

**FACULDADE DE TECNOLOGIA DA PARAÍBA – FATEC/PB
TECNOLOGIA EM SISTEMAS PARA INTERNET
DERIK REGINALDO DA CONCEIÇÃO**

**LEVANTAMENTO DE REQUISITOS PARA CRIAÇÃO DE UM PORTAL WEB
PARA CLIENTES DE UMA DISTRIBUIDORA DE MEDICAMENTOS**

**CABEDELLO
2017**

DERIK REGINALDO DA CONCEIÇÃO

**LEVANTAMENTO DE REQUISITOS PARA CRIAÇÃO DE UM PORTAL WEB
PARA CLIENTES DE UMA DISTRIBUIDORA DE MEDICAMENTOS**

Relatório técnico apresentado ao Curso de Sistemas para Internet na Faculdade de Tecnologia da Paraíba – FATEC/PB como requisito para obtenção do título de tecnólogo em Sistema para Internet.

ORIENTADOR: Prof. Dr. Marcelo Fernandes de Sousa

**CABEDELO
2017**

Dados Internacionais de Catalogação na Publicação (CIP)
Biblioteca Padre Joaquim Colaço Dourado

C7441 Conceição, Derik Reginaldo da

Levantamento de requisitos para criação de um portal web para clientes de uma distribuidora de medicamentos / Derik Reginaldo da Conceição. – Cabedelo, PB: [s.n], 2017.
38p.

Orientador. Prof. Marcelo Fernandes de Sousa. Monografia (Graduação Tecnológica em Sistemas para Internet) – Faculdade de Tecnologia da Paraíba – FATEC-PB.

1. Desenvolvimento de sistemas. 2. Sistemas web. 3. Levantamento de requisitos. 4. Distribuidora de medicamentos. I. Título.

CDU 004.43

DERIK REGINALDO DA CONCEIÇÃO

**LEVANTAMENTO DE REQUISITOS PARA CRIAÇÃO DE UM PORTAL WEB
PARA CLIENTES DE UMA DISTRIBUIDORA DE MEDICAMENTOS**

Relatório técnico apresentado ao Curso de Sistemas para Internet na Faculdade de Tecnologia da Paraíba – FATEC/PB como requisito para obtenção do título de tecnólogo em Sistema para Internet.

Aprovada em: 18 de Dezembro de 2017.

BANCA EXAMINADORA

Prof. Dr. Marcelo Fernandes de Sousa (orientador)
Instituto de Educação Superior da Paraíba

Prof. Fernanda Carolina Ferreira
Instituto de Educação Superior da Paraíba

Prof. Danilo Rangel Arruda Leite
Instituto de Educação Superior da Paraíba

AGRADECIMENTOS

Primeiramente a Deus, por ter me dado saúde e força para superar as dificuldades ao longo dessa jornada.

Agradeço ao meus pais, pelo amor, incentivo e apoio incondicional.

Agradeço aos meus amigos e companheiros de trabalho que fizeram parte da minha formação.

A todos meus professores, não somente por terem ensinado, mas por terem me feito aprender. A palavra mestre, nunca fará justiça aos professores dedicados aos quais sem nominar terão os meus eternos agradecimentos.

RESUMO

No presente trabalho o autor visa o processo de levantamento de requisitos para criação de um portal web, afim de diminuir o fluxo de solicitações internas e aumentar a qualidade dos serviços ofertados pela empresa. O mercado atual exige cada vez mais recursos tecnológicos, para agregar valor ao negócio e fidelizar seus clientes. Para tal fim, o relatório utiliza os conceitos da área da Tecnologia da Informação de Linguagem de Modelagem Unificada, focando na utilização do diagrama de caso de uso e baseado na ferramenta LucidChart, de Webservice, aplicado por meio da linguagem de programação AdvPL e com as ferramentas Totvs Developer Studio 11.3 e SoapUI. A partir desse referencial, o trabalho realiza o levantamento de requisitos funcionais e não funcionais visando a qualidade e solidez da solução a ser entregue. O processo de levantamento de requisito foi o primeiro passo para o desenvolvimento do projeto, onde foram identificados dados essenciais para criação do portal. Nessa etapa chegamos a conclusão de que há uma carência tecnológica para o fluxo de impressão de segunda via de Notas e boletos. Logo depois, são descritas as etapas e os responsáveis pelas entregáveis de desenvolvimento, testes e homologação. De modo geral, é possível identificar os benefícios da modelagem e documentação dos requisitos, tendo em vista o atendimento da necessidade com poucas mudanças do exposto no levantamento para o implantado. Além disso, com a adesão dos clientes ao sistema gera-se valor que tendência a soma de mais funcionalidade no futuro.

Palavras-chave: Portal Web. Linguagem de Modelagem Unificada. UML. Webservice.

ABSTRACT

In the present work the author aims the process of surveying the requirements for creating a web portal, in order to reduce the flow of internal requests and increase the quality of the services offered by the company. The current market demands more and more technological resources, to add value to the business and to retain its customers. To this end, the report uses the concepts of the Unified Modeling Language Information Technology area, focusing on the use of the use case diagram and based on the LucidChart tool of Webservice, applied through the programming language AdvPL and with the Totvs Developer Studio 11.3 and SoapUI tools. From this referential, the work carries out the survey of functional and not functional requirements aiming at the quality and solidity of the solution to be delivered. The requirement survey process was the first step in the development of the project, where key data was identified for portal creation. At this stage we conclude that there is a technological shortage for the flow of second printing of notes and tickets. Soon after, the stages and the responsible of the deliverables of development, tests and homologation are described. In general, it is possible to identify the benefits of modeling and documentation of the requirements, in order to meet the need with few changes from the above in the survey to the one implemented. In addition, with the adhesion of customers to the system generates value that tends to add more functionality in the future.

Keywords: Web Portal. Unified Modeling Language. UML. Webservice.

LISTA DE ILUSTRAÇÕES

Figura 1 – Tecnologias e Ferramentas Utilizadas	11
Figura 2 – Diagrama de Caso de Uso	13
Figura 3 – Componentes do Diagrama	13
Figura 4 – AdvPL	15
Figura 5 – LucidChart	17
Figura 6 – TOTVS Developer Studio.....	18
Figura 7 – SoapUI	19
Figura 8 – Diagrama de Caso de Uso do Portal	21
Figura 9 – Tela de Login	22
Figura 10 – Tela Criação de Conta.....	23
Figura 11 – Tela Cadastro Aguardando Aprovação	24
Figura 12 – Tela Solicitação de Cadastro	25
Figura 13 – Tela de Recuperação de Senha.....	26
Figura 14 – Tela Cadastro de Senha.....	27
Figura 15 – Tela de Senha Cadastrada com Sucesso.....	27
Figura 16 – Tela de Acesso Negado	28
Figura 17 – Tela de Títulos	29
Figura 18 – Método verifica CNPJ.....	32
Figura 19 – Método cadastrar ID.....	32
Figura 20 – Método GetTitulo.....	33
Figura 21 – Método gera boleto	34

LISTA DE ABREVIATURAS E SIGLAS

TI	Tecnologia da Informação
WS	Web Service
UML	Linguagem de Modelagem Unificada
AdvPL	Advanced Protheus Language
ID	Identificador

SUMÁRIO

1	INTRODUÇÃO	09
1.1	OBJETIVO GERAL	10
1.2	OBJETIVOS ESPECÍFICOS	10
2	TECNOLOGIAS UTILIZADAS	11
2.1	LINGUAGEM DE MODELAGEM UNIFICADA (UML)	11
2.1.1	Diagrama de Caso de Uso	12
2.2	WEB SERVICE	14
2.2.1	AdvPL	15
2.3	FERRAMENTAS UTILIZADAS	16
2.3.1	Lucidchart	16
2.3.2	Totvs Developer Studio 11.3	17
2.3.3	SOAPUI	18
3	LEVANTAMENTO DE REQUISITOS	20
3.1	DIAGRAMA E ESPECIFICAÇÕES	20
3.2	REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS	21
3.2.1	Requisitos Funcionais	22
3.2.2	Requisitos Não Funcionais	30
4	DESENVOLVIMENTO	31
5	CONSIDERAÇÕES FINAIS	35
	REFERÊNCIAS	36
	APÊNDICE A - Métodos do Webservice	37

1 INTRODUÇÃO

Diante de um mercado de distribuição de medicamentos cada vez mais concorrido e com os avanços tecnológicos, as empresas que buscam o desenvolvimento de sistemas web que possam melhorar processos internos e transformá-los em serviços que gerem valor para os seus clientes criam chances de predileção e fidelização deles.

Nesse contexto, fora idealizada uma proposta de solução tecnológica em uma organização desse ramo que buscava solidez e qualidade na construção de uma aplicação que sanasse a baixa satisfação de seus consumidores quanto aos seus processos financeiros de recebimento de títulos e cobrança.

Desse modo, o presente trabalho visa apresentar o processo de levantamento de requisitos para implementação de um portal online para cliente, com o propósito de ofertar serviços clientes da companhia de distribuição, por meio de um sistema web com segurança.

O projeto consiste na criação de uma interface administrativa que vai permitir aos clientes de uma distribuidora de medicamentos retirar segunda via de boletos e imprimir notas fiscais.

Hoje o processo é feito de forma manual, onde o setor de cobrança e o SAC (serviço de atendimento ao cliente) são os responsáveis pelo atendimento da demanda. Os prazos para atendimento dessas solicitações são de 48 horas a partir do contato realizado pelo cliente, seja por e-mail ou telefonema.

O processo para emissão de uma segunda via de boleto é dividido em três etapas. 1º Etapa: alteração de vencimento se o título estiver em atraso, senão, basta emitir uma segunda via e encaminhar para o cliente e o processo será concluído. 2º Etapa: aprovação e 3º etapa: emissão do boleto.

Na primeira, caso o boleto esteja vencido, o atendente terá que alterar a data de vencimento para o dia desejado pelo cliente e aplicar os juros de acordo com os dias em atraso, caso contrário, se o título está dentro do prazo para pagamento, é só o atendente imprimir uma segunda via do boleto e encaminhar ao cliente por e-mail, tudo isso é feito dentro do sistema Protheus.

Na segunda, é enviado um e-mail (workflow) para os aprovadores, onde contém um link com todas as informações alteradas, nesse momento o aprovador decide se é devido ou não as alterações feitas pelo atendente, se a resposta do

aprovador for positiva, após a confirmação o título já estará disponível no sistema com as novas configurações.

Em seguida, inicia-se a terceira etapa, onde o atendente irá emitir a segunda via do boleto com a nova data e valor ajustado.

O desenvolvimento do trabalho registrado neste relatório técnico foi realizado seguindo os seguintes passos: especificação das Tecnologias Utilizadas e Ferramentas Utilizadas contidas no Capítulo 2; a descrição da Levantamento de Requisitos no Capítulo 3; o desdobramento do Desenvolvimento do Projeto no Capítulo 4. Por fim, serão apresentadas as Considerações Finais, atreladas ao atendimento dos objetivos, no Capítulo 5.

1.1 OBJETIVO GERAL

Levantar todos os requisitos necessários para criação de um portal web para clientes de uma distribuidora de medicamentos, e desenvolver um webservice para integração dos sistemas.

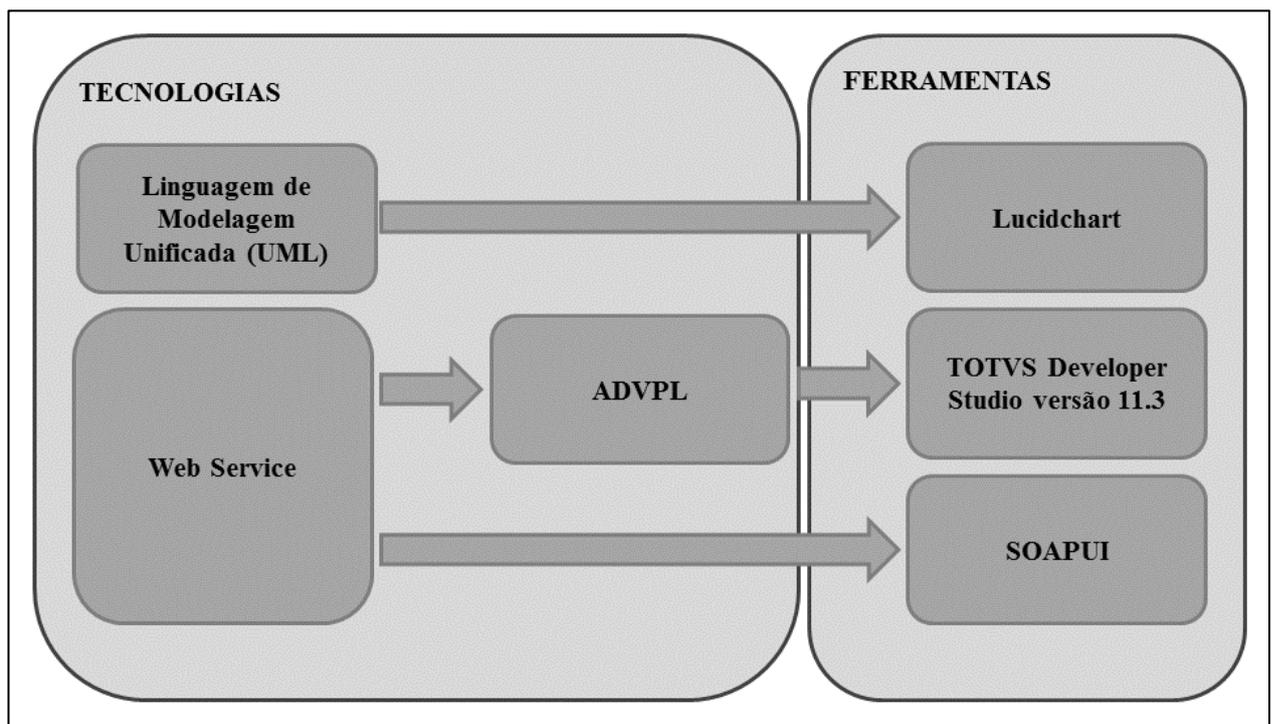
1.2 OBJETIVOS ESPECIFICOS

- Identificar tecnologias e ferramentas;
- Compreender problema;
- Documentar os requisitos;
- Relatar o desenvolvimento;
- Publicar site.

2 TECNOLOGIAS UTILIZADAS

Primeiramente, serão abordados os conceitos relacionados as tecnologias utilizadas no processo de levantamento de requisitos para criação do portal web. Conceitos de linguagens como UML, Advpl, ferramentas utilizadas na criação de diagramas e IDE para desenvolvimento do Webservice serão introduzidas nesse projeto.

Figura 1: Tecnologias e Ferramentas Utilizadas



Fonte: Próprio autor, 2017.

Posteriormente, serão apresentadas as ferramentas que são os meios de utilizam dessas tecnologias, conforme esquema lógico representado na Figura 1.

2.1 LINGUAGEM DE MODELAGEM UNIFICADA (UML)

UML (Linguagem de Modelagem Unificada) é uma linguagem visual destinada a documentação do software, utilizada para modelar requisitos e artefatos de um sistema. A UML segue padrões definidos que nos ajudam arquitetar projetos para criação de softwares, incluindo funções do sistema, regra de negócio, classes

concretas, esquema de banco de dados e programas reutilizáveis.

A modelagem de um software implica em criar modelos de software, [...]Um modelo de software captura uma visão de um sistema físico, é uma abstração do sistema com um certo propósito, como descrever aspectos estruturais ou comportamentais do software. (GUEDES, 2011, p.21).

Com a UML construímos modelos que nos ajudam a visualizar o sistema como ele é ou como gostaríamos que fosse. Os modelos ou diagramas criados, a partir da linguagem, permitem especificar comportamentos do sistema e disponibilizam um passo-a-passo para construção do mesmo, além de documentar todas as tomadas de decisões do projeto.

2.1.1 Diagrama de Caso de Uso

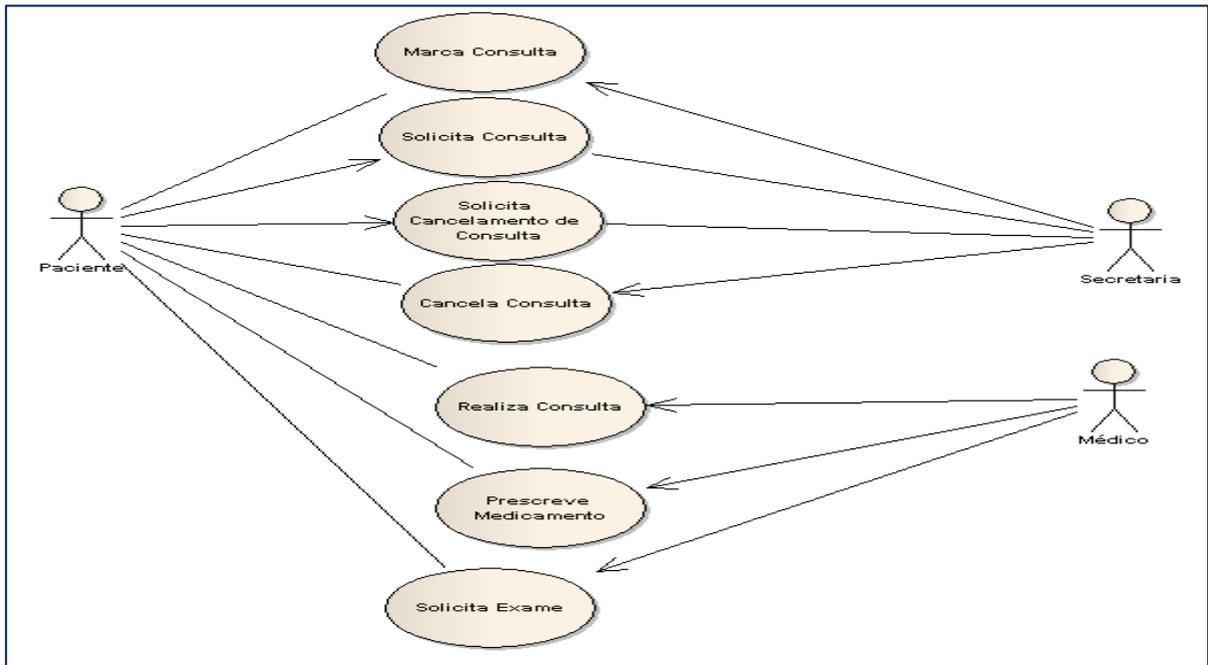
A UML disponibiliza diversos diagramas para modelagem do sistema. Nesse projeto utilizamos o diagrama de caso de uso, que é um dos primeiros e mais utilizados diagramas na fase de levantamento e análises de requisitos do sistema.

Segundo GUEDES, o diagrama de caso de uso

Apresenta uma linguagem simples e de fácil compreensão para que os usuários possam ter uma ideia geral de como o sistema irá se comportar. Procura identificar os autores (usuário, outros sistemas ou até mesmo algum hardware especial) que utilizarão de alguma forma o software, bem como os serviços, ou seja, as funcionalidades que o sistema disponibilizara aos atores, conhecidas nesse diagrama como caso. (GUEDES, 2011, p.30 e 31).

O diagrama de caso de uso a seguir, representa visualmente a interação entre o sistema, seus usuários e as funcionalidades.

Figura 2 - Diagrama de Caso de Uso

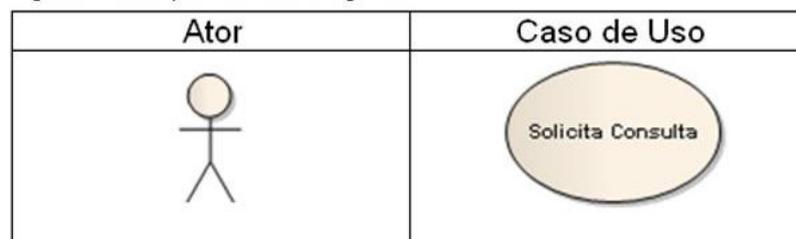


Fonte: Leandro Ribeiro, 2017.

Conforme Figura 2, diagrama de caso de uso é formado por basicamente 4 componentes:

- Cenário: Atividades de interação usuário e sistema
- Ator: Tipo de usuário do sistema;
- Caso de uso: Funcionalidade ou tarefa desempenhada pelo ator;
- Relacionamento: Comunicação entre os itens.

Figura 3 – Componentes do Diagrama



Fonte: Leandro Ribeiro, 2017.

Existem três tipos de relacionamentos muito importantes na modelagem orientada ao objeto: dependências, generalizações e associações.

- Dependência: relacionamento de dependência o próprio nome já diz, por exemplo: quando um determinado caso de uso X for executado, um outro caso

- de uso Y deverá ser executado também, ou seja, um depende do outro;
- Associação: relacionamento de associação ocorre quando: caso de uso X é acionado e o caso de uso Y pode ser ou não acionado;
 - Generalização: relacionamento de generalização é conhecido como subclasse/superclasse, ou seja, classes generalizadas conectam a outras mais especializadas.

Na UML, os modos pelos quais os itens podem estar conectados a outros, isto é, logicamente ou fisicamente, são modelados como relacionamentos. Na modelagem orientada a objetos, existem três tipos de relacionamento que são os mais importantes: dependências, generalizações e associações. (BOOCH, RUMBAUGH, JACOBSON, 2012, p.69).

Posteriormente, no Capítulo 3, serão aplicados os conceitos apresentados ao problema da empresa distribuidora de medicamento, a partir da ferramenta Ludichart, apresentada na seção 2.3.2. Possibilitando assim a compreensão visual da necessidade.

2.2 WEBSERVICE

É uma tecnologia utilizada para integração entre sistemas e aplicações diferentes, independente de linguagem de programação, ou seja, Webservice é uma solução que realiza a integração entre diferentes aplicações. Os principais requisitos do Webservice são:

- Comunicação padrão via protocolo de internet;
- Comunicação entre sistemas distribuídos;
- Comunicação padrão XML.

A tecnologia Webservice fica responsável por buscar a informação em outro sistema e retornar para o solicitante em formato XML, ou seja, em nenhum momento o solicitante saberá qual a plataforma ou qual a linguagem de programação está sendo utilizada no outro sistema.

Webservice é uma tecnologia baseada em XML e HTTP cuja a principal função é disponibilizar serviços interativos na WEB que podem ser acessados por qualquer outra aplicação independente da linguagem ou plataforma em que foi construída. (CAETANO, 2016).

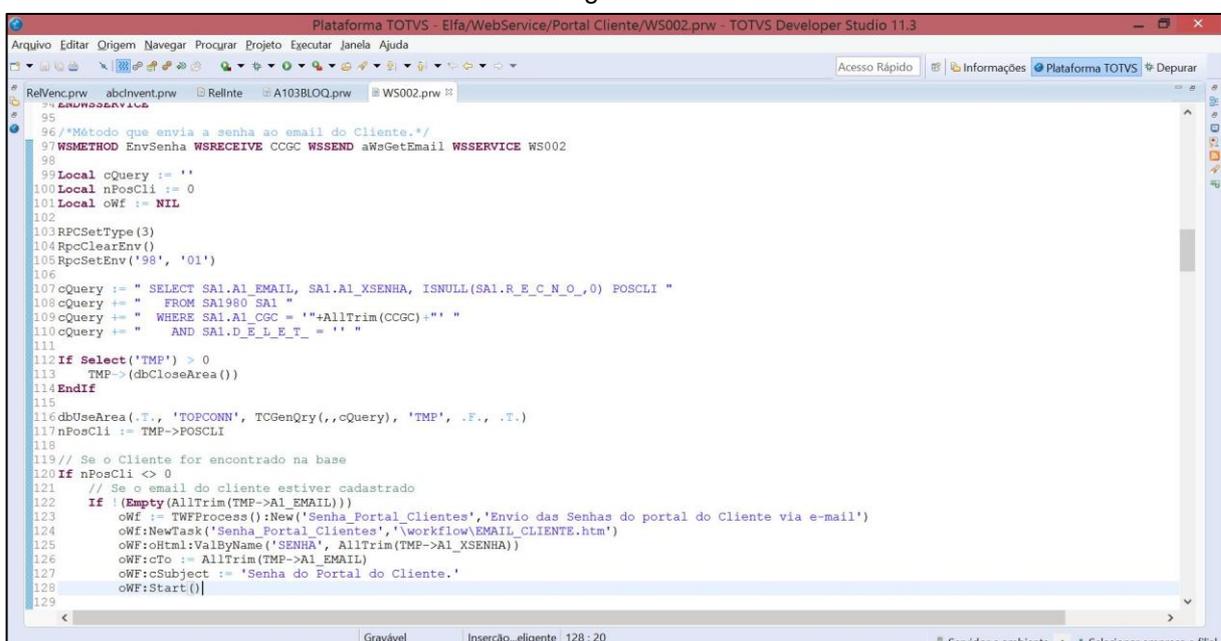
Para o presente trabalho, a utilização desse tipo de integração se faz essencial, tendo em vista que será necessário realizar a comunicação entre o sistema gerencial da empresa, utilizado majoritariamente pelos funcionários, e portal web, que será desenvolvido com foco os clientes da mesma. Nas seções seguintes, serão apresentadas tanto a linguagem utilizada para incremento da tecnologia, quanto as ferramentas meio para isso.

2.2.1 AdvPL

Para desenvolvimento do Webservice, foi utilizado a linguagem de programação AdvPL (Advanced Protheus Language), linguagem padrão de gestão empresarial ERP (*Enterprise Resource Planning*) do grupo TOTVS.

Desenvolvida em 1994, o AdvPL utiliza o padrão xBase (Clipper, Visual Objects e depois Fivewin), com funções e comandos, operadores, estruturas de controle de fluxo e palavras reservadas. Também permite o desenvolvimento de programas orientado a objetos ou procedural (SINKIKO, 2016).

Figura 4 – AdvPL



```

Plataforma TOTVS - Elfa/WebService/Portal Cliente/WS002.prw - TOTVS Developer Studio 11.3
Arquivo Editar Origem Navegar Procurar Projeto Executar Janela Ajuda
WS002.prw
95
96 /**Método que envia a senha ao email do Cliente.**/
97 WSMETHOD EnvSenha WSRECEIVE CCGC WSEND aWsGetEmail WSSERVICE WS002
98
99 Local cQuery := ''
100 Local nPosCli := 0
101 Local oWf := NIL
102
103 RPCSetType(3)
104 RpcClearEnv()
105 RpcSetEnv('98', '01')
106
107 cQuery := " SELECT SA1.A1_EMAIL, SA1.A1_XSENHA, ISNULL(SA1.R_E_C_N_O_,0) POSCLI "
108 cQuery += " FROM SA1980 SA1 "
109 cQuery += " WHERE SA1.A1_CGC = '"+AllTrim(CCGC)+"' "
110 cQuery += " AND SA1.D_E_L_E_T = '' "
111
112 If Select('TMP') > 0
113   TMP->(dbCloseArea())
114 EndIf
115
116 dbUseArea(.T., 'TOPCONN', TCGenQry(,cQuery), 'TMP', .F., .T.)
117 nPosCli := TMP->POSCLI
118
119 // Se o Cliente for encontrado na base
120 If nPosCli <> 0
121   // Se o email do cliente estiver cadastrado
122   If !(Empty(AllTrim(TMP->A1_EMAIL)))
123     oWf := TWfProcess():New('Senha_Portal_Clientes','Envio das Senhas do portal do Cliente via e-mail')
124     oWf:NewTask('Senha_Portal_Clientes','\workflow\EMAIL_CLIENTE.htm')
125     oWf:oHtml:ValByName('SENHA', AllTrim(TMP->A1_XSENHA))
126     oWf:cTo := AllTrim(TMP->A1_EMAIL)
127     oWf:cSubject := 'Senha do Portal do Cliente.'
128     oWf:Start()
129

```

Fonte: Próprio autor, 2017.

Na Figura 4, pode ser observado o ambiente, contendo trechos utilizando sintaxe AdvPL. A definição para ele é sintetizada, conforme afirmação a seguir:

O compilador e o interpretador da linguagem AdvPL é o próprio servidor de aplicação (Application Server), e existe um ambiente visual para desenvolvimento integrado (Development Studio) onde o código-fonte pode ser criado, compilado e depurado. (SINKIKO, 2016).

Na seção 2.3.2 serão esclarecidas as informações referentes a ferramenta citada que é utilizada como meio para desenvolvimento baseada na tecnologia AdvPL.

2.3 FERRAMENTAS UTILIZADAS

Nessa seção serão abordadas as ferramentas utilizadas para modelagem do diagrama de caso de uso e desenvolvimento do Webservice.

2.3.1 Lucidchart

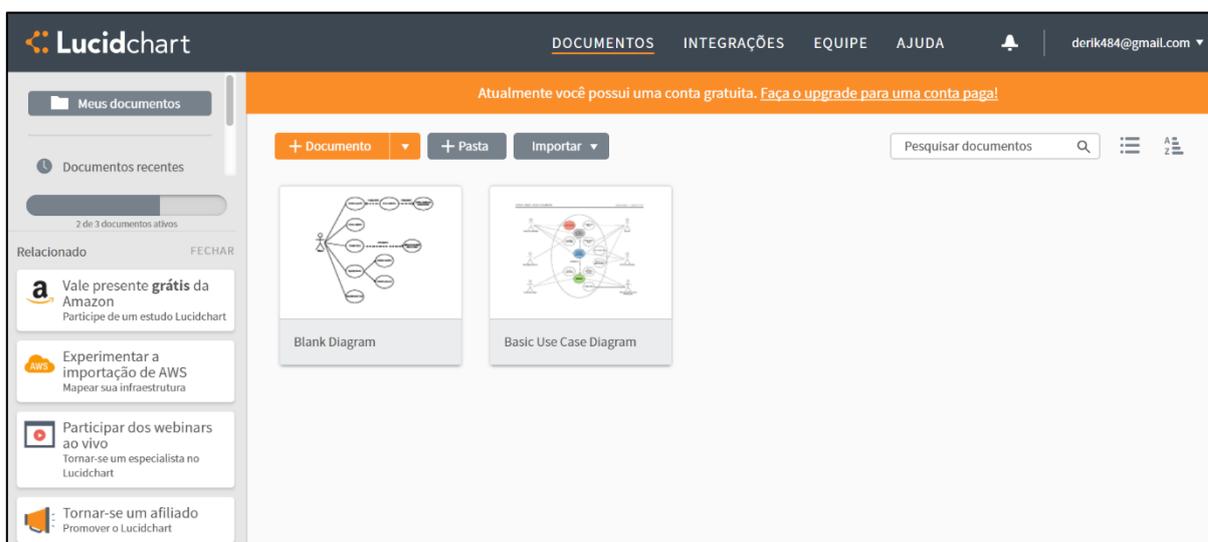
Lucidchart é uma ferramenta completa desenvolvida com HTML5 para criação de diagramas de forma simples e intuitiva. A ferramenta pode ser utilizada de forma online ou offline, com acessos colaborativos de diferentes membros da equipe. Além de conter funcionalidades importantes, como arrastar e soltar, a ferramenta possui diversos modelos de diagramas, como:

- UML
- Diagramas de rede
- Fluxogramas
- Modelos de processos de negócios
- Mapas mentais
- Mapas de sites
- Diagramas ER

A ferramenta foi escolhida pela simplicidade do uso e pelo fácil acesso, por ser uma ferramenta online e conter integração com o Google Drive. O Lucidchart é

encontrado através do endereço de URL www.lucidchart.com.br.

Figura 5 – LucidChart



Fonte: Próprio autor, 2017.

A Figura 5 mostra a tela inicial da ferramenta LucidChart e alguns diagramas que já estão em desenvolvimento.

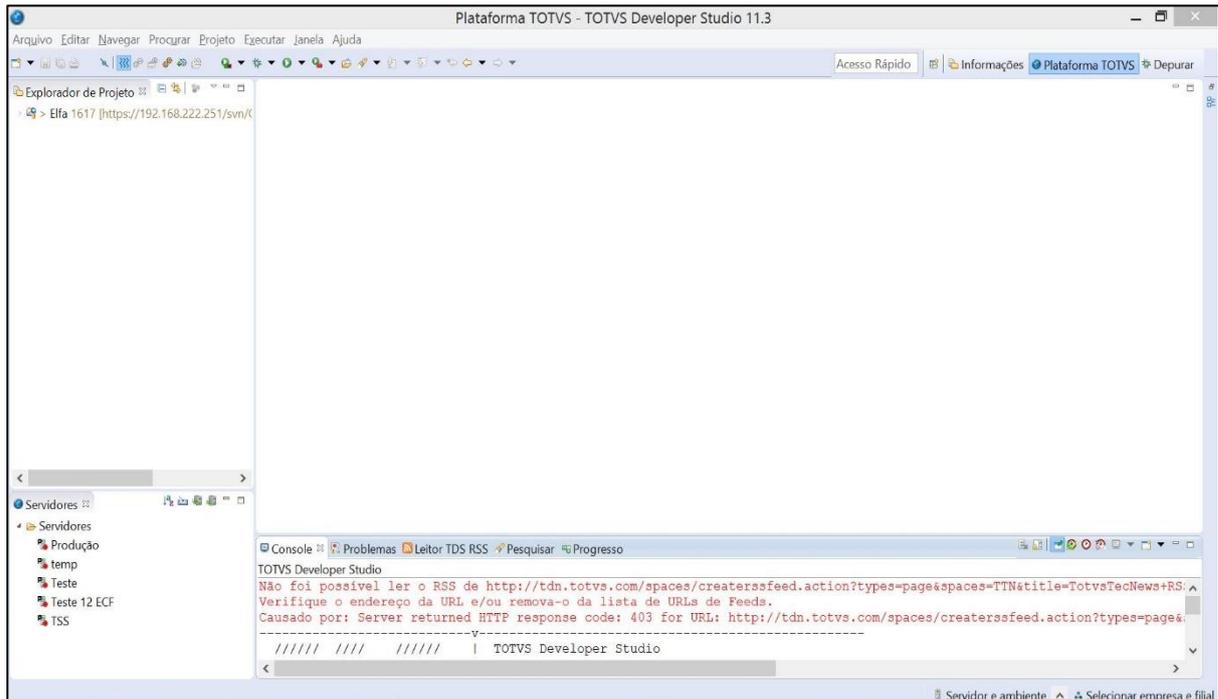
2.3.2 Totvs Developer Studio 11.3

No desenvolvimento do Webservice para integração dos sistemas, foi utilizado a ferramenta padrão da própria software house, o TOTVS Developer Studio versão 11.3. O programa foi construído pela própria TOTVS sob a plataforma Eclipse Luna, ferramenta utilizada para desenvolvimento em Java ou outras linguagens suportadas por ele.

O TOTVS Developer Studio é uma ferramenta de desenvolvimento integrada (IDE), fornecendo ao desenvolvedor ou administrador TOTVS uma vasta gama de funcionalidades, tais como edição de programas, execução, depuração, administração de produtos TOTVS e muitas outras funcionalidades. (WATANABE, 2016)

Essa variedade de funções, torna a ferramenta ideal para utilização da linguagem AdvPL na construção de programas e integrações.

Figura 6 – TOTVS Developer Studio



Fonte: Próprio autor, 2017.

A Figura 6, apresenta a ferramenta. Nela podemos aplicar pacotes de correção, visualizar datas de fontes e configurar conexões com o servidor de aplicação.

2.3.3 SoapUI

É uma ferramenta open source desenvolvida em linguagem de programação Java, onde sua função é consumir e testar Webservice s SOAP e REST.

Uma das ferramentas de teste de API mais confiável no mercado, a SOAPUI tem mais de 10 anos de experiência com uma grande rede comunitária sobre códigos abertos.

A SoapUI oferece testes funcionais do SOAP Webservices, testes funcionais de API REST, cobertura WSDL, testes de afirmação de mensagens e refatoração de testes (DEV MEDIA).

Figura 7 – SoapUI



Fonte: Próprio autor, 2017.

Os principais motivos para utilização da ferramenta foram a liberdade de uso, devido a natureza open source e também a necessidade de alta qualidade da integração, que para tal tem que experimentada durante o todo o processo de desenvolvimento do portal.

3 LEVANTAMENTO DE REQUISITOS

Conforme entrevista com pessoas da área de cobrança, setor responsável pelo atendimento da demanda, atualmente o principal motivo que levam os clientes entrarem em contato, seja no setor de cobranças ou com a SAC (Serviço de Atendimento ao Cliente) são reenvio de boletos e alteração no prazo de pagamento. Dados apontam que 85% das solicitações feitas ao SAC no mês de maio/17 para o setor financeiro, foram de reenvio de boleto e alteração no prazo de vencimento.

Quando é emitido uma Nota Fiscal para o cliente, além de ser enviado uma via física de cada documento com a mercadoria, o sistema ERP manda um e-mail com a nota fiscal, xml e o boleto, isso se o e-mail estiver preenchido corretamente no cadastro do cliente. Os motivos que levam esses clientes entrarem em contato são , principalmente, extravio interno do documento, ou seja, o documento é entregue ao cliente. Isso ocorre porque, por algum motivo, não chega ao setor responsável ou o cadastro do cliente não está com o e-mail atualizado.

Diante dessa situação, foi proposta a criação de um portal web com uma interface administrativa que irá permitir aos clientes tirar segunda via de boletos e imprimir suas notas fiscais.

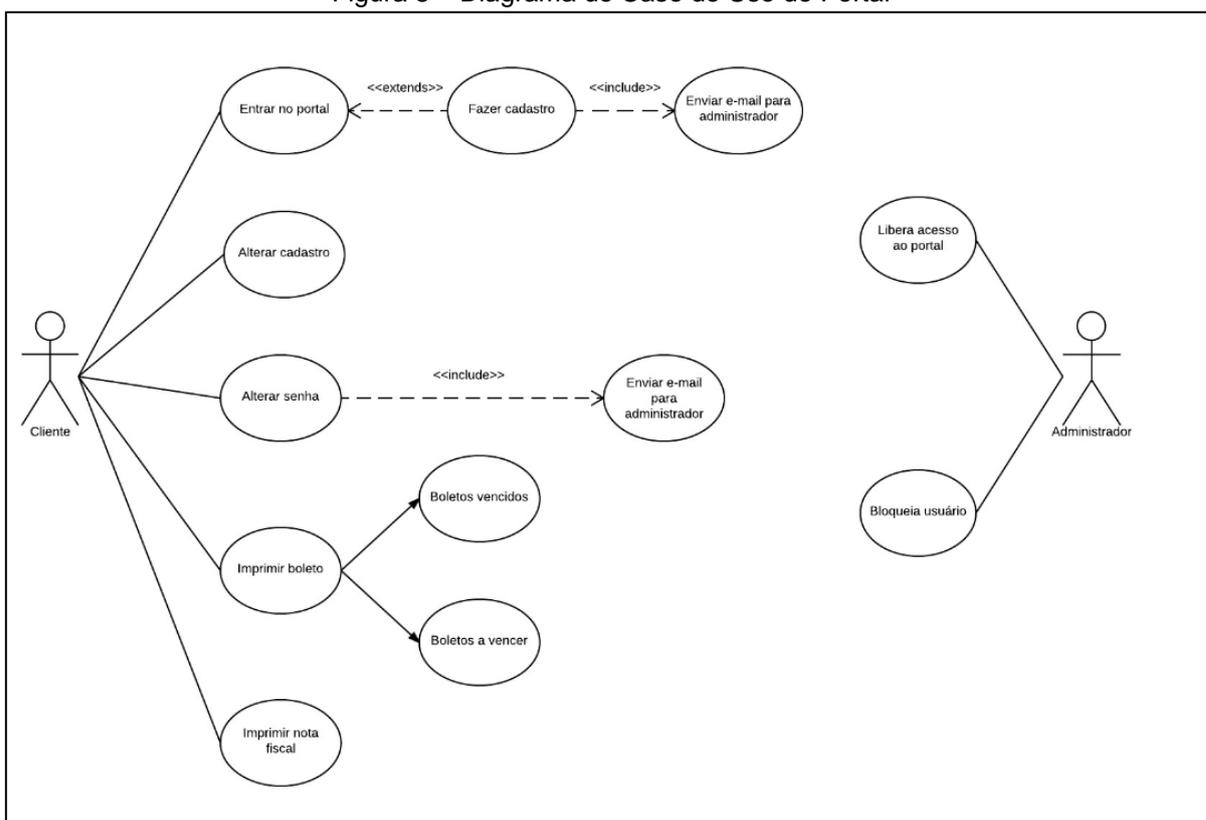
3.1 DIAGRAMA E ESPECIFICAÇÕES

Os diagramas de caso de uso especificam os sistemas de uma forma abstrata, sem muitos detalhes. Devem ser descritos em linguagem natural e estruturada, para que facilite a compreensão das funcionalidades do sistema.

Quando você modela as interações de um sistema com seu ambiente, deve usar uma abordagem abstrata sem muitos detalhes. Uma maneira de fazer isso é usar um modelo de caso de uso. [...]cada caso de uso representa uma interação com o sistema. (SOMMERVILLE, 2011, p.126).

Tendo em vista essas premissas do diagrama, os conceitos expostos na seção 2.1 e os resultados da entrevista anteriormente descrita, fora construído no Ludichart a modelagem para a aplicação.

Figura 8 – Diagrama de Caso de Uso do Portal



Fonte: Próprio autor, 2017.

O diagrama demonstrado na Figura 8 apresenta de maneira geral todas as funcionalidades de portal web, entre elas estão:

- Criar conta;
- Efetuar Login;
- Alterar cadastro;
- Alterar senha;
- Imprimir boletos;
- Imprimir notas fiscais;
- Liberar acesso ao portal;
- Bloquear acesso ao portal.

3.2 REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS

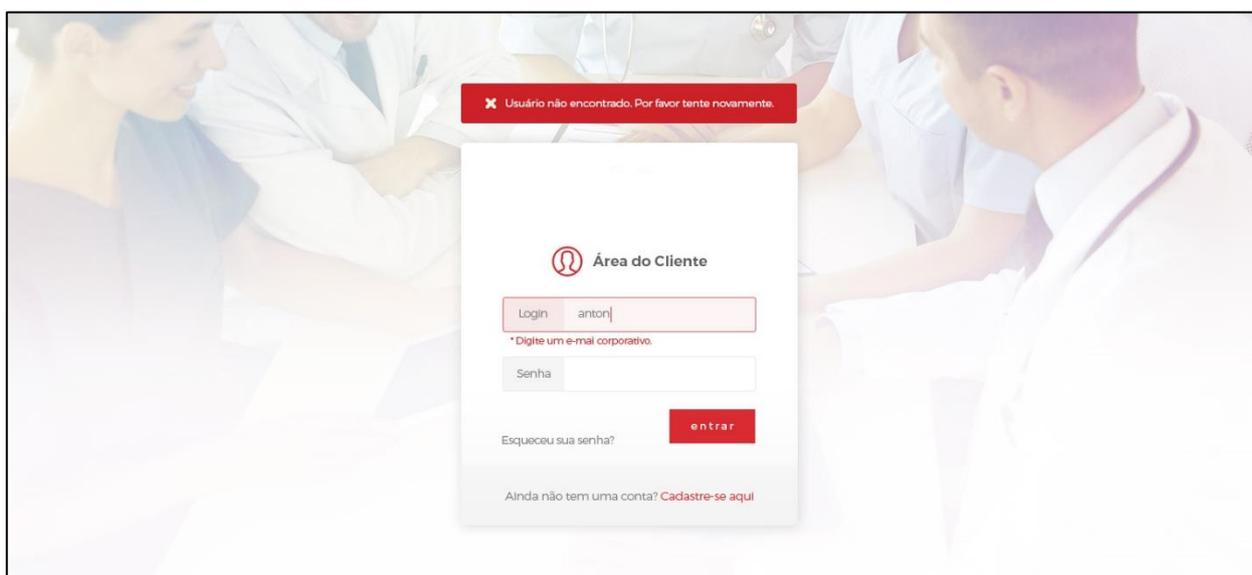
Após a compreensão do problema, entramos na etapa de levantamento de requisitos. Uma das etapas mais importantes de um projeto, pois qualquer análise malfeita poderá comprometer o projeto de diversas formas, como, atraso na entrega,

falhas e outros problemas que afetam a qualidade do produto final. Portanto, nesse tópico listamos todos os requisitos funcionais e não funcionais que o portal deverá oferecer para o cliente.

3.2.1 Requisitos Funcionais

Primeiramente, buscou-se identificar os processos e atividades que deverão ser realizadas pelos usuários. Desse modo, tendo em vista os requisitos funcionais “são declarações de funções que o sistema deve fornecer, como o sistema deve reagir a entradas específicas e como deve se comportar em determinadas situações” (SOMMERVILLE, 2003, p.82), fora concluído que o portal irá oferecer as seguintes funcionalidades para o cliente:

Figura 9 – Tela de Login



Fonte: Próprio autor, 2017.

- a) LOGIN: a tela inicial do portal será de autenticação com o login e senha, Figura 9. No campo login terá uma validação que irá verificar se o usuário existe, caso contrário, o sistema deverá exibir uma mensagem negativa;

Figura 10 – Tela Criação de Conta

Usuário não encontrado. Por favor tente novamente.

Cadastre-se.

Seja um parceiro do Grupo

1 Dados Pessoais

Nome anton

Cargo

E-mail

2 Dados Empresariais

CNPJ

Razão Social

Telefone

Caso seja responsável por outros CNPJs, gentileza os informe aqui.

Já possui cadastro? [Faça o login aqui.](#)

Enviar solicitação

Fonte: Próprio autor, 2017.

- b) **CRIAR CONTA:** na tela inicial do portal deverá ter um link para cadastro, ao clicar no link o usuário deverá ser direcionado para pagina de cadastro, onde será exibido um formulário com os campos: Login, senha, nome, cargo, e-mail, razão social, telefone e mensagem, Figura 10. No campo e-mail terá um filtro para não permitir cadastro de e-mails comuns. Por exemplo, @gmail, @hotmail e etc. Quando o formulário for submetido, as informações serão enviadas para o e-mail da equipe administrativa da empresa, que fará a validação do cadastro;
- c) **CADASTRO CONCLUÍDO:** Quando um cliente concluir seu cadastro com sucesso, o mesmo será redirecionado para uma nova tela, informando que sua solicitação está sendo analisada e, em breve, receberá um e-mail notificando se a conta foi ou não aprovada;

Figura 11 – Tela Cadastro Aguardando Aprovação



Cadastro aguardando aprovação

Obrigada pelo interesse em ser nosso parceiro.
Seu cadastro foi recebido por nossa equipe e está sendo analisado.

Em até 72h enviaremos outro e-mail sobre a aprovação de sua conta.

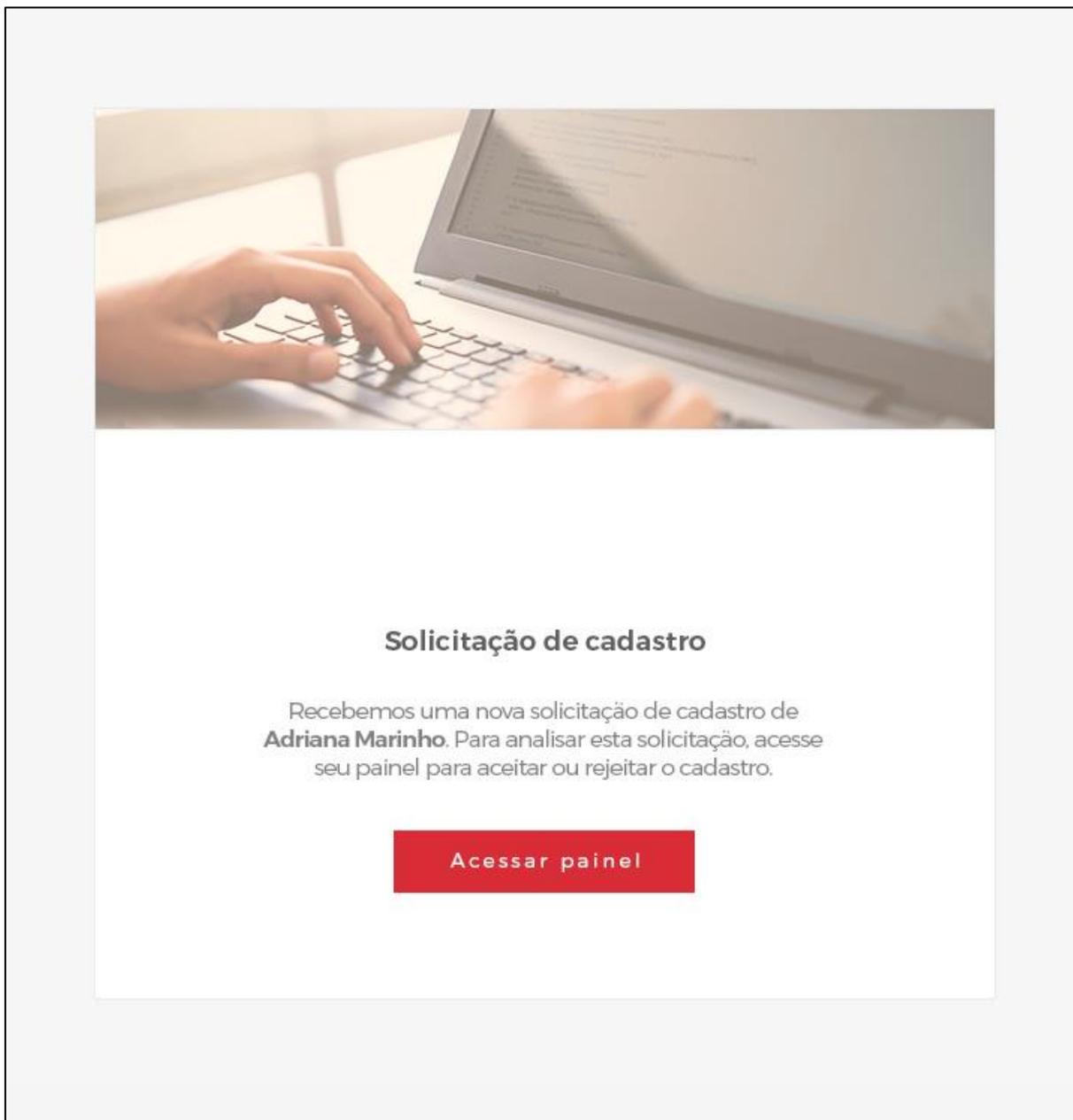
Se você não se cadastrou no Grupo elfa, entre em contato conosco através do e-mail elfa@elfa.com.br

Fonte: Próprio autor, 2017.

- d) E-MAIL AGUARDANDO APROVAÇÃO: enviar um e-mail explicando que sua solicitação está em análise e logo será notificado se a conta foi ou não aprovada, Figura 11;
- e) E-MAIL CADASTRO APROVADO/REJEITADO: se o cadastro for aprovado, o sistema deve enviar um e-mail de boas-vindas para o cliente, com um link onde

será direcionado para o portal, caso seja rejeitado, deverá enviar um e-mail com o motivo da rejeição;

Figura 12 – Tela Solicitação de Cadastro

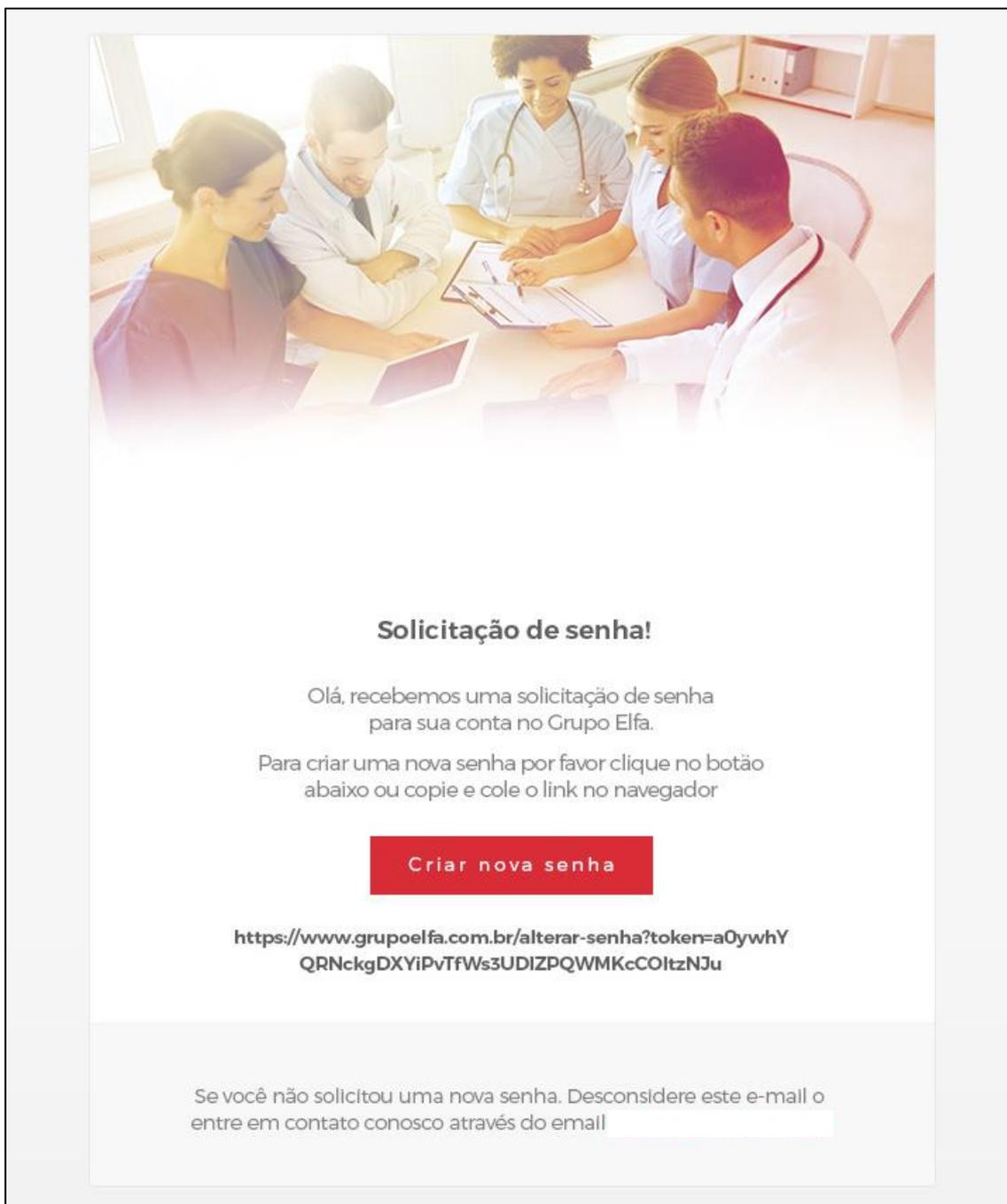


Fonte: Próprio autor, 2017.

- f) E-MAIL SOLICITAÇÃO DE CADASTRO: o sistema deve enviar um e-mail para os administradores do grupo, Figura 12, com um link para o painel do administrador, onde será possível autorizar ou rejeitar a solicitação;

- g) RECUPERAR SENHA: na tela inicial terá um link para recuperar senha que pode ser observado na Figura 9, ao clicar no link, o usuário será direcionado para uma nova página com um campo para digitar o e-mail de recuperação. Nesse campo terá uma validação do endereço informado, ou seja, irá verificar se o e-mail existe no sistema antes de submeter a solicitação;

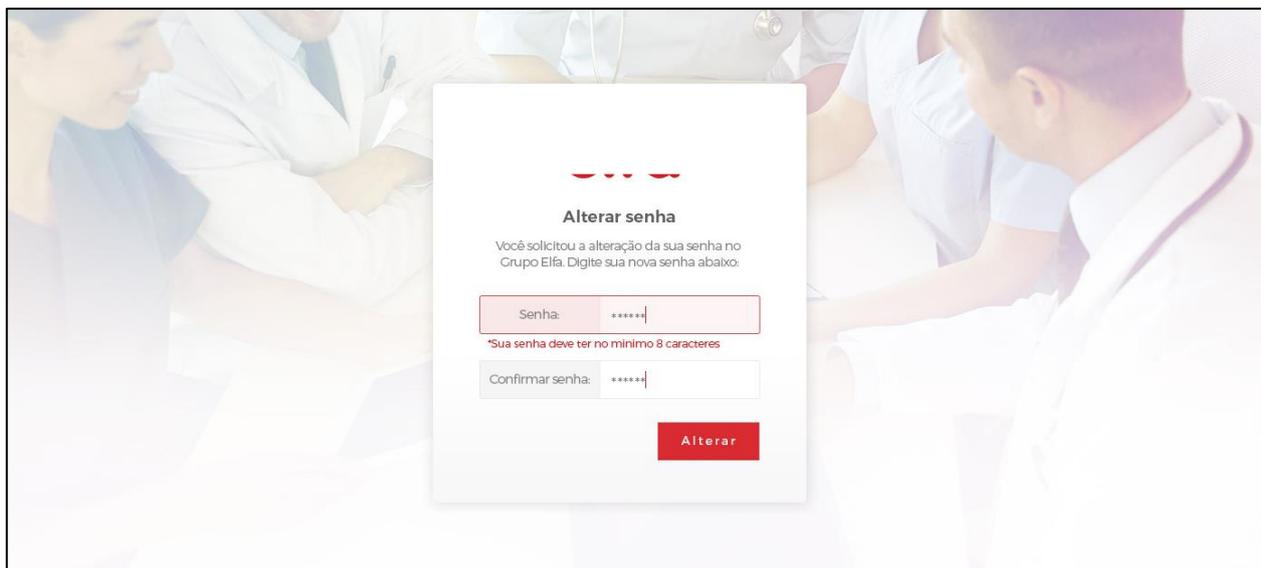
Figura 13 – Tela Recuperação de Senha



Fonte: Próprio autor, 2017.

- h) E-MAIL RECUPERAR SENHA: será enviado para o cliente, com um texto informando que uma nova senha foi solicitada e um link para cadastrar nova senha, Figura 13;

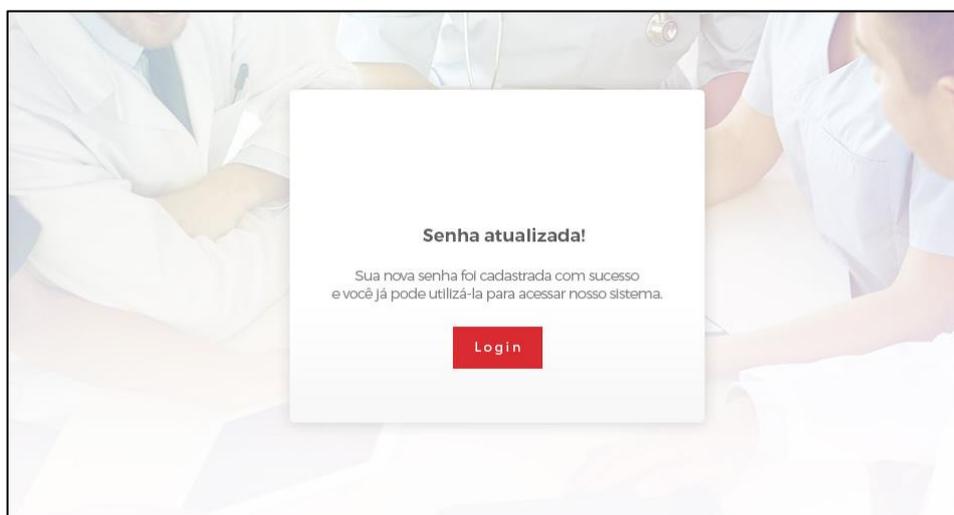
Figura 14 – Tela Cadastro de Senha

A screenshot of a web application interface for password change. The background shows a group of people in white lab coats. Overlaid on this is a white modal box with a red logo at the top. The text inside the modal reads: "Alterar senha", "Você solicitou a alteração da sua senha no Grupo Elfa. Digite sua nova senha abaixo:", "Senha: *****", "*Sua senha deve ter no mínimo 8 caracteres", "Confirmar senha: *****", and a red "Alterar" button at the bottom.

Fonte: Próprio autor, 2017.

- i) CADASTRAR NOVA SENHA: será uma tela com campos de senha e confirmação da senha, onde deverá ter no mínimo 8 caracteres alfa numéricos, conforme Figura 14;

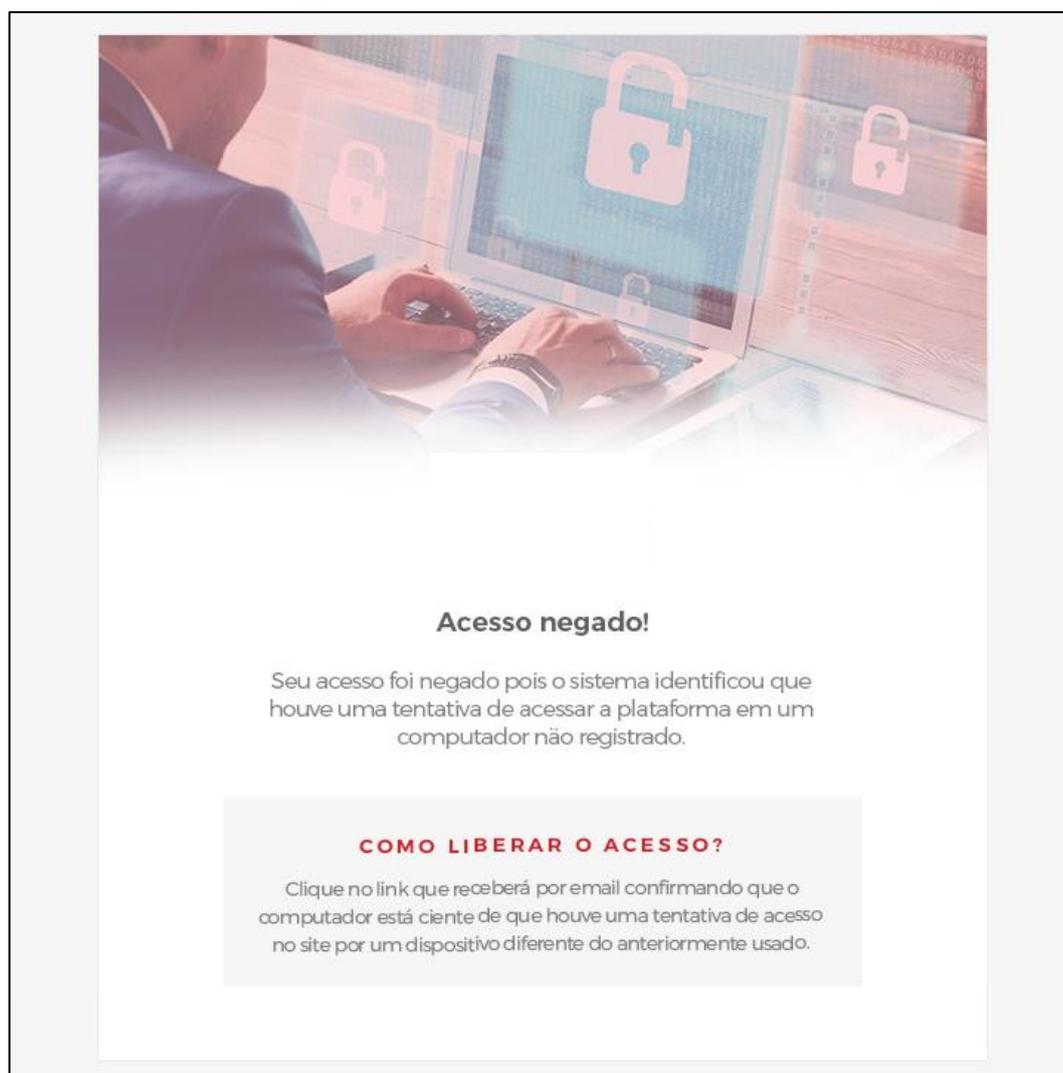
Figura 15 – Tela de Senha Cadastrada com Sucesso

A screenshot of a web application interface showing a success message. The background is the same group of people in white lab coats. Overlaid is a white modal box with a red logo at the top. The text inside the modal reads: "Senha atualizada!", "Sua nova senha foi cadastrada com sucesso e você já pode utilizá-la para acessar nosso sistema.", and a red "Login" button at the bottom.

Fonte: Próprio autor, 2017.

- j) **SENHA CRIADA COM SUCESSO:** após submeter a nova senha, o cliente será direcionado para uma nova tela com um botão para login e uma mensagem informando que a nova senha foi cadastrada com sucesso, que corresponde a Figura 15;
- k) **RESTRIÇÃO DE ACESSO AO SISTEMA:** quando o cliente fizer login pela primeira vez, um cookie será armazenado em seu computador que irá conter todas as informações sobre aquela máquina. Se no próximo acesso o sistema identificar que o cliente está acessando o site de outro computador, então o acesso será negado;

Figura 16 – Tela de Acesso Negado



Fonte: Próprio autor, 2017.

- l) **ACESSO NEGADO:** o sistema apresentará uma tela informando que seu acesso foi negado, como Figura 16, pois o sistema identificou que o mesmo tentou acessar a plataforma de outro computador. Para liberar o acesso, o cliente deverá entrar em contato com a empresa por e-mail ou telefone;
- m) **E-MAIL ACESSO NEGADO:** no momento do acesso negado, o sistema deverá enviar um e-mail para os administradores com as informações do cliente;

Figura 17 – Tela de Títulos

Olá, José Alberto
Sair

Meus dados **Títulos fiscais**

Empresa: TODOS

DATA DE VENCIMENTO:
De: 20/08/2017 Até: 30/08/2017 Situação: TODOS

Estabelecimento: TODOS Documento: 3467

Empresa	Estabelecimento	Emissão DOC	Tipo	Prefixo	Documento	Parcela	Valor	Saldo	Vencimento	Situação	2ª VIA	2ª VIA NF
Hospital A	Elifa BSB	29/09/17	NF	5	000011311	01	R\$10.000,00	R\$10.000,00	29/10/2017	Vencido		
Hospital A	Elifa BSB	29/09/17	NF	5	000011311	01	R\$10.000,00	R\$10.000,00	29/10/2017	Vencido		
Hospital A	Elifa BSB	29/09/17	NF	5	000011311	01	R\$10.000,00	R\$10.000,00	29/10/2017	Vencido		
Hospital A	Elifa BSB	29/09/17	NF	5	000011311	01	R\$10.000,00	R\$10.000,00	29/10/2017	Vencido		

Grupo - 2017

Fonte: Próprio autor, 2017.

- n) **TELA PRINCIPAL CLIENTE:** nessa tela terá duas abas, uma para acesso aos dados cadastrais (Meus dados) e outra para busca de títulos fiscais.
- **Títulos Fiscais:** essa será a tela principal do sistema, onde o cliente encontrará todos seus títulos vencidos ou a vencer, com opções de filtro por data de vencimento, número do documento e situação do título. Haverá uma tabela listando todos os títulos, conforme Figura 17, onde os campos serão: empresa, estabelecimento, data de emissão, tipo do documento, prefixo, número do documento, parcela, valor, saldo, data de vencimento, situação, botão de segunda via do boleto e botão de segunda via da nota fiscal;

- Meus dados: será uma tela com as informações preenchidas no cadastro, onde será possível alterar alguns campos, como, nome, cargo e e-mail.

Todas as figuras incluídas nessa seção correspondem ao que foi desenvolvido em realidade para o Portal. Vale ressaltar que foram incluídas as consideradas mais expressivas para o atendimento da necessidade da distribuidora.

3.2.2 Requisitos Não Funcionais

Além das funcionalidades, faz-se extremamente importante entender as necessidades que não são fundamentalmente respostas às ações e solicitações do usuário. Todavia, interferem na percepção da qualidade da experiência dele ao utilizar a aplicação.

Dessa forma, sabendo que não funcionais “são requisitos que não estão diretamente relacionados com os serviços específicos oferecidos pelo sistema a seus usuários. Requisitos não funcionais são frequentemente mais críticos que requisitos funcionais” (SOMMERVILLE, 2011, p.60), o sistema deverá atender aos seguintes pontos:

- a) **SEGURANÇA:** o portal deverá conter um sistema de segurança que impeça o acesso indevido as informações. Deverá conter um mecanismo que permite criptografia de dados para autenticação do usuário;
- b) **DISPONIBILIDADE:** o portal deverá estar disponível de segunda à sexta a partir das 07:00 às 21:00, e sábado e domingo das 07:00 às 18:00;
- c) **USABILIDADE:** o portal deverá rodar em uma plataforma web com um design responsivo, que se adeque a qualquer dispositivo utilizado. Com uma interface de fácil entendimento e uso da ferramenta.

Ao realizar tais levantamentos e somados os diagramas, pode-se partir de forma mais segura para o desenvolvimento da solução. Isso porque a tirou do âmbito das ideias, que pode ter diferentes interpretações para cada envolvido, e as tornou tangíveis, documentáveis e visuais.

4 DESENVOLVIMENTO

Logo depois da conclusão compreensão do problema e o desenho da solução, iniciou-se a etapa de desenvolvimento do sistema. Esta contou tanto com os recursos desenvolvimento da área de Tecnologia da Informação e Marketing da distribuidora, quanto com desenvolvedores e designs de um fornecedor da área de mídia e tecnologia. Com isso, foram planejadas etapas desenvolvimento, testes e homologação.

Quadro 1 – Macro Etapas do Desenvolvimento do Portal

Atividade	Responsável
Desenvolvimento do layout e front end	Fornecedor
Aprovação do layout	Marketing da Distribuidora
Desenvolvimento dos Web Services	TI da Distribuidora
Elaboração dos planos de teste	TI da Distribuidora
Testes com os envolvidos	TI da Distribuidora
Homologação	TI da Distribuidora
Implantação	TI da Distribuidora e Fornecedor
Acompanhamento pós-implantação	TI da Distribuidora e Fornecedor

Fonte: Próprio autor, 2017.

Vale ressaltar que o descrito no Quadro 1 são as macro etapas que foram desdobradas em entregáveis menores e acompanhadas por meio de reuniões quinzenais. Na fase de construção dos Webservices, por exemplo, foram construídos cinco métodos para atendimento dos requisitos do portal. Sendo eles:

- 1) Verificar CPNJ: o método receberá o CNPJ do cliente e verificará a existência do mesmo em nossa base, caso contrário o retorno será falso e exibirá uma mensagem de cliente não cadastrado.

Figura 18 – Método verifica CNPJ

```

74
75 WSMETHOD CheckCNPJ WSRECEIVE cCNPJ WSEND sCheckCNPJ WSSERVICE WSPORTCL
76
77 Local lRet := .F.
78 Local cTab := ""
79 Local cQuery
80
81 RpcClearEnv()
82 RpcSetType( 3 )
83 RpcSetEnv( '98', '01' )
84
85 cTab := GetNextAlias()
86
87 cQuery := " SELECT AI_NOME "
88 cQuery += " FROM " + RetSqlName("SA1")
89 cQuery += " WHERE AI_CGC = ' " + cCNPJ + "' AND "
90 cQuery += " AI_MSBLQL <> '1' AND "
91 cQuery += " D_E_L_E_T <> '*' "
92 dbUseArea(.T., "TOPCONN", TCGenQry(,cQuery), cTab, .F., .T.)
93
94 If (cTab)->(!Eof())
95 lRet := .T.
96 ::sCheckCNPJ:bFind := lRet
97 ::sCheckCNPJ:cRazaoSocial := (cTab)->AI_NOME
98 Else
99 ::sCheckCNPJ:bFind := lRet
100 ::sCheckCNPJ:cRazaoSocial := "CNPJ não existente na base cadastral !"
101 Endif
102
103 (cTab)->(DbCloseArea())
104 Return .T.
105
106 WSMETHOD SetIDCliente WSRECEIVE nIDPortal,cCNPJ WSEND sSetUser WSSERVICE WSPORTCL
107
108 Local lRet := .T.

```

Fonte: Próprio autor, 2017.

A figura 18 foi o primeiro método criado, serve para consultar se existe o CNPJ do cliente em nossa base.

- 2) Cadastrar ID no cliente: o método receberá o ID do cliente e gravará no CNPJ, se já existir o ID cadastrado para o CNPJ desejado, o WS retornará uma mensagem de ID já cadastrado.

Figura 19 – Método cadastrar ID

```

106 WSMETHOD SetIDCliente WSRECEIVE nIDPortal,cCNPJ WSEND sSetUser WSSERVICE WSPORTCL
107
108 Local lRet := .T.
109 Local cTab := ""
110 Local cQuery
111 Local aTmp := {}
112
113 RpcClearEnv()
114 RpcSetType( 3 )
115 RpcSetEnv( '98', '01' )
116
117 If cValToChar(::nIDPortal) == "0"
118 lRet := .F.
119 ::sSetUser:bRec := lRet
120 ::sSetUser:cMsg := "ID do Usuário inválido ou vazio, operação cancelada !"
121 Endif
122
123 cTab := GetNextAlias()
124
125 cQuery := " SELECT AI_COD, AI_LOJA, AI_XPORTCL "
126 cQuery += " FROM " + RetSqlName("SA1")
127 cQuery += " WHERE AI_CGC LIKE ' " + SubStr(cCNPJ,1,8) + "' AND "
128 cQuery += " AI_MSBLQL <> '1' AND "
129 cQuery += " D_E_L_E_T <> '*' "
130 dbUseArea(.T., "TOPCONN", TCGenQry(,cQuery), cTab, .F., .T.)
131
132 If (cTab)->(!Eof())
133 While (cTab)->(!Eof()) .and. lRet
134 DbSelectArea("SA1")
135 SA1->(DbSetOrder(1))
136 SA1->(MsSeek(xFilial("SA1") + (cTab)->AI_COD + (cTab)->AI_LOJA ))
137 If Empty(SA1->AI_XPORTCL)
138 ReClock("SA1",.F.)
139 SA1->AI_XPORTCL := cValToChar(::nIDPortal)
140 MsUnlock()

```

Fonte: Próprio autor, 2017.

A figura 19 mostra uma parte do algoritmo que serve para vincular um ID vindo do portal, ao CNPJ do cliente. O algoritmo identifica o cliente pelo CNPJ enviado e preenche um campo no nosso banco de dados com o ID cliente:

- 3) Retornar títulos do cliente: o método receberá o ID do cliente e retornará todos os títulos em abertos e vencidos.

Figura 20 – Método GetTitulo

```

170 WSMETHOD GetTitulos WSRECEIVE nIDPortal,sFiltro WSSEND sTitulos WSSERVICE WSPORTCL
171
172 Local lRet := .T.
173 Local cTab := ""
174 Local cQuery1 := ""
175 Local cQuery2 := ""
176 Local cEmp := ""
177 Local aEmp := {}
178
179 RpcClearEnv()
180 RpcSetType( 3 )
181 RpcSetEnv( '98', '01' )
182
183 cEmp := SuperGetMv( "EF_PORTCL", .F., "07/09/10/11", )
184 aEmp := Separa(AllTrim(cEmp),"|",-F.)
185 cTab := GetNextAlias()
186
187 cQuery1 += " SELECT E.NOMEFIL,TMP.* FROM ( "
188 For nX := 1 to len(aEmp)
189 If nX > 1
190 cQuery1 += " UNION ALL "
191 Endif
192 cQuery1 += " SELECT '" + aEmp[nX] + "' + E1_FILIAL AS CODEMP, A1_COD + A1_LOJA AS CODCLI, A1_NOME AS NOMECLI, "
193 cQuery1 += " E1_EMISSAO AS DTEMISSAO, E1_TIPO AS TIPO, E1_PREFIXO AS PREFIXO, E1_NUM AS NUMDOC, E1_PARCELA AS PARCELA, "
194 cQuery1 += " E1_VALOR AS VALOR, E1_SALDO AS SALDO, E1_VENCREA AS VENCIO, "
195 cQuery1 += " CASE WHEN E1_VENCREA < CAST(GETDATE() AS DATE) THEN 'VENCIDO' ELSE 'ABERTO' END AS [STATUS] "
196 cQuery1 += " FROM " + U.RetSqlEmp(aEmp[nX], "SE1") + " (NOLOCK) E1 "
197 cQuery1 += " LEFT JOIN " + U.RetSqlEmp(aEmp[nX], "SA1") + " (NOLOCK) A1 "
198 cQuery1 += " ON (A1.D.E.L.E.T <> ' ' AND E1_CLIENTE = A1_COD AND E1_LOJA = A1_LOJA) "
199 cQuery1 += " WHERE (A1_XPORTCL LIKE '" + cValtoChar(::nIDPortal) + "[ ]%' OR "
200 cQuery1 += " A1_XPORTCL LIKE '%" + cValtoChar(::nIDPortal) + "%' OR "
201 cQuery1 += " A1_XPORTCL LIKE '%" + cValtoChar(::nIDPortal) + "%') "
202 cQuery1 += " AND E1_SALDO > 0 "
203 cQuery1 += " AND E1_TIPO IN ('NF','FT') "
204 cQuery1 += " AND E1.D.E.L.E.T <> ' ' "

```

Fonte: Próprio autor, 2017.

A figura 20 foi criado para atender ao requisito (N), retornar os títulos do cliente através do ID enviado pelo portal:

- 4) Imprimir Boletos: ao clicar no botão de imprimir boleto, o WS retornará para o portal um arquivo binário base 64, onde o mesmo irá converter em imagem e exibir para o cliente.

Figura 21 – Método gera boleto

```

290
291 /*Gera o boleto e salva em uma pasta definida por parâmetro.*/
292 WSMETHOD GeraBoleto WSRECEIVE sDados WSEND RetornoImagem WSSERVICE WSPORTCL
293
294 Local lRet := .T.
295 Local cTab := ""
296 Local cQuery1 := ""
297 Local cQuery2 := ""
298 Local cEmp := ""
299 Local aEmp := {}
300 Local cArq := "\Boletos_Email\"
301 Local cTabQry := GetNextAlias()
302
303 cQuery:= " SELECT E1_FILORIG, E1_PREFIXO, E1_NUM, E1_PARCELA, E1_TIPO, E1_CLIENTE, "
304 cQuery+= " E1_LOJA, E1_NOMCLI, E1_EMISSAO, E1_VENCREA, E1_VALOR, E1_SALDO, E1_XINDSEE, E1_XENVMAI "
305 cQuery+= " FROM "+ U_RetSqlEmp (::sDados:cEmpresa, "SEI") + " AS SEI "
306 cQuery+= " WHERE E1_FILIAL = '"+:sDados:cFilial+"
307 cQuery+= " AND E1_NUM = '"+:sDados:cTitulo+"
308 cQuery+= " AND E1_PREFIXO = '"+:sDados:cPrefixo+"
309 cQuery+= " AND E1_PARCELA = '"+:sDados:cParcela+"
310 cQuery+= " AND SEL_D_E_L_E_T_ <> ' "
311
312 If Select (cTabQry) > 0
313 (cTabQry) -> (DbCloseArea ())
314 Endif
315
316 dbUseArea (.T., "TOPCONN", TCGenQry (, cQuery), cTabQry, .F., .T.)
317
318 nTot := Contar (cTabQry, "!Eof ()")
319
320 (cTabQry) -> (DbGoTop ())
321
322 If nTot > 0
323
324 U_BOLMAIL (Substr ((cTabQry) -> E1_XINDSEE, 3, 3), Substr ((cTabQry) -> E1_XINDSEE, 6, 5), Substr ((cTabQry) -> E1_XINDSEE, 11, 10), Substr ((cTabQry) -> E1_XINDSEE, 14, 10))
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Fonte: Próprio autor, 2017.

- 5) Imprimir Notas Fiscais: ao clicar no botão de imprimir Nota fiscal, o WS retornará para o portal um arquivo binário base 64, onde o mesmo irá converter em imagem e exibir para o cliente.

Após o processo de cerca de 15 dias de desenho da solução, o desenvolvimento da mesma teve duração de três meses. Nos anexos desse trabalho estão contidos alguns dos testes dos métodos utilizando a ferramenta SoapUI.

5 CONSIDERAÇÕES FINAIS

Este relatório técnico foi elaborado com base em conceitos de tecnologias importantes no cenário da Tecnologia da Informação e grandes nos da Engenharia de Software como Sommerville. Ao serem somadas as informações extraídas por meio do levantamento de requisitos, que foi responsável pela compreensão do problema, pode-se estruturar um bom planejamento de macro etapas de desenvolvimento. Melhorando assim a assertividade das ações para a construção dessa nova solução.

Durante a execução deste trabalho foi possível perceber que, apesar de não ser uma prática nova, a modelagem de requisitos ainda causa estranheza em alguns usuários, exigindo reforço constante da importância da mesma qualidade da entrega. Todavia, não inviabilizou a execução do projeto, que se destacou pelo baixo volume de mudanças e adequações da fase de planejamento à fase de homologação.

Vale ressaltar que para acompanhar a tendência global o portal web também é acessível via plataformas mobile e, futuramente, mais funcionalidades podem ser somadas ao mesmo visando agregar mais valor aos clientes da distribuidora.

REFERÊNCIA

BOOCH, Grady. RUMBAUGH, James. JACOBSON, Ivar. **UML: guia do usuário**. 2°. Ed. – Rio de Janeiro: Elsevier, 2012.

CAETANO, Cristiano. **SoapUI: testes de web services rápidos e descomplicado**. Disponível em: < <https://www.devmedia.com.br/soapui-testes-de-web-services-rapido-e-descomplicado/37461> >
Acessado em: 20 de novembro de 2017.

GUEDES, Gilleanes. **UML 2: uma abordagem na prática**. 2°. Ed. – São Paulo: Novatec, 2011.

LARMAN, Craig. **Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objeto e ao processo unificado**. 2°. Ed. – Porto Alegre: Bookman, 2004.

PRESSMAN, Roger. **Engenharia de software: uma abordagem profissional**. 7°. Ed. – Porto Alegre: AMGH, 2011.

RIBEIRO, Leandro. **O que é UML e Diagramas de Caso de Uso: Introdução Prática à UML**. Disponível em: < <https://www.devmedia.com.br/o-que-e-uml-e-diagramas-de-caso-de-uso-introducao-pratica-a-uml/23408>>. Acessado em: 20 de novembro de 2017.

SINKIKO, Eduardo. **AdvPL: a linguagem AdvPL**. Totvs, 2016.
Disponível em: < <http://tdn.totvs.com/display/tec/AdvPL> >
Acessado em: 10 de novembro de 2017.

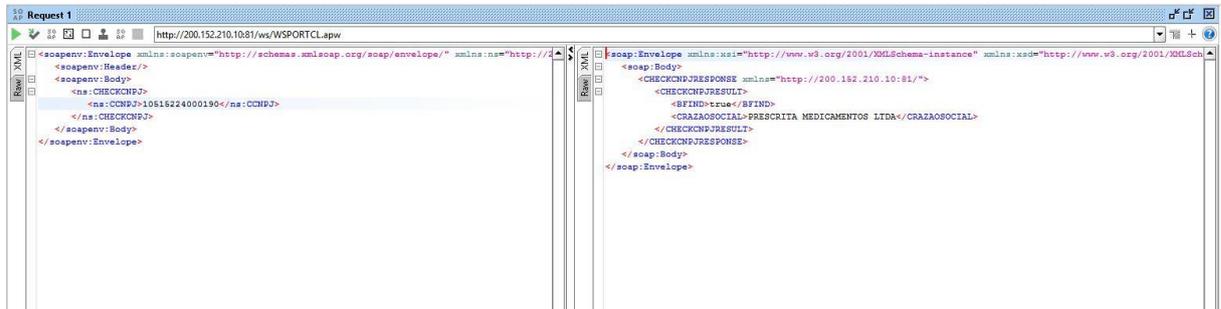
SOMMERVILLE, Ian. **Engenharia de software**. 6°. Ed. – São Paulo: Addison Wesley, 2003.

SOMMERVILLE, Ian. **Engenharia de software**. 9°. Ed. – São Paulo: Pearson Prentice Hall, 2011.

WATANABE, Leo. **TDS Versão 11.3: conceitos gerais**. Totvs, 2016.
Disponível em: < <http://tdn.totvs.com/display/tec/07-Conceitos+gerais> >
Acessado em: 18 de outubro de 2017

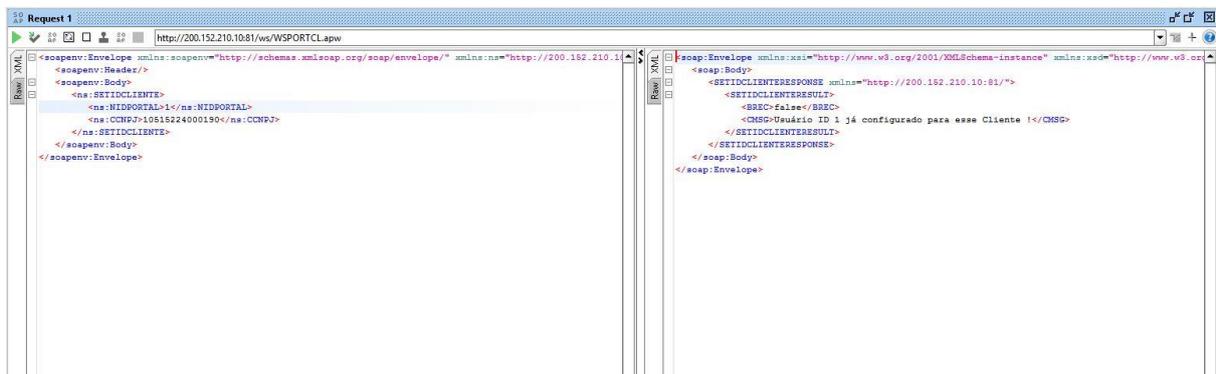
APÊNDICE A - Métodos do Webservice

Segue abaixo uma ilustração do método sendo testado com a ferramenta SoapUI:



Se o CNPJ estiver cadastrado em nossa base, o WS retorna uma mensagem verdadeira e em seguida retorna a razão social do cliente.

Teste do método incluir ID do portal ao cliente, o método solicita dois parâmetros: o primeiro é o ID que será vinculado ao cliente e o segundo parâmetro é o CNPJ desejado:



Retorno do WS quando consultamos títulos de um determinado cliente.

The image shows a browser window displaying the request and response of a SOAP web service. The URL is `http://200.152.210.10:81/ws/WSPORTCLapw`.

Request (Left Panel):

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns="http://200.152.210.10:81/ws/WSPORTCLapw">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:GETTITULOS>
      <ns:IDPORTAL?l</ns:IDPORTAL>
      <ns:SFILTRO?>
        <!--Optional:-->
      <ns:CCODCLI?</ns:CCODCLI>
        <!--Optional:-->
      <ns:CCODEMP?</ns:CCODEMP>
        <!--Optional:-->
      <ns:CNUNDOC?</ns:CNUNDOC>
        <!--Optional:-->
      <ns:CSTATUS?</ns:CSTATUS>
        <!--Optional:-->
      <ns:DVCTFIM?</ns:DVCTFIM>
        <!--Optional:-->
      <ns:DVCTINI?</ns:DVCTINI>
    </ns:SFILTRO>
  </ns:GETTITULOS>
</soapenv:Body>
</soapenv:Envelope>

```

Response (Right Panel):

```

<soap:Envelope xmlns:soap="http://www.w3.org/2001/XMLSchema-instance" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:ns="http://200.152.210.10:81/ws/WSPORTCLapw">
  <soap:Body>
    <GETTITULOSRESPONSE xmlns="http://200.152.210.10:81/ws/WSPORTCLapw">
      <GETTITULOSRESULT>
        <STITULO>
          <CCODCLI>22239501</CCODCLI>
          <CCODEMP>0701</CCODEMP>
          <CNOMECLI>PRESCRITA MEDICAMENTOS LTDA</CNOMECLI>
          <CNOMEEMP>ELFA BSB</CNOMEEMP>
          <CNUNDOC>00093897</CNUNDOC>
          <CPARCELA></CPARCELA>
          <CPREFIX>5</CPREFIX>
          <CSTATUS>VENCIDO</CSTATUS>
          <CTIPO>NF</CTIPO>
          <DEMISSAO>2017-05-11</DEMISSAO>
          <DVENCTO>2017-06-01</DVENCTO>
          <NSALDO>8483.9000000</NSALDO>
          <NVALOR>12664.07000000</NVALOR>
        </STITULO>
        <STITULO>
          <CCODCLI>22239501</CCODCLI>
          <CCODEMP>0701</CCODEMP>
          <CNOMECLI>PRESCRITA MEDICAMENTOS LTDA</CNOMECLI>
          <CNOMEEMP>ELFA BSB</CNOMEEMP>
          <CNUNDOC>00095789</CNUNDOC>
          <CPARCELA></CPARCELA>
          <CPREFIX>5</CPREFIX>
          <CSTATUS>VENCIDO</CSTATUS>
          <CTIPO>NF</CTIPO>
          <DEMISSAO>2017-06-16</DEMISSAO>
          <DVENCTO>2017-07-07</DVENCTO>
          <NSALDO>1737.36000000</NSALDO>
          <NVALOR>3961.36000000</NVALOR>
        </STITULO>
        <STITULO>
          <CCODCLI>22239501</CCODCLI>
          <CCODEMP>0701</CCODEMP>
          <CNOMECLI>PRESCRITA MEDICAMENTOS LTDA</CNOMECLI>
          <CNOMEEMP>ELFA BSB</CNOMEEMP>
          <CNUNDOC>00097139</CNUNDOC>
          <CPARCELA></CPARCELA>
          <CPREFIX>5</CPREFIX>
          <CSTATUS>VENCIDO</CSTATUS>
          <CTIPO>NF</CTIPO>
          <DEMISSAO>2017-07-10</DEMISSAO>
          <DVENCTO>2017-07-31</DVENCTO>
          <NSALDO>2550.25500000</NSALDO>
          <NVALOR>5550.25500000</NVALOR>
        </STITULO>
      </GETTITULOSRESULT>
    </GETTITULOSRESPONSE>
  </soap:Body>
</soap:Envelope>

```

At the bottom of the browser window, the status bar shows: `Auth Headers (0) Attachments (0) WS-A WS-RM JMS Headers JMS Property (0)` and `response time: 2517ms (37837 bytes)`.

O método possui alguns campos opcionais para otimizar a busca por títulos desejados.