP
--	--

Quadro 1 - Valores da XP.

Fonte: Beck (2004) e Pressman (2011, p.87). Adaptado pelos autores.

Ramos (2013) recomenda o uso da metodologia XP para desenvolvimento de projetos onde os requisitos não estão totalmente levantados, pois podem ocorrer mudanças constantes, e que a equipe de desenvolvedores não tenha mais que doze integrantes, facilitando o desenvolvimento iterativo utilizando programação orientada a objetos.

2.4.2 SCRUM

Foi desenvolvido no início da década de 90 por Jeff Sutherland, tendo como referência o trabalho de Nonaka e Takeuchi, e sendo formalizado em 1995 por Ken Schwaber. Neste modelo a equipe trabalha em conjunto e possui autonomia para definir a estratégia usada para alcançar seus objetivos (BASSI FILHO, 2008).

Possui grande ênfase no gerenciamento do projeto, sem definir como a equipe irá desenvolver, a fim de garantir a qualidade do produto propõem pequenas reuniões diárias para monitorar e acompanhar todo o grupo, auxiliando em eventuais problemas que impeçam a continuidade do desenvolvimento (SCHWABER, 2004).

O *Scrum* adota um ciclo de vida iterativo, representado na Figura 8, cada uma dessas iterações recebe o nome de *Sprint*. No início de cada *Sprint* é realizada uma reunião de planejamento (*Sprint Planning Meeting*), em que a equipe, o *Scrum Master* e o *Product Owner* definem o que deverá ser implementado (*Selected Product Backlog*), com isso a equipe consegue definir o *Sprint Backlog*, que são as tarefas necessárias para que se consiga desenvolver as funcionalidades selecionadas. A partir desse momento começa a execução da *Sprint*, diariamente são realizadas reuniões (*Daily Scrum Meeting*) com objetivo de localizar possível impedimentos no desenvolvimento e para a equipe dizer o que fez nos últimos dias e o que irão fazer nos próximos. No final do *Sprint* é realizada uma reunião para que sejam apresentadas pela equipe as alterações que foram realizadas durante a *Sprint*

(*Sprint Review Meeting*), e para finalizar é feita uma reunião para que sejam observados pontos de melhora (*Sprint Retrospective Meeting*) (SCHWABER, 2004).

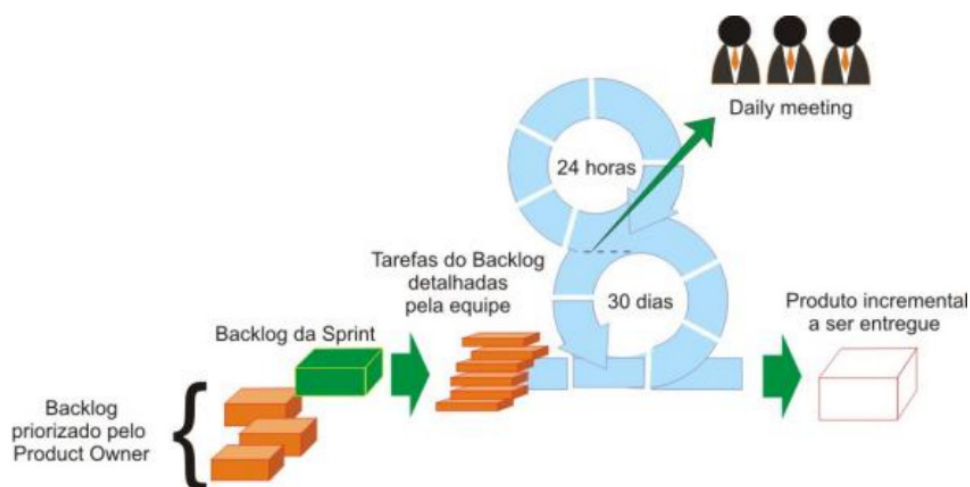


Figura 8 - Ciclo do Scrum
Fonte: Pereira, Torreão e Marçal. (2007)

Segundo Schwaber (2004) o Scrum é caracterizado por três papéis:

- *Product Owner*: gerencia os requisitos do produto, bem como a prioridade de desenvolvimento, a fim de entregar as principais funcionalidades rapidamente;
- Equipe: são os desenvolvedores do sistema, devem trabalhar unidos para alcançarem os objetivos;
- *Scrum Master*: realiza o acompanhamento da equipe e do desenvolvimento para garantir que tudo esteja planejado e seja executado;

Schwaber (2004) também definiu alguns artefatos que são usados como apoio durante todo o processo de desenvolvimento com Scrum. São eles:

- *Product Backlog*: lista de requisitos do sistema gerenciada pelo *Product Owner*, pode ser alterada sempre que for necessário;
- *Selected Product Backlog*: é um conjunto de requisitos levantados no *Product Backlog*, selecionados em conjunto com o cliente, para serem implementados no *sprint* atual;
- *Sprint Backlog*: são as tarefas necessárias para desenvolver os itens escolhidos em *Selected Product Backlog*;

- *Impediment Backlog*: lista de impedimentos encontrados pela equipe;
- *Product Burndown Chart*: gráfico que mostra a evolução do trabalho em relação ao *Product Backlog*.
- *Sprint Burndown Chart*: gráfico que mostra a evolução diária do trabalho em relação ao *Sprint Backlog*.

Algumas reuniões protocolares foram definidas por Schwaber (2004) como práticas a serem adotadas:

- *Sprint Planning Meeting*: reunião de planejamento onde todos os membros envolvidos no projeto devem estar presentes, primeiramente o *Product Owner* lista os itens de maior prioridade do *Product Backlog*, e em conjunto com a equipe é definido o *Selected Product Backlog*, após isso a equipe cria as tarefas necessárias gerando o *Sprint Backlog*;
- *Daily Scrum Meeting*: reunião rápida e diária entre a equipe e o *Scrum Master*, cada membro da equipe deve dizer o que fez no último dia, e o que fará no próximo, e se existem impedimentos no seu caminho;
- *Sprint Review Meeting*: reunião ao final de cada *Sprint*, para a equipe mostrar o que já foi desenvolvido, demonstrando as novas funcionalidades para o *Product Owner*;
- *Sprint Retrospective Meeting*: é uma extensão da *Sprint Review Meeting*, e serve para a equipe identificar formas de melhorar nos próximos *Sprints*;

2.5 FRAMEWORKS DE DESENVOLVIMENTO DE SOFTWARE

Um *framework* é uma arquitetura de *software* que a partir de um determinado contexto, por meio de abstrações, consegue prover uma estrutura e um comportamento genéricos para implementação (PRESSMAN, 2011).

Segundo a definição de Coad, *framework* é um modelo de classes, objetos e relacionamentos utilizados para desenvolver aplicações específicas (COAD, 1992) e, segundo Johnson, é um conjunto de classes tanto abstratas como concretas que fornecem uma estrutura genérica para solucionar determinados problemas

(JOHNSON, 1988). As classes podem ser específicas de uma aplicação ou podem pertencer a uma determinada biblioteca (MALDONADO et al., 2001).

Frameworks atuam quando há necessidade de reutilização, não ficando restrito apenas a componentes isolados, mas também podem fornecer toda arquitetura necessária para o desenvolvimento de um domínio específico (MALDONADO et al., 2001).

Por meio de códigos-fonte, classes, funções e metodologias, os *frameworks* tem como principal objetivo a reusabilidade, tornando o desenvolvimento mais ágil e fácil (SUPPI, 2012).

Com o objetivo de obter uma maior reusabilidade e manutibilidade, garantindo assim mais produtividade e qualidade no desenvolvimento de *softwares*, muitos *frameworks* têm sido desenvolvidos nas últimas décadas (MALDONADO et al., 2001).

Segundo Zemel (2009), existem muitas vantagens quanto a utilização de *frameworks*, as quais estão descritas abaixo:

- Utilidade: Um dos objetivos principais dos *frameworks* é auxiliar no desenvolvimento de aplicações e *softwares*.
- Segurança: Os *frameworks* mais eficazes são projetados de modo a garantir a segurança de quem programa.
- Extensibilidade: Os *frameworks* permitem estender funcionalidades nativas.
- Economia de tempo: As funções mais complexas, que precisam de mais tempo na realização, com o uso de um *framework* pode-se realizar-se em pouquíssimo tempo.
- Ajuda fácil: Os desenvolvedores de *frameworks* geralmente disponibilizam material de qualidade nos *Websites*, com uma vasta documentação a respeito. Além disso, a maioria dos *frameworks* tem uma comunidade de desenvolvedores dispostos a se ajudarem entre si.

2.5.1 CodeIgniter

O CodeIgniter foi desenvolvido por Rick Ellis, em 2006. Devido ao fato de os *frameworks* da época possuírem pouca documentação, e baixa qualidade do

software, motivaram Ellis a criar o CodeIgniter com uma ampla documentação e uma vasta quantidade de funções (GABARDO, 2010).

Por meio de uma estrutura simples de fácil entendimento, e um grande conjunto de bibliotecas permite ao desenvolvedor ganhar agilidade no processo de desenvolvimento, minimizando a escrita de código apenas ao que compete a aplicação, tendo o CodeIgniter como responsável pela arquitetura do sistema (GABARDO, 2010).

Devido a facilidade de instalação, em um primeiro momento não seja possível notar toda a capacidade do CodeIgniter, porém ele conta com uma grande quantidade de bibliotecas que provêm muitos métodos e funções extremamente úteis (ANDRADE, 2011).

Segundo Andrade (2011), o CodeIgniter foi criado com os seguintes objetivos:

- Instanciamento Dinâmico: apenas as requisições HTTP irão disparar os eventos. Sendo assim, os componentes e as rotinas são carregados e executados apenas quando requisitados.
- Acoplamento flexível: para o sistema ser reutilizável e flexível, os componentes dependem pouco um do outro.
- Singularidade do Componente: a fim de serem as mais úteis possíveis, cada classe e suas funções são altamente autônomas.

2.5.2 Zend

O Zend Framework foi criado e é mantido por Zend Technologies, um fornecedor de *software* comercial cujos fundadores, Andi Gutmans e Zeev Suraski, também foram responsáveis pela primeira grande reformulação do analisador PHP, lançado como PHP 3.0 em 1997. A primeira versão do Zend Framework, v1.0, foi lançado em julho de 2007 e era possível trabalhar com armazenamento em cache, autenticação, gerenciamento de configuração de acesso de banco de dados, RSS e Atom geração de *feeds*, e localização (ZEND, 2015).

Ele fornece uma implementação completa do padrão *Model-View-Controller* (MVC), que permite separar a lógica de negócios do aplicativo da interface de usuário e modelos de dados. Este padrão é recomendado para aplicações de média

a grande complexidade e é comumente usado para o desenvolvimento de aplicações *Web*, uma vez que incentiva a reutilização do código e produz uma estrutura de código mais gerenciável (ZEND, 2015).

Contém diversos componentes que são divididos em cinco categorias: MVC; desenvolvimento rápido de aplicações; banco de dados; internacionalização e localização; autenticação, autorização e gerenciamento de sessão (SUPPI, 2012).

Possui fraco acoplamento, ou seja, a maioria dos componentes que o compõem, funcionam de forma independente. Contribuindo para que os sistemas sejam mais leves, contendo apenas os recursos necessários para atender os requisitos (ZEND, 2015).

O Zend foi desenvolvido com o objetivo de ser um *framework* simples, capaz de prover agilidade e aumentar a produtividade no desenvolvimento de *softwares*. Para isso, busca trazer os recursos mais recentes em desenvolvimento *Web*. Além disso, conta com cobertura de testes desde o começo, a fim de garantir a qualidade do *framework*. Por meio das licenças do Zend, é possível construir um software proprietário, tendo assim os direitos de propriedade intelectual garantido (SUPPI, 2012).

2.5.3 Symfony

O Symfony é um *framework* PHP cujo o foco é a agilidade e produtividade no desenvolvimento e manutenção de aplicações *web*, bem como a reusabilidade de códigos, utilizando-se de implementações genéricas (SYMFONY, 2015).

Conforme seu site oficial (SYMFONY, 2015), do ponto de vista estratégico, existem seis motivos para a utilização deste *framework*, os quais estão dispostos abaixo:

- Reputação: possui uma comunidade ativa, recebendo o reconhecimento por desenvolver um *framework* estável;
- Permanência: desenvolvido a dez anos, possui uma base de suporte consolidada;
- Referências: sites de grandes corporações utilizam o Symfony, dentre elas destaca-se o Yahoo;

- Inovação: possui velocidade, flexibilidade e componentes reutilizáveis a fim de buscar uma melhoria contínua no desenvolvimento;
- Recursos: vasta coleção de recursos, todos com sua devida documentação;
- Interoperabilidade: não há necessidade de utilizar o *framework* inteiro para desenvolver, tendo liberdade para construir;

2.6 SISTEMAS DE GERENCIAMENTO DE BANCO DE DADOS RELACIONAIS

Antes do surgimento dos bancos de dados as informações eram gravadas em arquivos, contudo essa forma de armazenamento de dados traz alguns problemas, como a falta de uma estrutura definida de acesso, tanto para determinados dados contidos no arquivo, como para o acesso do usuário às informações (DAMAS, 2007).

Apenas em meados da década de 60 os sistemas de arquivos evoluíram com o surgimento dos primeiros Sistemas de Gerenciamento de Banco de Dados (SGBD), inicialmente com modelos hierárquicos e em redes (RAMAKRISHNAN; GEHRKE, 2008), apenas na década seguinte foi proposto um modelo de banco de dados relacional baseado na teoria matemática dos conjuntos (CODD, 1970), posteriormente esse modelo deu origem ao Modelo Entidade Relacionamento (CHEN, 1976).

Devido ao fato do modelo hierárquico armazenar as informações em forma de árvore, gerando assim uma dificuldade em acessar dados de forma aleatória, surgiu como alternativa o modelo em rede, permitindo que um mesmo registro possua várias associações por meio de links, facilitando assim o acesso a dados aleatórios. Contudo também apresenta limitações, principalmente de implementação. Com isso, o modelo relacional, mesmo inicialmente restrito ao meio acadêmico e com as limitações de hardware da época, se popularizou e tornou-se amplamente usado (DAMAS, 2007).

Um banco de dados estrutura os dados em tabelas, as quais normalmente são armazenadas em um ou mais arquivos associados. A existência de referências cruzadas entre as tabelas caracteriza o banco de dados como relacional. (KOFLER, 2005).

O Modelo Entidade Relacionamento é representado por entidades, que possuem atributos e através da representação da cardinalidade podem possuir relacionamentos entre si (CHEN, 1976).

Ramakrishnan e Gehrke (2008) citam as vantagens em utilizar um SGBD, as quais estão listadas:

- Independência de Dados: provê uma visão abstrata dos dados, ocultando do aplicativo toda representação e armazenamento dos dados.
- Acesso Eficiente aos Dados: deve ser eficiente quanto ao acesso aos dados, especialmente quando os dados estão armazenados em um dispositivo externo.
- Integridade e Segurança dos Dados: consegue garantir a integridade dos dados que estão sendo acessados ou salvos por meio de rotinas definidas pelo usuário.
- Administração dos Dados: centralização da administração dos dados, facilitando o controle de acesso.
- Acesso Concorrente e Recuperação de Falha: transparência na concorrência de dados e nas recuperações de falhas.
- Tempo Reduzido de Desenvolvimento de Aplicativo: interface de alto nível.

2.7 ESTUDO DE SISTEMAS SIMILARES

Nesta seção iremos estudar os sistema de orçamento Next Construção Civil e Controle de Obras, a fim de levantar requisitos, módulos e funcionalidades que façam parte deste domínio.

2.7.1 Next Construção Civil

O Next Construção Civil é uma solução de *Software* de Gestão Empresarial (ERP) voltada à indústria da construção. Este sistema atende todos os processos envolvidos no dia a dia de uma construtora, o sistema gerencia desde o planejamento da obra até a comercialização ao cliente, prestando o ferramental necessário para a evolução racional e otimizada de cada processo. Possui forte

ênfoque em Controle e Análise de Custos, Gerenciamento de Compras, Controle Financeiro e Contábil, atendendo a legislação tributária vigente da Construção Civil.

O ERP é dividido em seis módulos, conforme ilustrado na Figura 9, no módulo de engenharia possui o controle de orçamentos, requisito que faz parte do escopo do sistema WebFraming.



Figura 9 - Módulos Next Construção Civil
Fonte: BRASSOFT (2015)

A Figura 10 mostra a interface de Duplicata.

Dados da Duplicata		Datas		Juros	
Número	29/8	Emissão	05/05/2010	Tipo	Nenhum
Parcela	7/60	Vencimento	05/12/2010	Data Ini Juro	05/05/2010
Tipo	M - MENSALS	Previsão	05/12/2010	Taxa Mês	0,0000
Fluxo	1 - MENSALS	Ult. Correção	01/12/2010		
Status	Paga	Ult. Movimento	06/12/2010		
Moeda: 5 - IGP-M		Valores			
Localização: Local 0 - CAIXA GERAL, Oper 0 - COBRANÇA PROPRIA, Cod Bco		Original	2.800,00	Correção	189,07
		Moeda	578,6217849	Juros	0,00
		Cornag	0,00	Total	2.989,07
		Jornag	0,00	Recebido	2.989,07
		Seguro	0,00	Saldo	0,00

Figura 10 - Interface de Duplicata do sistema Next
Fonte: BRASSOFT (2015)

Na figura 11 mostra a interface de Consulta Contratos.

CONSULTA CONTRATOS

Código: 29

Executa Encerra

Contrato	Unidades	Parcelas	Parcelas Vencidas	História	Razão	Contatos	Observações	Atributos	Outros
Número	Seq	Parcela	Tipo	Vencimento	Moeda	Valor Corrigido	Valor Recebido	Saldo Local	Status
29/1	1	1/1	ENTRADA	05/05/2010	REAL	R\$ 90.000,00	R\$ 90.000,00	R\$ 0,00	0 Paga
29/2	2	1/60	MENSAIS	05/06/2010	IGP-M	R\$ 2.833,32	R\$ 2.833,32	R\$ 0,00	0 Paga
29/3	3	2/60	MENSAIS	05/07/2010	IGP-M	R\$ 2.857,40	R\$ 2.857,40	R\$ 0,00	0 Paga
29/4	4	3/60	MENSAIS	05/08/2010	IGP-M	R\$ 2.861,69	R\$ 2.861,69	R\$ 0,00	0 Paga
29/5	5	4/60	MENSAIS	05/09/2010	IGP-M	R\$ 2.883,73	R\$ 2.883,73	R\$ 0,00	0 Paga
29/6	6	5/60	MENSAIS	05/10/2010	IGP-M	R\$ 2.916,89	R\$ 2.916,89	R\$ 0,00	0 Paga
29/7	7	6/60	MENSAIS	05/11/2010	IGP-M	R\$ 2.946,35	R\$ 2.946,35	R\$ 0,00	0 Paga
29/8	8	7/60	MENSAIS	05/12/2010	IGP-M	R\$ 2.989,07	R\$ 2.989,07	R\$ 0,00	0 Paga
29/9	9	1/10	INTERMEDIARIAS	05/12/2010	IGP-M	R\$ 10.942,12	R\$ 10.942,12	R\$ 0,00	0 Paga
29/10	10	8/60	MENSAIS	05/01/2011	IGP-M	R\$ 2.989,07	R\$ 0,00	R\$ 2.989,07	0 Ativa
29/11	11	9/60	MENSAIS	05/02/2011	IGP-M	R\$ 2.989,07	R\$ 0,00	R\$ 2.989,07	0 Ativa
29/12	12	10/60	MENSAIS	05/03/2011	IGP-M	R\$ 2.989,07	R\$ 0,00	R\$ 2.989,07	0 Ativa
29/13	13	1/1	CHAVES	15/03/2011	IGP-M	R\$ 21.350,49	R\$ 0,00	R\$ 21.350,49	0 Ativa
29/14	14	11/60	MENSAIS	05/04/2011	IGP-M	R\$ 2.989,07	R\$ 0,00	R\$ 2.989,07	0 Ativa
29/15	15	12/60	MENSAIS	05/05/2011	IGP-M	R\$ 2.989,07	R\$ 0,00	R\$ 2.989,07	0 Ativa
29/16	16	13/60	MENSAIS	05/06/2011	IGP-M	R\$ 2.989,07	R\$ 0,00	R\$ 2.989,07	0 Ativa
29/17	17	2/10	INTERMEDIARIAS	05/06/2011	IGP-M	R\$ 10.942,12	R\$ 0,00	R\$ 10.942,12	0 Ativa
29/18	18	14/60	MENSAIS	05/07/2011	IGP-M	R\$ 2.989,07	R\$ 0,00	R\$ 2.989,07	0 Ativa
72						399.480,85	121.230,57	278.250,28	

(Status >> Renegociada)

Encerra

Figura 11 - Interface de Consulta Contratos
Fonte: BRASSOFT (2015)

Na figura 12 mostra a interface de Consulta de Ordens de Compra.

CONSULTA DE ORDENS DE COMPRA

Código: 10752

Executa Encerra

Ordem de Compra | Itens | História | Documentos | Endereços | Carimbos | Observações | Outros

Empresa 1 - DEMONSTRAÇÃO LTDA
Emprnd. 2 - RESIDENCIAL BARRA DOS CORAIS **Obra** 1 - RESIDENCIAL BARRA DOS CORAIS
Fornecedor 50 - CIMPOR - CIA DE CIMENTOS DO BRASIL

Código	10752	Valor Principal	9.750,00
Tipo	1	Ipi	0,00
Data	06/12/2010	Outros Acresc.	0,00
Data de Entrega	15/12/2010	Frete	0,00
Hora de Entrega	07:30	Desconto	0,00
Status	G - Geral	Total OC	9.750,00
Frete	Pago	Quantidade de Itens	1
Descarga	Fornecedor	Total Recebido	9.750,00
Comprador	NEXT	SALDO VALOR	
Cond. Pagamento	0054 - ENTRADA/30 DIAS	Liberada	<input type="checkbox"/>
Finalidade	Aquisição de cimento para a Infra-Estrutura	Impressa	<input type="checkbox"/>

Encerra

Figura 12 - Interface de Consulta Contratos
Fonte: BRASSOFT (2015)

2.7.2 Controle de Obra

O Controle de Obra é um sistema focado na gestão de construtoras com as seguintes funcionalidades: Cadastro de obras; Cadastro de unidades; Cadastro de

imobiliárias; Cadastro de fornecedores; Cadastro de fotos, vídeos e documentos; Gerenciamento de custos por insumo; Lançamentos por categorias; Conciliação bancária; Conciliação de cartão de crédito; Relatório de posição financeira; Análise do resultado da obra; Análise de gastos; Relatório de lançamentos diários; Gráfico de gastos; Gráfico de insumos; Gráfico de investimentos; Gráfico de Receita x Despesas; Página de exibição da obra.

Na figura 13 mostra a lista de insumos com a ação de editar insumo, requisito abstraído para o sistema Web Framing, otimizando suas ações para melhorar a interação com o usuário, atendendo os requisitos do cliente.



Figura 13 - Interface de Insumos
Fonte: Controle de Obra (2015)

Na figura 14 mostra a interface de Conciliação Bancária, onde esta interface não atende nenhum requisito solicitado pelo o cliente, na figura 15 ilustra a interface de Dados de Obras, a listagem de obras é um requisito do sistema Web Framing solicitado pelo cliente, este requisito foi abstraído e melhorado suas funcionalidade para melhor solucionar o problema.

Controle de Obra

Bom dia Valdir (sair)

minha conta usuários

obra ativa no momento
Obra Exemplo

Home Cadastros Financeiro Relatórios Gráficos

Financeiro > Conciliação > Contas Bancárias > Banco do Brasil S.A

Exibindo: Não conciliados

Data	Descrição	Entrada	Saída
<input checked="" type="checkbox"/> 19/07/2010	Investimento Sócio 2	150.000,00	
<input checked="" type="checkbox"/> 31/08/2010	Mercadorama- Cestas básica pipedreiros		150,00
<input checked="" type="checkbox"/> 02/09/2010	Transferência: Banco do Brasil S.A - Dinheiro		10.000,00
<input checked="" type="checkbox"/> 13/10/2010	Compra - Tijolo		16,00
<input type="checkbox"/> 13/10/2010	Compra - Madeira		80,00
<input type="checkbox"/> 13/10/2010	Compra - Tubos e conexões		6,70
<input type="checkbox"/> 13/10/2010	Compra - Pregos		30,00

Selecionar todos

TOTAL SELECIONADO: R\$ 139.834,00

Saldo do extrato: 139.834,00 (valor negativo)

Diferença: R\$ 0,00

Conciliar

Copyright 2010-2011 - Controle de Obra - Todos os direitos reservados

Figura 14 - Interface de Conciliação Bancária
Fonte: Controle de Obra (2015)

Controle de Obra

Bom dia Valdir (sair)

minha conta usuários

obra ativa no momento
Obra Exemplo

Home Cadastros Financeiro Relatórios Gráficos

Obras > Obra Exemplo

Obra Exemplo

Início: 01/01/2011

Previsão de término: 31/07/2011

Execução de 04 sobrados triplex com área de 111,43 m² cada um, totalizando 445,72m²

Unidades disponíveis:

Unidade	Descrição	Área	Valor	Ações
Unidade 01 (frente)	- 111 m² (R\$ 239.900,00)			[nova] [lupa] [x]
Unidade 02 (meio)	- 111 m² (R\$ 229.900,00)			[lupa] [x]
Unidade 03 (meio)	- 111 m² (R\$ 229.900,00)			[lupa] [x]
Unidade 04 (fundos)	- 111 m² (R\$ 239.900,00)			[lupa] [x]

Para gerar uma página de apresentação deste imóvel selecione uma das opções abaixo:

Imobiliária Fischer [gerar]

IMOBILIÁRIA TESTE [gerar]

fotos videos documentos

Figura 15 - Interface de Dados de Obras
Fonte: Controle de Obra (2015)

3 DESENVOLVIMENTO

3.1 GERENCIAMENTO DO PROJETO

Para o planejamento do sistema *web* aplicou-se a metodologia de gestão de projetos SCRUM, inicialmente o projeto dividiu-se em cinco *Sprints*, desde o gerenciamento do início do desenvolvimento até a fase de testes com o usuário. As ordens e prioridades dos *Sprints* mantiveram a sequência mostrada no Quadro 5:

<i>Sprint 01</i>	No primeiro <i>Sprint</i> definiu-se os requisitos do sistema, a criação de tabelas no banco de dados, cadastro de pessoas e a interface de cadastro de insumos;
<i>Sprint 02</i>	No segundo <i>Sprint</i> criou-se a interface de cadastro de usuário, interface de cadastro de composição e a interface de cadastro de obra;
<i>Sprint 03</i>	No terceiro <i>Sprint</i> foi desenvolvida a interface de cadastro de orçamento, a interface da curva ABC e geração de relatórios;
<i>Sprint 04</i>	No quarto <i>Sprint</i> foram aplicados ajustes e melhorias no sistema executando teste do fluxo completo;
<i>Sprint 05</i>	E no quinto <i>Sprint</i> focou-se nas melhorias no ambiente de teste para disponibilizar ao usuário;

Quadro 2 - *Sprints*
Fonte: Autoria Própria.

A equipe se reunia no final do dia para debater o fluxo do projeto como dificuldades e melhorias encontradas. Para acompanhamento do projeto com o cliente a equipe realizou reuniões ao executar todas as tarefas propostas em cada *Sprint*. Ao final do quinto *Sprint* notou a necessidade de acrescentar *Sprints* para a manutenção e suporte do projeto, os quais estão descritos no Quadro 3:

<i>Sprint 06</i>	Neste <i>Sprint</i> definiu-se os ajustes solicitado pelo cliente;
<i>Sprint 07</i>	Neste <i>Sprint</i> definiu-se os ajustes solicitado pelo cliente;
<i>Sprint 08</i>	Neste <i>Sprint</i> definiu-se os ajustes solicitado pelo cliente;
<i>Sprint 09</i>	Neste <i>Sprint</i> definiu-se os ajustes solicitado pelo cliente;
<i>Sprint 10</i>	Neste <i>Sprint</i> definiu-se os ajustes solicitado pelo cliente;

Quadro 3 - *Sprints* adicionais
Fonte: Autoria Própria.

No Quadro 4 mostra as tarefas mais relevantes de cada *Sprint* para o projeto.

<i>Sprint 01</i>	Criação da base de dados baseado no levantamento de requisitos;
<i>Sprint 02</i>	Criação da interface de cadastro de obra;
<i>Sprint 03</i>	Criação da interface de cadastro de orçamento;
<i>Sprint 04</i>	Teste do fluxo completo do sistema;
<i>Sprint 05</i>	Adicionar Submenus nos menus Cadastrar e Gerenciar, Submenu Patrocinador e Notícia;
<i>Sprint 06</i>	Gerenciar insumo, Tabela Quantidade por Embalagem;
<i>Sprint 07</i>	Tela Inicial do Painel Web Framing
<i>Sprint 08</i>	Globalizar Composição de construtora pelo usuário Nakamura;
<i>Sprint 09</i>	Não pode inserir dois insumos ou mais na mesma composição, e sim só alterar a quantidade do campo;
<i>Sprint 10</i>	Desabilitar campo valor para não ser alterado, pois seu valor é dado conforme a quantidade, para não editar o valor do campo;

Quadro 4 - Tarefas mais relevantes de cada *Sprint*

Fonte: Autoria Própria.

3.2 DIAGRAMAS

Neste capítulo será apresentado o caso de uso geral do sistema WebFraming, bem como a descrição dos principais casos de uso. Também será apresentado o modelo relacional do sistema com uma breve descrição das principais tabelas.

3.2.1 Caso de uso

A Figura 16 mostra o diagrama de casos de uso geral do sistema WebFraming. Como é possível observar foram necessários quinze casos de uso a fim de contemplar todos os requisitos do produto. O usuário interage com o sistema primeiramente efetuando login, após entrar no sistema o usuário poderá cadastrar: insumos, composições, obras e orçamentos, sempre sendo redirecionado para respectiva interface de gerenciamento. Caso o usuário tenha permissão de administrador, ele também poderá cadastrar: usuário, cliente e empresa tendo acesso ao devido gerenciamento após o cadastro.

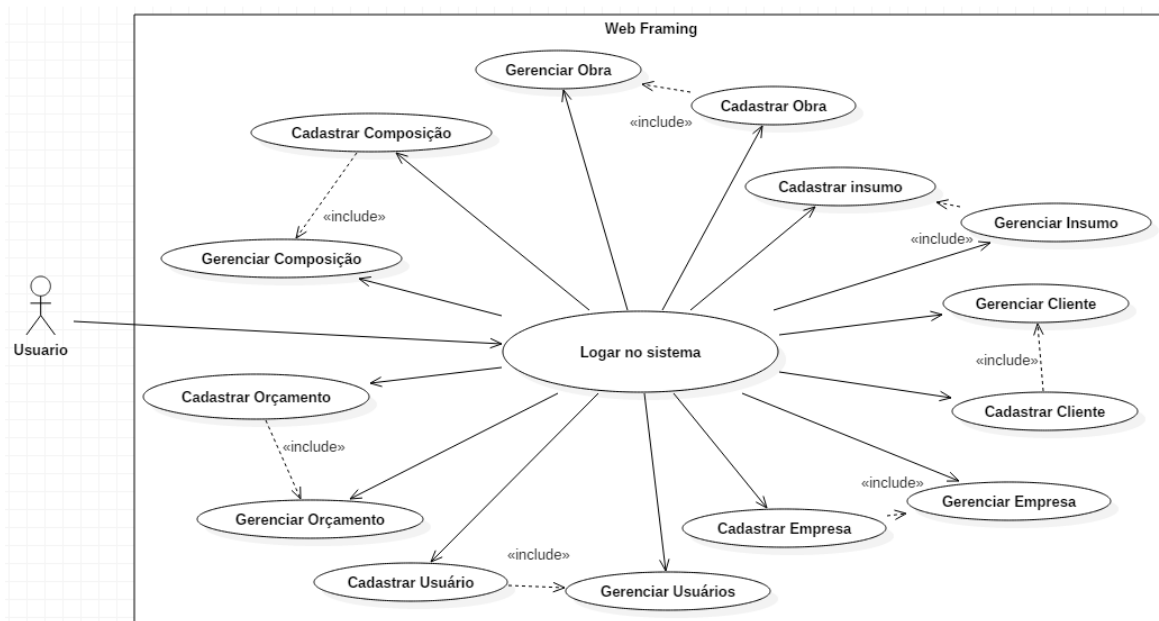


Figura 16 - Diagrama de Caso de Uso
Fonte: Autoria Própria.

O Quadro 5 descreve o caso de uso Logar no Sistema, o Quadro 6 o caso de uso Cadastrar Insumos, o Quadro 7 o caso de uso Cadastrar Composição, o Quadro 8 o caso de uso Cadastrar Obra e o Quadro 9 o caso de uso Cadastrar Orçamento.

As descrições dos casos de uso Gerenciar Insumo, Gerenciar Composição, Gerenciar Obra, Gerenciar Orçamento, Cadastrar Usuário, Gerenciar Usuário, Cadastrar Cliente, Gerenciar Cliente, Cadastrar Empresa e Gerenciar Empresa estão no Apêndice A.

Use case:	Logar no Sistema	
Descrição:	O usuário informa seu login e senha para acessar o sistema	
Pré-condição	Dados de acesso	
Cursos de Eventos Básicos		
Ação do Ator	Resposta do sistema	
1. Este caso de uso inicia quando o usuário acessa o sistema.	2. O sistema deixa o usuário informar seu login e sua senha.	
3. O usuário informa seu login e senha e confirma.	4. O sistema valida os dados do usuário e as permissões dele.	
5. O usuário pode acessar como administrador ou assinante.		
Curso Alternativo		
Caso as informações do usuário não estejam correta tem a opção de lembrar senha.		
Pós-Condição		
O usuário acessar o sistema e entra no painel de controle tendo acesso conforme suas permissões.		
Permissões de acesso		
Use Cases	Assinante	Administrador
Cadastrar Insumos		
Cadastrar Composição		
Cadastrar Obra		
Cadastrar Orçamento		
Gerenciar Insumos		
Gerenciar Composição		
Gerenciar Obra		
Gerenciar Orçamento		
Cadastrar Empresa		
Gerenciar Empresas		
Cadastrar Usuário		
Gerenciar Usuários		
Cadastrar Cliente		
Gerenciar Clientes		

Quadro 5 - Descrição caso de uso Logar no Sistema
Fonte: Autoria Própria.

Use case:	Cadastrar Insumos
Descrição:	O usuário informa dados de necessários para o cadastro de insumos.
Pré-condição	Estar logado.
Cursos de Eventos Básicos	
Ação do Ator	Resposta do sistema
1. Este caso de uso inicia quando o usuário acessa a interface de cadastro de insumos.	2. O sistema permite o usuário informar os dados de insumos.
3. O usuário informa os dados para o cadastro de insumos.	4. O sistema verifica os campos obrigatórios.
	5. O sistema retorna uma mensagem para o usuário informando que inseriu com sucesso, ou informa o usuário que existem dados inconsistentes.
Curso Alternativo	
Cancelar operação	
Pós-Condição	
É redirecionada a interface de Gerenciamento de Insumos.	

Quadro 6 - Descrição caso de uso Cadastrar Insumo
Fonte: Autoria Própria.

Use case:	Cadastrar Composição
Descrição:	O usuário informa dados de necessários para o cadastro de composição.
Pré-condição	Estar logado.
Cursos de Eventos Básicos	
Ação do Ator	Resposta do sistema
1. Este caso de uso inicia quando o usuário acessa a interface de cadastro de composição.	2. O sistema permite o usuário informar os dados de composição. 3. O sistema permite o usuário consultar insumos.
4. O usuário informar os dados para o cadastro de composição e relaciona os insumos necessários para a composição.	5. O sistema verifica os campos obrigatórios.
	6. O sistema retorna uma mensagem para o usuário informando que inseriu com sucesso, ou informa o usuário que existem dados inconsistentes.
Curso Alternativo	
Cancelar operação	
Pós-Condição	
É redirecionada a interface de Gerenciamento de Composições.	

Quadro 7 - Descrição caso de uso Cadastrar Composição
Fonte: Autoria Própria.

Use case:	Cadastrar Obra
Descrição:	O usuário informa dados de necessários para o cadastro de Obra.
Pré-condição	Estar logado.
Cursos de Eventos Básicos	
Ação do Ator	Resposta do sistema
1. Este caso de uso inicia quando o usuário acessa a interface de cadastro de obra.	2. O sistema permite o usuário informar os dados da obra. 3. O sistema permite o usuário consultar composição.
4. O usuário informar os dados para o cadastro da obra e relaciona as especificações necessárias para a obra.	5. O sistema verifica os campos obrigatórios.
	6. O sistema retorna uma mensagem para o usuário informando que inseriu com sucesso, ou informa o usuário que existem dados inconsistentes.
Curso Alternativo	
Cancelar operação	
Pós-Condição	
É redirecionada a interface de Gerenciamento de Obras.	

Quadro 8 - Descrição caso de uso Cadastrar Obra
Fonte: Autoria Própria.

Use case:	Cadastrar Orçamento
Descrição:	O usuário informa dados de necessários para o cadastro de orçamento.
Pré-condição	Estar logado.
Cursos de Eventos Básicos	
Ação do Ator	Resposta do sistema
1. Este caso de uso inicia quando o usuário acessa a interface de cadastro de orçamento.	2. O sistema permite o usuário informar os dados de orçamento. 3. O sistema permite o usuário consultar uma obra.
4. O usuário informar a obra para o orçamento.	5. O sistema verifica os campos obrigatórios.
	6. O sistema retorna uma mensagem para o usuário informando que inseriu com sucesso, ou informa o usuário que existem dados inconsistentes.
Curso Alternativo	
Cancelar operação	
Pós-Condição	
É redirecionada a interface de Gerenciamento de Orçamento.	

Quadro 9 - Descrição caso de uso Cadastrar Orçamento
Fonte: Autoria Própria.

3.2.2 Modelo Relacional

Para o armazenamento de dados do sistema Web Framing foi criada uma estrutura com dezenove tabelas para atender os requisitos funcionais do sistema.

Na Figura 17 estão modeladas as principais tabelas do sistema, o modelo relacional completo do sistema encontra-se no Apêndice B.

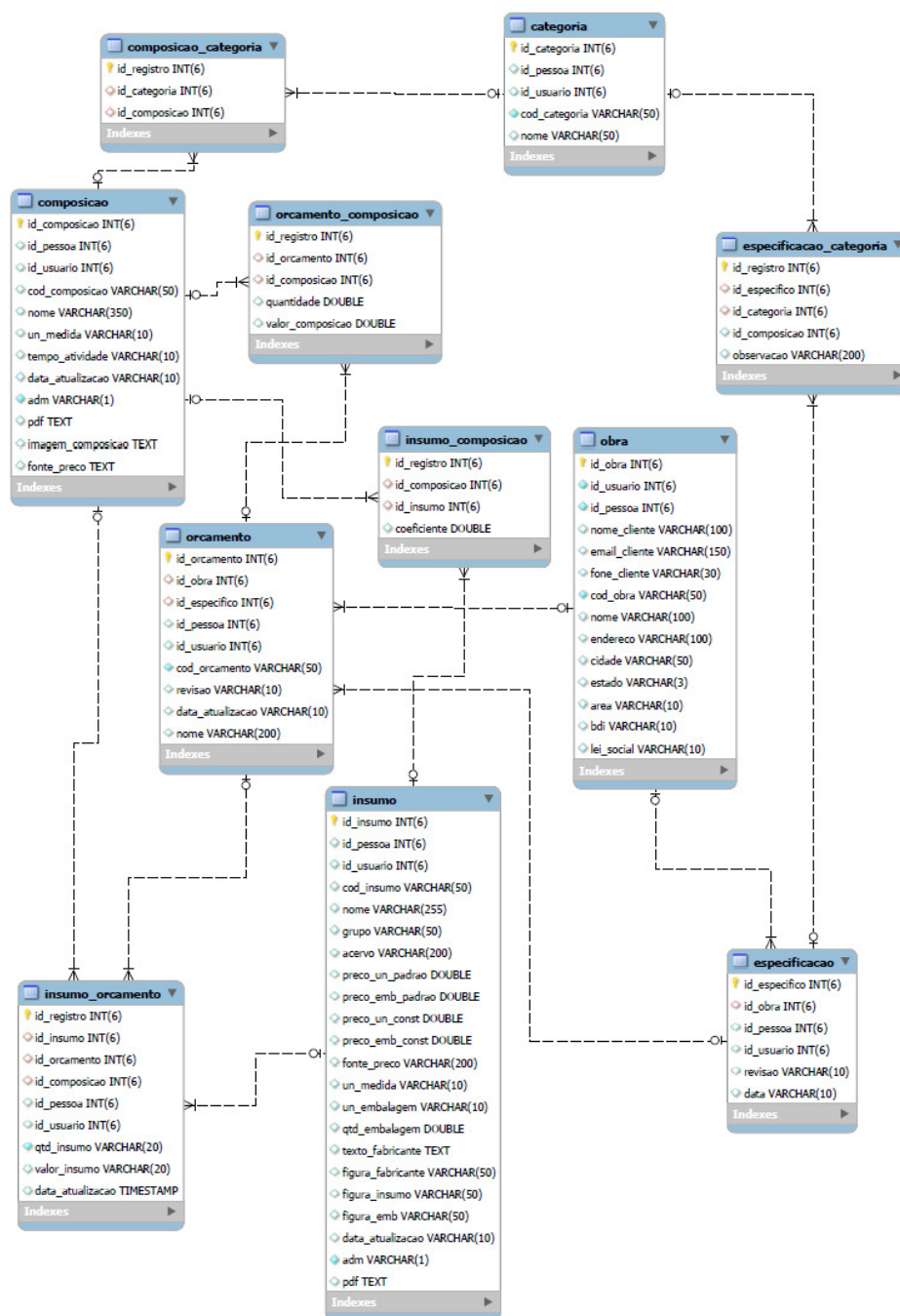


Figura 17 - Modelo Relacional Web Framing
Fonte: Autoria Própria.

No Quadro 10 estão descritas as principais tabelas do sistema.

Tabela	Descrição
Insumo	São os dados de todos os insumos cadastrados no sistema. Informações como valor e unidade de medida do insumo são salvos nessa tabela.
Composição	São as composições das obras, composta por vários insumos.
Categoria	Possuí as categorias referentes as composições e as especificações da obra.
Obra	São as obras, composta por várias composições e especificações.
Orçamento	São os orçamentos com base nas obras cadastradas no sistema.

Quadro 10 - Principais tabelas do sistema WebFraming

Fonte: Autoria Própria.

3.3 IMPLEMENTAÇÃO DO SOFTWARE

Tendo em vista as facilidades de desenvolvimento conquistadas com a utilização de um *framework*, a aplicação foi implementada utilizando o *framework* *Web CodeIgniter*, aplicando a linguagem de programação PHP, utilizando o banco de dados relacional MySQL.

O CodeIgniter tem sua estrutura baseada no padrão de desenvolvimento de *software* MVC, visando separar a lógica de programação das interfaces. A estrutura do sistema esta ilustrada na Figura 18, dentro da pasta *application* encontra-se as subpastas:

Controllers, nessa pasta contem a regra de negócio do sistema manipulando dados da *View* e *Model*. É responsável por capturar a requisição do usuário e por meio do endereço acessado mostra a respectiva *view*.

Models, essa pasta contem as funções de persistência com o banco de dados. Por meio dos respectivos arquivos são persistidos os dados de usuário, empresa, pessoa, insumo, composição, obra e orçamento.

Views, essa pasta contém o código da interface, fazendo iteração com o *controller* para mostrar os dados. Por meio das respectivas interfaces o usuário poderá interagir com o sistema a fim de gerenciar os dados de usuário, empresa, pessoa, insumo, composição, obra e orçamento.

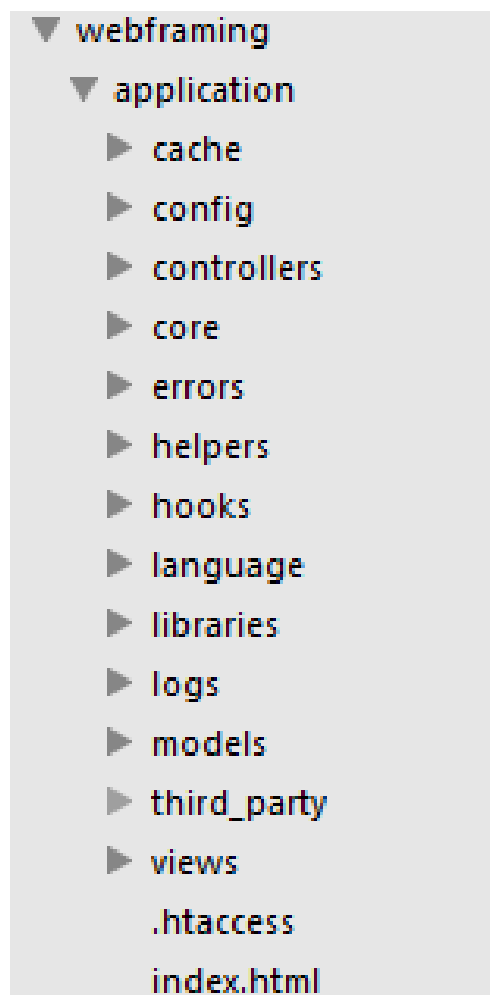


Figura 18 - Estrutura de pastas do sistema WebFraming
Fonte: Autoria própria

4 RESULTADOS

Neste capítulo são apresentados os resultados obtidos com o desenvolvimento do projeto.

A Figura 19 mostra a interface de *login* do sistema Web Framing, com os campos usuários e senha, onde os mesmos verificam os dados informados garantindo a segurança no acesso, caso o usuário informe a senha errado o sistema retorna uma mensagem ao usuário informando que a sua senha está errado, o usuário também tem a opção de lembrar senha, clicando no link “esqueci minha senha”.



Figura 19 - Interface de Login Web Framing
Fonte: Autoria própria

A Figura 20 mostra a interface principal do sistema Web Framing com os menus para acessar Perfil, Cadastros e Gerência. No menu Perfil contem os submenus de Visualizar Perfil e Alterar Senha. No menu Cadastros contem os submenus de Cadastrar Insumos, Cadastrar Composição, Cadastrar Obra e Cadastrar Orçamento. No menu Gerência contem os submenus Gerenciar Insumo, Gerenciar Composição, Gerenciar Obra e Gerenciar Orçamento. Sobre o menu contem os botões de alterar senha e sair do sistema.



Figura 20 - Interface Principal do sistema Web Framing
Fonte: Autoria própria

A Figura 21 mostra a interface de Cadastro de Insumo. No cabeçalho contem os menus, exibe as informações pessoais, informações sobre insumo, informações sobre preço e quantidade e informações sobre o fabricante, todos os campos são obrigatório o preenchimento, caso o usuário não preencher algum campo o sistema retorna uma mensagem informando que o campo precisa ser preenchido e destaca o campo em vermelho.

Web Framing

Boa noite Sr(a). **IVAN USER**
Sábado, 17 de Outubro de 2015

[Alterar senha](#) [Sair](#)

PERFIL ▾ CADASTROS ▾ GERÊNCIA ▾

[INÍCIO](#) / [INSUMO](#) / [CADASTRAR INSUMO](#)

Cadastrar Insumo

Informações Pessoais

Proprietário de sua conta
- Ivan De Jesus Ribeiro

Responsável pelo cadastro da obra
- Ivan User

Informações sobre insumo

Organização personalizada - digite o código do insumo

Nome do insumo

Grupo de insumos

Acervo

Informações sobre preços e quantidades

Clique para [Calcular!](#)

Preço unitário Personalizado
R\$

Quantidade em cada embalagem

Preço da embalagem Personalizado
R\$

Unidade de medida da embalagem

Unidade de medida do insumo

Link da fonte atualizada dos preços

Informações sobre fabricantes e imagens

Descrição sobre o fabricante

Faça upload da logo do fabricante

Faça upload da figura do insumo

Faça upload da figura da embalagem

Faça upload do Catálogo

[Cancelar](#) [Salvar Dados](#)

Up Rise 2015

Figura 21 - Interface de Cadastro de Insumo
Fonte: Autoria própria

A Figura 22 mostra a interface de Gerenciamento de insumo. Esta interface mostra uma lista de insumos cadastrados no sistema podendo Visualizar, Editar, Visualizar Catálogo e Excluir o insumo mediante confirmação.























Boa noite Sr(a). IVAN USER
Sábado, 17 de Outubro de 2015

Alterar senha Sair

PERFIL CADASTROS GERÊNCIA

INÍCIO / INSUMO / GERENCIAR INSUMO

Digite aqui para fazer sua pesquisa!

Imagem	Código	Nome do Insumo	Preço Unitário	Preço Embalagem	Quantidade por Embalagem	Ações
	SF01.01.42	Parafuso cabeça chata dentada ponta broca com asas - bitola: 4,2 mm / comprimento: 32 mm	R\$ 0/und	R\$ 0/und	1000 und/und	  
	SF01.01.43	Parafuso cabeça chata dentada ponta broca para OSB - bitola: 4,2 mm / comprimento: 32 mm	R\$ 0/un	R\$ 0/caixa	500 un/caixa	  
	SC01.01.08	Concreto - fck = 20 Mpa	R\$ 0/m³	R\$ 0/m³	1 m³/m³	  
	SC01.01.15	Pedra brita - número 2	R\$ 0/m³	R\$ 0/m³	1 m³/m³	  
	SC01.01.54	Lona Preta Comercial - dimensão: 1,0 x 10,0 m	R\$ 0/m	R\$ 0/rolo	6 m/rolo	  
	SF01.01.001	Perfil LSF, ZAR 230 - tipo: ME 90 / espessura: #1,25 mm	R\$ 0/kg	R\$ 0/kg	1 kg/kg	  

Mostrando um total de 436 registros


Up Rise 2015 

Figura 22 - Interface de Gerenciamento de Insumos
Fonte: Autoria própria

A figura 23 mostra a interface de Cadastrar Composição. No cabeçalho contem os menus, exibem as informações pessoais, Informações sobre Composição, Informações sobre insumos que fazem parte da composição, todos os campos são obrigatório o preenchimento, caso o usuário não preencher algum campo o sistema gera uma mensagem informando que o campo precisa ser preenchido e destaca o campo em vermelho.

The screenshot shows the 'Web Framing' application interface. At the top left is the logo, and at the top right, the user is logged in as 'IVAN USER' on 'Sábado, 17 de Outubro de 2015'. Navigation tabs include 'PERFIL', 'CADASTROS', and 'GERÊNCIA'. The breadcrumb trail is 'INÍCIO / COMPOSICAO / CADASTRAR COMPOSICAO'.

The main content area is titled 'Cadastrar Composição' and is divided into three sections:

- Informações Pessoais:** Shows the user as the owner ('Ivan De Jesus Ribeiro') and responsible for the work ('Ivan User'). A globe icon is present.
- Informações sobre Composição:** Contains several input fields:
 - 'Categoria da Composição' (dropdown menu)
 - 'Nome da Composição' (text input)
 - 'Unidade de Medida' (text input)
 - 'Link da fonte atualizada dos preços' (text input)
 - 'Organização personalizada - digite o código da composição' (text input)
 - 'Inserir Imagem' (file upload button: 'Choose File' / 'No file chosen')
 - 'Faça upload do Procedimento' (file upload button: 'Choose File' / 'No file chosen')
- Informações sobre Insumos:** Includes buttons for 'INSERIR INSUMOS' and 'BUSCAR EM OUTRA COMPOSIÇÃO'.

At the bottom of the form is a table header with columns: 'Imagem', 'Código', 'Nome do Insumo', 'Valor Unitário', 'Coeficiente', 'Remover', and 'Editar'. Below the table are 'Cancelar' and 'Salvar Dados' buttons.

The footer contains 'Up Rise 2015' and the 'Up Rise' logo.

Figura 23 - Interface de Cadastrar Composição
Fonte: Autoria própria

A Figura 24 mostra a interface de Gerenciar Composições. Esta interface mostra uma lista de composição cadastradas no sistema podendo Visualizar, Editar, Visualizar Procedimento e Excluir a composição mediante confirmação



Boa noite Sr(a). **IVAN USER**
Sábado, 17 de Outubro de 2015

[Alterar senha](#) [Sair](#)

PERFIL ▾ CADASTROS ▾ GERÊNCIA ▾

INÍCIO / COMPOSICAO / GERENCIAR COMPOSICAO

Digite aqui para fazer sua pesquisa!

Imagem	Código	Nome da Composição	Unidade de Medida	Ações
	Up 01	Perfil LSF, ZAR 230 - tipo: ME 90 / espessura: #1,25 mm	Kg	 
	Up 22	Perfil LSF, ZAR 230 - tipo: ME 250 / espessura: #1,64 mm	kg	 
	Up 86	Perfis estruturais - tipo: guia, montante, fita de contraventamento e chapa L/gusset	kg	 
	UP 27	Isotelha PUR para cobertura - dimensão: 1,0 x 8,0 m # 30 mm	m ²	 
	Up 88	Cumeeira Trapezoidal para telha PUR Comp. 1,2 m	m	 
	Up 89	Rufo lateral superior para isotelha - comprimento: 2,8 m	m	 
	Up 90	Telha trapezoidal 40 (dimensão: 0,99 x 12 m #0,65 mm) com OSB (dimensão: 1,20 x 2,40 m #11,1 mm)	m ²	 
	Up 91	Tratamento de juntas para placa cimentícia com Profort BC System	m ²	 

Mostrando um total de 42 registros

Up Rise 2015 

Figura 24 - Interface de Gerenciar Composições
Fonte: Autoria própria

A Figura 25 mostra a interface de Cadastrar Obra. No cabeçalho contem os menus, exibem as informações pessoais, Informação sobre seu Cliente, e Especificações onde, todos os campos são obrigatórios o preenchimento, caso o usuário não preencher algum campo o sistema gera uma mensagem informando que o campo precisa ser preenchido e destaca o campo em vermelho.

The screenshot displays the 'Web Framing' web application interface for registering a new work. The page is titled 'CADASTRAR NOVA OBRA' and is divided into several sections:

- Header:** Includes the 'Web Framing' logo, user information ('Boa noite Sr(a). IVAN USER', 'Sábado, 17 de Outubro de 2015'), and navigation links ('Alterar senha', 'Sair', 'PERFIL', 'CADASTROS', 'GERÊNCIA').
- Breadcrumbs:** 'INÍCIO / OBRA / CADASTRAR OBRA'.
- Informações Pessoais:** A box containing user details: 'Proprietário de sua conta - Ivan De Jesus Ribeiro' and 'Responsável pelo cadastro da obra - Ivan User', accompanied by a profile picture icon.
- Informações sobre seu cliente:** A section with input fields for 'Nome Cliente (proprietário da obra)', 'Email Cliente (proprietário da obra)', and 'Telefone Cliente (proprietário da obra)'.
- Informações sobre a obra:** A section with input fields for 'Organização personalizada - digite o código da obra', 'Nome da obra', 'Endereço da Obra', 'Cidade', 'Estado (2 dígitos - Ex.: PR)', 'Área da Obra', 'BDI Obra', and 'Lei Social Obra'.
- Especificações:** A section with a purple button labeled 'INSERIR E ESPECIFICAÇÃO'.
- Buttons:** A red 'Cancelar' button and a green 'Salvar Dados' button.

The footer of the page includes the text 'Up Rise 2015' and the 'Up Rise' logo.

Figura 25 - Interface de Cadastrar Obra
Fonte: Autoria própria

A Figura 26 mostra a interface de Gerenciamento de Obra. Esta interface mostra uma lista de obras cadastrados no sistema podendo Visualizar, Editar, Copiar Obra e Excluir a obra mediante confirmação.

Boa noite Sr(a). IVAN USER
Sábado, 17 de Outubro de 2015

[Alterar senha](#) [Sair](#)

PERFIL ▾ CADASTROS ▾ GERÊNCIA ▾

INÍCIO / OBRA / GERENCIAR OBRA

Digite aqui para fazer sua pesquisa!


Código	Nome do Cliente	Nome da Obra	Cidade	Estado	Ações
Up Rise 17.10.2015	Pedro Paulo	Barracão	PONTA GROSSA	PR	

Mostrando um total de 1 registros

Up Rise 2015

Figura 26 - Interface de Gerenciar Obras
Fonte: Autoria própria

A Figura 27 mostra a interface de Cadastrar Orçamento. No cabeçalho contem os menus, exibem as informações pessoais, informação sobre o orçamento, informação sobre obra, informação sobre composição, todos os campos são obrigatórios o preenchimento, caso o usuário não preencher algum campo o sistema gera uma mensagem informando que o campo precisa ser preenchido e destaca o campo em vermelho.



Web Framing

Boa noite Sr(a). **IVAN USER**
Sábado, 17 de Outubro de 2015

[Alterar senha](#) [Sair](#)

PERFIL ▾
CADASTROS ▾
GERÊNCIA ▾


[INÍCIO](#) / [ORÇAMENTO](#) / [CADASTRAR ORÇAMENTO](#)

Alterar uma Obra

Informações Pessoais

Proprietário de sua conta
- Ivan De Jesus Ribeiro

Responsável pelo cadastro da obra
- Ivan De Jesus Ribeiro



Informações sobre Orçamento

Organização personalizada - digite o código do orçamento
Up Rise 17.10.2015

Nome do Orçamento
Barracão

Informações sobre Obra

[INSERIR OBRA](#)

Código da Obra	Nome do Cliente	Nome da Obra	Área
Up Rise 17.10.2015	Pedro Paulo	Barracão	1000m²

Informações sobre Composição

[INSERIR COMPOSIÇÃO](#)

INFRAESTRUTURA [+](#)

Código Composição	Nome da Composição	Unidade de Medida	Quantidade	Valor	Remover
Up 51	Área de Forma para radier	vb	<input type="text" value="100"/>	R\$ <input type="text" value="1200"/>	X

ESTRUTURA [+](#)

Código Composição	Nome da Composição	Unidade de Medida	Quantidade	Valor	Remover
Up 65	Chapa de Gousset 150 x 150 #0,95, ZAR 230	un	<input type="text" value="100"/>	R\$ <input type="text" value="800"/>	X

GENÉRICO [+](#)

Código Composição	Nome da Composição	Unidade de Medida	Quantidade	Valor	Remover
Up 116	Verba para piso intertravado	m²	<input type="text" value="100"/>	R\$ <input type="text" value="100"/>	X

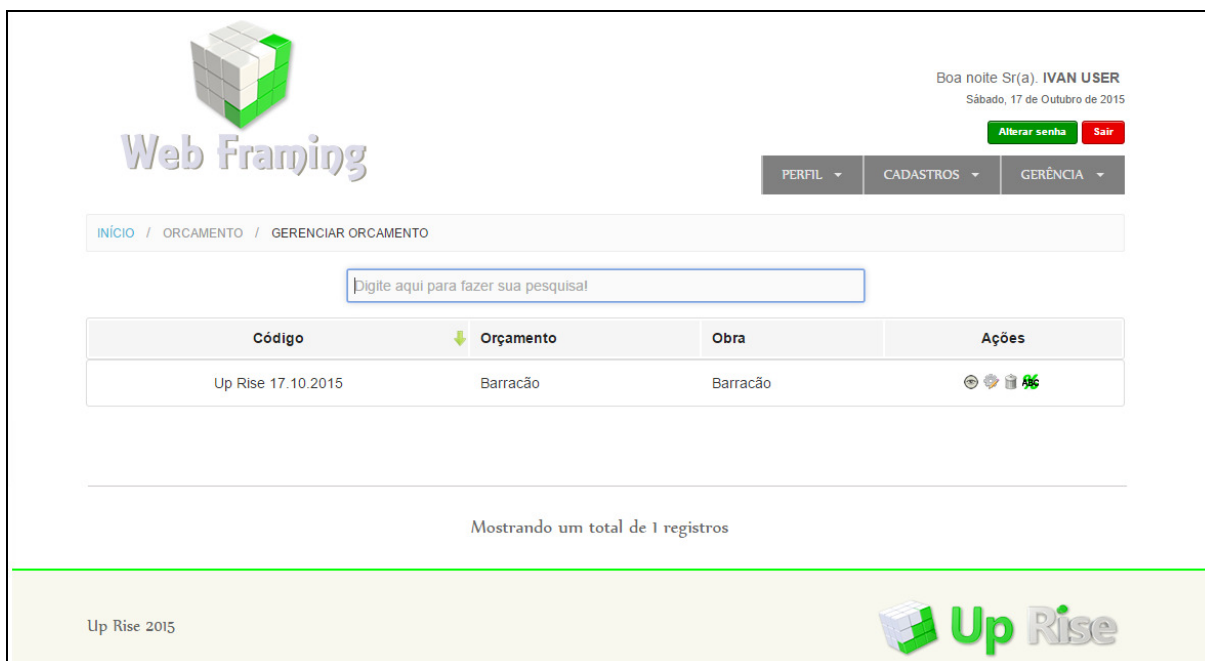
[Cancelar](#)
[Salvar Dados](#)

Up Rise 2015






Figura 27 - Interface de Cadastrar Orçamento
Fonte: Autoria própria

A Figura 28 mostra a interface de Gerenciar Orçamento. Esta interface mostra uma lista de orçamentos cadastrados no sistema podendo Visualizar, Editar, Gerar Curva ABC e Excluir o orçamento mediante confirmação.




The screenshot displays the 'Web Framing' interface. At the top left is the logo, and at the top right, the user is identified as 'IVAN USER' with the date 'Sábado, 17 de Outubro de 2015'. Navigation tabs for 'PERFIL', 'CADASTROS', and 'GERÊNCIA' are visible. The breadcrumb trail reads 'INÍCIO / ORÇAMENTO / GERENCIAR ORÇAMENTO'. A search bar contains the placeholder text 'Digite aqui para fazer sua pesquisa!'. Below this is a table with the following data:

Código	Orçamento	Obra	Ações
Up Rise 17.10.2015	Barracão	Barracão	  

Below the table, it states 'Mostrando um total de 1 registros'. The footer includes 'Up Rise 2015' and the 'Up Rise' logo.

Figura 28 - Interface de Gerenciar Orçamento
Fonte: Autoria própria

A Figura 29 mostra a interface de Gerar Curva ABC. Esta interface tem o objetivo de mostrar ao usuário aonde esta tendo mais gasto, qual insumo consome mais o orçamento, podendo alterar o preço do insumo para baratear mais o orçamento, e também editar o insumo para atualizar seu valor.



Boa noite Sr(a). **IVAN USER**
 Sábado, 17 de Outubro de 2015

[Alterar senha](#) [Sair](#)


PERFIL ▾
CADASTROS ▾
GERÊNCIA ▾

[INÍCIO](#) / [ORÇAMENTO](#) / [GERAR CURVA ABC](#)

Informações Pessoais

Proprietário de sua conta
 - Ivan De Jesus Ribeiro

Responsável pelo cadastro da obra
 - Ivan De Jesus Ribeiro











Informações sobre seu cliente

Nome do Cliente: Pedro Paulo
 Email do Cliente: ppa@hotmail.com
 Telefone do Cliente: (42)3239-9870

Informações sobre o Orçamento

Código Orçamento: Up Rise 17.10.2015
 Nome: Barracão

[Imprimir](#) [Planilha](#)

Fornecedor	Cód Insumo	Imagem	Nome do Insumo	Qtd Insumo	Qtd Insumo/Embalagem	Qtd Embalagem	Valor Embalagem	Valor Total ↓	Porcentagem	Alterar Valor/ Editar
	SF01.01.067		Área de forma para radier	100.00 m²	1.00 m²/m²	100 m²	R\$ 12.00	R\$ 1200.00	57.1429%	+ -
	SF01.01.078		Chapa de Gousset, ZAR 230 - dimensão: 150 x 150 mm #0,95 mm	110.00 kg	1.00 kg/kg	110 kg	R\$ 7.00	R\$ 770.00	36.6667%	+ -
	SF01.01.46 verificar		Verba para piso intertravado	100.00 vb	1.00 vb/vb	100 vb	R\$ 1.00	R\$ 100.00	4.7619%	+ -
	SF01.01.007		Parafuso cabeça flangeada ponta broca - bitola: 4,8 mm / comprimento: 19 mm	1000.00 un	500.00 un/caixa	2 caixa	R\$ 15.62	R\$ 31.24	1.4286%	+ -

Mostrando 1 a 4 de 4 registros

[OK! Voltar](#)

Up Rise 2015




Figura 29 - Interface de Gerar Curva ABC
 Fonte: Autoria própria

5 CONCLUSÃO

Através das práticas e métodos adotados pelo uso da metodologia SCRUM para a gestão do projeto e organização da equipe, utilização de *framework* e seus padrões para o desenvolvimento, foi possível desenvolver um sistema atendendo a necessidade do cliente, obtendo *feedback* do processo, documentação sobre o projeto, obtendo indicadores para um melhoramento contínuo.

Durante o desenvolvimento a alteração de requisito pelo cliente e a implementação em paralelo foram as principais dificuldades encontradas, o cliente constantemente solicitava novas funcionalidades ocasionando o aumento do número de *Sprints* proposto inicialmente, prorrogando a entrega do projeto, a participação direta do cliente no projeto foi importante para identificação dos requisitos e esclarecimento de dúvidas sobre o fluxo de dados, porém esse contato permite o cliente solicitar novas alterações não previstas anteriormente. Para atender os prazos iniciais de testes a equipe focou em desenvolver os requisitos levantados anteriormente e adicionando após os testes essas melhorias com novos prazos. A implementação em paralelo foi uma dificuldade pois não foi utilizado nenhuma ferramenta de controle de versão para gerenciamento dos arquivos tendo que estar em contato direto com a equipe.

5.1 TRABALHOS FUTUROS

Para trabalhos futuros limitar a inclusão de novos processos e funcionalidade desde o início do projeto para não alterar prazos de entrega e alocação da equipe para o desenvolvimento além do previsto, para a gestão dos arquivos de desenvolvimento a utilização de uma ferramenta de controle de versão para poder juntar alterações que envolvam mesmos arquivos.

REFERÊNCIAS

ALEXANDER, C et al. **A Pattern Language**. New York: Oxford University Press, 1977.

ANDRADE, F. F. de. **Desenvolvimento de aplicações Web com a utilização dos Frameworks CodeIgniter e Doctrine**. 2011. 54 f. TCC (Graduação) - Curso de Tecnologia em Desenvolvimento de Sistemas de Informação, Universidade Tecnológica Federal do Paraná, Medianeira, 2011.

BASSI FILHO, D. L. **Experiências com desenvolvimento ágil**. 170 pág. Dissertação (Mestrado) - Universidade de São Paulo, São Paulo, 2008.

BECK, K. **Extreme Programming Explained: Embrace Change**. 2ª Edição. Addison-Wesley, 2004. 256 p.

BECK, K. et al. **Manifesto para Desenvolvimento Ágil de Software**, 2001. Disponível em <<http://agilemanifesto.org>> Acessado em Agosto de 2015.

BRASSOFT. 2015. Disponível em <<http://brassoft.com.br>> Acessado em set. 2015.

BUSCHMANN, F. et al. **Pattern-Oriented Software Architecture: A System of Patterns**. John Wiley & Sons, 1996.

CHEN, P. P. **The Entity-Relationship Model – Toward a Unified View of Data**. Massachusetts Institute of Technology, 1976.

COAD, P. **Object-Oriented Patterns**. Communications of the ACM, v. 35, n. 9, p. 152-159, Setembro de 1992.

CODD, E. F. **A Relational Model of Data for Large Shared Data Banks**. 1970.

CONTROLE DE OBRA. 2015. Disponível em <<http://controledeobra.com.br>> Acessado em set. 2015.

DALL'OGGIO, P. **PHP: Programando com Orientação a Objetos**. São Paulo: Novatec, 2007. 576 p.

DAMAS, L. **SQL: structured query language**. 6 ed. Rio de Janeiro: LTC, 2007.

GABARDO, A. C. **CodeIgniter Framework PHP**. São Paulo: Novatec, 2010. 200 p.

GAMMA, E. et al. **Padrões de projeto: soluções reutilizáveis de software orientado a objetos**. Porto Alegre: Bookman, 2000. 364 p.

GARLAN, D.; SHAW, M. **An introduction to software architecture**. Pittsburgh: School Of Computer Science Carnegie Mellon University, 1994.

JANDL, P. J. **Mais Java**. São Paulo: Futura, 2003. 648 p.

JOHNSON, R. E.; FOOTE B. **Designing Reusable Classes**. Journal of Object Oriented Programming – JOOP, v. 1, n. 2, p. 22-35, Junho/Julho de 1988.

KOFLER, M. **The Definitive Guide to MySQL**. Nova Iorque, 2005.

MALDONADO, J. C. et al. **Padrões e Frameworks de Software**. 2001. Universidade de São Paulo, Instituto de Ciências Matemáticas e de Computação. Disponível em < <http://www.icmc.usp.br/~rtvb/apostila.pdf> > Acessado em Setembro de 2015.

MOREIRA, A. B. **Proposta de um processo ágil aplicado ao desenvolvimento de um software científico**. 2011. 127 f. Dissertação (Mestrado) - Programa de Pós-graduação em Modelagem Matemática e Computacional, Centro Federal de Educação Tecnológica de Minas Gerais, Belo Horizonte, 2011.

PEREIRA, P.; TORREÃO, P.; MARÇAL, A. S. **Entendendo Scrum para gerenciar projetos de forma ágil**. Mundo PM, v. 1, p. 3-11, 2007.

PERRY, D. E.; WOLF, A. L. **Foundations for the study of software architecture**.

SIGSOFT Software Engineering Notes, v. 17, n. 4, p. 40-52, Outubro de 1992.

PRESSMAN, R. S. **Engenharia de Software: Uma abordagem Profissional**. 7ª Edição. Porto Alegre: Mcgraw-Hill, 2011. 780 p.

RAMAKRISHNAN, R.; GEHRKE, J. **Sistemas de gerenciamento de banco de dados-3**. McGraw Hill Brasil, 2008.

RAMOS, E. A. **Metodologias Ágeis: Extreme Programming**. 2013. 42 f. Monografia (Especialização) - Curso de Desenvolvimento de Sistemas Para Web, Universidade Estadual de Maringá, Maringá, 2013.

SCHWABER, K. **Agile Project Management With Scrum**. Microsoft Press, 2004.

SUPPI, L. F. P. **Desenvolvimento de aplicações utilizando o Framework Zend**. 2012. 69 f. TCC (Graduação) - Curso de Sistemas de Informação, Universidade do Planalto Catarinense, Lages, 2012.

SYMFONY. **Learn Symfony**. 2015. Disponível em <<http://symfony.com>> Acessado em set. 2015.

VINCENZI, A. M. R. **Orientação a Objeto: Definição, Implementação e Análise de Recursos de Teste e Validação**. 2004. 231 f. Tese (Doutorado) - Curso de Ciências de Computação e Matemática Computacional, ICMC, USP, São Carlos, 2004.

ZEMEL, T. **O que é um framework: definição e benefícios de se usar frameworks**, 2009. Disponível em: <<http://codeigniterbrasil.com/passosiniciais/o-que-e-um-framework-definicao-e-beneficios-de-se-usar-frameworks/>>. Acessado em Setembro de 2015.

ZEND. **Programmer's Reference Guide of Zend Framework 2**. 2015. Disponível em <<http://framework.zend.com>> Acessado em 19 set. 2015.

APÊNDICE A - Descrição dos Casos de Uso do Sistema WebFraming

Use case:	Gerenciar Insumos
Descrição:	O usuário visualiza uma lista de insumos cadastrados podendo excluir um insumo, editar um insumo ou visualizar um insumo.
Pré-condição	Estar logado.
Cursos de Eventos Básicos	
Ação do Ator	Resposta do sistema
1. Este caso de uso inicia quando o usuário acessa a interface de gerenciar insumo.	2. O sistema mostra uma lista de insumos cadastrados.
3. O usuário o usuário pode escolher ação: Visualizar, Editar e Excluir um insumo.	4. O sistema verifica a ação do usuário e redireciona para uma das ações escolhidas
5. Usuário informa dado, visualiza dado ou exclui dados.	6. O sistema trata a ação do usuário e valida seus respectivos dados.
Curso Alternativo	
O sistema retorna uma mensagem ao usuário informando sobre o resultado de sua ação voltando a interface Gerenciar Insumos.	
Pós-Condição	
É redirecionada a interface de Gerenciamento de Insumos.	

Use case:	Gerenciar Composição
Descrição:	O usuário visualiza uma lista de composições cadastradas podendo visualizar uma composição, editar uma composição ou excluir.
Pré-condição	Estar logado.
Cursos de Eventos Básicos	
Ação do Ator	Resposta do sistema
1. Este caso de uso inicia quando o usuário acessa a interface de gerenciar composição.	2. O sistema mostra uma lista de insumos cadastrados.
3. O usuário o pode escolher uma ação: Visualizar, Editar e Excluir.	4. O sistema verifica a ação do usuário e redireciona para a escolhida.
5. Usuário informa dados, visualiza dados ou exclui dados.	6. O sistema trata a ação do usuário e valida seus respectivos dados.
Curso Alternativo	
O sistema retorna uma mensagem ao usuário informando sobre o resultado de sua ação voltando a interface Gerenciar Composição.	
Pós-Condição	
É redirecionada a interface de Gerenciamento de Composição.	

Use case:	Gerenciar Obra
Descrição:	O usuário visualiza uma lista de obras cadastradas podendo: visualizar, editar, copiar ou excluir uma obra.
Pré-condição	Estar logado.
Cursos de Eventos Básicos	
Ação do Ator	Resposta do sistema
1. Este caso de uso inicia quando o usuário acessa a interface de gerenciar obra.	2. O sistema mostra uma lista de obras cadastradas.
3. O usuário o pode escolher uma ação: Visualizar, Editar, Copiar e Excluir.	4. O sistema verifica a ação do usuário e redireciona para a escolhida.
5. Usuário informa dados, visualiza dados ou exclui dados.	6. O sistema trata a ação do usuário e valida seus respectivos dados.
Curso Alternativo	
O sistema retorna uma mensagem ao usuário informando sobre o resultado de sua ação voltando a interface Gerenciar Obras.	
Pós-Condição	
É redirecionada a interface de Gerenciamento de Obras.	

Use case:	Gerenciar Orçamento
Descrição:	O usuário visualiza uma lista de orçamentos podendo: Visualizar, Editar, Gerar Curva ABC e Excluir.
Pré-condição	Estar logado.
Cursos de Eventos Básicos	
Ação do Ator	Resposta do sistema
1. Este caso de uso inicia quando o usuário acessa a interface de gerenciar orçamento.	2. O sistema mostra uma lista de orçamentos cadastrados.
3. O usuário o pode escolher uma ação: Visualizar, Editar, Gerar Curva ABC e Excluir.	4. O sistema verifica a ação do usuário e redireciona para a ação escolhida.
5. Usuário informa dados, visualiza dados ou exclui dados.	6. O sistema trata a ação do usuário e valida seus respectivos dados.
Curso Alternativo	
O sistema retorna uma mensagem ao usuário informando sobre o resultado de sua ação voltando a interface Gerenciar Orçamento.	
Pós-Condição	
É redirecionada a interface de Gerenciamento de Composição.	

Use case:	Cadastrar Usuário
Descrição:	O usuário informa dados de necessários para o cadastro de usuário.
Pré-condição	Estar logado como administrador.
Cursos de Eventos Básicos	
Ação do Ator	Resposta do sistema
1. Este caso de uso inicia quando o usuário acessa a interface de cadastro de usuário.	2. O sistema permite o usuário informar os dados de usuário.
3. O sistema verifica os campos obrigatórios.	4. O sistema retorna uma mensagem para o usuário informando que inseriu com sucesso, ou informa o usuário que existem dados inconsistentes.
Curso Alternativo	
Cancelar operação.	
Pós-Condição	
É redirecionada a interface de Gerenciamento de Usuário.	

Use case:	Gerenciar Usuário
Descrição:	O usuário visualiza uma lista de usuários podendo: Editar, Alterar Senha e Excluir.
Pré-condição	Estar logado como administrador.
Cursos de Eventos Básicos	
Ação do Ator	Resposta do sistema
1. Este caso de uso inicia quando o usuário acessa a interface de gerenciar usuário.	2. O sistema mostra uma lista de usuários cadastrados.
3. O usuário o pode escolher uma ação: Editar, Alterar Senha e Excluir.	4. O sistema verifica a ação do usuário e redireciona para a ação escolhida.
5. Usuário informa dados ou exclui dados.	6. O sistema trata a ação do usuário e valida seus respectivos dados.
Curso Alternativo	
O sistema retorna uma mensagem ao usuário informando sobre o resultado de sua ação voltando a interface Gerenciar Usuário.	
Pós-Condição	
É redirecionada a interface de Gerenciamento de Usuário.	

Use case:	Cadastrar Cliente
Descrição:	O usuário informa dados de necessários para o cadastro de cliente.
Pré-condição	Estar logado como administrador.
Cursos de Eventos Básicos	
Ação do Ator	Resposta do sistema
1. Este caso de uso inicia quando o usuário acessa a interface de cadastro de cliente.	2. O sistema permite o usuário informar os dados de cliente.
3. O sistema verifica os campos obrigatórios.	4. O sistema retorna uma mensagem para o usuário informando que inseriu com sucesso, ou informa o usuário que existem dados inconsistentes.
Curso Alternativo	
Cancelar operação.	
Pós-Condição	
É redirecionada a interface de Gerenciamento de Cliente.	

Use case:	Gerenciar Cliente
Descrição:	O usuário visualiza uma lista de clientes podendo: Visualizar, Editar e Excluir.
Pré-condição	Estar logado como administrador.
Cursos de Eventos Básicos	
Ação do Ator	Resposta do sistema
1. Este caso de uso inicia quando o usuário acessa a interface de gerenciar cliente.	2. O sistema mostra uma lista de clientes cadastrados.
3. O usuário o pode escolher uma ação: Visualizar, Editar ou Excluir.	4. O sistema verifica a ação do usuário e redireciona para a ação escolhida.
5. Usuário informa dados, visualiza dados ou exclui dados.	6. O sistema trata a ação do usuário e valida seus respectivos dados.
Curso Alternativo	
O sistema retorna uma mensagem ao usuário informando sobre o resultado de sua ação voltando a interface Gerenciar Cliente.	
Pós-Condição	
É redirecionada a interface de Gerenciamento de Cliente.	

Use case:	Cadastrar Empresa
Descrição:	O usuário informa dados de necessários para o cadastro de empresa.
Pré-condição	Estar logado como administrador.
Cursos de Eventos Básicos	
Ação do Ator	Resposta do sistema
1. Este caso de uso inicia quando o usuário acessa a interface de cadastro de empresa.	2. O sistema permite o usuário informar os dados de empresa.
3. O sistema verifica os campos obrigatórios.	4. O sistema retorna uma mensagem para o usuário informando que inseriu com sucesso, ou informa o usuário que existem dados inconsistentes.
Curso Alternativo	
Cancelar operação.	
Pós-Condição	
É redirecionada a interface de Gerenciamento de Empresa.	

Use case:	Gerenciar Empresa
Descrição:	O usuário visualiza uma lista de empresas podendo: Visualizar, Editar e Excluir.
Pré-condição	Estar logado como administrador.
Cursos de Eventos Básicos	
Ação do Ator	Resposta do sistema
1. Este caso de uso inicia quando o usuário acessa a interface de gerenciar empresa.	2. O sistema mostra uma lista de empresas cadastradas.
3. O usuário o pode escolher uma ação: Visualizar, Editar ou Excluir.	4. O sistema verifica a ação do usuário e redireciona para a ação escolhida.
5. Usuário informa dados, visualiza dados ou exclui dados.	6. O sistema trata a ação do usuário e valida seus respectivos dados.
Curso Alternativo	
O sistema retorna uma mensagem ao usuário informando sobre o resultado de sua ação voltando a interface Gerenciar Empresa.	
Pós-Condição	
É redirecionada a interface de Gerenciamento de Empresa.	

APÊNDICE B - Modelo Relacional do Sistema WebFraming

