ANDRÉ TREFILIO
BIANCA ALT
GUILHERME BERTERO

SISTEMA DE PAGAMENTOS UTILIZANDO CÓDIGO QR

São Paulo
2018

ANDRÉ TREFILIO
BIANCA ALT
GUILHERME BERTERO

SISTEMA DE PAGAMENTOS UTILIZANDO CÓDIGO QR

Trabalho apresentado à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Engenheiro de Computação.

São Paulo
2018

ANDRÉ TREFILIO
BIANCA ALT
GUILHERME BERTERO

SISTEMA DE PAGAMENTOS UTILIZANDO CÓDIGO QR

Trabalho apresentado à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Engenheiro de Computação.

Orientador:
Anarosa Alves Franco Brandão

Co-orientador:
Marcos Antonio Simplicio Jr

São Paulo
2018

Anarosa Alves Franco Brandão

Marcos Antonio Simplicio Jr

Dedicatória

O trabalho aqui apresentado é dedicado às quatro grandes forças que nos fizeram completar essa caminhada.

AGRADECIMENTOS

Agradecemos à Professora Anarosa e ao Professor Marcos que, mesmo estando muito atarefados, aceitaram nos auxiliar nessa jornada e tiveram papel preponderante na organização e realização deste projeto.

Agradecemos às nossas famílias que sempre nos deram todo suporte, amor e carinho necessários para chegarmos até aqui.

Agradecemos a toda equipe Flash Pay que nos ajudou a identificar os problemas e passou horas desenhando as soluções.

RESUMO

Com a união dos campos de engenharia de *software* e segurança da informação, é possível pensar em uma solução de pagamentos que colabore com o desenvolvimento econômico do Brasil. O projeto tem como objetivo criar um sistema de pagamentos que concorra com o modelo padrão de adquirentes e bandeiras, de modo a trazer maior integração socioeconômica e contribuir com a formalização do mercado nacional. Utilizando uma abstração de um modelo de referência de engenharia de *software*, definiu-se os passos para a construção de um produto minimamente viável utilizando o código QR como tecnologia para comunicar o pagador e o recebedor das transações. A implementação foi dividida em três módulos: servidor e banco de dados; aplicativos de pagamento e recebimento; e um painel de controle *web*. Como resultado, obteve-se um sistema seguro capaz de viabilizar transações financeiras *peer-to-peer* num ecossistema fechado, sem a necessidade de intermediários.

Palavras-Chave – Pagamento móvel, Apple Pay, Google Pay, Alipay, Wechat Pay, *serverless*, ReactJS, Segurança de pagamentos.

ABSTRACT

By joining software engineer and information security concepts, it is possible to think of a payment solution that collaborates with the brazilian economic development. The project aims to create a payment system that competes with the standard model of acquirers and credit-debit cards companies, in order to bring greater socioeconomic integration and contribute to the formalization of the national market. Using an abstraction of a software engineering reference model, the steps were defined for constructing a minimal viable product using the QR code as the technology to communicate the payer and the receiver of the transactions. The implementation was divided into three modules: server and database; payment and receipt applications; and a web dashboard. As a result, we obtained a secure system capable of enabling peer-to-peer financial transactions in a closed ecosystem, without the need for intermediaries.

Keywords – Mobile Payment, Apple Pay, Google Pay, Alipay, Wechat Pay, serverless, ReactJS, Payment Security.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS

- QR: *Quick Response*
- PIB: Produto Interno Bruto
- NFC: *Near Field Communiction*
- API: *Application Programming Interface*
- RSA: Rivest, Shamir, Adleman
- XOR: *Exclusive Or*
- ECDSA: *Elliptic Curve Digital Signature Algorithm*
- EdDSA: *Edwards-curve Digital Signature Algorithm*
- NIST: *National Institute of Standards and Technology*
- NSA: *National Security Agency*
- IEEE: *Institute of Electrical and Electronics Engineers*
- ANSI: *American National Standards Institute*
- RM-ODP:*Reference Model of Open Distributed Processing*
- HTTP: *Hypertext Transfer Protocol*
- HTTPS: *Hypertext Transfer Protocol Secure*
- TLS: *Transport Layer Security*
- MST: *Magnetic Secure Transmission*
- PIN: *Personal Identification Number*
- P2P: conexão *peer-to-peer*
- AWS: Amazon *Web Services*
- ACID: Atomicidade, Consistência, Isolamento e Durabilidade
- JSON: Javacript *Object Notation*

SUMÁRIO

PARTE I

INTRODUÇÃO

1 INTRODUÇÃO

A economia brasileira pós crise econômica ainda se recupera de forma lenta, algo que pode ser visto nos números expressivos de informalidade. Além de prejudicial para o país como um todo, a perda de direitos dos trabalhadores nessa situação é crítica do ponto de vista social, uma vez que as marginaliza.

Olhando mais especificamente para o setor de pagamentos, nota-se uma concentração excessiva de poder nas mãos de adquirentes como Cielo, Rede e Getnet - que nada mais são do que diferentes facetas dos grande bancos no Brasil: Banco do Brasil, Bradesco, Itau e Santander.

Além disso, o mercado de benefícios corporativos se mostrou uma boa porta de entrada para o plano de negócios do projeto desenvolvido, já que é dominado por grandes empresas que cobram taxas abusivas dos lojistas que aceitam os vale-benefícios em seus estabelecimentos.

Combinando tecnologias já usadas para este fim - como código QR e NFC - é possível desenvolver um intermediário financeiro seguro que ajude a mudar esse paradigma atual.

1.1 Objetivo

O objetivo deste trabalho é aplicar os conhecimentos de segurança da informação e engenharia de software para desenvolver um sistema de pagamentos que auxilie a formalização da economia brasileira e uma inclusão socioeconômica.

1.2 Motivação

A motivação para o desenvolvimento do projeto foi dividida em duas partes: profissional, visto que a resolução de um problema de negócios se aproxima do mercado; e acadêmica, uma vez que os conceitos aprendidos na universidade são base para o desen-

rolar do trabalho.

1.2.1 Motivação profissional

A motivação profissional para a realização deste projeto veio do contato com questões de segurança em diversas áreas do mercado de trabalho. Apesar de essenciais em qualquer área de desenvolvimento, aspectos práticos desta área de estudo são pouco abordados no curso de Engenharia de Computação. Assim, objetivou-se aplicar esses conhecimentos na resolução de problemas reais.

Unindo as ferramentas de engenharia de software com os conhecimentos de segurança da informação, o projeto busca projetar e implementar o sistema de uma startup de pagamentos.

1.2.2 Motivação acadêmica

Este trabalho engloba o aprendizado e a aplicação de técnicas de engenharia de software para modelar um sistema de pagamentos seguro. Para tanto, envolveu-se as seguintes visões de referência: negócios, engenharia, segurança e tecnologia.

1.3 Organização

A primeira parte do trabalho é a introdução, em que são apresentados o contexto, motivação, objetivos e justificativas da realização do projeto. Na segunda parte, há o desenvolvimento, constituído de quatro capítulos:

- **Fundamentação teórica:** apresenta conceitos importantes empregados ao longo deste documento, além de contextualizá-los um pouco mais.
- **Metodologia:** apresenta a sequência lógica utilizada para estruturar o projeto e modularizar o processo de desenvolvimento.
- **Especificação do projeto:** trata de especificar todos os aspectos do projeto, passando por conceitos de negócios e de engenharia.
- **Implementação:** contempla a prototipação do projeto, selecionando tecnologias que cumpram da melhor maneira os requisitos desenhados na seção de especificação do projeto.

- **Considerações finais:** apresenta o quanto os resultados obtidos se aproximam dos objetivos iniciais e quais seriam possíveis discussões para trabalhos futuros.

PARTE II

DESENVOLVIMENTO

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta conceitos importantes empregados ao longo deste documento, além de contextualizá-lo um pouco mais.

2.1 Economia informal e o setor de pagamentos no Brasil

O grande mercado informal do país e o impacto que isso causa na economia tendo movimentado aproximadamente 1 trilhão de reais em 2015 (ou 16,2% do PIB daquele ano) são fatos relevantes de se observar quando o assunto é pagamento [1]. Além disso, a alta quantia subtraída de lojistas logo no momento da compra por efeito de intermediários apresenta uma oportunidade para, no mínimo, pesquisar mais a fundo o setor. A Figura 1 ilustra tanto a representatividade da economia informal frente ao PIB, quanto o valor absoluto movimentado ano a ano.

A participação de adquirentes, bandeiras de cartão e os próprios bancos emissores dos cartões cobram valores que chegam até 12% e estão em média por volta de 5% [2]. A Figura 2 ajuda a entender o mecanismo de pagamentos via cartão no Brasil, seja ele de crédito, débito, benefícios (como vale alimentação, refeição e transporte), entre outros.

Como a presença de intermediários tende a diminuir a receita dos vendedores, observa-se uma oportunidade para concorrer com o modelo atual de adquirente-bandeira por somente um interlocutor que conecte pagador e recebedor. Assim, abrem-se portas para algumas tecnologias que podem comunicar as duas pontas, como o código QR e o NFC.

2.2 Código QR

O código QR - ou código de Resposta Rápida (*Quick Response Code*) - foi inventado em meados de 2002 em fábricas japonesas para identificar peças de automóveis durante

Figura 1: A Economia Informal no Brasil

	% PIB	Em Milhões de Reais	
		Reais Correntes	Reais a Preços de 2015
2003	21,0%	361.116	870.089
2004	20,9%	409.324	915.291
2005	20,5%	444.139	924.443
2006	20,2%	485.836	947.074
2007	19,4%	527.910	966.838
2008	18,7%	581.011	978.217
2009	18,5%	615.500	965.660
2010	17,6%	685.367	991.737
2011	16,9%	738.451	986.484
2012	16,5%	794.587	984.531
2013	16,2%	862.675	995.368
2014	16,1%	915.909	988.909
2015	16,2%	956.898	956.898

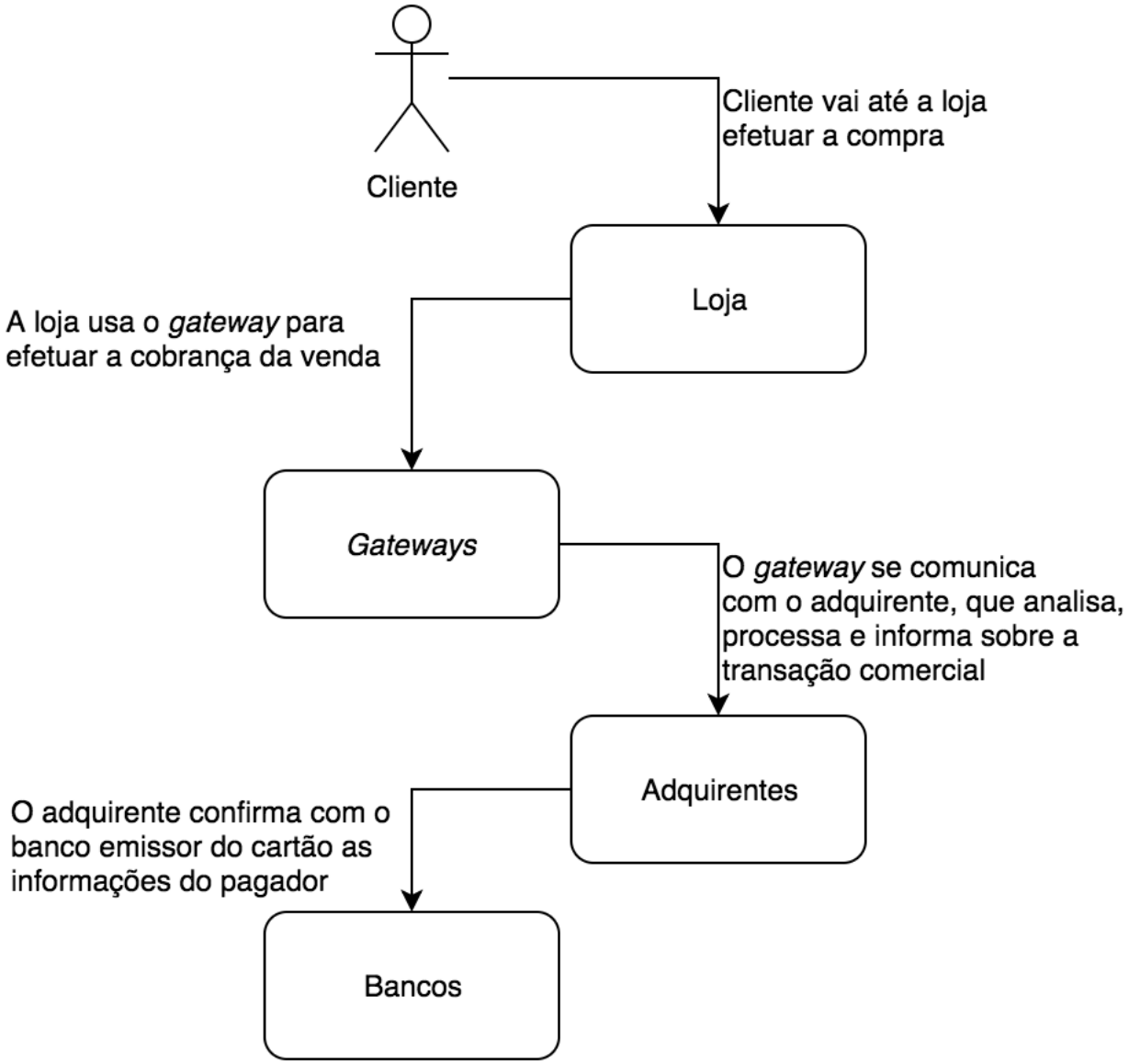
Fonte: Instituto Brasileiro de Ética Concorrencial (2016)

o processo de manufatura [3]. O que ninguém imaginava era que, praticamente dez anos depois, ele seria peça chave de economias tão importantes na Ásia e no mundo.

Desde aproximadamente 2012, países como China e Índia tem aplicativos com leitores de código QR como instrumentos imprescindíveis do dia-a-dia urbano de seus habitantes, sendo usado tanto em aluguéis de bicicletas, quanto em pagamentos de vendedores ambulantes. Muitos chineses, inclusive, dizem deixar suas carteiras literalmente em casa [4]. Vale notar que, dos 731 milhões de habitantes com acesso à internet, 95% utiliza o smartphone para isso; e metade desses realizou pelo menos um pagamento online em 2016 [5].

O sucesso na China se deve muito ao papel fundamental exercido por algumas de suas maiores empresas como a Alibaba e a Tencent. A primeira desenvolveu o app Alipay, que, hoje, funciona como um portal de serviços, unindo comodidade e segurança em apenas um lugar. O usuário tem uma carteira virtual e pode pedir desde um Uber até um delivery de sucos naturais dentro de um mesmo aplicativo, e todo processo de pagamento é realizado pela Alibaba. Essa centralização cortou uma série de custos dos estabelecimentos, visto que não havia a necessidade de intermediários nas transações [6]. Já a segunda, baseando-se no sucesso da primeira, implementou a funcionalidade de carteira virtual dentro do WeChat, maior aplicativo de mensagens instantâneas do país. Mesmo tendo ambientes diferentes, ambas empresas oferecem um serviço baseado nos códigos QR e na carteira

Figura 2: Modelo de execução de pagamento via cartão de débito no Brasil.



virtual.

Enquanto isso, na Índia, o governo teve papel muito importante na disseminação dos aplicativos de leitura de códigos QR. O grande peso da economia informal no país levou o Estado a pensar em soluções alternativas à fiscalização policial, surgindo, em 2013, o grande impulso ao uso de aplicativos de pagamento com leitores de código QR junto com a retirada de praticamente 80% das cédulas em circulação. O sucesso foi tremendo e o volume de transações digitais subiu 19% anualmente até 2016 [7].

2.3 NFC

Comunicação por campo de proximidade - ou NFC, do inglês *Near Field Communication* - é a tecnologia sem fio que fornece comunicação entre dois celulares que contém *tags* NFC, usando ondas de rádio de curto alcance. Para tal, é usada indução de campo magnético. Ambos os dispositivos podem se comunicar entre si requerendo um curto alcance de aproximadamente quatro centímetros. Pode-se fazer pagamentos usando celulares habilitados para NFC aproximando-os de outro celular ou uma máquina de cartão que também tenha a tecnologia [8].

2.4 QR Code e NFC como meios de pagamento

A diversificação das soluções de pagamento foi acompanhada pela difusão das tecnologias do QR Code e do NFC. O *software*, *hardware* e as APIs que possibilitam a leitura de QR Codes e *tags* NFC com um telefone celular dependem de seu sistema operacional e plataforma. Atualmente, existem vários sistemas operacionais e plataformas móveis compatíveis com ambas as tecnologias, diminuindo a resistência em adotá-las.

Vale ressaltar que já existem grandes empresas no mercado que oferecem o modelo de negócio de carteira digital utilizando as tecnologias em questão. Entre elas a Apple - com o *Apple Pay*, o Google - com o *Google Pay* - e a Samsung - com o *Samsung Pay* - [9]. Ao longo do documento, seus aspectos, fluxos de informação e processos serão relacionados com o projeto desenvolvido pelo grupo.

2.5 Segurança e conceitos de criptografia

Como o modelo de pagamento utilizando tecnologias como código QR e NFC exige uma conexão *peer-to-peer*, o projeto exige uma atenção extra com segurança.

2.5.1 Assinatura Digital

Começando com um conceito de garantia de autenticidade, numa assinatura digital, dada uma mensagem M de tamanho variável, utiliza-se uma função não invertível do tipo *Message digests* para se produzir um código de tamanho fixo associado univocamente à mensagem. Encripta-se o resultado da função anterior com a chave privada de quem está assinando a mensagem e o resultado é o que chamamos de assinatura digital.

A assinatura digital passa a fazer parte da mensagem, dizendo-se então que ela esta assinada pelo proprietário da chave privada utilizada no processo da assinatura. Para se verificar uma assinatura, utiliza-se a chave pública disponibilizada pelo assinante. Para tanto, decripta-se a assinatura com a chave pública, gera-se o *Message Digest* utilizando a mesma função hash usada inicialmente à partir dos dados da mensagem e compara-se os dois resultados anteriores: se forem iguais, está garantido que a mensagem de fato veio daquela pessoa e que ela não foi alterada.

2.5.2 Encriptação Híbrida

Quando se trata de criptografar grandes quantidades de dados com criptografia assimétrica tem-se um processo custoso e demorado. Para se garantir a eficiência e a segurança, uma solução é a criptografia híbrida [10]. Para tanto, utiliza-se dois algoritmos, um de criptografia simétrica e um assimétrica:

- Gera-se a chave simétrica e encripta-se os dados a serem enviados;
- Encripta-se então a chave simétrica com a chave pública de quem deve receber a mensagem;
- O destinatário utiliza sua chave privada para desencriptar a chave simétrica e utiliza a chave simétrica para desencriptar os dados recebidos.

2.5.3 Algoritmo RSA

RSA (Rivest, Shamir, Adleman) é um algoritmo de criptografia assimétrica. São geradas duas chaves, uma pública e uma privada. A encriptação é feita com a chave pública enquanto a decriptação é feita com a chave privada, portanto apenas o portador dessa chave pode acessar o conteúdo original [11].

Geração do par de chaves é feita da seguinte forma:

- Selecione p e q ambos números primos;
- Calcule $n = p * q$;
- Selecione um inteiro d, tal que: $mdc(\phi(n), d) = 1; 1 < d < \phi(n)$;
- Calcule e, tal que $e = mod(d^{-1}, \phi(n))$;

- Chave Pública $KU = \{e, n\}$;
- Chave Privada $KR = \{d, n\}$;
- Selecione dois números primos: $p = 7$ e $q = 17$;
- Calcule $n = p * q$, $n = 7 * 17 = 119$;
- Calcule $\phi(n) = \phi(119) = (p-1)(q-1) = 96$;
- Selecione e, tal que seja primo relativo a 96 e menor que 96. Neste caso escolhemos $e = 5$;
- Determine d tal que, $d * e = mod(1, 96)$ e $d < 96$. O valor calculado é $d = 77$;
- Chave Pública = {5,119} e Chave Privada = {77,119}.

$\phi(n)$: número de inteiros positivos menores do que n e primos relativos a n.
mod: operação módulo, que encontra o resto da divisão do primeiro número pelo segundo.
mdc: máximo divisor comum.

Encriptação e decriptação são feitas da seguinte maneira:

- Seja o texto limpo igual a $M < n$ e o texto cifrado igual a C:
  - Criptografia:
    - $C = mod(M^e, n)$
  - Decriptografia:
    - $M = mod(C^d, n)$

2.5.4 Algoritmo AES

Advanced Encryption Standard é um dos algoritmos de chave simétrica mais poderosos atualmente. Foi adotado como padrão pelo governo americano em 2002. Ele utiliza uma chave inicial, de 128, 192 ou 256 bits e realiza a encriptação em blocos (do mesmo tamanho da chave). A encriptação dos blocos consiste de 10 rodadas para chaves de 128 bits, 12 rodadas para chaves de 192 e 14 para chaves de 256 [12]. Cada rodada consiste de 4 etapas: *SubBytes*, *ShiftRows*, *MixColumns* e *AddRoundKey*.

- Primeiro deriva-se as chaves a serem usadas em cada rodada à partir da chave inicial

- Chave inicial de 128 bits em hexadecimal: 54 68 61 74 73 20 6D 79 20 4B 75 6E 67 20 46 75
- $w[0] = (54, 68, 61, 74), w[1] = (73, 20, 6D, 79), w[2] = (20, 4B, 75, 6E), w[3] = (67, 20, 46, 75)$
- Deslocamento circular de $w[3] : (20, 46, 75, 67)$
- Substituição dos bytes usando os bytes correspondentes da Rijndael *S-box* [13]: $(B7, 5A, 9D, 85)$
- Adiciona-se (XOR bit a bit) a constante de rodada $(01, 00, 00, 00)$ (estamos gerando a chave da primeira rodada), que resulta em $g(w[3]) = (B6, 5A, 9D, 85)$
- $w[4] = w[0] \oplus g(w[3]) = (54, 68, 61, 74) \oplus (B6, 5A, 9D, 85) = (E2, 32, FC, F1)$
- $w[5] = w[4] \oplus w[1] = (91, 12, 91, 88)$
- $w[6] = w[5] \oplus w[2] = (B1, 59, E4, E6)$
- $w[7] = w[6] \oplus w[3] = (D6, 79, A2, 93)$

Com $w[4]$, $w[5]$, $w[6]$ e $w[7]$ temos a chave da primeira rodada E2 32 FC F1 91 12 91 88 B1 59 E4 E6 D6 79 A2 93.

Com a chave da primeira rodada, repetimos os passos anteriores, usando a constante de rodada (02, 00, 00, 00) para gerar a chave da segunda rodada, e assim por diante.

Tendo todas as chaves de rodada, inicia-se a encriptação de fato de cada bloco. Na rodada 0 apenas fazemos a etapa *AddRoundKey*.

- Pegamos o bloco de dados e a chave 0;

Figura 3: Bloco de dados e chave 0.

$$\begin{pmatrix} 54 & 4F & 4E & 20 \\ 77 & 6E & 69 & 54 \\ 6F & 65 & 6E & 77 \\ 20 & 20 & 65 & 6F \end{pmatrix} \qquad \begin{pmatrix} 54 & 73 & 20 & 67 \\ 68 & 20 & 4B & 20 \\ 61 & 6D & 75 & 46 \\ 74 & 79 & 6E & 75 \end{pmatrix}$$

- Fazemos um XOR byte a byte;

Figura 4: Resultado de XOR entre bloco de dados e chave 0.

$$\begin{pmatrix} 00 & 3C & 6E & 47 \\ 1F & 4E & 22 & 74 \\ 0E & 08 & 1B & 31 \\ 54 & 59 & 0B & 1A \end{pmatrix}$$

Com esse novo bloco, iniciamos as etapas citadas anteriormente:

- *SubBytes*: substituimos cada byte pelo seu correspondente na tabela Rijndael S-box;

Figura 5: Resultado da etapa *SubBytes*.

$$\begin{pmatrix} 63 & EB & 9F & A0 \\ C0 & 2F & 93 & 92 \\ AB & 30 & AF & C7 \\ 20 & CB & 2B & A2 \end{pmatrix}$$

- *ShiftRows*: deslocamos circularmente cada linha de 0, 1, 2 e 3 posições;

Figura 6: Resultado da etapa *ShiftRows*.

$$\begin{pmatrix} 63 & EB & 9F & A0 \\ 2F & 93 & 92 & C0 \\ AF & C7 & AB & 30 \\ A2 & 20 & CB & 2B \end{pmatrix}$$

- *MixColumns*: multiplicamos o bloco atual por uma matriz constante onde cada coeficiente é um elemento no campo finito de Galois $GF(2^8)$.
  - Por exemplo, $(02.63) \oplus (03.2F) \oplus (01.AF) \oplus (01.A2)$ resulta em BA:
    - $02.63 = 00000010.01100011 = 11000110$ (lembrando que multiplicar por 2 é um *left shift*);

* $03.2F = (02.2F) \oplus 2F = (00000010.00101111) \oplus 00101111 = 01110001$;
* $01.AF = AF = 10101111$;
* $01.A2 = A2 = 10100010$;
* Somamos os quatro resultando em BA.

Figura 7: A etapa *MixColumns*.

$$\begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \begin{pmatrix} 63 & EB & 9F & A0 \\ 2F & 93 & 92 & C0 \\ AF & C7 & AB & 30 \\ A2 & 20 & CB & 2B \end{pmatrix} = \begin{pmatrix} BA & 84 & E8 & 1B \\ 75 & A4 & 8D & 40 \\ F4 & 8D & 06 & 7D \\ 7A & 32 & 0E & 5D \end{pmatrix}$$

- *AddRoundKey*: Adicionamos a chave de rodada;

Figura 8: Resultado da etapa *AddRoundKey*.

$$\begin{pmatrix} BA & 84 & E8 & 1B \\ 75 & A4 & 8D & 40 \\ F4 & 8D & 06 & 7D \\ 7A & 32 & 0E & 5D \end{pmatrix} \qquad \begin{pmatrix} E2 & 91 & B1 & D6 \\ 32 & 12 & 59 & 79 \\ FC & 91 & E4 & A2 \\ F1 & 88 & E6 & 93 \end{pmatrix}$$

- É realizado um XOR byte a byte, como mostrado na Figura 9.

Figura 9: Resultado da etapa de XOR byte a byte.

$$\begin{pmatrix} 58 & 15 & 59 & CD \\ 47 & B6 & D4 & 39 \\ 08 & 1C & E2 & DF \\ 8B & BA & E8 & CE \end{pmatrix}$$

Tomamos esse bloco como entrada para a próxima rodada e repetimos os mesmos passos, com exceção da última rodada onde pulamos a etapa *MixColumns*.

2.5.5 Algoritmo EdDSA

O EdDSA é um algoritmo assimétrico para assinaturas digitais baseado em curvas elípticas, assim como o ECDSA, porém com algumas diferenças.

O ECDSA foi aceito em 1999 como padrão ANSI e em 2000 como padrão IEEE e NIST. Sua eficácia já é amplamente comprovada e sua maior vantagem em relação a outros algoritmos assimétricos é que ele exige chaves de tamanho reduzido sem perder em segurança. Por exemplo, para um nível de segurança de 128 bits, usando RSA recomenda-se um tamanho de chaves de 3072 bits [14], ao passo que com algoritmos de curvas elípticas, recomenda-se chaves de 256 bits apenas. Um dos problemas com o ECDSA é que ele exige a geração de um número aleatório, e conseguir isso de forma segura pode ser desafiador em alguns casos. Há relatos de que a NSA inseriu um *backdoor* no padrão SP 800-90 publicado pela NIST com recomendações para a geração de números aleatórios [15], o que gerou desconfianças também em relação às curvas recomendadas pela NIST para a implementação do ECDSA.

O EdDSA possui algumas vantagens em relação ao ECDSA, dentre elas:

- EdDSA provê uma performance melhor em uma variedade de plataformas;
- Dispensa o uso de um gerador de números aleatórios.
- Maior resistência a ataques de canal lateral.

Por essas razões, optamos pelo uso do EdDSA (em específico, Ed25519 [16], implementação do EdDSA que utiliza a curva Curve25519 [17]), para assinaturas digitais.

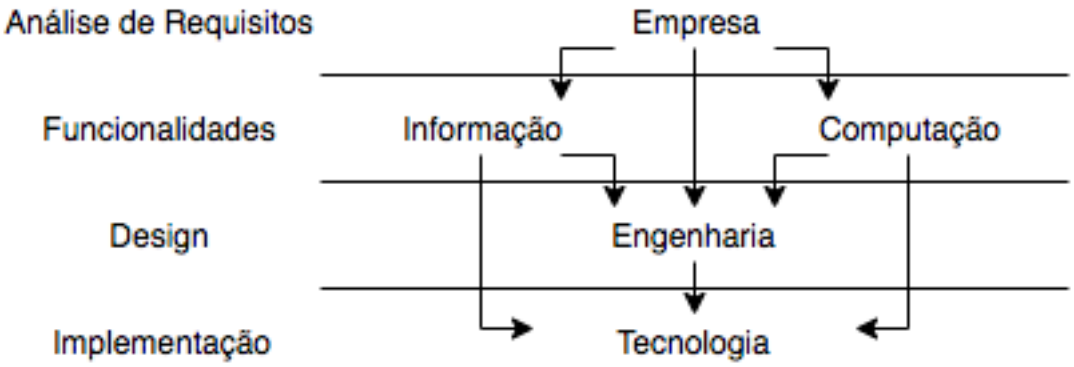
2.6 Estrutura arquitetural: RM-ODP

Dentre a enorme lista de possibilidades existentes no mundo de engenharia de software, algumas escolhas foram feitas de forma a organizar da melhor maneira possível os problemas de negócio a serem resolvidos e as tecnologias de segurança exigidas por eles.

Modelo de Referência para Processamento Distribuído Aberto - do inglês *Reference Model of Open Distributed Processing*, o RM-ODP é um modelo de referência para sistemas distribuídos abertos, desenvolvido a partir do trabalho conjunto da ISO (Organização Internacional de Normalização - do inglês *International Organization for Standardization*)

e da ITU (União Internacional de Telecomunicação - do inglês *International Telecommunication Union*) para criar uma arquitetura que suporte distribuição, interoperabilidade e portabilidade [18]. Estes Modelo utiliza o conceito de visões (economia, informação, informática, engenharia e tecnologia) para analisar sistemas de diferentes ângulos relevantes. A Figura 10 ilustra a relação dos pontos de vista com os processos de engenharia de software (análise de requisitos, especificação funcional, design e implementação).

Figura 10: Relação entre as visões do modelo RM-ODP e os processos de engenharia de software.



2.7 Design Thinking

Design Thinking é uma ferramenta para administrar ideias e incentivar o processo criativo em torno do ecossistema trabalhado. A metodologia abrange um espectro completo de possibilidades com um design centrado no usuário.

Tem como objetivo criar soluções que atendam melhor as necessidades e desejos do consumidor através da observação e estudo de suas preferências e comportamentos.

Vale ressaltar que nem sempre as ideias envolvem tecnologias inovadoras, mas sim projetos que sejam comercialmente viáveis e que gerem valor para o mercado e para o consumidor [19].

2.8 Tecnologias

Depois de estruturar a arquitetura do projeto, foi necessário selecionar as tecnologias que melhor se encaixariam, de forma a trazer melhores resultados no protótipo.

2.8.1 AWS

De acordo com a própria Amazon, proprietária da Amazon Web Services, ou AWS [20]:

> A Amazon Web Services (AWS) consiste em produtos e serviços em nuvem seguros que oferecem poder computacional, armazenamento de banco de dados, entrega de conteúdo e outras funcionalidades para ajudar as empresas a escalar e crescer.

Dentro da AWS, dois serviços foram utilizados durante a implementação do projeto: API Gateway e Lambda.

O primeiro, de acordo com a Amazon[21]:

> O Amazon API Gateway administra todas as tarefas envolvidas no recebimento e processamento de até centenas de milhares de chamadas de API simultâneas, inclusive gerenciamento de tráfego, controle de autorização e acesso, monitoramento, além de gerenciamento de versões de API. O Amazon API Gateway não tem taxas mínimas ou custos antecipados. Você paga apenas pelas chamadas de API recebidas e pela quantidade transferida de dados para fora.

Já o segundo, ainda de acordo com a Amazon [22]:

> Com o Lambda, você pode executar o código para praticamente qualquer tipo de aplicativo ou serviço de back-end, tudo sem precisar de administração. Basta carregar o código e o Lambda se encarrega de todos os it ens necessários para executar e alterar a escala do código com alta disponibilidade. Você pode configurar o seu código para que ele seja acionado automaticamente por meio de outros serviços da AWS ou chamá-lo diretamente usando qualquer aplicativo móvel ou da web.

2.8.2 MongoDB

O MongoDB é um banco de dados de código aberto e não estruturado que armazena dados em documentos flexíveis semelhantes a JSON, o que significa que os campos podem variar de documento para documento e a estrutura de dados pode ser alterada ao longo do tempo. Dentre suas principais vantagens estão a indexação e a agregação em tempo real, que fornecem maneiras poderosas de acessar e analisar os dados.

O MongoDB é um banco de dados distribuído desde seu núcleo, de modo que a alta disponibilidade, o dimensionamento horizontal e a distribuição geográfica são nativamente integrados e fáceis de usar [23].

2.8.3 Javascript

É uma linguagem de *script* interpretada, utilizada inicialmente nos navegadores para melhorar a experiência de usuário e tornar as páginas mais dinâmicas. Atualmente, é uma linguagem utilizada em diversos outros momentos, desde servidores até aplicativos. Sua popularidade é bastante alta na comunidade de desenvolvedores e possui um suporte muito amplo [24].

2.8.4 NodeJS

É um ambiente de execução de códigos Javscript. Seu ponto alto é o fato de operar em uma única *thread*, e com chamadas de entrada e saída não-bloqueadoras. Utilizando um laço de eventos, o NodeJS passa as chamadas para o sistema e, quando as operações dessas chamadas forem finalizadas, o kernel devolve a resposta ao processo. Isso torna a aplicação mais eficiente mas impossibilita uma escalabilidade vertical. Para isso o desenvolvedor pode aumentar o número de *threads* utilizado pelo arcabouço [25].

2.8.5 ReactJS e React-Native-JS

ReactJS é uma biblioteca de Javascript construído pela equipe de desenvolvedores do Facebook para melhorar interfaces de *front-end* para usuários. Ele permite que desenvolvedores criem grandes aplicativos da *web* que podem sofrer alterações de informações sem recarregar a página. O principal objetivo do ReactJS é ser rápido, escalável e simples [26].

Já o React Native é um *framework* que estrutura uma hierarquia de componentes de interface do usuário para criar o código JavaScript. Ele tem um conjunto de componentes para plataformas iOS e Android para criar um aplicativo móvel com aparência e comportamento nativos. Ele te permite codificar apenas uma vez e os aplicativos resultantes estão disponíveis para as plataformas iOS e Android, o que economiza tempo de planejamento, desenvolvimento e testes. Possui grande popularidade e também é apoiado pelo Facebook, que teve papel preponderante na sua criação [27] e [24].

3 METODOLOGIA

A metodologia empregada no trabalho consiste em uma sequência lógica de etapas: revisão da literatura e avaliação de trabalhos relacionados, estudo aprofundado de algoritmos de criptografia, estruturação do sistema e implementação e testes do projeto.

3.1 Revisão da literatura

Inicialmente, foi feito um levantamento bibliográfico de artigos e sites relevantes para o tema deste trabalho. Os principais tópicos pesquisados foram: como é a aceitação de usuários em relação a sistemas de pagamentos móveis; tecnologias utilizadas para realizar transações financeiras entre celulares; e métodos para garantir a segurança de um sistema de pagamentos.

3.2 Estudo de algoritmos de encriptação

Como resultado do estudado através da literatura, percebeu-se a necessidade de um aprofundamento em algumas tecnologias de segurança da informação em três pontos imprescindíveis: proteção de senhas; proteção de informação que é transmitida via internet; e assinatura de informações de pagamento transmitidas entre celulares. Desta maneira, esta seção detalha estas etapas e seleciona os algoritmos a serem utilizados.

3.3 Desenvolvimento do projeto do sistema de pagamentos

Outro resultado colhido da literatura foi o modelo de sistema de pagamentos que melhor seria aceito pelos usuários: o pagamento via código QR; somando-se com o aprofundamento de segurança da seção anterior e com o auxílio de ferramentas de engenharia de software, foi possível delinear a estrutura do projeto, com seus requisitos funcionais e

não funcionais.

3.4 Implementação do sistema de pagamentos

De forma a apresentar um resultado tangível, criou-se um PDC (Prova de Conceito) como forma de validar as escolhas de segurança e a aceitabilidade de usuários. Por fim, realizou-se testes fim-a-fim para garantir o correto funcionamento do sistema como um todo.

4 TRABALHOS RELACIONADOS

Diversos trabalhos sobre sistemas de pagamento móvel concentraram-se, principalmente, na descoberta de cadeias de valor e qualidades que podem promover a aceitação desse modelo. Kindberg et al. examinou as preocupações de confiança e segurança dos usuários sobre pagamentos móveis com uma carteira eletrônica [28]. Facilidade de uso, conveniência ou questões sociais foram relatados como tão importantes quanto questões de confiança e segurança. Kristoffersen et al. analisou a atitude de usuários noruegueses em relação aos pagamentos móveis em geral[29].

Ainda sobre esse tema, buscou-se na literatura informações sobre os modelos de pagamento digital utilizando somente o celular ou a combinação celular e terminal de adquirência para entender onde que a solução tratada no projeto melhor se encaixaria. Desta maneira, com as informações encontradas, foi possível delinear a Tabela 1 com dados diferencias de cada uma das soluções [30] e [31]:

Solução	Tecnologia P2P	Aceitação	Autenticação
Alipay	Código QR	Comerciantes cadastrados	Impressão digital, PIN, senha
Apple Pay	NFC	Terminais de adquirência com NFC	Impressão digital, FaceID
Google Pay	NFC	Terminais de adquirência com NFC	Impressão digital, PIN, senha
Samsung Pay	NFC e MST	Terminais de adquirência	Impressão digital, PIN, reconhecimento de íris
WeChat Pay	Código QR	Comerciantes cadastrados	Impressão digital, PIN, senha

Tabela 1: Detalhamento dos principais sistemas de pagamento móvel atualmente.

Os três modelos que utilizam a tecnologia de NFC, por não possuírem moeda digital - ou seja, não é possível depositar fundos na carteira, somente cadastrar cartões de débito ou crédito -, dependem de terminais de adquirência para que as transações possam ser

realizadas, sejam elas físicas ou na *internet*. Os aparelhos celulares da Samsung possuem uma tecnologia P2P extraque amplia a aceitação do Samsung Pay para praticamente todos os terminais de adquirência, mesmo aqueles sem NFC: o MST - do inglês *Magnetic Secure Transmission* [32].

Como mencionado no capítulo de fundamentação teórica, os sistemas Alipay e WeChat Pay são implementações de pagamentos móveis que foram bem sucedidos na Ásia, principalmente na China. Dentre os motivos para isso, destaca-se a acessibilidade do modelo, que requer, simplesmente, aparelhos celulares com câmeras simples [30]. Diferentemente dos outros três sistemas presentes na Tabela 1, os chineses utilizam um modelo com moeda digital, ou seja, é possível depositar fundos na sua carteira, além de cadastrar cartões de débito ou crédito.

Quando se trata de autenticação para a realização de um pagamento móvel, percebe-se que o *hardware* do aparelho tem papel primordial no nível de segurança que os modelos possuem. Ou seja, aparelhos mais novos e com mais funcionalidades - costumeiramente, mais caros - proporcionam transações com maior credibilidade.

Trabalhos anteriores também cobriam pagamentos baseados em NFC. Alguns fatores como usabilidade, custo e confiança foram estudados por A. Zmijewska para analisar a adequação das tecnologias sem fio para criar sistemas de pagamento móvel que grande parte dos usuário aceitaria [33]. NFC é tido como a tecnologia mais confiável e aceitável por parte dos usuários quando trata-se do assunto de pagamentos móveis. Há alguns anos, J. Ondrus e Y. Pigneur apresentaram uma revisão da tecnologia para ser usada em pagamentos no futuro e concluíram que o NFC tornar-se-ia um ponto de inflexão no meio financeiro [34]. M. Massoth e T. Bingel compararam o desempenho de vários serviços de pagamento tradicionais para dispositivos móveis como Resposta Interativa de Voz (*Interactive Voice Response* - IVR), Serviço de Mensagem Curta (*Short Message Service* - SMS) e protocolo de aplicação sem fio (*Wireless Application Protocol* - WAP) [9]. A aplicação baseada em NFC apresentou os melhores resultados de desempenho.

Mabel Vazquez-Briseno et al. desenvolveram um projeto analisando a interação entre os mundos físico e virtual e comparando as tecnologias utilizadas para tal fim [35]. Especificamente sobre NFC e código QR, definiram que o primeiro é mais caro e, portanto, com uma disponibilidade menor, porém mais seguro, uma vez que exige uma enorme proximidade para que a conexão seja feita.

Tratando-se de transações financeiras de modo geral, já há preocupação com questões de segurança; quando o assunto é pagamento utilizando dispositivos móveis, essa questão

fica ainda mais saliente. Analisando os algoritmos de encriptação mais famosos e renomados do atual cenário global, destaca-se: AES, RSA, EDDSA e Lyra2.

V. Mahalle e A. Shahade trabalharam em como manter a segurança de dados transacionados em serviços de nuvem utilizando criptografia híbrida com os algoritmos RSA e AES, além de como armazenar e gerenciar corretamente as chaves públicas e privadas [36].

S. Jaju e S. Chowhan estudaram o algoritmo de RSA focando no seu uso como assinatura digital [11]. Como forma de entender a eficiência dele, o compararam com, por exemplo, o EDDSA, demonstrando que a verificação é mais lenta no segundo, mesmo ele apresentando chaves com tamanhos menores em bytes.

M. Simplício et al. trabalharam em cima do algoritmo Lyra e produziram o novo Lyra2, que permite o desenvolvedor aplicar ajustes finos em usos de memória e processamento de acordo com o nível de segurança desejados [37]. O algoritmo trabalha de maneira significativamente sequencial, impedindo que ataques de força bruta paralelizem suas ações.

Mais especificamente sobre o mundo financeiro, M. Hossain e R. Islam foram mais além e publicaram um projeto propondo um modelo que classifica os dados de usuários e empresas antes de armazená-los em sistemas de nuvem distribuídos [38]. Segundo o artigo, a classificação, somada a três camadas de segurança, melhora, de fato, a segurança dos sistemas. Mesmo o sistema possuindo um único banco de dados, o acesso para cada um dos três tipos (público, moderadamente privado e privado) é totalmente separado e exige autenticações distintas.

5 ESPECIFICAÇÃO DO PROJETO

Este capítulo trata de especificar todos os aspectos do projeto. Para se ter embasamento sobre a estrutura do trabalho, estudou-se o modelo de referência RM-ODP citado no capítulo de Fundamentação Teórica.

Como o RM-ODP compõe uma estrutura bastante complexa e que aborda temas não relevantes a este trabalho, absorveu-se os principais tópicos que se adaptaram melhor ao assunto tratado.

Desta maneira, dividiu-se a especificação do projeto em três visões: negócios - detalhando as questões mercadológicas -, engenharia - particularizando todos os requisitos - e segurança - esmiuçando os requisitos mais críticos.

5.1 Visão de negócios

Sob a óptica do *Design Thinking* [19], apresentada no capítulo de Fundamentação Teórica, identificar as oportunidades através das dores dos usuários foi essencial para que o projeto não tivesse apenas valor técnico, mas também para o mercado e o consumidor.

A partir do que foi discutido no capítulo de fundamentação teórica, na seção Economia informal e o setor de pagamentos no Brasil, existe uma oportunidade no mercado de substituir o modelo atual de adquirente-bandeira por uma única empresa que seja responsável pela comunicação entre pagadores e recebedores. Desta maneira é possível identificar dois atores que se relacionam com o produto: lojista e consumidor.

Dentro do contexto de mercado, foi detectada também a oportunidade de envolver um terceiro ator no ecossistema do projeto: as empresas. Pessoas jurídicas que distribuem benefícios a seus colaboradores se mostraram um potencial cliente alvo, dado que não dispõem de uma boa plataforma que facilite este processo administrativo. Além disso, adquirir usuários através de serviços prestados a companhias é uma forma de tornar a estratégia de crescimento mais crível, dado que cada contrato negociado representa vários

usuários adquiridos de uma só vez.

As formas de pagamento viabilizadas pelos lojistas, de certa forma, impactam em seu potencial de venda, já que devem ser compatíveis com a forma que os clientes pretendem utilizar. Principalmente no ramo alimentício, é de grande interesse dos comerciantes atender colaboradores que recebem como benefício os cartões de vale-refeição. Entretanto, para que isso seja possível, eles se tornam reféns das altas taxas de adquirência cobradas pelas empresas que dominam o mercado de benefícios. Esta realidade faz com que os lojistas tenham que tomar a decisão estratégica de restringir as formas de pagamento ou arcar com as altas taxas.

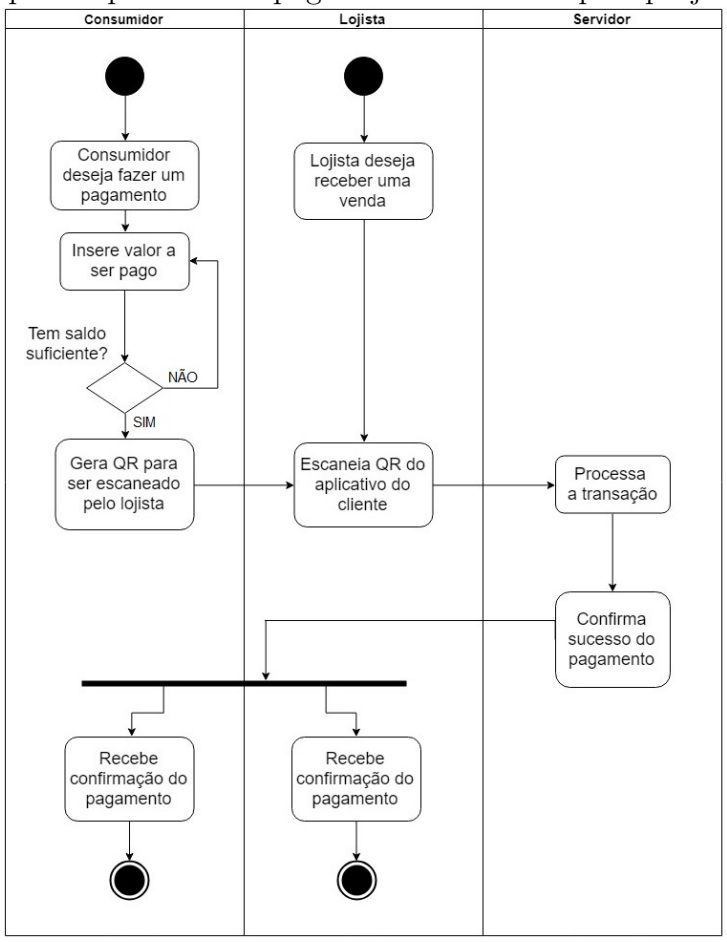
A proposta do projeto é eliminar o terminal de cartões como intermediário e, dessa forma, proporcionar maior liberdade para os comerciantes não dependerem das empresas dominantes.

As Figura 11 e 12 apresentam de forma mais clara e detalhada a sequência de etapas executadas durante o processamento do pagamento proposto pelo projeto e o implantado pelas empresas tradicionais, respectivamente. Além de o modelo convencional envolver mais intermediários, cada etapa é um serviço oferecido por uma empresa a parte. Como resultado, tem-se altas taxas e maiores vulnerabilidades de segurança. O projeto resolve esta questão englobando todo o processamento em um ecossistema único, eliminando a participação de terceiros.

Através da oferta de taxas menores, semelhantes àquelas cobradas em cartões de crédito e débito, o projeto tem como objetivo oferecer a oportunidade para lojistas que não são do ramo alimentício e não aceitam os vales-refeição, por conta da alta porcentagem cobrada pelas adquirentes, a aceitarem pagamentos via celular. O mesmo vale para os comerciantes informais apresentados na seção de Economia informal. Ou seja, há a criação de um cenário benéfico tanto para o comerciante, que passa a atender uma parcela de pessoas que não podiam antes comprar em suas lojas, quanto para os colaboradores, que passam a ter mais liberdade no momento da compra.

As empresas têm apresentado uma forte tendência em oferecer melhores benefícios com o objetivo de aumentar indiretamente a remuneração dos colaboradores e, desta forma, eliminar o absenteísmo, aumentar a produtividade e satisfação dos trabalhadores, atrair e reter talentos e valorizar a equipe. Além disso, foi detectado que as companhias enfrentam processos pouco otimizados relacionados a distribuição e gestão dos benefícios para seus colaboradores. Portanto, o projeto apresenta grande valor para estas empresas que têm interesse em oferecer bons benefícios e também buscam uma forma mais fácil e

Figura 11: Etapas do processo de pagamento executado pelo projeto desenvolvido



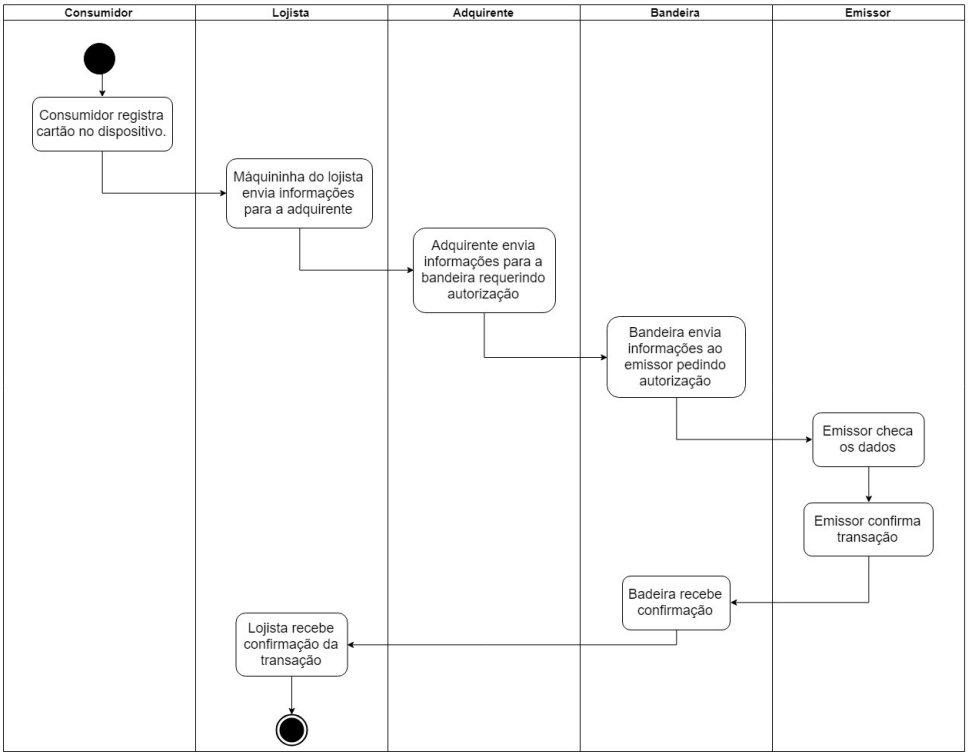
centralizada de controlar e administrar os recursos destinados a este fim.

Para os profissionais, trabalhos que oferecem vantagens além do salário também podem ser um destaque no momento de escolha de carreira e isso inclui não só montante de remuneração voltada para objetivos específicos, mas também a flexibilidade de poder decidir a forma como desejam aproveitar o benefício recebido.

Vale ressaltar que a estratégia vai de acordo com muitas organizações que já apresentam a tendência de entender a preferência de seus colaboradores e criar pacotes de benefícios flexíveis, que atendam às necessidades e desejo dos profissionais do mercado atual. Além disso, o projeto representa a modernização de um modelo de vales e adquirente-bandeira instaurados há anos e que apresentaram poucas mudanças diante do cenário de transformação digital em que estamos inseridos.

Conforme tratado no capítulo Trabalhos Relacionados, existem duas principais tecnologias aplicadas em pagamentos via celular: código QR e NFC. Durante a etapa de planejamento do projeto foram estudados os dois modelos que poderiam atender aos requisitos do projeto. Embora o NFC seja a opção mais segura [35], o grupo optou pelo

Figura 12: Etapas do processo de pagamento padrão executado pelas grandes empresas do mercado



código QR, principalmente porque se mostrou comercialmente mais viável, já que o único pré-requisito de hardware é um *smartphone* com câmera fotográfica. Vale ressaltar que foram tomadas medidas para aumentar a segurança da implementação do código QR para intermediar as transações realizadas. Estas serão esclarecidas na seção de Visão de Segurança.

A decisão pelo código QR foi impulsionada por três dificuldades na adoção do NFC. A primeira é a barreira econômica, dado que os *smartphones* que possuem NFC custam a partir de 1279 reais e os que possuem minimamente uma câmera para leitura do QR podem ser adquiridos a partir de 245 reais (valores consultados em novembro de 2018) [39]. A segunda barreira é a tecnológica, uma vez que por ser uma tecnologia mais cara e mais recente, não existem ainda tantos dispositivos com NFC em circulação quanto celulares com câmeras. Por fim, a barreira social, devido ao fato de apenas uma pequena parcela da população conhece ou se sente confortável em usar soluções com NFC [40].

5.2 Visão de engenharia

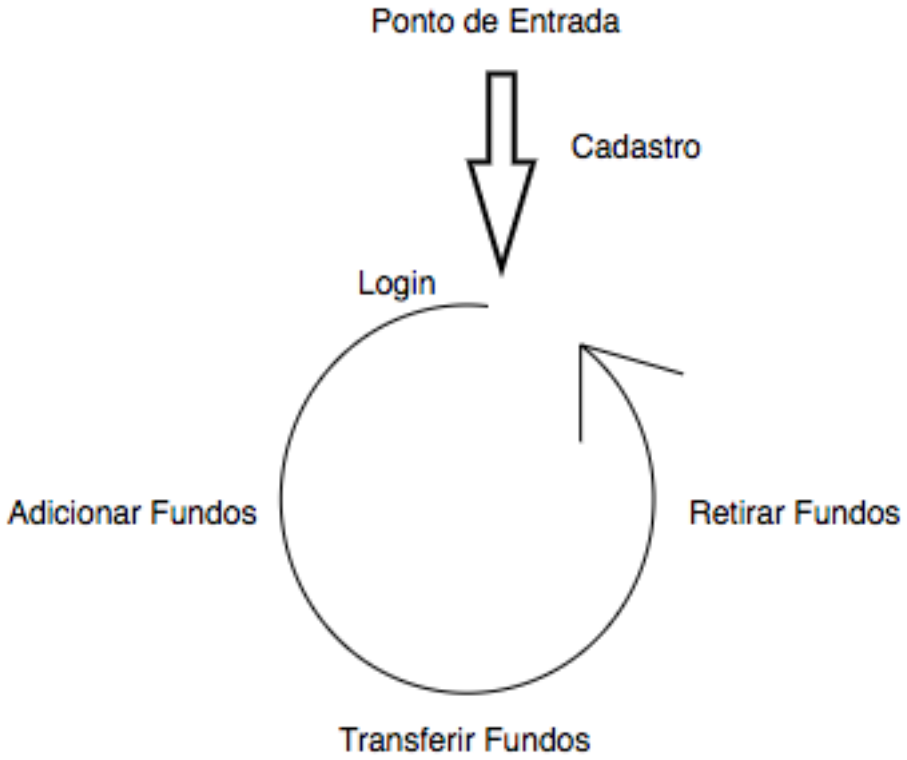
Nesta parte do projeto, há a especificação dos requisitos funcionais e não funcionais que dialogam com a visão de negócios e objetivam resolver os problemas observados para

cada um dos três atores envolvidos no sistema.

5.2.1 Requisitos funcionais

O trabalho está baseado no ciclo de vida dos fundos de um usuário em nossa plataforma, conforme ilustrado pela Figura 13. Seguindo essa linha de raciocínio, tudo se inicia em um cadastro de pessoa jurídica, seja ele de um lojista ou de uma empresa que deseja distribuir benefícios entre seus colaboradores. Após entrar dados específicos da empresa e pontos de contato válidos como telefone e e-mail, a empresa obtém acesso ao painel de controle.

Figura 13: Ciclo de vida de ações de usuários na plataforma.



O cadastro de usuários pessoa física pode ser ativo, em que a pessoa entra com informações pessoais próprias e uma senha, adiciona um ponto de contato - seja ele um e-mail ou um número de celular. Ou então na forma passiva, em que uma empresa deseja distribuir os benefícios de seus colaboradores e, para tal, é necessário informar os mesmos dados cadastrais e um ponto de contato.

Como resultado, cada usuário registrado recebe uma confirmação através de uma mensagem de texto ou correio eletrônico. Após essa primeira etapa e a confirmação do ponto de contato, é possível acessar o aplicativo utilizando o CPF como identificador e

uma senha para realizar login.

Para poder ter dinheiro na conta, qualquer usuário dependerá de uma empresa para lhe adicionar fundos, ou seja, nenhuma pessoa física terá o poder de movimentar fundos para dentro ou para fora do sistema. A empresa que deseja depositar fundos para distribuir entre seus colaboradores, terá um espaço no painel de controle para cadastrá-los, informar o valor que cada um deve receber e, então, um boleto será gerado e deverá ser pago fora da plataforma. Quando o boleto for corretamente quitado, os fundos estarão disponíveis para os usuários correspondentes.

Para realizar pagamentos, ambos os atores precisam ter contas distintas na plataforma: um usuário - o pagador - e um lojista - o recebedor. Para a realização do pagamento, o pagador deve gerar seu código de identificação e entrar com o valor a ser pago; o recebedor deve entrar com esse código no seu aplicativo e confirmar o valor.

Para retirar fundos do sistema, o lojista - que é o único ator com essa função disponível - deve entrar com informações bancárias e requisitar uma transferência.

Analisando o ciclo completo, é possível determinar oito requisitos funcionais que são detalhados a seguir: cadastro de pessoa jurídica, cadastro de pessoa física, login no aplicativo de pagamento, login no aplicativo de recebimento, login no painel de controle, adicionar fundos, pagar e receber, e retirada de fundos.

- **Cadastro de pessoa jurídica:** este requisito determina que o sistema deve recolher informações da empresa - que estão determinados no caso de uso de cadastro no Apêndice A - que desejar utilizar os serviços oferecidos. Após essa primeira interação, a empresa é contatada automaticamente exigindo uma confirmação do e-mail a fim de se garantir um ponto de contato.

- **Cadastro de pessoa física:** este requisito determina que o sistema deve recolher informações de pessoas físicas - que estão determinados no caso de uso de cadastro no Apêndice A. Este passo pode ser cumprido pela própria pessoa - caracterizando um cadastro ativo - ou por um representante da empresa na qual ela trabalha - um cadastro passivo.

- **Login no aplicativo de pagamento:** o sistema deve permitir acesso ao aplicativo de pagamento para todo usuário pessoa física que confirmar o seu ponto de contato, seja ele um e-mail ou número de celular.

- **Login no aplicativo de recebimento:** o sistema deve permitir acesso ao aplica-

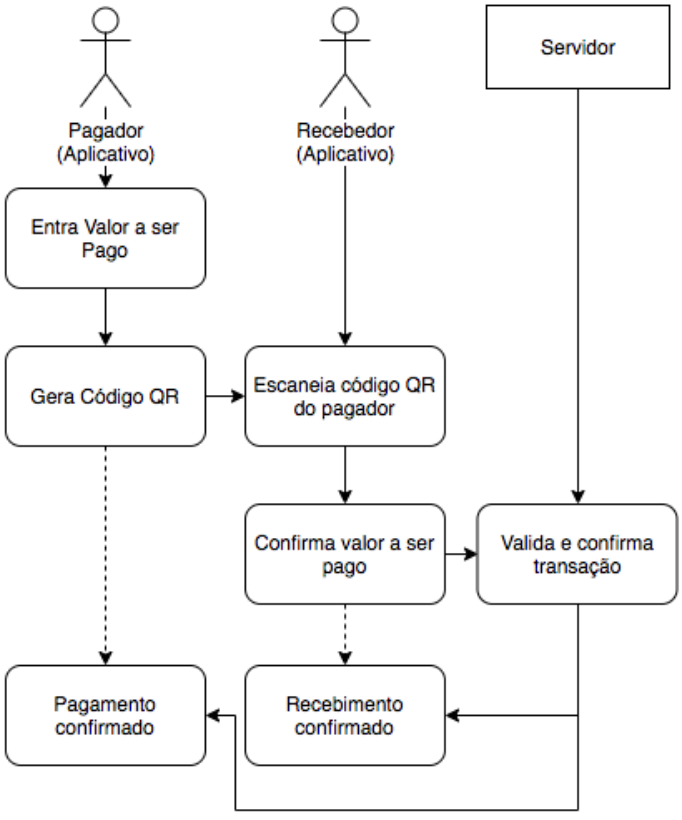
tivo de recebimento para todo usuário lojista que confirmar o seu ponto de contato.

- **Login no painel de controle:** O sistema deve permitir acesso ao painel de controle para todo usuário pessoa jurídica, seja ele um lojista ou uma empresa com colaboradores, que confirmar o seu ponto de contato.

- **Adicionar fundos:** o sistema deve permitir que uma empresa adicione fundos por meio do painel de controle para qualquer um de seus colaboradores registrados na plataforma através do cadastro passivo.

- **Pagamento e recebimento:** o sistema deve permitir que um usuário pessoa física, através do aplicativo de pagamento, efetue um pagamento direcionado a um lojista que esteja utilizando o aplicativo de recebimento. O processo está ilustrado na Figura 14.

Figura 14: Diagrama ilustrativo do processo de transferência de fundos entre um pagador e um recebedor. Além dos dois atores, há a presença do servidor como intermediário.



- **Retirada de fundos:** o sistema deve permitir que um lojista peça, através do aplicativo de recebimento, para que seus fundos sejam transferidos para sua conta bancária.

5.2.2 Requisitos não funcionais

A partir da definição dos requisitos funcionais do sistema, é possível determinar quais são os requisitos não funcionais requeridos para o correto funcionamento do projeto. São eles:

- Segurança:
  - garantir a confidencialidade e integridade de dados de usuários;
  - garantir a confidencialidade das senhas dos usuários;
  - garantir a confidencialidade e integridade de mensagens trocadas entre servidor e cliente;
  - garantir a autenticidade de cada usuário durante uma transação.
- Disponibilidade: o sistema deve mirar um funcionamento incessante, com 100% de disponibilidade.
- Portabilidade (ou interoperabilidade): o sistema deve estar disponível para plataformas iOS, Android e *Web*.

5.3 Visão de segurança

Esta seção detalha as especificações dos cumprimentos dos requisitos não funcionais de segurança do projeto. Há a descrição dos momentos críticos para garantir que as informações adicionadas por usuários estejam seguras e que as transações financeiras ocorram sem possíveis interferências maliciosas externas.

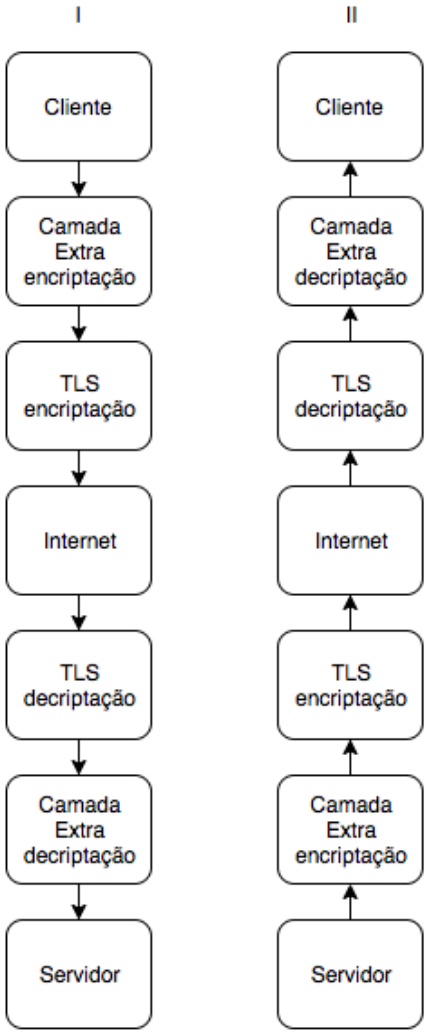
Conforme separado na seção da Visão de engenharia, os requisitos não funcionais de segurança se divide em duas principais facetas: as garantias de confidencialidade e integridade; e aquelas de autenticidade.

5.3.1 Confidencialidade e integridade

Os requisitos funcionais que envolvem transferências de dados entre servidor e cliente - seja ele *Web* ou aplicativo - ou cliente e servidor necessitam de cuidados ao transportar essas informações pela rede. Para tal, utilizamos o protocolo TLS 1.2, que pode ser configurado na AWS, onde o servidor está hospedado, podendo-se escolher a combinação

de algoritmos a ser utilizada. Além disso, optamos por utilizar uma camada extra de encriptação como ilustrado na figura 15.

Figura 15: Modelo de segurança com camada extra de proteção e HTTPS. No diagrama I, a informação trafega a partir do cliente para o servidor; no diagrama II o sentido é o oposto.



Utilizamos um modelo de encriptação híbrida, com RSA e AES, que foi previamente detalhada no capítulo de fundamentação teórica. Por isso, é necessário utilizar dois algoritmos criptográficos: um assimétrico e outro simétrico.

Com a aplicação desta camada extra, garante-se a integridade e a confidencialidade dos dados de qualquer ator que utilizar a plataforma.

5.3.2 Autenticidade

O principal requisito funcional do projeto é o de pagamento e recebimento, que constitui a ação mais crítica da plataforma, envolvendo uma transação financeira. Desta maneira, é imprescindível que o pagador tenha sua autenticidade garantida, por isso, a atenção com a configuração da assinatura digital é bastante importante.

Conforme explicado no capítulo de fundamentação teórica, a assinatura digital envolve o uso de um algoritmo de criptografia assimétrico, em que a chave privada permanece com o assinante (no caso, o pagador) e a chave pública pode ser usada por qualquer um para verificar a origem e autenticidade daquela assinatura.

Cada transação será assinada por ambas as partes. Dessa forma podemos garantir a autenticidade do pagador e se ele realmente possui saldo suficiente para realizar a transação. E garantimos a autenticidade do recebedor, para validarmos a transferência e atualizarmos os saldos das duas partes.

6 IMPLEMENTAÇÃO

A implementação do projeto foi modularizada. Assim, cada módulo corresponde a um serviço que pode ser isolado e olhado na forma de uma caixa preta do ponto de vista dos demais serviços. A divisão foi feita da seguinte maneira:

- Módulo I: constituído pelo servidor e pelo banco de dados;
- Módulo II: constituído pelos aplicativos de pagamento e recebimento;
- Módulo III: constituído pelo painel de controle *web*.

A Figura 16 ilustra um diagrama alto nível da integração entre os módulos.

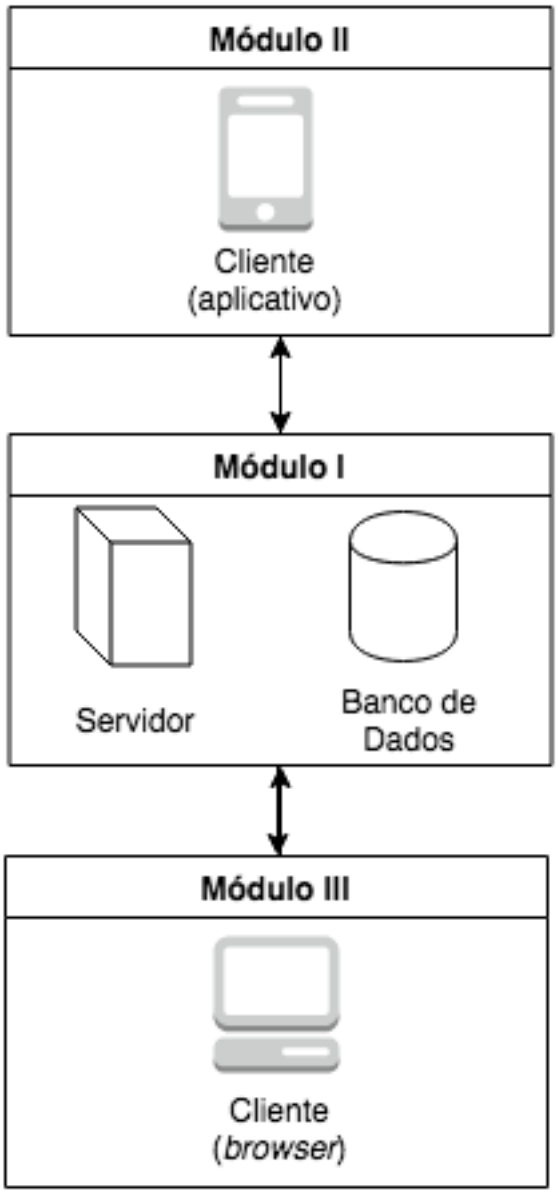
6.1 Módulo I: servidor e banco de dados

O primeiro módulo consiste numa parte centralizadora e única, ou seja, tanto os aplicativos quanto o cliente *web* se comunicam o servidor e, este, com o banco de dados.

6.1.1 Arquitetura

A arquitetura do servidor foi pensada e implementada a partir de uma estrutura *serverless*. Ou seja, cada ponto de contato que o servidor faz com qualquer sistema externo foi encapsulado em uma função e colocado no ar de forma independente. Estes pontos de contato referem-se aos requisitos funcionais, *i.e.* cada um dos requisitos definidos no capítulo de especificação do projeto constitui uma função e uma rota específicas. Desta maneira, o mal funcionamento de um elemento não interfere na execução de outro, contribuindo para aumentar a disponibilidade do sistema como um todo. A Figura 17 exemplifica o funcionamento de um serviço de nuvem que fornece hospedagem de servidores no modelo *serverless*, ou seja, as funções de *login* e cadastro estão funcionando de maneira independente.

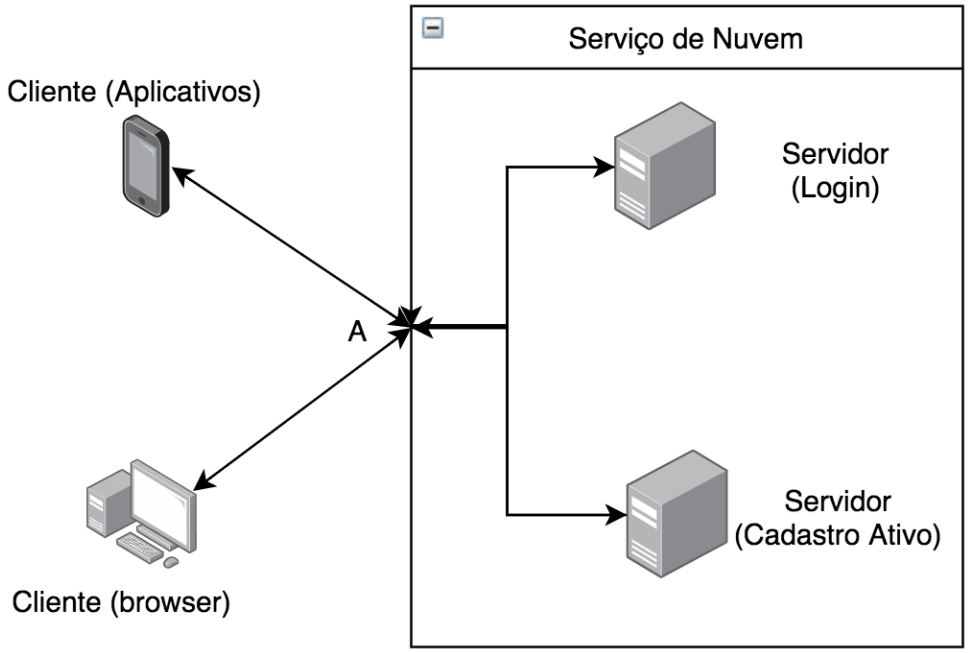
Figura 16: Diagrama alto nível de integração dos módulos do sistema.



Além da independência entre as funções, a estrutura *serverless* traz outros benefícios em relação a um modelo de servidor padrão, visto que há a terceirização de preocupações ligadas a manutenção, aumenta a velocidade de implementação de mudanças e novas funcionalidades e garante custos operacionais mais baixos [41].

O banco de dados - que está conectado somente às funções do servidor - escolhido foi no modelo não estruturado, visto que traz: maior flexibilidade em relação ao esqueleto de dados do banco; maior velocidade de escrita e leitura; e maior escalabilidade [42]. O banco de dados estruturado tem grande vantagem quando se trata da realização de pesquisas complexas, porém o projeto em questão não apresenta tal necessidade a ponto de se abrir mão dos benefícios da base não-estruturada. Como o projeto ainda está no

Figura 17: Hospedagem independente de funções encapsuladas em um serviço de nuvem.



início, a flexibilidade e facilidade de se adicionar um novo campo no banco não estruturado pesaram no momento de se decidir qual modelo usar.

6.1.2 Tecnologias

As ferramentas usadas neste módulo foram baseadas na decisão de se construir um sistema *serverless*. Assim, selecionou-se o AWS Lambda, que é o pioneiro no oferecimento de função como serviço - do inglês, *function-as-a-service*.

Como o Lambda oferece somente um serviço de encapsulamento de funções, foi necessário utilizar outro serviço para tratar o interfaceamento com os outros módulos: o AWS API Gateway, que é representado pelo ponto A na Figura 17. Além do interfaceamento, o API Gateway oferece outros benefícios ao sistema, como o controle de volume e registro de acessos, delineamento de códigos de resposta HTTP e gerenciamento de cache dados.

Baseando-se nas opções disponíveis a serem implantadas no AWS Lambda, escolheu-se NodeJS - um ambiente de execução de códigos em Javscript - para ser a base de codificação do projeto. Como discutido no capítulo de fundamentação teórica, a linguagem Javascript já possui papel relevante na comunidade internacional, sendo habitualmente usada para a construção de servidores.

Em relação ao banco de dados, por não existirem no projeto pesquisas complexas,

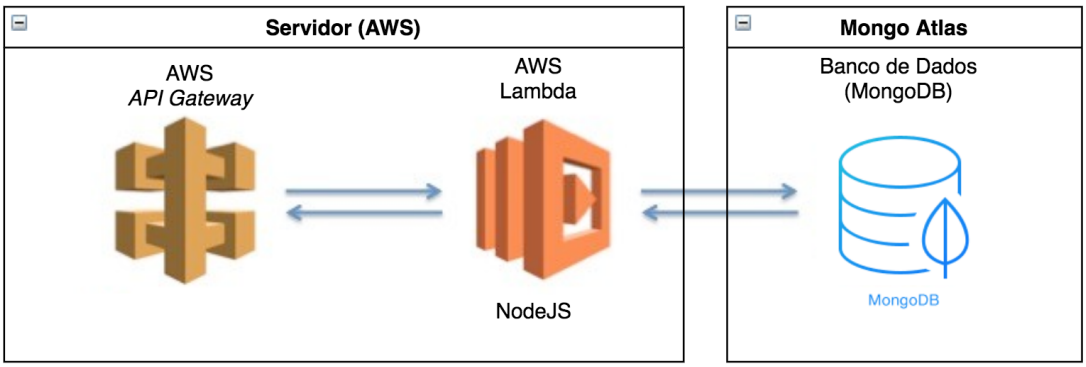
escolheu-se um banco não-estruturado e, dentre as possibilidades disponíveis, o MongoDB pareceu a mais apropriada. Além de ser o banco não-estruturado mais popular na comunidade [24], desde o lançamento da versão 4.0, possui funcionalidades apropriadas para a persistência de transações, ou seja, cumpre as propriedades ACID:

- Atomicidade: quando há múltiplos documentos sendo alterados, a operação somente é concretizada quando todos são corretamente salvos.
- Consistência: somente operações lícitas devem ser permitidas, de modo a não causar corrupção no banco.
- Isolamento: durante uma operação, leituras são realizadas em uma fotografia do banco, tirada logo antes de se iniciar as alterações.
- Durabilidade: garantia de que mesmo com alguma quebra ou instabilidade, os dados sempre estarão corretamente salvos no banco.

O banco de dados também está hospedado em um serviço terceirizado, nesse caso, em um específico para o MongoDB chamado Mongo Atlas. Novamente, essa estratégia via garantir a alta disponibilidade do sistema e reduzir eventuais custos de manutenção.

Assim, o primeiro módulo do sistema ficou construído de acordo com a Figura 18.

Figura 18: Tecnologias para implementação do módulo I do sistema.



6.2 Módulo II: aplicativos de pagamento e recebimento

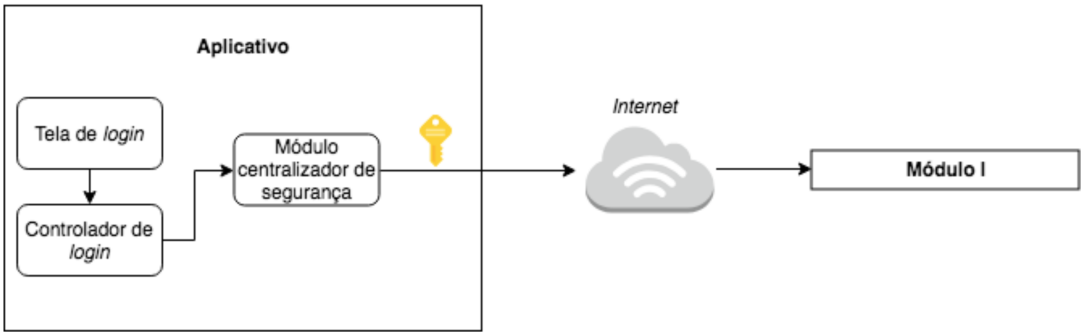
O segundo módulo, composto pelos dois aplicativos do sistema, possui uma arquitetura simples, priorizando a segurança das informações que passam por ele. O único elemento externo com o qual ele se conecta é o primeiro módulo do projeto, tornando possível a criptografia de toda informação que passa pela *internet*.

6.2.1 Arquitetura

Os aplicativos, mesmo tendo funcionalidades distintas, possuem uma organização arquitetural semelhante composta por telas e controladores. As telas são arquivos que contém construtores de elementos visuais e chamadas de funções dos controladores. Estas funções são responsáveis por conectar o cliente ao servidor.

Para tratar da segurança dos dados dos aplicativos, construiu-se um módulo centralizador que é chamado toda vez que uma requisição externa é necessária. A Figura 19 ilustra o papel do módulo quando informações de *login* são enviadas ao servidor.

Figura 19: Exemplo de utilização do módulo centralizador de segurança durante um evento de *login* nos aplicativos.



6.2.2 Tecnologias

Para se decidir qual seria a tecnologia a ser utilizada para realizar a implementação dos aplicativos, levou-se em consideração a facilidade de implementação do React-native-js por utilizar Javascript - assim como o servidor - e por desenvolver protótipos tanto para iOS quanto para Android a partir do mesmo código.

Além dessa facilidade, o React-native-js foi desenvolvido pelo Facebook, que ainda é um dos grande colaboradores e mantenedores da biblioteca, trazendo grande credibilidade ao *software*.

Buscando aproximar-nos de uma tecnologia de criptografia bem sucedida e segura, o TLS, os algoritmos escolhidos foram o AES (simétrico) e o RSA (assimétrico).

6.3 Módulo III: painel de controle

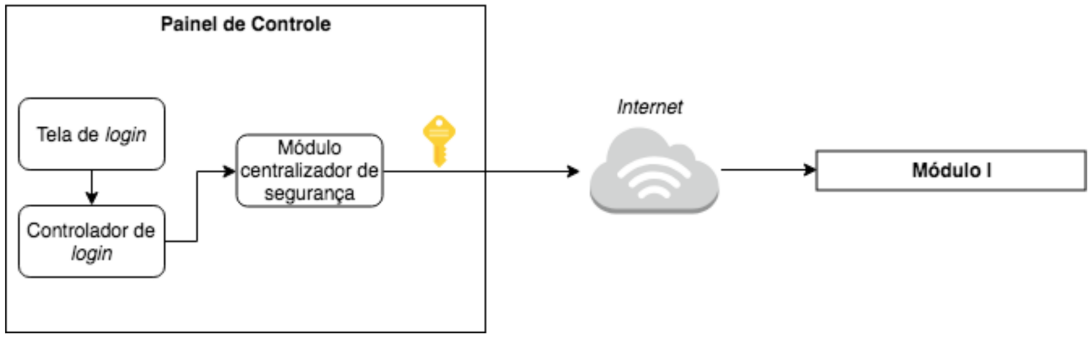
O terceiro módulo constitui o painel de controle em que os comerciantes podem realizar as ações de: cadastro, *login*, adição e retirada de fundos.

6.3.1 Arquitetura

A arquitetura do painel de controle, mesmo sendo um cliente *web*, se assemelha bastante com aquela estruturada para os aplicativos: telas e controladores.

Também semelhante ao módulo II, houve a construção de um elemento para intermediar as conexões com a *internet*, de modo a garantir a confidencialidade e integridade dos dados dos usuários. A Figura 20 é análoga à Figura 19, ou seja, da mesma maneira que o *login* realizado nos aplicativos possui suas informações criptografadas antes de viajar pela rede, o evento semelhante no painel de controle também tem seus dados codificados.

Figura 20: Exemplo de utilização do módulo centralizador de segurança durante um evento de *login* no painel de controle.



6.3.2 Tecnologias

Na escolha da implementação do *front-end* havia algumas opções a serem consideradas, todas utilizando a mesma linguagem dos módulos I e II, o Javascript: ReactJS, AngularJS e VueJS. Acabou-se decidindo utilizar o ReactJS devido a sua simplicidade e semelhança com as outras tecnologias utilizadas no projeto.

O React-native-js, utilizado no módulo II é uma biblioteca originada do ReactJS. Ou seja, as mesmas qualidades encontradas na versão para desenvolvimento móvel, são encontradas na versão *web* originária: uso frequente pela comunidade e manutenção de uma grande empresa como o Facebook.

6.4 Segurança dos módulos

A implementação das especificações de segurança abordaram os três módulos de maneira bastante semelhante, ou seja, quando dados são transferidos bidirecionalmente entre o cliente - sendo ele um dos aplicativos ou o painel *web* - e o servidor, o mesmo módulo

de criptografia, escrito em código Javascript, trata de codificar os conteúdos, através dos algoritmos simétrico e assimétrico.

O único ponto que o módulo II tem de diferença em relação aos outros é a assinatura digital durante a ação de pagamento e recebimento. Conforme explicado no capítulo de especificação do projeto, o pagador deve assinar o código QR a ser mostrado para o recebedor, que deve assinar a transação ao enviá-la ao módulo I.

6.4.1 Comunicação entre os módulos

As chaves douradas, nas Figuras 19 e 20, representam a codificação das informações que viajarão através da *internet* até o primeiro módulo do sistema. Quando chegar lá, a primeira tarefa a ser realizada é a decodificação dessas informações, conforme detalhado na seção de visão de segurança, no capítulo de especificação do projeto. Este processo utiliza o conceito de encriptação híbrida, conforme mencionado no capítulo de fundamentação teórica.

Buscando aproximar-nos de uma tecnologia de criptografia bem sucedida e segura, o TLS, os algoritmos escolhidos foram o AES (simétrico) e o RSA (assimétrico). Como os três módulos possuem a mesma lógica de segurança durante uma comunicação e são escritos em Javascript, a implementação desses algoritmos foi realizada a partir da biblioteca Node-forge [43], nativa de NodeJS.

6.4.2 Assinatura digital no pagamento

Como explicado na fundamentação teórica, por razões de performance e segurança, optamos pelo uso do EdDSA, com a curva Curve25519 para assinaturas digitais. Para tal, utilizamos a mesma biblioteca citada na subseção acima, a Node-forge [43]. Ela já possui a implementação puramente em JavaScript do Ed25519.

PARTE III

CONSIDERAÇÕES FINAIS

7 CONSIDERAÇÕES FINAIS

Esta capítulo apresenta o quanto os resultados obtidos se aproximam dos objetivos iniciais e quais seriam possíveis discussões para trabalhos futuros.

7.1 Cumprimento de objetivos

Relembrando o objetivo do projeto em partes, iniciou-se visando aplicar conceitos de segurança da informação e de engenharia de software. Conforme desenvolvido no capítulo de especificação do projeto, exercitou-se diversos tópicos de segurança da informação como forma de manter um sistema com requisitos não funcionais de segurança tão críticos. Além disso, o mesmo capítulo foi estruturado a partir da abstração de um modelo de referência - o RM-ODP. Desta maneira, é possível dizer que essa parcela do objetivo foi de fato cumprida com sucesso.

Continuando, o alvo central era a construção de um sistema de pagamentos com alguns requisitos mínimos; e, com a soma dos capítulos de especificação do projeto e implementação do projeto, é possível dizer que o sistema foi, de fato, construído.

Sobre o último tópico, relacionado às questões que o sistema deve combater, não foi possível tomar conhecimento deste atingimento de meta. Como não foram realizados testes suficientes e outras questões de negócios não foram discutidas, não se pode dizer que o produto final do trabalho pode já ir à mercado.

Por fim, com muitas metas cumpridas, foi possível identificar pontos a serem desenvolvidos em projetos derivados deste.

7.2 Trabalhos futuros

Existem algumas melhorias que podem constituir eventuais trabalhos derivados ou até mudanças de tema principal do projeto como forma de complementar a concepção do

sistema de pagamento em si. Alguns exemplos pensados são descritos a seguir:

- **Como estruturar testes para uma plataforma de pagamentos:** depois de implementar o projeto, percebeu-se a existência de diversos pontos a serem testados utilizando testes unitários, de integração e de sistema. Desta maneira, seria interessante constituir um projeto para testar todas as pontas de transações financeiras e garantir que o sistema fique, de fato, seguro.

- **Aprofundar a discussão de segurança:** o atual projeto chegou a discutir alguns algoritmos de criptografia e de assinatura digital, porém a oferta de alternativas na literatura é vasta. Assim, um projeto que discuta e teste possíveis implementações de algoritmos distintos ou, até mesmo num nível arquitetural, estruturas de proteção de dados e senhas com complexidades variadas.

- **Foco em usabilidade do usuário:** Assim como mencionado na seção anterior, existem etapas a serem cumpridas antes de o produto do trabalho poder ir à mercado. Um bom exemplo disso são testes de usabilidade das aplicações criadas e verificar a aceitação da população. Como o objetivo é ter um produto que seja usada de forma massificada, esses estudos são essenciais para garantir o sucesso do sistema.

- **Estudo de aceitabilidade do produto:** Entender se há necessidade de passos intermediários entre a implementação de um serviço em ecossistema fechado - ou seja, usar o aplicativo do projeto para pagar e receber transações. Por exemplo, verificar a viabilidade de integração do aplicativo de pagamento com terminais de adquirência como mecanismo de adoção dos comerciantes.

REFERÊNCIAS

[1] ETCO. *Índice de Economia Subterrânea*. [S.l.], 2016. Disponível em: <https://www.etco.org.br/11/wp-content/uploads/AF_RP1_Folder_ETCO_econsub_final.pdf>. Acesso em: 09 abr. 2018.

[2] GRADILONE, C. *A máquina de cartões do Safra*. 2017. Disponível em: <https://www.istoedinheiro.com.br/maquina-de-cartoes-do-safra/>. Acesso em: 10 mar. 2018.

[3] SOON, T. J. *QR Code*. 2008. Disponível em: <https://foxdesignsstudio.com/uploads/pdf/Three_QR_Code.pdf>. Acesso em: 12 nov. 2018.

[4] BELLINI, J. *China's Great Leap to Wallet-Free Living*. 2018. Disponível em: <https://www.wsj.com/articles/chinas-great-leap-to-wallet-free-living-moving-upstream-1516294172>. Acesso em: 25 mar. 2018.

[5] MILLWARD, S. *China now has 731 million internet users, 95% access from their phones*. 2017. Disponível em: <https://www.techinasia.com/china-731-million-internet-users-end-2016>. Acesso em: 25 mar. 2018.

[6] DOW JONES NEWSWIRES. *Fintech que nasceu do Alibaba chacoalha sistema bancário chinês*. 2018. Disponível em: <https://www.valor.com.br/financas/5697917/fintech-que-nasceu-do-alibaba-chacoalha-sistema-bancario-chines>. Acesso em: 21 ago. 2018.

[7] DELOITTE. *Leading the cashless charge – Evolution of the digital wallet industry in India*. [S.l.], 2017. Disponível em: <https://www2.deloitte.com/content/dam/Deloitte/in/Documents/strategy/in-strategy-leading-the-cashless-charge-noexp.pdf>. Acesso em: 15 abr. 2018.

[8] SHARMA, V.; GUSAIN, P.; KUMAR, P. Near field communication. 2013.

[9] MASSOTH, M.; BINGEL, T. Performance of different mobile payment service concepts compared with a nfc-based solution. In: *2009 Fourth International Conference on Internet and Web Applications and Services*. [S.l.: s.n.], 2009. p. 205–210.

[10] RAMARAJ, E.; KARTHIKEYAN, S.; HEMALATHA, M. A design of security protocol using hybrid encryption technique (aes- rijndael and rsa). 2009.

[11] JAJU, S. A.; CHOWHAN, S. S. Analytical study of modified rsa algorithms for digital signature. IJRITCC, 2015.

[12] ABDULLAH, A. M. Advanced encryption standard (aes) algorithm to encrypt and decrypt data. 2017.

[13] KAVALIRO. Aes example - input (128 bit key and message). 2014. Disponível em: <https://kavaliro.com/wp-content/uploads/2014/03/AES.pdf>. Acesso em: 11 nov. 2018.

[14] BARKER, E. Recommendation for key management. 2016.

[15] HALES, T. C. The nsa back door to nist. 2014.

[16] JOSEFSSON, S. *Edwards-Curve Digital Signature Algorithm (EdDSA)*. Disponível em: <https://tools.ietf.org/html/rfc8032>. Acesso em: 02 dez. 2018.

[17] BERNSTEIN, D. J. *A state-of-the-art Diffie-Hellman function*. Disponível em: <https://cr.yp.to/ecdh.html>. Acesso em: 02 dez. 2018.

[18] RAYMOND, K. Reference model of open distributed processing (rm-odp): Introduction. Springer Science+Business Media Dordrecht, 1995.

[19] BROWN, T. *Design Thinking*. 2008. Disponível em: <https://hbr.org/2008/06/design-thinking>.

[20] AMAZON. *Computação em nuvem com a Amazon Web Services*. Disponível em: <https://aws.amazon.com/pt/what-is-aws/>. Acesso em: 01 dez. 2018.

[21] AMAZON. *Amazon API Gateway*. Disponível em: <https://aws.amazon.com/pt/api-gateway/?nc2=type$_{a¿}$.*Acessoem* : 01*dez*.2018.

[22] AMAZON. *AWS Lambda*. Disponível em: <https://aws.amazon.com/pt/lambda/?nc2=type$_{a¿}$.A
01*dez*.2018.

[23] MONGODB. *What is MongoDB?* Disponível em: <https://www.mongodb.com/what-is-mongodb>. Acesso em: 02 dez. 2018.

[24] STACKOVERFLOW. *Developer Survey Results 2018*. 2018. Disponível em: <https://insights.stackoverflow.com/survey/2018/technology>. Acesso em: 01 dez. 2018.

[25] TEIXEIRA, P. *Professional NodeJS: Building JavaScript-Based Scalable Software*. Indianapolis, IN: John Wiley Sons, 2013.

[26] GACKENHEIMER, C. *Introduction to React*. Berkeley, CA: Apress, 2015.

[27] EISENMAN, B. *Learning React Native*. Sebastopol, CA: O'Reilly Media, 2016.

[28] KINDBERG, T.; SELLEN, A.; GEELHOED, E. Security and trust in mobile interactions: A study of users' perceptions and reasoning. Springer, Berlin, Heidelberg, 2004.

[29] KRISTOFFERSEN, S.; SYNSTAD, A.; SøRLI, K. Users' perception of mobile payment. 2008.

[30] KOW, Y. M.; GUI, X.; CHENG, W. Special digital monies: The design of alipayand wechat wallet for mobile payment practices in china. 2017.

[31] ENISA. Security of mobile payments and digital wallets. Enisa, 2016.

[32] SAMSUNG. *What is the difference between NFC and MST*. 2018. Disponível em: <https://www.samsung.com/us/support/answer/ANS00043949/>. Acesso em: 30 nov. 2018.

[33] ZMIJEWSKA, A. Evaluating wireless technologies in mobile payments - a customer centric approach. IEEE, 2005.

[34] ONDRUS, J.; PIGNEUR, Y. An assessment of nfc for future mobile payment systems. IEEE, 2007.

[35] VAZQUEZ-BRISENO, M. et al. Using rfid/nfc and qr-code in mobile phones to link the physical and the digital world. IntechOpen, 2012.

[36] MAHALLE, V. S.; SHAHADE, A. K. Enhacing the data security in cloud by implementing hybrid (rsa aes) encryption algorithm. Enisa, 2014.

[37] SIMPLICIO, M. et al. Lyra2: Efficient password hashing with high security against time-memory trade-offs. IEEE, 2016.

[38] HOSSAIN, M.; ISLAM, R. A model for ensuring data security to distributed financial system in cloud storage. IEEE, 2017.

[39] SMARTPHONES com NFC. Mais celular, 2018. Disponível em: <https://www.maiscelular.com.br/pesquisa/?aparelho=1funcao=2o=3z=2>. Acesso em: 30 nov. 2018.

[40] LUNA, R. de et al. Aceitação da tecnologia nfc para pagamentos móveis: Uma perspectiva brasileira. Revista Brasileira de Gestão de Negócios, 2017. Disponível em: <http://www.redalyc.org/articulo.oa?id=94749795005>. Acesso em: 30 nov. 2018.

[41] CABOT TECHNOLOGY SOLUTION. *The Benefits of Serverless Computing and its Impact on DevOps*. 2016. Disponível em: <https://hackernoon.com/the-benefits-of-serverless-computing-and-its-impact-on-devops-e75d82c47ac4>. Acesso em: 20 nov. 2018.

[42] XPLENTY. *The SQL vs NoSQL Difference: MySQL vs MongoDB*. 2017. Disponível em: <https://medium.com/xplenty-blog/the-sql-vs-nosql-difference-mysql-vs-mongodb-32c9980e67b2>. Acesso em: 20 nov. 2018.

[43] DIGITAL BAZAAR. *Forge Documentation*. Disponível em: <https://github.com/digitalbazaar/forge>. Acesso em: 02 dez. 2018.

GLOSSÁRIO

Adquirentes Fazem a liquidação financeira das transações por meio de cartão de crédito e cartão de débito. São empresas como Rede, Cielo, Elavon, GetNet, Stone, entre outras, que são responsáveis pela comunicação com as bandeiras e bancos emissores. 19

Bandeira é responsável por processar todas as transações que utilizem seu cartão de crédito, analisando também seu perfil de consumo e repassando essas informações para o banco ou instituição financeira que emitiu o cartão de crédito. É a bandeira do cartão que permite, através de um sistema integrado globalmente, que você utilize o mesmo cartão de crédito em qualquer lugar do mundo. 19

Conexão *peer-to-peer* é uma arquitetura em que cada um dos nós funciona tanto como cliente quanto como servidor, permitindo compartilhamentos de informação sem a necessidade de um servidor central. 22

FaceID sistema de reconhecimento facial desenvolvido pela Apple. 34

HTTPS *Hyper Text Transfer Protocol Secure*, que em português significa "Protocolo de Transferência de Hipertexto Seguro". É a união entre HTTP e o protocolo de segurança TLS, adicionando uma camada extra de criptografia ao primeiro. 45

M*essage digests* são funções hash unidirecionais seguras que pegam dados de tamanho arbitrário e geram um hash de tamanho fixo. 22

Pessoa física é todo ser humano enquanto indivíduo, do seu nascimento até a morte. Essa designação é um conceito jurídico e se refere especificamente ao indivíduo enquanto sujeito detentor de direitos e de deveres. 41

Pessoa jurídica é um entidade formada por indivíduos e reconhecida pelo Estado como detentora de direitos e deveres. O termo pode se referir a empresas, governos, organizações ou qualquer grupo criado com uma finalidade específica. 42

TLS *Transport Layer Security*, que em português significa "Segurança da Camada de Transporte". É uma camada de criptografia assimétrica adicionada em cima do protocolo HTTP para formar o protocolo HTTPS. 44

APÊNDICE A – CASOS DE USO

1. Cadastro

   (a) Descrição

   O usuário abre o aplicativo e clica no botão de registro (ou Sign Up), entra com informações básicas e se cadastra. O sistema faz algumas validações e envia uma mensagem para o usuário confirmar o seu e-mail.

   (b) Ator

   Usuário do aplicativo, seja uma pessoa física ou um representante de uma pessoa jurídica.

   (c) Pré-condição

   O celular deve estar conectado à internet.

   (d) Fluxo Básico

   i. Usuário clica no botão de registro;
   ii. Sistema pergunta se o cadastro será de pessoa física ou jurídica;
   iii. Usuário escolhe a opção de pessoa física;
   iv. Sistema redireciona para a tela de cadastro de pessoa física;
   v. Usuário entra com nome, sobrenome, cpf, data de nascimento, email e celular com DDD;
   vi. Sistema verifica se email, celular, cpf estão nos formatos corretos e não cadastrados no sistema;
   vii. Sistema retorna uma mensagem para o usuário confirmar o email cadastrado.

   (e) Fluxos Alternativos

i. FA1: Usuário deseja cadastrar pessoa jurídica (passo 3)

A. Usuário escolhe a opção de pessoa jurídica;
B. Sistema redireciona para a tela de cadastro de pessoa jurídica;
C. Usuário entra com nome, sobrenome, nome da empresa, cnpj, email e celular com DDD;
D. Sistema verifica se email, celular, cnpj estão nos formatos corretos e não cadastrados no sistema;
E. Sistema retorna ao passo 7.

ii. FA2: Entrada de pessoa física com erros ou já registrada no sistema (passo 7)

A. Sistema retorna mensagem informando que email, celular ou cpf está com erro e pede correção;
B. Sistema retorna ao passo 5.

iii. FA3: Entrada de pessoa jurídica com erros ou já registrada no sistema (passo FA1.e)

A. Sistema retorna mensagem informando que email, celular ou cnpj está com erro e pede correção;
B. Sistema retorna ao passo FA.c.

(f) Pós-condição
Usuário está cadastrado no sistema, porém sua conta ainda não está ativada. Para ativá-la, é necessário clicar no link enviado ao email cadastrado.

2. Recebimento

(a) Descrição
Usuário deseja receber dinheiro de outro, clica no botão de recebimento, escaneia código QR de identificação do pagador e confirma valor a ser pago.

(b) Ator
Usuário podendo ser pessoa física ou jurídica.

(c) Pré-condição
Usuário deve estar logado no sistema.

(d) Fluxo Principal

i. Usuário, a partir da página inicial do aplicativo, clica no botão de receber.
ii. Sistema redireciona para a tela de recebimento;
iii. Usuário utiliza a câmera do celular para escanear código QR de identificação do pagador;
iv. Sistema valida assinatura do pagador e mostra valor que o pagador deseja pagar;
v. Usuário confirma o valor;
vi. Sistema confirma o pagamento e mostra o comprovante;

(e) Fluxos Alternativos

i. FA1: Usuário possui celular sem câmera (passo 3)

A. Usuário escolhe a opção de digitar código de identificação do pagador;
B. Retorna ao fluxo principal no passo 4;

ii. FA2: Usuário não confirma a transação (passo 5)

A. Retorna ao passo 3;

iii. FA3: Sistema não confirma a transação (por qualquer motivo possível - passo 6)

A. Sistema devolve mensagem de erro detalhando o problema;
B. Retorna ao passo 3;

(f) Pós-condição
Recebimento efetuado.

3. Pagamento

(a) Descrição
Usuário clica no botão 'Pagar'. Entra com a quantidade a ser paga em 'Gerar QR code de pagamento'

(b) Ator

Usuário podendo ser pessoa física ou jurídica.

(c) Pré-condição

Usuário deve estar logado no sistema (não necessariamente online)

(d) Fluxo Principal

i. Usuário clica na opção de pagar; Sistema redireciona para a tela de pagamento;
ii. Usuário entra com um valor a ser transferido
iii. Sistema pede uma confirmação do pagamento, mostrando o valor;
iv. Usuário confirma o pagamento.
v. O QR code é gerado.

(e) Fluxos Alternativos

i. FA1: Usuário entra com um valor maior do que o saldo local informa (passo 3)

A. Usuário clica no botão de inverter a câmera para utilizar a frontal do aparelho;
B. Sistema retorna ao passo 3.

ii. FA2: Usuário escaneia código QR inválido (passo 3)

A. Sistema retorna mensagem de erro e pede para o usuário escanear o código QR novamente;
B. Sistema retorna ao passo 2.

iii. FA3: Usuário pagador com saldo insuficiente (passo 3)

A. Sistema retorna mensagem de erro falando que o usuário está com saldo insuficiente para executar a operação;
B. Sistema retorna ao passo 2.

iv. FA4: Usuário cancela compra (passo 4)

A. Usuário cancela a compra;

B. Sistema retorna ao passo 2.

(f) Pós-condição

Pagamento efetuado e dinheiro virtual transferido da carteira do pagador para a carteira do recebedor.

4. Retirar dinheiro

(a) Descrição

Usuário pessoa física ou jurídica clica no botão de retirar dinheiro. E o valor é subtraído da carteira virtual.

(b) Ator

Usuário pessoa jurídica.

(c) Pré-condição

Usuário deve estar logado no sistema e com acesso à internet.

(d) Fluxo Principal

i. Usuário clica na opção de retirar dinheiro;
ii. Sistema redireciona para a tela de retirar dinheiro;
iii. Usuário entra com valor a ser retirado;
iv. Sistema pede uma confirmação da retirada de dinheiro;
v. Usuário confirma a retirada de dinheiro.

(e) Fluxos Alternativos

i. FA1: Usuário com saldo insuficiente (passo 3)

A. Sistema retorna mensagem falando que o saldo em conta é insuficiente para efetuar a operação
B. Sistema retorna ao passo 3.

ii. FA2: Usuário cancela operação (passo 4)

A. Usuário cancela operação;
B. Sistema retorna ao passo 3.

(f) Pós-condição

Pagamento efetuado e dinheiro virtual transferido da carteira do pagador para a carteira do recebedor.