

UNIVERSIDADE FEDERAL DO PAMPA

Gustavo Alves Rodrigues

Emissão e gestão simplificada de recibos

Alegrete
2017

Gustavo Alves Rodrigues

Emissão e gestão simplificada de recibos

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação da Universidade Federal do Pampa como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Me. Diego Luis Kreutz

Alegrete
2017

Gustavo Alves Rodrigues

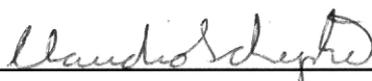
Emissão e gestão simplificada de recibos

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação da Universidade Federal do Pampa como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Trabalho de Conclusão de Curso defendido e aprovado em 27 de novembro de 2017
Banca examinadora:



Prof. Me. Diego Luis Kreutz
Orientador
UNIPAMPA



Prof. Dr. Claudio Schepke
UNIPAMPA



Prof. Dr. Fábio Kepler
UNIPAMPA

AGRADECIMENTOS

Agradeço a minha noiva Natalia Ferraz Mello, que me fez reagir e tornar tudo isso possível, de uma forma que só ela seria capaz de conseguir, ao meu orientador Diego Kreutz, que me orientou da melhor maneira e sempre me apoiou em qualquer situação e aos meus pais, que foram incríveis como sempre.

RESUMO

Muitos trabalhadores autônomos frequentemente fazem uso de recibos como meio de comprovação dos serviços prestados. Esses recibos são posteriormente utilizados nas declarações de Imposto de Renda (IR). Atualmente a maioria dos recibos são impressos em papel, o que colabora com o acúmulo desses documentos, dificultando a organização e facilitando o extravio. Nesse contexto, este trabalho propõe um sistema com a finalidade de simplificar e agilizar a emissão e gestão de recibos. Os principais diferenciais do sistema são: interface web e aplicação móvel para emissão personalizada de recibos, galeria e estúdio de modelos, histórico filtrado, relatórios para declaração do IR e web service. Este último é especialmente interessante para permitir que outros sistemas e plataformas utilizem este serviço. As principais contribuições do trabalho são a definição de uma arquitetura de software para resolver o problema apresentado, uma implementação da arquitetura e uma avaliação prática junto a usuários finais. Empresas e consumidores que testaram o sistema demonstraram empolgação e uma grande aceitação quanto a utilização em seus negócios, destacando a personalização dos recibos e a praticidade das emissões.

Palavras-chave: recibos, simplificação de processos, sistema online, aplicação web para dispositivos móveis, web service.

ABSTRACT

Many self-employed workers often make use of receipts as a mean of proof of the services provided. These receipts are subsequently used in the income tax declaration. Currently most of the receipts are printed on paper, which contributes to the accumulation of these documents, making it difficult to organize and facilitating the loss. In this context, this work proposes a system with the purpose of simplifying and speed up the issuance and management of receipts. The main differentials of the system are: web interface and mobile application for custom issuance of receipts, gallery and model studio, filtered history, reports for income tax declaration and web service. This last one is especially interesting to allow other systems and platforms to use this service. The main contributions of the work are the definition of a software architecture to solve the presented problem, an implementation of the architecture and a practical evaluation with end users. Companies and consumers, who have tested the system, showed excitement and a great acceptance as to the use in their businesses, highlighting the personalization of the receipts and the practicality of the issuances.

Key-words: receipts, process simplification, online system, web application for mobile devices, web service.

LISTA DE FIGURAS

Figura 1 – Arquitetura do sistema	21
Figura 2 – Modelo de recibo personalizado	23
Figura 3 – Modelo de recibo empresarial	23
Figura 4 – Pagamentos Efetuados no IRPF 2017	24
Figura 5 – Rendimentos Tributáveis Recebidos de Pessoas Físicas e do exterior pelo Titular no IRPF 2017	25
Figura 6 – Esquema básico de funcionamento do web service	27
Figura 7 – Implementação da arquitetura do sistema	30
Figura 8 – Fluxograma das etapas da emissão de recibos	32
Figura 9 – Emissão de recibo através do <i>web service</i>	35
Figura 10 – Estrutura da requisição para redefinição do <i>request_token</i>	37
Figura 11 – Tela inicial do painel do usuário	40
Figura 12 – Emissão de recibo básico, informações do emissor	40
Figura 13 – Emissão de recibo básico, informações do receptor	41
Figura 14 – Emissão de recibo básico, finalização	41
Figura 15 – Opções de emissão de recibos	42
Figura 16 – Emissão de recibo básico, terceira etapa	43
Figura 17 – Visualização e cancelamento de recibo	44
Figura 18 – Histórico de recibos	45
Figura 19 – Sub-sistema de notificações	46
Figura 20 – Configurações das notificações e e-mails	47
Figura 21 – Galeria de Modelos	48
Figura 22 – Relatórios Agregados para emissores	49
Figura 23 – Relatórios Agregados para receptores	49
Figura 24 – Estúdio de Modelos	50
Figura 25 – Menu superior do Estúdio de Modelos	50
Figura 26 – Menu interno dos elementos do Estúdio de Modelos	51
Figura 27 – Elemento original	52
Figura 28 – Filtros iniciais	52
Figura 29 – Par de filtros	53
Figura 30 – Filtro: corte	53
Figura 31 – Menu lateral do Estúdio de Modelos	54
Figura 32 – Menu de camadas do Estúdio de Modelos	55
Figura 33 – Menu de controle do Estúdio de Modelos	55
Figura 34 – Visualização de modelos	56
Figura 35 – Registro de novos usuários	57
Figura 36 – Aplicativo móvel: Menu principal e menu do usuário	58
Figura 37 – Aplicativo móvel: emissão de recibos para usuários identificados	59

Figura 38 – Aplicativo móvel: Busca de Recibos	60
Figura 39 – Aplicativo móvel: Notificações do usuário	61
Figura 40 – Aplicativo móvel: Relatórios Agregados para emissores	62
Figura 41 – Aplicativo móvel: Relatórios Agregados para receptores	63
Figura 42 – Aplicativo móvel: Visualização de modelo	63
Figura 43 – Emissão de recibo com logo no Recibo Online	65
Figura 44 – Emissão de recibo no Sem Multa	66
Figura 45 – Emissão de recibo empresarial no Controle na Net	67
Figura 46 – Emissão de recibo sem logo no Meu Recibo	68

LISTA DE TABELAS

Tabela 1 – Comparação dos sistemas de recibos	69
Tabela 2 – Requisições do Web Service	79
Tabela 3 – Respostas das requisições do Web Service	81
Tabela 4 – Respostas das requisições do Web Service (2)	82
Tabela 5 – Respostas das requisições do Web Service (3)	83
Tabela 6 – Respostas das requisições do Web Service (4)	84
Tabela 7 – Respostas das requisições do Web Service (5)	85

LISTA DE SIGLAS

API Application Programming Interface

BLOB Binary Large Object

CNPJ Cadastro Nacional da Pessoa Jurídica

CPF Cadastro de Pessoas Físicas

CSS Cascading Style Sheets

GDGL GD Graphics Library

HTML HyperText Markup Language

HTTP Hypertext Transfer Protocol

HTTPS Hyper Text Transfer Protocol Secure

IP Internet Protocol

IR Imposto de Renda

IRPF Imposto de Renda Pessoa Física

JPEG Joint Photographic Experts Group

JSON JavaScript Object Notation

MITM Man-In-The-Middle

MVC Model-View-Controller

MySQLi MySQL Improved

PHP PHP: Hypertext Preprocessor

PNG Portable Network Graphics

REST Representational State Transfer

SQL Structured Query Language

TSL Transport Layer Security

URI Uniform Resource Identifier

VPN Virtual Private Network

SUMÁRIO

1	INTRODUÇÃO	19
2	SOLUÇÃO PROPOSTA	21
2.1	Arquitetura do sistema	21
2.2	Emissão de recibos	22
2.3	Galeria e estúdio de modelos	22
2.4	Histórico e filtros	23
2.5	Relatórios Agregados para declaração de IR	24
2.6	Sub-sistema de notificações e e-mail	25
2.7	Aplicativo web para dispositivos móveis	25
2.8	Web service	26
3	IMPLEMENTAÇÃO	29
3.1	Ambiente de desenvolvimento	29
3.2	Arquitetura do Sistema	29
3.3	Cadastramento de Usuários	31
3.4	Emissão e Recebimento de Recibos	32
3.5	Estúdio de Modelos	33
3.6	Pesquisa de recibos	34
3.7	Aplicativo web para dispositivos móveis	34
3.8	Web Service	35
4	RESULTADOS	39
4.1	Avaliações	39
4.2	Sistema web	39
4.3	Aplicativo web para dispositivos móveis	57
5	ESTADO DA ARTE	65
5.1	Recibo Online	65
5.2	Sem Multa	66
5.3	Controle na Net	66
5.4	Meu Recibo	67
5.5	Análise comparativa	68
6	CONSIDERAÇÕES FINAIS	71
	REFERÊNCIAS	73

APÊNDICES	77
APÊNDICE A – PRIMEIRO APÊNDICE	79
APÊNDICE B – SEGUNDO APÊNDICE	81

1 INTRODUÇÃO

Trabalhadores autônomos, como freelancers, profissionais da saúde, microempreendedores individuais e seus respectivos clientes, frequentemente fazem uso de recibos para comprovar a realização e o pagamento dos serviços prestados. Esses recibos são utilizados, também, além da finalidade de comprovação, nas declarações de IR para conceder aos contribuintes o direito de abatimento de suas despesas (CRO-RN, 2009; RECEITA FEDERAL, 2017c).

A maioria dos recibos ainda são impressos em papel (WILLINS, 2011), o que colabora com o acúmulo desses documentos e, por conta disso, acaba dificultando o armazenamento e a organização, facilitando o extravio dos mesmos (VADDE; H.; SURHONNE, 2015; GANDINI; SALOMÃO; JACOB, 2001; STAPLES, 2017). Se um recibo não é encontrado e, por razão disso, não é declarado no IR, o emissor pode ser pego na malha fiscal, caracterizando incoerência das informações declaradas com as existentes na base de dados da Receita Federal (MINISTÉRIO DA FAZENDA, 2015).

Existem sistemas que oferecem uma informatização básica do processo de emissão de recibos. Os recursos mais comumente oferecidos são a emissão e informações históricas dos recibos. Porém, os sistemas disponíveis não oferecem recursos como: estúdio para personalização dos recibos, galeria de modelos, diferentes mecanismos de notificação quando do recebimento ou cancelamento de recibos e históricos filtrados. Além disso, os sistemas existentes não disponibilizam recursos como relatórios detalhados para facilitar as declarações anuais de IR. Por fim, outro recurso não encontrado nos sistemas atuais é um *web service* para disponibilizar as funcionalidades do sistema a outras aplicações e sistemas *Web*.

Com o intuito de contribuir na melhora do processo de emissão e o controle de recibos, será proposta uma arquitetura de sistema capaz de tornar o processo de gestão destes documentos simples e ágil, contendo uma interface web, uma aplicação para dispositivos móveis e um web service. O sistema oferece ao usuário recursos como a emissão de recibos, galeria e estúdio de modelos, relatórios agregados para declaração do IR, histórico e rápido acesso aos dados dos recibos. Como forma de verificar o funcionamento e o atendimento aos objetivos postos, o sistema foi colocado em utilização com usuários reais.

O restante deste documento está organizado da seguinte forma. O [Capítulo 2](#) apresenta os componentes e funcionalidades do sistema proposto. No [Capítulo 3](#) são listados os detalhes técnicos das funcionalidades implementadas. Os resultados são apresentados no [Capítulo 4](#). No [Capítulo 5](#) são introduzidos os principais sistemas existentes e uma comparação com o sistema desenvolvido. Por fim, as considerações finais são apresentadas no [Capítulo 6](#).

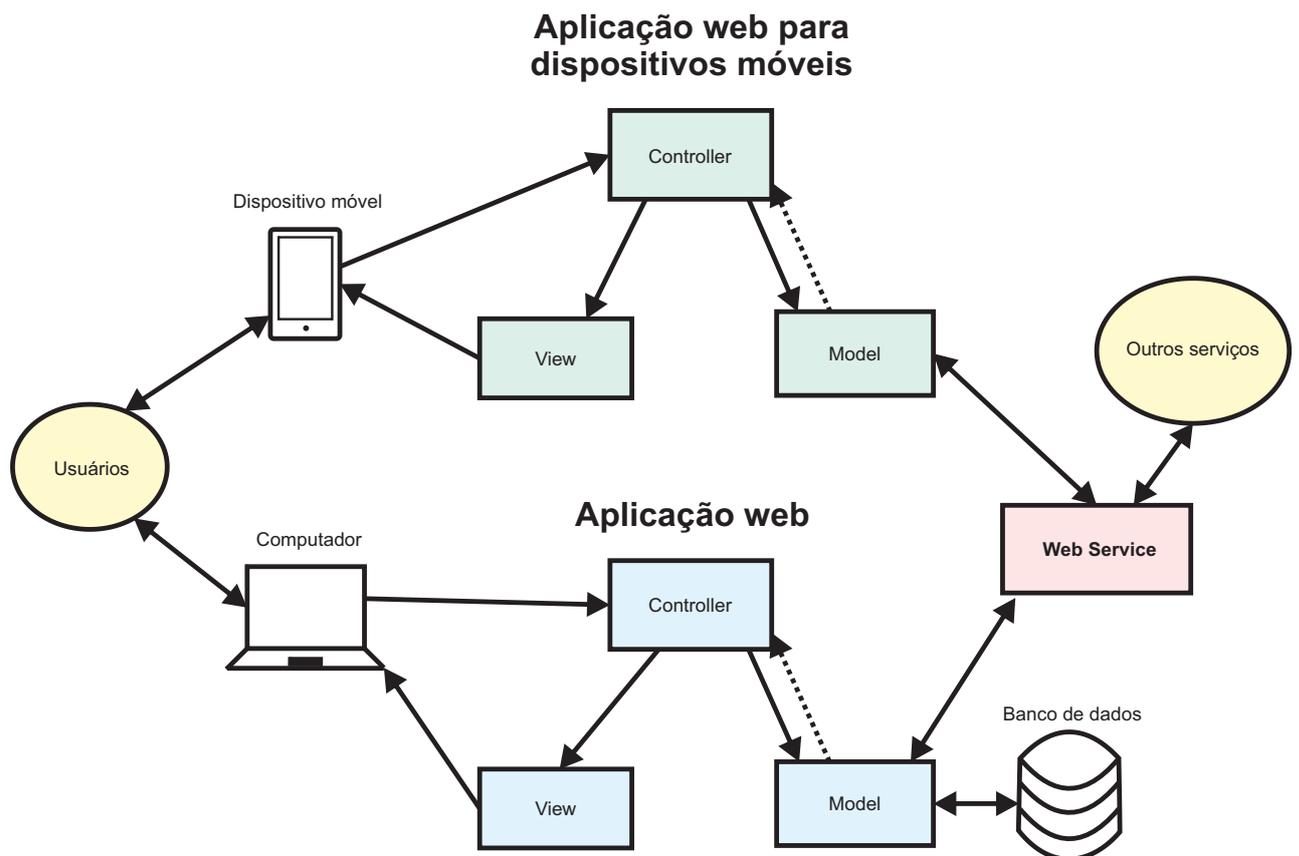
2 SOLUÇÃO PROPOSTA

A solução proposta é dividida em diferentes componentes e funcionalidades, como [Arquitetura do sistema](#), [Emissão de recibos](#), [Galeria e estúdio de modelos](#), [Histórico e filtros](#), [Relatórios Agregados para declaração de IR](#), [Sub-sistema de notificações e e-mail](#), [Aplicativo web para dispositivos móveis](#) e [Web service](#). As seções a seguir descrevem a arquitetura e cada uma das partes do sistema.

2.1 Arquitetura do sistema

As aplicações *Web* foram baseadas no padrão *Model-View-Controller (MVC)*, estruturando o sistema em três camadas: *model* (manipulação dos dados), *view* (interação do usuário com o sistema) e *controller* (controle de requisições do usuário e do sistema) (DEACON, 2005). A [Figura 1](#) apresenta graficamente a arquitetura do sistema.

Figura 1 – Arquitetura do sistema



A camada responsável pelo acesso ao banco de dados é a *model* das aplicações. Nos acessos às aplicações, o usuário somente pode acessar uma *controller* e visualizar uma *view*. A *controller* é responsável por invocar as *views* que os usuários visualizam e se comunicar com as *models*. Toda informação disposta em uma *view* é processada e

definida por uma *controller*, sendo que, para isso, a *controller* pode obter informações do banco de dados utilizando as *models*.

O [Aplicativo web para dispositivos móveis](#) e os serviços externos comunicam-se com o sistema através de requisições realizadas ao *Web service*.

2.2 Emissão de recibos

O sistema proposto tem como principal funcionalidade a emissão e o recebimento de recibos digitais. A emissão é realizada de duas maneiras: como pessoa física (utilizando Cadastro de Pessoas Físicas (CPF)) ou como pessoa jurídica (utilizando Cadastro Nacional da Pessoa Jurídica (CNPJ)).

Os recibos emitidos são baseados em modelos disponíveis na galeria, divididos em modelos para recibos comuns (emitidos por pessoas físicas) e modelos para recibos empresariais (emitidos por pessoas jurídicas).

O recebimento de recibos é a possibilidade dos receptores terem seus recibos disponíveis no sistema, mesmo não tendo emitido recibos. Nesse caso, os receptores que não possuem cadastro no sistema, serão informados por e-mail da emissão dos recibos. Na primeira emissão a um e-mail não cadastrado, uma senha, gerada automaticamente, é enviada para efetivar o cadastro e conceder o acesso aos recibos emitidos ao usuário até o momento. Não é obrigatório identificar-se no sistema para visualização do recibo, visto que um acesso direto ao documento também é constado no e-mail.

O sistema oferece também a funcionalidade de cancelamento de recibos. Esta opção pode ser necessária e utilizada em casos de equívocos nos modelos ou informações incorretas na emissão dos recibos. Tanto o emissor quanto o receptor podem solicitar o cancelamento de um recibo. Este processo é irreversível e ambas as partes precisam concordar para a ação ser concretizada.

2.3 Galeria e estúdio de modelos

O sistema conta com uma galeria de modelos, onde modelos pré-definidos estão disponíveis para a emissão de recibos. Inicialmente, somente o modelo de recibo básico está disponível na galeria do sistema.

É possível também criar modelos com o estúdio interno de modelos. Esta ferramenta conta com opções para inserir elementos textuais e gráficos, permitindo que cada elemento seja redimensionado e reposicionado dentro da área do recibo.

Os modelos da galeria também podem ser editados e salvos como modelos privados do usuário. Além de criar e salvar novos modelos de recibos, o usuário pode solicitar que seus modelos sejam adicionados à galeria do sistema. Porém, neste caso, há um processo de validação e aprovação, isto é, os novos modelos não são imediatamente incluídos na

galeria do sistema, evitando que modelos com conteúdos impróprios sejam compartilhados com outros usuários. A validação e aprovação serão objetos de análise futura.

A [Figura 2](#) demonstra um modelo de recibo simples.

Figura 2 – Modelo de recibo personalizado



Modelo de recibo personalizado. O formulário contém:

- Nome do Emissor: NOME DO EMISSOR
- CPF: CPF 000.000.000-00
- Endereço: ENDEREÇO RUA BAIRRO
- Recibo Nº: RECIBO Nº 130
- Moeda: R\$
- Recebemos de: Recebemos de
- CPF: CPF
- Importância: A importância de
- Referente a: Referente a
- Data: Cidade, 20 de Abril de 2017

Já a [Figura 3](#) apresenta um modelo mais completo, utilizando assinatura/carimbo, uma das funcionalidades do estúdio de modelos que permite ao usuário representar uma das autenticidades características destes documentos.

Figura 3 – Modelo de recibo empresarial



Modelo de recibo empresarial. O formulário contém:

- Logo: LOGO
- Recibo Nº: Nº 122
- Moeda: R\$
- Recebemos de: Recebi(emos) de
- Endereço: Endereço
- Importância: A importância de
- Referente a: Referente a
- Data: 20 de Abril de 2017
- Carimbo: NOME DA EMPRESA, CNPJ: 00.000.000/000-00, ENDEREÇO Rua 000 - Bairro - Cidade

2.4 Histórico e filtros

O histórico permite que os recibos sejam listados para acesso e controle. É possível filtrar a listagem dos recibos, a fim de obter estatísticas, tais como: média de valores em intervalos de tempo, predominância de serviços em lugares específicos, valores mais

significativos em determinados períodos de tempo ou lugares, dentre outras combinações de informações.

2.5 Relatórios Agregados para declaração de IR

Para os contribuintes do IR, existe a possibilidade de gerar relatórios que facilitam o preenchimento da declaração. Esses relatórios consideram os recibos equivalentes ao ano anterior ao exercício (de 01 de janeiro a 31 de dezembro), denominado ano base (RECEITA FEDERAL, 2017a), com as informações necessárias à inclusão no aplicativo IRPF (RECEITA FEDERAL, 2017b). Em outras palavras, o relatório gerado utiliza somente os dados dos recibos válidos para a declaração atual.

Na Figura 4 pode ser observada a ficha de declaração dos pagamentos efetuados, onde os contribuintes receptores dos recibos devem informar os dados do documento.

Figura 4 – Pagamentos Efetuados no IRPF 2017



Pagamentos Efetuados

Dados do Pagamento

Código
13 - Fisioterapeutas no Brasil.

Despesa realizada com
 Titular Dependente Alimentando

CPF do profissional prestador do serviço

Nome do profissional prestador do serviço

Valor pago Parcela não dedutível/valor reembolsado

Fonte: IRPF 2017

O relatório gerado pelo sistema proposto contém o valor total dos serviços, agrupados pela fonte emissora, com as seguintes informações: CPF/CNPJ do emissor e somatório dos valores dos serviços.

Na Figura 5 pode ser visto a ficha de declaração dos rendimentos tributáveis recebidos de pessoas físicas e do exterior pelo titular, onde os contribuintes emissores dos recibos devem informar os dados do documento.

Figura 5 – Rendimentos Tributáveis Recebidos de Pessoas Físicas e do exterior pelo Titular no IRPF 2017

A imagem mostra uma interface web para o lançamento de rendimentos tributáveis. No topo, há um cabeçalho azul com o ícone de uma pessoa e o título "Rendimentos Tributáveis Recebidos de Pessoa Física e do Exterior pelo Titular". Abaixo, há uma seção intitulada "Dados do Lançamento - Mês: janeiro".

Dentro desta seção, há um formulário com os seguintes campos:

- Titular do Pagamento:** Um campo de texto para o CPF (formato . . -) e uma caixa de seleção "Titular do pagamento é o próprio beneficiário do serviço".
- Beneficiário do Serviço:** Um campo de texto para o CPF (formato . . -) e uma caixa de seleção "CPF não informado".
- Valor:** Um campo de texto com o valor "0,00" exibido.

Fonte: IRPF 2017

O relatório gerado pelo sistema para esses contribuintes agrega por mês os recibos, exibindo somente o CPF do receptor e o somatório dos valores dos serviços prestados por esse emissor. A agregação por mês facilita o lançamento dos montantes pertencentes ao respectivo período, pois existe a opção "Incluir Novo", que lança o rendimento atual e redefine o formulário, possibilitando um novo lançamento para o mês supracitado.

2.6 Sub-sistema de notificações e e-mail

O sub-sistema de notificações serve para comunicar o usuário sobre acontecimentos e informações vinculadas a ele. É possível personalizar esse mecanismo, escolhendo a frequência e os tipos de informações a serem notificadas. Por padrão, as notificações são comunicadas no sistema web, no aplicativo web para dispositivos móveis e por e-mail, porém, o usuário pode desativar os meios de comunicação através dos quais não deseja receber notificações.

A priori, são notificados recebimento de recibos, solicitações de cancelamento, dicas de como aproveitar melhor o sistema, rendimentos, avisos do sistema, alertas, lembretes e outras informações.

2.7 Aplicativo web para dispositivos móveis

O aplicativo móvel oferece ainda mais agilidade e simplicidade ao processo, uma vez que a maioria das pessoas está a maior parte do tempo com o seu smartphone em

mãos. Estes dispositivos representam o que há de mais atual em termos de comodidade e praticidade (NGUYEN, 2008).

Através da plataforma móvel é possível emitir recibos, exibir o histórico, buscar recibos, gerar Relatórios Agregados para declaração de IR, solicitar e confirmar cancelamentos de recibos e receber as notificações do sistema.

2.8 Web service

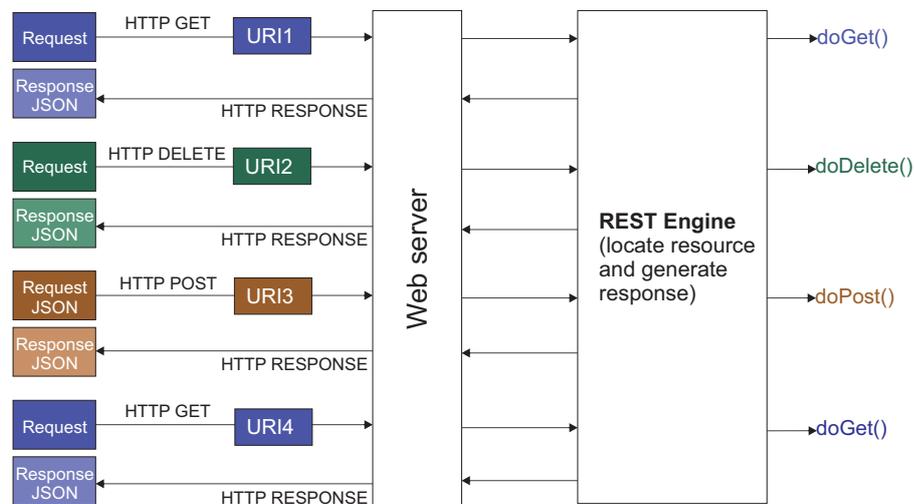
Um web service permite que diferentes aplicações interajam e compartilhem informações com o sistema, através de métodos e padrões, disponíveis em uma interface que independe de hardware e software (RODRIGUES; AFONSO; TOMÉ, 2011).

O *web service* do sistema é baseado no conjunto de padrões arquiteturais Representational State Transfer (REST), onde os Uniform Resource Identifiers (URIs) são utilizados para representar acessos aos recursos (dados e funcionalidades) (IBM, 2008). Os recursos são transferidos utilizando o HTTPS (GET, POST, PUT e DELETE). No sistema proposto, as informações são encaminhadas e retornadas no formato JavaScript Object Notation (JSON), característico do padrão REST.

Para tornar os encaminhamentos e requisições mais seguras, o protocolo HTTPS deve ser utilizado. Esse protocolo opera em uma porta padrão diferente da utilizada pelo HTTP, além de criptografar a camada entre o cliente e o servidor, utilizando o Transport Layer Security (TLS), um protocolo de segurança. Dessa maneira, ataques como o Man-In-The-Middle (MITM), quando as informações são interceptadas por invasores, são dificultados (RESCORLA, 2000).

Um exemplo de utilização é quando um serviço externo que necessita resgatar um recibo com identificador 12, requisita via HTTPS GET a URI <<https://www.3srecibos.com/services/api/recibo/12>>. Nesse caso serão retornadas as informações referentes ao recibo, no formato JSON. Da mesma maneira, se o serviço necessita executar o cancelamento do mesmo recibo, uma requisição via HTTPS DELETE deverá ser feita. Para emitir um recibo, deve ser feita uma requisição via HTTPS POST, com os dados do recibo em formato JSON, a <<https://www.3srecibos.com/services/api/recibo>>. Já uma requisição via HTTPS GET à URI <<https://www.3srecibos.com/services/api/recibo/emitidos>> retornará todos os recibos emitidos referentes ao usuário da sessão. A Figura 6 mostra o esquema de funcionamento do web service, onde as URIs 1, 2, 3 e 4 representam, respectivamente, estes exemplos.

Figura 6 – Esquema básico de funcionamento do web service



Os métodos e recursos do *web service* estão detalhados na [seção 3.8](#) da [Implementação](#). Na resposta da requisição existe um campo *data* e um campo *status*. O campo *data* contém as informações solicitadas ou uma mensagem do sistema e o campo *status* contém um código **HTTP** informando a situação da requisição, que pode ser, por exemplo, erro de permissão, erro interno do servidor ou sucesso na execução.

3 IMPLEMENTAÇÃO

Nas sessões [Ambiente de desenvolvimento](#) e [Arquitetura do Sistema](#) são apresentadas as ferramentas e padrões utilizados no desenvolvimento. Nas demais seções, é apresentada a implementação do [Cadastramento de Usuários](#), [Emissão e Recebimento de Recibos](#), [Estúdio de Modelos](#), [Pesquisa de recibos](#), [Aplicativo web para dispositivos móveis](#) e [Web Service](#).

3.1 Ambiente de desenvolvimento

A linguagem de programação utilizada é o PHP: Hypertext Preprocessor ([PHP](#)) ([PHP GROUP, 2017](#)), versão 5.6, em um servidor Apache ([THE APACHE SOFTWARE FOUNDATION, 2017](#)), versão 2.4, utilizando MySQL versão 5.7, um gerenciador de banco de dados de código aberto, que utiliza a linguagem Structured Query Language ([SQL](#)) para gerenciar o conteúdo armazenado no banco de dados basicamente através de declarações *INSERT* (adicionar registros), *SELECT* (acessar registros), *UPDATE* (atualizar registros) e *DELETE* (remover registros) ([ORACLE CORPORATION, 2017](#)). O framework [PHP CodeIgniter](#) ([BRITISH COLUMBIA INSTITUTE OF TECHNOLOGY, 2017](#)) foi utilizado para estruturar o sistema. A escolha do PHP foi baseada na boa documentação e na grande quantidade de conteúdos informativos, disponíveis na internet, devido a popularidade da linguagem. Já o CodeIgniter é um framework PHP que se destaca pela sua simplicidade e estrutura que segue o padrão [MVC](#).

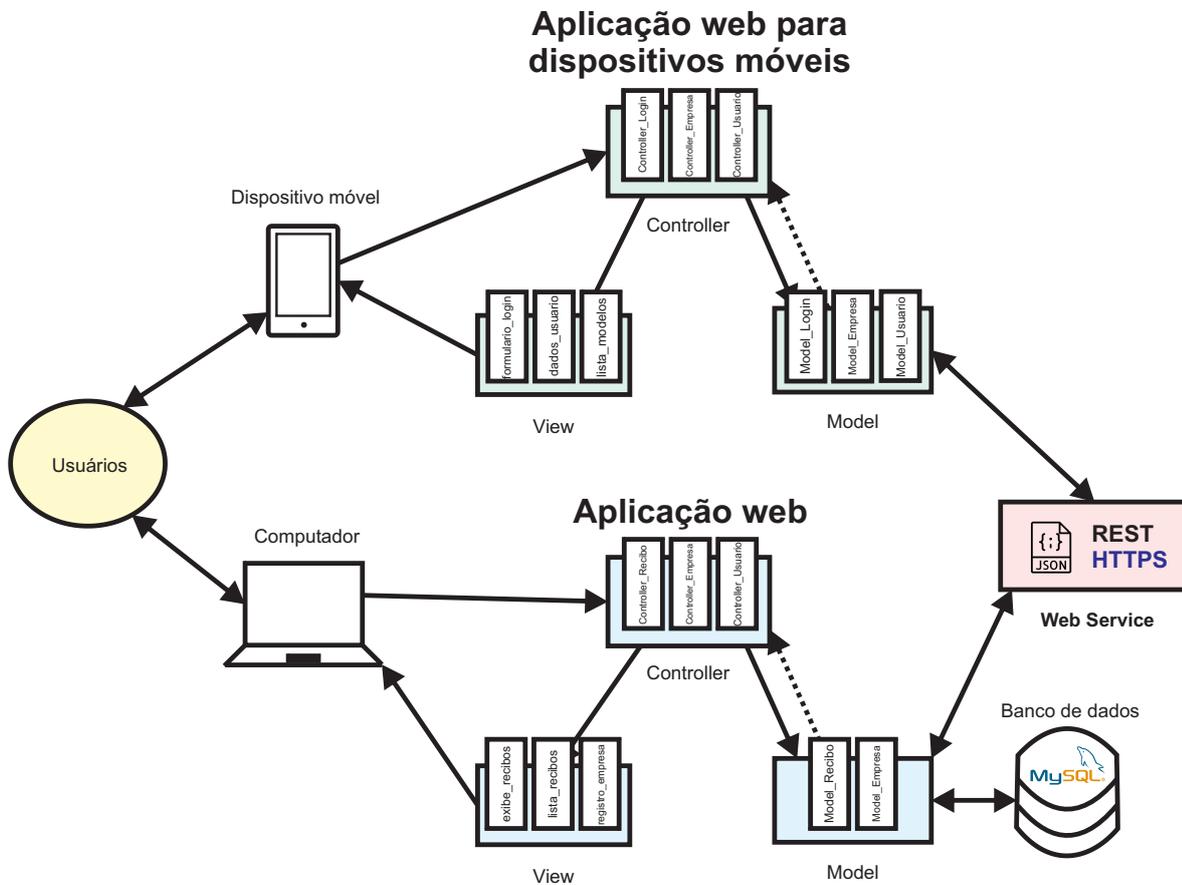
A interface do usuário foi implementada utilizando HyperText Markup Language ([HTML](#))⁵ ([W3C, 2017b](#)), Cascading Style Sheets ([CSS](#))³ ([W3C, 2017a](#)) e JavaScript ([EICH, 1995](#)), visto que são as tecnologias existentes para marcação, estilização e controle de ambientes web. O JQuery, um *framework* que facilita e torna mais prático o uso do JavaScript ([THE JQUERY FOUNDATION, 2017](#)), foi utilizado. Além do JQuery, não foram utilizados outros *frameworks front-end*, pois em algumas funcionalidades, como o [Estúdio de Modelos](#), foi necessário o uso de características mais específicas das linguagens.

O cliente Git ([SOFTWARE FREEDOM CONSERVANCY, 2017](#)) e o serviço de controle de versionamento de código GitLab ([GITLAB COMMUNITY, 2017](#)) foram utilizados no processo de desenvolvimento. Isto facilitou a manutenção das diferentes versões do código.

3.2 Arquitetura do Sistema

A implementação seguiu a arquitetura proposta para o sistema, conforme definido na [Capítulo 2](#). Ela é apresentada em partes: a estrutura dos componentes, o banco de dados e o web service.

Figura 7 – Implementação da arquitetura do sistema



Estrutura dos componentes

O CodeIgniter foi utilizado por ser um *framework* de produtividade, que segue o padrão MVC. Na aplicação móvel, este padrão foi seguido manualmente. Ao total, são 7 *models*, 32 *views* e 10 *controllers* na aplicação web e 13 *models*, 17 *views* e 14 *controllers* na aplicação móvel.

A página apresentada ao usuário final, *view*, pode ser invocada somente por um *controller*, que, por sua vez, pode somente obter informações de um banco de dados utilizando uma *model*. Os *controllers* podem obter informações internas, como dados de formulários originados de uma *view*, dados da sessão e outras informações não provenientes de bancos de dados. As *views* não devem fazer outra tarefa além da apresentação de informações ao usuário, ou seja, uma *view* não pode acessar nenhum tipo de informação diferente das que foram fornecidas pelo *controller*.

A Figura 7 ilustra a arquitetura implementada, onde as *views* `exibe_recibo` e `lista_recibos` utilizam o *controller* `Controller_Recibo` para apresentar as informações dos recibos ao usuário, assim como a *controller* `Controller_Recibo` utiliza a *model* `Model_Recibo` para obter essas informações. No caso da *view* `registro_empresa`, a *controller* `Controller_Empresa` é utilizada para adicionar uma nova empresa no banco de dados. A *controller*

Controller_Usuario é responsável por solicitar à *model* os dados dos usuários e apresentar através de uma *view*.

No aplicativo móvel, quando um usuário necessita se identificar, a *controller Controller_Login* deve ser requisitada para apresentar a *view formulario_login*, que possui um formulário com o destino sendo um método da própria *controller Controller_Login*, e solicitar que a *model Model_Login* requisiute do *web service* a autenticação das informações. Assim como as *views dados_usuario* e *lista_modelos*, são invocadas respectivamente pelas *controllers Controller_Usuario* e *Controller_Modelo*, que por sua vez, requisitam os dados das *models Model_Usuario* e *Model_Modelo*, respectivamente.

Banco de dados

O CodeIgniter também oferece suporte ao uso do banco de dados MySQL, através do MySQL Improved ([MySQLi](#)). Este oferece diferentes vantagens em relação ao antecessor. Uma das principais vantagens é a possibilidade de preparar os comandos *SQL* antes da execução, filtrando e dificultando ataques *SQL-Injection*, que são códigos mal intencionados injetados em campos de entrada de aplicações web ([W3C, 2017c](#)).

Web Service

A [Aplicativo web para dispositivos móveis](#) e os serviços externos comunicam-se com o sistema através de requisições *HTTPS* realizadas ao *Web service*, com dados no formato *JSON*. As requisições são recebidas e processadas pelo *Web service*. Este, por sua vez, envia as requisições de dados aos *models* correspondentes do servidor web. Após o *feedback* do *model*, o *Web service* envia um documento *JSON*, contendo os dados solicitados ou o status da operação realizada, ao solicitante.

O *Web service* constitui o conjunto das *controllers*, dentro da *aplicação web*, e é acessado através do *URI* [<https://www.3srecibos.com/services/api>](https://www.3srecibos.com/services/api).

3.3 Cadastramento de Usuários

O cadastramento de usuários é realizado de duas maneiras, manualmente pelo formulário de registro ou automaticamente durante a emissão de um recibo. O registro manual é encontrado na página inicial do sistema, sendo necessário informar o *nome completo*, *e-mail*, *senha* e *confirmação de senha*. Opcionalmente, também é possível informar o *CPF*. O registro automático ocorre quando um usuário emite um recibo, pela primeira vez, sem antes ter-se registrado no sistema. Neste caso, as contas de usuário serão criadas automaticamente considerando os emails informados no preenchimento do recibo. Os usuários serão notificados por e-mail da criação das contas no sistema.

3.4 Emissão e Recebimento de Recibos

Figura 8 – Fluxograma das etapas da emissão de recibos



A emissão de recibos é dividida em cinco etapas, como é mostrado na [Figura 8](#), requerendo informações mínimas para ser executada. Um recibo deve conter, obrigatoriamente, um emissor, um receptor, um modelo, os dados dos campos referentes ao modelo e uma localização. O dado que identifica tanto um emissor, quanto um receptor, em um recibo, é o seu respectivo identificador (*id*). O identificador do emissor é obtido a partir dos dados da sessão iniciada, se ela existir, caso contrário, o emissor é um usuário em sua primeira emissão e deverá informar seu nome e e-mail para ter sua conta criada automaticamente. Se um usuário informa um e-mail que já está cadastrado no sistema, ele será solicitado a identificar-se (iniciar uma sessão) para efetuar a emissão. Esta solicitação é importante, pois evita que um usuário em seu primeiro acesso emita recibos com autoria de usuários registrados. Após isso, o receptor é identificado, sendo necessário o nome e o e-mail do mesmo. Este parágrafo descreve os dados de entrada da emissão de um recibo.

A segunda etapa é a seleção do modelo de recibo a ser utilizado. Esse modelo pode ser escolhido a partir da galeria oficial do sistema ou da galeria do usuário, caso haja uma sessão iniciada. Usuários em seu primeiro acesso podem somente utilizar o *modelo 1*, que é o *recibo básico*, um modelo da galeria oficial. Esse é um modelo comum (emitido para pessoas físicas) e possui as informações mínimas para a emissão de um recibo.

Na terceira etapa, o recibo é finalizado, onde o usuário irá informar os dados de cada campo do modelo de recibo escolhido. Os campos do *recibo básico*, por exemplo, são a descrição do serviço prestado e o valor. Qualquer modelo do sistema deverá, obrigatoriamente, possuir estes dois campos. Os dados do modelo constituem o conteúdo dos recibos. A localização da emissão é obtida automaticamente na finalização, utilizando a Application Programming Interface (API) de localização do Free Geo IP ([FREE GEO IP, 2017](#)), que retorna informações de localização através de um Internet Protocol (IP). Para usuários que utilizam *proxies* ou Virtual Private Network (VPN)s, sua localização será objeto de análise futura.

Após a finalização, os usuários que não possuem registro são cadastrados automaticamente e o novo recibo é processado. O receptor recebe uma notificação automática de emissão do recibo através do sub-sistema de notificações e através do e-mail. Nos

e-mails, o recibo pode ser acessado via link, que utiliza uma chave de acesso direto, sem a necessidade de identificação do usuário. Cada recibo emitido sempre implicará em um recibo recebido, tornando o sistema uma ferramenta útil e prática também para pessoas que apenas recebem recibos.

3.5 Estúdio de Modelos

O Estúdio de Modelos foi implementado em *JavaScript* e *jQuery*, utilizando requisições feitas internamente à *controller Estudio*, que, através da *model Model_Estudio*, insere as informações no banco de dados. O estúdio utiliza *tags HTML* para representar os elementos do modelo. A estrutura de modelo gerada pelo estúdio é organizada separadamente no banco de dados, ou seja, cada elemento, assim como suas configurações e propriedades, são armazenados individualmente.

Com exceção dos elementos do tipo *imagem* e *assinatura/carimbo*, descritos nos parágrafos a seguir, existem propriedades de cor, aplicáveis ao texto e/ou ao fundo do elemento. Para isso, a ferramenta Tiny jQuery colorPicker (DEMATTE, 2017) foi utilizada, possibilitando a escolha personalizada das cores.

Os elementos do tipo *imagem* são utilizados para representar informações gráficas provenientes de arquivos, como, por exemplo, logomarcas. Estes elementos são limitados a arquivos de no máximo 1mb, dos tipos Portable Network Graphics (PNG) e Joint Photographic Experts Group (JPEG), com o objetivo de obter um padrão de arquivos de imagens. Esses elementos, quando inseridos, não são instantaneamente enviadas para o servidor. Durante a utilização do Estúdio de Modelos, as informações destas imagens, quando inseridos, ficam salvas no armazenamento interno do navegador, no formato Binary Large Object (BLOB) e, somente após a conclusão do modelo, são enviadas para o servidor. Com isso, o usuário trabalha localmente durante a criação, melhorando a experiência do usuário no uso do estúdio de modelos. Esses elementos, quando adicionados, são redimensionados para encaixar na área de criação. Eles possuem uma opção para redefinir suas dimensões para o valor inicial.

Os elementos do tipo *assinatura/carimbo* são utilizado para imitar a autenticidade tradicional destes documentos. Eles são similares aos elementos imagem, também provém de um arquivo, com a diferença de que o objeto é enviado e armazenado no servidor durante a criação do modelo. O motivo de seu armazenamento ser no servidor é que este elemento passará por filtros para remover sua cor de fundo, limitar suas dimensões à área preenchida e suavizar seus traçado. O objetivo dos filtros é aumentar a qualidade desses elementos para representar carimbos e/ou assinaturas. Para aumentar a qualidade do resultado, o arquivo utilizado deve representar uma assinatura e/ou carimbo escaneado de uma assinatura feita a caneta, preferencialmente preta, em uma folha branca, com um ambiente iluminado. Os filtros aplicados ao elemento, em ordem, são: 40% de correção gamma, escala de cinza, -100 de contraste e 30 de aumento de brilho. Após os filtros

iniciais, são aplicados pares de filtros cinco vezes consecutivas para suavizar a assinatura e/ou carimbo, sendo eles: 2 unidades suavização e -50 de contraste. Após a definição dos traçados, a imagem resultante é cortada para limitar-se a área preenchida, removendo a área clara e, por fim, a imagem passa por mais um filtro onde a cor branca é removida (transformada em transparente). Os filtros utilizados foram escolhidos levando em conta a capacidade da biblioteca utilizada GD Graphics Library ([GDGL](#)) do [PHP](#). A sequência de filtros foi definida através da análise manual de resultados empíricos.

Um modelo só pode ser finalizado quando os campos do tipo *valor* e *descritivo* estiverem presentes. Na finalização de um modelo, deve ser informado o nome do modelo, tipo (*comum* ou *empresarial*) e a opção de compartilhamento do modelo com outros usuários.

No momento que um modelo é criado, uma imagem miniatura é gerada com base no conteúdo da área de criação. Essa operação simula um *print*, utilizando o *CANVAS*, uma funcionalidade do [HTML5](#), que representa uma área de renderização dinâmica de elementos gráficos, junto com a ferramenta *html2Canvas* ([NIKLAS VON HERTZEN, 2017](#)). As informações do conteúdo da área de criação são geradas no formato [BLOB](#) e gravadas no servidor como um arquivo de imagem [PNG](#).

3.6 Pesquisa de recibos

A *view* que apresenta os resultados da pesquisa é a *lista_recibos*, esta *view* é utilizada em todas as partes do sistema que precisam exibem uma lista de recibos, como por exemplo, o histórico de recibos. A *controller* que processa o termo digitado e invoca a *view* dos resultados é a *controller Recibos*, que obtém as informações da *model Model_Recibos*.

O critério de busca utilizado para um recibo retornar na pesquisa é quando o termo digitado é subpalavra de, pelo menos, o *nome do emissor/receptor*, o *valor* de qualquer campo do recibo, podendo ser o valor do recibo, a descrição do recibo e/ou o valor de um texto estático do recibo, o *e-mail* do emissor/receptor, o [CPF](#) do emissor/receptor, o *nome da empresa geradora* ou o [CNPJ](#) da empresa geradora..

3.7 Aplicativo web para dispositivos móveis

O aplicativo móvel possui todas as funcionalidades da Aplicação Web, com exceção do Estúdio de Modelos. Como detalhado na seção [Capítulo 2](#), sua implementação é baseada no envio e recebimento de requisições ao *web service*.

As informações já processadas ficam disponíveis mesmo sem uma conexão com a internet, pois um armazenamento interno (*local storage*) é utilizado. Este armazenamento possui diferentes limites de espaço, porém, em média, possui 5mb disponíveis. Para isso, quando o usuário está conectado, a cada requisição, as informações obtidas são armazenadas no *localStorage*. As disponibilidade *offline* é útil para permitir a visualização

de recibos, porém, não é recomendada para gerar relatórios, visto que pode resultar em informações desatualizadas.

Com o objetivo de manter a validade das informações armazenadas, a cada 24 horas, uma atualização será realizada nas informações armazenadas, se necessário.

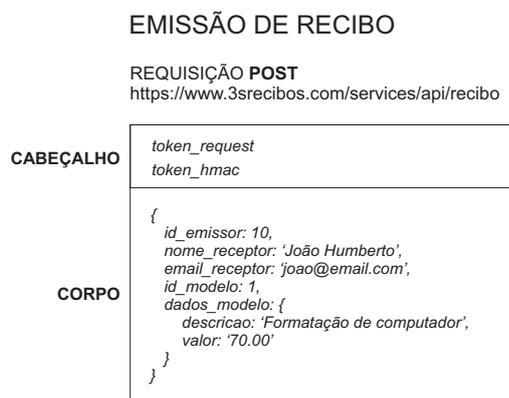
Além disso, o aplicativo é híbrido, o que significa que ele pode ser executado em qualquer ambiente que tenha suporte a aplicações web. O [HTML5](#), [CSS3](#), JavaScript e o [jQuery](#) foram utilizados na sua implementação.

3.8 Web Service

Com o *web service* é possível utilizar todas as funcionalidades do sistema em ambientes externos, com exceção do Estúdio de Modelos.

Para emitir um recibo através do *web service*, por exemplo, deve ser enviada uma requisição do tipo *POST* para o método *recibo*, com as informações do recibo contidas no corpo da mesma, sendo elas, o emissor, identificado através de um identificador ou de um nome e e-mail, o nome e e-mail do receptor, o identificador do modelo e os dados referentes aos campos do recibo. A informação retornada contém o identificador do novo recibo e a *chave* que permite acesso direto ao documento. A [Figura 9](#) representa a estrutura desta requisição.

Figura 9 – Emissão de recibo através do *web service*



O funcionamento do *web service* é baseado nas permissões dos usuários, sendo necessária a autenticação para ter acesso aos métodos permitidos a este usuário. Os métodos que dispensam autenticação são o *login*, o *recibo* (em requisições do tipo *POST*), o *usuario* (em requisições do tipo *POST*), o *modelo/id* (em requisições *GET*) e o *modelo/id/campos* (em requisições *GET*).

Todas as informações recebidas e transmitidas pelo *web service* estão contidas no corpo da requisição, no formato [JSON](#), o que é uma característica do protocolo [REST](#). Essas informações são criptografadas pelo protocolo [HTTPS](#), descrito no terceiro pará-

grafo da [seção 2.8](#) da [Solução proposta](#). O uso protocolo [HTTPS](#) é forçado, através de uma regra definida no servidor Apache, mostrada no código abaixo, que utiliza o módulo `mod_rewrite`, redirecionando as conexões da porta 80 ([HTTP](#)) para a porta 443 ([HTTPS](#)).

```
RewriteEngine On
RewriteCond %{HTTPS} !=on
RewriteRule ^/?(.*) https://\%{SERVER\_NAME}/\%1 [R,L]
```

O comando `RewriteEngine On` ativa o módulo `mod_rewrite`, com isso é possível reescrever as [URIs](#) acessadas pelos usuários através da definição de regras. Com o comando `RewriteCond %HTTPS !=on`, é possível aplicar uma condição para a regra ser executada, nesse caso é verificado se o protocolo [HTTPS](#) está sendo utilizado. O último comando define a regra a ser aplicada, alterando as [URIs](#) para o padrão [HTTPS](#).

Tokens são utilizados para autenticar e tornar mais seguras as requisições aos métodos do *web service*. Existem três tipos de *tokens*: `session_token`, que é único e serve para identificar os usuários, `request_token`, que serve para autenticar cada requisição enviada entre o cliente e o servidor, e `security_token`, que serve como uma assinatura, validando as informações contidas na requisição.

O `session_token` é obtido através de uma requisição do tipo `POST` ao método `login`, resultante da aplicação da `SHA-512`, uma função *hash*, que retorna uma expressão de *512 bits*, a um valor lido do `/dev/urandom`, que reusa o pool interno do sistema operacional para produzir bits aleatórios. O `e-mail` e `senha` do usuário são necessários para a sua geração, impedindo que o `session_token` seja obtido sem identificação. Cada requisição implica na geração de um novo `request_token`, tornando-o único por requisição. O primeiro `request_token` é obtido através da aplicação da `SHA-512` ao `session_token`, demonstrado na fórmula a seguir.

```
request_token = sha512(session_token)
```

Os `request_tokens` subsequentes, são obtidos através da aplicação da `SHA-512` ao `request_token` anterior, mostrado na fórmula abaixo. Este processo deve ocorrer também no cliente, fazendo com que exista uma sincronia entre `request_tokens` do cliente e do servidor.

```
request_token = sha512(request_token)
```

Security_tokens são gerados a partir da aplicação de uma `hash_hmac` ao corpo da requisição (formatado em [JSON](#)). A `hash_hmac` é similar as outras *hashs* utilizadas no sistema, porém combinada com uma chave de segurança. Com a chave de segurança, é possível verificar a integridade e a autenticidade da requisição. Neste caso, a chave de segurança é o `session_token` do usuário. Quando o servidor é requisitado, o mesmo

processo é realizado para gerar uma *security_token* e comparar com a recebida. A fórmula a seguir demonstra a geração das *security_tokens*.

```
security_token = sha512(request_body, session_token)
```

Tanto o *request_token* quanto o *security_token* são úteis para evitar ataques do tipo MITM, pois se um atacante intercepta as requisições, além de ter que passar pela proteção do HTTPS, ele terá que reconhecer o padrão utilizado para manter a sincronia (*request_tokens*) e o *session_token*.

Nos casos onde um atacante consegue reconhecer estes padrões, ele pode ocasionar uma perda de sincronia entre o usuário verdadeiro e o servidor. Para situações como esta, existe um recurso do *web service* (<<https://www.3srecibos.com/services/api/token>> [POST]) que serve para redefinir o *request_token*. A requisição a este recurso deve conter um número randômico e outra *hash* (*reset_token*), gerada a partir do *session_token* concatenado com este valor randômico. A fórmula abaixo demonstra a geração da *reset_token*.

```
reset_token = sha512(session_token+random_number)
```

A partir disso, o servidor pode realizar o mesmo processo, utilizando o número randômico e o *session_token* de cada usuário, e comparar as *reset_tokens* para redefinir o *request_token* do usuário identificado. O novo *request_token* passa a ser a própria *reset_token*. A Figura 10 representa a estrutura de uma requisição de redefinição.

Figura 10 – Estrutura da requisição para redefinição do *request_token*

REDEFINIÇÃO DE TOKEN

REQUISIÇÃO PUT
https://www.3srecibos.com/services/api/token

CORPO

```
{
  token_reset: ddaf35a193617abacc417349ae2...
  random_number: 0.82101973271
}
```

Todos os tipos de requisições e métodos do *web service*, assim como suas respostas, são descritos no Primeiro Apêndice e no Segundo Apêndice.

4 RESULTADOS

As sessões a seguir apresentam as [Avaliações](#) obtidas e os resultados do [Sistema web](#) e do [Aplicativo web para dispositivos móveis](#).

4.1 Avaliações

Foram realizadas simulações em empresas e consumidores reais. Estas simulações basearam-se nos serviços de quatro empresas, com o objetivo de testar a proposta do sistema, sem substituir os métodos atualmente utilizados por elas. O sistema foi avaliado pelas empresas durante um período de um mês. Nesse período, os consumidores foram informados da simulação e convidados a participar. Ao final dos testes, um total de 15 consumidores utilizaram o sistema e afirmaram, assim como as próprias empresas, que a proposta satisfaz suas necessidades, simplificando a prática de emissão e gestão de recibos.

4.2 Sistema web

Pode ser observado na [Figura 11](#) a tela inicial do painel do usuário, onde são exibidos os últimos recibos emitidos ou recebidos em ordem temporal decrescente.

O sistema conta com uma barra lateral em que são exibidas as informações do usuário, as opções de configurações e sair da conta. Nas configurações da conta as notificações e os dados pessoais e de acesso são gerenciados. Além disso, a barra lateral conta com o menu de navegação, composto pelo itens: Início (página inicial), Recibos (histórico de recibos), Empresas (gerenciamento de empresas do usuário), Galeria de Modelos (modelos oficiais e do usuário) e Relatórios (geração de relatórios).

Na barra superior, o sub-sistema de notificações, a opção para geração de recibos e a ferramenta de busca estão disponíveis.

Figura 11 – Tela inicial do painel do usuário

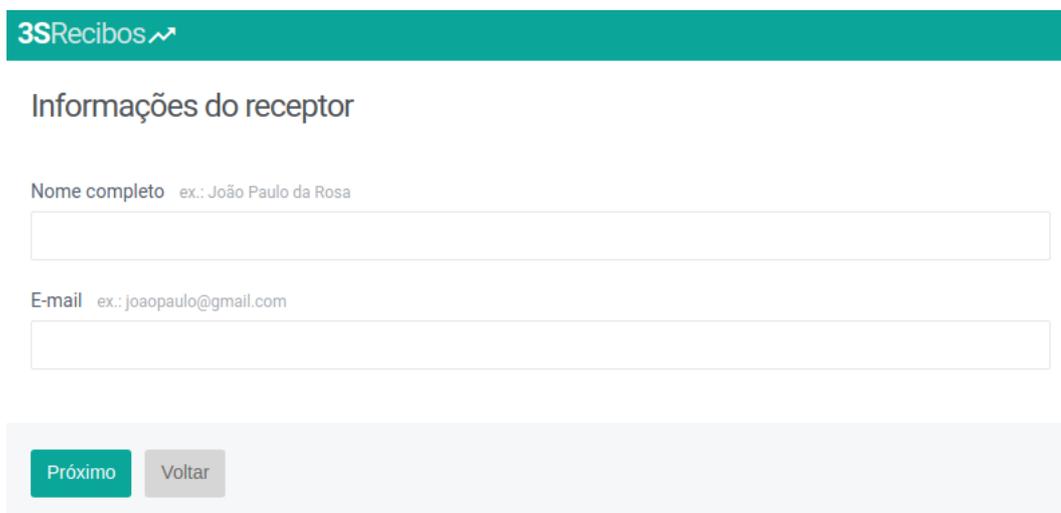


As etapas de emissão de um recibo, no primeiro acesso de um usuário, podem ser observadas na [Figura 12](#), [Figura 13](#) e [Figura 14](#). No primeiro acesso, a etapa de seleção de modelo é pulada, pois o modelo 1 (recibo básico) é automaticamente atribuído à emissão.

Figura 12 – Emissão de recibo básico, informações do emissor

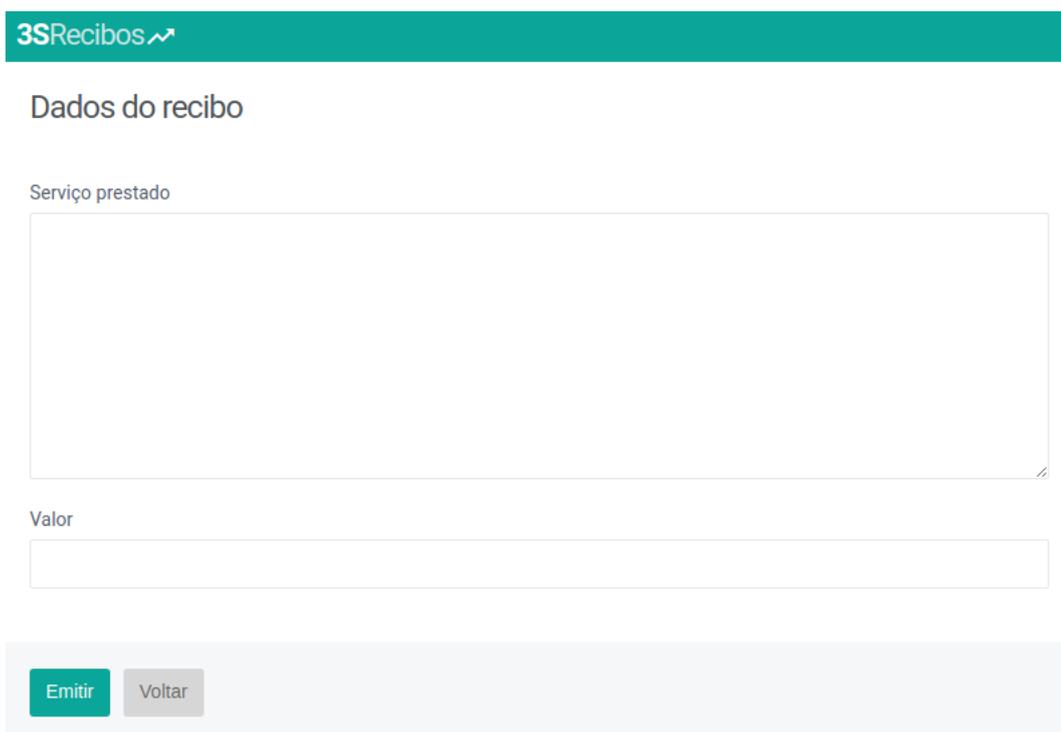
The screenshot shows the 'Emitir recibo básico' (Issue basic receipt) form. The form is titled 'Emitir recibo básico' and is located under the '3SRecibos' logo. Below the title, there is a section for 'Suas informações' (Your information). This section contains two input fields: 'Nome completo' (Full name) with an example 'ex.: João Paulo da Rosa' and 'E-mail' with an example 'ex.: joaopaulo@gmail.com'. At the bottom of the form, there are two buttons: 'Próximo' (Next) and 'Voltar' (Back).

Figura 13 – Emissão de recibo básico, informações do receptor



The screenshot shows the '3SRecibos' logo at the top left. Below it is the heading 'Informações do receptor'. There are two input fields: the first is labeled 'Nome completo' with the example 'ex.: João Paulo da Rosa', and the second is labeled 'E-mail' with the example 'ex.: joaopaulo@gmail.com'. At the bottom of the form area, there are two buttons: 'Próximo' (Next) and 'Voltar' (Back).

Figura 14 – Emissão de recibo básico, finalização



The screenshot shows the '3SRecibos' logo at the top left. Below it is the heading 'Dados do recibo'. There are two input fields: the first is labeled 'Serviço prestado' and is a large text area; the second is labeled 'Valor'. At the bottom of the form area, there are two buttons: 'Emitir' (Issue) and 'Voltar' (Back).

Na [Figura 15](#) é possível observar as opções de emissão de recibo, sendo elas, comum (CPF) e empresarial (CNPJ).

Figura 15 – Opções de emissão de recibos

Emitir recibo

Emitir como Pessoa Física

Gustavo Alves Rodrigues (99-99-9999)

Próximo

Emitir como Pessoa Jurídica

Minhas empresas

Brasil LTDA ▼

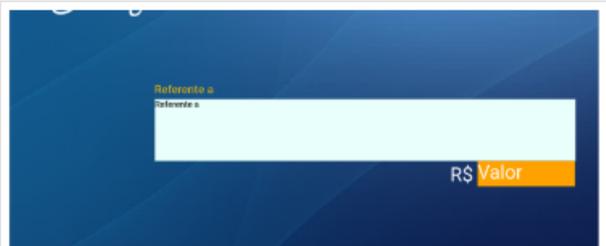
Próximo

Na [Figura 16](#) é possível observar a etapa de seleção do modelo de um recibo para usuários identificados, a escolha do modelo.

Figura 16 – Emissão de recibo básico, terceira etapa

Escolha o modelo

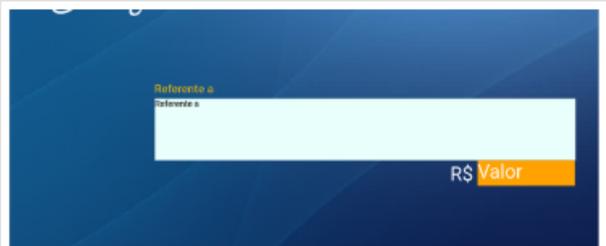
Galeria 3S



Recibo básico
Por Equipe 3S

Próximo

Minha galeria

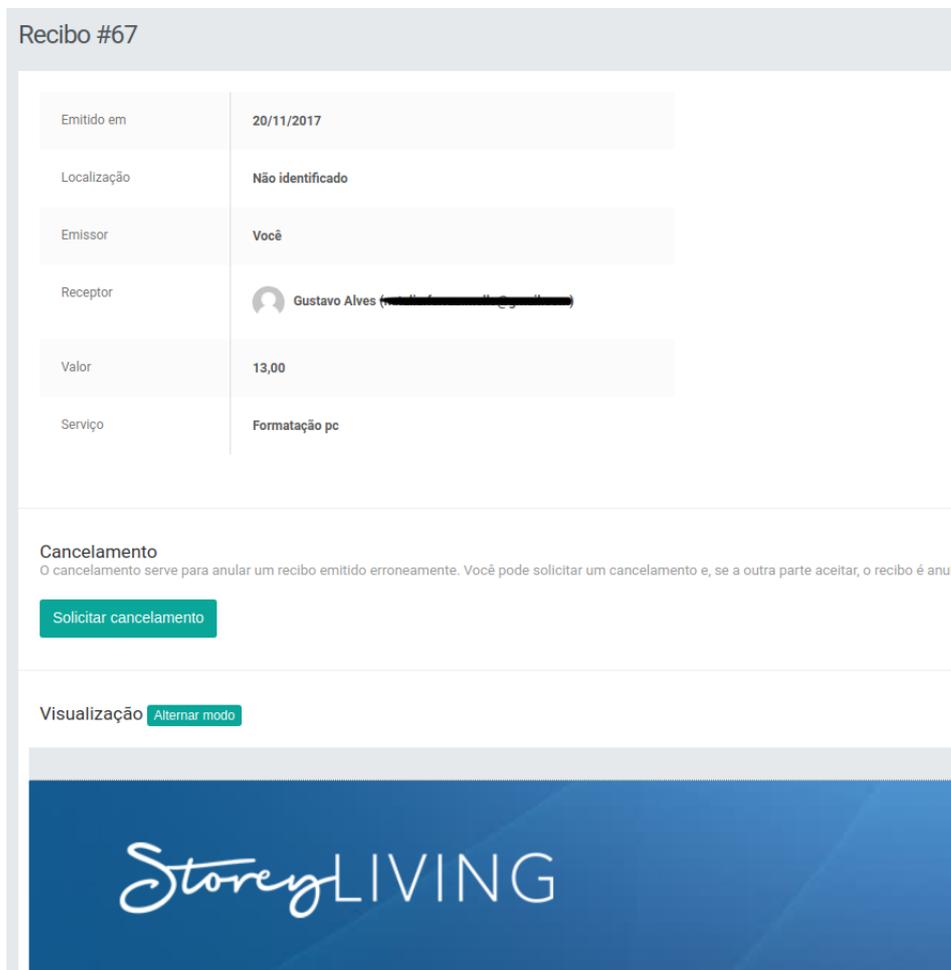


Modelo Comum 1

Próximo

A [Figura 17](#) mostra a tela de visualização do recibo, que inclui as ações disponíveis ao documento. As informações completas referentes aos recibos são encontradas nesta página, além de ser possível exibir o recibo modelado, com todos os campos pertencentes ao modelo utilizado.

Figura 17 – Visualização e cancelamento de recibo



A Figura 18 apresenta o histórico de recibos, juntamente com as opções de ordenação e filtragem. Com o histórico é possível exibir a lista completa de todos os recibos recebidos, emitidos e cancelados pelo usuário. Quando o recibo é do tipo empresarial (CNPJ), o nome da empresa é mostrado. O número total de recibos é exibido em cada categoria (recebidos, emitidos e cancelados).

O histórico pode ser ordenado em ordem crescente ou decrescente, pelas seguintes informações: *data*, *emissor/receptor*, *serviço*, *valor*, *localização*. Além disso, o histórico pode ser filtrado, de modo que os recibos listados devem bater com as informações definidas, por: *período*, informando data inicial, data final, horário inicial e/ou horário final, *tipo*, podendo ser recibos comuns (CPF), empresariais (CNPJ) ou ambos, *modelo*, *emissor*, *receptor*, *valor*, informando o valor mínimo e/ou máximo a ser considerado, e *localização*.

Em cada categoria é exibida a média e o valor total dos recibos, com isso é possível gerar relatórios estatísticos combinando os filtros desejados.

Figura 18 – Histórico de recibos

Recibos

ORDENAR > Data ↓ Emissor/receptor Serviço Valor Localização

FILTRAR > Período Tipo Modelo Emissor Receptor Aplicar

MARCAR/DESMARCAR TODOS

Recebidos
Você tem 1 recibo(s) recebido(s)

#	DATA	EMISSOR	SERVIÇO	RECEPTOR	VALOR
3	04/11/2017 às 02:50	Natalia Ferraz Mello (email@hotmail.com)	Serviço de a		R\$200,00
					R\$200,00
					R\$200,00

EMPRESAS

CAAL

JKE Nutricionistas

Brasil LTDA

RECEPTOR

Natalia Ferraz Mello

Gustavo Alves Rodrigues (Você)

EMITIDOS
Você tem 43 recibo(s) emitido(s)

#	DATA	RECEPTOR	SERVIÇO	LOCALIZAÇÃO	VALOR
7	04/11/2017 às 05:17	Natalia Ferraz Mello (email@hotmail.com)	Serviços de fisioterapia.	Alegrete - RS	R\$1.233,00
6	04/11/2017 às 04:54	Natalia (email@host.com) Emitido pela sua empresa Brasil LTDA	Serviços de encanamento residencial.	Alegrete - RS	R\$777,00
4	04/10/2017 às 02:56	Natalia Ferraz Mello (email@hotmail.com)	Serviços de fisioterapia.	Alegrete - RS	R\$21,00
2	04/11/2017 às 02:44	Natalia Ferraz Mello (email@hotmail.com)	Criação de sistema informativo.	Alegrete - RS	R\$123,99
					MÉDIA TOTAL R\$538,74 R\$2.154,99

CANCELADOS
Você tem 1 recibo(s) cancelado(s)

#	DATA	EMISSOR	RECEPTOR	SERVIÇO	VALOR
5	04/11/2017 às 04:07	Gustavo Alves Rodrigues (email@hotmail.com)	Natalia Ferraz Mello (email@hotmail.com)	Serviço de R\$ 915,00	R\$888,00
					MÉDIA TOTAL R\$888,00 R\$888,00

O sub-sistema de notificações pode ser visto na [Figura 19](#). Essas notificações possuem informações sobre *recibos*, notificando o recebimento ou cancelamento de um recibo, *dicas sobre o sistema*, *rendimentos*, informando rendimentos como total emitido na última semana e total gasto no último mês, *avisos*, notificando o usuário de informações ausentes na sua conta, como por exemplo, o [CPF](#), e *lembretes*.

É possível definir quais tipos de notificações o usuário deseja receber e, em alguns tipos, como *rendimentos*, *dicas*, *avisos* e *lembretes*, pode ser setada a frequência de recebimento. Para isso, existe uma sessão *Notificações*, nas configurações de conta, mostrada na [Figura 20](#).

Figura 19 – Sub-sistema de notificações



Figura 20 – Configurações das notificações e e-mails

Notificações

Frequência das notificações Não aplicável a notificações sobre recibos

Uma vez a cada três dias ▼

Tipos de notificações

- Rendimentos
- Avisos do sistema
- Dicas
- Alertas
- Lembretes

Salvar

E-mails

Frequência dos e-mails Não aplicável aos emails sobre recibos

Uma vez por mês ▼

Tipos de e-mail

- Rendimentos
- Avisos do sistema
- Dicas
- Alertas
- Lembretes

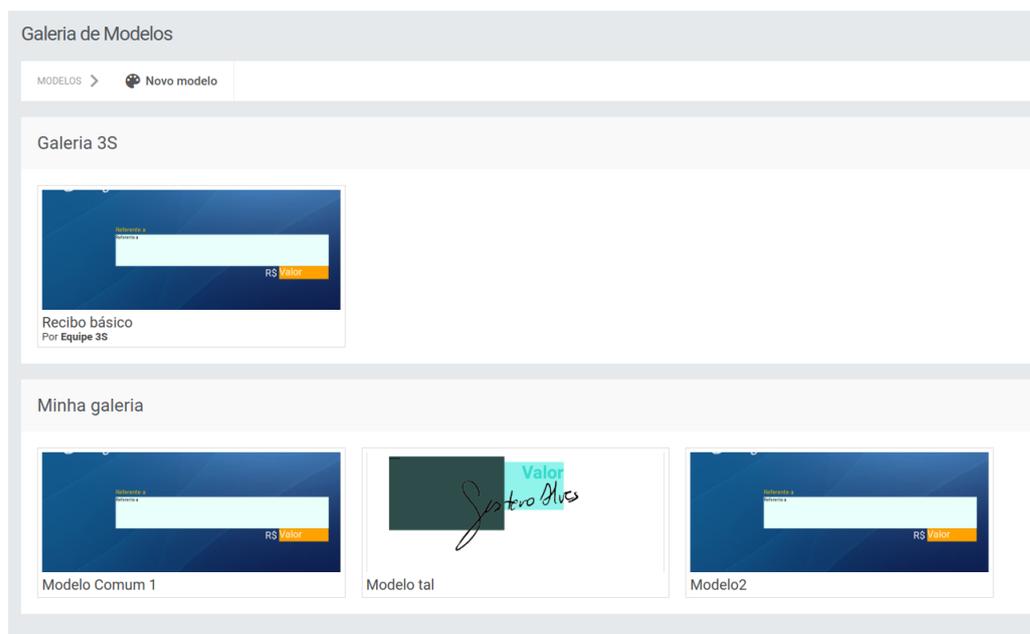
Salvar

Na [Figura 21](#), pode ser vista a Galeria de Modelos. A galeria serve para visualizar modelos a qualquer momento, não somente na hora da emissão. Isso é importante para possibilitar ao usuário conhecer os modelos disponíveis na galeria oficial, visualizar seus modelos e iniciar o [Estúdio de Modelos](#) a partir de um modelo existente. Os modelos da galeria oficial são modelos padrões do sistema, criados pela Equipe 3S ou modelos compartilhados por usuários, que foram aprovados.

Na exibição dos modelos, são exibidos o título, uma imagem miniatura do modelo e, na galeria oficial, o autor.

A galeria está presente também durante a emissão de um recibo. Nesse caso, são exibidos somente os modelos que coincidem com o tipo de recibo emitido, sendo eles, comum ([CPF](#)) ou empresarial ([CNPJ](#)).

Figura 21 – Galeria de Modelos



Os Relatórios Agregados para declaração de [IR](#) são apresentados na [Figura 22](#) e [Figura 23](#). Estes relatórios são baseados no ano base, que é o ano anterior ao ano corrente, e possuem dois tipos: *para emissores* e *para receptores*. Os relatórios para emissores são agregados por mês e por receptor, onde são listados os montantes de cada receptor em cada mês. Isso é importante pois no aplicativo [IRPF](#) existe a opção *incluir novo*, que redefine o formulário para uma nova inclusão no mesmo mês. Os relatórios para receptores são agregados somente por emissor, onde são listados os montantes gerais de cada um deles.

Figura 22 – Relatórios Agregados para emissores

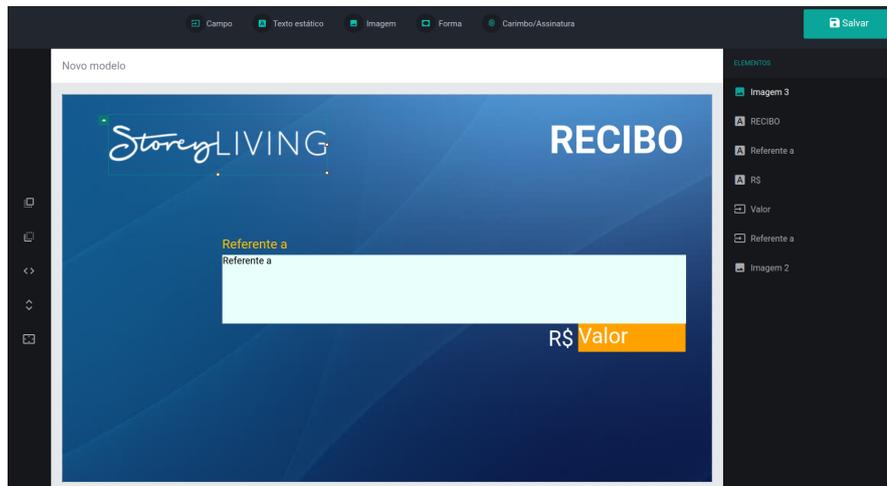
Relatório de Imposto de Renda para emissores (ano base: 2017)		
Outubro		
NOME DO RECEPTOR	CPF	VALOR TOTAL
Natalia Ferraz Mello (emaila@hotmail.com)	03063967009	R\$21,00
		MÉDIA TOTAL R\$21,00 R\$21,00
Novembro		
NOME DO RECEPTOR	CPF	VALOR TOTAL
Natalia Ferraz Mello (emaila@hotmail.com)	03063967009	R\$126.447,99
Usuário teste (teste@ymail.com)	Não informado	R\$12,00
Gustavo Alves (email@host.com)	Não informado	R\$22.377,95
John Lacer (grifindor_rox@hotmail.com)	Não informado	R\$34,00
		MÉDIA TOTAL R\$37.217,99 R\$148.871,94

Figura 23 – Relatórios Agregados para receptores

Relatório de Imposto de Renda para receptores (ano base: 2017)		
NOME DO EMISSOR	CPF	VALOR TOTAL
Gustavo Alves (email@host.com)	Não informado	R\$200,00
		MÉDIA TOTAL R\$200,00 R\$200,00

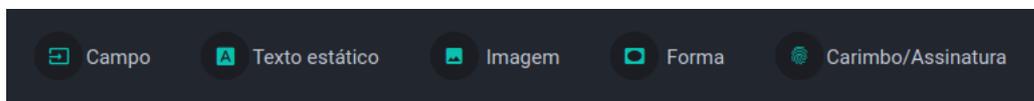
O estúdio de modelos, mostrado na [Figura 24](#), é o ambiente de criação de modelos do sistema.

Figura 24 – Estúdio de Modelos



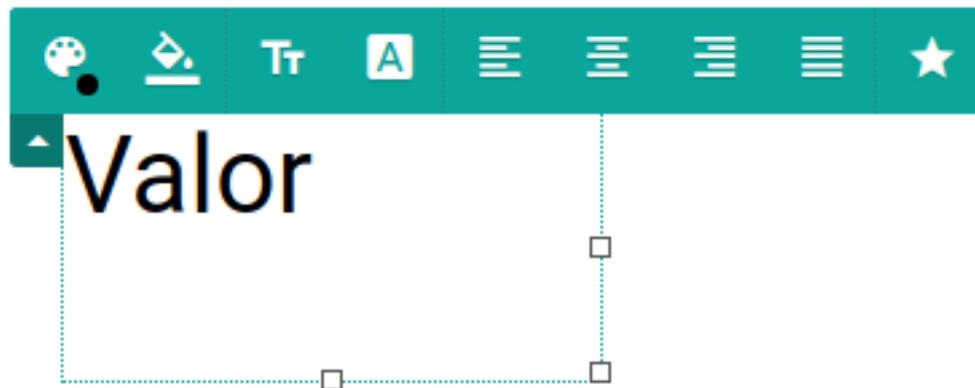
Um modelo pode conter elementos como textos estáticos, retângulos, círculos, imagens e assinaturas/carimbos. Estes elementos são adicionados utilizando o menu superior, mostrado na [Figura 25](#).

Figura 25 – Menu superior do Estúdio de Modelos



Cada elemento deve ter um nome (*label*), definido na sua criação. O estúdio se baseia na funcionalidade *drag-and-drop*, onde cada elemento pode ser arrastado livremente pela área de criação. Existem auxiliares de redimensionamento nas bordas do elemento, que podem ser utilizados para aumentar ou diminuir suas dimensões. Todos os elementos possuem um menu interno, que pode ser visto na [Figura 26](#). Esse menu contém propriedades e opções próprias do elemento.

Figura 26 – Menu interno dos elementos do Estúdio de Modelos



Os elementos do tipo *campo* são aqueles que possuem informações dinâmicas, informadas na finalização de um recibo. Eles possuem propriedades como cor do texto, cor de fundo da caixa, tamanho do texto, espessura do texto e alinhamento do texto. Elementos do tipo campo são separados em três subtipos: *comum*, *valor* e *descritivo*. O subtipo *comum* é opcional. Os subtipos *valor* e *descritivo* são obrigatórios e remetem, respectivamente, ao valor e serviço prestado do recibo.

Elementos do tipo *texto estático* servem para representar informações textuais presentes no modelos, como endereço e etiquetas que identificam campos no modelo. Esses elementos possuem as seguintes propriedades: cor do texto, cor de fundo da caixa, tamanho do texto, espessura do texto e alinhamento do texto na caixa. Nestes elementos também existe a opção para alterar sua informação textual.

Os elementos do tipo *assinatura/carimbo*, descritos no [seção 3.5 da Implementação](#), passam por filtros para remover sua cor de fundo, limitar suas dimensões à área preenchida e suavizar seus traçados. Esses filtros aumentam a qualidade desse elemento para representar carimbos e/ou assinaturas. Os resultados dos filtros iniciais aplicados ao elemento original ([Figura 27](#)) são: correção gamma, escala de cinza, aumento de contraste e aumento de brilho, para aproximar as cores do branco, reduzindo sombras e informações desnecessárias. A aplicação destes filtros resulta em uma imagem com menos borrões/sombras e branco/preto mais definidos ([Figura 28](#)).

Figura 27 – Elemento original

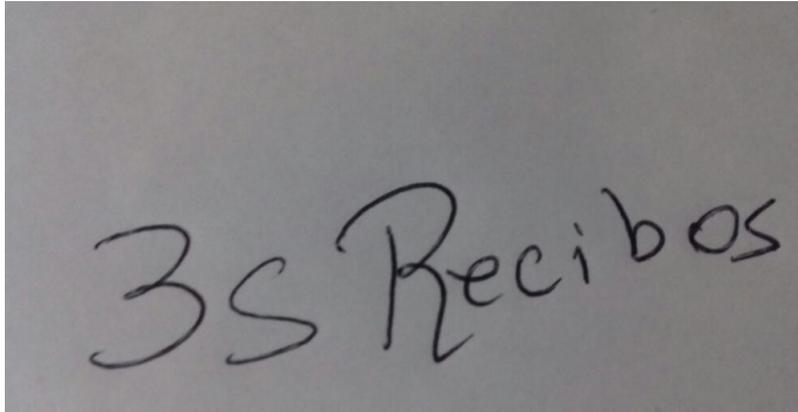
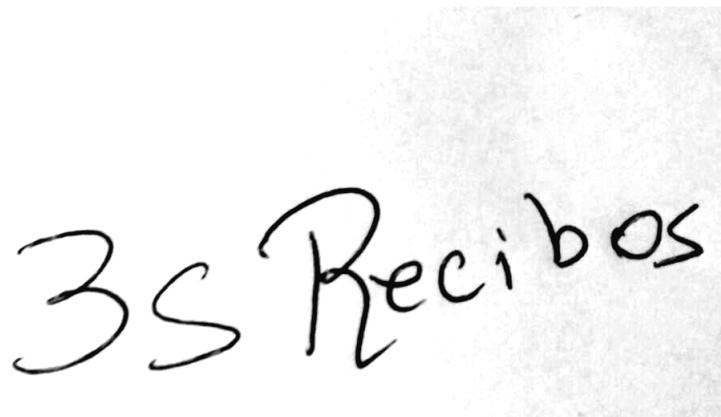


Figura 28 – Filtros iniciais



Após os filtros iniciais, já é possível localizar a assinatura e/ou carimbo e a área clara, sendo assim, são aplicados pares de filtros cinco vezes consecutivas para suavizar a assinatura e/ou carimbo, que são: suavização e contraste (Figura 29). O filtro suavização serve para aumentar a qualidade dos traçados da assinatura e/ou carimbo, aplicando um leve efeito esfumado e diminuindo a definição e retangularidade dos mesmos. Com a aplicação desses filtros, os traçados ficam mais espessos e mais definidos.

Figura 29 – Par de filtros



Após a definição dos traçados, a imagem resultante é cortada para limitar-se a área preenchida (Figura 30), removendo a área clara e, por fim, a imagem passa por mais um filtro onde a cor branca é removida (transformada em transparente), resultando em um carimbo/assinatura com traçado preto e fundo transparente.

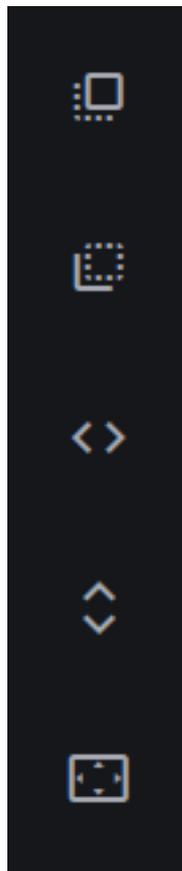
Figura 30 – Filtro: corte



Além dos elementos do tipo *campo*, *texto estático*, *imagem* e *assinatura/carimbo*, existem dois tipos de elementos que representam formas gráficas, os *retângulos* e *círculos*. Estes elementos possuem a propriedade de mudança cor de fundo.

Em cada elemento é possível aplicar propriedades que alteram sua representação dentro do modelo, a partir do menu de opções lateral (Figura 31). Com essas propriedades, é possível mover o elemento uma camada acima, trocando de nível com o primeiro elemento de nível superior, aplicar o efeito contrário, trocando de nível com o primeiro elemento de nível inferior, ajustar a largura do elemento para 100% da largura da área de criação, ajustar a altura do elemento para 100% da altura da área de criação e aplicar ambos os ajustes de largura e altura ao mesmo tempo.

Figura 31 – Menu lateral do Estúdio de Modelos



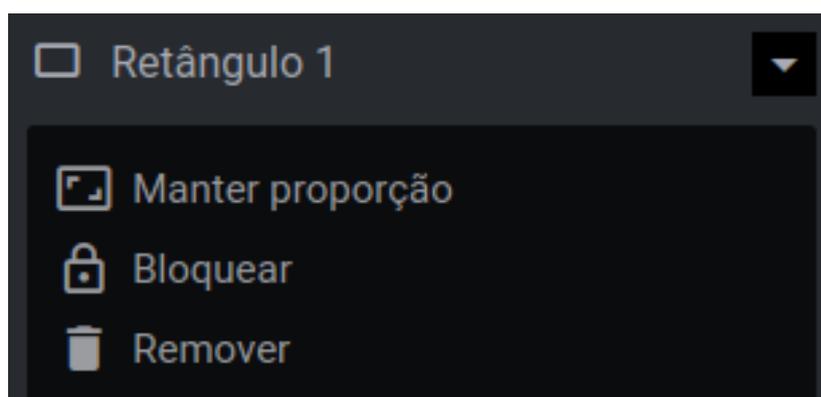
A ordem dos níveis (camadas) dos elementos é mostrada na sessão *elementos*, na lateral direita do estudio de modelos (Figura 24 e Figura 32). Também são mostrados os tipos de cada elemento e seus nomes (*labels*). Campos do tipo *valor* e *descritivo* possuem uma representação abaixo do nome do elemento.

Figura 32 – Menu de camadas do Estúdio de Modelos



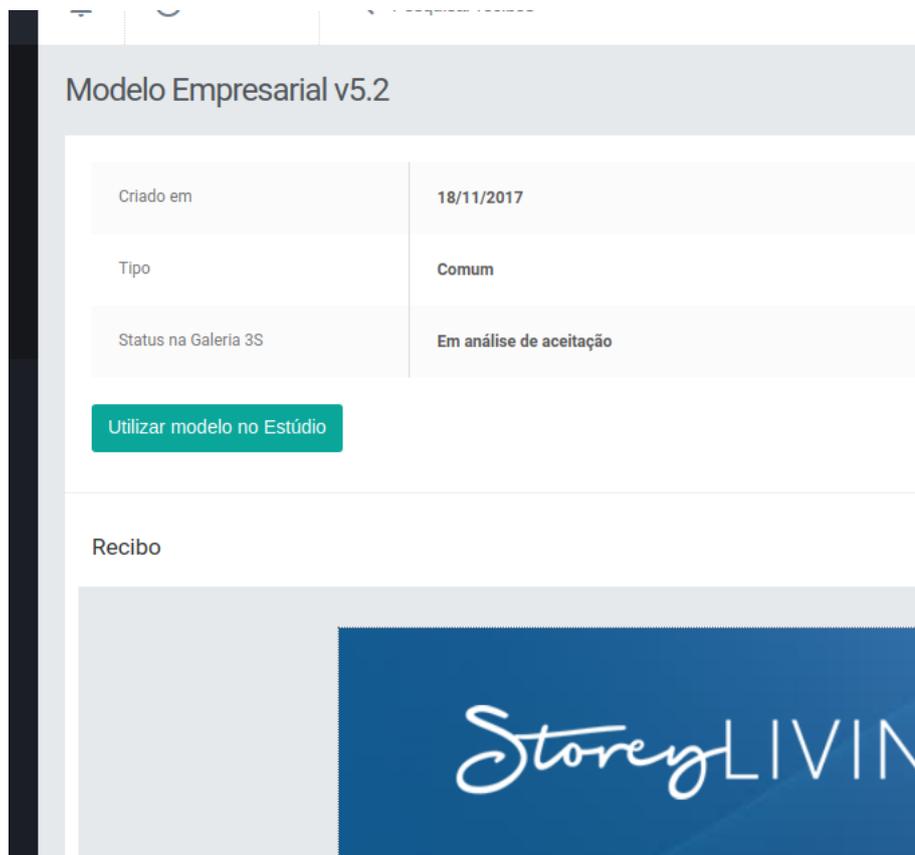
Existe também, nessa sessão, um menu para cada elemento, onde é possível definir opções de controle (Figura 33), sendo elas: */textit*manter proporção, que pode ser ativada ou desativada, mantendo o aspecto dimensional do elemento quando ativada, *bloquear*, que pode ser ativada ou desativada, evitando que o elemento sofra qualquer alteração quando ativada e *remover*, que serve para remover o elemento.

Figura 33 – Menu de controle do Estúdio de Modelos



É possível iniciar o Estúdio de Modelos utilizando um dos modelos disponíveis na galeria oficial. A Figura 34 apresenta a visualização de modelos de recibo.

Figura 34 – Visualização de modelos



A [Figura 35](#) mostra o formulário de registro de novos usuários. Atualmente são necessários o nome completo, [CPF](#), e-mail e senha.

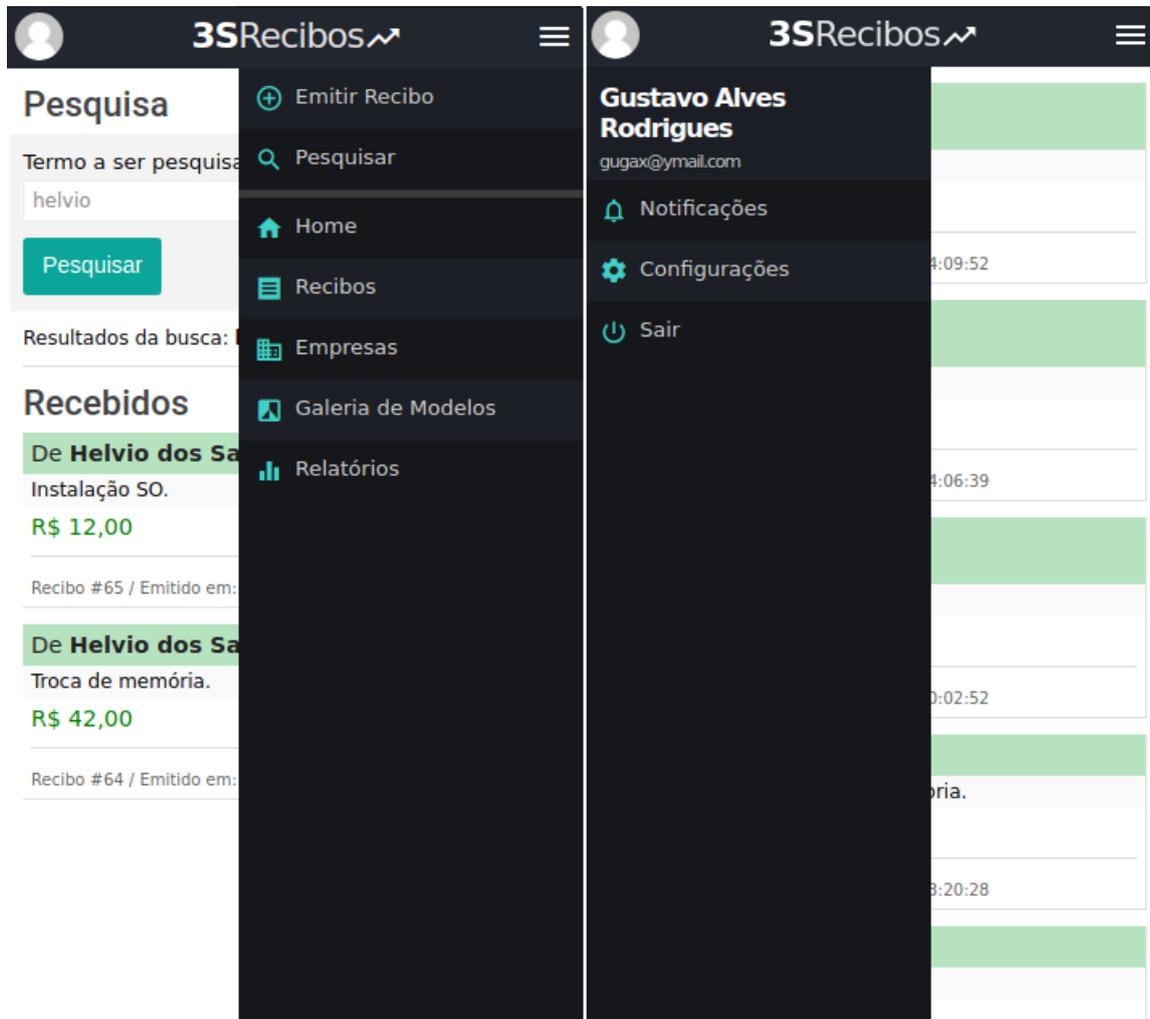
Figura 35 – Registro de novos usuários

The image shows a mobile web application interface for user registration. At the top, there is a teal header with the logo '3SRecibos' and a right-pointing arrow. Below the header, the title 'Registrar-se' is displayed. The form consists of several input fields: 'Nome completo' with an example 'João Paulo da Rosa', 'CPF' with the instruction 'Somente números. ex.: 02918273617', 'E-mail' with an example 'joaopaulo@gmail.com', 'Senha', and 'Confirmação de senha'. At the bottom of the form, there are two buttons: a teal 'Registrar-se' button and a grey 'Voltar' button.

4.3 Aplicativo web para dispositivos móveis

Na [Figura 36](#) pode ser vista a tela inicial e os menus principal e do usuário da aplicação móvel.

Figura 36 – Aplicativo móvel: Menu principal e menu do usuário



As etapas da emissão de recibos para usuários identificados pode ser visualizada na Figura 37.

Figura 37 – Aplicativo móvel: emissão de recibos para usuários identificados

The figure displays four screenshots of the 3SRecibos mobile application, arranged in a 2x2 grid, illustrating the receipt issuance process for identified users.

Top-Left Screenshot: The screen is titled "Emitir recibo". It offers two options: "Emitir como pessoa física" with an "Emitir" button, and "Emitir como pessoa jurídica" with a dropdown menu showing "Brasil LTDA" and an "Emitir" button.

Top-Right Screenshot: The screen is titled "Emitir recibo". It features two input fields for "Nome do receptor" and a "Próximo" button.

Bottom-Left Screenshot: The screen is titled "Emitir recibo". It shows a "Galeria oficial" with a preview of a "Recibo básico" (Por: Equipe 35) and a "Minha galeria" with a preview of a "Modelo Comum 1" (from Starry Living) and a "Próximo" button.

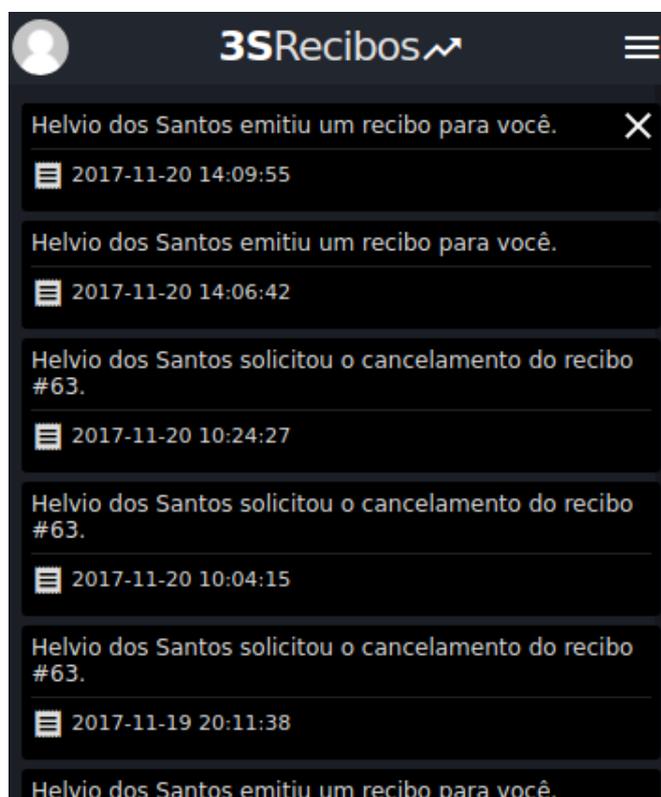
Bottom-Right Screenshot: The screen is titled "Emitir recibo". It features two input fields for "Valor" and "Referente a", and an "Emitir" button.

Na [Figura 38](#) e [Figura 39](#) são apresentadas, respectivamente, a busca de recibos e as notificações do usuário.

Figura 38 – Aplicativo móvel: Busca de Recibos

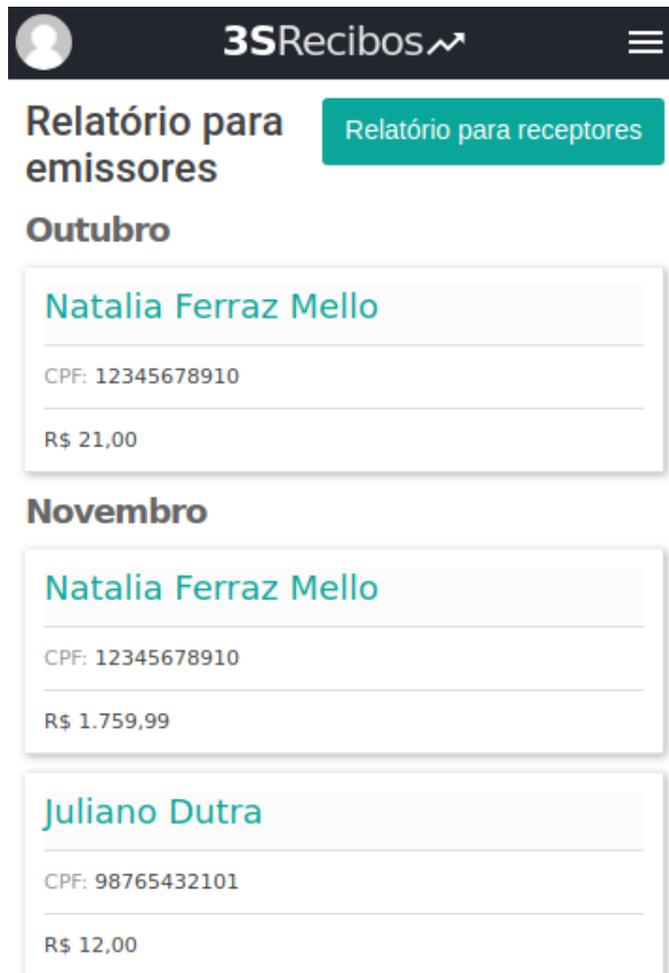


Figura 39 – Aplicativo móvel: Notificações do usuário



Os Relatórios Agregados para declaração de Imposto de Renda podem ser gerados a partir do menu principal, no item *Relatórios*. O relatório para emissores pode ser visto na [Figura 40](#).

Figura 40 – Aplicativo móvel: Relatórios Agregados para emissores



A [Figura 41](#) mostra o relatório para receptores.

Figura 41 – Aplicativo móvel: Relatórios Agregados para receptores



Por fim, a [Figura 42](#) exibe a visualização de um modelo da galeria.

Figura 42 – Aplicativo móvel: Visualização de modelo



5 ESTADO DA ARTE

Utilizando a ferramenta de busca da (GOOGLE, 2017), foram pesquisados, através de termos como *emissão de recibos*, *sistema de recibos*, *recibo digital* e *recibo online*, os sistemas que de alguma forma informatizam o processo de emissão e gestão de recibos. Nas seções a seguir, são introduzidos os principais sistemas encontrados, sendo eles: [Recibo Online](#), [Sem Multa](#), [Controle na Net](#) e [Meu Recibo](#). O capítulo é finalizado com uma [Análise comparativa](#) entre os sistemas existentes e a solução proposta.

5.1 Recibo Online

O Recibo Online (RECIBO ONLINE, 2017) é um sistema que possibilita a emissão de recibos de quatro maneiras: com logo, sem logo, advocatício, e de aluguel. Os recibos com logo caracterizam os documentos com a identificação gráfica do emissor. Já os recibos sem logo possuem as mesmas informações do com logo, com exceção de não ter identificação gráfica. Recibos advocatícios servem para emissão de honorários referentes ao trabalho exercido pelos advogados nas defesas de interesses de um cliente. Os recibos de aluguel servem para atestar o recebimento de uma determinada quantia do locatário referente ao imóvel alugado.

O sistema busca, através dessas quatro maneiras, atender diferentes tipos de prestadores de serviços que necessitam da emissão de recibos.

A [Figura 43](#) ilustra a tela de emissão de recibo com logo do sistema.

Figura 43 – Emissão de recibo com logo no Recibo Online

The screenshot shows the 'RECIBO COM LOGO' interface. At the top, there is a light blue box with a lightbulb icon and text: 'Para gerar o Recibo com Logo em PDF, preencha as informações abaixo e não esqueça de selecionar uma imagem JPG ou PDF em seu computador, preferencialmente a imagem deve ter 125 X 125 de tamanho.' Below this is the 'Dados Emissor' section with fields for 'Nome' (Example: Ricardo Santana), 'Endereço' (Example: Rua do Norte 105), 'Estado' (dropdown: 'Selecione um estado'), 'Cidade' (dropdown: 'Selecione uma cidade'), and 'CEP'. To the right, there is a green 'START DOWNLOAD' button with a download icon, and a section titled 'Find Your Templates Now' with the text 'Share Templates Online or Print them for Free!' and a right arrow. A sidebar menu on the right lists: > BLOG, > HOME, > RECIBOS (with sub-items: > RECIBO COM LOGO, > RECIBO SEM LOGO, > RECIBO ADVOCATÍCIO, > RECIBO DE ALUGUEL), > VALE, > CURRICULO, > DECLARAÇÕES, and > JUSTIÇA GRATUITA.

Fonte: Recibo Online

5.2 Sem Multa

O Sem Multa (BIG LEILÃO, 2017) é um sistema que emite apenas um tipo de recibo, com informações fixas. A Figura 44 apresenta o formulário de emissão de recibos.

Figura 44 – Emissão de recibo no Sem Multa

Os Dados informados serão utilizados somente para a geração do Recibo, sendo descartados automaticamente.

Valor RS:	<input type="text"/>		
Valor (por extenso):	<input type="text"/>		
Referente:	<input type="text"/>		
Nome do Pagador:	<input type="text"/>		
CPF/CNPJ do Pagador:	<input type="text"/>		
Cheque:	Nº: <input type="text"/>	Banco: <input type="text"/>	Agência: <input type="text"/>
	Quando o pagamento for realizado em dinheiro, deixar os campos em branco.		
Local onde está sendo emitido o Recibo (Cidade/Estado):	<input type="text"/>		
Nome do Emitente:	<input type="text"/>		
CPF/CNPJ do Emitente:	<input type="text"/>		
RG do Emitente:	<input type="text"/>		

Fonte: Sem Multa

Em outras palavras, o sistema é limitado a simples emissão de recibos utilizando um único modelo estático.

5.3 Controle na Net

Já o Cotrole na Net (CONTROLE NA NET, 2017) possui, além da emissão, diferentes modelos, registro de clientes, registro de logomarca, relatórios e um histórico de recibos. Antes da emissão do recibo, é necessário cadastrar o cliente no sistema. Isto, por vezes, pode ser algo um pouco burocrático uma vez que o cadastro pode estar automática e transparentemente associado à primeira emissão de um recibo para o referido cliente. Esse sistema possui um limite de 20 emissões diárias no plano gratuito. Os planos pagos somente aumentam o limite de emissões.

A Figura 45 apresenta a tela emissão de recibo empresarial.

Figura 46 – Emissão de recibo sem logo no Meu Recibo

Recibo sem Logo

Dados do Emissor Conforme Cadastro	Dados do Cliente e Recibo
Nome: José da silva	Nome: Arlindo cruz de malta
Endereço: arlindo cruz brasil 200 centro	CPF/CNPJ: 123245346564
Estado: MINAS GERAIS	Número do Recibo 2
Cidade: BELO HORIZONTE	Data: 20/09/2016
CEP: 3059020	Valor: Exemplo R\$ 100,00

Fonte: Meu Recibo

5.5 Análise comparativa

Um sistema de recibos poderia ir além da geração do documento para entrega manual e histórico simples. A possibilidade de não só emitir, mas também receber recibos, através de recursos como e-mail, notificações em uma aplicação móvel ou pelo próprio sistema, tornaria o acesso a esses documentos mais rápido e simples.

Funcionalidades como (a) histórico com opções de filtragem, que além da listagem de recibos, possibilitaria a geração de estatísticas de valores e serviços, (b) galeria e editor de modelos, para a personalização dos documentos, e (c) relatórios de IR, para facilitar as declarações, tornariam o sistema mais prático e completo.

Outra recurso útil para um sistema de recibos é um web service, através do qual seria possível oferecer os serviços de emissão e gestão destes documentos para aplicações externas.

Como pode ser observado na Tabela 1, o sistema proposto apresenta um conjunto de funcionalidades e características que o diferenciam em relação aos demais, como entrega por aplicação móvel, histórico personalizado, modelos personalizados, relatórios para declaração do IR e web service.

Tabela 1 – Comparação dos sistemas de recibos

	Recibo Online	Sem Multa	Controle na Net	Meu Recibo	Sistema proposto
Emissão de recibos digitais	X	X	X	X	X
Envio automático por e-mail				X	X
Aplicação móvel					X
Recebimento de recibos					X
Histórico de recibos emitidos			X	X	X
Histórico personalizado					X
Modelos personalizados					X
Relatórios para declaração do IR					X
Web service para integração de sistemas					X

Por fim, acreditamos que este é um passo importante na evolução de sistemas de emissão e gestão de recibos.

6 CONSIDERAÇÕES FINAIS

O objetivo do trabalho, que era desenvolver um sistema funcional, que facilita a gestão de recibos, através de funcionalidades práticas e rápidas, foi alcançado. Primeiro, foi proposta uma arquitetura inicial para o sistema, e, após isso, um protótipo dessa arquitetura foi implementado. A partir do protótipo, foi possível ter uma noção de como o sistema funcionaria e quais os desafios para chegar aos resultados apresentados.

O principal desafio foi a implementação dos recursos do sistema, visto que a proposta era composta por várias funcionalidades. Em especial, o Estúdio de Modelos foi o recurso mais desafiador, pois o objetivo era implementar um ambiente que possibilita uma personalização detalhada dos recibos, com diferentes tipos de elementos e diferenciados, como assinaturas/carimbos.

O sistema final contribui com uma melhor experiência e organização do usuário com esse tipo de documento. Além disso, outros sistemas, como por exemplo, sistemas de gestão de clínicas odontológicas e sistemas de controle de inscrições em eventos, tem a possibilidade de utilizar essas facilidades através do *web service* desenvolvido. Empresas e consumidores que testaram o sistema demonstraram aceitação quanto a utilização das funcionalidades em seus negócios.

Como possíveis trabalhos futuros, pode-se apontar: a divulgação do sistema, a implementação de uma aplicação móvel nativa para os sistemas operacionais móveis, suporte para escaneamento de assinatura/carimbo diretamente pela aplicação móvel, grades, controles de alinhamento e duplicação de elementos no Estúdio de Modelos, além do *backup* automático dos dados de um modelo em criação, para evitar a perda do trabalho em casos de problemas com o ambiente do usuário.

REFERÊNCIAS

- BIG LEILÃO. **Sem Multa**. 2017. Disponível em: <<http://www.semmulta.com.br/utilitarios/recibo/form-recibo.htm>>. Acesso em: 06 de junho de 2017. Citado na página 66.
- BRITISH COLUMBIA INSTITUTE OF TECHNOLOGY. **CodeIgniter Web Framework**. 2017. Disponível em: <<https://codeigniter.com/>>. Acesso em: 05 de junho de 2017. Citado na página 29.
- CONTROLE NA NET. **Controle na Net**. 2017. Disponível em: <http://www.conteudononet.com.br/controle_recibo/menu_principal.php>. Acesso em: 06 de junho de 2017. Citado na página 66.
- CRO-RN. **Comissão de Ética alerta sobre recibos para declaração de Imposto de Renda**. 2009. Disponível em: <<http://www.cro-rn.org.br/noticias/ver/352>>. Acesso em: 29 de abril de 2017. Citado na página 19.
- DEACON, J. Model-view-controller (mvc) architecture. **Citado em: 10 de março de 2006**. <http://www.jdl.co.uk/briefings/MVC.pdf>, 2005. Citado na página 21.
- DEMATTE. **Dematte**. 2017. Disponível em: <<http://www.dematte.at/>>. Acesso em: 09 de setembro de 2017. Citado na página 33.
- EICH, B. **JavaScript**. 1995. Citado na página 29.
- FREE GEO IP. **Free Geo IP**. 2017. Disponível em: <<http://freegeoip.net/>>. Acesso em: 09 de agosto de 2017. Citado na página 32.
- GANDINI, J. A. D.; SALOMÃO, D. P. d. S.; JACOB, C. A segurança dos documentos digitais. **Disponível em** <<http://www.jus.com.br>>. **Acesso em: Abril**, 2001. Citado na página 19.
- GITLAB COMMUNITY. **GitLab**. 2017. Disponível em: <<https://gitlab.com/>>. Acesso em: 05 de junho de 2017. Citado na página 29.
- GOOGLE. **Buscador da Google**. 2017. Disponível em: <<http://www.google.com>>. Acesso em: 10 de agosto de 2016. Citado na página 65.
- IBM. **RESTful Web services: The basics**. 2008. Disponível em: <<https://www.ibm.com/developerworks/library/ws-restful/index.html>>. Acesso em: 05 de junho de 2017. Citado na página 26.
- MEU RECIBO. **Meu Recibo**. 2017. Disponível em: <<http://meurecibo.com/>>. Acesso em: 06 de junho de 2017. Citado na página 67.
- MINISTÉRIO DA FAZENDA. **Saiba a diferença entre malha fiscal e malha débito**. 2015. Disponível em: <<http://portal.ouvidoria.fazenda.gov.br/mnoticias/saiba-a-diferenca-entre-malha-fiscal-e-malha-debito>>. Acesso em: 29 de abril de 2017. Citado na página 19.
- NGUYEN, D. L. Digital receipt system using mobile device technologies. **Disponível em** <<http://scholarworks.uno.edu/td/705>>. **Acesso em Junho**, 2008. Citado na página 26.

- NIKLAS VON HERTZEN. **HTML2CANVAS**. 2017. Disponível em: <<https://html2canvas.hertzen.com/>>. Acesso em: 10 de outubro de 2017. Citado na página 34.
- ORACLE CORPORATION. **MySQL**. 2017. Disponível em: <<https://www.mysql.com/>>. Acesso em: 05 de junho de 2017. Citado na página 29.
- PHP GROUP. **PHP: Hypertext Preprocessor**. 2017. Disponível em: <<http://php.net/>>. Acesso em: 20 de junho de 2017. Citado na página 29.
- RECEITA FEDERAL. **Ano do exercício a que corresponde a Declaração Anual do Imposto de Renda Pessoa Física**. 2017. Disponível em: <http://www31.receita.fazenda.gov.br/pirgateway/calcpfExerc_cio.asp>. Acesso em: 02 de junho de 2017. Citado na página 24.
- RECEITA FEDERAL. **Imposto de Renda Pessoa Física 2017**. 2017. Disponível em: <<https://idg.receita.fazenda.gov.br/interface/cidadao/irpf/2017/>>. Acesso em: 02 de junho de 2017. Citado na página 24.
- RECEITA FEDERAL. **Imposto sobre a renda – pessoa física: perguntas e respostas**. 2017. Disponível em: <<http://idg.receita.fazenda.gov.br/interface/cidadao/irpf/2017/perguntao/pir-pf-2017-perguntas-e-respostas-versao-1-1-03032017.pdf>>. Acesso em: 16 de junho de 2017. Citado na página 19.
- RECIBO ONLINE. **Recibo Online**. 2017. Disponível em: <<http://reciboonline.com/>>. Acesso em: 06 de junho de 2017. Citado na página 65.
- RESCORLA, E. Http over tls. 2000. Citado na página 26.
- RODRIGUES, C.; AFONSO, J.; TOMÉ, P. Mobile application webservice performance analysis: Restful services with json and xml. In: _____. **ENTERprise Information Systems: International Conference, CENTERIS 2011, Vilamoura, Portugal, October 5-7, 2011, Proceedings, Part II**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. p. 162–169. ISBN 978-3-642-24355-4. Disponível em: <http://dx.doi.org/10.1007/978-3-642-24355-4_17>. Citado na página 26.
- SOFTWARE FREEDOM CONSERVANCY. **Git**. 2017. Disponível em: <<https://git-scm.com/>>. Acesso em: 05 de junho de 2017. Citado na página 29.
- STAPLES. **The Pros and Cons of Digital vs. Printed Documents**. 2017. Disponível em: <<https://www.staples.com/sbd/cre/tech-services/explore-tips-and-advice/tech-articles/the-pros-and-cons-of-digital-vs-printed-documents.html>>. Acesso em: 23 de abril de 2017. Citado na página 19.
- THE APACHE SOFTWARE FOUNDATION. **Apache Software Foundation**. 2017. Disponível em: <<https://www.apache.org/>>. Acesso em: 05 de junho de 2017. Citado na página 29.
- THE JQUERY FOUNDATION. **jQuery**. 2017. Disponível em: <<https://jquery.com/>>. Acesso em: 05 de junho de 2017. Citado na página 29.
- VADDE, V.; H., N. C.; SURHONNE, A. P. An nfc based innovation for paperless retail transactions and digital receipts management. In: **2015 Annual IEEE India Conference (INDICON)**. [S.l.: s.n.], 2015. p. 1–6. Citado na página 19.

W3C. **CSS 3**. 2017. Disponível em: <<https://www.w3.org/>>. Acesso em: 05 de junho de 2017. Citado na página 29.

W3C. **HTML 5**. 2017. Disponível em: <<https://www.w3.org/>>. Acesso em: 05 de junho de 2017. Citado na página 29.

W3C. **SQL-Injection**. 2017. Disponível em: <<https://www.w3.org/>>. Acesso em: 05 de setembro de 2017. Citado na página 31.

WILLINS, B. **Secure electronic receipt systems and methods**. Google Patents, 2011. US Patent App. 12/630,215. Disponível em: <<https://www.google.com/patents/US20110137803>>. Citado na página 19.

Apêndices

APÊNDICE A – PRIMEIRO APÊNDICE

Tabela 2 – Requisições do Web Service

Método	Tipo	Descrição
.../api/empresa	GET	Lista todas empresas do usuario
.../api/empresa/id	GET	Informações da empresa id do usuário
.../api/empresa	POST	Cria nova empresa (dados enviados no corpo da requisição)
.../api/notificacao	GET	Todas as notificações do usuário
.../api/notificacao/lidas	GET	Todas notificações lidas do usuário
.../api/notificacao/naolidas	GET	Todas notificações não lidas do usuário
.../api/notificacao/lidas	PUT	Seta todas as notificações do usuário para lidas
.../api/notificacao/naolidas	PUT	Seta todas as notificações do usuário para não lidas
.../api/recibo/id	GET	Informações do recibo id do usuário
.../api/recibo/id/campos	GET	Informações dos valores dos campos do modelo do recibo id do usuário
.../api/recibo/id/cancelamento	GET	Informações sobre o cancelamento do recibo id do usuário
.../api/recibo/token/{token}	GET	Informações sobre o recibo da token {token}
.../api/recibo/emitidos	GET	Lista dos recibos emitidos do usuário
.../api/recibo/recebidos	GET	Lista dos recibos recebidos do usuário
.../api/recibo/cancelados	GET	Lista dos recibos cancelados do usuário
.../api/recibo/id/cancelamento	POST	Solicitar cancelamento do recibo id pelo usuário
.../api/recibo/	POST	Emite novo recibo (dados do recibo via corpo da requisição)
.../api/recibo/id/cancelamento	PUT	Confirmar cancelamento do recibo id pelo usuário
.../api/recibo/id/cancelamento	DELETE	Cancelar ou rejeitar o cancelamento do recibo id pelo usuário
.../api/modelo/id	GET	Informações do modelo id
.../api/modelo/id/campos	GET	Informações dos campos do modelo id
.../api/modelo/id/campos/dados	GET	Informações dos campos de entrada de dados do modelo id
.../api/modelo/meus	GET	Lista de todos os modelos do usuário
.../api/modelo/oficiais	GET	Lista de todos os modelos da galeria oficial
.../api/modelo/tipo/cpf	GET	Lista de todos os modelos (usuário e oficiais) do tipo comum (CPF)
.../api/modelo/tipo/cnpj	GET	Lista de todos os modelos (usuário e oficiais) do tipo empresarial (CNPJ)
.../api/modelo/todos	GET	Lista de todos os modelos (da galeria oficial e do usuario)
.../api/usuario	GET	Informações sobre a conta do usuário logado
.../api/usuario	PUT	Atualiza informações do usuário logado (dados enviados no corpo da requisição)
.../api/usuario	POST	Registro de novo usuário (dados enviados no corpo da requisição)
.../api/pesquisa/termo_tal	GET	Informações sobre recibos que foram encontrados na pesquisa com o termo termo_tal
.../api/relatorio/ir/receptor	GET	Informações do relatório agregado para IR do usuário como receptor
.../api/relatorio/ir/emissor	GET	Informações do relatório agregado para IR do usuário como emissor
.../api/login	POST	Autentica o usuário e retorna o novo api_token (dados enviados no corpo da requisição)
.../api/token	POST	Redefine a token de requisição

APÊNDICE B – SEGUNDO APÊNDICE

Tabela 3 – Respostas das requisições do Web Service

<pre> .../api/empresa GET { data: { {id: id1, cnpj: cnpj1, email: email1, nome: nome1, usuario: id_usuario}, {id: id2, cnpj: cnpj2, email: email2, nome: nome2, usuario: id_usuario}, {id: id3, cnpj: cnpj3, email: email3, nome: nome3, usuario: id_usuario} ... }, status: 200 } </pre>
<pre> .../api/empresa/id GET { data: {id: id, cnpj: cnpj, email: email, nome: nome, usuario: id_usuario}, status: 200 } </pre>
<pre> .../api/empresa POST { data: {new_id: id}, status: 200 } </pre>
<pre> .../api/notificacao GET .../api/notificacao/lidas GET .../api/notificacao/naolidas GET { data: { {id: id1, tipo: tipo1, texto: texto1, link: link1, lida: lida1, data: data1}, {id: id2, tipo: tipo2, texto: texto2, link: link2, lida: lida2, data: data2}, {id: id3, tipo: tipo3, texto: texto3, link: link3, lida: lida3, data: data3} ... }, status: 200 } </pre>
<pre> .../api/recibo/id GET { data: {id: id, modelo: modelo_id, valor: valor1, servico: servico1, data: data1, receptor: receptor_id, emissor: emissor_id, localizacao: localizacao1}, status: 200 } </pre>

Tabela 4 – Respostas das requisições do Web Service (2)

<pre> .../api/recibo/id/campos GET { data: { {id: id1, label: label1, c_valor: c_valor1, c_descritivo: c_descritivo1, valor: valor1}, {id: id2, label: label2, c_valor: c_valor2, c_descritivo: c_descritivo2, valor: valor2}, {id: id3, label: label3, c_valor: c_valor3, c_descritivo: c_descritivo3, valor: valor3} ... }, status: 200 } </pre>
<pre> .../api/recibo/id/cancelamento GET { data: {id: id1, solicitante: id_solicitante, recibo: id, status: status, data: data}, status: 200 } </pre>
<pre> .../api/recibo/token/{token} GET { data: {id: id, modelo: modelo_id, valor: valor1, servico: servico1, data: data1, receptor: receptor_id, emissor: emissor_id, localizacao: localizacao1}, status: 200 } </pre>
<pre> .../api/recibo/emitidos GET .../api/recibo/recebidos GET .../api/recibo/cancelados GET { data: { {id: id1, modelo: modelo_id, valor: valor1, servico: servico1, data: data1, receptor: receptor_id, emissor: emissor_id, localizacao: localizacao1}, {id: id2, modelo: modelo_id, valor: valor2, servico: servico2, data: data2, receptor: receptor_id, emissor: emissor_id, localizacao: localizacao2}, {id: id3, modelo: modelo_id, valor: valor3, servico: servico3, data: data3, receptor: receptor_id, emissor: emissor_id, localizacao: localizacao3} ... }, status: 200 } </pre>

Tabela 5 – Respostas das requisições do Web Service (3)

<pre> .../api/modelo/id GET { data: {tipo: tipo1, nome: nome1, modelo_global: modelo_global1, size_x: size_x1, size_y: size_y1, data: data}, status: 200 } </pre>
<pre> .../api/modelo/id/campos GET { data: { {id: id1, label: label1, c_valor: c_valor1, c_descritivo: c_descritivo1, valor: valor1}, {id: id2, label: label2, c_valor: c_valor2, c_descritivo: c_descritivo2, valor: valor2}, {id: id3, label: label3, c_valor: c_valor3, c_descritivo: c_descritivo3, valor: valor3} ... }, status: 200 } </pre>
<pre> .../api/modelo/id/campos/dados GET { data: { {id: id1, label: label1, c_descritivo: c_descritivo1, c_valor: c_valor1}, {id: id2, label: label2, c_descritivo: c_descritivo2, c_valor: c_valor2}, {id: id3, label: label3, c_descritivo: c_descritivo3, c_valor: c_valor3} ... }, status: 200 } </pre>
<pre> .../api/modelo/meus GET .../api/modelo/oficiais GET .../api/modelo/tipo/cpf GET .../api/modelo/tipo/cnpj GET .../api/modelo/todos GET { data: { {tipo: tipo1, nome: nome1, modelo_global: modelo_global1, size_x: size_x1, size_y: size_y1, data: data}, {tipo: tipo2, nome: nome2, modelo_globo2: modelo_global2, size_x: size_x2, size_y: size_y2, data: data}, {tipo: tipo3, nome: nome3, modelo_globo3: modelo_global3, size_x: size_x3, size_y: size_y3, data: data} ... }, status: 200 } </pre>

Tabela 6 – Respostas das requisições do Web Service (4)

<pre> .../api/usuario GET { data: {id: id, nome: nome, email: email, cpf: cpf, s_token: s_token}, status: 200 } </pre>
<pre> .../api/pesquisa/termo_tal GET { data: { {id: id1, modelo: modelo_id, valor: valor1, servico: servico1, data: data1, receptor: receptor_id, emissor: emissor_id, localizacao: localizacao1}, {id: id2, modelo: modelo_id, valor: valor2, servico: servico2, data: data2, receptor: receptor_id, emissor: emissor_id, localizacao: localizacao2}, {id: id3, modelo: modelo_id, valor: valor3, servico: servico3, data: data3, receptor: receptor_id, emissor: emissor_id, localizacao: localizacao3} ..., }, status: 200 } </pre>
<pre> .../api/relatorio/ir/receptor GET { data: { {cpf: cpf1, nome: nome1, email: email1, total: total1}, {cpf: cpf2, nome: nome2, email: email2, total: total2}, {cpf: cpf3, nome: nome3, email: email3, total: total3} ... }, status: 200 } </pre>
<pre> .../api/relatorio/ir/emissor GET { data: { {mes: mes1, cpf: cpf1, nome: nome1, email: email1, total: total1}, {mes: mes2, cpf: cpf2, nome: nome2, email: email2, total: total2}, {mes: mes3, cpf: cpf3, nome: nome3, email: email3, total: total3} ... }, status: 200 } </pre>

Tabela 7 – Respostas das requisições do Web Service (5)

<pre>.../api/login POST { data: {token: token, userid: id, username: username, message: "Sucesso"}, status: 200 }</pre>
<pre>.../api/notificacao/lidas PUT .../api/notificacao/naolidas PUT .../api/recibo/id/cancelamento POST .../api/recibo/id/cancelamento PUT .../api/recibo/id/cancelamento DELETE .../api/usuario PUT .../api/usuario POST .../api/token POST { data: "Sucesso", status: 200 }</pre>
<pre>Mensagens de erro gerais { data: "Erro", status: 400/401/404/500 }</pre>