

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

PUC Minas Virtual

Pós-graduação *Lato Sensu* em Arquitetura de *Software* Distribuído

Projeto Integrado

Relatório Técnico

SISTEMA DE PROSPECÇÃO DE LEADS DE SEGUROS

Gabriel de Souza Martins

Belo Horizonte
Agosto de 2023

Projeto Integrado – Arquitetura de Software Distribuído

Sumário

Projeto Integrado – Arquitetura de Software Distribuído	2
Introdução	3
2. Especificação Arquitetural da solução	6
2.1 Restrições Arquiteturais	6
2.2 Requisitos Funcionais	6
2.3 Requisitos Não-funcionais	6
2.4 Mecanismos Arquiteturais	7
3. Modelagem Arquitetural	8
3.1 Diagrama de Contexto	8
3.2 Diagrama de Container	9
3.3 Diagrama de Componentes	10
4. Avaliação da Arquitetura (ATAM)	11
4.1 Análise das abordagens arquiteturais	11
4.2 Cenários	12
4.3 Evidências da Avaliação	12
5. Avaliação Crítica dos Resultados	14
6. Conclusão	15
Referências	16

Introdução

Ao longo da história pessoas e empresas sempre estiveram sujeitas a diversos tipos de eventos imprevisíveis, ou seja, a diversos tipos de riscos como por exemplo morte, incêndio, acidente de automóveis e outros. Portanto o seguro pode ajudar pessoas ou empresas a diminuïrem as preocupações futuras que poderão gerar prejuízo (ESCOLA NACIONAL DE SEGUROS, 2019).

Segundo a Escola Nacional de Seguros (2019), seguro é um mecanismo de transferência de risco de uma pessoa ou empresa para uma seguradora que assumirá este risco, portanto ao transferir o risco, a pessoa ou empresa irá pagar determinado valor à seguradora e caso este risco aconteça, a seguradora reembolsará as perdas sofridas.

O setor de seguros, segundo a FENACOR (2022), teve arrecadação de R\$ 321,05 bilhões no acumulado até novembro de 2022, que representa um crescimento de 16,6% em relação ao mesmo período em 2021, isso evidência sua capacidade de crescimento ano após ano e estima-se que essa tendência deverá se manter fazendo com que o setor mantenha uma participação importante no PIB brasileiro.

Atualmente, a empresa moderna e inteligente sabe que ouvir o cliente e solucionar seus anseios e problemas é a única forma de continuar existindo e para vendas de seguros isto não é diferente ou menos importante, pois a compreensão das tendências atuais ajudará em vários aspectos, sendo assim é importante saber que a nova ordem em vendas de seguros é baseada nos seguintes fatores: clientes mais exigentes e concorrência mais acirrada (SANTOS, 2018).

Com o avanço das tecnologias e o alcance do acesso a à internet, o comércio eletrônico ocupa cada vez mais espaço no mercado e por sua vez as lojas virtuais têm grandes desafios como prover a melhor experiência no processo de compra, agilizar entregas, fidelizar clientes e tornar os preços competitivos.

De acordo com SEBRAE (2022), pesquisas realizadas pelo E-commerce Radar mostraram que 82% dos clientes abrem mão da compra antes de realizarem o seu pagamento, ou seja, desistem de comprar e entre os principais motivos de desistência estão: indecisão, lentidão de internet, preço, frete, condições de pagamento e entrega.

SISTEMA DE PROSPECÇÃO DE LEADS DE SEGUROS

Segundo Lippert (2021), a compra de um usuário no ambiente digital, geralmente, é dividida em três fases: lead, oportunidade e venda. No caso de um e-commerce, o lead seria o carrinho, a oportunidade seria o pedido e a venda a fatura, porém 50% dos usuários que colocam os pedidos no carrinho não concluem a venda e da mesma forma, há também usuários que efetuam o pedido, geram boleto de pagamento, mas não o pagam.

Considerando este cenário de forma geral, este trabalho tem por objetivo propor uma solução de arquitetura de software que visa otimizar as vendas de seguros por meio da prospecção de leads durante a desistência por parte do cliente.

O cliente ao iniciar uma simulação de seguro, preencher seus dados pessoais e não finalizar a compra irá gerar um lead para que este, por sua vez, possa ser processado e gerar uma oferta mais atrativa ao cliente informando-o da nova oportunidade.

Como principais benefícios desta solução destacam-se: disponibilidade, escalabilidade, performance, melhoria na experiência do usuário, maximização de lucro e coleta de dados para estudos futuros.

A figura 1 exemplifica por meio de desenho livre a solução proposta

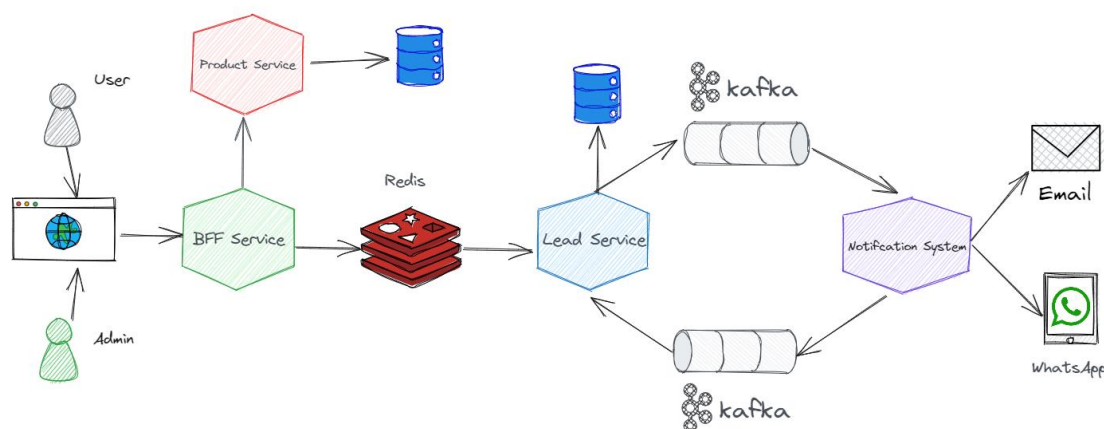


Figura 1 - Exemplo da solução Proposta. Fonte: do Autor

O objetivo geral deste trabalho é apresentar a descrição do projeto arquitetural de uma plataforma de prospecção de leads de vendas de seguros. O projeto visa fornecer uma plataforma na qual vendas de seguros não concluídas se transformem em dados que representem vendas em potencial e notifiquem o cliente para concluir a mesma. A plataforma deverá possuir alta disponibilidade, segurança, performance e poderá ser acessada por diversos dispositivos diferentes com acesso à internet através de um browser (desktops, notebooks, tablets e smartphones).

Os objetivos específicos propostos são:

- Criar um módulo responsável por simular vendas de seguros que permita o cliente informar dados pessoais e escolher o produto desejado.
- Criar mecanismos de integração que permitam a captura dos leads durante vendas não concluídas.
- Notificar o cliente da venda não concluída com base no lead capturado.

2. Especificação Arquitetural da solução

Esta seção apresenta a especificação arquitetural da solução proposta a ser desenvolvida, incluindo diagramas, restrições e requisitos funcionais e não-funcionais que permitam visualizar a macroarquitetura.

2.1 Restrições Arquiteturais

Nesta seção são definidos os requisitos arquiteturais da solução proposta levando em conta restrições tecnológicas que precisam ser satisfeitas.

R1: O frontend do software deve ser desenvolvido em linguagem JavaScript, com o <i>framework</i> Angular.
R2: O backend do software deve ser desenvolvido em linguagem Java, com o <i>framework</i> SpringBoot.
R3: As APIs devem seguir o padrão ReSTful.
R4: A captura de leads deve ocorrer primeiramente em um banco de dados em memória para fins de performance.
R5: O processamento dos leads deve ocorrer de forma assíncrona.
R6: O sistema deve ser hospedado em algum provedor de Nuvem.
R7: O backend deve gerar logs estruturados que permitam pesquisa e indexação.

2.2 Requisitos Funcionais

Esta seção tem por objetivo detalhes os requisitos funcionais da solução.

ID	Descrição Resumida	Dificuldade (B/M/A)*	Prioridade (B/M/A)*
RF01	Efetuar login	M	A
RF02	Manter produtos de seguros	M	A
RF03	O sistema deve permitir a simulação de seguro com base nos dados de CPF, Nome, Sexo e Data de Nascimento	A	A
RF04	O sistema deve validar o CPF informado	M	A
RF05	O sistema deve receber e validar o e-mail informado	M	A
RF06	O sistema deve notificar clientes que não concluírem a simulação com ofertas mais atrativas	A	A

*B=Baixa, M=Média, A=Alta.

2.3 Requisitos Não-funcionais

Esta seção tem por objetivo detalhes os requisitos não-funcionais da solução.

ID	Descrição	Prioridade B/M/A
RNF01	O sistema deve ser apresentar disponibilidade 24 X 7 X 365	A
RNF02	O sistema deve ser escalável	M
RNF03	O tempo de resposta de cada requisição deve ser menor que 500ms	M
RNF04	Não permitir que o cadastro de produtos possa ser feito por usuários não autenticados	A

2.4 Mecanismos Arquiteturais

Nesta seção são apresentados os mecanismos arquiteturais da solução proposta.

Análise	Design	Implementação
Frontend	Interface de comunicação com o usuário	Angular (AWS S3)
Backend	Regras de negócio da aplicação	Java com SpringBoot (AWS ECS)
Persistência de Produtos	ORM	JPA + Hibernate (AWS RDS0)
Persistência de Leads	ORM	MongoDB (AWS DocumentDB)
Integração com persistência de análise de leads	Publisher/Subscriber	Redis (AWS ElastiCache)
Integração com aplicação de análise de leads	Mensageria	Apache Kafka (AWS MSK)
Governança de APIs	Documentação e versionamento de APIs	OpenAPI 3.0 (AWS API Gateway)
Autenticação e Autorização	Verificação das credenciais para executar ações	OAuth0 (AWS Lambda)
Integração	Mensageria	Apache Kafka (AWS MSK)
Log do sistema	Framework de log	Logstash + Logback (AWS CloudWatch Logs)
Teste de Software	Testes automatizados	Cucumber + TestContainers (Github Actions)
CI/CD	Ferramenta para pipeline de integração e entrega contínua	Github Actions
Deploy	Hospedagem dos sistemas	Terraform (Github Action)
Versionamento	Controle de código-fonte	Git e Github

3. Modelagem Arquitetural

Esta seção apresenta a modelagem arquitetural da solução proposta por meio de diagramas com o objetivo de permitir seu completo entendimento.

3.1 Diagrama de Contexto

A figura 1 apresenta o diagrama de contexto da solução proposta exemplificando a utilização do sistema por parte do cliente para simulação, a utilização do administrador para cadastro e manutenção de produtos e a captura e processamento de leads.

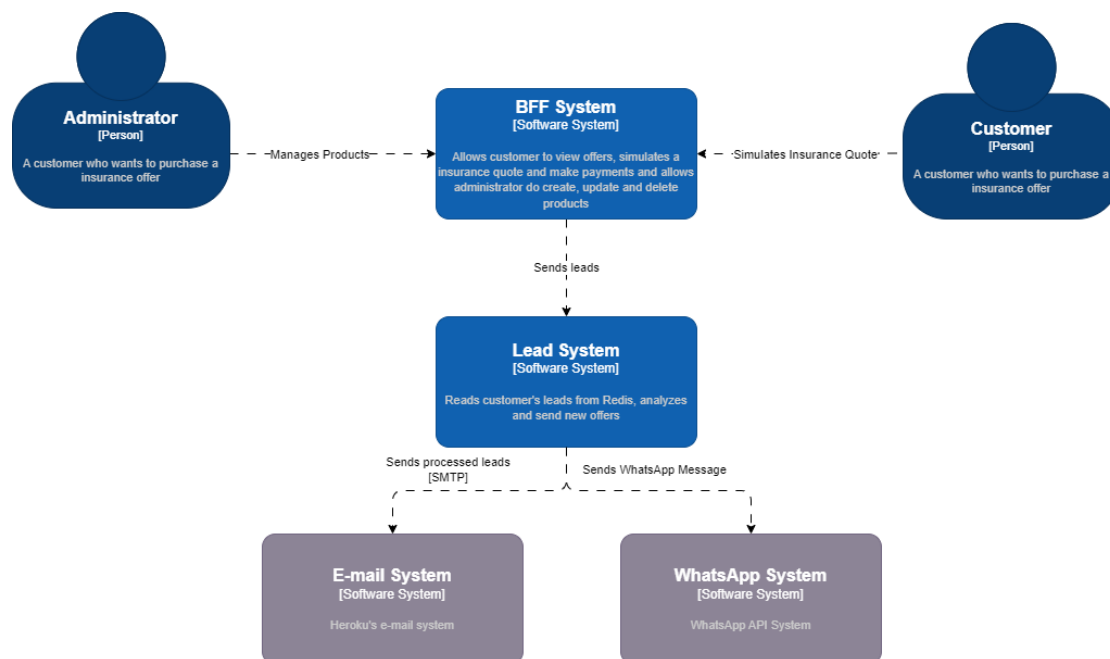


Figura 2 - Diagrama de Contexto da Solução. Fonte: do Autor

Como é possível observar, existem dois tipos de usuários: o cliente que irá realizar simulações/compra de seguros e o administrador responsável por cadastrar produtos.

3.2 Diagrama de Container

A Figura 3 apresenta o Diagrama de Container da solução contendo mais detalhes da mesma.

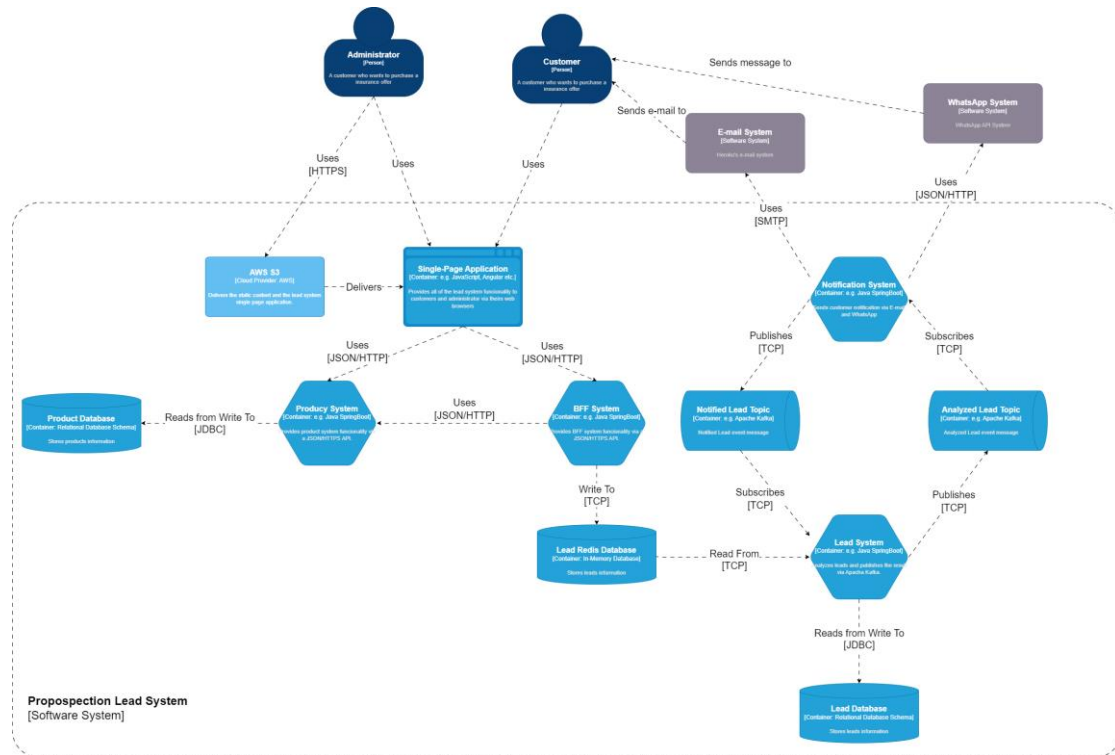


Figura 3 - Diagrama de Container da Solução. Fonte: do Autor

Nesta figura são representados os microsserviços que irão compor a solução, os bancos de dados que irão guardar informações de produto e lead e também a plataforma de mensageria responsável pela integração dos microsserviços.

É possível perceber a existência de dois tópicos: o tópico de lead analisado será utilizado pelo microsserviço de notificação para enviar e-mail e mensagem ao cliente que desistiu da compra e também o tópico de lead notificado que é usado para confirmação de que a notificação (e-mail/mensagem) foi entregue, seguindo assim o padrão SAGA.

3.3 Diagrama de Componentes

A Figura 4 apresenta o Diagrama de Componentes da solução contendo mais detalhes.

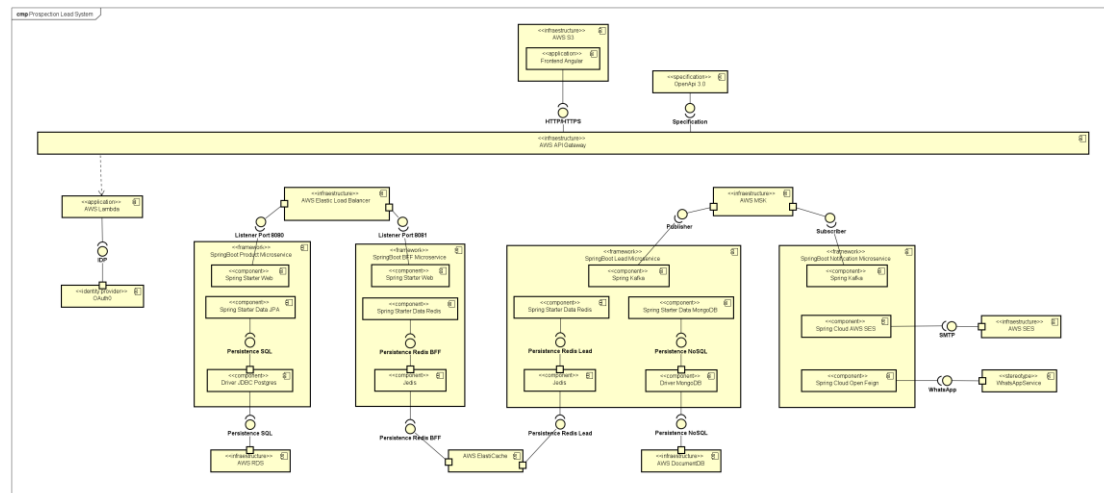


Figura 4 - Diagrama de Componentes da Solução. Fonte: do Autor

Conforme o diagrama apresentado na figura 4, os componentes participantes da solução são:

- Frontend Angular: Interface de comunicação com o usuário que permite que o mesmo crie produtos (quando administrador) e faça simulações de seguro (quando cliente).
- Product Microservice: Microserviço responsável por prover as funcionalidades de produto
- BFF Microservice: Microserviço responsável pelos dados que serão exibidos no frontend.
- Lead Microservice: Microserviço responsável pela análise dos leads capturados.
- Notification Microservice: Microserviço responsável por comunicar o cliente via e-mail e mensagem sobre nova ofertas.
- AWS S3: Responsável por armazenar o conteúdo estágio do frontend.
- API Gateway: Responsável por centralizar as requisições HTTP e encaminhar para o backend correspondente
- AWS Lambda: Serviço AWS que será utilizado como autorizador pelo API Gateway.

- AWS Elastic Load Balancer: Responsável por balancear o tráfego recebido pelo API Gateway e distribuir nos containers de aplicações.
- AWS SES: Serviço de e-mail da AWS.
- AWS MSK: Serviço da AWS com Apache Kafka gerenciado para comunicação entre microserviços.
- AWS ElastiCache: Serviço da AWS responsável pelo Redis gerenciado.
- AWS DocumentDB: Serviço da AWS responsável pelo MongoDB gerenciado.
- AWS RDS: Serviço da AWS responsável pelo Postgres gerenciado.
- WhatsApp Service: API do WhatsApp para envio de mensagens.
- OAuth0: Identity Provider que fará a gestão de acessos.
- Spring Data JPA: Módulo responsável pela implementação JPA de Persistência de Dados.
- Driver JDBC Postgre: Driver JDBC responsável pela comunicação com o Postgres.
- Spring Data MongoDB: Módulo responsável pela persistência dos dados no MongoDB.
- Driver MongoDB: Driver responsável pela comunicação com o MongoDB.
- Spring Data Redis: Módulo responsável pela persistência dos dados no ElastiCache Redis.
- Jedis: Java client responsável pela comunicação com o Redis.

4. Avaliação da Arquitetura (ATAM)

A avaliação da arquitetura desenvolvida neste trabalho é abordada nesta seção visando avaliar se ela atende ao que foi solicitado pelo cliente, segundo o método ATAM.

4.1 Análise das abordagens arquiteturais

Apresente aqui um breve resumo das principais características da proposta arquitetural. Para isto, utilize o método Architecture Tradeoff Analysis Method (ATAM), no qual são utilizados cenários para fazer essa análise.

Exemplo:

SISTEMA DE PROSPECÇÃO DE LEADS DE SEGUROS

Atributos de Qualidade	Cenários	Importância	Complexidade
Interoperabilidade	Cenário 1: O sistema deve se comunicar com sistemas de outras tecnologias.	A	M
Usabilidade	Cenário 2: O sistema deve prover boa usabilidade.	M	B
Manutenibilidade	Cenário 3: O sistema deve ter a manutenção facilitada.	M	M

4.2 Cenários

Mostre os cenários utilizados na realização dos testes da sua aplicação. Escolha cenários de testes que demonstrem os requisitos não funcionais (atributos de qualidade) sendo satisfeitos. Priorize os cenários para a avaliação segundo critérios quantitativos ou qualitativos.

Exemplos de cenários:

Cenário 1 - Interoperabilidade: Ao acessar a URL do serviço de informações gerenciais via HTTP GET, o mesmo deve retornar as informações no formato JSON.

Cenário 2 - Usabilidade: Ao navegar na tela, o sistema deve apresentar boa usabilidade. A navegação deve apresentar facilidade e o acesso as funcionalidades deve ser bem objetivo para a função que precisar ser realizada, o usuário deve ser capaz de efetuar uma compra em no máximo 5 minutos, assim garantindo a agilidade e a usabilidade para ficar de acordo com um dos requisitos não funcionais.

Cenário 3 - Manutenibilidade: Havendo a necessidade de alterar o gateway de pagamento somente será necessário fazer alteração no broker da funcionalidade de pagamento, facilitando a manutenção e os testes.

4.3 Evidências da Avaliação

Apresente as medidas registradas na coleta de dados. Para o que não for possível quantificar apresente uma justificativa baseada em evidências qualitativas que suportem o atendimento ao requisito não-funcional.

Atributo de Qualidade:	Interoperabilidade
Requisito de Qualidade:	O sistema deve se comunicar com outras tecnologias.
Preocupação:	

O sistema deve ter como resposta a uma requisição uma saída de fácil leitura por outro componente.	
Cenário(s):	
Cenário 1	
Ambiente:	
Sistema em operação normal	
Estímulo:	
O sistema de monitoramento envia uma requisição para o serviço REST do módulo de informações gerenciais.	
Mecanismo:	
Criar um serviço REST para atender às requisições do sistema de monitoramento	
Medida de resposta:	
Retornar os dados requisitados no formato JSON	
Considerações sobre a arquitetura:	
Riscos:	Alguma instabilidade na rede pode deixar a conexão lenta ou mesmo a perda de pacotes.
Pontos de Sensibilidade:	Não há
Tradeoff:	Não há

Acrescente imagens e descreva os testes realizados, de tal forma que se comprove a realização da avaliação.

Faça isto para todos os cenários apresentados no tópico 6.1.

5. Avaliação Crítica dos Resultados

Apresente aqui, de forma resumida, os principais pontos positivos e negativos da arquitetura proposta. Adote uma postura crítica que permita entender as limitações arquiteturais, incluindo os prós e contras das tecnologias. Você pode utilizar o formato textual ou produzir um quadro resumo.

Ex. de quadro resumo:

Ponto avaliado	Descrição
XXXXXXXXXXXXXXXXXX	XXXXXXXXXXXXXXXXXXXXXX

6. *Conclusão*

Descreva, de forma sucinta, quais foram as lições aprendidas na execução do seu projeto arquitetural. Procure apresentá-las de tal forma que fiquem configurados os *trade-offs* da arquitetura produzida, como por exemplo, Segurança X Desempenho, Granularidade X Manutenibilidade, etc.

Aqui deve ser apresentado também tudo que se aprendeu com esse projeto, de modo a servir como ajuda para outros profissionais.

Também se faz necessário evidenciar as possibilidades de melhoria do projeto, caso se deseje dar continuidade a ele. Nesse sentido, indique possíveis ajustes ou melhorias arquiteturais, que possam vir a ser realizados futuramente.

Lições aprendidas (ex.):

1. xxxxxxxxxxxxxxxxxxxx
2. xxxxxxxxxxxxxxxxxxxx
3. xxxxxxxxxxxxxxxxxxxx

Referências

ESCOLA NACIONAL DE SEGUROS, Diretoria de Ensino Técnico. **Teoria Geral do Seguro – 6.ed.** Rio de Janeiro: ENS, 2019.

FENACOR. **Mercado de seguros cresceu 16,6% até novembro.** 2023. Disponível em: <https://www.fenacor.org.br/noticias/mercado-de-seguros-cresceu-166-ate-novembro>. Acesso em: 25 de junho de 2023.

LIPPERT, Dener. **Cientista do Marketing Digital: Como vender para mais pessoas, mais vezes e pelo maior valor - 1.ed.** São Paulo: Editora Gente, 2021.

SANTOS, André. **Seguros: como vender mais e melhor! técnicas e muitas dicas para aumentar seu desempenho – 1.ed.** Rio de Janeiro: FUNENSEG, 2008.

SEBRAE. **6 motivos para consumidores desistirem no momento de concluir a compra.** 2022. Disponível em: <https://sebrae.com.br/sites/PortalSebrae/artigos/6-motivos-para-consumidores-desistirem-no-momento-de-concluir-a-compra,ab57ff3754a42810VgnVCM100000d701210aRCRD>. Acesso em: 25 de junho de 2023.