

INTELIGÊNCIA ARTIFICIAL: **AUTOMATIZANDO SISTEMAS DE COBRANÇA**

Vinicius Jairo Rocha Queiroz de Albuquerque
Hercilio de Medeiros Sousa



ISBN: 978-85-5597-058-0

Inteligência artificial: automatizando sistemas de cobrança

**Vinicius Jairo Rocha Queiroz de Albuquerque
Hercilio de Medeiros Sousa**
(Autores)

Instituto de Educação Superior da Paraíba - IESP

Cabedelo
2018



INSTITUTO DE EDUCAÇÃO SUPERIOR DA PARAÍBA – IESP

Diretora Geral

Érika Marques de Almeida Lima Cavalcanti

Diretora Acadêmica

Iany Cavalcanti da Silva Barros

Diretor Administrativo/Financeiro

Richard Euler Dantas de Souza

Editores

Cícero de Sousa Lacerda

Hercilio de Medeiros Sousa

Jeane Odete Freire Cavalcante

Josemary Marcionila Freire Rodrigues de Carvalho Rocha

Corpo editorial

Antônio de Sousa Sobrinho – Letras

Daniel Vitor da Silveira da Costa – Publicidade e Propaganda

Hercilio de Medeiros Sousa – Computação

José Carlos Ferreira da Luz – Direito

Marcelle Afonso Chaves Sodré – Administração

Maria da Penha de Lima Coutinho – Psicologia

Rafaela Barbosa Dantas – Fisioterapia

Rogério Márcio Luckwu dos Santos – Educação Física

Thiago BizerraFideles – Engenharia de Materiais

Thiago de Andrade Marinho – Mídias Digitais

Thyago Henriques de Oliveira Madruga Freire – Ciências Contábeis

Copyright © 2018 – Editora IESP

É proibida a reprodução total ou parcial, de qualquer forma ou por qualquer meio. A violação dos direitos autorais (Lei nº 9.610/1998) é crime estabelecido no artigo 184 do Código Penal.

O conteúdo desta publicação é de inteira responsabilidade do(os) autor(es).

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

A345i

Albuquerque, Vinicius Jairo Rocha Queiroz de

Inteligência artificial: automatizando sistemas de cobrança [recurso eletrônico] / Vinicius Jairo Rocha Queiroz de, Hercílio de Medeiros Sousa. - Cabedelo, PB: Editora IESP, 2018.
50 p.

Tipo de Suporte: E-book
Modo de Acesso: Digital via página web
ISBN: 978-85-5597-058-0

1. Inteligência artificial. 2. Sistemas para internet.
3. Sistemas de cobranças. 4. Computação. I. Sousa, Hercílio de Medeiros. II. Título.

CDU: 004.8

Bibliotecária: Angélica Maria Lopes Silva – CRB-15/023

Editora IESP

Rodovia BR 230, Km 14, s/n,
Bloco Central - 2 andar - COOPERE
Morada Nova. Cabedelo - PB.
CEP 58109-303

SUMÁRIO

1. INTRODUÇÃO	9
2 PLANEJAMENTO E FERRAMENTAS	10
2.2 FUNDAMENTAÇÃO TEÓRICA	11
2.2.1 Inteligência Artificial.....	11
2.2.2 Machine Learning – Aprendizado de máquina.	12
2.2.3 Processamento de Linguagem Natural.	13
2.2.4 Computação nas Nuvens.	14
2.2.5 O IBM Watson	15
2.3 INDICAÇÃO DE METODOLOGIA.....	16
2.4 PROCEDIMENTOS ADOTADOS.....	16
2.4.1 Métodos	16
2.4.2 Técnicas.....	21
2.4.3 Ferramentas	23
3 RESULTADO	25
3.1 DECLARAÇÃO DE ESCOPO	25
3.2 IDENTIFICAÇÃO DOS REQUISITOS.....	25
3.2.1 Técnica de Levantamento de Requisitos	25
3.2.2 Descrição Simples dos Requisitos	26
3.3 ANÁLISE DE REQUISITOS	27
3.3.1 Diagrama de Caso de Uso – UC	27
3.3.2 Especificação e descrição dos casos de uso.	29
3.3.2.1 Sistema web	29
3.3.2.2 Sistema Mobile	31
3.4 DIAGRAMA E CLASSES.....	33
3.5 DIAGRAMA DE SEQUÊNCIA.....	34
3.6 PROJETO DE BANCO DE DADOS.....	35
3.6.1 Modelo Conceitual.....	35
3.6.2 Modelo Lógico	37
3.6.3 Modelo Físico.....	38
3.6.3 Dicionário de Dados	39
3.7 MANUAL DO USUÁRIO DO PROTÓTIPO	41
3.7.1 Manual Sistema usuário PagueTudo (Móvel)	41

4 CONSIDERAÇÕES FINAIS	45
REFERÊNCIAS.....	47
APÊNDICE A (Modelo Físico do Banco de Dados).....	49

1. INTRODUÇÃO

Este trabalho tem como objetivo descrever o desenvolvimento de um Sistema Automatizado de Meios de Cobrança, utilizando técnicas especiais de inteligência artificial, chamado de PagueTudo, que é um conjunto de soluções que servem para coletar os dados dos devedores, efetuar a cobrança utilizando o meio de contato definido através de análise das preferências de cada devedor, realizar automaticamente a negociação dos valores e ainda gerar os boletos para que possam ser pagos.

A intenção desta ferramenta é diminuir os índices de inadimplência, realizando cobranças pontuais guiadas a dados, em que as negociações dos valores sejam feitas sem qualquer intervenção humana, diminuindo o constrangimento gerado, buscando assim aumentar o incentivo para o devedor pagar as suas contas. Como uma das grandes vantagens ainda temos a diminuição drástica dos custos de operação do meio de cobrança em si, que atualmente é feito através de escritórios de cobrança que gastam milhões com *call centers*.

A coleta dos dados dos devedores será feita através de um sistema web, em que os credores poderão exportar os dados nos mais variados formatos, então o sistema trata e realiza automaticamente o cadastro do devedor no banco de dados. Validada as informações, no primeiro momento, o PagueTudo vai entrar em contato através de e-mails, mensagem eletrônica, ou até mesmo ligar com o uso de um robô capacitado com processamento de linguagem natural, convidando para o futuro usuário baixar o aplicativo mobile, ou acessar o site, para realizar a negociação dos seus débitos.

O sistema é uma plataforma multilateral, que segue a proposta da economia colaborativa e tem inúmeras soluções para os dois lados interessados. Na solução para o credor, a possibilidade de mandar pacotes de mensagens, ligações e notificações para controlar a cobrança de forma mais eficaz. Para o usuário, ele terá acesso via aplicativo ou plataforma web a todas os seus débitos de forma organizada, onde poderá realizar sem constrangimento toda a negociação.

A presente pesquisa, pretende desenvolver uma plataforma multilateral que automatizará os meios de cobrança atuais, composto de duas partes: Um aplicativo mobile que terá por interface um robô digital para realização da negociação das dívidas. Um sistema *WEB* de painel administrativo para os credores realizarem o cadastro dos inadimplentes.

2 PLANEJAMENTO E FERRAMENTAS

De acordo com dados levantados no *Serasa Experian*, 59 milhões de brasileiros estão com seus nomes negativados no ano de 2016, um aumento de mais de 9.1% em relação a 2015. Conforme dados da mesma empresa, o montante aproximado dos valores devidos ultrapassa os R\$250.000.000.000,00(duzentos e cinquenta bilhões), dados que mostra uma incrível demanda comercial para este mercado de cobrança a inadimplentes. Neste contexto a cobrança é feita atualmente por meio de escritórios especializados, que se mostram ineficazes na realização efetiva das cobranças e também devido aos seus altos custos de processos, visto que gastam milhões com Call Centers e uma abordagem ineficaz.

Outro dado levantado por pesquisa da FECOMERCIO, mostrou que em média 73,3% das pessoas se sentem constrangidas ao receberem cobranças de outras pessoas, preferindo serem abordadas por e-mail, mensagem eletrônica ou outro meio impessoal. De acordo com esses dados, se constata que o número de pessoas descontentes com a forma de abordagem é consideravelmente alta. Em algumas entrevistas *in loco*, realizadas pelo próprio autor desta peça, ficou evidenciado que a esmagadora maioria das pessoas que ligam cobrando são mal-educadas, não dão a devida atenção e dificultam a negociação dos débitos, gerando uma situação em que torna improvável que o devedor vá quitar suas dívidas. Então, chega-se a conclusão que o modelo atual é um processo de alto custo e ineficaz.

A solução proposta é o desenvolvimento de uma plataforma multilateral de fácil gerenciamento em ambas as pontas da plataforma, onde os devedores poderão via aplicação mobile encontrarem os seus débitos e negociarem via interface inovadora de assistente digital. Na parte dos credores é facilitado a

inserção dos dados de devedores no sistema, pois o mesmo aceitará todo o tipo de planilha e extensão. Como também serviços de mensagens e ligações a preço reduzido.

Atualmente as cobranças dos débitos atrasados é feita através de escritórios especializados que utilizam milhares de funcionários no esquema de *Call Centers*, a grande maioria destes colaboradores são despreparados e que realizam uma abordagem invasiva ao realizarem a cobrança aos devedores. Essa situação é constrangedora, e que de acordo com os dados levantados, se comprova totalmente ineficaz. Ainda por cima o alto custo de manutenção deste modelo o torna uma alternativa totalmente ultrapassada para se realizar o serviço proposto pela alternativa nesta peça descrita.

O PagueTudo é uma plataforma multilateral que trabalha com sistemas para dispositivos móveis e também *WEB*. Ela funciona na essência da economia compartilhada em que a tecnologia surge para diminuir os custos operacionais, automatizando tarefas e aproximando partes interessadas. De um lado o credor poderá facilmente lançar os dados dos devedores no sistema e do outro o usuário inadimplente poderá realizar sua negociação com um assistente digital.

2.2 FUNDAMENTAÇÃO TEÓRICA

2.2.1 Inteligência Artificial

O conceito de inteligência já foi visto como uma capacidade inata de todos os indivíduos, uma faculdade herdada desde o nascimento. Entretanto segundo Flavell (1975) realizando um estudo acerca da visão do psicólogo Jean Piaget, relata que para este psicólogo, não herdamos a inteligência em sua totalidade, ao invés disto, nós nascemos com certas capacidades neurológicas e sensoriais, que são o princípio que dá origem as estruturas mentais, ou seja, nós pegamos dos nossos pais biológicos a estrutura que irá se desenvolver de acordo com a influência da interação ao ambiente e dessas perspectiva teremos como resultado estruturas cognitivas que funcionarão de forma semelhante ao dos nossos pais ao longo da vida.

Seguindo esta corrente a partir deste processo que temos como origem da inteligência, Vygotsky traz a luz de que todo ser humano nasce apresentando um potencial atômico cognitivo, esse é o ímpeto que servirá para desenvolver outros, explicando de forma mais clara, é a capacidade de aprender a aprender.

Então para esse artigo nós chegamos a visão de Gardner(1995) que diz que a inteligência:

(...)implica na capacidade de resolver problemas ou elaborar produtos que são importantes em determinado ambiente ou comunidade cultural. A capacidade de resolver problemas permite à pessoa abordar uma situação em que um objetivo deve ser atingido e localizar a solução adequada para esse objetivo.

Então para a plena compreensão do significado convencionamos que inteligência é a capacidade de aprender através de experiências e partindo desse aprendizado resolver problemas. Através da inteligência o indivíduo tem a capacidade de abordar situações cuja a reação é adequada para se chegar a determinado fim utilizando a informação anteriormente aprendida.

Já o sentido de artificial, segundo AURÉLIO (1999), é tudo aquilo feito não de forma natural, mas sim pela mão do homem. Tudo aquilo criado pelo homem. Por definição temos então que a inteligência Artificial é a capacidade de aprender e tomar decisões para resolução de problemas construída e atribuída a um sistema computacional pela mão do homem. É a capacidade não natural de um sistema criado pelo homem de resolver problemas através de aprendizado.

Dentro do contexto da inteligência artificial existem inúmeros algoritmos para os mais variados tipos de necessidade, entretanto o foco neste trabalho é explicar o uso dessas técnicas aplicadas ao software de serviço escopo do estudo.

2.2.2 Machine Learning – Aprendizado de máquina.

Atualmente é um dos campos de maior importância para o mercado teórico e prático da computação. De acordo com Shalev-Shwartz e Ben-David

(2014), o aprendizado de máquina é a capacidade de um computador de converter experiência em conhecimento ou habilidade. Melhor exemplificando, na realidade, isso significa que através dos algoritmos criados para esse aprendizado os softwares conseguem hoje simular a tomada de decisão e a interpretação, até mesmo pelos sentidos, da realidade do próprio cérebro humano.

No sistema PagueTudo, nas entrevistas realizadas ao público alvo ficou claro que um dos maiores problemas que as pessoas enfrentavam era a burocracia para se conseguir informações, muitas vezes isso não estava claro, ou até mesmo não se conseguia falar com um atendente. Então algoritmos de aprendizado de máquina poderiam melhorar o serviço ao entregar para as pessoas acesso a informação de forma instantânea e elas só precisariam pedir por isso. Para criar a funcionalidade ideal para esse fim se faz o uso de um algoritmo de aprendizado de máquina chamado Processamento de Linguagem Natural.

2.2.3 Processamento de Linguagem Natural.

O Processamento de Linguagem Natural ou PNL, para melhor entendermos o que significa o termo é necessário analisarmos primeiramente o sentido da palavra processamento, para AURÉLIO (1999), no sentido aplicado ao ambiente de tecnologia da informação, processar é organizar dados de acordo com uma sequência de instruções codificadas em um programa. Já por Linguagem nós temos um o grande escopo de estudo da linguística. Para AMARAL (2015) ao falar sobre o estudo da linguística no século 20 nos explica:

“No século 20, estudos na área da linguagem possibilitaram chegar a um consenso: a linguagem é uma faculdade humana e depende de um sistema organizado de signos. Um signo é uma unidade dotada de sentido em um código. Pode ser decomposto em um elemento material, que pode ser percebido - pela audição, pela visão, pelo tato e até pelo gosto - e em um elemento conceitual: a ideia.”

Pela palavra natural, também de acordo com AURÉLIO (1999), nós temos que é tudo aquilo que advém da natureza; que faz parte do indivíduo

desde o seu nascimento; congênito, inato. Então temos por definição de que linguagem natural é um sistema organizado de signos, ou símbolos, que representam uma forma de se comunicar inata dos humanos. Para nós esses símbolos são a nossa língua, seja português, inglês, libras ou tantas outras.

Por definição convencionamos que o termo PNL significa a organização dos signos que formam determinada língua de forma coerente a se identificar o sistema de comunicação natural do ser humano através de algoritmos computacionais.

Conduzindo para área da computação a definição de acordo com CALADO (2017) é que PNL é uma subárea da Inteligência Artificial que estuda a capacidade e as limitações de uma máquina para compreender a linguagem natural. Ele nos evidencia que o maior desafio da mesma é além de identificar os símbolos da língua, também precisa fazer uma interpretação analítica sintática, semântica, léxica e morfológica. Além disso é necessário que o algoritmo também possa criar resumos, interpretar os sentidos, analisar sentimentos. Para isso é necessário um algoritmo de aprendizado de máquina que possibilite o software a aprender conceitos com os textos processados.

2.2.4 Computação nas Nuvens

Um dos ramos da tecnologia que dominou o mercado mundial é o da computação distribuída. Que nos trás o conceito de adicionar o poder computacional de várias máquinas conectadas por uma rede. A interligação desses diversos computadores tem o objetivo de compartilhar a execução de tarefas tornando o serviço mais eficaz e eficiente.

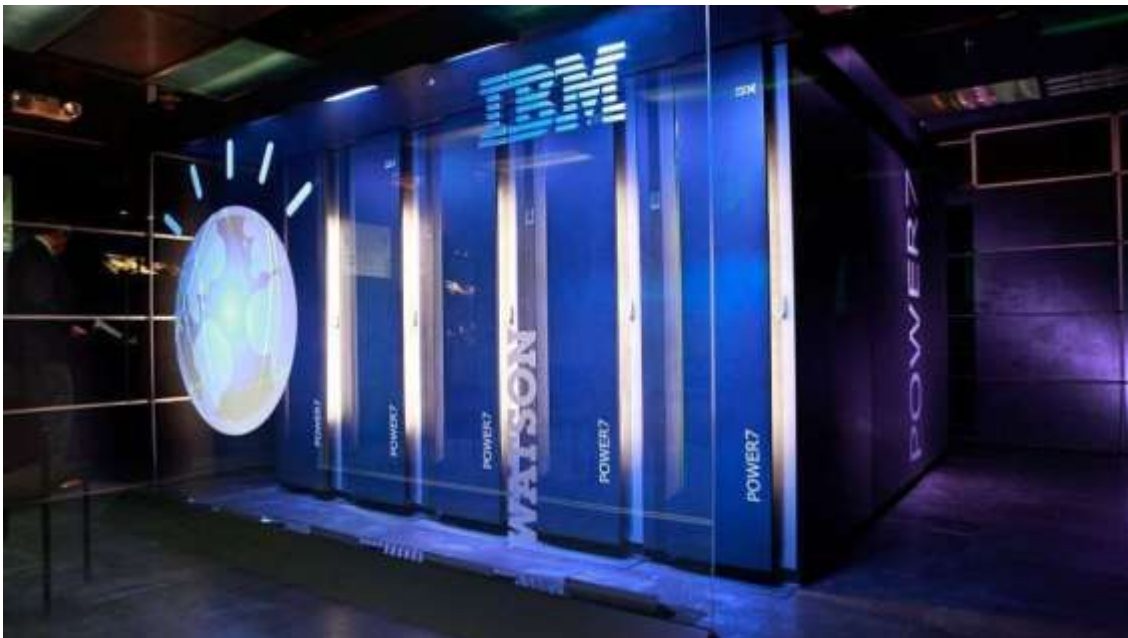
Dentro do escopo da computação distribuída nós temos a computação nas nuvens. Que de acordo com ALECRIM (2015) se refere a noção de utilizarmos, seja onde for e independente de qualquer plataforma, as mais variadas aplicações por meio da rede mundial de computadores. Para o autor é muito importante que as aplicações distribuídas pela internet tenham a mesma facilidade de uso e disponibilidade, ou até mais, do que as que são instaladas em computadores locais.

O grande gargalo das aplicações baseadas na nuvem em contraposição aquelas *onpremise*, ou seja, aquelas instaladas no computador ou no celular do usuário, era a disponibilidade em um ambiente offline. Contudo muito foi feito ao longo dos anos em relação a isso, hoje esse problema é totalmente contornado e não gera mais problemas para o usuário. Pelo contrário, as aplicações nas nuvens oferecem uma grande disponibilidade dos dados, pois eles não ficam restritos apenas ao ambiente em que a aplicação foi instalada.

2.2.5 O IBM Watson

Um dos grandes exemplos de tecnologia hoje criada baseada em todos esses escopos acima referidos é o IBM Watson, uma suíte de soluções de inteligência artificial que engloba aprendizado de máquina, processamento de linguagem natural e tantos outros serviços e tudo baseado nas nuvens.

Figura 1: IBM Watson



Fonte: http://bit.ly/ia_ibmwatson

O Watson é peça fundamental para utilização e viabilidade do presente projeto, pois os custos e tempo para se criar todas as soluções *fromscratch* inviabilizaria o projeto. Na realidade o Watson é um super computador,

conforme imagem 1, equipado com softwares de inteligência artificial construído pela centenária empresa de tecnologia norte americana a Internacional Business Machines – IBM.

2.3 INDICAÇÃO DE METODOLOGIA

A ferramenta PagueTudo, fruto desenvolvido e objeto da descrição deste presente relatório, utiliza a metodologia de desenvolvimento RPROCESS^[1] (LIMA, 2014), que por sua vez é uma simplificação do IBM *RationalUnifiedProcess* – RUP^[2](LEVY e DEVLIN, 1992). Segundo essa estrutura, o desenvolvimento do sistema é dividido em fases, e em cada fase terá um conjunto de componentes-atividades, são elas: Fase de Planejamento, Fase de Construção, Fase de Implementação e fase de Avaliação/Manutenção.

2.4 PROCEDIMENTOS ADOTADOS

2.4.1 Métodos

No intuito de elucidar os métodos de desenvolvimento utilizados desde a idealização da ferramenta até a sua definitiva implantação ao mercado, segue as técnicas e processos escolhidos e o motivo pelo qual assim o foram.

Para a produção da plataforma foi usado o método de desenvolvimento Iterativo Incremental proposto por Barry Boehm(1988)^[3], e citado na obra de PRESSMAN(2016)^[4]. Segue também os princípios do desenvolvimento ágil de softwares, visto que em detrimento de uma abordagem mais tradicional e engessada como o caso do Modelo em Cascata criado por W.W. Royce^[5], citado também por Pressman, tal abordagem, a iterativa incremental, é mais dinâmica e possibilita a entrega de incrementos levando valor de forma mais rápida as partes interessadas.

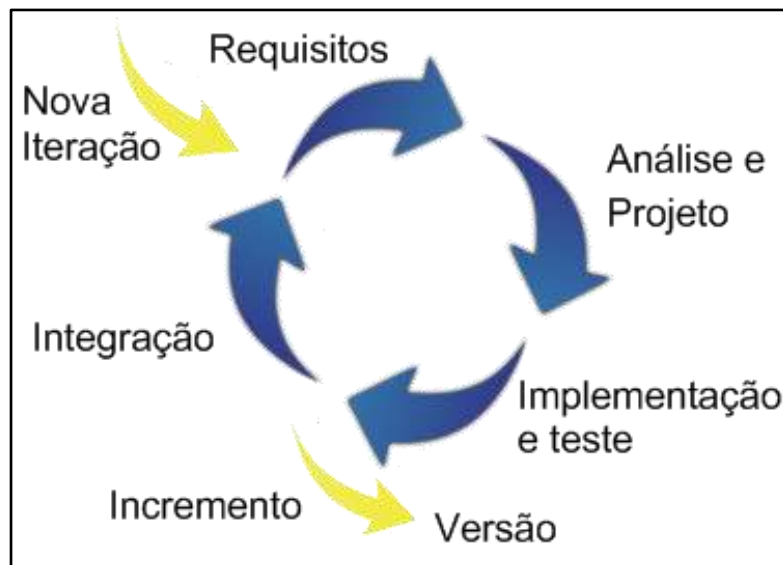
O desenvolvimento no modelo escolhido é dividido em etapas denominadas “incrementos”, que uma vez prontas e testadas, são implementadas ao sistema e já entregam funcionalidades aos *stakeholders* ou

interveniente. Cada incremento é implementado como se fosse um software em sua totalidade, passando por todas as etapas do ciclo de vida de desenvolvimento, desde a modelagem de negócios até os testes, que por sua vez geram componentes, ou partes de componentes que podem ser montados no sistema.

O modelo de desenvolvimento Iterativo Incremental nos direciona a experimentação e as rotineiras mudanças em fases distintas no projeto, diferentemente do caminho encorajado no ciclo de vida do modelo Cascata, ou em outros que seguem um princípio estruturado (PRESSMAN, 2016, p.215), onde cada fase só deve começar quando a anterior é finalizada completamente. O modelo supracitado como escolhido para esta obra é totalmente colaborativo, visto que todas as partes intervenientes podem interagir em qualquer fase do desenvolvimento, pois que em cada implantação de um incremento ao sistema, o cliente pode ir acompanhando e homologando da forma como for necessária.

Na figura 2 é possível perceber o funcionamento do ciclo de desenvolvimento de software via a abordagem iterativo incremental.

Figura 2: Abordagem de desenvolvimento iterativa incremental



Fonte: <http://bit.ly/2nTN28o>

Para uma melhor gestão do tempo, foi decidido seguir os direcionamentos consagrados no manifesto ágil. O mesmo é composto por

valores e princípios que regem a atitude perante o desenvolvimento do software. Dentre as diretrizes, nós temos quatro valores que dão um norte a esse manifesto, são eles:

- Os indivíduos e suas interações acima de procedimentos e ferramentas;
- Funcionamento do software acima de documentação abrangente;
- A colaboração dos clientes acima de negociação de contratos;
- A capacidade de resposta a mudanças acima de um plano pré-estabelecido;

Não se trata, como poderia parecer à primeira vista, de um desprezo aos elementos e ferramentas tradicionais do desenvolvimento de software, mas sim do estabelecimento de uma escala de valores, na qual a flexibilidade e a colaboração são mais relevantes do que a rigidez de processos e planejamentos clássicos.

Em cima desses valores centrais, nós temos oito princípios que ditam o caminho a ser seguido a quem deseja implantar a essência do movimento ágil, segue os princípios:

- Garantir a satisfação do cliente, entregando rápida e continuamente software funcionais;
- Software funcionais são entregues frequentemente (semanal, ao invés de mensal);
- Software funcionais são a principal medida de progresso do projeto;
- Até mesmo mudanças tardias de escopo do projeto são bem-vindas;
- Cooperação constante entre as pessoas que entendem do 'negócio' e os desenvolvedores;
- Projetos surgem por meio de indivíduos motivados, devendo existir uma relação de confiança;
- Design do software deve prezar pela excelência técnica;
- Simplicidade;
- Rápida adaptação às mudanças;

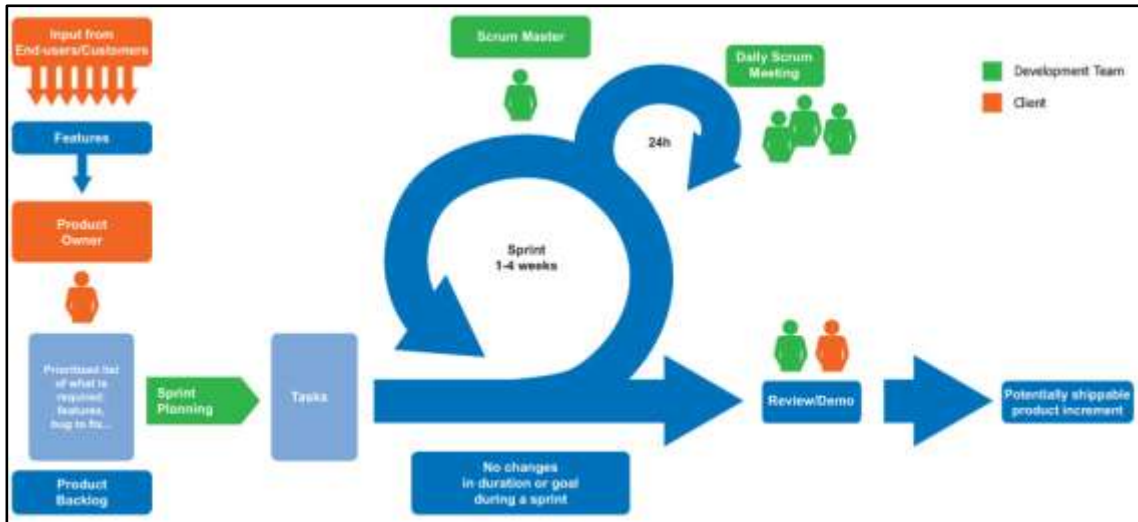
Só a título de curiosidade, o manifesto ágil contou inicialmente com 17(dezessete) signatários originais: Kent Beck, Mike Beedle, Arievan Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, AndrewHunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland e Dave Thomas. Atualmente diversas empresas ao redor do mundo já adotam estes princípios e até mesmo metodologias inteiras foram criadas a partir deste manifesto.

Dentre as várias metodologias consolidadas no mercado, e que foram criadas baseadas no manifesto ágil, foi escolhido para o presente trabalho a utilização do framework SCRUM. Ferramenta que possui foco no gerenciamento e projeto da organização onde é difícil planejar com o que ainda está por vir. Esta ferramenta não traz diretrizes de um processo descritivo, em que se direciona o que se fazer em cada situação, mas é utilizado para trabalhos complexos onde fica impossível prever a totalidade de tudo que será feito. O SCRUM traz em seu escopo um compêndio de valores, princípios e práticas que fornecem a base para que o projeto possa ser realizado de forma ágil. Princípios e ferramentas que foram utilizados neste projeto.

Existem algumas nomenclaturas utilizadas para separar os papéis que são: *ProductOwner* que é o cliente propriamente dito, ou o seu representante. O *team*, ou a equipe, é preferencialmente formada por poucas e multidisciplinares pessoas, pois equipes pequenas são comprovadamente mais eficientes. Quem gerencia todo o processo é o *ScrumMaster*. Alguns artefatos são nomeados para o uso eficaz do processo, o *Product Backlog*, ou a lista com requisitos e suas prioridades definidas pelo *ProductOwner*. *Sprint Backlog*, consiste em uma lista de produtos selecionados do *Product Backlog*. *Sprints*, é o desenvolvimento de um incremento de um software a ser entregue em determinado prazo. *Daily Scrum*, são reuniões diárias realizadas para resolver as questões envolvidas no desenvolvimento de determinada Sprint.

Para uma melhor visualização do conceito abordado pelo *framework*, a figura 3 mostra a arquitetura de todo o processo proposto pelo SCRUM.

Figura 3: Processo passo a passo do SCRUM



Fonte: <http://www.nutcache.com/blog/how-to-use-scrum-to-boost-teams-productivity>

Seguindo os ditames propostos pelo paradigma na figura 1 deste trabalho técnico, o PagueTudo utiliza a metodologia iterativo incremental, na qual também se encaixa o *Framework* Scrum para gerenciamento ágil do desenvolvimento, conforme os processos evidenciados na figura 2. Entretanto, na fase de concepção e planejamento descritas a seguir, foi decidido fazer algumas readequações para um melhor desempenho. O processo utilizado é dividido em 4 fases descritas abaixo:

a) Concepção - Fase em que se define todas as funcionalidades do sistema, como também o seu escopo e modelagem de negócio. A seguir estão definidos todos os passos da definição dos requisitos e modelagem.

- Técnicas usadas para o levantamento de requisitos e modelagem de negocio:

I. Entrevistas: Como o próprio nome já diz, é a forma em que o analista se comunica com as partes interessadas, seja através do meio verbal ou escrito. É um método primitivo de colher informações, entretanto ainda se mostra um dos mais eficazes.

II. Observação: É o método que consiste na percepção direta do observador sobre os fenômenos que acontecem ao redor. É realizada no intuito de colher informações mais criteriosas sobre o problema a ser resolvido.

III. Estudo da documentação: É um método onde o analista pesquisa a documentação disponível sobre o assunto, buscando criar conhecimento válido sobre o tema, que por sua vez deverá gerar valor ao desenvolvimento das regras de negócio do sistema.

b) Elaboração – Fase em que é definida toda a arquitetura do sistema, assim como o projeto, buscando complementar a documentação dos casos de uso voltado para a arquitetura do sistema.

- Técnica utilizada para a modelagem do Sistema:

- Linguagem de Modelagem Unificada (UML): De acordo com um dos criadores da linguagem o engenheiro de softwares GradyBooch:

Método usado para visualizar a estruturação e o comportamento desejado que o sistema deva apresentar. O UML Provê a notação gráfica para a geração de modelos visuais que facilitam o entendimento por parte dos desenvolvedores de parte ou de toda a estrutura do sistema (BOOCH et al., 2000)

c) Construção – Fase em que se realiza o desenvolvimento físico do sistema, produção de código, testes alfa.

d) Entrega Final e Evolução – Fase em que o sistema é testado na sua totalidade, é feito o plano de entrega e finalmente implantado. Começa então a atividade de refinamento de modo a atender as modificações das necessidades do cliente, prática possível graças ao modelo iterativo incremental.

2.4.2 Técnicas

Dada a natureza da própria arquitetura dos sistemas de informações para o consumidor final, o mercado atualmente utiliza uma divisão clara entre duas frentes no desenvolvimento de softwares: *front-end* (parte da frente, em tradução livre) (EIS, Diego. 2015), que consiste em todo o desenho da interface

e desenvolvimento de telas baseadas no estudo da experiência do usuário. Como também o *back-end* (parte de trás, em tradução livre) (EIS, Diego. 2015), onde acontece a codificação em linguagem de programação na parte do servidor. As técnicas usadas seguem essa divisão clara, todavia abaixo, segue-se um modelo para apresentar como e onde cada uma delas foi utilizada no presente sistema, em vista que algumas informações estão contidas antes do desenvolvimento que se enquadra no modelo *front-end/back-end*. Segue:

Concepção – Na fase de concepção foram utilizadas técnicas de entrevistas e observações para a coleta dos dados, e já para a análise dos requisitos foi utilizada técnicas oriundas do RPROCESS (LIMA, 2014).

Elaboração – Na arquitetura do sistema, para a modelagem geral foi utilizado técnicas padronizadas do UML, que resultou na documentação necessária para a geração dos artefatos. Para o banco de dados, encontrado na esfera *back-end*, o planejamento e modelagem foi feito seguindo as técnicas de normalizações consagradas e vastamente disponíveis na literatura, como também a técnica relacional de armazenar os dados, em que as informações são tidas como entidades e agrupadas conforme os seus relacionamentos. A modelagem de negocio foi revista usando técnicas inspiradas nos ditames do *RationalUnifiedProcess* (LEVY, DEVLIN, 1981), ou como chamam também de RUP. Já a arquitetura do *front-end* foi modelada utilizando a técnica de pesquisa de mercado nos exemplos de interfaces e navegações de produtos similares.

Construção - Para a construção do sistema, foi utilizado técnicas de padronização e documentação de código de programação, como também a escolha do paradigma de programação orientada a objetos (CARVALHO, Thiago, 2015.) Para o desenvolvimento da parte *mobile*, paradigma onde os conceitos de abstração, encapsulamento herança e polimorfismo são utilizados. Na esfera do *front-end* o design foi inspirado nas técnicas minimalistas, onde se procura dizer muito com pouca informação. Para a criação das telas propriamente ditas foi seguida as técnicas fornecidas pelo Material Design (GOOGLE, 2014). Na gestão dos processos as técnicas do SCRUM foram utilizadas, assim como as entregas dos artefatos ao final de cada *sprint*.








Entrega Final e Evolução - Para a entrega final foram utilizadas as técnicas também disponibilizadas pelo RUP, onde se construiu manual do usuário e o plano de implantação. O plano de gestão de mudanças foi feito com as técnicas oferecidas pela obra de Boehm(1988), para o refinamento e modificação dos requisitos e de outros componentes do sistema, em que porventura seja solicitado pelos clientes.









2.4.3. Ferramentas

Para o desenvolvimento do PagueTudo, foram utilizadas várias ferramentas. Desde um editor de texto até servidores web e sistemas de gerenciamento de banco de dados, inclusive linguagens de programação, modelagem, marcação, estilos e outras. Segue na figura 4 a descrição de todas as que foram utilizadas:

Figura 4: Tabela de ferramentas

 Photoshop	Ferramenta de edição de imagens	http://www.photoshop.com/
 node.js	Servidor para hospedagem de aplicações back-end javascript	https://nodejs.org/en/

Ferramentas	Breve descrição	
	Linguagem de marcação usada na construção das interfaces front-end	http://www.w3schools.com/html/
	Tecnologia para desenvolvimento de folhas de estilo(Cascading Style Sheets) para auxílio no design das interfaces front-end	http://www.w3schools.com/html/html_css.asp
	Linguagem de programação client-side, para o tratamento de formulários ou partes dinâmicas do front-end	https://www.javascript.com/
	Linguagem de programação server-side para a parte lógica de uma página web.	https://secure.php.net/manual/pt_BR/index.php
	Programa desenvolvido em PHP que auxilia no desenvolvimento dos bancos de dados	https://www.phpmyadmin.net/
	ferramenta de banco de dados relacional	https://www.mysql.com/
	Sistema de gerenciamento de banco de dados relacional mysql	http://www.mysql.com/products/workbench/

	ferramenta de geração dos diagramas UML	http://astah.net/
	Suite de edição de textos, planilhas e apresentação	https://www.microsoft.com/pt-br/download/office.aspx
	Editor de textos programado para HTML, CSS e JavaScript	https://www.sublimetext.com/
	Ferramenta para desenvolvimento na plataforma Android	https://developer.android.com/studio/index.html?hl=pt-br
	Ferramenta para desenvolvimento na plataforma IOS	https://developer.apple.com/xcode/
	Linguagem de programação para desenvolvimento da plataforma Android	https://www.java.com/pt_BR/
	Linguagem de programação para desenvolvimento da plataforma IOS	http://www.swift.com.br/
	Ferramenta para desenvolvimento com a linguagem PHP	https://www.jetbrains.com/phpstorm/

Fonte: Próprio Autor

3 RESULTADO

Com a efetivação do uso de todas as técnicas, métodos e ferramentas anteriormente citados, temos por esperado o resultado de atender os objetivos gerais e específicos elencados nos itens 1.1.1 e 1.1.2 deste relatório técnico. A finalidade é atender as expectativas do usuário e todas as partes interessadas, gerando valor com usabilidade simples e intuitiva.

3.1 DECLARAÇÃO DE ESCOPO

O PagueTudo é um sistema construído como plataforma de *fintech* para facilitar a cobrança por parte dos credores e pagamento por parte dos devedores. Utilizando sistemas web integrados com aplicação mobile e inteligência artificial.

3.2 IDENTIFICAÇÃO DOS REQUISITOS

Conforme PRESSMAN (2016), a identificação dos requisitos é o resultado da especificação das características funcionais e não funcionais do sistema de software. Logo é um meio pelo qual o analista levanta todas as necessidades que o sistema deverá resolver, assim como as necessidades não funcionais que o sistema precisará oferecer. É o roteiro que foi seguido com norte para a especificação das necessidades do PagueTudo.

3.2.1 Técnica de Levantamento de Requisitos

A primeira técnica utilizada para o levantamento dos requisitos foram as entrevistas, que segundo a literatura é uma das várias formas de se facilitar a comunicação entre os desenvolvedores e os usuários. Já de acordo com SANTOS (2015), para realização de uma boa entrevista, é indispensável a organização de um roteiro, em outras palavras um planejamento da entrevista.

Também é primordial a escolha dos entrevistados alvo, pois é importante para o contexto específico. Para esses requisitos foram entrevistadas pessoas que já passaram pelo constrangimento de receberem cobranças por parte das dos escritórios e também donos de negócios que sofrem com a inadimplência.

3.2.2 Descrição Simples dos Requisitos

A descrição simples dos Requisitos é uma visão geral e objetiva de cada uma das funcionalidades do sistema. Abaixo seguem as figuras 5 e 6 com as tabelas com os requisitos funcionais e não funcionais, assim como o seu identificador, dependências e descrição de prioridades para o PagueTudo.

Figura 05 – Requisitos Funcionais

IDENTIFICADOR	REQUISITOS DEPENDENTES	DESCRIÇÃO DOS REQUISITOS FUNCIONAIS	PRIORIDADES
RF001	NENHUM	Registrar débitos por parte das empresas conveniadas	Essencial
RF002	RF001	O usuário poderá acessar os debitos cadastrados em seu cpf	Essencial
RF003	RF002	O usuário poderá aceitar as propostas de acordos por parte da empresa	Essencial
RF004	RF002	oferecer novas propostas para os seus debitos listados	Essencial
RF005	NENHUM	Interface de assistente virtual com inteligencia artificial	Essencial
RF006	RF001, RF002, RF003, RF004	Gerenciar propostas e acordos pela empresa	Essencial
RF007	Todos	Gerar relatórios para as empresas	Essencial

Fonte: Próprio Autor

Figura 06 – Requisitos não Funcionais

IDENTIFICADOR	REQUISITOS DEPENDENTES	REQUISITOS	DESCRIÇÃO DOS REQUISITOS FUNCIONAIS
RNF001	NENHUM	Usabilidade	Um novo usuário deverá saber utilizar o sistema intuitivamente
RNF002	NENHUM	Banco de dados	Mysql e MongoDB
RNF003	NENHUM	Compatibilidade	poder cadastrar os debitos em pelo menos 90% dos formatos de

Fonte: Próprio Autor

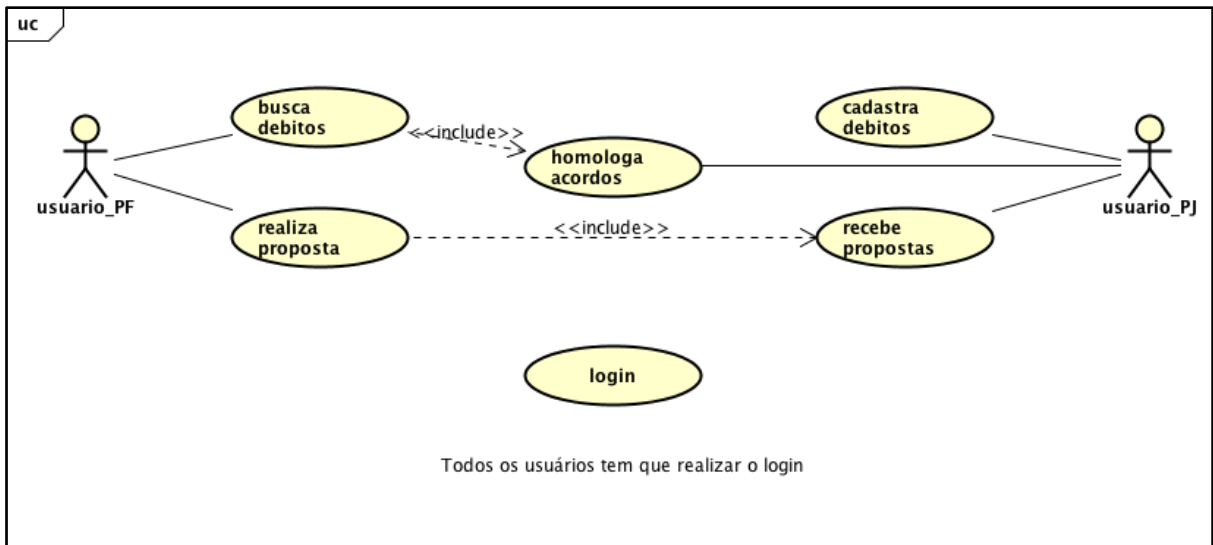
3.3 ANÁLISE DE REQUISITOS

De acordo com Pressman (2016, p.167) a Análise de Requisitos resulta na especificação das características operacionais do software, indica a interface do software com outros elementos do sistema e estabelece restrições a que o software deve atender.

3.3.1 Diagrama de Caso de Uso – UC

Este diagrama é uma das diretrizes consagradas no UML mencionada já anteriormente. Utilizada mundialmente e tem como um dos seus principais objetivos a descrição de apoiar “o cliente, os usuários e os desenvolvedores a obterem acordo no modo de uso do sistema” (JACOBSON; BOOCH; RUMBAUGH, 2001, P.40).

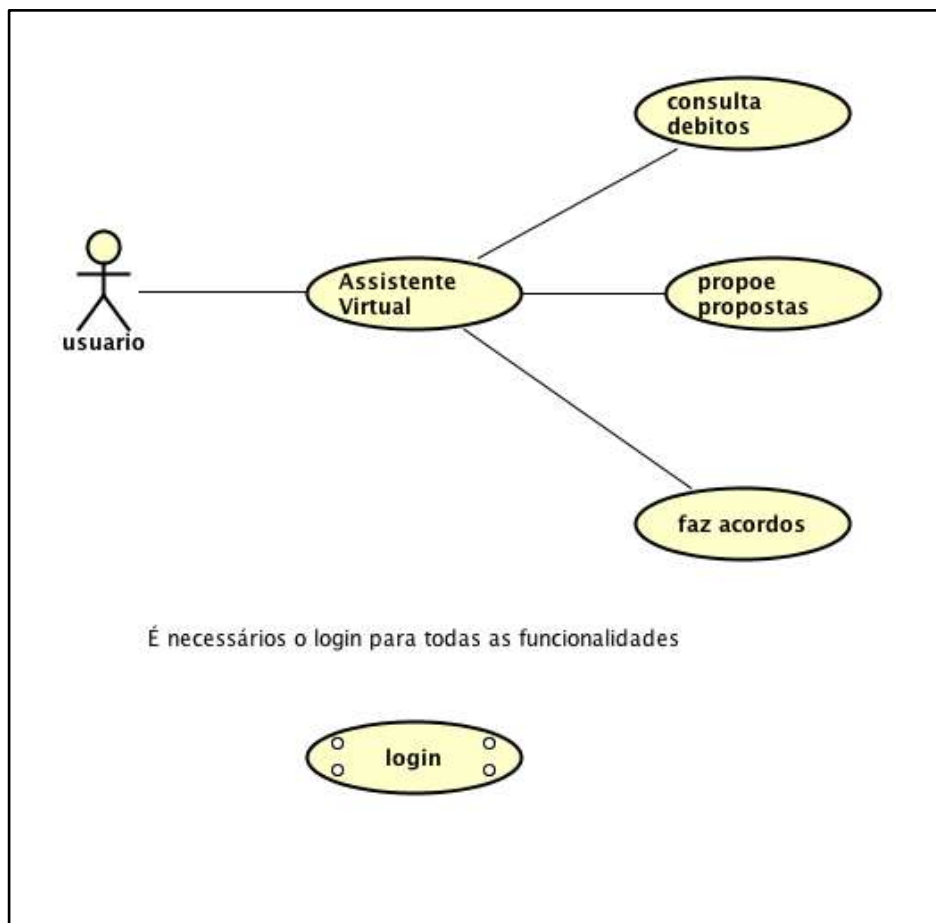
Figura 07 – Diagrama de Casos de Uso– Sistema Web



Fonte: Próprio Autor.

Nas figuras 07 e 08, são apresentados de forma simplificada, os diagramas de uso, desenvolvidos para a aplicação web e para o Sistema Mobile da aplicação PagueTudo.

Figura 08 – Diagrama de Casos de Uso – Sistema Mobile



Fonte: Próprio Autor

3.3.2 Especificação e descrição dos casos de uso.

Uma especificação de caso de uso descreve uma sequência de ações “com o objetivo de demonstrar o comportamento do sistema (ou parte dele), através de interações com atores”. (MELO, 2002).

Cada especificação de caso de uso é o detalhamento de um ou mais requisitos funcionais do sistema e sua descrição deve ser de alto nível. Embora os clientes precisem validar o fluxo de eventos dos casos de uso, outras pessoas também farão uso dos mesmos. Dentre elas, estão gestores, arquitetos de software, analistas, especificadores, implementadores e testadores. Dessa forma, este artefato pode ser refinado até atingir um nível de detalhamento que atenda às necessidades das pessoas envolvidas no projeto. (RUP, 2017)

3.3.2.1 Sistema web

Caso de uso: (UC 01) – Realizar autenticação

- **Atores:** usuario_PF e usuario_PJ
- **Descrição do caso de uso:** Para acessar a plataforma os atores deverão se cadastrar no sistema, oferecendo suas credenciais que incluem nome e CPF, para os usuario_PF e CNPJ para o usuario_PJ, além do E-mail que será usado como Usuário para autenticação.
- **Pré-condição:** Nenhuma
- **Pós-condição:** Ter acesso ao sistema.

Caso de uso: (UC 02) – Buscar Débitos

- **Ator:** usuario_PF
- **Descrição do caso de uso:** Este caso de uso refere-se à funcionalidade do usuário efetuar a busca pelos débitos que ele

procura quitar usando o sistema. Esses débitos serão cadastrados pelas empresas, e outros credores conveniados com a plataforma.

- **Pré-condição:** o ator ter feito a autenticação no sistema
- **Pós-condição:** apresentar a lista dos débitos vinculados ao CPF.

Caso de uso: (UC 03) – Homologa Acordos

- **Ator:** usuario_PF
- **Descrição do caso de uso:** Uma vez localizados os débitos, o ator pode enxergar as condições oferecidas pelas empresas ou credores, caso aceite os termos ali oferecidos ele pode aceitar e homologar o acordo.
- **Pré-condição:** o ator ter feito a busca pelos seus débitos.
- **Pós-condição:** apresentar termos, opções de pagamentos e enviar acordo.

Caso de uso: (UC 03/1) – Homologa Acordos

- **Ator:** usuario_PJ
- **Descrição do caso de uso:** Quando os devedores aceitarem os termos dos acordos e realizarem o pagamento o ator deverá receber a confirmação e o pedido para que se dê baixa no cadastro do debito.
- **Pré-condição:** o devedor ter realizado acordo e pago.
- **Pós-condição:** dar baixa no débito agora quitado.

Caso de uso: (UC 04) – Realiza Proposta

- **Ator:** usuario_PF
- **Descrição do caso de uso:** Caso o ator não esteja satisfeito com as condições oferecidas para pagamento do debito, ele pode fazer uma proposta conforme deseje. Também pode entrar em contato com o remetente daquele debito para maiores informações.
- **Pré-condição:** O ator ter feito busca pelos seus débitos.

- **Pós-condição:** Abrir tela para proposta e meios de contatos.

Caso de uso: (UC 05) – Cadastra Débitos

- **Ator:** usuario_PJ
- **Descrição do caso de uso:** Neste caso de uso o ator poderá cadastrar no sistema todos os débitos que as pessoas têm para com ele.
- **Pré-condição:** o ator ter feito a autenticação no sistema
- **Pós-condição:** apresentar a lista dos débitos pagos e não pagos.

Caso de uso: (UC 06) – Recebe Proposta

- **Ator:** usuario_PJ
- **Descrição do caso de uso:** Apresentar para o ator todos as propostas enviadas acerca dos débitos que as pessoas têm com ele.
- **Pré-condição:** o UC 4 ter sido feito pelo usuario_PF.
- **Pós-condição:** Não tem.

3.3.2.2 Sistema Mobile

Caso de uso: (UC 01) – Realiza Autenticação

- **Ator:** usuario
- **Descrição do caso de uso:** Para acessar a plataforma o ator deverá se cadastrar no sistema, oferecendo suas credenciais que incluem nome e CPF, além do E-mail que será usado como Usuário para autenticação.
- **Pré-condição:** Não tem
- **Pós-condição:** Ter acesso ao sistema.

Caso de uso: (UC 02) – Assistente Virtual

- **Ator:** usuario
- **Descrição do caso de uso:** Utilizando os algoritmos de inteligência artificial, o ator poderá interagir com o sistema por meio de comandos de voz. Onde ele poderá acessar as funcionalidades do sistema de forma intuitiva.
- **Pré-condição:** Ter feito a autenticação no sistema
- **Pós-condição:** Apresentar a funcionalidade solicitada pelo ator.

Caso de uso: (UC 03) – Consulta Débitos

- **Ator:** usuario
- **Descrição do caso de uso:** Ao ser solicitado, deve-se apresentar todos os débitos vinculados ao CPF do ator na tela.
- **Pré-condição:** Ter feito autenticação no sistema
- **Pós-condição:** Apresentar os débitos em tela.
-

Caso de uso: (UC 04) – Faz Acordos

- **Ator:** usuario
- **Descrição do caso de uso:** Uma vez tendo acesso aos débitos, o ator pode ver quais são os termos oferecidos pelo credor.
- **Pré-condição:** Ter consultado os débitos
- **Pós-condição:** Não tem.

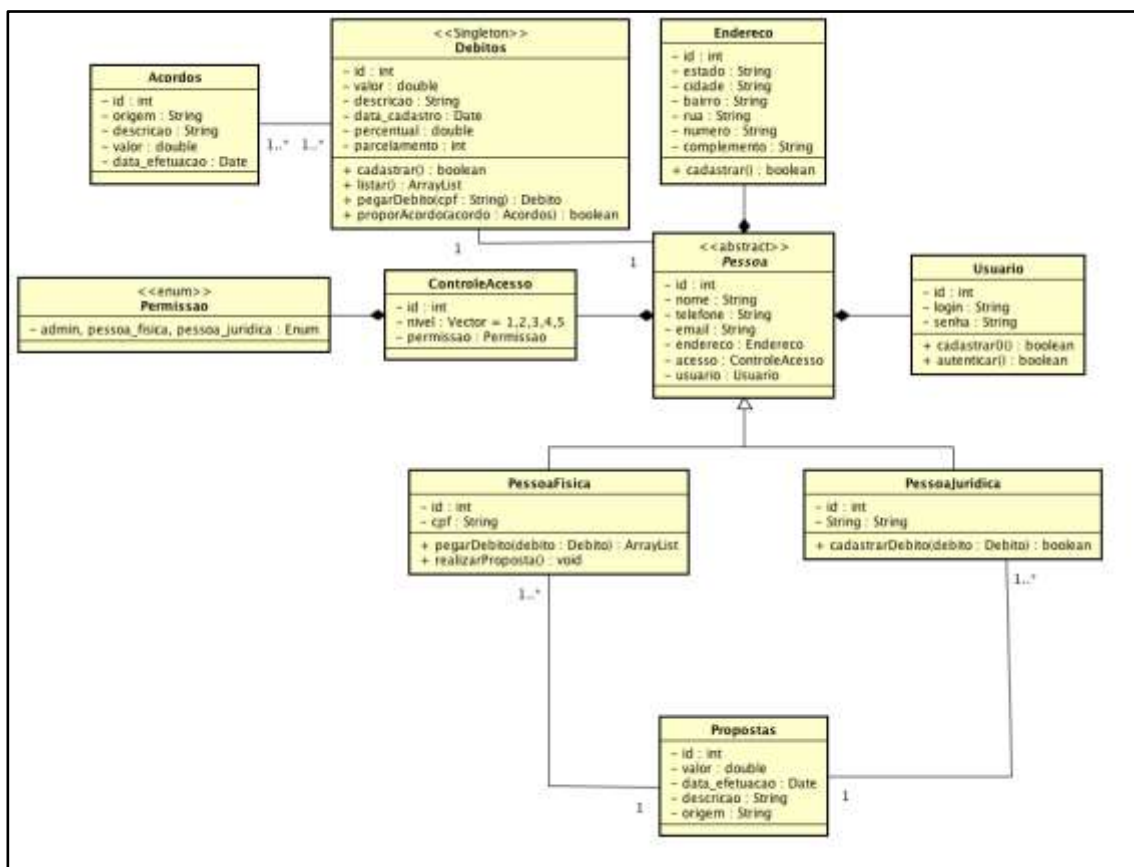
Caso de uso: (UC 05) – Propõe Propostas

- **Ator:** usuario
- **Descrição do caso de uso:** Para acessar a plataforma o ator deverá se cadastrar no sistema, oferecendo suas credenciais que incluem nome e CPF, além do E-mail que será usado como Usuário para autenticação.
- **Pré-condição:** Não tem
- **Pós-condição:** Ter acesso ao sistema.

3.4 DIAGRAMA E CLASSES

De acordo com Fowler (2005), um diagrama de classe descreve os tipos de objetos presentes no sistema e os vários tipos de relacionamentos existentes entre eles, além de mostrar as propriedades e as operações de uma classe e as restrições que se aplicam a estes relacionamentos. Na *UnifiedModelingLanguage* (UML) na parte em que se evidência o diagrama de classe, a classe é representada por um retângulo subdividido em três partes: Nome da Classe, Atributos e Métodos. A maior importância de se encontrar as classes do projeto é que cada classe representa fisicamente uma tabela no banco de dados, logo a modelagem do sistema fica mais coerente e a implementação mais simples.

Figura 9: Diagrama de Classes

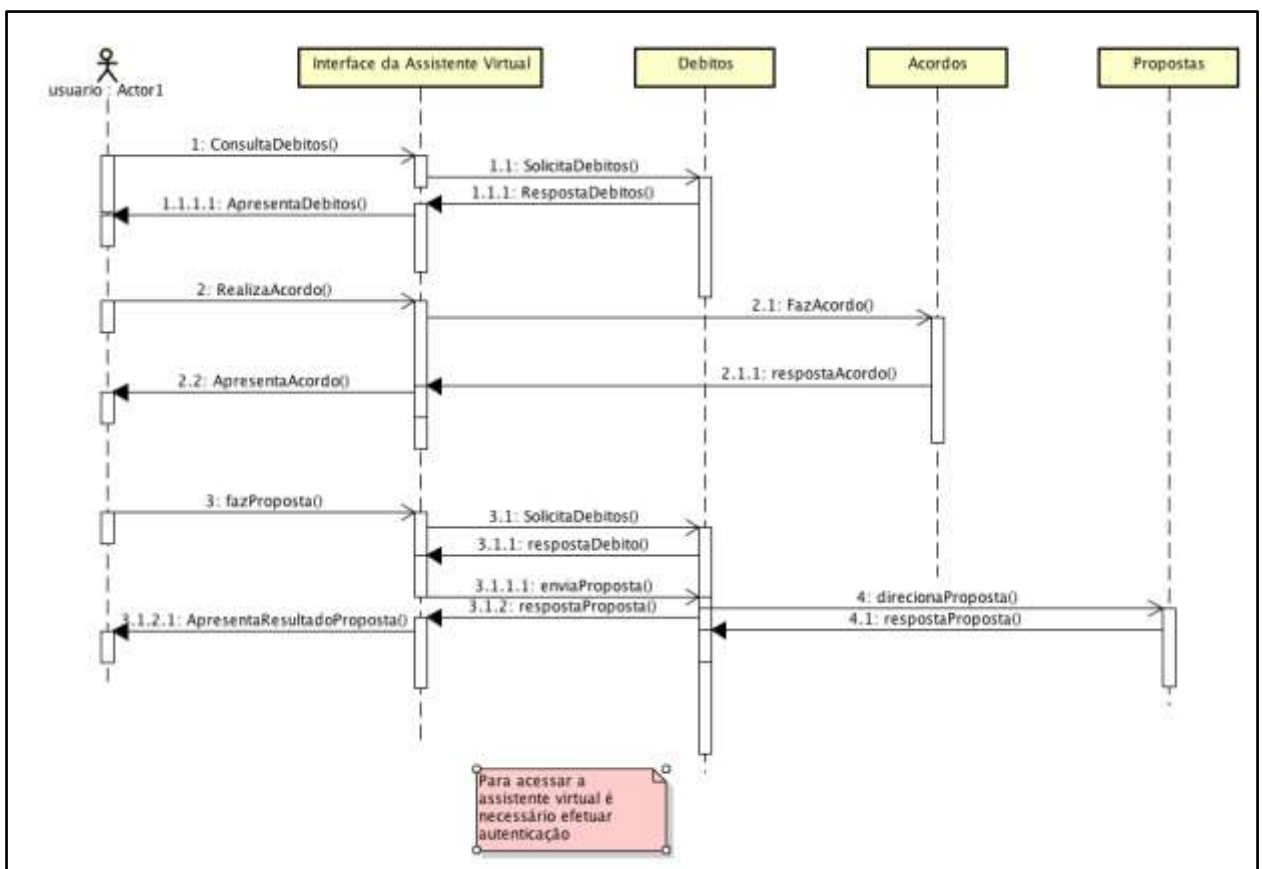


Fonte: Próprio Autor

3.5 DIAGRAMA DE SEQUÊNCIA

De acordo com CRISTINA (2014), o Diagrama de Sequência é o principal dos quatro diagramas de interação. Que são: Diagrama de Comunicação, Visão e Temporal. Logo é justo evidenciar qual o conceito geral antes de entrar no específico. Com este intuito a responsabilidade central destes diagramas é mostrar a interação entre os objetos de um sistema por meio de uma visão dinâmica. A interação entre os objetos é representada por meio de mensagens. Ao se identificar as mensagens, estamos identificados os serviços oferecidos pelas classes, que ao serem identificados podemos elencar quais os métodos necessários em cada classe. Ainda segundo Cristina, Diagrama de Sequência é aquele que enfatiza a troca de mensagens dentro de uma linha de tempo sequencial. A seguir na figura 10 é apresentado o diagrama de sequência utilizado para a construção do PagueTudo.

Figura 10: Diagrama de Sequência da aplicação Mobile do PagueTudo



Fonte: Próprio Autor

3.6 PROJETO DE BANCO DE DADOS

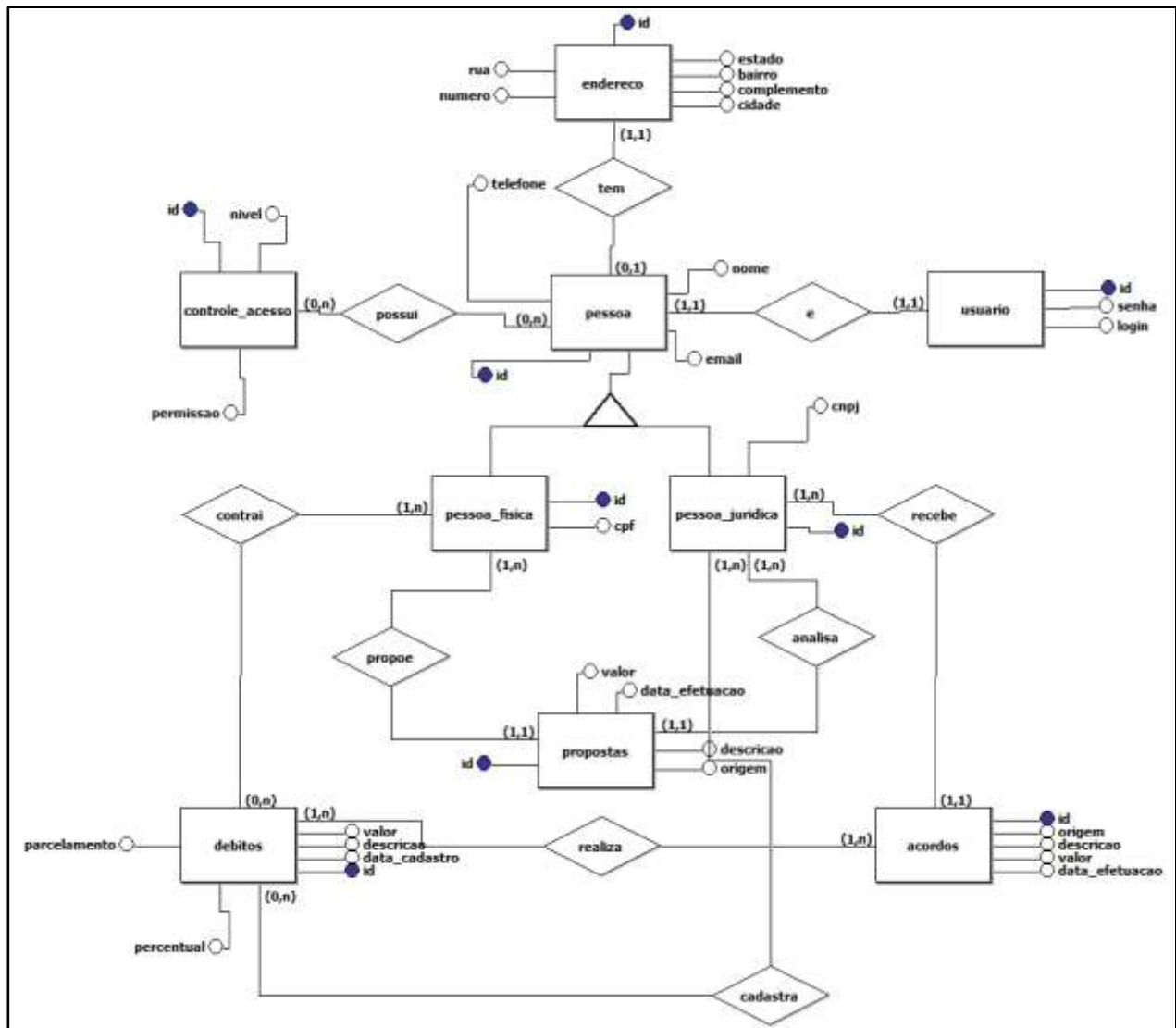
Para melhor entendimento o projeto de banco de dados foi construído usando o Modelo Entidade Relacionamento (MER). Onde temos por Entidade todo o conteúdo concreto da realidade do dia-a-dia. Trazendo para o mundo da computação, podemos entender como uma abstração do conceito de objetos do paradigma de programação orientado a objetos. Como por exemplo usuário é uma Entidade que tem como objeto pai a entidade Pessoa. Relacionamento é a descrição de como essas entidades se relacionam no escopo da aplicação. De acordo com CHEN (1990), para se tornar ainda mais claro, o projeto de banco de dados deve ser apresentado em subseções, modelo conceitual, modelo lógico, modelo físico e dicionário de dados, e assim é feito nesse relatório.

O modelo conceitual compõe os detalhes das entidades, seus relacionamentos e atributos; o modelo lógico é uma descrição das estruturas que são armazenadas, que inclui a nomeação de componentes; o modelo físico é uma adequação dos dados modelados nos modelos anteriores as restrições do SGBD escolhido; por ultimo, o dicionário de dados, onde se apresenta as definições e representações de elementos de dados, ou seja, a representação das tabelas e seu conjunto de campos bem como restrições.

3.6.1 Modelo Conceitual

O Modelo Conceitual é formado a partir do projeto conceitual, que é “a descrição concisa dos requisitos de dados dos usuários e inclui detalhes dos tipos de entidades, relacionamentos e restrições.”. (ELMASRI & NAVATHE, 2011). É um diagrama composto por figuras geométricas, onde a forma do quadrado representa as Entidades e o losango os Relacionamentos. Os Atributos que de acordo com RODRIGUES (2016), são todas as características que descrevem cada entidade dentro do domínio, logo são os valores que em sua totalidade representem o conceito concreto da Entidade.

Figura 11 – Modelo Conceitual de Banco de Dados do PagueTudo



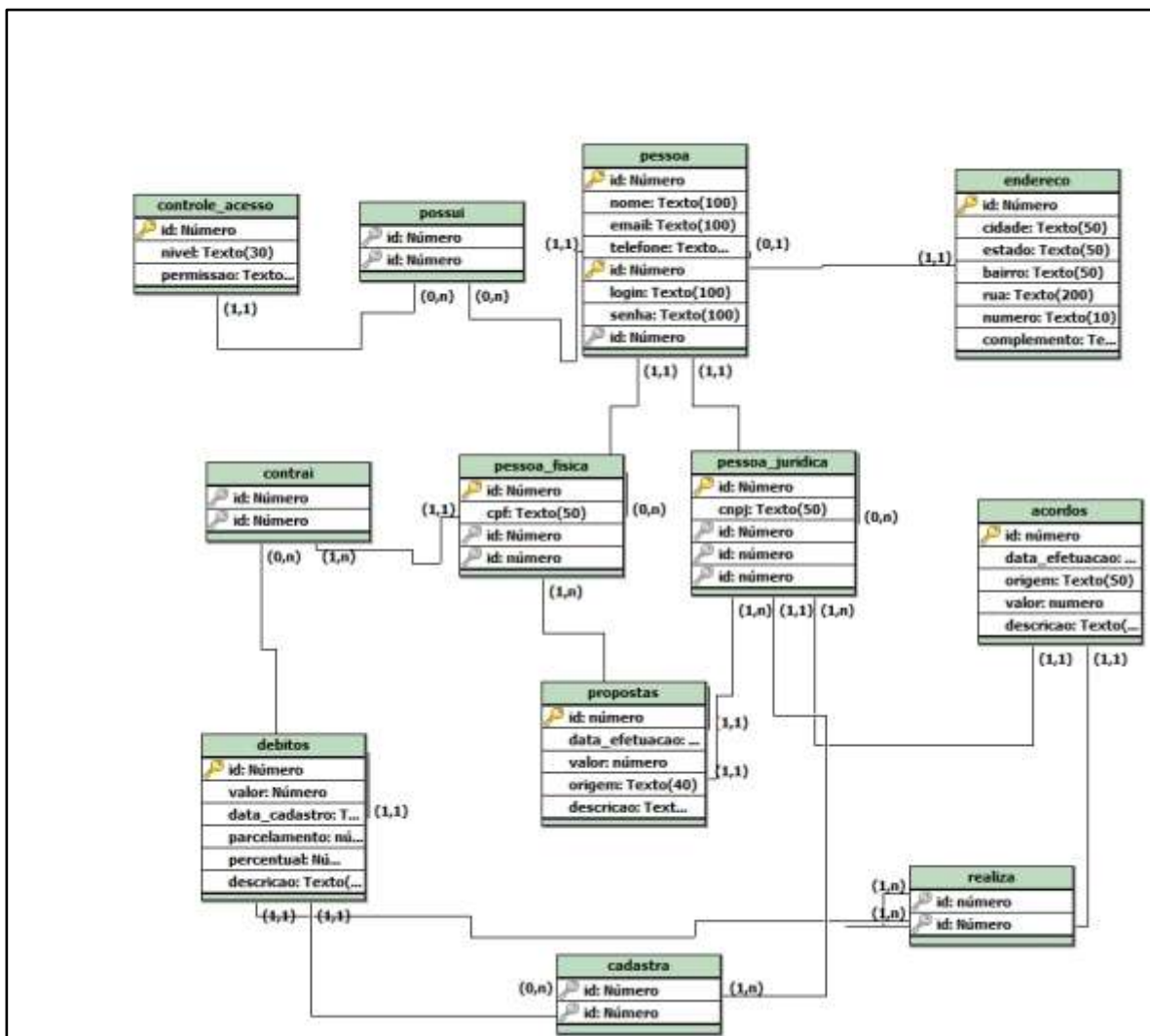
Fonte: Próprio Autor

Temos ainda a cardinalidade, que segundo VIANA (2015), é um número que expressa o comportamento (número de ocorrências) de determinada entidade associada a uma ocorrência da entidade em questão através do relacionamento. Na figura 11 é apresentado o modelo conceitual do PagueTudo.

3.6.2 Modelo Lógico

Já o modelo lógico implementa recursos como adequações de padrões e nomenclatura, define as chaves primárias e estrangeiras que seguindo os dizeres do Analista de Tecnologia da Informação MACÊDO (2011), chave primária se refere a um ou mais atributos, cujo os valores, nunca se repetem e que podem ser usados como um índice para os demais campos da tabela do banco de dados. Já a chave estrangeira, ainda segundo o Analista, é um campo, ou atributo utilizado para criar os relacionamentos entre as tabelas.

Figura 12 – Modelo Lógico de dados do PagueTudo.



Fonte: Próprio Autor

O modelo lógico ainda implementa conceitos de normalização e integridade referencial, que fogem do escopo de explicação deste trabalho, mas estão contidos dentro do âmbito geral da criação de Banco de Dados. Na figura 12 é apresentado o modelo lógico do PagueTudo.

3.6.3 Modelo Físico

No modelo físico de dados é feita a representação física, isto é, a adequação dos dados modelados seguindo as restrições típicas impostas pelo Sistema de Gerenciamento de Banco de Dados – SGBD. Isto é, de acordo com SILBERCHATZ (2006), mostra como o banco de dados é efetivamente implantado. Consiste em uma sequência de comandos computacionais em uma linguagem específica chamada *Structured Query Language*, ou Linguagem de Consulta Estruturada, doravante denominada como SQL, a fim de criar as tabelas, estruturas, ligações projetadas e por fim banco de dados. Neste modelo são detalhadas as estruturas do banco, índices, campos, tabelas, tipos de valores e entre várias outras informações. Para o desenvolvimento do PagueTudo foi escolhido o PHPMYADMIN como o SGBD e o banco o MySQL. Abaixo segue trecho do modelo físico de dados do PagueTudo.

```
CREATE TABLE pessoa_juridica (  
  id int PRIMARY KEY,  
  cnpjvarchar(50),  
  REFERENCES pessoa (id)  
)
```

```
CREATE TABLE debitos (  
  id int PRIMARY KEY,  
  valor double,  
  data_cadastrovarchar(50),  
  parcelamento int,  
  percentual double,
```

```
descriçãomediumtext  
)
```

```
CREATE TABLE acordos (  
id int PRIMARY KEY,  
data_efetuacaovarchar(50),  
origem varchar(50),  
valor int,  
descriçãomediumtext  
)
```

O modelo físico completo está presente no Apêndice A – MODELO FÍSICO DO BANCO DE DADOS.

3.6.3 Dicionário de Dados

Um dicionário de dados nada mais é do que um conjunto de metadados que contêm definições e representações de elementos de dados. Em outras palavras, metadados são dados sobre dados. Dentro do contexto dos Sistemas de Gerenciamento de Banco de Dados, um dicionário de dados consiste em um grupo de tabelas, feitas com intuito de apenas servirem para leitura ou consulta, ou seja, é um banco de dados propriamente dito, entretanto contém informações específicas que são: definições precisas sobre elementos de dados, perfis de usuários, papéis e privilégios, descrição de objetos, restrições de integridade, estrutura geral da base de dados, informações de verificação, alocação de espaço e índices. As tabelas a seguir são uma parte do dicionário de dados do PagueTudo.

Tabela 01 - Endereço

Coluna	Tipo	Nulo	Predefinido
Id_endereco	Int PK	Não	
estado	varchar(50)	Sim	Null
cidade	varchar(50)	Sim	Null
bairro	varchar(50)	Sim	Null
rua	varchar(200)	Sim	Null
numero	varchar(10)	Sim	Null
complemento	Varchar(200)	Sim	Null

Fonte: Próprio Autor

Tabela 02 - Pessoa

Coluna	Tipo	Nulo	Predefinido
Id_pessoa	Int PK	Não	
nome	varchar(100)	Não	Null
email	varchar(100)	Não	Null
telefone	varchar(20)	Sim	Null

Fonte: Próprio Autor

Tabela 03 - Possui

Coluna	Tipo	Nulo	Predefinido
Id_pessoa	Int FK	Não	
Id_controleAcesso	Int FK	Não	

Fonte: Próprio Autor

Tabela 04 – Controle de Acesso

Coluna	Tipo	Nulo	Predefinido
Id_controleAcesso	Int PK	Não	
nivel	varchar(30)	Não	
permissao	varchar(30)	Não	

Fonte: Próprio Autor

Tabela 05 – Pessoa Física

Coluna	Tipo	Nulo	Predefinido
Id_pessoaFisica	Int PK	Não	
cpf	varchar(50)	Não	Null
Id_pessoa	Int FK	Não	
Id_proposta	Int FK	Não	

Fonte: Próprio Autor

Tabela 06 – Pessoa Jurídica

Coluna	Tipo	Nulo	Predefinido
Id_pessoaJuridica	Int PK	Não	
cnpj	varchar(50)	Não	Null
Id_proposta	Int FK	Não	
Id_pessoa	Int FK	Não	
Id_acordos	Int FK	Não	

Fonte: Próprio Autor

3.7 MANUAL DO USUÁRIO DO PROTÓTIPO

Apesar de ser bastante intuitivo, este manual serve para ensinar como operar as funcionalidades e navegação da aplicação mobile. A seguir estão os protótipos das principais telas e a descrição de como utilizá-las.

3.7.1 Manual Sistema usuário PagueTudo (Móvel)

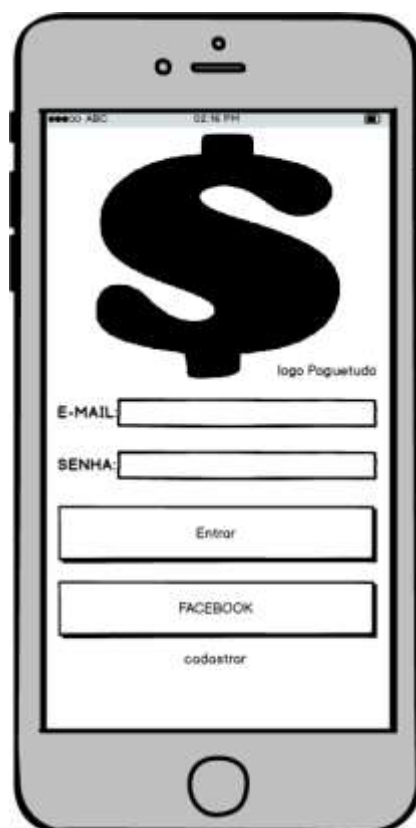
Na aplicação mobile, todo o sistema foi construído de forma simples, funcional e intuitivo, utilizando as tecnologias de inteligência artificial para facilitar ainda mais a interação do usuário com as funcionalidades oferecidas pelo sistema. Essa solução é proposta pela utilização de uma Assistente Virtual

que irá conduzir a navegabilidade da aplicação por meio de comandos de voz, interpretando e processando linguagem natural.

Ao abrir o aplicativo no celular surge a *LauncherScreen*, ou tela de apresentação, que nada mais é que uma tela contendo a logo do PagueTudo e o seu Slogan. Automaticamente o usuário é levado para a segunda tela que é a convencional tela de Acesso, ou popularmente chamada de tela de *Login*. A tela de apresentação não necessita de instrução, pois ela é apresentada e automaticamente fechada, servindo apenas como estética, logo este manual irá contemplar apenas as telas em que existe interação entre o usuário e a aplicação.

Finalmente na tela de Acesso, é apresentada mais uma vez a logomarca do PagueTudo, logo embaixo o usuário é convidado a inserir o seu E-MAIL e sua SENHA para a autenticação no sistema, caso já tenha sido cadastrado previamente, uma vez inseridos os dados é necessário apenas apertar o botão ENTRAR.

Figura 13: Tela de Acesso da aplicação mobile



Fonte: Próprio Autor

É também oferecido a facilidade de autenticação por meio do botão FACEBOOK, que serve como cadastro e autenticação por meio das credenciais de uma conta no sistema deles. Caso o usuário não tenha se cadastrado previamente, além da possibilidade de se cadastrar via botão FACEBOOK, ele também pode fazer por meio do botão CADASTRAR, logo embaixo o botão FACEBOOK. Na figura 13 é apresentado o esboço da tela.

Uma vez cadastrado e autenticado a navegação do sistema leva o usuário a terceira tela do fluxo, que é a Tela da Assistente Virtual. Nesta tela existe uma barra de navegação que é nela estão contidos um botão de MENU, representado pelas já conhecidas três barrinhas e o nome da aplicação PagueTudo.

Figura 14 : Tela da Assistente Virtual da aplicação mobile



Fonte: Próprio Autor

Apertando o botão de MENU o usuário terá acesso as suas informações cadastrais e do sistema, como meios de pagamentos, acordos e propostas. Abaixo da barra de navegação é apresentada a imagem da assistente virtual, em que servirá como tela para apresentação de algumas informações

solicitadas pelo usuário a ela. Embaixo dessa imagem temos o botão de PLAY, que deverá ser apertado para começar a interagir com a assistente por meio de comandos de voz. Vários comandos são aceitos por ela. Como solicitações de débitos, respostas de propostas, acordos celebrados e até algumas frases de humor para captar a atenção dos usuários. Na figura 14 se mostra o protótipo desta tela.

Uma vez solicitado a Assistente Virtual a apresentação da lista dos débitos, o usuário é conduzido a tela de Débitos e Propostas. Na parte superior da tela se encontra a barra de navegação, com o mesmo botão de MENU e o nome do sistema da tela anterior. Abaixo da barra de navegação o usuário encontrará uma mensagem de boas vindas personalizada com o seu nome, não existe interação com essa mensagem.

Figura 15 – Tela de Débitos e Propostas da aplicação mobile



Fonte: Próprio Autor

Logo após existe um outro texto escrito DEBITOS que mostra que a partir dali estão apresentados os dados solicitados pelo usuário. Adiante ele terá acesso a uma LISTA com todos os débitos cadastrados em seu CPF pelas empresas ou credores conveniados. Uma vez selecionado o debito, é possível clicar no botão FAZER ACORDO para receber as informações de pagamentos e parcelamentos, origem do débito, data e várias outras informações pertinentes a transparência das informações. Entretanto nesse campo de acordo, o usuário só poderá aceitar ou negar os valores já oferecidos pelas empresas, ou credores. Já o botão REALIZAR PROPOSTA o usuário será encaminhado para campo onde ele pode oferecer uma proposta diferente para os débitos que o mesmo possui em seu nome, oferecendo valores à vista, parcelamentos, formas de pagamentos ou até mesmo dúvidas sobre a origem do débito. A figura 15 mostra o protótipo da tela de Débitos e Propostas.

4 CONSIDERAÇÕES FINAIS

Como um dos intuitos principais deste projeto é a obtenção do grau de tecnólogo em Sistemas para Internet, é justo evidenciar que foram abordadas diversas competências ministradas por toda a grade da graduação, desde modelagem de negocio, até o desenvolvimento mobile e de aplicações na nuvem. Possibilitando assim conhecimento prático de um ambiente real de produção que leva a uma melhor preparação para o mercado de trabalho.

A experiência obtida com a busca para solucionar os problemas inesperados, na procura por ferramentas, *frameworks* e outras alternativas, são essenciais para expandir a visão de mundo, aguçando assim a curiosidade e a busca cada vez maior pelo aprendizado. Outro aspecto importante foi o desenvolvimento de habilidades interpessoais, como comunicação clara, trabalhos em conjuntos e postura de liderança, habilidades que juntas formam um grande aperfeiçoamento da inteligência emocional.

Logo a conclusão é que além dos ganhos acadêmicos e práticos, após exaustiva análise dos dados levantados com as pesquisas, os benefícios criados com o desenvolvimento da ferramenta batizada de PagueTudo, se mostram como uma grande oportunidade de criação de valor real para a sociedade. Diminuindo constrangimentos, preservação da natureza, economia

por parte das empresas, por parte dos usuários, gastos de transporte e muitas outras coisas ligadas diretamente ou indiretamente ao escopo econômico do sistema.

Em relação aos trabalhos futuros esta obra fornece várias opções no que diz respeito a continuidade dos esforços desde o interesse econômico, passando pelo desenvolvimento de software e que finalmente desagua nas pesquisas acadêmicas. Temas como inteligência artificial e automatização de meios de cobrança são atualmente estado da arte no ecossistema de fomento de startups, se mostrando como um campo prospero para pesquisas, investimentos e aprendizado técnico de desenvolvimento de software.

REFERÊNCIAS

- AMARAL, Suely. **Noções linguísticas: O que é linguagem?** Disponível em: <<https://educacao.uol.com.br/disciplinas/portugues/nocoas-linguisticas-o-que-e-linguagem.htm>>. Acesso em 07 de novembro de 2018.
- AURÉLIO. **Dicionário Aurélio Eletrônico** - Século XXI. [S.l.]: Editora Nova Fronteira, 1999.
- BOOCH, G; RUMBAUGH, J e JACOBSON, I: **UML: Guia do Usuário: tradução;** Fábio Freitas da Silva, Rio de Janeiro, Campus ,2000.
- CALADO, Caio. **O que é o Processamento de Linguagem Natural?** Disponível em: <<https://medium.com/botsbrasil/o-que-%C3%A9-o-processamento-de-linguagem-natural-49ece9371cff>>. Acesso em 07 de novembro de 2018.
- CHEN, Peter. **Modelagem de Dados: A abordagem Entidade Relacionamento para Projeto Lógico;** Tradução Cecília Camargo Bartalotti São Paulo, McGraw-Hill, 1990.
- CRISTINA, Ana. 2014. **Artigo Engenharia de Software 15 – UML – Diagrama de Sequências.** Disponível em: <<http://www.devmedia.com.br/artigo-engenharia-de-software-15-uml-diagrama-de-sequencias/13820>>. Acesso em 17 de maio de 2017.
- Disponível em: < <http://www.nutcache.com/blog/how-to-use-scrum-to-boost-teams-productivity/>>. Acesso em: 12 março. 2017.
- EIS, Diego. **O que é tableless.** Disponível em: <www.tableless.com.br>. Acesso em: 21/09/2016.
- ELMARSRI, Ramez; NAVATHE, Shamkant B. **Sistemas de Banco de Dados.** 6. Ed. São Paulo: Addison Wesley, 2011.
- FLAVELL, J.H. **A Psicologia do Desenvolvimento de Jean Piaget.** São Paulo, Liv. Pioneira Ed., 1975.
- GARDNER, H. **Inteligências Múltiplas: a Teoria na Prática.** Porto Alegre: Artes Médicas, 1995.
- KORTH, H.F.; SUDARSHAN, S; SILBERSCHATZ, A. **Sistema de Banco de Dados.** 5ª ed. Editora Camputs, 2006.

MELO, Ana Cristina. **Desenvolvendo Casos de Uso com UML: Do conceitual a implementação**. 1. Ed. Rio de Janeiro: Editora Brasport, 2002.

PREESMAN, Roger; MAXIM, Bruce. **Engenharia de Software: Uma abordagem profissional**. 8 ed. Porto Alegre: AMGH, 2016

RODRIGUES, Joel. **Modelo Entidade Relacionamento (MER) e Diagrama Entidade-Relacionamento (DER)**. 2016. Disponível em: <<http://www.devmedia.com.br/modelo-entidade-relacionamento-mer-e-diagrama-entidade-relacionamento-der/14332>>. Acesso em: 17 de maio de 2017.

RUP, RationalUnifiedProcess. Disponível em: <https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf>. Acesso em: 18 de maio 2017.

SANTOS, Julio Cesar. **Entrevistador: Aprenda Técnicas de Entrevistas**. 2015. Disponível em: <<http://www.rhportal.com.br/artigos-rh/entrevistador-aprenda-tcnicas-de-entrevistas/>>. Acesso em: 15 de maio de 2017

VIANA, Reinaldo. **Tecnologias de Banco de Dados e Modelagem de Dados Parte 2**. 2015. Disponível em: <<http://www.devmedia.com.br/tecnologias-de-banco-de-dados-e-modelagem-de-dados-parte-2/1871>>. Acesso em: 17 de maio de 2017

APÊNDICE A (Modelo Físico do Banco de Dados)

```
CREATETABLEpessoa (  
  idNumero,  
  nomeTexto(100),  
  emailTexto(100),  
  telefoneTexto(50),  
  idNumero,  
  loginTexto(100),  
  senhaTexto(100),
```

```
  PRIMARYKEY(id)  
)
```

```
CREATETABLEcontrole_acesso (  
  idNumeroPRIMARYKEY,  
  nivelTexto(30),  
  permissaoTexto(30)  
)
```

```
CREATETABLEendereco (  
  idNumeroPRIMARYKEY,  
  cidadeTexto(50),  
  estadoTexto(50),  
  bairroTexto(50),  
  ruaTexto(200),  
  numeroTexto(10),  
  complementoTexto(200)  
)
```

```
CREATETABLEpessoa_fisica (  
  idNumeroPRIMARYKEY,
```

```
cpfTexto(50),
idNumero,
FOREIGNKEY(id) REFERENCESpessoa (id)
)
```

```
CREATETABLEpessoa_juridica (
idNumeroPRIMARYKEY,
cnpjTexto(50),
REFERENCESpessoa (id)
)
```

```
CREATETABLEdebitos (
idNumeroPRIMARYKEY,
valorNumero,
data_cadastroTexto(50),
parcelamentoNumero,
percentualNumero,
descricaoTexto(1000)
)
```

```
CREATETABLEacordos (
idNumeroPRIMARYKEY,
data_efetuacaoTexto(50),
origemTexto(50),
valornumero,
descricaoTexto(1000)
)
```

```
CREATETABLEpropostas (
idNumeroPRIMARYKEY,
data_efetuacaoTexto(30),
valorNumero,
origemTexto(40),
descricaoTexto(1)
```


)

```
CREATETABLEpossui (  
idN?mero,  
idN?mero,  
FOREIGNKEY(id) REFERENCEScontrole_acesso (id),  
FOREIGNKEY(/*erro: ??*/) REFERENCESpessoa (id,id)  
)
```

```
CREATETABLEcontrai (  
idN?mero,  
idN?mero,  
FOREIGNKEY(id) REFERENCESpessoa_fisica (id),  
FOREIGNKEY(id) REFERENCESdebitos (id)  
)
```

```
CREATETABLErealiza (  
idn?mero,  
idN?mero,  
FOREIGNKEY(id) REFERENCESacordos (id),  
FOREIGNKEY(id) REFERENCESdebitos (id)  
)
```

```
CREATETABLEcadastra (  
idN?mero,  
idN?mero,  
FOREIGNKEY(id) REFERENCESpessoa_juridica (id)  
)
```

```
ALTERTABLEpessoaADDFOREIGNKEY(id) REFERENCESendereco  
(id)
```

```
ALTERTABLEpessoa_fisicaADDFOREIGNKEY(id)  
REFERENCESpropostas (id)
```

```
ALTER TABLE pessoa_juridica ADD FOREIGN KEY (id)
REFERENCES sacordos (id)
ALTER TABLE pessoa_juridica ADD FOREIGN KEY (id) REFERENCES propostas
(id)
```