

Melhoria de Processo na captação de recursos para Instituição do 3º setor com Implementação baseada na Arquitetura REST

Dione A. Faria, Fábio Mendes Vieira de Almeida, Renato Cardoso

Instituto de Informática – Centro Universitário do Triângulo (UNITRI)
Caixa Postal 309 – 38.411-106 – Uberlândia – MG – Brasil

{dione_castell,fabio_mendesvieira,renatotio21}@hotmail.com

Resumo. *O terceiro setor tem desempenhado um papel muito importante na sociedade, embora seu intuito não seja de lucratividade, as entidades filantrópicas necessitam de recursos financeiros para se manterem ativas e desempenharem seu papel social. Grandes processos realizados internamente nessas instituições não têm aproveitamento adequado para a demanda. Através de um estudo empírico, foi possível aplicar os conceitos vivenciados durante o curso de sistemas de informação, identificando problemas na busca de recursos na instituição Grupo Luta Pela Vida, e propor otimizar as atividades existentes através de uma aplicação Web e uma aplicação Mobile, ambas para gerenciar os processos de arrecadação das doações.*

1. Introdução

Com o intuito de proporcionar uma melhor qualidade de vida às pessoas e aos meios em que estão envolvidos, o terceiro setor desempenha um papel muito importante na sociedade. Quando o assunto é o terceiro setor, logo é preciso entender que são instituições filantrópicas sem fins lucrativos, onde seu papel é captar recursos para que possam exercer suas atividades e desempenhar seu trabalho na sociedade, porém, para muitas instituições se torna complexo o processo das arrecadações, embora existem vários meios para a captação desses recursos [BAZOLI 2009].

Para que as instituições possam permanecer ativas e realizar todo o propósito relacionado à sua criação, as buscas por parceiros e recurso são feitas arduamente para manter-se em funcionamento. Sabendo que a informação cresce a partir que esta é utilizada no processo decisório, faz necessário que as organizações promovam esforços no sentido de desenvolver sistemas para auxílio em seus recebimentos.

Ao iniciar os estudos de sistemas já existentes e as tecnologias já utilizadas para o terceiro setor, o presente projeto se iniciou na intenção de melhorar os processos de captação de recursos para instituição Grupo Luta Pela Vida, onde a parte do processo da geração dos recibos das arrecadações, são feitas manualmente, demandando tempo e recursos. Preocupados com meio ambiente e redução de gastos da instituição, foi desenvolvido uma aplicação Web e uma *Mobile* para o controle dos dados obtidos na captação dos recursos.

Nos próximos tópicos serão apresentados os conceitos sob uma visão ampla do problema realmente encontrado e a busca para sua solução utilizando-se de tecnologias avançadas e meios para garantir que todo o processo de captação de recursos seja mais seguro, ágil e contribua para os fins da instituição.

2. Melhoria de processos e arquitetura de software

A melhoria de processos consiste na busca de meios e/ou alternativas para que possam através delas, melhorar e otimizar processos existentes dentro de uma organização/empresa, de modo que facilite e organize as atividades existentes de forma clara e objetiva. Para que isto ocorra, é necessário entender como é o funcionamento de uma organização como um todo, de maneira a conhecer seus objetivos e a própria estrutura organizacional. Logo após o entendimento de suas obrigações de modo geral, é possível entender e encontrar os possíveis problemas que possam estar causando gargalos, contribuindo de forma negativa com um grande acúmulo de atividades e consequentemente um atraso significativo [DAVENPORT 1994].

Dentre todos os pontos negativos encontrados durante o período de análise dos processos existentes, a melhoria de processos vem com o intuito de identificar as possíveis causas e definir os meios aos quais serão utilizados para que os problemas possam ser corrigidos, modelando novamente para que todo o processo seja otimizado. Para isso são utilizadas ferramentas para melhoria de processos tais como podem ser citadas o BPM(*Business Process Management*) que no português significa gestão de processos de negócios, onde são construídos fluxogramas e/ou diagramas de processos para identificar os melhores caminhos e atividades a serem executadas, seu intuito é tornar todo o processo existente claro e objetivo agregando valor ao cliente, usando os mínimos recursos existentes mas que não afete a qualidade do produto ou serviço [IPROCESS 2018].

Pode-se destacar também a matriz BÁSICO, levando em consideração alguns pontos bastante importantes que podem ser destacados da seguinte maneira: benefícios para a empresa, quer dizer que em relação a otimização do processo haja redução de custos, otimização na transição de informações entre outros. Abrangência, onde se verifica de modo quantitativo quantos funcionários terão seu trabalho facilitado com essa melhoria, satisfação do cliente onde será analisado se as equipes de trabalho ficarão satisfeitas ou não com a implementação, investimentos necessários, quer dizer o que a empresa terá que investir para conseguir tal melhoria, cliente externo satisfeito, o quanto seu produto ou serviço está agregando valor aos clientes, e por fim operação simples, quer dizer se esta melhoria será ou não difícil de ser implementada, levando em consideração os processos já existentes.

O que é realmente importante de citar é que existem inúmeras ferramentas para se utilizar na melhoria de processos, basta apenas identificar a qual melhor se adapta as suas preferências e aos fins que necessita. No decorrer do artigo será apresentado alguns processos em relação a captação de recursos em uma Instituição do terceiro Setor, a qual é a Instituição Grupo Luta Pela Vida, onde foram avaliados os processos, identificados os problemas, uma proposta como solução e consequentemente será apresentado o novo processo, após a definição e implementação das melhorias.

A complexidade na utilização de sistemas de software tem crescido arduamente, tornando-se mais que necessário obter-se de meios para garantir a confiabilidade e integrabilidade das informações que estão sendo manipuladas. A engenharia de software surgiu para resolver esses impasses, criando técnicas chamadas de abstração para que fosse possível acompanhar esse grande aumento da complexidade, não baseando somente nos algoritmos ou formas que estão sendo utilizados para o desenvolvimento,

mas também preocupando com a estrutura global do projeto que se tornou mais que imprescindível ser criada antes de iniciá-lo [FALBO 2005].

A função da arquitetura de software é explorar e analisar as características de cada subsistema, proporcionando outras vantagens como desenvolver outras linhas de produto cujo podem ser criados diversos sistemas com funcionalidades distintas. Entre outras palavras pode se dizer que a arquitetura de software se trata de um estudo da organização de um sistema de maneira global e das inter-relações de seus subsistemas como citado anteriormente [MENDES 2002]. Nos seguintes tópicos decorrentes, será apresentado os princípios e tecnologias utilizados na implementação da solução proposta, contendo todo referencial teórico para um bom entendimento dos assuntos abordados durante o artigo.

3. Técnicas e Tecnologias

Esse tópico tem como objetivo demonstrar ao leitor de forma geral as tecnologias utilizadas no desenvolvimento do projeto.

3.1 Tecnologias Lado-Servidor (*Server-Side*)

Para a construção do nosso aplicativo *server side* foi desenvolvido seguindo alguns preceitos da arquitetura REST, que pode ser entendida como um estilo arquitetônico, algo que seja abstrato, esse estilo define os atributos e/ou características que implementam um conjunto de restrições que buscam amenizar a latência e comunicações em rede. Esse estilo arquitetônico possui vários princípios que definem bem sua arquitetura, dentre eles recursos que podem ser identificados em cada requisição, como se tivesse descrevendo seu serviço através de uma URL (*Uniform Resource Locator*), como, por exemplo, <http://nomedoservidor/usuario/10>, seja em questão uma pesquisa por determinado usuário ou a exclusão do mesmo, irá depender do método que estiver utilizando no serviço.

Tal arquitetura possui vários métodos que implicam em ações baseadas em cada recurso fornecido, utiliza-se seis dos nove métodos existentes do protocolo HTTP (*HyperText Transfer Protocol*), porém os mais utilizados são os métodos *GET* para obter a representação de um recurso, *POST* para criar um novo recurso, *PUT* para atualizar ou criar um novo recurso e *DELETE* para remover um recurso existente. Cada requisição é composta por recursos que são identificados em qual formato se refere ao conteúdo enviado, normalmente esse formato é identificado no cabeçalho da requisição por um campo chamado *Content-type*, que se tratando da arquitetura REST utiliza-se o formato JSON (*JavaScript Object Notation*) e XML (*Extensible Markup Language*), não se limitando somente a estes citados. Outro ponto bastante importante do estilo arquitetural REST é o desenvolvimento por camadas, cada camada de serviços seja de componentes ou de acesso a dados apenas tem visibilidade de sua camada imediata, o principal motivo desse princípio é garantir que as camadas não tenham sobrecarga [FILDER 2008].

Em contrapartida, uma das desvantagens de um sistema em camadas é o retardamento das informações de um extremo ao outro, porém em se tratando da arquitetura REST logo é contornado através do compartilhamento de cache entre as camadas, permitindo que uma requisição já feita possa ser reaproveitada sem precisar

fazer todo o percurso anteriormente [TOBALDINI 2008].

Um nível bastante importante de se mencionar é o HATEOAS (*Hypermedia As The Engine Of Application State*), no português significa hipermídia como o mecanismo de estado do aplicativo, que também é uma restrição da arquitetura REST, que trata de informar como está o estado da aplicação, fornecendo informações para navegar dinamicamente incluindo os links hipermídia para as respostas.

Na figura 1, uma reposta baseada em HATEOAS, contendo não somente os dados relacionados ao funcionário, mas também a url onde ele está localizado.

```
{
  "idFunc" : 1,
  "nome" : "ADRIEL FERNANDES",
  "endereço" : "RUA ALVARO LENZA",
  "numero" : "39"
  "links" : [
    {
      "rel" : "auto",
      "href" : "http://"localServidor"/quickaction/funcionarios/1"
    }
  ]
}
```

Figura 1: Resposta HATEOAS.
Fonte: Elaborado pelo Autor

É possível construir relacionamentos mais complexos, HATEOAS facilita a forma de comunicar-se com outro serviço sem precisar procurar outro arquivo externo, mostrando todas as urls de cada recurso, deixando clara a forma de acessar esses recursos. É considerado o nível final do REST, pois cada link é baseado nos verbos HTTP padrão *GET*, *POST*, *PUT* e *DELETE*. Fornecendo assim as informações necessárias para utilizar-se dos serviços disponíveis [SPRING 2018].

Também e não menos importante, foi utilizado o Spring que para Afonso (2017, p.12), o *Spring* não se trata somente de um framework, mas sim de um conjunto de projetos que resolve várias situações do cotidiano, ajudando o programador a desenvolver aplicações Java com mais simplicidade e flexibilidade. Existem muitas áreas cobertas pelo *Spring*, como por exemplo, o *Spring Data* utilizado para acesso a banco de dados, o *Spring Security* para prover segurança, entre outros.

O próprio framework *Spring* nada mais é do que a base para todo o projeto utilizando *Spring*, desenvolvido principalmente para que as aplicações pudessem focar mais na regra de negócio do que na própria infraestrutura. O *Spring Boot* facilita muito todo o andamento do projeto a ser desenvolvido, justamente pelo fato de trazer várias configurações já prontas, como por exemplo, é só você dizer que módulos irá utilizar (*Web*, *template*, entre outros), que ele vai reconhecer e configurar, basta escolher os modelos e incluir no *pom.xml*, que trata-se de um local onde são instanciadas todas as dependências do projeto [AFONSO 2017].

Como a solução proposta esta trabalhando com informações que desrespeito somente ao Grupo Luta Pela Vida, veio a preocupação de garantir a integridade dos dados que seriam manipulados pela aplicação. Devido a este cenário, veio a necessidade

de construir uma segurança robusta com controle de todas as extremidades da aplicação. A seguir será apresentado as principais configurações de segurança desenvolvidas no presente projeto, a qual se refere a utilização do protocolo OAuth2.

A especificação *OAuth 2.0* define que é um protocolo de delegação útil para transferir decisões de autorização por meio de *tokens* aos aplicativos de terceiros sejam capazes de ter permissão a recursos em um serviço HTTP sem expor credenciais do usuário [OAUTH, 2018]. O protocolo se torna mais fácil quando você conhece as entidades envolvidas. Basicamente, *OAuth* especifica quatro entidades como ilustrado na figura 2.

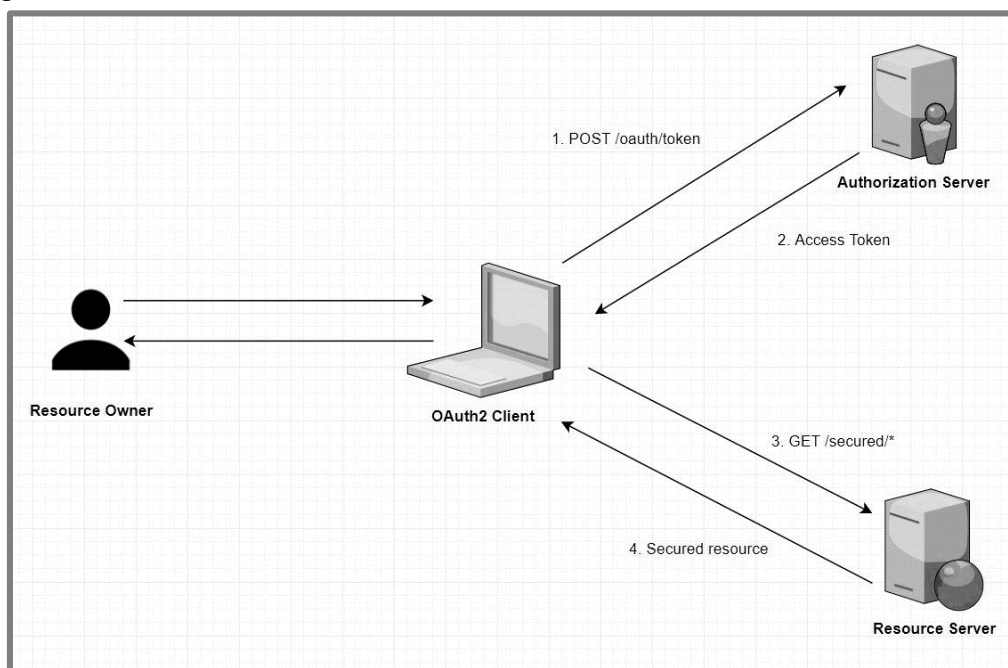


Figura 2: Fluxo concessão por senha.

Fonte: Elaborado pelo Autor

- Dono do Recurso (*Resource Owner*): é a entidade capaz de conceder acesso a um recurso protegido (por exemplo, usuário final).
- Servidor do Recurso (*Resource server*): Servidor da aplicação REST que disponibiliza os recursos protegidos, recebe e responde às requisições de acesso aos recursos com os *tokens* do protocolo;
- Cliente (*Client*): O cliente se refere ao aplicativo que deseja acessar as credencias do usuário
- Servidor de Autorização: Servidor que verifica a identidade do usuário e disponibiliza o *token* de acesso após verificar ao cliente.

O sistema de autorização fornece vários tipos de concessão para diferentes casos de uso. Alguns tipos definidos são:

- Código de autorização
- Senha
- Credenciais do Cliente
- *Refresh Token*

O JWT (*JSON Web Token*) é um padrão aberto que define um sistema de transferência de informações segura, compacta e leve que pode ser enviada em um

cabeçalho HTTP (*Header*), ou como parâmetro no *Body* (corpo da requisição) entre duas extremidades como um objeto JSON. Um mecanismo de autenticação confiável porque é assinado digitalmente. Os *Json Web Tokens* podem ser assinados usando um segredo (com o algoritmo HMAC) ou um par de chaves pública/privada usando o RSA. [JWT.IO 2018].

Geralmente um cabeçalho constitui-se em duas partes: o tipo do *token*, que é JWT, e o algoritmo de *hashing* é usado, como HMAC SHA256 ou RSA. Basicamente, informações que irão conter no cabeçalho são de como a assinatura JWT deve ser calculada. Um exemplo de cabeçalho pode ser:

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

Figura 3: Cabeçalho Web Token
Fonte: Elaborado pelo Autor

A segunda parte que constitui o *token* é o *payload*, que contém as declarações e metadados adicionais. Existem várias declarações padrões diferentes, para *payload* do JWT, como:

- Iss: O emissor do *token*;
- Exp: O tempo de vencimento;
- Iat: O horário em que o JWT foi emitido;
- Jti: Um identificador exclusivo para o JWT que é usado para impedir que o JWT seja reutilizado ou reproduzido novamente. Um exemplo de *payload* utilizado na aplicação:

```
{
  "exp": 1524973413,
  "user_name": "admin",
  "authorities": ["ADMINISTRADOR", "LIST_DONATION",
    "SAVE_DONATION", "LIST_TAXPAYER", "LIST_EMPLOYEE",
    "SAVE_TAXPAYER"],
  "jti": "3f08aced-a89f-4868-862b-32d2aa92a474",
  "client_id": "web",
  "scope": ["read", "write"]
}
```

Figura 4: Exemplo payload
Fonte: Elaborado pelo Autor

O *Payload* é, então, codificado em Base64Url para formar a segunda parte do JWT. A última parte do JWT é uma assinatura gerada com base na codificação do cabeçalho (*Header*), na codificação do *payload*, com o segredo e com algoritmo especificado no cabeçalho.

3.2 Tecnologias Lado-Cliente (*Client-Side*)

A aplicação web ela foi desenvolvida com o intuito de ser uma ferramenta de gestão que auxilia se o gestor responsável de forma intuitiva. Para que o usuário final tenha uma melhor performance e uma boa experiência navegando no sistema do aplicativo foi desenvolvido utilizando *AngularJS*, que é um framework robusto desenvolvido na

linguagem *JavaScript* e casa muito bem com o modelo SPA (*Single Page Application*). As SPAs são um novo modelo para desenvolver as aplicações web dando a sensação ao usuário como se ele tivesse executando um aplicativo desktop [ANGULARJS 2018].

O projeto conta com o desenvolvimento de um aplicativo para a plataforma *android* para auxiliar os mensageiros no processo de captação dos recursos para a Instituição Grupo Luta pela Vida. E para o desenvolvimento deste aplicativo utilizamos a o ambiente de desenvolvimento Oficial, o *Android Studio* que é uma plataforma robusta e ajuda o desenvolvedor a construir suas aplicações. Durante o processo de desenvolvimento utilizamos recursos internos, um exemplo é a utilização do Bluetooth para a comunicação com a impressora térmica portátil. Para a comunicação com nossa API Rest foi utilizado a biblioteca *Volley* que facilita as requisições HTTP.

```
//Json Request
String url = "https://json_url/";
JsonObjectRequest jsonObjectRequest=new JsonObjectRequest
(Request.Method.GET, url, null,
    new Response.Listener<JSONObject>() {
        @Override
        public void onResponse(JSONObject response) { }
    },
    new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {}
    });
//add request queue
requestQueue.add(jsonObjectRequest);
```

Figura 5: Solicitação JsonObjectRequest.
Fonte: Elaborado pelo Autor

```
JSONArrayRequest jsonArrayRequest = new JSONArrayRequest
(Request.Method.GET, url, null,
    new Response.Listener<JSONArray>() {
        @Override
        public void onResponse(JSONArray response) {
        }
    },
    new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
        }
    });
//add request queue
requestQueue.add(jsonArrayRequest);
```

Figura 6: Solicitação JSONArrayRequest.
Fonte: Elaborado pelo Autor

Com base na documentação, o *Volley* trabalha com apenas duas classes, *RequestQueue* e *Request*. A *RequestQueue* administra segmentos de trabalho para executar as operações de rede, já a *Request* é um construtor que sempre usa como parâmetro o tipo de método *GET*, *POST*, *PUT* e *DELETE*, a URL do recurso e um evento. O *Volley* fornece as classes *JSONArrayRequest* e *JsonObjectRequest*, as figuras 5 e 6, ilustram as solicitações [DEVELOPER 2018].

4. Estudo de caso

Nesta seção será analisado o processo anterior e o novo processo da instituição, como o objetivo de identificar os pontos de melhoria para a implementação do sistema proposto.

4.1 Processo Existente

O processo de arrecadação da Instituição Grupo Luta Pela Vida é visto conforme está apresentada na figura 7 a seguir. O departamento de telemarketing entra em contato com o doador, caso o doador aceite a efetivação da doação na forma de pagamento de recibo, será definido o valor, e os dados serão registrados no CRM da instituição para o recebimento. Após operador efetivar o cadastro da contribuição do doador, as informações são acessadas pelo departamento SAP, onde os recibos são impressos. Assim que os recibos são impressos por impressora a laser, são separados manualmente por auxiliares e entregues para os mensageiros responsáveis por suas rotas.

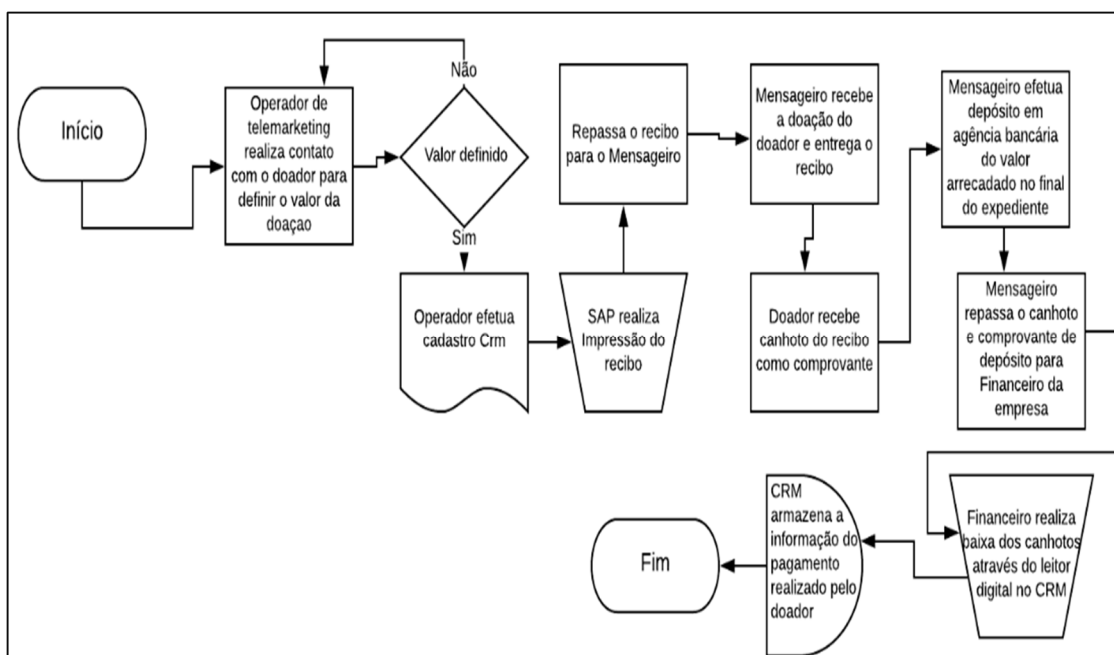


Figura 7: Processo de negócio atual.

Fonte: Elaborado pelo Autor

No ato do recolhimento da doação, o mensageiro recolhe o valor combinado com o doador e é entregue ao doador um recibo em folha A5. Após o expediente do colaborador mensageiro ser finalizado, ele se conduz até a uma agência bancária para o depósito do valor arrecadado.

O processo de conferência dos valores arrecadados das doações pelos mensageiros é realizado pelo departamento Financeiro, onde os mensageiros precisam se deslocar-se até a instituição para a entrega dos comprovantes de depósitos e os canhotos dos recibos. Todo processo de conferência é feito manualmente, contando com leitores de código de barras para efetivar a baixa dos recibos recebidos e não recebidos no sistema CRM da instituição.

A pesquisa teve como foco a área de captação de recursos que conta com o apoio de um Telemarketing, a área financeira da instituição e a área de Mensageiros. A área do telemarketing possui uma equipe composta por operadores, auxiliares administrativos e coordenador, já a área financeira conta com coordenador e auxiliares administrativos, a área de Mensageiro é composta por auxiliares, mensageiros/Moto boy e coordenador. Os departamentos são diretamente responsáveis pela execução e

finalidade do processo das doações.

Com o processo descrito da instituição, foram identificadas as atividades realizadas para o processo de captação de doações da instituição, demonstrando as dificuldades de acordo com a descrição apresentada a seguir

- Dificuldade com impressão dos recibos por serem impressos em impressoras.
- Mão de obra desnecessária para separação de recibos.
- Acumulo de papel para a impressão dos recibos.
- Acumulo de papel para a impressão dos recibos.
- Volume excessivo de papel para os mensageiros.
- Impacto do uso do papel para o meio ambiente.
- Gasto mensal com papel.
- Manutenção com impressoras.
- Informações transitando de forma insegura.
- Falta de informações em tempo real para tomada de decisões.

A partir do cenário atual, foi identificado que a utilização dos recursos para a impressão e as buscas das contribuições por mensageiros, demanda muito tempo para o processo ser realizado, custo com papéis e funcionários para a execução das atividades. A busca de contribuições exige que os mensageiros façam acertos diariamente, tanto para a entrega de arrecadações quanto para a busca de novos recibos, onde exige tempo e dificuldade para deslocar-se com frequência até a instituição.

4.2 Novo Processo

Visando focar no escopo deste trabalho e iniciar o processo de otimização das atividades nas áreas relacionadas de captação da instituição Grupo Luta Pela Vida, foi realizado um estudo dos processos que poderiam ser melhorados e como poderiam ser melhorados. Para tal melhoria, foi desenvolvido uma aplicação *Web* e uma *Mobile*, onde a aplicação *Web* conta com o gerenciamento de dados, e a aplicação *Mobile* executa a impressão dos recibos através de uma impressora portátil.

A aplicação *Web* tem todas as funcionalidades necessárias para garantir que todo o processo de doações e recebimento das mesmas sejam práticos e claros, onde se tem todo o controle de doação com seus respectivos contribuintes cadastrados no sistema e o quadro de funcionários da instituição, ambos podem ser alterados e editados conforme sendo o usuário do sistema e suas permissões, como por exemplo, um funcionário cujo seu cargo seja mensageiro/motoboy, ele somente terá acesso as informações relacionadas a suas respectivas doações, as quais será responsável por recebê-las, por outro lado, um coordenador terá outro nível de permissão, tanto para listar, editar e/ou excluir funcionários de seu quadro, e assim sucessivamente. Em se tratando das doações não será possível um usuário *Web* ter acesso à edição da mesma, isso ficará a cargo do colaborador mensageiro/motoboy no momento do recebimento da doação através da aplicação *Mobile*.

Para garantir que a aplicação atendesse todos os requisitos e funcionalidades propostas como solução dos problemas encontrados foi utilizado os princípios da arquitetura REST como base para construir uma aplicação que obtivesse bom desempenho e atendesse os requisitos de segurança.

Logo abaixo na figura 8, é apresentado o *Dashboard* da aplicação, contendo todo o valor captado de doações em tempo real do ano corrente, a quantidade de doadores captados no mês e a quantidade de doações do respectivo mês, sendo representados também graficamente.



Figura 8: Dashboard da aplicação
Fonte: Elaborado pelo Autor

Todo o processo de captação dos recursos pode ser visualizado pela aplicação web, que traz as informações em tempo real, as quais foram executadas pelo Mensageiro/motoboy pela aplicação *Mobile*, facilitado assim qualquer ação que deva ser executada e/ou alguma decisão que deva ser tomada.

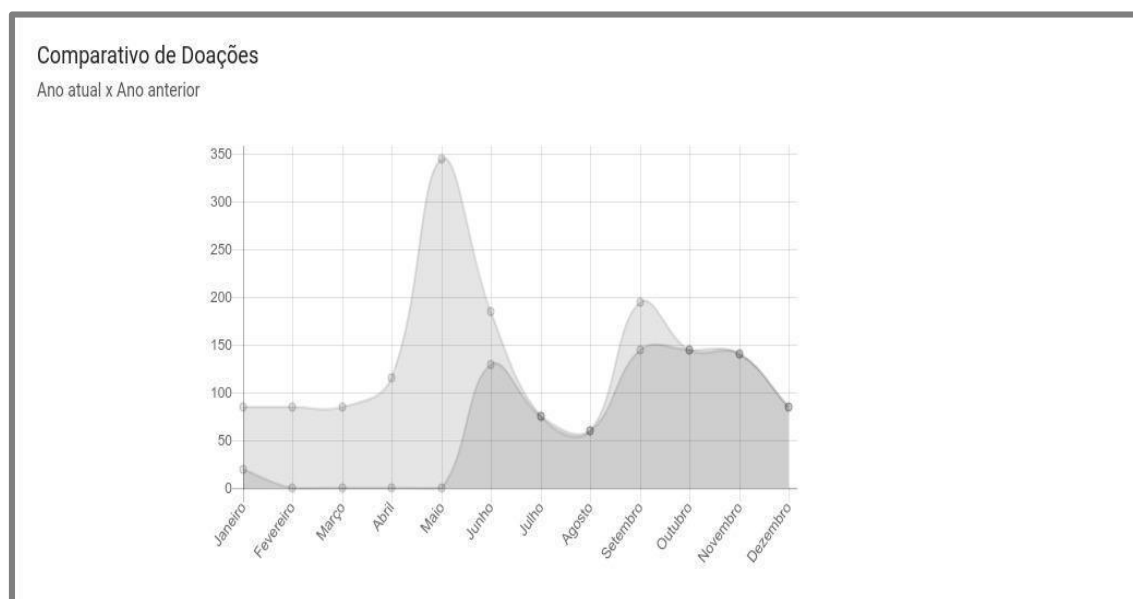


Figura 9: Comparativo de doações
Fonte: Elaborado pelo Autor

O gráfico demonstrado na figura 9, se depara com o comparativo do ano atual e

o ano anterior, visualizando claramente os desempenhos obtidos, caso coordenadores desejam traçar um plano de ação.

A aplicação *Mobile* tem como foco a impressão das doações diárias já agendadas pelo operador de Telemarketing da instituição. Os usuários/mensageiros e recibos são gerenciados pela aplicação web, contando com cadastro e edição apenas do colaborador usuário/mensageiro, já as alterações dos recibos, são feitas pelos mensageiros no ato do recolhimento ou não das doações.

Para o usuário/mensageiro autenticado no aplicativo, os recibos chegaram por identificação de *login* e dia corrente do vencimento do recibo.

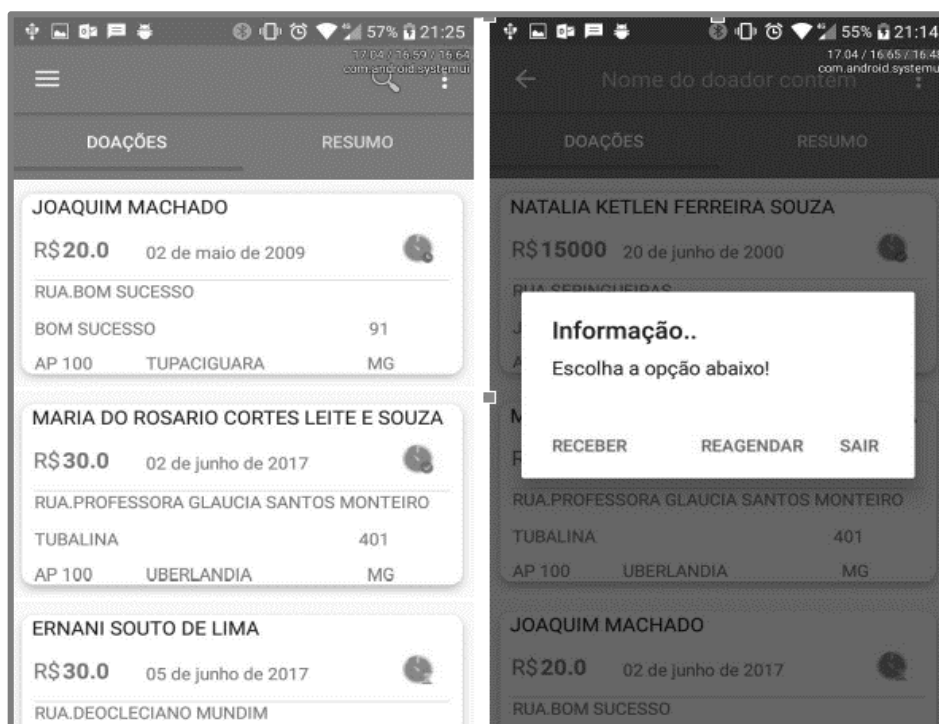


Figura 10: Tela de Listagem de contribuintes e ação
Fonte: Elaborado pelo Autor

Em um determinado dia, pode haver mais de um tipo de ação pelo usuário/mensageiro, cada uma com sua propriedade, brevemente descritas a seguir:

- Receber: O usuário/mensageiro recebe o valor combinado no ato do contato do Telemarketing, onde já foi combinado o valor da doação.
- Reagendar: O usuário/mensageiro reagenda a doação do doador não encontrado em sua residência e/ou que se não dispôs a doar no dia.
- Alterar valor: usuário/mensageiro recebe um valor diferente do combinado com o operador de Telemarketing da instituição.

Na figura 10 mostrada anteriormente, é possível visualizar as informações de cada doador, onde são identificados por ícones que indicam a ação que usuário/mensageiro executou. Os ícones são identificados como: relógio com sinal verde de confirmação, significa que a doação foi recebida; em relação ao que contém um relógio com a ampulheta significa que ainda não foi recebido a doação; por fim o relógio com sinal de hora, que significa o reagendamento da doação caso o doador

queira pagar em outro momento. Para facilitar a busca de doações, foi implementado um filtro que busca o doador pelo nome. A figura 11 é mostrada o fluxo para a impressão do recibo no ato do recebimento da doação.

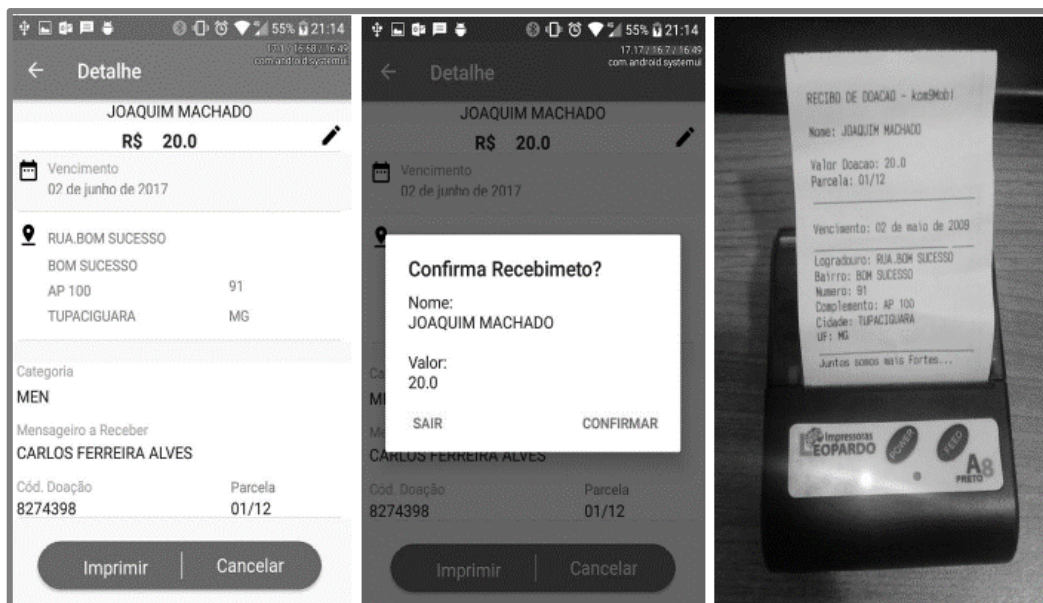


Figura 11: Tela de Detalhe da Doação, confirmação e recibo impresso
Fonte: Elaborado pelo Autor

Assim que o usuário/mensageiro escolher a opção de receber a doação, demonstrado na figura 10, o usuário/mensageiro é direcionado para a tela de detalhe, onde contém todos os dados para o recebimento. A tela de detalhe de doação, também possibilita a alteração do valor da doação para impressão do recibo, selecionando o ícone de edição ao lado do valor. Assim que o usuário/mensageiro solicita a impressão, abre-se uma caixa de confirmação para a impressão do recibo, conforme figura 11, após a confirmação, se inicia a transmissão das informações via *Bluetooth* para impressora térmica portátil, imprimindo o recibo não fiscal para o doador.



Figura 12: Tela de Reagendamento de doação
Fonte: Elaborado pelo Autor

Na tela de reagendamento figura 12, é onde o usuário/mensageiro consegue alterar o vencimento da doação para o doador, percebe-se que além da opção de inserir uma nova data para o vencimento da doação, contém a opção motivo do reagendamento. A inserção da opção motivo reagendamento para o usuário/mensageiro registrar, é considerada para a instituição, como uma identificação para a tratativa das doações reagendadas para futuros contatos.

Percorrendo o fluxo na figura 13, foi proposto à otimização completa dos dados que são transitados durante o processo de arrecadação da instituição.

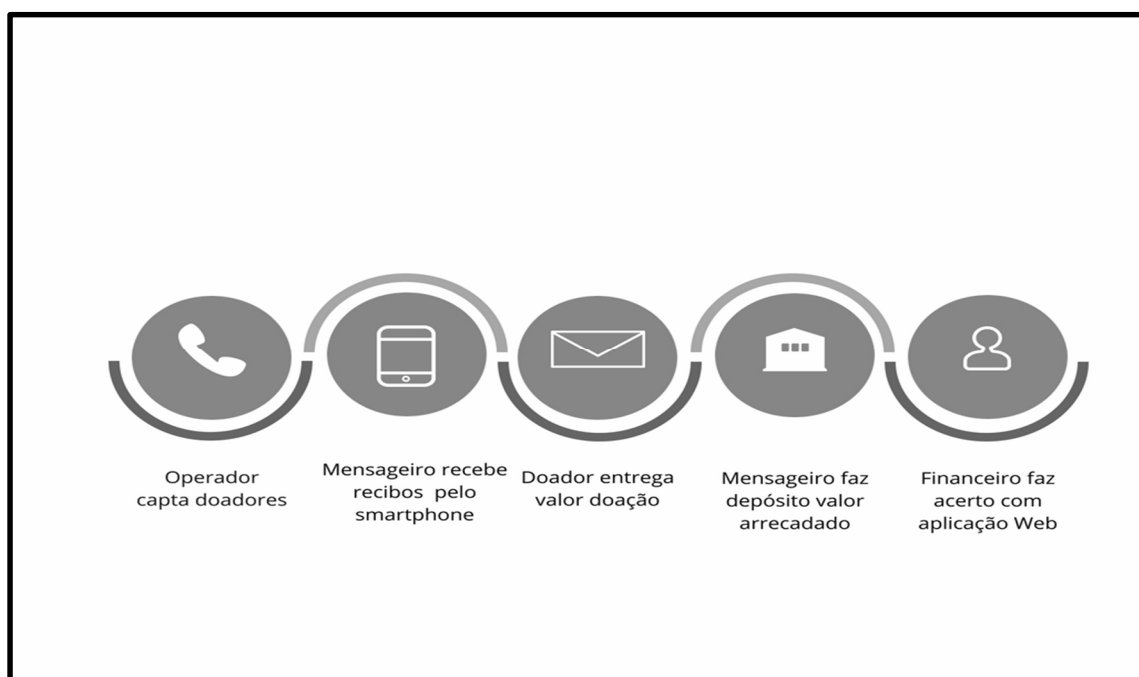


Figura 13: Processo do sistema de arrecadação.

Fonte: Elaborado pelo Autor

Após os dados serem registrados no CRM pelos operadores do Telemarketing da instituição, automaticamente já são enviados para a aplicação proposta. Onde conta com a parte gerencial Web e a aplicação *Mobile* para o gerenciamento das doações. Caso o *Smartphone* esteja conectado a um sinal de dados, as informações das doações são descarregadas para o usuário/mensageiro em tempo real. Assim que o usuário/mensageiro finaliza as arrecadações das doações, é possível o departamento financeiro já visualizar os valores da receita do dia de cada usuário/mensageiro, possibilitando que os mensageiros vão a instituição para a entrega dos comprovantes um único dia do mês.

As informações contidas na aplicação proposta, também são utilizadas no CRM da instituição, para geração de campanhas futuras e controle de ações feitas pelos usuários/mensageiros.

5. Conclusão

Nos últimos tempos, o aumento das doações acarretou em mudanças nos processos já existentes. Fez-se necessário, então, juntamente com a vasta propriedade das tecnologias associadas, a criação da aplicação proposta.

Neste cenário, foi demonstrado que a utilização de um sistema Web juntamente com uma aplicação *mobile*, facilitou o recebimento das arrecadações da instituição, mostrando-se aderente aos objetivos propostos neste trabalho. O sistema já foi testado na instituição, onde foram feitos alguns ajustes para se obter o maior aproveitamento. De acordo com (AYRES 2018), coordenador da área de Tecnologia da Informação da instituição, “a aplicação superou as expectativas da instituição Grupo Luta Pela Vida”.

Durante o processo de desenvolvimento da solução, tivemos algumas dificuldades, umas delas foi a comunicação de dados da aplicação *mobile* via Bluetooth com a impressora térmica portátil. Outra dificuldade encontrada durante o processo de desenvolvimento foi a configuração do Servidor de Autorização OAuth2 por ter uma complexidade alta, exigindo disciplina e estudos aprofundados sobre essa tecnologia, pois a mal configuração poderia causar falhas de segurança comprometendo todo o projeto desenvolvido.

Melhorias podem ser implementadas futuramente, como foi entendido a necessidade de se criar formas para possíveis eventos não esperados dos doadores, além de transformar toda a estrutura construída na plataforma *Android*, para uma plataforma híbrida que atendesse todos os tipos de usuários sem restrição, deixando a aplicação muito mais acessível.

Vale ressaltar a importância de cada contribuinte em um todo, justamente pelo fato de que a instituição vive através de doações, e nada mais justo buscarem maneiras de trazer comodidade, que facilite e que melhore a transição entre doador e instituição.

6. Referências

- AFONSO, Alexandre. Produtividade no desenvolvimento de aplicações web com Spring boot. 3. ed. Uberlândia: Algoworks, 2017
- ANGULARJS. AngularJs, 2018. Disponível em: <https://angularjs.org/>. Acessado em: 28 de abril de 2018.
- AYRES, Erik. Citação ao Projeto [Mensagem Pessoal]. Mensagem recebida por <erick@hospitaldocancer.org.br> em 08 de março de 2018
- BAZOLI, Thiago Nunes. Terceiro setor: parcerias com o Estado à luz do desenvolvimento social sustentável. Londrina: Saúde em Destaque, 2009.
- CAMARGO, Mariângela Franco de. Gestão do terceiro setor no Brasil. São Paulo: Futura, 2001.
- DAVENPORT, T. H. Reengenharia de processos: como inovar na empresa através da Tecnologia da Informação. Rio de Janeiro: Campus, 1994.
- DEVELOPER. Arquitetura Android, 2018. Disponível em: <https://developer.android.com/latform/?hl=pt-br>. Acessado em: 03 de maio de 2018.
- DEVELOPER. Volley, 2018. Disponível em: <https://developer.android.com /volley/>. Acessado em: 03 de maio de 2018.
- DEVELOPER. Android Studio, 2018. Disponível em: <https://developer.android.com/>. Acessado em: 03 de maio de 2018.

- DEVELOPER. Bluetooth, 2018. Disponível em: <https://developer.android.com/topics/connectivity/bluetooth?hl=pt-br>. Acessado em: 04 de maio de 2018.
- FALBO, R. de A. Engenharia de software. Universidade Federal do Espírito Santo, UFES, 2005.
- FILDING. Roy Thomas. Architectural Styles and the Design of Network-based Software Architectures. Dissertação de Doutorado - University of California, Irvine, 2000 Disponível em: http://www.ics.uci.edu/fielding/pubs/dissertation/rest_arch_style.htm. Acessado em 02 de abril de 2018
- JWT. Json Web Token, 2018. Disponível em: <https://jwt.io/introduction/>. Acessado em: 03 de maio de 2018.
- MENDES, António. Arquitetura de Software, 2002. Desenvolvimento orientado para arquitetura. Rio de Janeiro
- OAuth. OAuth 2.0, 2018. Disponível em: [https://oauth.net /](https://oauth.net/). Acessado em: 04 de maio de 2018.
- SPRING. Spring, 2018. Disponível em: <https://spring.io/>. Acessado em: 24 de abril de 2018.
- TOBALDINI, Ricardo Ghisi. Aplicação do Estilo Arquitetural REST a um Sistema de Congressos. Florianópolis: 2008. Monografia em Ciências da Computação – UFSC