

**INSTITUTO FEDERAL DO SERTÃO PERNAMBUCANO
CURSO DE GESTÃO EM TECNOLOGIA DA INFORMAÇÃO**

ANTÔNIO JERRY SÁ SANTOS

**UMA ANÁLISE DO DESENVOLVIMENTO WEB ATRAVÉS DE FRAMEWORKS:
DESENVOLVIMENTO DO SISTEMA *IFACADÊMICO***

**FLORESTA
2013**

ANTÔNIO JERRY SÁ SANTOS

**UMA ANÁLISE NO DESENVOLVIMENTO WEB ATRAVÉS DE FRAMEWORKS:
DESENVOLVIMENTO DO SISTEMA *IFACADÊMICO***

Trabalho de Conclusão de Curso apresentado para obtenção do título de Graduado no Curso Tecnólogo em Gestão de Tecnologia da Informação, do Instituto Federal do Sertão Pernambucano / Campus Floresta – PE.

Orientador: Prof. MSc. Cassiano Henrique de Albuquerque

FLORESTA

2013

FICHA CATALOGRÁFICA

Santos, Antônio Jerry Sá.

Uma análise do desenvolvimento web através de frameworks: desenvolvimento do IFACADÊMICO. / Antônio Jerry Sá Santos . – Floresta, 2013.
134 p. il.

Monografia (Graduação) – Instituto Federal de Educação, Ciência e Tecnologia do Sertão Pernambucano – Campus Floresta. Gestão da Tecnologia da Informação.

Orientador: Prof^o. Msc. Cassiano Henrique de Albuquerque.

1. Desenvolvimento web. 2. Frameworks. 3. Engenharia de software. I.
Título.

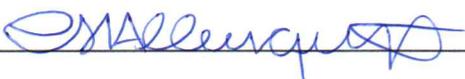
CDD 005.3

ANTÔNIO JERRY SÁ SANTOS

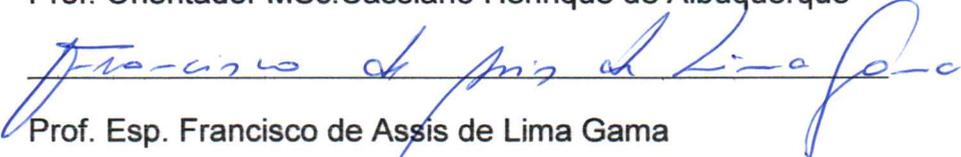
**UMA ANÁLISE DO DESENVOLVIMENTO WEB ATRAVÉS DE FRAMEWORKS:
DESENVOLVIMENTO DO SISTEMA IFACADÊMICO**

Trabalho de Conclusão de Curso – TCC – apresentado para obtenção do título de Graduado no Curso Tecnólogo em Gestão de Tecnologia da Informação, do Instituto Federal do Sertão Pernambucano / Campus Floresta - PE

Banca Examinadora

Presidente: _____ 

Prof. Orientador MSc. Cassiano Henrique de Albuquerque

Membro: _____ 

Prof. Esp. Francisco de Assis de Lima Gama

Membro: _____ 

Prof. Breno Leonardo Gomes de Menezes Araújo

FLORESTA, 30 de julho de 2013

Dedico este trabalho a minha mãe por simplesmente ser a minha fonte de inspiração e ao meu pai por me motivar e acreditar no meu potencial.

Agradeço a DEUS pela motivação e força que me proporciona em todos os momentos da minha vida. A minha família pelo apoio e o incentivo durante todos os anos do curso. Aos meus amigos da turma GTI.2009.1. A todos os professores que contribuíram repassando conhecimentos, pois sem estes verdadeiros mestres não seria possível concretizar este momento. Ao professor Lincoln Tavares dos Santos pelas contribuições intangíveis para realização desta idéia. Ao professor orientador Cassiano Henrique de Albuquerque pela idéia inicial, pelo incentivo, paciência e cooperação.

RESUMO

Aplicações na web são uma realidade há muitos anos, e adotar ferramentas ou bibliotecas para o desenvolvimento de tais softwares tem-se tornado, cada vez mais frequente, devido aos benefícios e facilidades que tais tecnologias oferecem. Este trabalho apresenta a experiência obtida da utilização, observação e análise do jQuery UI Framework, como suporte para concepção de aplicações web, através da sua aplicação no desenvolvimento de um software acadêmico de gerenciamento de coordenações de curso. Este é um trabalho direcionado a interessados em estudos sobre a tecnologia jQuery, bem como discentes, docentes e demais que tenham desejo em conhecer detalhes sobre a realização da pesquisa que resultou na criação do sistema *IFACADÊMICO*.

PALAVRAS-CHAVES: Desenvolvimento Web, Framework, Engenharia de software.

ABSTRACT

Web applications are a reality for many years and adopt tools or libraries for the development of such software has become increasingly common due to the benefits and facilities that these technologies offer. This research presents the experience of use, observation and analysis of the jQuery UI Framework as support for designing web applications, through its application in developing an academic software management coordination of courses. This work is aimed at interested in studies on jQuery technology, as well as students, teachers and others who have a desire to know details about the research that resulted in the creation of the IFACADÊMICO system.

KEYWORDS: Web Development, Framework, Software Engineering.

LISTA DE FIGURAS

Figura 1 -	Esquema de utilização de um framework de software.....	28
Figura 2 -	Exemplo de frameworks para desenvolvimento web.....	30
Figura 3 -	Logotipo da biblioteca javascript jQuery.....	32
Figura 4 -	John Resig, criador do projeto jQuery.....	32
Figura 5 -	Comparativo de interesse de busca entre "DHTML" e "jQuery" em produtos da Google Inc	34
Figura 6 -	Comparativo de interesse de busca entre "AJAX" e "jQuery" em produtos da Google Inc	34
Figura 7 -	Resultados da pesquisa realizada pelo <i>W3Techs</i>	35
Figura 8 -	Logotipo do framework web jQuery UI.....	36
Figura 9 -	Exemplo de interface gráfica do sistema IFACADÊMICO: Interface do gerenciamento de professores do módulo coordenação.....	39
Figura 10 -	Ciclo de desenvolvimento do sistema IFACADÊMICO.....	41
Figura 11 -	Tecnologias utilizadas para a concepção do sistema IFACADÊMICO	42
Figura 12 -	Interface do aplicativo <i>Theme Roller</i>	49
Figura 13 -	Exemplo de utilização do widget <i>Dialog</i> : Interface para alterar os dados do aluno.....	50
Figura 14 -	Exemplo de marcações HTML para instanciar o widget <i>Dialog</i>	51
Figura 15 -	Exemplo de código jQuery para instanciar o widget <i>Dialog</i>	51
Figura 16 -	Exemplo de código jQuery para manipulação de elementos através	52

do widget <i>Dialog</i>	
Figura 17 - Exemplo de utilização do widget <i>Tabs</i> : Interface para gerenciar alunos.....	53
Figura 18 - Exemplo de marcações HTML para instanciar o widget <i>Tabs</i>	53
Figura 19 - Exemplo de código jQuery para instanciar o widget <i>Tabs</i>	54
Figura 20 - Exemplo de utilização do widget <i>Acoordion</i> : Menu sobre informações do sistema <i>IFACADÊMICO</i>	54
Figura 21 - Exemplo de marcações HTML para instanciar o widget <i>Acoordion</i> ..	55
Figura 22 - Exemplo de código jQuery para instanciar o widget <i>Acoordion</i>	56
Figura 23 - Exemplo de utilização do widget <i>Datapiecker</i> : Campo "data" da funcionalidade para gerar relatórios.....	56
Figura 24 - Exemplo de marcação HTML para instanciar o widget <i>Datapiecker</i>	57
Figura 25 - Exemplo de código jQuery para instanciar o widget <i>Datapiecker</i>	57
Figura 26 - Desenvolvimento de sistemas através da reutilização de componentes.	57
Figura 27 - Interface do aplicativo <i>jQuery Plugin Registry</i>	58
Figura 28 - Interface de edição do memorando virtual.....	60
Figura 29 - Código jQuery para implementar o <i>CKEditor</i>	60
Figura 30 - Interface da funcionalidade para enviar arquivos.....	62
Figura 31 - Interface da funcionalidade para cadastrar turmas.....	64
Figura 32 - Código jQuery para implementar o cadastro de turmas.....	65

Figura 33 - Interface da funcionalidade para visualizar os anexos enviados.....	66
Figura 34 - Código jQuery para implementar a exclusão do anexo.....	66
Figura 35 - Exemplo de atualização assíncrona de dados: Interface da funcionalidade para realizar a troca de mensagens no <i>IFACADÊMICO</i>	67
Figura 36 - Código jQuery para implementar a verificação do status dos participantes da funcionalidade para realizar conversas.....	69
Figura 37 - Exemplo de atualização assíncrona de dados: Fragmento " <i>Dados Acadêmicos</i> " da funcionalidade para cadastrar alunos.....	70
Figura 38 - Código jQuery para implementar a busca de turmas cadastradas....	71
Figura 39 - Interface da funcionalidade para selecionar destinatários.....	72
Figura 40 - Código jQuery para implementar a pesquisa do destinatário.....	73
Figura 41 - Interface da funcionalidade para criar horários acadêmicos.....	74
Figura 42 - Código jQuery para implementar a busca de início e término de uma determinada aula.....	75
Figura 43 - Código jQuery para implementar a busca de professores relacionados a uma determinada disciplina escolhida no horário acadêmico.....	76
Figura 44 - Código jQuery para implementar o cadastro de uma determinada aula na criação do horário acadêmico.....	77
Figura 45 - Fragmento " <i>Dados de Identificação</i> " do cadastro de alunos: Verificação do <i>CPF</i>	78
Figura 46 - Exemplo de código jQuery para implementar a verificação de dados: Verificação de <i>CPF</i>	79
Figura 47 - Interface do chat de mensagens rápidas.....	80

Figura 48 - Criação personalizada do editor no chat de mensagens rápidas.....	81
Figura 49 - Código jQuery para implementar o envio de mensagens do chat.....	81
Figura 50 - Código jQuery para atualizar o chat.....	82

SUMÁRIO

1. INTRODUÇÃO.....	15
1.1. TEMA E PROBLEMA.....	16
1.2. OBJETIVO GERAL.....	17
1.3. OBJETIVOS ESPECÍFICOS.....	17
1.4. JUSTIFICATIVA.....	18
2. FUNDAMENTAÇÃO TEÓRICA.....	20
2.1. O DESENVOLVIMENTO DE SOFTWARE VOLTADO PARA O AMBIENTE EDUCACIONAL.....	20
2.1.1. A Importância da Comunicação Virtual entre Docentes e Discentes no Processo de Coordenação, Ensino e Aprendizagem.....	20
2.1.2. Ponderações para o Sucesso de um Ambiente Virtual de Compartilhamento de Informações.....	23
2.1.2.1. <i>Adequação ao Futuro Ambiente de Execução</i>	23
2.1.2.2. <i>Segurança Reforçada</i>	24
2.1.2.3. <i>Usabilidade</i>	25
2.2. CONCEITOS GERAIS A CERCA DE UM FRAMEWORK.....	27
2.2.1. Expectativas e Motivação para a Utilização de Frameworks.....	29
2.3. ENTENDENDO A TECNOLOGIA JQUERY.....	32
2.3.1. Visão Geral do jQuery UI Framework.....	35
2.3.2. Desenvolvimento Subsidiado pela Utilização de Plugins.....	37
3. DESENVOLVIMENTO DO SISTEMA IFACADÊMICO.....	39
3.1. ANÁLISE E PROJETO.....	40
3.2. IMPLEMENTAÇÃO.....	41

4. METODOLOGIA.....	43
4.1. DELINEAMENTO DA PESQUISA.....	43
4.2. PROCEDIMENTO DE COLETA E ANÁLISE DOS DADOS.....	44
4.3. O OBJETO DE ESTUDO.....	46
5. ANÁLISE DO OBJETO DE ESTUDO.....	47
5.1. ANÁLISE DA UTILIZAÇÃO DO JQUERY UI FRAMEWORK NO SISTEMA IFACADÊMICO.....	47
5.1.1. Implementação da Interface com o Usuário.....	48
5.1.1.1. Theme Roller.....	48
5.1.1.2. Widgets.....	52
5.1.1.2.1. Dialog.....	52
5.1.1.2.2. Tabs.....	52
5.1.1.2.3. Acoordion.....	54
5.1.1.2.4. Datapiecker.....	56
5.1.2. Reutilização de Software.....	57
5.1.2.1. Desenvolvimento jQuery Através de Plug-ins.....	58
5.1.2.1.1. CKEditor Adapter.....	59
5.1.2.1.2. jQuery File Upload.....	61
5.1.3. Dinâmica de Interação Cliente-Servidor.....	63
5.1.3.1. Envio Assíncrono de Dados.....	63
5.1.3.2. Atualização Assíncrona de Dados.....	67
5.2. ANÁLISE DA ADEQUAÇÃO DA TECNOLOGIA À IMPLEMENTAÇÃO DE FUNCIONALIDADES COMPLEXAS DO SISTEMA IFACADÊMICO.....	71
5.2.1. Selecionar Destinatário.....	71
5.2.2. Criar Horário Acadêmico.....	74

5.2.3. Verificar a Disponibilidade de Dados.....	77
5.2.4. Chat de Mensagens Rápidas.....	79
6. CONCLUSÃO.....	83
6.1. TRABALHOS FUTUROS.....	85
REFERÊNCIAS.....	87
GLOSSÁRIO.....	91
APÊNDICES.....	93

1. INTRODUÇÃO

Há muitos anos a web deixou de ser apenas um repositório de informações e páginas estáticas. A evolução do usuário na internet, através das mudanças sociais, das novas formas de acesso e da modernização das características computacionais, bem como, das plataformas e tecnologias envolvidas no processo de criação de aplicações direcionadas à web, modificaram determinantemente as tradicionais premissas existentes sobre o desenvolvimento de tais softwares, onde a criação de aplicações dependia fortemente de tecnologias nativas, despendia considerável esforço e tempo e possibilitava apenas mínimos recursos de interface, limitados a links, textos, imagens e formulários.

Deitel (2007) afirma que apesar do crescimento tecnológico da internet, a usabilidade de aplicações web desenvolvidas sob arquiteturas tradicionais, desprovidas de recursos sofisticados de interface e baseadas na relação cliente-servidor, tornou-se obsoleta. Pois, a cada interação do usuário na aplicação, resulta em um período de espera, onde: todo o processamento é realizado no servidor; e, ao final do processamento, a página web é apenas apresentada estaticamente.

Além disso, desenvolver um produto de software direcionado à web exige muito mais que um bom projeto e práticas de implementação bem organizadas. Para Tacin e Zorzo (2012), é indiscutível que a utilização de técnicas da Engenharia de Software traz grandes benefícios para o sucesso de uma aplicação web. Isso se dá porque a aplicação de tais conceitos favorece, dentre outros aspectos, um bom funcionamento técnico da aplicação, assim como, fornece um nível maior de manutenibilidade. No entanto, num contexto globalizado e baseado em conceitos cada vez mais sofisticados e dinâmicos, no que diz respeito à navegabilidade e usabilidade, apenas boas práticas de desenvolvimento de softwares para web não são suficientes, pois, à medida que a tecnologia web foi crescendo, o usuário tornou-se cada vez mais exigente frente às aplicações na internet.

Paralelamente a este cenário, um dos grandes desafios em desenvolver aplicações web está em projetar, em tempo hábil, sistemas flexíveis adaptáveis de

acordo com necessidades e preferências estabelecidas. Promovendo assim, aumentos de eficiência nos referidos sistemas de informações, e proporcionando ótimos níveis de padronização e compatibilidade frente às inúmeras formas de acesso disponíveis aos usuários atuais.

Muito se tem comentado a cerca desta nova realidade, e neste contexto, surgem inúmeras abordagens para o desenvolvimento destas aplicações, porém, uma delas se destaca predominantemente entre as mais importantes, bem como, faz-se presente na maioria das aplicações web existentes hoje: o desenvolvimento através de frameworks.

A tecnologia de frameworks possibilita que uma família de produtos de software seja gerada a partir de uma única estrutura que captura os conceitos mais gerais necessários a estas aplicações (JOHNSON *apud* MATTSSON, p. 45, 2000). O desenvolvimento de aplicações web utilizando frameworks se expandiu de tal forma que tornou-se imprescindível o uso de tais ferramentas para a construção de sistemas de pequeno ou grande porte (SOUZA, 2010).

Segundo Gimenes e Huzita (2005), um dos principais objetivos da Engenharia de Software é o reuso. Através da reutilização de software, além de obter-se mais qualidade, também é possível reduzir e padronizar o esforço de desenvolvimento. Olhando deste ponto, é exatamente isso que um framework deve oferecer através da sua utilização, e pensando nisso, este trabalho pretende descrever situações, características, vantagens e desvantagens do uso de um framework em uma aplicação web compacta e dinâmica. Nesta pesquisa será possível analisar na prática a participação da tecnologia de frameworks sobre o desenvolvimento web, estabelecendo, por tanto, elos entre os resultados obtidos em campo e esta abordagem de desenvolvimento.

1.1. Tema e problema

A utilização de frameworks para desenvolver aplicações web é o alvo desta pesquisa porque é uma abordagem que vem ganhando grande popularidade frente

ao desenvolvimento web nos últimos anos. Segundo Souza (2010), a utilização de frameworks web em larga escala representa uma tendência real para o desenvolvimento de aplicações voltadas a internet, pois, através dos seus resultados, vem sendo comprovada como uma solução consideravelmente viável às necessidades existentes.

A decisão de escolher o *jQuery UI Framework* como ferramenta chave para realização deste trabalho, se deu através de uma comparação prévia diante de outros frameworks semelhantes. Nesta comparação, os motivos para definir o framework a ser utilizado na pesquisa levaram em consideração características como: linguagem de programação melhor adaptável as demais tecnologias; interface com o usuário; dependência de outras tecnologias ou plataformas; liberdade de alteração e utilização da ferramenta; capacidade de integrar novos componentes ao projeto; e desempenho satisfatório no resultado final de software.

1.2. Objetivo geral

Observar e analisar a participação da tecnologia de framework no desenvolvimento de software a partir do *jQuery UI Framework*, levando em consideração a implementação de interfaces e serviços web;

1.3. Objetivos específicos

- Observar a tecnologia de framework em atuação a partir do desenvolvimento de um sistema web voltado para o ambiente educacional;
- Analisar o uso do framework, como tecnologia para o desenvolvimento web atual, quanto à implementação e usabilidade de interfaces, recursos e serviços;

- Estabelecer relações de causa e consequência entre os resultados de software obtidos e a abordagem de desenvolvimento utilizada;
- Expor resultados provenientes do desenvolvimento realizado em campo;
- Avaliar a viabilidade da utilização de frameworks para o desenvolvimento de aplicações web.

1.4. Justificativa

Este trabalho aborda um exemplo do desenvolvimento de aplicações web através de framework. Nos últimos anos, muito se tem visto quanto aos benefícios existentes da aplicação de frameworks no desenvolvimento web, bem como, também, este tipo de implementação se tornou extremamente popular por todo mundo. De maneira geral, esta tecnologia atende aos diversos tipos de aplicações e traz uma proposta de desenvolvimento diferente dos padrões comuns existentes no cenário web atual.

Com relação a essa realidade, a questão central levantada aqui é que à medida que este tipo de desenvolvimento se torna comum, nem sempre se tem conceitos bem definidos de como estas tecnologias atuam na prática. Faz-se pertinente então, ter o mínimo de conhecimento sobre as características, a viabilidade e o potencial de tais ferramentas.

Além disso, observando a realidade atual das coordenações de curso do *IF Sertão-PE Campus Floresta*, a busca por soluções que auxiliem a produção e gerenciamento de informações é pertinente, já que a comunicação entre docentes e discentes do campus não foi considerada satisfatória pela coordenação entrevistada durante a realização desta pesquisa; e, além do mais, o cenário encontrado atualmente nas coordenações da instituição não dispõe de nenhum canal para comunicação virtual.

Assim, tendo como foco central a percepção real da ferramenta, movido pelo intuito de promover melhorias contínuas através da aplicação de novas tecnologias e motivado na busca de benefícios para a comunidade de docentes e discentes do Campus Floresta do *IF Sertão-PE*, este trabalho vem apresentar uma análise da utilização do *jQuery UI Framework*, como base para o desenvolvimento de aplicações direcionadas a web, através do desenvolvimento do Sistema de Gerenciamento de Coordenações de Curso *IFACADÊMICO*.

O propósito da realização desta pesquisa através do desenvolvimento e implementação de um sistema desta natureza é, em primeiro plano, observar, investigar e elucidar como a tecnologia pode ser útil para o desenvolvimento de aplicações web atualmente; e, promover um ambiente virtual de cooperação em que os seus próprios beneficiários possam comunicar-se frequentemente sem a necessidade de estarem presentes no mesmo tempo ou lugar. E, em segundo plano, fornecer conhecimentos práticos que possam ser úteis para a realização de trabalhos futuros relacionados, bem como, também, incentivar o uso da tecnologia de frameworks entre desenvolvedores interessados, para que possam dar vida a projetos brilhantes através desta tecnologia.

2. FUNDAMENTAÇÃO TEÓRICA

Este capítulo irá demonstrar conceitos importantes, dados estatísticos, considerações e ponderações extremamente relevantes na realização desta pesquisa. Estes conceitos serviram como base principal para que o desenvolvimento desta pesquisa não fosse concebido sob conceitos precipitados, fora de contexto ou alheios aos seus reais objetivos. Através das próximas seções deste capítulo serão descritas informações sobre a aplicação de tecnologias no ambiente escolar, bem como definições, panoramas e análises sobre as metodologias e ferramentas necessárias para a concretização desta pesquisa.

2.1. O desenvolvimento de software voltado para o ambiente educacional

A busca pela solução ideal de um software, independente de qual software seja, sempre exige muitos conhecimentos prévios a cerca do ambiente de atuação, todos os envolvidos na utilização, o comportamento de cada um e suas contribuições, as restrições e políticas de segurança necessárias, sempre é o melhor subsídio para que o desenvolvimento de um software atenda criteriosamente seu destino final: um produto de software em conformidade com as expectativas existentes. Para isso, esta seção deste capítulo irá descrever informações relevantes sobre a dinâmica da comunicação e colaboração entre os indivíduos, da construção do conhecimento junto ao coletivo e do compartilhamento das informações existentes. Assuntos defendidos e analisados sobre o ponto de vista de estudiosos da dinâmica de formação do conhecimento na escola, com ou sem o uso da tecnologia da informação.

2.1.1. A importância da comunicação virtual entre docentes e discentes no processo de coordenação, ensino e aprendizagem

De acordo com Magdalena e Costa, p. 49, 2003:

Só atingimos o pleno desenvolvimento da inteligência e a construção de novos conhecimentos em situações de confronto, de descentralização de nosso ponto de vista, de intensa cooperação, de momentos em que precisamos nos colocar em xeque pela desestruturação e reestruturação de argumentos e de posições que são auxiliados pelas questões colocadas por outros.

A comunicação entre indivíduos de uma determinada comunidade é a principal premissa para que os objetivos traçados coletivamente sejam alcançados. Curvello (2012) descreve a comunicação interna como item estratégico responsável pela construção de um universo simbólico que aproxima e integra os públicos aos princípios e objetivos centrais de uma determinada organização. E, através deste tipo de comunicação é possível conhecer e compreender a cultura de um ambiente, além de captar a lógica das relações internas, suas contradições, suas mediações, para melhor compreender os estágios administrativos, os sucessos e fracassos e as facilidades ou dificuldades impostas às mudanças.

No meio acadêmico, semelhante ao cenário apresentado, a interação entre indivíduos deve ser um costume tão frequente e importante quanto às próprias aulas, pelos seguintes motivos:

- um aluno que pouco interage com o ambiente, perde muito no que diz respeito à exposição das suas dificuldades, ideias e necessidades.
- segundo, um professor que pouco interage com o ambiente em que atua, em nada pode ajudar no esclarecimento de dúvidas dos alunos fora do horário normal de aula, no incentivo ao cumprimento das atividades propostas e na contribuição adicional à formação de futuros profissionais comprometidos em exercer suas funções no melhor nível possível.
- e, entre os dois cenários, uma coordenação que não possui canais de comunicação eficientes nunca estará apta a assessorar alunos e professores na melhoria do processo de ensino e aprendizagem, a incentivar o uso de melhores práticas pedagógicas, a obter e compartilhar informações frequentemente, a solucionar os problemas que por ventura venha a ocorrer.

A falta de comunicação nos dias de hoje se assemelha, em grande parte, à utilização de tecnologias obsoletas, onde por melhor que se tente realizar algo bem feito, jamais será possível conseguir. Atualmente, graças ao advento das novas tecnologias, uma alternativa para amenizar os problemas de comunicação é a comunicação virtual, um meio, que ganhou grande aceitação entre os diversos seguimentos da sociedade moderna.

Para Lévy (1996) o processo de virtualização afeta não apenas a informação e a comunicação, mas também as pessoas, o funcionamento, os quadros coletivos da sensibilidade e até o exercício da inteligência. Ele atinge mesmo as modalidades do estar junto, a constituição do “nós”: comunidades virtuais, empresas virtuais, democracia virtual, etc. Moran (1997) defende que as redes atraem os estudantes porque eles gostam de navegar, de descobrir coisas novas, de divulgar suas descobertas, de comunicar-se com outros colegas. Moran também afirma que:

- a educação presencial se modifica significativamente com as redes eletrônicas;
- as paredes das escolas e das universidades se abrem, as pessoas se intercomunicam, trocam informações e dados;
- a educação é otimizada pela possibilidade de integração de várias mídias, acessando-as tanto em tempo real como assincronicamente, isto é, no horário favorável a cada indivíduo, e também pela facilidade de pôr em contato educadores e educandos.

Aliada à comunicação presencial, a comunicação virtual visa possibilitar ou complementar a interação entre os seus envolvidos em situações onde, teoricamente, os mesmos estejam separados em tempos ou espaços diferentes. Trazendo, assim, versatilidade, flexibilidade e liberdade à construção e veiculação das informações e à criação de uma permanente relação de troca do indivíduo com o ambiente.

2.1.2. Ponderações para o sucesso de um ambiente virtual de compartilhamento de informações

Para Valentini (2003) o sucesso do ambiente virtual acadêmico está ligado ao contexto pedagógico do seu uso: o cenário pedagógico, o grau de envolvimento do professor, a dimensão de tempo, a infra-estrutura técnica, etc. Uma das vantagens dos ambientes virtuais de aprendizagem é que eles dão suporte a interações sociais de múltiplas formas: síncrona ou assíncrona, baseadas em texto, áudio ou vídeo, de um para um ou de um para muitos. Essas possibilidades definem os efeitos potenciais da interação e da aprendizagem em ambientes virtuais.

2.1.2.1. Adequação ao futuro ambiente de execução

A adequação de um sistema à sua proposta de desenvolvimento é uma das primordiais características que mais influenciam no resultado final de um produto de software. Conforme Burd (1999), todo desenvolvimento tecnológico deve buscar o melhor equilíbrio entre o problema que se pretende resolver e a sua solução. E, nos vários âmbitos em que a adequação de um sistema pode estar, as principais características são interface, eficiência e viabilidade.

Em relação à interface, Burd (1999) descreve que o sistema deve trazer consigo um modelo visual simples, explicativo e intuitivo. Na questão funcional atrelada à interface cada informação visual oferecida pelo sistema deverá ser retratada fielmente ao usuário. A proposta de interface está intimamente ligada aos níveis de produtividade que cada usuário pode atingir no sistema e deve ser a melhor dentre as quais forem possíveis através dos recursos disponíveis.

. Segundo Burd (1999), a eficiência viria da facilidade de uso da tecnologia, das formas com que ajuda na implantação e acompanhamento da dinâmica educacional, na sua integração com o resto do ambiente, entre outros. No que se refere à eficiência, um sistema eficiente deve atender criteriosamente cada premissa necessária ao desempenho das funções propostas. Não adianta se pensar em uma solução que seja esteticamente atraente, mas que não atenda ao objetivo desejado.

Por último, tendo em vista a viabilidade, a análise de uma tecnologia para a educação deve considerar os gastos de desenvolvimento e, além deles, dedicar uma atenção especial aos custos de implantação e manutenção. Para Burd (1999), é importante frisar que a não consideração dos elementos sócio-histórico-culturais envolvidos na implantação e na manutenção pode condenar uma tecnologia, por mais eficaz ou eficiente que ela pareça.

Mesmo assim, Burd (1999) afirma que não existe nenhuma tecnologia educacional que mereça o adjetivo “mais educacional” ou que possa ser considerada “melhor” do que qualquer outra, sem que se faça a consideração de um contexto maior. Cada caso é um caso e a tecnologia mais eficiente será aquela que melhor suprir as necessidades da situação educacional que está sendo criada.

2.1.2.2. Segurança reforçada

Segurança é um item muito importante no desenvolvimento de softwares. Segundo Sommerville (2007), a segurança é um atributo que reflete a capacidade do sistema proteger-se de ataques externos acidentais ou propositais. Mello *et al* (2006) descreve que a segurança é vista como uma qualidade que garante o fornecimento do serviço, mesmo diante de ações de indivíduos não autorizados no sistema, sem que ocorram violações de segurança.

A confiança de um sistema está intimamente ligada à segurança que ele possui sobre as informações e os usuários que o utilizam. Nos dias de hoje ter um sistema vulnerável a acessos indevidos pode custar não só prejuízos na integridade do próprio software, como também, a exposição de sua privacidade.

Em sistemas computacionais as ameaças são constantes, e uma maneira de evitá-las é identificar e corrigir as vulnerabilidades que por ventura possam existir. Certamente, um sistema seguro deve ser confiável, sem falhas e em conformidade com sua especificação.

Atualmente ataques podem vir de qualquer lugar. Sommerville (2007) explica que “a importância da proteção cresceu, à medida que mais e mais sistemas foram

conectados à internet". Como explicado por Sommerville, a exposição mundial das informações de um sistema pode ser algo muito preocupante se as políticas de seguranças necessárias ao sistema não forem devidamente implementadas. Um sistema provedor de serviço com essa possibilidade de acesso, além da preocupação de ter que gerenciar inúmeras identidades e credenciais, deve preocupar-se, a cima de tudo, com a política de segurança que o rege rigorosamente.

As políticas definem pontos importantes a se considerar durante a execução de um sistema por partes dos usuários. Conforme Mello *et al* (2006) a política de segurança de um sistema é um conjunto de diretrizes, normas e procedimentos, os quais estabelecem os limites de operação dos principais envolvidos. Ela é concebida sob medida para um sistema específico, visto que cada sistema pode possuir diferentes necessidades. As diretrizes ditadas em uma política de segurança indicam o que cada componente do sistema (usuários, máquinas, etc) pode ou não pode fazer. As normas indicam o que cada componente está habilitado a fazer e como deverá ser feito.

2.1.2.3. Usabilidade

A usabilidade de software é uma característica determinante para o melhor aproveitamento dos recursos existentes em uma aplicação. Ter usabilidade representa, antes de tudo, expor de forma explícita toda a potencialidade que um produto pode ter frente a sua utilização.

Conforme a norma 9241 da ISO, citada por Filho (2008), a usabilidade é compreendida como a capacidade que um sistema interativo oferece a seu usuário em determinado contexto de uso para realizar tarefas com eficácia, eficiência e satisfação. Torres e Mazzoni (2004), seguindo a mesma norma regulamentadora, definem que a usabilidade ocorre em função da eficiência, eficácia e satisfação com a qual os usuários podem alcançar seus objetivos em ambientes específicos, quando utilizam determinado produto ou serviço.

Segundo Nielsen e Loranger (2007), ter usabilidade, por um lado, oferece suporte aos objetivos almejados no desenvolvimento; por outro lado, simplifica e aprimora a interação entre o usuário e a tecnologia. Onde, com um design mais útil, os usuários encontram e gerenciam informações facilmente; e, com uma arquitetura de informação apropriada, os usuários sentem-se à vontade para explorar as possibilidades provenientes da utilização do produto. A seguir, a *Tabela 1* mostra alguns critérios operacionais que definem, de maneira mensurável, como o desenvolvimento de software pode obter usabilidade no seu produto final.

Critério	Descrição
Facilidade de aprendizado	Tempo necessário para aprender um conjunto básico de operações que permitem ao usuário iniciar suas tarefas
Facilidade de memorização	O sistema deve ser fácil de lembrar, de maneira que o usuário ocasional seja capaz de reutilizá-lo sem ter que reaprendê-lo
Eficácia	O conjunto de tarefas especificado deve ser completado acima do patamar de performance definido
Eficiência	Uma vez que o usuário aprendeu a utilizar o sistema, um alto nível de produtividade é atingido
Versatilidade	Número de diferentes funcionalidades que o sistema oferece
Flexibilidade	O sistema deve permitir variações na forma de realização da tarefa e no ambiente de realização
Atitude	A tarefa deve ser realizada dentro de níveis definidos como aceitáveis em termos de cansaço, desconforto, frustração e esforço pessoal
Satisfação	Uso continuado do sistema de forma estimulada

Tabela 1 - Critérios operacionais de usabilidade para o desenvolvimento de software. (Fonte: adaptado de Gomes e Padovani (2005))

Nos dias de hoje, a usabilidade de software pode ser o fator decisivo, tanto para o sucesso, quanto para o fracasse de uma aplicação, seja ela educacional ou não. O desenvolvimento deve estar focado o tempo todo em oferecer a melhor experiência de utilização possível. A preocupação com a usabilidade de um projeto de software deve ser constante durante todo o processo de desenvolvimento, levando sempre em consideração o design de interface projetado e a sua validação frente aos usuários, evitando, ao máximo, a falta de acessibilidade e os erros de usabilidade; (ALVES e PIRES, 2002).

2.2. Conceitos gerais a cerca de um framework

Um conceito muito importante desta pesquisa denomina-se framework, e sem o entendimento deste assunto torna-se muito difícil compreender os elementos constitutivos deste trabalho. A fim de informar ao leitor a cerca de tal conceito, esta seção objetiva a compreensão de significados, utilidade, aplicação e benefícios relacionados à tecnologia de frameworks.

Em uma tradução do inglês para o português a partir do dicionário Michaelis (2013), o significado da palavra framework é encontrado como o substantivo “estrutura”. Naturalmente percebe-se nesse conceito que um framework se trata de algo que serve para moldar, dar padronização ou reaproveitamento a alguma coisa através de uma estrutura predefinida já existente, tal como uma receita que pode gerar resultados incríveis, porém, que por si só não pode existir como elemento físico. Dentro do contexto desse trabalho esta “receita” se refere ao auxílio no processo de desenvolvimento de softwares web, algo pronto que serve de base para criar rapidamente novas soluções de software padronizadas a partir de sua estrutura.

Para ser mais claro e através da visão de estudiosos de frameworks, em referência a tecnologia do desenvolvimento de softwares, a definição para framework pode ser:

Um framework consiste em um conjunto de classes, interfaces e padrões dedicados a solucionar um grupo de problemas através de uma arquitetura de programação flexível e extensível (Govoni, 1999).

Uma pequena aplicação completa com uma estrutura estática e outra dinâmica, desenvolvidas para resolver um conjunto restrito de problemas (Fayad, 2000).

Gomes (2002) explica que um framework descreve a arquitetura de um sistema, os tipos de objetos que ele gera, e como eles interagem. Seguindo a afirmação de Gomes, é possível apontar que uma aplicação baseada em framework herda toda a estrutura dele no seu modo de ser, de comunicar, de agir e de apresentar-se frente ao usuário. E, qualquer conceito que seja dado aos frameworks, os seguintes fatores devem ser levados em conta:

- é um modelo de aplicação orientado ao objeto;
- está incompleto;
- pode ser estendido, portanto é flexível;
- é reutilizável;
- pertence a um domínio específico de aplicações;
- é constituído principalmente por um conjunto de classes e suas interconexões;
- disponibiliza um padrão a ser seguido e uma série de funcionalidades pré-definidas;
- tem o propósito de gerar aplicações completas através da personalização de suas instâncias.

Para melhor exemplificar estes conceitos, a *Figura 1* apresenta muito bem como segue a utilização de um framework no processo de desenvolvimento de um software.

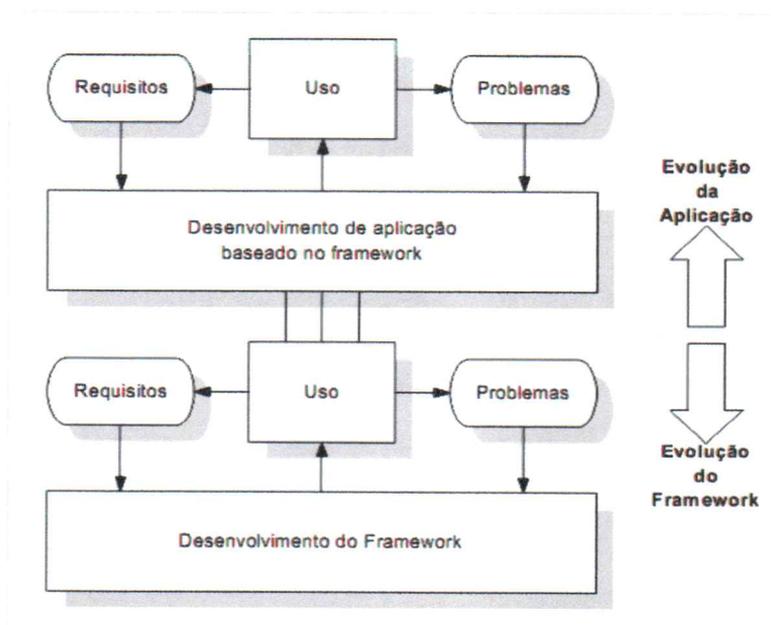


Figura 1 - Esquema de utilização de um framework de software. (Fonte: Gomes, p. 44, 2002.)

2.2.1. Expectativas e motivação para a utilização de frameworks

Um bom framework pode reduzir sensivelmente o custo de desenvolvimento de uma aplicação porque ele permite a reutilização tanto do código como do projeto de um sistema. De uma maneira geral, pode-se dizer que ele acumula o conhecimento e modela o comportamento de toda uma família de aplicações (Gomes, p. 22, 2002).

Embora, o uso de frameworks seja apresentado nesta pesquisa como uma solução de desenvolvimento web moderna, está prática não é uma realidade tão nova assim entre os desenvolvedores de softwares tradicionais. Há muito tempo já se usa frameworks como base para o desenvolvimento de aplicações, e doze anos atrás, em sua publicação, Gomes (2002) já apontava que a grande utilização de frameworks ocorreria futuramente através de fatores como:

- a evolução e o aumento de complexidade dos sistemas de informação;
- a diversidade de plataformas para software base, linguagens de programação e bancos de dados;
- e a adaptação para utilização de novas tecnologias de hardware e software.

Gomes (2002) descreve perfeitamente três dos inúmeros fatores considerados decisivos para se optar por desenvolver um software através de um framework nos dias de hoje. Os motivos para se utilizar um framework como base para um aplicação de software divergem bastante, mas sempre permeiam premissas como ganho de produtividade e tempo, redução de custos, riscos e falhas, utilização de estruturas padronizadas e modernas, etc. Sommerville (2007) aponta fatores como: confiança aumentada, risco de processo reduzido, uso eficiente de especialistas, conformidade com padrões e desenvolvimento acelerado como benefícios deste tipo de reutilização. Para Sommerville, esta prática possibilita que apenas poucos componentes de software precisem ser especificados, projetados, implementados e validados.

Gomes (2002) analisa que a utilização de frameworks constitui uma alternativa para minimização de problemas relacionados ao processo de desenvolvimento de uma aplicação. Segundo Gomes, um framework diminui

sensivelmente o esforço de desenvolvimento, testes e manutenção, promove alta propriedade e reduz sensivelmente o custo de desenvolvimento necessário.

Atualmente existe uma infinidade de frameworks, cada um segundo seus objetivos, como é o exemplo do próprio jQuery UI, do Adobe Flex¹, do YUI², entre outros. Como pode ser visto na *Figura 2*, enumerar benefícios comuns universais para qualquer framework seria no mínimo insensato. Porém ao analisar um framework como solução de software pode se obter expectativas sobre os futuros benefícios que ele deverá proporcionar a futura aplicação.



Figura 2 - Exemplo de frameworks para desenvolvimento web. (Fonte: Krify (2013).)

Na primeira das hipóteses, segundo Gomes (2002), a utilização de frameworks pode significar uma melhoria significativa do processo de desenvolvimento de sistemas. Os frameworks proporcionam a possibilidade de superação de alguns desafios que empresas e profissionais que desenvolvem sistemas desejam vencer:

¹ O Adobe Flex é um framework multi-plataforma desenvolvido pela empresa *Adobe Systems, Inc.*, que utiliza a linguagem *ActionScript* para desenvolver aplicações para a tecnologia *Flash*. Mais informações no endereço da web:

http://www.oficinadanet.com.br/artigo/1482/o_que_e_adobe_flex

² O YUI é um framework livre e de código aberto desenvolvido pelo grupo de desenvolvimento da empresa *Yahoo! Inc.*, que utiliza as linguagens *JavaScript* e *CSS* para construção de aplicações web. Mais informações no endereço da web:

<http://developer.yahoo.com/yui/>

- diminuição do esforço de desenvolvimento;
- aumento da qualidade da aplicação;
- equalização da curva de aprendizagem;
- maior eficiência e agilização;
- melhor integrabilidade;
- facilidade de validação e manutenção;
- maior padronização.

Contudo, a concretização destas expectativas estará sempre em função da correta aplicação dessa técnica de desenvolvimento.

Na segunda das hipóteses, Fayad e Schmidt (1997) destacam os benefícios primários que um framework concede à produção de sistemas de uma forma muito parecida:

- reusabilidade;
- modularidade;
- flexibilidade (possibilidade de extensão);
- inversão de Controle;

Conclui-se que frameworks constituem uma alternativa para minimização de problemas relacionados ao processo de desenvolvimento de uma aplicação. Objetivando maximizar as propriedades de um framework como a reusabilidade e flexibilidade, além de diminuir sensivelmente o esforço de desenvolvimento, testes e manutenção de um sistema, uma alternativa para isso seria anexar soluções no framework, que possibilitassem a parametrização de parte da lógica de programação em tempo de execução, ou seja, a possibilidade de parametrização de regras de negócio. Para que essa solução funcione corretamente e que seja intuitiva para

usuários comuns, faz-se necessário o desenvolvimento de uma interface amigável no framework que faça a gestão destas regras.

2.3. Entendendo a tecnologia jQuery

Escreva menos, faça mais... (John Resig, Criador da biblioteca jQuery).

jQuery é uma biblioteca javascript (*Figura 3*) criada por John Resig (*Figura 4*) e disponibilizada como software livre e aberto sob às licenças MIT (Massachusetts Institute of Technology) e GPL (GNU General Public License). Segundo o The jQuery Foundation (2013), a tecnologia jQuery torna mais simples a exploração de documentos HTML e manipulação de eventos e interações AJAX³ através de uma API simples, que funciona em um grande número de browsers e que elimina o uso do código javascript a praticamente zero em uma página.



Figura 3 - Logotipo da biblioteca javascript jQuery. (Fonte: JQuery (2013).)



Figura 4 - John Resig, criador do projeto jQuery. (Fonte: Cnet (2010).)

³ A sigla AJAX vem da expressão *Asynchronous JavaScript and XML*. É o uso sistemático de *JavaScript* e *XML* (entre outras tecnologias) para tornar o navegador mais interativo com o usuário, utilizando-se solicitações assíncronas de informações (NIEDERAUER, 2007).

Conforme Silva (2010), a finalidade do uso de jQuery é controlar o comportamento de toda ou partes de uma página web. A biblioteca destina-se a adicionar interatividade e dinamismo às páginas web, proporcionando ao desenvolvedor, funcionalidades necessárias à criação de scripts que visem a incrementar, de forma progressiva e não obstrutiva, a usabilidade, a acessibilidade e o design, enriquecendo a experiência do usuário. Ainda segundo o autor a utilização desta biblioteca é útil para:

- adicionar efeitos visuais e animações;
- acessar e manipular o DOM;
- buscar informações no servidor sem necessidade de recarregar a página;
- prover interatividade;
- alterar conteúdos;
- modificar apresentação e estilização;
- simplificar tarefas específicas de JavaScript;
- realizar outras tarefas relacionadas às descritas.

A utilização da tecnologia jQuery tornou-se muito popular recentemente. Em um comparativo do interesse de busca dos usuários na internet, a partir do aplicativo Google Trends (2013a), fornecendo os termos “DHTML”⁴ e “jQuery” e comparando dados entre os anos de 2004 a 2013 (*Figura 5*), é possível perceber facilmente que o interesse por jQuery aumentou vertiginosamente nos últimos anos. No entanto, o interesse por DHTML nas buscas, mesmo vindo de um período crescente no início da progressão, não atingiu a mesma popularidade do jQuery e não conseguiu manter-se em alta durante os anos analisados na comparação.

⁴ Segundo About (2013), assim como a tecnologia jQuery, DHTML é uma prática desenvolvimento web que visa o melhor aproveitamento de uma aplicação através da combinação entre Javascript, XML e CSS.

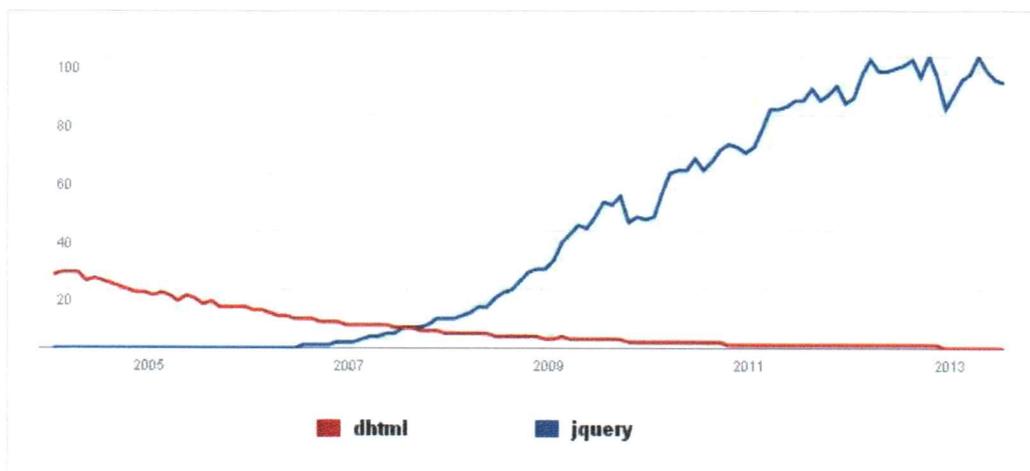


Figura 5 - Comparativo de interesse de busca entre "DHTML" e "jQuery" em produtos da Google Inc.

Similar ao caso anterior, comparando a tecnologia AJAX com a tecnologia jQuery a partir de suas nomenclaturas no aplicativo Google Trends (2013b), considerando dados de 2004 a 2013, temos mais um panorama muito interessante. Neste panorama é possível perceber como o interesse por jQuery superou o interesse por AJAX nas buscas em alguns produtos de software da empresa Google Inc. durante os anos analisados. Onde, ao analisar os resultados obtidos no aplicativo (*Figura 6*), é possível perceber que a popularidade do termo "AJAX" entre as buscas era crescente até meados do ano de 2007, no entanto, a partir deste ponto, decaiu e, por volta de 2010, foi superado pelo interesse de pesquisa direcionado a jQuery.

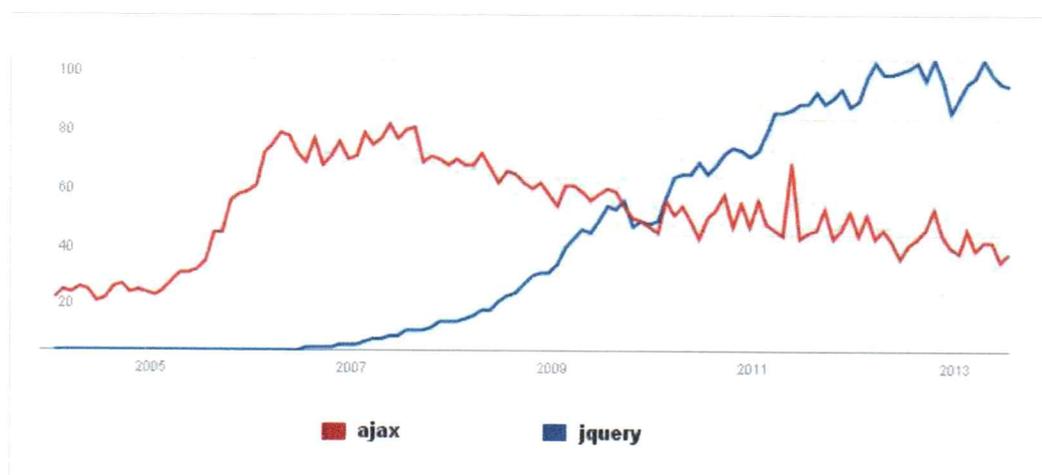


Figura 6 - Comparativo de interesse de busca entre "AJAX" e "jQuery" em produtos da Google Inc.

Segundo pesquisas realizadas pelo site W3Techs (2013), jQuery é a biblioteca javascript mais popular atualmente, sendo usada em cerca de 54,2% dos sites analisados em sua pesquisa, como mostra a *Figura 7* os resultados obtidos a partir dos dados de utilização das diversas tecnologias do mercado.

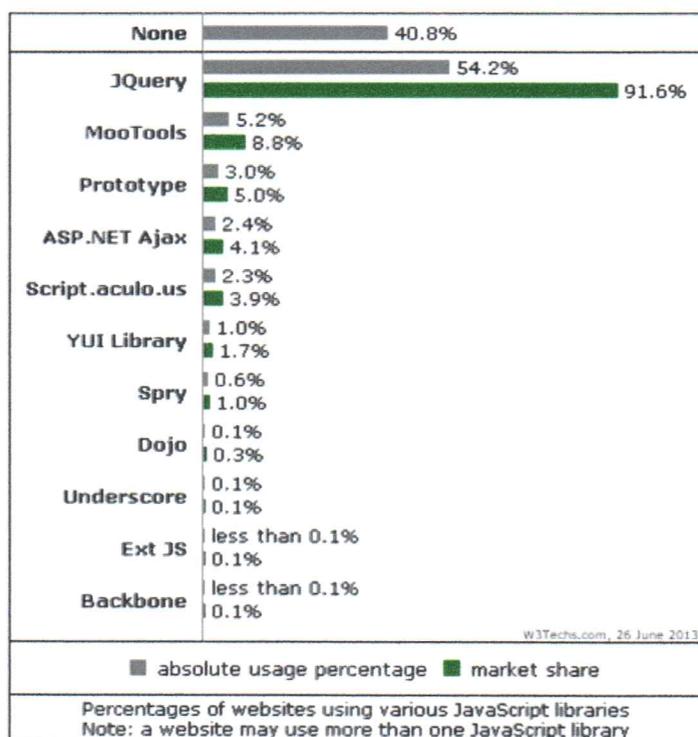


Figura 7 - Resultados da pesquisa realizada pelo W3Techs. (Fonte: W3techs (2013).)

Além da grande aceitação entre os desenvolvedores atuais, sendo utilizada em cinco (05) de dez (10) sites que utilizam bibliotecas javascript, segundo JQuery (2013), organizações muito conhecidas como o Google, Bank of America, Wordpress, Digg, Netflix, Mozilla.org, Dell, entre outras fazem uso da sua tecnologia.

2.3.1. Visão geral do jQuery UI Framework

Não é necessário começar a codificação de um programa do zero, é possível selecionar de uma biblioteca de classes, aquelas que possuem características que mais se aproximam de suas necessidades e aplicá-las no projeto (Gomes, p. 14, 2002).

O *jQuery UI Framework* é um conjunto de soluções de interações de interface com usuário, efeitos, widgets e temas construídos a partir da Biblioteca JavaScript jQuery (JQUERY UI, 2013). Segundo Silva (2012), este framework foi criado em 1º de julho de 2007 por um grupo de desenvolvimento com o propósito de ser uma ferramenta destinada à criação de elementos ricos de interface com o usuário, elementos esses, determinados “ricos” porque simulam a experiência oferecida pelos programas utilizados no ambiente desktop.



Figura 8 - Logotipo do framework web jQuery UI. (Fonte: JQuery UI (2013).)

Assim como a própria biblioteca jQuery, o uso do framework *jQuery UI* (Figura 8) possibilita que o desenvolvedor adicione elementos de interface de usuário utilizando o mínimo de código. Seu princípio de funcionamento é a modificação do Modelo de Objetos de Documentos (DOM⁵). Para modificar o DOM os mecanismos nativos da biblioteca identificam a marcação HTML criada pelo autor e insere elementos adicionais e atributos classes, de modo a criar condições para a estilização adequada à criação do elemento de interface.

Silva (2012) classifica os elementos de interface disponíveis na API do jQuery UI de acordo com sua finalidade, em grupos conforme mostrado no *Quadro 1*.

Grupo	Nome	Descrição
Componentes	Draggable (Arrastar)	Cria elementos capazes de serem arrastados pela interface.
	Droppable (Soltar)	Cria elementos para receber os elementos arrastáveis
	Resizable (Dimensionar)	Cria elementos cujas dimensões podem ser controladas pelo usuário.

⁵ DOM (Document Object Model - Modelo de Objetos de Documentos) é uma especificação da W3C onde pode-se dinamicamente alterar e editar a estrutura, conteúdo e estilo de um documento eletrônico, permitindo que o documento seja mais tarde processado e os resultados desse processamento, incorporados de volta no próprio documento.

	Selectable (Selecionar)	Cria elementos que possam ser selecionados pelo usuário, quer individualmente quer em grupos.
	Sortable (Ordenar)	Cria elementos que possam ser ordenados pelo usuário com ação de arrastar e soltar.
Widgets	Accordion (Acordeão)	Cria o efeito acordeão, para ocultar/mostrar conteúdos.
	Autocomplete (Atocompletamento)	Apresenta ao usuário uma lista de sugestões de palavras à medida que ele digita em um campo de texto.
	Button (Botão)	Cria de vários tipos de botão
	Datepicker (Seletor de datas)	Cria uma janela pop-up para seleção da data a ser digitada em um campo destinado a coletar uma data.
	Dialog (Janela de diálogo)	Cria vários tipos de janela de diálogo, entre elas janelas modais.
	Progressbar (Barra de progresso)	Cria uma barra indicativa do andamento de uma tarefa.
	Slider (Controle deslizante)	Cria um botão arrastável em uma guia para seleção de um valor compreendido em determinada faixa.
	Tabs (Abas)	Cria uma interface cuja navegação é feita com o uso de abas.
Efeitos	Color Animation	Esse efeito destina-se a animar as cores de um elemento.
	Toggle class	Esses efeitos destinam-se a animar elementos baseados na manipulação dos seus atributos classe.
	Add class	
	Remove class	
	Switch class	
	Effect	Aplica em um elemento uma série de efeitos de animação padrão da biblioteca jQuery, tais como os efeitos de esmaecimento, pulsação, balanço, sacudir etc.
	Toggle	Aplica um dos efeitos padrão da biblioteca jQuery com a finalidade de alternar entre ocultar e esconder um conteúdo.
	Hide	Efeito destinado a ocultar um conteúdo.
Show	Efeito destinado a mostrar um conteúdo.	
Utilidades	Position	Destina-se a controlar e manipular o posicionamento de elementos na interface.
	Widgets	Destina-se à criação de widgets personalizados.

Quadro 1: Classificação dos itens de interface que formam a API do jQuery UI Framework.

2.3.2. Desenvolvimento subsidiado pela utilização de plug-ins

Por ser distribuída como software livre, jQuery tem o apoio e o envolvimento de uma considerável comunidade. Desenvolvedores de diversos locais do mundo tem contribuído em larga escala com novas ideias que incrementem não só a

biblioteca, mas também as técnicas de desenvolvimento jQuery. Estas colaborações melhoram e evoluem continuamente a biblioteca e seus produtos, e favorecem uma característica muito importante para os desenvolvedores interessados em implementar aplicações a partir desta tecnologia, o método de desenvolvimento jQuery subsidiado por plugins.

O método de desenvolvimento subsidiado por plugins é uma característica marcante da Biblioteca jQuery. A partir da integração de plugins a facilidade encontrada na utilização da biblioteca jQuery se multiplica bastante, pois similarmente a aplicativos, os plugins eliminam a implementação de tarefas comuns relacionadas ao desenvolvimento de softwares web, e que apenas na utilização desta tecnologia não seria muito simples desenvolver. O método funciona tanto para a validação de um formulário como para o gerenciamento de arquivos de um servidor, reduzindo a preocupação em criar tais funcionalidades, abstraindo as interações de mais baixo nível, simplificando e padronizando a realização de tarefas um tanto mais complexas.

3. DESENVOLVIMENTO DO SISTEMA *IFACADÊMICO*

O Sistema de Gerenciamento de Coordenações de Curso *IFACADÊMICO* é um software que foi desenvolvido com a finalidade de automatizar e padronizar, de forma dinâmica e ágil, as necessidades de docentes e discentes que formam as diversas coordenações de curso do *IF Sertão-PE, Campus Floresta*.

O *IFACADÊMICO* (Figura 9), como uma aplicação acadêmica desenvolvida para o ambiente web, surge para suprir virtualmente as necessidades de comunicação encontradas no Campus Floresta. Isto porque, com este aplicativo, é possível fornecer, obter ou disponibilizar informações referentes aos alunos, professores, cursos, turmas e coordenações em pouco tempo, tão antes quanto possível; manter um canal de comunicação seguro, confiável e flexível acessível aos envolvidos; e, gerenciar situações comumente encontradas frente aos usuários no processo de interação existente no seu meio. E, dessa forma, promover a participação permanente de todos os envolvidos no processo educacional existente na instituição.

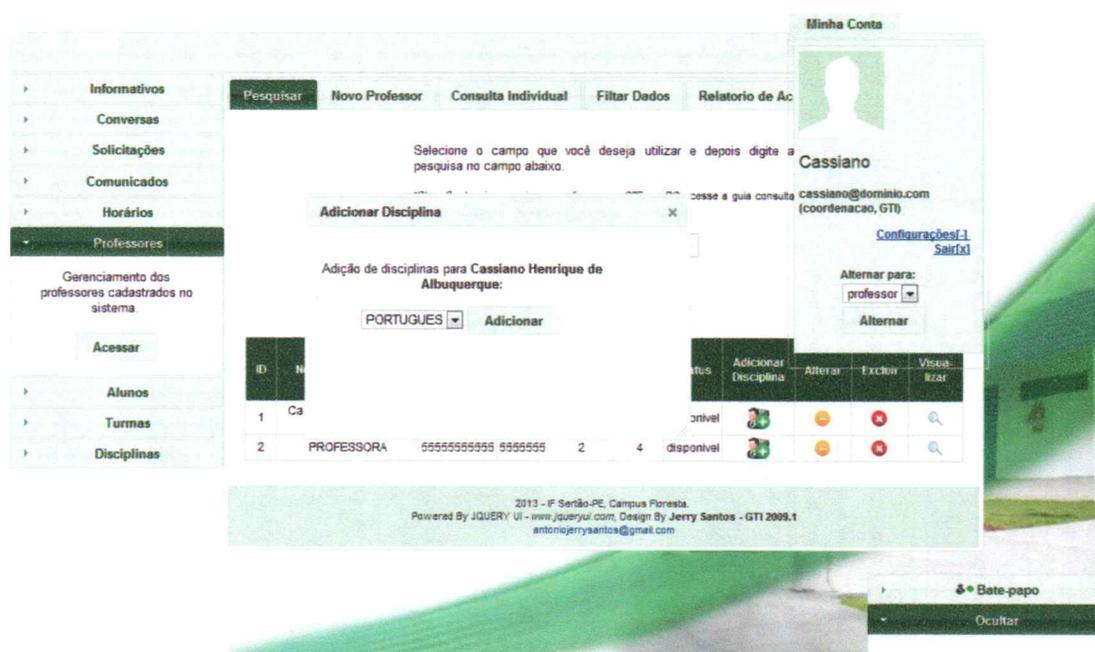


Figura 9 - Exemplo de interface gráfica do sistema *IFACADÊMICO*: Interface do gerenciamento de professores do módulo coordenação.

Por meio de um ambiente simples e ótimos níveis de segurança, ele oferece maior precisão, controle e segurança à forma como os problemas são resolvidos; minimizar o considerável acúmulo de documentos físicos atual; fornecer acesso seguro e autenticado em disponibilidade interna ou externa; possibilitar a comunicação e compartilhamento de documentos.

A partir da facilitação da troca de informações entre os envolvidos, o sistema se propõe a beneficiá-los:

- contribuindo de forma significativa para a melhoria do ambiente escolar;
- aumentando a comunicação e produtividade;
- oferecendo garantia na cobrança de documentos e solicitações;
- diminuindo o prazo de resposta entre as partes;
- promovendo acompanhamento de cursos, turmas, alunos e professores virtualmente;
- incentivando o uso de novas tecnologias no *Campus Floresta do IF Sertão-PE*;
- proporcionando ambiente virtual seguro e confiável.

3.1. Análise e projeto

Segundo Spinola (2008), a atividade de identificação e especificação de requisitos de software é uma atividade bastante desafiadora para o desenvolvimento de uma aplicação. Pois, esta fase é crucial para que se possa descobrir, avaliar e definir os objetivos a serem alcançados com o desenvolvimento do software.

Na fase de análise, o analista deve interpretar, da melhor forma possível, a necessidade do cliente e adaptar as características necessárias para o software em artefatos interpretáveis para o desenvolvimento a ser realizado. Existem várias

técnicas para realizar este tipo de análise, no entanto, a técnica utilizada para o desenvolvimento do sistema *IFACADÊMICO* foi a entrevista. Onde, a partir de entrevistas realizadas com responsáveis na coordenação do curso de Gestão da Tecnologia da Informação, foi possível conhecer a necessidade do cliente⁶ e definir artefatos para o desenvolvimento do sistema.

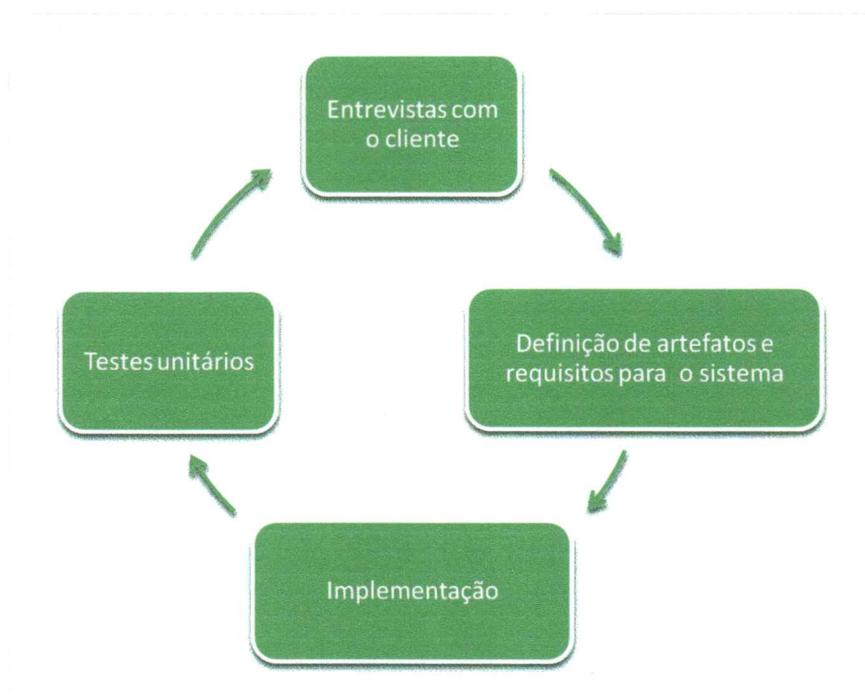


Figura 10 - Ciclo de desenvolvimento do sistema *IFACADÊMICO*.

3.2. Implementação

O *IFACADÊMICO*⁷ foi desenvolvido e implantado⁸ na plataforma web, utilizando o *MySQL* como sistema de gerenciador de banco de dados. A linguagem de programação *PHP* foi adotada para a implementação do sistema, e, para complementar toda esta estrutura técnica, foi utilizado o *jQuery UI Framework* como

⁶ A denominação “cliente” refere-se aquele que solicita o desenvolvimento do software.

⁷ Para mais informações sobre o sistema *IFACADEMICO* consulte a documentação do software disponível na seção *APÊNDICES* deste relatório.

⁸ O sistema *IFACADÊMICO* está hospedado na web e possui o domínio *www.ifacademico.com*.

base para a implementação da interface RIA⁹ (*Rich Internet Applications*) do sistema, o *CKEditor* como editor rich-text e o *jQuery Fileupload* como gerenciador de arquivos. Também, foi utilizada a linguagem de estilos CSS, que dá suporte a organização do conteúdo emitido pelo software, bem como ao framework RIA. Tudo, com o intuito de permitir o funcionamento de recursos avançados de interface, manter a alta produtividade no processo de desenvolvimento e estabelecer um alto padrão de usabilidade da aplicação.

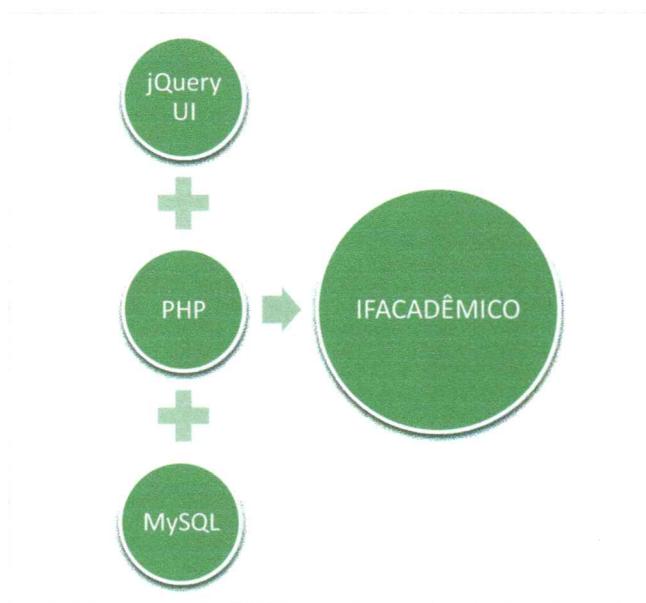


Figura 11 - Tecnologias utilizadas para a concepção do sistema IFACADÊMICO.

⁹ Conforme Deitel (2008) define RIAs como aplicações web que oferecem a sensibilidade, recursos e funcionalidades "ricos", que se aproximam das aplicações de desktop.

4. METODOLOGIA

Identificou-se no *capítulo 2* deste trabalho características importantes para que um ambiente virtual seja utilizado para mediar a comunicação de docentes e discentes e para que se desenvolva um sistema de gerenciamento de coordenações de curso (*seção 2.1*), para que se use um framework no desenvolvimento de aplicações de software (*seção 2.2*), para que se aplique a tecnologia jQuery no desenvolvimento web (*seção 2.3*). O aprendizado destes conceitos, provenientes do conhecimento de diferentes teóricos e pesquisadores dos múltiplos aspectos que formam esta pesquisa, indicou que a utilização do *jQuery UI Framework* (*subseção 2.3.1*) poderia ser uma solução extremamente vantajosa para o desenvolvimento atual de aplicações web. Com o intuito de avaliar essa hipótese, definiu-se por realizar uma análise prática sobre a sua real veracidade.

4.1. Delineamento da pesquisa

Esta pesquisa situa-se na interface da aplicação da tecnologia de frameworks para melhorar o desenvolvimento de softwares voltados à web, apoiando-se em áreas do desenvolvimento de software como: análise e desenvolvimento de sistemas, desenvolvimento de aplicações web e engenharia e qualidade de software. Esta é uma pesquisa de caráter descritivo e investigativo que pretende analisar a aplicação do *jQuery UI Framework* em campo, utilizando-se dos pressupostos da abordagem qualitativa de dados para obtenção de seus resultados.

Uma pesquisa qualitativa é entendida como uma investigação que tem como preocupação central o exame dos dados em um tipo de profundidade que não é captada por números, tabelas e dados quantitativos. Este tipo de pesquisa pretende observar o seu escopo de forma abrangente e em direções variadas, fazendo uso de objetos para descrever a realidade de forma “*holística*” (VIGORENA e BATTISTI, 2011).

Neves (1996) descreve a pesquisa qualitativa como um método cuja abordagem costuma ser direcionada ao longo de seu desenvolvimento; que não busca enumerar ou medir eventos; e, geralmente, não emprega instrumental estatístico para análise dos dados. Para Neves, o foco de interesse da abordagem qualitativa é a obtenção de dados descritivos mediante contato direto e interativo do pesquisador com a situação do objeto de estudo. Logo, é natural nesse tipo de pesquisa que o pesquisador procure entender os fenômenos, segundo a perspectiva dos participantes da situação estudada, e, a partir daí, situe a sua interpretação dos fenômenos estudados na pesquisa.

Godoy (1995a, p. 62), citado por Neves (1996), enumera características essenciais capazes de identificar uma pesquisa desse tipo, da seguinte forma:

1. o ambiente natural é fonte direta de dados;
2. o pesquisador é considerado instrumento fundamental;
3. o caráter é descritivo;
4. o enfoque é dedutivo.

Ainda seguindo o pensamento apresentado por Neves (1996), o autor aponta que em sua maioria, os estudos qualitativos não impedem o pesquisador de empregar a lógica do empirismo científico, mas partem da suposição de que seja mais apropriado empregar a perspectiva da análise fenomenológica, quando se trata de fenômenos singulares e dotados de certo grau de ambiguidade.

4.2. Procedimento de coleta e análise dos dados

Partindo dos conceitos da metodologia desta pesquisa, o método qualitativo a ser aplicado é o estudo de caso. O objetivo do estudo de caso é a análise profunda de uma unidade de estudo (NEVES, 1996). Lüdke e André (1986, p. 17) *apud* Oliveira (2008), descrevem que um estudo de caso:

- visa à descoberta;

- enfatiza a 'interpretação em contexto';
- busca retratar a realidade de forma completa e profunda;
- usa uma variedade de fontes de informação;
- revela experiência vicária e permitem generalizações naturalísticas;
- procura representar os diferentes e às vezes conflitantes pontos de vista presentes numa situação social;
- utiliza uma linguagem e uma forma mais acessível do que os outros relatórios de pesquisa.

O estudo de caso não é uma técnica específica, mas uma análise minuciosa e objetiva de uma situação a ser investigada, com foco em somente um problema. A partir deste tipo de metodologia pode-se tanto descobrir fatos-chave da situação em seu estado atual ou em desenvolvimento, quanto descobrir relações significativas de todos os fatos entre si.

Segundo Vigorena e Battisti (2011), dentro do enfoque qualitativo realizado através do estudo de caso destacam-se alguns procedimentos de coleta de dados que podem ser utilizados para investigar o objeto que se pretende conhecer, como, a exemplo, podemos citar: documentação, entrevista, entrevista em profundidade, observação direta e participante, história de vida e *focus group*.

Esta pesquisa, dentro dos seus conceitos reais, realiza um estudo de caso que coloca o *jQuery UI Framework* sob observação. Onde, a partir da observação participante, é possível demonstrar características importantes sobre eficiência, viabilidade e potencialidade provenientes deste tipo de abordagem para o desenvolvimento de software. Na observação participante o pesquisador busca relatar a realidade por meio de um tratamento de acontecimentos em tempo real, porém não a faz como um mero observador passivo. Para Vigorena e Battisti (2011), este tipo de observação é uma oportunidade de se perceber a realidade do ponto de vista de alguém que faz parte daquele contexto, e que existe ainda a possibilidade

de manipulação dos eventos vivenciados na realidade investigada por parte do pesquisador.

4.3. O objeto de estudo

A realização desta pesquisa tomou como caso o desenvolvimento de um software educacional específico, onde se puderam empregar métodos, técnicas e políticas que possibilitassem a observação do *jQuery UI* como base para o desenvolvimento de software web atual.

O objeto de estudo desta pesquisa é o produto final de software obtido a partir da aplicação do framework em campo, o sistema *IFACADEMICO*. Como descrito no *capítulo 3* deste relatório, este é um software acadêmico desenvolvido como um recurso para melhorar a dinâmica de comunicação existente entre docentes e discentes das coordenações de curso do *IF Sertão-PE Campus Floresta*. Nele é possível fornecer, obter ou disponibilizar informações referentes aos alunos, professores, cursos, turmas e coordenações em pouco tempo, tão antes quanto possível. Para tanto as funcionalidades e o código-fonte desta aplicação compreendem um subsídio muito importante para gerar informações e dados nesta pesquisa.

5. ANÁLISE DO OBJETO DE ESTUDO

Levando em conta o método e a abordagem de pesquisa apresentados no *capítulo 4* deste trabalho, a análise do objeto de estudo considerou as funcionalidades implementadas e o próprio código-fonte do sistema *IFACADÊMICO* como bases seguras para determinar os resultados finais desta pesquisa. Com esse pressuposto, e tendo como foco o escopo aqui definido, a análise das funcionalidades do sistema subdividiu a interpretação do objeto de estudo em duas (02) categorias: *Análise da Utilização do jQuery UI Framework no Sistema IFACADÊMICO* e *Análise da Adequação da Tecnologia à Implementação de Funcionalidades Complexas do Sistema IFACADÊMICO*.

As próximas seções deste trabalho descrevem a análise qualitativa resultante da observação participante realizada no desenvolvimento do sistema *IFACADÊMICO*. Para complementar a análise são apresentados imagens de funcionalidades do sistema e fragmentos do código referente à sua implementação na tecnologia jQuery. Este capítulo, a partir do conhecimento obtido em campo, objetiva fornecer um panorama de como a aplicação da tecnologia do *jQuery UI Framework* proporcionou resultados satisfatórios ao desenvolvimento proposto.

5.1. Análise da utilização do jQuery UI Framework no sistema *IFACADÊMICO*

Esta seção tem como objetivo analisar o objeto de estudo, segundo a dinâmica de implementação entre o framework *jQuery UI* e as funcionalidades do sistema *IFACADÊMICO*. Interpretando o que a tecnologia do framework proporcionou ao sistema, levando em consideração a contribuição do desenvolvimento para a aplicação final, o quê e como aconteceu. Nesta categoria, três (03) características chamaram a atenção: a implementação de interfaces, a reutilização de software e a dinâmica de interação da tecnologia na transmissão de dados.

5.1.1. Implementação da interface com o usuário

A interface com o usuário é um fator crítico para a aceitação de qualquer software. Filho (2008) apresenta a importância da interface de um software, defendendo que assim como qualquer outro produto, a interface de um programa precisa ser simples, fácil de usar e deve possibilitar que o usuário realize suas tarefas de maneira eficiente e com satisfação.

A análise do objeto de estudo a partir dos critérios definidos, compreendeu dois aspectos da implementação do *jQuery UI* no sistema *IFACADÊMICO*: o aplicativo *Theme Roller* e o grupo *Widgets* presente na API do framework.

5.1.1.1. Theme roller

O *Theme Roller* é um recurso disponível no site¹⁰ *jQuery UI* que fornece mecanismos para a configuração do produto de software a ser gerado pelo *jQuery UI Framework*. A partir deste aplicativo on-line, o desenvolvedor define a aparência da aplicação que deseja, e, após o processo, baixa os arquivos do Framework *jQuery* para utilização local.

¹⁰ O aplicativo *Theme Roller* pode ser acessado através do endereço da web: <http://www.jqueryui.com/themeroller>.

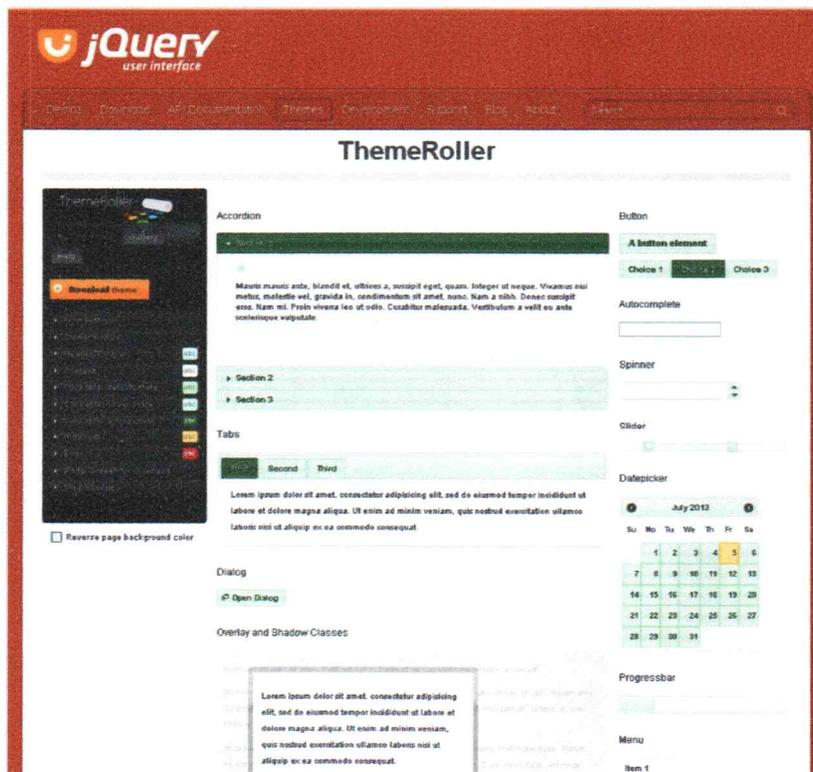


Figura 12 - Interface do aplicativo *Theme Roller*.

Como mostrado na *Figura 12*, o *Theme Roller* possibilita toda a configuração de cores, fontes e estilos para o *jQuery UI Framework*, de forma muito simples e prática. Ao utilizar a ferramenta, é possível alterar todas as características visuais do framework sem a necessidade de alterações, tanto no seu código-fonte, quanto no código-fonte da aplicação desenvolvida.

Ao analisar a importância do *Theme Roller* no sistema *IFACADÊMICO*, tornou-se possível mensurar o quanto a aplicação do desenvolvimento jQuery contribuiu para a melhoria e aperfeiçoamento dos processos de desenvolvimento do sistema. Esta característica da tecnologia jQuery possibilita, além de outras coisas, uma interface totalmente independente do projeto, promovendo, assim, melhores níveis de produtividade e padronização para o desenvolvimento, expansão e manutenção do software.

5.1.1.2. Widgets

O *jQuery UI* reúne bons recursos para criação de interfaces e manipulação de dados na web. Conforme descrito na *subseção 2.3.1*, Silva (2012) classifica esses recursos em quatro (04) grupos distintos (*Quadro 1*), e, entre esses grupos está o alvo da análise desta subseção: o grupo *Widgets*.

Widgets imprimem as características do *jQuery UI Framework* na aplicação. No contexto da tecnologia *jQuery*, widgets são componentes de interface com o usuário que permitem a interação homem-máquina segundo padrões, se não idênticos, superiores aos padrões utilizados em aplicações desktop normais. Para fins de análise, os próximos itens relatam a experiência real da implementação dos widgets no do sistema *IFACADÊMICO*.

5.1.1.2.1. Dialog

Similar a uma janela de dialogo comum, o widget *Dialog* fornece características praticamente idênticas às características de uma janela normal. A *Figura 13* descreve bem o widget, à primeira vista apenas uma janela, porém, ao observar a sua aplicação no sistema, o conceito do componente muda.



Figura 13 - Exemplo de utilização do widget *Dialog*: Interface para alterar os dados do aluno.

A primeira consideração a se fazer sobre o *Dialog* é que este componente é apenas uma marcação *div* da linguagem HTML, assim como é mostrado na *Figura 14*.

```
<!-- Janela de dialogo de alteração -->  
<div id="alteracao" title="Alteração de Dados">  
  <p>  
    Conteúdo da Janela aqui.  
  </p>  
</div>
```

Figura 14 - Exemplo de marcações HTML para instanciar o widget *Dialog*.

A partir da marcação HTML o *jQuery UI* entra em ação, transformando-a no componente. A *Figura 15* imprime o código necessário para instanciar um novo *Dialog* a partir do framework.

```
<!-- Janela de Dialogo -->  
$( "#alteracao" ).dialog();
```

Figura 15 - Exemplo de código jQuery para instanciar o widget *Dialog*.

Além da praticidade envolvida na criação do *Dialog*, a utilização deste widget também é útil no gerenciamento de conteúdo de uma página. Através desta prática, o desenvolvedor pode gerenciar diversas janelas na interface da aplicação usando o mínimo possível de linhas de código. A *Figura 16*, por exemplo, mostra a implementação de alguns comandos assim que o um determinado *Dialog* do sistema *IFACADÊMICO* é fechado.

```
<!-- Atualização do preview do memorando -->
$("#editormemorando").dialog({ beforeclose : function (event) {
    $("#previewMemorando").html($("#editor").val());
    $("#previewMemorando").show();
    $("#accordionMemorando").accordion({autoHeight: false, navigation: t
    $("#addmemorando").show();
    },
    autoOpen: false
});
```

Figura 16 - Exemplo de código jQuery para manipulação de elementos através do widget *Dialog*.

O código impresso na *Figura 16* tem duas características principais: a forma como o framework implementa os comandos e a quantidade de linhas necessárias para implementá-los. Ao analisar esta situação, bem como a forma como o *jQuery UI* implementa o componente, é possível ter uma pequena noção de quão útil o widget *Dialog* pode ser no desenvolvimento de interfaces. Onde, além de proporcionar uma aparência visual sofisticada e prática, o widget também pode manipular tanto suas próprias características, quanto todo o restante da página.

5.1.1.2.2. Tabs

O widget *Tabs* (*Figura 17*) se assemelha em muito com o conceito de guias existentes no padrão de desenvolvimento de interfaces desktop, no entanto, o componente de interface web se sobressai devido às suas possibilidades de aplicação com os demais itens do framework.

Figura 17 - Exemplo de utilização do widget *Tabs*: Interface para gerenciar alunos.

Assim como o *Dialog* e os demais widgets presentes na API do *jQuery UI*, o *Tabs* também é, em essência, apenas marcação HTML, porém, neste caso o código é um pouco mais elaborado.

```

<!-- Guias para Login, Chat, Recuperação, Sobre e Ajuda -->
<div id="tabs">
  <ul>
    <li><a href="#tab1">Login</a></li>
    <li><a href="#tab2">Chat</a></li>
    <li><a href="#tab3">Recuperação</a></li>
    <li><a href="#tab4">Sobre</a></li>
    <li><a href="#tab5">Ajuda?</a></li>
  </ul>

  <!-- Login -->
  <div id="tab1"> Conteúdo da guia Login aqui. </div>

  <!-- Chat -->
  <div id="tab2"> Conteúdo da guia Chat aqui. </div>

  <!-- Recuperação -->
  <div id="tab3"> Conteúdo da guia Recuperação aqui. </div>

  <!-- Sobre -->
  <div id="tab4"> Conteúdo da guia Sobre aqui. </div>

  <!-- Ajuda -->
  <div id="tab5"> Conteúdo da guia Ajuda aqui. </div>
</div>

```

Figura 18 - Exemplo de marcações HTML para instanciar o widget *Tabs*.

A criação do widget segue o mesmo padrão dos demais componentes deste grupo, onde, primeiro monta-se a estrutura HTML e depois a transforma através do código jQuery. Para ilustrar melhor, a *Figura 18* mostra como o exemplo de marcação mostrada na *Figura 19* foi implementado no sistema *IFACADÊMICO*.

```
<!-- Criação das Guias -->
$("#tabs").tabs();
```

Figura 19 - Exemplo de código jQuery para instanciar o widget Tabs.

O enfoque da utilização do *Tabs* está intimamente ligado à usabilidade do sistema. Quanto aos benefícios analisados, foi evidente notar o quanto ele pode melhorar a experiência oferecida na interface com o usuário. Ao observar sua aplicação em campo, este widget apresentou características muito proveitosas para organização de conteúdo no *IFACADÊMICO*, onde, além de ser um componente de fácil implementação, também proporcionou níveis de usabilidade e organização consideravelmente modernos no dias de hoje.

5.1.1.2.3. Acoordion

O *Acoordion* é uma marca característica da utilização do *jQuery UI*. Não por ser usado exclusivamente no framework, mas, sim por trazer este conceito tão raro nos padrões de interface desktop para as páginas web.

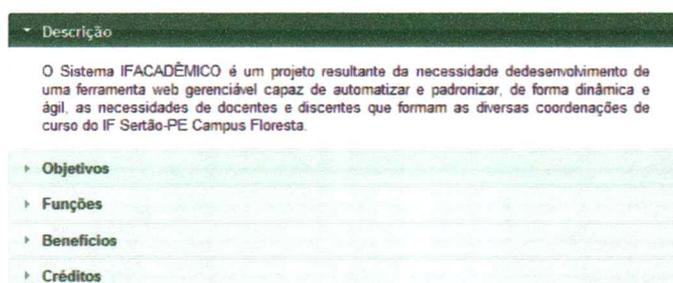


Figura 20 - Exemplo de utilização do widget *Acoordion*: Menu sobre informações do sistema *IFACADÊMICO*.

Similar ao *Tabs*, o widget *Accordion* objetiva dar organização e interatividade às informações da página. Neste componente, a informação pode ser agrupada em sessões bem organizadas, ocupando muito menos espaço na interface.

O funcionamento do *Acoordion* é muito intuitivo. Nele, durante a interação do usuário, enquanto uma sessão do componente está visível, as demais sessões estarão ocultas. Além disso, há um detalhe muito interessante nesta interação: a execução de efeitos visuais no widget proporciona a sensação de estar modificando o elemento fisicamente.

A criação do widget também segue o mesmo padrão dos demais componentes deste grupo. Como mostrado nas *Figuras 21 e 22*, inicialmente monta-se a estrutura HTML e depois a transforma através do código jQuery.

```

<div id="accordionSobre">
  <!-- Item Descrição do Menu -->
  <h1><a>Descrição</a></h1>
  <div>
    <div>
      O Sistema IFACADÊMICO é um projeto resultante da necessidade de
      de uma ferramenta web gerenciável capaz de automatizar e padroniz
      dinâmica e ágil, as necessidades de docentes e discentes que form
      coordenações de curso do IF Sertão-PE Campus Floresta.
    </div>
  </div>

  <!-- Item Objetivos do Menu -->
  <h1><a>Objetivos</a></h1>
  <div>
    Conteúdo aqui...
  </div>

  <!-- Item Funções do Menu -->
  <h1><a>Funções</a></h1>
  <div>
    Conteúdo aqui...
  </div>

  <!-- Item Benefícios do Menu -->
  <h1><a>Benefícios</a></h1>
  <div>
    Conteúdo aqui...
  </div>

  <!-- Item Créditos -->
  <h1><a>Créditos</a></h1>
  <div>
    Conteúdo aqui...
  </div>
</div>

```

Figura 21 - Exemplo de marcações HTML para instanciar o widget *Acoordion*.

```
<!-- Criação do menu da guia sobre -->
$("#accordionSobre").accordion();
```

Figura 22 - Exemplo de código jQuery para instanciar o widget *Acoordion*.

5.1.1.2.4. Datapiecker

Para finalizar a análise da utilização dos wigets do *jQuery UI* na implementação de interfaces do *IFACADÊMICO*, esta subseção apresenta a análise a cima do widget *Datapiecker*.

O *Datapiecker* não é uma novidade nem no desenvolvimento de interfaces para desktop, nem no desenvolvimento de interfaces para a web. Assim como pode ser mostrado na *Figura 23*, é um componente de interface similar a um calendário que possibilita que o usuário selecione uma data a ser adicionada a um campo da página.



Figura 23 - Exemplo de utilização do widget *Datapiecker*: Campo "data" da funcionalidade para gerar relatórios.

Além do nível de usabilidade visivelmente percebido na *Figura 23*, o *Datapiecker* chama atenção na forma que é instanciado na interface da aplicação, pois, instanciar o widget é ainda mais fácil que os demais widgets presentes no framework. A princípio, como mostra nas figuras 24 e 25, basta apenas implementar o campo da data no código HTML, e, por fim, declarar o código do framework na página.

```
Data: <br/> <input type="text" id="dataRelatorio"/>
```

Figura 24 - Exemplo de marcação HTML para instanciar o widget *Datapiecker*.

```
<!-- Criação dos Datapickers -->
$("#dataRelatorio").datepicker();
```

Figura 25 - Exemplo de código jQuery para instanciar o widget *Datapiecker*.

5.1.2. Reutilização de software

A reutilização de software é o princípio-chave de tudo que envolve a tecnologia jQuery. A primeira impressão que se tem ao iniciar o desenvolvimento a partir desta tecnologia, é que não é necessário ser um “*expert*” em programação para começar a criar aplicações web inteligentes.

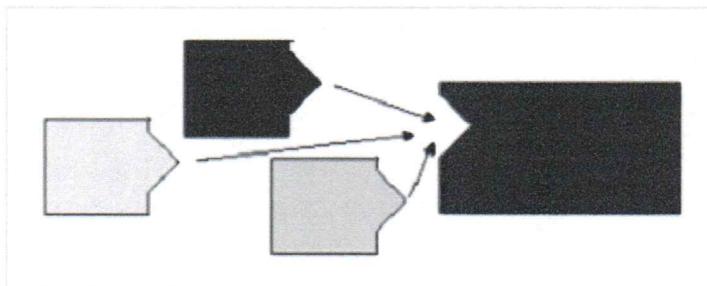


Figura 26 - Desenvolvimento de sistemas através da reutilização de componentes. (Fonte: Gomes, p. 15, 2002.)

Como mostrado na *Figura 26*, a prática de reutilização de software evita o esforço exagerado no desenvolvimento e dá padronização e uniformidade à interface de aplicações web. Ao utilizar as técnicas de reutilização de software presentes na biblioteca jQuery e seus produtos, como é o exemplo do *jQuery UI Framework*, é possível obter resultados finais de software surpreendentes. Muito do sucesso deste conceito parte do princípio de desenvolvimento subsidiado por plug-ins, já apresentado neste trabalho e disponível na *subseção 2.3.2*. E, para descrever

melhor a análise da reutilização de software, a próxima sessão descreve como a prática de desenvolvimento através de plug-ins contribuiu para que o sistema *IFACADÊMICO* fosse implementado.

5.1.2.1. Desenvolvimento jQuery através de plug-ins

A possibilidade de integrar plug-ins a uma aplicação jQuery abre um universo de possibilidades ao desenvolvedor. Semelhante ao caso *Theme Roller*, descrito na subseção 4.1.1.1, o site da tecnologia jQuery fornece um aplicativo que auxilia o desenvolvedor a encontrar plug-ins para o seu projeto, o *jQuery Plugin Registry*. Em um conceito prático, o aplicativo¹¹ compreende um grande repositório onde é possível registrar, buscar e baixar plug-ins jQuery para as mais diversas funcionalidades existentes no desenvolvimento web. A *Figura 27* mostra a interface do aplicativo *jQuery Plugin Registry*.



Figura 27 - Interface do aplicativo *jQuery Plugin Registry*.

¹¹ O aplicativo *jQuery Plugin Registry* pode ser acessado através do endereço da web: <http://www.plugins.jquery.com>.

O sistema *IFACADÊMICO*, partindo dos objetivos desta pesquisa (seções 1.2 e 1.3), compreendeu um total de seis (06) plug-ins em seu desenvolvimento. Porém, para fins de análise, podemos citar dois (02) deles que retratam perfeitamente o princípio de desenvolvimento proposto pela prática de integração de plug-ins: o *CKEditor Adapter* e o *jQuery Fileupload*.

5.1.2.1.1. Ckeditor adapter

No primeiro caso desta análise, a utilização do plugin *CKEditor adapter* possibilitou que o editor rich-text *CKEditor* fosse instanciado a partir dos conceitos do *jQuery UI Framework*.

O *CKEditor* é um editor de texto HTML projetado para simplificar a criação de conteúdo web. Ele é um editor WYSIWYG¹² que traz características comuns em processadores de texto diretamente para as páginas web (CKEDITOR, 2013). Através do editor, o usuário tem uma experiência de uso muito semelhante à experiência de um processador de texto desktop, podendo desde modificar elementos de formatação do texto até criar páginas web completas.

¹² WYSIWYG é o termo usado para classificar ferramentas de edição e desenvolvimento que permitem visualizar, em tempo real, exatamente aquilo que será publicado ou impresso (TECMUNDO, 2013).

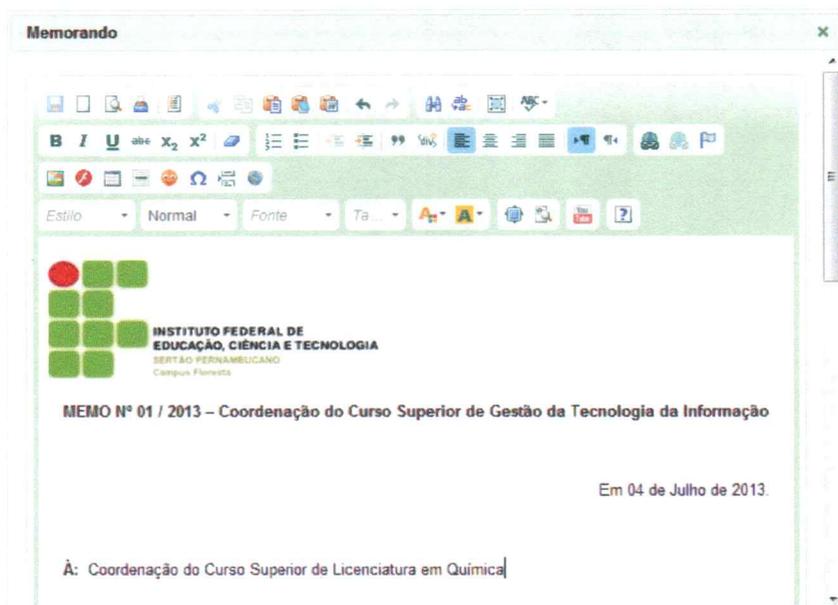


Figura 28 - Interface de edição do memorando virtual.

A *Figura 28* mostra a implementação do editor na funcionalidade editar memorando do sistema. Na edição do memorando o *CKEditor* fornece inúmeros recursos para que o usuário altere a formatação do documento no próprio sistema, de acordo com a sua necessidade. Além disso, uma coisa em particular chamou a atenção nesta análise: a praticidade para implementar o editor no código jQuery. Para dar uma dimensão melhor desta situação, a *Fig 19* apresenta o código jQuery necessário para instanciar o *CKEditor* no sistema *IFACADÊMICO*.

```
<!-- Editor -->
$("#editor").ckeditor();
```

Figura 29 - Código jQuery para implementar o CKEditor.

O código jQuery apresentado na *Figura 29* mostra quão fácil é instanciar o editor através do plugin *CKEditor Adapter*. Essencialmente, para realizar esta função o desenvolvedor informa o identificador de um campo do tipo *textarea* existente no código e chama a função *"ckeditor"*.

Em uma abordagem prática, a implementação do editor compreendeu várias funcionalidades no *IFACADÊMICO*. No sistema é possível visualizar o editor no chat, no atendimento e acompanhamento de solicitações e em alguns cadastros. As contribuições do *CKEditor* para o sistema foram ótimas, ao analisar a sua participação nestas funcionalidades, é possível ter uma pequena demonstração do potencial que a utilização do plug-in trouxe para a aplicação em questão.

A análise do *CKEditor Adapter* pôde perceber quão útil um plug-in pode ser no desenvolvimento jQuery. A aplicação em campo permitiu descobrir níveis de produtividade incríveis para este tipo de desenvolvimento. Em uma definição final, o *CKEditor Adapter* transformou o editor *CKEditor*, um editor de texto independente, em mais um *widget* do *jQuery UI Framework*, compatível, fácil de instanciar e personalizar.

5.1.2.1.2. JQuery file upload

No segundo caso da análise de reutilização de componentes está o *jQuery File Upload*. A utilização deste plug-in permite o envio, e até o gerenciamento de arquivos em diversos tipos de servidores web, a partir da tecnologia jQuery.

Uma característica marcante do *jQuery File Upload* é o quanto ele pode sofisticar o envio de arquivos de uma aplicação, tanto em nível de interface quanto em nível de interação. A *Figura 30*, por exemplo, mostra uma derivação deste plug-in desenvolvida especialmente para o desenvolvimento através do *jQuery UI Framework*.



Figura 30 - Interface da funcionalidade para enviar arquivos.

A Figura 30 imprime a funcionalidade que envia arquivos no sistema IFACADÊMICO. Ao analisá-la é possível observar dois aspectos predominantes na versão do plug-in utilizada no sistema: o nível de usabilidade obtido e a forma como os arquivos são transmitidos para o servidor.

Quanto ao primeiro aspecto, esta funcionalidade proporcionou itens bastante úteis:

- Através de um design muito similar aos designs de padrão desktop, o usuário obtém a dimensão real do estado da execução da tarefa;
- Através de uma dinâmica de utilização bem distribuída, o usuário possui flexibilidade necessária para realizar os processos segundo a sua necessidade.

Quanto ao segundo aspecto observado, esta funcionalidade dá introdução a uma das características mais importantes do desenvolvimento jQuery: a transmissão de dados assíncrona¹³. Este tipo de interação *cliente-servidor* caracteriza-se pela troca de informações de forma isolada, realizando envios e atualizações sem a necessidade de submissão de todos os dados da página.

¹³ Este tema será abordado na seção 5.1.3 deste trabalho.

Em uma abordagem prática, pode-se dizer que a implementação do *jQuery File Upload* abstraiu o processo de envio de arquivos do sistema *IFACADÊMICO*, seguindo conceitos de funcionalidade e interface que, embora similares aos padrões desktop, são considerados altamente modernos e produtivos.

A análise do *jQuery File Upload* mostrou um lado diferente do desenvolvimento jQuery subsidiado por plug-ins. A aplicação em campo gerou experiências construtivas muito úteis para melhoria e aprimoramento dos conceitos de desenvolvimento web existentes atualmente. Em uma definição final, o *jQuery File Upload* transformou o envio de arquivos do sistema *IFACADÊMICO* em uma funcionalidade moderna, dinâmica e usável.

5.1.3. Dinâmica de interação cliente-servidor

Um ponto altamente relevante na implementação do framework *jQuery UI* no sistema *IFACADÊMICO* foi a transmissão assíncrona de dados. Este tipo de interação torna possível a comunicação *cliente-servidor* sem a necessidade de submissão completa dos dados da página. A implementação desta técnica, em primeiro lugar, evitou o envio de dados desnecessários ao servidor; e, em segundo, tornou a utilização da aplicação mais simples para o usuário.

No âmbito desta pesquisa, a partir da aplicação prática em campo, a funcionalidade de transmissão assíncrona de dados do framework representou uma técnica muito prática, onde, entre os resultados analisados (*subseções 4.1.3.1 e 4.1.3.2*), uma característica prevaleceu como cargo-chefe desta funcionalidade: a simplicidade para troca de dados.

5.1.3.1. Envio assíncrono de dados

O envio assíncrono de dados ocorre quando uma página web transmite dados para processamento posterior em um servidor, sem a necessidade de sua submissão completa. O envio assíncrono de dados observado na coleta de dados

desta pesquisa esteve intimamente ligado a implementação das funcionalidades de cadastro do sistema. Para ilustrar a implementação destas funcionalidades no *IFACADÊMICO*, a *Figura 31* apresenta como exemplo a interface utilizada para a funcionalidade cadastrar turma.

Informe os dados abaixo para que possamos realizar o seu cadastro.

*Obs.: Para que o aluno seja cadastrado como representante você deve digitar o id no campo aluno. Se quiser também pode pesquisá-lo automaticamente digitando seu nome no mesmo campo.

Dados de Identificação

Nome:

Curso:

Início em: / /

Capacidade de alunos:

Turno:

 Aguarde, Enviando dados...

Figura 31 - Interface da funcionalidade para cadastrar turmas.

Nesta interface o usuário pode enviar as informações necessárias para cadastrar uma turma sem a necessidade de submissão dos dados à outra página, como era muito comum tempos atrás no desenvolvimento web. Porém esta não é a vantagem de utilização desta funcionalidade do framework, até porque a tecnologia AJAX também pode fazer isso. A grande vantagem da troca de dados assíncrona presente no *jQuery UI* está na estrutura de utilização oferecida pelo framework. A estrutura básica para implementar esta funcionalidade consiste basicamente em definir quais dados serão usados; o alvo da troca de dados; e, a partir daí definir o que vai acontecer assim que o alvo responda à interação. Para exemplificar melhor, a *Figura 32* mostra o código-fonte necessário para realizar a funcionalidade “*cadastrar turma*” do *IFACADÊMICO*, no contexto da tecnologia jQuery.

```

<!-- Clique do botão de Cadastro -->
$("#cadastrar").click(function(){
    $("#resultadoCadastro").html("<img src=\"http://localhost/ifaacademico/ima

$.post(
    <!-- Endereço do serviço a ser usado -->
    "http://localhost/ifaacademico/servicos/cadastrarTurma.php",

    <!-- Dados que serão enviados -->
    {
        nome : $("#nomeCadastro").val(),
        curso : $("#cursoCadastro").val(),
        capacidade : $("#capacidadeCadastro").val(),
        turno : $("#turnoCadastro").val(),
        anoInicio : $("#anoInicioCadastro").val(),
        mesInicio : $("#mesInicioCadastro").val(),
        diaInicio : $("#diaInicioCadastro").val()
    },

    <!-- Tratamento das informações recebidas do servidor -->
    function(data){
        if(data == "ok"){
            $("#resultadoCadastro").html("<img src=\"http://localhost/ifa
            $("#nomeCadastro").val("");
            $("#capacidadeCadastro").val("");
        }

        else{
            $("#resultadoCadastro").html("<div class=\"tvermelho\">Não fo
        }
    }
});
});

```

Figura 32 - Código jQuery para implementar o cadastro de turmas.

O código jQuery apresentado na *Figura 32*, mostra quão fácil implementar uma funcionalidade de cadastro pode ser. Este código, em sua essência, simplesmente altera um determinado status disponível na interface; utiliza dados a partir de campos existentes na página; envia as informações para o endereço determinado; e, informa o usuário do resultado da interação por meio de informações visuais. Tudo isso, ocorrendo apenas através do clique do botão determinado na funcionalidade.

Outra funcionalidade implementada que retrata muito bem esta interação é a exclusão de um anexo na edição de serviços de envio que a aplicação fornece. A *Figura 33* mostra o layout básico implementado para realizar esta funcionalidade.



Figura 33 - Interface da funcionalidade para visualizar os anexos enviados.

Assim como na funcionalidade para cadastro de turmas, a exclusão do anexo ocorre a partir da troca de dados *cliente-servidor* assíncrona. Logo que o usuário clica com o mouse no ícone de exclusão (✖) a página envia a requisição para que o servidor realize esta operação. A Figura 34 mostra o código jQuery utilizado para realizar esta interação.

```

<!-- Excluir -->
$(".excluir").click(function(){
    $.post(
        <!-- Endereço do serviço a ser usado -->
        "http://localhost/ifacademico/servicos/excluirAnexo.php",

        <!-- Dados que serão enviados -->
        {
            id : this.id
        },

        <!-- Tratamento das informações recebidas do servidor -->
        function(data){
            if(data == "ok"){
                document.location.href = "<?PHP echo $_SERVER['PHP_SELF']; ?>";
            }
        }
    );
});

```

Figura 34 - Código jQuery para implementar a exclusão do anexo.

Como no código jQuery mostrado na Figura 32, a Figura 34 apresenta a enorme simplicidade para realizar interações desta natureza. O código para excluir o anexo é ainda mais simples. Inicialmente, após o clique do item que faz parte da classe *“excluir”*, o código obtém o identificador do item clicado e o envio ao endereço

Esta funcionalidade utiliza da atualização assíncrona de dados para verificar ou atualizar o status dos usuários participantes da conversa. Nesta interface, o status do usuário destinatário pode ser alterado, e, se o usuário emissor não estiver on-line, o sistema pode restringir o acesso do usuário remetente à página.

A implementação desta funcionalidade proporcionou a obtenção de dados do servidor sem a necessidade de submissão de toda a página para realizar uma simples consulta. A estrutura básica para implementar a verificação do status dos usuários participantes da conversa segue a mesma dinâmica descrita na *subseção 5.1.3*, e consiste basicamente em: definir quais dados serão usados, se necessário; o alvo da interação; e, a partir daí definir o que vai acontecer assim que o alvo responda à interação. Para dar melhor entendimento de como a tecnologia atuou nesta funcionalidade, a *Figura 36* mostra o código jQuery para implementá-la no sistema.

```

$(document).ready(function(){
    loop();
    $("#mensagem").ckeditor({uiColor: '#cee0cc', toolbar: [
        { name: 'basicstyles', groups: [ 'basicstyles', 'cleanup' ], item
        { name: 'links', items: [ 'Link' ] },
        { name: 'insert', items: [ 'Image', 'Flash', 'Smiley', 'SpecialCh
        { name: 'styles', items: [ 'Font', 'FontSize' ] },
        { name: 'colors', items: [ 'TextColor' ] },
        { name: 'others', items: [ 'youtube' ] }
    ]
    });
    document.mensagens.location.href = "http://localhost/ifaceademico/servicos
});

var loop = function(){
    document.mensagens.location.href = "http://localhost/ifaceademico/servicos
    setTimeout('loop()', 20000);

    $.post(
        <!-- Endereço do serviço a ser usado -->
        "http://localhost/ifaceademico/servicos/checkOnline.php",

        <!-- Dados que serão enviados -->
        {
            usuario : $("#destinatario").val()
        },

        <!-- Tratamento das informações recebidas do servidor -->
        function(data){
            if(data == "ok"){
                $("#statusRemetente").html("<img src=\"http://localhost/iface
                $("#statusDestinatario").html("<img src=\"http://localhost/ifa
            }

            else{
                if(data == "erro"){
                    $("#statusRemetente").html("<img src=\"http://localhost/i
                    $("#statusDestinatario").html("<img src=\"http://localhos
                }

                else{
                    document.location.href = "http://localhost/ifaceademico/er
                }
            }
        }
    );
}
}

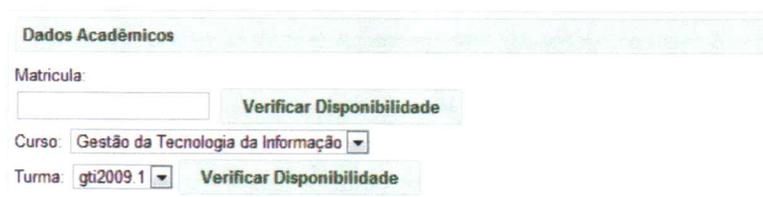
```

Figura 36 - Código jQuery para implementar a verificação do status dos participantes da funcionalidade para realizar conversas.

A partir do código jQuery mostrado na *Figura 36*, entra em questão uma característica que será melhor apresentado na *seção 5.2* deste trabalho: a implementação de funcionalidades complexas que o framework proporciona. O referido código segue etapas tão simples como as etapas descritas anteriormente neste trabalho:

- inicialmente, ele define que a função “loop” será iniciada assim que a página for completamente carregada no navegador;
- após, a estrutura da referida função atualiza o frame de mensagens e define que, a cada 20 segundos, o frame deverá ser atualizado novamente;
- em seguida, para realizar a interação com o servidor, o campo que contem o identificador do usuário de destinatário é enviado para o alvo determinado;
- por fim, se o resultado for verdadeiro, os dois usuários estão on-line. Se o resultado for falso o destinatário está off-line e o remetente está on-line. E se o servidor responder algo diferente, o remetente não está mais on-line no sistema.

Outra situação que retrata muito bem esta interação é a atualização de disciplinas que ocorre na troca de curso, presente na funcionalidade para cadastrar alunos. A *Figura 37* mostra a seção “dados acadêmicos” da referida funcionalidade, onde é possível observar um exemplo da atualização assíncrona de dados no sistema.



Dados Acadêmicos

Matricula:

Curso: ▾

Turma: ▾

Figura 37 – Exemplo de atualização assíncrona de dados: Fragmento “Dados Acadêmicos” da funcionalidade para cadastrar alunos.

Assim como na funcionalidade de troca de mensagens do sistema, a atualização de disciplinas também ocorre a partir da busca assíncrona de dados. Como mostrado na *Figura 37*, logo que o usuário altera o curso no objeto *select*¹⁴, a

¹⁴ Elemento de interface web para escolha de itens.

página envia a requisição para que o servidor busque as turmas referentes ao item selecionado. A *Figura 38*, mostrada na sequência, mostra o código jQuery utilizado para realizar esta interação.

```

<!-- Escolha do curso, busca das turmas referentes a ele -->
$("#cursoCadastro").change(function(){
    $("#turmasCadastro").html("<img src=\"http://localhost/ifacademico/imagen

$.post(
    <!-- Endereço do serviço a ser usado -->
    "http://localhost/ifacademico/servicos/carregarTurmas.php",

    <!-- Dados que serão enviados -->
    {
        curso : $("#cursoCadastro").val()
    },

    <!-- Tratamento das informações recebidas do servidor -->
    function(data){
        if(data == "erro"){
            $("#turmasCadastro").html("<select id=\"turmaCadastro\"><opti
        }

        else{
            data = "Turma: <select id=\"turmaCadastro\">"+data+"</select>
            $("#turmasCadastro").html(data);
        }
    }
});
});

```

Figura 38 - Código jQuery para implementar a busca de turmas cadastradas.

O código jQuery apresentado na *Figura 38* realiza a busca de turmas a partir de um determinado curso. Este código inicialmente altera um determinado status disponível na interface; em seguida, envia o identificador do curso desejado e inicia a requisição dos dados; E, ao final, após a resposta do servidor, ele mostra as disciplinas obtidas a partir do curso no conteúdo da página.

5.2. Análise da adequação da tecnologia à implementação de funcionalidades complexas do sistema IFACADÊMICO

Implementar funcionalidades complexas em um software é uma tarefa árdua para desenvolvedores e projetistas de interface atualmente. Realizar este tipo de

implementação, na maioria das vezes, ou resulta em inúmeras linhas de código ou transforma o código-fonte em um verdadeiro *cubo mágico*¹⁵. Visando descobrir se o framework *jQuery UI* responde bem a esta situação, a presente coleta e análise de dados buscou funcionalidades consideradas complexas existentes no sistema *IFACADÊMICO*.

Esta seção do trabalho irá interpretar como a tecnologia do framework alvo desta pesquisa reage à implementação de funcionalidades complexas, levando em consideração o nível de esforço necessário para desenvolver itens considerados difíceis no desenvolvimento web comum, analisando o quê e como foi feito. Nesta categoria, quatro (04) das funcionalidades analisadas puderam ser classificadas sob este critério, cada uma segundo um ponto de vista diferente do sistema: a escolha de destinatários, a criação de horários acadêmicos, a verificação da disponibilidade de determinados dados no sistema e o chat de mensagens rápidas presente na aplicação.

5.2.1. Selecionar destinatário

A funcionalidade selecionar destinatário é muito interessante. Como é mostrada na *Figura 39*, ela permite que o usuário do sistema pesquise o destinatário a partir de partes do seu nome.

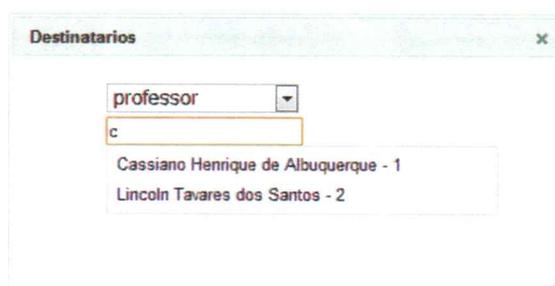


Figura 39 - Interface da funcionalidade para selecionar destinatários.

¹⁵ O *cubo mágico* é um jogo de raciocínio lógico que funciona através da reorganização das cores que formam os seus lados. A comparação com o jogo é pertinente, pois, assim como no jogo, nestes casos, as alterações realizadas modificam praticamente tudo.

Na maioria das vezes, oferecer o nível de usabilidade encontrado na interface mostrada na *Figura 39* não é fácil. Porém, neste caso especificamente, ao analisar o código da funcionalidade (*Figura 40*), é possível afirmar que não foi muito difícil implementá-la no sistema.

```

<!-- Autocomplete professores -->
$("#input#professor").autocomplete ({
  source : function (request, callback){
    var data = { valor : request.term, campo : "nome" };

    $.ajax ({
      url : "http://localhost/ifaacademico/servicos/getListProfessores.
      data : data,
      complete : function (xhr, result){
        if (result != "success") return;

        var response = xhr.responseText;
        var ids = [];

        $(response).find("li id").each (function (){
          ids.push ($(this).text ());
        });

        // atribuição dos ids recebidos como valores do autocomplete
        callback (ids);

        // visualização dos nomes no autocomplete
        var $ul = $("#input#professor").autocomplete ("widget");

        $(response).find ("li nome").each (function (index){
          var nome = $(this).text () || "?";
          $ul.find ("li:eq(" + index +") a").prepend (nome+ " - ");
        });
      }
    });
  },

  open : function (event){
    var $ul = $(this).autocomplete ("widget");
    $ul.css ("width", "300px");
  }
});

```

Figura 40 - Código jQuery para implementar a pesquisa do destinatário.

O código jQuery da *Figura 40*, mostra como o campo para inserção do destinatário se transforma em um campo de pesquisa autocompletável e dinâmico. Essa característica não é somente uma alteração no campo, mas sim, a utilização de mais um widget disponível no framework *jQuery UI*.

No código mostrado na *Figura 40*, a função “*autocomplete*”, simplesmente, adiciona dados ao parâmetro “*source*”. Este parâmetro é a fonte de dados onde o widget pesquisa o que está sendo digitado no momento. Neste caso particularmente, a funcionalidade utiliza-se da função “*ajax*” para transmitir assincronamente os dados para que o servidor realize a pesquisa, e só após a resposta, adiciona os dados ao componente de interface.

5.2.2. Criar horário acadêmico

A funcionalidade “*criar horário*” (*Figura 41*), é, entre todas as funcionalidades, a mais complexa. Além disso, criar essa funcionalidade compreendeu um desafio muito importante para o sistema, pois a dinâmica para realizar a atividade virtualmente deveria seguir padrões similares a dinâmica real existente na coordenação. Para tanto, isso não representa em nada a sua implementação no sistema *IFACADÊMICO*.

HORARIO Nº 1 / 13 AULAS CADASTRADAS
Turma: gti2009.1, 1º Período, noite
Curso: Gestão da Tecnologia da Informação

Aula / Dia	Segunda-Feira	Terça-Feira	Quarta-Feira	Quinta-Feira	Sexta-Feira
Cadastro / Alteração de Aulas					
Disciplina:	<div style="border: 1px solid black; padding: 5px;"> Seleccione... Seleccione... Inglês Instrumental Português Matemática Princípios Organizacionais Lógica de Programação Sistemas Digitais Engenharia e Qualidade de Software Técnicas e Métodos de Pesquisa </div>				
5ª aula 21:15:00 as 22:00:00		Lógica de Programação (Outro Professor)			Princípios Organizacionais (Professor Tal)
6ª aula Sem duração					Princípios Organizacionais (Professor Tal)
			Inglês para da Tal		Matemática (Outro Professor)
					Inglês Instrumental (Professora da Silva Tal)
					Inglês Instrumental (Professora da Silva Tal)

Novo Horário

Figura 41 - Interface da funcionalidade para criar horários acadêmicos.

Ao analisar a funcionalidade “*criar horário acadêmico*”, podemos dividi-la em três (03) interações básicas: a busca do início e término da aula, a busca de professores disponíveis para uma determinada disciplina e o envio das informações para cadastrar as aulas do horário.

No primeiro caso, como mostrado na *Figura 42*, o início e o término da aula são solicitados ao servidor da aplicação através do seu número e turno. Após receber os dados, o código jQuery adiciona-os em campos existentes na página.

```

<!-- Segunda-Feira -->
$("#segunda").click(function(){
    $.post(
        <!-- Endereço do serviço a ser usado -->
        "http://localhost/ifacademico/servicos/carregarInicioAula.php",

        <!-- Dados que serão enviados -->
        {
            aula : this.id,
            turno : $("#turno").val()
        },

        <!-- Tratamento das informações recebidas do servidor -->
        function(data){
            if(data == "erro"){
                $("#inicio").val("");
            }

            else{
                $("#inicio").val(data);
            }
        }
    );

    $.post(
        <!-- Endereço do serviço a ser usado -->
        "http://localhost/ifacademico/servicos/carregarTerminoAula.php",

        <!-- Dados que serão enviados -->
        {
            aula : this.id,
            turno : $("#turno").val()
        },

        <!-- Tratamento das informações recebidas do servidor -->
        function(data){
            if(data == "erro"){
                $("#termino").val("");
            }

            else{
                $("#termino").val(data);
            }
        }
    );

    $("#dia").val("segunda");
    $("#aula").val(this.id);
    $("#cadastroAula").dialog("open");
});

```

Figura 42 - Código jQuery para implementar a busca de início e término de uma determinada aula.

No segundo caso, como mostrado na *Figura 43*, o código jQuery solicita professores disponíveis assim que o usuário escolhe uma disciplina.

```

<!-- Change para carregar professores -->
$("#disciplina").change(function() {
    $("#professores").html("<img src=\"http://localhost/ifacademico/imagens/i

$.post(
    <!-- Endereço do serviço a ser usado -->
    "http://localhost/ifacademico/servicos/carregarProfessores.php",

    <!-- Dados que serão enviados -->
    {
        disciplina : $("#disciplina").val(),
        aula : $("#aula").val(),
        dia : $("#dia").val(),
        horario : $("#horario").val()
    },

    <!-- Tratamento das informações recebidas do servidor -->
    function(data) {
        if(data == "erro"){
            $("#professores").html("<div class=\"tvermelho\">Conflito de
            $("#cadastrarAula").hide();
        }

        else{
            data = "Professor: <select id=\"professor\">"+data+"</select>
            $("#professores").html(data);
            $("#cadastrarAula").show();
        }
    }
});
});

```

Figura 43 - Código jQuery para implementar a busca de professores relacionados a uma determinada disciplina escolhida no horário acadêmico.

No terceiro e último caso (*Figura 44*), assim que o usuário confirma o cadastro, o código jQuery envia os dados para serem armazenados no servidor da aplicação. Neste processo, a tecnologia do framework altera um determinado status da tarefa na interface; utiliza dados a partir de campos existentes na página; envia as informações para o endereço determinado; e, informa o usuário do resultado da interação por meio de informações visuais.

No sistema *IFACADÊMICO* a verificação de dados está presente em algumas funcionalidades de cadastro. Como descrito na *Figura 45*, assim que o usuário clica em “*verificar disponibilidade*”, o sistema pesquisa se o dado informado já existe na base de dados.

Dados de Identificação

Nome:

CPF:

CPF ja cadastrado.

RG:

Figura 45 - Fragmento “Dados de Identificação” do cadastro de alunos: Verificação do CPF.

Tomando esta situação como exemplo, implementar tal recurso com a mesma sofisticação e a partir de arquiteturas tradicionais, desprovidas de recursos sofisticados de interface e baseadas na relação cliente-servidor, torna-se uma tarefa extremamente sacrificante ao desenvolvedor. Essencialmente, partindo do contexto da tecnologia do framework *jQuery UI*, esta implementação é extremamente simples. Para tanto, o código *jQuery* mostrado na *Figura 46* exemplifica muito bem como esta tarefa, considerada complexa em outras metodologias de desenvolvimento, pôde ser implementada no sistema *IFACADÊMICO*.

```

<!-- Clique do botão para verificar a disponibilidade do CPF -->
$("#verificarDisponibilidadeCpf").click(function(){
    $("#statusDisponibilidadeCpf").html("<img src=\"http://localhost/ifacademico/imagens/im
    if($("#cpfCadastro").val() == ""){
        $("#statusDisponibilidadeCpf").html("<div class=\"tamarelo\">Você ainda não dig
    }

    else{
        $.post(
            <!-- Endereço do serviço a ser usado -->
            "http://localhost/ifacademico/servicos/checkDisponibilidadeCpf.php",

            <!-- Dados que serão enviados -->
            {
                cpf : ($("#cpfCadastro").val())
            },

            <!-- Tratamento das informações recebidas do servidor -->
            function(data){
                if(data == "ok"){
                    $("#disponibilidadeCpf").val("sim");
                    $("#statusDisponibilidadeCpf").html("<div class=\"tverdel\">CPF dis
                }

                else{
                    $("#disponibilidadeCpf").val("nao");
                    $("#statusDisponibilidadeCpf").html("<div class=\"tvermelho\">CPF j
                }
            }
        );
    }
});
}
});

```

Figura 46 – Exemplo de código jQuery para implementar a verificação de dados: Verificação de CPF.

O código jQuery representado na *Figura 46*, inicialmente, verifica se o valor do campo do *CPF* está vazio; se o campo estiver vazio, o recurso informa ao usuário que ele ainda não digitou o *CPF*; se o campo não estiver vazio, o código realiza uma interação assíncrona com o servidor para verificar a disponibilidade do dado no sistema.

5.2.4. Chat de mensagens rápidas

Implementar um chat de mensagens rápidas foi um requisito de software levantado para o sistema *IFACADÊMICO*. Segundo a documentação de software do sistema (disponível na sessão *APÊNDICES* deste relatório), este requisito deve ser implementado sob conceitos de usabilidade práticos, dinâmicos e intuitivos.

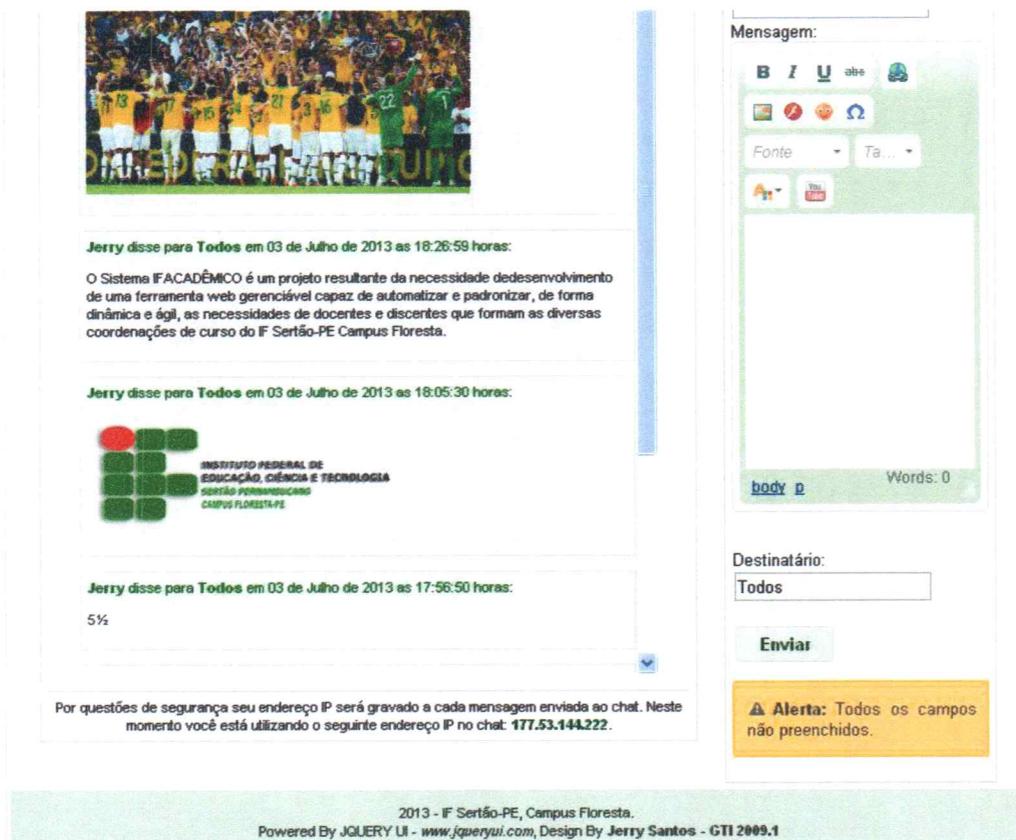


Figura 47 - Interface do chat de mensagens rápidas.

Assim como descreve a *Figura 47*, o usuário que acessa a página inicial do *IFACADÊMICO* pode comunicar-se com outros usuários que também estejam utilizando o recurso no momento. A utilização do chat consiste basicamente em: o usuário informa seu nome, sua mensagem e o destinatário, e, em seguida, confirma o envio; o sistema identifica os usuários e envia e atualiza as mensagens. Analisando a participação do *jQuery UI* na implementação deste chat, é possível dividir a participação do framework em três situações distintas: a atualização dos dados, a adaptação do editor à interface e o envio das mensagens.

Na primeira das três participações analisadas na implementação do chat, assim como mostrado na *Figura 48*, o chat adiciona o editor a interface, definindo a cor e as opções de edição que estarão disponíveis para o usuário.

```

<!-- Editor rich-text, Mensagem -->
$("#mensagem").ckeditor({uiColor: '#cee0cc', toolbar: [
  { name: 'basicstyles', groups: [ 'basicstyles', 'cleanup' ], items: [
  { name: 'links', items: [ 'Link' ] },
  { name: 'insert', items: [ 'Image', 'Flash', 'Smiley', 'SpecialChar'
  { name: 'styles', items: [ 'Font', 'FontSize' ] },
  { name: 'colors', items: [ 'TextColor' ] },
  { name: 'others', items: [ 'youtube' ] }
  ]
});

```

Figura 48 - Criação personalizada do editor no chat de mensagens rápidas.

No segundo caso (Figura 49), o código verifica se os dados estão preenchidos. E, se os campos “mensagem”, “remetente” e “destinatário” estiverem preenchidos, envia assincronamente os dados para cadastrar a mensagem no servidor.

```

<!-- Clique do botão de envio de mensagens -->
$("#enviarMensagem").click(function(){
  if($("#mensagem").val() != "" && $("#remetente").val() != "" && $("#destinatario").val() != "" && $("#ip").val() != ""){
    $("#resultadoEnvioMensagem").html("<img src='\"http://localhost/ifaacademico/servicos/envioMensagens.php\"',
    $.post(
      <!-- Endereço do serviço a ser usado -->
      "http://localhost/ifaacademico/servicos/envioMensagens.php",
      <!-- Dados que serão enviados -->
      {
        remetente : $("#remetente").val(),
        mensagem : $("#mensagem").val(),
        destinatario : $("#destinatario").val(),
        ip : $("#ip").val()
      },
      <!-- Tratamento das informações recebidas do servidor -->
      function(data){
        if(data == "ok"){
          $("#resultadoEnvioMensagem").html("<div class='\"ui-state-highlight ui-state-active\"'>
            $("#mensagem").val(\"");
          document.mensagens.location.href = "http://localhost/ifaacademico/servicos/envioMensagens.php";
        }
        else{
          $("#resultadoEnvioMensagem").html("<div class='\"ui-state-error\"'>
        }
      }
    );
  }
  else{
    $("#resultadoEnvioMensagem").html("<div class='\"ui-state-highlight ui-state-active\"'>
  }
});

```

Figura 49 - Código jQuery para implementar o envio de mensagens do chat.

No terceiro e último caso, assim como mostrado na *Figura 50*, o chat atualiza a visualização das mensagens a cada trinta (30) segundos.

```
$(document).ready(function(){
    loop();
    $('#emailLogin').focus();
});

var loop = function(){
    document.mensagens.location.href = "http://localhost/ifacademico/servicos";
    setTimeout('loop()', 30000);
}
```

Figura 50 - Código jQuery para atualizar o chat.

6. CONCLUSÃO

Esta pesquisa se propôs, em seu objetivo geral, a observar e analisar os vários aspectos existentes no desenvolvimento web a partir do *jQuery UI Framework*. Para que o trabalho não se limitasse somente à teoria, buscou, através da implementação do sistema *IFACADÊMICO*, situações condizentes com a realidade comum neste tipo de desenvolvimento.

Após a implementação da tecnologia em campo, tornou-se possível capturar situações que puderam dar base de sustentação para o estudo sobre o framework na pesquisa. E, através da observação participante realizada no sistema, características da tecnologia do framework foram levantadas e relações de uso e resultado foram estabelecidas.

Tendo como base a análise dos resultados obtidos, foi possível identificar que o *jQuery UI* posiciona o desenvolvimento web em uma nova perspectiva, sendo possível apresentar características evolutivas que compreendem muito bem a atual dinâmica de utilização dos recursos existentes na internet, e que podem ser divididas em quatro (04) conceitos básicos: implementação de interfaces gráficas, utilização do framework, reutilização de software e potencialidade da ferramenta.

Quanto à implementação de interfaces gráficas, é possível evidenciar que a partir da utilização do *jQuery UI* pode se obter uma experiência de utilização moderna, funcional e intuitiva. Esta característica ocorre por meio de uma estrutura pré-definida que promove a interação do usuário com o software sob conceitos de usabilidade e funcionalidade muito semelhantes aos conceitos empregados no design de aplicações para a plataforma desktop. Além disso, a análise de interface mostrou que a tecnologia do framework fornece elementos que dão suporte a possibilidades de criações de software fascinantes.

Quanto à utilização do framework, é possível evidenciar que o mesmo permite que a implementação de recursos e funcionalidades demande menos tempo e

esforço, simplificando a implementação de funcionalidades consideradas complexas e possibilitando a reestruturação do sistema sem modificar o seu comportamento.

Quanto à reutilização de software, foi possível evidenciar que os níveis de reutilização do *jQuery UI* superam bastante os níveis tradicionais para realizar essa técnica de reaproveitamento de software. Esta característica do framework, em suma, abstrai o desenvolvimento e permite reutilizar, desde um simples componente até um aplicativo em sua totalidade.

Quanto à potencialidade da ferramenta, foi possível evidenciar que a estrutura do framework possibilita a realização de projetos surpreendentes. A partir desta categoria, o desenvolvimento web dispensa a utilização de inúmeras tecnologias para criar aplicações com inteligência própria, maior grau de eficiência e custos reduzidos.

O princípio da “*nova perspectiva*” sugere que, uma vez encontrada uma solução para determinado contexto, esta solução deve ser adotada em situações semelhantes. Quando esta pesquisa articula a proposta de que “*o jQuery UI posiciona o desenvolvimento web em uma nova perspectiva*”, fala-se de um desenvolvimento que contempla:

- uma interface gerenciável, que permite a manipulação de cada item de forma isolada, que adiciona interação a visualização gráfica, que permite que o usuário usufrua o máximo do conteúdo disponível;
- uma implementação completa, que engloba todos os aspectos necessários para desenvolver produtos de software web, e que possui recursos de uso descomplicado;
- uma reutilização de software evoluída, que reutiliza itens muito fácil, que reduz o esforço, que possui inúmeros subsídios disponíveis para uso;
- uma potencialidade de uso intangível, que proporciona resultados de softwares expressivos no desenvolvimento web, e que atende a uma aplicação complexa tão bem, quanto atende a uma aplicação simples.

Por fim, apesar dos resultados não poderem ser generalizados, visto que se tratou de um estudo de caso em apenas um único software, pode-se concluir, com base nas as observações realizadas no âmbito deste trabalho, que o *jQuery UI Framework* é uma solução de software comprovada para o desenvolvimento web atual. No entanto é válido ponderar o quanto as características apresentadas neste trabalho atendem os objetivos do software que se pretende desenvolver.

A conclusão final é que através da aplicação de um framework em campo, foi possível perceber na prática como o desenvolvimento web pode ser útil para gerar aplicações atualmente. Onde, mesmo em situações extrema do desenvolvimento, o framework respondeu satisfatoriamente a esta pesquisa.

6.1. Trabalhos futuros

Visando futuras extensões, este trabalho oferece algumas sugestões para que sejam realizados futuramente:

- Versão do sistema *IFACADÊMICO* para dispositivos móveis, através do jQuery Mobile Framework: abordar o desenvolvimento de uma versão do sistema *IFACADÊMICO*, através do framework jQuery Mobile, que possibilite o acesso ao sistema por meio de dispositivos móveis como celulares, palmtops, tablets, etc.
- Desenvolvimento de plug-ins jQuery: abordar a experiência de desenvolvimento de plug-ins jQuery específicos para o sistema *IFACADÊMICO*, podendo publicá-los na comunidade jQuery através do jQuery Plugin Registry, ou, até, comercializá-los.
- Desenvolvimento de aplicações multimídia com jQuery UI: abordar o desenvolvimento jQuery UI, com ênfase em efeitos, widgets e plug-ins para a criação de aplicações altamente interativas na web. Se possível, por meio da linguagem de marcação HTML5, visando tirar proveito de todas as suas ferramentas nativas, funcionalidades para acessibilidade e semântica.

- Testes do sistema *IFACADÊMICO*: realizar testes mais rigorosos e abrangentes.

REFERÊNCIAS

- ABOUT. **What is Dynamic HTML (DHTML)?**. Disponível em: <<http://webdesign.about.com/od/dhtml/a/aa030298.htm>>. Acesso em: 20 jul. 2013.
- ALVES, Paulo; PIRES, José Adriano. **A usabilidade em software educativo: princípios e técnicas**. In: IV Simpósio Internacional de Informática Educativa. Vigo. VII Taller Internacional de Software Educativo. 2002. Disponível em: <<https://bibliotecadigital.ipb.pt/handle/10198/1950>> Acesso em: 20 jun. 2013.
- BURD, Leo. **Desenvolvimento de software para atividades educacionais**. Campinas, UNICAMP, 1999.
- CNET. **John_Resig-1-2.jpg**. 2010. Altura: 363 pixels. Largura: 350 pixels. 57 Kb. Formato JPEG. Disponível em: <http://news.cnet.com/8301-30685_3-20018599-264.html>. Acesso em: 20 jul. 2013.
- CKEDITOR. **About**. Disponível em: <<http://ckeditor.com/about>>. Acesso em: 8 jul. 2013.
- DEITEL, Paul J.; DEITEL, Harvey M. **Ajax, Rich Internet Applications e desenvolvimento Web para programadores**. São Paulo: Pearson Prentice Hall, 2008.
- FAYAD, Mohamed. **Introduction to the computing surveys' electronic symposium on object-oriented application frameworks**. ACM Comput. Surv., 2000.
- FAYAD, Mohamed; SCHMIDT, Douglas C. **Object-oriented application frameworks**. ACM. v. 40, n. 10, 1997. Disponível em: <<http://www.cse.wustl.edu/~schmidt/CACM-frameworks.html>>. Acesso em: 12 jun. 2013.
- FILHO, Antonio Mendes da Silva. **Usabilidade de Software: A Importância da Usabilidade no Desenvolvimento de Sistemas Interativos**. Engenharia de Software Magazine. 5 ed. Devmedia, 2008.
- MICHAELIS. **Significado de "framework"**. Disponível em: <http://www.michaelis.uol.com.br/moderno/ingles/definicao/ingles-portugues/framework%20_451666.html>. Acesso em: 25 jun. 2013.

GIMENES, Itana Maria de Souza; HUZITA, Elisa Hatsue Moriya. **Desenvolvimento baseado em componentes: conceitos e técnicas**. Rio de Janeiro, Ciência Moderna, 2005.

GOMES, Alex Sandro; PADOVANI, Stephania. **Usabilidade no ciclo de desenvolvimento de software educativo**. In: Simpósio Brasileiro de Informática na Educação SBIE 2005, 2005, Juiz de Fora. Anais do Simpósio Brasileiro de Informática na Educação SBIE 2005, v. 1, 2005. Disponível em: <http://www.cin.ufpe.br/~asg/publications/files/gomes_padovani_mini_curso_sbie_2005.pdf> Acesso em: 20 jun. 2013.

GOMES, André Luiz Pereira. **FLEXFRAME: Framework para desenvolvimento de sistemas de informação flexíveis, que parametrizam regras de negócios**. Fortaleza, Universidade de Fortaleza, 2002.

GOOGLE TRENDS. **Pesquisa na web do Google interesse: jquery dhtml**. 2013. Disponível em: <<http://www.google.com.br/trends/explore?#q=jquery%2C%20dhtml&cmpt=q>>. Acesso em: 21 jul. 2013.

_____. **Pesquisa na web do Google interesse: jquery ajax**. 2013. Disponível em: <<http://www.google.com.br/trends/explore?#q=jquery%2C%20ajax&cmpt=q>>. Acesso em: 21 jul. 2013.

GOVONI, Darren. **Java Application Frameworks**. New York, Wiley, 1999.

JQUERY. **Write less, do more**. Disponível em: <<http://www.jquery.com>>. Acesso em: 21 jul. 2013.

JQUERY UI. **JQuery User Interface**. Disponível em: <<http://www.jqueryui.com>>. Acesso em: 21 jul. 2013.

KRIFY. **Web-frameworks.png**. 2013. Altura: 232 pixels. Largura: 634 pixels. 105 Kb. Formato PNG. Disponível em: <<http://www.krify.com/solutions/web-application-development/web-application-frameworks>>. Acesso em: 20 jul. 2013.

LÉVY, Pierre. **O que é o virtual?**. São Paulo, Editora 34, 1996.

MAGDALENA, Beatriz Corso; COSTA, Iris Elisabeth Tempel. **Internet em sala de aula: com a palavra, os professores**. Porto Alegre, Artmed, 2003.

MATTSSON, Michael. **Evolution and Composition of Object-Oriented Frameworks**. Sweden, Karlskrona, 2000.

MELLO, Emerson Ribeiro de. et al. **Segurança em Serviços Web**. Minicursos do SBSeg 2006. Porto Alegre, Sociedade Brasileira de Computação, 2006.

MORAN, José Manuel. **Como utilizar a Internet na educação**. Ci. Inf. n.2, maio/ago 1997. Disponível em: <www.scielo.br/scielo.php?pid=s0100-19651997000200006&script=sci_arttext>. Acesso em: 8 jun. 2013.

NEVES, José Luis. **Pesquisa qualitativa: características, usos e possibilidades**. Caderno de pesquisas em administração FEA-USP. São Paulo, v. 1, n. 3, 1996.

NIEDERAUER, Juliano. **Web Interativa com Ajax e PHP**. São Paulo, Novatec Editora, 2007.

NIELSEN, Jakob; LORANGER, Hoa. **Usabilidade na Web**. Rio de Janeiro, Elsevier, 2007.

OLIVEIRA, Cristiano Lessa de. **Um apanhado teórico-conceitual sobre a pesquisa qualitativa: tipos, técnicas e características**. TRAVESSIAS. v. 2, n. 3, 2008. Disponível em: <http://www.unisc.br/portal/upload/com_arquivo/um_apanhado_teorico_conceitual_sobre_a_pesquisa_qualitativa_tipos_tecnicas_e_caracteristicas.pdf>. Acesso em: 26 jun. 2013.

SILVA, Maurício Samy. **JQuery UI: componentes de interface rica para suas aplicações web**. São Paulo, Novatec Editora, 2012.

_____. **JQuery: A Biblioteca do Programador JavaScript**. 2 ed. São Paulo, Novatec Editora, 2010.

SOMMERVILLE, Ian. **Engenharia de software**. 8 ed. São Paulo, Pearson Addison-Wesley, 2007.

SOUZA, Francisco. **Uso de frameworks para desenvolvimento web e mitos que já deveriam ter desaparecido**. 2010. Disponível em: <<http://www.franciscosouza.com.br/2010/01/22/uso-de-frameworks-para-desenvolvimento-web-e-mitos-que-ja-deveriam-ter-desaparecido/>>. Acesso em: 21 jun. 2013.

SPINOLA, Rodrigo Oliveira; SPINOLA, Eduardo Oliveira. **Especificação de requisitos com casos de uso**. Engenharia de Software Magazine. 5 ed. Devmedia, 2008.

TACIN, Rodolfo Vendrasco; ZORZO, Sergio Donizetti. **Usabilidade de sistemas web com o uso de frameworks para aplicações ricas para a internet: um estudo de caso**. Revista TIS. v. 1, n. 1, 2012. Disponível em:

<<http://revistatis.dc.ufscar.br/index.php/revista/article/view/14/18>>. Acesso em: 21 jun. 2013.

TECMUNDO. **O que é WYSIWYG?**. Disponível em: <<http://www.tecmundo.com.br/institucional/2057-o-que-e-wysiwyg-.htm>>. Acesso em: 15 jul. 2013.

TORRES, Elisabeth Fátima; MAZZONI, Alberto Angel. **Conteúdos digitais multimídia: o foco na usabilidade e acessibilidade**. Ci. Inf., Brasília, v. 33, n. 2, p. 152-160, maio/ago. 2004. Disponível em: <http://www.scielo.br/scielo.php?pid=S0100-19652004000200016&script=sci_arttext> Acesso em: 20 jul. 2013.

VALENTINI, Carla Beatris. **Tecendo e aprendendo: redes sociocognitivas e autopoieticas em ambientes virtuais de aprendizagem**. Porto Alegre, UFRGS, 2003.

VIGORENA, Débora Andrea Liessem; BATTISTI, Patrícia Stafusa Sala. **Procedimentos de coleta de dados em trabalhos de conclusão do curso de Secretariado Executivo da Unioeste/PR**. Passo Fundo, Revista do Secretariado Executivo. n. 7, p. 95-111, 2011.

W3C. **Document Object Model**. Disponível em: <<http://www.w3.org/dom/>>. Acesso em: 25 jun. 2013.

W3TECHS. **Usage and market share of JavaScript libraries for websites**. Disponível em: <http://w3techs.com/technologies/overview/javascript_library/all>. Acesso em: 25 jun. 2013.

GLOSSÁRIO

AJAX: Asynchronous JavaScript and XML. Uso sistemático de JavaScript e XML (entre outras tecnologias) para tornar o navegador mais interativo com o usuário, utilizando-se de solicitações assíncronas de informações.

API: Application Programming Interface. Conjunto de instruções e padrões de programação disponíveis para que os desenvolvedores possam acessar e reutilizar características e comportamentos de softwares, linguagens, e demais tecnologias.

CSS: Cascading Style Sheets. Linguagem de folhas de estilo que descreve a apresentação das páginas na world wide web.

DHTML: Dynamic HyperText Markup Language. Prática de desenvolvimento que visa tornar as páginas web mais interativas e dinâmicas, utilizando-se de HTML, CSS e JavaScript.

DOM: Document Object Model. Especificação da W3C onde pode-se dinamicamente alterar e editar a estrutura, conteúdo e estilo de um documento eletrônico.

GPL License: General Public License. Licença copyleft para softwares e outros tipos de obras.

HTML: HyperText Markup Language. Linguagem de marcação de texto que descreve a estrutura das páginas na world wide web.

Javascript: Linguagem de programação client-side direcionada ao desenvolvimento de aplicações web, que funciona através de scripts incorporados no código HTML de uma página.

jQuery: Biblioteca JavaScript livre e de código aberto.

jQuery UI: Framework para desenvolvimento de aplicações web a partir da biblioteca jQuery.

MIT License: Massachusetts Institute of Technology License. Licença para softwares originalmente desenvolvidos no Massachusetts Institute of Technology.

MySQL: Gerenciador de banco de dados relacional de código aberto.

PHP: Hypertext Preprocessor. Linguagem de programação server-side direcionada ao desenvolvimento de aplicações web, que funciona através de scripts incorporados no código HTML de uma página.

RIA: Rich Internet Application. Aplicações web que oferecem a sensibilidade, recursos e funcionalidades que se aproximam dos padrões utilizados em aplicações desktop.

W3C: World Wide Web Consortium. Comunidade internacional que desenvolve padrões abertos para garantir o crescimento em longo prazo da web.

WYSIWYG: Termo usado para classificar ferramentas de edição e desenvolvimento que permitem visualizar, em tempo real, exatamente aquilo que será publicado ou impresso.

XHTML: Extensible HyperText Markup Language. Variante da linguagem HTML que usa a sintaxe XML.

XML: Extensible Markup Language. Formato baseado em texto para representar informações estruturadas de dados.

APÊNDICES

APÊNDICE A – DOCUMENTAÇÃO DE SOFTWARE VERSÃO 1.6 DO SISTEMA *IFACADÊMICO*

INSTITUTO FEDERAL DE EDUCAÇÃO CIÊNCIA E TECNOLOGIA DO SERTÃO PERNAMBUCANO
TRABALHO DE CONCLUSÃO DE CURSO - GESTÃO SUPERIOR EM GESTÃO DA TECNOLOGIA
DA INFORMAÇÃO
PROFESSOR ORIENTADOR: CASSIANO HENRIQUE DE ALBUQUERQUE

DOCUMENTAÇÃO DE SOFTWARE Sistema IFACADÊMICO

VERSÃO 1.6 - AGOSTO DE 2013

Equipe Responsável pela Elaboração

Antônio Jerry Sá Santos
Cassiano Henrique de Albuquerque
Lincoln Tavares dos Santos

Público Alvo

Este manual destina-se aos desenvolvedores e responsáveis pelo sistema acadêmico destinado às coordenações de curso do Instituto Federal de Educação Ciência e Tecnologia do Sertão Pernambucano, Campus Floresta.

Versão 1.6 – Floresta-PE, Agosto de 2013

Dúvidas, críticas e sugestões devem ser encaminhadas por escrito para o seguinte endereço postal:

suporte@ifacademico.com

Ou para o seguinte endereço eletrônico:

antoniojerrysantos@gmail.com

Recomendamos que o assunto seja identificado com o título desta obra. Alertamos ainda para a importância de se identificar o endereço e o nome completos do remetente para que seja possível o envio de respostas.

Windows e Microsoft Word são marcas registradas da Microsoft Corporation

HISTÓRICO DE ALTERAÇÕES

Data	Versão	Descrição	Autor
19/12/2012	1.0	Fechamento do escopo com definição de todos os requisitos a serem implementados no sistema.	Antônio Jerry Sá Santos
20/01/2013	1.1	Adição do diagrama e das especificações de caso de uso, dos diagramas de sequência e do modelo conceitual do sistema.	Antônio Jerry Sá Santos
31/01/2013	1.2	Adição dos itens 13.2, 13.3, 13.4, 13.5 aos requisitos não-funcionais do item [RF013] dos requisitos funcionais; Criação dos requisitos [RF014] e [RF015].	Antônio Jerry Sá Santos
04/02/2013	1.3	Adição do diagrama de classes; Alteração no modelo conceitual do sistema e no diagrama de sequência 6.3.	Antônio Jerry Sá Santos
18/07/2013	1.4	Modificações nos itens RF009, RF010, RF012, RF013, UC003, UC005 e UC007.	Antônio Jerry Sá Santos
12/08/2013	1.5	Modificação no diagrama de casos de uso.	Antônio Jerry Sá Santos
20/08/2013	1.6	Adição dos resultados dos testes do sistema.	Antônio Jerry Sá Santos

SUMÁRIO

1.	INTRODUÇÃO.....	5
1.1.	CONVENÇÕES, TERMOS E ABREVIACÕES.....	5
1.1.1.	Identificação dos Requisitos.....	5
1.1.2.	Prioridade dos Requisitos.....	6
2.	DESCRIÇÃO GERAL DO SISTEMA.....	7
2.1.	ESCOPO DO PRODUTO.....	7
2.2.	FUNÇÕES.....	7
2.3.	RESTRIÇÕES GERAIS.....	8
2.4.	BENEFÍCIOS.....	8
2.5.	DESCRIÇÃO DOS USUÁRIOS.....	9
3.	REQUISITOS FUNCIONAIS.....	10
4.	REQUISITOS NÃO-FUNCIONAIS DO SISTEMA.....	20
5.	CASOS DE USO.....	21
5.1.	DIAGRAMA E RELAÇÃO DE CASOS DE USO.....	21
6.	ESPECIFICAÇÃO DOS CASOS DE USO.....	23
7.	DIAGRAMAS DE SEQUÊNCIA DO SISTEMA.....	30
7.1.	CRIAÇÃO DE HORÁRIOS ACADÊMICOS.....	30
7.2.	ATENDIMENTO DE SOLICITAÇÕES.....	30
7.3.	ENVIO DE SOLICITAÇÕES.....	31
7.4.	ENVIO DE MENSAGENS.....	32
8.	MODELAGEM DO SISTEMA	33
8.1.	MODELO CONCEITUAL.....	33
8.2.	DIAGRAMA DE CLASSES.....	34
9.	TESTES.....	35

1. INTRODUÇÃO

Este documento especifica o sistema acadêmico desenvolvido para as coordenações dos cursos do Instituto Federal do Sertão Pernambucano - Campus Floresta, fornecendo informações necessárias para o projeto e implementação, assim como para a realização dos testes e homologação do sistema.

1.1. CONVENÇÕES, TERMOS E ABREVIACÕES

A correta interpretação deste documento exige o conhecimento de algumas convenções e termos específicos, que são descritos a seguir.

1.1.1. Identificação dos Requisitos

Por convenção, a referência a requisitos é feita através do nome da subseção onde eles estão descritos, seguido do identificador do requisito, de acordo com o esquema abaixo:

[nome da subseção.identificador do requisito]

Por exemplo, o requisito [Autenticação de Usuário.RF001] está descrito em uma subseção chamada "Autenticação de Usuário", em um bloco identificado pelo número [RF001].

1.1.2. Prioridades dos Requisitos

Para estabelecer a prioridade dos requisitos foram adotadas as denominações "essencial", "importante" e "desejável".

- **Essencial** - é o requisito sem o qual o sistema não entra em funcionamento. Requisitos essenciais são requisitos imprescindíveis, que têm que ser implementados impreterivelmente.
- **Importante** - é o requisito sem o qual o sistema entra em funcionamento, mas de forma não satisfatória. Requisitos importantes devem ser implementados, mas, se não forem, o sistema poderá ser implantado e usado mesmo assim.
- **Desejável** - é o requisito que não compromete as funcionalidades básicas do sistema, isto é, o sistema pode funcionar de forma satisfatória sem ele. Requisitos desejáveis são requisitos que podem ser deixados para versões posteriores do sistema, caso não haja tempo hábil para implementá-los na versão que está sendo especificada.

2. DESCRIÇÃO GERAL DO SISTEMA

O Sistema IFACADÊMICO é um projeto resultante da necessidade de desenvolvimento de uma ferramenta web gerenciável capaz de automatizar e padronizar, de forma dinâmica e ágil, as necessidades de docentes e discentes que formam as diversas coordenações de curso do IF Sertão-PE - Campus Floresta.

2.1. ESCOPO DO PRODUTO

Após entrevista realizada com responsáveis na coordenação do curso de Gestão da Tecnologia da Informação do instituto, a situação atual nos revelou que a instituição já dispõe de um sistema escolar, que ele não atende as necessidades específicas do setor e que toda a tecnologia disponível atualmente ainda não é bem aproveitada.

Pensando nisso, este projeto visa:

- Trazer maior precisão, controle e segurança à forma como os problemas são resolvidos;
- Minimizar o considerável acúmulo de documentos físicos atual;
- Fornecer acesso seguro e autenticado com disponibilidade interna e externa à instituição.

2.2. FUNÇÕES

Este sistema tem por principal objetivo fornecer melhor comunicação entre seus usuários finais. Por meio de um ambiente simples e ótimos níveis de segurança, ele irá proporcionar as seguintes funções aos seus usuários:

- Troca de mensagens;
- Compartilhamento de avisos e comunicados;
- Envio e acompanhamento de solicitações;
- Consulta de horários, disciplinas e professores;
- Envio e compartilhamento de arquivos.

2.3. RESTRIÇÕES GERAIS

Todas as informações cadastradas são extremamente restritas aos usuários do sistema.

2.4. BENEFÍCIOS

O sistema *IFACADÊMICO* se propõe a beneficiar docentes, discentes e coordenação da referida instituição de ensino, contribuindo de forma significativa para a melhoria do ambiente escolar, aumentando a comunicação e produtividade. Além disso, irá oferecer também:

- Garantia na cobrança de documentos e solicitações;
- Diminuição no prazo de resposta da coordenação;
- Acompanhamento de cursos, alunos e professores virtualmente;
- Incentivo ao uso de novas tecnologias no Campus Floresta do IF Sertão-PE;
- Ambiente virtual seguro e confiável.

2.5. DESCRIÇÃO DOS USUÁRIOS

Este sistema é inicialmente destinado as coordenações de curso do instituto federal aqui mencionado. As permissões adotadas para os usuários do sistema seguem a dinâmica atual descrita pela coordenação do curso de Gestão da Tecnologia da Informação mediante entrevista:

- **Aluno** - Usuário com acesso em nível de consulta. Possibilidade de envio e acompanhamento de solicitações, mensagens e documentos, visualização de horários e comunicados referentes à sua turma.
- **Representante de classe** - Aluno mediador entre a turma e o curso ao qual pertence. Possibilidade de envio e acompanhamento de solicitações, mensagens e documentos, visualização de horários acadêmicos, criação e visualização de comunicados.
- **Professor** - Mediador entre sua disciplina e o curso ao qual ela pertence. Possibilidade de envio e acompanhamento de solicitações, mensagens e documentos, visualização de horários referentes as turmas que atua, criação e visualização de comunicados.
- **Coordenação** - Usuário com poder de decisão no sistema. Responsável direto pelo atendimento de solicitações, montagem de horários e criação de comunicados, envio de documentos e mensagens, consulta de turmas, alunos e professores cadastrados.
- **Administrador do sistema** - Super-usuário. Responsável direto pela manutenção de tudo que é ou será cadastrado. Poder de inserir, visualizar, alterar ou excluir qualquer coisa na base de dados do sistema.

3. REQUISITOS FUNCIONAIS

Requisitos são descrições dos principais recursos de um produto de software, seu fluxo de informações, comportamento e atributos. Um requisito de software fornece uma estrutura básica para o desenvolvimento de um software. Os requisitos funcionais deste sistema estão organizados com base nas principais funcionalidades do sistema apresentadas mediante entrevista com responsáveis.

[RF001] ATENTICAR USUÁRIOS

O sistema deverá identificar cada usuário cadastrado, assim como seu nível de acesso e participação, por intermédio de dados únicos verificáveis fornecidos no login. Adaptar suas funcionalidades ao tipo de acesso identificado e restringir acessos indevidos.

Prioridade: Essencial Importante Desejável

Entradas e pré condições: Os dados de acesso fornecidos pelo usuário deverão ser válidos, cadastrados no sistema.

Saídas e pós condições: Se a busca no banco de dados for bem sucedida o sistema redireciona o usuário à conta correspondente à sua categoria.

Requisitos Não-Funcionais

Nome	Restrição	Categoria	Desejável	Importante	Permanente
1.1. Identificação dos usuários	Cada usuário deverá possuir email e senha para realizar login no sistema.	Segurança	()	(x)	()
1.2. Dados do login	Cada login bem sucedido deverá ser gravado como acesso. Cada acesso deverá possuir um status, bem como, conter o email do usuário, sua categoria, a data e a hora de início da sessão.	Segurança	()	(x)	()

[RF002] ADMINISTRAÇÃO DO SISTEMA

O sistema deverá possibilitar o cadastro, alteração e exclusão de tudo que é cadastrado na base de dados.

Prioridade: Essencial Importante Desejável

Entradas e pré condições: Usuário administrador autenticado no sistema.

Saídas e pós condições: Ação é realizada na base de dados.

Requisitos Não-Funcionais

Nome	Restrição	Categoria	Desejável	Importante	Permanente
2.1. Alertas	O sistema deverá alertar o administrador sempre antes de realizar uma ação na base de dados.	Segurança	()	()	(x)
2.2. Sessão de login	O login do administrador deve expirar em 5 minutos inativos no sistema.	Segurança	(x)	()	()

[RF003] GERENCIAMENTO DA PRÓPRIA CONTA DE USUÁRIO

O sistema deverá possibilitar que o próprio usuário altere ou exclua os seus dados cadastrados.

Prioridade: Essencial Importante Desejável

Entradas e pré condições: Usuário autenticado no sistema.

Saídas e pós condições: Os dados serão alterados na base de dados.

Requisitos Não-Funcionais

Nome	Restrição	Categoria	Desejável	Importante	Permanente
------	-----------	-----------	-----------	------------	------------

3.1. Alteração	O sistema deverá questionar se o usuário deseja realmente alertar seus dados cadastrais.	Segurança	()	()	(x)
3.2. Alteração de senha	O sistema deverá solicitar a senha antiga para que a alteração seja realizada.	Segurança	()	()	(x)
3.3. Exclusão de conta	O sistema deverá alertar o usuário e solicitar a senha de acesso novamente para a exclusão da conta.	Segurança	()	()	(x)

[RF004] RECUPERAÇÃO DE SENHA

O sistema deverá possibilitar que o próprio usuário altere sua senha em caso de esquecimento.

Prioridade: Essencial Importante Desejável

Entradas e pré condições: Usuário comprova sua identidade no sistema.

Saídas e pós condições: O usuário cadastra uma nova senha no seu cadastro.

Requisitos Não-Funcionais

Nome	Restrição	Categoria	Desejável	Importante	Permanente
4.1. Localização da recuperação de senha no sistema	A interface para recuperação de senha deverá ser implementada na página inicial do sistema.	Interface	()	(x)	()
4.2. Identificação do usuário	O usuário deverá informar seu email de acesso e o código de recuperação da sua conta.	Segurança	()	(x)	()

[RF005] CRIAR HORÁRIO ACADÊMICO

O sistema deverá fornecer mecanismos de cadastro dos horários acadêmicos da coordenação.

Prioridade: Essencial Importante Desejável

Entradas e pré condições: Usuário responsável autenticado no sistema.

Saídas e pós condições: Após o preenchimento correto dos dados, o horário será salvo na base de dados.

Requisitos Não-Funcionais

Nome	Restrição	Categoria	Desejável	Importante	Permanente
5.1. Acesso	A função só será desempenhada por usuários com perfil da coordenação.	Segurança	()	(x)	()
5.2. Montagem do horário	A montagem do horário deverá manter-se fiel a atividade original, ser rápida e descomplicada.	Interface	()	(x)	()

[RF006] VISUALIZAR HORÁRIO ACADÊMICO

O sistema deverá fornecer mecanismos de visualização de cada horário cadastrado.

Prioridade: Essencial Importante Desejável

Entradas e pré condições: Usuário solicitante vinculado com a turma a qual o horário pertence.

Saídas e pós condições: Se o horário existir, a visualização é carregada para o usuário.

Requisitos Não-Funcionais

Nome	Restrição	Categoria	Desejável	Importante	Permanente
6.1. Acesso	Alunos, professores e representantes de classe só poderão visualizar horários vinculados às suas turmas.	Segurança	()	(x)	()
6.2. Layout da Visualização	A visualização do horário deverá ser auto-explicativa e clara o máximo possível.	Interface	()	(x)	()

[RF007] CRIAR COMUNICADO

O sistema deverá fornecer mecanismos para criação de novos comunicados de forma simples e prática.

Prioridade: Essencial Importante Desejável

Entradas e pré condições: Usuário responsável autenticado no sistema.

Saídas e pós condições: Após o preenchimento correto dos dados, o comunicado será salvo na base de dados.

Requisitos Não-Funcionais

Nome	Restrição	Categoria	Desejável	Importante	Permanente
7.1. Escolha dos destinatários	O sistema deve fornecer mecanismos de pesquisa para a escolha dos destinatários.	Usabilidade	(x)	()	()
7.2. Notificação do envio	Após a criação do comunicado, os usuários vinculados a ele deverão ser notificados.	Usabilidade	()	(x)	()

[RF008] VISUALIZAR COMUNICADO

O sistema deverá fornecer mecanismos de visualização de cada comunicado cadastrado.

Prioridade: Essencial Importante Desejável

Entradas e pré condições: Usuário solicitante vinculado ao comunicado.

Saídas e pós condições: Após a busca das informações, a visualização é carregada para o usuário.

Requisitos Não-Funcionais

Nome	Restrição	Categoria	Desejável	Importante	Permanente
8.1. Acesso	O comunicado só poderá ser visualizado por usuários vinculados a ele.	Segurança	()	()	(x)

[RF009] CRIAR SOLICITAÇÃO

O sistema deverá fornecer mecanismos que possibilitem o envio de solicitações entre os usuários.

Prioridade: Essencial Importante Desejável

Entradas e pré condições: Usuário emissor autenticado no sistema.

Saídas e pós condições: Após o preenchimento correto dos dados, a solicitação será salva na base de dados.

Requisitos Não-Funcionais

Nome	Restrição	Categoria	Desejável	Importante	Permanente
9.1. Notificação	Cada solicitação	Usabilidade	()	(x)	()

do envio	cadastrada deve ser apresentada via notificação ao destinatário.				
9.2. Prazo definido	Cada solicitação enviada possuirá um prazo médio de até 5 dias úteis para resposta.	Funcionalidade	()	(x)	()

[RF010] ATENDER SOLICITAÇÃO

O sistema deverá fornecer mecanismos que possibilitem o atendimento das solicitações enviadas à coordenação do referido curso.

Prioridade: Essencial Importante Desejável

Entradas e pré condições: Usuário responsável autenticado no sistema.

Saídas e pós condições: O usuário emissor é notificado através de feedbacks referentes ao atendimento.

Requisitos Não-Funcionais

Nome	Restrição	Categoria	Desejável	Importante	Permanente
10.1. Alerta de atraso	Cada prazo descumprido resultará em alertas de cobrança do sistema.	Funcionalidade	()	(x)	()
10.2. Acompanhamento da solicitação	O usuário emissor terá notificações sobre a solicitação enquanto ela não for completamente resolvida.	Funcionalidade	(x)	()	()

[RF011] ENVIAR MENSAGEM

O sistema deverá possibilitar o envio de mensagens entre os usuários.

Prioridade: Essencial Importante Desejável

Entradas e pré condições: Usuário emissor autenticado no sistema.

Saídas e pós condições: A mensagem é salva na base de dados, o destinatário é notificado da mensagem.

Requisitos Não-Funcionais

Nome	Restrição	Categoria	Desejável	Importante	Permanente
11.1. Escolha dos destinatários	O sistema deve fornecer mecanismos de pesquisa para a escolha dos destinatários.	Usabilidade	(x)	()	()
11.2. Notificação do envio	Após o envio da mensagem, os usuários vinculados a ele deverão ser notificados.	Usabilidade	()	(x)	()

[RF012] ANEXAR ARQUIVO

O sistema deverá possibilitar que o usuário anexe arquivos durante o envio de documentos no sistema.

Prioridade: Essencial Importante Desejável

Entradas e pré condições: Arquivo compatível às especificações de envio.

Saídas e pós condições: As informações do arquivo são salvas na base de dados, o arquivo enviado é armazenado.

Requisitos Não-Funcionais

Nome	Restrição	Categoria	Desejável	Importante	Permanente
12.1. Número máximo de arquivos anexados ao envio	Cada envio de documento poderá ter no máximo 10 anexos.	Funcionalidade	(x)	()	()
12.2. Tamanho de arquivo máximo permitido por envio	Cada arquivo anexado a um envio de documento deverá ter no máximo 2MB de tamanho.	Especificação	()	(x)	()
12.3. Extensões de arquivo aceitas no envio	Cada arquivo anexado a um envio de documento deverá ter uma das seguintes extensões de arquivo: pdf, doc, docx, xls, xlsx, ppt, pptx, odt, ods, odp, jpg, jpeg, png, gif, XML.	Segurança	(x)	()	()

Obs.: As extensões associadas ao requisito não-funcional 12.3 foram selecionadas por critérios de segurança tanto para o sistema como para o usuário. A troca de arquivos entre usuários na internet é considerada atualmente uma das maiores causas de disseminação de de virus e spywares.

[RF013] CHAT

O sistema deverá apresentar um chat de mensagens rápidas.

Prioridade: Essencial Importante Desejável

Requisitos Não-Funcionais

Nome	Restrição	Categoria	Desejável	Importante	Permanente
13.1. Características do chat	O sistema deverá fornecer um chat de fácil acesso e utilização.	Funcionalidade	()	(x)	()
13.3. Atualização	O chat deverá ser atualizado a cada 30 segundos.	Funcionalidade	(x)	()	()
13.4. Apresentação	O chat deverá carregar no máximo 40 mensagens por	Usabilidade	(x)	()	()

das mensagens	atualização.				
13.5. Identificação e segurança	Cada mensagem enviada deverá gravar o endereço IP utilizado para o envio.	Segurança	()	(x)	()

[RF014] ADMINISTRAÇÃO DO CHAT

O sistema deverá possibilitar o gerenciamento das mensagens enviadas no chat, bem como todos os responsáveis pelos envios.

Prioridade: Essencial Importante Desejável

Requisitos Não-Funcionais

Nome	Restrição	Categoria	Desejável	Importante	Permanente
14.1. Bloqueio de emissor	O sistema deverá fornecer mecanismos que possibilitem o bloqueio de utilizadores inconvenientes.	Segurança	()	(x)	()

[RF015] PAINEL INFORMATIVO

O sistema deverá fornecer mecanismos que possibilitem que a coordenação do curso crie informativos, tal quais notícias, para seus alunos e professores.

Prioridade: Essencial Importante Desejável

Requisitos Não-Funcionais

Nome	Restrição	Categoria	Desejável	Importante	Permanente
			()	()	()

4. REQUISITOS NÃO-FUNCIONAIS DO SISTEMA

[RNF001] TECNOLOGIAS UTILIZADAS

O desenvolvimento deste projeto deverá utilizar o *jQuery UI* como interface da aplicação, a linguagem *PHP* no processamento do servidor, e a base de dados *MySQL* para armazenamento dos dados.

Prioridade: Essencial Importante Desejável

[RNF002] REQUISITOS MÍNIMOS

O sistema *IFACADÊMICO* é melhor visualizado com resolução de vídeo de 1024 x 768 ou 1280 x 800. A versão atual do sistema só é 100% compatível com o navegador web Google Chrome. A velocidade de conexão mínima recomendada para utilização deste software é de 300Kb.

Prioridade: Essencial Importante Desejável

5. CASOS DE USO

Os casos de uso tem o objetivo de auxiliar a comunicação entre os analistas e o cliente. Um diagrama de caso de uso descreve um cenário que mostra as funcionalidades do sistema do ponto de vista do usuário.

5.1. DIAGRAMA E RELAÇÃO DE CASOS DE USO

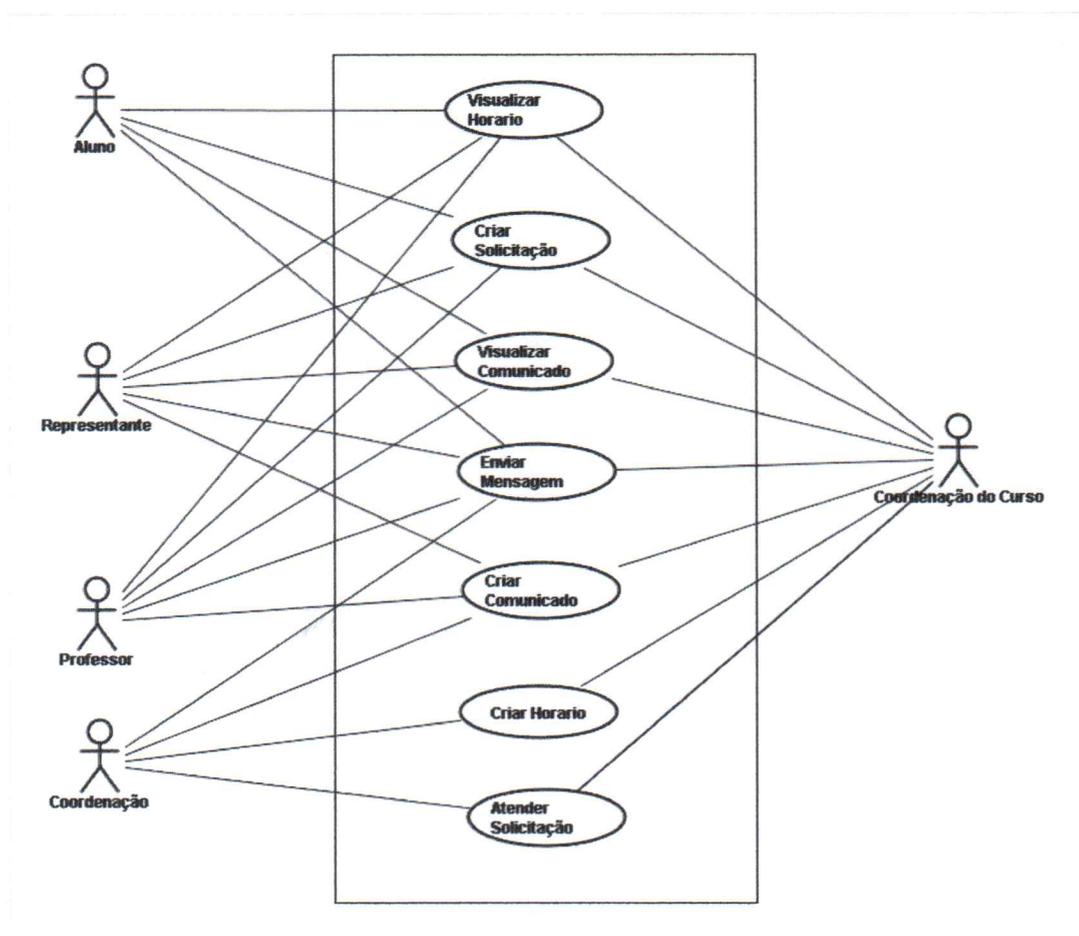


Figura 1 - Diagrama de casos de uso.

Caso de uso	Atores	Descrição	Referências Cruzadas
Criar horário	Coordenação	Usuário cadastra horários acadêmicos vinculadas à coordenação do curso.	RF005
Visualizar horário	Aluno, Representante, Professor	Usuário visualiza horário acadêmico vinculado à turma a qual ele pertence.	RF006
Criar comunicado	Representante, Professor, Coordenação	Usuário compartilha comunicados e avisos a outros usuários do sistema.	RF007
Visualizar comunicado	Aluno, Representante, Professor	Usuário visualiza comunicados cadastrados vinculados ao seu nome.	RF008
Enviar solicitação	Aluno, Representante, Professor, Coordenação	Usuário envia solicitações aos demais usuários.	RF009
Atender solicitação	Aluno, Representante, Professor, Coordenação	Usuário atende solicitações recebidas	RF010
Criar mensagem	Aluno, Representante, Professor, Coordenação	Usuário contata os demais usuários do sistema através de mensagens cadastradas no sistema.	RF011

Tabela 1 - Relação do diagrama de casos de uso.

6. ESPECIFICAÇÃO DE CASOS DE USO

[UC001] CRIAR HORÁRIO ACADÊMICO

1.1. Atores

Coordenação.

1.2. Entradas e Pré-condições

Usuário responsável autenticado.

1.3. Breve descrição

O usuário configura informações, disciplinas e professores que farão parte das aulas semanais de uma determinada turma.

1.4. Saídas e Pós-condições

O horário deverá ser gravado na base de dados do sistema.

1.5. Fluxo de eventos

1.5.1. Fluxo básico

1. Usuário com perfil coordenação acessa o item "Novo horário".
2. Formulário de criação de horários é apresentado pelo sistema.
3. Formulário de criação de horários solicita a turma e o período a serem vinculada ao novo horário.
4. Usuário informa a turma e o período.
5. Formulário verifica a disponibilidade para criar o horário.
6. Formulário identifica o turno a partir da turma informada e apresenta a estrutura de aulas para montar o horário.
7. Usuários seleciona aula.
8. Formulário carrega as disciplinas disponíveis em seu conteúdo.
9. Usuário seleciona a disciplina desejada para a aula.
10. Após a escolha da disciplina, o formulário carrega os professores aptos e disponíveis a ministrá-la.
11. Usuário seleciona o professor desejado para a aula.
12. Usuário solicita o cadastro da aula.
13. Formulário envia as informações para o servidor.
14. Servidor processa as informações enviadas.

15. Servidor grava as informações na base de dados.
16. Servidor responde ao formulário.
17. Formulário informa o resultado do cadastro ao usuário.

1.5.2. Fluxos alternativos

5-a. Formulário verifica que a turma informada pelo usuário já possui um horário cadastrado no sistema.

1. Formulário redireciona o usuário para a alteração do horário.

1.6. Requisitos Especiais

- A montagem do horário deverá ser rápida e descomplicada

[UC002] VISUALIZAR HORÁRIO ACADÊMICO

2.1. Atores

Aluno, Representante, Professor.

2.2. Entradas e Pré-condições

Usuário autenticado e vinculado à turma associada ao horário.

2.3. Breve descrição

O usuário visualiza informações, disciplinas e professores referentes à rotina de aulas de uma determinada turma, através dos dados cadastradas em um horário acadêmico do sistema.

2.4. Saídas e Pós-condições

As informações do horário são carregadas no conteúdo da página.

2.5. Fluxo de eventos

2.5.1. Fluxo básico

1. Usuário acessa o item “Visualizar horário acadêmico”.
2. Página de visualização do horário é apresentada pelo sistema.
3. Página identifica o usuário solicitante.

4. Página identifica o vínculo do usuário com a turma associada ao horário em questão.
5. Página solicita as informações necessárias ao servidor.
6. Servidor busca as informações referentes na base de dados.
7. Servidor responde as informações à página.
8. Página de visualização carrega as informações do horário no seu conteúdo.

2.6. Requisitos Especiais

- A visualização do horário deverá ser simples, sem falhas de carregamento e sobrecarga de conteúdo. As cores e fontes devem atender as regras básicas de usabilidade.
- O layout de apresentação do comunicado deverá seguir os padrões atuais do instituto federal.

[UC003] CRIAR COMUNICADO

3.1. Atores

Coordenação, Professor, Representante.

3.2. Entradas e Pré-condições

Usuário responsável autenticado

3.3. Breve descrição

O usuário compartilha comunicados aos usuários do sistema.

3.4. Saídas e Pós-condições

O comunicado deverá ser gravado na base de dados.

3.5. Fluxo de eventos

3.5.1. Fluxo básico

1. Usuário acessa o item “*Novo comunicado*”.
2. Formulário de criação de comunicados é apresentado pelo sistema.

3. Formulário de criação solicita destinatários e informações para a criação do novo comunicado.
4. Usuário preenche os campos do formulário de acordo com o comunicado que deseja.
5. Usuário sinaliza que tudo está preenchido corretamente.
6. Formulário envia as informações para o servidor.
7. Servidor processa as informações enviadas.
8. Servidor grava as informações na base de dados.
9. Servidor notifica os destinatários do comunicado.
10. Servidor responde ao formulário.
11. Formulário informa o resultado do cadastro ao usuário.

[UC004] VISUALIZAR COMUNICADO

4.1. Atores

Professor, Representante, Aluno.

4.2. Entradas e Pré-condições

Usuário autenticado e vinculado ao comunicado.

4.3. Breve descrição

O usuário visualiza comunicados associados a seu nome.

4.4. Saídas e Pós-condições

As informações do horário são carregadas no conteúdo da página.

4.5. Fluxo de eventos

4.5.1. Fluxo básico

1. Usuário acessa o item “*Visualizar comunicado*”.
2. Página de visualização do comunicado é apresentada pelo sistema.
3. Página identifica o usuário solicitante.
4. Página solicita as informações necessárias ao servidor.
5. Servidor busca as informações referentes na base de dados.
6. Servidor desativa a notificação do comunicado para o usuário.

7. Servidor responde as informações à página.
8. Página de visualização carrega o comunicado no seu conteúdo.

4.6. Requisitos Especiais

- A visualização do comunicado deverá ser simples, sem falhas de carregamento e sobrecarga de conteúdo. As cores e fontes devem atender as regras básicas de usabilidade.
- O layout de apresentação do comunicado deverá seguir os padrões atuais do instituto federal.

[UC005] CRIAR SOLICITAÇÃO

5.1. Atores

Coordenação, Professor, Representante, Aluno.

5.2. Entradas e Pré-condições

Usuário responsável autenticado.

5.3. Breve descrição

Usuário envia solicitações aos demais usuários.

5.4. Saídas e Pós-condições

A solicitação deverá ser gravada na base de dados.

5.5. Fluxo de eventos

5.5.1. Fluxo básico

1. Usuário acessa o item “Nova solicitação”.
2. Formulário de envio de solicitações é apresentado pelo sistema.
3. Formulário de criação solicita as informações necessárias para o envio da nova solicitação.
4. Usuário preenche as informações pertinentes a solicitação.
5. Usuário sinaliza que tudo está preenchido corretamente.
6. Formulário envia as informações para o servidor.
7. Servidor processa as informações enviadas.

8. Servidor grava as informações na base de dados.
9. Servidor notifica a nova solicitação aos destinatários.
10. Servidor responde ao formulário.
11. Formulário informa o resultado do cadastro ao usuário.

[UC006] ATENDER SOLICITAÇÃO

6.1. Atores

Coordenação.

6.2. Entradas e Pré-condições

Usuário responsável autenticado.

6.3. Breve descrição

O usuário atende solicitações recebidas.

6.4. Saídas e Pós-condições

O emissor da solicitação é notificado sobre o atendimento.

6.5. Fluxo de eventos

6.5.1. Fluxo básico

1. Usuário acessa o item “*Visualizar solicitação*”.
2. Página de visualização de solicitações é apresentada pelo sistema.
3. Página identifica o usuário solicitante.
4. Página solicita as informações necessárias ao servidor.
5. Servidor busca as informações referentes na base de dados.
6. Servidor responde as informações à página.
7. Página carrega as informações da solicitação.
8. Usuário da procedência no atendimento da solicitação.
9. Usuário sinaliza a conclusão do atendimento.
10. Página envia as informações da solicitação ao servidor novamente.
11. Servidor atualiza a solicitação.
12. Servidor notifica o usuário emissor através de feedback do atendimento.
13. Servidor responde as informações à página.

14. Página informa o resultado do processo no servidor.

6.6. Requisitos Especiais

- O usuário responsável pelo atendimento da solicitação deve ser explicitamente informado sobre os prazos e regras para atendimento das solicitações no sistema.

[UC007] ENVIAR MENSAGEM

7.1. Atores

Coordenação, Professor, Representante, Aluno.

7.2. Entradas e Pré-condições

Usuário responsável autenticado.

7.3. Breve descrição

O usuário contata os demais usuários através de mensagens cadastradas no sistema.

7.4. Saídas e Pós-condições

A mensagem deverá ser gravada na base de dados.

7.5. Fluxo de eventos

7.5.1. Fluxo básico

1. Usuário seleciona o destinatário da mensagem.
2. Formulário de envio de mensagens é apresentado pelo sistema.
3. Usuário informa a mensagem desejada e solicita o envio da mensagem.
4. Formulário envia as informações para o servidor.
5. Servidor processa as informações enviadas.
6. Servidor grava as informações na base de dados.
7. Servidor notifica o destinatário da mensagem.
8. Servidor responde ao formulário.
9. Formulário informa o resultado do envio ao usuário.

7. DIAGRAMAS DE SEQUÊNCIA DO SISTEMA

7.1. CRIAÇÃO DE HORÁRIOS ACADÊMICOS

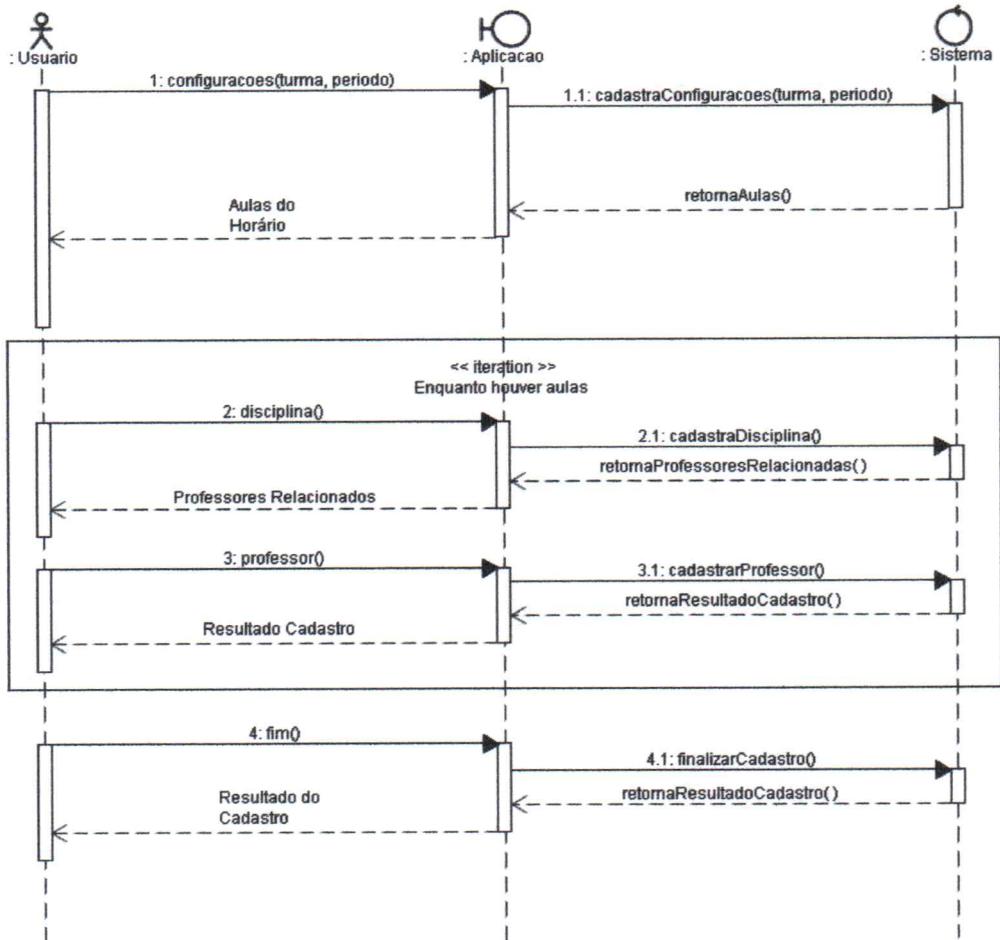


Figura 2 - Diagrama de sequência da criação de horários acadêmicos do sistema.

7.2. ATENDIMENTO DE SOLICITAÇÕES

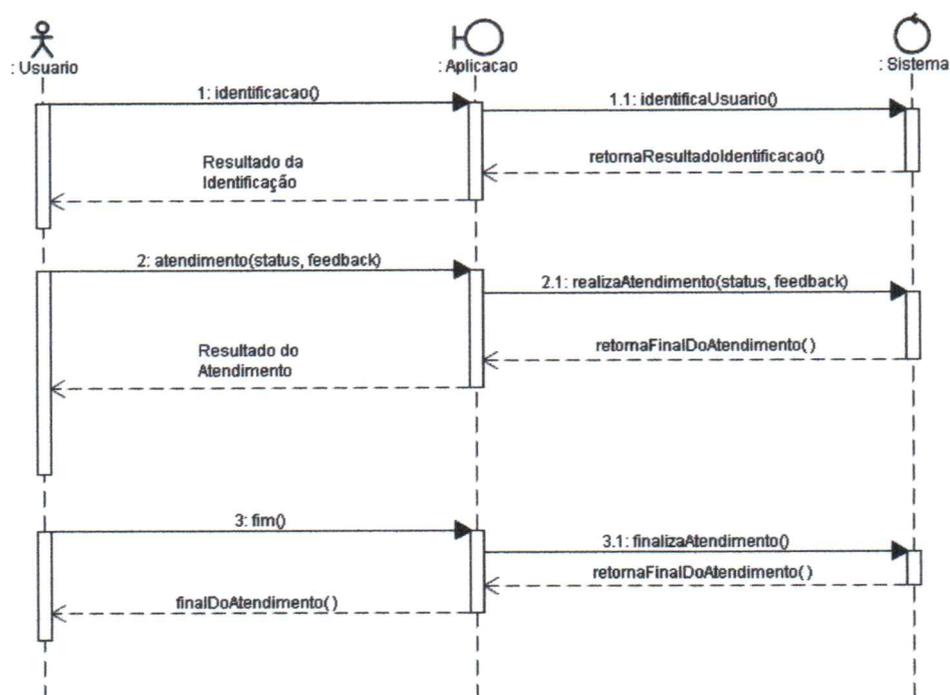


Figura 3 - Diagrama de seqüência do atendimento de solicitações do sistema.

7.3. ENVIO DE SOLICITAÇÕES

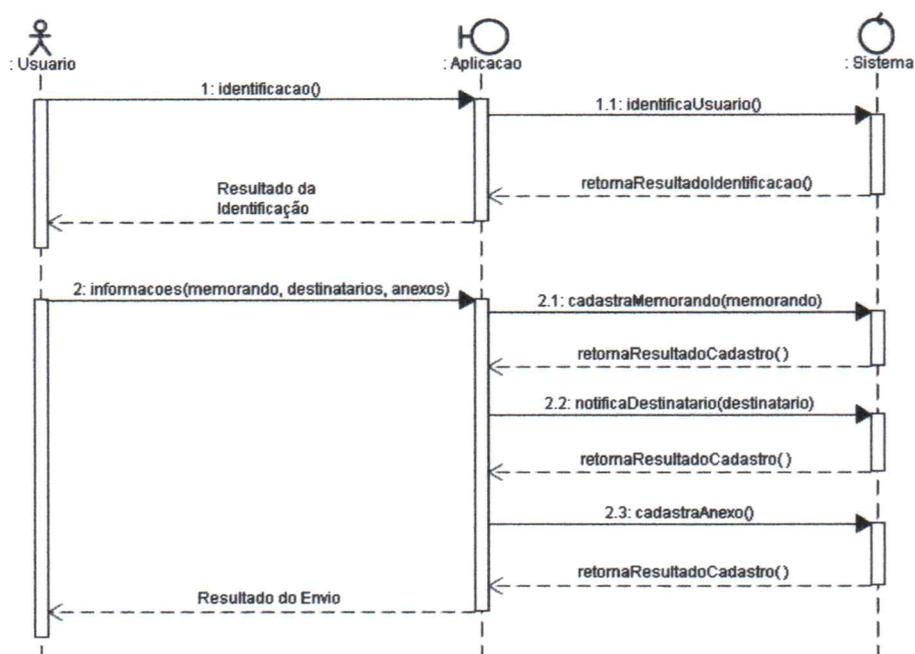


Figura 4 - Diagrama de seqüência do envio de solicitações do sistema.

7.4. ENVIO DE MENSAGENS

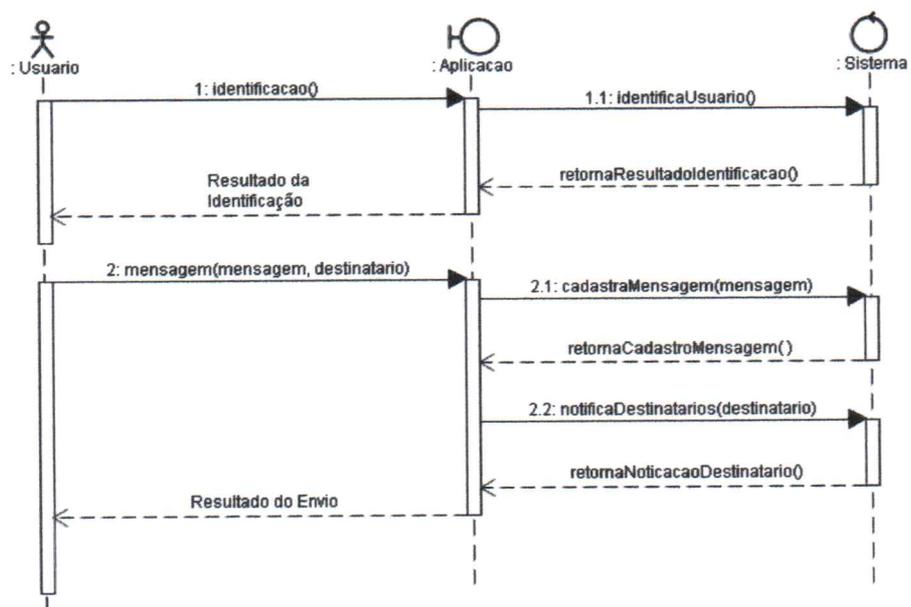


Figura 5 - Diagrama de sequência do envio de mensagens do sistema.

9. TESTES

Os testes foram realizados nos seguintes navegadores: Google Chrome 28.0, Mozilla Firefox 20.0, e Internet Explorer 8.

Caso de teste	Pré-condição	Resultado esperado	Resultado	Incompatibilidade
Login	Informar e-mail e senha; clicar em "login".	Usuário é redirecionado a sua conta.	ok.	
Alternar login	Escolher o login; clicar em "alternar".	Usuário é redirecionado a conta selecionada.	ok.	
Recuperação	Informar e-mail e código de recuperação; clicar em "recuperar".	Usuário é redirecionado à página para alterar a senha.	ok.	
Enviar mensagem no chat	Informar nome, mensagem e destinatário; clicar em "enviar".	A mensagem é enviada; o usuário emissor é notificado do envio; a mensagem é atualizada no chat.	ok.	Mozilla Firefox não atualiza as mensagens; Internet Explorer não possibilita personalizar a cor do editor.
Cadastrar curso	Informar nome, sigla, duração, modalidade, e habilidades do curso; clicar em "cadastrar".	A mensagem "cadastro concluído" é apresentada; os campos nome, sigla, duração, e habilidades são limpos;	ok.	
Pesquisar curso	Selecionar o tipo da pesquisa; informar o termo a ser pesquisado; clicar em "pesquisar";	Resultado da pesquisa é informado ao usuário.	ok.	
Filtrar cursos	Informar as opções do filtro; clicar em "filtrar".	Os dados são selecionados de acordo com as opções selecionadas.	ok.	
Visualizar curso	Informar o identificador do curso ou clicar no botão visualizar referente ao curso.	A visualização dos dados do curso é apresentada.	ok.	
Alterar curso	Alterar os dados do curso; clicar em "salvar alterações".	Os dados do curso são atualizados pelos dados informados na alteração; usuário é redirecionado a página de visualização dos dados do curso que alterou.	ok.	
Excluir curso	Informar o identificador do curso ou clicar no botão excluir referente ao	Todos os dados referentes ao curso são excluídos.	ok.	

	curso.			
Cadastrar turma	Informar nome, carga horária, descrição, objetivos da disciplina; clicar em "cadastrar".	A mensagem "cadastro concluído" é apresentada; os campos nome, carga horária, descrição, objetivos são limpos;	ok.	
Pesquisar turma	Selecionar o tipo da pesquisa; informar o termo a ser pesquisado; clicar em "pesquisar";	Resultado da pesquisa é informado ao usuário.	ok.	
Filtrar turmas	Informar as opções do filtro; clicar em "filtrar".	Os dados são selecionados de acordo com as opções selecionadas.	ok.	
Visualizar turma	Informar o identificador da turma ou clicar no botão visualizar referente à turma.	A visualização dos dados da turma é apresentada.	ok.	
Alterar turma	Alterar os dados da turma; clicar em "salvar alterações".	Os dados da turma são atualizados pelos dados informados na alteração; usuário é redirecionado a página de visualização dos dados da turma que alterou.	ok.	
Excluir turma	Informar o identificador da turma ou clicar no botão excluir referente à turma.	Todos os dados referentes à turma são excluídos.	ok.	
Cadastrar disciplina	Informar nome, carga Horária, descrição, objetivos; clicar em "cadastrar".	A mensagem "cadastro concluído" é apresentada; os campos nome e capacidade são limpos;	ok.	
Pesquisar disciplina	Selecionar o tipo da pesquisa; informar o termo a ser pesquisado; clicar em "pesquisar";	Resultado da pesquisa é informado ao usuário.	ok.	
Filtrar disciplinas	Informar as opções do filtro; clicar em "filtrar".	Os dados são selecionados de acordo com as opções selecionadas.	ok.	
Visualizar disciplina	Informar o identificador da disciplina ou clicar no botão visualizar referente à disciplina.	A visualização dos dados da disciplina é apresentada.	ok.	
Alterar disciplina	Alterar os dados da disciplina; clicar em "salvar alterações".	Os dados da disciplina são atualizados pelos dados informados na alteração; usuário é redirecionado a página de visualização dos	ok.	

		dados da disciplina que alterou.		
Excluir disciplina	Informar o identificador da disciplina ou clicar no botão excluir referente à disciplina.	Todos os dados referentes à disciplina são excluídos.	ok.	
Cadastrar professor	Informar nome, CPF, RG, sexo, data de nascimento, endereço, bairro cidade, CEP, UF, telefone1, telefone2, formação, áreas de atuação, email, senha; clicar em "cadastrar".	A mensagem "cadastro concluído" é apresentada; os campos nome, CPF, RG, endereço, bairro, telefone1, telefone2, formação, áreas de atuação, email, senha são limpos;	ok.	
Pesquisar professor	Selecionar o tipo da pesquisa; informar o termo a ser pesquisado; clicar em "pesquisar";	Resultado da pesquisa é informado ao usuário.	ok.	
Filtrar professores	Informar as opções do filtro; clicar em "filtrar".	Os dados são selecionados de acordo com as opções selecionadas.	ok.	
Visualizar professor	Informar o identificador do professor ou clicar no botão visualizar referente ao professor.	A visualização dos dados do professor é apresentada.	ok.	
Gerar relatório de acesso dos professores	Informar a data; clicar em "relatório".	Relatório referente à data informa é gerado.	ok.	
Alterar professor	Alterar os dados do professor; clicar em "salvar alterações".	Os dados do professor são atualizados pelos dados informados na alteração; usuário é redirecionado a página de visualização dos dados do professor que alterou.	ok.	
Excluir professor	Informar o identificador do professor ou clicar no botão excluir referente ao professor.	Todos os dados referentes ao professor são excluídos.	ok.	
Cadastrar aluno	Informar nome, CPF, RG, sexo, data de nascimento, endereço, bairro cidade, CEP, UF, telefone1, telefone2, matrícula, curso, turma, email, senha; clicar em "cadastrar".	A mensagem "cadastro concluído" é apresentada; os campos nome, CPF, RG, endereço, bairro, telefone1, telefone2, matrícula, email e senha são limpos;	ok.	
Pesquisar aluno	Selecionar o tipo da pesquisa; informar o	Resultado da pesquisa é informado ao	ok.	

	termo a ser pesquisado; clicar em "pesquisar";	usuário.		
Filtrar alunos	Informar as opções do filtro; clicar em "filtrar".	Os dados são selecionados de acordo com as opções selecionadas.	ok.	
Visualizar aluno	Informar o identificador do aluno ou clicar no botão visualizar referente ao aluno.	A visualização dos dados do aluno é apresentada.	ok.	
Gerar relatório de acesso dos alunos	Informar a data; clicar em "relatório".	Relatório referente à data informa é gerado.	ok.	
Alterar aluno	Alterar os dados do aluno; clicar em "salvar alterações".	Os dados do aluno são atualizados pelos dados informados na alteração; usuário é redirecionado a página de visualização dos dados do aluno que alterou.	ok.	
Excluir aluno	Informar o identificador do aluno ou clicar no botão excluir referente ao aluno.	Todos os dados referentes ao aluno são excluídos.	ok.	
Cadastrar coordenador	Informar identificador do professor a ser cadastrado; clicar em "cadastrar".	A mensagem "cadastro concluído" é apresentada; o campo professor é limpo;	ok.	Internet Explorer não pesquisa os professores cadastrados.
Visualizar coordenador	Informar o identificador do coordenador ou clicar no botão visualizar referente ao coordenador.	A visualização dos dados do coordenador é apresentada.	ok.	
Gerar relatório de acesso dos coordenadores	Informar a data; clicar em "relatório".	Relatório referente à data informa é gerado.	ok.	
Adicionar coordenador ao curso	Coordenador a ser adicionado deve estar cadastrado.	Coordenador é vinculado ao curso.	ok.	
Excluir coordenador	Informar o identificador do coordenador ou clicar no botão excluir referente ao coordenador.	Todos os dados referentes ao coordenador são excluídos.	ok.	
Cadastrar diretor	Informar identificador do professor a ser cadastrado; clicar em "cadastrar".	A mensagem "cadastro concluído" é apresentada; o campo professor é limpo;	ok.	Internet Explorer não pesquisa os professores cadastrados.
Visualizar diretor	Informar o identificador do diretor ou clicar no botão visualizar referente ao diretor.	A visualização dos dados do diretor é apresentada.	ok.	

Gerar relatório de acesso dos diretores	Informar a data; clicar em "relatório".	Relatório referente à data informa é gerado.	ok.	
Adicionar diretor ao curso	Diretor a ser adicionado deve estar cadastrado.	Diretor é vinculado ao curso.	ok.	
Excluir diretor	Informar o identificador do diretor ou clicar no botão excluir referente ao diretor.	Todos os dados referentes ao diretor são excluídos.	ok.	
Criar informativo	Informar título e texto referentes ao informativo; clicar em "criar informativo".	Informativo é criado.	ok.	
Visualizar informativo	Informar o identificador do informativo ou clicar no botão visualizar referente ao informativo.	A visualização dos dados referentes ao informativo é apresentada.	ok.	
Excluir informativo	Informar o identificador do informativo ou clicar no botão excluir referente ao informativo.	Todos os dados referentes ao informativo são excluídos.	ok.	
Criar comunicado	Informar título referente ao comunicado; clicar em "criar comunicado".	Comunicado é criado; usuário é redirecionado à edição do comunicado.	ok.	
Editar comunicado	Editar o memorando; definir destinatários; anexar arquivos; clicar em "finalizar o cadastro".	Memorando é salvo; destinatários são notificados; arquivos anexados são cadastrados e armazenados no sistema; mensagem "cadastro concluído" é apresentada ao usuário.	ok.	
Visualizar comunicado	Informar o identificador do comunicado ou clicar no botão visualizar referente ao comunicado.	A visualização dos dados referentes ao comunicado é apresentada.	ok.	
Excluir comunicado	Informar o identificador do comunicado ou clicar no botão excluir referente ao comunicado.	Todos os dados referentes ao comunicado são excluídos.	ok.	
Criar solicitação	Informar título referente à solicitação; clicar em "criar solicitação".	Solicitação é criada; usuário é redirecionado à edição da solicitação.	ok.	
Editar solicitação	Editar o memorando; definir destinatários;	Memorando é salvo; destinatários são	ok.	

	anexar arquivos; clicar em "finalizar o cadastro".	notificados; arquivos anexados são cadastrados e armazenados no sistema; mensagem "cadastro concluído" é apresentada ao usuário.		
Visualizar solicitação	Informar o identificador da solicitação ou clicar no botão visualizar referente à solicitação.	A visualização dos dados referentes à solicitação é apresentada.	ok.	
Excluir solicitação	Informar o identificador da solicitação ou clicar no botão excluir referente à solicitação.	Todos os dados referentes à solicitação são excluídos.	ok.	
Criar horário acadêmico	Informar turma, período e observações; clicar em "iniciar horário".	Horário acadêmico é criado; usuário é redirecionado à página de configuração do horário acadêmico.	ok.	
Adicionar aula ao horário acadêmico	Selecionar aula; informar disciplina e professor; clicar em "cadastrar aula".	Aula é adicionada ao horário acadêmico.	ok.	
Filtrar horários acadêmicos	Informar as opções do filtro; clicar em "filtrar".	Os dados são selecionados de acordo com as opções selecionadas.	ok.	
Visualizar horário acadêmico	Informar o identificador do horário acadêmico ou clicar no botão visualizar referente ao horário acadêmico.	A visualização dos dados do horário acadêmico é apresentada.	ok.	
Excluir horário acadêmico	Informar o identificador do horário acadêmico ou clicar no botão excluir referente ao horário acadêmico.	Todos os dados referentes ao horário acadêmico são excluídos.	ok.	
Mostrar usuários on-line	Estar logado no sistema; clicar em "bate-papo".	Usuários on-line são mostrados	ok.	Mozilla Firefox não atualiza a lista de usuários; Internet Explorer não posiciona a lista no local definido no layout.
Conversar com um usuário	Selecionar um usuário para conversar.	Usuário é redirecionado à página para realizar a conversa.	ok.	
Atualizar o status dos usuários de	Acessar a conversa.	O status dos usuários é alterado.	ok.	Mozilla Firefox não o status dos usuários da