

ISO TEST: UMA FERRAMENTA PARA TESTES DE MENSAGENS NO FORMATO ISO 8583

Bruno de Sousa Much¹

RESUMO

Em meios de pagamento utilizando cartões de pagamento, utiliza-se como convenção de mercado a especificação definida pela norma ISO 8583 para troca de mensagens entre emissores, bandeiras e adquirentes. Essa especificação possui diferentes versões e implementações para os diferentes tipos de transações, emissores, bandeiras e adquirentes, tendo assim um número considerável de variações que a implementação de uma mensagem pode possuir. Ao enviar mensagens fora do padrão definido pela bandeira, existe o risco da rejeição do pagamento, onde os componentes deixam de receber parte dos valores da transação e um risco de multa por trafegar mensagens fora da implementação exigida pelas bandeiras de cartão. Com isso, torna-se complexa e arriscada a implementação e manutenção de sistemas que utilizam mensagens da norma ISO 8583 para comunicação. Por isso, este trabalho apresenta um protótipo que permite a criação de casos de teste para diferentes mensagens do padrão ISO 8583, permitindo validação de forma automatizada de diferentes variações das mensagens através dos casos de teste.

Palavras-chave: ISO 8583; Testes; Cartões de pagamento.

ABSTRACT

In payment methods using payment cards, the specification defined by the ISO 8583 standard for exchanging messages between issuers, brands and acquirers is used as a market convention. This specification has different versions and implementations for different types of transactions, issuers, brands and acquirers, so there are a considerable number of variations that the implementation of a message can have. When sending messages outside the standard defined by the card brand, there is a risk of payment rejection, whereby the components no longer receive part of the transaction value, and a risk of fines for sending messages outside the implementation required by the card brands. Thus, this work presents a prototype that allows the creation of test cases for different messages of ISO 8583 standard, which allows the automated validation of different variations of the messages through the test cases.

Palavras-chave: ISO 8583; Tests; Payment Cards.

1 INTRODUÇÃO

Em transações utilizando cartões de pagamento, utiliza-se como convenção de mercado a especificação definida pela norma ISO 8583 (IBM) para troca de

¹ Discente do Curso de Ciência da Computação da Universidade La Salle- Unilasalle, matriculada na disciplina Trabalho de Conclusão de Curso II. E-mail: bruno.202010559@unilasalle.edu.br, sob a orientação Prof. Patricia Mangan Kayser. E-mail: patricia.mangan@unilasalle.edu.br Data de entrega: 02 Dezembro. 2024.

mensagens entre sistemas de pagamento. A especificação apresenta um formato padrão para as mensagens de pagamento, definindo campos padrão e campos reservados, utilizados por sistemas autorizadores e bandeiras para definir sua própria implementação. Sendo assim, uma mensagem mesmo seguindo um formato padrão, possui diferentes variações de implementação de acordo com o sistema.

Ao enviar mensagens fora do padrão especificado pelo sistema autorizador e/ou bandeira, existe o risco da rejeição do pagamento, onde os componentes deixam de receber parte dos valores da transação, e risco de multa por trafegar mensagens fora da especificação adequada. Com isso, torna-se complexa e arriscada a implementação e manutenção de sistemas que utilizam a norma ISO 8583 para comunicação. Isso implica a necessidade de inúmeros testes manuais e regressivos com o objetivo de validar possíveis falhas e/ou impactos adversos entre as diferentes implementações. Deste modo, foi definido como problema de pesquisa: como apoiar o processo de validação de mensagens ISO 8583, sua manualidade e criticidade relacionadas ao desenvolvimento?

A partir deste contexto e deste problema de pesquisa, o presente trabalho tem como objetivo geral desenvolver uma ferramenta que permita a criação de casos de teste para diferentes mensagens do padrão ISO 8583, permitindo validação de forma automatizada de diferentes variações das mensagens através dos casos de teste.

Para atender o objetivo geral proposto, são definidos os seguintes objetivos específicos:

- Realizar o análise e levantamento de requisitos;
- Definir as tecnologias mais apropriadas para a implementação;
- Implementar um protótipo da ferramenta proposta;
- Realizar testes e avaliações;

Do ponto de vista metodológico, a pesquisa realizada neste trabalho é do tipo aplicada, experimental e qualitativa, conforme os objetivos e os métodos adotados para o desenvolvimento do protótipo proposto.

Na continuidade deste artigo, será apresentado sobre o mercado de pagamentos e o processo de implementação do protótipo para atender aos objetivos estabelecidos, após apresentar os conceitos relacionados à norma ISO 8583.

2 Mercado de pagamentos por cartão

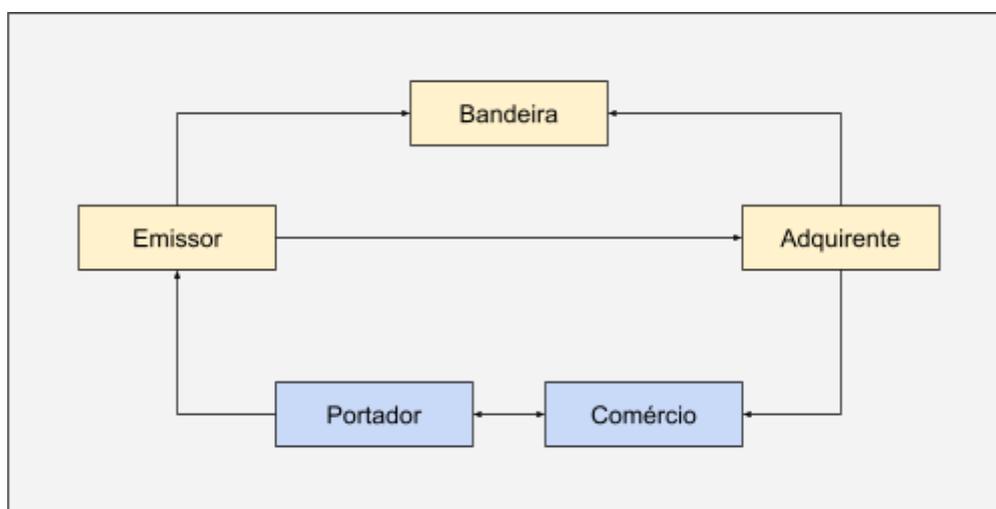
O mercado de pagamentos por cartão, tanto de crédito quanto de débito, possui relevância em relação às formas de pagamento no país, uma vez que segundo SANTOS, em 2019, foi responsável por movimentar mais de R\$1,8 trilhão. Ainda que o pagamento por PIX venha ganhando cada vez mais adesão, o volume de transações deste tipo ainda é muito relevante. Nesse mercado, existem diferentes atores envolvidos ao longo da cadeia.

- **Portador:** pessoa titular de um cartão de pagamento;

- **Emissor:** instituição financeira que emite o cartão de pagamentos a um portador, essa instituição possui relacionamento com a Bandeira, e assume as responsabilidades de crédito do portador junto aos demais atores da cadeia de pagamentos;
- **Cartão de Pagamento:** cartão gerado por um emissor para um determinado portador. Possui as seguintes modalidades: Crédito, Débito e Pré-Pago.
 - **Crédito:** nesse tipo de cartão, o emissor financia o pagamento para o portador, pagando ao adquirente os valores da transação, sendo posteriormente reembolsado pelo portador.
 - **Débito:** nesse tipo de cartão, o pagamento é realizado utilizando valores que o portador possui em conta, sem necessidade de financiamento por parte do emissor.
 - **Pré-pago:** são cartões que podem ser recarregados, funcionam de forma semelhante aos cartões de débito, porém existem benefícios para sua utilização. Exemplos: cartão alimentação, cartão refeição, vale combustível, etc
- **Bandeira:** define regras e normativas para o funcionamento do sistema, funciona como ponto de conexão entre emissores e adquirentes. As bandeiras mais conhecidas são: Visa, Mastercard e American Express.
- **Adquirente:** também conhecido como credenciadora, credencia estabelecimentos comerciais para aceitar pagamentos por cartão e realiza o intermédio das operações de pagamento junto à bandeira.
- **Comércio:** estabelecimento comercial, credenciado em uma Bandeira, por intermédio de um Adquirente, podendo realizar a cobrança de um portador.

Esses atores, se comunicam ao longo do fluxo de transações com cartões de pagamento, conforme ilustrado na figura 1.

Figura 1 - Fluxo de transações com cartões de pagamento

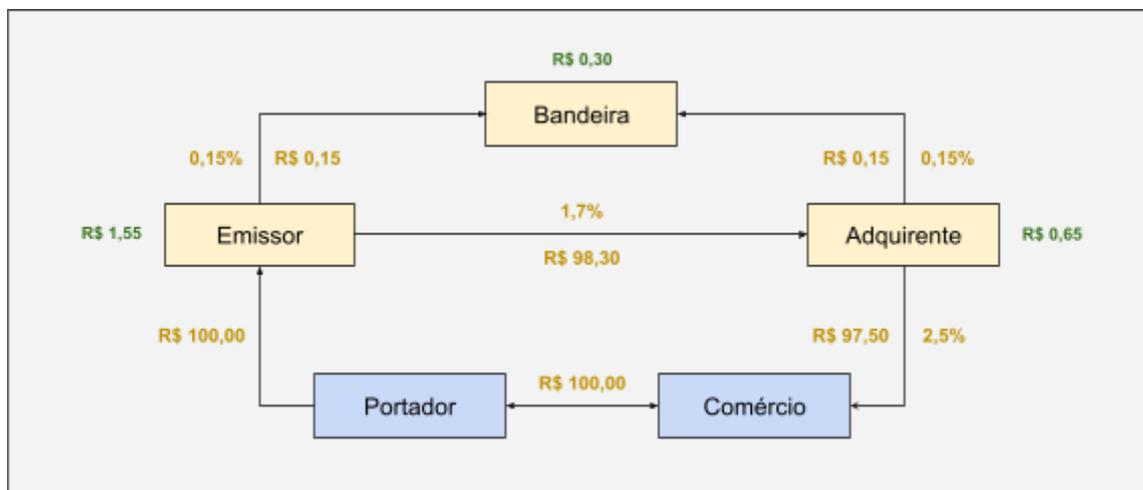


No fluxo de transação, um Comércio realiza a cobrança de um Portador. Tanto o Comércio, quanto o Portador possuem cadastro junto a uma Bandeira, para o Portador, seu cadastro ocorre por intermédio de um Emissor, e para um Comércio, seu cadastro ocorre por intermédio de um Adquirente. Emissores e Adquirentes trocam mensagens utilizando a norma ISO 8583 no momento da transação por intermédio da Bandeira, posteriormente Emissores e Adquirentes realizam trocas financeiras entre si, e sinalizam a bandeira sobre as trocas realizadas.

A remuneração referente a transação é dada para os atores envolvidos no processamento: Emissor, Bandeira e Adquirente. O Emissor cobra uma taxa de intercâmbio do Adquirente, essa taxa pode variar de acordo com o cartão, já um Adquirente cobra uma taxa do Comércio para processar a transação, essa taxa é conhecida como MDR - *Merchant Discount Rate*, podendo variar de acordo com características da transação - tipo de operação, canal de venda, entre outros. A Bandeira cobra uma taxa por transação tanto do Emissor quanto do Adquirente. Em uma transação, o faturamento do Adquirente é dado pela diferença entre a taxa cobrada do comércio, pela taxa cobrada pelo Emissor.

A Figura 2 demonstra um exemplo com taxas fictícias, referente a uma compra de R\$ 100,00. O Comércio realiza uma cobrança de R\$ 100,00 do Portador, que por sua vez irá pagar esse mesmo valor ao Emissor.

Figura 2 - Exemplo de taxas e valores no fluxo de transações com cartões de pagamento



Fonte: Adaptado de Payments 4.0 (2020)

O Emissor, possui uma taxa de intercâmbio de 1,7%, com isso, ele irá descontar R\$ 1,70 da transação, e pagar ao Adquirente R\$ 98,30. O Emissor também precisa pagar uma taxa para a Bandeira, essa taxa é 0,15%. Para essa transação, o Emissor recebeu uma remuneração final de R\$ 1,55.

O Adquirente cobra do Comércio uma taxa de 2,5% para essa transação, logo, nesse processo, será descontado R\$ 2,50, e o Comércio irá receber do Adquirente R\$ 97,50.

O Adquirente recebe do Emissor R\$ 98,30, referente ao valor da transação descontado da taxa de intercâmbio (1,7%), ao descontarmos a taxa de intercâmbio, da taxa cobrada do Comércio (MDR), o Adquirente recebeu R\$ 0,80. Porém, da mesma forma que o Emissor, o Adquirente também precisa pagar uma taxa para a Bandeira, essa taxa é 0,15%. No caso dessa transação, o Adquirente recebeu uma remuneração final de R\$ 0,65.

3 Especificação ISO 8583

A ISO 8583 foi estabelecida inicialmente em 1987, e atualizada posteriormente em 1997 e 2003. Ainda que pudesse ser desejável que todos adotassem a versão mais atual, ainda coexistem diferentes versões e variações em cada uma dessas versões.

Uma mensagem seguindo a especificação ISO 8583, é formada pelos seguintes elementos:

- MTI - Message Type Indicator
- Bitmap - mapa de bits
- Data Elements

Esses três elementos são considerados parte da estrutura básica. Assim, a dificuldade para quem vai criar soluções utilizando este formato, está relacionada às variações entre as implementações, pois ainda que a estrutura básica seja mantida, dentro desses elementos existem variações, que variam de acordo com sistemas autorizadores e bandeiras que têm suas próprias implementações.

A Figura 3 apresenta um exemplo de mensagem no formato ISO 8583.

Figura 3 - Exemplo mapa de bits

```
0200721A040128C0880A1645396206599220970120000000005300001129150358003758112911299000
0000000999374539620659922097=1111111111111111111100000012345600000001000000000000010
3986000000000000000400300800812300000
```

Fonte: Autoria Própria, 2024.

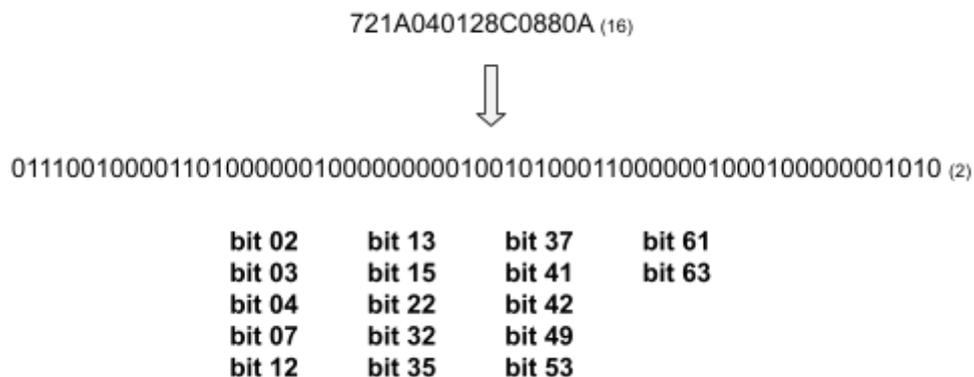
3.1 Message Type Indicator

O *MTI (Message Type Indicator)* é composto por um número de 4 dígitos, que traz informações sobre a mensagem, como versão da ISO utilizada, classificação da mensagem e origem.

3.2 Bitmap

O bitmap é composto por 16 ou 32 dígitos hexadecimais, que indicam os *Data Elements* presentes na mensagem. Ao transformar os dados de hexadecimal para binário, as posições com valor '1' indicam a presença do *Data Element* na mensagem. A Figura 4 apresenta um exemplo de bitmap, conforme a mensagem de exemplo apresentada na Figura 3.

Figura 4 - Exemplo mapa de bits



Fonte: Autoria Própria, 2024.

3.3 Data Elements

Os Data Elements, também chamados de 'bits', são os campos da mensagem de pagamento, são apresentados de forma serializada e possuem diferentes formatos e tamanho fixo ou variável, campos variáveis são precedidos por um valor indicando o tamanho do conteúdo.

Ainda que a estrutura básica seja mantida, dentro desses campos existem variações, que variam de acordo com sistemas autorizadores e bandeiras que têm suas próprias implementações.

A Figura 5 apresenta um exemplo de campos em uma mensagem ISO 8583.

Figura 5 - Exemplo de campos em mensagem ISO 8583

16453962065992209701200000000053000011291503580037581129112990000000000999374539620659922097=11111111111111111111000000123456000000100000000000001039860000000000000040300800812300000

bit 2 = 4539620659922097
 bit 3 = 012000
 bit 4 = 000000530000
 bit 7 = 1129150358
 bit 12 = 003758
 bit 13 = 1129
 bit 15 = 1129
 bit 22 = 900
 bit 32 = 00000000999
 bit 35 = 4539620659922097=11111111111111111111
 bit 37 = 000000123456
 bit 41 = 00000001
 bit 42 = 0000000000000001
 bit 49 = 986
 bit 53 = 0000000000000004
 bit 61 = 008
 bit 63 = 12300000

Fonte: Autoria Própria, 2024.

4 Testes

Testes de software são essenciais para o processo de desenvolvimento, eles garantem que o funcionamento do software obedeça aos requisitos esperados. Segundo Pressman, testes no contexto de software, são um conjunto de atividades que podem ser planejadas, empregando técnicas de cenário e caso de teste.

Existem diferentes tipos de teste e conceitos envolvidos, como testes unitários, que validam pequenas partes do código, testes de integração, que validam a integração entre diferentes partes do software, e teste regressivo, que tem como objetivo verificar se novas alterações não introduziram falhas em funcionalidades já existentes. Esses testes podem ser realizados de forma manual ou automatizada.

No processo de testes, é comum o uso de cenário de teste e casos de teste para elaboração de testes.

- **Cenário de Teste:** descreve uma situação a ser testada
- **Caso de Teste:** descreve com maiores detalhes a situação a ser testada, observando entradas e resultados esperados.

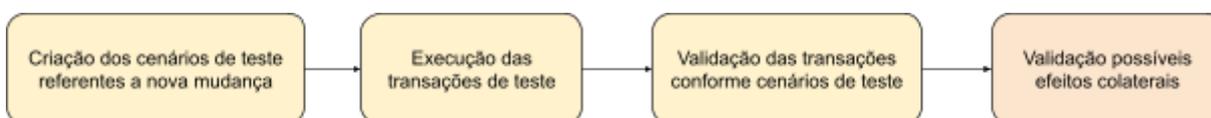
Esses conceitos são essenciais para garantir a abrangência dos testes.

4 DESENVOLVIMENTO

Para o desenvolvimento do trabalho, foram consultados especialistas, que atuam no mercado de meios de pagamento utilizando a norma ISO 8583, sobre os processos de desenvolvimento e testes em mensagens envolvendo a norma.

Foi analisado o processo de teste conforme apresentado na Figura 6, onde inicialmente, são definidos os casos de teste, seguido da execução de transações de teste, e posteriormente são realizadas validações manuais relacionadas a mudança desenvolvida, e ao final, é realizada uma validação manual de eventuais efeitos colaterais que um desenvolvimento pode ter gerado para outras mensagens ISO.

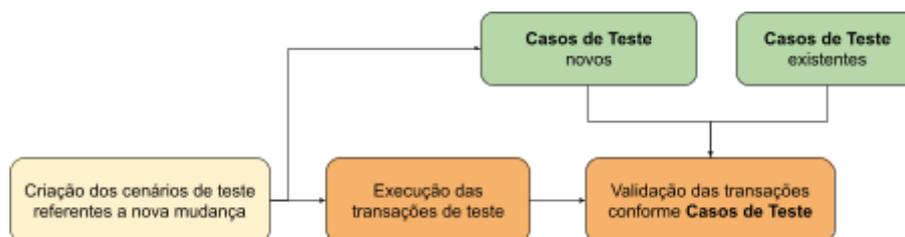
Figura 6 - Fluxo de testes atual



Fonte: Autoria Própria, 2024.

Observando o fluxo de testes atual, foi possível perceber um esforço repetitivo para validação de eventuais efeitos colaterais, que ocorre sem uma análise muito profunda. Com isso, foi planejado que, com a ferramenta proposta no presente trabalho, deveria auxiliar na validação das mensagens novas e auxiliar no processo de validação das mensagens já existentes, diminuindo assim o esforço manual nos processos de teste e validação.

Figura 7 - Fluxo de teste proposto



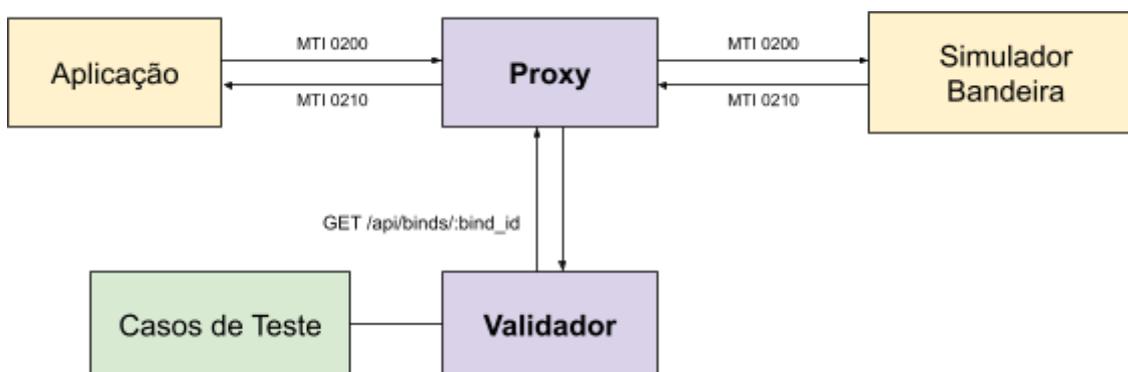
Fonte: Autoria Própria, 2024.

Após a elaboração dos casos de teste, a validação das transações utiliza como base tanto os casos de teste novos quanto os já existentes, expandindo os testes em relação à mudança desenvolvida de forma mais ampla, sem uma necessidade de aumento de força manual.

Para a parte de programação, foi escolhida a linguagem JavaScript e o interpretador NodeJS, por existir uma maior familiaridade de conhecimentos.

No desenvolvimento do trabalho, foi idealizado um protótipo que atua de forma generalista, sem acoplamento com tecnologias e/ou ferramentas específicas para sua utilização, desenvolvendo sua arquitetura conforme representado na Figura 8.

Figura 8 - Arquitetura da aplicação



Fonte: Autoria Própria, 2024.

O Proxy é responsável por receber e repassar as mensagens entre a aplicação e o simulador de bandeira, armazenando de forma temporária a mensagem ISO 8583 de pedido e resposta. Essa mensagem armazenada, é recuperada pelo elemento Validador, responsável por executar os casos de teste escritos.

Para execução do Proxy, é necessário executar o programa enviando como argumento, o arquivo de configuração em formato YAML, conforme apresentado na Figura 9.

Figura 9 - Exemplo de execução do elemento proxy

```
▼ TERMINAL
○ → its-proxy git:(main) x ./iso-proxy.js example-config.yaml
[proxy_port_8000] - Listening on 8000
[proxy_port_8002] - Listening on 8002
[API] - API running on port 3000
□
```

Fonte: Autoria Própria, 2024.

É possível definir diferentes configurações para execução do proxy, permitindo que uma mesma instância em execução, receba e envie mensagens de diferentes origens para diferentes destinos. A Figura 10 apresenta um exemplo do arquivo de configuração esperado pelo Proxy em formato YAML.

Figura 10 - Exemplo de arquivo de configuração

```
1  api_port: 3000
2  binds:
3    - id: 'proxy_port_8000'      # the config name
4      description:             # the config description
5        'Test payment switch A'
6      version: 'iso_87'        # the iso version - 87 => ISO 8583 1987
7      message_id:              # define the key to retrieve message -
8        - 4
9        - 37
10     port: 8000                # port to client consume
11     to:
12       host: 'localhost'      # destination host
13       port: 8010             # destination port
14
15     - id: 'proxy_port_8002'
16       description:
17         'Test payment swich B'
18       version: 'iso_87'
19       port: 8002
20       to:
21         host: 'localhost'
22         port: 8020
23
```

Fonte: Autoria Própria, 2024.

Os casos de teste são escritos em arquivo JSON com estrutura específica, no qual o Validator realiza a leitura e aplica as regras de validação.

Figura 13 - Exemplo de arquivo de caso de teste

```

() example_test_case.iso_test.json M x
iso-validator > {} example_test_case.iso_test.json > title
You, 1 second ago | 1 author (You)
1
2 "title": "Example Test Case",
3 "description": "test case example",
4 "message_id": "0000003000000000123456",
5 "request": [
6   {
7     "field": "0",
8     "description": "mti equal '0200'",
9     "expect": {
10      "condition": "equal",
11      "value": "0200"
12    }
13  },
14  {
15    "field": "4",
16    "description": "amount equal '00000123456'",
17    "expect": {
18      "condition": "equal",
19      "value": "00000123456"
20    }
21  },
22  {
23    "field": "35",
24    "description": "track 1 valid",
25    "expect": {
26      "condition": "regex",
27      "value": "%B\\d{0,19}\\^[\\w\\s\\|]{2,26}\\^\\d{7}\\w*\\?$"
28    }
29  }
30 ],
31 "response": [
32   {
33     "field": "0",
34     "description": "mti equal '0210'",
35     "expect": {
36       "condition": "equal",
37       "value": "0210"
38     }
39   }
40 ]
41

```

Fonte: Autoria Própria, 2024.

Para execução do Validador, é necessário executar o programa enviando como argumentos: a configuração esperada, o endereço do Proxy, e os arquivos de casos de teste. A Figura 9 apresenta um exemplo de execução do processo de validação.

Figura 14 - Exemplo execução da validação de casos de uso pelo Validador

```

TERMINAL
└─ iso-validator git:(main) x ./iso-validator proxy_port_8000 http://localhost:3000 "*.iso_test.json"
  /api/binds/proxy_port_8000/messages/0000000300000000000123456
  File "example_test_case.iso_test.json"
  Example Test Case - Message ID: "00000003000000000123456"

Request:
  ✓ Field 0 - SUCCESS
  x Field 4 - FAIL
    - expect value "000000030000" be equal to "00000123456"
  x Field 35 - FAIL
    - expect value "1234567891234567=111111111111111111" matches to pattern "%B\d{0,19}\^[\\w/s/]{2,26}\^d{7}\w*\?$"

Response:
  x Field 0 - FAIL
    - expect value "0200" be equal to "0210"

/Users/muchsousa/Documents/La Salle/TCC/iso-validator/lib/validator.js:183
  throw new Error('Test cases failed')
  ^
Error: Test cases failed
    at validate (/Users/muchsousa/Documents/La Salle/TCC/iso-validator/lib/validator.js:183:15)
    at processTicksAndRejections (node:internal/process/task_queues:96:5)
└─ iso-validator git:(main) x █

```

Fonte: Autoria Própria, 2024.

No processo de validação, são comparados os valores encontrados no Proxy com as regras definidas no caso de teste, caso o valor encontrado seja divergente da regra de validação são apresentadas as diferenças encontradas.

5 Discussão e Resultados

Para critério de avaliação, foi apresentado o protótipo desenvolvido para dois especialistas em TI, nos cargos de Arquiteto de Soluções e Product Owner, com experiência de 10 e 12 anos respectivamente, ambos atuam no mercado de meios de pagamento utilizando a norma ISO 8583, há mais de 4 anos. Na apresentação, foi demonstrado para ambos o protótipo em um exemplo prático de funcionamento, ao fim da apresentação, foi realizado um questionamento sobre a aplicabilidade e serventia do protótipo proposto.

Ambos concordaram sobre a aplicabilidade e serventia do protótipo apresentado, afirmando que ajudaria os processos de teste no fluxo de desenvolvimento, melhorando os critérios de qualidade. Como considerações, ambos especialistas sugeriram o desenvolvimento de uma interface gráfica para criação dos casos de teste, permitindo criar o cenário de teste e validar o teste na própria interface gráfica, essa sugestão será avaliada como trabalho futuro.

6 Conclusão

Neste trabalho de conclusão de curso, foi buscado aliar conceitos e técnicas trabalhadas ao longo do curso, com tecnologias e demandas experimentadas no mercado de trabalho. A definição do problema se deu a partir da observação de rotinas de trabalho, onde percebeu-se uma oportunidade de propor melhorias por meio da automatização de testes. Ao final do trabalho, podemos concluir que os objetivos foram atendidos, uma vez que obteve-se um protótipo operacional capaz de auxiliar no processo de desenvolvimento e testes de soluções para pagamento com cartões a partir da norma ISO 8583.

Como trabalhos futuros, planeja-se implementar uma ferramenta gráfica para facilitar a criação de casos de teste, de forma a permitir a criação do cenário de teste, e sua validação na própria interface gráfica, seguindo as considerações feitas pelos especialistas que avaliaram a solução proposta.

REFERÊNCIAS

SANTOS, Edson Luiz dos; CAVALCANTI, Luis Filipe. **PAYMENTS 4.0: as forças que estão transformando o mercado brasileiro**. São Paulo: Linotipo Digital, 2020. ISBN 978-65-990263-3-1

ROEHRS, Alex. **4iPay: Modelo para Sistemas de Pagamento Móvel em Comércio Ubíquo**. 2012. Dissertação (Mestrado) - Programa Interdisciplinar de Pós-Graduação, Universidade do Vale do Rio dos Sinos, São Leopoldo, 2012. Disponível em: <https://repositorio.jesuita.org.br/bitstream/handle/UNISINOS/3140/4iPay.pdf>

IBM. **Padrão do sistema de mensagens ISO8583**. Disponível em: <https://www.ibm.com/docs/pt-br/integration-bus/10.0?topic=formats-iso8583-messaging-standard>

DOCK. **Autorização externa - SOCKET TCP ISO8583**. Disponível em: <https://lighthouse.dock.tech/docs/hybrid-platform/cards-and-credit/processing-cards/authorization/socket-tcp-iso-8583>

IR. **An Introduction to ISO 8583: What You Need To Know**. Disponível em: <https://www.ir.com/guides/introduction-to-iso-8583>

MERWE, Arthur Van Der. **Parsing AS2505/8583 Messages**. Disponível em: <https://arthurvandermerwe.com/2014/07/06/parsing-as25058583-messages/>

PRESSMAN, Roger S. **Engenharia de Software Uma Abordagem Profissional**. 9. ed. São Paulo: McGraw-Hill, 2011.

SPIRLANDELI, Cleber; ROLAND, Carlos Eduardo de França. **A utilização de testes automatizados no desenvolvimento de software** *Revista*. EduFatec: educação, tecnologia e gestão, Franca, v.2, n.2, p.1-24, ago./dez. 2019. Disponível em: <https://revistaedufatec.fatecfranca.edu.br/wp-content/uploads/2020/03/edufatec-n02v2a01.pdf>

REITZ, Lindomar Peixinho; FERNANDES, Anita Maria da Rocha. **Uma Comparação entre Testes Manuais e Testes Automatizados para Garantia da Qualidade em Softwares para Dispositivos Móveis**. *Revista Eletrônica Argentina-Brasil de Tecnologias da Informação e da Comunicação*, v. 1. Disponível em: https://www.researchgate.net/profile/Lindomar-Reitz/publication/314236449_Uma_Comparacao_entre_Testes_Manuais_e_Testes_Automatizados_para_Garantia_da_Qualidade_em_Softwares_para_Dispositivos_Moveis/links/58ea4338458515e30dcb4a1/Uma-Comparacao-entre-Testes-Manuais-e-Testes-Automatizados-para-Garantia-da-Qualidade-em-Softwares-para-Dispositivos-Moveis.pdf