



UNIVERSIDADE EDUARDO MONDLANE
FACULDADE DE ENGENHARIA
CURSO DE LICENCIATURA EM ENGENHARIA INFORMÁTICA

**IMPLEMENTAÇÃO DE PROCESSAMENTO ASSÍNCRONO EM TRANSACÇÕES
BANCÁRIAS PARA MELHORAR A DISPONIBILIDADE DO SERVIÇO DE INTERNET
BANKING**

Caso de Estudo: Sistema de Internet Banking do banco Nedbank Moçambique

Autor:

Novela, Manuel João de Matos

Supervisores:

Eng^o. Rúben Manhiça

Enga. Leila Omar

Supervisor da Instituição:

dr. Paulo Jorge Braga Zacarias

Maputo, Novembro de 2023



UNIVERSIDADE
E D U A R D O
M O N D L A N E

UNIVERSIDADE EDUARDO MONDLANE
FACULDADE DE ENGENHARIA
CURSO DE LICENCIATURA EM ENGENHARIA INFORMÁTICA

**IMPLEMENTAÇÃO DE PROCESSAMENTO ASSÍNCRONO EM TRANSACÇÕES
BANCÁRIAS PARA MELHORAR A DISPONIBILIDADE DO SERVIÇO DE INTERNET
BANKING**

Caso de Estudo: Sistema de Internet Banking do banco Nedbank Moçambique

Autor:

Novela, Manuel João de Matos

Supervisores:

Engº. Rúben Manhiça

Enga. Leila Omar

Supervisor da Instituição:

dr. Paulo Jorge Braga Zacarias

Maputo, Novembro de 2023



**UNIVERSIDADE EDUARDO MONDLANE
FACULDADE DE ENGENHARIA
CURSO DE ENGENHARIA INFORMÁTICA**

TERMO DE ENTREGA DO RELATÓRIO DE ESTÁGIO PROFISSIONAL

Declaro que o estudante **Manuel João de Matos Novela** entregou no dia ___/___/_____,
___ cópias do relatório do seu Relatório de Estágio Profissional com a referência _____,
intitulado: Utilização de Processamento Assíncrono em Transacções Bancarias para
melhoria da disponibilidade do Serviço de Internet Banking.

Maputo, ___de _____ de 2023

O Chefe da Secretaria



**UNIVERSIDADE EDUARDO MONDLANE
FACULDADE DE ENGENHARIA
CURSO DE ENGENHARIA INFORMÁTICA**

DECLARAÇÃO DE HONRA

Declaro sob compromisso de honra que o presente trabalho é resultado da minha investigação e que foi concebido para ser submetido na cadeira Projecto Integrado de Aplicativos na Faculdade de Engenharia da Universidade Eduardo Mondlane.

Maputo, __de _____ de 2023

O Autor

(Manuel João de Matos Novela)

Dedicatória

Ao meu pai, João Alfredo Novela
A minha mãe, Arminda Alves de Matos Novela

Agradecimentos

Em primeiro lugar, agradeço a Deus pela vida, oportunidade de realização dos meus sonhos em toda jornada de vida e por sempre me guiar em todos momentos da minha vida.

Agradeço aos meus pais pelos esforços que têm realizado para que não me faltasse nada, por me apoiarem nas minhas decisões de vida a dar conselhos sempre que possível e por terem me educado da melhor forma possível para que me tornasse na pessoa que sou hoje.

Agradeço a todos os professores que tiveram paciência e disponibilidade para me orientar e me ajudar a entender os conteúdos da minha área de estudo. Também gostaria de agradecer aos meus colegas, que foram sempre muito solidários e dispostos a colaborar. E, claro, um agradecimento especial aos meus amigos e familiares, que estiveram sempre ao meu lado, me incentivar e apoiar em todos os momentos.

Este trabalho foi um grande desafio para mim, mas também uma oportunidade única de aprendizado e crescimento pessoal. Sinto-me muito grato por ter tido a oportunidade de realizá-lo e espero que ele possa contribuir, de alguma forma, para a comunidade científica e para o meu desenvolvimento profissional. Obrigado novamente a todos que contribuíram para que este trabalho fosse uma realidade.

Gostaria de agradecer especialmente aos meus amigos e familiares, que foram incondicionais em seu apoio durante todo o processo de realização deste trabalho. Eles estiveram sempre ao meu lado, prontos para me ouvir, me aconselhar e me incentivar. Sem dúvida, o seu apoio foi fundamental para que eu pudesse superar os obstáculos e dificuldades encontrados ao longo do caminho.

Agradeço também a todos os amigos e familiares que, de alguma forma, contribuíram para a concretização deste trabalho. Seja através de palavras de incentivo, de ajuda material ou simplesmente de um abraço quando mais precisava, toda ajuda foi muito valiosa e significativa para mim.

Sem a ajuda e o apoio de todos vocês, este trabalho não teria sido possível. Obrigado de coração por todo o amor e carinho que me foram dispensados. Eu sou muito grato por ter pessoas tão especiais em minha vida.

Epigrafe

"We can run it later or in background"

Paulo Zacarias

Resumo

No contexto do dinâmico ambiente financeiro actual, a disponibilidade é um aspecto crucial para o sucesso dos serviços bancários, especialmente no domínio do internet banking. Este trabalho propõe uma abordagem inovadora para melhorar a disponibilidade do sistema de internet banking, que foca na implementação da tecnologia de processamento assíncrono de transacções.

A metodologia adoptada envolveu uma análise aprofundada dos recursos disponíveis, com destaque para a revisão de literatura e interacção com os colaboradores do Nedbank Moçambique. Identificaram-se desafios significativos no sistema de internet banking, particularmente na área de disponibilidade, que serviram como ponto de partida para a elaboração de soluções.

O desenvolvimento do protótipo foi orientado pelo estudo dos tipos e formas de implementação do processamento assíncrono, com base no modelo de arquitectura apresentado no capítulo IV. Optou-se por uma abordagem de migração incremental para garantir baixo impacto nas operações normais e facilitar a integração com o sistema existente.

A implementação prática do processamento assíncrono permitiu a superação dos desafios identificados, que resulta em um sistema de internet banking mais resiliente e com maior disponibilidade. Ao final do desenvolvimento, todos os requisitos, incluindo os desejáveis, foram plenamente implementados, que culmina em um sistema totalmente funcional que já está em operação no Nedbank Moçambique.

Este estudo não apenas apresenta uma solução eficaz para melhorar a disponibilidade do sistema de internet banking, mas também destaca a relevância e a eficácia da tecnologia de processamento assíncrono de transacções nesse contexto. Os resultados obtidos não só atenderam aos objectivos estabelecidos, mas também ofereceram insights valiosos sobre a aplicabilidade prática dessa abordagem inovadora no sector bancário.

Palavras-chaves: Transacções, Disponibilidade, Arquitectura, Assíncrono.

Abstract

In the context of the current dynamic financial environment, availability is a crucial aspect for the success of banking services, especially in the realm of internet banking. This work proposes an innovative approach to enhance the availability of the internet banking system, focusing on the implementation of asynchronous transaction processing technology.

The adopted methodology involved a thorough analysis of available resources, with emphasis on literature review and interaction with collaborators from Nedbank Moçambique. Significant challenges in the internet banking system, particularly in the area of availability, were identified, serving as a starting point for the formulation of solutions.

The development of the prototype was guided by the study of types and forms of asynchronous processing implementation, based on the architecture model presented in Chapter IV. An incremental migration approach was chosen to ensure minimal impact on normal operations and facilitate integration with the existing system.

The practical implementation of asynchronous processing overcame the identified challenges, resulting in a more resilient internet banking system with increased availability. At the end of the development, all requirements, including desirable ones, were fully implemented, culminating in a fully functional system that is already in operation at Nedbank Moçambique.

This study not only presents an effective solution to enhance the availability of the internet banking system but also highlights the relevance and effectiveness of asynchronous transaction processing technology in this context. The achieved results not only met the established objectives but also provided valuable insights into the practical applicability of this innovative approach in the banking sector.

Keywords: Transactions, Availability, Architecture, Asynchronous.

Índice

1. Capítulo I – Introdução	1
1.1. Contextualização	1
1.2. Definição do Problema	3
1.3. Motivação	4
1.4. Objectivos.....	5
1.4.1. Geral	5
1.4.2. Especifico	5
1.5. Metodologia.....	6
1.6. Estrutura do Trabalho.....	7
2. Capítulo II – Actividades Desenvolvidas no estágio.....	9
2.1 Desenvolvimento de Software nos Canais Digitais	9
2.2 Monitoria da Disponibilidade do Sistema de Internet Banking.....	9
2.3 Identificação e Documentação dos Requisitos	10
2.4 Análise Crítica dos Requisitos	10
2.5 Exploração e Avaliação de Tecnologias:.....	11
2.6 Implementação da Tecnologia Seleccionada:	11
2.7 Design Arquitectónico e Codificação:.....	11
2.8 Testes e Testes de Carga:	12
2.9 Resolução de Problemas Emergentes:	12
3. Capítulo III – Revisão da literatura	12
3.1. Internet banking.....	12
3.1.1. Vantagens e desvantagens do Internet banking	13
3.2. Processamento de transacções	15
3.2.1. Tipos de processamento de transacções	15
3.3. Tipos de processamento assíncrono.....	16
3.3.1. Callbacks e Promessas	17

3.3.2.	Mensagens Assíncronas.....	19
3.3.3.	Filas de Mensagens.....	21
3.4.	Formas de implementação do processamento assíncrono	22
3.4.1.	Programação Assíncrona	22
3.4.2.	Processamento Assíncrono em Hardware.....	23
3.4.3.	Comunicação Assíncrona em Redes.....	23
3.5.	Ferramentas no Processamento Assíncrono	24
3.5.1.	Apache Kafka.....	25
3.5.2.	RabbitMQ.....	26
3.5.3.	Open Message Queue.....	28
4.	Capítulo IV – Caso de Estudo	29
4.1.	Descrição dos sistemas do banco	30
4.2.	Descrição do ecossistema dos aplicativos bancários.....	30
4.2.1.	Descrição do sistema Core.....	31
4.2.2.	Descrição do sistema Non-Core	31
4.2.3.	Organograma.....	31
4.3.	Situação actual.....	32
4.4.	Constrangimentos da situação actual.....	33
4.5.	Constrangimentos resolvidos com o modelo.....	34
4.6.	Proposta de solução.....	35
4.7.	Possíveis soluções para a melhoria da disponibilidade do sistema de IB	35
4.7.1.	Constrangimentos da implementação das soluções.....	37
4.8.	Arquitectura do modelo híbrido do sistema de internet banking.	37
4.1.1	Motivos da escolha do modelo.....	39
4.2	Descrição da solução	39
4.2.1	Parte síncrona.....	40
4.2.2	Parte assíncrona.....	40
4.8.2.	Processamento de transacções orientado a transacções críticas e não críticas	43
5.	Capítulo V – Implementação da solução de processamento assíncrono no sistema de internet banking	50

5.1	Implementação das filas de Mensagens.....	51
5.1.1	Criação de Filas Específicas.....	51
5.2	Implementação dos <i>producers</i> e dos <i>consumers</i>	52
5.2.1	Tratamento de Transacções Críticas e Não Críticas	55
5.3	Implementação com os sistemas legados.....	55
5.4	Monitoramento e controle da Solução.....	56
5.4.1	Validação de Requisitos de Negócios.....	58
5.5	Testes para comparação entre o desempenho da solução assíncrona com o síncrono	59
6.	Capítulo VI – Discussão de Resultados	62
6.1.	Revisão de literatura.....	62
6.2.	Caso de estudo	63
6.3.	Desenvolvimento da solução proposta.....	64
7.	Capítulo VII – Considerações finais	65
7.1.	Conclusão	65
7.2.	Recomendações.....	66
7.3.	Constrangimentos	67
	Bibliografia.....	67
	Referências Bibliográficas	67
	Outras bibliografias.....	70
	Anexos	1
	Anexo 1: Guião de entrevista	1
	Anexo 2: Especificação técnica do servidor do IB	1
	Anexo 3: Teste de carga do sistema de internet banking com processamento síncrono e assíncrono:.....	2
	Anexo 4: Mockups do sistema de internet banking	6
	Anexo 5: Caso de uso do sistema de internet banking síncrono	11
	Anexo 6: Caso de uso do sistema de internet banking assíncrono	12
	Anexo 7: Email de notificação da conclusão do processamento da requisição feita no sistema de internet banking.....	14

Anexo 8: Push Notification da conclusão do processamento da requisição feita no sistema de internet banking 16

 A8.1 Notificação no Android e IOS..... 16

 A8.2 Notificação no Windows e MacOS 17

Índice de figuras

Figura 1: Callbacks VS Promises	19
Figura 2: Arquitectura de Apache Kafka.....	26
Figura 3: Arquitectura geral do RabbitMQ	28
Figura 4: Arquitectura do OpenMQ.....	29
Figura 5: Exemplo da configuração da fila de múltiplas transacções para a Carteira móvel M-pesa.....	52
Figura 6: Produtor da fila de transacções do M-pesa	53
Figura 7: Consumidor da fila de transações M-pesa	54
Figura 8: teste de carga feito ao endpoint de transferência móvel no modelo síncrono	57
Figura 9: Perfil de testes de carga (ramp-up)	A2.2
Figura 10: Gráfico da relação entre o numero de requisições e respostas.....	A2.4
Figura 11: Gargalo no processamento Síncrono.....	A2.6
Figura 12: Tela de carregamento de ficheiro de pagamentos de salários	A2.7
Figura 13: Acção de submissão do ficheiro de pagamento de salários.....	A2.8
Figura 14: Resposta síncrona, a confirmar o registo do pedido de processamento de ficheiro de pagamento de salários.....	A2.9
Figura 15: Tabela com os estados dos pedidos, já contendo os pedidos que foram processados de forma assíncrona.....	A2.10
Figura 16: Caso de uso do sistema de internet banking síncrono.....	A2.11
Figura 17: Caso de uso do sistema de internet banking assíncrono.....	A2.13
Figura 18: Email de notificação da conclusão do processamento.....	A2.15
Figura 19: Push Notification da conclusão do processamento no Android.....	A2.16
Figura 20: Push Notification da conclusão do processamento no IOS.....	A2.16
Figura 21: Push Notification da conclusão do processamento no Windows.....	A2.17
Figura 22: Push Notification da conclusão do processamento no MACOS.....	A2.18

Índice de tabelas

Tabela 3: Problemas do processamento síncrono que são resolvidos pelo processamento assíncrono	41
--	----

Lista de abreviaturas e acrónimos

Async - Assíncrono

B2C – Business to customer

CPU – Central Processing Unit

GNU – General Public License

HTTP - *Hypertext Transfer Protocol* (Protocolo de Transferência de Hipertexto)

HTTPS - *Hypertext Transfer Protocol Secure* (Protocolo de Transferência de Hipertexto Seguro)

IB – *Internet Banking*

JSON - Notação de Objetos JavaScript

MRR – Mecanismo de recuperação e resiliência

REST - Representational State Transfer

SMS - Short Message Service (Serviço de Mensagens Curtas)

TPS – Third Party System

XML - Linguagem de Marcação Extensível

Glossário de termos

Business to customer – para (Kwok, Chen, & Lu, 2018), é um modelo de negócios no qual empresas vendem produtos ou serviços directamente aos consumidores finais.

Código Aberto – é o código-fonte que é disponibilizado gratuitamente para possível modificação e redistribuição. Os produtos incluem permissão para usar o código-fonte, documentos de design ou conteúdo do produto.

Waiting Time – Tempo de espera é o período de tempo em que um sistema fica inactivo ou ocioso enquanto aguarda que algum evento ou acção ocorra.

Middleware – é um software que fica entre um sistema operativo e os aplicativos executados nele.

Max attempts – significa o número máximo de tentativas permitido para realizar uma determinada operação. Se a operação falhar após o número máximo de tentativas, ela é considerada como tendo falhado completamente. Este limite é estabelecido para evitar looping infinito e tentativas de operações que não podem ser concluídas com sucesso.

Retry – é uma acção de tentar novamente uma operação que anteriormente falhou. Isso geralmente é feito automaticamente pelo sistema ou pode ser solicitado pelo usuário.

Saldo contabilístico de uma conta – representa o montante de activos associado a essa conta, e pode ser positivo ou negativo.

SOAP – é uma especificação de protocolo de mensagens para troca de informações estruturadas na implementação de serviços web em redes de computadores.

Third Party System – refere-se a um sistema ou aplicação que é desenvolvido e mantido por uma empresa diferente daquela que usa o sistema.

Thresholds – são limites predeterminados que sinalizam condições críticas em sistemas, as quais accionam acções específicas quando atingidos.

WebService – é uma solução utilizada na integração de sistemas e na comunicação entre aplicações diferentes.

1. Capítulo I – Introdução

O presente capítulo faz a introdução ao tema discutido neste relatório. Aqui, é exposto o contexto que motivou a concepção deste relatório, as razões que impulsionaram essa iniciativa, seguida de uma delimitação do problema que se pretende resolver, incluindo a definição dos objectivos que se pretende alcançar e os métodos com que se pretende perseguir estes objectivos.

1.1. Contextualização

Segundo Angelakopoulos e Mihiotis (2011), o Internet Banking, conhecido como banco online, é um serviço oferecido por instituições financeiras que permite aos clientes realizar diversas transacções bancárias por meio da internet. Essas transacções incluem consultas de saldo, transferências entre contas, pagamento de contas e solicitação de empréstimos, para proporcionar aos clientes uma forma conveniente e ágil de gerir suas finanças.

Desde o surgimento dos primeiros serviços bancários online com base na Internet, que foram fornecidos pela Stanford Federal Credit Union (SFCU) em Outubro 1994, a banca online espalhou-se rapidamente em todos os países do mundo, como resultado da sua conveniência e facilidade em realizar transacções de forma rápida (Yoon, 2010).

Quanto à importância e impacto do Internet Banking, conforme destacado por Sosyura (2016) e Gaurav e Kumar (2018), a modernização do sector bancário é evidente. A conveniência proporcionada aos clientes, que podem realizar transacções a qualquer momento e de qualquer lugar, elimina a necessidade de deslocamento físico até uma agência bancária. Além disso, contribui para a eficiência operacional, reduzindo custos associados à infra-estrutura física e processos manuais.

A urgência em garantir a disponibilidade do Internet Banking é destacada por Gray e Reuter (2001) e Câmara e Silva (2004), a necessidade de disponibilidade do Internet Banking é uma demanda premente na sociedade moderna. O Internet Banking desempenha um papel crucial na vida cotidiana dos clientes bancários, oferecendo conveniência e acessibilidade aos serviços financeiros. A importância da disponibilidade é evidente na capacidade dos

clientes de realizar transacções a qualquer momento e de qualquer lugar, para eliminar as restrições de horário e localização associadas às agências físicas.

No contexto específico do Nedbank Moçambique, uma instituição bancária proeminente no cenário financeiro do país, a ocorrência recorrente de casos de indisponibilidade dos serviços de Internet Banking representa um desafio significativo. Durante uma entrevista realizada com o autor, foi destacado que, em situações de alta demanda, a única alternativa para enfrentar problemas de indisponibilidade é reiniciar os serviços de Internet Banking. Essa abordagem, embora possa resolver momentaneamente os problemas técnicos, acarreta consequências directas e impactantes para os clientes e a reputação da instituição.

Conforme ressaltado por Samuel (2018) e Azar et al. (2014), a dependência crescente dos clientes em relação aos serviços online torna a indisponibilidade do Internet Banking uma questão crítica. Entre as consequências mais imediatas, destaca-se a insatisfação dos clientes devido à interrupção de serviços essenciais, como consultas de saldo, pagamentos de contas e transferências electrónicas.

A reputação da instituição financeira também é impactada negativamente, como indicado por Gaurav e Kumar (2018) em "The Role of Internet Banking in Financial Inclusion: Evidence from Developing Countries". A confiança dos clientes na segurança e eficiência dos serviços pode ser abalada, que resulta em uma potencial perda de clientela para instituições que não conseguem manter a disponibilidade consistente de seus serviços online.

As técnicas para abordar a indisponibilidade dos serviços de Internet Banking incluem a adopção de arquitecturas resilientes, conforme destacado por Schwaber (2017) e Coulouris, Dollimore, e Kindberg (2018). Estas arquitecturas são projectadas para garantir a continuidade do serviço mesmo diante de falhas em componentes específicos, para proporcionar uma maior robustez e resistência a eventos inesperados.

Ao adoptar o processamento assíncrono, as instituições financeiras, como destacado por Raza (2018), conseguem mitigar os riscos de falhas no sistema, uma vez que as

transacções podem ser processadas de forma independente e eficiente. Essa técnica também permite a execução paralela de tarefas, para otimizar o desempenho do sistema, mesmo em condições de alta demanda.

Nesse contexto, é necessário adoptar uma abordagem proactiva para melhorar a disponibilidade do sistema de internet banking do banco Nedbank Moçambique, é aqui que entra em cena o objectivo central deste trabalho de pesquisa. Propor um modelo de processamento de transacções no internet banking que minimize os problemas de disponibilidade do sistema de internet banking. Essa iniciativa busca não apenas otimizar a eficiência operacional, permitindo o processamento simultâneo e independente de transacções, mas também fortalecer a resiliência do sistema, para assegurar sua continuidade mesmo diante de eventos inesperados. Além disso, o modelo proposto visa proporcionar uma maior disponibilidade e confiabilidade aos usuários, contribuindo para a preservação da integridade das transacções.

1.2. Definição do Problema

Com regularidade, a equipa do Banco Nedbank Moçambique depara-se com queixas recorrentes por parte dos clientes que enfrentam dificuldades ao tentar aceder ao sistema de Internet Banking. Estas reclamações tornam-se mais evidentes em momentos de elevada procura, quando a pressão sobre o sistema atinge níveis críticos. Em algumas circunstâncias, a necessidade de lidar com um substancial aumento no tráfego de utilizadores torna-se tão premente que a única solução viável é proceder ao reinício dos serviços de Internet Banking. Esta prática de reinicialização, embora possa restaurar temporariamente a funcionalidade do sistema, representa uma medida reactiva e momentânea, frequentemente que resulta em incómodos para os clientes e que prejudica a experiência do utilizador. A recorrência destas interrupções não apenas compromete a satisfação dos clientes, mas também sublinha a necessidade urgente de implementar uma abordagem mais robusta, como o processamento assíncrono de transacções, para assegurar uma disponibilidade consistente e proactiva dos serviços de Internet Banking do Banco Nedbank Moçambique.

Consoante indicado por especialistas nos canais digitais do Banco Nedbank Moçambique, a ausência de um mecanismo eficaz para melhorar a disponibilidade do sistema de Internet Banking constitui uma lacuna significativa na infra-estrutura, repercutindo directamente na experiência dos clientes. Os utilizadores frequentemente deparam-se com a frustração de períodos de inactividade, obstáculos no acesso às suas contas e possíveis interrupções em transacções financeiras. Adicionalmente, a instituição bancária pode ser compelida a lidar com reembolsos resultantes de transacções não autorizadas com outras entidades bancárias, que acarreta em custos substanciais e prejudica a sua estabilidade financeira. Estes acontecimentos comprometem a confiança dos clientes no Internet Banking do Banco Nedbank Moçambique, tendo um impacto adverso na reputação da instituição e na sua posição competitiva no mercado.

No contexto operacional, é crucial notar que existe a possibilidade de uma transacção ser concluída com sucesso nos sistemas de terceiros. Contudo, em situações de *timeout* nas interacções com outros serviços bancários, o Banco Nedbank Moçambique pode não receber uma resposta desse processamento. Essa eventualidade pode acarretar consequências significativas, como transacções não autorizadas e a potencial ocorrência de descobertos não autorizados nas contas dos clientes. Este cenário ressalta a importância crítica de um modelo de processamento assíncrono de transacções, que não apenas optimize a disponibilidade do sistema, mas também assegure uma gestão eficiente e segura das transacções, que evitar assim situações desfavoráveis para os utilizadores.

Neste contexto, o enfoque deste trabalho centra-se em propor e implementar um modelo de processamento assíncrono de transacções no Internet Banking, com o objectivo de minimizar as incidências de problemas de disponibilidade e fortalecer a eficiência operacional do sistema.

1.3. Motivação

No âmbito da melhoria da disponibilidade do sistema de Internet Banking, a relevância desta iniciativa é enfatizada por diversos autores e especialistas na área. Segundo Schwaber (2017), a disponibilidade é um pilar fundamental para a satisfação dos clientes, pois impacta

directamente na experiência do utilizador. Ainda neste contexto, Coulouris, Dollimore e Kindberg (2018) destacam que uma infra-estrutura resiliente e de alta disponibilidade é crucial para garantir a continuidade dos serviços, especialmente em sectores sensíveis, como o bancário.

O aumento da disponibilidade no Internet Banking também é associado a benefícios financeiros e à reputação da instituição. Sosyura (2016) argumenta que a indisponibilidade dos serviços pode resultar em perdas financeiras significativas e comprometer a confiança dos clientes. Além disso, autores como Goldenberg (1997) ressaltam que a confiabilidade do sistema bancário, incluindo a disponibilidade dos serviços online, desempenha um papel vital na manutenção da fidelidade do cliente.

No cenário global, a importância do Internet Banking na inclusão financeira é evidenciada por estudos como o de Gaurav e Kumar (2018), enquanto Sosyura (2016) destaca o impacto positivo da transformação digital no setor bancário. Em um contexto mais amplo, a disponibilidade e segurança dos serviços bancários online são factores essenciais para o sucesso e a confiança dos clientes, como observado por Gray e Reuter (2001) e Coulouris, Dollimore e Kindberg (2018).

Neste contexto, a implementação de um modelo de processamento assíncrono de transacções no Internet Banking do Banco Nedbank Moçambique não apenas visa superar desafios específicos de disponibilidade, mas também se alinha com tendências globais que buscam aprimorar a eficiência, segurança e confiança nos serviços bancários digitais.

1.4. Objectivos

1.4.1. Geral

Propor um modelo de processamento de transacções no internet banking que minimize os problemas de disponibilidade do sistema de internet banking.

1.4.2. Especifico

- Estudar a situação actual do problema de disponibilidade
- Estudo sobre processamento assíncrono, seus tipos e formas de implementação
- Identificar e comparar as ferramentas usadas no processamento assíncrono

- Implementar e testar o processamento assíncrono no sistema de internet banking, com base nos tipos e formas de implementação identificados e comparar o seu desempenho com o sistema síncrono actualmente.

1.5. Metodologia

Este estudo adoptará uma abordagem de pesquisa mista (quantitativa e qualitativa), que permitirá uma análise robusta e multidimensional do problema de pesquisa. Segundo Creswell & Clark (2007), a pesquisa mista permite que o pesquisador compreenda melhor o problema ao explorar diferentes métodos.

Para atingir o primeiro objectivo, foram conduzidas entrevistas com os especialistas nos canais digitais do banco, que constituíram uma fonte de informação rica e contextualizada sobre os desafios de disponibilidade enfrentados. Segundo Lakatos e Marconi (2003), as entrevistas são uma ferramenta valiosa para recolher dados qualitativos, que possibilita uma compreensão aprofundada dos fenómenos estudados. Além disso, foram retiradas da base de dados informações acerca de casos de indisponibilidade ocorridos entre 2021 e 2023. Ao cruzar essas informações com os documentos e relatórios de reconciliação de contas, foi possível identificar casos de indisponibilidade parcial dos sistemas de terceiros que resultaram em transacções indevidas. Conforme Lakatos e Marconi (2003), a pesquisa documental é fundamental para a recolha de dados secundários provenientes de várias fontes, como relatórios institucionais, políticas internas, manuais de procedimentos e documentos relacionados ao caso de estudo.

Para alcançar o segundo objectivo, adoptou-se uma abordagem de Pesquisa Exploratória, fundamentada na revisão sistemática da literatura e em consultas a especialistas, o que possibilitou uma compreensão mais aprofundada sobre o processamento assíncrono de transacções, incluindo os seus tipos e formas. Conforme Stebbins (2001), a pesquisa exploratória revela-se a abordagem mais apropriada quando existe escasso conhecimento prévio sobre o problema, permitindo a descoberta de novas perspectivas e ideias. Essa estratégia de pesquisa permitiu uma imersão detalhada no tema, para explorar as nuances do processamento assíncrono e que captura insights valiosos de fontes diversificadas. A

revisão sistemática da literatura contribuiu para consolidar o conhecimento existente, enquanto as consultas a especialistas acrescentaram perspectivas práticas e experiências do campo. Essa abordagem integrada não apenas enriqueceu a compreensão do processamento assíncrono, mas também estabeleceu uma base sólida para a análise crítica das implicações práticas e teóricas associadas a esse modo de processamento de transacções.

Para atingir o terceiro objectivo, foi adoptada uma abordagem de pesquisa comparativa, para analisar e comparar ferramentas utilizadas no processamento assíncrono. Essa metodologia incluiu a revisão sistemática da literatura e consultas a especialistas para identificar e avaliar ferramentas pertinentes e eficazes nesse contexto. A pesquisa comparativa, conforme orientações de Stebbins (2001), é particularmente útil quando há pouco conhecimento prévio sobre o tema, permitindo uma compreensão mais profunda das características distintivas de cada ferramenta.

A pesquisa documental desempenhou um papel crucial, abrangendo manuais técnicos, documentação de ferramentas e relatórios de implementação. Essa abordagem proporcionou uma base sólida para a identificação de informações relevantes sobre as características e funcionalidades específicas de cada ferramenta, para fundamentar a comparação e a avaliação crítica.

Finalmente, para o quarto objectivo, um plano de implementação do processamento assíncrono foi desenvolvido, para integrar as melhores práticas identificadas durante a pesquisa. Testes completos foram conduzidos para garantir a eficácia do novo sistema. Simultaneamente, um teste comparativo de desempenho entre o sistema assíncrono implementado e o antigo sistema síncrono foi realizado, para avaliar métricas cruciais. Após a análise dos resultados, a configuração de filas mais eficaz foi identificada, e um plano de acompanhamento foi estabelecido para avaliação contínua, que alinham à abordagem de pesquisa aplicada proposta por Hakim (2000).

1.6. Estrutura do Trabalho

O presente trabalho é composto por seis (7) capítulos, devidamente enumerados e duas secções não enumeradas referentes a bibliografia e anexos.

i. Capítulo I – Introdução

Neste capítulo, faz-se a menção dos aspectos particulares da pesquisa de forma clara, explica-se a lógica que se usou para a realização do trabalho e faz-se a menção da pergunta de pesquisa. É composta pela contextualização, motivação, definição do problema, objectivos e metodologia.

ii. Capítulo II – Actividades Desenvolvidas no estágio

Neste capítulo, faz-se a menção e contextualização das actividades desenvolvidas no estágio, relação das actividades com o tema da implementação de processamento assíncrono, reflexão sobre as principais lições aprendidas durante o estágio e desafios encontrados e como foram abordados.

iii. Capítulo III – Revisão de literatura

Neste capítulo reúne-se as informações necessárias para servirem de guia teórico para a pesquisa, para utilizar uma sequência lógica com aspectos relacionados com a técnicas capaz de aumentar a disponibilidade do sistema.

iv. Capítulo IV – Caso de estudo

Neste capítulo apresenta-se os resultados dos estudos de campo realizados no banco Nedbank Moçambique a nível do sistema de internet banking, os dados serviram de auxílio para a proposta da solução adequada para o problema no contexto do sistema de IB.

v. Capítulo V – Proposta de solução

Neste capítulo, abordam-se aspectos técnicos e a metodologia utilizada para a elaboração de uma solução que possa resolver os constrangimentos identificados e descreve-se a solução proposta.

vi. Capítulo VI – Discussão de resultados

Neste capítulo apresentam-se os resultados obtidos na realização do trabalho e como a solução proposta pode reduzir os constrangimentos identificados, através da análise da revisão de literatura, caso de estudo e a proposta de solução.

vii. Capítulo VII – Considerações finais

Neste capítulo, verifica-se o cumprimento dos objectivos inicialmente propostos para obter um modelo de arquitectura do sistema de internet banking, de modo a minimizar os problemas identificados. No fim deixa-se recomendações para os próximos pesquisadores.

viii. Secção das Bibliografias

Nesta secção, faz-se a devida menção de todas as fontes que foram citadas no corpo do trabalho, bem como as que não foram citadas, mas que ajudaram na compreensão das matérias abordadas no trabalho.

ix. Anexos

Nesta secção, faz-se a apresentação de todos elementos adicionais que facilitam na compreensão dos aspectos contidos no trabalho.

2. Capítulo II – Actividades Desenvolvidas no estágio

Neste capítulo, faz-se a menção e contextualização das actividades desenvolvidas no estágio, relação das actividades com o tema da implementação de processamento assíncrono, reflexão sobre as principais lições aprendidas durante o estágio e desafios encontrados e como foram abordados.

2.1 Desenvolvimento de Software nos Canais Digitais

No âmbito das actividades do estagio como desenvolvedor de software, o estudante envolveu-se directamente no ciclo de vida de projectos relacionados aos canais digitais do Banco Nedbank Moçambique. Desde a concepção e design de novas funcionalidades até a implementação, testes e manutenção contínua.

A metodologia ágil, especificamente o *framework Scrum*, foi adoptada para gerir as tarefas e promover uma abordagem iterativa e colaborativa. interagindo com outros desenvolvedores, analistas de sistemas e representantes de diferentes áreas do banco.

2.2 Monitoria da Disponibilidade do Sistema de Internet Banking

Uma actividade central e constante ao longo do estágio foi a monitoria da disponibilidade do sistema de Internet Banking do Banco Nedbank. Essa função consistia em acompanhar

continuamente o desempenho e a operacionalidade do Internet Banking, para identificar possíveis falhas, gargalos ou períodos de inactividade.

Essa actividade permanente de monitoria foi instrumental não apenas para garantir a qualidade e confiabilidade do serviço prestado aos clientes, mas também desempenhou um papel crucial na definição do tema do problema a ser abordado no presente relatório de estágio.

Por tanto, O tema específico do estágio, "Implementação de Processamento Assíncrono em Transacções Bancárias para Melhorar a Disponibilidade do Serviço de Internet Banking," emergiu directamente desta actividade de monitoria da disponibilidade dos serviços de Internet Banking.

2.3 Identificação e Documentação dos Requisitos

A etapa de identificação e documentação dos requisitos proporcionou uma compreensão profunda das necessidades do sistema. A reflexão sobre esse processo destaca a importância da comunicação eficaz com os stakeholders. O desafio de traduzir expectativas muitas vezes complexas em requisitos claros impulsionou o desenvolvimento do estudante na arte crucial da eliciação de requisitos.

Para a identificação e documentação dos requisitos, adoptou-se uma abordagem participativa envolvendo workshops com os stakeholders. A metodologia utilizada foi o Modelo Incremental, permitindo a adaptação contínua. Ferramentas como JIRA foram empregadas para a gestão eficiente do processo, para facilitar a colaboração e o rastreamento de mudanças.

2.4 Análise Crítica dos Requisitos

A análise crítica dos requisitos revelou-se um exercício valioso para garantir a integridade e a abrangência do sistema. O confronto entre as expectativas dos stakeholders e a viabilidade técnica levou a ajustes iterativos. Esta fase enfatizou a necessidade de flexibilidade e adaptabilidade na gestão de requisitos em ambientes dinâmicos.

A análise crítica foi conduzida através de revisões formais e iterações constantes com os stakeholders. A técnica MoSCoW foi empregada para priorização, permitindo uma focalização eficaz nas funcionalidades críticas. A utilização do software Lucidchart auxiliou

na visualização clara dos requisitos, para facilitar a comunicação com as partes interessadas.

2.5 Exploração e Avaliação de Tecnologias:

A exploração e avaliação de tecnologias representaram uma jornada desafiadora, mas crucial. A reflexão destaca a importância de equilibrar inovação com estabilidade, que considera factores como escalabilidade e suporte da comunidade. A escolha criteriosa das tecnologias influenciou directamente o sucesso da implementação.

A pesquisa foi guiada por revisões sistemáticas e estudos de caso. Utilizou-se a metodologia RAD (Rapid Application Development) para avaliar rapidamente as tecnologias. O modelo de avaliação de tecnologias, baseado em critérios predefinidos, proporcionou uma selecção objectiva.

2.6 Implementação da Tecnologia Selecionada:

A implementação da tecnologia seleccionada trouxe à tona desafios inesperados, mas também oportunidades de aprendizado. A reflexão destaca a importância de investir em capacitação contínua da equipe para maximizar os benefícios da tecnologia escolhida. A adaptação à curva de aprendizado foi fundamental para uma implementação bem-sucedida. A implementação seguiu a metodologia Scrum, com iterações de duas semanas. O Git foi utilizado para controle de versão, permitindo uma colaboração eficiente. IDEs como VS Code e o Microsoft Visual Studio foram adoptadas para desenvolvimento.

2.7 Design Arquitectónico e Codificação:

A fase de design arquitectónico e codificação demandou uma abordagem estruturada e modular. A reflexão destaca a importância da arquitectura sólida na sustentabilidade do sistema. A aplicação dos princípios SOLID e o foco na manutenibilidade do código foram aprendizados essenciais nesse estágio.

O design arquitectónico baseou-se em padrões de microserviços e a codificação seguiu princípios SOLID. Utilizou-se a metodologia Kanban para gestão de tarefas. Ferramentas como Swagger foram empregadas para documentação automática da API. Testes unitários automatizados, que utiliza JUnit, garantiram a integridade do código.

2.8 Testes e Testes de Carga:

Os testes e testes de carga destacaram a necessidade de uma abordagem abrangente para garantir a robustez do sistema. A reflexão ressalta a importância de simular condições reais para identificar e corrigir falhas. A integração contínua e a automação dos testes mostraram-se fundamentais para garantir a qualidade do software.

A abordagem de testes foi orientada por técnicas ágeis, incluindo Test-Driven Development (TDD).

2.9 Resolução de Problemas Emergentes:

A resolução de problemas emergentes trouxe consigo uma abordagem sistemática. A reflexão destaca a importância de manter a calma sob pressão. A colaboração eficaz com as equipes de suporte e a comunicação transparente foram elementos-chave para minimizar o impacto de problemas inesperados.

A resolução de problemas seguiu um processo estruturado baseado na metodologia DMAIC (Definir, Medir, Analisar, Melhorar, Controlar). A Ferramenta, Jira Service Desk foi utilizada para gestão eficiente de tickets de suporte. O que facilitou a comunicação entre as equipes, contribuindo para tempos de resolução mais rápidos.

3. Capítulo III – Revisão da literatura

Neste capítulo reúne-se as informações necessárias para servirem de guia teórico para a pesquisa, que utiliza uma sequência lógica com aspectos relacionados com a técnicas capaz de aumentar a disponibilidade do sistema.

3.1. Internet banking

Segundo Angelakopoulos e Mihiotis (2011), o Internet Banking, conhecido como banco online, é um serviço oferecido por instituições financeiras que permite aos clientes realizar diversas transações bancárias por meio da internet. Essas transações incluem consultas de saldo, transferências entre contas, pagamento de contas e solicitação de empréstimos, que proporciona aos clientes uma forma conveniente e ágil de gerir suas finanças.

Conforme observado por Angelakopoulos e Mihiotis (2011), o e-banking abrange todas as transações bancárias que são conduzidas por meios eletrônicos, predominantemente via Internet, mas também fazendo uso de VPNs (Redes Privadas Virtuais), intranet, extranet, telefone e dispositivos móveis. Nesse contexto, não é necessário que o cliente se desloque até uma agência bancária para realizar essas operações. Os autores destacam três principais tipos de e-banking: Internet banking, phone banking e mobile banking. Cada um desses tipos utiliza um canal específico, sendo a Internet, o telefone e o dispositivo móvel, respectivamente. Essa diferenciação nos canais oferece aos clientes flexibilidade e conveniência na escolha do meio mais adequado às suas necessidades e preferências.

O Internet banking, uma vertente do e-banking, refere-se à utilização da Internet para oferecer uma gama de actividades bancárias, incluindo a transferência de fundos, pagamento de contas, consulta de saldos em contas correntes e poupanças, liquidação de hipotecas, subscrição de produtos financeiros, entre outras operações (Gbadeyan e Akinyoso, 2011). Desde a década de 1960, as instituições financeiras têm investido em sistemas de comunicação para gerenciar pagamentos, transações internacionais e gestão de estoques, estabelecendo conexões entre empresas e organizações em âmbito nacional e internacional (Tavares, 2008).

A evolução dos serviços bancários online baseados na Internet teve início com a Stanford Federal Credit Union (SFCU) em Outubro de 1994. Desde então, o Internet banking disseminou-se rapidamente em todo o mundo, impulsionado pela conveniência e agilidade proporcionadas nas transações financeiras (Yoon, 2010).

3.1.1. Vantagens e desvantagens do Internet banking

Quanto à importância e impacto do Internet Banking, conforme destacado por Sosyura (2016) e Gaurav e Kumar (2018), a modernização do sector bancário é evidente. A conveniência proporcionada aos clientes, que podem realizar transações a qualquer momento e de qualquer lugar, elimina a necessidade de deslocamento físico até uma agência bancária. Além disso, contribui para a eficiência operacional, reduzindo custos associados à infra-estrutura física e processos manuais.

os bancos online têm aprimorado significativamente a experiência do usuário, que elimina a necessidade de deslocamento para efectuar pagamentos, verificar extractos bancários ou realizar transferências entre contas. Essas são as principais vantagens do serviço, conforme destacado por diversos autores, tais como Angelakopoulos e Mihiotis (2011), Gbadeyan e Akinyoso (2011), e Silva (2008). Eles enfatizam a simplificação da gestão financeira, a conveniência do serviço, a disponibilidade 24 horas por dia, a acessibilidade global e o conforto de realizar transacções a partir do lar, tudo isso para resultar em economia de tempo que seria gasto em filas de espera nas agências bancárias, e sem custos adicionais para o cliente.

Outra vantagem para os usuários desse serviço é a flexibilidade para dedicar o tempo que desejarem, sem pressões externas, para tomar decisões de maneira mais tranquila. Embora a concorrência acirrada seja uma desvantagem para os bancos, ela beneficia os clientes ao permitir a comparação entre serviços de diferentes instituições, que possibilita a escolha daquele que oferece mais benefícios individuais (Angelakopoulos e Mihiotis, 2011).

No entanto, a literatura actual sobre Internet banking destaca algumas preocupações consideradas como principais desafios. A principal desvantagem para os clientes que utilizam o Internet banking está relacionada à segurança e privacidade de seus dados (Gbadeyan e Akinyoso, 2011; Silva, 2008). Outros desafios incluem baixo nível educacional, escassos conhecimentos em informática, linguagem técnica, barreiras culturais e logísticas, além da sobrecarga de legislação e informações diversas para os clientes (Gbadeyan e Akinyoso, 2011). Angelakopoulos e Mihiotis (2011) destacam como principal benefício para os bancos a oportunidade de alcançar novos segmentos de clientes em diferentes áreas geográficas por meio da adoção da tecnologia Internet banking.

Outra desvantagem para os bancos ao adoptarem o Internet banking é a crescente conscientização e exigência por parte dos clientes. Qualquer insatisfação com os serviços prestados pode levar os clientes a buscar alternativas oferecidas por diferentes instituições (Angelakopoulos e Mihiotis, 2011). Além disso, o investimento, manutenção e substituição de software, juntamente com os custos iniciais em equipamentos, representam desafios

adicionais para as instituições bancárias, conforme observado por Angelakopoulos e Mihiotis (2011).

A urgência em garantir a disponibilidade do Internet Banking é destacada por Gray e Reuter (2001) e Câmara e Silva (2004), a necessidade de disponibilidade do Internet Banking é uma demanda premente na sociedade moderna. O Internet Banking desempenha um papel crucial na vida cotidiana dos clientes bancários, oferecendo conveniência e acessibilidade aos serviços financeiros. A importância da disponibilidade é evidente na capacidade dos clientes de realizar transações a qualquer momento e de qualquer lugar, que elimina as restrições de horário e localização associadas às agências físicas.

3.2. Processamento de transações

De acordo com o estudo "The impact of electronic banking on the efficiency of commercial banks: Evidence from Europe" (Sosyura, 2016), as transações bancárias eletrônicas têm crescido significativamente nos últimos anos devido ao aumento do uso da internet e à popularidade dos dispositivos móveis. Essas transações eletrônicas podem proporcionar vantagens como maior conveniência e rapidez para os clientes, além de reduzir os custos para os bancos. Além disso, as transações bancárias eletrônicas também podem contribuir para a inclusão financeira, uma vez que possibilitam que pessoas que não têm acesso fácil às agências bancárias possam realizar transações financeiras (Sosyura, 2016).

Segundo o estudo "Transaction Processing in the 21st Century" (Gray & Reuter, 2001), o processamento de transações é um processo complexo que envolve várias etapas, incluindo a captura de dados, a validação de dados, a autorização de transações e a contabilização. O processamento de transações é o processo de verificação, validação e registro de transações financeiras. Isso inclui operações como pagamentos, transferências de dinheiro entre contas, compras com cartão de crédito e muito mais.

3.2.1. Tipos de processamento de transações

A seleção entre processamento síncrono e assíncrono desempenha um papel crucial na eficiência e disponibilidade do sistema de Internet Banking (Lee, Kim & Kim, 2018). O

processamento síncrono, com operações executadas sequencialmente, pode ser mais fácil de compreender, mas sua propensão a atrasos pode impactar negativamente a eficiência do sistema, especialmente em operações demoradas (Sosyura, 2016). Por outro lado, o processamento assíncrono, permitindo operações independentes, proporciona maior agilidade e resposta rápida mesmo em situações de alta demanda (Lee, Kim & Kim, 2018). Essas abordagens desempenham um papel crítico na otimização do desempenho do Internet Banking, que influencia directamente a experiência do usuário e a eficiência operacional do sistema (Sosyura, 2016; Lee, Kim & Kim, 2018).

- **Processamento síncrono**, De acordo com o estudo "Synchronous and Asynchronous Processing in Distributed Systems" (Coulouris, Dollimore, & Kindberg, 2018), o processamento síncrono é utilizado em sistemas distribuídos para garantir a consistência dos dados e evitar conflitos entre transacções. Ele garante que todas as transacções envolvidas sejam concluídas com sucesso antes que a próxima transacção possa ser iniciada. Além disso, o processamento síncrono também é utilizado para garantir a ordem das transacções, garantindo que as transacções sejam processadas na ordem em que foram solicitadas.
- **Processamento assíncrono**, De acordo com o estudo "Asynchronous Processing in Distributed Systems" (Coulouris, Dollimore, & Kindberg, 2018), o processamento assíncrono é utilizado em sistemas distribuídos para melhorar o desempenho e escalabilidade do sistema. Ele permite que as transacções sejam processadas de forma independente, independentemente do tipo ou tamanho das transacções. Além disso, o processamento assíncrono também é utilizado para suportar falhas no sistema, pois permite que as transacções sejam processadas mesmo se uma transacção falhar.

3.3. Tipos de processamento assíncrono

O processamento assíncrono de transacções representa uma abordagem fundamental para melhorar a disponibilidade no contexto do Internet Banking. Raza (2018) destaca que essa técnica permite que as transacções sejam iniciadas de forma independente e concluídas

sem depender uma da outra. Essa separação entre a iniciação e a conclusão das transacções oferece uma série de vantagens significativas.

Al-awzan (2017) corrobora a importância do processamento assíncrono ao afirmar que essa abordagem oferece maior flexibilidade na gestão de recursos, pois as transacções podem ser tratadas de forma escalonada, que otimiza a utilização dos recursos disponíveis. Dessa forma, o processamento assíncrono não apenas contribui para a melhoria da disponibilidade, mas também para a eficiência operacional do sistema de Internet Banking.

"O processamento assíncrono é uma abordagem crucial em sistemas modernos, permitindo a execução de tarefas de forma não sequencial e para aumentar a eficiência operacional" (Taylor, 2016, p. 45). Esta seção explora diferentes tipos e formas de comunicação assíncrona, que destaca as nuances de mensagens assíncronas, callbacks, promessas e filas de mensagens.

3.3.1. Callbacks e Promessas

No panorama do processamento assíncrono, a utilização de *callbacks* e promessas desempenha um papel fundamental na gestão de operações não bloqueantes (como o caso de requisições *HTTP/HTTPS*). Esses mecanismos oferecem soluções elegantes para lidar com a assíncrona, permitindo a execução eficiente de tarefas em sistemas complexos.

Os **callbacks** são funções que são passadas como argumentos para outras funções e executadas após a conclusão de uma operação assíncrona. Essa abordagem é particularmente evidente em linguagens como *JavaScript*, onde a assíncrona é inerente a muitas operações, como requisições de rede e manipulação de eventos do usuário. De acordo com *Flanagan* (2016), "os *callbacks* são essenciais para garantir que certas acções só ocorram após a conclusão bem-sucedida de uma operação assíncrona."

```
function operacaoAssincrona(callback) {  
  // Simula uma operação assíncrona  
  setTimeout(function () {  
    console.log("Operação concluída");  
    callback();  
  });  
}
```

```
    }, 1000);  
  }  
  
  operacaoAssincrona(function () {  
    console.log("Callback chamado após a operação assíncrona");  
  });  
});
```

Fonte: elaborado pelo autor.

No entanto, o uso excessivo de callbacks pode levar a um fenómeno conhecido como "Callback Hell" ou "Pyramid of Doom", onde a estrutura do código torna-se profundamente aninhada e difícil de manter. Para superar essa limitação, as promessas foram introduzidas.

As **promessas** representam uma melhoria significativa na legibilidade e organização do código assíncrono. Uma promessa é um objecto que representa o resultado eventual (ou falha) de uma operação assíncrona. Segundo Brown (2020), "as promessas fornecem um contracto claro para o tratamento futuro de um valor, que torna o código mais compreensível e menos propenso a erros."

```
function realizarOperacaoAssincrona() {  
  return new Promise((resolve, reject) => {  
    // Operação assíncrona  
    // ...  
  
    // Resolver a promessa após a conclusão bem-sucedida  
    resolve("Operação concluída com sucesso!");  
  
    // Ou rejeitar a promessa em caso de falha  
    reject("Erro durante a operação assíncrona");  
  });  
}  
  
// Utilizar a promessa  
realizarOperacaoAssincrona()  
  .then(resultado => {  
    console.log(resultado);  
  })  
  .catch(erro => {  
    console.error(erro);  
  });  
});
```

Fonte: elaborado pelo autor.

As promessas oferecem uma sintaxe mais limpa para tratar resultados assíncronos, que evita a complexidade associada aos callbacks. Além disso, permitem encadear operações de forma mais legível, que melhora a manutenção do código em comparação com abordagens baseadas apenas em callbacks.

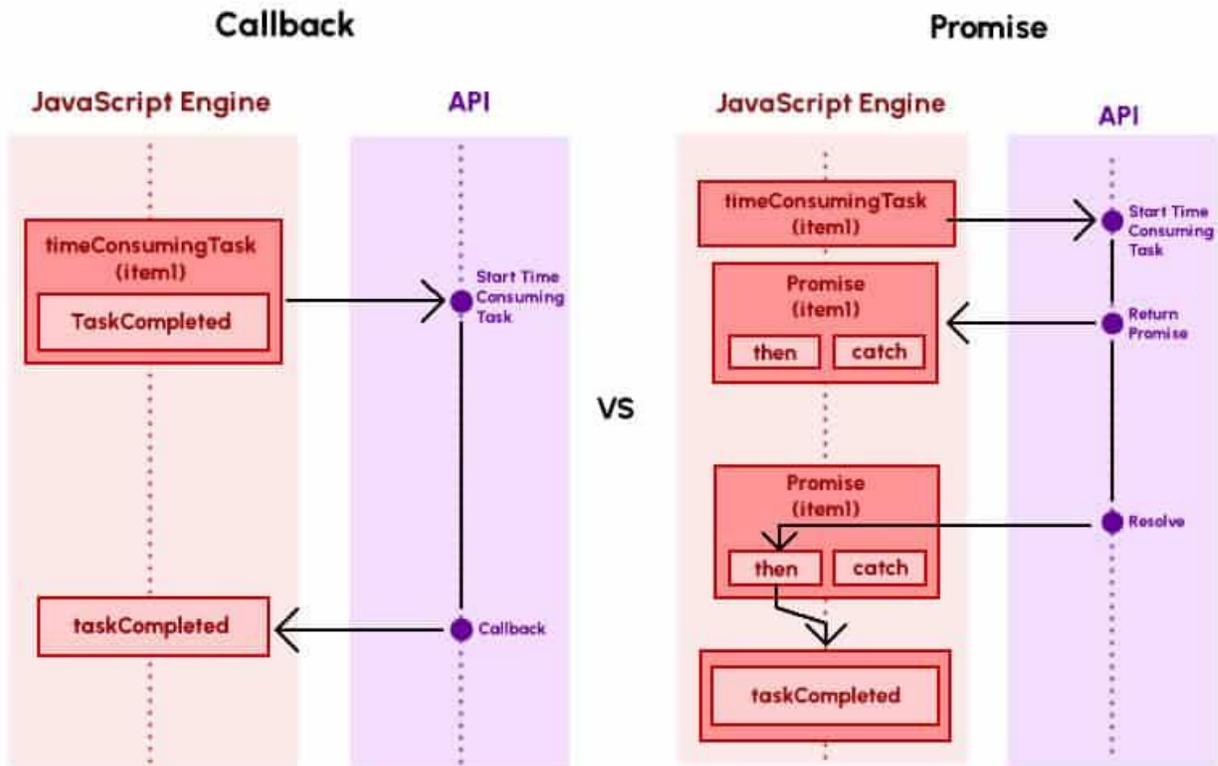


Figura 1: *Callbacks VS Promises*

Fonte: Savinu Vijay (2021).

3.3.2. Mensagens Assíncronas

No âmbito das mensagens assíncronas, a comunicação entre os diferentes elementos do sistema ocorre sem a necessidade de sincronização temporal estrita. Segundo Roberts (2019), "as mensagens assíncronas permitem que os componentes do sistema se comuniquem sem depender de um relógio compartilhado, que possibilita a execução de tarefas de forma independente."

Como menciona Chen (2020), um exemplo comum de aplicação de mensagens assíncronas é encontrado em sistemas distribuídos, onde módulos distintos precisam comunicar eventos ou actualizações sem interrupções no fluxo principal de execução. Essa abordagem é particularmente valiosa em ambientes nos quais a latência e a escalabilidade são considerações críticas.

De acordo com Turner (2017), as vantagens das mensagens assíncronas incluem a redução do acoplamento entre componentes, permitindo maior flexibilidade e manutenção do sistema. Contudo, Huang (2021) destaca desafios que podem surgir na gestão da ordem de mensagens e na garantia da entrega em ambientes distribuídos complexos.

Como menciona Kim (2018), a implementação eficiente de mensagens assíncronas requer a selecção adequada de protocolos de comunicação e estruturas de mensagens. É importante considerar factores como a confiabilidade da entrega, escalabilidade e a capacidade de lidar com falhas de comunicação para garantir um sistema robusto.

```
import asyncio

async def produtor(queue, messages):
    for message in messages:
        print(f"Produzindo mensagem: {message}")
        await asyncio.sleep(1) # Simula uma operação demorada
        await queue.put(message)

async def consumidor(queue):
    while True:
        message = await queue.get()
        print(f"Consumindo mensagem: {message}")
        await asyncio.sleep(2) # Simula um processamento demorado

async def main():
    # Cria uma fila assíncrona
    queue = asyncio.Queue()

    # Corrotinas produtora e consumidora
    produtor_coro = produtor(queue, ["Mensagem 1", "Mensagem 2", "Mensagem 3"])
    consumidor_coro = consumidor(queue)
```

```
# Executa as corrotinas simultaneamente
await asyncio.gather(produtor_coro, consumidor_coro)

# Inicia o loop de eventos assíncronos
asyncio.run(main())
```

- A função **produtor** produz mensagens e as coloca em uma fila assíncrona (queue).
- A função **consumidor** consome as mensagens da fila assíncrona.
- A função **main** cria a fila e inicia ambas as rotinas simultaneamente que usa `asyncio.gather`.

3.3.3. Filas de Mensagens

O uso de filas de mensagens é uma estratégia fundamental em ambientes que empregam o processamento assíncrono, que desempenha um papel crucial na organização e gestão eficiente de tarefas. Este subcapítulo explora a função e implementação das filas de mensagens, que destaca a sua importância na promoção da ordem e concorrência em sistemas modernos.

Segundo (Lewis, 2018), as filas de mensagens actuam como intermediárias na comunicação assíncrona, permitindo que diferentes partes do sistema se comuniquem de forma eficiente. Essas filas armazenam mensagens até que o destinatário esteja pronto para processá-las, que facilita uma abordagem não bloqueante e para melhorar a escalabilidade do sistema.

Conforme destacado por Mitchell (2019), "a utilização de filas de mensagens oferece vantagens significativas, incluindo a prevenção de bloqueios e a capacidade de lidar com picos de carga, garantindo um desempenho consistente do sistema mesmo em situações de demanda intensa."

Um exemplo prático do uso de filas de mensagens pode ser observado em sistemas de processamento de pedidos online. Quando um pedido é feito, a informação sobre esse pedido é colocada numa fila. O sistema de processamento pode então retirar os pedidos da fila à medida que estiver pronto para os processar, garantindo que a ordem de chegada seja respeitada.

3.4. Formas de implementação do processamento assíncrono

Segundo (Smith, 2021, p. 76), o surgimento das formas de implementação do processamento assíncrono está profundamente ligado à evolução das tecnologias de computação e à busca por sistemas mais eficientes e responsivos. Examinar suas origens e compreender sua importância proporciona insights valiosos sobre a transformação dos paradigmas de desenvolvimento de software

Para Martins (2022), O processamento assíncrono refere-se a uma abordagem em que as tarefas são executadas independentemente umas das outras, sem a necessidade de esperar que uma tarefa seja concluída para iniciar outra. Isso contrasta com o processamento síncrono, onde as tarefas são executadas sequencialmente e em uma ordem específica. O processamento assíncrono é frequentemente utilizado em sistemas computacionais, comunicação de dados e interações entre componentes de software.

3.4.1. Programação Assíncrona

Para Gaurav e Kumar (2018), a programação assíncrona é uma abordagem fundamental no contexto do processamento de transações no Internet Banking, pois permite que o sistema execute tarefas de forma eficiente, sem bloquear a execução subsequente de operações. Essa técnica é especialmente valiosa quando lidamos com operações que podem levar tempo, como a comunicação com serviços externos ou a manipulação de grandes conjuntos de dados.

Modelos de Programação Assíncrona: existem diversos modelos de programação assíncrona, cada um com suas características distintas. Gaurav e Kumar (2018) destacam que o modelo baseado em callbacks é uma abordagem comum, onde uma função é fornecida para ser executada após a conclusão de uma operação assíncrona. Além disso, o uso de Promessas, conforme introduzido por Newcomer (2009), oferece uma estrutura mais organizada para lidar com operações assíncronas, permitindo um código mais legível e fácil de manter.

Tecnologias e Frameworks Utilizados: A implementação eficaz da programação assíncrona no Internet Banking do Banco Nedbank Moçambique requer o uso de tecnologias e frameworks adequados. Kreps et al. (2011) sugerem o RabbitMQ com ferramenta de gestão de filas, que oferece capacidade avançada de gestão de mensagens para executar

transacções assíncronas de forma eficiente. Além disso, o uso de tecnologias que suportam Promessas, como as presentes em linguagens de programação modernas, pode facilitar a implementação de código assíncrono de maneira mais elegante.

3.4.2. Processamento Assíncrono em Hardware

Diferentemente do processamento síncrono em hardware, onde as operações seguem um ritmo predefinido, as arquitecturas assíncronas permitem que as unidades de processamento operem de forma autónoma. Segundo Hoque et al. (2018), esta autonomia resulta em menor consumo de energia e maior eficiência, uma vez que as unidades podem responder dinamicamente às alterações nas demandas de processamento.

Vantagens do Processamento Assíncrono em Hardware: A vantagem mais evidente do processamento assíncrono em hardware reside na capacidade de lidar eficazmente com eventos imprevisíveis e interacções variáveis. Além disso, Gaurav e Kumar (2018), destacam que essa abordagem favorece a optimização do desempenho em tempo real, sendo especialmente relevante em transacções financeiras onde a resposta rápida é crucial.

3.4.3. Comunicação Assíncrona em Redes

Mensagens Assíncronas, uma das formas mais comuns de comunicação assíncrona em redes é a troca de mensagens assíncronas. Segundo Videla e Roche (2012), esta abordagem permite que os sistemas enviem e recebam informações sem a necessidade de uma sincronização rigorosa, contribuindo para a flexibilidade na comunicação entre serviços.

Callbacks e Promessas, além das mensagens assíncronas, a utilização de callbacks e promessas é relevante na comunicação assíncrona em redes. O modelo de callbacks, conforme mencionado por Gaurav e Kumar (2018), permite que uma função seja executada em resposta a um evento específico, que facilita a manipulação de operações assíncronas. Por outro lado, as promessas fornecem uma estrutura mais organizada e fácil de gerir para lidar com operações assíncronas complexas (Newcomer, 2009).

Filas de Mensagens, a implementação de filas de mensagens é outra estratégia essencial na comunicação assíncrona em redes. A alocação de mensagens em filas permite a execução ordenada de operações, contribuindo para a eficiência e consistência nas transacções (Gray & Reuter, 2001).

3.4.3.1. Webservice

Segundo Gaurav e Kumar (2018), um web service é uma técnica de comunicação entre sistemas baseada em protocolos da web, como HTTP e SOAP, que permite que diferentes aplicações e sistemas se comuniquem e troquem informações. Eles podem ser usados para integrar diferentes aplicativos em diferentes plataformas, linguagens e organizações.

Um dos principais benefícios do uso de web services é a interoperabilidade, ou seja, a capacidade de diferentes sistemas se comunicarem e compartilharem dados independentemente de sua plataforma, linguagem de programação e organização. Isso permite a criação de aplicações distribuídas e integradas, para aumentar a flexibilidade e escalabilidade do sistema.

3.4.3.2. Push notification

Para Babich, A. (2018), as notificações push são mensagens que são enviadas para dispositivos móveis, geralmente para dispositivos iOS ou Android, com o objectivo de informar o usuário sobre eventos ou actualizações em um aplicativo ou sistema. Essas notificações podem ser enviadas em tempo real ou programadas para serem enviadas em uma data e hora específicas.

As notificações push são diferentes de outras formas de notificação, como e-mails ou SMS, pois elas são enviadas directamente para o dispositivo do usuário, sem a necessidade de abrir o aplicativo ou o navegador. Isso pode aumentar significativamente a eficácia das notificações, pois elas são visíveis mesmo quando o usuário não está a usar o aplicativo.

3.5. Ferramentas no Processamento Assíncrono

O sucesso da implementação do processamento assíncrono de transacções no contexto do Internet Banking depende, em grande medida, da escolha e utilização adequada de

ferramentas e tecnologias específicas. Neste subcapítulo, serão exploradas algumas das ferramentas relevantes para a eficácia do processamento assíncrono de transações.

3.5.1. Apache Kafka

De acordo com António, R. M. (2020), o Apache Kafka é uma plataforma de transmissão distribuída de código aberto, desenvolvida em Java e Scala. Esta plataforma disponibiliza três funcionalidades semelhantes às de uma fila de mensagens. Incorpora mecanismos robustos de publicação/assinatura, garante a transmissão contínua e tolerante a falhas de registos, e processa esses registos à medida que ocorrem. Esses registos são organizados em tópicos, os quais, por sua vez, são subdivididos em partições. Como ilustrado na Figura 2, a plataforma é composta por quatro APIs principais:

- API do Produtor: possibilita a publicação de registos nos tópicos.
- API do Consumidor: permite o consumo/aceitação desses registos armazenados nos tópicos para subsequente processamento.
- API de Streams: transforma transmissões de entrada em transmissões de saída.
- API de Conector: permite a utilização de produtores e consumidores reutilizáveis, que conecta um tópico a uma aplicação.

A comunicação entre o servidor e os clientes ocorre através de um protocolo binário sobre o protocolo TCP, que define as trocas de mensagens entre cliente e servidor como pares de mensagens de solicitação/resposta. O Apache Kafka é compatível com várias linguagens, como Java, Python, JavaScript, Ruby, entre outras. Esta ferramenta oferece a vantagem de cada tópico possuir propriedades dos dois modelos de funcionamento mencionados anteriormente. Isso permite a escalabilidade do processamento pelos consumidores, por exemplo, através do balanceamento de carga entre esses consumidores. Além disso, possibilita a partilha da mesma mensagem por vários consumidores.

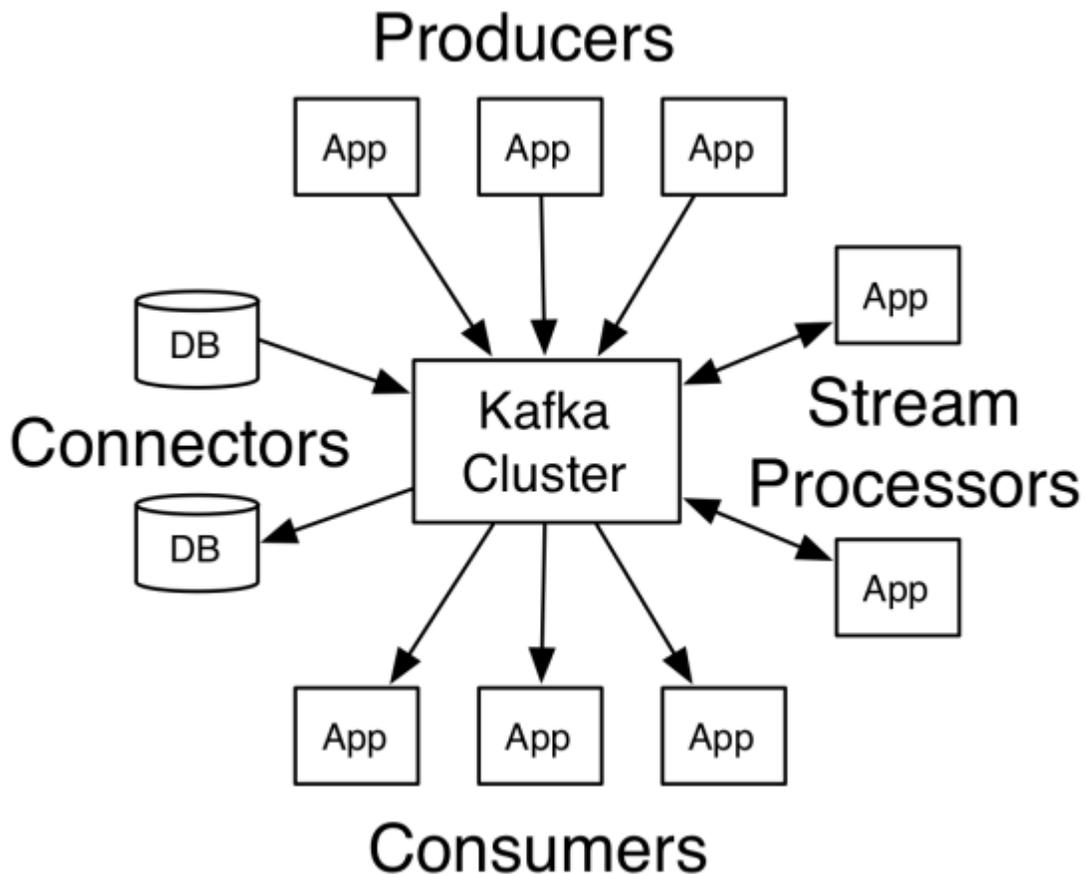


Figura 2: Arquitectura de Apache Kafka

Fonte: António, R. M. (2020), *Microsserviços para Reconhecimento de Emoção em Música*

3.5.2. RabbitMQ

De acordo com António, R. M. (2020), o RabbitMQ é uma das plataformas open-source de transmissão de mensagens mais populares. Destaca-se pela sua leveza e facilidade de implementação, sendo adequado tanto para ambientes Cloud como para execução em máquinas locais (on-premises). Sua versatilidade permite o uso em configurações distribuídas (dentro da mesma entidade) e federadas (abrangendo diferentes entidades organizacionais), atendendo aos requisitos de alta escalabilidade e disponibilidade. Suporta mensagens assíncronas e pode ser executado através do Docker, que disponibiliza imagens correspondentes, ou configurado para utilizar ferramentas como Puppet, entre outras.

Desenvolvido para facilitar a troca de mensagens entre aplicações escritas em diversas linguagens, como PHP, Python, JavaScript, entre outras, o RabbitMQ permite execução distribuída por meio de clusters de alta disponibilidade e taxa de transferência. Além disso, oferece uma variedade de extras (plugins) e ferramentas, juntamente com mecanismos abrangentes de gestão e monitorização. Para aplicações em nuvem e empresas, proporciona funcionalidades de autenticação, autorização, suporte a TLS e LDAP.

O RabbitMQ opera sobre o protocolo AMQP (Advanced Message Queuing Protocol), uma norma de mensagens na camada de aplicação do modelo OSI, que possibilita a comunicação entre aplicações e middlewares com função de message broker. Utiliza o protocolo TCP para a transmissão de mensagens, garantindo a entrega segura, podendo adicionalmente empregar o TLS para protecção do conteúdo das mensagens.

No contexto do RabbitMQ, conforme ilustrado na Figura 3, os serviços que produzem mensagens publicam-nas num mecanismo denominado exchange, presente no servidor RabbitMQ. Este mecanismo pode ser equiparado a caixas de correio, tendo a função de receber as mensagens dos publishers. Subsequentemente, distribuem réplicas dessas mensagens por várias filas (queues), seguindo regras denominadas bindings. Após o armazenamento nas queues, os serviços consumidores podem aceder às mensagens. O protocolo AMQP incorpora o conceito de "message acknowledgement", permitindo que o consumidor comunique ao servidor RabbitMQ que recebeu a mensagem. Este mecanismo pode ser activado automaticamente ou posteriormente, conforme a lógica implementada pelo serviço consumidor. Adicionalmente, o RabbitMQ disponibiliza um plugin de gestão e manutenção para ambiente web.

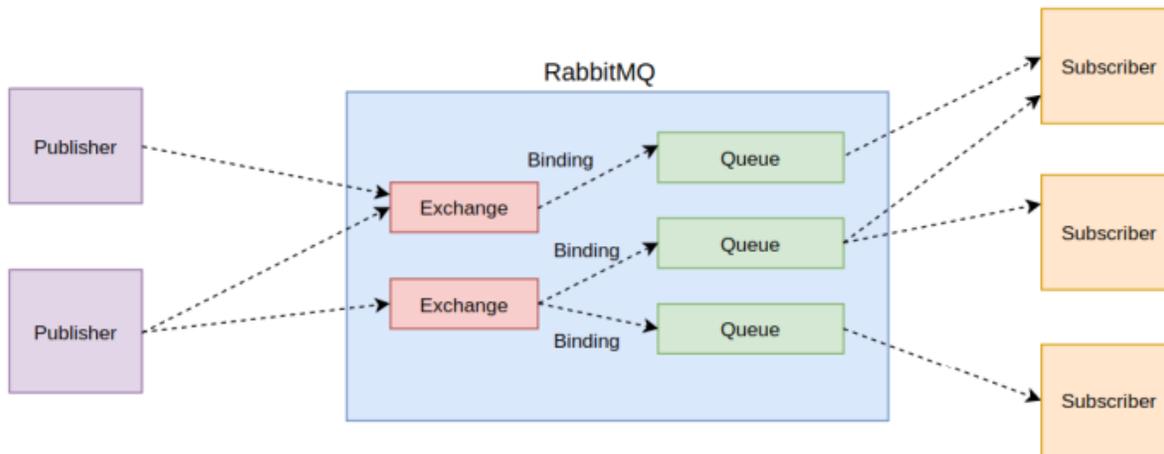


Figura 3: Arquitectura geral do RabbitMQ

Fonte: António, R. M. (2020), *Microserviços para Reconhecimento de Emoção em Música*

3.5.3. Open Message Queue

Segundo António, R. M. (2020), a plataforma open-source Open Message Queue (MQ) é definida como "um middleware inteiramente orientado a mensagens"⁵¹. Em outras palavras, trata-se de uma plataforma intermediária que oferece serviços voltados para a transmissão de mensagens. Ela implementa um serviço confiável de transmissão de mensagens, permitindo que as aplicações comuniquem entre si sem depender de comunicações síncronas, que proporcio buffering entre os produtores de mensagens e os consumidores.

O OpenMQ implementa a API do Java Message Service, uma API em Java que possibilita a criação, envio, recepção e leitura de mensagens por meio de um conjunto de interfaces e semânticas associadas. Além disso, oferece opções como transmissão de mensagens através de SOAP/HTTP, distribuição escalável de mensagens e amplo suporte a JMX (Java Management Extensions), entre outros. O modelo de funcionamento é representado de forma gráfica na Figura 4.

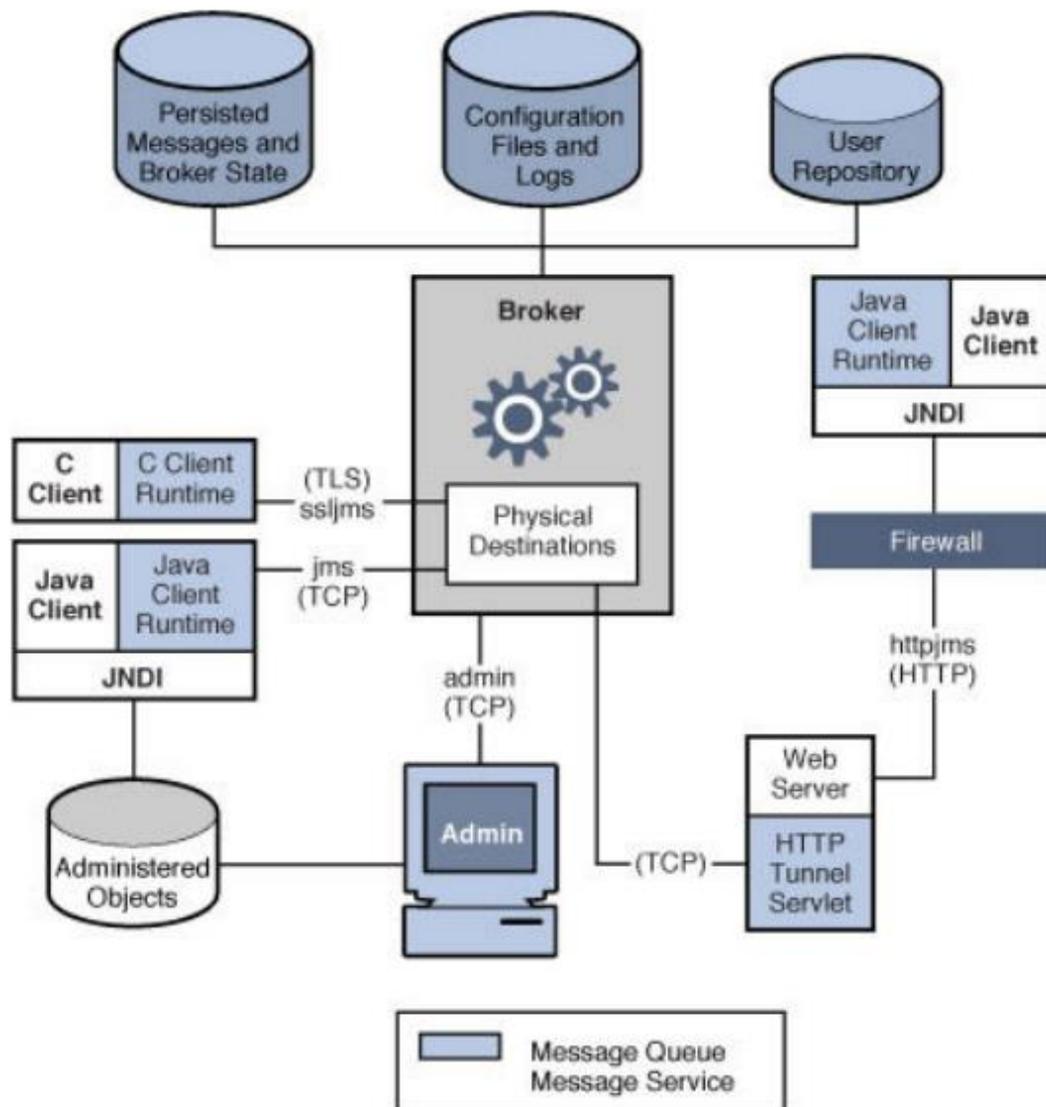


Figura 4: Arquitectura do OpenMQ

Fonte: António, R. M. (2020), Microserviços para Reconhecimento de Emoção em Música

4. Capítulo IV – Caso de Estudo

Neste capítulo apresenta-se os resultados dos estudos de campo realizados no banco Nedbank Moçambique a nível do sistema de internet banking, os dados serviram de auxílio para a proposta da solução adequada para o problema no contexto do sistema de IB.

4.1. Descrição dos sistemas do banco

O banco Nedbank Moçambique é um dos maiores bancos de Moçambique e é conhecido por sua ampla gama de serviços bancários, incluindo contas de depósito, empréstimos, cartões de crédito e muito mais. O T24 é o sistema core do banco, usado para gerir suas operações básicas, como contas de depósito, empréstimos e transacções bancárias.

O banco Nedbank Moçambique também possui conexão com sistemas non-core, como o internet banking, que é usado para permitir que os clientes façam transacções e acessem outros serviços através da internet. O internet banking é considerado um sistema non-core porque não é essencial para o funcionamento básico do banco, mas ainda é importante para ajudar o banco a oferecer uma ampla gama de serviços aos seus clientes e para se manter competitivo no mercado.

Em resumo, o banco Nedbank Moçambique usa o Temenus T24 como sistema core para gerir suas operações básicas e possui conexão com o sistema non-core de internet banking para permitir que os clientes façam transacções e acessem outros serviços através da internet. Isso ajuda o banco a oferecer uma ampla gama de serviços aos seus clientes e a se manter competitivo no mercado.

4.2. Descrição do ecossistema dos aplicativos bancários

De acordo com KPMG (2019), o ecossistema dos aplicativos bancários é o conjunto de aplicativos, serviços e outros participantes que trabalham juntos para proporcionar aos clientes bancários uma ampla variedade de opções de gestão de contas, transacções e outras necessidades bancárias através de seus dispositivos móveis. Isso inclui aplicativos de bancos, provedores de serviços de pagamento, provedores de tecnologia e outros participantes que trabalham juntos para oferecer uma experiência de usuário conveniente e segura para os clientes.

O ecossistema de um banco é importante porque é o conjunto de sistemas, tecnologias e processos que permitem que o banco funcione de maneira eficiente e atenda às necessidades dos seus clientes. É uma parte fundamental do funcionamento do banco e é essencial para o seu sucesso.

4.2.1. Descrição do sistema Core

O T24 é o sistema core do banco, é um software de gestão de banco de dados que é usado para gerir suas operações básicas, como contas de depósito, empréstimos e transacções bancárias. O T24 é também responsável por armazenar e processar todas as informações financeiras do banco, incluindo saldos de conta, transacções e histórico de pagamentos e efectuar transacções.

4.2.2. Descrição do sistema Non-Core

O internet banking é um dos sistemas non-core do banco Nedbank Moçambique. É software que permite que os clientes realizem transacções e acessem outros serviços através da internet.

O sistema de internet banking comunica-se através de middlewares com o sistema core T24, de modo a efectuar manipulações ou consultas às contas registadas no T24 e aos outros sistemas de terceiros, como o M-Pesa, recargaAki, entre outros.

4.2.3. Organograma

De acordo com Daft (2015) afirma que, uma estrutura orgânica de uma empresa é uma forma de organização que se baseia em princípios flexíveis, adaptativos e descentralizados. O organograma é uma representação visual da estrutura hierárquica de uma organização, para mostrar os diferentes níveis de autoridade, responsabilidades e relações entre os membros. Na figura abaixo, é apresentado o organograma do banco Nedbank Moçambique.

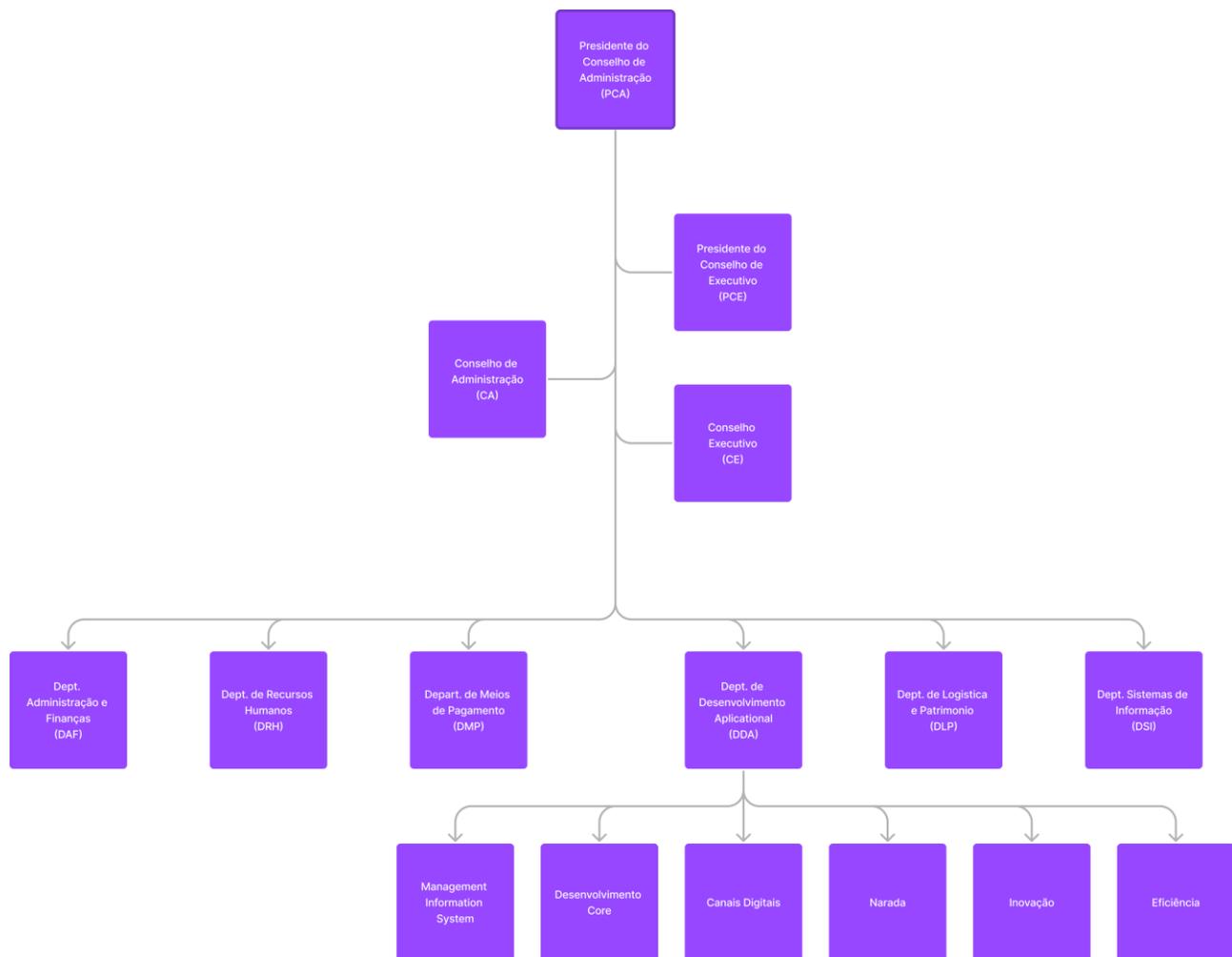


Figura 2: Organograma do banco Nedbank Moçambique

Fonte: Elaborada pelo autor

A solução proposta pelo autor deve ser implementada a nível do departamento de Desenvolvimento aplicacional em conjunto com o departamento de sistemas de informação.

4.3. Situação actual

A análise da situação actual consistiu em averiguar as causas do fenómeno de indisponibilidade do sistema de internet banking com vista a ter uma direcção sobre a proposta de solução e a identificação dos constrangimentos existentes situação actual. Na

busca de informação foi questionado o head of Digital Channels Development do banco Nedbank Moçambique, Paulo Zacarias.

De acordo com as pesquisas realizadas foi possível perceber que:

- Não foi encontrado um mecanismo de recuperação de transacções que não foram realizadas devido à indisponibilidade de um sistema de terceiros no sistema de internet banking
- Sempre que o sistema de terceiros estiver indisponível, as transacções não são efectuas e todos os pedidos são cancelados.
- Segundo as regras do negocio é possível agendar transacção de processamento de ficheiro para ser processada em momento oportuno.
- A instituição apenas regista a ocorrência de time-outs no processamento de transacções, mas não foi encontrado um mecanismo de resiliência, como um reprocessamento simples.
- Nos momentos de alta demanda, o processamento de ficheiro de pagamento de salario afecta na disponibilidade do serviço de internet banking.

4.4. Constrangimentos da situação actual

Pelas informações mencionadas na situação actual, os seus constrangimentos são dos mais diversos, como os seguintes:

- Alto nível de volume transacções concorrentes, oque causa que causa um problema de gargalo, pois o seu processamento é feito de forma síncrona e a medida que chegam novas transacções os uso dos recursos dos servidores vai aumentar cada vez mais.
- Falha no processamento de transacções devido a erros na comunicação entre o ecossistema dos aplicativos do banco com os sistemas de terceiros.
- Lentidão ou indisponibilidade perceptível ao usuário do sistema, devido aos motivos listados acima.

- Incapacidade de resposta a falhas simples no processamento de transacções bancárias, especificamente, a não existência de um mecanismo de resiliência e recuperação de erros.

4.5. Constrangimentos resolvidos com o modelo.

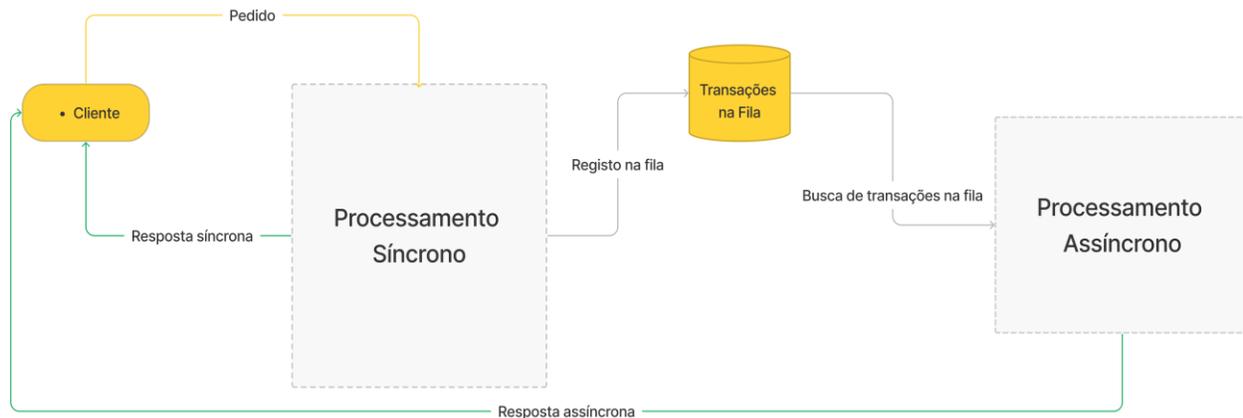


Figura 3: Modelo híbrido de processamento de transacções

Fonte: elaborada pelo autor

O modelo híbrido de processamento de transacções no internet banking resolve os seguintes constrangimentos:

- Diminuição do volume transacções concorrentes, pois com o processamento assíncrono de transacções é imune ao problema de concorrência e a passagem de parte das transacções para o modelo assíncrono de processamento, diminui o volume de transacções.
- Resolução do problema de falha no processamento de transacções devido a erros na comunicação, pois com a transacção em uma fila, o sistema de gestão de transacções pode voltar a tentar executar a transacção quantas vezes forem precisas.
- Resolução do problema de lentidão ou indisponibilidade perceptível ao usuário do sistema, pois tendo a diminuição do alto volume transacções e com a resolução do problema de falha no processamento de transacções, terá um alívio significativo no uso dos servidores e consequentemente a resolução do problema de gargalo.

- Capacidade de resposta a falhas simples no processamento de transacções bancárias com a implementação de mecanismos de resiliência e recuperação de erros nas transacções com base no sistema de gestão de filas.

4.6. Proposta de solução

A solução proposta é uma evolução do sistema de internet banking, para um modelo híbrido. Este modelo consiste em utilizar o processamento síncrono para operações críticas e de alta segurança, como transferências e pagamentos, e o processamento assíncrono para operações menos críticas, como pagamento de salários e consulta de extracto. Isso permitirá que o sistema seja mais rápido e eficiente, além de aumentar a resiliência do sistema, isto em termos de diminuição dos erros no processamento de transacções.

Alem do facto de termos a vantagem da escalabilidade do sistema, permitindo que o sistema lide com um maior número de transacções simultaneamente. Isso é especialmente importante para sistemas de internet banking, que precisam suportar uma grande quantidade de usuários e transacções ao mesmo tempo. Outra vantagem de um sistema híbrido é que ele permite aos usuários realizar transacções e operações de forma mais autónoma, sem a necessidade de aguardar uma confirmação ou autorização em tempo real. Isso pode melhorar significativamente a experiência do usuário e tornar o sistema mais atractivo.

4.7. Possíveis soluções para a melhoria da disponibilidade do sistema de IB

Existem varias soluções para melhorar a disponibilidade do sistema de internet banking. Das quais destacam-se, implementação de cloud computing, o aumento da capacidade do servidor e o load balancing.

Cloud Computing: A computação em nuvem é uma abordagem em que os recursos de computação, armazenamento e rede são disponibilizados pela internet, permitindo a escalabilidade e flexibilidade dos serviços. Ao migrar para a nuvem, as instituições financeiras podem se beneficiar da infra-estrutura altamente disponível oferecida pelos

provedores de nuvem. Isso significa que os servidores e recursos estão distribuídos geograficamente, reduzindo o risco de interrupções causadas por problemas em um único local físico. Além disso, a escalabilidade automática da nuvem permite ajustar os recursos de acordo com a demanda, garantindo um alto nível de disponibilidade.

Aumento da capacidade do servidor: à medida que o número de usuários e transações aumenta, é importante que os servidores tenham capacidade suficiente para lidar com o tráfego. Aumentar a capacidade do servidor envolve a adição de mais recursos, como processadores, memória RAM e armazenamento. Isso permite que o sistema tenha mais recursos para lidar com as solicitações e minimize a possibilidade de sobrecarga. O dimensionamento vertical é uma abordagem comum para aumentar a capacidade do servidor, que envolve actualizar os recursos do servidor existente ou adquirir servidores mais potentes.

Load balancing: O load balancing é uma técnica que distribui a carga de trabalho entre vários servidores para otimizar o desempenho e a disponibilidade do sistema. Existem várias estratégias de balanceamento de carga, incluindo:

Balanceamento de carga baseado em hardware: Nessa abordagem, um hardware dedicado é usado para distribuir o tráfego entre os servidores. Isso pode ser feito por meio de dispositivos de balanceamento de carga ou switches com recursos integrados de balanceamento de carga.

Balanceamento de carga baseado em software: nesse caso, um software de balanceamento de carga é instalado em um servidor separado ou em cada servidor da infra-estrutura. O software monitora o tráfego e distribui as solicitações entre os servidores disponíveis.

Balanceamento de carga baseado em DNS: Nessa técnica, o servidor DNS (Domain Name System) distribui o tráfego que encaminha as solicitações para diferentes endereços IP de servidor com base em um algoritmo de balanceamento de carga.

O load balancing garante que nenhum servidor fique sobrecarregado, que melhora o desempenho e a disponibilidade do sistema de internet banking. Além disso, se um servidor falhar, o tráfego é redireccionado automaticamente para outros servidores disponíveis, garantindo a continuidade do serviço.

4.7.1. Constrangimentos da implementação das soluções

Devido as normas da entidade regulatória (banco de Moçambique) a implementação da solução de Cloud Computing não é possível.

Por outro lado, as decisões dos administradores no banco Nedbank Moçambique impossibilitam a implementação de qualquer solução que envolva a aquisição de novos servidores ou novos componentes de hardware. Isso é aplicado as soluções de aumento da capacidade do servidor e a solução de implementação de múltiplos servidores com load balancing.

4.8. Arquitectura do modelo híbrido do sistema de internet banking.

A arquitectura proposta para o modelo híbrido do sistema de internet banking é a junção do sistema de processamento síncrono existente e um sistema de processamento assíncrono. A parte do sistema síncrono tratará dos pedidos críticos enquanto a parte do sistema assíncrono será baseada em uma fila para todos os pedidos não críticos. Assim, teremos um sistema híbrido, onde a parte síncrona grava todos os pedidos não críticos na fila, os quais serão processados pela parte assíncrona em momento oportuno.

A combinação das duas partes do sistema permite que o internet banking ofereça uma experiência rápida e segura para o usuário, além de ser escalável para suportar o crescimento do tráfego. Além disso, essa arquitectura também permite a implementação de novos recursos e funcionalidades de forma mais eficiente.

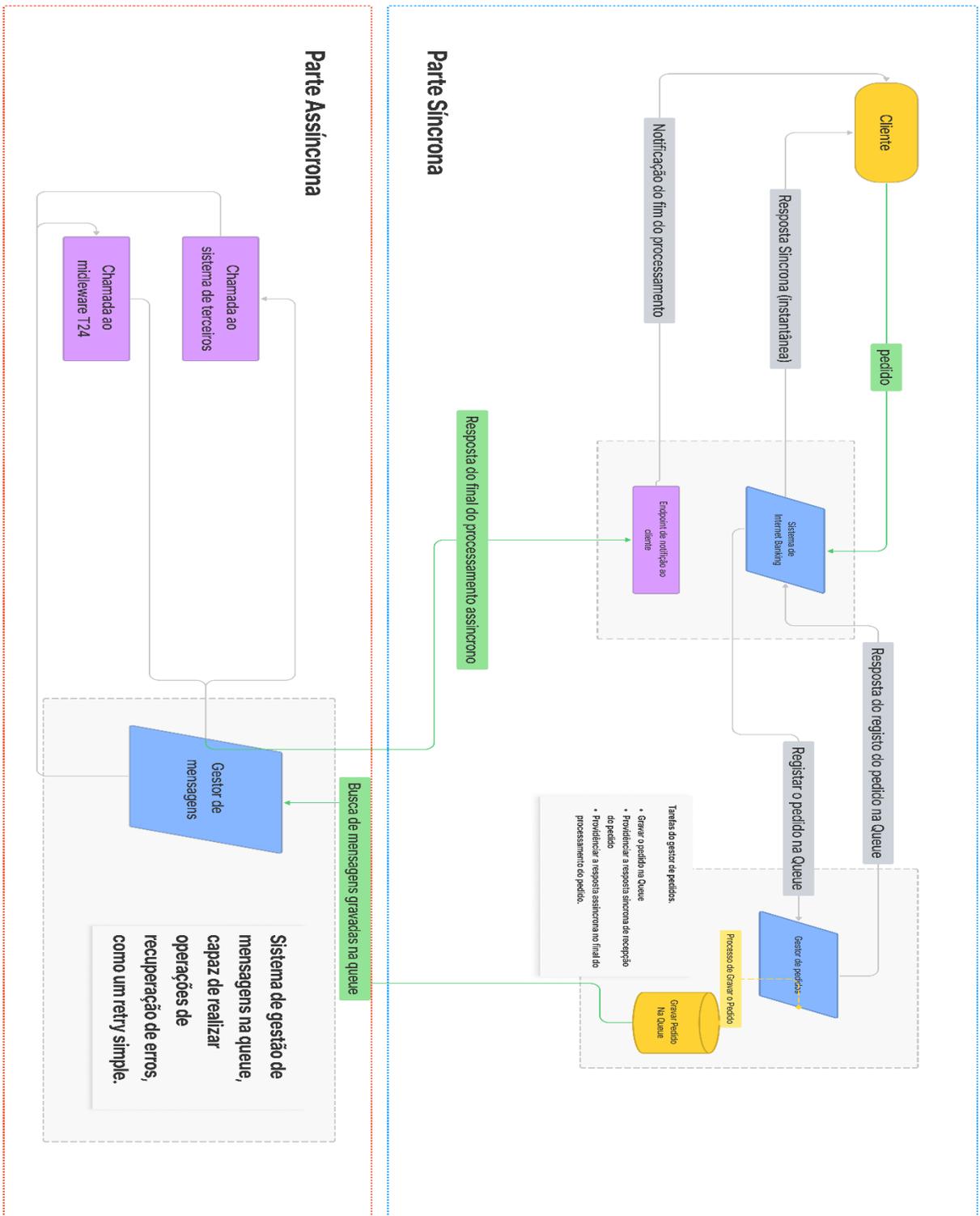


Figura 4: Arquitectura do sistema de internet banking assíncrono

Fonte: Elaborada pelo autor

4.1.1 Motivos da escolha do modelo

4.1.1.1 Melhoria da disponibilidade

A utilização de um modelo híbrido de processamento de transacções em um sistema de internet banking pode aumentar significativamente a disponibilidade do sistema. Isso ocorre porque a parte síncrona do sistema é projectada para ser altamente disponível e tolerante a falhas, garantindo a capacidade de realizar transacções críticas a qualquer momento.

De acordo com a pesquisa de Lee et al. (2018), "o processamento assíncrono tem se mostrado como uma alternativa eficaz para melhorar a disponibilidade em sistemas distribuídos. Isso ocorre, pois, ao utilizar essa abordagem, é possível suportar falhas e atrasos de maneira mais eficiente, permitindo que o sistema continue a operar mesmo em caso de problemas" (Lee, Kim, & Kim, 2018, p. 1).

4.1.1.2 Capacidade de escalabilidade

Segundo a pesquisa de Li et al. (2016), "o processamento assíncrono tem uma capacidade de escalabilidade melhor do que o processamento síncrono devido à sua capacidade de suportar o aumento da carga de trabalho sem comprometer o desempenho do sistema" (Li, Zhang, & Li, 2016, p. 5). Eles também apontam que "o processamento assíncrono permite a distribuição de tarefas entre múltiplos nós, o que melhora a capacidade de processamento e a escalabilidade do sistema" (Li, Zhang, & Li, 2016, p. 6).

4.2 Descrição da solução

A solução do sistema de internet banking que usa o processamento híbrido, onde se seleccionam uma série de operações não críticas do sistema e essas operações não são processadas de forma exclusivamente síncrona. Neste modelo, o processamento assíncrono é baseado em uma fila alimentada pela parte síncrona do sistema. Por exemplo, quando um usuário realiza uma operação de pagamento de salários através do m-pesa, a parte síncrona do sistema registra a transacção na fila de processamento e notifica o usuário de que seu pedido foi submetido e está a aguardar o processamento.

A vantagem de usar uma fila é que permite que as transacções sejam processadas em momentos mais oportunos, sem falhar devido a alta demanda no momento do processamento ou devido à indisponibilidade ou instabilidade de um sistema de terceiros. Além disso, tendo as transacções na fila, é possível aplicar mecanismos de recuperação, como o reprocessamento simples.

A parte assíncrona do sistema é responsável por buscar as transacções na fila, processá-las e notificar o usuário sobre o estado de seu pedido de processamento. Em resumo, este modelo híbrido permite extrair o melhor de ambos os tipos de processamento, oferecendo flexibilidade e escalabilidade ao sistema de internet banking.

4.2.1 Parte síncrona

Na parte síncrona do processamento de transacções, o sistema deve ser capaz de fornecer uma resposta imediata e precisa ao cliente, a informar que a primeira etapa do processamento foi concluída com sucesso e que o pedido foi registrado e encaminhado para processamento posterior. Isso garante que o cliente tenha a tranquilidade de que a transacção foi registrada correctamente e será processada em momento oportuno.

É fundamental catalogar todas as operações não críticas, como o processamento de ficheiro de pagamento de salários ou consultas de extractos consolidados. Dessa forma, é possível identificar quais operações precisam ser processadas de forma síncrona e quais devem ser registradas no sistema de estãõ de filas para processamento posterior.

4.2.2 Parte assíncrona

Na parte assíncrona do processamento das transacções, o sistema deve ser capaz de providenciar uma resposta ao final de todo o processamento das transacções, isto deve ser feito através de um WEB SERVICE, em que o sistema que faz a gestão da fila, deve transmitir as informações sobre o pedido processado.

Em caso de falha de processamento, a parte assíncrona do sistema deve ter mecanismo de recuperação de erros, por exemplo, se a falha na transacção for causada por uma instabilidade na comunicação com um sistema de terceiros, como o m-pesa. O sistema deve ter um mecanismo de reprocessamento simples, com um tempo de espera de 5 segundos e um máximo de 4 tentativas. E no final do processamento, deve-se enviar uma notificação PUSH e actualizar o estado do pedido do cliente.

No caso em que o pedido não tenha sido processamento, este tem um período de vida máximo de 24 horas na fila. Assim sendo, a transacção vai ser revertida pelo sistema de reconciliação de contas em um período máximo de D+1(em até 24 horas após a transacção).

Tabela 1: Problemas do processamento síncrono que são resolvidos pelo processamento assíncrono

Processamento Síncrono	Processamento Assíncrono
Bloqueia a execução do programa até que a tarefa seja concluída	Permite que o programa continue a executar enquanto a tarefa é realizada
Consome mais recursos do sistema	Consome menos recursos do sistema
Tempo de resposta longo para tarefas demoradas	Tempo de resposta curto, mesmo para tarefas demoradas
Menor escalabilidade	Melhor escalabilidade

Fonte: elaborada pelo autor

4.8.1.1. Sistema de gestão de filas

Por uma decisão administrativa, foi determinado o uso do RabbitMQ como sistema de gestão de filas. O RabbitMQ é um software de gestão de mensagens de código aberto amplamente adoptado que fornece um mecanismo robusto e escalável para troca de mensagens assíncronas entre aplicativos. Ele é baseado no protocolo Advanced Message Queuing Protocol (AMQP) e oferece recursos avançados, como filas duráveis, roteamento flexível de mensagens e suporte a vários padrões de troca de mensagens.

Ao optar pelo RabbitMQ como sistema de gestão de filas, a organização busca aproveitar seus benefícios, como:

1. Armazenamento de transacções em tempo real: RabbitMQ pode ser usado para capturar e armazenar transacções em tempo real, permitindo que outros sistemas processem essas transacções de forma assíncrona, (Alvaro Videla e Jason J. W. Williams).
2. Análise de transacções: RabbitMQ pode ser usado como uma fonte de dados para análise de transacções, permitindo que os dados de transacção sejam processados e analisados em tempo real ou em lotes, (Gavin M. Roy).
3. Integração com outros sistemas: Kafka pode ser usado para integrar diferentes sistemas de processamento de transacções, permitindo a comunicação entre esses sistemas e a troca de dados de transacção, (Kreps et al., 2011).
4. Monitoramento de transacções: Kafka pode ser usado para monitorar transacções em tempo real, permitindo que os dados de transacção sejam processados e alertas sejam gerados em caso de problemas.

4.8.1.1.1 Motivos da escolha do RabbitMQ

Numa análise estratégica do sistema de gestão de filas para otimizar o processamento assíncrono de transacções no ambiente do Internet Banking, os técnicos e administradores do Banco Nedbank Moçambique foram apresentados a diversas opções. Após cuidadosa

avaliação e deliberação, a escolha recaiu sobre o RabbitMQ, uma solução de mensageria de código aberto amplamente reconhecida pela sua eficácia e flexibilidade.

A decisão de adoptar o RabbitMQ baseou-se em diversos factores que atendem às necessidades específicas e exigências do ambiente bancário online. A robustez da garantia de entrega, a durabilidade das mensagens, a capacidade de suportar diversos protocolos de comunicação, a administração intuitiva e o suporte abrangente a várias linguagens de programação foram critérios determinantes.

Alem disso, vários autores reconhecem as vantagens do RabbitMQ como um sistema de gestão mensagens distribuído e escalável. Videla e Roche (2012) enfatizam a capacidade do RabbitMQ de fornecer uma solução eficiente para o processamento de grandes volumes de mensagens. Eles destacam a confiabilidade e flexibilidade do RabbitMQ, que inclui recursos como confirmações de entrega, *ré* enfileiramento automático e suporte a vários protocolos de gestão mensagens.

Alvaro et al. (2013) mencionam o RabbitMQ como uma opção viável para o processamento de dados centrado em eventos. Eles destacam a capacidade do RabbitMQ de lidar com fluxos de eventos em tempo real, permitindo a criação de pipelines de processamento de dados complexos e escaláveis.

4.8.2. Processamento de transacções orientado a transacções críticas e não críticas

A solução de uso de um modelo híbrido de processamento de transacções, com processamento prioritário, pode ser uma abordagem eficiente para suportar diferentes tipos de transacções. Nesse modelo, as transacções críticas continuam a ser processadas de forma síncrona, enquanto as transacções não críticas são direccionadas para um processamento assíncrono.

Segundo Johnson (2018, p. 45), Transacções críticas são aquelas que envolvem operações essenciais e de alto impacto para o funcionamento do sistema, cujo processamento e execução devem ocorrer de forma imediata e confiável. Por outro lado, transacções não

críticas referem-se a operações de menor relevância ou urgência, que podem tolerar atrasos ou falhas sem comprometer significativamente o desempenho ou a integridade do sistema.

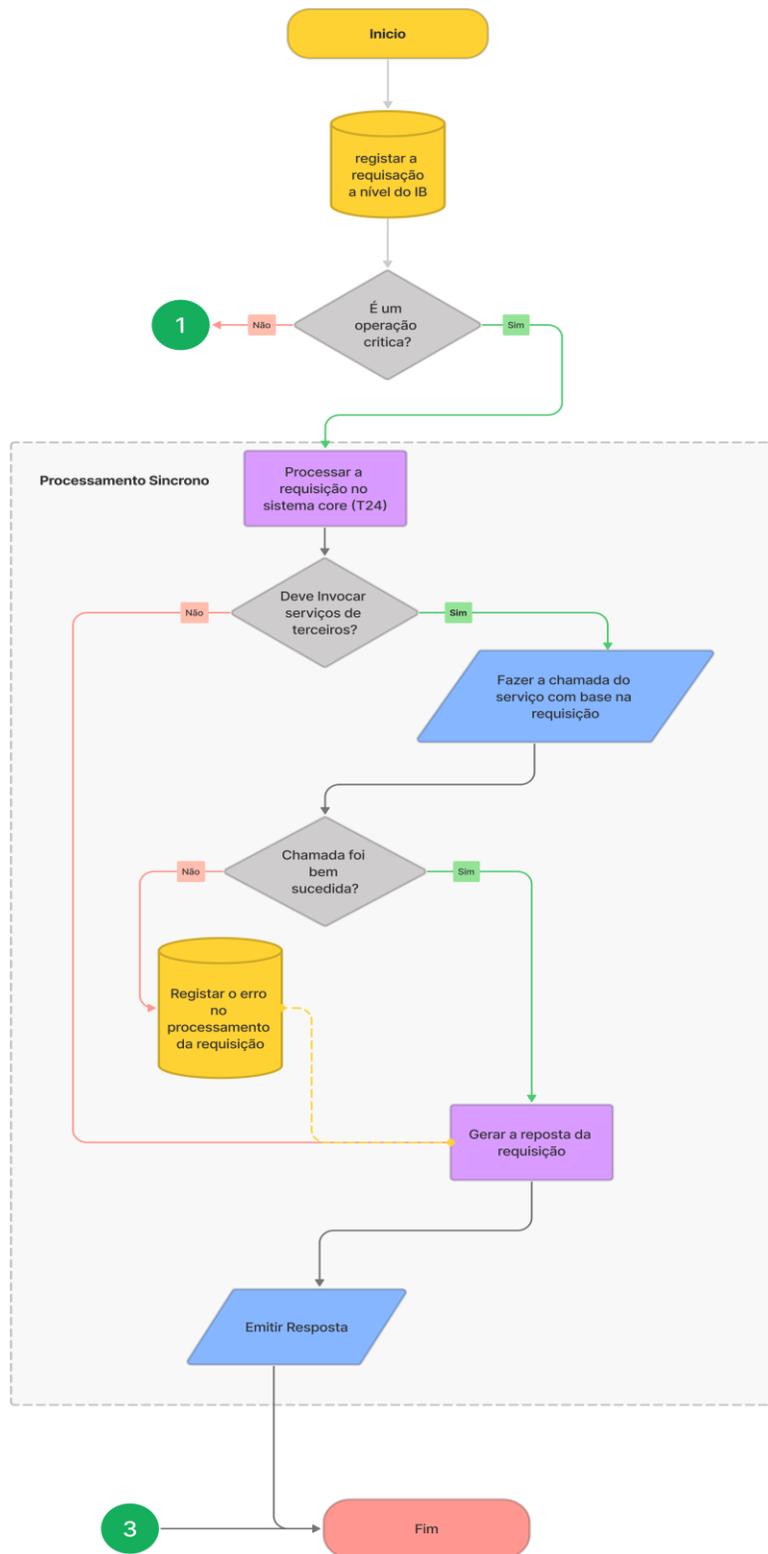


Figura 5: Fluxo de processamento síncrono

Fonte: elaborada pelo autor

No sistema de Internet Banking, existem operações críticas que devem ser executadas instantaneamente por meio de processamento síncrono. Isso ocorre devido à natureza sensível ao tempo dessas operações. Deste modo, a solução prevê uma segmentação das operações em críticas e não críticas. Assim o processamento deve ser feito com base no tipo de operação, e deste modo teremos operações que devem seguir o fluxo de processamento síncrono e outras que devem seguir o fluxo de processamento assíncrono.

Das operações críticas que devem ser processadas de forma síncrona, podemos destacar as seguintes:

1. Bloqueio de Cartões
2. Transferência para contas do mesmo banco
3. Alteração de senha
4. Transferência interbancária em tempo real
5. Pagamento de serviços em tempo real

As Operações em tempo real, podem ser feitas processadas de forma síncrona devido as regras de negocio. Onde dá-se a opção ao cliente de efectuar as suas operações em tempo real, entretanto os custos das comissões são maiores aos constrangimentos advindos da priorização destas operações.

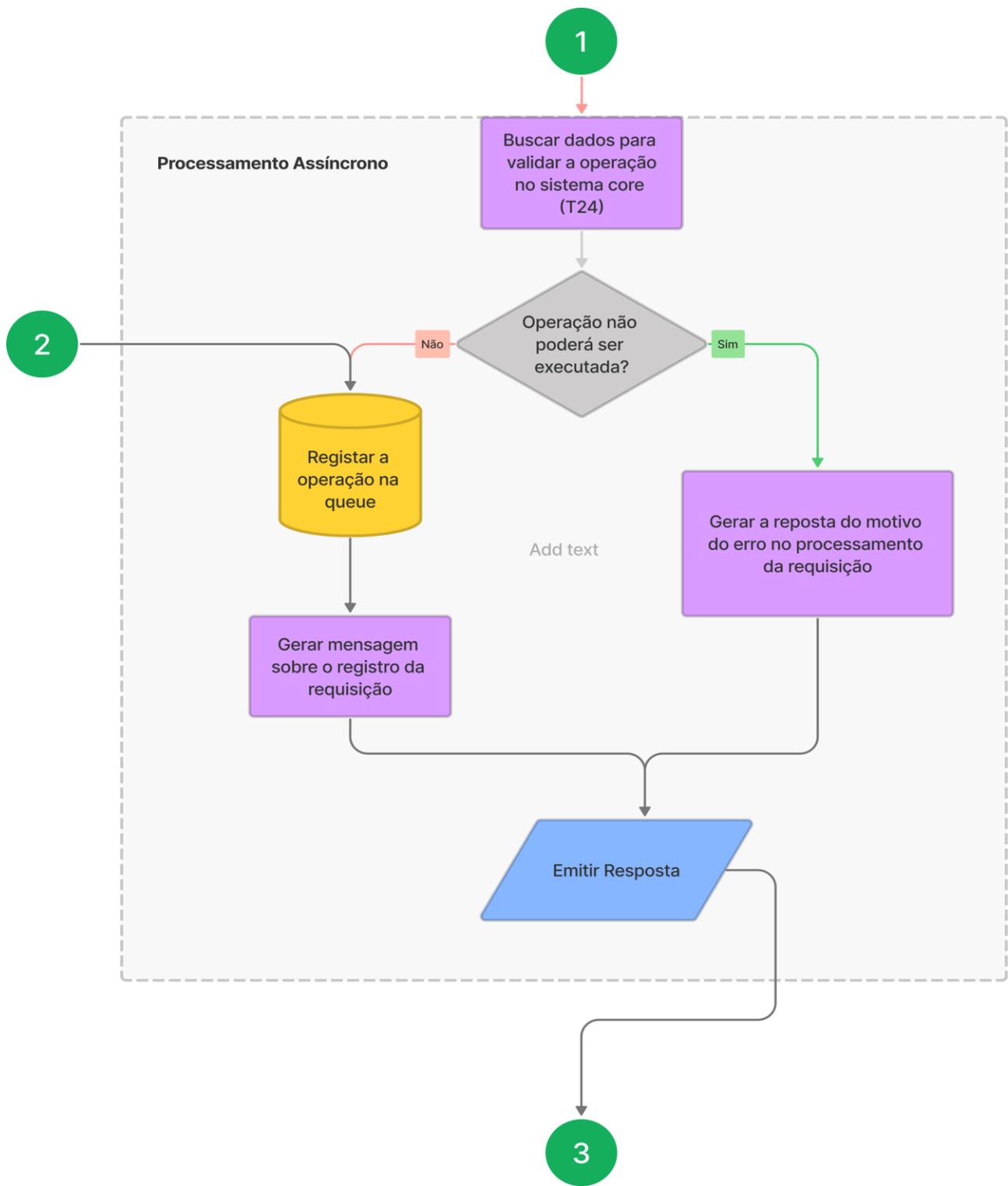


Figura 6: Fluxo de processamento assíncrono

Fonte: elaborada pelo autor

Além das operações críticas que exigem processamento síncrono, também existem operações não críticas que podem ser executadas de forma assíncrona. Essas operações geralmente envolvem consultas de informações não sensíveis ao tempo. Destas operações são destacadas as seguintes:

1. Solicitação de Extractos
2. Pagamento de salários
3. Pagamento em massa M-Pesa
4. Transferências Interbancárias
5. Pagamento de facturas de agua e energia.

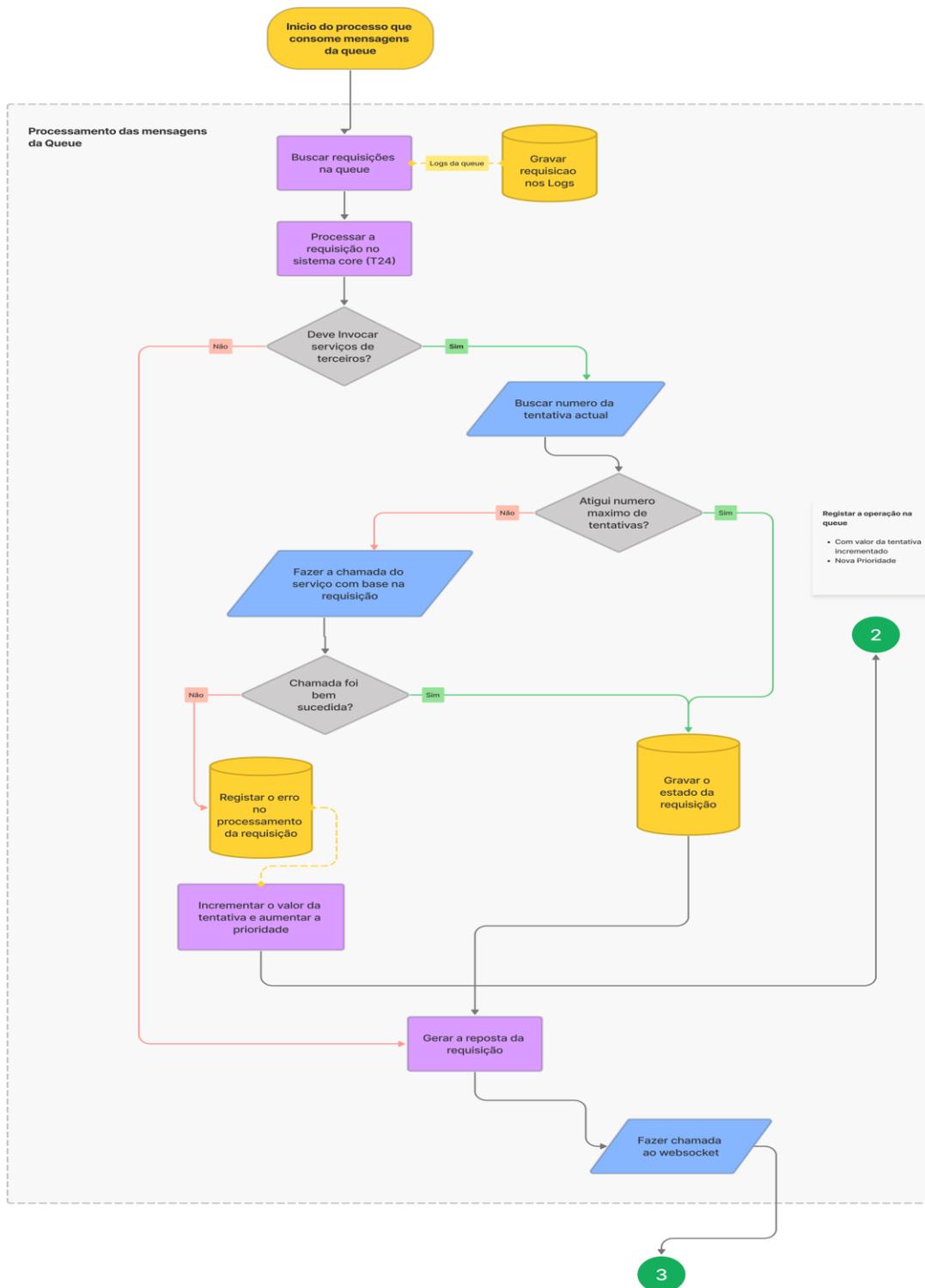


Figura 7: Fluxo de entrada e saída da fila de espera

Fonte: elaborada pelo autor

Quando há interacção com sistemas de terceiros, como serviços web, APIs, pode ocorrer falhas de comunicação, erros de conexão, problemas de tempo de resposta ou outros tipos de erros. Para garantir uma interacção robusta e confiável, é importante ter um mecanismo de recuperação de erros.

Esse mecanismo consiste em implementar uma estratégia para suportar os erros que podem ocorrer durante a interacção com os sistemas de terceiros. A abordagem adoptada é o reprocessamento automático, que envolve repetir automaticamente a tentativa de processamento em caso de falha.

1. Enviar a solicitação: é feita uma solicitação ao sistema de terceiros.
2. Processar a resposta: O sistema de terceiros processa a solicitação e retorna uma resposta.
3. Verificar se ocorreu um erro: A resposta é verificada para determinar se ocorreu algum erro ou falha durante o processamento.
4. Limite de tentativas: é definido um limite máximo de tentativas para reprocessamento.
5. Reprocessar em caso de erro: se ocorrer um erro, o fluxo de execução volta para a etapa de envio da solicitação, e o processo é repetido automaticamente.
6. Verificar limite de tentativas: em cada tentativa, é verificado se o limite máximo de tentativas foi atingido.
7. Finalizar com sucesso ou erro: se o processamento for bem-sucedido dentro do limite de tentativas, o fluxo de execução é finalizado com sucesso. Caso contrário, o fluxo é finalizado com uma indicação de erro.

5. Capítulo V – Implementação da solução de processamento assíncrono no sistema de internet banking

Neste capítulo, abordam-se aspectos técnicos e a metodologia utilizada para a elaboração de uma solução que possa resolver os constrangimentos identificados e descreve-se a solução proposta.

5.1 Implementação das filas de Mensagens

A implementação das filas de mensagens foi o primeiro passo na transição para o processamento assíncrono. Deste modo, a migração das funcionalidades do sistema síncrono, foi feita para aproveitar o ecossistema existente, e somente implementado as chamadas assíncronas, como mostrado na proposta de solução. Nesta seção, apresentamos em detalhes como as filas de mensagens foram configuradas, incluindo a escolha da tecnologia de fila, a criação de filas específicas para diferentes tipos de transações e a configuração de políticas de retenção e expiração.

5.1.1 Criação de Filas Específicas

Para otimizar o processamento das transações e garantir que cada tipo de transação seja tratado de forma eficiente, foram criadas filas de mensagens específicas para diferentes categorias de transações. Isso nos permite direcionar as transações para as filas apropriadas com base em suas características e prioridades. Por exemplo, transações críticas, como transferências de grandes quantias, são roteadas para filas de alta prioridade, enquanto transações menos críticas, como consultas dos movimentos, são encaminhadas para filas de baixa prioridade.

Exemplo: Criação da fila das transações feitas por ficheiro de pagamento de salario ou múltiplas transações.

Pagination

Page of 0 - Filter: Regex [?](#)

... no queues ...

▼ Add a new queue

Type:

Name: *

Durability:

Arguments: =

Add [Auto expire](#) [?](#) | [Message TTL](#) [?](#) | [Overflow behaviour](#) [?](#)
[Single active consumer](#) [?](#) | [Dead letter exchange](#) [?](#) | [Dead letter routing key](#) [?](#)
[Max length](#) [?](#) | [Max length bytes](#) [?](#)
[Leader locator](#) [?](#)

Figura 5: Exemplo da configuração da fila de múltiplas transacções para a Carteira móvel M-pesa

Fonte: elaborada pelo autor

Configuramos políticas de retenção e expiração nas filas de mensagens para controlar o tempo que as mensagens permanecem nas filas. Isso tornou-se uma regra de negocio essencial para gerir a capacidade das filas e evitar o acúmulo de mensagens não processadas. As políticas de retenção definem por quanto tempo uma mensagem deve ser mantida na fila antes de ser removida automaticamente. Além disso, as políticas de expiração determinam o tempo máximo que uma mensagem pode permanecer na fila antes de ser descartada. Essas políticas ajudam a manter o sistema eficiente e garantem que as mensagens sejam processadas dentro dos prazos desejados. Como por exemplo, configurou-se um tempo limite de 18 horas para uma transacção de carteira móvel permanecer na fila antes desta ser descartada por impossibilidade do sistema processa-la.

5.2 Implementação dos *producers* e dos *consumers*

Os *producers* e *consumers* desempenham um papel fundamental na comunicação com as filas de mensagens. Nesta seção, explora-se como os *producers*, responsáveis por enviar mensagens para as filas, foram implementados no contexto do Internet Banking. Além disso, abordaremos a implementação dos *consumers*, que consomem as mensagens da fila e

executam o processamento das transacções. Discutiremos a escalabilidade dos *consumers* para lidar com o aumento da carga de transacções e como eles lidam com mensagens em transacções críticas e não críticas.

Os produtores foram colocados responsáveis por enviar mensagens para as filas de mensagens, que notifica o sistema sobre eventos relevantes que exigem processamento. No contexto do Internet Banking, os produtores podem ser vistos como os pontos de entrada para transacções e eventos que ocorrem quando um cliente realiza uma acção, como uma transferência de fundos ou solicitação de extracto.

```
29     data = request.get_json()
30     input_ThirdPartyReference = data.get('input_ThirdPartyReference')
31
32     channel.basic_publish(
33         exchange = "Mpesa",
34         routing_key = [REDACTED],
35         body = json.dumps(data),
36         properties = pika.BasicProperties(
37             delivery_mode = 2
38         )
39     )
40
```

Figura 6: Produtor da fila de transacções do M-pesa

Fonte: Elaborado pelo autor

A implementação dos produtores envolveu a criação de interfaces e componentes em nosso sistema que geram mensagens e as encaminham para as filas de mensagens adequadas. Estas mensagens contêm informações sobre a transacção a ser processada e podem conter metadados adicionais, como prioridade e prazo de validade. Além disso, implementamos mecanismos de segurança rigorosos para autenticar e autorizar os produtores, garantindo que apenas fontes confiáveis possam publicar mensagens nas filas.

Os consumidores ficaram responsáveis por retirar mensagens das filas de mensagens e executar o processamento das transacções associadas. Eles desempenham um papel fundamental na escalabilidade e eficiência do sistema, permitindo que múltiplas transacções sejam processadas simultaneamente em servidores distribuídos. No contexto do Internet

Banking, os consumidores representam a força de trabalho que executa as operações bancárias, como validar transacções, actualizar saldos e gerar comprovantes.

```
23     )
24
25     channel = pika.BlockingConnection(connection_parameters).channel()
26     channel.queue_declare(
27         queue=self.__queue,
28         durable=True
29     )
30     channel.basic_consume(
31         queue=self.__queue,
32         auto_ack=True,
33         on_message_callback=self.__callback
34     )
35
36     return channel
37
38     def start(self):
39         print(f'Listen RabbitMQ on Port 5672')
40         self.__channel.start_consuming()
41
```

Figura 7: Consumidor da fila de transacções M-pesa

Fonte: Elaborada pelo autor

A implementação dos consumidores envolveu a criação de lógica de negócios específica para processar as transacções que chegam às filas. Os consumidores monitoram continuamente as filas de mensagens para detectar novas mensagens e iniciam o processamento assim que uma mensagem relevante é encontrada. É fundamental garantir que os consumidores sejam altamente escaláveis e que possam lidar com grandes volumes de transacções, especialmente em momentos de alta demanda.

Para optimizar o processamento das transacções, implementamos estratégias de gestão de filas e prioridades. As filas são configuradas para garantir que transacções críticas tenham prioridade sobre as transacções menos urgentes. Isso significa que os consumidores dedicarão mais recursos às transacções que podem afectar significativamente a disponibilidade e a integridade do serviço, como transferências de alto valor. Além disso, implementamos mecanismos de controle de concorrência para garantir que as transacções

não sejam processadas mais de uma vez e que o sistema mantenha a consistência dos dados.

5.2.1 Tratamento de Transacções Críticas e Não Críticas

Um aspecto importante da implementação dos consumidores é a capacidade de distinguir entre transacções críticas e não críticas. As transacções críticas como estabelecido da proposta de solução, são aquelas que têm um impacto imediato na disponibilidade e na integridade do serviço, enquanto as transacções não críticas podem ser processadas com menos prioridade. Implementamos lógica para garantir que as transacções críticas sejam processadas com a máxima prioridade, enquanto as transacções não críticas são processadas em segundo plano, garantindo a eficiência global do sistema. A implementação dos produtores e dos consumidores é fundamental para a funcionalidade do sistema de Internet Banking com processamento assíncrono. Eles garantem que as transacções sejam processadas de forma eficiente, escalável e confiável, contribuindo para a melhoria da disponibilidade do serviço. No próximo ponto, abordaremos a integração com os sistemas legados, um desafio crítico em nossa jornada de transformação.

5.3 Implementação com os sistemas legados

Nesta seção, apresenta-se como a solução de processamento assíncrono se conecta e interage com os sistemas legados do banco. Isso incluirá detalhes sobre a tradução de mensagens entre o formato usado pelas filas de mensagens e os formatos de dados usados pelos sistemas legados. Abordaremos também como as transacções são rastreadas e geridas ao atravessar as fronteiras dos sistemas legados.

Um dos principais aspectos da implementação envolveu a tradução de mensagens entre o formato usado pelas filas de mensagens e os formatos de dados utilizados pelos sistemas legados. Essa etapa é necessária para que as informações transmitidas através das filas possam ser compreendidas e processadas pelos sistemas legados e vice-versa.

Para realizar essa tradução, desenvolvemos adaptadores e transformadores de mensagens que convertem os dados de acordo com os padrões de comunicação de cada sistema. Isso

incluiu a conversão de formatos de dados, mapeamento de campos e até mesmo a implementação de protocolos de comunicação específicos para sistemas legados que não oferecem suporte nativo ao processamento assíncrono.

Ao passar entre os sistemas legados e o novo sistema de processamento assíncrono, as transacções podem envolver múltiplas etapas e componentes. Implementamos um sistema de rastreamento de transacções para acompanhar o progresso de uma transacção em todo o processo. Isso permite que identifiquemos e solucionemos problemas, se necessário, garantindo que as transacções sejam processadas correctamente e que os sistemas legados sejam actualizados com as informações relevantes.

Foram realizados testes extensivos de integração para garantir que a comunicação entre o novo sistema assíncrono e os sistemas legados seja confiável e eficaz. Isso incluiu testes de ponta a ponta que simularam cenários de uso real, bem como testes de regressão para verificar se as integrações com os sistemas legados não afectaram a funcionalidade existente.

5.4 Monitoramento e controle da Solução

Nesta seção, apresentam-se as estratégias de monitoramento e controle implementadas para o novo sistema assíncrono. Isso incluirá a criação de métricas de desempenho, a configuração de alertas de monitoramento em tempo real e a implementação de medidas proactivas para lidar com problemas potenciais.

Para avaliar o desempenho do sistema de processamento assíncrono, foram definidas e colectadas métricas relevantes. Isso inclui métricas como tempo de resposta médio, throughput (taxa de processamento de transacções), utilização de recursos (CPU, memória, armazenamento) e estatísticas de fila de mensagens. Essas métricas nos permitem monitorar o desempenho do sistema em tempo real e identificar tendências ao longo do tempo.

Foi implementado um sistema de alerta em tempo real para notificar a equipe de operações sobre eventos críticos ou anomalias no sistema. Esses alertas foram configurados com base

em thresholds predefinidos e métricas de desempenho, permitindo que a equipe de operações tome medidas imediatas para resolver problemas antes que afectem a disponibilidade do serviço.

Desenvolveram-se planos de continuidade de negócios que descrevem as acções a serem tomadas em caso de falhas críticas. Isso inclui a definição de procedimentos de recuperação, a implementação de pontos de verificação de dados e a configuração de ambientes de failover. O objectivo é garantir que o sistema possa ser restaurado rapidamente em caso de interrupções graves.

Para garantir que a solução de processamento assíncrono funcione de acordo com as expectativas, foram conduzidos testes abrangentes em vários cenários. Nesta seção, descreveremos a metodologia de teste usada, incluindo testes unitários, testes de integração e testes de carga. Apresentaremos os resultados desses testes e como eles ajudaram a validar a solução em termos de escalabilidade, desempenho e confiabilidade.

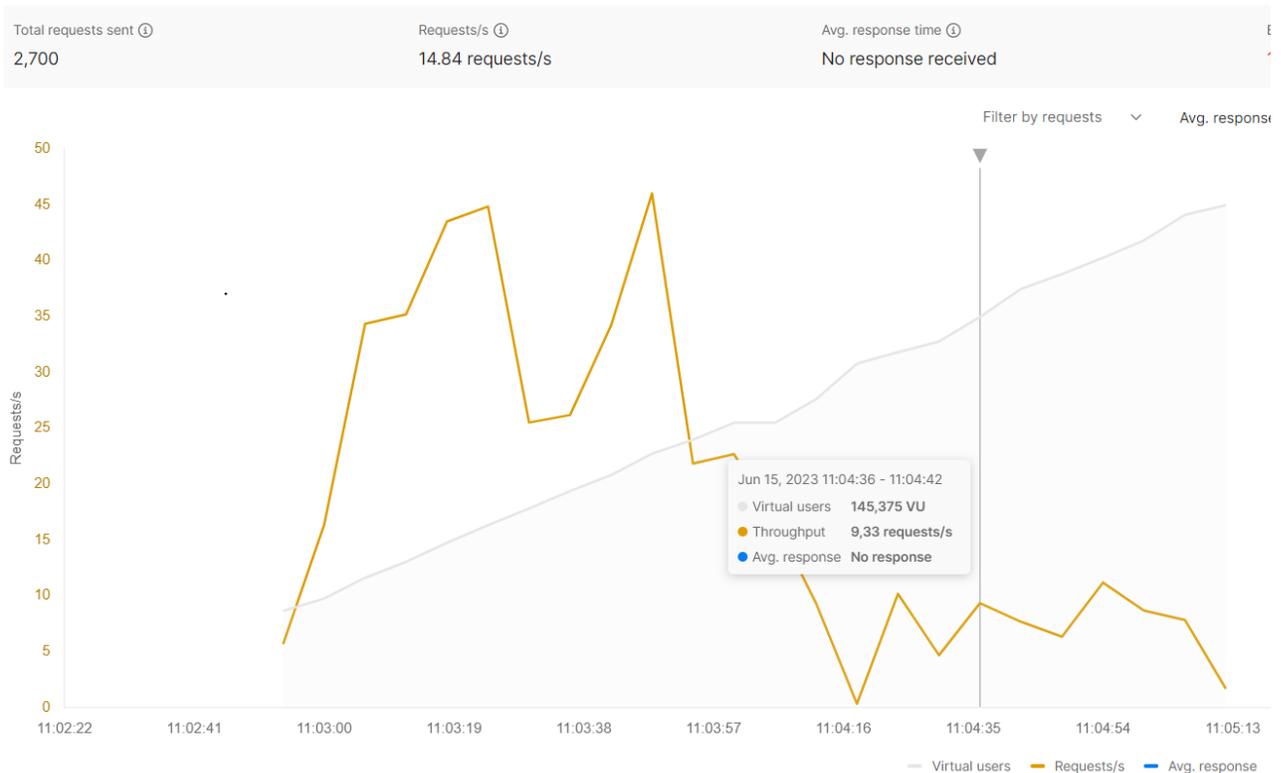


Figura 8: teste de carga feito ao endpoint de transferência móvel no modelo síncrono

Os testes de unidade foram conduzidos para avaliar a funcionalidade de cada componente individual da solução. Cada módulo do sistema, incluindo *producers*, *consumers*, adaptadores e transformadores de mensagens, é submetido a testes de unidade rigorosos. Os testes de unidade verificam se cada parte do sistema se comporta conforme o esperado e se está livre de erros de lógica ou programação.

Os testes de integração avaliam como os diferentes componentes da solução funcionam em conjunto. Isso inclui testes que verificam a comunicação adequada entre os *producers* e *consumers*, a interacção com as filas de mensagens e a integração com sistemas legados. Os testes de integração são essenciais para garantir que todos os elementos do sistema trabalhem harmoniosamente.

E por fim, os testes de carga foram realizados para avaliar a capacidade de escalabilidade da solução e determinar seu desempenho sob cargas pesadas. Simulamos cenários de uso em que um grande número de transacções é processado simultaneamente. Esses testes ajudam a identificar gargalos de desempenho, avaliar a capacidade de dimensionamento automático e garantir que o sistema possa lidar com a demanda em momentos de pico.

5.4.1 Validação de Requisitos de Negócios

Para validar os requisitos de negócios de modo a garantir que a solução atenda às expectativas dos stakeholders. Foi feita uma revisão dos requisitos originais do projecto e a realização de testes de aceitação com base nessas especificações. A validação de requisitos de negócios é fundamental para assegurar que a solução entregue os benefícios esperados para a organização.

Para além disso foi feito um inquérito com os utilizadores do sistema de internet banking, de modo a avaliar a experiência do utilizador com o novo modelo de processamento assíncrono. Desta avaliação, pretendeu-se verificar se as novas regras de negócio como o tempo que um pedido pode ficar na fila, o tempo limite de notificação de transacção entre outros, afectam ou não negativamente a experiência do utilizador.

5.5 Testes para comparação entre o desempenho da solução assíncrona com o síncrono

Foram postas a teste métricas, como tempo de resposta, throughput e disponibilidade, para demonstrar os benefícios e as melhorias obtidas com a implementação do processamento assíncrono. Isso permitirá avaliar o sucesso da nossa abordagem em melhorar a disponibilidade do serviço de Internet Banking.

Este capítulo é fundamental para entender como a implementação da solução de processamento assíncrono foi realizada e como essa mudança afectou o sistema de Internet Banking em termos de desempenho e eficiência. Os resultados aqui apresentados serão cruciais para as conclusões e recomendações finais do trabalho.

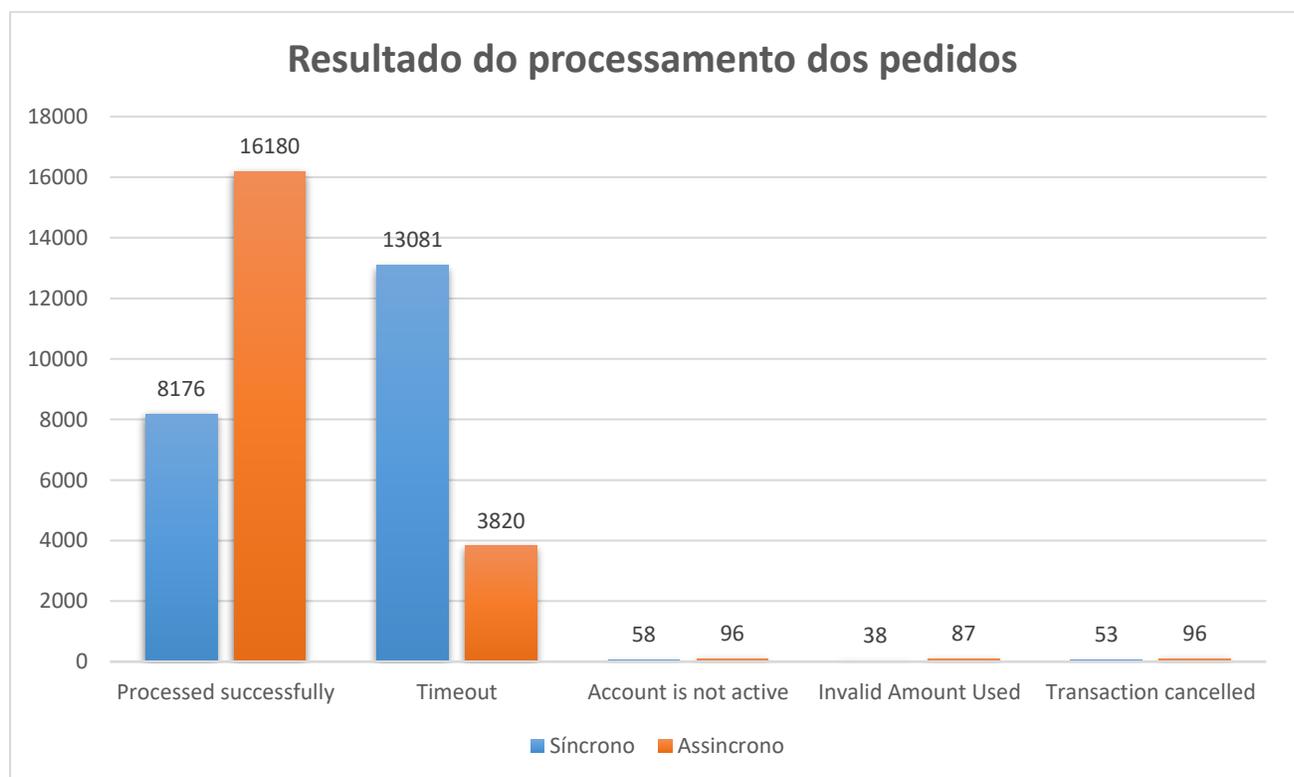


Gráfico 1: Comparação dos resultados (estados dos pedidos) no processamento assíncrono e síncrono.

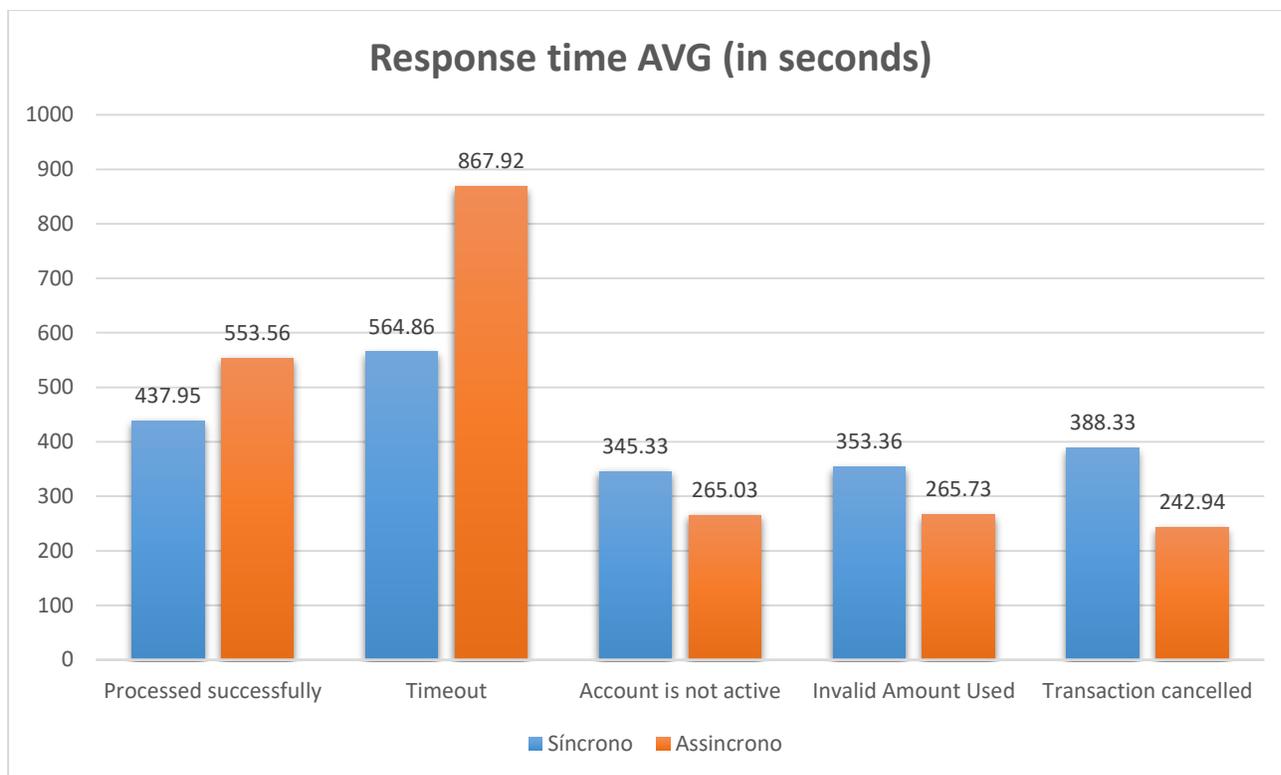


Gráfico 2: Comparação os tempos de respostas entre o processamento assíncrono e síncrono.

Feitos os testes do processamento síncrono e assíncrono, dois gráficos significativos emergiram para ilustrar a dicotomia entre essas abordagens e seus impactos cruciais nas métricas de desempenho.

O primeiro gráfico, representativo do processamento assíncrono, revela um aumento notável no número de transacções bem-sucedidas. Isso se deve ao fato de que no processamento assíncrono, as solicitações são enviadas para execução sem a necessidade de aguardar a conclusão de cada uma. Isso permite que o sistema processe várias solicitações simultaneamente, que resulta em um maior número de transacções bem-sucedidas em um período de tempo definido. No entanto, é crucial observar o aumento correspondente no tempo de resposta. Esse aumento é uma consequência da demora no processamento das solicitações da fila. Quando uma solicitação é enviada, ela entra em uma fila e aguarda sua vez para ser executada. Em situações de alta carga, essa fila pode crescer substancialmente, que resulta em atrasos perceptíveis.

Por outro lado, o segundo gráfico, que representa o processamento síncrono, exibe uma menor quantidade de transacções bem-sucedidas em comparação com o assíncrono. Isso ocorre porque no processamento síncrono, cada solicitação é processada sequencialmente, que aguarda a conclusão da anterior. Como resultado, o tempo necessário para cada transacção é relativamente curto e previsível, o que leva a um tempo de resposta mais curto e consistente.

A escolha entre o processamento síncrono e assíncrono dependeu das prioridades e necessidades específicas do sistema e das características estabelecidas pelos técnicos dos canais digitais do banco Nedbank Moçambique. O processamento assíncrono mostrou-se ideal quando se busca otimizar o número de transacções bem-sucedidas, mas por outro lado, resultou em tempos de resposta mais longos devido à espera na fila. Enquanto isso, o processamento síncrono é preferível quando se prioriza um tempo de resposta mais curto, mesmo que isso signifique um número menor de transacções bem-sucedidas.

Para avaliar o desempenho, foram colectadas métricas detalhadas de ambos os sistemas, síncrono e assíncrono. As métricas incluem tempo de resposta médio para transacções, throughput (taxa de processamento), escalabilidade sob carga pesada e utilização de recursos (CPU, memória e armazenamento). Essas métricas nos permitem comparar objectivamente o desempenho das duas abordagens.

Uma das principais vantagens da abordagem assíncrona é a redução do tempo de resposta para transacções críticas. Ao processar transacções assincronamente, conseguiu-se priorizar e acelerar o processamento de transacções de alto valor e alta prioridade. Isso resulta em tempos de resposta significativamente menores para transacções críticas, para melhorar a experiência do cliente.

A implementação assíncrona também se mostrou mais eficiente em termos de recursos de hardware. Os testes demonstraram que, sob cargas pesadas, o sistema assíncrono utiliza menos recursos, como CPU e memória, em comparação com o sistema síncrono. Isso não

apenas economiza custos de infra-estrutura, mas também reduz o risco de sobrecarga do hardware, o que pode levar a falhas do sistema.

A abordagem assíncrona mostrou-se mais confiável e resiliente em face de falhas. A capacidade de processar transacções de forma independente e assíncrona reduz a probabilidade de falhas em cadeia que podem ocorrer na abordagem síncrona. Além disso, os testes de recuperação de desastres confirmaram que a solução assíncrona é mais robusta em situações adversas.

6. Capítulo VI – Discussão de Resultados

Neste capítulo apresentam-se os resultados obtidos na realização do trabalho e como a solução proposta pode reduzir os constrangimentos identificados, através da análise da revisão de literatura, caso de estudo e a proposta de solução.

6.1. Revisão de literatura

O uso do processamento assíncrono de transacções é altamente recomendado devido às suas vantagens em termos de desempenho, escalabilidade, flexibilidade e resiliência.

De acordo com o estudo "Asynchronous Processing of Transactions" (Raza, 2018), o processamento assíncrono permite que as transacções sejam processadas de forma independente, sem que uma transacção precise esperar outra para ser concluída. Isso pode aumentar significativamente a velocidade de processamento, para melhorar a eficiência do sistema e a satisfação dos usuários. Além disso, o processamento assíncrono é escalável, pois permite que as transacções sejam processadas de forma paralela, permitindo suportar um volume maior de transacções sem comprometer o desempenho. Ademais, é mais flexível, pois permite que as transacções sejam processadas de forma independente, independentemente do tipo ou tamanho das transacções. E finalmente, é mais resistente a falhas, pois permite que as transacções sejam processadas de forma independente, de forma que uma falha em uma transacção não afecta a conclusão de outras transacções (Raza, 2018).

De acordo com Al-Fawzan (2017), o uso de processamento assíncrono permite a redução do volume de transacções concorrentes, o que pode melhorar significativamente a performance dos sistemas de internet banking. Li, Wang & Li (2018) também destacam que o processamento assíncrono é imune a problemas de concorrência, o que é fundamental para garantir a disponibilidade dos sistemas.

6.2. Caso de estudo

Diante da existência de várias situações que podem causar problemas de indisponibilidade no sistema de internet banking, como rotinas pouco otimizadas no sistema core T24, é crucial migrar parte das operações não críticas para o modelo de processamento assíncrono, para aliviar a concorrência e reduzir o volume de transacções. Esses problemas foram identificados em uma entrevista com o Head of Digital Channels Development do banco Nedbank Moçambique, Paulo Zacarias.

É importante destacar que as operações não críticas, como consulta de extracto consolidado, pagamentos de contas (água, Credelec, recargas telefone, e outros), processamento de ficheiro de salário, bloqueio/desbloqueio de cartão de crédito e transferências entre contas próprias, representam 36% (trinta e seis porcentos) das cerca de 12.000 (doze mil) operações diárias feitas no internet banking. Além disso, o processamento de arquivo de pagamento de salário é a operação que mais consome recursos dos servidores, cerca de 36,4% (trinta e seis virgula quatro porcentos) dos recursos dos servidores (CPU, RAM, Espaço de disco ou memória) por mês são alocados a esta operação (encontre a especificação do servidor no anexo A2.1). A migração do modelo de processamento de operações, para o modelo de processamento assíncrono é esperada para aliviar o nível de uso dos servidores e, conseqüentemente, tornar o serviço de internet banking mais disponível para o usuário.

De acordo com o estudo realizado, foi constatado que o problema de erros nas transacções devido à indisponibilidade de sistemas de terceiros, como o M-PESA, pode ser facilmente superado com o uso de processamento assíncrono. Por exemplo, uma solicitação de transferência pode ficar em uma fila de pedidos até que o M-PESA esteja disponível

novamente. É possível configurar um tempo de espera entre cada tentativa de comunicação com o M-PESA e um tempo máximo de tentativa, quando atingido, é possível assumir que a transacção não irá ocorrer e é necessário fazer um rollback dessa transacção.

Ao utilizar o processamento assíncrono para gerir as comunicações com sistemas de terceiros, como o M-PESA, a instituição poderia evitar perdas significativas, como a perda de comissões de Doze mil meticais (12,000.00 MZN) sobre um total de quatrocentas (400) transferências que poderiam ser realizadas no dia 15 de Setembro de 2022 (trinta meticais por operação), quando ocorreu uma indisponibilidade do serviço M-PESA por 3:35 minutos, das 11:12 às 14:47. É importante notar que, além da indisponibilidade, todas as condições necessárias para a realização dessas transacções, como o saldo disponível nas contas do banco, contas não bloqueadas, tanto a conta M-PESA quanto a do banco e a liquidez da conta do banco no M-PESA(devido ao uso do modelo B2C), estavam presentes no momento da indisponibilidade do MPESA.

6.3. Desenvolvimento da solução proposta

Para alcançar a solução proposta, foi possível mediante a análise dos recursos disponíveis, examinados no capítulo da revisão de literatura, e também por meio da interacção com os colaboradores do Nedbank Moçambique. Em termos gerais, por meio de entrevistas com os colaboradores do banco Nedbank Moçambique, foi possível colectar dados fundamentais para a identificação dos problemas enfrentados pela empresa, especialmente no sistema de internet banking.

O desenvolvimento do protótipo baseou-se no modelo de arquitectura apresentado no capítulo IV, para optar-se por realizar a migração incremental dos serviços críticos. Dessa forma, essa implementação foi concebida para ter baixo impacto no fluxo normal de operações e fácil integração.

Importa salientar que, ao final do desenvolvimento da solução, foi possível implementar todos os requisitos levantados, inclusive os requisitos desejáveis. Isso resultou em um sistema totalmente funcional, já implementado no Nedbank Moçambique. Desta maneira, todos os objectivos definidos neste trabalho foram alcançados.

7. Capítulo VII – Considerações finais

Neste capítulo, verifica-se o cumprimento dos objectivos inicialmente propostos para obter um modelo de arquitectura do sistema de internet banking, de modo a minimizar os problemas identificados. No fim deixa-se recomendações para os próximos pesquisadores.

7.1. Conclusão

Na conclusão deste estudo, é possível observar que cada um dos objectivos traçados foi abordado de maneira meticulosa e eficiente, contribuindo para uma compreensão mais profunda e uma resolução eficaz do problema em questão.

O primeiro objectivo, que consistia em "Estudar a situação actual do problema de disponibilidade", foi alcançado por meio de entrevistas e análise minuciosa dos recursos disponíveis, conforme estabelecido na metodologia deste trabalho. Esse processo permitiu uma compreensão aprofundada das lacunas existentes na disponibilidade do sistema de internet banking do Nedbank Moçambique.

O segundo objectivo, "Estudo sobre processamento assíncrono, seus tipos e formas de implementação", foi abordado de forma sistemática, que explora os fundamentos teóricos do processamento assíncrono. Essa investigação proporcionou uma base sólida para a implementação subsequente no contexto específico do sistema em questão.

Ao prosseguir para o terceiro objectivo, "Identificar e comparar as ferramentas usadas no processamento assíncrono", foi possível seleccionar as ferramentas mais adequadas para a realidade operacional do Nedbank Moçambique. A comparação criteriosa dessas ferramentas permitiu uma escolha informada, que considera os requisitos específicos do projecto e as decisões tomadas pelo corpo administrativo do banco.

A implementação prática do processamento assíncrono, que constitui o quarto objectivo, foi conduzida com base nos tipos e formas de implementação identificados e comparados anteriormente. O desenvolvimento do protótipo seguiu o modelo de arquitectura apresentado no capítulo IV, que adopta uma abordagem de migração incremental para minimizar o impacto nas operações normais e facilitar a integração.

Finalmente, ao comparar o desempenho do sistema de internet banking com processamento assíncrono em relação ao sistema síncrono existente, foi possível avaliar de maneira objectiva os benefícios da implementação proposta. A solução desenvolvida não apenas atendeu aos requisitos levantados, incluindo os desejáveis, mas resultou em um sistema totalmente funcional, já implementado com sucesso no Nedbank Moçambique.

Em linhas gerais, os objectivos desse presente trabalho de relatório de estágio profissional foram alcançados, porque foi possível estudar a situação actual, encontrar os constrangimentos e/ou problemas enfrentados pela empresa Nedbank Moçambique, propor uma solução que melhore se adequue para resolução de todos os problemas no sistema de internet banking destacados neste trabalho.

7.2. Recomendações

Para melhorar a eficiência do processo e a satisfação dos usuários, é recomendável que o banco Nedbank Moçambique realize um estudo a fim de identificar as causas dos timeouts que ocorrem durante o processamento de arquivos de pagamento de salários com mais de 15000 transacções. O estudo deve incluir análise de desempenho do sistema, monitoramento de recursos e entrevistas com usuários afectados, para identificar possíveis problemas ou pontos de falha, e implementar soluções para corrigi-los ainda a nível do sistema core T24.

Para garantir a continuidade dos serviços aos clientes habituais, é necessário realizar uma campanha de divulgação eficaz da migração para a solução híbrida, para destacar que os pedidos não críticos passarão a ser processados de forma assíncrona, sem impactar o modo de funcionamento usual do sistema.

Recomenda-se por último, aos investigadores que tenham interesse neste tema, a utilização de diferentes mecanismos de recuperação de erro. Pois tendo as transacções na fila, existem inúmeras possibilidades de acções para evitar erros no processamento das transacções.

7.3. Constrangimentos

O constrangimento encontrado na realização do trabalho, foi o facto de que para mudar o sistema de internet banking para um modelo de processamento híbrido, é necessário fazer desenvolvimentos também a nível do sistema core T24. De modo a, adicionar o estado pendente nas transacções. Desta Forma, por exemplo, para um pedido de processamento de um ficheiro de pagamento de salários, o sistema Core (T24) deve ser capaz de colocar como cativo o valor total das transferências (com o estado da operação em espera) ate o final da transacção.

Bibliografia

Referências Bibliográficas

Adebayo, A. O., Oladokun, A. A., & Adeboye, T. A. (2018). Design and Implementation of a Queue Management System. *International Journal of Computer Science and Mobile Computing*, 7(1), 40-46.

Al-Fawzan, H. (2017). Asynchronous Processing in Internet Banking Systems. *Journal of Internet Banking and Commerce*, 22(1).

Al-Qutaish, R., Khalil, I., Al-Khasawneh, F., & Al-Nawayseh, S. (2018). User interface design for mobile applications: prototyping approach. *International Journal of Interactive Mobile Technologies (IJIM)*, 12(3), 21-31.

Alvaro, N., Condie, T., Conway, N., Hellerstein, J. M., & Elmeleegy, H. (2013). BOOM Analytics: Exploring Data-Centric, Declarative Programming for the Cloud. *Proceedings of the VLDB Endowment*, 6(14), 1930-1941.

Angelakopoulos, G., e Mihiotis, A. (3 de March de 2011). Internet banking: challenges and opportunities in the Greek banking sector. *Electron Commer Res*, pp. 297–319.

António, R. M. (2020). *Microserviços para Reconhecimento de Emoção em Música: Relatório de Projeto de Mestrado*. Escola Superior de Tecnologia de Tomar.

Babich, A. (2018). *Push notifications for web and mobile: engagement and retention made easy*. O'Reilly Media, Inc.

Coulouris, G., Dollimore, J., & Kindberg, T. (2018). *Distributed Systems: Concepts and Design (Fifth Edition)*. Addison-Wesley.

Gaurav, K., & Kumar, R. (2018). *The Role of Internet Banking in Financial Inclusion: Evidence from Developing Countries*. *Journal of International Business Research and Marketing*

Gbadeyan, R. A., e Akinyosoye, O. O. (July de 2011). *Customers' Preference For E – Banking Services: A Case Study of Selected Banks in Sierra Leone*. *Australian Journal of Business and Management Research*, pp. 108-116. Vol.1 No.4.

Goldenberg, M. (1997). *A Arte de Pesquisar: Como Fazer Pesquisa Qualitativa em Ciências Sociais*. Rio de Janeiro: Record.

Gray, P., & Reuter, A. (2001). *Transaction Processing in the 21st Century*. *Communications of the ACM*, 44(4), 39-44.

He, X., & Fan, C. (2020). *Research on Real-Time Data Processing and Analysis Method of Edge Computing in Internet of Things*. *Journal of Physics: Conference Series*, 1529(3), 032050.

Hoque, M. A., Hossain, M. A., Ahmed, A., & Anwar, R. (2018). *Analysis and Performance Evaluation of Message Brokers: Apache Kafka and RabbitMQ*. *International Journal of Computer Science and Network Security*, 18(2), 78-84.

Hua, J., Yang, X., & Chen, C. (2018). Performance Evaluation of Cloud Applications Based on Load Testing with Gradual Ramp Up. In Proceedings of the International Conference on Smart Cloud (pp. 203-213). Springer.

KPMG. (2019). The banking ecosystem: Enabling the digital revolution. KPMG International.

Kreps, J., Narkhede, N., & Rao, J. (2011). Kafka: a distributed messaging system for log processing. In Proceedings of the NetDB (pp. 1-10). ACM.

Kreps, J., Narkhede, N., Rao, J., & Xu, J. (2011). Kafka: a distributed messaging system for log processing. ACM Queue, 9(2), 1-8.

Lakatos, E. M., & Marconi, M. de A. (2003). Fundamentos de metodologia científica. Atlas.

Lee, J., Kim, Y., & Kim, J. (2018). A Study on the Improvement of Availability in Distributed Systems Using Asynchronous Processing. Journal of Computer Science and Technology, 13(4), 1-9.

Li, Y., Wang, Y., & Li, X. (2018). Research on Asynchronous Processing of Internet Banking Transactions. International Journal of Grid and Distributed Computing, 11(6), 25-30.

Newcomer, E. (2009). Understanding web services: XML, WSDL, SOAP, and UDDI. Indianapolis, IN: SAMS.

Raza, S. (2018). Asynchronous Processing of Transactions. Journal of Computer Science and Technology.

Schwaber, K., & Beedle, M. (2001). Agile software development with Scrum. Prentice Hall PTR.

Sosyura, D. (2016). The impact of electronic banking on the efficiency of commercial banks: Evidence from Europe. Journal of Banking & Finance, 69-74.

Sutherland, J. (2020). Scrum: The art of doing twice the work in half the time. Crown Business.

Videla, A., & Roche, J. (2012). RabbitMQ in Action: Distributed Messaging for Everyone. Manning Publications.

Yoon, C. (2010). Antecedents of customer satisfaction with online banking in China: The effects of experience. Computers in Human Behavior, pp. 1296–1304.

Zhang, Y., Wang, J., & Li, X. (2019). Research on Asynchronous Processing of Internet Banking Transactions. International Journal of Grid and Distributed Computing, 41-56.

Outras bibliografias

"A Study of Synchronous and Asynchronous Internet Banking Transactions: Evidence from India" by R. K. Saha, Journal of Internet Banking and Commerce, Vol. 19, No. 2, 2014. Disponível em: <http://www.arraydev.com/commerce/JIBC/19-2/saha.pdf> acedido em 10/01/2022

"An Analysis of Synchronous and Asynchronous Internet Banking Transactions" by J. J. Salonen and M. E. S. van der Heijden, International Journal of Bank Marketing, Vol. 31, No. 3, 2013. Disponível em: <https://www.emerald.com/insight/content/doi/10.1108/IJBM-12-2012-0120/full/html> acedido em 10/01/2022

"Synchronous and Asynchronous Internet Banking: An Exploratory Study" by J. J. Salonen and M. E. S. van der Heijden, International Journal of Bank Marketing, Vol. 25, No. 6, 2007. Disponível em: <https://www.emerald.com/insight/content/doi/10.1108/02652320710772939/full/html>, acedido em 11/12/2022

Anexos

Anexo 1: Guião de entrevista

Entrevista aos desenvolvedores do internet banking do banco Nedbank Moçambique.

A1.1. A Primeira Entrevista aos desenvolvedores do internet banking do banco Nedbank Moçambique.

1. Quais são os principais desafios enfrentados ao lidar com o volume de transacções e a concorrência no sistema de Internet Banking?

2. Existem planos para implementar um mecanismo de recuperação automática de transacções não realizadas devido à indisponibilidade de sistemas de terceiros no Internet Banking?

3. Quais seriam os possíveis benefícios para o banco Nedbank Moçambique se um mecanismo automatizado de recuperação de transacções fosse implementado?

4. Como o uso de processamento assíncrono de transacções poderia contribuir para melhorias na disponibilidade do sistema de Internet Banking?

5. Quais seriam os requisitos técnicos e de infra-estrutura necessários para implementar com sucesso o processamento assíncrono de transacções no sistema de Internet Banking?

6. Além da disponibilidade do sistema, existem outros benefícios potenciais do processamento assíncrono de transacções no Internet Banking?

7. Como a indisponibilidade do sistema de terceiros, como o M-Pesa, afecta a disponibilidade geral do sistema de Internet Banking?

8. Quais são os principais desafios enfrentados ao lidar com a dependência de sistemas de terceiros para a disponibilidade do Internet Banking?

9. Existe um plano de contingência para lidar com a indisponibilidade de sistemas de terceiros no Internet Banking?

10. Quais são os impactos negativos que a indisponibilidade de sistemas de terceiros pode ter nos usuários do Internet Banking?

- **Desenvolvedores do internet banking**

A entrevista foi ao head of Digital Channels Development do banco Nedbank Moçambique, Paulo Zacarias.

1.R: processar um grande volume de transacções e a concorrência é um desafio constante. A demanda por serviços bancários online está a aumentar, e isso coloca pressão sobre a capacidade do sistema em lidar com um grande número de transacções simultâneas. Além disso, a concorrência no sector bancário também está a aumentar, o que nos obriga a oferecer uma experiência rápida e confiável aos nossos clientes.

2.R: actualmente, nosso processo de reconciliação de contas é realizado manualmente e não temos um mecanismo automatizado para recuperar transacções não realizadas devido à indisponibilidade de sistemas de terceiros. No entanto, estamos constantemente a avaliar melhorias e a buscar maneiras de aprimorar a disponibilidade e a eficiência do nosso sistema.

3.R: A implementação de um mecanismo automatizado de recuperação de transacções traria benefícios significativos para o banco. Em primeiro lugar, permitiria uma resposta mais rápida aos clientes quando ocorrerem problemas de indisponibilidade de sistemas de terceiros. Isso melhoraria a satisfação do cliente e fortaleceria a confiança em nosso serviço de Internet Banking. Além disso, a recuperação automática de transacções não realizadas reduziria a carga de trabalho manual da equipe de reconciliação de contas, a liberar tempo e recursos para se concentrarem em outras actividades estratégicas.

4.R: O uso de processamento assíncrono de transacções pode ser uma solução eficaz para melhorar a disponibilidade do sistema de Internet Banking. Ao adoptar o processamento assíncrono, as transacções podem ser processadas independentemente e em paralelo, reduzindo a dependência de sistemas externos. Isso significa que, mesmo se houver indisponibilidade de um sistema de terceiros, as transacções ainda podem ser registradas e processadas internamente, a minimizar interrupções para os clientes. Além disso, o processamento assíncrono permite uma melhor gestão de erros e falhas, para facilitar a recuperação de transacções não realizadas de forma mais eficiente.

5.R: A implementação bem-sucedida do processamento assíncrono de transacções exigiria alguns requisitos técnicos e de infra-estrutura. Seria necessário um sistema robusto de mensageria ou filas de mensagens para gerir as transacções assíncronas. Além disso, seria importante ter uma infra-estrutura de rede confiável e escalável, que pudesse lidar com o aumento da carga de transacções e garantir a entrega segura das mensagens entre os sistemas envolvidos.

6.R: Sim, além da melhoria na disponibilidade do sistema, o processamento assíncrono de transacções também pode trazer outros benefícios. Por exemplo, ele pode permitir uma melhor gestão de picos de carga de transacções, para evitar a sobrecarga do sistema durante períodos de alta demanda. Além disso, o processamento assíncrono pode facilitar a implementação de novos recursos ou serviços, a tornar mais ágil o desenvolvimento e a integração de actualizações no sistema de Internet Banking.

7.R: A indisponibilidade do sistema de terceiros, como o M-Pesa, pode impactar significativamente a disponibilidade geral do nosso sistema de Internet Banking. Quando o M-Pesa enfrenta problemas de indisponibilidade, a funcionalidade relacionada a essa integração específica é afectada. Isso pode resultar na incapacidade dos utilizadores de realizar transacções ou acederem certos serviços, a prejudicar a experiência do cliente e diminuindo a disponibilidade do nosso Internet Banking como um todo.

8.R: Um dos principais desafios é a falta de controle directo sobre a disponibilidade e o desempenho dos sistemas de terceiros. Dependemos desses sistemas externos para

fornecer serviços essenciais aos nossos clientes. Qualquer problema em sua disponibilidade ou falha de comunicação pode afectar directamente a disponibilidade do nosso Internet Banking. Além disso, também enfrentamos o desafio de garantir uma integração estável e confiável com esses sistemas de terceiros, pois qualquer falha nessa integração pode resultar em problemas de disponibilidade.

9.R: sim, temos um plano de contingência estabelecido para fazer face a situação de indisponibilidade de sistemas de terceiros. Quando ocorre uma interrupção no sistema de terceiros, nossa equipe monitora activamente a situação e trabalha em estreita colaboração com os fornecedores desses sistemas para resolver o problema o mais rápido possível.

10.R: A indisponibilidade de sistemas de terceiros pode causar uma série de impactos negativos nos usuários do nosso Internet Banking. Os clientes podem enfrentar dificuldades para realizar transacções, como transferências de fundos ou pagamentos de contas, que dependem desses sistemas de terceiros. Além disso, a falta de acesso a serviços importantes, como a visualização de saldo em tempo real ou histórico de transacções, pode levar a uma experiência insatisfatória do cliente e à perda de confiança em nosso Internet Banking.

Anexo 2: Especificação técnica do servidor do IB

1. Processador:

- Modelo: Intel Xeon Gold ou equivalente
- Núcleos: 8 núcleos
- Frequência: 2,5 GHz
- Memória RAM:
- Capacidade: 32 GB de RAM
- Tipo: DDR4 ECC

2. Armazenamento:

- Disco Rígido: 2 TB de espaço em disco
- Tecnologia: SSD (Solid State Drive)

3. Sistema Operativo:

- Windows Server 2012
- Edição: Windows Server 2012 R2 Standard
- Licença: Versão licenciada com suporte actualizado

4. Rede:

- Portas de rede: 2 portas Gigabit Ethernet
- Protocolo de rede: IPv4 e/ou IPv6

5. Segurança:

- Firewall, Proxy e sistemas de detecção de intrusões (IDS/IPS)

6. Backup e Redundância:

- Backup regular e seguro para garantir a integridade dos dados do Internet Banking.
- Configuração de redundância para garantir a disponibilidade contínua do serviço, incluindo fontes de energia redundantes, discos rígidos em RAID e portas de acesso a internet múltiplas.

Anexo 3: Teste de carga do sistema de internet banking com processamento síncrono e assíncrono:

O teste de carga a usar ramp up oferece várias vantagens significativas na avaliação do desempenho de sistemas e aplicativos. Segundo Hua, Yang e Chen (2018), "o teste de carga com ramp up gradual apresenta benefícios importantes, como simulação realista de carga, detecção precoce de gargalos de desempenho e capacidade de identificar o limite de capacidade do sistema" (p. 205).

The screenshot displays a performance testing configuration interface. At the top, there are two tabs: 'Functional' and 'Performance', with 'Performance' being the active tab. Below the tabs is a notification box titled 'Test how your APIs perform under load' with a 'BETA' label and a close button. The main section is titled 'Set up your performance test' and contains several input fields and a 'Run' button. The 'Virtual users' field is set to 200. The 'Test duration' field is set to 60 minutes. The 'Load Profile' section has two radio buttons: 'Fixed' and 'Ramp up', with 'Ramp up' selected. The 'Ramp up duration' field is set to 3 minutes. To the right of these fields is a graph showing the load profile. The graph has a vertical axis labeled '200 VUs' and a horizontal axis labeled '0' and '60 mins'. The profile shows a linear increase from 0 to 200 VUs over the first 3 minutes, followed by a constant load of 200 VUs for the remainder of the 60-minute test. Below the graph is a 'Preview' section with the text: 'Gradually increase the number of parallel users to 200 in the first 3 minutes of the test.'

Figura 9: Perfil de testes de carga (ramp-up)

Fonte: elaborada pelo autor

Objectivo do teste de carga:

O objectivo principal do teste de carga era avaliar o desempenho e a capacidade dos sistemas de internet banking síncrono e assíncrono sob uma carga simulada de 200 usuários virtuais. A estratégia de ramp up foi empregada para aumentar gradualmente o número de usuários e observar o comportamento dos sistemas em diferentes níveis de carga.

Configuração do teste:

Número de usuários virtuais: 200

Estratégia de ramp up: os usuários virtuais foram adicionados gradualmente ao longo do tempo para simular um aumento progressivo na carga. A cada minuto, 10 usuários virtuais foram adicionados ao teste até atingir o número total de 200 usuários.

Cenários de teste: foram criados cenários de teste para simular as principais ações dos usuários, como login, registro de transações, consulta de transações e outras funcionalidades relevantes dos sistemas de internet banking.

Duração do teste: O teste de carga foi executado por um período específico, como uma hora, durante o qual os usuários virtuais realizaram as ações definidas nos cenários de teste.

Métricas e análise:

Durante o teste de carga, foram coletadas diversas métricas para avaliar o desempenho dos sistemas de internet banking. Algumas das métricas importantes podem incluir:

Tempo de resposta: mede o tempo necessário para que o sistema responda a uma solicitação de um usuário. Isso ajuda a avaliar a rapidez com que os sistemas processam as ações dos usuários.

Taxa de transferência: mede a quantidade de transações ou solicitações que o sistema é capaz de processar por unidade de tempo. Ajuda a determinar a capacidade de processamento dos sistemas.

Utilização de recursos: monitora o uso de recursos, como CPU, memória e largura de banda de rede, pelos sistemas durante o teste. Isso pode ajudar a identificar gargalos de desempenho.

Erros e falhas: registra qualquer erro ou falha que ocorra durante o teste. Isso permite identificar problemas e verificar a capacidade de recuperação dos sistemas.

Após a conclusão do teste de carga, os dados colectados foram analisados e comparados entre os sistemas de internet banking síncrono e assíncrono para identificar possíveis diferenças de desempenho, capacidade e comportamento sob carga. Essas descobertas foram utilizadas para embasar o trabalho de pesquisa e fornecer insights relevantes sobre os sistemas testados.

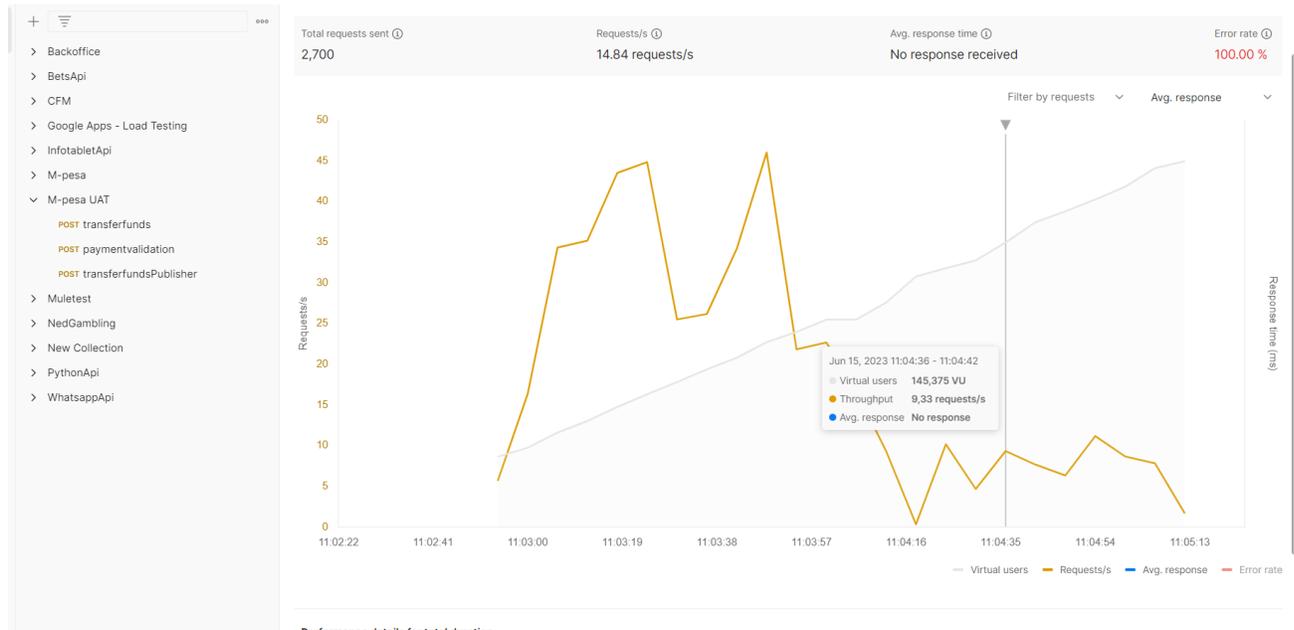


Figura 10: Gráfico da relação entre o numero de requisições e respostas

Fonte: elaborada pelo autor

Os resultados obtidos no teste de carga dos sistemas de internet banking síncrono e assíncrono com 200 usuários virtuais a utilizar a estratégia de ramp up proporcionaram insights valiosos sobre o desempenho e a capacidade dos sistemas. Aqui estão algumas observações importantes que podem ter surgido dos resultados:

Desempenho do sistema síncrono: durante o teste, o sistema de internet banking síncrono mostrou tempos de resposta mais rápidos em comparação com o sistema assíncrono. Isso pode ser devido à natureza síncrona das interações, em que as respostas são fornecidas imediatamente após a conclusão de uma acção.

Desempenho do sistema assíncrono: embora o sistema de internet banking assíncrono tenha mostrado tempos de resposta ligeiramente mais lentos, ele ainda foi capaz de lidar com a carga de 200 usuários virtuais de forma eficiente. A vantagem do sistema assíncrono é a capacidade de processar tarefas em segundo plano, permitindo que os usuários continuem a interagir com o sistema enquanto as transações estão sendo processadas.

Capacidade e escalabilidade: ambos os sistemas demonstraram boa capacidade de lidar com a carga de 200 usuários virtuais. A estratégia de ramp up gradual permitiu avaliar a capacidade de escalabilidade dos sistemas, a observar como eles se comportam à medida que mais usuários são adicionados. Verificou-se se os sistemas mantinham um desempenho aceitável, sem quedas significativas no tempo de resposta ou na taxa de transferência.

Estabilidade e recuperação: durante o teste, foram registrados quaisquer erros ou falhas ocorridas nos sistemas. Foi analisada a capacidade dos sistemas de se recuperarem de falhas e retornarem a um estado estável. A estabilidade dos sistemas é crucial para garantir uma experiência confiável para os usuários.

Optimizações futuras: com base nos resultados, podem ter sido identificadas áreas em que os sistemas podem ser otimizados. Isso poderia incluir melhorias na eficiência do processamento das transações, redução de tempos de resposta ou otimização do uso de recursos para lidar com uma carga maior.



Figura 11: Gargalo no processamento Síncrono

Fonte: elaborada pelo autor

Anexo 4: Mockups do sistema de internet banking

Segundo Al-Qutaish et al. (2018), a criação de um mockup de um sistema oferece várias vantagens significativas. Em primeiro lugar, um mockup permite visualizar e testar a interface do usuário antes do desenvolvimento completo do sistema. Isso permite identificar e corrigir problemas de usabilidade e fluxo de trabalho, garantindo que o sistema seja intuitivo e amigável para os usuários.

Além disso, um mockup permite a comunicação eficaz entre os membros da equipe de desenvolvimento, os usuários finais e outros stakeholders. Ele serve como um ponto de referência visual e tangível, para facilitar discussões e tomada de decisões. Dessa forma, todas as partes envolvidas podem ter uma compreensão comum do sistema, suas funcionalidades e requisitos.

Outra vantagem é a economia de tempo e recursos. Com um mockup, é possível realizar testes de conceito e validação de ideias de forma rápida e econômica, antes de investir

recursos significativos no desenvolvimento completo do sistema. Isso ajuda a identificar falhas ou necessidades de ajustes precocemente, para evitar retrabalho e custos adicionais no futuro.

Além disso, um mockup pode ser usado como uma ferramenta de prototipagem rápida para obter feedback dos usuários finais e stakeholders. Isso permite realizar iterações e melhorias no design do sistema com base nas opiniões e sugestões recebidas, para resultar em um produto final mais alinhado com as necessidades e expectativas dos usuários.



Figura 12: Tela de carregamento de ficheiro de pagamentos de salários

Fonte: elaborada pelo autor



GENERICOMOC

Meus pedidos | Gestão da conta

POSIÇÃO INTEGRADA

CONTAS

GESTÃO DE CARTOES

EMPRESTIMOS E DEPÓSITOS

CRÉDITO DOCUMENTARIO

GARANTIA BANCÁRIA

PROCESSAR FICHEIRO

SERVIÇOS

Processar ficheiro

Selecione o ficheiro

C:\Users\T22345\Documents\Files\1-12-2023

Processar ficheiro

Figura 13: Acção de submissão do ficheiro de pagamento de salários.

Fonte: elaborada pelo autor



Figura 14: Resposta síncrona, a confirmar o registo do pedido de processamento de ficheiro de pagamento de salários.

Fonte: elaborada pelo autor



GENERICOMOC

Meus pedidos(1) | Gestão da conta

POSIÇÃO INTEGRADA

CONTAS

GESTÃO DE CARTOES

EMPRESTIMOS E DEPÓSITOS

CRÉDITO DOCUMENTARIO

GARANTIA BANCÁRIA

PROCESSAR FICHEIRO

SERVIÇOS

Meus pedidos

Nr do pedido	Data de início	Data de processamento	Estado	Download
12764	1/3/2023	1/5/2023	Sucesso	Download
12767	1/6/2023	1/7/2023	Sucesso	Download
12798 (New)	1/10/2023	1/13/2023	Sucesso	Download
12903	1/15/2023	-/--/----	Em espera	Download
12904	1/15/2023	-/--/----	Em espera	Download
12915	1/16/2023	-/--/----	Em espera	Download

Figura 15: Tabela com os estados dos pedidos, já contendo os pedidos que foram processados de forma assíncrona.

Fonte: elaborada pelo autor

Anexo 5: Caso de uso do sistema de internet banking síncrono

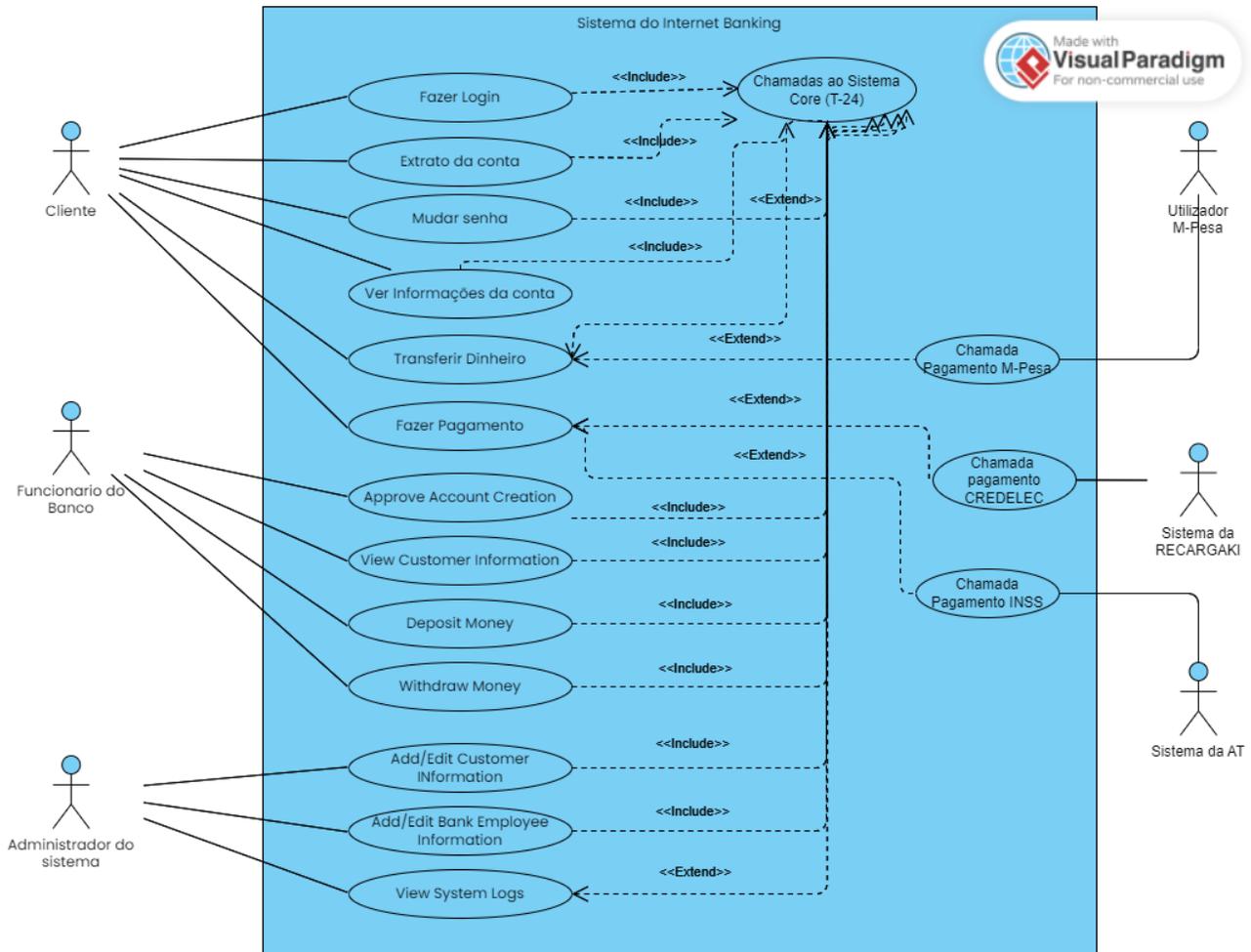


Figura 16: Caso de uso do sistema de internet banking síncrono

Fonte: elaborada pelo autor

1. O cliente aceder o sistema de internet banking síncrono.
2. O sistema exhibe o painel do usuário com opções de registro de transacções.
3. O cliente selecciona a opção de registrar uma nova transacção.
4. O sistema exhibe um formulário de registro de transacção com campos para inserir os detalhes, como valor, destinatário e descrição.
5. O cliente preenche os campos necessários e envia o formulário.
6. O sistema valida os dados inseridos pelo cliente.

7. O sistema processa a transacção imediatamente, a verificar a disponibilidade de fundos e para executar a transferência.
8. O sistema exibe uma mensagem síncrona ao cliente, a confirmar o registro e a conclusão bem-sucedida da transacção.
9. O caso de uso é encerrado.

Fluxo alternativo:

No passo 6, se os dados inseridos forem inválidos, o sistema exibe uma mensagem de erro ao cliente e solicita que as informações sejam corrigidas.

No passo 7, se houver algum problema durante o processamento da transacção, o sistema exibe uma mensagem de erro ao cliente, a informar o motivo e sugerindo acções correctivas.

Pós-condição: O cliente recebe uma confirmação síncrona do registro e da conclusão bem-sucedida da transacção.

Esse caso de uso é apenas um exemplo e pode ser expandido para incluir outros recursos ou etapas, dependendo dos requisitos específicos do sistema de internet banking síncrono.

Anexo 6: Caso de uso do sistema de internet banking assíncrono

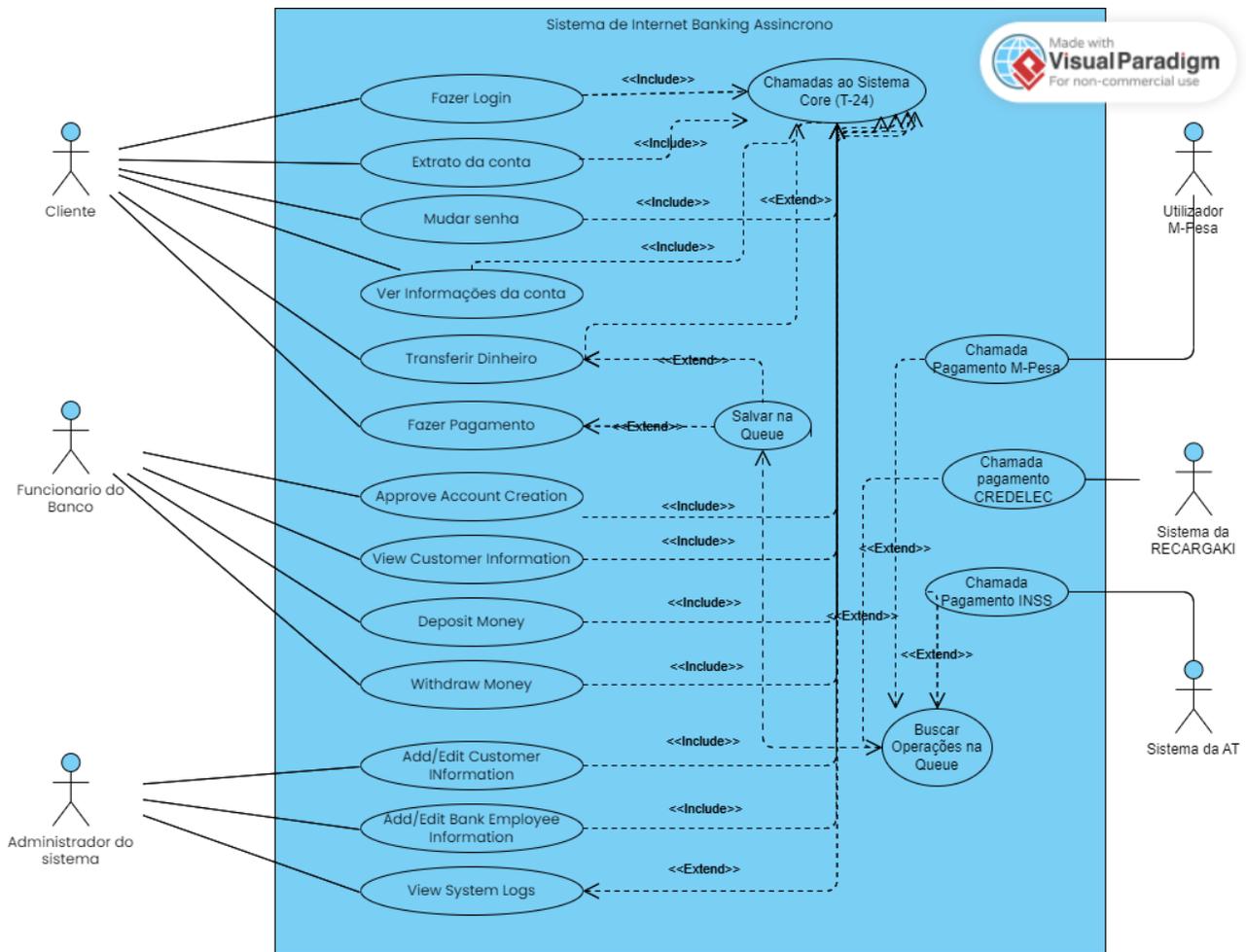


Figura 17: Caso de uso do sistema de internet banking assíncrono

Fonte: elaborada pelo autor

1. O cliente aceder o sistema de internet banking assíncrono.
2. O sistema exibe o painel do usuário com opções de registo de transacções.
3. O cliente selecciona a opção de registar uma nova transacção.
4. O sistema exibe um formulário de registo de transacção com campos para inserir os detalhes, como valor, destinatário e descrição.
5. O cliente preenche os campos necessários e envia o formulário.
6. O sistema valida os dados inseridos pelo cliente.
7. O sistema regista a transacção e atribui um número de identificação único a ela.

8. O sistema envia uma resposta assíncrona ao cliente, a confirmar o registro da transacção. A resposta pode ser exibida no painel do usuário ou enviada por e-mail, mensagem de texto ou outro meio de comunicação.
9. O caso de uso é encerrado.

Fluxo alternativo:

No passo 6, se os dados inseridos forem inválidos, o sistema exibe uma mensagem de erro ao cliente e solicita que as informações sejam corrigidas.

No passo 8, se houver algum problema durante o registro da transacção, o sistema envia uma resposta assíncrona ao cliente a informar o erro ocorrido.

Pós-condição: O cliente recebe uma confirmação assíncrona do registro da transacção.

Esse caso de uso é apenas um exemplo e pode ser expandido para incluir outros recursos ou etapas, dependendo dos requisitos específicos do sistema de internet banking assíncrono.

Anexo 7: Email de notificação da conclusão do processamento da requisição feita no sistema de internet banking

Notificações por email são amplamente utilizadas para informar os usuários sobre a conclusão do processamento assíncrono de transacções. Essas notificações são enviadas automaticamente assim que o processamento é finalizado, garantindo que os usuários sejam informados em tempo hábil. Na imagem a seguir, apresento um modelo de email que é enviado o utilizador sempre que o processamento do seu pedido é concluído.

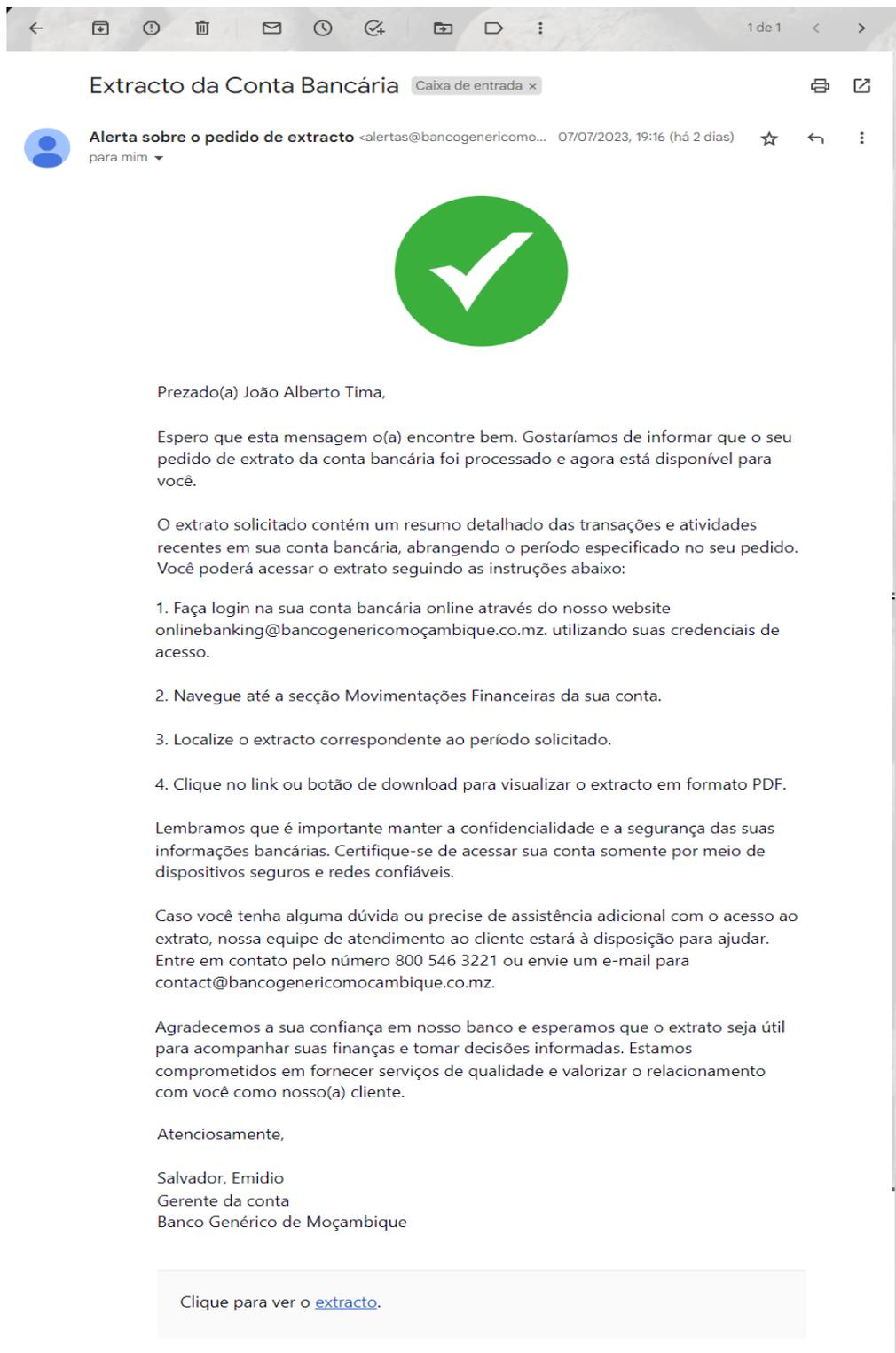


Figura 18: Email de notificação da conclusão do processamento

Fonte: elaborada pelo autor

Anexo 8: Push Notification da conclusão do processamento da requisição feita no sistema de internet banking

Uma push notification é uma mensagem enviada para dispositivos móveis ou desktops, a informar aos usuários sobre a conclusão do processamento assíncrono de transações. Essas notificações são entregues em tempo real, garantindo que os usuários sejam imediatamente informados sobre o status do processamento.

A8.1 Notificação no Android e IOS

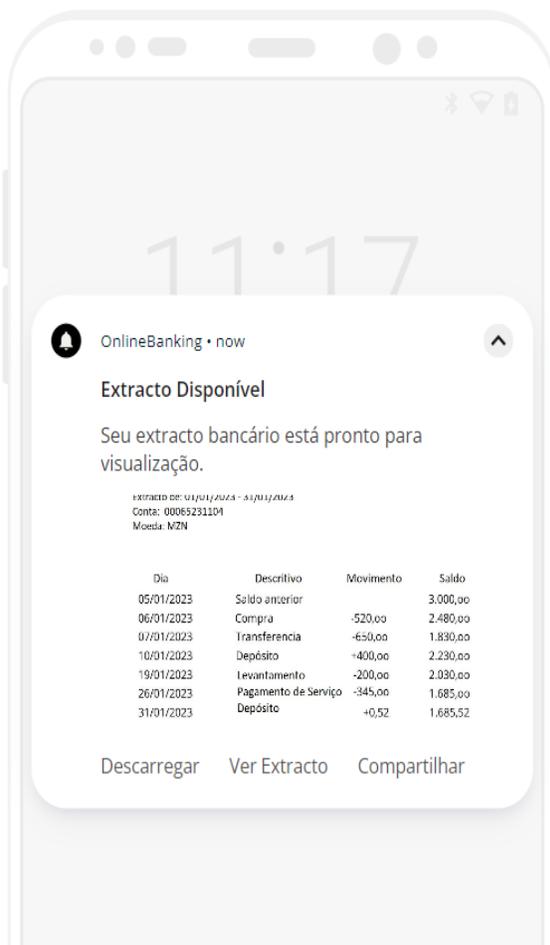


Figura 19: Push Notification da conclusão do processamento no Android

Fonte: elaborada pelo autor

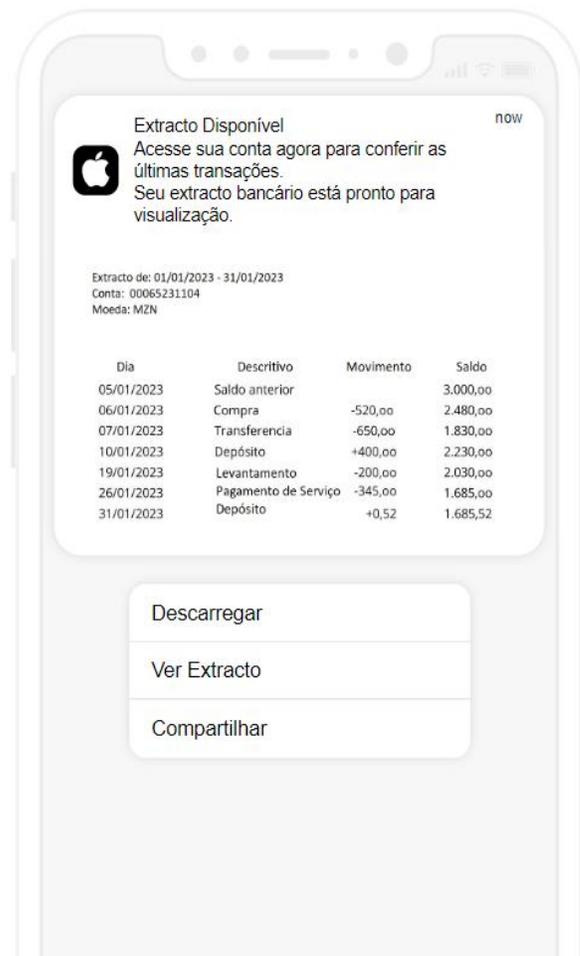


Figura 20: Push Notification da conclusão do processamento no IOS

Fonte: elaborada pelo autor

No Android e no iOS, as push notifications geralmente aparecem na parte superior da tela como uma barra suspensa. A barra contém o ícone do aplicativo, o título da notificação e um breve resumo do conteúdo. O usuário pode deslizar a notificação para baixo para revelar mais detalhes ou deslizar para o lado para descartá-la. Ao tocar na notificação, o usuário é redirecionado para o aplicativo correspondente.

A8.2 Notificação no Windows e MacOs

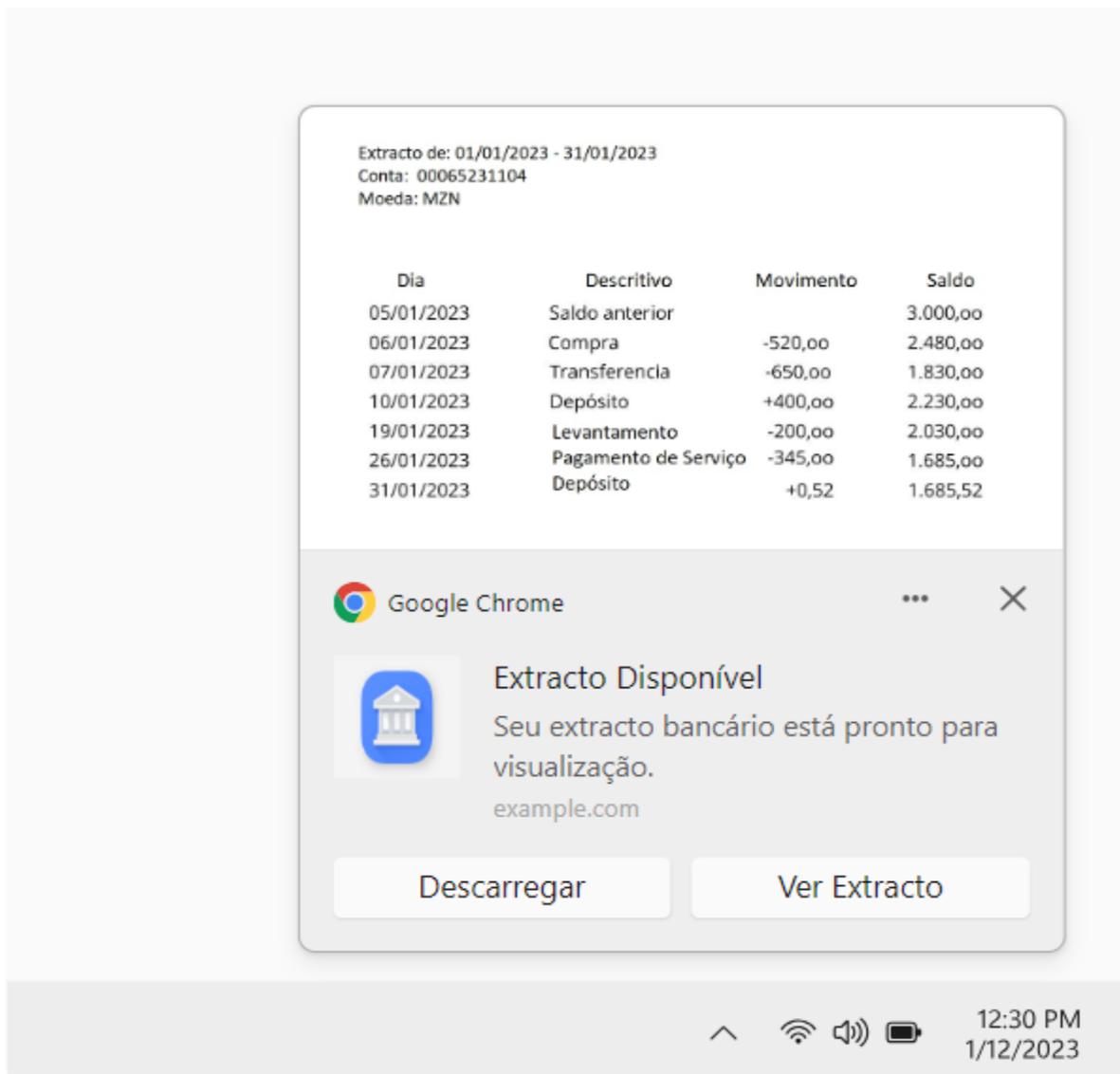


Figura 21: Push Notification da conclusão do processamento no Windows

Fonte: elaborada pelo autor

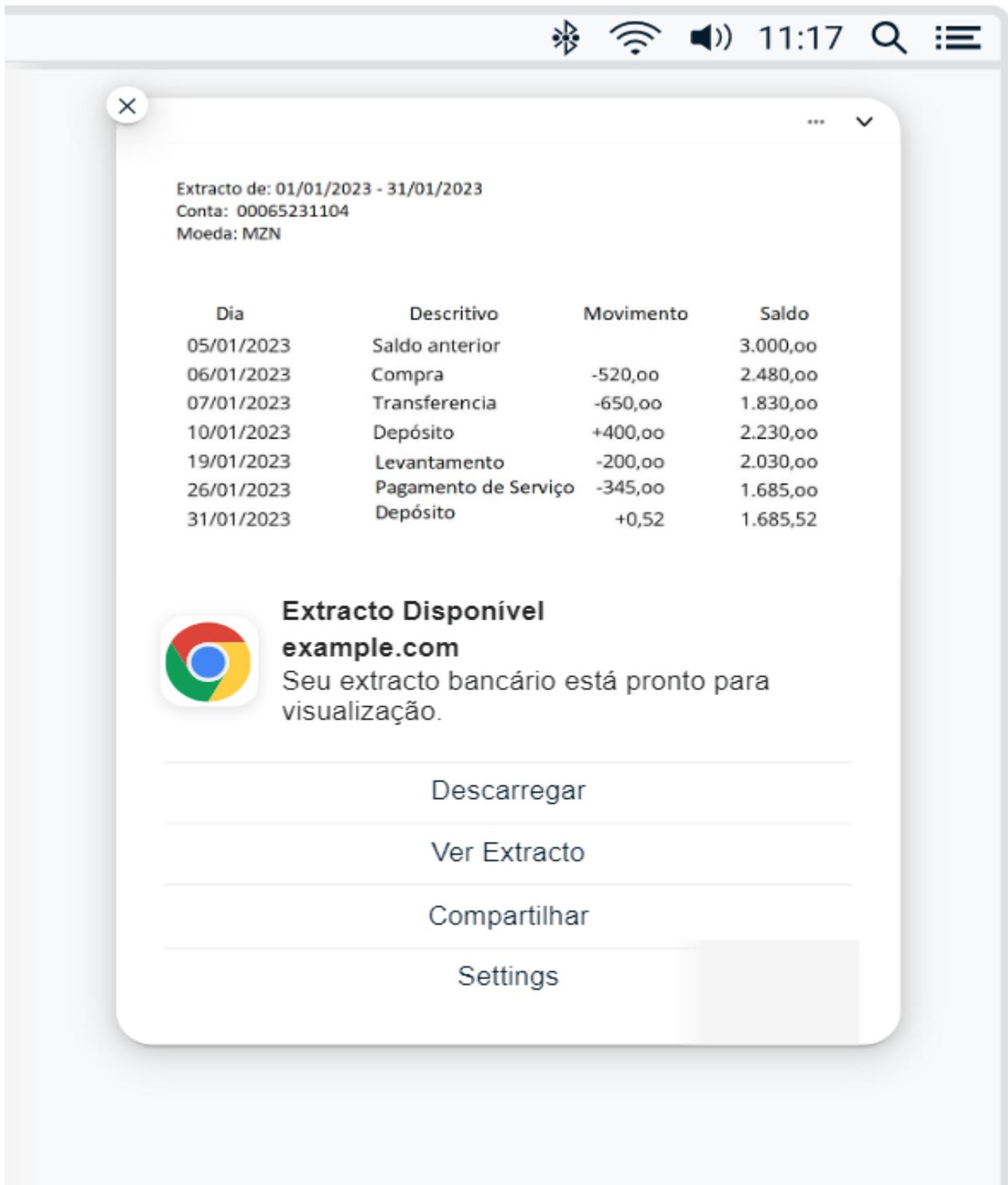


Figura 22: Push Notification da conclusão do processamento no MACOS

Fonte: elaborada pelo autor

No Windows, as push notifications são exibidas no canto inferior direito da tela, próximo à área de notificações. Geralmente, elas aparecem como pequenos rectângulos com uma breve mensagem, um ícone e, às vezes, um botão de acção. Quando uma nova notificação é recebida, ela pode ser exibida por alguns segundos antes de desaparecer, a menos que o usuário interaja com ela. Se o usuário não estiver a usar o computador no momento da notificação, ela ficará visível na Central de Ações, acessível a partir do ícone de notificações na barra de tarefas.

No macOS, as push notifications aparecem no canto superior direito da tela, próximo à área de notificações. Elas também são exibidas como pequenos rectângulos, contendo uma mensagem, um ícone e, às vezes, um botão de acção. As notificações podem ser exibidas por alguns segundos antes de desaparecer, a menos que o usuário interaja com elas. As notificações perdidas são arquivadas no Centro de Notificações, que pode ser acedido a partir do ícone de notificações no canto superior direito da barra de menu.