

UNIVERSIDADE FEDERAL DO PIAUÍ
CAMPUS SENADOR HELVÍDIO NUNES DE BARROS
CURSO BACHARELADO EM SISTEMAS DE INFORMAÇÃO

GIVANALDO RODRIGUES SILVA

**DESENVOLVIMENTO DE UM SISTEMA *WEB* PARA CONTROLE E GESTÃO DE
CONDOMÍNIOS DA PICOS-IMOBILIÁRIA UTILIZANDO PRINCÍPIOS DE
RESPONSIVIDADE**

PICOS - PI

2015

GIVANALDO RODRIGUES SILVA

DESENVOLVIMENTO DE UM SISTEMA *WEB* PARA CONTROLE E GESTÃO DE
CONDOMÍNIOS DA PICOS-IMOBILIÁRIA UTILIZANDO PRINCÍPIOS DE
RESPONSIVIDADE

Trabalho de Conclusão de Curso apresentado ao Curso de Bacharelado em Sistemas de Informação do Campus Senador Helvídio Nunes de Barros (CSHNB) da Universidade Federal do Piauí (UFPI) como parte dos requisitos para obtenção do Grau de Bacharel em Sistemas de Informação, sob orientação do Professor Francisco das Chagas Imperes Filho.

PICOS - PI

2015

FICHA CATALOGRÁFICA
Serviço de Processamento Técnico da Universidade Federal do Piauí
Biblioteca José Albano de Macêdo

S5861d Silva, Givanaldo Rodrigues.

Desenvolvimento de um sistema web para controle e gestão de condomínios da Picos-Imobiliária utilizando princípios de responsividade / Givanaldo Rodrigues Silva . – 2015.

CD-ROM : il.; 4 ¾ pol. (69 f.)

Monografia(Bacharelado em Sistemas de Informação) – Universidade Federal do Piauí, Picos, 2015.

Orientador(A): Profº. Esp. Francisco das Chagas Imperes Filho.

1. Sistemas de Informação-Ciclo de vida. 2. Sistemas de Informação-Usabilidade. 3. Web Design Responsivo 2. I. Título.

CDD 005

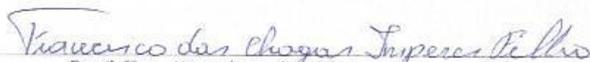
DESENVOLVIMENTO DE SISTEMA *WEB* PARA CONTROLE E GESTÃO DE
CONDOMÍNIOS DA PICOS-IMOBILIÁRIA UTILIZANDO PRINCÍPIOS DE
RESPONSIVIDADE

GIVANALDO RODRIGUES SILVA

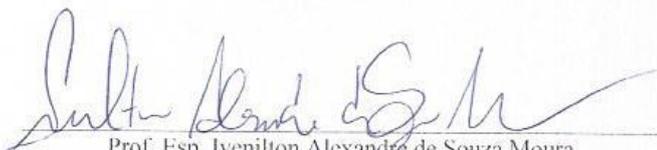
Monografia aprovada como exigência parcial para obtenção do
grau de Bacharel em Sistemas de Informação.

Data de Aprovação

Picos - Pl. 26 de junho de 2015



Prof. Esp. Francisco das Chagas Imperes Filho
Orientador



Prof. Esp. Ivenilton Alexandre de Souza Moura
Membro



Prof. Esp. Eder Jânio Francisco Gomes
Membro

Dedico este trabalho à Deus, pela sua força e proteção diária a mim concebida, e a minha família, em especial aos meus pais e irmão, Severino Almeida da Silva, Viliane Rodrigues Lima e Givanildo Rodrigues da Silva por sempre estarem comigo, apoiando, incentivando, criticando e acreditando no meu potencial.

AGRADECIMENTOS

Agradeço primeiramente à Deus, o qual acredito estar sempre comigo, e a minha família pelo incentivo, carinho e confiança. Em especial ao meu pai, Severino Almeida da Silva, minha mãe, Viliane Rodrigues Lima e meu irmão, Givanildo Rodrigues da Silva pelo apoio, ensinamentos e críticas, mas principalmente por nunca me abandonarem, sendo eles minha maior motivação.

Ao meu orientador, Francisco Imperes, e ao meu amigo e coorientador, Ivenilton Alexandre por sempre estarem dispostos a dividir o seu conhecimento comigo. E a todos os professores do curso de Sistemas de Informação.

Aos meus amigos da minha cidade natal (Otávio, Bruno, Pulinha, galera dos boias, companheiros de ensino médio). Agradeço também aos que tive a honra de dividir apartamento e compartilhar momentos únicos nessa temporada como universitário (Obedio Albuquerque, Leonan Bezerra, Bell Cristina, Jane Minerva e Josiel Santos) e aos que tive o prazer de dividir a sala de aula (Abimael, Laronso, Kaio, Gilberlon, Lívia, Viviane, Renan, Denes, Marcelino e outros).

Muito obrigado a todos!

“Somos todos geniais. Mas se você julgar um peixe pela sua capacidade de subir em árvores, ele passará a vida inteira acreditando ser estúpido.”

Albert Einstein.

RESUMO

Os sistemas de informações estão constantemente sendo adotados pelas empresas. Essa crescente é dada principalmente pelos benefícios agregados com a sua utilização o que inclui sua importância na tomada de decisões, estabelecimento de estratégias e vantagem competitiva. O presente trabalho demonstra o desenvolvimento de um sistema de informação gerencial para a administradora de condomínios Picos-Imobiliária. O objetivo deste projeto foi implementar um sistema *web* que automatize os processos de gerenciamento das atividades, resolvendo assim suas carências percebidas através da análise e coleta de dados realizadas na própria administradora. Com isso, é possível um maior controle dos condomínios e os interessados poderão realizar solicitações de serviços e consultas diversas pela *internet*. Para implantação do aplicativo foram utilizados os conceitos de sistemas, sistemas de informações e engenharia de *software*, esta última com o uso da metodologia *Scrum* de desenvolvimento, a Linguagem de Programação (LP) *Ruby*, o *Framework Ruby on Rails*, o *Framework* de *front-end Bootstrap*, a linguagem de marcação de texto HTML (*Hyper Text Markup Language*), a linguagem de folha de estilo CSS (*Cascading Style Sheets*), a linguagem interpretada *JavaScript*, o sistema gerenciador de banco de dados (SGBD) *MySQL* e a linguagem de modelagem UML (*Unified Modeling Language*).

Palavras-chave: Sistemas de informações. Sistema *Web*. *Ruby On Rails*. Controle dos condomínios.

ABSTRACT

The Information systems are constantly being adopted by companies. This growing is given mainly by aggregates benefits with its use what includes its importance in decision making, establishment of strategies and competitive advantage. This work demonstrates the development of a management information system for managing condominiums Picos-Real Estate. The objective of this project was to implement a web system that automates the management processes of activities, thereby addressing their needs perceived by the analysis and data collection performed in the administrator. With this, it is possible greater control of condominiums and interested may carry out requests services and several consultations by internet. For application deployment were used the concepts of systems, information systems and software engineering, the latter with the use of Scrum methodology of development, the Programming Language (LP) Ruby, the Framework Ruby on Rails, the Framework of front-end Bootstrap, the HTML text markup language (Hyper Text Markup Language), the CSS style sheet language (Cascading Style Sheets), the interpreted language JavaScript, the database manager system (DBMS) MySQL and the language of UML modeling (Unified Modeling Language).

Keywords: *Information systems. Web System. Ruby On Rails. Control of condominiums.*

LISTA DE FIGURAS

Figura 1 – <i>Ranking</i> das linguagens de programação mais populares. Maio-2015	27
Figura 2 – Diagrama de Casos de Uso da Visão do Usuário Externo	42
Figura 3 – Diagrama de Casos de Uso da Visão Geral do Sistema	43
Figura 4 – Diagrama de Classes	45
Figura 5 – Diagrama Entidade Relacionamento (DER)	46
Figura 6 – Diagrama de atividades de solicitação de apartamento	47
Figura 7 – Diagrama de Atividades de autenticação no sistema	48
Figura 8 – Diagrama de Atividades de cadastro de condomínio e apartamento	49
Figura 9 – Página Inicial da Aplicação acessada por <i>notebook</i>	51
Figura 10 – Página inicial da aplicação: (a) acessada por <i>smartphone</i> ; (b) acessada por <i>tablet</i>	52
Figura 11 – Solicitação de apartamento. (a) ícone "Solicitar Apartamento"; (b) formulário de solicitação	53
Figura 12 – Buscar solicitação (página exibida em <i>tablet</i>)	54
Figura 13 – Lista de solicitações (resultado da busca exibida pela Figura 12 – página exibida em <i>notebook</i>)	54
Figura 14 – Autenticação no sistema. (a) Tela de autenticação (exibida por <i>smartphone</i>); (b) Página inicial para usuário logado (exibida por <i>tablet</i>)	55
Figura 15 – Página de condomínios. (a) Exibida por <i>smartphone</i> ; (b) Exibida por <i>tablet</i>	56
Figura 16 – Lista de apartamentos de um condomínio	57
Figura 17 – Página de cadastro de cliente (<i>notebook</i>)	57
Figura 18 – Tela de cadastro de cliente (<i>tablet</i>)	58
Figura 19 – Página de cadastro de funcionário. Mensagens e alertas	59
Figura 20 – Exemplos de presteza no sistema	60
Figura 21 – Exemplo de ações mínimas no sistema (Novo Condomínio). (a) cliques necessários; (b) resultado	61
Figura 22 – Menus do sistema: (a) usuário externo; (b) usuário funcionário	62
Figura 23 – Análise de usabilidade (presteza)	63
Figura 24 – Análise de usabilidade (ações mínimas)	64
Figura 25 – Análise de usabilidade (densidade informacional)	64
Figura 26 – Análise de usabilidade (controle do usuário)	64
Figura 27 – Análise de usabilidade (mensagens de erro)	65

Figura 28 – Análise de usabilidade (compatibilidade)	65
Figura 29 – Questionário da análise de usabilidade	70

LISTA DE QUADROS

Quadro 1 – Requisitos funcionais	37
Quadro 2 – Requisitos não-funcionais	38
Quadro 3 – Regras de negócio	38
Quadro 4 – Atores do sistema	39
Quadro 5 – Fluxo de solicitação de apartamentos (representação do diagrama de atividades Figura 6)	45
Quadro 6 – Fluxo de autenticação (representação do diagrama de atividades Figura 7)	47
Quadro 7 – Fluxo de cadastro de condomínio e apartamento (representação do diagrama de atividades Figura 7)	48

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i>
ASD	<i>Adaptive Software Development</i>
CSS	<i>Cascading Style Sheets</i>
CoC	<i>Convention over Configuration</i>
CPF	Cadastro de Pessoa Física
DER	Diagrama Entidade Relacionamento
DRY	<i>Don't Repeat Yourself</i>
EAD	Ensino a Distância
ES	Engenharia de <i>Software</i>
FDD	<i>Feature Driven Development</i>
GNU-GPL	<i>General Public Licence</i>
HTML	<i>HiperText Markup Language</i>
IaaS	<i>Infrastructure as a Service</i>
IHC	Interação Humano-Computador
IRB	<i>Interactive Ruby</i>
JSON	<i>JavaScript Object Nonation</i>
LP	Linguagem de Programação
LSD	<i>Lean Software Development</i>
MVC	<i>Model-View-Controller</i>
PaaS	<i>Platform as a Service</i>
RWD	<i>Responsive Web Design</i>
SaaS	<i>Software as a Service</i>
SAD	Sistema de Apoio a Decisão
SAE	Sistema de Apoio ao Executivo
SE	Sistemas de Escritório
SGBD	Sistema Gerenciador de Banco de Dados
SIG	Sistema de Informação Gerencial
SO	Sistema Operacional
SPT	Sistema de Processamento de Transações
STC	Sistemas de Trabalhadores do Conhecimento
TIC	Tecnologias da Informação e Comunicação
UML	<i>Unified Modeling Language</i>

URL Uniform Resource Locator
XP *Extreme Programming*

SUMÁRIO

1	INTRODUÇÃO	17
1.1	Objetivo	18
1.2	Organização do Documento	18
2	REFERENCIAL TEÓRICO	19
2.1	Sistemas	19
2.1.1	Sistemas de Informação	20
2.1.2	Ciclo de Vida e Especificação dos Sistemas	20
2.1.3	Sistemas <i>Web</i>	21
2.2	Usabilidade	22
2.2.1	<i>Web Design</i> Responsivo	24
2.3	Tecnologias e Metodologias de Desenvolvimento	25
2.3.1	Linguagem de programação <i>Ruby</i>	25
2.3.2	<i>Framework Ruby on Rails</i>	27
2.3.3	<i>Framework de Front-end Bootstrap</i>	29
2.3.4	HTML (<i>Hyper Text Markup Language</i>)	29
2.3.5	<i>JavaScript</i>	30
2.3.6	CSS (<i>Cascading Style Sheets</i>)	31
2.3.7	MySQL	31
2.3.8	Engenharia de Software	32
2.3.9	Modelos de Processos Ágeis	33
2.3.10	<i>Scrum</i>	34
2.3.11	Engenharia de requisitos	35
2.3.12	UML	36
3	SISTEMA WEB PARA CONTROLE E GESTÃO DE CONDOMÍNIOS UTILIZANDO PRINCÍPIOS DE RESPONSABILIDADE	38
3.1	Requisitos do Sistema	39
3.1.1	Requisitos Funcionais	39
3.1.2	Requisitos Não-Funcionais	40
3.1.3	Regras de Negócio	40
3.2	Principais Diagramas do Sistema	41
3.2.1	Diagramas de Casos de Uso	41

3.2.2	Diagrama de Classes.....	45
3.2.3	Diagrama de Atividades	47
4	FUNCIONAMENTO DO SISTEMA.....	51
4.1	Principais Interfaces.....	51
5	AVALIAÇÃO DE USABILIDADE	60
5.1	Análise do Autor	60
5.1.1	Presteza.....	60
5.1.2	Ações Mínimas	61
5.1.3	Densidade Informacional.....	61
5.1.4	Controle do Usuário.....	62
5.1.5	Mensagens de Erro	62
5.1.6	Compatibilidade.....	62
5.2	Análise dos Usuários.....	63
6	CONSIDERAÇÕES FINAIS.....	66
	REFERÊNCIAS	67
	APÊNDICES.....	69

1 INTRODUÇÃO

Com o aumento na quantidade de informações disponíveis nas empresas surgiu a necessidade do trabalho integrado entre as áreas que a compõe, o que dificulta o armazenamento, a recuperação e a integração dessas informações, tornando-se uma tarefa onerosa e pouco confiável. Também movidas pelos avanços constantes das Tecnologias da Informação e Comunicação (TICs) e as rotineiras mudanças no mercado, as instituições buscam a automação de seus processos através de sistemas de informações, motivadas em muito pelos seus benefícios e possíveis vantagens agregadas.

Os sistemas de informações são responsáveis pelos dados acerca de tudo aquilo que for considerado significativo para a organização, além disso são capazes de estruturar os dados e informações e auxiliar no controle e tomada de decisões cooperando com os responsáveis na análise de problemas e complexidades (LAUDON; LAUDON, 2004). É possível notar sistemas dessa natureza em hospitais, universidades, escritórios, administradoras de condomínios, empresas de viagem e muitas outras.

Nas organizações os sistemas são utilizados de forma a integrar todos os níveis, e com a utilização dos sistemas de informações elas podem alcançar seus objetivos através de benefícios como: redução da carga de trabalho, redução de custos e desperdícios, redução de erros, aumento de produtividade, qualidade e segurança, e maximização do lucro e competitividade.

Na administradora de condomínios Picos-Imobiliária, as atividades de gestão encontram-se estáticas e as ferramentas disponíveis e utilizadas não favorecem a realização das tarefas, distanciando o contato com os clientes. Os interessados em alugar apartamentos devem se deslocar até a administradora de condomínios e mesmo estando nas dependências físicas da empresa, não tem nenhuma ou pouca informação sobre os condomínios e imóveis disponíveis, o que torna esse método desgastante para todos os envolvidos no processo.

Após a identificação dos problemas e diagnóstico das necessidades da organização, foi aprovada a proposta de desenvolvimento de um sistema *web* para controle e gestão das atividades condominiais que facilite a interação da administradora com seus clientes e favoreça o acesso através de diversos dispositivos.

A pesquisa realiza um estudo de caso a fim de buscar evidências quanto aos benefícios na utilização de um sistema informatizado para automatização de processos de uma empresa que atua na área de locação de imóveis. Desta forma, o trabalho pretende apresentar

uma ferramenta capaz de gerenciar informações úteis e que possa auxiliar tantos clientes como gestores na tomada de decisões.

1.1 Objetivo

O objetivo desse trabalho é desenvolver um sistema *Web* para a administradora de condomínios Picos-Imobiliária, que automatize o processo de gerenciamento de suas atividades (controle de clientes, funcionários, apartamentos), facilite o acesso a plataforma e contato com clientes.

1.2 Organização do Documento

O presente projeto está organizado em 6 capítulos. O capítulo 2, intitulado Referencial Teórico, apresenta o embasamento teórico para produção do trabalho. Nele são citados trabalhos e conceitos de diversos autores, em várias áreas de conhecimento, para auxiliar na compreensão do trabalho proposto e, por consequência, subsidiar meios para desenvolver a aplicação objeto deste trabalho de pesquisa. No capítulo 3, nomeado de Sistema *Web* para Controle e Gestão de Condomínios, será exibida a análise e especificação de requisitos para o sistema, as informações relacionadas ao desenvolvimento e os diagramas de modelagem visual que facilitam o entendimento do problema. No capítulo 4, Funcionamento do Sistema, são apresentadas as principais funcionalidades e os atores que interagem com a aplicação. No capítulo 5, nomeado Avaliação de Usabilidade, são expostos os resultados obtidos na avaliação de usabilidade do software. Por fim, o capítulo 6, intitulado Considerações Finais, como o próprio nome sugere, contempla as considerações finais e indicações de trabalhos futuros. Além dos capítulos mencionados, o trabalho contém seções para Referências Bibliográficas e Apêndices.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta o embasamento teórico dos temas relacionados à pesquisa e referencia fontes de autores para os temas abordados, que incluem definições de sistemas, usabilidade - com foco em Interação Humano Computador (IHC), tecnologias e metodologias de desenvolvimento e, por fim, trabalhos relacionados à temática deste projeto.

2.1 Sistemas

Os sistemas podem ser percebidos através de três componentes básicos: entrada (informações recebidas pelo sistema), processamento (informações transformadas pelo sistema) e saída (resultado das informações processadas). De acordo com Rezende (2007) os sistemas são um composto de elementos ou partes interconectadas que se comunicam entre si, constituindo-se para alcançar um todo unificado e que realizam determinadas funções de forma organizada. Nas organizações a composição de sistemas supera os três componentes básicos de forma a agregar novos componentes como:

- Ambiente do sistema;
- Aplicabilidade do sistema;
- Objetivos do sistema;
- Recursos do sistema;
- Comportamento do sistema;
- Componentes humanos do sistema; e
- Gestão do sistema.

A interação entre os elementos do sistema é de extrema importância para que os objetivos sejam alcançados. Para Puga, França e Goya (2013), um sistema de informação é “... sistema computacional ou manual utilizado para manipular dados, composto por um conjunto de elementos interdependentes e logicamente associados, tendo como objetivo prover informações”. Portanto, pode-se concluir que um sistema de informação é um tipo de sistema, que pode ou não ser automatizado, e que quando bem utilizado é uma poderosa ferramenta de gerenciamento.

2.1.1 Sistemas de Informação

Os sistemas de informação podem assumir inúmeras responsabilidades e trazer incontáveis benefícios para as organizações. Para isso é necessário estudo, preparação e adequação do ambiente organizacional para que se possa tirar maior proveito desse recurso.

Um sistema de informação é um conjunto de elementos interligados que coletam e manipulam dados gerando informações que possibilitam um mecanismo de realimentação para alcançar um objetivo e facilitar a gestão da instituição. Organizações e pessoas que detenham informações diferenciadas e organizadas são capazes de atuar de forma diferenciada e promissora (REZENDE, 2007).

Para Laudon e Laudon (2004) os sistemas de informação possuem informações consideráveis sobre a organização e seus componentes (pessoas, ambiente, locais), sendo capazes de dar suporte aos gerentes e sua equipe na concepção de assuntos complexos, análise e solução de problemas e busca por inovação, contribuindo dessa forma para a coordenação, controle e tomada de decisões na empresa.

Nota-se que os sistemas de informação são capazes de causar grande impacto na gestão de uma organização agregando inúmeros benefícios. Porém é importante perceber que o simples ato da implantação do sistema não irá ascender a organização e solucionar seus problemas. É necessário estudo prévio das necessidades, dos riscos e das metodologias para que as melhorias sejam alcançadas.

2.1.2 Ciclo de Vida e Especificação dos Sistemas

O ciclo de vida de um sistema pode ser entendido como um conjunto sequencial de operações que vai desde a sua concepção até a sua eliminação. De acordo com Rezende (2007) um sistema de informação automatizado pode ser curto quando não ajustado ou passado por melhorias, e abrange as seguintes fases: concepção; construção; disponibilização; ajustes e melhorias; maturidade; declínio; manutenção; e “morte”. As três primeiras etapas precisam de cuidado especial, pois quando mal elaboradas são capazes de apressar a morte do sistema.

Os sistemas de informação precisam estar integrados e ser capazes de render as informações necessárias para a organização, em cada um de seus níveis, independente da heterogeneidade e grau de importância entre eles. Para Laudon e Laudon (2004) existem quatro tipos de sistemas de informação tidos como principais e responsáveis por atender aos

diferentes níveis organizacionais e que subdividem-se em sistemas específicos responsáveis pelas funções de negócios em cada nível gerencial. A seguir, é feito um comentário sucinto sobre os principais tipos de sistemas de informação de acordo com o nível organizacional:

- Sistemas de nível operacional – ajudam os encarregados deste setor e tem como objetivo fundamental auxiliar no acompanhamento de transações na organização. Os sistemas de processamento de transações (SPTs) atuam de forma específica nesse nível.
- Sistemas de nível de conhecimento – apoiam os dados da organização e trabalhadores deste setor, e tem como principais objetivos a adaptação de novas tecnologias e organização de documentos. Os sistemas de trabalhadores do conhecimento (STCs) e os sistemas de escritório (SE) atuam de forma específica nesse nível.
- Sistemas de nível gerencial – preocupam-se principalmente com a direção em que a organização está seguindo, em realizar projeções futuras sobre os impactos da tomada de decisões, em fazer o devido controle das atividades, identificando se os negócios estão de fato indo bem. Os sistemas de informações gerenciais (SIGs) e os sistemas de apoio a decisão (SADs) atuam de forma específica nesse nível.
- Sistemas de nível estratégico – tem como objetivo preparar o ambiente interno para possíveis mudanças do mercado e auxiliar nas questões estratégicas, desta natureza, a longo prazo. Os sistemas de apoio ao executivo (SAEs) atuam de forma específica nesse nível.

2.1.3 Sistemas *Web*

A *web* pode ser retratada como um sistema de informação em hipertexto, gráfico, distribuído, disponível para qualquer plataforma, interativo, dinâmico e global, utilizado na *Internet*. Desde a década de 1960 com o surgimento da *Internet*, a *web* vem agregando inúmeros serviços de forma acelerada, e estes estão em constante processo de desenvolvimento, surgindo a todo momento novidades e tecnologias que podem acelerar, substituir ou agregar novas funcionalidades aos serviços já existentes (RODRIGUES, 2010).

Segundo Torres (2012), um sistema *web* pode ser entendido como um produto *web*, e definido pela sua capacidade de armazenar dados, realizar funcionalidades, disponibilizar conteúdo e possibilitar que os usuários sejam reconhecidos durante sua utilização. Eles podem

ser classificados de três diferentes formas: do ponto de vista da tecnologia utilizada; de acordo com o que o sistema é capaz de fazer; ou para quem ele resolve um problema.

Do ponto de vista da tecnologia utilizada:

- SaaS (*Software as a Service*) ou *Software* como um serviço.
- PaaS (*Platform as a Service*) ou Plataforma como um serviço.
- IaaS (*Infrastructure as a Service*) ou Infraestrutura com um serviço.

Do ponto de vista do que o sistema é capaz de fazer:

- Comércio eletrônico;
- Pagamentos;
- Gestão de conteúdo;
- Educação;
- Comunicação;
- Colaboração; entre outros.

Para quem o sistema resolve um problema:

- Para o consumidor final. Exemplos: Netflix (www.netflix.com); LinkedIn (www.linkedin.com); ContaCal (www.contacal.com.br).
- Para as empresas. Exemplos: Locaweb (www.locaweb.com.br), Google AdWords (www.google.com.br/AdWords), Cobre Grátis (www.cobregratis.com.br).
- Mistos. Exemplos: Buscapé (www.buscape.com.br), Mercado Livre (www.mercadolivre.com.br), UOL (www.uol.com.br).

Dessa forma é possível perceber diversos padrões dos sistemas *web* que garantem maior acessibilidade e visibilidade, podendo ser compreendidos pela disponibilidade de seus serviços ou pela sua utilização através da *Internet*.

O sistema desenvolvido e base para este trabalho se enquadra na segunda forma mencionada: Do ponto de vista do que o sistema é capaz de fazer.

2.2 Usabilidade

Há anos a *internet* encontra-se consolidada no mercado e vem proporcionando facilidades consideráveis aos seus usuários, como por exemplo: a comunicação entre pessoas fisicamente distantes (redes sociais); a Educação à Distância (EAD); a realização de compras, vendas e pagamentos *online*; entre outras atividades que tornaram-se viáveis com a sua

utilização. Porém as pessoas exigem facilidade e comodidade quando estão conectadas à rede mundial de computadores, e é esse o foco da IHC.

A IHC como seu próprio nome sugere, preocupa-se diretamente com a comunicação dos usuários com a aplicação, analisando fatores como: desempenho; usabilidade; ergonomia; praticidade; comportamento; entre outros. De acordo com Netto (2006) as interfaces tem como principal objetivo oferecerem uma interação amigável aos usuários, e é a partir das interfaces que a IHC avalia a aprendizagem de uso. Dessa forma, tem-se a interface como uma parcela essencial de um sistema computacional, pois é através dela que o usuário vai interagir com o *software*, e decidir pela sua aprovação ou rejeição. Elas precisam tornar a comunicação com o sistema o mais simples e atrativa possível, abstraindo complexidades, disponibilizando sequências simples, claras, e mostrando nitidamente cada passo da interação, permitindo que o usuário navegue de forma fácil e segura pela aplicação.

É de suma importância durante a utilização de um sistema, que o usuário consiga, com facilidade, resolver seus problemas e entender as funcionalidade disponíveis. Barbosa e Silva (2010) descrevem algumas características principais, responsáveis pela qualidade de uso de sistemas, são elas:

- Usabilidade – caracteriza-se pela facilidade de uso, aprendizado e pela satisfação do usuário com a interface, destacando características, como: sua compreensão, sua capacidade de agir sobre a interface e perceber as respostas do sistema.
- Experiência do Usuário – determinada pelos sentimentos e emoções dos usuários.
- Acessibilidade – caracteriza-se pela eliminação de barreiras que impeçam que mais usuários (deficientes ou não) sejam capazes de interagir com sistema.
- Comunicabilidade – é a forma como o *designer* interpreta os usuários e então toma decisões de como apoiá-los, tornando o sistema acessível e usual.

Outros fatores também podem ser analisados para se considerar a usabilidade de um sistema, como: facilidade de aprendizado; facilidade de recordação; eficiência; flexibilidade; segurança no uso; satisfação do usuário; e outros.

Outra ferramenta que está diretamente ligada a usabilidade e tem ganhado bastante força no mercado, principalmente devido à crescente utilização de dispositivos móveis, é o *web design* responsivo.

2.2.1 *Web Design* Responsivo

De acordo com Silva (2014) o *Responsive Web Design* (RWD), ou Web Design Responsivo, surgiu em 2010 depois que Ethan Marcotte divulgou um artigo com o título “*Responsive Web Design*”, e nele evidenciou o crescimento e a mudança de comportamento dos usuários *mobile*, que queriam seus *websites* disponíveis para os mais diferentes dispositivos, alertando que essas mudanças já estariam afetando as empresas.

Um *site* ou um sistema com *Web Design* Responsivo é capaz de ser acessado por qualquer dispositivo móvel, seja ele: notebook, TV, *smartphone*, *tablet*, geladeira, independente da sua resolução de tela, bastando que possua acesso à rede. Zemel (2012) destaca algumas estatísticas do mundo *mobile* que justificam a utilização do *Web Design* Responsivo, por exemplo:

- Em 2010, as vendas de celulares ultrapassaram as vendas combinadas de desktops e notebooks algo que só estava previsto apenas para o ano de 2012.
- Em 2011, as vendas de *smartphones* cresceu cerca de 179% no Brasil.

Os dispositivos *mobiles* possuem uma taxa de crescimento 4 vezes maior do que a da população mundial, em um comparativo entre a venda de dispositivos móveis e a taxa de natalidade mundial. E financeiramente os números também são fantásticos, por exemplo:

- Em 2010, as vendas mundiais somente através de dispositivos móveis chegaram a quase US\$ 2 bilhões no eBay¹.
- Somente por meio de pagamentos *mobile* o PayPal² movimenta US\$ 10 milhões diariamente.

É evidente que os dispositivos móveis já são uma realidade, e os usuários exigem maior flexibilidade e facilidade para acessar seus *sites* e aplicações, e a nova prática para proporcionar essa funcionalidade é por meio do desenvolvimento responsivo. Este, por sua vez, oferece facilidade na leitura e adequação do conteúdo disponível independentemente do dispositivo que esteja garantindo o acesso, dessa forma, tornando as aplicações mais interativas e agradáveis de serem utilizadas pelos usuários.

Silva (2014) destaca três componentes fundamentais para o desenvolvimento do *Design* Responsivo, são eles:

¹ <www.ebay.com>

² <www.paypal.com.br>

- *Grid* fluído ou *Layout* fluído – é o responsável por garantir uma adaptação natural do conteúdo ao dispositivo, e faz isso definindo unidades de medida CSS relativas, como: porcentagens e *ems* (a unidade de porcentagem é relativa ao bloco onde o elemento estiver contido, enquanto que o *em* representa o tamanho da fonte).
- Imagens e Mídia flexíveis – são responsáveis por garantir a adaptação de imagens, vídeos e *widgets* (pequenos aplicativos que compõem uma interface gráfica) em geral, garantido a contração ou expansão de suas dimensões.
- *Media queries* – é uma tecnologia CSS que permite ao usuário servir folhas de estilo ocultando, ou reposicionando elementos baseadas nas características do dispositivo ao qual o *layout* será servido.

A seguir serão citadas três tecnologias necessárias que compõem a base de sustentação do *Design* Responsivo:

- HTML5 (versão 5);
- JavaScript para compatibilidades;
- CSS3 (versão 3).

Com a utilização adequada desses componentes tecnológicos é possível criar *Designs* Responsivos.

2.3 Tecnologias e Metodologias de Desenvolvimento

Diversas tecnologias e métodos foram utilizadas no desenvolvimento da aplicação, como: a linguagem de programação, *Ruby*; o *framework* para desenvolvimento *web*, *Ruby on Rails*; o *framework* para *front-end*, *Twitter Bootstrap*; o método ágil, *Scrum*; entre outros. Cada tecnologia possui tarefas específicas e juntas colaboraram para os resultados do trabalho.

As principais tecnologias e métodos utilizadas durante esse projeto são abordadas, de forma mais específica, a seguir.

2.3.1 Linguagem de programação *Ruby*

Criada para ser uma linguagem de *script* e que fosse mais poderosa que o *Perl*, e mais orientada a objetos que o *Python*, a linguagem de programação *Ruby* surgiu em 1993, criada pelo programador japonês Yukihiro Matsumoto, também conhecido como Matz, mas

só foi apresentada ao público pela primeira vez em 1995. Matz criou o *Ruby* para ser uma linguagem simples e de fácil entendimento, de forma a facilitar o desenvolvimento e manutenção de sistemas com ela escritos (SOUZA, 2012).

O *Ruby* possui algumas características que fazem dela uma linguagem poderosa e flexível, como:

- É uma linguagem interpretada, ou seja, seu código fonte é executado por um interpretador.
- É multiparadigma, pois suporta dois ou mais paradigmas de programação, o que permite ao programador lidar com vários estilos (por exemplo: funcional, orientado a objetos, reflexivo e imperativo).
- É multiplataforma, pois pode ser executado em mais de uma plataforma.
- Possui tipagem forte e dinâmica já que a declaração de tipos de dados é opcional e podendo alterá-la durante a compilação, mas exige que o tipo de dado de um valor seja do mesmo tipo da variável.
- É puramente orientada a objetos, pois tudo na linguagem é tratado como objeto.

Além dessas características anteriormente citadas, o Ruby-Lang (2015) demonstra outros atributos bastante interessantes da linguagem, e são eles:

- É altamente portátil.
- Pode carregar bibliotecas de extensão dinamicamente se o Sistema Operacional (SO) permitir.
- Possui capacidade de tratamento de exceções, o que facilita o tratamento de erros.
- Possui *multithreading*, independente do SO.
- Dispõem de *garbage collector* para todos os objetos *Ruby*.
- Escrever extensões C em *Ruby* é mais fácil do que *Perl* ou *Python*, com uma API (*Application Programming Interface*) refinada para chamar *Ruby* desde o código C.

Além dessas qualidades que viabilizam uma escrita mais curta de código e fácil entendimento, o *Ruby* possui um recurso bastante admirado pelos seus desenvolvedores, o IRB (*Interactive Ruby*), onde os comandos são interpretados a medida que vão sendo inseridos no seu terminal, tudo de forma bem interativa.

A linguagem *Ruby* foi inspirada principalmente nas linguagens *Python*, *Perl*, *Smalltalk*, *Lisp*, *Eiffel* e *Ada*, extraindo o que há de melhor dessas linguagens. Atualmente encontra-se entre as linguagens mais utilizadas e essa popularização deve-se em muito pela

propagação do seu principal *framework*, o *Ruby on Rails*, destacando-se principalmente pela sua boa performance, simplicidade e elegância (SOUZA, 2012).

De acordo com Índice TIOBE (2015) a linguagem *Ruby* é atualmente a décima quinta (15ª) linguagem de programação mais popular do mundo, e apesar da queda em relação ao mesmo período do ano passado, o *Ruby* encontra-se em uma variação de leve crescimento (+0,03%) em uma lista que tem a LP *Java* no seu topo. A figura 1 apresenta o *ranking* das linguagens de programação mais populares.

Mai 2015	Mai 2014	Mudança	Linguagem De Programação	Classificações	Mudança
1	2	▲	Java	16,869%	-0,04%
2	1	▼	C	16,847%	-0,08%
3	4	▲	C ++	7,875%	+ 1,89%
4	3	▼	Objective-C	5,393%	-6,40%
5	6	▲	C #	5,264%	+ 1,52%
6	8	▲	Python	3,725%	+ 0,67%
7	9	▲	JavaScript	3,127%	+ 1,34%
8	11	▲	Visual Basic .NET	2,968%	+ 1,70%
9	7	▼	PHP	2,720%	-0,67%
10	-	▲▲	Visual Básico	1,893%	+ 1,89%
11	10	▼	Perl	1,820%	+ 0,35%
12	33	▲▲	R	1,444%	+ 1,06%
13	15	▲	Delphi / Object Pascal	1,302%	+ 0,33%
14	19	▲▲	MATLAB	1,283%	+ 0,57%
15	12	▼	Rubi	1,273%	+ 0,03%
16	27	▲▲	COBOL	1,169%	+ 0,58%
17	22	▲▲	PL / SQL	1,127%	+ 0,49%
18	-	▲▲	Rápido	1,115%	+ 1,12%
19	18	▼	Pascal	0,960%	+ 0,21%
20	37	▲▲	ABAP	0,887%	+ 0,57%

Figura 1 - *Ranking* das linguagens de programação mais populares. Maio-2015
 Fonte: < <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html> >

2.3.2 *Framework Ruby on Rails*

Um *framework* é um conjunto de *scripts* (instruções a serem seguidas) que dão suporte a uma programação mais rápida, simples e eficiente, possuindo funções, rotinas e

variáveis que poupam bastante tempo de desenvolvimento. O *Ruby on Rails* é um *metaframework* de desenvolvimento *web*, escrito em *Ruby* (mas que não exige grande conhecimento na linguagem), de código aberto e gratuito, e que foi elaborado para facilitar e acelerar o desenvolvimento de aplicações *web* (HARLT, 2012). Foi criado por David Heinemeier Hansson, também conhecido como DHH, a partir do desenvolvimento de projetos na empresa *37signals*, e oficialmente lançado em 2004. O *Rails* é considerado um *metaframework* por ser um *framework* composto de *frameworks*. Entre os *frameworks* que o compõem temos (RAILSGUIDES, 2015):

- *Active Record* – é a base para os modelos em uma aplicação, sendo responsável pela abstração dos dados e pela interação entre o banco de dados e a aplicação.
- *Action Pack* – é responsável pelos controles, apresentações e rotas da aplicação. Abrange o *Action View* (responsável por gerar as visualizações), *Action Controller* (responsável pelo controle de fluxo) e o *Action Dispatch* (responsável pelo gerenciamento de rotas).
- *Action Mailer* – é responsável pela função de recebimento e envio de e-mails.
- *Active Support* – abrange diversas classes e extensões de bibliotecas bastante úteis e utilizadas no desenvolvimento de aplicações.

Segundo Fuentes (2012) o *Rails* utiliza o padrão de arquitetura MVC (*Model-View-Controller*) dividindo o projeto em três camadas, são elas:

- *Models* (Modelos) – auxiliam o armazenamento e a busca dos dados no repositório de dados, além de comporem as regras de negócios e cederem aos controladores a possibilidade de acesso às funções da aplicação.
- *Views* (Apresentação) – representam a forma como os dados da aplicação serão apresentados aos usuários. Normalmente são representadas pelas páginas *web* que utilizam tecnologias como: CSS3, HTML5 ou até JSON (*JavaScript Object Notation*).
- *Controllers* (Controladores) – é responsável por compreender as ações vindas da URL através das ações de usuários e repassar para os modelos, que responderão ao *controller*, e em seguida redirecionam para uma *view* a resposta solicitada.

Dessa forma o *Rails* divide o projeto em três camadas bem definidas, garantindo a separação das tarefas e uma comunicação desacoplada, o que facilita bastante a manutenção das aplicações.

Fuentes (2012) ainda destaca outro grande atrativo do *Ruby on Rails* é que ele permite que as funcionalidades das aplicações possam ser implementadas de forma

incremental, apresentando um código mais simples e elegante por conta de duas de suas filosofias, a saber:

- *Don't Repeat Yourself* (DRY) – que sugere a não repetição de código na aplicação incentivando a sua reutilização.
- *Convention over Configuration* (CoC) – que permite a geração de código pelo próprio *Rails*, podendo o desenvolvedor escrever menos código e evitando configurações desnecessárias.

2.3.3 Framework de Front-end Bootstrap

De acordo com Cochran (2012) o *Twitter Bootstrap* ou somente *Bootstrap* foi produzido por Mark Otto e Jacob Thornton e lançado pela primeira vez em 2011, enquanto desenvolvedores do *Twitter*. É um *framework* de código aberto, com estrutura de *front-end* elegante, intuitiva e extremamente significativa para um desenvolvimento *web* mais simples e rápido. Em 2012, o *Twitter Bootstrap* chegou a sua versão 2.0 agregando inúmeras melhorias significativas, sendo a principal delas a de um *layout* responsivo capaz de se adaptar aos mais variados dispositivos: *tablets*, *smartphones*, *desktops* e outros, ganhando reconhecimento e aceitabilidade.

O *Bootstrap* vem equipado com algumas tecnologias como: HTML, CSS e *JavaScript*. Para Magno (2013) o *Bootstrap* é adequado para qualquer tipo de desenvolvedor, seja ele experiente ou não, pois agrega soluções que se adequam para qualquer cenário. Com ele não é necessário definir elementos de interface a partir do zero, já que possui ferramentas com convenções e padrões de classes simples e bem definidas, prontas para serem usadas e personalizadas a critério do desenvolvedor.

Por meio do *Twitter Bootstrap*, de seus padrões e de suas tecnologias é possível a criação de *designs web* responsivos para qualquer plataforma, facilitando a navegação e interação com o usuário.

2.3.4 HTML (*Hyper Text Markup Language*)

De acordo com Rodrigues (2010) o HTML é uma linguagem de marcação de código criada em 1990 por Tim Berners-Lee. Através do HTML é possível a apresentação de informações na *internet*, onde o mesmo determina onde os elementos (textos, imagens, tabelas, vídeos, entre outros elementos) devem ser visualizados nas páginas *web*, o processo é

possível por meio da interpretação do código pelo *browser*. Junto com o CSS, formam a base para o desenvolvimento *front-end*, i.e., as páginas de apresentação de uma aplicação.

O HTML, assim como o CSS e o *JavaScript*, são tecnologias importantes no desenvolvimento de *designs web* responsivos e também encontra-se presente e de forma indispensável na camada de apresentação do *framework ruby on rails* (citado na seção 2.3.2).

Atualmente o HTML encontra-se na sua versão 5 e de acordo com Silva (2014), proporciona, entre outras, as seguintes vantagens no desenvolvimento *web*:

- Código claramente mais semântico, estando agora mais simples e fácil de ser compreendido.
- Melhor estruturação do código, de forma a facilitar a leitura e o suporte.
- Melhor acessibilidade, sendo capaz de ser executado em dispositivos de baixa potência.
- Melhores benefícios para estratégias de otimização de *sites*.

2.3.5 *JavaScript*

Criada em 1995 por Brendan Eich, o *JavaScript* é uma linguagem de programação interpretada, com orientação a objetos, e projetada como linguagem de *script* capaz de criar programas e realizar ações em uma página *web* e complementar as capacidades do HTML (criar efeitos, botões animados, sons, etc). As páginas *webs* podem ser percebidas em três categorias: o conteúdo, responsabilidade do HTML; a formatação, encargo do CSS; e o comportamento, compromisso do *JavaScript*. E quando bem construídas proporcionam satisfação e êxito nos resultados (RODRIGUES, 2010).

De acordo com Balduino (2012) a escolha do nome *JavaScript* foi uma estratégia inteligente para quebrar a resistência das pessoas em aprender uma nova linguagem e fazer com que a mesma pudesse pegar carona com o sucesso da LP *java*, porém alguns problemas tornaram a linguagem bastante ignorada. Mas com o passar dos anos tornou-se melhor compreendida e aceita, especialmente devido ao desenvolvimento e popularização de ferramentas como o *node.js*³, que permite o desenvolvimento de servidores com *JavaScript*; o *Rhino*⁴, que permite a execução de código *JavaScript*; e o *jQuery*⁵, que simplifica e abstrai o desenvolvimento no *browser*.

³ <https://nodejs.org/>

⁴ <https://developer.mozilla.org/en-US/docs/Mozilla/Projects/Rhino>

⁵ <https://jquery.com/>

2.3.6 CSS (*Cascading Style Sheets*)

De acordo com Rodrigues (2010) a CSS ou folhas de estilo em cascata, teve sua proposta de criação em 1994 por Hakon Wium Lie, e sua primeira versão lançada em 1996, surgindo para superar as diferenças de implementações dos *browsers* e melhorar o visual do *sites*, pois com a popularização do HTML, o mesmo estava ficando sobrecarregado e complexo. A CSS implementou um aspecto inovador quanto ao *design* e desenvolvimento de páginas *web*, resolvendo problemas, economizando tempo, e oferecendo maiores possibilidades de aparência, podendo ser utilizado de três formas:

- *In-line* – com o CSS sendo aplicado diretamente no elemento HTML;
- Interno – criando-se blocos de estilos dentro do código HTML; e
- Externo – criando-se um arquivo separado e dentro do código HTML tem-se apenas *links* para o arquivo. Sendo esta a forma mais utilizada.

Atualmente a CSS encontra-se na sua versão 3 (CSS3) agregando diferentes módulos e funcionalidades de estilização que vieram para facilitar as obrigações do desenvolvedor. Entre as principais novas funcionalidades encontram-se: *text-shadow*, *box-shadow*, *linear-gradient*, *opacity*, *border-image*, *animation*, e entre essas e outras tem-se também as regras de *media queries*, que possibilitam a criação de folhas de estilo para diversos dispositivos, permitindo a criação de *designs* responsivos (SILVA, 2014).

2.3.7 MySQL

De acordo com Milani (2007) o MySQL é um SGBD reacional e com licença de *software* livre baseada nas cláusulas da GNU-GPL (*General Public Licence*) que define o que pode ser feito com a ferramenta e seus recurso, baseando-se em princípios como:

- Utilização – o *software* pode ser utilizado para qualquer finalidade.
- Distribuição – permite a distribuição do *software* entre as pessoas.
- Didática – permite o estudo de seu funcionamento a partir de código-fonte.
- Colaboração – Permite a alteração do código-fonte para evolução da ferramenta, desde que continue livre.

Atualmente é o banco de dados *open-source* (tradução livre: código aberto) mais utilizado em aplicações para a *internet*. E além de ser compatível com a maioria dos SOs

existentes, outras características também justificam sua grande utilização, entre elas podem ser citadas as seguintes:

- Portabilidade – é altamente portátil entre diferentes sistemas, plataformas e compiladores.
- *Multithreads* – aumenta a velocidade do processamento e facilita a integração com *hardwares*.
- Formas de Armazenamento – disponibiliza vários tipos de tabela para armazenamento dos dados, possibilitando a escolha de um tipo de acordo com a situação.
- Segurança – trabalha com criptografia no tráfego de senha.
- Capacidades – alto poder de execução e de armazenamento. É capaz de executar mais de um bilhão de consultas diárias em um *site* ou processar milhares de transações por minuto.

Por padrão o *Ruby on Rails* utiliza o banco de dados SQLite, mas devido a algumas características e vantagens do MySQL optou-se por sua utilização nesse projeto.

2.3.8 Engenharia de Software

Na visão de Sommerville (2011) a Engenharia de *Software* (ES) preocupa-se com toda a perspectiva da elaboração de um sistema, que vai desde as fases iniciais de especificação até a sua manutenção quando já encontra-se em uso, tratando de toda documentação e dados de configurações para que sua operação ocorra corretamente. Dessa forma, cabe aos engenheiros de *software* utilizarem os métodos e técnicas da ES para planejar, controlar e simplificar um projeto, de forma a assegurar sua qualidade e confiabilidade. Devido as diversas categorias de sistemas de *softwares*, que variam desde as mais simples até as mais complexas, não é possível definir um padrão absoluto para a engenharia de *software*, pois as metodologias e técnicas utilizadas variam de acordo com cada projeto. Porém, apesar da heterogeneidade dos sistemas de *software*, todos os processos devem incluir quatro atividades fundamentais para a ES:

- Especificação de *software* – são definidas as características e funcionalidades do sistema.
- Projeto e implementação de *software* – planejamento e desenvolvimento do sistema de acordo com as especificações.
- Validação do *software* – garantir que o sistema atenda às demandas do cliente.

- Evolução do *software* – preparar o sistema para eventuais mudanças.

As ações do processo de *software* mantém a aplicação sob análise durante todas as etapas, de forma a garantir o sucesso da aplicação. Para Pressman (2011) independentemente das características comuns e fundamentais para a ES, as mesmas precisam ser complementadas por outras ações que irão enriquecer o projeto. Dentre elas, as atividades de suporte dominantes são:

- Controle e Acompanhamento do Projeto – é importante que a equipe gerencie e acompanhe o projeto, principalmente para que os prazos sejam cumpridos.
- Administração de Riscos – controla os riscos para que eles não venham a afetar o produto final.
- Garantia da Qualidade de *Software* – determina e direciona as ações necessárias para a obtenção de qualidade do *software*.
- Revisões Técnicas – Avaliam elementos da engenharia de *software*, para que possíveis erros sejam encontrados e corrigidos.
- Medição – Contribui para que o sistema atenda aos requisitos predefinidos.
- Gerenciamento da Configuração do *Software* – Controla as consequências de possíveis mudanças no decorrer das atividades.
- Gerenciamento da Reusabilidade – Define métodos de reutilização e obtenção de componentes reutilizáveis.
- Preparo e Produção de Artefatos de *Software* – Envolve as ações necessárias para criar os elementos necessários para a produção.

As atividades essenciais e de suporte quando utilizadas de forma conjunta no desenvolvimento de sistemas, tendem a agregar maior qualidade e confiabilidade à aplicação.

2.3.9 Modelos de Processos Ágeis

De acordo com Sommerville (2011) um processo de *software* engloba tarefas relacionadas que levam a elaboração de um produto. Já os modelo de processos representam uma visão particular dos processos representando-os de forma simplificada e podendo ser usados para explicar diferentes abordagens de desenvolvimento.

Para se desenvolver um sistema, inúmeras técnicas e metodologias são disponibilizadas pela engenharia de *software*, de forma que se alcance um nível aceitável de organização, segurança e qualidade. Porém, seguir efetivamente esses conceitos, pode levar

tempo, e no mundo corporativo isso nem sempre é aceitável o que leva a priorizar um desenvolvimento ágil, que é principalmente implementado em aplicações *web*. Para isso existem inúmeros modelos de processos ágeis adequados para o desenvolvimento desses sistemas constantemente mutáveis (PRESSMAN, 2011), sendo os mais conhecidos listados a seguir:

- XP (*Extreme Programming*) – tem seu trabalho baseado principalmente em cinco valores. São eles: comunicação, simplicidade, retorno, coragem e respeito. E a partir desse ponto envolve um conjunto de regras e práticas constantes no âmbito de quatro atividades: planejamento, projeto, codificação e testes.
- ASD (*Adaptive Software Development*) – o desenvolvimento de *software* adaptativo é principalmente utilizado na construção de sistemas complexos, baseando-se na colaboração humana e organização automática das equipes.
- Scrum – auxilia no desenvolvimento de tarefas em um processo que englobe atividades como: requisitos, análise, projeto, evolução e entrega. E se mostra bastante eficaz para projetos com curtos prazos de entrega, requisitos mutáveis e críticos de negócio.
- FDD (*Feature Driven Development*) – o desenvolvimento dirigido a funcionalidades é aplicado principalmente em projetos de porte moderado e baseia-se na colaboração entre membros da equipe, gerencia problemas e complexidades, e comunica detalhes através de meios verbais, gráficos e texto.
- LSD (*Lean Software Development*) – o desenvolvimento enxuto de *software* baseia-se em eliminar desperdícios, incorporar qualidade, criar conhecimento, adiar compromissos, entrega rápida e otimização total.

Todos os modelos citados são utilizados no desenvolvimento ágil de *software* e possuem características peculiares, cabendo ao desenvolvedor optar por aquela(s) que melhor se adequar(em) ao seu projeto. Na implementação da aplicação proposta neste trabalho, optou-se pelo modelo de desenvolvimento ágil *Scrum* pelas características descritas no próximo tópico.

2.3.10 Scrum

De acordo com Sabbagh (2013) o *scrum* é um *framework* ágil que caracteriza-se pela sua clareza, simplicidade e bom desempenho em diferentes contextos, quando utilizado de

forma conjunta com outras técnicas e práticas. O método lista uma série de benefícios agregados com sua correta utilização, entre eles estão:

- Entregas frequentes de retorno ao investimento dos clientes.
- Redução dos riscos do projeto.
- Maior qualidade no produto gerado.
- Mudanças utilizadas como vantagem competitiva.
- Visibilidade do progresso do projeto.
- Redução do desperdício.
- Aumento de produtividade.

Além da metodologia de desenvolvimento, que é essencial em um projeto de *software*, outra etapa indispensável e extremamente importante é a de levantamento e análise de requisitos, que entre outras características permite uma melhor compreensão das necessidades e aumenta as chances de êxito do produto final, i.e., o *software*.

2.3.11 Engenharia de requisitos

Para Pádua (2000) os requisitos são atributos que determinam as normas de aceitação de um produto. A engenharia de requisitos está inserida na engenharia de *software* e apresenta diversas técnicas de levantamento, documentação e análise de requisitos. Ela busca apresentar a importância de uma elaboração qualificada e convencer todos os envolvidos em um projeto de que:

- Boas especificações de requisitos são indispensáveis;
- Uma boa especificação de requisitos custa tempo e dinheiro, mas a sua ausência custa muito mais;
- A participação dos usuários na engenharia de requisitos é essencial para que suas necessidades sejam atendidas;
- São investimentos necessários.

De acordo com Sommerville (2011) um dos principais problemas enfrentados pela engenharia de requisitos é não fazer uma correta distinção entre os níveis de descrição do sistema. Ainda segundo o autor, a engenharia de requisitos é subdividida em dois tipos:

- Requisitos de usuário – que mostram os serviços que deverão ser oferecidos pelo sistema e as restrições com as quais o usuário deve operar.

- Requisitos de sistema – especifica de forma detalhada as funções, serviços e restrições do sistema. Define exatamente o que deve ser implementado na aplicação.

Uma forma bastante aceita e utilizada no levantamento de requisitos é através de uma linguagem de modelagem, como a UML por exemplo, apresentando diversas formas de visualização do sistema e facilitando a sua implementação.

2.3.12 UML

Na visão de Guedes (2011) a UML, é a linguagem-padrão de modelagem adotada pela engenharia de *software*, sendo completamente independente do processo de desenvolvimento e da linguagem de programação adotados em um projeto de *software*. A UML é de extrema importância na descrição das características de um sistema, tais como seus requisitos, sua estrutura lógica, seu comportamento, a dinâmica de seus processos e até mesmo as necessidades físicas para a implantação do sistema.

De acordo com Booch *et al.* (2005) a modelagem é um componente essencial para um projeto de *software* ser bem-sucedido e a UML é uma linguagem bastante expressiva e capaz de abranger todas as visões necessárias para a construção e implantação de sistemas. Ela pode ser utilizada para a visualização, a especificação, a construção e a documentação de itens que utilizam sistemas de *software*, podendo alcançar objetivos como:

- Auxílio na visualização do sistema em sua forma atual e futura.
- Permissão para especificar a estrutura ou o comportamento de um sistema.
- Disposição de um roteiro para a construção do sistema.
- Documentação das decisões tomadas.

A UML agrega inúmeras facilidades ao processo de desenvolvimento de *software* disponibilizando inúmeras formas de visualização do sistema, encurtando a relação cliente-desenvolvedor e reduzindo as taxas de erros e atualizações do projeto. Segundo Guedes (2011) inúmeros diagramas são disponibilizados pela UML com o propósito de oferecer inúmeras visões, e tornar possível a descoberta de falhas e suas correções. Entre os principais diagramas da UML estão:

- Diagrama de casos de uso – Apresentam uma visão geral do comportamento do sistema e por meio de linguagem simples e de fácil compreensão identificam os usuários e quais as funções de cada um no sistema.

- Diagrama de classes – Proporciona a visão da estrutura de classes do sistema, exibindo seus atributos, seus métodos e os relacionamentos entre as classes. Servindo de base para outros diagramas.
- Diagrama de objetos – Está diretamente ligado ao diagrama de classes, de forma a complementá-lo fornecendo uma visão dos valores armazenados pelos objetos.
- Diagrama de atividade – Demonstra o controle de fluxo de uma atividade, caracterizando os passos a serem seguidos até a sua conclusão.

Por fim, os diagramas disponibilizados pela UML são úteis e um aliado valioso para o desenvolvedor, tornando-se uma ferramenta indispensável para a produção de um software.

3 SISTEMA WEB PARA CONTROLE E GESTÃO DE CONDOMÍNIOS UTILIZANDO PRINCÍPIOS DE RESPONSABILIDADE

Nesse capítulo serão apresentados os principais diagramas da aplicação, além do seu levantamento de requisitos que juntos irão especificar o sistema e possibilitar a identificação dos seus usuários. A proposta desse trabalho é o desenvolvimento de um sistema *web* para controle e gestão de condomínios, que a partir de agora será referenciado como SCGC.

A aplicação foi desenvolvida sob duas perspectivas: a primeira, perspectiva dos usuários externos, que poderão navegar pelo sistema, visualizar condomínios e apartamentos, verificar se os imóveis encontram-se disponíveis ou não e, finalmente, realizar solicitações de aluguel de apartamentos; a segunda, perspectiva dos usuários internos, que representam a cúpula que compõem a administradora de condomínios. Os responsáveis por essas atribuições serão dignados a manter condomínios, apartamentos, clientes e atender as solicitações dos usuários. Para tanto, os usuários internos precisam estar autenticados na plataforma.

O SCGC trata-se de um sistema de informação *web* que foi concebido utilizando a metodologia *scrum* de desenvolvimento. Por se tratar de um método simples, claro e compatível a diferentes ambientes, ele também tem a vantagem de ser acompanhado e analisado de perto pelos usuários, que vão avaliando o desenvolvimento gradativamente e com isso é capaz de agregar uma série de benefícios ao projeto.

Para o desenvolvimento da aplicação foi utilizado o *framework Ruby on Rails* que disponibiliza uma grande quantidade de recursos e tem como linguagem de programação base, o *Ruby*. Além disso, faz uso da arquitetura MVC que facilita bastante a manutenção dos sistemas. A LP *Ruby* além de ser a base do *framework* utilizado, caracteriza-se por ser uma linguagem simples e de fácil entendimento, o que também facilita o desenvolvimento e manutenção de sistemas. Por estas razões e facilidades foi usada para a produção do *software*. O SGBD utilizado como repositório dos dados foi o *MySQL*, pois apesar de o *Rails* conceber o *SQLite* por padrão, o *MySQL* é considerado mais seguro e ideal para o ambiente de produção, pois implementa um sistema de autenticação robusto, sendo este recurso ausente no *SQLite*. Outro fator que culminou na escolha do SGBD *MySQL*, foi sua capacidade de manter uma base de dados bem superior e realizar consultas de maneira simultânea com eficiência, sendo dessa forma uma melhor opção para esse projeto. Para o *front-end* da aplicação foi utilizado o *framework bootstrap*, que caracteriza-se pela sua flexibilidade na utilização das suas inúmeras ferramentas, que permitem a criação de interfaces personalizadas ou o uso de temas prontos, servindo de base para o *design web* responsivo.

O projeto passou por diversas etapas que vão desde a sua concepção, coleta e análise de requisitos, produção de diagramas e codificação e chegando a sua conclusão com as fases de testes e diagnóstico de usabilidade, que serão apresentados mais adiante.

3.1 Requisitos do Sistema

Os requisitos do sistema são uma forma de expressar as necessidades do cliente para que o *software* possa ser construído de acordo com as imposições especificadas. Para a coleta de requisitos do SCGC foram realizadas reuniões em um modelo de entrevista com o chefe da administradora de condomínios, onde foram identificados os problemas, especificadas as necessidades e definidas as funcionalidades do sistema. A seguir serão apresentados os requisitos funcionais (RF), os requisitos não-funcionais (RNF) e as regras de negócio (RN) da aplicação.

3.1.1 Requisitos Funcionais

Os requisitos funcionais representam as funcionalidade do *software*, i. e., as atividades que o cliente deseja realizar através do sistema (GUEDES, 2011). Os requisitos funcionais do SCGC são exibidos a seguir no Quadro 1.

Quadro 1 - Requisitos funcionais

Identificador	Descrição	Depende de
RF01	O sistema deverá possuir cadastro de usuários associando-os a um cargo.	
RF02	Somente funcionários possuirão <i>login</i> e senha para acesso ao sistema e atuarão com restrição de acordo com seu cargo.	RF01
RF03	Visitantes poderão acessar a sistema sem necessidade de autenticação, mas terão acesso restrungido: visualizar condomínios e apartamentos com seus respectivos detalhes e solicitação de apartamentos para possível aluguel.	
RF04	Os níveis de acesso aos recursos do sistema serão de acordo com o cargo do usuário cadastrado. A exceção são os visitantes.	RF02
RF05	O sistema permitirá cadastro básico de: cargos, endereços, clientes, funcionários, condomínios, e apartamentos.	RF04
RF06	Os apartamentos cadastrados possuirão um status: disponível e indisponível.	RF05
RF07	O sistema permitirá a solicitação de apartamentos disponíveis.	RF06
RF08	O sistema permitirá a visualização de condomínios e apartamentos assim como detalhes de suas estruturas e localidades.	RF05
RF09	O sistema permitirá o gerenciamento dos recursos cadastrados.	RF05

3.1.2 Requisitos Não-Funcionais

Os requisitos não-funcionais são restrições válidas que devem ser levadas em consideração para a consistência dos requisitos funcionais (GUEDES, 2011). O Quadro 2 exhibe os requisitos não-funcionais do SCGC.

Quadro 2 - Requisitos não-funcionais

Identificador	Descrição	Categoria	Depende de
RNF01	O sistema deve controlar o acesso às funcionalidades. O gerenciamento dos recursos do sistema devem ser acessíveis somente por usuários autenticados.	Segurança de Acesso	RF02
RNF02	O sistema deve permitir o acesso a visitantes, porém com restrição de acesso.	Segurança de Acesso	
RNF03	O sistema deve possuir <i>interface</i> intuitiva que permita aos usuários utilizarem a plataforma e realizarem suas atividades sem esforços.	Usabilidade	
RNF04	O sistema deve possuir mensagens ou alertas que orientem os usuários durante a utilização da plataforma.	Usabilidade	
RNF03	O sistema deve estar disponível pela Internet, e capaz de ser acessado por meio dos principais navegadores disponíveis no mercado.	Portabilidade	
RNF04	A persistência das informações deve ser implementada em um Sistema Gerenciador de Bancos de Dados Relacionais (SGBDR) livre (o MySQL foi o escolhido).	Manutenibilidade	

3.1.3 Regras de Negócio

As regras de negócio são identificadas pelos requisitos não funcionais que são as normas e condições estabelecidas pelo cliente e que devem ser seguidas na execução de alguma atividade do sistema (GUEDES, 2011). As regras de negócio do SCGC são apresentadas no Quadro 3.

Quadro 3 - Regras de negócio

Identificador	Descrição	Depende de
RN01	Não serão permitidas solicitações por apartamentos indisponíveis.	
RN02	Visitantes não autenticados podem realizar solicitações de apartamentos, desde que se identifiquem e determinem o objetivo.	

RN03	Usuários não-autenticados não podem realizar atividades de gestão.	
RN04	As solicitações somente podem ser excluídas pelos usuários solicitantes e pelos administradores do sistema.	
RN05	O sistema permite que mais de um apartamento seja solicitado, desde que esteja disponível.	

3.2 Principais Diagramas do Sistema

Após a especificação e análise de requisitos, as informações colhidas podem ser utilizadas para a diagramação do sistema. Esse recurso serve para apresentar aos usuários o que será desenvolvido e se está de acordo com suas necessidades. A seguir serão exibidos e detalhados os principais diagramas do SCGC.

3.2.1 Diagramas de Casos de Uso

Por meio dos diagramas de casos de uso é possível compreender os requisitos e facilitar a compreensão do comportamento do sistema, apresentando uma visão geral das funcionalidades, identificando os usuários e quais seus papéis e funções na interação com a aplicação (GUEDES, 2011).

O SCGC possui essencialmente dois tipos de usuários: os funcionários da administradora e os visitantes externos. O administrador do sistema possui total acesso e é encarregado pelo cadastramento dos funcionários (ou no mínimo a criação de um funcionário com cargo de “gerente”, que possuirá as mesmas funções), o seu acesso às funcionalidades do sistema vão de acordo com seu cargo. O Quadro 4 apresenta os atores da aplicação e uma breve descrição do que eles podem realizar no sistema.

Quadro 4 – Atores do Sistema

Ator	Descrição
Usuário não autenticado/externo	São visitantes do sistema que podem colher informações sobre a empresa e seus produtos, são capazes de navegar pela aplicação e obter essas informações e partir desse ponto, solicitar interesse aos produtos disponíveis sem a necessidade de uma autenticação.
Usuário Funcionário	São as pessoas responsáveis pelas atividades de gestão da administradora de condomínios. Suas ações no sistema dependem do cargo atribuído no momento de seu cadastro.
Usuário de Cargo Gerente	Possui as mesmas características do administrador da aplicação, obtendo acesso total ao sistema (cadastros de clientes, funcionários, condomínios, apartamentos, cargos, endereços, bem como a manutenção dos mesmos e de solicitações).

Usuário de Cargo Auxiliar Administrativo	Possui características semelhantes ao “usuário de cargo gerente”, também possuindo acesso às atividades de gestão do sistema, porém não incluem o cadastro e manutenção de cargos e funcionários.
Usuário de Cargo Operacional	São usuários que não possuem atividades de gestão na aplicação (pedreiro, eletricista, zelador), possuem acesso às mesmas funcionalidades que um usuário externo.

O diagrama de casos de uso apresentado na Figura 2 mostra as funcionalidades dos usuários-externos (que não precisam autenticar-se), acerca da aplicação. Esses usuários são representados pelos bonecos magros, as elipses representam os casos de uso e dentro delas tem-se descrita a sua funcionalidade, e as associações entre atores e casos de uso é representada por uma linha.

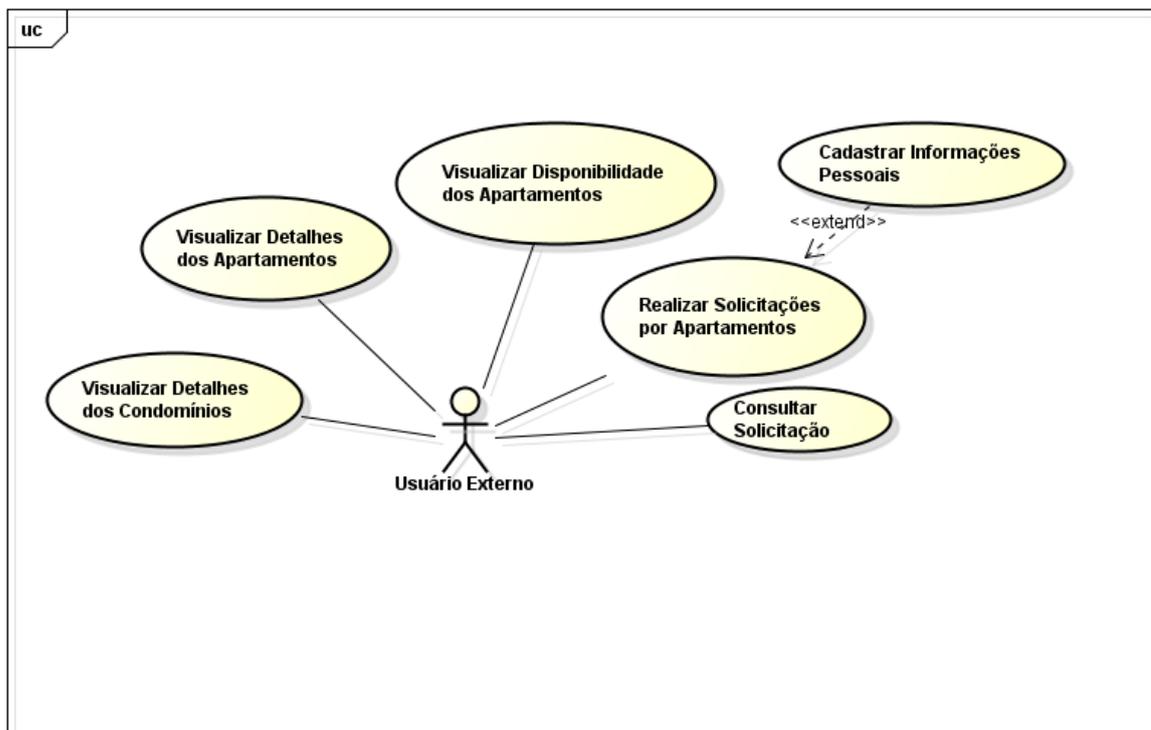


Figura 2 - Diagrama de Casos de Uso da Visão do Usuário Externo

O diagrama de casos de uso mostrado na Figura 3 apresenta a visão geral do sistema, nele é possível perceber as ações dos usuários-funcionários que precisam autenticar-se na plataforma.

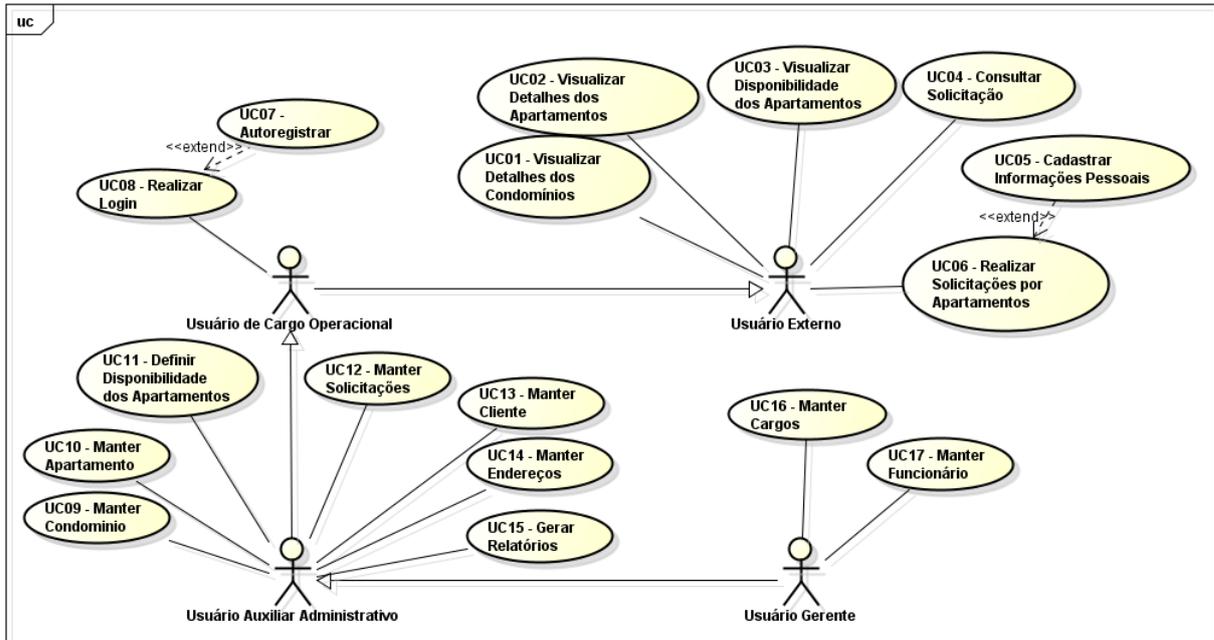


Figura 3 - Diagrama de Casos de Uso da Visão Geral do Sistema

Além do acesso a todas as funções que os usuários-externos possuem, já que os usuários-funcionários as herdam por meio da associação de generalização/especialização (determinada pelo uso de uma seta) eles também possuem permissão para realizar atividades de gestão (precisando, para isso, autenticarem-se no sistema) que estão disponíveis de acordo com seu cargo.

A seguir são exibidos mais detalhes dos casos de uso da visão geral do sistema apresentado na Figura 3, evidenciando os atores que estão associados ao caso de uso e descrevendo suas ações:

- **Usuário Externo:**

- **Casos de Uso:** UC01, UC02, UC03, UC04, UC05, UC06;
- **Descrição:** Os usuários externos são aqueles que não precisam autenticar-se na plataforma (não possuem *login* e senha de acesso), com isso eles possuem acesso limitado podendo realizar as seguintes atividades no sistema: Visualizar detalhes dos condomínios (descrição, apartamentos e outros), Visualizar detalhes dos apartamentos e sua disponibilidade (se está ou não ocupado, valor, descrição do imóvel), Realizar solicitações por apartamentos (no ato da solicitação o solicitante preenche um pequeno cadastro com informações pessoais e interesses) e Consultar solicitação (o usuário pode editar e excluir somente sua própria solicitação e para visualiza-la ele precisa informar seu cpf).

- **Usuário de Cargo Operacional:**

- **Casos de Uso:** UC07 e UC08 (além de todos os casos de uso apresentados anteriormente para o Usuário Externo, pois tem-se um relacionamento de generalização/especialização onde o usuário de cargo operacional herda as associações do usuário externo com seus casos de uso);
- **Descrição:** Os usuários de cargo operacional não realizam atividades de gestão na plataforma, porém, eles são funcionários da empresa e precisam ser gerenciados. Esses usuários tem permissões semelhantes aos externos, mas por se tratarem de funcionários, possuem *login* e senha de acesso.

- **Usuário Auxiliar Administrativo:**

- **Casos de Uso:** UC09, UC10, UC11, UC12, UC13, UC14 e UC15 (além de todos os casos de uso associados e herdados pelo usuário de nível operacional);
- **Descrição:** Esses usuários possuem papel importante na gestão da administradora de condomínios já que apenas o gerente possui maiores responsabilidades. Entre as ações disponíveis para esses usuário, temos: Manter condomínios, Manter apartamentos, Manter solicitações, Manter clientes, Manter endereços (onde o “Manter” significa a possibilidade de inclusão, edição e exclusão do referido caso de uso), Definir disponibilidade dos apartamentos e geração de relatórios.

- **Usuário Gerente:**

- **Casos de Uso:** UC16 e UC17 (inclui os casos de uso associados e herdados pelo usuário auxiliar administrativo);
- **Descrição:** O usuário Gerente possui acesso total à plataforma, sendo responsável pela manutenção dos Cargos e Funcionários além das demais ações descritas anteriormente.

O SCGC busca tornar as atividades administrativas mais simples e organizadas ao mesmo tempo que se torna mais acessível, tanto para gestores quanto para clientes.

Apresentada a visão geral do sistema por meio dos diagramas de casos de uso, foi possível perceber os atores que utilizarão o *software* e as funcionalidades que estarão disponibilizadas para os mesmos. Com base nas informações disponíveis nos diagramas de caso de uso, foi possível iniciar a construção do diagrama de classes.

3.2.2 Diagrama de Classes

Por meio do diagrama de classes é apresentada a estrutura de classes do sistema e como elas se relacionam entre si, além de definidos os métodos e atributos de cada uma. Dessa forma, tem-se uma referência conceitual sobre as informações necessárias para o desenvolvimento da aplicação.

A Figura 4 apresenta o diagrama de classes do SGCG, bem como suas classes, atributos e métodos e ainda como as mesmas se associam e trocam informações. As classes são representadas pelos retângulos que carregam seu nome na parte superior, seus atributos na parte central e seus métodos na parte inferior.

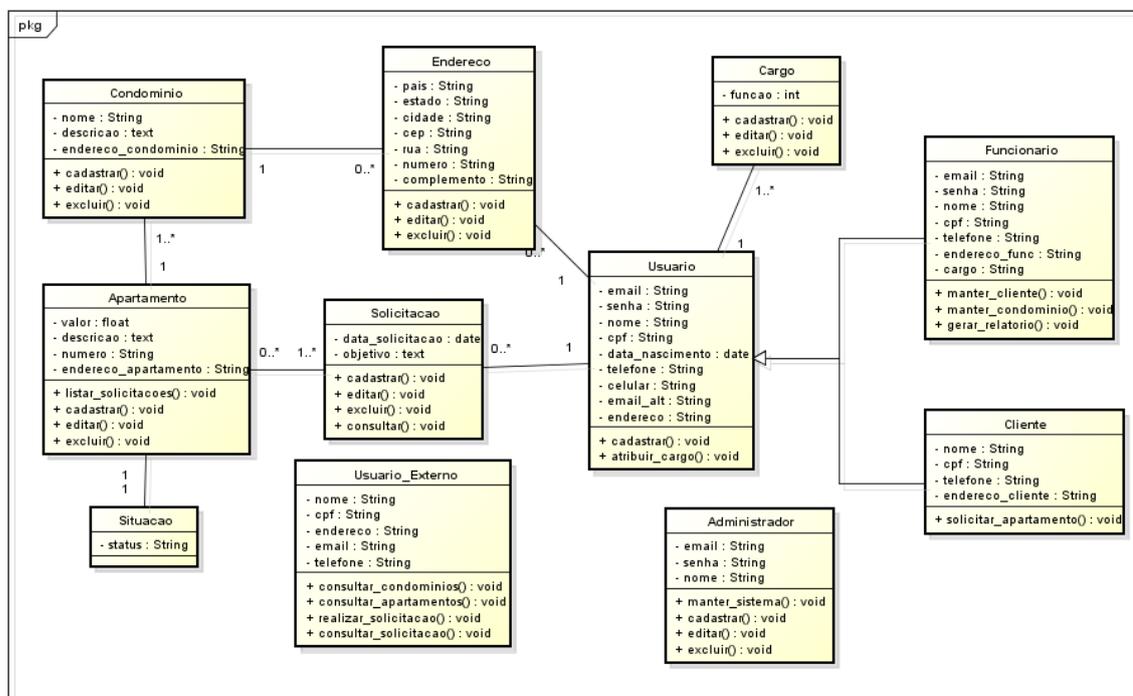


Figura 4 - Diagrama de Classes

O sistema distingue seu usuário pelo seu cargo e dessa forma controla o acesso dos mesmos à plataforma.

A classe “Usuario” manterá as informações dos usuários. Ela se relaciona com as classes “Cargo”, “Endereco” e “Solicitacao”, além de possuir uma relação especial de generalização/especialização com as classes “Funcionário” e “Cliente” que herdam suas características. Os usuários pertencentes à classe “Funcionario” possuirão um cargo e de acordo com esse cargo serão definidas as classes que poderão ser por eles gerenciadas. Esses usuários também poderão realizar solicitações por apartamentos. A classe “Apartamento”

relaciona-se com as classes “Solicitação” (já que se trata do produto a ser solicitado), “Condominio” (que representa o prédio o qual esse apartamento pertence) e “Situação” (que demonstra o *status* em que o apartamento se encontra, i. e., disponível ou indisponível). A classe “Usuario” bem como a classe “Condominio” relacionam-se com a classe “Endereco”, que carrega consigo informações de localidade.

A classe “Usuario_Externo” é apresentada sem associações, pois serve apenas para manter as informações dos usuários não-autenticados no momento de realizar as solicitações por apartamentos e a partir disso ficam disponíveis para os usuários de cargo gerencial. E a classe “Administrador” é assim também representada pois se trata de um usuário único com acesso total e responsabilidade de manter o *software* em funcionamento.

O diagrama de classes é quem mais se aproxima da forma como as informações são armazenadas no banco de dados. A Figura 5 mostra o diagrama entidade relacionamento (DER) do SCGC.

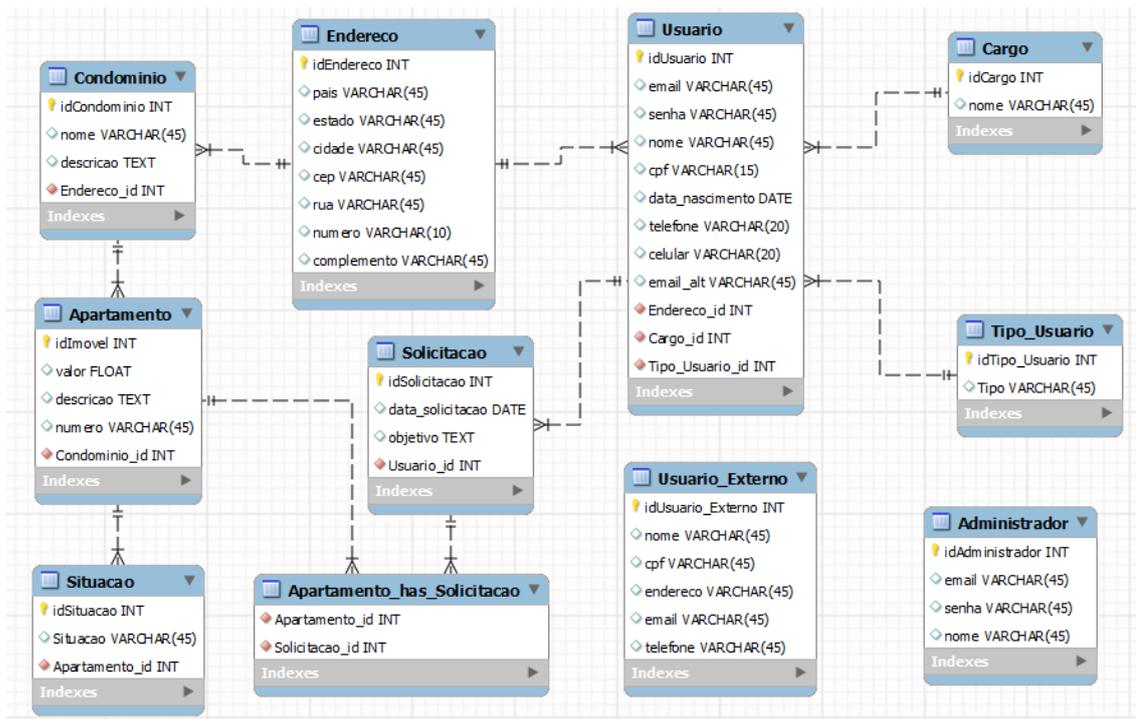


Figura 5 - Diagrama Entidade Relacionamento (DER)

Exibidos o diagrama de classes da aplicação e o diagrama entidade relacionamento, trazendo consigo a forma como as informações são inseridas no bando de dados, outros aspectos do sistema ainda podem ser expostos. A seguir o *software* será apresentado de uma forma mais dinâmica por meio do diagrama de atividades.

3.2.3 Diagrama de Atividades

O diagrama de atividades preocupa-se em demonstrar os caminhos que devem ser seguidos para se chegar à conclusão de uma determinada atividade no sistema. Tendo como base duas perspectivas (usuário externo/não-autenticado e usuário funcionário/autenticado), serão apresentados alguns diagramas de atividades que representem ações desses usuários na utilização da plataforma.

A Figura 6 mostra o diagrama de atividades de solicitação de apartamento apresentando o fluxo de ações necessárias sob a perspectiva de um usuário não-autenticado.

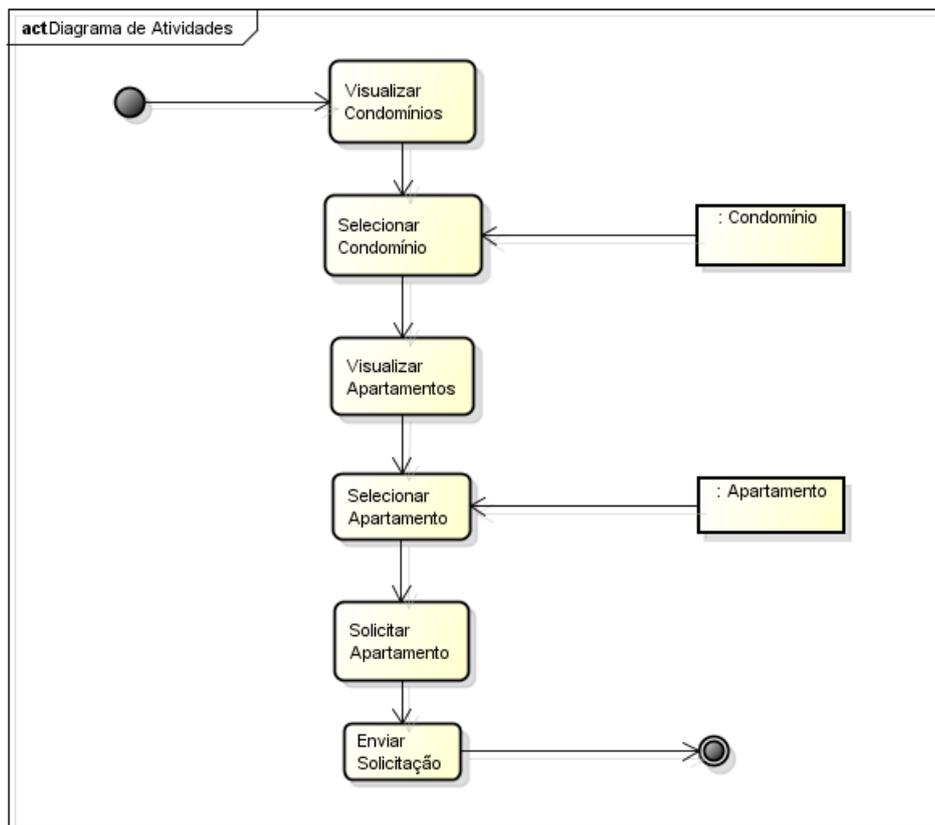


Figura 6 - Diagrama de atividades de solicitação de apartamento

Os passos necessários para solicitação de um apartamento, como o mostra o fluxo de atividades do diagrama da Figura 6, é melhor detalhado no Quadro 5.

Quadro 5 – Fluxo de solicitação de apartamentos (representação do diagrama de atividades Figura 6)

Fluxo de Solicitação de Apartamento	
Ações do Usuário	Ações do Sistema
1- O fluxo inicia com o ator na página inicial do sistema e acessa a área “condomínios”;	

	2- O sistema, então, exibe uma página com todos os condomínios cadastrados;
3- O usuário acessa a opção “detalhes do condomínio” (cada condomínio possui essa alternativa de forma individual);	
	4- O sistema direciona o usuário a uma página com todos os apartamentos do condomínio acionado;
5- O usuário acessa a opção “solicitar apartamento” (cada apartamento possui essa alternativa de forma individual);	
	6- O sistema exibe um formulário com os campos que precisam ser preenchidos para que uma solicitação seja realizada;
7- O usuário submete seu formulário devidamente preenchido;	
	8- Fluxo de solicitação de apartamento encerrado.

Exibido o diagrama de atividades com o passo a passo para a solicitação de um apartamento no sistema, a seguir serão apresentados diagramas das atividades que podem ser realizadas por usuários funcionários, i. e., que possuem responsabilidades administrativas.

A Figura 7 apresenta o diagrama de atividades de autenticação na aplicação expondo o fluxo de ações impostas para o processo de autenticação na plataforma.

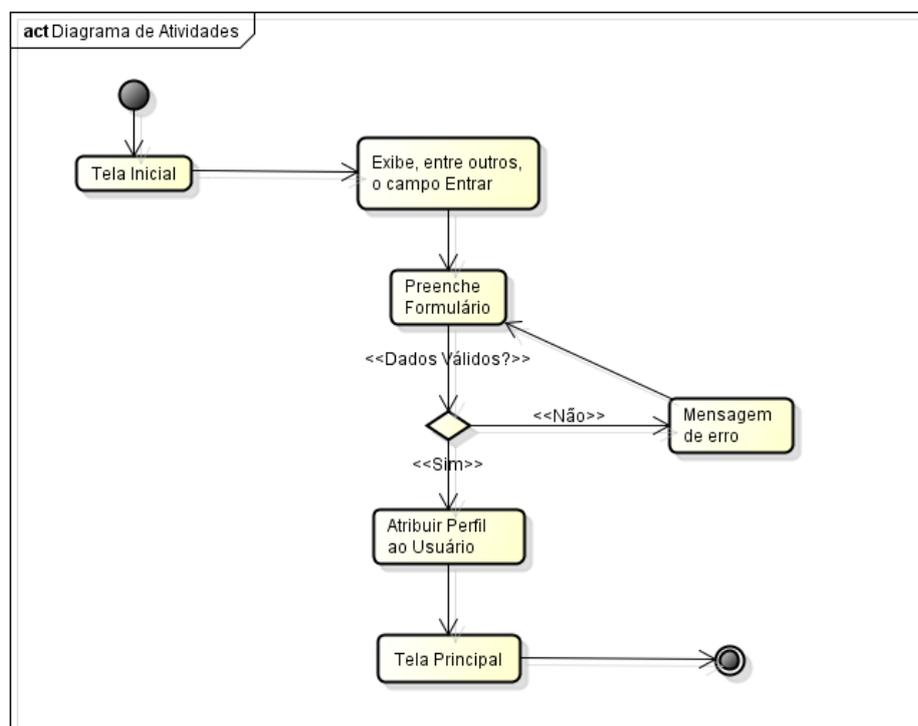


Figura 7- Diagrama de Atividades de autenticação no sistema

O Quadro 6 melhor detalha o fluxo de ações mostradas pelo diagrama de atividades da Figura 7.

Quadro 6 – Fluxo de autenticação (representação do diagrama de atividades Figura 7)

Fluxo de Autenticação	
Ações do Usuário	Ações do Sistema
1- O fluxo inicia com o ator na página inicial do sistema e acessa a área “Entrar”;	
	2- O sistema, então, exibe uma página com um formulário contendo os campos email e senha a serem preenchidos;
3- Caso o usuário preencha os campos corretamente;	
	4- O sistema atribui um perfil ao usuário e o direciona à página principal;
5- Caso o usuário preencha os campos de forma incorreta;	
	6- O sistema exibe uma mensagem de erro e mantém-se na página de autenticação;
	7- Fluxo de autenticação encerrado.

A Figura 8 mostra o diagrama de atividades de cadastro de condomínio e apartamento além de exibir as ações que precisam ser seguidas no sistema. É importante ressaltar que para o usuário seguir os passos apresentados na Figura 8, é necessário realizar as ações representadas anteriormente pelo diagrama de atividades de autenticação no sistema, como mostrado na Figura 7 e detalhadas pelo fluxo de autenticação do Quadro 6.

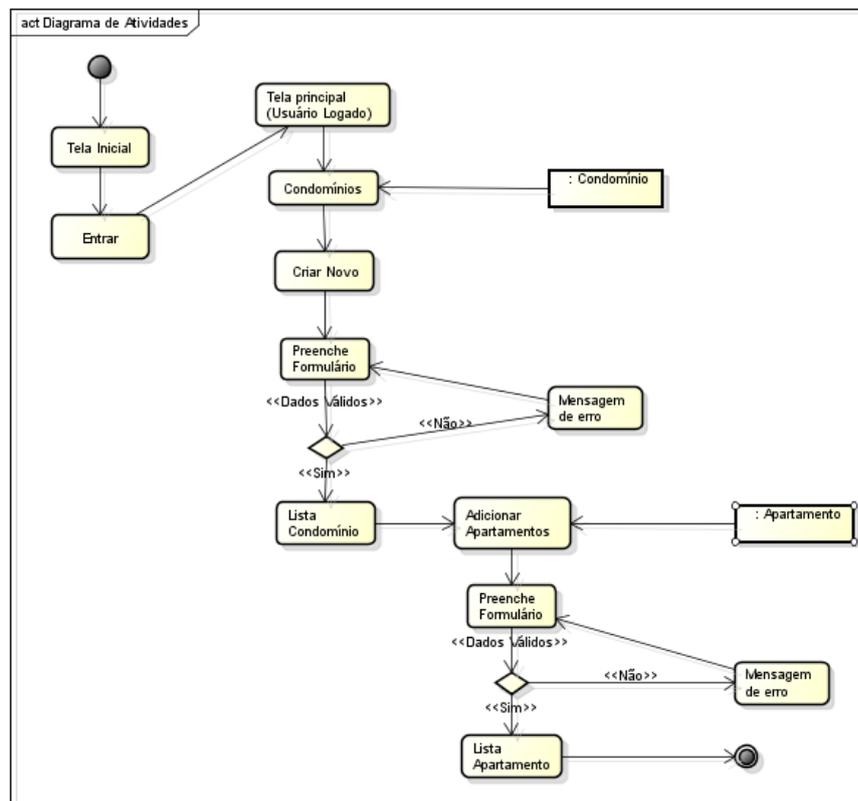


Figura 8 - Diagrama de Atividades de cadastro de condomínio e apartamento

O Quadro 7 especifica as ações realizadas pelo usuário e as reações do sistema no decorrer das atividades de cadastro de condomínio e apartamento na plataforma, como explicitado na Figura 8.

Quadro 7 – Fluxo de cadastro de condomínio e apartamento (representação do diagrama de atividades Figura 8)

Fluxo de Cadastro de Apartamento	
Ações do Usuário	Ações do Sistema
1- O fluxo inicia com o ator logado (autenticado) na página principal do usuário autenticado no sistema e acessa a área “Condomínios”;	
	2- O sistema, então, exibe uma página onde deverão ficar listados todos os condomínios cadastrados;
3- O usuário acessa a opção “criar novo”;	
	4- O sistema direciona o usuário a uma página com um formulário contendo os campos que devem ser preenchidos para a realização de um novo cadastro de condomínio;
5- O usuário submete o formulário do condomínio, por meio da opção “Salvar”;	
	6- O sistema, então, retorna para a página que lista os condomínios cadastrados (agora contendo o condomínio submetido pelo usuário);
7- O usuário agora acessa a opção “Adicionar Apartamentos” (cada condomínio cadastrado possui essa alternativa de forma individualizada);	
	8- O sistema encaminha o usuário a um formulário de cadastro de apartamentos, e nele estão os campos que precisam ser preenchidos para que um apartamento seja devidamente vinculado a um condomínio;
9- O usuário submete o formulário do apartamento, por meio da opção “Salvar”;	
	10- O sistema, dessa forma, exibe uma página onde estão listados todos os apartamentos deste condomínio (incluído o apartamento submetido pelo usuário);
	11- Fluxo de cadastro de condomínio e apartamento encerrado.

4 FUNCIONAMENTO DO SISTEMA

Esse capítulo apresenta as principais funcionalidades da aplicação. Serão demonstradas telas do *software* em uso por meio de diferentes dispositivos (*notebook*, *smartphone* e *tablet*), evidenciando a responsividade do sistema. Também serão exibidas e explicadas as atividades disponíveis sob as visões dos usuários externos e internos. É válido ressaltar que as imagens e informações dispostas nas telas são meramente ilustrativas, apenas como forma de representar o funcionamento do sistema.

4.1 Principais Interfaces

Como descrito anteriormente (seção 3.2.1) o SCGC possui sobretudo dois tipos de usuários: os externos, que seriam qualquer pessoa que acesse a plataforma sem a necessidade de autenticação (as funcionalidades disponíveis a esses usuários foram mostradas na Figura 2); e os internos (funcionários da administradora), que possuem responsabilidades de gerenciamento e precisam autenticar-se no sistema (as ações disponíveis a esses usuários foram exibidas na Figura 3). As Figuras 9 e 10 apresentam a tela inicial do sistema, bem como ele se encontra disponível para qualquer visitante em diferentes dispositivos, garantindo, desta forma, o uso da tecnologia intitulada responsividade entre equipamentos com plataformas distintas (*notebooks*, *tablets* e *smartphones*).

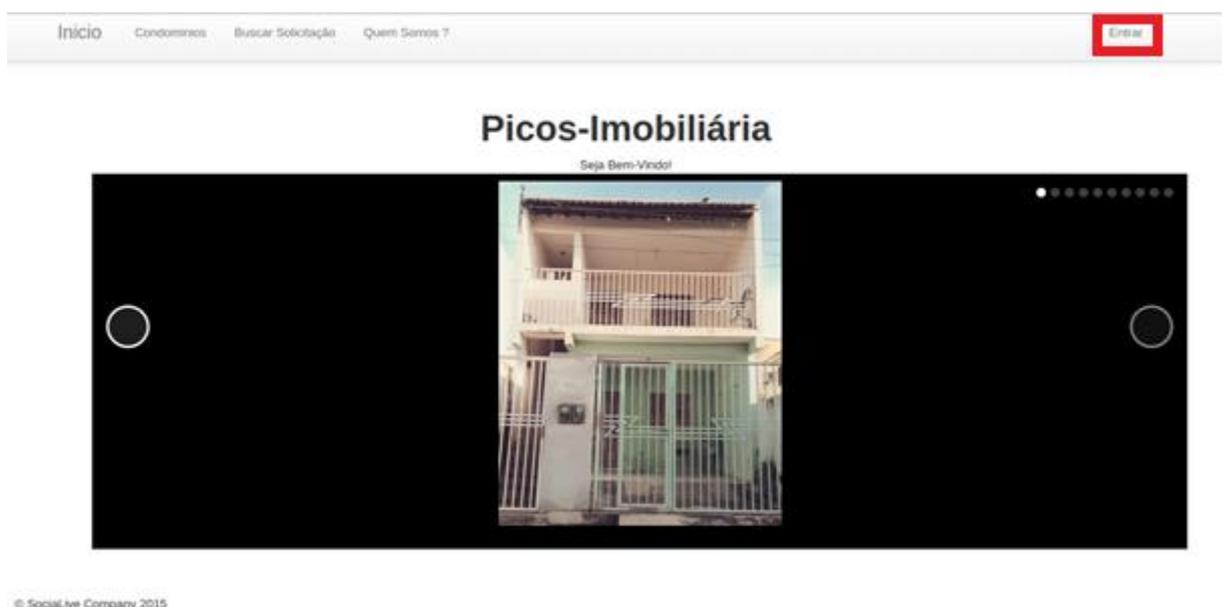


Figura 9 - Página Inicial da Aplicação acessada por *notebook*



Figura 10 - Página inicial da aplicação: (a) acessada por *smartphone*; (b) acessada por *tablet*

As Figuras 9 e 10 apresentam o mesmo método de acesso inicial à aplicação (página inicial). Nelas é possível perceber a presença de um *carousel* (ferramenta do *bootstrap* para exibição de imagens) onde são mostrados os condomínios disponíveis e gerenciados pela administradora. Nota-se na Figura 9, com o sistema sendo acessado por um *notebook*, que as funcionalidades disponíveis aos usuários ficaram visíveis e dispostas em uma barra superior. Enquanto que na Figura 10(a), com a aplicação acessada por meio de um *smartphone*, a barra superior se manteve, mas devido as dimensões do aparelho, as funções foram alocadas a um menu bastando um toque para que as mesmas possam ser vistas, como mostra a Figura 10(b), com a aplicação por meio de um *tablet*.

Além de poderem visualizar os condomínios que são disponibilizados pela administradora, assim como seus respectivos apartamentos, os usuário que estiverem visitando a plataforma podem realizar solicitações demonstrando interesse pelos apartamentos. Para isso, basta que o mesmo siga as ações detalhadas pelo Quadro 5 (Fluxo de solicitação de apartamentos) e demonstradas pelo digrama de atividades da Figura 6

(Diagrama de atividades de solicitação de apartamento). A Figura 11 apresenta o ícone (a) e o formulário de solicitação de apartamento (b).

Descrição	Numero	Valor	Situação	Opções
Apartamento com: 1 - Varanda; 3 - Quartos; 1 Cozinha; 2 - Banheiros; 1 - Sala;	01	550.0	Indisponível	
Apartamento com: 2 - Quartos; 2 - Salas; 1 - Cozinha; 1 - Dispensa; 1 - Área de serviço; 2 - Banheiros.	02	500.0	Disponível	Solicitar Apartamento
Apartamento com: 2 - Quartos; 1 - Banheiro; 1 - Cozinha; 1 - Sala; 1 - Lavanderia.	03	350.0	Disponível	

(a)

Nova Solicitação

Nome
Givanaldo

Cpf
10110210300

Telefone
999112233

Email
givanaldo@email.com

Data solicitacao
12/05/2015

Objetivo
Estou precisando de um apartamento com essas características, para dividir com mais dois amigos.

Salvar Cancelar

(b)

Figura 11 - Solicitação de apartamento. (a) ícone "Solicitar Apartamento"; (b) formulário de solicitação

Na Figura 11(a) o usuário encontra-se na página de listagem dos apartamentos e por se tratar de um acesso por pessoa não autenticada, tem-se apenas duas opções que são: “Detalhes do Apartamento” e ao seu lado “Solicitar Apartamento”. Clicando na segunda opção o usuário é direcionado à página exibida pela Figura 11(b), que disponibiliza um formulário para ser preenchido e conseqüentemente concretizar o interesse pelo imóvel.

As pessoas que realizarem solicitações podem gerencia-las bastando, para isso, a realização de uma busca por meio do seu CPF (Cadastro de Pessoa Física) e em seguida todas as suas solicitações serão listadas. A Figura 12 demonstra a tela de busca, destacando o *link* que precisa ser acessado para que a página seja exibida e em seguida, a Figura 13 apresenta o

resultado da busca, listando as solicitações retornadas e evidenciando os campos que as diferenciam.



Figura 12 - Buscar solicitação (página exibida em *tablet*)

Nome	CPF	Telefone	Email	Data da Solicitação	Objetivo	Numero apartamento	Valor	Opções
Givaldo	10110210300	999112233	givaldo@email.com	2015-06-10	Estou precisando de apartamento com essas características, para dividir com mais dois amigos.	02	500.0	
Givaldo	10110210300	999112233	givaldo@email.com	2015-05-12	Estou precisando de um apartamento com essas características, para dividir com mais dois amigos.	01	550.0	

Figura 13 - Lista de solicitações (resultado da busca exibida pela Figura 12 – página exibida em *notebook*)

Os usuários com responsabilidades administrativas possuem maiores atribuições no sistema, mas para garantirem acesso às funcionalidades que asseguram o gerenciamento da administradora de condomínios, faz-se necessário realizar autenticação na plataforma fazendo uso da opção “Entrar” que pode ser percebida na página inicial e que encontra-se em destaque

na Figura 9 (canto superior direito da tela). É importante lembrar que somente o administrador do *software* (e posteriormente um usuário de cargo “gerente”) tem acesso à página de registro, onde são criados os perfis dos funcionários, e atribuídos seu e-mail e senha de acesso. A Figura 14(a) mostra um usuário autenticando-se na aplicação e a Figura 14(b) apresenta a atribuição do perfil confirmando sua autenticação.

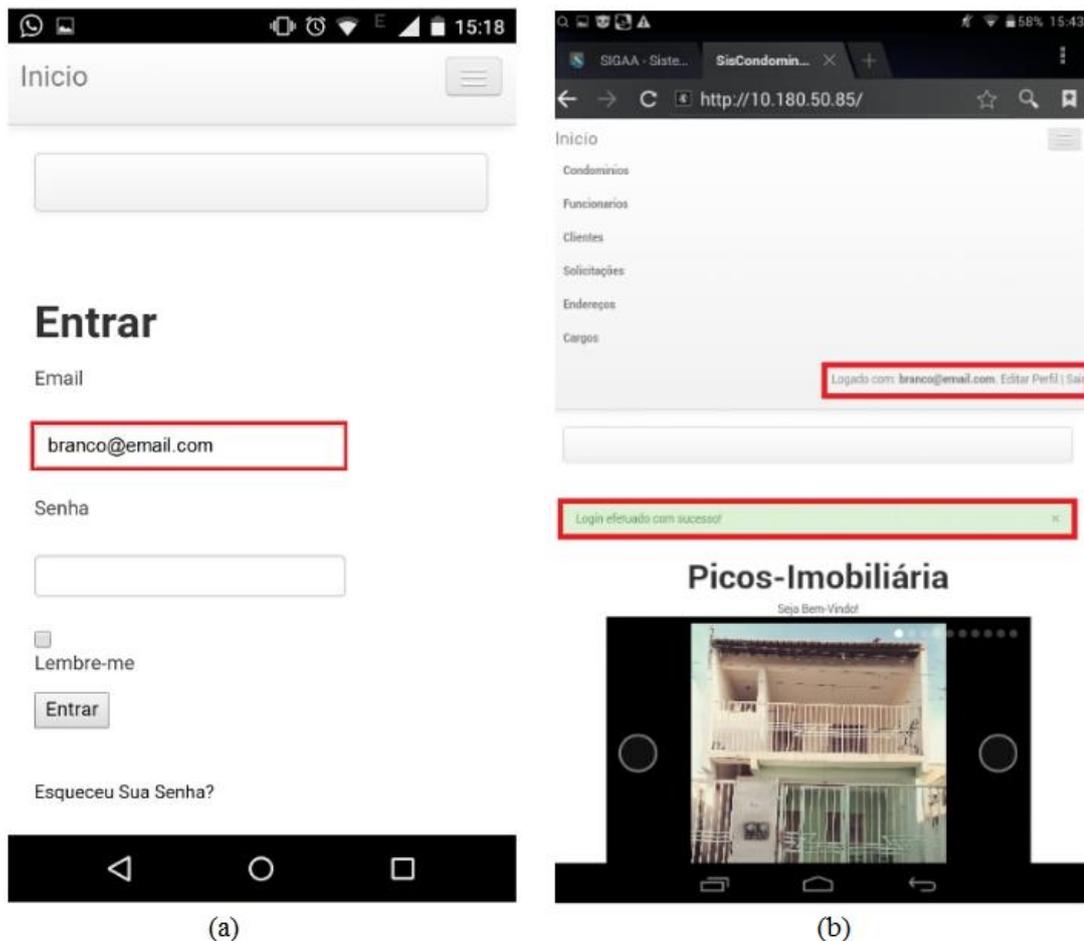


Figura 14 - Autenticação no sistema. (a) Tela de autenticação (exibida por *smartphone*); (b) Página inicial para usuário logado (exibida por *tablet*)

Nota-se pela mensagem de confirmação, pelo e-mail gravado e pelas novas funcionalidades disponíveis, que o usuário foi de fato reconhecido. Tratando-se agora de um funcionário da administradora. As ações disponíveis são: “Condomínios”, “Funcionários”, “Clientes”, “Solicitações”, “Endereços” e “Cargos”. A Figura 15 apresenta a página de condomínios sendo acessada por *smartphone* (a) e *tablet* (b). Nela os condomínios cadastrados são listados, e a frente de cada um, as opções de gerenciamento são representadas por ícones.

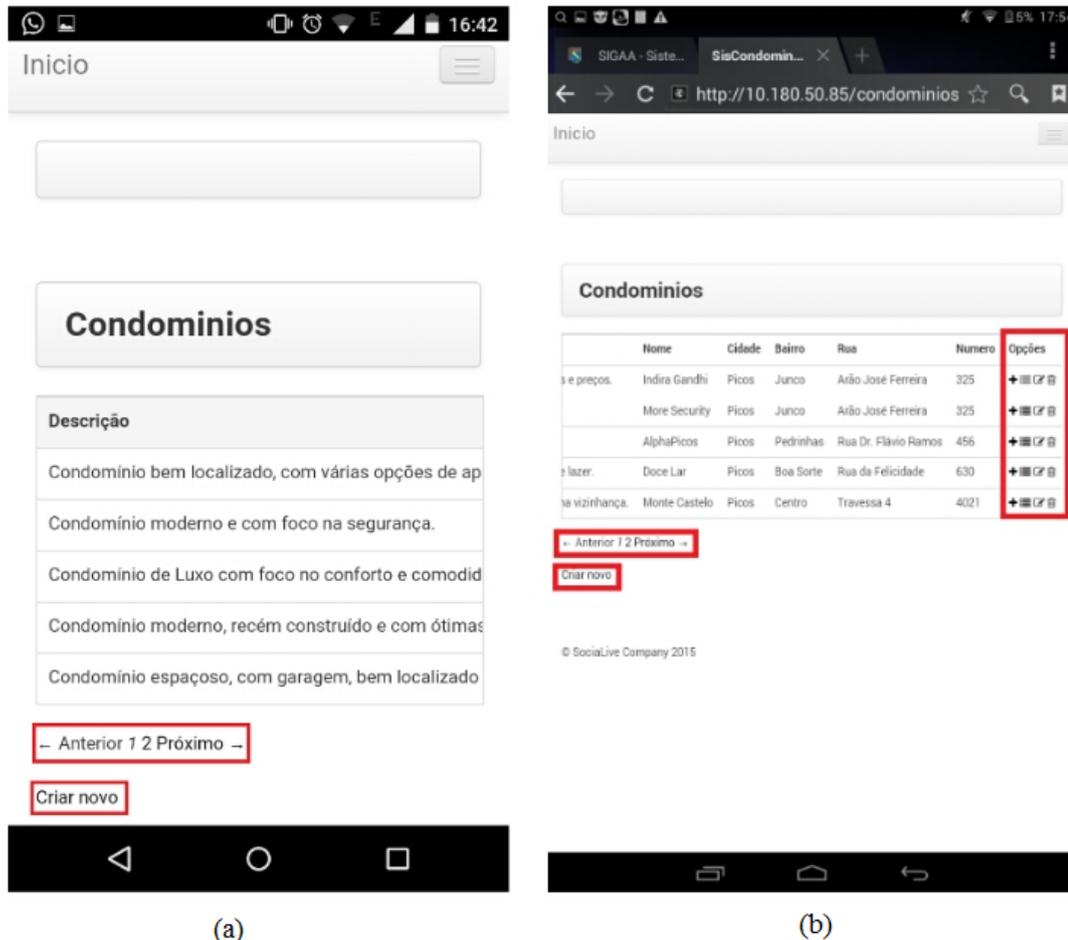


Figura 15 - Página de condomínios. (a) Exibida por *smartphone*; (b) Exibida por *tablet*

Os condomínios são listados em uma tabela, e na mesma tela tem-se a presença de um paginador (indicando que há mais de uma página) e um *link* “Criar novo”, que leva a inserção de informações sobre um novo imóvel. Percebe-se pela Figura 15(a), que apesar de estar faltando conteúdo na tabela que lista os condomínios, as informações não forçam sua total aparição na página, e deslizando o dedo sobre a tabela em um movimento da esquerda para a direita, o *software* é levado a apresentar o conteúdo da Figura 15(b). A ausência de barras horizontais e a manutenção estática do conteúdo da página, mesmo com ação do usuário sobre a tabela, indicam uma perfeita atuação do *web design* responsivo na aplicação.

Sobre as opções destacadas na Figura 15(b), elas representam as ações disponíveis e quando passado um indicador sobre elas, o nome da operação é visualizada. A cruz significa “Adicionar Apartamento”; a lista indica “Detalhes do Condomínio”; o lápis sobre o papel representa “Editar”; e a lixeira simboliza “Excluir” o item em questão.

Acionando a opção “Detalhes do Condomínio” o utilizador do sistema é direcionado a uma página contendo os apartamentos deste condomínio, como mostra a Figura 16.

Início Condomínios Funcionarios Clientes Solicitações Endereços Cargos Logado com: branco@email.com. Editar Perfil | Sair

Apartamentos

Descrição	Numero	Valor	Situação	Opções
Apartamento com: 1 - Varanda; 3 - Quartos; 1 Cozinha; 2 - Banheiros; 1 - Sala;	01	550.0	Disponível	
Apartamento com: 2 - Quartos; 2 - Salas; 1 - Cozinha; 1 - Dispensa; 1 - Área de serviço; 2 - Banheiros.	02	500.0	Disponível	
Apartamento com: 2 - Quartos; 1 - Banheiro; 1 - Cozinha; 1 - Sala; 1 - Lavanderia.	03	350.0	Disponível	
Apartamento com: 1 - Quarto; 1 - Banheiro; 1 - Cozinha; 1 - Sala.	04	250.0	Disponível	
Apartamento com: 3 - Quartos; 3 - Banheiros; 2 - Salas; 1 - Cozinha; 1 - Varanda; 1 - Lavanderia.	101	650.0	Disponível	

-- Anterior 1 2 Próximo --

© Socialive Company 2015

Figura 16 - Lista de apartamentos de um condomínio

A lista de apartamentos agora apresenta mais opções do que as exibidas na Figura 11(a), quando se tratava de um usuário externo. Além dos ícones: “Detalhes do apartamento” (mostra as características individuais do apartamento) e “Solicitar Apartamento” (exibe o formulário de solicitação do imóvel), foram acrescentados: “Adicionar Cliente” (o funcionário preenche um formulário e vincula um cliente a um apartamento); “Editar” (permite alteração nos dados do imóvel); “Excluir” (permite a exclusão de um cadastro); “Disponível” (indica que o apartamento encontra-se desocupado); e “Indisponível” (informa que o imóvel já está ocupado por um cliente). O ícone do canto inferior esquerdo, também destacado na imagem, permite que o usuário volte à página que lista os condomínios. A Figura 17 mostra um novo cliente sendo cadastrado.

Início Condomínios Funcionarios Clientes Solicitações Endereços Cargos Logado com: branco@email.com. Editar Perfil | Sair

Novo Cliente

Nome
Abimael

Email
Abimael@email.com

Cpf
12345678901

Telefone
34229188

Data nascimento
10/09/1990

Celular
999010203

Salvar Cancelar

© Socialive Company 2015

Figura 17 - Página de cadastro de cliente (notebook)

A Figura 18 apresenta a mesma tela de cadastro de cliente, porém, sendo acessada por um dispositivo de dimensões inferiores (*tablet*) como forma de evidenciar a adequação do sistema aos critérios que inclui uma aplicação utilizando conceitos de responsividade.

The image shows a mobile browser interface for a registration form titled "Novo Cliente". The browser's address bar displays "http://10.180.50.85/usuario_novo". The form consists of several input fields stacked vertically: "Nome", "Cpf", "Data nascimento" (with a placeholder "dd/mm/aaaa"), "Email" (with a placeholder "name@example.com"), "Telefone", and "Celular". At the bottom of the form are two buttons: "Salvar" and "Cancelar". Below the form, there is a copyright notice: "© SocialLive Company 2015". The browser's status bar at the top shows the time as 17:55 and 5% battery. The Android navigation bar is visible at the bottom of the screen.

Figura 18 - Tela de cadastro de cliente (*tablet*)

Observa-se que na Figura 17 o formulário de cadastro encontra-se disposto na plataforma em duas colunas verticais, ocupando o espaço disponível de maneira harmoniosa. Já na Figura 18, os campos de registro alinham-se em uma única coluna vertical como forma de adaptar-se ao espaço de visualização do novo dispositivo, neste caso o *tablet*.

O SCGC possui ainda mensagens e alertas que orientam os usuários durante a utilização da plataforma. A Figura 19 apresenta avisos de erro que norteiam o acesso ao sistema.

A Figura 19 expõem a tela de cadastro de funcionário, mas na tentativa de registrar um novo sem o preenchimento adequado das áreas indicadas, o sistema retorna uma mensagem caracterizando o problema, “Por favor, preencha todos campos!”, e alerta para a sua solução.

Início Condomínios Funcionários Clientes Solicitações Endereços Cargos Logado com: branco@email.com. Editar Perfil | Sair

Por favor, preencha todos os campos! ✕

Novo Funcionário

Nome	<input type="text"/>	Cidade	<input type="text"/>
Cpf	<input type="text"/>	Bairro	<input type="text"/>
Telefone	<input type="text"/>	Endereço	<input type="text"/>
Email	<input type="text"/>		

Cargo
admin ▼

Salvar Cancelar

Figura 19 – Página de cadastro de funcionário. Mensagens e alertas

5 AVALIAÇÃO DE USABILIDADE

Este capítulo apresenta, descreve e analisa os resultados obtidos na avaliação de usabilidade do sistema, baseado nos módulos do *ErgoList*⁶. O *Ergolist* é uma ferramenta que utiliza técnicas de inspeção da ergonomia de IHC, com o objetivo de verificar a qualidade ergonômica da interface com o usuário e o sistema inspecionado.

5.1 Análise do Autor

A análise de usabilidade foi realizada a partir dos seguintes conceitos definidos no *ErgoList*: presteza; ações mínimas; densidade informacional; controle do usuário; mensagens de erro; e compatibilidade. Os sub tópicos a seguir apresentam uma avaliação do próprio autor acerca destes conceitos em investigação ao projeto desenvolvido.

5.1.1 Presteza

Presteza significa verificar se o sistema informa e conduz o usuário durante a interação lhe poupando o aprendizado de uma série de comandos. Por exemplo, dando título às janelas ou indicando o formato adequado dos valores aceitáveis. A partir da Figura 20 a presteza da aplicação pode ser notada.

Inicio

Novo Cliente

Nome

Cpf

Data nascimento

dd/mm/aaaa

Email

name@example.com

Telefone

Celular

Salvar Cancelar

Figura 20 - Exemplos de presteza no sistema

⁶ < <http://www.labiutil.inf.ufsc.br/ergolist/index.html> >

5.1.2 Ações Mínimas

O conceito de ações mínimas leva em consideração os passos necessários para a realização dos objetivos do usuário. No SCGC usuário passa por poucas telas do sistema para realizar suas ações, p. ex., para a criação de novos registros na plataforma, dois cliques são o suficiente para levar o usuário ao formulário de cadastro. A Figura 21 demonstra as ações necessárias para acessar a página de criação de um novo item.

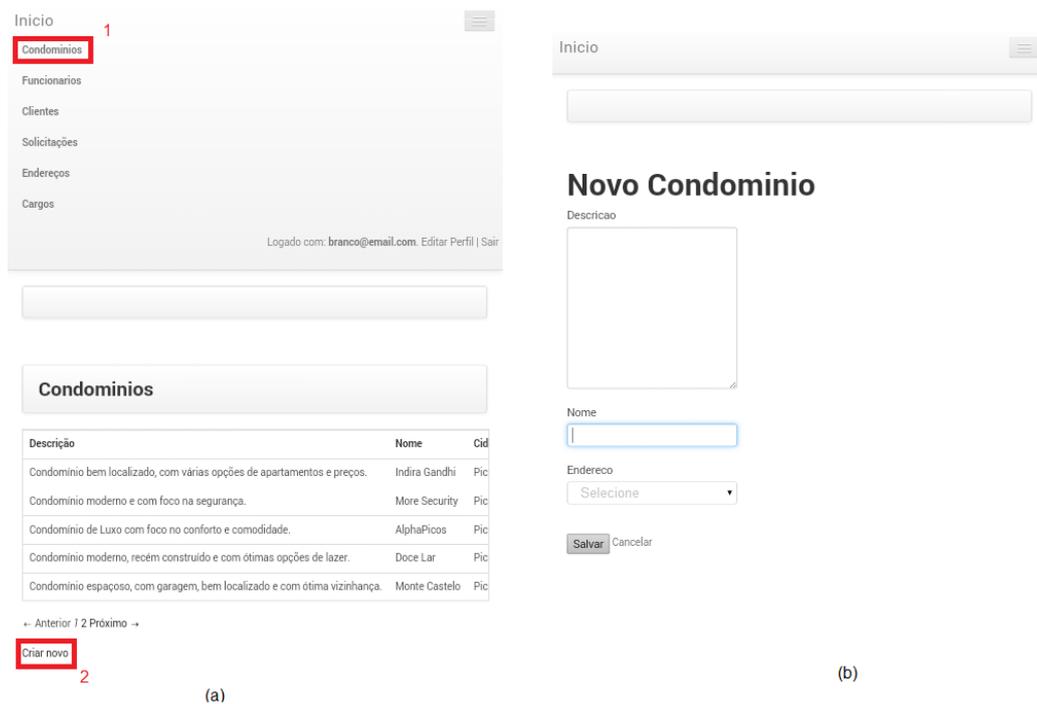


Figura 21 - Exemplo de ações mínimas no sistema (Novo Condomínio). (a) cliques necessários; (b) resultado

O exemplo de item citado na Figura 21 é o de condomínio, porém as ações são as mesmas para os demais *links* do menu. A Figura 21(a) encontra-se numerada indicando, em ordem, onde devem ocorrer os cliques para se chegar ao formulário de criação percebido na Figura 21(b).

5.1.3 Densidade Informacional

A característica densidade informacional avalia a quantidade de informações apresentadas pelo sistema. No caso do SCGC, a simplicidade da informações é priorizada contendo, basicamente, os *links* que indicam claramente para onde o usuário será direcionado.

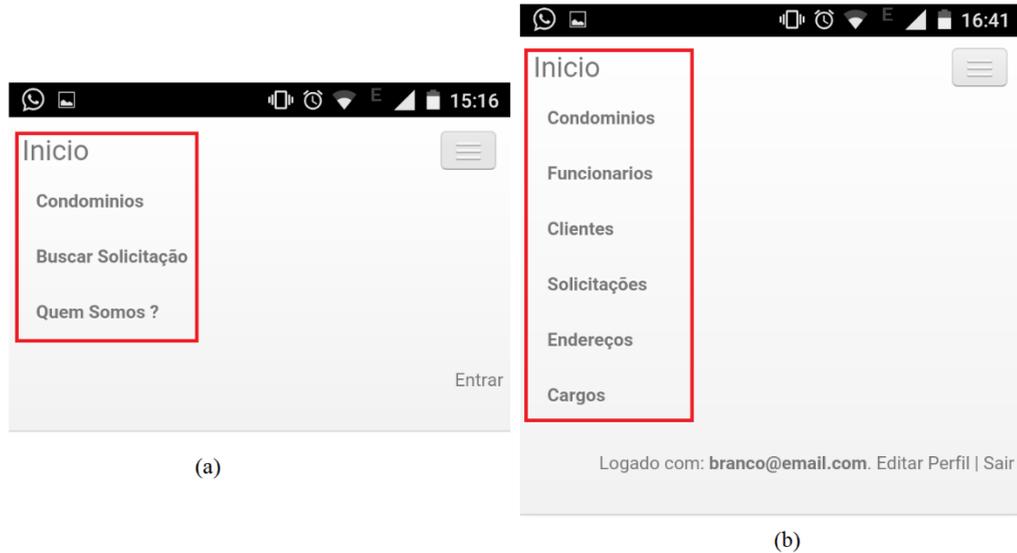


Figura 22 - Menus do sistema: (a) usuário externo; (b) usuário funcionário

5.1.4 Controle do Usuário

O controle do usuário avalia o domínio do usuário na realização de ações no sistema. Por exemplo: possibilitar ao usuário o controle do ritmo de suas entradas de dados ou cancelar o processo atual apagando qualquer mudança que tenha sido feita. O SCGC permite total controle do usuário sobre suas ações, não impondo limites de tempo de navegação ou de realização de determinada atividade na plataforma. Fornece também a opção “Cancelar” sempre que alguma tarefa específica está sendo realizada, como pode ser percebida na Figura 21(b).

5.1.5 Mensagens de Erro

O conceito de mensagens de erro avalia a qualidade das mensagens de erro e se as mesmas conduzem o usuário a realização correta da ação. O SCGC deixa claro a partir da Figura 19 que cumpre essa tarefa muito bem. Na situação lá demonstrada, o usuário tentou cadastrar um novo funcionário sem preencher todos os campos, e o sistema permaneceu na mesma página indicando o erro e como resolve-lo.

5.1.6 Compatibilidade

O último conceito avaliado, mas não menos importante, é o de compatibilidade. Esta característica analisa a compatibilidade do sistema com as expectativas e necessidades do usuário na realização de suas tarefas. O SCGC foi testado em uso de diferentes plataformas e

em nenhuma delas apresentou seu conteúdo de forma anormal ou teve seu funcionamento prejudicado, como pode ser percebido no capítulo 4 desta pesquisa.

5.2 Análise dos Usuários

Com o objetivo de oferecer maior credibilidade à avaliação de usabilidade do sistema desenvolvido, foi coletada uma amostra onde 25 pessoas que utilizaram o SCGC, responderam a um questionário (APÊNDICE A) e puderam compartilhar suas experiências na utilização da plataforma, gerando os resultados obtidos. Foram elaboradas seis perguntas, sendo que quatro delas pediam resposta em escala variando de 1 à 5 (onde a melhor e a pior resposta variam de acordo com o contexto da questão) e as outras duas de múltipla escolha. As questões elaboradas seguiram os mesmos conceitos determinados no *ErgoList* e apresentados anteriormente no diagnóstico do autor (seção 5.1).

A presteza do sistema foi a mais questionada pelos usuários, onde a maioria (44%) ficou indecisa quanto a orientação do *software* na realização das atividades. A Figura 23, apresentada a seguir, exibe a pergunta abordando o conceito de presteza e os resultados obtidos a partir da análise e respostas dos usuários.

O sistema lhe orienta na realização de suas atividades na plataforma?

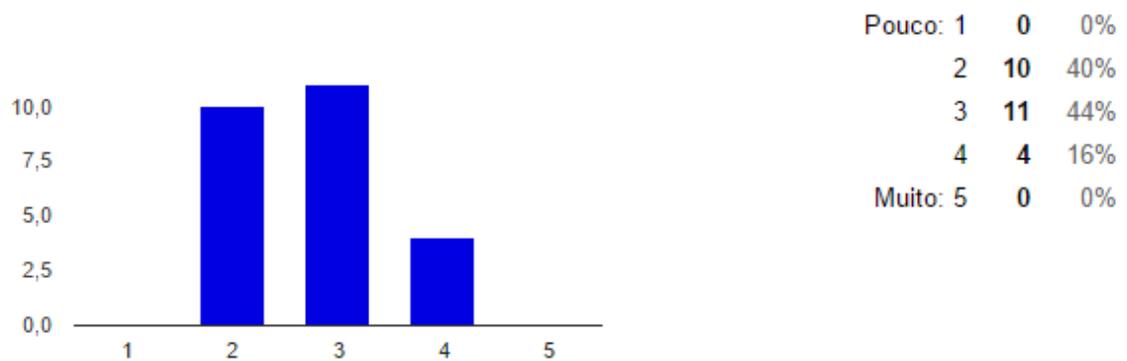


Figura 23 - Análise de usabilidade (presteza)

Quanto as ações mínimas, a maior parte dos entrevistados afirmou perder pouco tempo (72%) ou próximo disso (24%) para encontrar na plataforma, onde solucionar seus problemas. A Figura 24 mostra a conclusão dos responsáveis pelo diagnóstico acerca do concepção de ações mínimas.

Você considera que perdeu muito tempo navegando na plataforma até encontrar a página de solução de sua atividade?

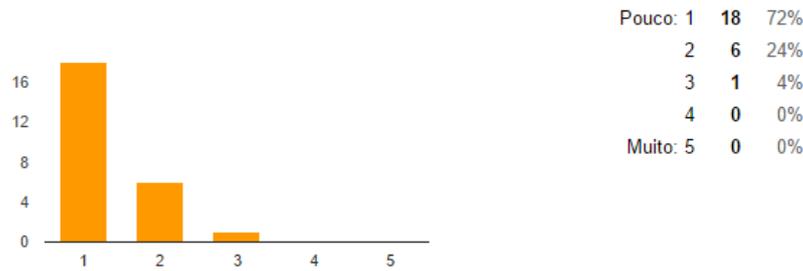


Figura 24 - Análise de usabilidade (ações mínimas)

Em relação a densidade informacional, que avaliou a quantidade de informações no sistema, o *feedback* foi positivo, com 60% das pessoas afirmando utilizar o *software* normalmente e 32% assegurando haver pouca informação, mas que não atrapalhou. A Figura 25 exibe o diagnóstico dos entrevistados em referência à densidade informacional.

A aplicação apresenta muita ou pouca informação, de forma que lhe confunda ou não lhe oriente, durante a navegação acarretando em erros?

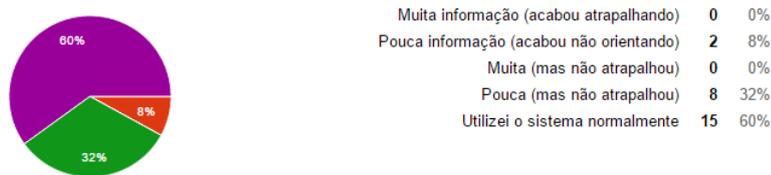


Figura 25 - Análise de usabilidade (densidade informacional)

Com referência ao controle do usuário, foi quase unânime (96%) a afirmação de autonomia garantida pela aplicação na realização das atividades. A Figura 26 evidencia a opinião dos entrevistados em relação a definição de controle do usuário no sistema.

O sistema lhe dá autonomia na realização das atividades e durante a navegação?

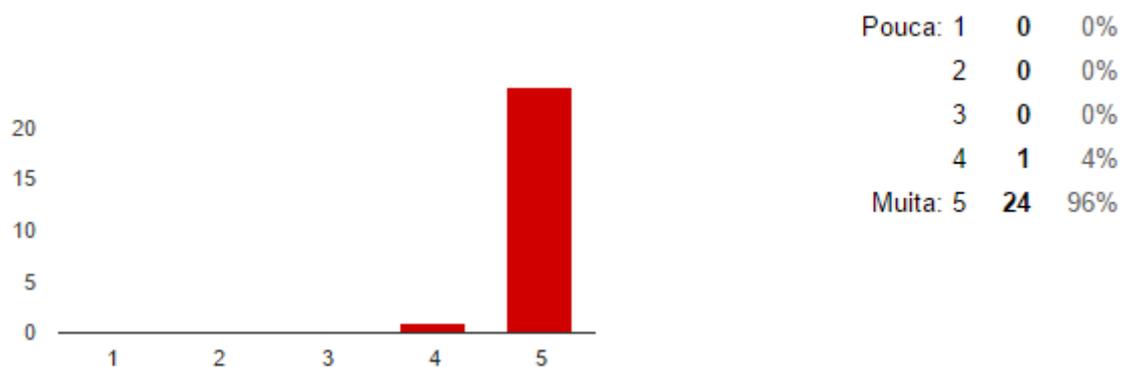


Figura 26 - Análise de usabilidade (controle do usuário)

Sobre as mensagens de erro, 80% da amostra assegurou sua presença e orientação durante os testes. A Figura 26 mostra os resultados abondando o conceito de mensagens de erro.

O sistema apresenta mensagens de erros (quando necessário) e que orientem a forma correta da ação desejada?



Figura 27 - Análise de usabilidade (mensagens de erro)

E por fim, a respeito da compatibilidade, que analisou a utilização do sistema por meio de outros dispositivos (*smartphone* e *tablet*), a grande maioria garantiu que obteve bom (56%) ou ótimo (28%) desempenho durante o acesso através de outro aparelho. A Figura 28 apresenta os resultados da verificação da compatibilidade da aplicação.

Na utilização do sistema por meio de outros dispositivos (smartphone; tablet), como foi seu desempenho?

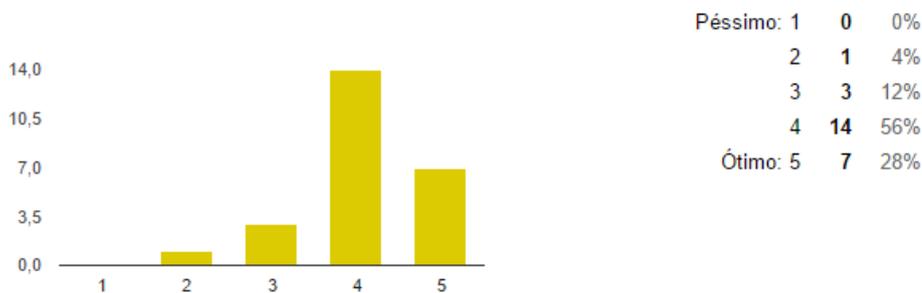


Figura 28 - Análise de usabilidade (compatibilidade)

Após a inspeção do questionário e a observação dos gráficos gerados a partir das respostas dos entrevistados, os resultados adquiridos que avaliaram a usabilidade do *software* desenvolvido foram considerados satisfatórios.

Por fim, o *ErgoList* dispõe de uma série de questionários e listas de verificações que permitem ao desenvolvedor analisar o *software* produzido sobre diferentes aspectos. As listas de verificação que formam o *ErgoList* destinam-se a apoiar na inspeção da interface de uma aplicação de maneira a levar a equipe de produção a descobrir falhas ergonômicas mais flagrantes em uma interface com o usuário e, desta forma, corrigi-las e ou melhorá-las.

6 CONSIDERAÇÕES FINAIS

Um sistema de informação automatizado é capaz de agregar inúmeros benefícios para uma organização. O trabalho descrito foi desenvolvido com o propósito de apresentar um sistema *web* de controle de gestão de condomínios para a empresa Picos-Imobiliária. Como foi desenvolvido para plataforma *web*, a aplicação possibilita seu uso em qualquer local que possua acesso à *Internet* dando, desta forma, maior poder de interação com o usuário final. Outra característica desse *software* é a utilização da tecnologia do *web design* responsivo que permite sua utilização por meio de qualquer dispositivo sem prejudicar as ações do usuário.

No decorrer da produção do *software*, foi realizado um estudo do problema, analisando o funcionamento das atividades de gerenciamento e identificadas as possíveis formas de otimização desses processos. Em seguida, foram detalhadas as tecnologias e métodos de desenvolvimento utilizadas, além de apresentadas as principais funcionalidades do sistema em utilização em diferentes dispositivos e, por fim, a análise de usabilidade da aplicação.

Finalmente, como trabalhos futuros sugere-se realizar testes da aplicação em ambiente real e inserir novas funcionalidades, tais como: a utilização de técnicas de *real time* para mudar a forma de comunicação e troca de informações do servidor do sistema com as máquinas clientes; a criação de um módulo exclusivo aos clientes, onde eles terão acesso a um *Chat* de comunicação com funcionários e poderão gerar boletos de pagamento dos seus respectivos imóveis; a implementação da função “Estado da reserva”, com as solicitações de apartamentos sendo avaliadas e podendo passar por três situações: “Aprovada” (quando os termos da reserva forem aceitos); “Reprovada” (quando os termos forem negados), neste caso será necessário uma justificativa; e “Em processamento” (quando a mesma ainda encontrar-se em processo de avaliação); e ainda a ação de envio de e-mails, para que os usuários solicitantes possam acompanhar o estado das suas reservas.

REFERÊNCIAS

- BALDUINO, Plínio. **Dominando JavaScript com jQuery**. São Paulo: Casa do Código, 2012.
- BARBOSA, Simone Diniz Junqueira; SILVA, Bruno Santana. **Interação Humano-Computador**. Rio de Janeiro: ELSEVIER, 2010.
- BOOCH, Grady; JACOBSON, Ivar; RUMBAUGH, James. **UML: Guia Do Usuário**. 2ª Edição. Rio de Janeiro: ELSEVIER, 2005.
- COCHRAN, David. **Twitter Bootstrap Web Developmente How-To: A hands-on introduction to building websites with Twitter bootstrap's powerful front-end development framework**. Birmingham: Packt Publishing, 2012.
- FUENTES, Vinícius B. **Ruby on Rails: Coloque sua aplicação web nos trilhos**. São Paulo: Casa do Código, 2012.
- GUEDES, Gilleanes T. A. **UML 2 Uma abordagem prática**. 2ª Edição. São Paulo: Novatec, 2011.
- HARTL, Michael. **Ruby on Rails Tutorial: Learn Web Development with Rails**. 2ª Edição. Michigan: Addison-Wesley, 2012.
- LABIUTIL.INF; **ErgoList** Disponível em: <<http://www.labiutil.inf.ufsc.br/ergolist/index.html>> Acesso em: 10 junho. 2015.
- LAUDON, Kenneth C; LAUDON, Jane P. **Sistemas de Informação Gerenciais Administrando a empresa digital**. 5ª Edição. São Paulo: Pearson Prentice Hall, 2004.
- MAGNO, Alexandre. **Mobile First Bootstrap: Develop advanced websites optimized for mobile devices using the Mobile First feature of Bootstrap**. Birmingham: Packt Publishing, 2013.
- MILANI, André. **MySQL: Guia do Programador**. Novatec, 2007. Disponível em: <<http://www.novateceditora.com.br/livros/mysqlcompleto/capitulo8575221035.pdf>> Acesso em: 06 maio 2015.
- NETTO, Alvim. **Interação humano computador. IHC**. Modelagem e gerência de interfaces com o usuário. Florianópolis: VisualBooks, 2006.
- PÁDUA, Wilson de. **Engenharia de Software: fundamentos, métodos e padrões**. [S.l]: LTC, 2000.
- PRESSMAN, Roger S. **Engenharia de Software: uma abordagem profissional**. 7ª Edição. Porto Alegre: AMGH, 2011.
- PUGA, Sandra; FRANÇA, Edson; GOYA, Milton. **Banco de Dados: Implementação em SQL, PL\SQL e Oracle 11g**. São Paulo: Pearson Education - Br, 2013.

RAILSGUIDES; **Ruby on rails guides: getting started with rails**. Disponível em < http://guides.rubyonrails.org/getting_started.html >. Acesso em: 28 abr. 2015.

RUBY-LANG.ORG. About Ruby. **Ruby - A Programmer's Best Friend**, abril 2015. Disponível em: < <http://www.ruby-lang.org/pt/about/> >. Acesso em: 26 abr. 2015.

REZENDE, Denis Alcides. **Sistemas de Informações Organizacionais**: Guia prático para projetos. 2ª Edição. São Paulo: Atlas S.A, 2007.

RODRIGUES, Andréa. **Desenvolvimento para Internet** Curitiba: LT, 2010.

SABBAGH, Rafael. **SCRUM**: Gestão ágil para projetos de sucesso. São Paulo: Casa do Código, 2013.

SILVA, Maurício Samy. **Web Design Responsivo**. São Paulo: Novatec, 2014.

SOMMERVILLE, Ian. **Engenharia de Software**. 9ª Edição. São Paulo: Pearson Prentice Hall, 2011.

SOUZA, Lucas. **Ruby**: Aprenda a programar na linguagem mais divertida. São Paulo: Casa do Código, 2012.

TIOBE.COM. **Índice TIOBE para abril de 2015**. Disponível em: < <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html> >. Acesso em: 26 abr. 2015.

TORRES, Joaquim. **Guia da Startup**: Como startups e empresas estabelecidas podem criar produtos web rentáveis. São Paulo: Casa do Código, 2012.

ZEMEL, Tércio. **Web Design Responsivo**: Páginas adaptáveis para todos os dispositivos. São Paulo: Casa do Código, 2012.

APÊNDICES

APÊNDICE A – Formulário utilizado na amostra de usabilidade do sistema

A Figura 29 apresenta as perguntas e opções de respostas utilizadas na coleta de dados que foram diagnosticados e levaram aos resultados obtidos e exibidos no item 5.2 Usabilidade de Usuários.

Análise de Usabilidade

Os respondentes analisaram o sistema sob as duas perspectivas (externo/funcionário) e foram colocados em situações que pudessem avaliar as perguntas.

O sistema lhe orienta na realização de suas atividades na plataforma?

Ex: Dá nome às páginas atuais; indica o formato adequado dos valores.

1 2 3 4 5

Pouco Muito

Você considera que perdeu muito tempo navegando na plataforma até encontrar a página de solução de sua atividade?

Analise se o sistema lhe obrigou a passar por várias telas até chegar ao local desejado.

1 2 3 4 5

Pouco Muito

A aplicação apresenta muita ou pouca informação, de forma que lhe confunda ou não lhe oriente, durante a navegação acarretando em erros?

- Muita informação (acabou atrapalhando)
- Pouca informação (acabou não orientando)
- Muita (mas não atrapalhou)
- Pouca (mas não atrapalhou)
- Utilizei o sistema normalmente

O sistema lhe dá autonomia na realização das atividades e durante a navegação?

Ex: impõe limite de tempo; ou permite cancelar determinada ação depois de iniciada.

1 2 3 4 5

Pouca Muita

O sistema apresenta mensagens de erros (quando necessário) e que orientem a forma correta da ação desejada?

- Sim
- Não
- As vezes

Na utilização do sistema por meio de outros dispositivos (smartphone; tablet), como foi seu desempenho?

1 2 3 4 5

Péssimo Ótimo

Figura 29 - Questionário da análise de usabilidade



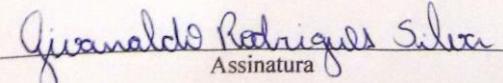
**TERMO DE AUTORIZAÇÃO PARA PUBLICAÇÃO DIGITAL NA BIBLIOTECA
“JOSÉ ALBANO DE MACEDO”**

Identificação do Tipo de Documento

- () Tese
- () Dissertação
- (X) Monografia
- () Artigo

Eu, **Givaldo Rodrigues Silva**, autorizo com base na Lei Federal nº 9.610 de 19 de Fevereiro de 1998 e na Lei nº 10.973 de 02 de dezembro de 2004, a biblioteca da Universidade Federal do Piauí a divulgar, gratuitamente, sem ressarcimento de direitos autorais, o texto integral da publicação **Desenvolvimento de um Sistema Web para controle e gestão de condomínios da Picos-Imobiliária utilizando princípios de responsividade** de minha autoria, em formato PDF, para fins de leitura e/ou impressão, pela internet a título de divulgação da produção científica gerada pela Universidade.

Picos-PI 30 de Outubro de 2015.


Assinatura