

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE SISTEMAS DE INFORMAÇÃO – BACHARELADO

APLICAÇÃO WEB PARA GERENCIAMENTO DE
MENSAGENS DE DIFERENTES SERVIÇOS DE E-MAIL

ANDERSON RISSARDI

BLUMENAU
2016

ANDERSON RISSARDI

**APLICAÇÃO WEB PARA GERENCIAMENTO DE
MENSAGENS DE DIFERENTES SERVIÇOS DE E-MAIL**

Trabalho de Conclusão de Curso apresentado ao curso de graduação em Sistemas de Informação do Centro de Ciências Exatas e Naturais da Universidade Regional de Blumenau como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informação.

Prof. Samuel Cristhian Schwebel, Mestre - Orientador

**BLUMENAU
2016**

APLICAÇÃO WEB PARA GERENCIAMENTO DE MENSAGENS DE DIFERENTES SERVIÇOS DE E-MAIL

Por

ANDERSON RISSARDI

Trabalho de Conclusão de Curso aprovado para obtenção dos créditos na disciplina de Trabalho de Conclusão de Curso II pela banca examinadora formada por:

Presidente: _____
Prof. Samuel Cristhian Schwebel, Mestre – Orientador, FURB

Membro: _____
Prof. Francisco Adell Péricas, Mestre – FURB

Membro: _____
Prof. Gabriele Jennrich Bambineti, Especialista – FURB

Blumenau, 08 de Dezembro de 2016

Dedico este trabalho à minha família, meus amigos e todos que contribuíram para a realização deste e de toda a minha formação acadêmica.

AGRADECIMENTOS

A Deus por me dar a oportunidade e a força para realizar este trabalho.

À minha família por sempre me apoiarem em meu objetivo.

Aos meus amigos que sempre me ajudaram e estiveram lá quando precisei deles.

Ao meu orientador Samuel Schwebel que sempre esteve presente e disposto a me auxiliar e por nunca duvidar da realização deste trabalho.

RESUMO

Este trabalho apresenta uma aplicação Web de leitura e envio de mensagens de diferentes serviços de e-mail que são acessados através dos protocolos SMTP e IMAP. Disponibiliza um ambiente Web que possui interfaces com design responsivo, tornando a aplicação independente do dispositivo utilizado para realizar o acesso, exigindo apenas um navegador de internet. A aplicação permite o cadastro de contatos e a realização de pesquisas avançadas em todas as mensagens de todas as contas de e-mail cadastradas. No desenvolvimento da interface utilizou-se as bibliotecas de estilo Bootstrap e AdminLTE, juntamente com a tecnologia do HTML5 permitindo assim atingir o design responsivo. A aplicação foi desenvolvida utilizando o modelo MVC em que o lado servidor da aplicação foi desenvolvido utilizando C# e a persistência realizada no banco de dados MSSQL Server 2014. Para realizar o acesso aos diferentes serviços de e-mail são utilizados os componentes EAGetMail e EASendMail que executam em ambiente .Net e possuem código gerenciado. Como resultado obteve-se uma aplicação Web, responsiva a fim de agilizar e simplificar o acesso há diversos serviços de e-mail.

Palavras-chave: Web. Responsiva. E-mail. MVC.

ABSTRACT

This work presents a Web application for reading and sending messages from different e-mail services that are accessed through SMTP and IMAP protocols. It provides a Web environment that has interfaces with responsive design, making the application independent of the device used to access, requiring only a web browser. The application allows you to register contacts and perform advanced searches on all messages from all registered email accounts. In the development of the interface used whether the style library Bootstrap and AdminLTE, together with the HTML5 so as to achieve responsive design. The application was developed using the MVC model in which the server side of the application was developed using C# and the persistence performed in the MSSQL Server 2014 database. To access different e-mail services, the EAGetMail and EASendMail components that run in .Net environment and have managed code. As a result, we obtain a Web application, responsive to speed and simplify access there are several e-mail services.

Key-words: Web. Responsive. E-mail. MVC.

LISTA DE FIGURAS

Figura 1 – Exemplo da tela inicial do Microsoft Outlook.....	18
Figura 2 – Exemplo da tela de envio de mensagens do Outlook.....	18
Figura 3 – Exemplo da tela inicial do Gmail.....	19
Figura 4 – Exemplo da tela de leitura de mensagens do Gmail	20
Figura 5 – Diagrama de casos de uso de configurações	23
Figura 6 – Diagrama de casos de uso gerais do sistema.....	24
Figura 7 – Diagrama de implantação.....	25
Figura 8 – Diagrama UML dos principais controles	27
Figura 9 – As views da aplicação.	28
Figura 10 – Disposição de botões na resolução extremamente pequena.....	33
Figura 11 – Disposição de botões a partir da resolução pequena.....	33
Figura 12 – Tela inicial da aplicação.....	37
Figura 13 – Tela de cadastro de usuário.	37
Figura 14 – Tela de login da aplicação.....	38
Figura 15 – Tela de exibição das contas de e-mail.....	39
Figura 16 – Tela de cadastro de conta	39
Figura 17 – Tela de exibição de contatos	40
Figura 18 – Tela de cadastro de contatos.	41
Figura 19 – Tela de edição de contas de e-mail.	41
Figura 20 – Tela de visualização de mensagens das contas monitoradas.	42
Figura 21 – Tela de leitura de mensagem.....	42
Figura 22 – Tela de responder mensagem de e-mail.....	43
Figura 23 – Tela de envio de mensagens.....	43
Figura 24 – Tela de visualização de contas em dispositivos extremamente pequenos	44
Figura 25 – Tela de pesquisa avançada de e-mails.....	45
Figura 26 – Tela de perfil do usuário.	45

LISTA DE QUADROS

Quadro 1 – Principais características EAGetMail.....	15
Quadro 2 – Requisitos funcionais.....	22
Quadro 30 – Características dos trabalhos correlatos.....	47

LISTA DE ABREVIATURAS E SIGLAS

CSS – Cascade Style Sheets

GMAIL – Google Mail

HTML – Hyper Text Markup Language

IMAP – Internet Message Access Protocol

MVC – Model View Controller

POP – Post Office Protocol

RFC – Request for Change

SMTP – Simple Mail Transfer Protocol

SSL – Secure Sockets Layer

TCP – Transmission Control Protocol

TLS – Transport Layer Security

SUMÁRIO

1 INTRODUÇÃO.....	11
1.1 OBJETIVOS.....	12
1.2 ESTRUTURA.....	12
2 FUNDAMENTAÇÃO TEÓRICA	13
2.1 SERVIÇO E PROTOCOLOS DE E-MAIL.....	13
2.2 COMPONENTES DE ENVIO E RECEBIMENTO DE E-MAIL.....	14
2.2.1 EAGetMail POP3 e IMAP4 Component	14
2.2.2 EASendMail SMTP Component.....	15
2.3 DESIGN RESPONSIVO.....	15
2.4 TRABALHOS CORRELATOS	16
2.4.1 Ferramenta para controle de e-mails e anti-spam	16
2.4.2 Microsoft Office Outlook.....	17
2.4.3 Mozilla Thunderbird	18
2.4.4 Google Mail (Gmail).....	19
3 DESENVOLVIMENTO DA APLICAÇÃO.....	21
3.1 INFORMAÇÕES	21
3.2 ESPECIFICAÇÕES	22
3.2.1 Requisitos.....	22
3.2.2 Diagramas	23
3.3 IMPLEMENTAÇÃO	26
3.3.1 Modelo de desenvolvimento	26
3.3.2 Técnicas e ferramentas utilizadas.....	29
3.3.3 Operacionalidade da implementação	36
3.4 RESULTADOS E DISCUSSÕES.....	46
4 CONCLUSÕES.....	48
4.1 EXTENSÕES	48
APÊNDICE A – DESCRIÇÃO DOS CASOS DE USO.....	53

1 INTRODUÇÃO

Atualmente a disseminação da informação atingiu uma velocidade nunca antes vista, uma bomba explode no outro lado do mundo e em minutos, após alguns poucos cliques, é possível estar ciente de todo o acontecimento. Isso só foi possível com o advento da internet, que possibilitou a comunicação através dos mais diversos meios.

Um dos serviços eletrônicos mais utilizados pelas pessoas para se comunicar é o e-mail. Segundo The Radicati Group (2015, p. 3, tradução nossa), no final de 2015, o tráfego total de e-mails no mundo foi estimado em mais de 205 bilhões de mensagens por dia.

Apesar de nos seus primórdios o e-mail servir apenas para a recepção e envio de mensagens (CARNEIRO, 2008, p. 2), atualmente o serviço de e-mail possui suas próprias características e funcionalidades. Segundo Carneiro (2000, p.2), “o e-mail deixou de servir apenas como ferramenta de comunicação assíncrona e passou a incorporar muitas outras funcionalidades fora do âmbito original para o qual foi criado.” Isto exigiu que várias aplicações fossem criadas para acessar esses serviços de e-mail, atendendo assim as características específicas de cada serviço.

Atualmente, é comum as pessoas possuírem várias contas de e-mails nos mais diferentes domínios; um para o trabalho, um para uso pessoal, outro de uso acadêmico entre outros e que constantemente, verifiquem as mensagens destas contas. Contas de e-mails nos mais diferentes domínios implicam em diferentes meios de acesso a esses serviços, sistemas *webmail*, aplicações *desktop*, aplicativos para dispositivos móveis entre outros. Segundo The Radicati Group (2015, p. 3, tradução nossa), em 2015, cada usuário possui, em média aritmética, 1,7 contas de e-mail e no final de 2019 a média será de 1,9 conta de e-mail por usuário.

Com o aumento da quantidade de serviços prestados pelos sistemas *webmail* atuais, rapidamente verificou-se um novo fenômeno denominado Email Overload que é descrito como o acúmulo de funções do e-mail e o fato do usuário não conseguir gerir o seu e-mail (CARNEIRO, 2000, p. 2). O fato de ter que acessar diferentes aplicações, com diferentes contas exige muito tempo e esforço do usuário o que acaba se tornando de certa forma maçante para ele.

No dia-a-dia, acessar diversos sistemas diferentes com contas e senhas diferentes para verificar se há novas mensagens, torna-se um esforço e tempo desprezível considerável. Algumas vezes, se faz necessário trocar de plataforma, quando determinados recursos não são

atendidos, por exemplo, quando aplicações *webmail* não funcionam de forma responsiva, ficando praticamente inviável acessá-los de dispositivos móveis.

Visando agilizar e simplificar o processo de leitura e envio de mensagens de e-mail este trabalho propõe o agrupamento de várias contas e serviços de e-mail em apenas um sistema Web com interface responsiva, facilitando o acesso e o controle dessas mensagens de e-mail, independentemente do meio de acesso ao sistema sem se preocupar com o dispositivo utilizado para realizar o acesso, bem como sem a necessidade de realizar o download das mensagens, sem ter que realizar a instalação de programas terceiros e independentemente de sistema operacional.

1.1 OBJETIVOS

O objetivo geral do trabalho proposto é o desenvolvimento de uma aplicação Web que reúna mensagens de vários serviços de e-mail diferentes, com domínios diferentes e que permita ler e enviar mensagens de e-mail.

Os objetivos específicos do trabalho proposto são:

- a) acessar diferentes serviços de e-mail através dos protocolos SMTP (Simple Mail Transfer Protocol) e IMAP (Internet Message Access Protocol);
- b) permitir o envio e recebimento de mensagens de e-mail em um ambiente Web e responsivo;
- c) permitir o acesso através dos navegadores de internet de maneira responsiva, que permita o acesso a partir de dispositivos móveis, *desktops* e notebooks.

1.2 ESTRUTURA

No primeiro capítulo tem-se a introdução da temática principal deste trabalho com a identificação dos problemas e a apresentação da justificativa e dos objetivos deste trabalho.

No segundo capítulo apresenta-se a fundamentação teórica pesquisada que apresenta sobre a história do correio eletrônico, os principais protocolos do correio eletrônico, interfaces responsivas e as bibliotecas de envio e recebimento de mensagens de e-mail além de trabalhos correlatos.

No terceiro capítulo é apresentado o desenvolvimento da aplicação, mostrando os requisitos, o diagrama de casos de uso e o diagrama de implantação. Após essa apresentação é detalhado o desenvolvimento da aplicação e a operacionalidade da aplicação. Finalizando com os resultados obtidos. Por fim, no quarto capítulo apresentam-se as conclusões e extensões deste trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo aborda o serviço de e-mail e os protocolos utilizados para fazer uso desse serviço. O componente EAGetMail realiza a conexão com serviços de e-mail e o EASendMail realiza o envio de mensagens. O modelo de desenvolvimento MVC e o conceito de design responsivo. Também são abordados trabalhos correlatos a este.

2.1 SERVIÇO E PROTOCOLOS DE E-MAIL

Segundo Carneiro (2000, p. 31), “Normalmente chamado de e-mail, o correio eletrônico é o serviço disponibilizado na rede mundial de computadores, a Internet, para troca de mensagens, a princípio constituídas apenas de texto (cadeia de caracteres).” Então o e-mail é um serviço que permite a troca de mensagens entre os usuários do serviço, além de diversas outras funcionalidades.

O e-mail é um serviço utilizado para o envio e recebimento de mensagens eletrônicas. O serviço de e-mail é composto por diversos protocolos, entre eles: o SMTP (Single Mail Transfer Protocol), [...] como sendo o protocolo a ser utilizado para a transferência da mensagem de um usuário *A* até o servidor do usuário *B*, onde a mensagem será armazenada. Quando a mensagem já estiver armazenada no servidor de destino, o cliente dispõe de alguns protocolos para efetuar a leitura destes emails. Os mais difundidos são o POP (Post Office Protocol) [...] e o outro é o IMAP (Internet Message Access Control). (KNORST, 2000, p. 2, grifo do autor)

Para Levine (2013, p. 135) o e-mail é o serviço mais popular da internet, mesmo sendo um dos mais antigos e pouco chamativo. Mas ainda é a forma das empresas se comunicarem entre si e com seus clientes. Isso ilustra a importância do serviço de e-mail tanto para o uso pessoal quanto profissional atualmente, mesmo que é um serviço bastante antigo, continua funcionando e sendo utilizado por milhões de usuários.

Para realizar a troca de mensagens, serviço de e-mail faz uso de alguns protocolos, cada qual com um propósito específico. O protocolo SMTP utilizado para realizar o envio de mensagens de e-mail. O objetivo do SMTP é transferir mensagem com eficiência e segurança entre dois hosts independentemente do subsistema de transmissão (CARVALHO, 1996). Já Regis (2000, p. 27) define SMTP como, “um conjunto de componentes funcionais residentes nos sistemas transmissor e receptor cooperam entre si de forma a garantir o sucesso dessa comunicação.”

Segundo dados do Internet Engineering TaskForce, o protocolo SMTP foi padronizado pela primeira vez em 1982 através do RFC 821 e em 2008 foi atualizado pela padronização RFC 5321 que é a vigente.

O cliente SMTP se conecta a um servidor SMTP para entregar mensagens

eletrônicas. Esse cliente pode ser a própria máquina do usuário, ou uma máquina que também é servidor de SMTP de uma rede. [...] Para a transferência das mensagens, o protocolo utiliza uma conexão TCP, e a comunicação entre as máquinas é feita através de mensagens em ASCII. (CARNEIRO, 2000, p. 31)

Para realizar o recebimento de mensagens os serviços de e-mail implementam os protocolos POP (Post Office Protocol) e IMAP (Internet Message Access Protocol). O protocolo POP utilizado para acesso remoto a uma caixa de correio eletrônico. Permite que as mensagens contidas em uma caixa de e-mail, possam ser transferidas para um computador local. Assim o usuário, poderá ler as mensagens recebidas, apagá-las, respondê-las e/ou armazená-las (EBERHARDT, 2007, p. 16).

Segundo o Internet Engineering TaskForce (1996a, tradução nossa), o protocolo POP permite que uma estação de trabalho possa acessar dinamicamente a caixa de e-mail de um servidor de forma útil. Isto significa que o protocolo é utilizado para permitir que uma estação de trabalho recupere o que o servidor de correio está armazenando para ele.

Para Knorst (2000, p. 9), o protocolo IMAP além de possuir as mesmas características do protocolo POP, permite que o usuário mantenha todas as mensagens armazenadas no servidor, podendo ser acessadas a partir de qualquer computador que tenha um software que implemente o protocolo IMAP. O IMAP armazena todos os seus e-mails em um servidor, de maneira que é possível acessá-los a partir de qualquer programa de e-mail. (LEVINE, 2013, p. 139)

Permite que um cliente possa acessar e manipular mensagens de correio eletrônico sobre um servidor. IMAP4 permite a manipulação de pastas de mensagens remotas, chamadas "caixas de correio", de uma maneira que é funcionalmente equivalente a caixas de correio locais. [...] Inclui operações para criar, excluir e renomear caixas de correio; verificação de novas mensagens; remover permanentemente as mensagens; remoção de *flags* [...] (INTERNET ENGINEERING TASKFORCE, 1996b, p. 1).

2.2 COMPONENTES DE ENVIO E RECEBIMENTO DE E-MAIL

São componentes instalados nos sistemas operacionais e que aplicações terceiras podem fazer uso desses componentes. Os componentes apresentados realizam operações específicas utilizando os protocolos POP3, IMAP4 e SMTP. E são executados em ambientes .Net.

2.2.1 EAGetMail POP3 e IMAP4 Component

O EAGetMail é um componente de comunicação que suporta todas as operações dos protocolos POP3/IMAP4/MIME/Exchange/WebDav/SSL (ADMINSYSTEM SOFTWARE

LIMITED, 2016a, tradução nossa). No Quadro 1 estão listadas as principais características do componente.

Quadro 1 – Principais características EAGetMail

POP3/IMAP4	Comandos POP3 suportados: USER, PASS, AUTH NTLM, APOP, NOOP, RSET, TOP, LIST, STAT, UIDL, RETR E QUIT. Suporta todos os comandos do protocolo IMAP4. Suporta os comandos do Exchange do tipo WebDAV e Web Service – EWS. Obtém apenas o cabeçalho do e-mail, sem fazer o download do e-mail todo. Gerencia as pastas e move e-mails. Realiza pesquisa avançada nos e-mails. Autenticação utilizando LOGIN, CRAM-MD5, NTL. Suporte a servidores de proxy: Socks 4/ Socks5/ Http.
SSL/TLS	Suporte ao SSL v2, SSL v3, TLS v1. Algoritmo de chave RSA e DH. Implementa os comandos ESMTP: TLS e STARTTLS. Tudo conforme as especificações RFC 2487 e RFC2595.
Mail Parser	Conversão do Remetente, Participantes, Assunto, Data Recebimento, Corpo do e-mail e Anexos. Conversão de arquivos winmail.dat (arquivo TNEF). Conversão dos e-mails em arquivos compatíveis com o Microsoft Office Outlook (.MSG). Converte arquivos .EML em arquivos do Microsoft Office Outlook (.MSG) e vice e versa.

Fonte: Adaptação de Adminsystem Software Limited (2016a, tradução nossa).

O componente permite que desenvolvedores entreguem aplicações seguras e completas em VB6, ASP, C++, C#, VB.NET, JScript.NET, ASP.NET baseadas no framework .net ou COM mais facilmente (ADMINSYSTEM SOFTWARE LIMITED, 2016a, tradução nossa).

2.2.2 EASendMail SMTP Component

EASendMail é um componente baseado no protocolo SMTP com suporte ao Exchange WebService e WebDav, S/MIME, HTML, imagens *embedded* no corpo do e-mail e segurança através de SSL (ADMINSYSTEM SOFTWARE LIMITED, 2016b, tradução nossa).

EASendMail foi desenvolvida em C# com código gerenciado e permite a construção e envio de mensagens em VB.NET através de seu `Namespace` com classes para auxiliar no desenvolvimento e envio para qualquer servidor de e-mail arbitrário ou sem um DNS (Domain Name System) especificado (ADMINSYSTEM SOFTWARE LIMITED, 2016b, tradução nossa).

2.3 DESIGN RESPONSIVO

O conceito de design responsivo é aquele que responde a quaisquer dispositivos/resoluções e, devido a uma série de características bem específicas, é bem

apresentado em qualquer um deles (ZEMEL, 2013). Para Lopes (2013) no design moderno, as palavras chave são adaptação e flexibilidade, a mesma página executa de forma diferente dependendo do navegador, se adaptando as novas condições.

O design responsivo é uma nova forma de pensar a web e dentro de pouco tempo, será tão vital e importante aos desenvolvedores e à experiência do usuário quanto o próprio HTML ou CSS (ZEMEL, 2013). Lopes afirma que o design responsivo traz independência de plataforma e portabilidade da aplicação.

Segundo Zemel (2013) para implementar um design responsivo, três tecnologias principais estão envolvidas:

- a) o Layout Fluido: são sites que prezam pela não especificação de medidas fixas, permitindo que haja uma adaptação natural do que se apresenta na tela;
- b) as Imagens e Recursos flexíveis: são técnicas variadas em que recursos como imagens e vídeos do site sejam flexíveis para garantir a melhor experiência do usuário, independentemente do dispositivo usado;
- c) as Media Queries: são ações aonde é possível ocultar, apresentar e reposicionar elemento e interações conforme a resolução atual.

O design responsivo é uma solução técnica para programar um site de forma que os elementos que o compõem se adaptem automaticamente à largura de tela do dispositivo no qual ele está sendo visualizado (TEIXEIRA, 2011).

2.4 TRABALHOS CORRELATOS

Nas pesquisas realizadas foram encontrados produtos com a finalidade semelhante ao desenvolvido neste trabalho: aplicação para gerenciar mensagens de vários serviços de e-mails diferentes. Dentre as pesquisadas destacam-se o Microsoft Office Outlook, Mozilla Thunderbird, Google Mail e a monografia Ferramenta para o controle de e-mails e *anti-spam*. Esta seção tem por objetivo apresentar esses produtos e a monografia.

2.4.1 Ferramenta para controle de e-mails e anti-spam

O trabalho de Eberhardt (2007) foi a criação de uma ferramenta Web para gerenciar e-mails de forma mais simples aonde o principal foco foi o de fazer um controle *anti-spam*, para identificar e bloquear mensagens indesejadas.

O principal objetivo foi desenvolver uma ferramenta que permite gerenciar o uso do serviço de e-mail em ambiente corporativo coletando as principais informações das

mensagens como remetente e destinatário, assunto, data e hora entre outros, armazenando esses dados em uma base integrada com o gerenciador de caixa postal Mail Transfer Agent Postfix. Dessa maneira organizando todos esses dados para que seja possível realizar uma auditoria nas trocas de mensagens do serviço de e-mail (EBERHARDT, 2007).

O projeto utilizou a linguagem de programação Perl em conjunto com o banco de dados MySQL para armazenar as regras de armazenamento de e-mails e os principais dados extraídos das mensagens de e-mail e de todo o controle *anti-spam*. O servidor do serviço de e-mail do projeto utilizava o Postfix para realizar todo o gerenciamento de caixa postal.

O projeto desenvolvido tratava mais do gerenciamento das mensagens de e-mail recebidas pelos usuários do serviço, a fim de bloquear os *spams* no ambiente corporativo. Outro ponto marcante foi o fato de que o sistema gerava relatório com as informações obtidas das mensagens de e-mail trocadas. Permitindo a apresentação dessas informações em uma possível auditoria ou para controle gerencial das mensagens que chegam e saem da empresa.

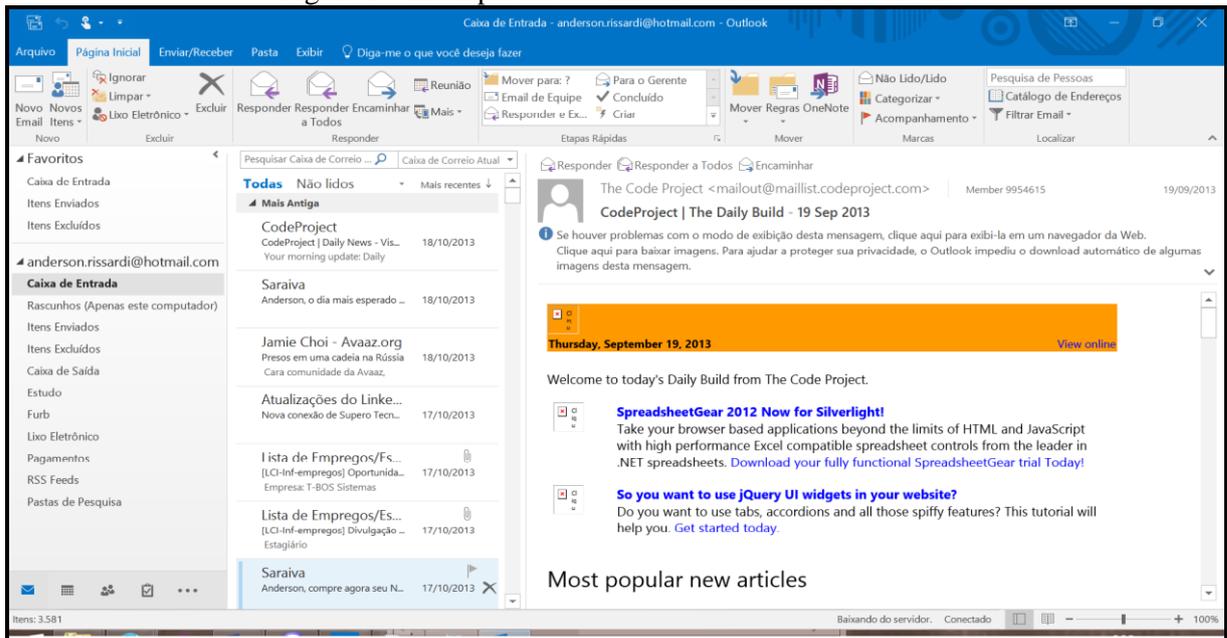
2.4.2 Microsoft Office Outlook

O Microsoft Office Outlook é a solução para ambiente *desktop* da Microsoft desde 1997 para gerenciar e-mails nos ambientes Windows e Mac OS (Microsoft Corporation, 2016). É possível cadastrar e-mails de todos os domínios, desde que, seja possível acessar o servidor através dos protocolos POP3 ou IMAP e do SMTP. Também é possível configurar o acesso a servidores através de proxy. É um dos produtos do pacote Office da Microsoft.

O Microsoft Outlook 2016 é a versão mais recente do Outlook. As versões anteriores incluem o Outlook 2013, o Outlook 2010 e o Outlook 2007. O Outlook 2016 é compatível com o Windows 10, o Windows 8.1 e o Windows 7 (MICROSOFT CORPORATION, 2016).

O Microsoft Outlook apresenta na sua tela inicial as contas de e-mail cadastradas como se fossem grupos no canto esquerdo da tela, algo parecido com uma estrutura de pastas, aonde é possível acessar itens que abrem outros itens, como uma estrutura com níveis. Abaixo de cada conta as pastas daquela conta de e-mail são criadas. Ao selecionar uma pasta o Microsoft Outlook exibe uma lista com os e-mails que estão dentro da pasta e exibe um resumo do e-mail. O usuário tem as opções de responder o e-mail, criar um novo e-mail, encaminhar o e-mail, excluir o e-mail, entre diversas outras. A Figura 1 exemplifica a tela inicial do Outlook.

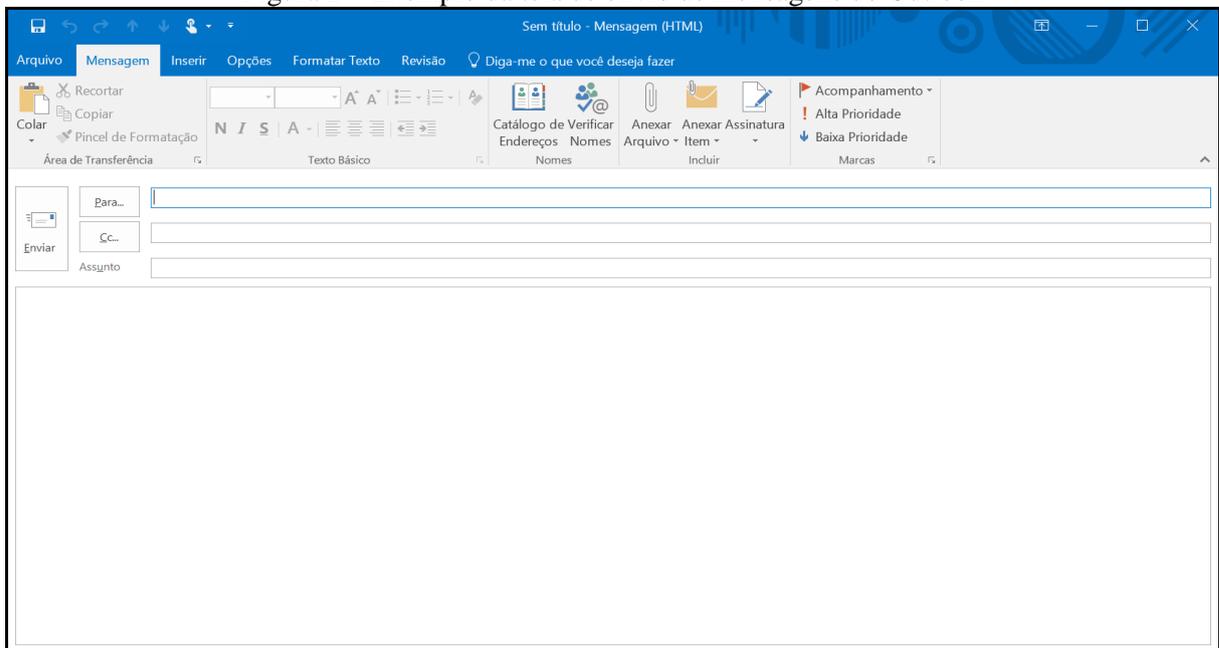
Figura 1 – Exemplo da tela inicial do Microsoft Outlook



Fonte: Do Autor.

Já a tela de envio de e-mails do Outlook é uma janela apresentada ao usuário que permite que o usuário informe o destinatário do e-mail, escreva as mensagens com formatações avançadas entre outras diversas funcionalidades. A Figura 2 exemplifica a tela de envio de e-mails.

Figura 2 – Exemplo da tela de envio de mensagens do Outlook



Fonte: Do Autor.

2.4.3 Mozilla Thunderbird

O Mozilla Thunderbird é um programa disponibilizado pela fundação Mozilla de forma gratuita para gerenciar contas de e-mails de diferentes domínios.

Desenvolvido desde 1998 para ambiente desktop. Ele funciona nos sistemas operacionais Windows, Mac OS e Linux. Ele é mantido pela comunidade de desenvolvedores da fundação Mozilla e parte de seu código fonte é aberto (MOZILLA ORGANIZATION, 2016).

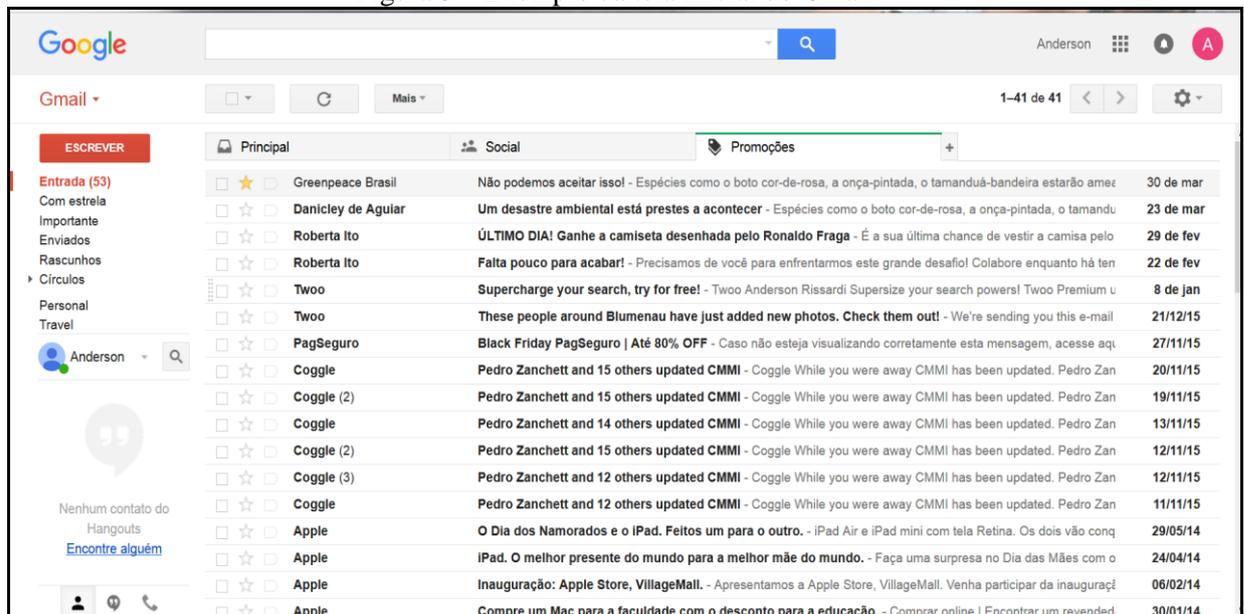
É bastante parecido com o Microsoft Outlook, tanto as funcionalidades quanto as telas do sistema. Ambos os produtos foram evoluindo em conjunto com o passar dos anos até chegar nas versões mais recentes.

2.4.4 Google Mail (Gmail)

O Google Mail também conhecido como Gmail é um serviço gratuito de e-mails. É um dos sistemas *webmail* mais utilizados e estáveis atualmente. Segundo Garretti (2015) foi criado e é mantido pela Google desde 2004 e se destacou logo no início por oferecer muito mais espaço e ser muito mais rápido que os concorrentes.

Ele funciona nas plataformas desktop e para dispositivos móveis, não é responsivo, mas existe um aplicativo para dispositivos móveis. A Figura 3 apresenta a tela inicial do Gmail. A estrutura é bastante parecida com a do Outlook, porém além de apresentar os e-mails que estão na pasta selecionada ele também faz uma divisão aonde seleciona os e-mails pelo tipo. Dividindo em 3 abas, principal, social e promoções.

Figura 3 – Exemplo da tela inicial do Gmail

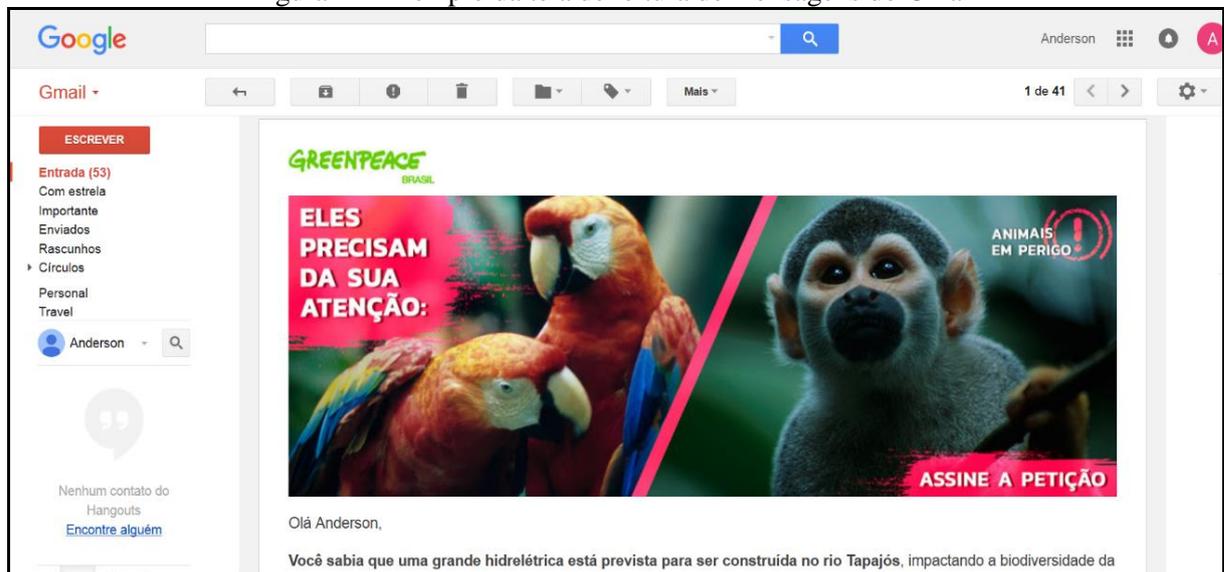


Fonte: Do Autor.

Ao clicar sobre o e-mail é realizada a visualização do e-mail. A tela de visualização permite realizar diversas ações com o e-mail como por exemplo enviar para a lixeira, marcar

como não lido, marcar o e-mail como favorito, mover para outra pasta entre outras funcionalidades. A Figura 4 ilustra a tela de visualização de e-mail.

Figura 4 – Exemplo da tela de leitura de mensagens do Gmail



Fonte: Do Autor

No Gmail é possível adicionar até 5 contas de outros serviços de e-mail, de maneira gratuita, desde que seja possível acessar esses serviços através do protocolo POP3 (GOOGLE, 2016b). É possível importar mensagens e contatos de outros serviços de e-mail e ativar o monitoramento de conta, sempre que a conta nova receber um e-mail esse e-mail é encaminhado para a conta Gmail do usuário.

3 DESENVOLVIMENTO DA APLICAÇÃO

Neste capítulo serão descritas as especificações técnicas da aplicação desenvolvida e maiores detalhes tais como a descrição, a apresentação dos requisitos funcionais e não funcionais, diagrama de casos de uso, definições das funcionalidades desenvolvidas. Também serão apresentadas bibliotecas utilizadas no desenvolvimento da aplicação.

3.1 INFORMAÇÕES

Foi desenvolvido uma aplicação Web para leitura e envio de mensagens de e-mail através de diferentes serviços de e-mail, que permite ser utilizada através de computadores e dispositivos móveis. Para utilizar a aplicação é necessário criar uma conta e utilizar essa conta para acessar a aplicação que possui apenas o papel usuário.

Ao acessar o sistema o usuário já pode navegar pelo sistema, porém nenhuma ação estará disponível até que ele cadastre ao menos uma conta de e-mail. Até o usuário realizar esse cadastro somente será possível alterar as informações do perfil e cadastrar contatos.

Após cadastrar ao menos uma conta de e-mail o sistema buscará os e-mails recebidos no último mês. Não há limite para a quantidade de contas a serem cadastradas pelo usuário. Dessa maneira o usuário pode adicionar todas as contas que ele utiliza no dia-a-dia e verificar todas as mensagens através da aplicação.

O usuário pode ler todas as mensagens de e-mail recebidas e também é possível escrever e enviar novas mensagens de e-mail. Caso o usuário queira ver mensagens mais antigas a aplicação permite a realização de uma pesquisa avançada e simultânea em todas as contas de e-mail, fazendo filtros por data de recebimento, remetente e assunto. Dessa maneira o usuário não precisa saber em qual conta está a mensagem que ele deseja ler novamente, basta realizar a pesquisa que todas as contas serão vasculhadas.

A aplicação não armazena as informações das mensagens recebidas ou enviadas pelo usuário, nem no servidor e nem na base. Os dados são sempre enviados/recebidos pela aplicação diretamente para /do serviço de e-mail da conta que o usuário está utilizando. Dessa maneira, garante segurança e privacidade ao usuário.

É possível acessar a aplicação através de dispositivos móveis e também computadores *desktop* através de navegadores de internet. É possível realizar acesso simultaneamente, através de diversos dispositivos.

3.2 ESPECIFICAÇÕES

Nesta seção serão apresentados os requisitos da aplicação desenvolvida, funcionais e não funcionais. Também serão apresentados alguns diagramas desenvolvidos com a ferramenta Enterprise Architect como diagrama de casos de uso e o diagrama de implantação da aplicação.

3.2.1 Requisitos

Nesta seção serão apresentados os requisitos funcionais (RF) e não funcionais (RNF) da aplicação desenvolvida. O Quadro 2 apresenta os requisitos funcionais da aplicação com a definição de qual caso de uso atende esse requisito, apresentando a rastreabilidade entre requisitos e casos de uso. Os casos de uso serão apresentados no item 3.2.2.

Quadro 2 – Requisitos funcionais

Requisitos Funcionais	Caso de Uso
RF01 - O sistema deve permitir o cadastro de usuários.	UC02
RF02 - O sistema deve permitir o cadastro de contas de e-mail.	UC07
RF03 – O sistema deve permitir manter as credenciais do usuário ao realizar o login e auto preencher para o usuário.	UC08
RF04 - O sistema deve permitir o enviar e responder mensagens de e-mails.	UC09
RF05 - O sistema deve listar e-mails de diversas contas de diversos serviços de e-mail.	UC10
RF06 - O sistema deve permitir que o usuário cadastre uma lista de contatos.	UC11
RF07 – O sistema deve permitir parar o monitoramento de determinada conta de e-mail	UC03
RF08 - O sistema deve permitir ao usuário realizar uma pesquisa avançada nas mensagens das contas de e-mail cadastradas.	UC13
RF09 – O sistema deve permitir o cadastro de um perfil para o usuário.	UC04
RF10 – O sistema deve permitir realizar <i>login</i> no sistema	UC06
RF11 – O sistema deve permitir que o usuário insira anexos ao enviar e-mails.	UC05
RF12 – O sistema deve permitir visualizar as mensagens de e-mails em que o corpo do e-mail for do formato HTML.	UC12

Fonte: Do autor.

O Quadro 3 apresenta os requisitos não funcionais da aplicação.

Quadro 3 – Requisitos não funcionais

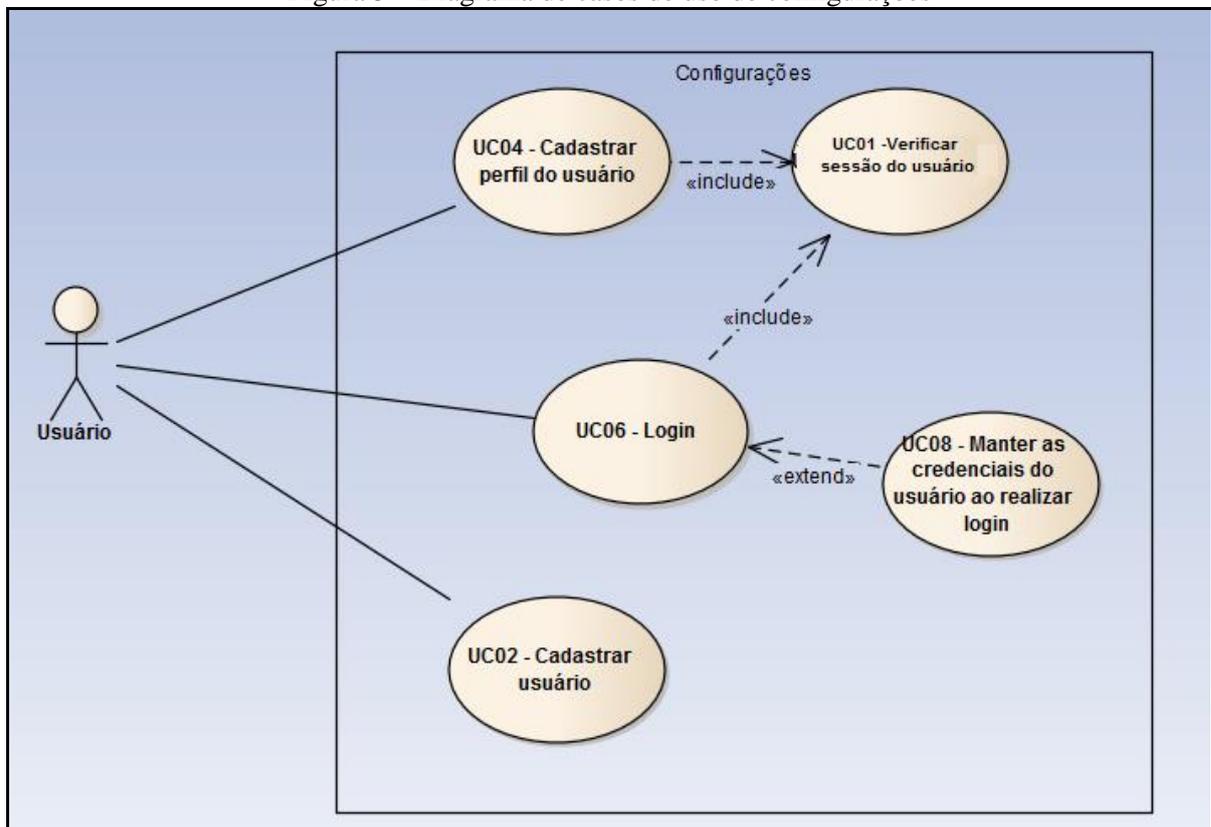
Requisitos Não Funcionais
RNF01: O sistema deverá utilizar o banco de dados MS SQLServer.
RNF02: O sistema deverá ser acessado através de navegadores de internet Chrome, Internet Explorer a partir da versão 9, Edge, Mozilla Firefox, Safari e Opera.
RNF3: O sistema deverá ser desenvolvido utilizando o modelo MVC e as interfaces HTML 5, CSS 3.0 e o uso da biblioteca de estilos Bootstrap.
RNF4: Utilização da biblioteca Entity framework como ORM (Mapeamento objeto-relacional) da aplicação.
RNF5: Acesso à internet através dos protocolos SMTP, IMAP4 e sem a utilização de <i>proxys</i> .
RNF6: Anexos podem ter no máximo 4Mb.
RNF7: Sistema deverá ser responsivo. Deverá ser possível acessar o sistema através de dispositivos móveis.
RNF8: O servidor do serviço de e-mail precisa disponibilizar acesso através do protocolo IMAP 4.

Fonte: Do autor.

3.2.2 Diagramas

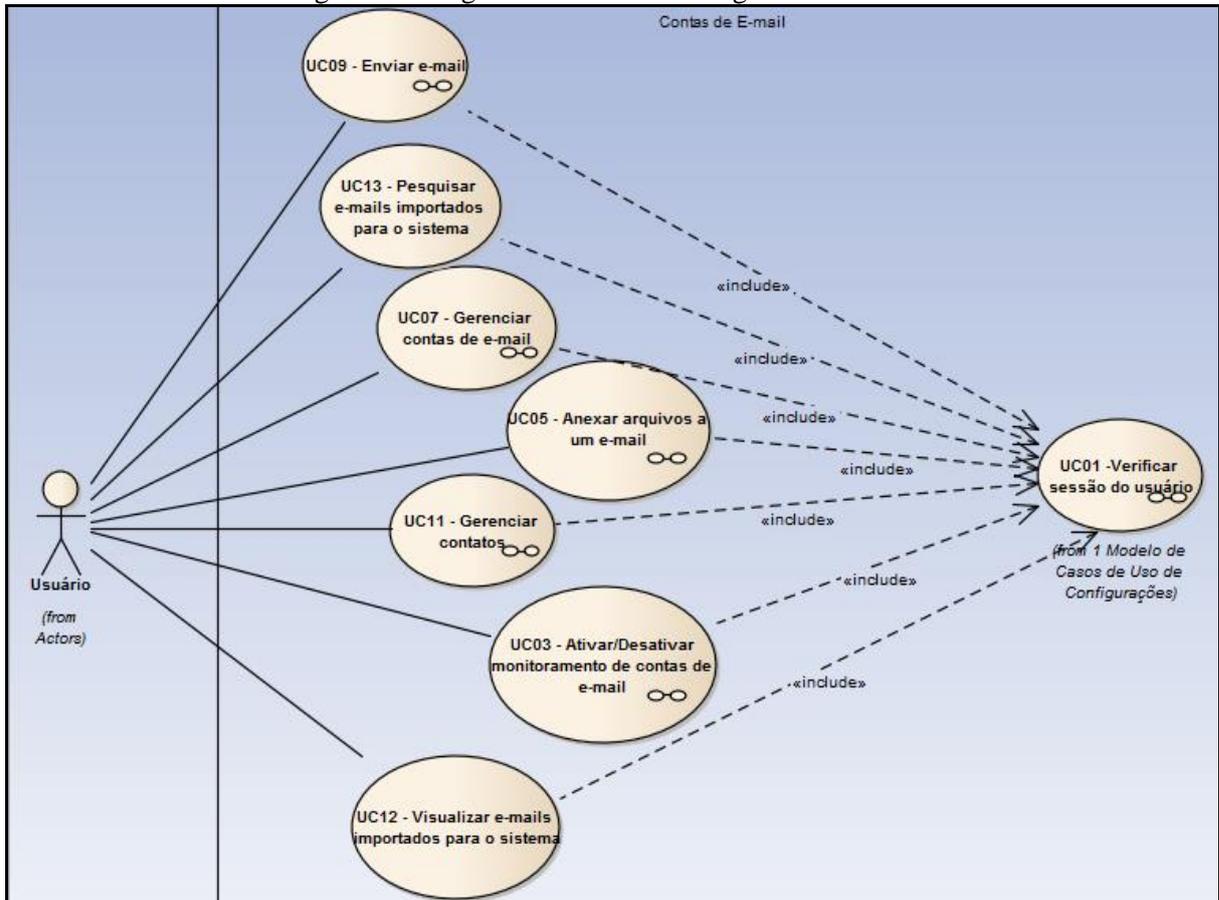
Nessa seção serão apresentados os diagramas de casos de uso e de implantação da aplicação desenvolvida. A descrição dos casos de uso está sendo apresentada no Apêndice A. A Figura 5 apresenta o diagrama de casos de uso de configurações e a Figura 6 o diagrama de casos de uso de itens gerais do sistema.

Figura 5 – Diagrama de casos de uso de configurações



Fonte: Do autor.

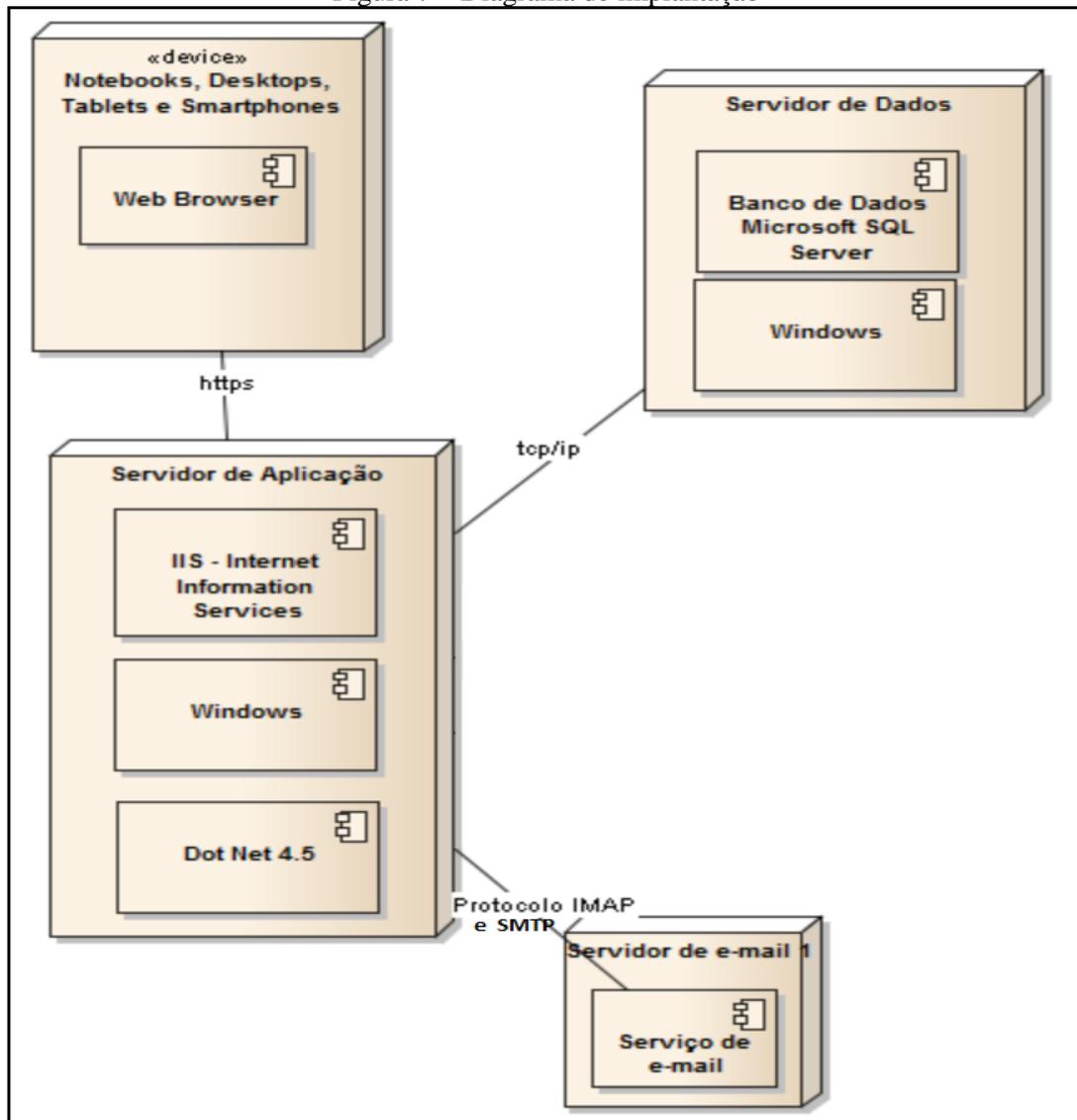
Figura 6 – Diagrama de casos de uso gerais do sistema



Fonte: Do autor.

A Figura 7 apresenta o diagrama de implantação da aplicação, ilustrando a comunicação que a aplicação faz com os clientes, banco de dados e serviços de e-mail externos.

Figura 7 – Diagrama de implantação



Fonte: Do autor.

No diagrama de implantação no servidor em que a aplicação estiver hospedada deve utilizar o sistema operacional Windows com o .Net Framework na versão 4.5 instalado, além de utilizar o Internet Information Services da Microsoft para publicar a aplicação. O servidor da aplicação ainda deve poder acessar através dos protocolos IMAP4 e SMTP os serviços de e-mail dos respectivos servidores de e-mail que forem solicitados através da aplicação, dessa maneira, não deve ser feito nenhum bloqueio que envolva esses protocolos.

O servidor de dados da aplicação deve utilizar o sistema operacional Windows e executar o banco de dados Microsoft SQL Server de maneira que permita a conexão entre a aplicação que está em outro servidor e o banco de dados.

Como a aplicação estará publicada na internet os usuários da aplicação utilizam os navegadores de internet, como por exemplo, Mozilla Firefox para acessar a aplicação, portanto o dispositivo que está sendo usado no lado cliente pode ser um *desktop*, notebook ou

qualquer dispositivo móvel que possua um navegador de internet compatível com os navegadores apresentados no requisito não funcional 2.

3.3 IMPLEMENTAÇÃO

Nesta seção serão apresentadas informações sobre o desenvolvimento da aplicação, como ferramentas e técnicas utilizadas. Além do modelo de implementação utilizado e as características oriundas desse modelo.

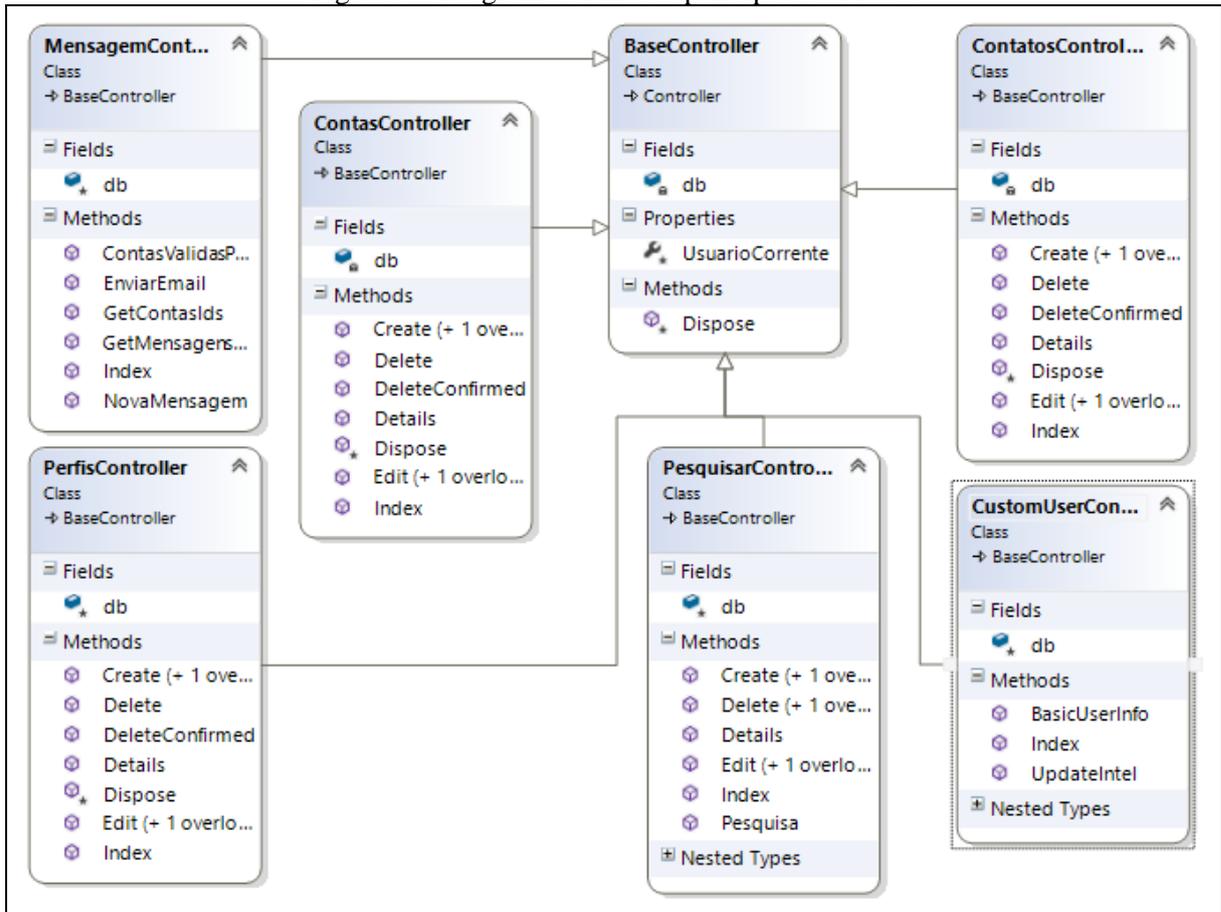
3.3.1 Modelo de desenvolvimento

A aplicação desenvolvida é Web e segue o padrão MVC de desenvolvimento. O desenvolvimento se divide em 3 camadas. A parte cliente da aplicação, que é executada pelo navegador do usuário, é a interface gráfica que apresenta e recebe dados do usuário e representa a *view* do padrão de desenvolvimento. E a parte servidor, que é onde a aplicação fica hospedada, aplica as regras do negócio e faz a persistência dos dados da aplicação. Representando os *controllers* e o *model* do padrão de desenvolvimento respectivamente.

A camada modelo da aplicação ficou bastante parecida com o DER (Diagrama Entidade Relacionamento) da aplicação e as classes foram utilizadas para servir de modelo para persistência de dados.

A camada de controladores da aplicação foi desenvolvida para garantir a segurança da informação e também atender as regras de negócio da aplicação. A Figura 8 ilustra em um diagrama UML os principais controladores da aplicação.

Figura 8 – Diagrama UML dos principais controles



Fonte: Do autor.

Os principais controladores da aplicação herdam do controlador chamado `BaseController`. Esse controlador garante a segurança da aplicação, como por exemplo, aplicar os filtros nas requisições que chegam a aplicação para garantir que é um usuário conectado no sistema que está fazendo essa chamada e que o usuário tem acesso a esse recurso. Ele também disponibiliza para os outros controladores a propriedade `UsuarioCorrente` com as informações de qual usuário do sistema está realizando a ação. O Quadro 4 apresenta como é carregada essa propriedade.

Quadro 4 – Identificando o usuário realizando a ação.

```

protected ApplicationUser UsuarioCorrente
{
    get
    {
        return db.Users.FirstOrDefault(nn => nn.UserName ==
User.Identity.Name);
    }
}
  
```

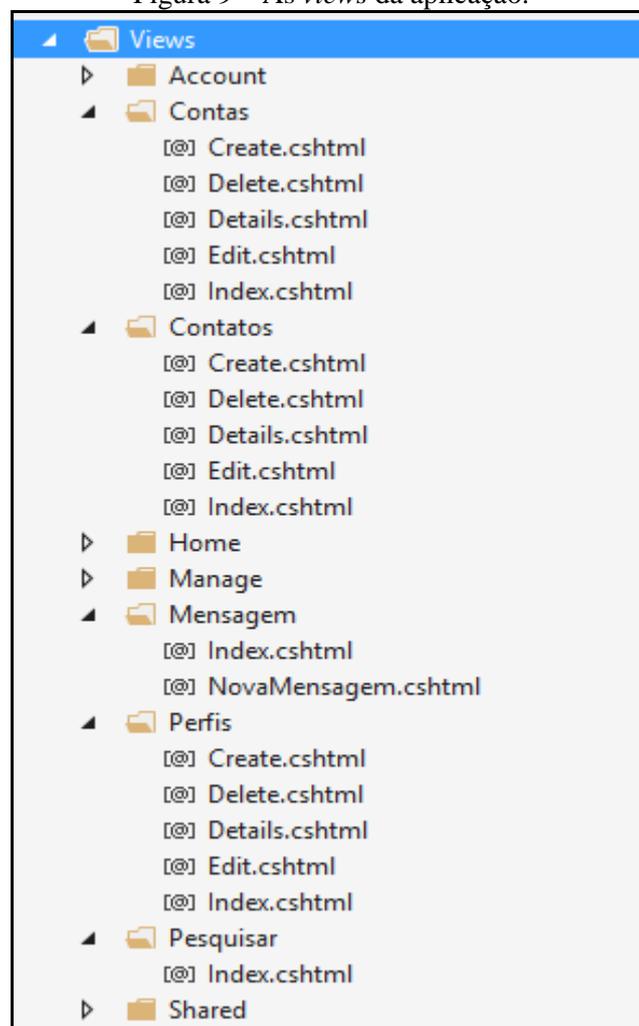
Fonte: Do autor.

No Quadro 4 o .Net Framework disponibiliza através da biblioteca `Identity` disponibiliza a propriedade `User.Identity.Name` após identificar na requisição que chegou, os cookies atrelados ao usuário que está fazendo a requisição. A biblioteca identifica e carrega

um objeto com as informações de identidade do usuário. Essas informações são comparadas com o cadastro de usuários do sistema e retorna todas as informações do usuário que está realizando essa ação no sistema.

A parte de visualização de recursos da aplicação é composta por interfaces disponíveis aos usuários, desde consulta de contas, por exemplo, como telas para cadastrar, editar informações de contas, contatos entre outras funcionalidades em que a interação com o usuário é feita através da interface. A Figura 9 apresenta a estrutura de *views* criadas na aplicação.

Figura 9 – As *views* da aplicação.



Fonte: Do autor.

Nas *views* é possível identificar a outra parte da aplicação, o lado cliente. Isso por que o lado cliente recebe os arquivos `cshtml` processados e renderiza o HTML gerado no navegador do cliente.

3.3.2 Técnicas e ferramentas utilizadas

O lado servidor da aplicação foi desenvolvido na linguagem de programação C# na versão 6.0, no ambiente .NET Framework 4.5. O C# é uma linguagem de programação criada para o desenvolvimento de uma variedade de aplicações que executam sobre o .NET Framework (MICROSOFT CORPORATION, 2016).

O C# é uma linguagem poderosa e simples, voltada para os desenvolvedores, herdando grande parte dos melhores recursos do C++ e Visual Basic e pouco das inconsistências e anacronismos, resultando uma linguagem limpa e lógica (SHARP, 2011, p.21).

O lado cliente da aplicação foi desenvolvida utilizando as linguagens de programação HTML e JavaScript. As telas da aplicação foram codificadas tendo como objetivo deixar responsiva as interfaces para que o usuário possa utilizar a aplicação a partir de qualquer dispositivo.

Para desenvolver a aplicação foi utilizada a ferramenta de desenvolvimento Visual Studio Community. O Visual Studio Community é uma IDE bastante completa, extensível e gratuita para a criação de aplicações modernas (MICROSOFT CORPORATION, 2016). A ferramenta disponibiliza um ambiente de desenvolvimento compatível com várias linguagens de programação, incluindo suporte às linguagens utilizadas no lado cliente e no lado servidor da aplicação.

3.3.2.1 Detalhamento desenvolvimento lado cliente

Para fazer a implementação conforme o conceito de interface responsiva conforme o item 2.4, que fundamenta o conceito, foram utilizadas as bibliotecas Bootstrap e jQuery, nas versões 3.3 e 2.1 respectivamente. O jQuery é uma biblioteca JavaScript rápida, pequena e cheia de funcionalidades, que funciona através de vários navegadores (THE JQUERY FOUNDATION, 2016, tradução nossa).

O jQuery foi escolhido como principal biblioteca de desenvolvimento JavaScript da aplicação por causa da compatibilidade que ele tem, de funcionar em todos os navegadores modernos. O Quadro 5 apresenta a função que monta a tela de visualização e leitura de mensagens de e-mail.

Quadro 5 – Função Javascript que monta a tela de visualização de mensagens

```

var DefinicaoLinguagemGridMensagens = {
  lengthMenu: "Exibindo _MENU_ mensagens por página.",
  zeroRecords: "Nenhuma mensagem encontrada.",
  info: "Página PAGE de PAGES ",
  infoEmpty: "Nenhuma mensagem para exibir.",
  infoFiltered: "Total de registros: _MAX_",
  search: "Localizar",
  processing: "Pesquisando...",
  decimal: ",",
  thousands: ".",
  oPagate: {
    sNext: "Próximo",
    sPrevious: "Anterior",
    sFirst: "Primeiro",
    sLast: "Último"
  },
  oAria: {
    sSortAscending: ": Ordenar colunas de forma ascendente",
    sSortDescending: ": Ordenar colunas de forma descendente"
  }
};

$(document).ready(function () {
  $.getJSON('../Mensagem/GetContasIds', {}, function (data) {
    var listaTabs = $("#listaTabs");
    var contentTabs = $("#tabsContent");

    if (listaTabs && contentTabs) {
      // debugger;
      for (var i = 0; i < data.length; i++) {
        var currentConta = data[i];
        var classActiveControl = i == 0;
        var idPanel = "panel_" + currentConta.Id;

        var item = classActiveControl ? "<li class='active'>"
: "<li>";
        item += ("<a href='#" + idPanel + "' data-
toggle='tab' aria-expanded='true'>" + currentConta.Email + "<i id='" +
currentConta.Id + "' style='float: left;position: relative;cursor:
pointer;' onclick='atualizarGrid(this);'><span class='fa fa-
refresh'></span></i>" + "</a>");
        item += "</li>";

        var finalClassPanel = classActiveControl ? " in
active'" : "";
        var panelItem = "<div id='" + idPanel + "'
class='tab-pane fade " + finalClassPanel + ">";

        var tableID = 'table_' + currentConta.Id;

        panelItem += "<table id='" + tableID + "'
class='table table-hover' width='99%' style='width: 99% !important;'
cellspacing='0'><thead>";
        panelItem += "<tr role='row'>";
        panelItem += "<th class='sorting' tabindex='0'
rowspan='1' colspan='1'>";
        panelItem += "De";
        panelItem += "</th>";
        panelItem += "<th class='sorting' tabindex='0'
rowspan='1' colspan='1'>";

```

```

        panelItem += "Assunto";
        panelItem += "</th>";
        panelItem += "<th class=' ' tabindex='0' rowspan='1'
colspan='1'>";
        panelItem += "Recebido em";
        panelItem += "</th>";
        panelItem += "</tr>";
        panelItem += "</thead></table>";
        panelItem += "</div>";

        contentTabs.append(panelItem);
        listaTabs.append(item);
    }
}

$('a[data-toggle="tab"]').on('shown.bs.tab', function (e) {
    $.fn.dataTable.tables({ visible: true, api: true
}).columns.adjust();
});

for (var i = 0; i < data.length; i++) {
    var contaID = data[i].Id;
    var tableID = 'table_' + contaID;
    $('#' + tableID).DataTable({
        ajax:
        "../Mensagem/GetMensagensFromContaById?IdConta=" + contaID,
        language: DefinicaoLinguagemGridMensagens,
        //scrollY: 275,
        select: true,
        info: false,
        pagingType: "simple_numbers",
        scrollCollapse: true,
    });

    $('#' + tableID + ' tbody').on('click', 'tr', function ()
{ clickGridEmail(this); });
}
});
});

```

Fonte: Do autor.

A rotina utiliza o jQuery para disparar no evento 'ready' da página, quando o navegador cliente terminou de receber os dados e montar a tela da aplicação, nesse momento, utilizando jQuery, é executada uma requisição que busca todas as contas de e-mail do usuário e cria na página uma aba para cada conta de e-mail cadastrada pelo usuário, apresentando as mensagens de e-mail.

Após montar as tabelas, ainda utilizando o jQuery, é feita uma nova chamada para o servidor, buscando as mensagens de e-mail de cada conta, após a resposta, o jQuery é utilizado para preencher os dados das tabelas. Isso passa a impressão de que a tela está sendo carregada, alguns serviços de e-mail respondem mais rapidamente que outros, assim, ao carregar essas mensagens de maneira assíncrona é possível que o usuário comece a ler as

mensagens do serviço que responder primeiro, sem ter que esperar todas as mensagens de todas as contas carregarem.

Ainda buscando manter a interface responsiva, para que se adapte da melhor maneira a todos os dispositivos, foram utilizadas as bibliotecas de estilos Bootstrap e a Admin-Lte. Todas as interfaces do sistema fazem uso dessas bibliotecas que disponibilizam classes de estilo que se adaptam conforme a resolução da tela do usuário. O Quadro 6 apresenta um trecho de código da estrutura da tela de edição das informações, ela ajusta a disposição dos campos e botões da tela conforme a resolução da tela.

Quadro 6 – Estrutura tela cadastro de contas

```

<div class="box-footer">
  <div class="col-sm-2 col-xs-12">
    @Html.ActionLink("Voltar", "Index", new { },
new { @class = "btn btn-info", @style = "width: 100%;" })
  </div>
  <div class="col-sm-2 col-xs-12 pull-right">
    @Html.ActionLink("Excluir", "Delete", new {
id = Model.ContaID }, new { @class="btn btn-danger", @style = "width:
100%;" })
  </div>
  <div class="col-sm-2 col-xs-12 pull-right">
    <input type="submit" value="Salvar"
class="btn btn-primary" style="width: 100%;" />
  </div>
</div>

```

Fonte: Do autor.

No trecho de código HTML demonstrado no quadro 5 faz-se uso das classes de estilo do Bootstrap `col-sm-2`, `col-xs-12`, `pull-right` e `btn` para definir o tamanho e comportamento dos botões que ficam no rodapé da interface que o usuário visualiza. Em dispositivo com a resolução extremamente pequena cada botão ocupa uma linha, a partir da resolução pequena cada botão ocupa 2/12 avos da resolução da tela. O Bootstrap assume que o dispositivo tem a resolução extremamente pequena quando a resolução tem até 576px, pequena a partir de 576px até 768px, média até 992px, grande até 1200px e extremamente grande a partir disso (BOOTSTRAP, 2016, tradução nossa). A Figura 10 e a Figura 11 apresentam a disposição dos botões nas resoluções pequena e extremamente pequena respectivamente.

Figura 10 – Disposição de botões na resolução extremamente pequena



Fonte: Do autor.

Figura 11 – Disposição de botões a partir da resolução pequena.



Fonte: Do autor.

3.3.2.2 Detalhamento de desenvolvimento do lado servidor

As rotinas que são executadas na parte servidor da aplicação fazem a persistência dos dados na base utilizando a biblioteca Entity Framework e também se encarregam das rotinas de conexão e utilização dos serviços de e-mail. Para realizar a conexão da aplicação com os serviços de e-mail cadastrados pelos usuários são utilizados os componentes EAGetMail e EASendMail que estão fundamentadas nos itens 2.2.1 e 2.2.2 respectivamente. Para exemplificar a utilização dos componentes são apresentados os controladores `MensagemController` e `PesquisarController`.

Para realizar a conexão da aplicação com o serviço de e-mail da conta do usuário através do protocolo IMAP o componente necessita de 3 informações: endereço, usuário e senha. Todas as conexões são realizadas utilizando o padrão global de segurança de conexão SSL, se possível utiliza o mais novo, TLS 1.2. O Quadro 7 apresenta a rotina utilizada no controlador `PesquisarController` para realizar a conexão do serviço de e-mail com a aplicação.

Quadro 7 – Conexão com os serviços de e-mail

```

var servicoEmail = new MailServer(conta.ServidorIMAP, conta.Email,
conta.Senha, ServerProtocol.Imap4);
servicoEmail.SSLConnection = true;//Garante que será uma conexão
segura
servicoEmail.Port = conta.PortaIMAP;//Define a porta

var clienteConexao = new MailClient("TryIt");//Cria o "cliente" que
enviará a mensagem para o servidor
try
{
    clienteConexao.Connect(servicoEmail);//Realiza a conexão entre
cliente e serviço
}
catch
{
    continue;
}

```

Fonte: Do autor.

A rotina cria uma instância de `MailServer` que representa o serviço de e-mail informando o usuário, senha, endereço e o protocolo de comunicação. Também é definido que o serviço utiliza o padrão de criptografia SSL e define a porta que o serviço utiliza. A rotina também cria uma instância de `MailClient` e tenta realizar a conexão entre o serviço e o cliente. Após ter realizado a conexão é possível buscar os e-mails da conta de e-mail utilizada para realizar a conexão.

O controlador `PesquisarController` é o encarregado por realizar a pesquisa avançada em todas as contas. O componente `EAGetMail` é utilizado para realizar pesquisa nas mensagens de e-mail do serviço. O componente permite filtrar por propriedades da mensagem como por exemplo remetente, destinatário, corpo do e-mail, assunto e data. O componente primeiro faz uma pesquisa buscando apenas algumas propriedades da mensagem como por exemplo categorias, *flags*, se já foi lida entre outras, sem realizar o download da mensagem, deixando a pesquisa mais rápida e permitindo que buscar as informações completas da mensagem seja realizada apenas quando explicita no código. O Quadro 8 apresenta a rotina de pesquisa de mensagens utilizando o componente.

Quadro 8 – Filtros da pesquisa de mensagens

```

if (De.IsNotNullOrEmpty())
    clienteConexao.GetMailInfosParam.SenderContains = De;

if (Assunto.IsNotNullOrEmpty())
    clienteConexao.GetMailInfosParam.SubjectContains = Assunto;

if (Corpo.IsNotNullOrEmpty())
    clienteConexao.GetMailInfosParam.BodyContains = Corpo;

clienteConexao.GetMailInfosParam.DateRange.SINCE = dt;
clienteConexao.GetMailInfosParam.DateRange.BEFORE =
DateTime.Now.AddDays(1);
clienteConexao.GetMailInfosParam.GetMailInfosOptions |=
GetMailInfosOptionType.OrderByDateTime;
clienteConexao.GetMailInfosParam.GetMailInfosOptions |=
GetMailInfosOptionType.DateRange;

var mails = clienteConexao.GetMailInfos();

foreach (MailInfo mail in mails)
{
    //Realiza o acesso a informações do e-mail
    var item = clienteConexao.GetMail(mail);

    mailsToReturn.Add(new ItemMail()
    {
        De = item.From.Address,
        Assunto = item.Subject,
        HTMLBody = item.HtmlBody,
        Data = item.ReceivedDate.ToString("dd/MM/yyyy")
    });
}

```

Fonte: Do autor

Os filtros para realizar pesquisas no serviço de e-mail são adicionados na propriedade `GetMailInfosParam` do cliente que realizou a conexão. Aonde são aplicados filtros para que a propriedade contenha determinada palavra, propriedades como remetente, assunto e corpo do e-mail. Também é realizado um filtro por data de recebimento, se nenhuma data de início for especificada a aplicação toma como padrão a data 01/01/1900. O método `GetMailsInfo` do cliente realiza a pesquisa no serviço retornando apenas as mensagens que atendem ao filtro e o método `GetMail` busca as informações completas do e-mail.

O controlador `MensagemController` é responsável por diversas ações realizadas sobre as mensagens de e-mail, inclusive enviar e-mail. O componente `EASendMail` é o componente utilizado para fazer esse envio de mensagem de e-mail. Para realizar o envio a aplicação realiza uma conexão com o serviço de e-mail, monta a mensagem com seu corpo em HTML e envia através do protocolo SMTP. O Quadro 9 apresenta a rotina de envio de mensagem de e-mail.

Figura 12 – Tela inicial da aplicação



Fonte: Do autor.

Caso o usuário ainda não esteja cadastrado no sistema ele deve clicar no link *Registrar*, que redirecionará o usuário para a tela cadastro de usuários que é apresentada na Figura 13.

Figura 13 – Tela de cadastro de usuário.

Fonte: Do autor.

Após preencher todos os campos com os dados necessários e clicar no botão *Registrar* o usuário é redirecionado para a tela de *login* da aplicação. Para realizar a conexão

com o sistema o usuário deve informar o apelido e senha, a aplicação também permite a opção de continuar conectado neste aplicativo, dessa maneira as credenciais de acesso ficam armazenadas. A Figura 14 apresenta a tela de *login* da aplicação.

Figura 14 – Tela de *login* da aplicação.

A imagem mostra a interface de login de uma aplicação web. No topo, há uma barra azul com o nome 'Jaiminho Aglomera' à esquerda e os botões 'Entrar' e 'Registrar' à direita. O formulário centralizado contém o seguinte:

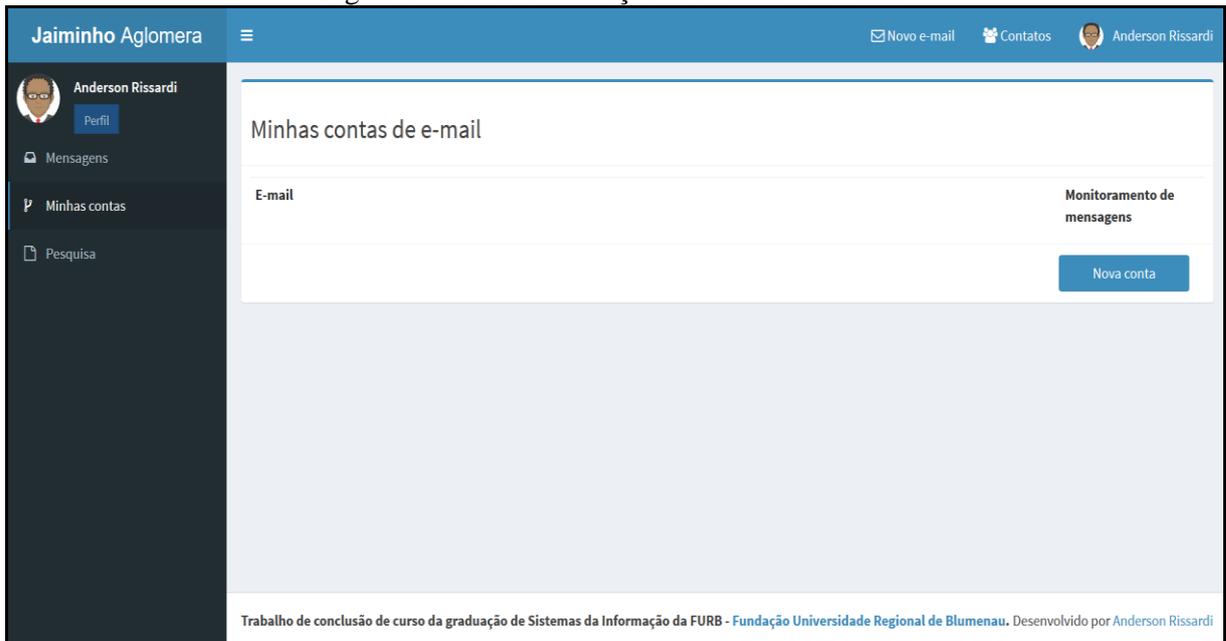
- Um título: 'Realize a conexão para acessar suas mensagens'.
- Um campo de texto para o apelido com o valor 'asdf' e um ícone de usuário.
- Um campo de senha com pontos e um ícone de cadeado.
- Uma caixa de seleção desativada com o texto 'Continuar conectado neste dispositivo?'.
- Um botão azul 'Entrar'.
- Um botão vermelho 'Novo por aqui? Registre-se'.

Na base da tela, há uma barra cinza com o texto: 'Trabalho de conclusão de curso da graduação de Sistemas da Informação da FURB - Fundação Universidade Regional de Blumenau.' e 'Desenvolvido por Anderson Rissardi'.

Fonte: Do autor.

Após realizar a conexão o usuário é redirecionado para a tela de exibição das contas de e-mail. Como o usuário ainda não possui nenhuma conta cadastrada apenas o botão *Nova conta* é apresentado. Conforme a Figura 15 apresenta, outras funcionalidades ficam disponíveis para o usuário, como editar as informações do perfil ou acessar o cadastro de contatos, e outras funcionalidades que são apresentadas, porém não estão disponíveis até o usuário cadastrar uma conta de e-mail.

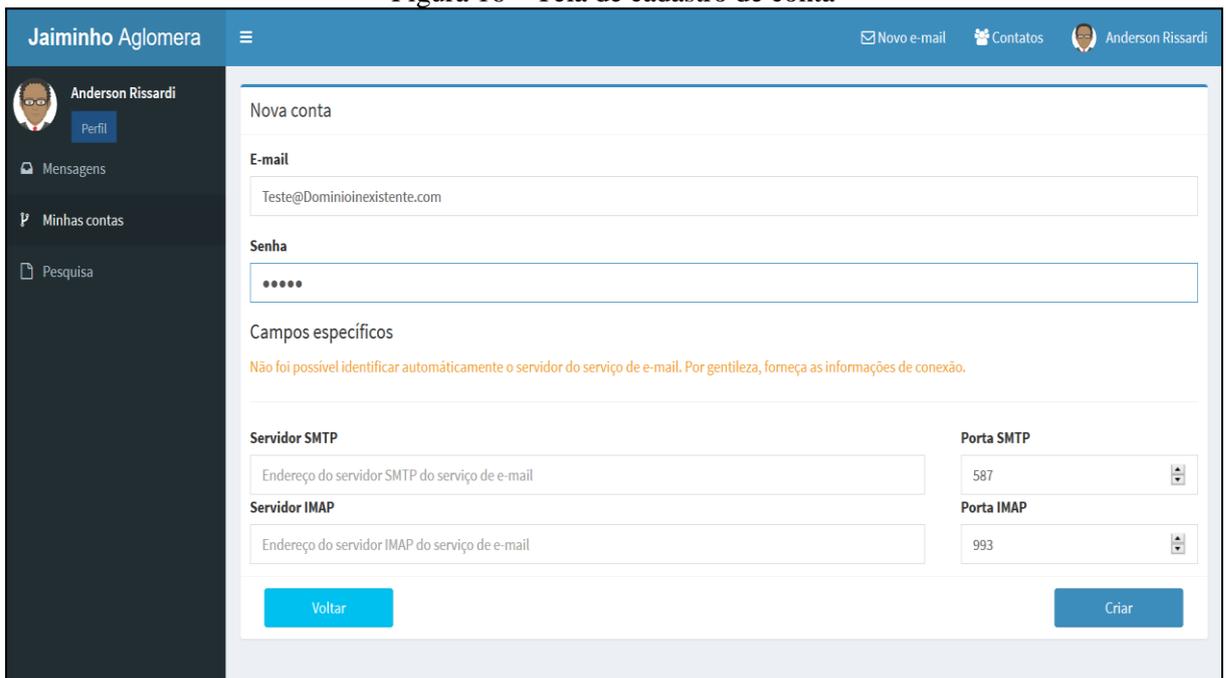
Figura 15 – Tela de exibição das contas de e-mail



Fonte: Do autor.

O usuário clica no botão *Nova conta* e é redirecionado para a tela de cadastro de contas de e-mail conforme ilustra a Figura 16. Devem ser informados os campos *E-mail* e *Senha*; se o sistema não identificar o domínio na sua base de conhecimento de conexões o sistema exibirá os campos *Endereço SMTP*, *Porta SMTP*, *Endereço IMAP* e *Porta IMAP* para que o usuário informe de maneira manual os dados de conexão com o serviço de e-mail da conta informada.

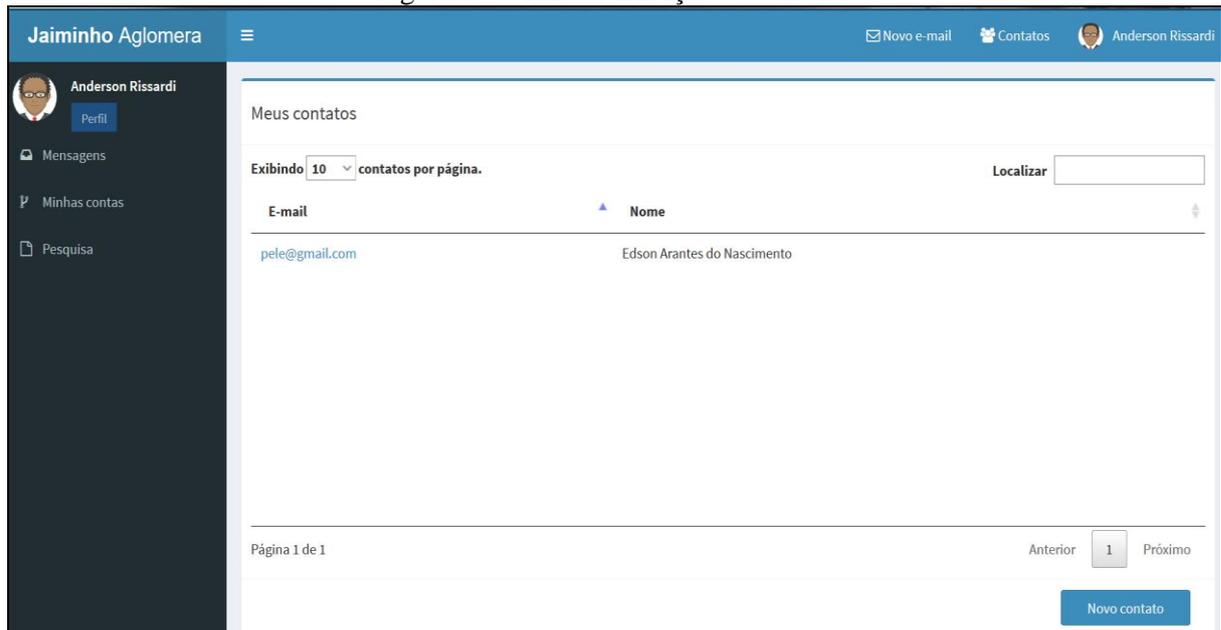
Figura 16 – Tela de cadastro de conta



Fonte: Do autor.

Ao cadastrar a conta a aplicação define que o monitoramento da conta está ativo, dessa maneira o usuário já tem acesso a todas as funcionalidades da aplicação, como ler as mensagens de e-mail e pesquisar por mensagens em todas as contas de e-mail cadastradas. Ao clicar no link `Contatos` o usuário acessa a tela de exibição de contatos conforme apresentado na Figura 17, aonde o usuário pode pesquisar pelo `Nome` ou pelo `endereço de E-mail` cadastrado, visualizar todos os contatos cadastrados e inserir novos contatos.

Figura 17 – Tela de exibição de contatos



Fonte: Do autor.

O botão `Novo contato` apresenta a tela de cadastro de novos contatos do usuário conforme a Figura 18 apresenta. Aonde é necessário informar o `Nome` do contato e o `Endereço de e-mail` e clicar em `Criar`.

Figura 18 – Tela de cadastro de contatos.

The screenshot shows the 'Novo contato' (New contact) form. The header includes the system name 'Jaiminho Aglomera' and navigation links for 'Novo e-mail', 'Contatos', and the user profile 'Anderson Rissardi'. The left sidebar contains navigation options: 'Perfil', 'Mensagens', 'Minhas contas', and 'Pesquisa'. The main form area has the title 'Novo contato' and two input fields: 'Nome' and 'E-mail'. Below the fields are two buttons: 'Contatos' and 'Criar'. At the bottom of the page, there is a footer: 'Trabalho de conclusão de curso da graduação de Sistemas da Informação da FURB - Fundação Universidade Regional de Blumenau. Desenvolvido por Anderson Rissardi'.

Fonte: Do autor.

Ao clicar no link `Mensagens` o usuário é redirecionado para a tela de visualização de mensagens das contas cadastradas que estão com o monitoramento ativo. O usuário pode parar de monitorar determinada conta de e-mail, basta acessar a tela de edição do cadastro de contas conforme ilustra a Figura 19 e desmarcar o campo `Monitorar está conta de e-mail` e clicar em `Salvar`.

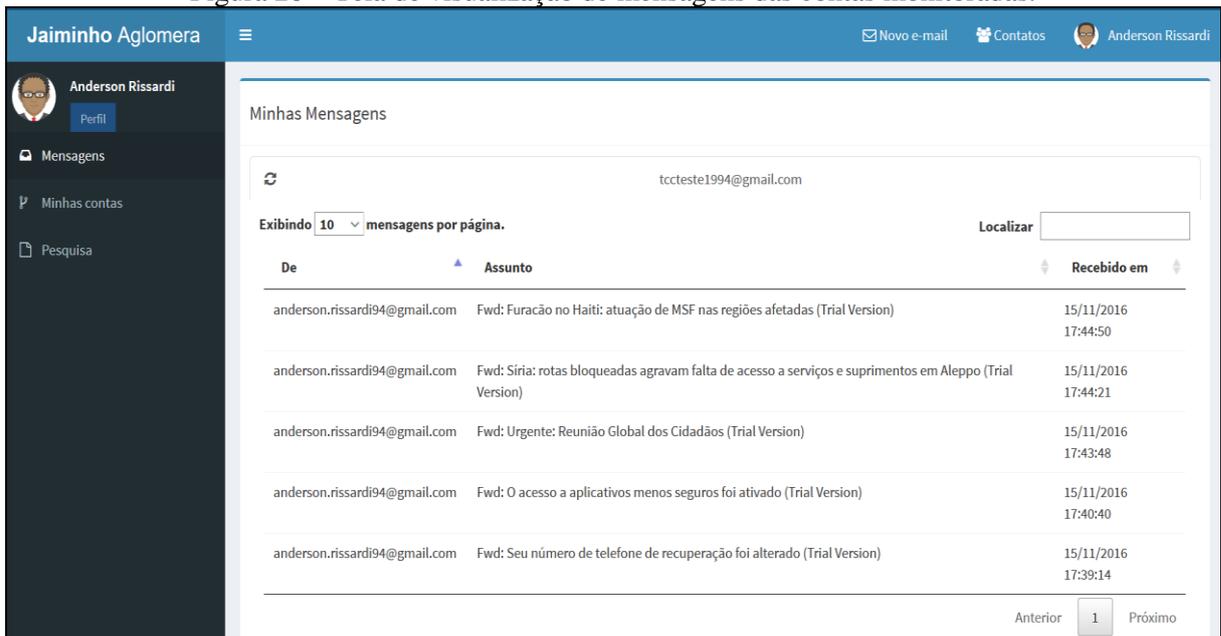
Figura 19 – Tela de edição de contas de e-mail.

The screenshot shows the 'Editar informações da conta' (Edit account information) form. The header and sidebar are identical to Figure 18. The main form area has the title 'Editar informações da conta' and several input fields: 'E-mail' (containing 'tcc teste1994@gmail.com'), 'Senha' (containing 'SenhaMaster123!'), 'Servidor SMTP' (containing 'smtp.gmail.com'), 'Porta SMTP' (a dropdown menu with '587' selected), 'Servidor IMAP' (containing 'imap.gmail.com'), and 'Porta IMAP' (a dropdown menu with '993' selected). There is a checkbox labeled 'Monitorar está conta de e-mail?' which is checked. At the bottom are three buttons: 'Voltar', 'Salvar', and 'Excluir'.

Fonte: Do autor.

A Figura 20 apresenta a tela de visualização das mensagens recebidas pelas contas de e-mail monitoradas no último mês. Aonde o usuário pode ordenar por data de recebimento, remetente ou assunto, além de poder realizar pesquisas por palavras nas mensagens listadas. Nesta imagem o usuário possui apenas uma conta de e-mail cadastrada.

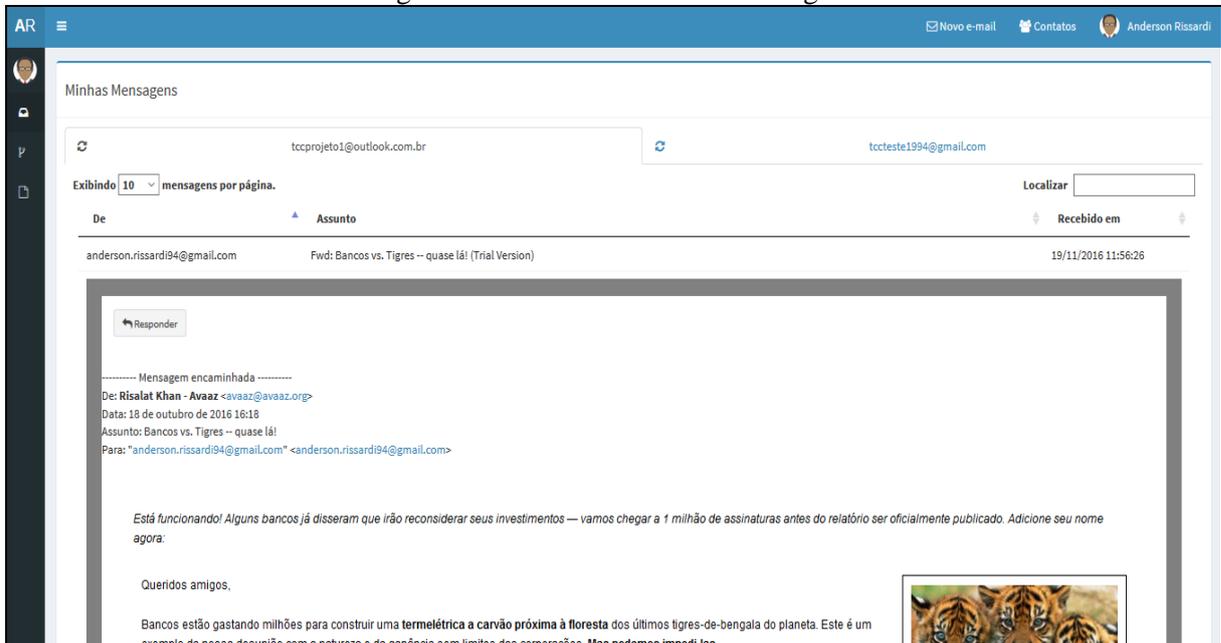
Figura 20 – Tela de visualização de mensagens das contas monitoradas.



Fonte: Do autor.

Ao clicar sobre uma mensagem o corpo da mensagem é apresentada e a possibilidade de responder à mensagem fica disponível conforme apresenta a Figura 21. Pode-se observar também que este usuário possui duas contas de e-mail monitoradas.

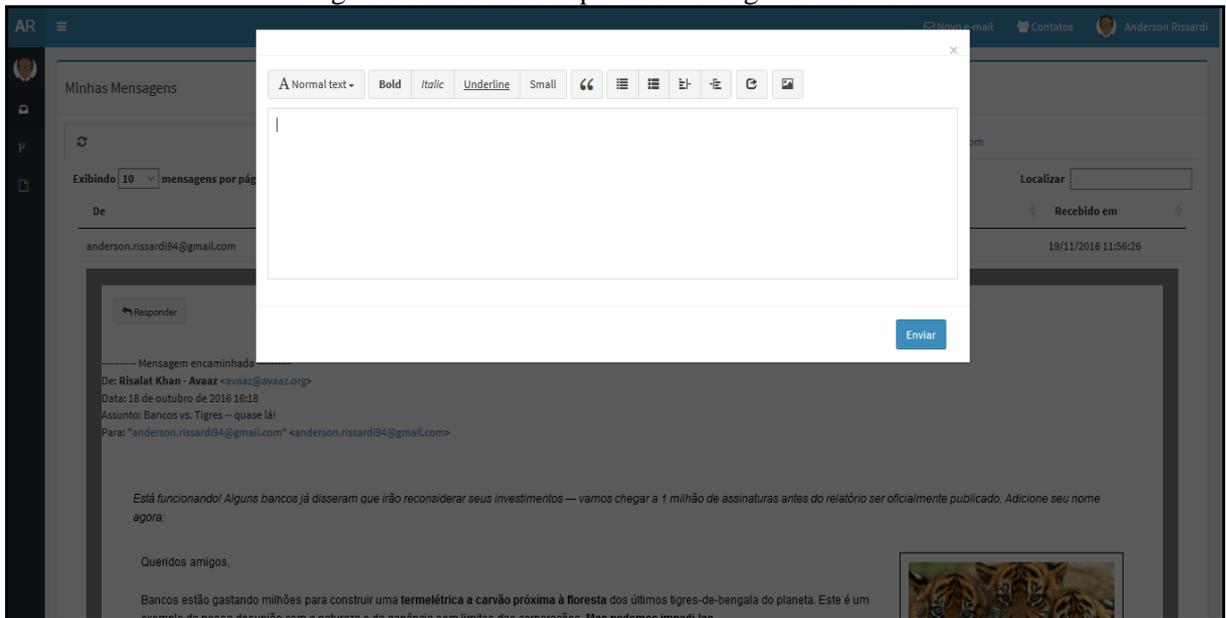
Figura 21 – Tela de leitura de mensagem



Fonte: Do autor.

Ao clicar em Responder a tela para resposta aparece para o usuário conforme apresentado na Figura 22. O usuário insere a mensagem que deseja enviar e clica em Enviar.

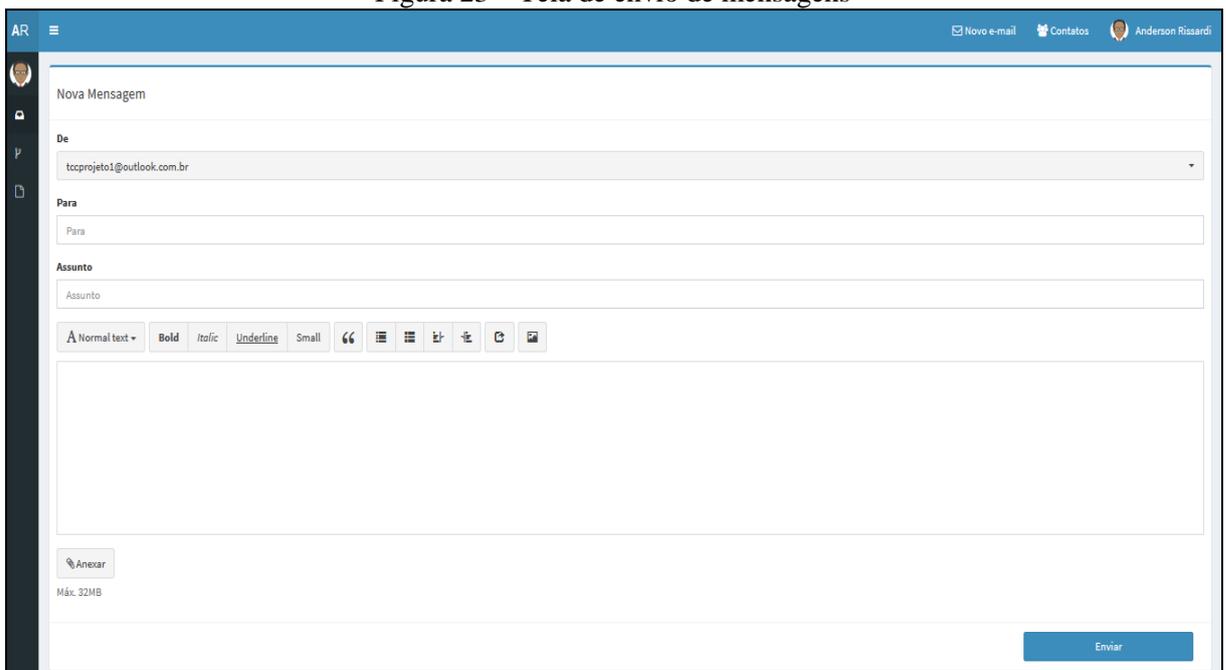
Figura 22 – Tela de responder mensagem de e-mail.



Fonte: Do autor.

Ao clicar no link **Novo e-mail** que fica disponível depois que o usuário cadastrar pelo menos 1 conta de e-mail, a tela de envio de mensagem de e-mail é apresentada conforme a Figura 23. Nela o usuário seleciona de qual conta de e-mail deseja enviar a mensagem, preenche o destinatário, assunto e é possível criar uma mensagem com diversas formatações e incluir anexos a essa mensagem de e-mail. A mensagem é enviada ao clicar em **Enviar**.

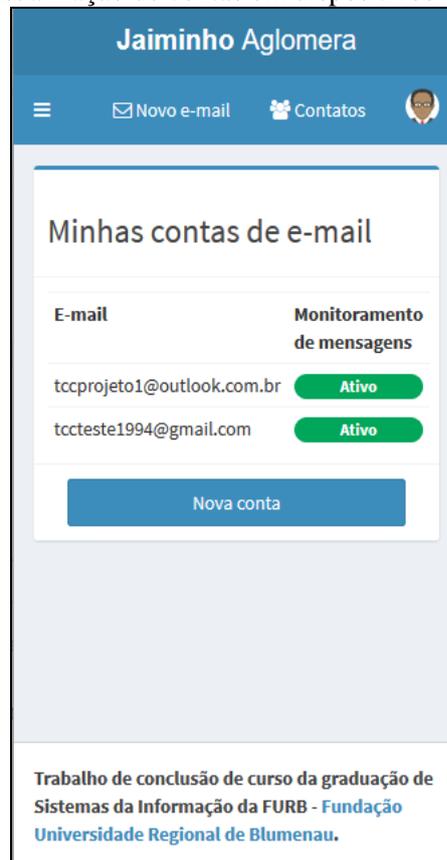
Figura 23 – Tela de envio de mensagens



Fonte: Do autor.

As telas do sistema foram desenvolvidas para atuarem de maneira responsiva, conforme o dispositivo que o usuário está utilizando. A Figura 24 apresenta de visualização de contas cadastradas, a mesma da Figura 15, porém no formato de dispositivos extremamente pequenos.

Figura 24 – Tela de visualização de contas em dispositivos extremamente pequenos



Fonte: Do autor.

Ao clicar no link *Pesquisa* é apresentada a tela de pesquisa avançada de e-mails, ela realiza uma pesquisa em todas as contas de e-mail que estão sendo monitoradas pela aplicação, aonde é possível realizar filtros pelo remetente, data de recebimento, assunto e por palavras dentro do corpo do e-mail, conforme apresenta a Figura 25, após clicar em *Pesquisar* as mensagens resultantes da pesquisa são apresentadas logo abaixo do botão *Pesquisar*, tendo as mesmas funcionalidades relatadas da tela de leitura de mensagens de e-mail, apresentadas na Figura 21.

Figura 25 – Tela de pesquisa avançada de e-mails.

Jaiminho Aglomera

Novo e-mail Contatos

Perfil

Mensagens

Minhas contas

Pesquisa

Pesquisa avançada

Assunto

De

A partir de

Mensagem

Tigres

Pesquisar

Exibindo 10 mensagens por página. Localizar

De	Recebido em	Assunto
anderson.rissardi94@gmail.com	19/11/2016	Fwd: Bancos vs. Tigres -- quase lá! (Trial Version)

Anterior 1 Próximo

Fonte: Do autor.

O usuário pode a qualquer momento alterar as informações do seu perfil de usuário, tais como Nome, Data de nascimento e a Imagem do perfil conforme apresenta a Figura 26, basta alterar as informações que foram salvas no momento do registro da conta e clicar em OK.

Figura 26 – Tela de perfil do usuário.

Jaiminho Aglomera

Novo e-mail Contatos Anderson Rissardi

Anderson Rissardi Perfil

Mensagens

Minhas contas

Pesquisa

Meu perfil

Selecionar arquivo... Nenhum arquivo selecionado. Atualizar imagem

Nome

Anderson Rissardi

Data de nascimento

14/07/1994

E-Mail

tcpprojeto1@outlook.com.br

tcteste1994@gmail.com

Monitoramento de mensagens

Ativo

Ativo

OK

Fonte: Do autor.

Foram apresentadas as telas do sistema e o comportamento da tela ao ser acessada por dispositivos com resoluções extremamente pequenas. Assim concluindo a apresentação da operacionalidade da aplicação desenvolvida.

3.4 RESULTADOS E DISCUSSÕES

A aplicação tem como finalidade apresentar ao usuário de forma simples e prática as mensagens de e-mail mais recentes das contas do usuário, sem se preocupar com o meio de acesso ao serviço de e-mail. O usuário também pode enviar mensagens, realizar pesquisas avançadas em todos os e-mails e manter um cadastro único de contatos.

Por ser responsiva, a aplicação se adapta e funciona em dispositivos desde resoluções muito baixas como 320px x 480px até as resoluções maiores e mais comumente utilizadas. Este recurso tornou o desenvolvimento desafiador, mas agregou um grande aprendizado, principalmente da linguagem HTML em conjunto com as bibliotecas de estilos, que deram forma à aplicação, como por exemplo a biblioteca Bootstrap e a AdminLTE que dispõem de diversas funcionalidades e ampla documentação para realizar o desenvolvimento responsivo.

No desenvolvimento da aplicação foi utilizado as bases conceituais do padrão MVC o que permitiu um grande aprendizado do funcionamento desse padrão que é atualmente referência no desenvolvimento de aplicações Web. A utilização da plataforma padrão da Microsoft de projetos MVC possibilitou conhecer as novas bibliotecas de desenvolvimento como a Microsoft Identity que garante a segurança da aplicação mantendo o acesso apenas dos usuários conectados à aplicação.

Ainda no desenvolvimento da camada servidora da aplicação, foi utilizada a biblioteca EntityFramework uma ORM (*Object-Relational Mapping*) para realizar o acesso a base de dados, deixando mais abstrato o acesso da aplicação à respectiva base de dados. Isso foi de muito aprendizado e como uma quebra de paradigma de acesso a bases de dados relacionais.

Alguns trabalhos e produtos foram usados como referência para a criação desta aplicação. O Microsoft Outlook permite ao usuário receber e enviar mensagens além de permitir o agendamento de reuniões e compartilhar os contatos entre todas as contas cadastradas, porém é executado apenas em ambiente *desktop* e nos sistemas operacionais Windows e Mac, obrigando o usuário a utilizar um dos sistemas operacionais disponíveis através de um dispositivo específico. Tendo que realizar o download e instalação da aplicação.

O Mozilla ThunderBird em sua essência é bastante parecido com o Microsoft Outlook, até em suas interfaces e funcionalidades. Porém além de poder ser utilizado nos sistemas

operacionais Windows e Mac também pode ser utilizado no Linux. Também sendo necessário realizar o download da aplicação e a sua instalação.

O Google Mail diferentemente das outras aplicações apresentadas permite o acesso através do navegador de internet do usuário permitindo realizar o envio e recebimento de mensagens independente do sistema operacional e sem necessidade de realizar instalação de aplicações. Porém não foi desenvolvida de forma responsiva, assim não é possível utilizar o Google Mail através de navegadores de internet em dispositivos com a resolução pequena. Para atender a demanda de dispositivos móveis a Google disponibiliza um aplicativo, dessa maneira obrigando o usuário a realizar o download de uma aplicação.

O trabalho de Eberhardt faz uma abordagem mais gerencial das mensagens de e-mail. Um controle de mensagens que chegam e saem do serviço. Apresentando relatórios gerenciais e controle *anti-span*.

Analisando os trabalhos correlatos pode se verificar que cada aplicativo possui suas características e funcionalidades específicas. O Quadro 10 faz uma comparação entre os aplicativos comerciais Microsoft Outlook, Mozilla ThunderBird e Google Mail com a aplicação desenvolvida.

Quadro 10 – Características dos trabalhos correlatos.

Aplicativo	Conecta em serviços de diferentes domínios	É Web	É responsivo
Microsoft Outlook	X		Não se aplica
Mozilla ThunderBird	X		Não se aplica
Google Mail	Parcialmente	X	
Aplicação Desenvolvida	X	X	X

Fonte: Do autor.

O Google Mail exige que o usuário tenha uma conta com o domínio gmail e disponibiliza o serviço que permite que o usuário cadastre de maneira gratuita, até 5 contas de e-mail de domínios diferentes para a realização do monitoramento de mensagens.

Quando uma nova mensagem chega na conta de e-mail monitorada o serviço encaminha essa mensagem para a conta de e-mail com o domínio gmail, assim replicando a mensagem e sem deixar que o usuário responda a mensagem, já que ela é um encaminhamento de outra conta do usuário.

No começo do desenvolvimento da aplicação outra biblioteca foi escolhida para realizar a integração da aplicação com os serviços de e-mail. A biblioteca Lumisoft que é *open source* e gratuita para uso não respondeu bem aos primeiros testes e devido a escassa documentação acabou-se desistindo da sua utilização, aonde perdeu-se um bom tempo pesquisando por outras bibliotecas que fizessem o mesmo que Lumisoft, porém mais estáveis.

4 CONCLUSÕES

O objetivo deste trabalho, de desenvolver uma aplicação Web e responsiva que permita o envio e leitura de mensagens de e-mail dos mais variados serviços de e-mail foi alcançado. As mensagens de e-mail de diferentes contas são agrupadas em apenas uma aplicação, permitindo assim, o acesso através de apenas uma aplicação e a partir de dispositivos de diferentes resoluções, sem a necessidade de realizar o download e instalação de outros programas de e-mail.

O desenvolvimento dessa aplicação segue a tendência apresentada na pesquisa de THE RADICATI GROUP (2015) que afirma que o número de usuários dos serviços de e-mail vai continuar aumentando e em 2019 irá atingir a marca de 2,9 bilhões de usuários e que cada usuário terá em média 1,9 contas de e-mail de serviços diferentes, com diferentes meios de acesso a esses serviços. Analisando essa informação e visando agilizar o processo de acesso a diferentes serviços de e-mail, observou-se que é possível atender esses usuários, permitindo o acesso de apenas uma aplicação e a partir dos mais diversos dispositivos.

A partir da aplicação desenvolvida foi possível aprimorar os conhecimentos de desenvolvimento responsivo e que no futuro será cada vez mais e mais comum, aplicações Web desenvolvidas para serem acessadas a partir de dispositivos com as mais diferentes resoluções. A biblioteca de estilos Bootstrap se mostrou bastante confiável e eficiente assim como as bibliotecas Entity Framework e jQuery.

A aplicação se mostrou uma grande fonte de conhecimento e que todo o conhecimento gerado será utilizado futuramente, seja com o padrão de desenvolvimento MVC ou com as bibliotecas utilizadas para a realização deste trabalho.

4.1 EXTENSÕES

Para a continuidade do presente trabalho, sugere-se:

- a) permitir que o usuário armazene uma lista de e-mails favoritos que fiquem sempre disponíveis ao usuário;
- b) permitir configurar a data de início da listagem de mensagens de e-mail;
- c) permitir o compartilhamento dos contatos entre os usuários;
- d) permitir categorizar as mensagens;
- e) permitir realizar ações no serviço de e-mail, tais como: excluir, mover, marcar como não lida;
- f) permitir encaminhar uma mensagem de e-mail;
- g) realizar um destaque nas mensagens não lidas;

- h) destacar a quantidade de mensagens não lidas;
- i) apresentar a caixa de saída das contas cadastrada;

REFERÊNCIAS

- ADMINSYSTEM SOFTWARE LIMITED. **EAGetMail POP3 & IMAP4 Component Overview**. [S. l.], [2016?]. Disponível em: <<https://www.emailarchitect.net/eagetmail/>>. Acesso em 22 out. 2016.
- ADMINSYSTEM SOFTWARE LIMITED. **EASendMail SMTP Component Overview**. [S. l.], [2016?]. Disponível em: <<https://www.emailarchitect.net/eagetmail/>>. Acesso em 04 nov. 2016.
- ADMINSYSTEM SOFTWARE LIMITED. **Retrieve Email and Parse Email in C# - Tutorial**. [S. l.], [2016?]. Disponível em: <<https://www.emailarchitect.net/eagetmail/kb/csharp.aspx?cat=0>>. Acesso em 22 out. 2016.
- BOOTSTRAP. **Grid System**. [S. l.], [2016?]. Disponível em: <<https://v4-alpha.getbootstrap.com/layout/grid/>>. Acesso em 06 nov. 2016.
- CARNEIRO, Márcio Rodrigo de Freitas. **Treinamento em UNIX(Linux/Solaris) para usuários do IME**. 2000. 76 f. Monografia – Instituto de Matemática e Estatística da Universidade de São Paulo, São Paulo, 2000. Disponível em: <<https://linux.ime.usp.br/~cef/mac499-00/monografias/pipo/mac499/texto/curso.pdf>>. Acessado em: 30 mar. 2016.
- CARNEIRO, Rui A. R. **Unificação e expansão da interface de pesquisa no webmail**. [S. l.], [2008]. Disponível em: <<http://www.inesporto.pt/~jsc/students/2009RuiCarneiro/2009relatorioRuiCarneiro.pdf>>. Acesso em: 26 mar. 2016.
- CARVALHO, Tereza C. M. de B. **Arquiteturas de redes de computadores OSI e TCP/IP**. São Paulo: Makron Books, 1996.
- EBERHARDT, Guilherme L. **Ferramenta para o controle de e-mails e anti-spam**. 2007. 49 f. TCC (Graduação) - Curso de Sistemas de Informação, Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau, 2007. Disponível em: <<http://dsc.inf.furb.br/arquivos/tccs/monografias/TCC2007-2-08-VF-GuilhermeLEberhardt.pdf>>. Acesso em: 30 mar. 2016.
- Garrett, Filipe. **A história do Gmail**. [S. l.], [2015]. Disponível em <<http://www.techtudo.com.br/noticias/noticia/2015/11/historia-do-gmail-conheca-origem-do-email-do-google.html>>. Acesso em: 19 nov. 2016.
- GOOGLE INC. **Sobre o Gmail**. [S. l.], [2016?]. Disponível em <<https://www.google.com/intl/pt-BR/mail/help/about.html>>. Acesso em: 27 mar. 2016.
- GOOGLE INC. **Usar o Gmail para verificar e-mails de outras contas**. [S. l.], [2016?]. Disponível em: <<https://support.google.com/mail/answer/21289?hl=pt-BR>>. Acesso em: 30 mar. 2016.
- INTERNET ENGINEERING TASKFORCE. **RFC 1939: POST OFFICE PROTOCOL**. Pittsburgh, Pensilvânia. 1996. 23 p. Disponível em: <<https://www.rfc-editor.org/rfc/rfc1939.txt>>. Acesso em: 31 mar. 2016.
- INTERNET ENGINEERING TASKFORCE. **RFC 2060: INTERNET MESSAGE ACCESS PROTOCOL**. Washington, Seattle. 1996. 79 p. Disponível em: <<https://www.rfc-editor.org/rfc/rfc2060.txt>>. Acesso em: 31 mar. 2016.

INTERNET ENGINEERING TASKFORCE. **RFC 5321: SIMPLE MAIL TRANSFER PROTOCOL**. Marina del Rey: Information Sciences Institute, 2008. 95 p. Disponível em: <<https://tools.ietf.org/html/rfc5321>>. Acesso em: 31 mar. 2016.

INTERNET ENGINEERING TASKFORCE. **RFC 821: SIMPLE MAIL TRANSFER PROTOCOL**. Marina del Rey: Information Sciences Institute, 1982. 61 p. Disponível em: <<http://www.isi.edu/in-notes/rfc821.txt>>. Acesso em: 31 mar. 2016.

KNORST, Alexandre Eduardo. **Protótipo de software para notificação de recebimento de novos e-mail's, baseado no protocolo IMAP**. 2000. 81 f. TCC (Graduação) - Curso de Ciências da Computação, Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau, 2000. Disponível em: <<http://dsc.inf.furb.br/arquivos/tccs/monografias/2000-2alexandreduardoknorstvf.pdf>>. Acesso em: 30 mar. 2016.

LEVINE, John R.; Young, Margaret L. **Internet para leigos**. 13. ed. Rio de Janeiro: Alta Books, 2013.

LOPES, Sérgio. **A Web Mobile: Design Responsivo e além para uma Web adaptada ao mundo mobile**. São Paulo: Casa do Código. 2013. Disponível em: <https://books.google.com.br/books?id=E3CCCwAAQBAJ&pg=PT132&dq=web+design+responsivo&hl=pt-BR&sa=X&ved=0ahUKEwi_uvzOlb_QAhVIQZAKHc-xBDsQ6AEIOjAB#v=onepage&q=web%20design%20responsivo&f=false>. Acesso em: 23 nov. 2016.

MICROSOFT CORPORATION. **Conectado aos seus contatos – Ajuda do Outlook**. [S. l], 2016. Disponível em: <<http://windows.microsoft.com/pt-br/windows/outlook/connected-people>>. Acesso em: 02 abr. 2016.

MICROSOFT CORPORATION. **C#**. [S. l], [2016?]. Disponível em: <<https://msdn.microsoft.com/pt-BR/library/kx37x362.aspx>>. Acesso em: 04 nov. 2016.

MICROSOFT CORPORATION. **Microsoft Office Outlook**. [S. l], [2016?]. Disponível em <<https://products.office.com/pt-br/outlook/email-and-calendar-software-microsoft-outlook>>. Acesso em: 27 mar. 2016.

MICROSOFT CORPORATION. **A Brief History of Microsoft's mail and Exchange**. [S.l], 2004. Disponível em: <<https://blogs.msdn.microsoft.com/gilg/2004/08/18/a-brief-history-of-microsofts-mail-and-exchange-clients-on-the-mac-part-i/>>. Acesso em: 23 out. 2016.

MICROSOFT CORPORATION. **Visual Studio Community**. [S. l], [2016?]. Disponível em: <<https://www.visualstudio.com/vs/community/>>. Acesso em: 05 nov. 2016.

MOZILLA ORGANIZATION. **Thunderbird**. [S. l], [2016?]. Disponível em <<https://www.mozilla.org/pt-BR/thunderbird/>>. Acesso em: 27 mar. 2016.

OLIVEIRA, Leonardo; DAVI, Arnaut; MAZIERO, Carlos. **Um servidor de e-mail distribuído, flexível, escalável e tolerante a intrusões**. [S. l], 2005. Disponível em: <https://tspnyg-sn3301.files.1drv.com/y3mKT2BAhiGswJWdg0wqNqKJfVy1ZLQZa2OU5XRNYgKpi_FRWaH0pwPNdFZt9uWkHNu7ErVf9cfL6EeNdw_JLOloW95lnTN8uuZD9H9-VFJLg_b4lzDZhpnGCTu_7zSoOwJ/2005-sbseg.pdf?psid=1>. Acesso em: 27 mar. 2016.

REGIS, Eduardo Jean. **Protótipo de software para distribuição de arquivos recebidos por e-mail via internet**. 2000. 62 f. TCC (Graduação) - Curso de Ciências da Computação, Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau, 2000. Disponível em: <<http://dsc.inf.furb.br/arquivos/tccs/monografias/2000-1eduardojeanregisvf.pdf>>. Acesso em: 31 mar. 2016.

SHARP, John. **Microsoft Visual C# 2010, passo a passo**. Porto Alegre: Bookman, 2011. 780 p.

TEIXEIRA, Fabricio. **O que é Responsive Web Design?** [S. l], 2011. Disponível em: <<https://brasil.uxdesign.cc/o-que-%C3%A9-responsive-web-design-ab292eb616b7#.ywkhdyrbp>>. Acesso em: 23 nov. 2016.

THE JQUERY FOUNDATION. **What is jQuery?** [S. l], 2016. Disponível em <<https://jquery.com/>>. Acesso em: 05 nov. 2016.

THE RADICATI GROUP. **Email Statistics Report, 2015-2019**. Palo Alto, California. 2015. Disponível em: <<http://www.radicati.com/wp/wp-content/uploads/2015/02/Email-Statistics-Report-2015-2019-Executive-Summary.pdf>>. Acessado em: 19/11/2016.

ZEMEL, Tércio. **Web Design Responsivo: Páginas adaptáveis para todos os dispositivos**. São Paulo, 2013. Disponível em: <https://books.google.com.br/books?id=82SCCwAAQBAJ&pg=SA1-PA1&dq=interface+responsiva&hl=pt-BR&sa=X&redir_esc=y#v=onepage&q=interface%20responsiva&f=false>. Acesso em: 23 nov. 2016.

APÊNDICE A – Descrição dos Casos de Uso

Esta seção apresenta a descrição dos casos de uso conforme previstos nos diagramas apresentados nas Figuras 5 e 6.

Para os principais casos de uso, sendo eles o UC07 e UC12, foram definidas as pré-condições e pós-condições, bem como os cenários com fluxo de eventos principal, alternativo e de exceção.

UC01. Verificar sessão do usuário

Verifica a sessão de acesso ao sistema para identificar se existe algum usuário ativo, se a sessão é inválida ou esta expirada.

UC02. Cadastrar usuário

Permite o cadastro de uma conta de usuário do sistema. Informando os dados básicos para o cadastro.

UC03. Ativar/Desativar monitoramento de contas de e-mail

Permite que o usuário definir se a conta de e-mail cadastrada no sistema está ou não sendo monitorada, definindo se a aplicação deve buscar as mensagens, realizar pesquisa avançada e realizar o envio de mensagens a partir da conta.

UC04. Cadastrar perfil do usuário

Permite que o usuário altere as informações básicas da conta e permite o cadastro de informações sobre o usuário.

UC05. Anexar arquivos a um e-mail

Permite que o usuário anexar arquivos ao enviar uma mensagem de e-mail.

UC06. Login

Realiza o *login* no sistema dando acesso a todas as outras funcionalidades do sistema.

UC07. Gerenciar contas de e-mail

Permitir que o usuário cadastre, visualize, edite e exclua as contas de e-mail.

Constraints

Pré-condição. O usuário deve estar cadastrado e conectado no sistema.

Pós-condição. Contas de e-mails foram cadastradas, alteradas, excluídas.

Pós-condição. Servidor de serviço de e-mail cadastrado ou consultado.

Cenários

Cadastrar conta de e-mail {Principal}.

1 - O usuário seleciona contas de e-mail.

2 – O usuário seleciona novo.

3 - O usuário entra com a conta de e-mail, usuário e senha.

4 - O sistema busca no cadastro de servidores um servidor host do serviço de e-mail informado.

5 - O sistema faz um teste de conectividade utilizando as informações inseridas e o servidor selecionado.

6 - O sistema apresenta uma mensagem informando o resultado do teste de conectividade.

7 - O sistema grava as informações e conclui o caso de uso.

Cadastrar contas de e-mail e servidor *host* do serviço de e-mail {Alternativo}.

No passo 4, o sistema não identifica um servidor em sua base de dados e pede para o usuário inserir as especificações

4.1. O sistema apresenta os campos para o usuário informar os dados de conectividade do servidor.

4.2. O usuário informa os dados do servidor.

Consulta de contas cadastradas {Alternativo}.

No passo 1, o sistema exibe uma lista das contas cadastradas.

1.1. O sistema apresenta uma lista as contas cadastradas através do nome da conta e o endereço de e-mail.

1.2 - O usuário clica em cima do registro e é apresentada uma tela com todas as informações da conta cadastrada.

Alterar uma conta de e-mail {Alternativo}.

No passo 1, o sistema exibe uma lista das contas cadastradas.

1.1. O sistema apresenta uma lista as contas cadastradas através do nome da conta e o endereço de e-mail.

1.2 - O usuário clica em cima do registro e é apresentada uma tela com todas as informações da conta cadastrada.

1.3 – O usuário altera as informações.

1.4 – O sistema grava as alterações

Excluir conta de e-mail {Alternativo}.

No passo 1, o sistema exibe uma lista das contas cadastradas.

1.1. O sistema apresenta uma lista as contas cadastradas através do nome da conta e o endereço de e-mail.

1.2 - O usuário seleciona a(s) conta(s) que deseja excluir.

1.3 - O sistema exclui as contas selecionadas e encerra o caso de uso.

UC08. Manter as credenciais do usuário ao realizar *login*

Permitir que as credenciais utilizadas pelo usuário para realizar *login* continuem armazenadas no dispositivo e que ao acessar novamente a aplicação seja possível realizar a conexão de maneira automática.

UC09. Enviar e-mails

Permitir que o usuário envie e-mails através das contas cadastradas.

UC11. Gerenciar contatos

Permitir que o usuário cadastre, altere, filtre e visualize os seus contatos.

UC12. Visualizar e-mails importados para o sistema

O sistema deve apresentar as mensagens de e-mails dos serviços de e-mails das contas cadastradas recebidas nos últimos 30 dias.

Constraints

Pré-condição. O usuário deve estar cadastrado e conectado.

Pré-condição. As contas de e-mail devem estar cadastradas.

Pré-condição. O serviço de e-mail da conta deve estar operante.

Pós-condição. Os e-Mails são apresentados para o usuário.

Cenários

Apresentar e-mails para o usuário {Principal}.

1- O sistema deve buscar as contas de e-mail cadastradas.

2 - O sistema deve buscar as mensagens de e-mail que atenderem o filtro informado nas regras de sincronização (mensagens recebidas no último mês) do serviço de e-mail da conta e cadastrar no sistema o e-mail.

Contas de e-mail não cadastradas {Exceção}.

No passo 1, o sistema não identifica nenhuma conta de e-mail cadastrada.

1.1 - O sistema apresenta uma mensagem informando que não é possível encontrar nenhuma mensagem por que não existe nenhuma conta de e-mail cadastrada.

1.2 - Encerra o caso de uso.

UC13. Pesquisar e-mails importados para o sistema

Permitir que o usuário realize pesquisas avançadas nas mensagens de e-mails diretamente no serviço nativo de e-mail. Pesquisando por conteúdo do e-mail, remetente, assunto ou data de recebimento da mensagem.