

# **SIB**

## **Smart Insurance Blockchain**

## Controle do SIBDoc - documentação geral do projeto

### Histórico de revisões

Data	Autor	Versão	Resumo da atividade
<07/02/2023>	<Pedro Rezende>	1.1	Criação do documento.
<08/02/2023>	<Pedro Rezende>	1.2	Atualizações das seções do Documento.
<08/02/2023>	<Kathlyn Diwan>	1.3	Descrição dos tópicos 1.2 e 2.1.
<10/02/2023>	<Pedro Rezende>	1.4	Implementação dos conteúdos do tópico 3.
<10/02/2023>	<Patrick Miranda> <Larissa Carvalho>	1.5	Atualização do tópico 2.1; Implementação do tópico 4.1; Descrições dos tópicos: 3.2.2, 3.2.3, 4.1, 4.3.
<12/02/2023>	<Dayllan Alho>	1.6	Atualização do tópico 3.2.4; Implementação dos tópicos: 1, 1.1 e 2.2; Descrições dos tópicos: 3.1, 4 e 4.3.
<12/02/2023>	<Pedro Rezende>	1.7	Revisão e organização geral do documento; Descrição do tópico 3.2; Implementação do tópico 5.5 e seus subtópicos.
<13/02/2023>	<Patrick Miranda> <Larissa Carvalho>	2.1	Revisão do Documento; Implementação da descrição dos tópicos 4.2 e 5.2.
<13/02/2023>	<Pedro Rezende>	2.2	Implementação das 3 Jornadas do usuário (tópico 4.2); Implementação da Arquitetura da solução (5.2).
<23/02/2023>	<Dayllan Alho> <Patrick Miranda> <Larissa Carvalho>	2.3	Reorganização das User Stories e Personas
<27/02/2023>	<Pedro Rezende>	2.4	Mudança da Arquitetura da Solução
06/04/2023	Dayllan	2.5	Revisão documental e acréscimos necessários

# Sumário

<b>1. Introdução</b>	<b>5</b>
1.1 Parceiro de Negócios	6
1.2. Definição do Problema	6
<b>2. Objetivos e Justificativa</b>	<b>7</b>
2.1 Objetivos	7
2.2 Justificativa	7
<b>3. Análise do Negócio</b>	<b>8</b>
3.1. Contexto da indústria	8
3.2. Ferramentas	13
3.2.1. Análise SWOT	14
3.2.2. Value Proposition Canvas	15
3.2.3. Matriz de Riscos	16
3.2.4 Matriz de avaliação de valor Oceano Azul	17
3.2.5 Análise financeira do projeto	19
<b>4. Análise de Experiência do Usuário</b>	<b>20</b>
4.1. Personas	20
4.2. Jornadas do Usuário e/ou Storyboard	23
4.3. User Stories	25
4.4. Protótipo de interface com o usuário	32
<b>5. Solução proposta</b>	<b>32</b>
5.1. Solução	32
5.2. Arquitetura proposta	32
5.3. Diagrama Macro da Solução	33
5.4. Descrição da Solução	33
5.5. Análise de Risco (Segurança da Informação)	33
5.5.1. Requisitos de segurança da informação	35

5.5.2. Identificar as vulnerabilidades existentes	35
5.5.3. Atacantes e levantar ataques possíveis	37
5.5.4. Priorizar os ataques de acordo com a sua probabilidade e impacto	40
5.6. Algoritmos e Equações	<b>Error! Bookmark not defined.</b>
<b>6. Desenvolvimento e Resultados</b>	40
6.1. Nome do Módulo	40
6.1.1. Descrição	41
6.1.2. Tecnologias adotadas	46
6.1.3. User Stories	48
6.1.4. Prototipação	52
6.1.5. Diagramas	6
6.2. Avaliação	6
<b>7. Conclusões e Recomendações</b>	5
<b>8. Referências</b>	5
<b>Anexos</b>	<b>Error! Bookmark not defined.</b>
ANEXO I – Sprint 1	<b>Error! Bookmark not defined.</b>
ANEXO II – Sprint 2	<b>Error! Bookmark not defined.</b>
ANEXO III – Sprint 3	<b>Error! Bookmark not defined.</b>
ANEXO IV – Sprint 4	<b>Error! Bookmark not defined.</b>

# 1. Introdução

A Coover é uma empresa líder no setor de seguros digitais, fundada em 2015 na cidade de Vitória, no Espírito Santo. Com a missão de transformar a forma como as pessoas protegem o que é importante para elas, a Coover tornou o processo de compra de seguros mais fácil e acessível. Em 2019, a empresa foi autorizada pela Superintendência de Seguros Privados (Susep), tornando-se uma seguradora regulamentada e uma das primeiras seguradoras digitais a atuar no mercado. Em 2021, a Coover uniu-se à Zurich, uma das maiores seguradoras do mundo, para ampliar sua presença e oferecer ainda mais segurança aos seus clientes. Recentemente, a Coover abriu um escritório em São Paulo e está expandindo suas linhas de negócios e operações.

A equipe de desenvolvimento da Coover tem como objetivo trazer soluções inovadoras para o mercado de seguros, explorando as vantagens da tecnologia blockchain para criar novos modelos de proteção. A empresa busca oferecer soluções baseadas em mutualismo e gerenciadas por contratos inteligentes, possibilitando a criação de soluções securitárias peer-to-peer.

Infelizmente, mais de 70% da população brasileira atualmente não possui nenhuma forma de proteção securitária privada. A Coover acredita que é possível mudar essa realidade, oferecendo soluções acessíveis e transparentes a todos. O MVP da Coover se concentrará no design e nas funcionalidades de contratos inteligentes na blockchain, proporcionando soluções eficientes e acessíveis à população.

A necessidade de projetos como o da Coover surge devido às dificuldades encontradas pelas empresas de seguros em oferecer soluções intuitivas e acessíveis aos seus clientes. A Coover oferece uma ampla gama de proteções, incluindo seguros para celular e o primeiro seguro de saúde pet autorizado no país, atuando no setor de seguros digitais. A empresa espera fomentar a estabilidade econômica e promover a segurança de indivíduos, empresas e comunidades, oferecendo soluções eficazes para a carência de proteção securitária.

## **1.1 Parceiro de Negócios**

A Coover é uma startup de seguros digitais que tem como objetivo tornar a proteção de bens mais acessível e simples para as pessoas. Com o uso da tecnologia blockchain, a Coover deseja oferecer soluções securitárias inovadoras, baseadas em mutualismo, gerenciadas por smart contracts. O MVP da Coover é focado no design e na criação de smart contracts inteligentes na blockchain, buscando proporcionar soluções mais acessíveis e transparentes para a população, suprimindo a necessidade do parceiro em fomentar a estabilidade econômica e promover a segurança dos indivíduos, empresas e comunidades.

## **1.2. Definição do Problema**

Atualmente, os seguros são percebidos como instrumentos valiosos de proteção, que oferecem estabilidade financeira e segurança aos indivíduos e empresas. Eles criam uma barreira contra eventos incertos e futuros, prevenindo possíveis danos econômicos ou emocionais. Infelizmente, apesar de sua importância, a maioria da população brasileira ainda não tem acesso a esses meios de proteção. Mais de 70% dos brasileiros vivem sem nenhuma forma de segurança, o que os coloca em uma posição vulnerável e expostos a riscos e ameaças. Dessa forma, em parceria com a empresa Coover, pretendemos implementar uma solução em blockchain para garantir um novo tipo de aplicação de seguros, que traga maior acessibilidade e participação dos próprios integrantes do grupo de seguros, a partir da cultura de seguro mútuo.

## 2. Objetivos e Justificativa

### 2.1 Objetivos

Com o surgimento de novas pessoas sem acesso a nenhuma proteção securitária, o objetivo geral deste projeto é implementar um MVP de um protocolo em blockchain ethereum para proteção para proteção contra roubo/furto de celulares; e uma aplicação Web 3 que atenda tanto a área do cliente quanto da seguradora. Criando assim uma plataforma em web 3 baseada em smart contracts, fornecendo aos usuários a oportunidade de visualizar e aceitar um contrato do qual deseja participar e posteriormente, realizar seus pedidos de seguro . E a empresa de seguros terá a oportunidade de gerenciar e aprovar seguros para liberar fundos a partir desse smart contract. Aumentando assim a segurança dos usuários, e diminuindo os custos operacionais e de contratação de um seguro.

#### **Objetivos específicos**

- Desenvolver de um sistema em blockchain para proteção de celulares
- Criação de uma plataforma em web 3.0 com área do usuário de do gerente de seguros
- Criação de um smartcontrast funcional para solução
- Parte da interface que possibilite que o gerente de seguros, aprove ou rejeite solicitações
- Oferecer um processo eficiente e com pouca burocracia

### 2.2 Justificativa

Nós acreditamos que é possível resolver os problemas enfrentados pelos clientes, utilizando tecnologia e inovação. Ao adotar nossa solução, os parceiros terão acesso a uma plataforma intuitiva, prática, segura e rápida de contratação de seguros baseados

no mutualismo, permitindo ampliar sua rede de clientes e oferecer soluções completas e personalizadas. Além disso, nossa tecnologia baseada em blockchain garante transparência e segurança nas transações.

Em resumo, nossa proposta de solução se diferencia pela sua facilidade de uso, personalização e segurança, contribuindo para a ampliação da rede de clientes do parceiro e a melhoria dos processos de contratação de seguros.

## 3. Análise do Negócio

Descreva as etapas que irão compor a análise do negócio do parceiro. Faça a apresentação ao leitor dos tópicos desenvolvidos.

### 3.1. Contexto da indústria

As 5 forças de Porter é uma ferramenta de análise de mercado criada por Michael Porter, para avaliar a intensidade da concorrência em um setor e a força das forças competitivas que determinam a atratividade de um mercado. Essas 5 forças são:

1. Rivals Competitivos: a concorrência direta e indireta de outras empresas no setor.
2. Poder de negociação de compradores: o poder de barganha dos compradores em relação ao preço, qualidade e outros requisitos.
3. Poder de negociação de fornecedores: o poder de barganha dos fornecedores em relação a preços e condições de fornecimento.
4. Ameaça de novos entrantes: a facilidade de entrada de novas empresas no mercado, incluindo barreiras à entrada, economias de escala e recursos financeiros.
5. Ameaça de produtos substitutos: a presença de produtos ou serviços substitutos que possam afetar a demanda pelos produtos atuais do setor.



Essas 5 forças são utilizadas para avaliar o potencial de lucro de um setor e para identificar as principais fontes de pressão competitiva, ajudando as empresas a tomar decisões estratégicas e a se posicionar de forma mais forte no mercado.

Abaixo segue o infográfico das 5 forças de Porter da Coover

## Infográficos das 5 forças de Porter - Coover



A partir da análise de forças, proposta por Michael Porter, se inicia a análise do contexto da indústria a qual a startup Coover se localiza, a fim de entender os principais competidores/*players*, modelos de negócios e tendências do mercado.

### Modelo de Negócio:

A Coover é uma startup de seguros digitais (*insurtech*) que está revolucionando a forma de cuidar e administrar as necessidades que os clientes apresentam de forma menos complicada e mais acessível, a partir da implementação de novas tecnologias e de uma nova cultura, que é o "seguro mútuo". A empresa trabalha no segmento de Seguros, atualmente focado em smartphones e, logo mais, no seguro de PET's, além de visar abranger seus serviços para novas vertentes.

Funciona mais ou menos assim: uma turma de colegas combina de formar um grupo de proteção para, por exemplo, smartphones. Eles definem o valor mínimo de cobertura

como R\$500,00, e a quantidade mínima de pessoas para garantir 100% de um evento de perda total. No caso de R\$500,00, se o grupo tem ao menos 20 pessoas, cada um precisa entrar com pelo menos R\$25,00 para o grupo arrecadar o valor total da cobertura. Suponha então que uma pessoa do grupo seja roubado e precise de um celular novo. Então o reembolso deve ser aprovado pelo grupo e o dinheiro é transferido para o participante. Em seguida, cada participante deve repor seu saldo mínimo individual.

Assim, o investimento mínimo para cada participante varia de acordo com o tamanho do grupo e com o bem que as pessoas querem proteger. No viés da empresa, ela ganha apenas com as taxas administrativas pagas por quem usa a plataforma, além disso, a Coover não fica responsável pelo dinheiro do grupo e não pode movimentá-lo, já que tudo funcionar em Blockchain.

### **Rivalidade entre concorrentes / Principais Players:**

A questão das *Insurtechs* é nova e cheia de startups e empresas que estão se consolidando no mercado, sua principal inovação é a utilização de tecnologias inovadoras que mudam a forma como os consumidores contratam planos de seguro. Assim como as *fintechs*, as *insurtechs* são startups que pretendem inserir o poder das novas tecnologias em um mercado conservador, e fazê-lo se beneficiar muito delas, de forma que reduzam drasticamente a burocracia, falta de personalização, inacessibilidade financeira e dificuldade para entender os planos.

Nesse quesito, a Coover também é uma startup que faz parte desse processo de iniciar esse novo tipo de mercado, porém, junto dela, há empresas que iniciaram seus negócios com diferentes propostas e tecnologias. Nesse sentido, escolhemos 3 principais concorrentes/*players* que revolucionaram a maneira de contratar seguros e compartilham esse novo mercado de *Insurtechs* com a Coover:

#### **Player 1 → Kakau**

- Empresa focada em seguros de celular e seguros de bicicletas, sendo 100% online. Além disso, tem bastante influencia em posts e divulgação de produtos que influenciem em seus seguros, como por exemplo: equipamentos de bicicleta.

- Porém, possuem certas limitações no que o seguro de aparelhos celulares abrange...

### **Player 2 → Minuto Seguros**

- A Minuto é a maior corretora de seguro auto do país e a maior do segmento online. Além disso, vedem outros tipos de seguros, como por exemplo: seguro residencial, seguro viagem, seguro de vida, seguro saúde empresarial e seguro saúde patrimonial.
- A corretora funciona 100% online e possui uma enorme quantidade de parcerias com outras empresas e até mesmo outras seguradoras.

### **Player 3 → Youse**

- A Youse é uma empresa 100% digital, que oferece 3 tipos de seguros, sendo eles: seguro auto, seguro residencial e seguro de vida. Aplicando a forma de pagamento mensal, podendo ser até por pix, e habilitando quaisquer alterações pelo app, o que facilita a vida de muitos usuários.
- Possuem uma cultura clara de uma comunidade de que se ajuda e recebe por trazer novas pessoas.

### **Ameaça de novos concorrentes:**

A Coover é uma startup que está "inaugurando" o mercado de seguros com blockchain, então, ela está no período de aprender e trabalhar em paralelo com novos concorrentes. Porém, mesmo com esse fator, a base de uma empresa é buscar lucro, nesse viés, a Coover possui diferenciais claros, com a implementação da tecnologia Blockchain e sua cultura de seguro mútuo. Dessa forma, as possíveis principais preocupações da empresa, deveria ser com Insurtechs que estão se consolidando no

mercado e ganhando cada vez mais espaço, já que, por ser uma startup bem investida, é difícil que apareça novos concorrentes ameaçadores.

### **Ameaça de Produtos substitutos:**

O tipo de produto/serviço que a Coover disponibiliza já é novo, além de usar novas tecnologias, como blockchain. Então, o produto da empresa se encontra numa posição “confortável”, perante possíveis substitutos, já que inova na forma de prestar o serviço e vendê-los, vende-los, com a aplicação de uma nova cultura.

Porém, uma possibilidade de substituição seria a compra de garantias na hora da compra do produto. Como por exemplo, na hora da compra de um celular ou de um computador, já é possível fazer a compra de uma garantia que habilite reparos e cuidados com o equipamento.

### **Poder de negociação dos clientes:**

No que diz respeito ao poder negocial dos clientes, os quais possuem alta palavra de barganha nesse tipo de setor, fica claro os seguintes fatores: a demanda por melhor qualidade e acessibilidade, faz com que as empresas tenham que aumentar drasticamente sua eficiência, na logística dos apps e atendimento ao cliente.

Pela Coover ser uma startup com um novo tipo de serviço no mundo de seguros e por utilizar novas tecnologias, sua abordagem na negociação e apresentação com os clientes é delicada e essencial. Isso se deve ao fato que é muito importante demonstrar o diferencial da empresa, sem complicar, para melhor entendimento do cliente.

### **Poder de negociação dos fornecedores:**

A análise de Porter, em relação ao poder de negociação dos fornecedores, se resume à capacidade das empresas parceiras e fornecedores terceirizados barganhar com a Coover, no sentido de fornecer produtos e parcerias que auxiliem ou, até mesmo, influenciem no desenvolvimento dos serviços. Outro fator influenciador, é o fato da

Coover ser uma startup que precisa de investimentos para o desenvolvimento e divulgação de seus serviços, o que acaba por necessitar de negociações com novos fornecedores e parceiros.

### **Tendências de Mercado:**

Atualmente, a maior tendência de mercado, para esse tipo de setor (seguros digitais), é a capacidade de desenvolvimento tecnológico e o marketing para a chamada de clientes. Existem diversas outras empresas *insurtechs*, porém, elas abordam outros tipos de produtos e até outras tecnologias, os quais não são acatados pela Coover. Dessa forma, ela se encontra em uma situação confortável perante outros concorrentes, já que aplica Blockchain em seus produtos (demonstrando já se adaptar para a evolução em massa com blockchain) e por utilizar um outro tipo de cultura, que é o seguro mútuo, além de já ser uma startup "bem-visada" em um âmbito novo como esse.

### **Conclusão:**

Em geral, a iminente inovação e melhora a partir dos meios tecnológicos utilizados pela empresa, além da implementação de uma nova cultura de negócio ("seguro mútuo"), são meios que possam assegurar o crescimento da Coover no setor de seguros digitais. Assim, cativar os usuários a partir do novo meio de negócio, faria com que assegurasse mais esses clientes para a Coover, consequentemente, faria com que os fornecedores percebessem o crescimento da empresa, gerando mais receita para aplicações.

## **3.2. Ferramentas**

Para esse tópico, utilizamos diversas ferramentas como auxílio da nossa análise de negócio da empresa parceira. Os seguintes subtópicos apresentam a ferramenta utilizada, uma pequena descrição do motivo de seu uso, além do conteúdo de análise. Em ordem, estará respectivamente:

- Análise de SWOT - Strengths, Weaknesses, Opportunities, e Threats.
- Value Proposition Canvas

- Matriz de Riscos
- Matriz de avaliação de valor Oceano Azul
- Análise financeira do projeto

### **3.2.1. Análise SWOT**

A análise Swot é uma técnica de organização empresarial que possibilita a empresa a realizar análises referentes aos ambientes e cenários, internos e externos a em relação a mesma. Assim é explicitado como é sua situação no setor, suas forças, fraquezas, oportunidades e ameaças. Sendo esses, os quatro fatores que a compõem:

#### **I. Forças:**

É possível identificar como pontos fortes para a empresa, os casos a seguir:

- A) Smart Contracts (Contratos auto executáveis, assim otimizam tempo de trabalho), minimizando o trabalho necessário para firmar um contrato.
- B) Facilidade em obter o seguro pelo site e cancelar facilmente, pouca burocracia
- C) Número de funcionários reduzido em relação a uma empresa normal de seguro, perante o funcionamento e manutenção dos processos de seguro.
- D) Oportunidade de crescimento e inovação diferente de uma empresa tradicional.

#### **II. Fraquezas:**

É possível identificar como pontos fracos para a empresa, os casos a seguir:

- A) Ter poucos tipos de seguros disponíveis para o usuário contratar.
- B) Discrepância entre o poder financeiro da empresa, por ser um startup, comparado a grandes empresas de seguro que já estão no mercado.
- C) Seguro não é visto como um serviço essencial pelas pessoas, por isso, há o risco de cancelamento a qualquer momento.

### **III. Oportunidades:**

É possível identificar como oportunidades para a empresa, os casos a seguir:

- A) Cerca de 70% da população ainda não possui seguros privados, sendo um grande mercado a se explorar.
- B) Mais de 40% da população já sofreu um assalto, tendo um grande mercado de pessoas para contratar seguros anti-furto.
- C) Quase nenhuma empresa atua no mercado de seguros através da blockchain.

### **IV. Ameaças:**

É possível identificar como ameaças para a empresa, os casos a seguir:

- A) Apenas cerca de 30% da população tem o costume de contratar seguros privados.
- B) Falta de afinidade e desconfiança da população em relação a sistemas baseados em blockchain.
- C) A alta taxa de assaltos no Brasil, pode causar prejuízos para a empresa a assegurar os clientes.

### **3.2.2. Value Proposition Canvas**

O modelo de Value Proposition é uma ferramenta que divide e identifica alguns pontos cruciais da criação de uma solução, ou serviço para um cliente em específico. Com seu uso, podemos determinar o valor gerado em relação à solução e as necessidades dos clientes em questão. Dentre eles, podemos citar as dores, ganhos, trabalhos, produtos e serviço, criadores de ganhos (“gain creators”), aliviadores (“pain relievers”).

Na parte das dores, os riscos, experiências negativas e outros impedimentos são listados. Já na parte de ganhos, são os benefícios, desejos conquistados, aspirações e ambições, relacionada a solução, para o cliente. O trabalho se refere à atuação, em qual área e outros detalhes organizacionais.

## Proposta de Valor



## Perfil do Cliente



Proposta de valor (Imagem 1)  
Dados dos autores (2023)

### 3.2.3. Matriz de Riscos

A matriz de riscos é uma ferramenta que proporciona uma análise ampla das ameaças e oportunidades do projeto. Com ela podemos definir quais são as ameaças com maiores probabilidades e impactos no nosso projeto, além das oportunidades que são vigentes dentro do desenvolvimento. Sua principal função é auxiliar a empresa a tomar decisões baseadas nos impactos e na probabilidade desses riscos acontecerem.

Segue a lista de todos os riscos (ameaças e oportunidades) validados pela equipe:



		Ameaças					Oportunidades				
Probabilidade	90%						Maior satisfação profissional	A equipe aprender sobre o processo de criptografia	A equipe aprender mais sobre Blockchain		
	70%				Conhecimento insuficiente sobre Blockchain		Especialista em trabalhar com uma rede de seguros por parte da equipe				
	50%		Falta de dados dos clientes da Coover	Pouco contato com os clientes da Coover	Baixo conhecimento técnico das ferramentas	Problemas na parte de integração da solução	Prontidão na implementação da criptografia no projeto	O time entender sobre o setor de seguros no Brasil	Maior conhecimento do grupo sobre novas ferramentas	Flexibilidade na escolha das ferramentas	Atrair mais clientes para adquirir o serviço de seguros
	30%		Falta de credibilidade no produto	Desequilíbrio de conhecimento técnico no grupo					Inclusão de mais pessoas na Web3		
	10%	UX design não ser tão eficiente	Parte do grupo ficar desiste ou não conseguir acompanhar	Falha na criptografia	Problemas na rede do Ethereum	Não conseguir finalizar o produto a tempo	Ocorrer vazamento dos dados disponibilizados	Maior conhecimento por parte dos clientes sobre o mercado Blockchain		Conseguir trazer o sistema para mobile	Possibilidade de mais opções de criptomoedas
		Muito Baixo	Baixo	Moderado	Alto	Muito Alto	Muito Alto	Alto	Moderado	Baixo	Muito Baixo
Impacto						Impacto					

Imagem 2: Matriz de Riscos  
Dados dos autores (2023)

### 3.2.4 Matriz de avaliação de valor Oceano Azul

A análise da Matriz de Avaliação de Valor Oceano Azul é uma boa maneira de avaliar a posição da Coover em relação à concorrência e identificar oportunidades para criar valor para os clientes. Aqui está um exemplo de como a matriz poderia ser preenchida para a hipotética start-up de seguros em blockchain Coover:

Atributos: Preço, Conveniência, Transparência, Rapidez, Personalização, Tecnologia, Segurança, Suporte ao cliente

Reduzir:

- Preço: A Coover pode oferecer preços mais baixos que a concorrência, uma vez que a tecnologia blockchain permite a eliminação de intermediários e a automação de processos, o que resulta em economia de custos.

Eliminar:

- Descontos: A Coover pode não oferecer descontos, mas ao invés disso, oferecer preços acessíveis e justos para todos os clientes.

Aumentar:

- Conveniência: A Coover pode oferecer uma experiência de compra de seguros mais conveniente, como a possibilidade de comprar seguros online e acessar informações sobre seguros 24 horas por dia.
- Confiança:

Criar:

- Personalização: A Coover pode criar soluções de seguros personalizadas, baseadas nas necessidades específicas de cada cliente, usando a tecnologia blockchain para garantir transparência e segurança.

A análise da matriz de avaliação de valor Oceano Azul permite à Coover identificar suas vantagens competitivas e potenciais oportunidades de melhoria, para criar valor para os clientes. No exemplo acima, a Coover tem a oportunidade de se destacar oferecendo preços acessíveis e soluções de seguros personalizadas, que não são comuns no mercado.

Abaixo há a tabela de atributos da Avaliação do Oceano Azul, com dois players do mercado, onde é possível observar os principais pontos de diferença da Coover em relação a estes. Importante destacar que, a coover perde no SUporte ao Cliente, principalmente, por não oferecer um serviço tão efetivo quanto os outros players.

Atributos	Coover	MinutoSeguros	Kakau
Preço	8	6	5
Conveniência	7	9	7
Transparência	10	7	7
Rapidez	10	6	5
Personalização	10	8	4
Segurança	10	8	6

<b>Suporte ao Cliente</b>	<b>0</b>	<b>9</b>	<b>7</b>
<b>Blockchain</b>	<b>10</b>	<b>0</b>	<b>0</b>

A transparência, rapidez, personalização, segurança e blockchain são os pontos de destaque da Coover, principalmente pois é nestes quesitos que ela se destaca em dos outros players do mercado, principalmente por oferecer um serviço diferenciado e com altíssima segurança como o blockchain.

Abaixo segue a matriz de avaliação de valor Oceano Azul, comparando a Coover com dois players do mercado.

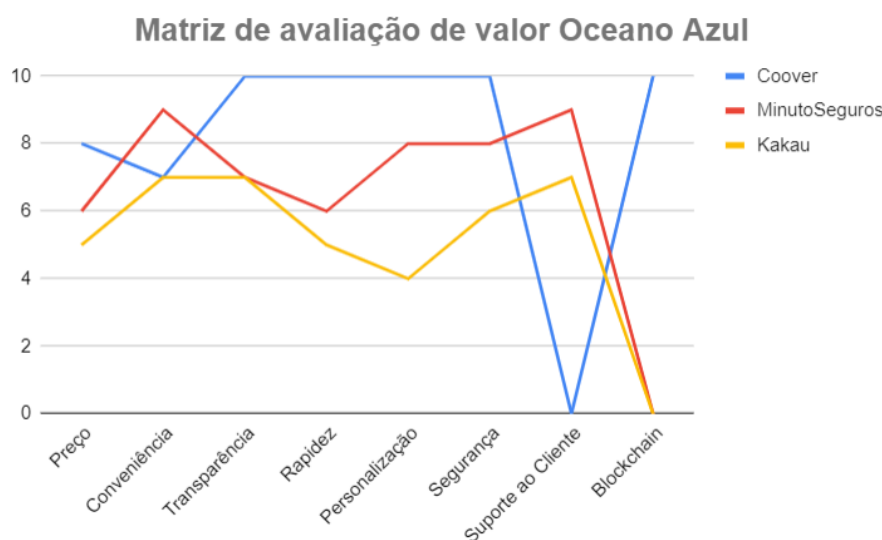


Imagem 3: Gráfico da matriz de avaliação de valor Oceano Azul  
Dados dos autores (2023)

### 3.2.5 Análise financeira do projeto

A relação entre nossas equipes e a Coover se assemelha às relações criadas entre Software houses e clientes (tirando os aspectos de burocracia, jurisdição e exigências reais). Nesse modelo, se prevê um custo fixo (mesmo que negociável) para o desenvolvimento do MVP. Devemos considerar também que a Coover é uma startup e a nossa solução não é exatamente nova ou essencial para o modelo deles. Assim, o parceiro esperaria ter custos baixos e fixos com o projeto, provavelmente ele também não daria continuidade ao MVP conosco, pois possuem uma equipe de desenvolvimento interna. Ainda em gastos, a Coover economizaria recursos em jurisdições por conta da

segurança garantida pela tecnologia Blockchain, porém, manter e desenvolver tecnologias por Blockchain pode aumentar consideravelmente os custos de operação. Dessa forma, o possível lucro real da Coover, usando nossa solução, seria captando novos clientes atraídos por um modelo novo de seguros, mais prático e seguro.

- O investimento estimado pelo cliente foi de cerca de 300 mil reais.
- Outros gastos e lucros não foram especificados pelo cliente.

## 4. Análise de Experiência do Usuário

Análise de Experiência do Usuário (UX, User Experience Analysis) é uma metodologia que busca compreender e melhorar a interação do usuário com um produto. Ela é baseada em dados de pesquisas com os usuários, que permitem identificar pontos de dor, dificuldades, desafios e necessidades. No caso, utilizamos as seguintes ferramentas para melhor compreender os usuários do sistema: Personas, jornada do Usuário e User Story.

### 4.1. Personas

Em relação à nossa persona, podemos citar ela como uma representação do nosso cliente. Consequentemente, possuem as principais características condizentes com a dor, necessidade e sua compatibilidade com a solução. Elas têm como propósito criar uma orientação através de um perfil que sintetize as principais características dos clientes e beneficiários. Para que possamos verificar a capacidade de atender suas demandas e manter o projeto focado nas necessidades do cliente.ixo:

**Persona 1:** A persona "Gustavo Almeida", foi criada visando as necessidades de um usuário comum da plataforma, logo, nela estão contidas informações relevantes

sobre as características de quem vai participar do seguro como um membro comum do contrato, como é possível observar na representação abaixo:



**Persona 2:** A persona "Sergio Alcantra", foi criada visando as necessidades de um gerente de seguros da empresa que acessa a plataforma, logo, nela estão contidas

informações relevantes sobre as características de quem vai administrar os pedidos de seguros do smart contract, como é possível observar na representação abaixo:



As personas estruturadas nos ajudam, pois criam um perfil detalhado e fictício de um usuário-alvo que, com base nas informações coletadas, é considerado representativo da maioria dos usuários reais do produto. Isso nos permite a entender melhor as

necessidades, desejos, expectativas e comportamentos de seus usuários, o que ajuda a informar as decisões de design e desenvolvimento.

## **4.2. Jornadas do Usuário e/ou Storyboard**

A jornada do usuário é uma ferramenta utilizada para mapear a experiência do usuário de maneira gráfica, explicitando os sentimentos, ações e etapas que o usuário passa. De modo que é utilizado as personas para a sua modelagem e através das características dela é pensado nas responsabilidades, oportunidades e expectativas dela. Para que assim, seja pensado na melhor experiência possível do usuário que utiliza o produto. Segue abaixo as jornadas de usuário referente à nossas personas:

### **Persona 1:**



## Gustavo Almeida

**Cenário:** Gustavo Almeida precisa de um sistema de protocolo P2P que utilize uma tecnologia transparente e segura. Além disso, como cliente de uma seguradora, deseja ter um sistema de contrato menos burocrático e automatizado.

## Expectativas

Gustavo espera melhorar a sua usabilidade com a plataforma da seguradora e tornar o seu processo de assinatura de contratos mais eficaz e seguro.

FASE 1 (Cadastro na plataforma)	FASE 2 (Entrada em um grupo)	FASE 3 (Recebimento do smart contract)	FASE 4 (Validação do smart contract)	FASE 5 (Solicitando uma indenização)
<ol style="list-style-type: none"><li>1. Realizar o seu cadastro na plataforma com o seu e-mail e senha.</li><li>2. O login também pode ser realizado com a sua carteira digital da metamask.</li></ol> <div>Gostaria de fazer o meu cadastro. Será que consigo entrar com a minha carteira da metamask?</div>	<ol style="list-style-type: none"><li>1. Na página inicial da plataforma, terá opções de grupos de seguros mútuos que ele pode escolher para participar</li><li>2. O cliente também pode criar um grupo novo.</li></ol> <div>Qual grupo será que faz mais sentido com o meu perfil e com as minhas necessidades?</div>	<ol style="list-style-type: none"><li>1. Cada grupo receberá um smart contract, o qual conterà as regras e os termos gerais para ser membro de determinado grupo.</li><li>2. No smart contract terão regras definidas pelos clientes, como por ex: qual valor cada membro depositará mensalmente.</li></ol> <div>Acabei de receber o smart contract, será que concordo com as regras?</div>	<ol style="list-style-type: none"><li>1. Assim que estiver inserido em um grupo, os membros receberão um smart contract de acordo com o consenso de cada participante.</li><li>2. O cliente terá que aceitar os termos do smart contract, e linkar sua carteira, para ser membro do grupo.</li></ol> <div>Agora que recebi o smart contract, posso validá-lo e ver se realmente quero participar desse sistema.</div>	<ol style="list-style-type: none"><li>1. Dentro da plataforma, terá serviços oferecidos para os clientes, como por ex: processo de adesão, pagamento e pedido de indenização.</li><li>2. O cliente pode solicitar um pedido de indenização.</li><li>3. O seguro será acionado e poderá aceitar ou rejeitar o pedido.</li></ol> <div>Preciso de uma indenização, pois o meu celular caiu dentro da piscina, será que o seguro aceitará?</div>

## Oportunidades

As oportunidades serão que os clientes poderão utilizar a plataforma de seguros de uma maneira menos burocrática, mais segura, mais eficiente e mais acessível. Portanto, utilizarão um sistema de contrato inteligente, em que não haverá intermediações, ou seja, será mais ágil e trará uma maior segurança para os usuários.

## Responsabilidades

A responsabilidade será atribuída para os clientes, uma vez que eles escolhem o grupo que desejam participar, leem os termos de aceitação dos smart contracts e validam se realmente é aquele tipo de contrato e grupo que desejam participar.

## Persona 2:





### Sergio Alcantra

**Cenário:** Como gestor de seguros da Coover, Sergio é o responsável por aprovar as retiradas do contrato. Após, entrar na plataforma, e receber uma solicitação de ativação do seguro, ele precisa analisar e tomar uma decisão.

### Expectativas

Sergio espera conseguir analisar e tomar uma decisão quanto a um pedido de ativação de seguro.

FASE 1 (Acessar a plataforma)	FASE 2 (Reber solicitação para ativa do seguro)	FASE 3 (Analisar a solicitação de seguro)	FASE 4 (Emitir sua decisão)	FASE 5 (dar Feedback para o membro que solicitou)
<ol style="list-style-type: none"><li>1. Acessa plataforma através de sua conta de Gerente de projetos.</li><li>2. Carrega a main page da solução Web.</li><li>3. Acessa a área de solicitações de seguro.</li></ol>	<ol style="list-style-type: none"><li>1. Confirma quais foram as solicitações de seguro que recebeu, referentes aos contratos os quais ele gerência.</li><li>2. Acessa um dos pedidos de seguro.</li></ol>	<ol style="list-style-type: none"><li>1. Lê atentamente a toda solicitação do seguro.</li><li>2. De acordo com os fatos, dados de suporte e sua análise, toma uma decisão quanto a o caso.</li></ol>	<ol style="list-style-type: none"><li>1. Aprova ou rejeita a solicitação.</li></ol>	<ol style="list-style-type: none"><li>1. Fornece um feedback ao membro do contrato sua sua solicitação de ativação do seguro.</li></ol>
<div>Hora de começar o trabalho.</div>	<div>É, vamos ver quanto trabalho tenho para hoje.</div>	<div>Vamos ver, qual a situação dessa vez?</div>		<div>Como posso dar essa notícia para ele?</div>

### Oportunidades

A solução proporciona para ele uma forma rápida e eficiente, para aprovar ou rejeitar solicitações de ativação de seguro. Algo que normalmente teria mais burocracia e gastaria mais tempo.

### Responsabilidades

Aprovar ou rejeitar as solicitações de seguro emitidas pelos cliente dos contratos de que ele faz parte.

## 4.3. User Stories

O template de User Story é uma ferramenta utilizada para descrever as necessidades e os requisitos do usuário de um produto, sob o ponto de vista deste usuário. Ele busca simplificar a descrição dessas necessidades, tornando-as objetivas e concisas.

O template utilizado de User Story consiste em dois aspectos principais: o Cartão e a Confirmação.

O Cartão descreve a necessidade do usuário utilizando três parâmetros: “QUEM”, “O QUÊ” e “POR QUÊ”. Esses parâmetros representam, respectivamente, quem é o

usuário que tem a necessidade, qual é a necessidade e qual é o benefício direto obtido pelo usuário com a solução.

Já a Confirmação consiste em critérios e testes de aceitação que estabelecem como a funcionalidade deve se comportar uma vez implementada. Esses critérios são representados por enunciados pequenos e de fácil entendimento, e os testes são criados para verificar se a funcionalidade está se comportando conforme o esperado.

Em resumo, o template de User Story é uma ferramenta importante para garantir que as necessidades e requisitos do usuário estejam claramente definidos e sejam priorizados no desenvolvimento de um produto. Ele ajuda a garantir que o produto atenda às expectativas e necessidades dos usuários e fornece uma base sólida para a criação de soluções eficazes.

Para acessar as User Stories da Coover basta clicar no link abaixo:

[User Story - Coover](#)

Número	Descrição	Critérios de aceitação	Teste de aceitação	Tamanho	Prioridade	Status
1	Como gerente de seguros, quero ter a permissão para aprovar as solicitações de retirada dos smart contracts do qual faço parte, para ter controle sobre a ativação do seguro.	Precisa-se de um contrato para que os gerentes tornem-se pessoas com permissão para aprovar as solicitações de retiradas dos smart contracts.	<p>O gerente acessa a página web, e visualizar suas solicitações de retirada de seguro.</p> <p>Se o gerente tem essa permissão aceita = Correto.</p> <p>Se essa permissão é rejeitada = Incorreto, deve-se implementar.</p>	Médio	Alta	Pendente
2	Como gerente de seguro, quero poder visualizar todas as regras e	Após os smart contracts terem sido concluídos, é	Tentativa do gerente de seguros realizar o acesso a um smart contract.	Grande	Alta	Pendente

	condições do smart contract antes de ele entrar em vigor, para validar antes de os subir na rede.	necessário uma página de avaliação para que os gerentes possam avaliá-los e validá-los. Se os contratos forem validados pelos gerentes, deve-se ter um botão de ok/aceito que encaminha diretamente os smart contracts aceitos para a rede.	Se os gerentes de seguro conseguem visualizar todas as regras e condições dos smart contracts = Correto.  Caso seja rejeitada essa visualização = Incorreto.			
3	Como usuário, quero uma forma de solicitar o seguro, para poder retirar o dinheiro do smart contract.	Caso o usuário deseje retirar o seu dinheiro do smart contract, é necessário um meio de solicitação.	O usuário através do frontend, realiza uma solicitação de ativação do seguro.  O gerente de seguros recebeu = Correto A solicitação não chegou a na plataforma web = Incorreto, deve ser corrigido	Médio	Alta	Pendente
4	Como gerente de seguro, quero ter permissão para	Apenas o gerente de seguros terá a	Um usuário comum tenta executar uma função only owner:	Grande	Baixa	Em andamento

	executar funções que apenas o administrador pode executar.	permissão de chamar certas funções do contrato.	<p>Aceitou = Incorreto, a permissão deve ser revista Recusou = Correto</p> <p>O gerente tenta executar uma função only owner: Aceitou = Correto Rejeitou = Incorreto, a permissão deve ser revista.</p>			
5	Como gerente de seguro, quero cobrar de uma carteira geral o percentual da Coover referente aos seus serviços prestados, para conseguir ter caixa na empresa.	Deve existir uma carteira geral para que todas as cobranças dos grupos, sejam encaminhadas para ela.	<p>O contrato deve definir uma carteira central do projeto que será depositado o valor de todos os usuários dos grupos</p> <p>Define = Correto</p> <p>Não define = Incorreto, necessário adicionar carteira para receber os fundos.</p>	Grande	Alta	Em andamento
6	Como usuário, quero poder visualizar as pessoas dentro do meu contrato de forma segura, para poder ver a	Ao requisitar, o usuário deve visualizar as carteiras e quantidade de usuários em seu contrato.	<p>O usuário realiza uma requisição para receber informações sobre os membros de seu contrato;</p> <p>Aceita = Correto</p>	Médio	Alta	Em andamento

	quantidade de usuários dentro do mesmo contrato que o meu.		Recusou = Incorreto, precisa ser adicionado permissão			
7	Como usuário, quero poder verificar se o smart contract está chegando no vencimento, para poder estar ciente das mudanças em seu funcionamento e/ou entrar em outro seguro mútuo.	Caso o contrato esteja chegando na validade, como membro desse contrato, deve receber uma notificação indicando que o contrato foi alterado.	O usuário acessa a plataforma, após a aproximação da data de validade em um contrato do qual ele participa. Recebeu notificação = Correto Nada ocorreu = Incorreto, deve ser corrigido	Médio	Alta	Em andamento
8	Como gerente de seguros, quero que o smart contract só entre em pleno vigor, caso as condições de funcionamento sejam atendidas. Para que tudo funcione conforme o acordado.	Certas funções só devem funcionar caso o contrato esteja atendendo aos requisitos de funcionamento.	O Gerente de seguros, caso tente executar uma função que necessita que as condições de funcionamento tenham sido atingidas, mesmo elas não tendo sido realmente correspondidas.  Solicitação aceita: Incorreto, a permissão deve ser revisada.  Recusou = Correto.	Pequeno	Alta	Pendente
9	Como usuário, quero saber a quantidade de pessoas dentro dos grupos para	A função de entrar em vigor deve conter o número	O usuário acessa o grupo e verifica a quantidade de pessoas, Se o grupo tiver o mínimo de pessoas	Médio	Alta	Em andamento

	saber se ele está em progresso ou ativo, para ser segurado o mais rápido possível.	mínimo e máximo de usuário permitido, juntamente com a validade do contrato com a data de entrada em vigor, caso cumpra os requisitos.	<p>viáveis para a execução do contrato e o usuário clicar em entrar e concordar com o contrato</p> <p>-Solicitação aceita = entrou no grupo</p> <p>Caso não possuo o mínimo de pessoas, e ainda não chegou a data de entrada de vigor</p> <p>-Solicitação em andamento = mensagem de grupo sendo montado</p> <p>Caso, não possua o número mínimo de pessoas e alcance a data de entrada, sem atingi-lo.</p> <p>- Solicitação cancelada = mensagem para o usuário.</p>			
10	Como gerente de seguros, quero que após uma solicitação de indenização ser aprovada, quero poder transferir os dinheiros do fundo do smart contract para a	O gerente de contratos deve poder acionar uma função para transferir dinheiro do smart contract para um usuário de teve sua solicitação de	<p>O gerente de seguros realiza a ativação de uma função para transferir dinheiro do smart contract para a carteira de um usuário</p> <p>Solicitação de transferência aceita = correto</p> <p>Solicitação de transferência negada =</p>	Médio	Alta	Em andament o

	carteira de quem deve ser assegurado. Para garantir que os usuários recebam o dinheiro.	seguros aprovada.	<p>incorreto, a permissão deve ser revisada.</p> <p>Caso a transferência tenha sido aceita:</p> <p>Usuário recebeu o dinheiro = correto</p> <p>Não recebeu o dinheiro = incorreto, função de transferência deve ser revisada</p>			
11	Como gerente do smart contract quero receber uma taxa de adm com um percentual definido pelo contrato.	Deve ser transferido um valor para o gerente do smart contract, valor o qual trata-se de uma taxa percentual do adm.	<p>Transferência de um valor (taxa do adm), ou seja, um percentual do valor pago pelos clientes do contrato = Correto</p> <p>Caso a transferência não tenha sido feita = Incorreto, a função de transferência deve ser revisada.</p>	Médio	Alta	Em andamento
12	Como gerente quero que os clientes paguem pelo seguro um percentual correspondente ao valor do seu celular.	O cliente deve transferir um valor correspondente e ao percentual do valor do seu celular.	<p>Transferência de valor pelo usuário</p> <p>Transferiu um valor correspondente ao percentual do seu celular = Correto</p> <p>Não transferiu um valor correspondente ao percentual do seu celular = Incorreto.</p>	Médio	Alta	Em andamento

## 4.4. Protótipo de interface com o usuário

Protótipo da interface/parte visual da solução.

[Link do Figma - WireFrame](#)

- O seguinte link acima, possui 2 páginas principais a serem consideradas, sendo elas:
  - Wireframe novo - Usuário
  - Wireframe novo - Seguradora

## 5. Solução proposta

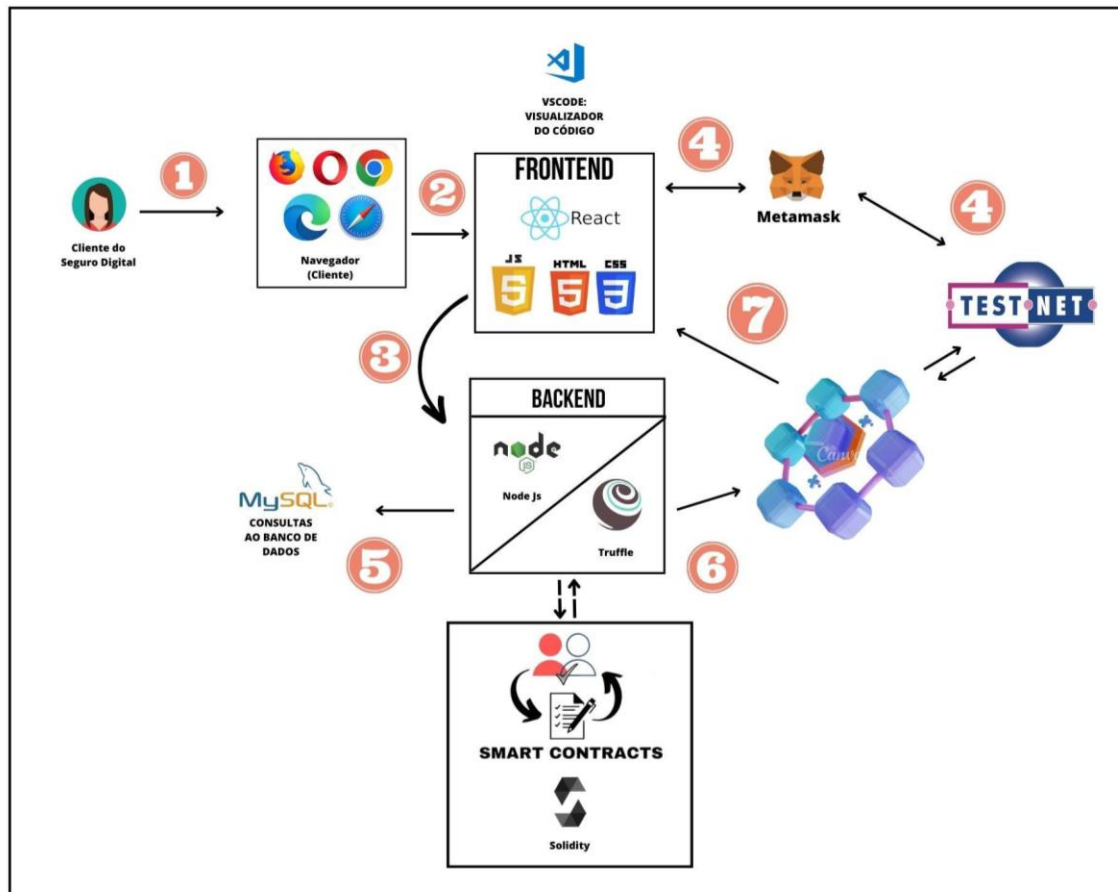
### 5.1. Solução

Nós do grupo BlockCoovers desenvolvemos, a partir da tecnologia blockchain, uma plataforma Web3 para criar e desenvolver contratos inteligentes, os quais permitam a realização de operações e validações de processos de forma segura e eficiente em um TestNet aberto. Este modelo inovador de seguro mútuo combinado com a tecnologia blockchain garantirá uma cultura de transparência e confiança na indústria de seguros.

### 5.2. Arquitetura proposta

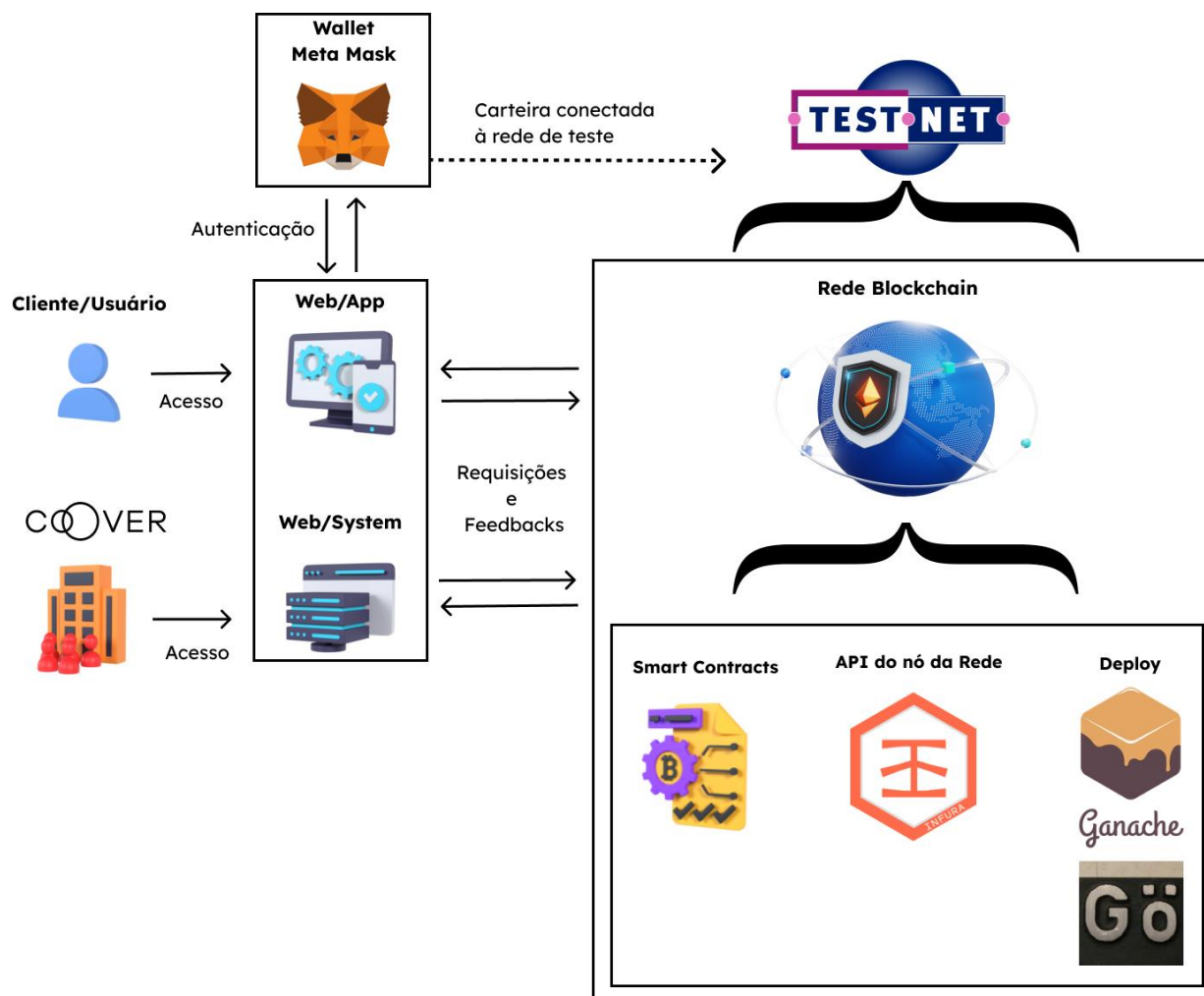
A arquitetura da nossa solução é uma parte muito importante para entendimento do projeto, e implementação da solução. Ela é necessária, pois complementa a viabilidade do projeto como um todo, exemplificando as etapas presentes na solução:





### 5.3. Diagrama Macro da Solução

Abaixo segue o diagrama da Macro solução representada como um diagrama de blocos.



## 5.4. Descrição da Solução

A diagramação em blocos é uma técnica de organização visual de informações em que o conteúdo é dividido em blocos, cada um com uma função específica e sendo possível fazer linhas que representem as conexões ou interações entre as funções. A diagramação em blocos será usada como forma de criar uma hierarquia visual clara, com os elementos mais importantes e relevantes em destaque, refletindo em um melhor entendimento.

## 5.5. Análise de Risco (Segurança da Informação)

### 5.5.1. Requisitos de segurança da informação

**Autenticidade** - verdadeira autoria da informação, por exemplo você está tendo a certeza de que está falando com x pessoa e não alguém se passando por ela.

- Garante que os dados são de fato provenientes de determinada fonte, assegurando que foram criados, expedidos, alterados ou destruídos por certo órgão, entidade ou sistema.

**Disponibilidade** - funcionamento adequado da rede e à eficácia dos sistemas.

- Garante que as informações estejam à disposição de gestores e colaboradores para consulta a qualquer momento, assegurando sua acessibilidade.

### 5.5.2. Identificar as vulnerabilidades existentes

#### 1. Reentrancy

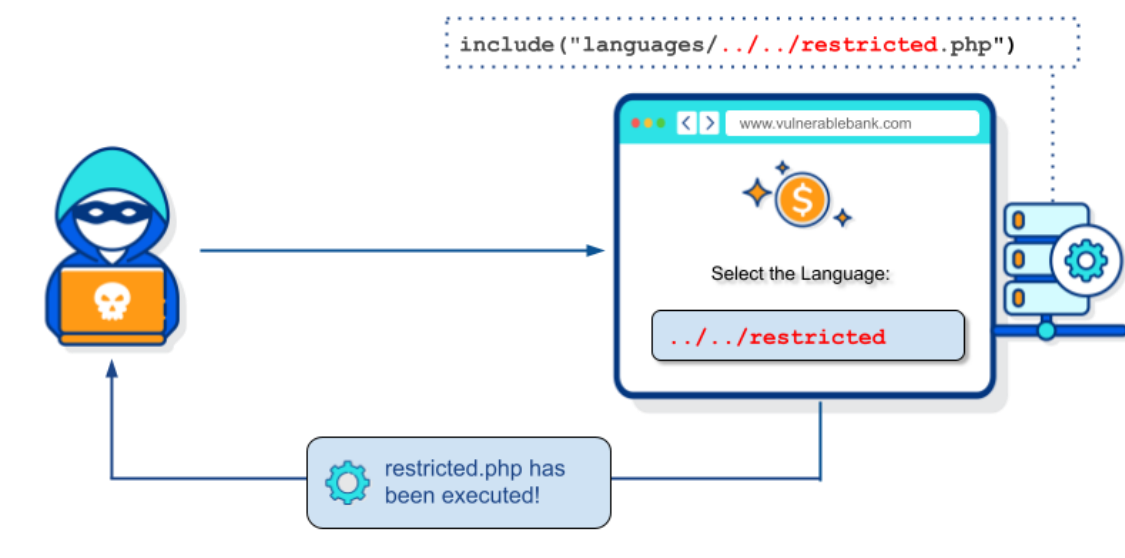


A reentrancy é uma vulnerabilidade presente em smart contracts que permite a um invasor drenar os fundos do contrato. Isso ocorre quando uma função de um contrato confiável chama uma função de um contrato não confiável, permitindo que este contrato não confiável faça uma chamada recursiva de volta à função original.

### Exemplo:

Um exemplo típico de reentrancy é quando um contrato mantém um registro de saldo em uma variável interna e possui uma função de saque. Se o contrato vulnerável transferir fundos antes de atualizar o saldo para zero, um invasor pode chamar repetidamente a função de saque de forma recursiva, esvaziando todo o saldo do contrato. Para evitar esse tipo de vulnerabilidade, é importante implementar medidas de segurança apropriadas em todos os contratos smart.

## 2. File inclusion



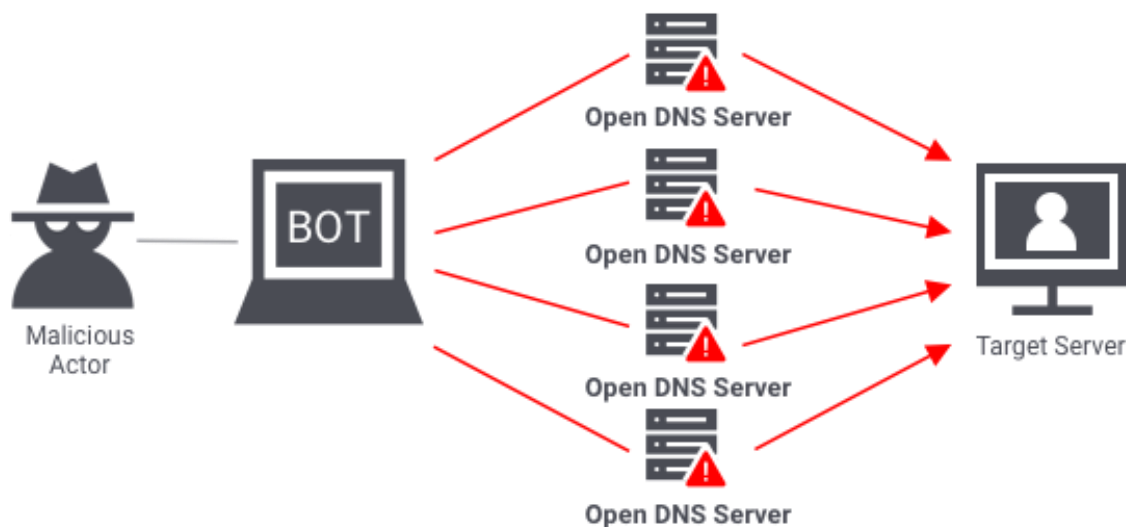
File Inclusion é uma vulnerabilidade de segurança de aplicativos web que permite a um atacante incluir arquivos remotos no sistema. Isso pode ser explorado para executar código malicioso no servidor, acessar informações sensíveis ou causar danos ao sistema. As duas formas mais comuns de file inclusion são Local File Inclusion (LFI) e Remote File Inclusion (RFI). A prevenção inclui validação rigorosa dos inputs do usuário, configuração correta do servidor e aplicação de patches de segurança quando disponíveis.

### Exemplo:

A funcionalidade de incluir fotos de perfil na página pode ser uma vulnerabilidade de segurança, se a validação dos arquivos não for realizada corretamente. Um atacante pode explorar essa vulnerabilidade ao inserir scripts maliciosos, ao invés de uma imagem de perfil. Esses scripts maliciosos serão carregados dentro do servidor e podem causar uma série de problemas de segurança, como execução de código malicioso, revelação de informações sensíveis e tomada de controle do sistema.

### 5.5.3. Atacantes e levantar ataques possíveis

#### 1. DDOS (distributed denial-of-service)



**DDOS** (Distributed Denial of Service) ocorre quando um atacante envia diversas requisições ao servidor em um curto espaço de tempo, resultando em sobrecarga. Isso faz com que o servidor pare de responder, não só para ele, mas para todos os usuários que estariam tentando utilizar o servidor no momento.

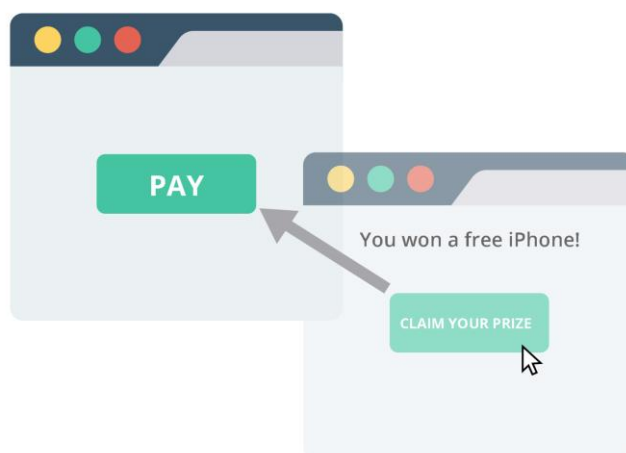
### Exemplo:

Imagine que um site de comércio eletrônico é alvo de um ataque DDOS. O atacante coleta centenas de computadores infectados com malware e os usa para

enviar solicitações simultâneas ao site de comércio eletrônico. Como resultado, os servidores do site ficam sobrecarregados e não conseguem lidar com o tráfego legítimo, tornando o site inacessível para os usuários.

Este é apenas um exemplo básico de como um ataque DDOS pode funcionar. Na prática, esses ataques podem ser muito mais complexos e utilizar várias técnicas diferentes para tornar um site ou serviço inacessível.

## 2. Clickjacking



Clickjacking é uma técnica de segurança da informação que permite a um atacante controlar cliques de um usuário em uma página da web, de forma a direcioná-los para realizar ações não intencionais, como por exemplo, clicar em um botão malicioso ou instalar malware. Isso é conseguido colocando uma página legítima por cima de outra página maliciosa, de tal forma que o usuário acredita estar clicando em elementos da primeira página, enquanto na verdade está clicando em elementos da página maliciosa. É importante que as páginas sejam protegidas contra esse tipo de ataque para garantir a segurança dos usuários.

## 3. Phishing



Phishing é uma técnica de fraude cibernética que visa obter informações confidenciais, como nomes de usuário, senhas e informações financeiras, através da imitação de fontes confiáveis, como instituições financeiras, redes sociais e outros serviços on-line. Isso geralmente é feito através de mensagens de e-mail ou links para páginas falsas que parecem ser legítimas, mas na verdade são criadas pelos atacantes para roubar informações pessoais

**Exemplo:**

Imagine que você receba um e-mail aparentemente de sua instituição financeira, solicitando que você atualize suas informações de conta clicando em um link fornecido no e-mail. No entanto, ao clicar no link, você é direcionado a um site falso que parece ser a página de login da instituição financeira, mas na verdade é controlado por criminosos. Quando você insere suas informações de login, elas vão diretamente para os criminosos, que podem usá-las para acessar sua conta bancária e roubar seu dinheiro. Esse é um exemplo de phishing, onde os criminosos estão fingindo ser uma empresa confiável para obter informações pessoais sensíveis de suas vítimas.

#### 5.5.4. Priorizar os ataques de acordo com a sua probabilidade e impacto

			Matriz de Risco de Segurança da Informação		
PROBABILIDADE					
	Alta	3		DDOS (Distributed Denial fo Service)	Reentrancy
	Médio	2	Clickjacking		File inclusion
	Baixa	1	Phishing		
			1	2	3
			Baixo	Médio	Alta
			IMPACTO		

## 6. Desenvolvimento e Resultados

Nessa sessão iremos apresentar os dois módulos que temos no nosso contrato: O Smart Contract e o FrontEnd.

### 6.1. Módulo Smart Contract

O smart contract possui como objetivo geral armazenar e garantir que os parâmetros para o funcionamento do contrato sejam atendidos, armazenando informações úteis sobre os membros do contrato, garantindo que as transações sejam feitas de forma segura e previsível. Garantindo que o que foi acordado vai ser executado de forma segura e imutável.



### 6.1.1. Descrição do Smart Contract

O arquivo smartContract.sol possui as seguintes funcionalidades e características. Abaixo é possível visualizar as variáveis responsáveis para o funcionamento do contrato.

#### Variáveis do contrato:

As variáveis globais ou variáveis de estado são declarações de variáveis que ficam no nível do contrato e podem ser acessadas por todas as funções do contrato. Elas são usadas para armazenar informações importantes para o funcionamento do contrato. No código fornecido, existem várias variáveis globais, como

- **quantUsuario**: variável do tipo uint, ela é responsável por armazenar a quantidade de usuários dentro do contrato.
- **minPessoas** e **maxPessoas**: variável do tipo uint, são responsável por armazenar a quantidade mínima e máxima de usuários permitidos por contrato.
- **taxaAdmin**: variável tipo uint que armazena a taxa de administração do contrato.
- **carteiraCentral**: variável do tipo address, que é responsável por armazenar informações sobre a carteira da Coover.
- **fundosAdm**: variável tipo uint que armazena o total de fundos coletados como taxa de administração.
- **carteira**: armazena a lista de carteiras de usuários, que são definidas usando a struct Carteira.
- **admin**: variável do tipo address, onde armazena o endereço do proprietário do contrato, que é o administrador.
- **Aprovar[]**: Armazena a lista de usuários que estão esperando aprovação para entrar no contrato. A estrutura Aprovar tem os campos carteiraUsuario (endereço da carteira do usuário), valorAparelho (valor do aparelho do usuário) e IMEI (número de identificação do aparelho).

- **Indenizacao[]**: Armazena a lista de solicitações de indenização feitas pelos usuários do contrato. A estrutura Indenizacao tem os campos carteiraUsuario (endereço da carteira do usuário solicitante), valorPedido (valor do pedido de indenização), justificativa (texto explicando a razão da solicitação) e imeiSolicitacao (número de identificação da solicitação).

### **Structs do contrato:**

O "struct Carteira" é uma estrutura de dados que é usada para armazenar informações sobre a carteira de um usuário. No código fornecido, a struct Carteira armazena o endereço da carteira do usuário, o valor do aparelho celular, o saldo da carteira do usuário e o número de IMEI do aparelho celular.

Abaixo há a estrutura dentro do struct Carteira definida no nosso smart contract.

- **carteiraUsuario**: é um elemento do tipo "address" e armazena o endereço da carteira do usuário.
- **valorAparelho**: é um elemento do tipo "uint" e armazena o valor do aparelho celular do usuário.
- **saldo**: é um elemento do tipo "uint" e armazena o saldo da carteira do usuário.
- **IMEI**: é um elemento do tipo "uint" e armazena o número de IMEI do aparelho celular do usuário.

Esses elementos são usados para armazenar informações relevantes sobre a carteira do usuário dentro do smart contract.

O "struct Aprovar" é uma estrutura para armazenar um pedido de entrada no smart contract, contendo os seguintes atributos:

Abaixo há a estrutura dentro do struct Aprovar definida no nosso smart contract.

- **carteiraUsuario**: endereço da carteira do usuário que solicitou a entrada no contrato;
- **valorAparelho**: valor do aparelho do usuário que solicitou a entrada no contrato;

- **IMEI**: número de identificação do aparelho do usuário que solicitou a entrada no contrato.

Essas informações são armazenadas na estrutura Aprovar até que o proprietário do contrato as aprove ou rejeite.

O "struct Indenizacao" é utilizado para armazenar os pedidos de indenização realizados pelos usuários. Se um usuário precisar de uma indenização por ter sofrido um dano em seu aparelho, ele poderá enviar uma solicitação contendo o endereço de sua carteira, o valor do pedido, a justificativa e o número IMEI da solicitação.

Abaixo há a estrutura dentro do struct Indenizacao definida no nosso smart contract.

- **carteiraUsuario**: o endereço da carteira do usuário que solicitou a indenização;
- **valorPedido**: valor solicitado pelo usuário na indenização;
- **justificativa**: descrição da justificativa para a indenização solicitada pelo usuário;
- **imeiSolicitacao**: número de identificação da solicitação de indenização.

Essas informações são armazenadas na estrutura Indenizacao para posterior avaliação e resolução do proprietário do contrato.

### **Constructor:**

O constructor é uma função especial que é executada apenas uma vez, quando o contrato é criado. É usado para inicializar as variáveis globais do contrato com valores iniciais. No código fornecido, o constructor recebe três parâmetros: `_minPessoas`, `_maxPessoas` e `_taxaAdmin`.

Dentro do constructor, são executadas as seguintes ações:

- **quantUsuario = 0**;: inicializa a variável global `quantUsuario` com o valor zero.
- **admin = msg.sender**;: define o endereço que criou o contrato como o proprietário/administrador do contrato.

- **minPessoas = \_minPessoas;** define o valor do parâmetro `_minPessoas` como o valor inicial da variável global `minPessoas`.
- **maxPessoas = \_maxPessoas;** define o valor do parâmetro `_maxPessoas` como o valor inicial da variável global `maxPessoas`.
- **taxaAdmin = \_taxaAdmin;** define o valor do parâmetro `_taxaAdmin` como o valor inicial da variável global `taxaAdmin`.
- **carteiraCentral = payable(msg.sender);** define o endereço que criou o contrato como a carteira central, que é a carteira que receberá a taxa de administração do contrato. O `payable` indica que a carteira pode receber pagamentos.
- **fundosAdm = 0;** inicializa a variável global `fundosAdm` com o valor zero.

### Modifiers:

O nosso smart contract define um modificador chamado `apenasAdmin`. Esse modificador é utilizado para limitar o acesso a uma função somente ao proprietário do contrato (administrador).

O modificador é uma função especial que modifica o comportamento de outra função. No caso desse modificador, ele é adicionado antes de uma função que deve ser executada somente pelo administrador. Se a pessoa que chama a função não for o administrador, a execução da função será interrompida e a mensagem de erro "Somente o proprietario do contrato pode executar esta funcao." será exibida.

O modificador "`apenasAdmin`" ajuda a garantir que somente o proprietário do contrato possa executar funções críticas e sensíveis que poderiam afetar o estado do contrato ou os fundos mantidos dentro dele.

- **apenasAdmin():** modificador responsável por garantir que apenas o proprietário do contrato irá poder executar uma função.

### Funções do contrato:

Uma função no smart contract é um trecho de código que realiza uma tarefa específica dentro do contrato. As funções podem ser chamadas por outras funções

dentro do contrato ou por usuários externos ao contrato, através de transações na blockchain.

Uma função pode receber parâmetros de entrada, executar um conjunto de instruções e retornar um valor ou uma transação. As funções podem modificar o estado do contrato, que é a informação armazenada na blockchain, ou apenas consultar o estado atual do contrato, sem modificá-lo.

As funções são essenciais para a interação dos usuários com o contrato e para a execução das regras de negócio definidas pelo contrato. Abaixo há a lista de todas as funções utilizadas no smart contract:

- **adicionarUsuario:** Essa função adiciona um novo usuário ao projeto, verifica se a quantidade máxima de usuários já foi atingida, adiciona o usuário à lista de carteiras com seu saldo e valor do aparelho celular e incrementa a quantidade de usuários.
- **adicionarDinheiro:** Essa função é utilizada para uma carteira adicionar dinheiro no smart contract, ela também armazena a taxa do administrador, armazena o valor do depósito, armazena o valor que o usuário deve receber, uma vez calculada a taxa administrativa. Essa função também passa pelas carteiras de membros do contrato, verifica se a carteira do usuário corresponde ao endereço de quem chamou a função, adiciona os fundos a carteira do usuário, tirando a taxa administrativa e a transferindo para a Coover.
- **verSaldo:** Essa função é utilizada para ver o saldo da carteira que está chamando a função, onde verifica se a carteira do usuário corresponde ao endereço fornecido e retorna o saldo do usuário.
- **contratoSaldo:** Essa função é utilizada para ver quanto de dinheiro tem dentro do contrato e retorna quanto de fundo o contrato possui.
- **visualizarCarteiras:** Essa função que permite visualizar carteiras participantes do contrato e retorna as carteiras.

- **verSaldoAdmin:** Essa função é utilizada para ver o saldo do administrador do contrato e retorna o saldo do administrador.
- **retirarTaxas:** Essa função é utilizada para o administrador conseguir retirar as taxas do contrato (retirar o valor que o administrador ganha, o qual já estará destinado a ele), ela também confere se o contrato tem dinheiro suficiente e transfere o valor já "separado" das taxas cobradas para o administrador, quando chama esta função.
- **indenizar:** Essa função é utilizada para transferir a indenização a um usuário, conferindo se o contrato tem dinheiro suficiente. Ela também retira primeiro todo o saldo de quem pediu a indenização e transfere o valor da indenização para o usuário.

### 6.1.2. Tecnologias adotadas no Smart Contract

Para melhor compreensão das tecnologias adotadas para a construção do Smart Contract apontamos como tópicos abaixo:

#### Cliente/Usuário

- Os clientes são os proprietários dos aparelhos e os que desejam contratar um seguro. Nesse sentido, além de fornecerem as informações referentes aos seus próprios aparelhos (para contratação do seguro), irão se conectar à plataforma para navegarem nas diversas oportunidades.
- Obs: Também precisarão se conectar com uma carteira a partir da MetaMask.

#### Metamask

- MetaMask é uma carteira digital para armazenamento de criptomoedas que permite ao usuário interagir com aplicativos descentralizados baseados em Ethereum. Dessa forma, ela engloba diversas funcionalidades em uma única extensão – todas atreladas a aplicativos descentralizados (dApps). No caso da

Coover, a Metamask é utilizada para realizar transacionar de forma mais simples e segura.

## **Smart Contracts**

- Sistema onde os termos e os conteúdos são estruturados, englobando todas as regras de negócio do seguro mútuo. São executados na rede blockchain e possuem todas as cláusulas que são acordadas entre os integrantes e a seguradora, de acordo com a regra de negócio. Não só isso, mas automatizam as transações relacionadas à indenizações, ao momento que são aprovadas pela Coover.

## **Deploy**

- Deploy é o processo de implantação de um smart contract na blockchain, tornando-o disponível para ser usado e executado pelos usuários da rede. A principal diferença entre o deploy na rede Goerli e na rede Ganache é que a primeira é uma rede de testes pública que permite aos desenvolvedores testar seus contratos em um ambiente semelhante ao da rede principal da Ethereum, enquanto a segunda é uma rede de testes privada que permite aos desenvolvedores testar contratos em um ambiente controlado localmente.

## **API do Nó da Rede**

- A Infura é uma plataforma de infraestrutura em nuvem que fornece acesso a APIs de nós de blockchain, incluindo Ethereum, IPFS e outros. A API Infura permite que desenvolvedores e usuários finais se conectem a esses nós sem a necessidade de executar seus próprios Nós. Isso simplifica e acelera o desenvolvimento de aplicativos descentralizados e serviços relacionados à blockchain.

## **Rede Blockchain**

- Rede responsável por guardar os Smart contracts, além de garantir segurança e imutabilidade dos registros, de forma que as transações não necessitem de um intermediário.

## TestNet

- Rede de testes na Blockchain, usada com objetivo de prevenir erros e riscos de gastos desnecessários, além de ser visto como local para testes e validações, antes de ser aplicado na Mainnet (Rede principal).

### 6.1.3. User Stories referentes ao Smart Contract

As User Stories utilizadas para atender o Smart Contract são, as user stories 4, 5, 6, 8 e 10. Abaixo uma explicação de como elas foram utilizadas.

Para a User Story 4, o requisito de negócio implementado é referente à permissão para executar funções exclusivas do administrador do contrato. Foi desenvolvido um modifier chamado "apenasAdmin()" que garante que apenas o dono do contrato tenha acesso às funcionalidades direcionadas ao administrador. Para utilizar esse código, é necessário utilizar esse modifier nas funções que devem ser restritas apenas ao administrador.

Para relembrar a **User Story 4: Como gerente de seguro, quero ter permissão para executar funções que apenas o administrador pode executar.**

Para a User Story 5, o requisito de negócio implementado é referente à cobrança de um percentual da Coover em uma carteira geral. Foi desenvolvido um struct chamado "carteiraCentral" que define essa carteira central para receber fundos do contrato. Para utilizar esse código, é necessário instanciar esse struct e transferir os fundos da carteira geral para a carteira da Coover.

Para relembrar a **User Story 5: Como gerente de seguro, quero cobrar de uma carteira geral o percentual da Coover referente aos seus serviços prestados, para conseguir ter caixa na empresa**



Para a User Story 6, o requisito de negócio implementado é referente à visualização dos membros do contrato de forma segura. Foi desenvolvida a função "visualizarCarteiras()" que retorna todos os membros daquele contrato. Para utilizar esse código, é necessário chamar a função e passar como argumento o endereço do contrato.

Para relembrar a **User Story 6: Como usuário, quero poder visualizar as pessoas dentro do meu contrato de forma segura, para poder ver a quantidade de usuários dentro do mesmo contrato que o meu.**

Para a User Story 8, o requisito de negócio implementado é referente à validade do contrato e ao número de membros necessários para o funcionamento do mesmo. Foi desenvolvida a função "viabilidadeContrato()" que confere se o contrato está na validade e possui o número definido de membros. Para utilizar esse código, é necessário chamar a função e passar como argumento o endereço do contrato.

Para relembrar a **User Story 8: Como gerente de seguros, quero que o smart contract só entre em pleno vigor, caso as condições de funcionamento sejam atendidas. Para que tudo funcione conforme o acordado.**

Para a User Story 10, o requisito de negócio implementado é referente à transferência de fundos para a carteira de quem deve ser assegurado após a aprovação de uma solicitação de indenização. Foi desenvolvida a função indenizar que transfere dinheiro do fundo do smart contract para a carteira que for indicada. Para utilizar esse código, é necessário chamar a função e passar como argumento o endereço da carteira que deve receber os fundos e o valor a ser transferido.

Para relembrar a **User Story 10: Como gerente de seguros, quero que após uma solicitação de indenização ser aprovada, quero poder transferir os dinheiros do fundo do smart contract para a carteira de quem deve ser assegurado. Para garantir que os usuários recebam o dinheiro.**

## 6.2. Módulo Frontend

A solução que estamos desenvolvendo em parceria com a seguradora Coover é um aplicativo mobile para os usuários e uma plataforma desktop para o gestor responsável da Coover. Para garantir maior acessibilidade e participação dos próprios integrantes do grupo de seguros, implementamos uma solução em blockchain baseada na cultura de seguro mútuo. O objetivo principal dessas telas é fornecer aos usuários e ao gestor uma experiência de usuário intuitiva e agradável. Para garantirmos isso, desenvolvemos técnicas de design de interface do usuário, como a criação de telas claras e concisas, com elementos gráficos visualmente atraentes e de fácil utilização.

### 6.2.1. Descrição do Frontend

Na construção do Front-end da aplicação, foram utilizadas as seguintes linguagens: HTML - linguagem de marcação, CSS - linguagem de estilo, JavaScript - linguagem de programação e React - biblioteca JavaScript front-end gratuita e de código aberto.

O React é usado para a criação de interfaces de usuário interativas e aplicativos da web de maneira rápida e eficiente com muito menos código do que faria com o JavaScript básico. Com o React, é possível criar componentes reutilizáveis que podem ser utilizados em diferentes partes da aplicação. Isso reduz o tempo de desenvolvimento e facilita a manutenção do código, o que torna mais fácil o processo de criação de interface e suas funcionalidades.

### 6.2.2. Tecnologias adotadas no Frontend

Na construção do Front-end da aplicação, foram utilizadas as seguintes linguagens: HTML - linguagem de marcação, CSS - linguagem de estilo, JavaScript - linguagem de programação e React - biblioteca JavaScript front-end gratuita e de código aberto.

O React é usado para a criação de interfaces de usuário interativas e aplicativos da web de maneira rápida e eficiente com muito menos código do que faria com o JavaScript básico. Com o React, é possível criar componentes reutilizáveis que podem ser utilizados

em diferentes partes da aplicação. Isso reduz o tempo de desenvolvimento e facilita a manutenção do código, o que torna mais fácil o processo de criação de interface e suas funcionalidades.

Com o objetivo de criarmos um ambiente de desenvolvimento de teste e das funcionalidades dos Smart Contracts, utilizamos uma carteira da Meta Mask para podermos então identificar e autenticar os usuários da aplicação e disponibilizar os serviços oferecidos para o usuário.

Além disso, o Truffle é uma ferramenta de desenvolvimento que utiliza a Ethereum Virtual Machine (EVM) para simplificar os processos de desenvolvimento em áreas de blockchain e dApps. Com o objetivo de fornecer diversas implementações, compilações, testes e gerenciamento, a ferramenta visa atender às diferentes necessidades dos desenvolvedores e criar um ambiente mais facilitador e eficiente. O nosso projeto então utiliza o Truffle com o intuito de criar interações de front-end com os smart contracts, compilar e implantar automaticamente os contratos inteligentes em diferentes redes, e desenvolver testes automatizados.

### 6.2.3. User Stories referentes ao Frontend

1	Como gerente de seguros, quero ter a permissão para aprovar as solicitações de retirada dos smart contracts do qual faço parte, para ter controle sobre a ativação do seguro.
2	Como gerente de seguro, quero poder visualizar todas as regras e condições do smart contract antes de ele entrar em vigor, para validar antes de os subir na rede.
3	Como usuário, quero uma forma de solicitar o seguro, para poder retirar o dinheiro do smart contract.

7	Como usuário, quero poder verificar se o smart contract está chegando no vencimento, para poder estar ciente das mudanças em seu funcionamento e/ou entrar em outro seguro mútuo.
9	Como usuário, quero saber a quantidade de pessoas dentro dos grupos para saber se ele está em progresso ou ativo, para ser segurado o mais rápido possível.
12	Como gerente quero que os clientes paguem pelo seguro um percentual correspondente ao valor do seu celular.

### 6.3. Prototipação

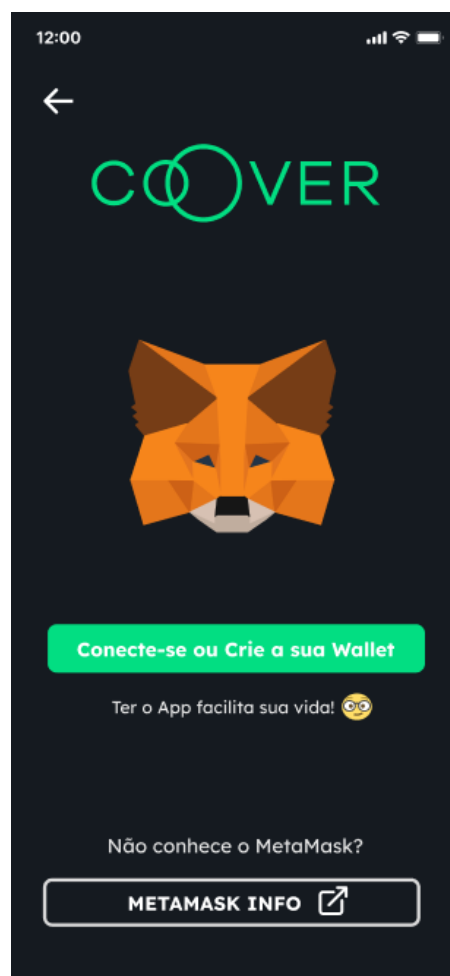
Abaixo segue a prototipação

Frame 1 - Tela Inicial



Tela inicial para o usuário entrar na plataforma.

Frame 2 - Tela de Entrada



Tela de entrada para o usuário se conectar com a sua carteira digital da Metamask na plataforma.

Frame 3 - Tela de Entrada 2



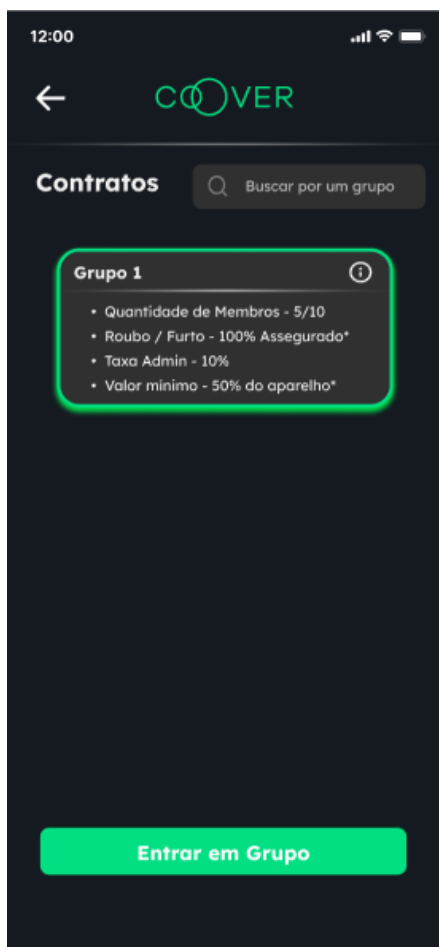
Essa tela é apresentada após o usuário ter cadastrado sua carteira na metamask.

Frame 4 - Tela de Menu Inicial



Essa tela é a de menu principal da plataforma.

Frame 5 - Tela de Contratos



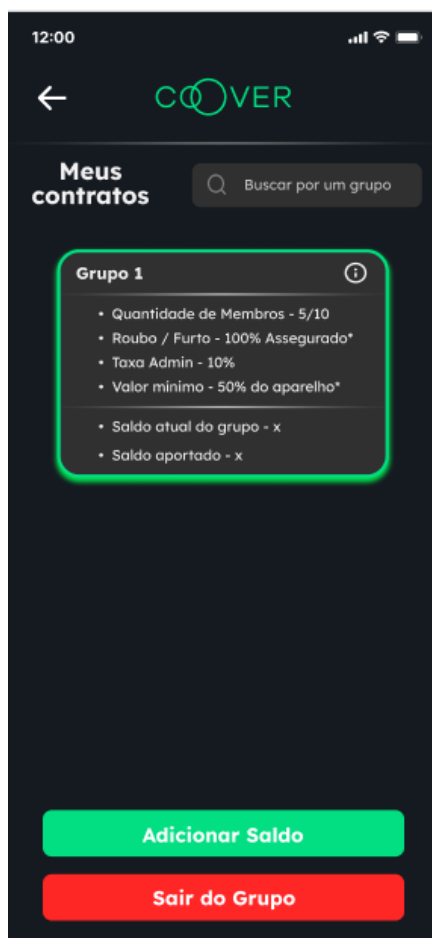
Essa tela mostra a opção de grupos disponíveis.

Frame 6 - Tela de Dados do Aparelho



Essa tela mostra um pedido para o usuário, informando que para se tornar membro de um grupo é necessário informar o valor do celular e o IMEI do mesmo.

Frame 7 - Tela de Meus Contratos



Depois de informado todos os dados o usuário pode pedir para entrar no grupo, selecionando-o.

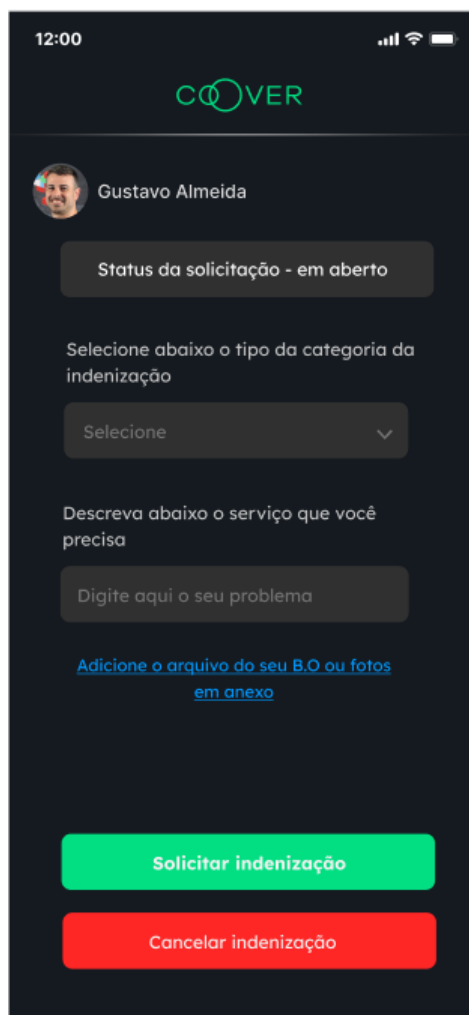
Frame 8 - Tela de Solicitação Enviada



E confirmando.




Frame 9 - Tela de Solicitar Indenização



12:00

COOVER

 Gustavo Almeida

Status da solicitação - em aberto

Selecione abaixo o tipo da categoria da indenização

Selecione ▾

Descreva abaixo o serviço que você precisa

Digite aqui o seu problema

[Adicione o arquivo do seu B.O ou fotos em anexo](#)

Solicitar indenização

Cancelar indenização

Frame 10 - Tela de Solicitação Enviada



12:00

COOVER



**Pedido de indenização enviado com sucesso!**

Sua solicitação já foi enviada! Agora só falta aguardar pela resposta da seguradora.

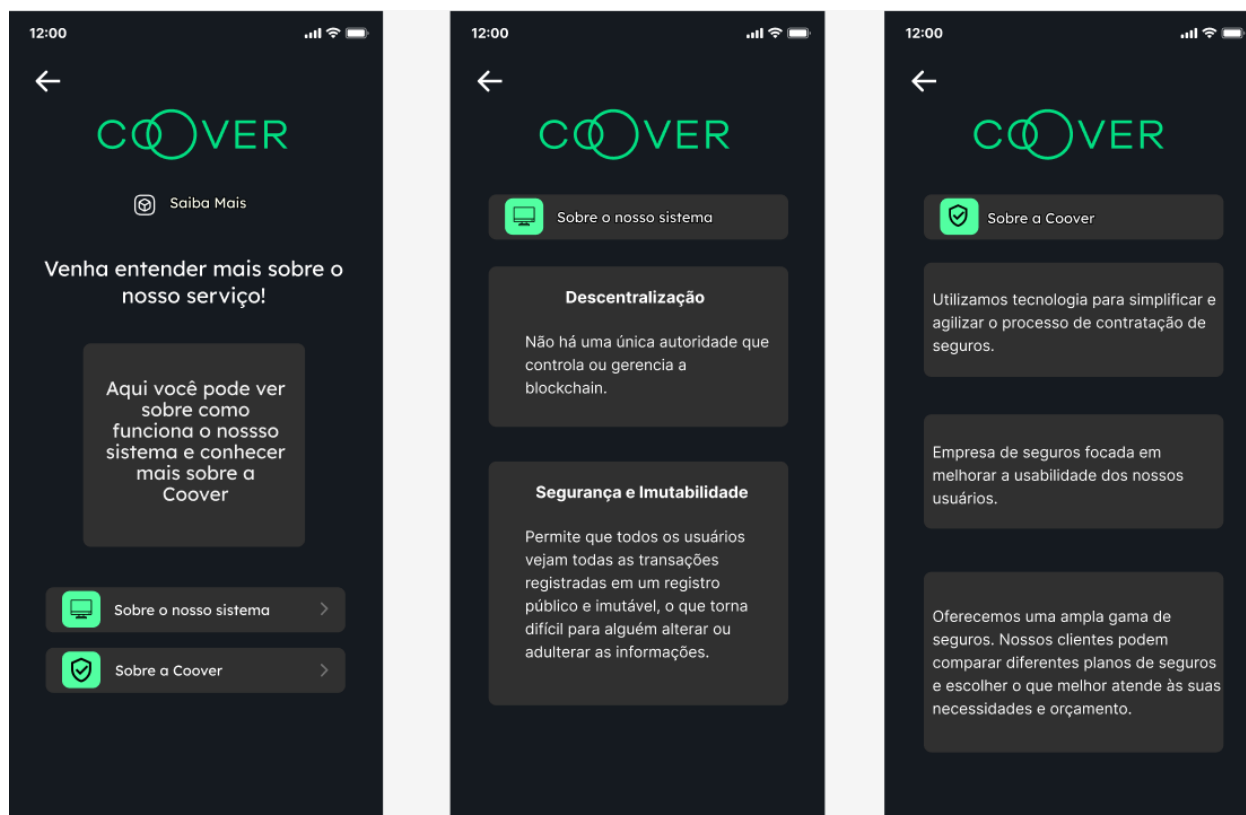
Status da solicitação

Em aberto

Abrir minha solicitação

Voltar para menu inicial

## Frame 11, 12 e 13 - Telas Saiba Mais



## 6.2. Avaliação

Para avaliar nosso MVP, criamos testes automatizados para verificar de algumas funções dentro do Smart Contract. Ao todo realizamos 5 testes, segue abaixo:

A documentação a seguir é uma descrição detalhada dos Testes de Aprendizado. Cada um dos testes apresentam suas pré-condições, resultados esperados, procedimentos de teste e pós-condições.

Ela pode ir incluir algumas informações como

- **Teste número 1;**
- **Teste número 2;**
- **Teste número 3;**
- **Teste número 4;**
- **Teste número 5;**

Nesse momento, realizamos dois testes seguindo metodologias utilizadas em testes de software. Esses testes são realizados para verificação das pequenas partes do código, chamadas de unidades, que devem apresentar o funcionamento perfeitamente, seguindo uma pré-condição, procedimento e pós-condição. Os testes foram executados em um ambiente isolado do contrato principal, em um arquivo denominado "coin\_test.sol". Essa metodologia ajuda a garantir que cada parte do código esteja funcionando corretamente antes de ser integrada a outras partes.

Instruções de teste unitário no ambiente de desenvolvimento:

### **Teste número 1**

Na primeira função de teste, "testarTaxaAdmin()", é verificado se a taxa de administração do contrato foi definida corretamente.

**Pré-condição:** é criado um novo contrato, passando a taxa de administração como parâmetro. Em seguida, o valor esperado é definido como 2. Como mostrado abaixo:

**Procedimento de teste:** o teste verifica se o valor retornado pela função "taxaAdmin()" é igual ao valor definido no deploy da nova instância do contrato.

**Resultado esperado:** o resultado esperado é que a função "taxaAdmin()" retorne 2, valor esse que foi definido na criação do contrato.

**Pós-condição:** o teste verifica se o valor retornado pela função "taxaAdmin()" é igual ao valor esperado, e se não for, a mensagem de erro "taxaIncorreta" é exibida.

## **Teste número 2**

Na segunda função de teste, "testarValorPosTaxa()", é verificado se o valor final, com a taxa de administração incluída, está sendo calculado corretamente.

**Pré-condição:** é criado um novo contrato, passando a taxa de administração como parâmetro, 10% no caso. Em seguida, o valor do depósito é definido como 10.

**Resultado Esperado:** o valor esperado é definido como 11 (o depósito original mais a taxa de administração).

**Procedimento de teste:** o teste calcula o impacto da taxa administrativa no valor do depósito através da função valorPosTaxa(), para depois ser comparado ao valor esperado.

**Pós-condição:** o teste verifica se o valor retornado pela função "valorPosTaxa()" é igual ao valor esperado, e se não for, a mensagem de erro "valor Incorreto" é exibida.

## **Teste número 3**

Na Terceira função de teste, "testSolicitacaoAprovacao()", é verificado se o usuário enviar uma solicitação de entrada no grupo, as suas informações são salvas corretamente no contrato

**Pré-condição:** é criado um novo contrato, e posteriormente, definimos o valor do aparelho "10" e imei "123" de usuário a ser adicionado no contrato.

**Resultado Esperado:** os valores esperados são os mesmos fornecidos anteriormente no caso valor do aparelho "10" e imei "123".

**Procedimento de teste:** O valor do aparelho e o imei definidos são fornecidos para função solicitacaoAprovacao() para o usuário ser adicionado ao contrato, depois as informações salvas são recuperadas pelas funções teste3A() e teste 3B(), para serem comparadas ao resultado esperado.

**Pós-condição:** o teste verifica se o valor retornado pelas funções teste3A() e teste 3B() são iguais aos resultados registrados previamente, e retorna “Erro ao registrar Imei” caso o imei esteja errado e “Erro ao registrar valor do aparelho” caso o valor do aparelho seja registrado errado.

#### **Teste número 4**

Na Quarta função de teste, "testeMaximoIndenizavel()", é verificado se o máximo indenizável, baseado no saldo restante do membro do contrato, está sendo calculado corretamente”.

**Pré-condição:** é criado um novo contrato, após isso é definimos o valor do aparelho “10” e o saldo do usuário “4”.

**Resultado Esperado:** o valor esperado é 8. equivalente a 80% de cobertura do valor do aparelho.

**Procedimento de teste:** O valor para o aparelho e o saldo do usuário são passados para função MaximoIndenizavel(), que irá calcular o máximo indenizável com base nas informações e retornar esse valor para ser comparado posteriormente com o esperado.

**Pós-condição:** o teste verifica se o valor retornado pela funçãoMaximoIndenizavel() é igual ao valor espererado”8”, e retorna “Maximo indenizavel incorreto” caso o cálculo de errado.

#### **Teste número 5**

Na Quinta função de teste, "testeViabilidadeContrato()", é verificado se a função identifica corretamente se o contrato possui a quantidade correta de membros para o seu funcionamento”.

**Pré-condição:** é criado um novo contrato, passando como parâmetro o mínimo de pessoas como 1 e o máximo como 3, e são adicionados 2 usuários no contrato através da função adicionarUsuario.

**Resultado Esperado:** o valor esperado é 1, que ocorre quando o contrato está avaliado, uma vez que ele possui 2 usuários, valor este que está entre o mínimo e o máximo.

**Procedimento de teste:** A função adicionarUsuario() é chamada 2 vezes, para adicionar 2 novos usuários ao contrato, após isso é chamada a função viabiliadeContrato(), que irá verificar

se o número de pessoas é válido e retornar um feedback, que será utilizado para ser comparado com resultado esperado.

**Pós-condição:** o teste verifica se o valor retornado pela `viabilidadeContrato()` é igual ao Resultado esperado “1”, e retorna “Resposta Incorreta” caso o resultado esteja errado.

Em resumo, os testes de automatizados são importantes para garantir que pequenas partes do código estejam funcionando corretamente antes de serem integradas a outras partes. Os testes têm pré e pós-condições definidas e verificam se o resultado esperado foi alcançado. Isso ajuda a garantir que o código funcione corretamente e minimize a possibilidade de erros e bugs em versões futuras.

## 7. Conclusões e Recomendações

O projeto de smart contract com blockchain para a seguradora de Coover apresentou resultados positivos, principalmente no que diz respeito à eficiência e transparência na gestão dos contratos de seguros. O uso da tecnologia blockchain permitiu que os contratos fossem automatizados e executados de forma autônoma, sem a necessidade de intermediários, reduzindo os custos e aumentando a confiabilidade.

Além disso, a tecnologia blockchain permitiu a criação de um registro imutável e seguro de todas as transações realizadas, garantindo a integridade dos dados e evitando fraudes. O projeto também possibilitou a criação de uma plataforma de gerenciamento de seguros online, acessível e amigável para os clientes. Assim como também uma plataforma para os administradores da Coover para gerenciar os contratos.

Com base nesses resultados, recomenda-se que a Coover continue investindo em tecnologias disruptivas como blockchain e smart contracts, a fim de aprimorar ainda mais a eficiência de seus serviços e aumentar a satisfação dos clientes. É importante também que a empresa mantenha-se atualizada em relação às novidades e tendências do mercado, buscando sempre inovações que possam agregar valor ao negócio.