

**UNIVERSIDADE FEDERAL DE MATO GROSSO
INSTITUTO DE COMPUTAÇÃO
COORDENAÇÃO DE ENSINO DE GRADUAÇÃO EM
CIÊNCIAS DA COMPUTAÇÃO**

**UMA PROPOSTA DE CUSTOMIZAÇÃO DA PÁGINA DE
AUTENTICAÇÃO DA REDE WI-FI "IC ACADÊMICO" DO
INSTITUTO DE COMPUTAÇÃO**

RODRIGO VENANCIO VERISSIMO

CUIABÁ – MT

2019

UNIVERSIDADE FEDERAL DE MATO GROSSO

INSTITUTO DE COMPUTAÇÃO
COORDENAÇÃO DE ENSINO DE GRADUAÇÃO EM
CIÊNCIA DA COMPUTAÇÃO

UMA PROPOSTA DE CUSTOMIZAÇÃO DA PÁGINA DE
AUTENTICAÇÃO DA REDE WI-FI "IC ACADÊMICO" DO
INSTITUTO DE COMPUTAÇÃO

RODRIGO VENANCIO VERISSIMO

Relatório apresentado ao Instituto de
Computação da Universidade Federal de
Mato Grosso, para obtenção do título de
Bacharel em Ciência da Computação.

CUIABÁ – MT
2019

UNIVERSIDADE FEDERAL DE MATO GROSSO
INSTITUTO DE COMPUTAÇÃO
COORDENAÇÃO DE ENSINO DE GRADUAÇÃO EM
CIÊNCIAS DA COMPUTAÇÃO

RODRIGO VENANCIO VERISSIMO

Relatório de Estágio Supervisionado apresentado à Coordenação do Curso de Ciência da Computação como uma das exigências para obtenção do título de Bacharel em Ciências da Computação da Universidade Federal de Mato Grosso

Aprovado por:

Prof. M.e Jivago Medeiros Ribeiro
(ORIENTADOR)

Prof. M.e Nelcilenio Virgílio de Souza Araújo
(SUPERVISOR)

M.e. César Eduardo Guarienti
(AVALIADOR EXTERNO)

DEDICATÓRIA

*A Deus pela saúde e oportunidades,
a minha família pelo apoio aos estudos em especial minha mãe, Elizabete que
dedicou sua vida para educação de seus filhos.*

AGRADECIMENTOS

Agradeço, à todos os professores e supervisores, aos quais tive a oportunidade de caminhar junto, e de um modo muito especial:

Ao professor Jivago por me orientar e me ajudar com o estágio com grande atenção e dedicação.

Ao professor Nelcileo por me lembrar e cobrar as datas acadêmicas, e ser um grande amigo durante e depois do curso.

Ao Professor Carlos Wesley, pela sua boa vontade em transmitir novos ensinamentos, fazendo nos crescer para a vida.

À Professora Cláudia, pela sua disponibilidade e pronto atendimento.
Aos meus pais, pelo amor e pelo apoio que me transmitiram durante essa caminhada.

Aos meus colegas de sala, pelo apoio e ajuda mútua.

Ao meu amigo Rondinely que esteve me apoiando no curso desde o primeiro semestre.

SUMÁRIO

LISTA DE FIGURAS	7
LISTA DE SIGLAS E ABREVIATURAS	9
RESUMO	10
1. APRESENTAÇÃO DO LOCAL DE ESTÁGIO	13
2. JUSTIFICATIVA E OBJETIVOS	14
3. REVISÃO DE LITERATURA	15
3.1. Engenharia web	15
3.1.1. Desenvolvimento Web Responsivo	17
3.1.2. Interação Humano Computador	18
3.2. Estrutura Cliente-Servidor	19
3.3. SSL/TLS e HTTPS	20
3.4. Autenticação / RADIUS	20
3.5. Framework	21
3.6. Tecnologias e ferramentas utilizadas	21
3.6.1. Php	21
3.6.2. JavaScript	22
3.6.3. jQuery	22
3.6.4. Bootstrap	23
3.6.5. PhpStorm	23
3.6.6. MySQL	23
3.6.7. PfSense	24
3.6.8. Captive Portal	24
3.6.9. Máquina Virtual	25
3.6.10. Infraestrutura	25
4. JUSTIFICATIVA DAS ATIVIDADES NO PLANO DE ESTÁGIO	28
4.1. Atividades desenvolvidas no primeiro mês	31
4.1.1. Levantamento de requisitos	31
4.1.2. Criação de protótipos	31
4.1.3. Implementação	32
4.2. Atividades desenvolvidas no segundo mês	36
4.2.1. Implementação	36
4.2.1.1. Página inicial	36
4.2.1.2. Envio do mural	39
4.2.1.3. Exibição de postagens	41
4.2.1.4. Autenticação do painel administrador	42
4.2.1.5. Painel administrador	43
4.2.1.6. Aprovação do mural	44
4.2.1.7. Página criação de notícias	46
4.2.1.8. Página de criação de usuário	47
4.2.1.9. Páginas de gerenciamento	47

4.2.1.10. Detalhes do desenvolvimento	48
4.2.2. Implementação do HTTPS	48
4.3. Atividades desenvolvidas no terceiro mês	51
4.3.1. Testes	51
4.3.1.1. Criação do ambiente de emulação	51
4.3.1.2. Teste 1 - Captive Portal	53
4.3.1.3. Teste 2 - Permissões do acesso SSH	56
4.3.1.4. Teste 3 - Instalando aplicação web	56
4.3.2. Implantação	57
4.3.3. Relatório Final	58
5. DIFERENÇAS ENTRE O PLANEJADO E O EXECUTADO	59
6. DIFICULDADES ENCONTRADAS	60
7. CONCLUSÕES	61
8. REFERÊNCIAS BIBLIOGRÁFICAS	62

LISTA DE FIGURAS

Figura 1 - Página de autenticação do IC acessado por um dispositivo celular.....	14
Figura 2 - Página de autenticação do IC com zoom aplicado e conteúdo reposicionado.....	14
Figura 3 - Engenharia de software em camadas.....	15
Figura 4 - Página oficial de login do Instagram, acessado por um dispositivo móvel, sem um design responsivo.....	17
Figura 5 - Aplicando zoom na página oficial de login do Instagram.....	17
Figura 6 - Página oficial de login do Instagram com seu design responsivo projetado para o dispositivo móvel.....	18
Figura 7 - Funcionamento Cliente-Servidor.....	19
Figura 8 - Card padrão do Bootstrap.....	31
Figura 9 - Wireframe do card na horizontal.....	31
Figura 10 - Wireframe da página de inicial.....	32
Figura 11 - Wireframe da página de publicação.....	32
Figura 12 - Página inicial da primeira versão do sistema web de autenticação em telas grandes....	35
Figura 13 - Página inicial da primeira versão do sistema web de autenticação em telas médias....	36
Figura 14 - Página inicial do sistema web de autenticação em telas pequenas.....	36
Figura 15 - Código em HTML do botão Carregar Mais em Mural	37
Figura 16 - Código em Javascript da função de carregar mais notícias	38
Figura 17 - Página com o formulário de envio de uma nova postagem do mural.....	39
Figura 18 - Mensagem mostrada após o envio de uma nova postagem do mural.....	40
Figura 19 - página de visualização de uma postagem no sistema web de autenticação.....	41
Figura 20 - Página com os campos de autenticação para acesso a área de administradora.....	42
Figura 21 - Mensagem de erro ao entrar com o usuário errado.....	42
Figura 22 - Mensagem de erro ao entrar com a senha errada.....	42
Figura 23 - Aviso ao tentar entrar sem digitar o usuário ou senha.....	42
Figura 24 - Template original SB Admin 2.....	43
Figura 25 - Página inicial do painel administrador.....	43
Figura 26 - Janela de visualização de uma postagem de mural vista no painel administrador.....	44

Figura 27 - Card de mural no painel administrador após o aceite do moderador.....	44
Figura 28 - Card de mural no painel administrador após o moderador recusar a publicação.....	45
Figura 29 - Card do mural no painel administrador.....	45
Figura 30 - Página de envio de uma nova notícia.....	46
Figura 31 - página de criação de um novo usuário para o painel administrador.....	46
Figura 32 - página de listagem das postagem no mural.....	47
Figura 33 - Recorte de texto feito do site oficial do Let's Encrypt	48
Figura 34 - Passos para configurar a rede interna no VirtualBox	51
Figura 35 - Tela de instalação do pfSense	51
Figura 36 - Terminal do pfSense com a configuração de rede em destaque	52
Figura 37 - Personalização das configurações do captive portal.....	53
Figura 38 - Terminal da VM do Windows 10 com o comando ping executado antes e depois da autenticação do captive portal.....	54
Figura 39 - Página padrão do captive portal.....	54

LISTA DE SIGLAS E ABREVIATURAS

BSD	<i>Berkeley Software Distribution</i>
CGI	<i>Common Gateway Interface</i>
CPU	<i>Central Processing Unit</i>
CRUD	<i>Create, Read, Update, Delete</i>
CSS	<i>Cascading Style Sheets</i>
DAO	<i>Data Access Object</i>
DHCP	<i>Dynamic Host Configuration Protocol</i>
DNS	<i>Domain Name System</i>
GB	<i>Giga Byte</i>
HD	<i>Hard Disk drive</i>
HTML	<i>HyperText Markup Language</i>
HTTP	<i>HyperText Transfer Protocol</i>
IBM	<i>International Business Machines</i>
IC	Instituto de Computação
IDE	<i>Integrated Development Environment</i>
IP	<i>Internet Protocol address</i>
LAN	<i>Local Area Network</i>
LTS	<i>Long Term Support</i>
MB	<i>Mega Byte</i>
MVC	<i>Model-View-Controller</i>
PC	<i>Personal Computer</i>
PHP	<i>Hypertext Preprocessor, originalmente Personal Home Page</i>
RADIUS	<i>Remote Authentication Dial In User Service</i>

RESUMO

Este documento tem como objetivo descrever as atividades desenvolvidas durante o Estágio Supervisionado no Instituto de Computação (IC), com a finalidade de obtenção do título de Bacharel em Ciência da Computação pela Universidade Federal de Mato Grosso. Teorias vistas em sala de aula, como Orientação a Objetos, Banco de Dados, Desenvolvimento Web, foram colocados em prática neste trabalho. Foi desenvolvido funcionalidades de um sistema web responsivo para publicações de postagens e autenticação na rede sem fio do Instituto de Computação. Foi utilizado o ambiente de desenvolvimento *PHPStorm* para desenvolvimento geral e o *framework Bootstrap 4* para desenvolvimento do *layout*. O trabalho realizado separou as publicações em duas áreas, são elas Notícias do IC e Mural Público.



**TERMO DE COMPROMISSO DE ESTÁGIO
ESTÁGIO CURRICULAR OBRIGATORIO**
(INSTRUMENTO JURÍDICO QUE TRATA A LEI 11.788, DE 25 DE SETEMBRO DE 2008)



Em 14 de novembro de 2018, na cidade de Cuiabá neste ato, as partes a seguir nomeadas:

EMPRESA/INSTITUIÇÃO CONCEDENTE

CNPJ

33.004.540/0001-00

Razão Social

Fundação Universidade Federal de Mato Grosso

Sector

Instituto de Computação

Endereço

Avenida Fernando Correa da Costa, 2367

Representante

Claudia Aparecida Martins

Supervisor

Neicileno Virgilio de Souza Araújo

Cargo

Diretora do Instituto de Computação

Cargo

Coordenador do curso de Sistemas da Informação

INSTITUIÇÃO DE ENSINO

CNPJ

33.004.540/0001-00

Razão Social

Fundação Universidade Federal de Mato Grosso

Matrícula RG

201511310007

Curso

Ciência da Computação

ESTUDANTE/ESTAGIÁRIO

CPF

08895324943

Nome

Rodriqo Venâncio Veríssimo

Coordenador de Estágios

Carlos Ueslei Rodrigues de Oliveira

Endereço

Av. Fernando Corrêa da Costa nº 2367, Cidade Universitária "Gabriel Novis

Neves" Bairro: Boa Esperança Cidade: Cuiabá-MT CEP: 78060-900

Endereço

Av. Dr. José Feliciano Figueiredo, 36

Celebram entre si este TERMO DE COMPROMISSO DE ESTÁGIO, convencionando as cláusulas seguintes:

CLÁUSULA 1ª - Este termo tem por objetivo formalizar e particularizar a relação jurídica especial existente entre o ESTAGIÁRIO, EMPRESA/INSTITUIÇÃO CONCEDENTE e INSTITUIÇÃO DE ENSINO, caracterizando a não vinculação empregatícia.

CLÁUSULA 2ª - O estágio curricular **OBRIGATORIO** dos acadêmicos atende ao Projeto Pedagógico do curso, conforme seu regulamento nos termos da Lei n.º 11.788/08.

CLÁUSULA 3ª - Ficam comprometidas entre as partes as seguintes condições básicas para a realização do estágio:

a) Vigência de: 10/11/2018 até 22/04/2019;

b) Horário de estágio: das 13:00 às 19:00;

c) Carga Horária semanal: 30 horas;

d) As atividades a serem executadas pelo ESTAGIÁRIO constarão no documento intitulado **PLANO DE ESTÁGIO** onde será discriminado o projeto a ser desenvolvido com o objetivo de aprovação na disciplina de Estágio Supervisionado.

CLÁUSULA 4ª - O Seguro de Acidentes Pessoais em favor do estagiário fica a cargo da UFMT, na vigência do presente Termo, pela APÓLICE DE SEGURO ACIDENTES PESSOAIS COLETIVO Nº **1018200518981**, com vigência até 25/07/2019.

Carla

CLÁUSULA 5ª – Cabe à INSTITUIÇÃO DE ENSINO:
A COORDENAÇÃO DO CURSO:

- Aprovar, acompanhar e avaliar o estágio, visando à complementação do ensino e da aprendizagem, conforme proposta pedagógica do curso;
- Indicar professor orientador, na área a ser desenvolvida no estágio, como responsável pelo acompanhamento e avaliação das atividades do estagiário;
- Avaliar as instalações da parte concedente do estágio e sua adequação à formação cultural e profissional do educando;
- Avaliar e aprovar Plano de Atividades, conforme competências próprias da atividade profissional e à contextualização curricular;
- Informar à CONCEDENTE do estágio as datas das avaliações acadêmicas, no início do seu período letivo;
- Disponibilizar cópia do termo de compromisso ao aluno.

CLÁUSULA 6ª – Cabe à EMPRESA/INSTITUIÇÃO CONCEDENTE:

- Oferecer ao ESTAGIÁRIO, instalações que tenham condições de proporcionar ao educando atividades de aprendizagem social, profissional, cultural e compatíveis com o respectivo curso de formação;
- Nos períodos de avaliação acadêmica, informados previamente pelo ESTAGIÁRIO ou INSTITUIÇÃO DE ENSINO, reduzir a jornada de estágio para garantir o bom desempenho do estudante;
- Proporcionar à Instituição de Ensino, com periodicidade mínima de 4 (quatro) meses, relatório individual de atividades, devidamente assinado pelo Supervisor de estágio, com vista obrigatória do estagiário;
- Por ocasião de desligamento do estagiário, entregar termo do estágio com indicação resumida das atividades desenvolvidas, dos períodos e da avaliação de desempenho;
- Em caso de Rescisão do presente termo, informar imediatamente à Instituição de Ensino para as devidas providências;
- Manter, à disposição da fiscalização, documentos que comprovem a relação de estágio;
- Garantir que as atividades de estágio iniciarão somente após a celebração deste termo, devidamente assinado pelas partes envolvidas;
- Indicar funcionário de seu quadro de pessoal com formação ou experiência profissional na área de conhecimento desenvolvida no curso do estagiário, para orientá-lo e supervisioná-lo no desenvolvimento das atividades de estágio;
- Requerer, sempre que julgar necessário, documentos que comprovem a regularidade escolar, condição determinante para a realização do estágio.

CLÁUSULA 7ª – Cabe ao ESTAGIÁRIO:

- Cumprir a programação estabelecida para seu ESTÁGIO;
- Obedecer às normas internas da EMPRESA/INSTITUIÇÃO CONCEDENTE;
- Manter confidencial e não divulgar a quaisquer terceiros as Informações Confidenciais, sem a prévia autorização por escrito da EMPRESA/INSTITUIÇÃO CONCEDENTE;
- Apresentar os documentos comprobatórios da regularidade da sua situação escolar, sempre que solicitado pelas partes;
- Atualizar dados cadastrais e escolares junto à CONCEDENTE;
- Informar, qualquer alteração na sua situação escolar, tais como o abandono, a transferência do curso, trancamento da matrícula e alterações cadastrais gerais;
- Encaminhar, à INSTITUIÇÃO DE ENSINO e à EMPRESA/INSTITUIÇÃO CONCEDENTE, uma via do presente termo assinado por todas as partes;
- Comprometer-se a preencher, relatório de atividades, com periodicidade mínima de seis meses ou quando solicitado.

CLÁUSULA 8ª – O presente instrumento e o Plano de Atividades de Estágio serão alterados ou prorrogados através de TERMOS ADITIVOS.

E por estarem de inteiro e comum acordo com as condições e diretrizes do TERMO DE CONVÊNIO, do decorrente TERMO DE COMPROMISSO DE ESTÁGIO e do PLANO DE ATIVIDADES as partes assinam em 3 (três) vias de igual teor.



EMPRESA/INSTITUIÇÃO, COM SEDE EM MATRIM
(carimbo e assinatura) de Computação
Matrão, 1124026
SIAPE: 1124026



ESTAGIÁRIO

COORDENAÇÃO DE ESTÁGIO IC/UFMT
(carimbo e assinatura)



1. APRESENTAÇÃO DO LOCAL DE ESTÁGIO

O Instituto de Computação da Universidade Federal de Mato Grosso (UFMT) foi criado em 2007. Possui os cursos de Ciência da Computação e Sistemas da Informação, totalizando cerca de 400 alunos. Possui 5 laboratórios com 30 computadores em cada e 5 salas de aula. Além dos alunos do instituto abriga vários alunos de outros cursos da UFMT. Sendo para usar os laboratórios ou salas de aula.

O local além de ser o local de estágio, também é o local de minha formação. Grande parte do trabalho foi realizado em *Home Office*. As necessidades e demandas são sugeridas pelos professores do Instituto. E o trabalho é acompanhado por reuniões frequentes com o orientador, supervisor e diretor do bloco.

2. JUSTIFICATIVA E OBJETIVOS

A rede sem fio do Instituto de Computação é utilizado por muitos alunos da UFMT. Ao se conectar é necessário a autenticação, sendo que o aluno ou professor acessa a página várias vezes por dia. Da maneira que está atualmente a página de autenticação não está responsiva, trazendo uma baixa experiência de usuário. Na [Figura 1](#) podemos ver a página atual de autenticação do Instituto de Computação acessado por um dispositivo celular. Podemos perceber também que para identificar o conteúdo da página é necessário aplicar o zoom ([Figura 2](#)). Após preencher e enviar os campos de autenticação, o acesso a internet é liberado.



Figura 1 - Página de autenticação do IC acessado por um dispositivo celular.

Fonte: Print screen da página de autenticação acessado por um dispositivo celular no sistema android Pie



Figura 2 - Página de autenticação do IC com zoom aplicado e conteúdo reposicionado.

Fonte: Print screen da página de autenticação acessado por um dispositivo celular no sistema android Pie

Ainda, acredita-se que a área seja subutilizada, uma vez que conta apenas com o formulário de autenticação, não possuindo nenhum outro recurso ou conteúdo que permitiria, por exemplo, a comunicação entre as direções e coordenações do IC com alunos e demais atores do Instituto de Computação ou ainda a comunicação entre os próprios usuários. Dado o exposto, este trabalho de Estágio Supervisionado teve como objetivo a proposta de uma nova página de autenticação para a rede “IC ACADÊMICA” do Instituto de Computação; uma página que seja responsiva melhorando assim a experiência de uso dos usuários e possua recursos que visem explorar um espaço bastante acessado porém atualmente ocioso. Ainda, no decorrer do trabalho pretende-se explorar questões de segurança relacionadas a página de autenticação e a autenticação em si, como por exemplo, o uso de *Hyper Text Transfer Protocol Secure* (HTTPS).

3. REVISÃO DE LITERATURA

Nesta seção serão apresentados os conceitos fundamentais para a compreensão deste trabalho. Destacando a importância da engenharia *web*, desenvolvimento *web* responsivo, *frameworks*, tecnologias e ferramentas utilizadas.

3.1. Engenharia web

Segundo Pressman (2006), sistemas e aplicativos da *web* são caracterizados por disponibilizar grande quantidade de conteúdo e funcionalidade para grande população de usuários. A engenharia para a *web* é, portanto, o processo utilizado para criar aplicativos *web* de alta qualidade.

Para Pressman a engenharia de *software* é extremamente importante com enfoque na qualidade. Podemos observar na [Figura 3](#) que a qualidade é justamente a base da pirâmide na engenharia de *software* proposta por Pressman.



Figura 3 - Engenharia de software em camadas.

Fonte: PRESSMAN (2006)

Pressman cita as principais diferenças entre desenvolver um aplicativo para a *web* e desenvolver um *software* tradicional:

- **Imediatismo:** o tempo que um site completo precisa ficar pronto pode ser apenas alguns poucos dias ou semanas. Desenvolvedores devem, portanto, utilizar métodos de planejamento, análise, projeto, implementação e teste que estejam adaptados para estes cronogramas comprimidos necessários no desenvolvimento para a *web*.
- **Segurança:** aplicativos para a *web* estão disponíveis via rede, é difícil ou até mesmo impossível limitar a população de usuários que irão acessar o aplicativo. Para poder proteger o conteúdo e fornecer métodos seguros de transmissão de dados é preciso implementar medidas rígidas de segurança no aplicativo e na infra-estrutura do mesmo.
- **Estética:** é inegável que boa parte do apelo dos aplicativos para a *web* é o seu visual. Quando um aplicativo é projetado para vender produtos ou idéias, estética pode ser tão importante para o sucesso quanto o projeto técnico.

A construção de um sistema para a *web* necessita do conhecimento de pessoas de diferentes áreas. Como resultado, a Engenharia para a *web* é multidisciplinar, e dela participam áreas como: análise de sistemas e projetos;

engenharia de *software*; engenharia de hipermídia e hipertexto; engenharia de requisitos; interação humano computador; desenvolvimento de interface de usuário; engenharia de informação; indexação e recuperação de informações; teste; modelagem e simulação; gerenciamento de projetos; e projeto gráfico e apresentação (GINIGE, 2001).

3.1.1. Desenvolvimento Web Responsivo

O Desenvolvimento *web* responsivo possibilita que as páginas web se adaptem a qualquer tamanho de tela e continuem proporcionando uma boa experiência do usuário, garantindo a leitura e navegação confortável (ZEMEL, 2015; DA SILVA, 2014).

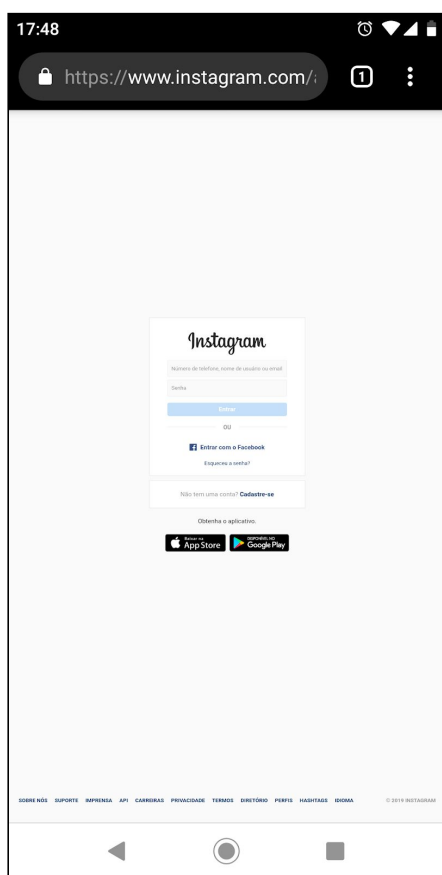


Figura 4 - Página oficial de login do Instagram, acessado por um dispositivo móvel, sem um design responsivo.

Fonte: Print screen da página do Instagram acessado por um dispositivo celular no sistema android Pie



Figura 5 - Aplicando zoom na página oficial de login do Instagram.

Fonte: Print screen da página do Instagram acessado por um dispositivo celular no sistema android Pie

Ao dizer “páginas web que se adaptam” não é simplesmente funcionar em qualquer dispositivo. O conceito de *design* responsivo vai além disso: uma página responsiva se reorganiza, em tempo real, da melhor forma com a finalidade de apresentar o seu conteúdo de acordo com o dispositivo em questão. O que descarta a ideia de fazer diferentes páginas para diferentes dispositivos, tendo apenas um código a ser exibida em qualquer tela (DA SILVA, 2014; ALDABRA, 2017).

Desta forma, é ilustrado na [Figura 4](#) a página oficial de login do Instagram, acessado por um dispositivo móvel, sem um design responsivo (usando a opção de versão para desktop). Na [Figura 5](#) podemos ver a necessidade de aproximar mais o conteúdo (aplicar *zoom*) e de reposicionar para o conteúdo desejado na tela.



Figura 6 - Página oficial de login do Instagram com seu design responsivo projetado para o dispositivo móvel.

Fonte: Print screen da página do Instagram acessado por um dispositivo celular no sistema android Pie

Já na [Figura 6](#) é apresentado a mesma página, porém com seu design responsivo projetado para o dispositivo móvel. Podemos ver que o conteúdo da página é ajustado ao tamanho da tela e uma opção de baixar o aplicativo do instagram aparece na mesma.

3.1.2. Interação Humano Computador

A Interação Humano-Computador (IHC) é uma disciplina que diz respeito ao *design*, avaliação e implementação de sistemas de computação interativos para uso humano em um contexto social e com os estudos dos principais fenômenos que os cercam (HEWETT et al, 1992).

A IHC atualmente é uma ciência multidisciplinar, que engloba diversas áreas como: Ciência da Computação, Psicologia Cognitiva, Psicologia Organizacional e Social, Ergonomia e Fatores Humanos, Engenharia, Design, Antropologia, Sociologia, Filosofia, Linguística e Inteligência Artificial (PREECE, 2002 e OLIVEIRA NETTO, 2004). O conjunto de todas estas áreas visa apoiar de forma sistemática diversas etapas e procedimentos das metodologias de Interação Humano-Computador, uma vez que esta é uma área que trabalha tanto com aspectos de competência das ciências exatas (os sistemas computacionais) quanto com os aspectos físicos, psicológicos e comportamentais dos indivíduos, considerando também o contexto social em que um determinado sistema se insere para seus usuários.

O principal objetivo da IHC é estudar e definir métodos para o projeto de sistemas ou dispositivos que sejam de fácil utilização, eficientes, eficazes e que possibilitem conforto aos indivíduos que irão utilizá-los. Ela visa compreender como e porque uma pessoa utiliza determinada tecnologia (AGNER, 2006)

Portanto, um dos princípios da IHC é tratar como o usuário se comunica com a máquina e como a tecnologia responde à interação do usuário.

3.2. Estrutura Cliente-Servidor

A arquitetura Cliente-Servidor é composta por um ou mais servidores e por um ou

mais clientes. Na verdade, este conceito representa uma interação entre processos de *software* executando de forma concorrente. Os processos que executam no computador do usuário final são chamados de *client-side*, sendo que estes processos solicitam a outros processos, que no caso estariam executando em um servidor, conhecido como *server-side*, para continuar sua execução, ou seja: há uma comunicação entre processos em máquinas separadas (GRANDI, 1996).

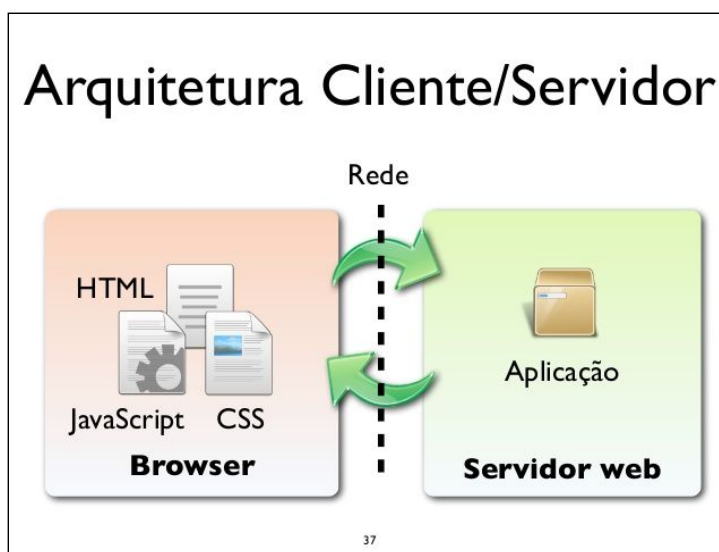


Figura 7 - Funcionamento Cliente-Servidor.

Fonte: <https://pt.slideshare.net/watinha1/apresentacao-17018075>

Para melhor compreensão do funcionamento de *server-side* e *client-side*, é ilustrado na [Figura 7](#) como os dois lados se comunicam. O cliente requisita uma página ao servidor, que por sua vez faz os procedimentos necessários para responder ao cliente com a página solicitada. O cliente faz o *download* da página enviada pelo servidor e só então começa a interpretar seu conteúdo.

3.3. SSL/TLS e HTTPS

SSL significa *Secure Sockets Layer*, um tipo de segurança digital que permite a comunicação criptografada entre um site e um navegador. Atualmente a tecnologia se encontra depreciada e está sendo completamente substituída pelo TLS.

TLS é uma sigla que representa *Transport Layer Security* e certifica a proteção de dados de maneira semelhante ao SSL. Como o SSL não está mais de fato em uso, esse é o termo correto que deveria ser utilizado.

O HTTPS é uma extensão segura do HTTP (*Hypertext Transfer Protocol*). Os sites que configurarem um certificado SSL/TLS podem utilizar o protocolo HTTPS para estabelecer uma comunicação segura com o servidor.

O objetivo do SSL/TLS é tornar segura a transmissão de informações sensíveis como dados pessoais, de pagamento ou de login. É uma alternativa à transferência de dados em texto simples na qual a conexão ao servidor não é criptografada e torna mais difícil para que *hackers* possam interceptar a conexão e roubar dados (ARIANA G, 2019).

3.4. Autenticação / RADIUS

Autenticação é o ato de estabelecer ou confirmar algo (ou alguém) como autêntico, isto é, que reivindica a autoria ou a veracidade de alguma coisa. A autenticação também remete à confirmação da procedência de um objeto ou pessoa, neste caso, frequentemente relacionada com a verificação da sua identidade.

O RADIUS (*Remote Authentication Dial In User Service*) é um protocolo de autenticação, autorização e *accounting*. A autenticação verifica a identidade digital do usuário de um sistema banda larga, a autorização garante que um usuário autenticado somente tenha acesso aos recursos autorizados e, por fim, a *accounting* refere-se à coleta de informações sobre o uso dos recursos de um sistema pelos seus usuários (TELECO, Banda Larga).

3.5. Framework

Para Johnson (1997), *framework* é um projeto reutilizável de todo ou parte de um sistema, representado por um conjunto de classes abstratas e a forma como suas instâncias interagem. Também pode-se dizer que um *framework* é o esqueleto da

aplicação, que pode ser *customizado* pelo desenvolvedor. Com *frameworks* a reutilização de código é bem maior, o que é bem mais fácil que criá-los. Pois é mais fácil desenvolver utilizando uma biblioteca de componentes, ainda mais quando não há a necessidade de entender profundamente sobre o componente a ser utilizado. *frameworks* utilizam das vantagens das três características distintas do modelo Orientado a Objetos, sendo eles: abstração de dados, polimorfismo e herança. Onde, assim como um tipo de dado abstrato, uma classe abstrata representa uma interface cujo pode ser modificada por implementação. O polimorfismo é a capacidade de uma única referência invocar métodos diferentes, dependendo do seu conteúdo. E, por fim, herança é proporciona o reuso do código, podendo-se criar novas classes a partir de classes já existentes, herdando seus atributos e métodos e adicionando seus próprios. É uma técnica de programação Orientada a Objetos amplamente utilizada pela indústria, e é parte do motivo de os desenvolvedores que a utilizam serem tão produtivos.

3.6. Tecnologias e ferramentas utilizadas

3.6.1. Php

Segundo sua documentação, o “PHP é uma linguagem de script *open source* de uso geral, muito utilizada, e especialmente adequada para o desenvolvimento *web* e que pode ser embutida dentro do HTML”. Seu código é executado no lado servidor, trazendo as páginas HTML prontas para serem processadas pelo navegador. Portanto, ela pode fazer qualquer coisa que outro programa CGI pode fazer, como coleta de dados de um formulário, gerar páginas com conteúdo dinâmico, etc. Possui suporte à uma ampla variedade de banco de dados e, pode ser executado em vários sistemas operacionais de servidores *web*. O que mais chama a atenção para esta linguagem é sua curva de aprendizagem, o que torna-a uma linguagem ótima para iniciantes. Oferece os recursos necessários para pequenos, médios e, até mesmo grandes projetos. É justamente por isso que a definimos como a linguagem a ser utilizada no projeto *web*, também por ter tido um contato com tal linguagem em disciplinas como

“Programação em Ambiente Web I”, o que facilitará futuras melhorias, caso outro aluno dê continuidade ao projeto.

3.6.2. JavaScript

JavaScript é uma linguagem de programação que permite a criação de conteúdos web dinâmicos, manipulando diretamente a página apresentada ao cliente requisitante. Sendo assim, esta linguagem script é interpretada e executada no próprio navegador (*browser*) do usuário, ou seja, ao lado do cliente (*client-side*), diferentemente do PHP que é executado ao lado do servidor como explicado no [Tópico 3.2](#) (JAVASCRIPT, 2019). O *JavaScript* é a base para qualquer desenvolvimento *front-end* (ou *client-side*). A partir dele foram construídos diversos *plug-ins*, bibliotecas e *frameworks* para melhorar ainda mais a interação com o usuário. Para este projeto foi utilizado *jQuery* que depende do *JavaScript* para o funcionamento.

3.6.3. jQuery

O *jQuery*, segundo a *jQuery* (2019), (tradução própria), é uma biblioteca de *JavaScript* rápida, pequena, versátil, extensível e rica em recursos. Tornando a manipulação de documentos HTML, eventos, animações e requisições aos servidores muito mais simples do que o *JavaScript* puro. Esta biblioteca facilita o uso de *JavaScript* em páginas web. Além disso sem o *jQuery* não seria possível trabalhar com outro recurso utilizado para este trabalho.

3.6.4. Bootstrap

Bootstrap é um conjunto de ferramentas de código aberto destinados a criar *layouts* para o *front-end*, é uma das ferramentas mais populares do mundo. Desenvolvido em CSS, *JavaScript* e *jQuery*, permite um desenvolvimento rápido, organizado e responsivo de páginas web (DOCUMENTAÇÃO, 2019). O Bootstrap facilita o desenvolvimento no *front-end*, pois entrega um conjunto de componentes com botões

personalizados, tabelas, *ícones*, *modais*, facilitando muito a vida do programador na criação de páginas com HTML, deixando o desenvolvedor livre para se preocupar com as regras de negócio.

3.6.5. PhpStorm

PhpStorm é um *IDE* (ambiente de desenvolvimento integrado) comercial para PHP criado na plataforma *IntelliJ IDEA* da *JetBrains*. *PhpStorm* fornece um editor para PHP, HTML e *JavaScript* com análise de código em tempo real, prevenção de erros e refatoração automatizada para código PHP e *JavaScript*. Inclui um editor de SQL completo com resultados de consulta editáveis (PHPSTORM, 2019). Com estes recursos a implementação do código se torna mais fluida e rápida ajudando muito o programador.

3.6.6. MySQL

MySQL é um banco de dados bem popular e muito simples, utilizado por pequenas e grandes empresas. Para o desenvolvimento do sistema, havia a necessidade de armazenamento de dados como controle de usuários e salvar as publicações. Foi escolhido o *MySQL* pois já estava disponível no servidor onde aplicação iria executar.

3.6.7. PfSense

O *software* livre e de código aberto *pfSense* é uma distribuição baseada no sistema operacional *FreeBSD*, adaptado para ser utilizado como um *firewall* e/ou roteador. O objetivo é fornecer um *firewall* poderoso, seguro e completamente configurável usando o *hardware* de um PC comum. Ele pode ser configurado e atualizado através de uma interface baseada na *web* e não requer nenhum conhecimento do sistema *FreeBSD* para gerenciar (BUECHLER; PINGLE, 2009).

3.6.8. Captive Portal

O Portal *Captive* é um dos serviços oferecido pelo *pfSense* e pode ser usado para redirecionar tanto usuários da rede *wireless* como da rede cabeada para uma página de autenticação, onde devem inserir suas credenciais para continuar navegando na Internet.

Segundo Lockhart (2004), ele deve permitir que o administrador de rede/sistema monitore quem está utilizando sua rede sem fio. Cada usuário possui um *login* e senha ou algum *voucher* que conceda a ele o acesso (limitado ou não) a rede.

Após habilitar o serviço clicando no *checkbox* “*Enable Captive Portal*”, o administrador de rede pode editar algumas configurações, tais como: limite máximo de conexões simultâneas por endereço IP, cujo propósito é "prevenir que um único host esgote todos os recursos do *firewall*" (BUECHLER; PINGLE, 2009); tempo limite de ociosidade, usuários inativos durante tempo definido são automaticamente desconectados da rede; janela *pop up* de *logout*, visível para que o usuário se desconecte da rede quando desejar; e muitos outros.

3.6.9. Máquina Virtual

A IBM define uma máquina virtual como uma cópia isolada de um sistema físico, e esta cópia está totalmente protegida. Ao invés de ser uma real, isto é, um computador real feito de *hardware* e executando um sistema operacional específico, uma máquina virtual é um computador fictício criado por um programa de simulação. Sua memória, processador e outros recursos são virtualizados. A virtualização é a interposição do *software* (máquina virtual) em várias camadas do sistema. É uma forma de dividir os recursos de um computador em múltiplos ambientes de execução.

Neste trabalho foi utilizada máquinas virtuais devidos as seguintes vantagens:

- Facilidade na replicação de computadores, aplicações ou sistemas operacionais.

- O isolamento da máquina virtual assegura que as aplicações e serviços que serão executados dentro de uma máquina virtual não poderão interferir no sistema operacional original e nem em outras máquinas virtuais.
- Teste de diversos sistemas operacionais sem precisar particionar o HD, ou formatar o computador.

3.6.10. Infraestrutura

Todo o desenvolvimento foi realizado em um ambiente de programação contando com uma memória RAM de 8GB, SSD de 120GB, processador Intel® Core™ i3-2310M CPU @ 2.10GHz x 2 no sistema operacional Xubuntu 18.04.2 LTS (*Bionic Beaver*). Para teste de responsividade da aplicação foi utilizado um recurso dos próprios navegadores para simulação de dispositivos. Além deste recurso foi utilizado um *smartphone* Xiaomi, modelo MI A2, sistema operacional Android Pie versão 9.

PLANO DE ESTÁGIO

Dados do Discente

Matrícula 20151131007	Curso Ciência da Computação	Nome Rodrigo Venâncio Veríssimo	E-mail rodrigovenancioverissimo@gmail.com
---------------------------------	---------------------------------------	---	---

Local de Estágio

CNPJ 33.004.540/0001-00	Nome ou Razão Social Fundação Universidade Federal de Mato Grosso
Nome do Supervisor Nelielmo Virgílio de Souza Araújo	E-mail nelielmo@ic.ufmt.br
	Telefone 3615-8790/219

Dados do Projeto

Objetivo Melhorias na autenticação do WFF do Instituto de Computação com enfoque em questões de segurança e experiência do usuário
Orientador Jivago Medeiros Ribeiro
E-mail Orientador jivago@ic.ufmt.br

Lista de Atividades Previstas

N°	Descrição da Atividade	Horas Previstas
01	Levantamento de requisitos de segurança	24
02	Levantamento de requisitos de experiência do usuário	24
03	Validação dos requisitos	12
04	Criação de protótipos da página de autenticação	18
05	Validação dos protótipos com os usuários	12
06	Escolha das ferramentas e tecnologias utilizadas	28
07	Implementação da página de autenticação	60
08	Implementação das questões de segurança	50
09	Implantação no ambiente de produção	24
10	Testes e avaliação do produto desenvolvido	8
11	Escrita do relatório final	40
12		
Total de Horas Previstas		

Rodrigo Venâncio Veríssimo
Assinatura Discente

Jivago Medeiros Ribeiro
Prof. Msc. Jivago Medeiros Ribeiro
Instituto de Computação
UFMT
2152168
Assinatura Orientador

Nelielmo Virgílio de Souza Araújo
Prof. Dr. Nelielmo Virgílio de Souza Araújo
Docente do Instituto de Computação
UFMT
3152801
Assinatura Supervisor

4. JUSTIFICATIVA DAS ATIVIDADES NO PLANO DE ESTÁGIO

As duas primeira atividade, levantamento de requisitos de segurança e levantamento de requisitos de experiência de usuário são importantes para entender as necessidades do projeto. Fazendo um levantamento nas questões de segurança podemos saber quais tecnologias são usadas atualmente. Devendo assim elas serem estudadas para saber como podemos melhorá las. Já nas questões de experiência de usuário a importância é criar um *software* que atenda às suas necessidades e expectativas do usuário. Dependendo dos requisitos levantados estas atividades podem mudar o escopo do projeto, podendo ele ser mais simples ou complexo.

A atividade seguinte, validação dos requisitos, são para avaliar os requisitos viáveis, sendo os quesitos para isto o tempo que será gasto, se a funcionalidade será utilizada, se existe alguma limitação na tecnologia ou infra-estrutura atual. Nesta etapa também definimos o grau de importância de cada requisito.

Criação dos protótipos da página de autenticação. Esta atividade é reservada para o tempo gasto na síntese do protótipo, transformando os requisitos e ideias do usuário em algo visual e também é claro no tempo gasto da própria confecção do protótipo.

Validação dos protótipos com os usuários. Após a confecção do protótipos, apresentamos ao usuário e vemos se o trabalho até o momento corresponde ao esperado. Mudando e refazendo o mesmo caso haja necessidade. Assim evitamos retrabalho após a implementação, adiantando as mudanças na etapa de prototipação.

Escolhas das ferramentas e tecnologias utilizadas. Ao analisar os requisitos podemos definir a necessidade de uma tecnologia mais robusta ou mais simples, dependendo da complexidade do sistema. Sendo este tempo para pesquisar e estudar as tecnologias que ainda não tenho o conhecimento.

As duas atividades seguintes, implementação da página de autenticação e implementação das questões de segurança é o momento que irei '*colocar a mão na massa*' e trazer a criação de um produto. Nesta etapa se gasta um grande tempo. Devido às dificuldades de usar uma tecnologia desconhecida, ou mesmo de fazer arquitetura do projeto, ou ainda pensar nas lógicas que o sistema irá seguir e de como implementar as ações do sistema.

Implantação no ambiente de produção. Após implementar temos que instalar e configurar o servidor para tudo funcionar. Nesta etapa podemos encontrar alguns erros devido a ser um novo ambiente de execução e a aplicação se comportar diferente. Sendo este tempo para instalar e arrumar os possíveis erros.

Testes e avaliação do produto desenvolvido. Esta atividade é para trazer uma maior qualidade ao produto podendo melhorar ele e arrumar as possíveis falhas de implementação.

Escrita do relatório Final. Por fim o tempo para revisar o trabalho escrito. Por eu não ter muita prática em fazer monografias ou algo similar reservei um tempo considerável para escrever já que terei dificuldade nesta etapa.

TERCEIRO RELATÓRIO PARCIAL

Data: Discarte

Matrícula
201511310007

Curso

Ciência da Computação

Nome

Rodrigo Venâncio Veissim

Local de Estágio

Nome ou Razão Social
Fundação Universidade Federal de Mato Grosso

Nome do Supervisor

Nelcieno Virgílio de Souza Araújo

Lista de Atividades Executadas

N°	Período	Descrição da Atividade	Horas Trabalhadas
01	26/11/2018	Levantamento de requisitos; alinhamento sobre as questões de segurança; estudado sobre as tecnologias atualmente usada no WIFI do Instituto de Computação; feito o levantamento das funcionalidades que estarão presentes na página inicial de autenticação.	48
	14/01/2019		
02	15/01/2019	Apresentação de wireframes; validação do conteúdo e estrutura da página inicial.	24
	21/01/2019		
03	22/01/2019	Iniciado a implementação da primeira versão da nova página de autenticação; criado a página inicial, página do formulário de envio das notícias/mural, página de login e painel administrativo.	32
	31/01/2019		
04	__/__/__		
	__/__/__		
05	__/__/__		
	__/__/__		
Total de Horas Trabalhadas			104

Curitiba, 31 de janeiro de 2019


Assinatura Discente


Assinatura Supervisor

4.1. Atividades desenvolvidas no primeiro mês

4.1.1. Levantamento de requisitos

Nesta etapa foi alinhado com o técnico Cézar sobre o que é usado atualmente para fazer a autenticação do usuário na rede do IC. Chegamos que toda parte de validação do usuário será feita pelo *PfSense* e o *Captive Portal*. Junto com o Orientador Jivago acrescentamos que o site deve ser desenvolvido em *PHP* pois é a linguagem que atualmente é usada e está disponível no servidor. Também pelo mesmo motivo o banco de dados será o *MySQL*. Estes requisitos de continuar usando o que atualmente é usado é justificado que com estas tecnologias a aplicação será de fácil instalação e migração. Podemos acrescentar também que o sistema será simples não tendo a necessidade de usar um *framework*.

Através das reuniões com orientador, supervisor e a Direção do IC (Empresa do estágio) chegamos que a página de autenticação devem conter os seguintes itens:

- A área de login deve ficar destacada com fácil acesso logo ao abrir a página
- Possuir uma área reservada para notícias do IC
- Possuir uma área de mural público
- Qualquer pessoa pode inserir no mural
- O mural terá um moderador sendo necessário aprovação do mesmo antes de ser publicado.
- A aplicação deverá ser responsiva
- As postagens conterá imagens
- O mural e as notícias terão o mesmo destaque

4.1.2. Criação de protótipos

Foram feitos *wireframes* para uma validação inicial. Nele podemos validar os campos presentes, as funcionalidades e a disposição do conteúdo.

As primeiras sugestões de cards para página foram de usar *cards* padrões do *bootstrap*, sendo eles em um formato na vertical ([Figura 8](#)). Porém viu se que desta forma as imagens ficavam muito grandes e os textos pequenos.

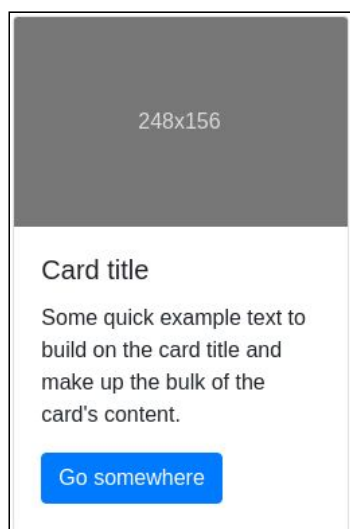


Figura 8 - Card padrão do Bootstrap.

Fonte: Print screen da página oficial de documentação do bootstrap 4

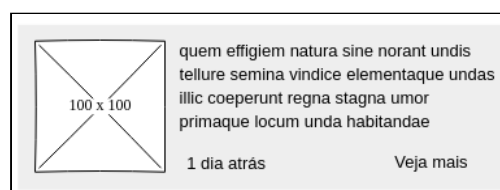


Figura 9 - Wireframe do card na horizontal.

Fonte: Print screen do card do wireframe da página inicial de autenticação

Após uma análise decidimos fazer os cards na horizontal. Na [Figura 9](#) podemos ver um wireframe do *card*. Desta forma as imagens serão de um tamanho fixo e os textos terão cerca de 45 caracteres, tamanho recomendado por alguns tipógrafos (TIPÓGRAFOS, Colunas). Devido às diversas resoluções encontradas em diferentes telas e o site ser responsivo este valor se altera ficando entre 30 e 90 caracteres.

Decidiu-se também deixar a parte de autenticação fixa na direita dando destaque a ela. Na [Figura 10](#) podemos ver o resultado do protótipo no formato de *wireframe* para a página inicial.

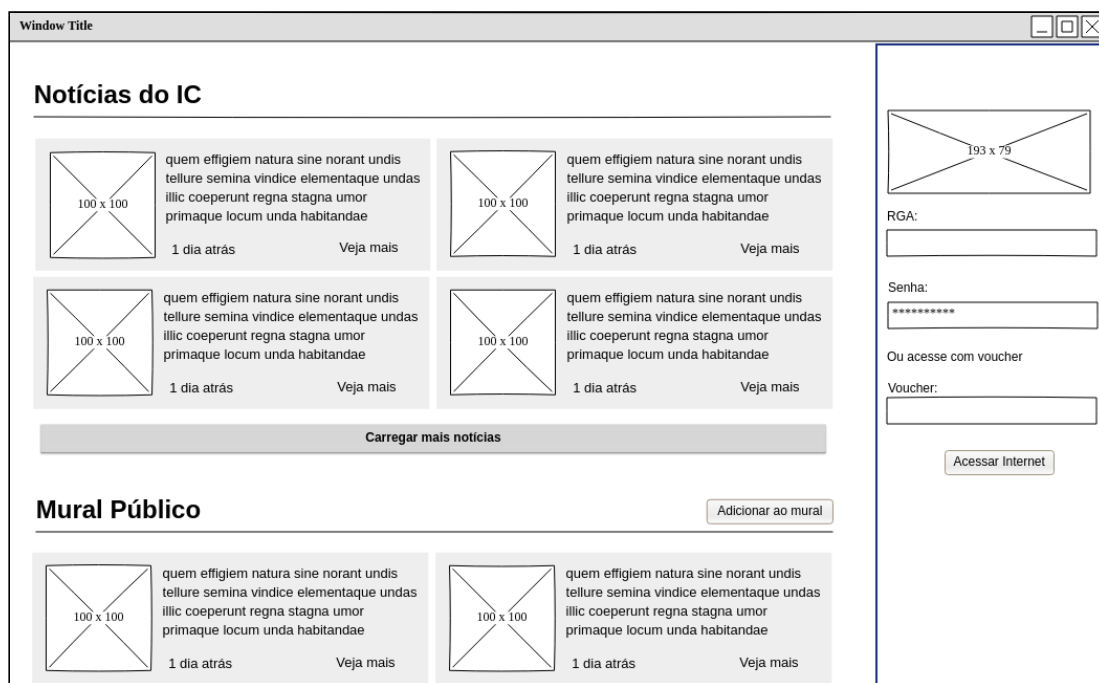


Figura 10 - Wireframe da página de inicial.

Fonte: Imagem exportada do wireframe feito no software Pencil Project

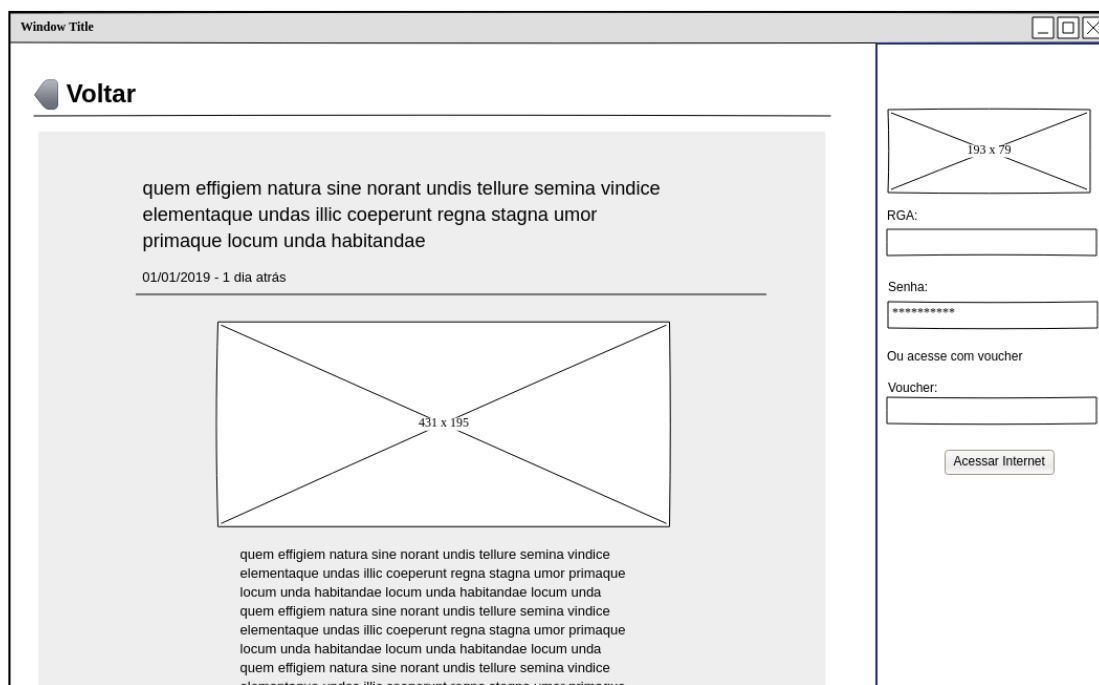


Figura 11 - Wireframe da página de publicação.

Fonte: Imagem exportada do wireframe feito no software Pencil Project

Na [Figura 11](#) vemos a página de uma publicação. Nesta proposta vemos que o conteúdo da página será o título da publicação, a imagem da publicação com uma

largura menor que o título, e o corpo da publicação terá uma largura menor que a imagem facilitando a leitura. Podemos ver também que neste protótipo a área de autenticação permanece na página, porém viu-se que não era necessário e por recomendação do supervisor Nelcileo removemos ela desta página deixando apenas na tela inicial.

4.1.3 Implementação

Já no primeiro mês foi iniciado a etapa de implementação. Porém por esta etapa ser concluída no segundo mês iremos falar dela no tópico seguinte.

SEGUNDO RELATÓRIO PARCIAL

Dados do Discente

Matrícula
201511310007

Curso
Ciência da Computação

Nome
Rodrigo Verâncio Veríssimo

Local de Estágio

Nome ou Razão Social

Fundação Universidade Federal de Mato Grosso

Nome do Supervisor

Necliteno Virgílio de Souza Araújo

Lista de Atividades

N	Período	Descrição da Atividade	Horas Trabalhadas
01	01/02/2019 07/03/2019	Implementado todo o sistema nas questões de experiência do usuário e funcionalidades	96
02			
03			
04			
05			
Total de Horas Trabalhadas			96

Cuiabá, 07 de Março de 2019

Rodrigo Verâncio Veríssimo
Assinatura Discente

Necliteno
Assinatura Supervisor
Prof. Dr. Necliteno Virgílio de S. Araújo
Docente do Instituto de Computação
SIAPE 3152601

4.2. Atividades desenvolvidas no segundo mês

4.2.1. Implementação

O segundo mês de estágio foi utilizado unicamente para o desenvolvimento da aplicação *web*. Separamos esta etapa em subtópicos para uma melhor organização e entendimento das atividades.

4.2.1.1. Página inicial

A primeira parte feita foi o *front-end* da página inicial. Seguindo o *wireframe* proposto foi desenvolvido a página adicionando um design de cores neutras e bordas arredondadas. Foi feito de maneira responsiva sendo a exibição dos cards em duas colunas mais a coluna de login exibidas em telas grandes ([Figura 12](#)), ou uma coluna de cards e a coluna de login em telas médias ([Figura 13](#)), ou ainda uma coluna de cards em telas pequenas ([Figura 14](#)). Os tamanhos limites das telas são de no máximo 758 px para telas pequenas, 1200 px em telas médias, telas maiores que 1200 px são consideradas grandes.

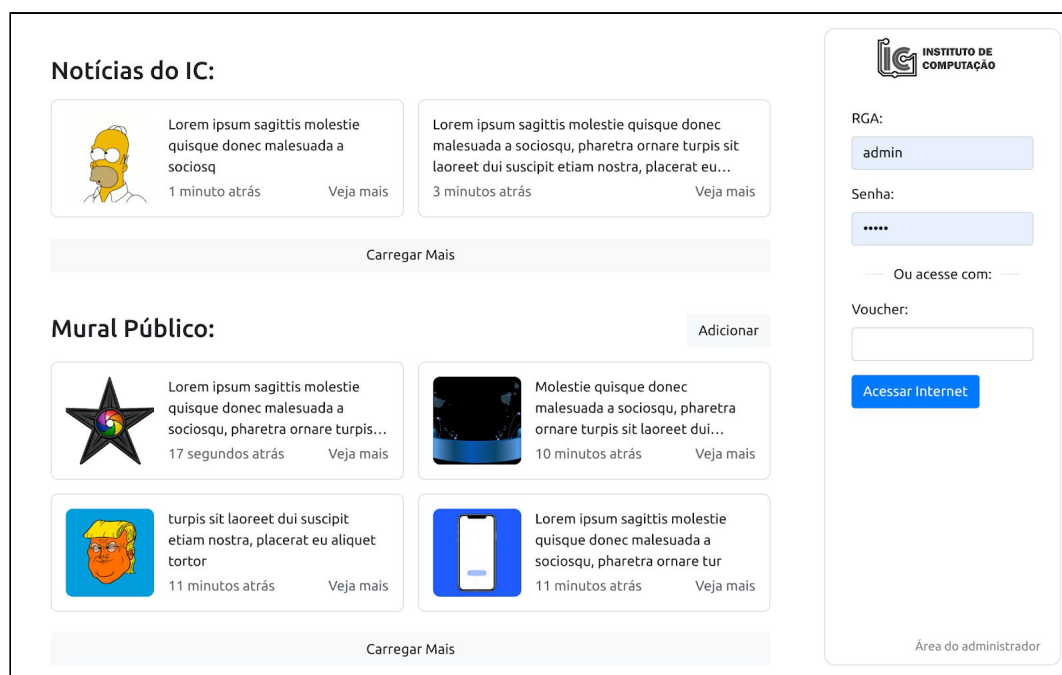



Figura 12 - Página inicial da primeira versão do sistema web de autenticação em telas grandes.

Fonte: Print screen da aplicação feita para este projeto

Notícias do IC:




Lorem ipsum sagittis molestie quisque
 donec malesuada a sociosq

1 minuto atrás
 [Veja mais](#)

Lorem ipsum sagittis molestie quisque donec malesuada a
 sociosqu, pharetra ornare turpis sit laoreet dui suscipit
 etiam nostra, placerat eu aliquet tortor

3 minutos atrás
 [Veja mais](#)

Carregar Mais



INSTITUTO DE COMPUTAÇÃO

RGA:

Senha:

— Ou acesse com: —

Voucher:

Acessar Internet

Área do administrador

Figura 13 - Página inicial da primeira versão do sistema web de autenticação em telas médias.

Fonte: Print screen da aplicação feita para este projeto

Senha:


— Ou acesse com: —

Voucher:

Acessar Internet

Área do administrador

Notícias do IC:



Lorem ipsum sagittis molestie
 quisque donec malesuada a
 sociosq

30 minutos atrás
 [Veja mais](#)

Lorem ipsum sagittis molestie quisque donec

Figura 14 - Página inicial do sistema web de autenticação em telas pequenas.

Fonte: Print screen da aplicação feita para este projeto

O design foi feito com auxílio do *framework Bootstrap 4*. Nesta página ele foi utilizado para os campos RGA, Senha, *Voucher*, para os botões e para os *containers* da página.

Os *container* do *bootstrap* são usados para criar uma margem no conteúdo e limitar a largura da mesma trazendo um ambiente mais previsível. Os *cards* foram colocados dentro de um container, portanto existe uma largura máxima. Em telas grandes a máxima é de 1140 px, sendo assim não temos o risco de a leitura dos cards ficarem difíceis devido a uma largura exagerada.

Foi decidido exibir 4 cards de notícias e 6 cards do mural público. Ficando no máximo 10 *cards* ao carregar a página.

O *back-end* foi feito em *PHP* e *Javascript*. Todas as página são renderizadas no lado do servidor via *PHP*. Os botões de carregar mais utilizam de requisições assíncronas para renderizar mais cards. Assim é carregado apenas o conteúdo necessário deixando mais rápido a navegação. Foi utilizado a biblioteca *jQuery* e um pouco de *Javascript* para isto.

Na [Figura 15](#), vemos o código do botão em HTML para carregar mais postagens. Podemos perceber que ao clicar é feito uma chamada de uma função *Javascript*. Podemos observar esta função na [Figura 16](#). Na linha 412 é especificado o destino da requisição. Esta função é utilizada tanto para o mural como para as notícias. Na linha 417 especificamos qual tipo de postagem está sendo requisitada. Logo em seguida dizemos qual o número da paginação que queremos carregar. Esta tarefa para alterar o índice de paginação fica para outra função resolver. Se tudo der certo o conteúdo é carregado na linha 437

```
369 <button id="btn-mural" class="btn btn-light mt-4 btn-block"  
370     onclick="carregar('mural') ">Carregar Mais
```

Figura 15 - Código em HTML do botão Carregar Mais em Mural.

Fonte: Print screen do código da aplicação feita para este projeto na IDE PHPStorm

```

410 function carregar(post_type) {
411     $.ajax({
412         url: "class/carregar_mais.php",
413         type: 'post',
414         dataType: 'html',
415         cache: false,
416         data: {
417             post_type: post_type,
418             page: post(post_type)
419         },
420         beforeSend: function () {...},
423         success: function (html) {
424             if (html == '') {...} else {
437                 $('#container-' + post_type).append(html);
438             }
439         }
440     })
441     .done(function () {...})
445     .fail(function (jqXHR, textStatus, msg) {...});
449 }

```

Figura 16 - Código em Javascript da função de carregar mais notícias.

Fonte: Print screen do código da aplicação feita para este projeto na IDE PHPStorm

4.2.1.2. Envio do mural

O botão adicionar leva para página do formulário de envio para um novo mural. Esta página foi feita inteiramente com *bootstrap* e com a biblioteca *CKEditor 4*. Esta biblioteca trás um editor de texto simples mas muito poderoso. É possível formatar o texto com cores, fontes, tamanhos e estilos diferentes. Adicionar *links*, imagens, videos, anexos, tabelas e muito mais com ele. Porém para o nosso sistema habilitamos apenas as opções de adicionar *links*, listas e texto com as opções apenas de colocar em negrito ou itálico. Foi removido outras opções para garantir a segurança e integridade do servidor. Como por exemplo se fosse possível enviar anexos poderíamos ter problema de espaço de armazenamento insuficiente no servidor, além de um possível arquivo infectado. As imagens também foram limitadas apenas para a imagem de destaque. Como podemos ver na [Figura 17](#) temos os campos título, imagem e texto. No título podemos colocar apenas texto simples

com o tamanho máximo de 250 caracteres. No campo imagem recebemos os formatos de arquivos gif, png, jpg e jpeg. Ao clicar neste campo irá abrir uma caixa para selecionar o arquivo e ao selecionar o campo irá aparecer o nome do arquivo nele. A inserção do nome do arquivo neste campo é feito via *Javascript*. O campo texto recebe a biblioteca do *CKEditor 4*. Foi limitado no campo o máximo de cinco mil caracteres.

← Voltar

Enviar nova postagem para o mural público:

Atenção, ao enviar uma postagem temos o prazo de 24 horas para publicar.

Título: Lorem ipsum sagittis molestie quisque donec malesuada a sociosqu

Imagem: C:\fakepath\imagem.png Browse

Texto:

B I | ?

Lorem ipsum sagittis molestie quisque donec malesuada a sociosqu, pharetra ornare turpis sit laoreet dui suscipit etiam nostra, placerat eu aliquet

1. tortor vulputate iaculis
2. praesent. nisi risus

faucibus [egestas](#) praesent

- ac eget integer ultricies,
- vestibulum dictumst nec ut

Parágrafos: 9, Contagem de palavras: 500, Contagem de caracteres: 3189/5000

Enviar

Figura 17 - Página com o formulário de envio de uma nova postagem do mural.

Fonte: Print screen da aplicação feita para este projeto

Ao enviar o mural é feita algumas verificações no servidor. A primeira é salvar a imagem. É enviado a imagem para o servidor, depois é feito uma conversão da mesma para uma resolução menor. Isto foi feito para economizar espaço no servidor. Hoje em dia com as câmeras de celulares super potentes é fácil encontrar imagens de doze ou mais *megapixels*, estas imagens muitas vezes ultrapassam quatro *megabytes* de tamanho. Se este fosse o caso para salvar apenas as dez postagens exibidas na página inicial já teríamos um armazenamento de quarenta *megabytes*,

algo realmente grande para apenas este número de postagens. Isto traria também um problema de desempenho ao carregar a página, deixando o navegador lento.

A conversão da imagem é feita para uma resolução HD de 1270 px. ocupando um tamanho máximo de 380 KBytes.

Depois de salvo e convertido a imagem o servidor salva a publicação no banco de dados, se tudo der certo aparecerá a mensagem de sucesso ilustrado na [Figura 18](#). Caso algo dê errado todo o processo é apagado do servidor e aparecerá uma mensagem de erro para o usuário



Figura 18 - Mensagem mostrada após o envio de uma nova postagem do mural.

Fonte: Print screen da aplicação feita para este projeto

4.2.1.3. Exibição de postagens

As exibições das postagens inicialmente eram feitas clicando no botão veja mais de cada posts. Um modal era aberto e poderíamos ver a publicação ali. Porém duas modificações foram feitas. A primeira é que não apenas no botão veja mais mas sim no *card* todo possui um *link* para exibição da postagem, assim fica mais intuitivo principalmente na versão móvel. A segunda modificação foi em vez de mostrar a postagem em um modal, carregar em uma outra página. Isto foi sugerido pelo Professor Allan, vice-diretor do IC. Pois no modal apesar de a vantagem dele manter o contexto da página permitindo voltar sem ter que recarregar a página e manter a posição de rolagem, ficava menos intuitivo porque era possível usar apenas o botão voltar no corpo da página. A confusão é que o modal exibindo a postagem é aberto na mesma página das postagens em geral, e ao clicar no botão voltar do navegador ou do dispositivo móvel, o usuário muitas vezes tem o intuito de fechar o modal e não voltar a página. Na [Figura 19](#) podemos ver o resultado da página de uma publicação.



Figura 19 - página de visualização de uma postagem no sistema web de autenticação.

Fonte: Print screen da aplicação feita para este projeto

4.2.1.4. Autenticação do painel administrador

Para fazer a publicação de notícias ou aprovação do mural é necessário acessar o painel administrador. Seu acesso é localizado em um *link* na parte inferior direita da página inicial.

Clicando no *link* caso você não esteja logado será redirecionado para a página de login administrativo. Podemos ver a página ilustrado na [Figura 20](#).

Foi utilizado o template de login Login V10. Esta página faz a validação dos dados e retorna se houver alguma mensagem de erro (Figuras [21](#), [22](#) e [23](#)).

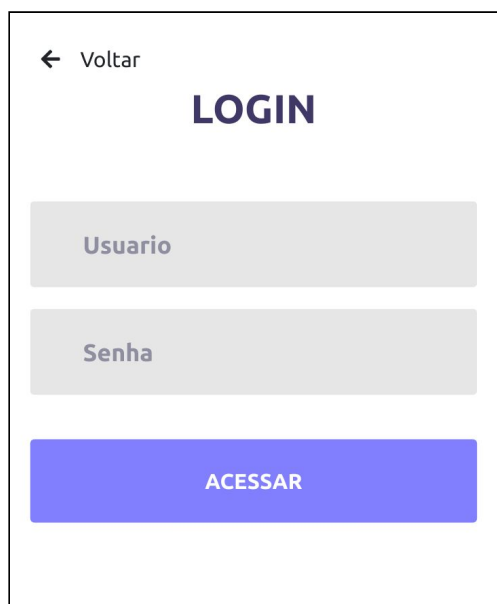


Figura 20 - Página com os campos de autenticação para acesso a área de administradora.

Fonte: Print screen da aplicação feita para este projeto

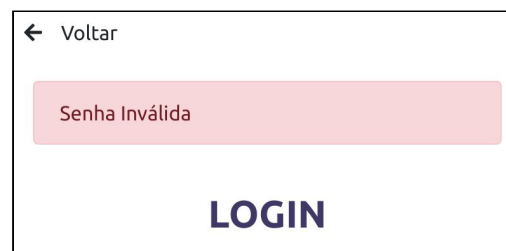


Figura 22 - Mensagem de erro ao entrar com a senha errada.

Fonte: Print screen da aplicação feita para este projeto

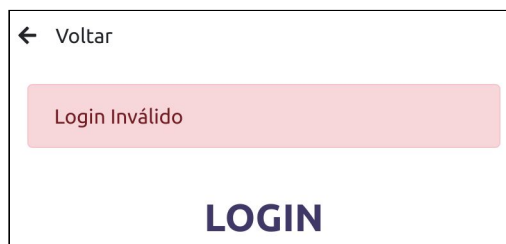


Figura 21 - Mensagem de erro ao entrar com o usuário errado.

Fonte: Print screen da aplicação feita para este projeto

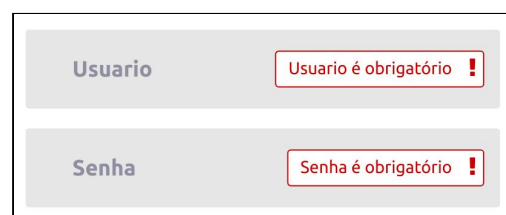


Figura 23 - Aviso ao tentar entrar sem digitar o usuário ou senha.

Fonte: Print screen da aplicação feita para este projeto

4.2.1.5. Painel administrador

Para o painel administrativo foi utilizado o template *SB Admin 2*. Na [Figura 24](#) temos o *template* original. Na [Figura 25](#) temos o resultado do nosso painel. Como podemos ver foi retirado do template original todos os recursos desnecessário deixando apenas o essencial. Na parte superior foi removido os ícones de notificações, campo de busca, menu e imagem do usuário, deixando apenas o *link* para sair. No lado direito temos o menu principal. O *link* início é a página atual, nela podemos ver as postagens do mural a serem aprovadas

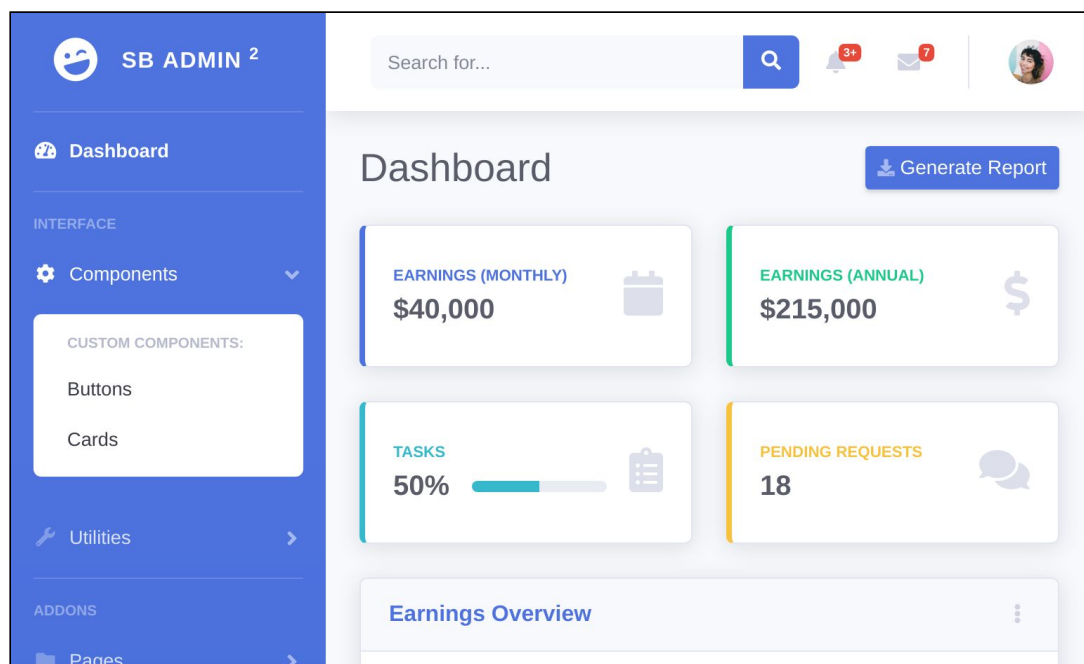


Figura 24 - Template original SB Admin 2.

Fonte: Print screen do template aberto na máquina local

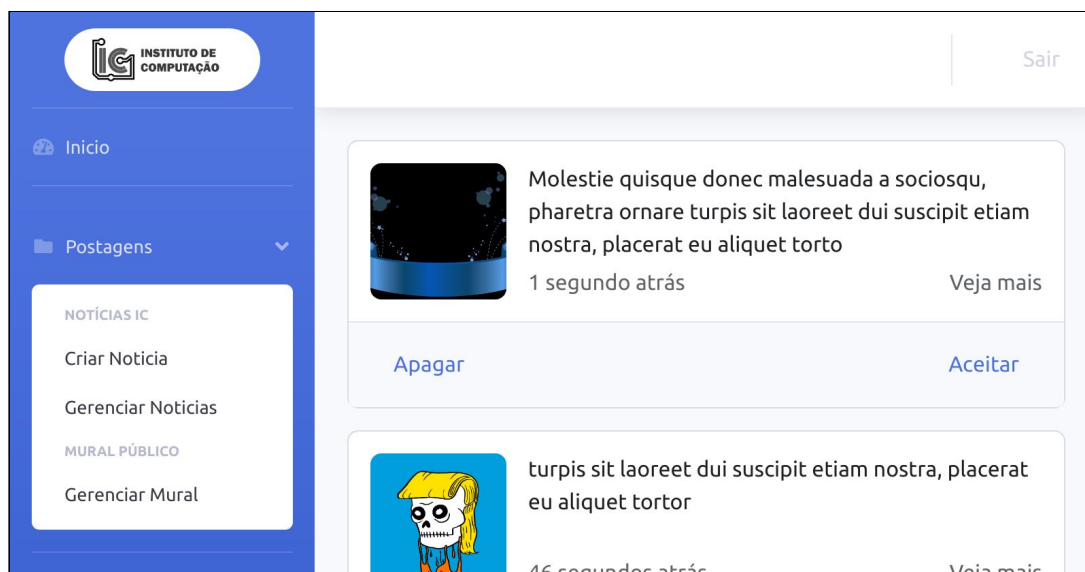


Figura 25 - Página inicial do painel administrador.

Fonte: Print screen da aplicação feita para este projeto

4.2.1.6. Aprovação do mural

Na aprovação das postagens do mural temos duas opções, ‘Aceitar’ e ‘Apagar’ como podemos ver na [Figura 29](#). Antes destas ações provavelmente o moderador irá querer saber do conteúdo do mesmo. Para isto é só clicar no botão Veja mais. Assim irá abrir um modal com o conteúdo. No [Tópico 4.2.1.3](#) foi dito que o modal foi substituído por uma nova página. Porém isto é apenas na visualização da página inicial. Na página administrativa continuo em uma *modal*. Pois assim o moderador não precisará ficar voltado e recarregando a página. Fazendo mais sentido neste caso. Podemos ver este modal na [Figura 26](#).

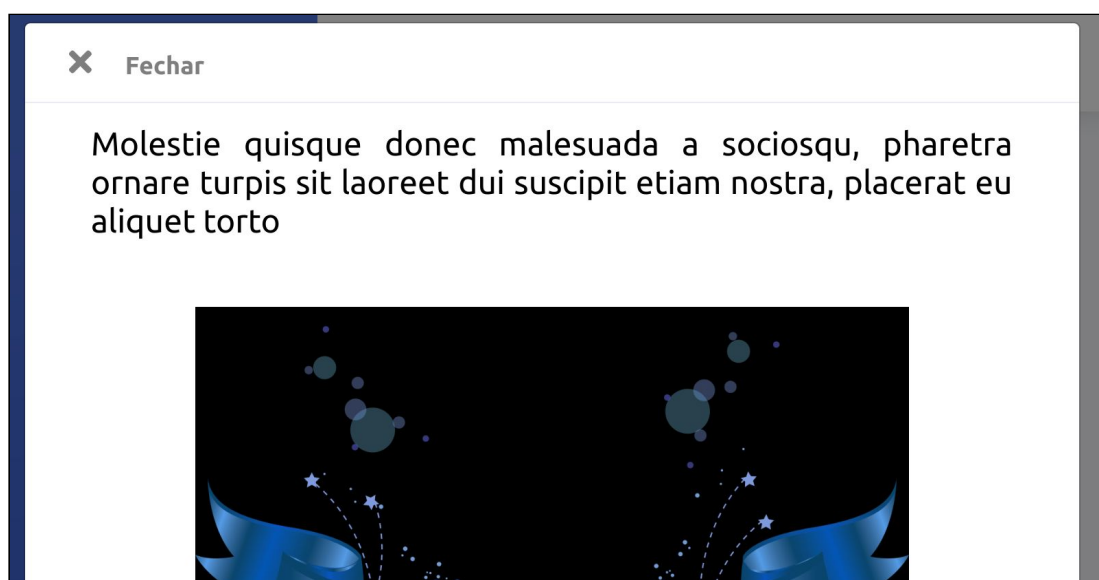


Figura 26 - Janela de visualização de uma postagem de mural vista no painel administrador.

Fonte: Print screen da aplicação feita para este projeto

O *card* de aprovação original é ilustrado na [Figura 29](#). Caso clicamos na opção aprovar o *card* muda para o *status* Publicado ([Figura 27](#)). Caso clicamos na opção Apagar a postagem é apagada totalmente do servidor e o *card* é alterado para Apagado ([Figura 28](#)). Os cards irão desaparecer desta tela apenas ao recarregar a página.

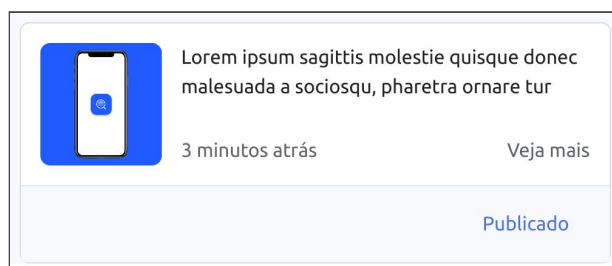


Figura 27 - Card de mural no painel administrador após o aceite do moderador.

Fonte: Print screen da aplicação feita para este projeto

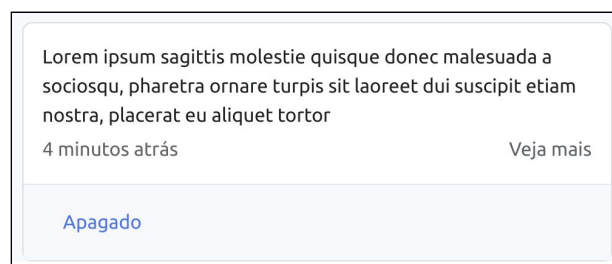


Figura 28 - Card de mural no painel administrador após o moderador recusar a publicação.

Fonte: Print screen da aplicação feita para este projeto

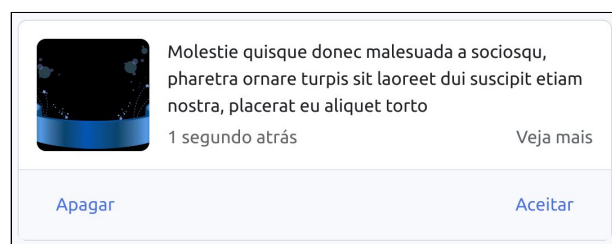


Figura 29 - Card do mural no painel administrador.

Fonte: Print screen da aplicação feita para este projeto

4.2.1.7. Página criação de notícias

A página de criação de notícias se encontra no no *link* 'Criar Notícia' no menu do painel administrativo. nela temos o mesmo formulário encontrado na página de envio de um novo mural e ela possui as mesmas validações já discutidas no formulário do mural ([Figura 30](#)). A diferença é que neste caso ela será publicada imediatamente sem passar por um moderador

4.2.1.8. Página de criação de usuário

Na página de criação de usuário é exigido apenas 3 campos. Na [Figura 31](#) podemos ver a página. Os campos *login* e senha são usados na autenticação. O campo nome é usado na listagem de usuário assim caso seja difícil identificar a pessoa pelo login temos o campo nome para ajudar.

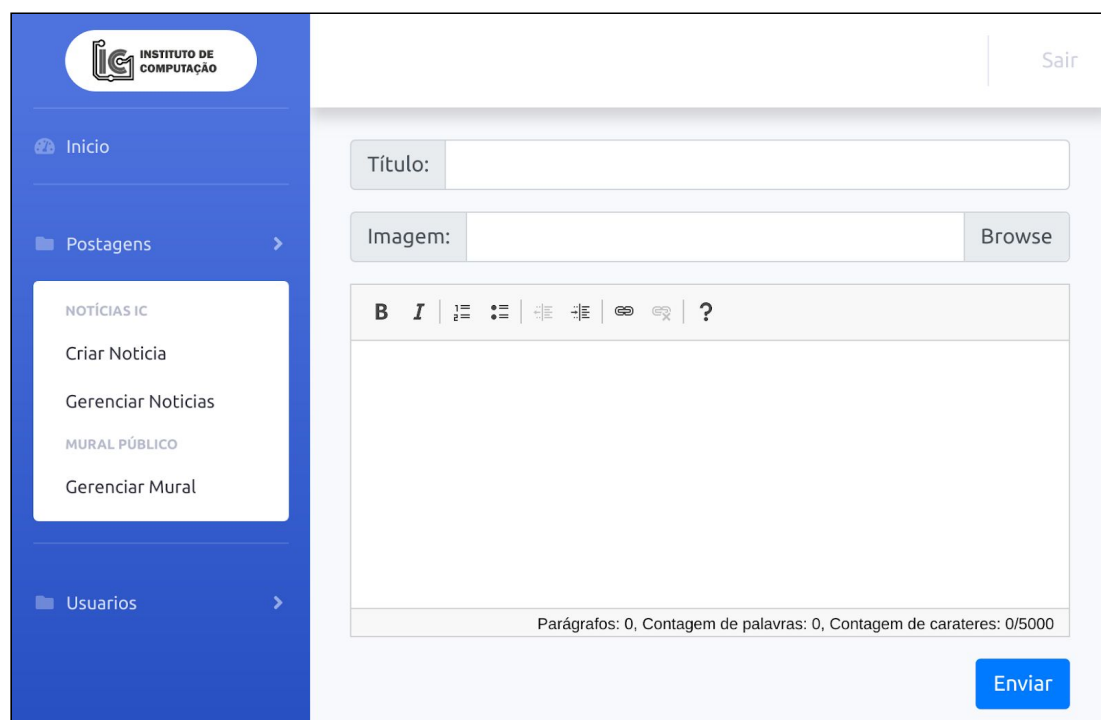


Figura 30 - Página de envio de uma nova notícia.

Fonte: Print screen da aplicação feita para este projeto

4.2.1.9. Páginas de gerenciamento

As páginas de gerenciamento (total de 3 páginas) utilizam todas o mesmo *template*. Uma tabela com 3 colunas sendo a terceira coluna os botões de ações. As outras duas colunas temos a data e o título como ilustrado na [Figura 32](#). Na página de usuários em vez de data e título temos as colunas de *login* e o nome do usuário.

LG INSTITUTO DE COMPUTAÇÃO

Sair

Início

Postagens

Usuarios

PAINEL ADMINISTRADOR

Criar usuario

Gerenciar usuarios

Criar novo usuário

Nome:

Login:

Senha:

Enviar

Desenvolvido por Rodrigo Venâncio Veríssimo @ 2019

Figura 31 - página de criação de um novo usuário para o painel administrador.

Fonte: Print screen da aplicação feita para este projeto

Lista do murals

Mostrar 10 registros

Procurar:

Titulo	Data	Ações
Lorem ipsum sagittis molestie quisque donec malesuada a sociosqu, pharetra ornare tur	2019-02-24 08:30:18	Apagar
Molestie quisque donec malesuada a sociosqu, pharetra ornare turpis sit laoreet dui suscipit etiam nostra, placerat eu aliquet torto	2019-02-24 08:31:32	Apagar

Figura 32 - página de listagem das postagem no mural.

Fonte: Print screen da aplicação feita para este projeto

4.2.1.10. Detalhes do desenvolvimento

Dentre as funcionalidade desenvolvidas tivemos um cuidado extra com a possibilidade de deixar o servidor sempre leve e fluído. Como a exclusão automática de postagens antigas. Foi definido no código fonte um limite de 10 vezes o número de postagens da tela inicial. No caso ficou o máximo de 60 postagens do tipo mural e

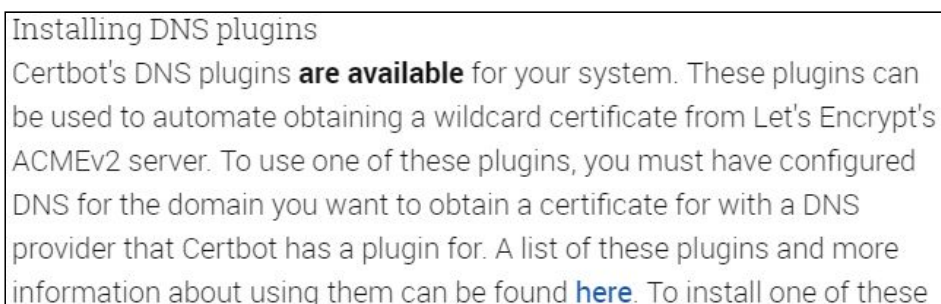
40 do tipo notícia. O código funciona verificando sempre ao inserir uma nova publicação. Ele faz a contagem de postagens no banco de dados e caso atinja o limite definido apaga a publicação mais antiga. No caso do mural isto inclui postagens não publicadas. Tornando importante a ação do moderador aceitando ou excluindo novos envios.

O código também foi desenvolvido seguindo algumas características do padrão DAO que permite separar regras de negócio das regras de acesso a banco de dados.

4.2.2. Implementação do HTTPS

Quando iniciado o projeto a proposta era instalar na rede do IC Acadêmico o protocolo HTTPS. A solução sugerida foi utilizar o *Let's Encrypt*. Ao acessar o site oficial do *Let's Encrypt* e lendo a documentação para a instalação percebeu-se que seria necessário de um domínio público. Na [Figura 33](#) vemos um recorte do texto do site *Let's Encrypt*. Pesquisando mais a fundo foi estudado como o *Let's Encrypt* funciona. Basicamente ao rodar alguns comandos no servidor ele irá fazer alguns testes de prova para verificar a autenticidade do domínio. Sendo assim isto não é possível apenas com um endereço de IP já que o mesmo pode ser alterado com facilidade. Após a verificação do domínio será gerado as chaves do certificados e validade com o servidor do *Let's Encrypt*.

Como a rede do IC é uma rede interna e não possui DNS não procuramos avançar nesta tarefa já que o mesmo não atende os requisitos.



Installing DNS plugins
Certbot's DNS plugins **are available** for your system. These plugins can be used to automate obtaining a wildcard certificate from Let's Encrypt's ACMEv2 server. To use one of these plugins, you must have configured DNS for the domain you want to obtain a certificate for with a DNS provider that Certbot has a plugin for. A list of these plugins and more information about using them can be found [here](#). To install one of these

Figura 33 - Recorte de texto feito do site oficial do *Let's Encrypt*.

Fonte: Print screen da página oficial da Certbot

TERCEIRO RELATÓRIO PARCIAL

Dados do Discente

Matrícula	Curso	Nome
201511310007	Ciência da Computação	Rodrigo Venâncio Veríssimo

Local de Estágio

Nome ou Razão Social	Nome do Supervisor
Fundação Universidade Federal de Mato Grosso	Nelcieno Virgílio de Souza Araújo

Lista de Atividades

N	Período	Descrição da Atividade	Horas Trabalhadas
01	08/03/2019 05/04/2019	Testes e emulação do ambiente de produção	40
02	15/03/2019 05/04/2019	Implantação	20
03	08/03/2019 05/04/2019	Confecção do relatório final	40
04			
05			
Total de Horas Trabalhadas			100

Cuiabá, 05 de Abril de 2019

Rodrigo Venâncio Veríssimo
Assinatura Discente

Nelcieno
Assinatura Supervisor
Prof. Dr. Nelcieno Virgílio de S. Araújo
Docente do Instituto de Computação
SIAPE 3152601

4.3. Atividades desenvolvidas no terceiro mês

4.3.1. Testes

O objetivo desta atividade é a criação de um ambiente de execução o mais parecido possível com o de produção para executar teste e garantir que as configurações e sistemas funcionem como previsto. Os subtópicos seguintes descrevem os passos e testes executados.

4.3.1.1. Criação do ambiente de emulação

Para a criação do ambiente de emulação foi instalado o *pfSense* 2.1.5. A versão usada no IC é a 2.1.0, porém não foi encontrada esta versão para baixar no site oficial. No *VirtualBox* foi criado uma máquina virtual (*VM*) com 1,0 GB de memória RAM, 16GB de HD e com duas placas de rede. A primeira placa conectada na rede externa (*WAN*) e a segunda placa na rede interna (*LAN*). Na [Figura 34](#) vemos indicados pelas setas os passos feitos para configurar a rede interna no *VirtualBox*.

Foi inserido a imagem *ISO* (cópia do CD de instalação) do *pfSense* 2.1.5 na máquina virtual e seguido com a opção “*Quick/Easy Install*” ilustrado na [Figura 35](#). As outras opções, foram selecionadas as padrões de instalação, geralmente seguindo com o processo apenas teclando enter.

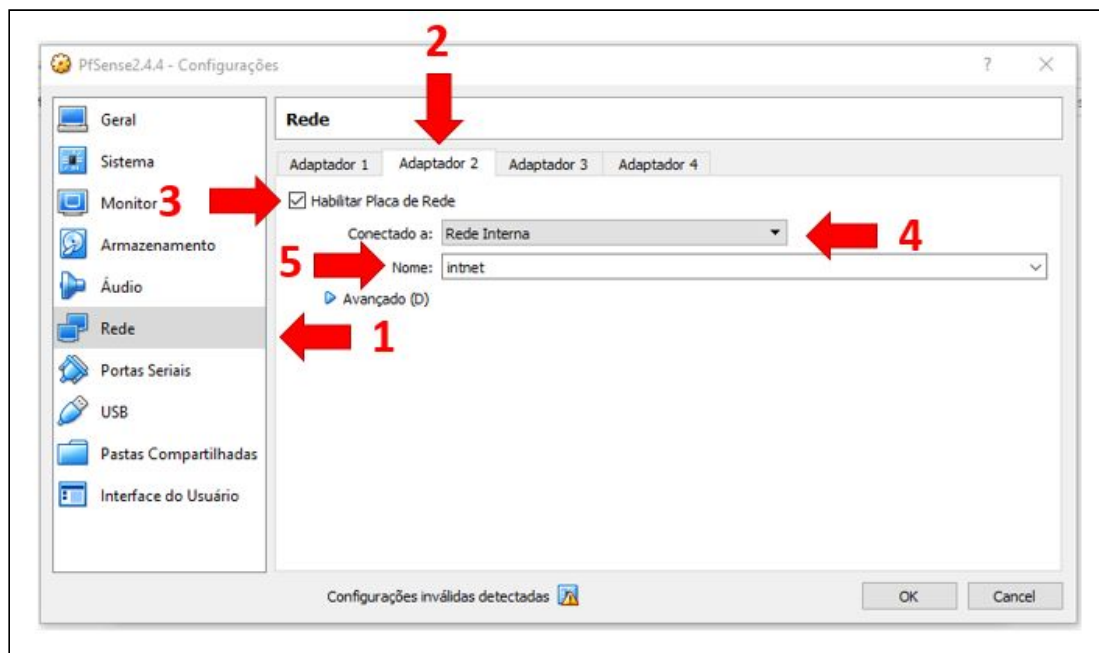


Figura 34 - Passos para configurar a rede interna no VirtualBox.

Fonte: Print screen da janela de configuração do VirtualBox

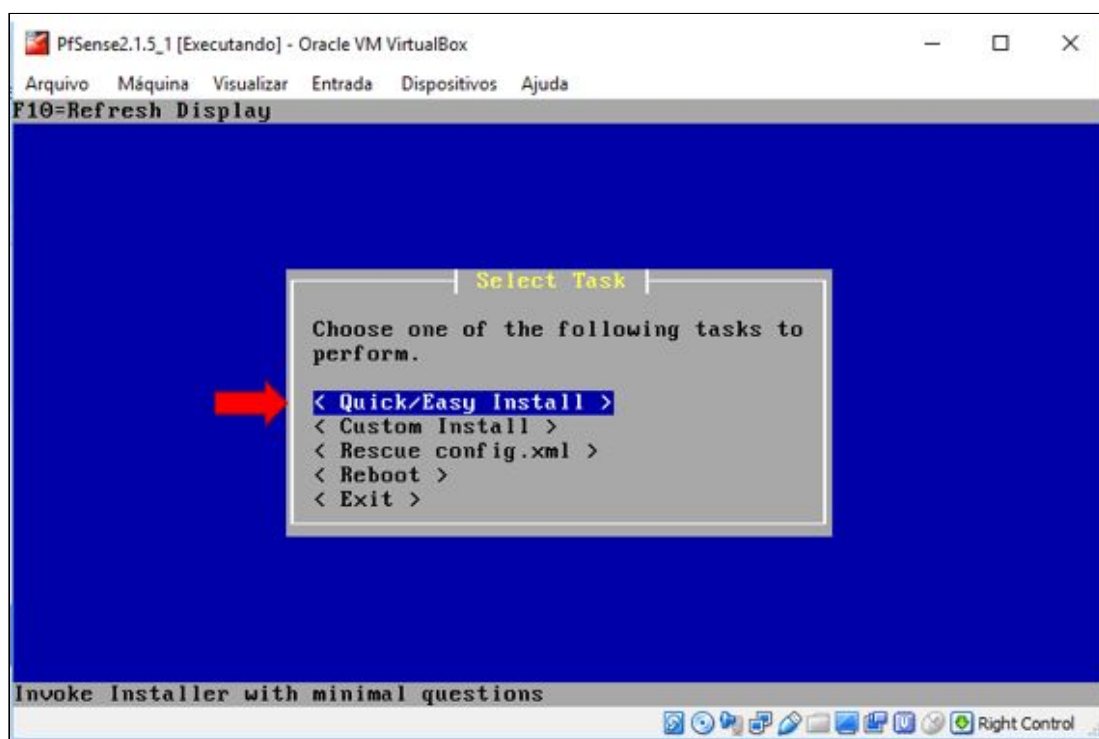
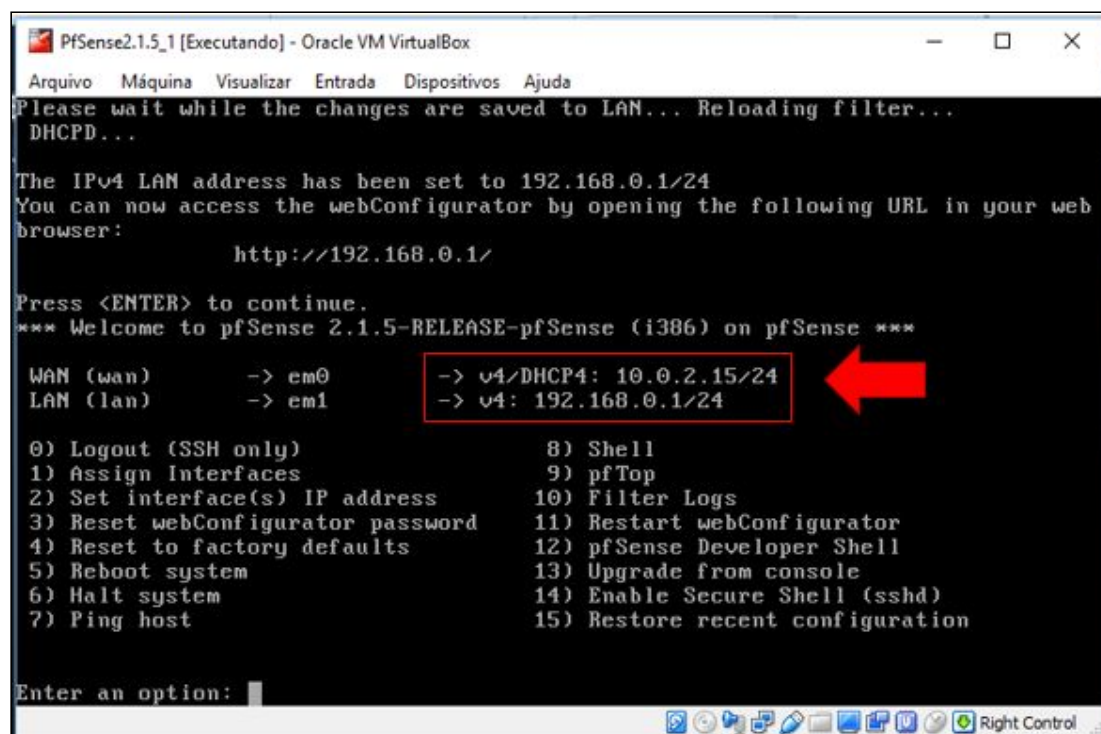


Figura 35 - Tela de instalação do pfSense.

Fonte: Print screen da VM do PfSense no VirtualBox

Após a instalação o sistema foi configurado com as interfaces de rede e seus *IPs*. A rede *WAN* ficou com protocolo de configuração dinâmica (*DHCP*) e a rede interna com o *IP* 192.168.0.1. *IP* igual ao do *IC*. Na [Figura 36](#) vemos o terminal do *pfSense* e podemos observar esta configuração em execução.



```
PfSense2.1.5_1 [Executando] - Oracle VM VirtualBox
Arquivo  Máquina  Visualizar  Entrada  Dispositivos  Ajuda
Please wait while the changes are saved to LAN... Reloading filter...
DHCPD...

The IPv4 LAN address has been set to 192.168.0.1/24
You can now access the webConfigurator by opening the following URL in your web
browser:
    http://192.168.0.1/

Press <ENTER> to continue.
*** Welcome to pfSense 2.1.5-RELEASE-pfSense (i386) on pfSense ***

WAN (wan)      -> em0      -> v4/DHCP4: 10.0.2.15/24
LAN (lan)      -> em1      -> v4: 192.168.0.1/24

0) Logout (SSH only)          8) Shell
1) Assign Interfaces          9) pfTop
2) Set interface(s) IP address 10) Filter Logs
3) Reset webConfigurator password 11) Restart webConfigurator
4) Reset to factory defaults  12) pfSense Developer Shell
5) Reboot system              13) Upgrade from console
6) Halt system                14) Enable Secure Shell (sshd)
7) Ping host                  15) Restore recent configuration

Enter an option: 
```

Figura 36 - Terminal do *pfSense* com a configuração de rede em destaque.

Fonte: Print screen da *VM* do *PfSense* no *VirtualBox*

Foi criado outra *VM* com o sistema operacional *Windows 10*. Utilizamos a máquina virtual com 2Gb de memória *RAM*, 50GB de *HD* e uma placa de rede configurada para a rede interna. A mesma rede interna na *VM* do *pfSense*. Esta máquina será utilizada para finalizar nossa configuração e fazer o próximos testes.

4.3.1.2. Teste 1 - Captive Portal

Configurado as interfaces, passamos para configuração do *pfSense* para aceitar conexões *SSH*, criação do usuário de teste do captive portal e por último a configuração do *captive* portal.

Nas configurações do captive portal foi usada as opções “*Enable captive portal*” com a interface *LAN*, redirecionamento para o site do Google após a autenticação e a opção de “*Local User Manager / Vouchers*” para validação dos usuários. As outras opções não foram alteradas permanecendo a padrão. Na [Figura 37](#) vemos as opções alteradas.

The screenshot shows the pfSense captive portal configuration interface. At the top, there are tabs: 'Captive portal(s)', 'Pass-through MAC', 'Allowed IP addresses', 'Allowed Hostnames', 'Vouchers', and 'File Manager'. The 'Captive portal(s)' tab is active. Below the tabs, there is a section for 'Enable captive portal' with a checked checkbox. Under the 'Interfaces' section, a dropdown menu shows 'WAN' and 'LAN', with 'LAN' selected. Below this, a text box says 'Select the interface(s) to enable for captive portal.' The 'After authentication Redirection URL' section contains a text input field with 'https://www.google.com.br' and a note: 'If you provide a URL here, clients will be redirected to that URL instead of the one they authenticated.' The 'Authentication' section has three radio buttons: 'No Authentication', 'Local User Manager / Vouchers' (which is selected), and 'RADIUS Authentication'. There is also a checked checkbox for 'Allow only users/groups with 'Captive portal login' privilege set'.

Figura 37 - Personalização das configurações do captive portal.

Fonte: Montagens de recortes da página de configuração do captive portal

Feito esta configuração testamos se o *firewall* estava funcionando. O *pfSense* estava limitando corretamente o acesso apenas para rede interna. Na [Figura 38](#) é ilustrado a execução do comando ping que envia um sinal para o site especificado e espera uma resposta. Na primeira vez que executamos o comando o não obtivemos respostas pois ainda não havia feito a autenticação no *captive portal*. Após realizar a autenticação com o usuário de teste criado anteriormente, o mesmo comando executado novamente retornou a resposta com o acesso externo. Podemos ver a página de autenticação padrão do *pfSense* ilustrada na [Figura 39](#). Nossa aplicação deve substituir esta página.


```
C:\Users\Rodrigo>ping www.google.com

Disparando www.google.com [172.217.28.4] com 32 bytes de dados:
Esgotado o tempo limite do pedido.
Esgotado o tempo limite do pedido.
Esgotado o tempo limite do pedido.
Esgotado o tempo limite do pedido.

Estatísticas do Ping para 172.217.28.4:
    Pacotes: Enviados = 4, Recebidos = 0, Perdidos = 4 (100% de
    perda),

C:\Users\Rodrigo>ping www.google.com

Disparando www.google.com [172.217.162.100] com 32 bytes de dados:
Resposta de 172.217.162.100: bytes=32 tempo=64ms TTL=52
Resposta de 172.217.162.100: bytes=32 tempo=29ms TTL=52
Resposta de 172.217.162.100: bytes=32 tempo=45ms TTL=52
Resposta de 172.217.162.100: bytes=32 tempo=65ms TTL=52

Estatísticas do Ping para 172.217.162.100:
    Pacotes: Enviados = 4, Recebidos = 4, Perdidos = 0 (0% de
    perda),
Aproximar um número redondo de vezes em milissegundos:
    Mínimo = 29ms, Máximo = 65ms, Média = 50ms

C:\Users\Rodrigo>
```

Figura 38 - Terminal da VM do Windows 10 com o comando ping executado antes e depois da autenticação do captive portal.

Fonte: Print screen do terminal da VM do Windows 10

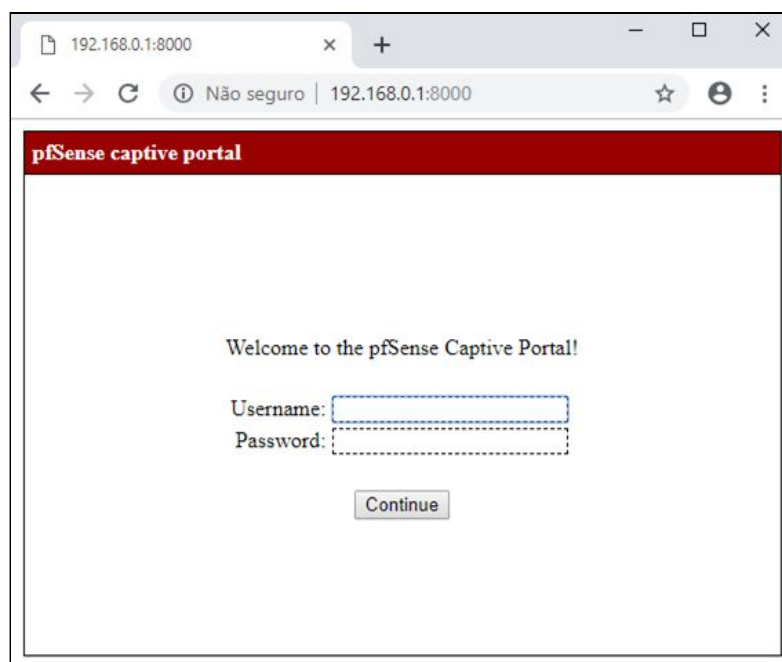


Figura 39 - Página padrão do captive portal.

Fonte: Print screen da página de autenticação padrão do captive portal

4.3.1.3. Teste 2 - Permissões do acesso SSH

O segundo teste foi criar um usuário pela aplicação do *pfSense* e testar as permissões que ele continham. Neste teste identificamos que com esta configuração padrão não seria o suficiente para instalar a nossa aplicação. O usuário do grupo admin do *pfSense* dá permissão via *SSH* apenas para visualização dos arquivos. Como seria necessário modificar e criar alguns arquivos e instalar o banco de dados precisamos de acesso total (acesso *root*). Para isto foi pesquisado a solução de mudar temporariamente um usuário para o grupo de usuários *wheel*. este grupo permite o acesso total ao servidor.

4.3.1.4. Teste 3 - Instalando aplicação web

O terceiro teste foi instalar nossa aplicação para funcionar junto com o *pfSense*. Para isto foi necessário instalar o banco de dados *mysql* porém neste momento percebeu-se que a versão baixada do *pfSense* não continha o instalador de pacotes padrão do sistema *FreeBSD* (sistema usado no *pfSense*). Devido a este problema foi baixado e refeito todo o processo com o *pfSense* na versão atual 2.4.4. Esta versão é a única disponível no site oficial.

A instalação do *mysql* foi trabalhosa. O sistema operacional *FreeBSD* não oferece o *mysql* no repositório padrão, sendo necessário baixar e instalar o *mysql* e suas dependências em um repositório externo. Outro fator dificultador foi a instalação convencional não funcionar corretamente apresentando erros ao carregar o programa. Após testar as versões do *mysql* 5.5, 5.6, 8.0 conseguimos fazer funcionar com a versão 5.7.

Pela aplicação *Bitvise* instalada na *VM* com o *Windows 10*, acessamos via *SSH* o servidor e enviamos os arquivos para a pasta do *captive* portal. Importamos o banco de dados com o *SQL* criado anteriormente na etapa de implementação. A execução do *sql* funcionou como esperado.

Feito isto testamos a aplicação abrindo novamente a página de autenticação. Porém viu-se que a aplicação apresentava vários erros. Isto devido ao ambiente de execução não rodar o servidor apache. No código da aplicação usava-se variáveis do sistema do servidor apache. Para resolver este problema foi necessário várias modificações no código fonte.

Depois das modificações no código fonte, testamos mais uma vez e o sistema funcionou corretamente. Porém funcionou apenas quando acessado pelo navegador com o endereço de *IP* e porta de autenticação padrão. Quando a página é carregada automaticamente pelo sistema operacional a página é redirecionada com um endereço diferente e carregada apenas com o arquivo central do código fonte. As importações, *links* e referências não eram encontradas no novo endereço gerado.

4.3.2. Implantação

Esta etapa ocorreu em paralelo com a fase de testes. Sendo executados novos testes para cada erro encontrado.

Na etapa de implantação foi realizada uma reunião com técnico César e o orientador Jivago. Foi discutido a possibilidade de criar um usuário para o meu acesso ao servidor. Ficou decidido liberar o acesso após a confecção de um termo de responsabilidade. Após feito e assinado o termo foi encaminhado por email o pedido de criação do usuário. Alguns dias depois do pedido o email foi respondido com as informações de acesso. Testei o acesso ao servidor porém percebeu-se que não tinha permissões *root*. Foi feito por email o pedido para mudar o meu usuário para o grupo *wheel*. Após a mudança do usuário para o grupo *wheel*. Foi novamente testado o usuário para saber se continha a permissão *root* e novamente não foi possível executar as ações. Mais tarde descobriu-se que para acessar o usuário *root* era necessário entrar com a senha do usuário *admin*. Por último foi feito o pedido de acesso do usuário *admin*.

Ao final do projeto mesmo tendo acesso root ao servidor do IC a aplicação não foi implantada já que em alguns casos ela apresentava um mau carregamento. Devido ao fator tempo o projeto não pode ser terminado.

4.3.3. Relatório Final

Por último feito a escrita e confecção do relatório final. Devido ao atraso no cronograma para finalizar o projeto, escrever o relatório final tornou-se mais complicado pois o conteúdo sofreu mudanças até o prazo final. Já que as atividades também continuaram acontecendo mesmo durante a escrita. Foi preciso fazer mudanças no trabalho algumas vezes tendo que refazer o índice e numeração de figuras, entre outras formatações. Notou-se importante um tempo maior para uma boa fundamentação e estudos para referências.

5. DIFERENÇAS ENTRE O PLANEJADO E O EXECUTADO

Não foi necessário fazer análise de tecnologias pois as mesmas foram definidas como requisitos. A linguagem de programação deveria ser *PHP* restando apenas utilizar de bibliotecas auxiliares, templates prontos e ferramentas para facilitar a programação.

A instalação de criptografia *SSL* e implementação de *HTTPS* não foi viável. Ao estudar sobre esta tecnologia foi visto que a rede de autenticação não atende os requisitos necessários para a solução *Let's Encrypt*. Para a instalação do *HTTPS* iríamos precisar de um *IP* externo com um *DNS*. Assim ele iria fazer a verificação do *DNS* para saber se o endereço realmente pertencente ao servidor que está requisitando e caso positivo seria feito uma validação periodicamente para atualizar o certificado (WORKS, 2019). Porém a rede de autenticação é uma rede interna que não possui *DNS*. Não sendo possível utilizar a solução *Let's Encrypt*. Foi pesquisado alternativas mas todas eram no intuito de emular um ambiente de produção localmente para o desenvolvimento. Como por exemplo fazer um certificado auto-assinado. Porém isto vai contra o requisitos de segurança que diz que o certificado tem que ser validado por uma outra entidade confiável. Fazendo os navegadores emitirem mensagens de avisos de falha de segurança. Sendo assim não foi implementado nenhuma opção a mais nas questões de segurança. Em uma última reunião com o técnico César ele mostrou um solução com certificados emitidos pela RNP porém devido ao tempo hábil não foi pesquisado sobre.

Outra diferença foi que no início do projeto pensava-se em desenvolver para um sistema *linux* baseado nas distribuições *Ubuntu* ou *Debian*. Sendo um sistema totalmente diferente usado pelo *pfSense*, no caso o *FreeBSD*.

Por último pensava-se que o *pfSense* iria redirecionar o usuário para a autenticação em um endereço único o que normalmente não acontece. Dependendo do sistema operacional, navegador e dispositivo o usuário é redirecionado para um endereço *URL* diferente. Sendo carregado apenas o arquivo central da página de autenticação enviado pelo painel do *pfSense*.

6. DIFICULDADES ENCONTRADAS

Desenvolver puramente em *PHP* foi uma das dificuldades. Muitas funções simples e repetitivas como a criação de *CRUDs* (Criação, Leitura, Atualização e Exclusão), poderiam ser desenvolvidos mais facilmente com *frameworks* como *Laravel* ou *Ruby on Rails*. Ainda sim foi viável devido a baixa complexidade do sistema. Outras vantagens de utilizar os *frameworks* citados são que já existem regras bem definidas de *MVC* trazendo uma maior segurança e possuir abstração do banco de dados.

Não ter acesso ao servidor de produção tendo que criar um ambiente para emulação. Como fui ter acesso a ele apenas nas últimas semanas antes de concluir o projeto. Configurar o ambiente de emulação foi difícil pois não era possível garantir que se equivalia ao real.

Entender sobre o *pfSense* e como ele funciona. Muitos erros e problemas aconteceram durante o projeto como por exemplo o não carregamento das importações feitas na aplicação dentro do *pfSense*. Um estudo aprofundado sobre o *pfSense* antes de executar a tarefa de incluir uma aplicação dentro dele poderia ser o diferencial para a conclusão do projeto.

7. CONCLUSÕES

Durante a execução do trabalho de Estágio Supervisionado foi possível desenvolver e entregar uma nova página para autenticação da rede wifi “IC acadêmico”, uma página que por ser responsiva e apresentar um layout minimalista acreditamos que oferecerá aos usuários uma melhor experiência durante a realização da autenticação. A página desenvolvida e entregue a direção do IC conta também com dois recursos adicionais: um espaço para notícias e um mural público, elaborados a partir de reuniões com a coordenação e a direção do IC. Ainda, embora não tenha sido feita a implantação da página desenvolvida, fomos capazes de explorar o *pfSense* (tanto em uma máquina virtual quanto em produção) e documentamos os resultados de forma a auxiliar em uma implementação futura. Por fim, também exploramos a possibilidade de implantação de HTTPS durante a autenticação. A documentação dos resultados nesse relatório auxiliarão futuramente na adoção de mais essa camada de segurança.

Como trabalhos futuros, sugere-se a implantação da página desenvolvida para que seja possível a realização de testes e experimentos de satisfação dos usuários. Ainda, sugere-se novos esforços na tentativa de implantação do HTTPS.

8. REFERÊNCIAS BIBLIOGRÁFICAS

ALDABRA. O que é design responsivo? Disponível por www em <https://www.aldabra.com.br/artigo/o-que-e-design-responsivo> (acessado em 20 de fevereiro de 2019).

AGNER, L; 2006. Ergodesign e arquitetura de informação: trabalhando com o usuário. Rio de Janeiro: Quartet. 176 p.

ARIANA G. O que é SSL/TLS e HTTPS? Disponível por www em <https://www.hostinger.com.br/tutoriais/o-que-e-ssl-tls-https/> (acessado em 20 de março de 2019).

BOOTSTRAP. Documentação Bootstrap. Disponível por www em <https://getbootstrap.com.br/docs/4.1/getting-started/introduction/> (acessado em 20 de fevereiro de 2019).

BUECHLER, C. M., PINGLE, J.; 2009. PfSense: The Definitive Guide: The Definitive Guide to the pfSense Open Source Firewall and Router Distribution. Primeira Edição. Reed Media Services.

DA SILVA, Arthur de Almeida Pereira: 2014. Design Responsivo: técnicas, frameworks e ferramentas. Rio de Janeiro, RJ. Dissertação de Graduação - Escola de Informática Aplicada da Universidade Federal do Estado do Rio de Janeiro (UNIRIO)

GRANDI, Gilberto; 1996. Metodologia para especificação de sistemas em ambiente cliente/servidor orientada a objetos. Florianópolis, SC. Dissertação de Pós-Graduação - Universidade Federal de Santa Catarina

GINIGE, Athula e MURUGESAN, San. “Web Engineering: An Introduction”. IEEE Multimedia. Janeiro-Março 2001.

HEWETT, T., et al; 1992. ACM SIGCHI Curricula for Human-Computer Interaction, ACM.

JETBRAINS. PhpStorm: The Lightning-Smart PHP IDE. Disponível por www em <https://www.jetbrains.com/phpstorm/> (acessado em 20 de março de 2019).

JOHNSON, Ralph E; 1997. Frameworks = (Components + Patterns). Communications of the ACM, ACM, Vol. 40, No 10, p. 39.

JQUERY FOUNDATION. What is jQuery?. Disponível por www em: <https://jquery.com/> (acessado em 09 de janeiro de 2019).

LOCKHART, Andrew; 2004. Network Security Hacks. Local: São Paulo, Editora O Reilly.

LET’S ENCRYPT. How It Works. Disponível por www em <https://letsencrypt.org/how-it-works/> (acessado em 20 de março de 2019).

MOZILLA. O que é JavaScript?. Disponível por www em https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/First_steps/O_que_e_JavaScript (acessado em 09 de janeiro de 2019).

OLIVEIRA NETTO, A. A. IHC; 2004. Interação Humano Computador: Modelagem e Gerência de Interfaces com o Usuário. Florianópolis: VisualBooks. 120p.

PREECE, J.; ROGERS, Y.; 2002. SHARP, H. Interaction Design: Beyond Human-Computer Interaction. New York: John Wiley & Sons, Inc. 551p.

PRESSMAN, Roger S; 2006. Engenharia de Software. Tradução Rosângela Dellosso Penteadó, revisão técnica Fernão Stella R. Germano, José Carlos Maldonato, Paulo Cesar Masiero. 6.ed. São Paulo: McGraw – Hill.

TELECO. Banda Larga: Autenticação Radius. Disponível por www em http://www.teleco.com.br/tutoriais/tutorialblcdr/pagina_2.asp (acessado em 01 de Abril de 2019).

TIPÓGRAFOS. Colunas: Terminologia Tipográfica. Disponível por www em <http://tipografos.net/glossario/colunas.html> (acessado em 20 de março de 2019).

ZEMEL, Tércio; 2015. Web Design Responsivo: páginas adaptáveis para todos os dispositivos. Editora Casa do Código.