

Sistema de Gerenciamento de Processos Jurídicos no Modelo Software as a Service (SaaS)

Matheus Viane de Freitas¹

¹UniRitter Laureate International Universities

Rua Orfanatrópio, 555 – Alto Teresópolis – 90840-440 – Porto Alegre – RS – Brazil

mvdfreitas@gmail.com

Resumo. *Este artigo apresenta o desenvolvimento de um protótipo de uma aplicação que oferece serviços para gerenciamento de processos jurídicos com foco em pequenos escritórios. Esta aplicação permite que as empresas tenham um controle interno maior sobre seus processos, como os prazos e os andamentos processuais, custos do processo e também controlar os recebimentos dos honorários advocatícios. O protótipo será desenvolvido utilizando o Adianti framework, criado pela empresa Adianti, que usa como base a linguagem de programação PHP.*

Abstract. *This article presents the development of a prototype of an application that provides services for managing legal processes focusing on small offices. This application allows companies to have greater control over their internal processes, such as time limits and procedural tempos, process costs and also control the receipt of legal fees. The prototype will be developed using the Adianti framework set up by Adianti company, which builds upon the PHP programming language.*

1. Introdução

O modelo SaaS não surgiu logo de cara com a chegada da internet, primeiro veio o modelo ASP (*Application Service Provider*). O ASP consiste em oferecer o software pela internet, do mesmo modo que o SaaS, só que o ASP apresenta inúmeras soluções para atender a todas as demandas que um cliente possa ter. Desta forma, o software acaba ficando pesado e o cliente acaba pagando o mesmo valor que gastaria caso tivesse comprado o software para instalar em seu computador. Com isso, o SaaS foi criado para solucionar o problema de escalabilidade que o ASP apresenta.[Asaas 2013]

O SaaS (*Software as a service*), é um modelo de distribuição de software, na qual não é vendido e instalado localmente, mas sim é liberado apenas o acesso ao serviço oferecido por este software e é licenciado para a utilização através da internet[Pozzebon 2011]. Esse modelo de negócio iniciou em 2001 nos Estados Unidos, mas foi a partir de 2006 que começou a decolar.[Wagner 2015]

Atualmente, muito se fala no modelo de computação na nuvem, e esse tipo de software está integrado nesse tipo de serviço[Pozzebon 2011]. O SaaS cresce por tornar o nicho mais adaptado à necessidade do usuário final. Afinal, hoje em dia as pessoas querem agilidade nos processos e para isso é preciso estar conectado o tempo todo. O modelo SaaS atende usuários a todo o momento, desde que haja conexão com a internet[Asaas 2013].

Neste contexto, este artigo apresenta a implementação de um protótipo que permite a utilização de alguns serviços que são propostos pela aplicação. Este artigo está organizado da seguinte forma: a seção 2 apresenta um referencial teórico sobre a arquitetura da plataforma SaaS que será utilizada no desenvolvimento da aplicação. A seção 3 apresenta algumas aplicações existentes no mercado que utilizam a plataforma SaaS. Na seção 4 é apresentada a solução implementada. Encerrando o artigo, apresentam-se as considerações finais e as referências utilizadas.

2. Referencial Teórico

Esta seção apresenta conceitos e técnicas para desenvolvimento dos estudos realizados sobre SaaS, com foco no trabalho desenvolvido.

Um fator primordial que as empresas buscam ao contratar um serviço SaaS é a disponibilidade no atendimento, a implantação e claro, a confiabilidade[Pozzebon 2011]. Existem duas categorias principais de software como serviço:[Cássio A. Melo 2008]

- **Serviços de linha de negócios:** oferecidos a empresas e organizações de todos os tamanhos. Os serviços de linha de negócios geralmente são soluções de negócios grandes e personalizáveis direcionadas para facilitar processos de negócios como finanças, gerenciamento da cadeia de suprimentos e relações com o cliente. Normalmente esses serviços são vendidos aos clientes como assinatura.
- **Serviços orientados a cliente:** oferecidos ao público em geral. Os serviços orientados a cliente às vezes são vendidos como assinatura, mas geralmente são fornecidos sem custo e financiados por anúncios.

2.1. Benefícios

No entanto, para quem fornece um software SaaS deve estar preparado pois ele é o responsável por todo o tipo de infraestrutura necessária. Uma das grandes vantagens em adotar o SaaS é a facilidade de atualização do software adquirido, toda a responsabilidade de infraestrutura é por conta de quem está prestando o serviço[Pozzebon 2011].

A utilização do SaaS possui vários benefícios:[Pozzebon 2011]

- **Agilidade:** pois não é mais necessário comprar um servidor ou instalar sistemas;
- **Flexibilidade:** pois, por ser um sistema integrado na web, diferentes usuários em lugares distintos podem utilizar o mesmo sistema sem algum problema;
- **Diminuição do Risco:** a empresa ao invés de investir em compra de servidores, licenças e no treinamento de pessoal o usuário só paga conforme a utilização do sistema;
- **Retorno no Investimento:** no modelo que era usado e, continua ainda por muitos, é necessário esperar meses, ou até anos, para ter o retorno do investimento inicial, o que não acontece quando feito no SaaS, o retorno costuma ser muito mais rápido, já que o investimento inicial requer pouco custo e;
- **Redução de Custos:** há uma significativa redução de custos, já que o fornecedor de SaaS consegue reduzir e consolidar suas despesas, pois uma mesma equipe pode monitorar, realizar backups e prover o serviço pretendido a um número elevado de clientes.

2.2. Comparativo entre o Modelo Tradicional e o SaaS

As diferenças podem ser sumarizadas na tabela a seguir:[Cássio A. Melo 2008]

Pacotes de software tradicional	Software como Serviço
Projetado para os clientes instalarem, gerenciarem e manterem	Projetado para ser distribuído como serviço pela Internet
A solução é arquitetada para ser executada por uma companhia individual em uma infra-estrutura dedicada	Projetado para executar em milhares de clientes diferentes em um único código
Pouco freqüente, atualizações principais a cada 18-24 meses, vendido individualmente para cada base de cliente instalada	Freqüente, atualizações "digeríveis" a cada 3-6 meses para minimizar o impacto no cliente e melhorar a satisfação
Controle de Versão Taxa de Upgrade	Ao corrigir um problema para um cliente, é corrigido para todos
Funcionalidade repetível via Web services, APIs abertas e conectores padrões	Pode usar APIs abertas e Web services para facilitar a integração, mas cada cliente tipicamente tem que pagar pelo trabalho de integração

Tabela 1. Comparação de características entre os modelos de pacotes de software tradicional e de software como serviço.

2.3. Cloud Computing, o alicerce do SaaS

O software como serviço não poderia existir se não fosse pela Cloud Computing (Computação em nuvem), uma tecnologia que oferece espaço para armazenagem de dados na rede virtual da internet. Daí o significado do nome, uma vez que o espaço de armazenagem não é físico, e sim, um espaço que fica online.

Todo software como serviço precisa de um conjunto de servidores compartilhados para oferecer ao cliente um armazém de dados que ele possa acessar de qualquer computador, bastando apenas estar conectado à internet.[Asaas 2013]

2.4. Técnicas para desenvolvimento de uma aplicação SaaS

Abaixo serão listados alguns elementos-chave específicos para o desenvolvimento de software que podem determinar se um aplicativo SaaS torna-se bem-sucedido e, se vale a pena lançá-lo. Essas são dicas destinadas a fornecerem uma perspectiva esclarecedora sobre o desenvolvimento baseado em SaaS.[Orlando 2009]

1. **Faça do UXD(*User Experience Design*) seu bem mais valioso**

Verificar o que os usuários necessitam e o que eles avaliam, além também das suas limitações e suas habilidades. Para assim promover o aperfeiçoamento da qualidade de interação dos usuários e das percepções de seu produto, e de todos os serviços relacionados.

2. **Adapte-se aos requisitos em constante mudança**

Estar sempre preparado para implementar novas metodologias de desenvolvimento ou variantes das metodologias existentes a cada projeto. Sendo que o cliente mudará os requisitos do projeto depois de concluído.

3. **Adote padrões abertos**

As empresas baseadas em SaaS devem considerar a adoção de padrões abertos, pois a compatibilidade com outros dispositivos, plataformas, serviços e aplicativos da Web significa menos codificação em futuras iterações e maior adoção do produto pelo consumidor.

4. **Esquematize antes de projetar**

Um esquema é simplesmente uma visualização conceitual de um estado específico de uma UI (*User Interface*) de um programa de software a partir de um ponto de vista funcional.

5. **Forneça uma infraestrutura de nuvem para o SaaS**

Por exemplo, o Amazon EC2 (*Amazon Elastic Compute Cloud*) tem o potencial de oferecer economias consideráveis às empresas que executam aplicativos SaaS.

6. **Produza documentação completa de projeto antes de iniciar a codificação**

Documentação de projeto de alta qualidade serve como um roteiro para os responsáveis pela execução do projeto e tem o potencial de acelerar consideravelmente a codificação do software. Uma excelente maneira de comunicar tais princípios é utilizando o córtex visual como diagramas, fluxogramas, entre outros.

7. **Seja obcecado por testes de unidade**

Os projetos que implementam testes de unidade tardiamente fracassam. Temos muitos ganho com essa estratégia, como:[Gama 2013]

- Feedback rápido sobre a nova funcionalidade e sobre as outras funcionalidades existentes no sistema
- Código mais limpo, já que escrevemos códigos simples para o teste passa
- Segurança no Refactoring pois podemos ver o que estamos ou não afetando
- Segurança na correção de bugs
- Maior produtividade já que o desenvolvedor encontra menos bugs e não desperdiça tempo com depuradores
- Código da aplicação mais flexível, já que para escrever testes temos que separar em pequenos "pedaços" o nosso código, para que sejam testáveis, ou seja, nosso código estará menos acoplado.

8. **Construa protótipos utilizáveis**

Um protótipo nada mais é que uma prova de conceito. Desenvolvedores inteligentes de SaaS tiram proveito do tempo que têm para criar o protótipo. Considere quanto pode se conseguir durante esse tempo:

- Projete e construa a base da arquitetura.
- Crie o esquema do banco de dados SaaS construindo um XML (*eXtensible Markup Language*) DTD (*Document Type Definition*) personalizado, e use o XML como origem de dados do protótipo.
- Crie o pacote, interface e estrutura de classe organizacionais do aplicativo em tamanho total, mesmo se os arquivos não fizerem nada além de declarar o nome da classe e a interface que implementa inicialmente.

É importante reconhecer que o mundo do desenvolvimento de software evolui tão rapidamente que os principais elementos para o desenvolvimento bem sucedido de SaaS eventualmente mudarão.

3. Estado da Arte

Como referência de sistemas que utilizam SaaS, apresento 3 aplicações:

3.1. Spotify

O Spotify foi lançado em outubro de 2008 pela startup sueca Spotify AB. Em 15 de setembro de 2010.

É um serviço de música comercial em streaming, podcasts e vídeo comercial que fornece conteúdo provido de restrição de gestão de direitos digitais[4] de gravadoras e empresas de mídia.[Wikipédia 2015]

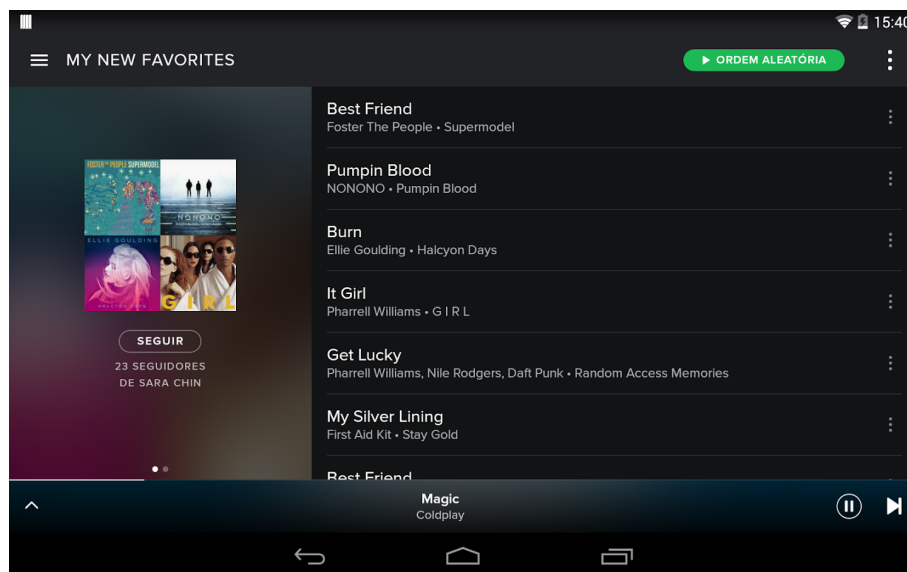


Figura 1. interface da aplicação Spotify

3.2. Gmail

Lançado em 1 de abril de 2004, o Gmail revolucionou o mercado de correios eletrônicos com a variada oferta de recursos para o usuário. Inicialmente, o acesso ao serviço Gmail só era possível através de convite, mas em 2007 o serviço passou a ser acessível a todo o público.

O serviço Gmail oferece ainda aos usuários a possibilidade de utilização do próprio domínio em uma conta do Gmail. É um serviço interessante para quem quer continuar a usar o Gmail, mas divulgar um endereço profissional no cartão de visitas.[Significados 2015]

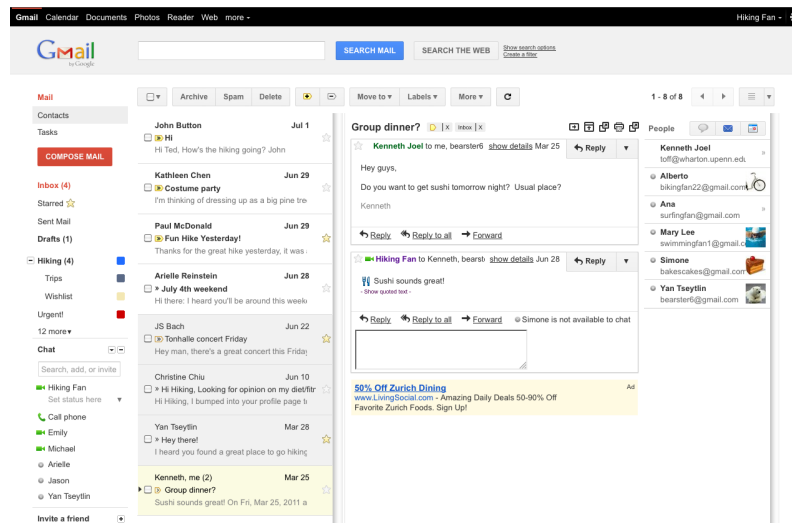


Figura 2. interface da aplicação Gmail

3.3. Salesforce

Fundado em 1999 por um executivo da Oracle chamado Marc Benioff, pode-se dizer que a Salesforce.com, inovou completamente o mercado de softwares CRM (*Customer Relationship Management*) utilizando o conceito de SaaS.[Cássio A. Melo 2008]

É um sistema CRM bem-sucedido envolve muitas áreas distintas de sua empresa, começando por vendas e ampliando para todas as áreas de interação com o cliente, como marketing e atendimento ao cliente.[Salesforce 2015]

A Salesforce oferece uma solução de tecnologia para todas essas áreas e muito mais. Com a Salesforce e nossa plataforma de computação na nuvem Force.com, você pode cuidar dos clientes e do seu orçamento ao mesmo tempo. Tudo pronto e funcionando em 30 dias com a solução de CRM mais comprovada do mundo.[Salesforce 2015]

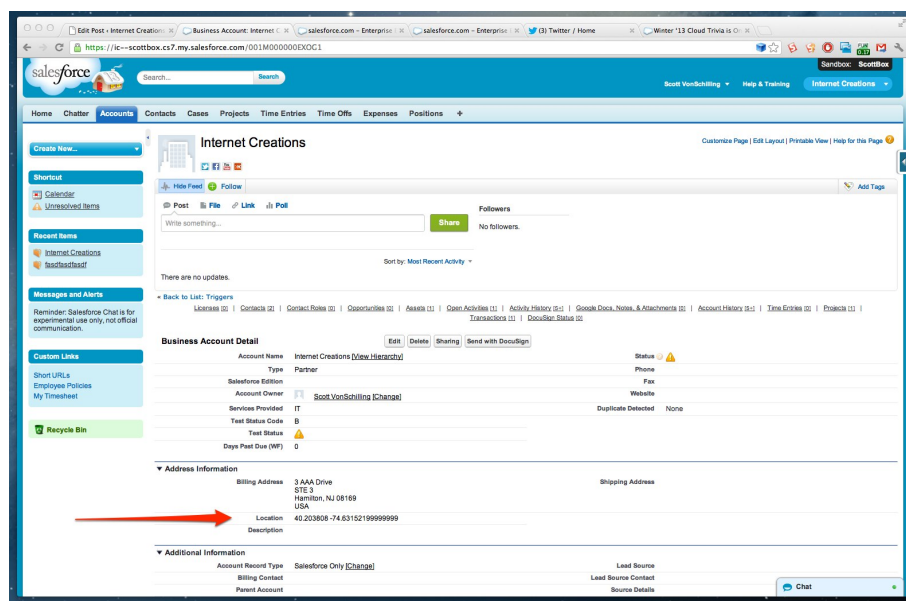


Figura 3. interface do Sistema Salesforce

4. Solução Implementada

A solução implementada será empregada para auxiliar os escritórios de advocacia ou profissionais autônomos na área do direito, que não possuem ferramentas, infraestrutura e até mesmo poder aquisitivo necessários para o melhor gerenciamento dos seus processos, pois sabemos como é custoso investir em infraestrutura e também em um software que pode não atender por completo o seu objetivo.

Para tanto, o protótipo desenvolvido possui um cadastro de clientes (os contratantes do serviço), cadastro dos processos e as partes que o compõem como, histórico do processo, tramitação do processo, custas processuais, valor da causa, partes (réu e autor) que compõem o processo e demais informações. Bem como disponibilizar a informação sobre determinado processo onde o cliente estiver, sendo essa a grande vantagem do SaaS.

4.1. Modelagem e Solução da Ferramenta Proposta

Esta seção apresenta os diagramas de Casos de Uso, DER (*Diagrama Entidade Relacionamento*) e arquitetura da aplicação. Estas modelagens apresentam a estrutura da ferramenta implementada.

4.1.1. Diagrama de Casos de Uso

Este diagrama procura, por meio de uma linguagem simples, demonstrar o comportamento externo do sistema, buscando apresentar o sistema por uma perspectiva do usuário, demonstrando as funções e os serviços oferecidos e quais usuários poderão utilizar cada serviço. Este diagrama é, dentre todos os diagramas da UML (*Unified Modeling Language*), o mais abstrato, flexível e informal, sendo utilizado principalmente no início da modelagem do sistema, embora venha a ser consultado e possivelmente modificado durante todo o processo de engenharia e sirva de base para a modelagem de outros diagramas.[Guedes 2007]

O caso de uso da ferramenta implementada (Figura 4) apresenta dois atores: o ator "Administrador", que é o responsável pela administração da aplicação, pelo cadastramento de novos clientes habilitando o acesso ao sistema. O ator "Usuário" que é o cliente que pagará pelo serviço que disponibilizado para o gerenciamento dos processos.

Após as seguintes tarefas serem realizadas pelo Administrador da aplicação:

- Cadastrar Cliente / Cadastrar Usuário: Registra novos cliente e usuários que utilizarão o serviço.
- Cadastrar Áreas de acesso: Registra as telas na qual o cliente terá acesso.
- Gerar Cobrança: Gera a cobrança que será enviada aos clientes para pagamento.

O Usuário já estará apto a realizar as seguintes tarefas:

- Cadastrar Pessoas: permite que o cliente cadastre novas pessoas na base de dados;
- Cadastrar Advogado: com base no cadastro de pessoas, é setado se a pessoa é advogado ou não;
- Cadastrar Processos: parte que engloba todos os demais itens da aplicação;
- Cadastrar Partes do Processo: insere no processo o(s) réu(s) e autor(es) que o compõem;

- Cadastrar Ação: cadastra o tipo de ação do processo;
- Cadastrar Alvará: cadastra o pedido de alvará judicial;
- Cadastrar Tipo de despesas: cadastro das custas do processo;
- Cadastrar Tramitação: cadastra o andamento processual;
- Cadastrar Comarca: cadastra o estado e a cidade em que processo foi distribuído;
- Cadastrar de Carga: cadastra a pasta de um processo que consta sobre a carga de um determinado advogado;
- Cadastrar Nota de Expediente: cadastra o despacho exarado pelo Juiz;
- Gerar Relatórios:
 - Por data de audiência: Gera um relatório de processos em uma determinada data ou período;
 - Por Advogado: Gera um relatório de processos que em que o advogado participa;
 - De Carga: Gera um relatório de histórico do processo, por quais advogados determinado processo já passou;
 - Por Tipo de Ação: Gera um relatório de processos de um determinado tipo de ação.

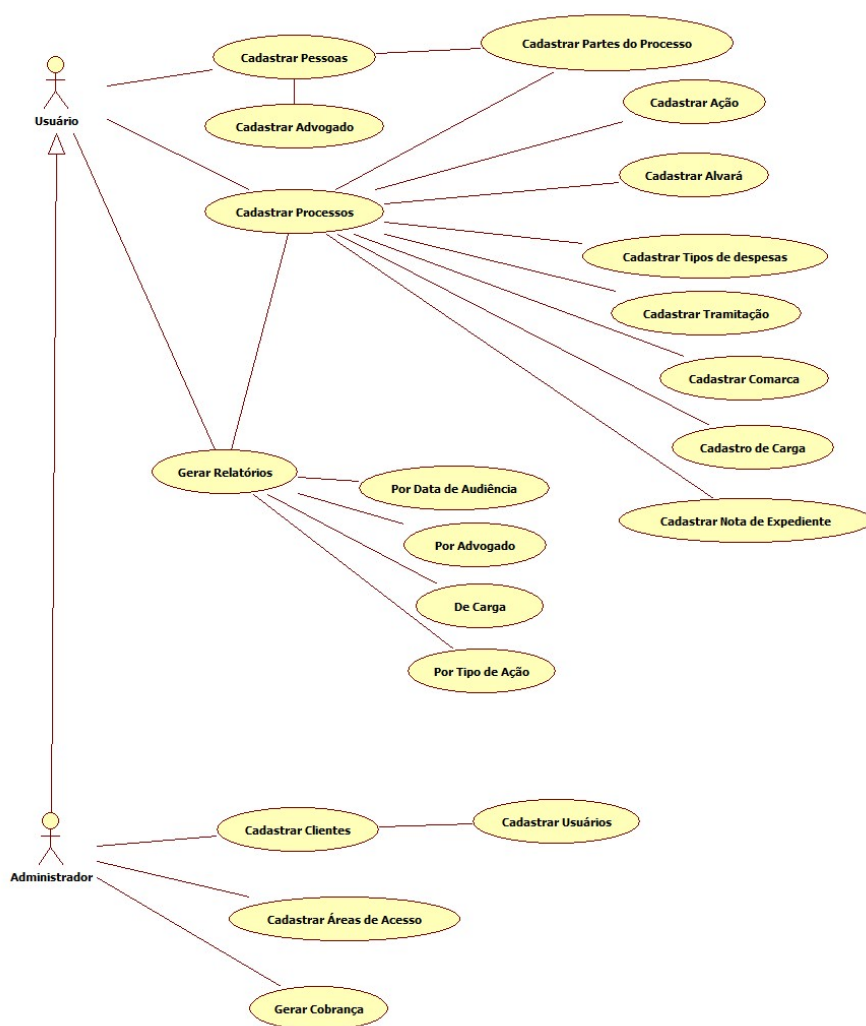


Figura 4. Diagrama de Caso de Uso da Aplicação Implementada

4.1.2. DER (Diagrama Entidade Relacionamento)

O Diagrama Entidade Relacionamento é composto por um conjunto de objetos gráficos que visa representar todos os objetos do modelo Entidade Relacionamento. O DER fornece uma visão lógica do banco de dados, fornecendo um conceito mais generalizado de como estão estruturados os dados de um sistema.

Pelo fato da aplicação possuir muitas tabelas, citarei apenas as mais importantes.

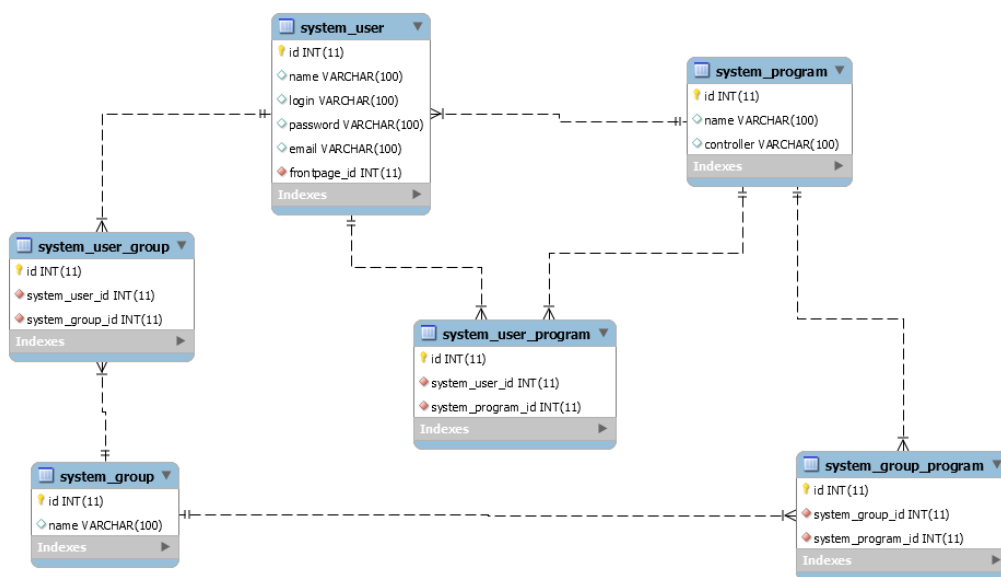


Figura 5. Tabelas de Administração do Sistema

Analisando-se a figura 5, tem-se a seguinte estrutura:

- A tabela *system_user* armazena os dados dos usuários que terão acesso ao sistema.
- A tabela *system_program* armazena as telas que o sistema possui.
- A tabela *system_user_program* armazena as telas que cada usuário terá acesso.
- A tabela *system_group* armazena os grupos.
- A tabela *system_user_group* armazena os grupos dos quais os usuários fazem parte.
- A tabela *system_group_program* armazena quais acessos determinado grupo de usuários terá acesso.

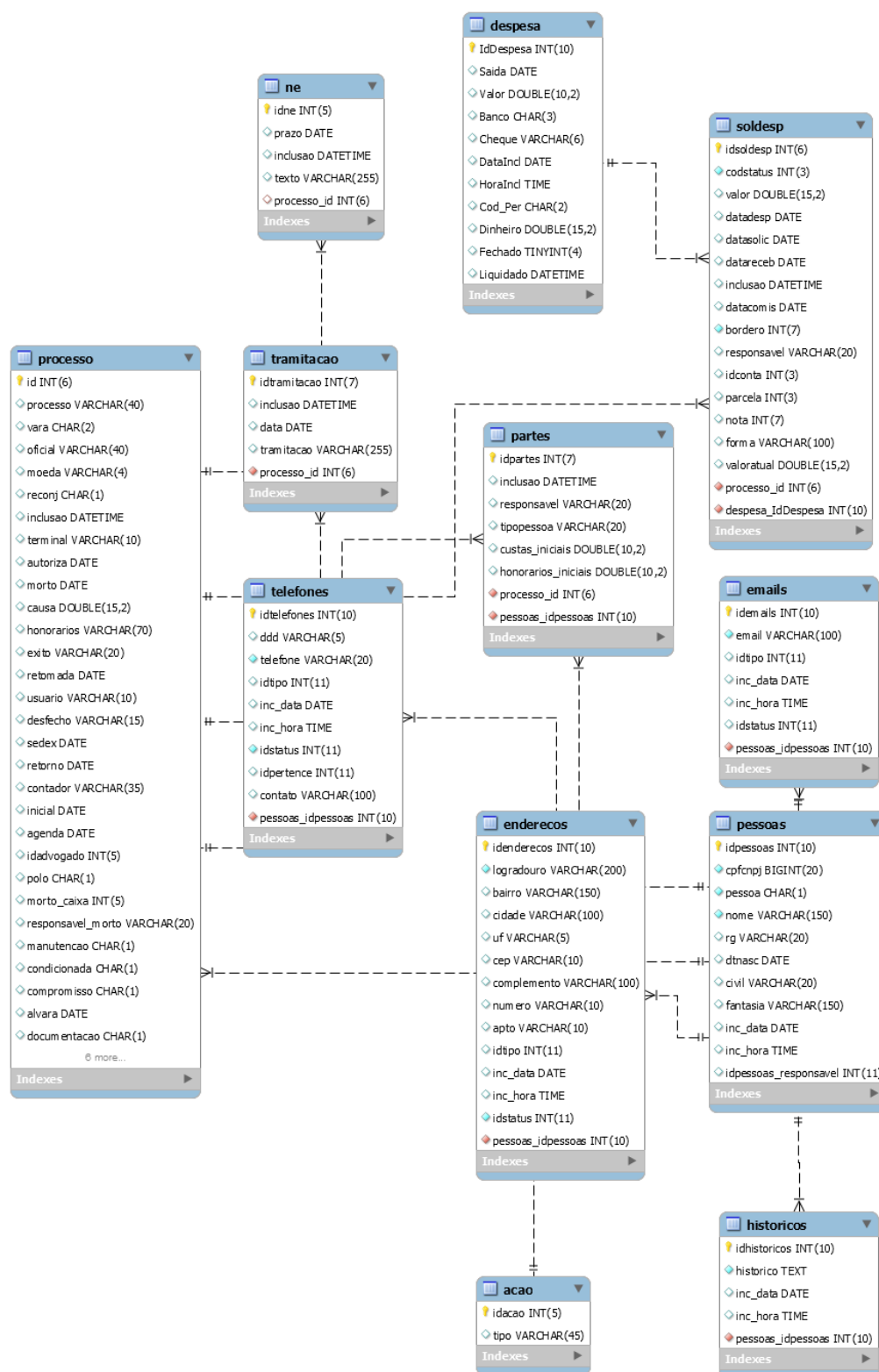


Figura 6. Principais tabelas que fazem parte do cadastro de processo.

Analisando-se a figura 6, tem-se a seguinte estrutura:

- A tabela *pessoas* armazena os dados cadastrais de pessoas físicas e jurídicas.

- A tabela *enderecos* armazena os dados endereços como logradouro, cep, bairro, etc.
- A tabela *emails* armazena os emails da pessoa.
- A tabela *telefones* armazena os telefones que a pessoa possui;
- A tabela *historicos* armazena todo contato que é realizado com a pessoa.
- A tabela *acao* armazena o tipo de ação como ação revisional, ação contra o estado, entre outros.
- A tabela *processo* armazena todos os dados detalhados que fazem parte de um processo.
- A tabela *partes* armazena as pessoas que fazem parte do processo (réu e autor).
- A tabela *tramitacao* armazena as informações do andamento do processo.
- A tabela *ne* armazena as informações da nota de expediente.
- A tabela *despesa* armazena os tipos de despesa como xerox, cartório, entre outros.
- A tabela *soldesp* armazena as solicitações de despesas que são autorizadas para dar andamento ao processo.

4.1.3. Arquitetura da Aplicação

O esquema apresentado na figura 5 apresenta os componentes que compõe o SaaS. O software que é disponibilizado através da internet como serviço.

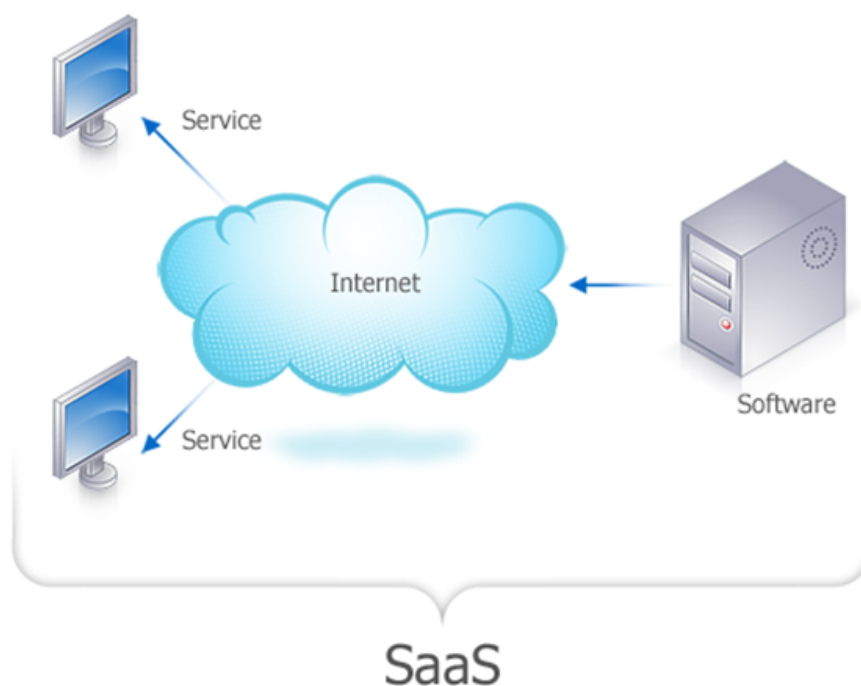


Figura 7. Solução SaaS

4.1.4. Tecnologias Empregadas

As tecnologias empregadas no desenvolvimento deste trabalho foram o framework *Adianti*, que é uma ferramenta open-source que fornece uma arquitetura completa para o desenvolvimento de aplicações PHP, é um framework baseado em componentes e orientado à eventos que utiliza padrões de projeto popularmente conhecidos como o MVC (*Model View Controller*), Front Controller e padrões de projeto para ORM (*Mapeamento Objeto Relacional*) como Active Record e Repository. Para o armazenamento de dados, foi utilizado o banco de dados MySQL que também é uma ferramenta open-source. Estas tecnologias foram escolhidas por serem softwares livres, e claro o custo, já que as soluções proprietárias são muito mais caras.

4.1.5. Funcionalidades Implementadas

O protótipo apresentado possui as seguintes funcionalidades implementadas:

- **Cadastros:**

Comarca: Cadastro responsável pela localização de cada TJ(Tribunal de Justiça) em cada cidade.

Tipo de despesas: Cadastro responsável por setar se determinada despesa é um custo ou honorário e se o mesmo deve ser reembolsado ou não.

Fases do processo: Cadastro de fases que cada processo pode possuir.

Juízo: Cadastro de cada magistrado na justiça estadual responsável por uma vara. Ex.: vara cível, vara criminal, etc.

Enviar o processo para o morto: Cadastro responsável por quando o processo chegar ao fim e ordem judicial foi arquivado definitivamente.

Pessoas: Cadastro das partes que farão parte do processo.

The screenshot displays the 'Juridico Web' application interface. On the left is a dark sidebar with navigation links: 'Matheus', 'Menu', 'Consultas', 'Cadastros', and 'Relatórios'. The main content area is titled 'Cadastro de Pessoas' and contains several form sections. The 'Endereço' section includes input fields for 'Logradouro', 'Complemento', 'Bairro', 'CEP', 'Cidade', and 'UF', with a table below for data entry. The 'Telefones' section has fields for 'DDD' and 'Telefone'. The 'Email' section has an 'Email' field. Each section has 'Register', 'Delete', and 'Cancel' buttons. A 'Save' button is at the bottom left.

Figura 8. interface do cadastro de pessoas

Retomada do bem: Cadastro responsável pela retomada do bem por inadimplemento.

Tipo de ação: Cadastro dos tipos de ações que cada processo pode ter. Ex.: ação de busca e apreensão, ação ordinária, etc.

ID	Tipo
1	BUSCA E APREENSÃO
2	REINTEGRAÇÃO DE POSSE
3	REVISIONAL
4	EXECUÇÃO DE TÍTULO EXTRAJUDICIAL
5	EXECUÇÃO DE SENTENÇA
6	MONITÓRIA
7	PRECATÓRIA DE CITAÇÃO E PENHORA
8	EMBARGOS A EXECUÇÃO
9	ORDINÁRIA
10	CAUTELAR SPOUSERASA

Figura 9. interface do cadastro de tipo de ação

Processo: Nesse cadastro abrange todas as informações que foram cadastradas anteriormente.

Partes	Tramitações	Contratos	Situação do Pedido
Código	Tipo	Custas Iniciais	Honorários Iniciais

Figura 10. interface do cadastro do processo

- **Consultas:**

Agenda: Filtra por período e status os processos de determinado advogado, retornado como resultado os prazos e a quantidade de processos dentro do período e status selecionados.

Processo: A consulta pode ser realizada pelo número do processo, nome das partes, entre outras.

- **Relatórios:**

Audiência: Filtra por período as audiências dos processos que estão a cargo de determinado advogado.

Processos: Nesse relatório é possível fazer uma consulta mais detalhada dos processos.

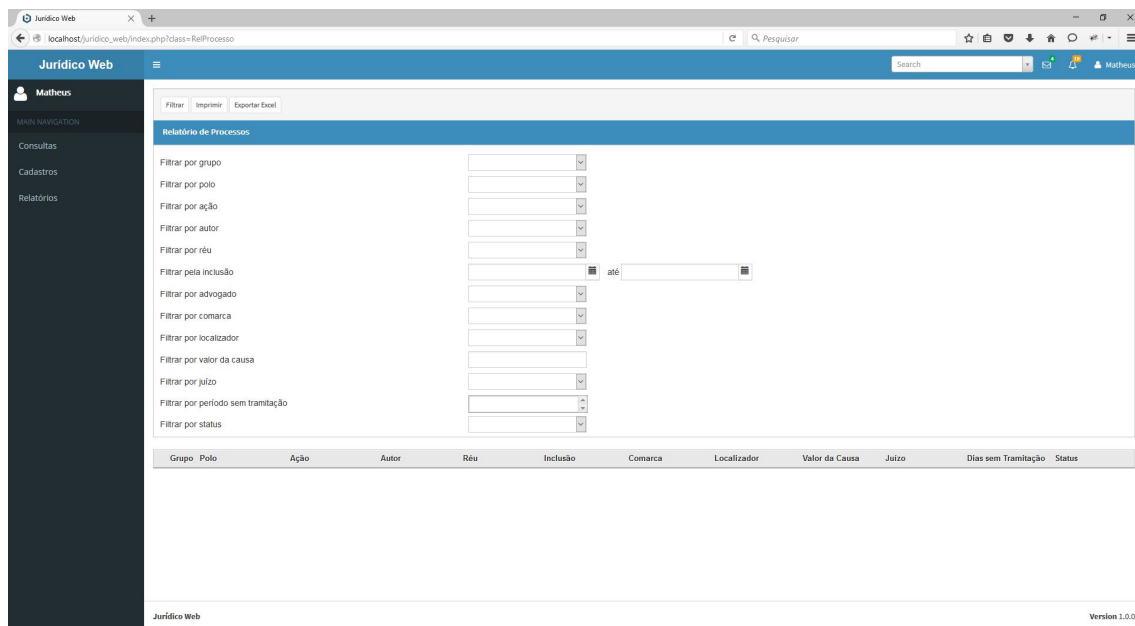


Figura 11. interface do relatório de processos

Receitas e despesas: Filtra por período as despesas e receitas dos processos que estão cadastrados.

5. Considerações Finais

Os sistemas apresentados no estado da arte, verificou-se que a arquitetura de SaaS é bem elaborada, forte no mercado e utilizamos às vezes sem perceber que se trata de SaaS.

O protótipo desenvolvido realizou com sucesso todos os serviços que foram propostos. Outros serviços não menos importantes serão disponibilizados para o enriquecimento do aplicação, como por exemplo, as partes que compõem o processo, possam também consultar o andamento processual.

Por se tratar de um protótipo, o mesmo pode sofrer alterações. Futuramente a intenção seria criar novas funcionalidades e vender o serviço, já que a quantidade de serviços que podem ser oferecidos são muitos. O foco para a venda desse serviço, acredito eu que seria no gasto que o cliente teria com a aquisição de uma infraestrutura cara e um software de prateleira que não o atenderia por completo, gerando gastos desnecessários, em relação ao serviço que estaria sendo oferecido.

6. Referências

Referências

- Asaas, E. (2013). Saas o crescimento do uso no brasil.
- Cássio A. Melo, Daniel F. Arcoverde, f. R. A. M. J. H. C. P. R. Q. F. (2008). Software como serviço: Um modelo de negócio emergente.
- Gama, A. (2013). Test driven development: Tdd simples e prático.
- Guedes, G. T. A. (2007). Uml 2 guia prático. In Guedes, G. T. A., editor, *UML 2 Guia Prático*. Novatec Editora.
- Orlando, D. (2009). 10 principais dicas para criar software como serviço de sucesso.
- Pozzebon, R. (2011). Saas - software as a service: O que é?
- Salesforce (2015). Salesforce crm.
- Significados (2015). Gmail.
- Wagner, D. (2015). Guia de crescimento para o saas brasileiro.
- Wikipédia (2015). Spotify.