

**Controle do Documento**

**Histórico de revisões**

| **Data** | **Autor** | **Versão** | **Resumo da atividade** |
| --- | --- | --- | --- |
| 08/02/2023 | Sophia Mello Dias | 1.1 | Criação do documento,Introdução, Definição do Problema, Objetivos e Justificativa, Matriz de Risco e Canvas Value Proposition |
| 08/02/2023 | Vitor Moura  Lucas Vieira | 1.2 | Criação das User Stories |
| 09/02/2023 | Raduan Muarrek | 1.3 | Elaboração do contexto da indústria, SWOT, 5 forças de Porter |
| 10/02/2023 | Vitor Moura | 1.4 | Elaboração dos cards de User Stories |
| 10/02/2023 | Sophia Mello Dias | 1.5 | Revisão e correção ortográfica: Contexto da Indústria, 5 forças de Porter, SWOT Analysis e Justificativa (2.2) |
| 25/02/2023 | Vitor Moura | 1.6 | Refinamento das User Stories |

**Sumário**

[1. Introdução 4](#_heading=h.1t3h5sf)

[2. Objetivos e Justificativa 1](#_heading=h.2p2csry)

[2.1. Objetivos 1](#_heading=h.147n2zr)

[2.2. Justificativa 1](#_heading=h.3o7alnk)

[3. Análise do Negócio 2](#_heading=h.23ckvvd)

[3.1. Contexto da indústria 2](#_heading=h.ihv636)

[3.2. Ferramentas 2](#_heading=h.32hioqz)

[4. Análise de Experiência do Usuário 3](#_heading=h.1hmsyys)

[4.1. Personas 3](#_heading=h.41mghml)

[4.2. Jornadas do Usuário e/ou Storyboard 3](#_heading=h.2grqrue)

[4.3. User Story 4](#_heading=h.vx1227)

[4.4. Protótipo de interface com o usuário 5](#_heading=h.3fwokq0)

[5. Solução Proposta 6](#_heading=h.1v1yuxt)

[5.1. Solução 6](#_heading=h.4f1mdlm)

[5.2. Arquitetura Proposta 6](#_heading=h.2u6wntf)

[5.3. Diagrama Macro da Solução 6](#_heading=h.19c6y18)

[5.4. Descrição da Solução 6](#_heading=h.3tbugp1)

[5.5. Algoritmos e Equações 6](#_heading=h.28h4qwu)

[6. Desenvolvimento e Resultados 7](#_heading=h.nmf14n)

[6.1. Nome do Módulo 7](#_heading=h.37m2jsg)

[6.1.1. Descrição 7](#_heading=h.1mrcu09)

[6.1.2. Tecnologia adotada 7](#_heading=h.46r0co2)

[6.1.3. User Stories 7](#_heading=h.2lwamvv)

[6.1.4. Prototipação 7](#_heading=h.111kx3o)

[6.1.5. Diagramas 7](#_heading=h.3l18frh)

[6.2. Avaliação 7](#_heading=h.2bn6wsx)

[7. Conclusões e Recomendações 8](#_heading=h.206ipza)

[8. Referências 9](#_heading=h.4k668n3)

[Anexos 10](#_heading=h.2zbgiuw)

[ANEXO I – Sprint 1 10](#_heading=h.1egqt2p)

[ANEXO II – Sprint 2 10](#_heading=h.3ygebqi)

[ANEXO III – Sprint 3 10](#_heading=h.2dlolyb)

[ANEXO IV – Sprint 4 10](#_heading=h.sqyw64)

# 1. Introdução

**1.1. Parceiro de Negócios**

A Coover é uma startup de pequeno porte da área de seguros (insutech), que busca democratizar o acesso a seguros à população brasileira, visto que 70% desta não possui bens cobertos por um seguro privado. Atualmente cobre apenas os de furto e roubo de celular e, em breve, plano de saúde para pet. Ela está inserida em um mercado super competitivo com concorrentes como Porto e Bradesco, por outro lado participa da revolução do mercado de seguros, pois seus processos são todos realizados por plataforma, desde a contratação, até mesmo a vistoria do aparelho celular e indenização do seguro e tudo com muito pouca burocracia.

Apesar de ter tido sucesso com o MVP de seguros peer-to-peer e uma boa aceitação do mercado, como mencionado pelo CTO Fabrício Matos, o órgão regulador responsável pelos seguros, Susep, impediu a continuação de suas operações, o que levou à parceria com a Zurich Seguros e, consequentemente, a plena realização de seguros mútuos que democratizam seu acesso aos brasileiros principal objetivo do desenvolvimento do projeto, é permitir que qualquer pessoa contrate de forma fácil um seguro de celular pelo método de seguro mútuo (peer-to-peer). Além disso, visando o cliente do negócio, fornecer um produto seguro, que resguarde quaisquer dados sensíveis, forneça transparência de transações e simplicidade na indenização de um sinistro. Ademais, seja mais prático em relação aos métodos utilizados atualmente.

Por outro lado, da Coover, são diminuir os custos de operação, visto que a tecnologia blockchain não requer um servidor próprio e os custos de transação são incluídos na parte do cliente, além de desburocratizar a criação de apólices, contratos e indenizações. Por fim, fornecer segurança e transparência acerca de todas as suas decisões para os clientes.

**1.2. Definição do Problema**

**1.2.1. Problema**

O Stakeholder Coover identificou um grande gap da adesão e contratação de seguros tradicionais na população brasileira, em que apenas 30% possuem algum tipo de proteção privada. Tal problema acentua a falta de segurança e estabilidade econômica da população, que tem produtos de baixa capilaridade para atender as necessidades populacionais e com alto custo comparado ao valor gerado.

Dessa maneira, os seguros mútuos aparecem como uma solução disruptiva ao mercado tradicional, permitindo que grupos de pessoas criem reservas de dinheiro inacessíveis, e que garantam o pagamento de indenizações aos segurados. Portanto, seguindo a transformação digital do mundo contemporâneo e, ao mesmo tempo, o surgimento de novas tecnologias mais seguras e confiáveis, os smart contracts da tecnologia blockchain inovam o mercado brasileiro como uma solução à população sem bens resguardados por um seguro.

# 2. Objetivos e Justificativa

## 2.1. Objetivos

O principal objetivo do desenvolvimento do projeto, é permitir que qualquer pessoa contrate de forma fácil um seguro de celular pelo método de seguro mútuo (peer-to-peer). Além disso, visando o cliente do negócio, fornecer um produto seguro, que resguarde quaisquer dados sensíveis, forneça transparência de transações e simplicidade na indenização de um sinistro. Ademais, seja mais prático em relação aos métodos utilizados atualmente.

Tais objetivos são cobertos com o uso da tecnologia blockchain, englobando a ativação de smart contracts, que garantem a criação de grupos mútuos de seguro, o pagamento correto de indenizações e a transparência de quaisquer transações, além de uma aplicação Web 3.0 que, com o uso de wallets, permite uma maior proteção dos dados sensíveis dos usuários.

Por outro lado, da Coover, são diminuir os custos de operação, visto que a tecnologia blockchain não requer um servidor próprio e os custos de transação são incluídos na parte do cliente, além de desburocratizar a criação de apólices, contratos e indenizações. Por fim, fornecer segurança e transparência acerca de todas as suas decisões para os clientes.

## 2.2. Justificativa

Para atingir os objetivos descritos pelo Stakeholder Coover, tanto pela visão do cliente, quanto da empresa, a tecnologia blockchain, que será utilizada como cerne do desenvolvimento da solução, apresenta alto potencial de crescimento, visto que é previsto um crescimento de 1.337% entre 2022 e 2030, segundo o relatório divulgado pela Custom Market Insights, projetando uma capitalização total da indústria de US$69 bilhões.

Demonstra-se, a partir destes dados, uma grande oportunidade de entrar no mercado de seguros tradicional brasileiro, com uma tecnologia cada dia mais adotada por jovens e de crescimento exponencial. A solução engloba a substituição de apólices e contratos que envolvem advogados e demais partes interessadas para smart contracts, que além de estabelecerem as cláusulas do contrato, também realiza as funções de recebimento do pagamento mensal dos segurados e transações de indenizações, que não podem ser mudadas por uma característica intrínseca do funcionamento da blockchain. Ademais, com o avanço das normas LGPD, que garantem mais segurança e controle acerca de seus dados sensíveis, a aplicação web onde a contratação do serviço e o processo de indenização ocorrem não contém nenhum destes dados, somente uma chave de carteira, o que garante o cumprimento de tais obrigações e maior segurança ao contratante.

# 3. Análise do Negócio

## 3.1. Contexto da indústria

O mercado de seguros é extremamente ligado ao comportamento e mudanças da sociedade, sendo volátil para garantir a continuidade de segurança da população. Pode-se dizer que é volátil, pois seguradoras que já apresentaram papel protagonista no Brasil a três décadas atrás, a Bamerindus, não existe mais, assim como a integração de duas grandes seguradoras: Marítima Seguros e Yasuda Seguros, que resultou na Sompo Seguros. Também é volátil no sentido de adaptação de seus produtos, como a criação de seguro celular após a alta adoção desses pela comunidade, visto que, segundo o IBGE, são 242 milhões de celulares ativos para 214 milhões de brasileiros.

Com esse cenário macro dos seguros e o posicionamento da Coover, que vem como a disrupção dos seguros tradicionais brasileiros, introduzindo tecnologia e novos modelos de negócio aos seguros de celular, a Porto Seguro Seguradora: A Porto é, talvez, a maior referência quando o assunto é seguros no Brasil. Além de atuarem há mais de 70 anos no mercado e líder em inovação,ilustrada pela Oxigênio, aceleradora que, por mais que seja equity free, procura aproximar a inovação com a Porto. Assim, é uma é uma empresa que opera em diversas frentes, como seguro celular, carro, saúde, entre outros, o que causa, muitas vezes, vendas casadas aos seus clientes. Todos esses pontos fazem com que a Porto seja uma gigante no setor que deve ser observada constantemente no mercado.

Por outro lado, a Pier Seguradora é uma empresa que se assemelha bastante à Coover. Ela é uma startup criada por três amigos, que começaram sem nenhum capital e com anseio de melhorar a relação entre seguro e cliente. Segundo eles: “Ninguém gosta de seguradora”. A Pier é uma empresa que atende, hoje, celular e carros, ambos com seguros próprios. Sua comunicação é irônica com os seguros tradicionais e seu site tenta ser o mais simples e direto ao ponto possível. Pelo fato de ser uma empresa parecida com a Coover, que está tendo um bom crescimento e por ter boas práticas, a Pier é a segunda empresa que destacamos aqui.

Dado os concorrentes na indústria de seguros, é necessário também entender como esse setor funciona. Além de servir como um contexto para a solução a ser desenvolvida, serve também para eventuais features a serem adicionadas ou para guiar a maneira como conduzir a comunicação e a user experience da solução de acordo com características da indústria. Por isso utilizamos as 5 forças de Porter e SWOT Analysis para complementar a análise do mercado:

## 3.1.1 5 forças de Porter



Imagem 1 - 5 forças de Porter

1. Ameaça de novos entrantes.

Com o cenário macroeconômico e das atuais ventures capitals e startups, temos uma ameaça de novos entrantes, de maneira geral, a curto prazo .

Pensando em médio e curto prazo, têm-se fatores que também deixam a ameaça de novos entrantes baixa, como a grande barreira de entrada que existe no setor criado pelo capital de investimento necessário para abrir uma seguradora.

Entretanto, o mercado de seguros mostrou ser um pouco volátil nos últimos anos, pois, por mais que temos a Bradesco Seguros e a Porto Seguros há anos de mercado, como foi citado anteriormente, mas existem também empresas como a Tokio Marine que acabaram passando por instabilidades em suas operações, podendo assim criar oportunidades de mercado para novos entrantes. Ou seja, por mais que classifiquemos a ameaça de novos entrantes como baixa, enxergamos também o risco iminente de surgir oportunidades de mercado, movidas por instabilidades de grandes empresas no setor.

1. Ameaça de produtos substitutos.

A solução de seguros é muito bem aceita e requisitada pelo mercado, não à toa seus lucros estão em constante crescimento, inclusive em ano de pandemia. Além disso, ela possui muitas regulações, criando dificuldades para novas frentes de seguro, sendo assim difícil de se imaginar novas soluções que substituam a atual sendo utilizada.

Entretanto, a própria solução proposta pela Coover é uma espécie de produto substituto, em que se conclui que, no futuro próximo, pode-se ter novas tecnologias e tendências que não mudem completamente a dinâmica atual, mas atualizem a maneira como o seguro é concedido atualmente. Sendo assim, atribuímos uma ameaça de nível médio para novos produtos substitutos.

1. Poder de barganha com fornecedores.

Esse é um dos aspectos mais difíceis de analisar quando se trata de seguradoras, pois é difícil analisar primeiro o que é fornecedor e, segundo, como eles se comportam. Para essa análise partiremos de algumas premissas que podem ser questionadas e não vamos considerar o “backstage” do financiamento de seguros, como por exemplo: Como de fato é a relação dos bancos com os seguros.

Como citado anteriormente, a dinâmica de um seguro é um pouco diferente que outras operações, pois, de certo modo, os clientes também são os próprios fornecedores do principal insumo. Eles são os próprios fornecedores, porque com o dinheiro que ele utiliza para contratar o seguro é o que permite a seguradora pagar o sinistro para demais clientes. Mas se considerarmos esse caixa disponível para criar os fundos para seguros como os fornecedores, temos um cenário neutro para as empresas.

Da mesma forma, temos 70% da população sem nenhum tipo de seguro, criando assim oportunidades de formação de grandes círculos de seguro, têm-se também poucas instituições financeiras no Brasil de grande porte para auxiliar na construção de produtos para seguradoras. Considerando que o papel dos bancos seja mais influente que o papel dos clientes, temos uma força neutra para o poder de barganha com consumidores.

1. Poder de barganha com clientes.

Diferente do varejo, por exemplo, uma seguradora não tem tanto espaço para diminuir sua margem, para conquistar novos clientes, pois o preço do seguro está diretamente atrelado à relação do valor do celular com o seu risco. Sendo assim, não existe muito espaço para criar grandes campanhas e alcançar mais clientes.

Por outro lado, o público alcançável é muito grande, 70% da população brasileira não possui nenhum seguro, mas ainda pode-se considerar os outros 30% que podem ter apenas um tipo de seguro e querem passar a ter outros. Dado a um Serviceable Addressable Market massivo, qualquer ressalva negativa que a operação pode trazer fica quase que imperceptível. A quantidade de demanda cria uma boa posição para as seguradoras que não precisam brigar tanto assim pelos clientes e desse modo, cria-se um poder de barganha alto para as seguradoras brasileiras.

1. Rivalidade entre concorrentes

Para concluirmos nossa linha de raciocínio de nossa análise, a rivalidade entre concorrentes, de modo geral, não é alta. Temos poucos grandes players que detém a maior parte do market share há um bom tempo, somado a isso, o total adresseble market é bem alto, esse cenário favorável cria-se quase que uma zona de conforto aos players atuais. O histórico de instabilidade do setor e o atual caminhar da tecnologia como o blockchain, entretanto, é algo que acaba fazendo com que essas grandes empresas arrisquem mais, criando assim uma maior competitividade entre eles.

## 3.1.2 Análise SWOT



Imagem 2 - Análise SWOT

Por fim, para finalizar o entendimento do contexto da indústria, foi feita uma análise SWOT, que visa compreender mais da Coover, com suas forças e fraquezas internas e externas. Sendo assim, os seguintes pontos foram levantados:

1. Forças (interno):

* DNA em tecnologia e inovação para criar valor real para o cliente e
* Desenvolvimento interno de soluções de acordo com tendências tecnológicas.

1. Fraquezas (interno):

* Ainda sem um diferencial competitivo para oferecer para os clientes;
* Porte de startup onde existem empresas muito grandes e tracionadas competindo;
* Dependência de empresas externas como a Zurich.

1. Oportunidades (externo):

* Problema real que tem muito espaço para explorar. 70% dos brasileiros não tem um seguro;
* Blockchain em constante crescimento. Adoção segue no mesmo ritmo da internet nos anos 2000 e
* Aprovações de leis no Brasil acerca da Blockchain.

1. Ameaças (externo):

* Setor com histórico incerto. Grandes empresas já faliram;
* Dependência da SUSEP e regulações governamentais e
* Oceano vermelho. Muitos concorrentes grandes como Porto, Bradesco, Mercado Pago etc.

Conclui-se que, apesar de ameaças fortes, que envolvem tanto concorrentes quanto legislações, tem muito espaço para a Coover conquistar clientes e desenvolver soluções inovadoras e tecnológicas, pois as fraquezas apontadas são factíveis de serem solucionadas e, assim, ganhar uma parcela importante de um grande mercado brasileiro.

## 3.2. Ferramentas

## 3.2.1. Matriz Oceano Azul

Após o entendimento de algumas concorrentes e características do setor, foram analisadas a relação entre elas no próprio setor. É uma reflexão dos pontos que, dentro dessa dinâmica, a Coover pode Reduzir, Eliminar, Criar e Aumentar.

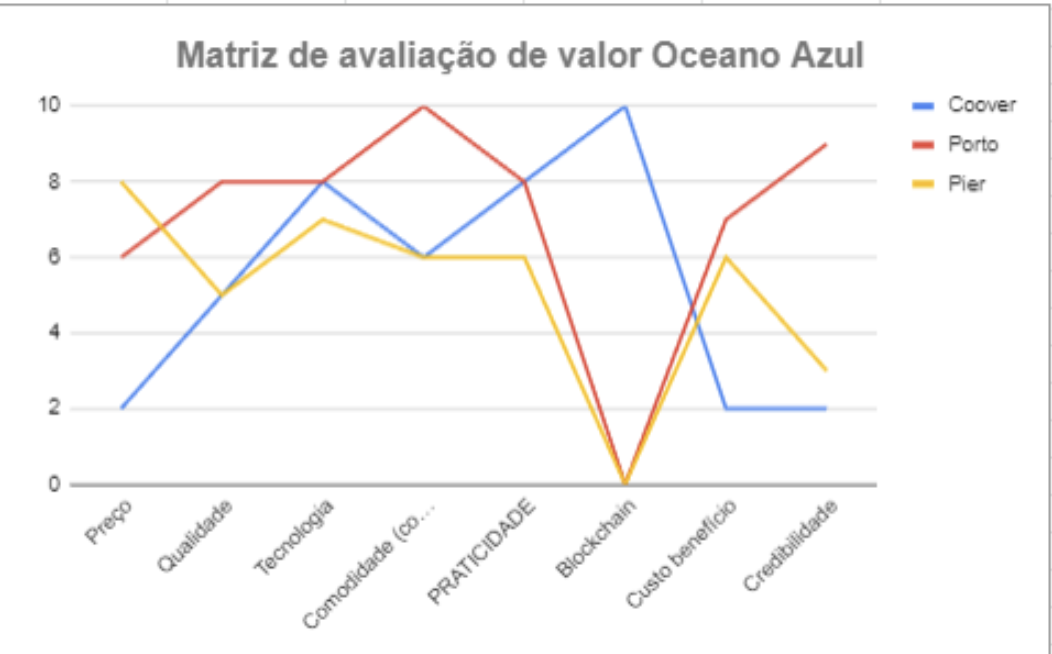


Imagem 3 - Matriz Oceano Azul

| Atributos | Coover | Porto Seguro | Pier |
| --- | --- | --- | --- |
| Preço | 2 | 6 | 8 |
| Qualidade | 5 | 5 | 8 |
| Tecnologia | 8 | 8 | 7 |
| Comodidade | 6 | 10 | 6 |
| Praticidade | 8 | 8 | 6 |
| Blockchain | 10 | 0 | 0 |
| Custo benefício | 2 | 7 | 6 |
| Credibilidade | 2 | 9 | 3 |

Tabela - Matriz Oceano Azul

A partir da pesquisa do grupo acerca da principal seguradora do mercado no seguro de celular, Porto Seguro, e a maior concorrente atual da Coover, e a startup Pier, foi possível concluir os seguintes pontos com a finalidade de tornar a Coover mais competitiva no mercado: Deve REDUZIR seu preço, visto que a qualidade dos seus concorrentes encontra-se bem mais alta; Deve CRIAR credibilidade, pois sem uma clientela fiel e recorrente, sua parcela de market share ficará comprometida; Deve AUMENTAR seu custo benefício, intrinsecamente ligado à sua proposta de valor, o custo deve condizer com a mudança de comportamento do cliente de uma seguradora tradicional à uma tecnológica que utiliza blockchain e, por fim, não foi identificado nenhum atributo a ser ELIMINADO que permita a Coover ser mais competitiva com suas concorrentes e não afete o valor percebido pelo cliente.

## 3.2.2. Matriz de Risco

Para compreender melhor as possíveis oportunidades e ameaças que englobam a solução proposta: smart contracts de seguros mútuos com aplicação Web 3.0 na Blockchain, foi utilizada a Matriz de Risco pelos parâmetros de probabilidade do evento ocorrer e o impacto para a Coover.

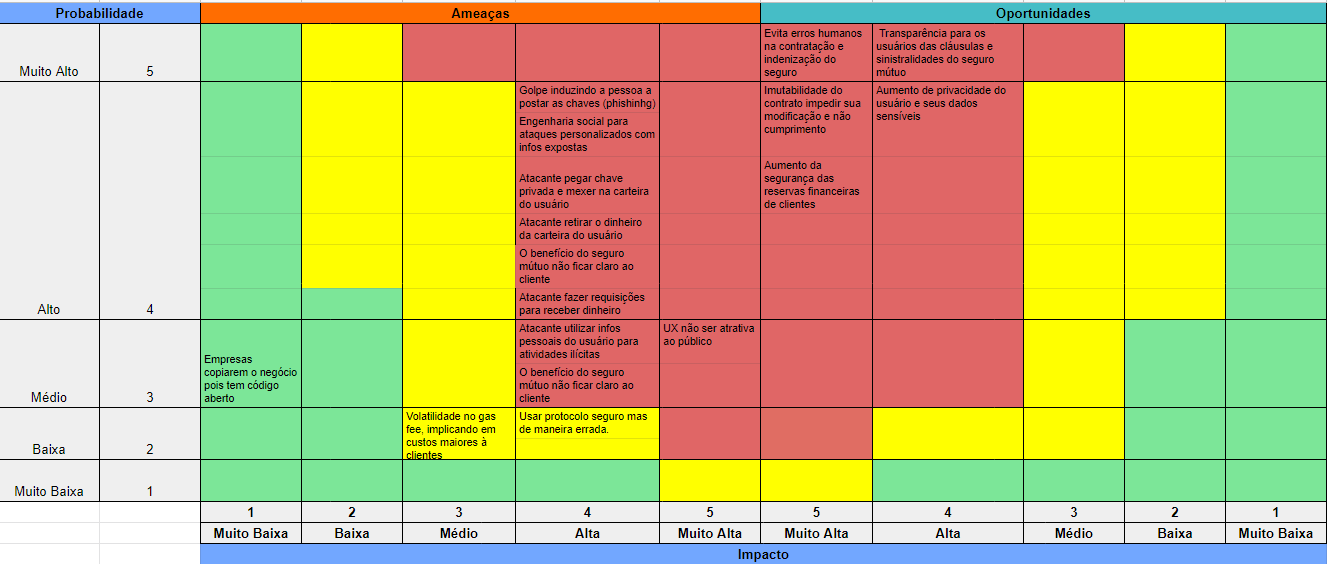


Imagem 4 - Matriz de risco

1. Empresas copiarem o negócio pois tem código aberto: como na tecnologia blockchain, todos os códigos (smart contracts) ficam na rede aberta, qualquer pessoa pode facilmente ver qualquer informação, assim um player que tem maior fatia do mercado consegue copiar o modelo de negócio. Entretanto, o valor gerado ao cliente não se restringe somente ao produto, mas toda a cadeia que a Coover gera, como comentado pela própria no kick off, garantindo um risco menor de prejudicar seu crescimento.
2. Volatilidade no gas fee, implicando em custos maiores à clientes: o valor do gas fee representa algo muito insignificante do total valor do seguro, sem contar que não é uma transação recorrente, diária. Visto isso, é algo que poderia ter um impacto maior a quem não é familiarizado com o termo, mas levando em consideração o que foi dito na reunião com Fabrício, todos os usuários estão acostumados com a tecnologia, representando uma probabilidade média com impacto baixo.
3. UX não ser atrativa ao público: a experiência do usuário é um fator de muita discussão da comunidade blockchain na Web 3.0, pois a falta de entendimento acerca do uso das carteiras impossibilita um usuário usar o serviço de maneira plena, assim, por mais que tenha uma probabilidade média, o impacto para a Coover é muito alto pela adesão de usuários ao sistema.
4. O benefício do seguro mútuo não ficar claro ao cliente: o principal benefício e valor ao cliente ao utilizar a solução blockchain da Coover baseia-se no conceito de seguros mútuos. Portanto, caso um cliente não entenda esse conceito, podem surgir diversos conflitos empresa-cliente, a partir disso, representando um impacto alto para a Coover, com uma probabilidade média, levando em consideração uma boa comunicação entre Coover-usuário.
5. Erros no deploy do smart contract, aumentando seu processo de ativação: erros na programação são comuns, principalmente no deploy, além de que as requisições de smart contracts para ingressarem na primeira camada da EVM (Ethereum Virtual Machine) são altas e demandam certo tempo até serem concluídas, o processo de ativação oficial do smart contract não é instantâneo, mas não é algo que o cliente consegue visualizar. Portanto, representam uma probabilidade alta, de resolução factível à Coover, mas de impacto baixo por não ter tempo de abalar os clientes.
6. Aumento da segurança das reservas financeiras de clientes: a tecnologia blockchain, por ser totalmente criptografada, fornece uma alta segurança aos seus ativos financeiros,pela wallet possuir chaves públicas e uma privada, assim tem uma probabilidade alta de ocorrência e impacto positivo muito alto ao negócio da Coover.
7. Aumento de privacidade do usuário e seus dados sensíveis: como dito no item anterior, a blockchain possui uma segurança mais alta que quaisquer tecnologias, garantindo privacidade dos dados pessoais dos usuários, por isso possui probabilidade e impacto altos.
8. Imutabilidade do contrato impedir sua modificação e não cumprimento: pelo conceito de blockchain, em que cada bloco é conectado ao anterior e ao seguinte, assim quando houver uma modificação em um bloco, todos os anteriores à ele deve ser mudados também, garante imutabilidade do contrato (código) e que ele funcione de forma confiável, evitando conflitos de interesse entre cliente e Coover por isso suas probabilidade e impacto são alto e muito alto, respectivamente.
9. Transparência para os usuários das cláusulas e sinistralidades do seguro mútuo: pela tecnologia blockchain ser open source, ou seja, aberta à toda rede mundial para acessar os smart contracts publicados, evita quaisquer inconsistências, além de ser completamente auditável, dessa forma tem uma probabilidade muito alta, pois é 100% open source, garantindo transparência das cláusulas contratuais e seu ledger, e impacto alto à Coover, por ser auditável.
10. Evita erros humanos na contratação e indenização do seguro: como os smart contracts gerem todos os processos de um seguro, desde a apólice a sua indenização, é garantido que ela ocorrerá de forma correta e para as carteiras fornecidas na sua ativação, evitando quaisquer erros humanos no meio do processo, assim tem uma probabilidade muito alta, pois refere-se ao funcionamento de um smart contract, e um impacto positivo muito alto à Coover.

## 3.2.3. Canvas Value Proposition

Depois de entendermos o risco do projeto, passamos então para entender o projeto em si. Ou seja, o que ele de fato agrega ao cliente. Aliviando uma dor ou criando ganho para nossa persona (que está descrita posteriormente no item 4.1).

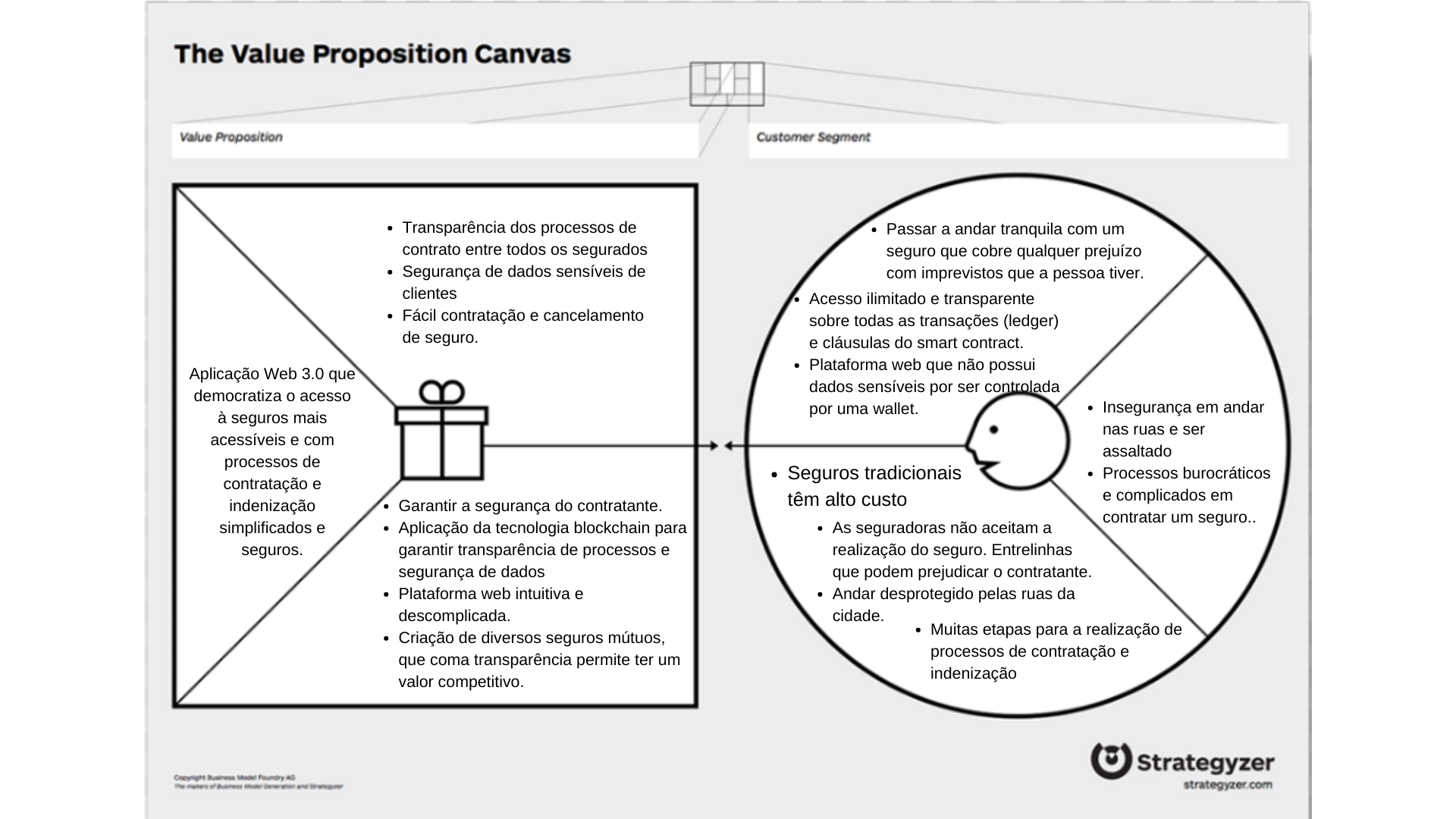


Imagem 5 - Canvas Value Proposition

Para a confecção do Canvas Value Proposition foram utilizadas as Personas (que podem ser vistas na seção abaixo), pensando nas suas necessidades core e dores. Dessa forma, tais pontos foram levantados:

Value proposition:

1. Produtos e serviços: Aplicação Web 3.0 que democratiza o acesso à seguros mais acessíveis e com processos de contratação e indenização simplificados e seguros.
2. Criadores de ganho:

* Segurança de dados sensíveis de clientes;
* Transparência dos processos de contrato entre todos os segurados;
* Fácil contratação e cancelamento de seguro.

1. Aliviadores de dor:

* Criação de diversos seguros mútuos, que com a transparência permite ter um valor competitivo;
* Garantir a segurança do contratante;
* Plataforma web intuitiva e descomplicada;
* Aplicação da tecnologia blockchain para garantir transparência de processos e segurança de dados.

Perfil do consumidor:

1. Atividades do cliente:

* Insegurança em andar nas ruas e ser assaltado;
* Processos burocráticos e complicados em contratar um seguro.

1. Ganhos:

* Acesso ilimitado e transparente sobre todas as transações (ledger) e cláusulas do smart contract;
* Plataforma web que não possui dados sensíveis por ser controlada por uma wallet;
* Passar a andar tranquilamente com um seguro que cobre qualquer prejuízo com imprevistos que a pessoa tiver.

1. Dores:

* As seguradoras não aceitam a realização do seguro. Entrelinhas que podem prejudicar o contratante;
* Seguros tradicionais têm alto custo;
* Andar desprotegido pelas ruas da cidade;
* Muitas etapas para a realização de processos de contratação e indenização.

## 3.2.4. Análise Financeira do Projeto

Após entendermos todo o mercado e o business da Coover. Criamos, para finalizar para finalizar a análise de negócio, uma projeção para entender a viabilidade econômica do projeto.

Fazendo uma especulação de quanto seria gasto neste projeto no período de um ano, podemos destacar os seguintes tópicos:

* Profissionais com entendimento em blockchain e web3;
* Profissionais com ênfase em UX;
* Equipe de Marketing;
* Equipe de gestão de projetos;
* Profissionais para a parte judicial.

Para fazer uma hipótese de quanto que gastaria neste projeto, deveríamos saber quais são as prioridades do cliente, o esperado da solução pelo cliente e sua pretensão futura com o projeto. Considerando o projeto completo como de porte médio, podemos estipular uma média de custo de R$300.000. A criação de um smart contract é um processo que deve ser feito com muita cautela e necessita de profissionais bem qualificados, com revisões bem rigorosas dos códigos, o que torna o processo mais custoso. Além de toda equipe de UX e Marketing para deixar o usuário confortável com a ferramenta utilizando Blockchain.

Ps: Quando essa informação foi levantada com o cliente, não tivemos nenhuma informação concreta, portanto essa análise não passa de uma especulação.

# 4. Análise de Experiência do Usuário

## 4.1. Personas

### 4.1.1. Rebeca



Imagem 6 - Persona 1

Rebeca é estagiária de arquitetura que gosta de conhecer lugares. Comprou um novo celular por ter tido o antigo roubado na volta de um happy hour, por isso tem medo de usar o novo nas ruas de seu bairro pois tem receio de ser novamente assaltada. Para diminuir sua angústia, ela entrou em contato com uma seguradora tradicional para contratar um plano de reembolso para seu smartphone novo, porém foi recusada. Por esse motivo, ela deseja proteger sua recente compra, o celular, assim como seus dados pessoais na internet. Para isso ela quer contratar um seguro de maneira simplificada e poder resolver qualquer problema da mesma forma, como faz atualmente em sua vida em outras áreas, como o banco digital, pedir caronas etc.

### 4.1.2. Augusto



Imagem 7 - Persona 2

Augusto é curioso e gosta de estudar sobre tecnologia. Formado em sistemas de informação, ele descobriu o interesse por revoluções tecnológicas e começou a empreender na área. Criou a Coover com a intenção de ampliar a indústria de seguradoras do Brasil a partir do uso de blockchain. Ele é entusiasta por essa tecnologia como solução, pois pelos seus estudo crê que ela pode reduzir diversos custo e tempo de processos, além de garantir segurança aos seus usuários.

Frustrado com o mercado atual de seguros, por não entregarem valor suficiente aos clientes e, além de concorrentes terem preços mais altos, muitos brasileiros não conseguem proteger seu bem de maior valia: o celular.

## 4.2. User Story

User stories são descrições curtas e simples de funcionalidades que um usuário precisa para alcançar um objetivo específico em relação a um produto ou sistema. Elas nos ajudam fornecendo uma descrição clara e concisa dos requisitos que o sistema deve ter, mantendo o foco na solução das dores do usuário.

Seguindo este princípio, juntamente com as personas do projeto, montamos o seguinte fluxo de requisitos para estruturar as nossas user stories:

Criação do contrato (pré ativação) -> Usuário aceita o convite -> Monitoramento dos grupos e indivíduos -> Publicação do contrato -> Aporte do cliente -> Pedido de indenização -> Monitoramento dos pedidos -> Autorização ou recusa das indenizações.

### 4.2.1. Versão 1.0

| **User Story - Augusto** |
| --- |
| 1. Eu, como agente da seguradora, quero ter uma interface que permita a criação de cláusulas de smart contracts para definir as regras de organização dos grupos. |
| 1. Eu, como agente da seguradora, quero ter como publicar o contrato a fim de torná-lo válido e funcional. |
| 1. Eu, como agente da seguradora, quero ter como visualizar as solicitações de indenização realizadas pelos participantes a fim de avaliar o caso. |
| 1. Eu, como agente da seguradora, desejo ter uma interface para, de acordo com a veracidade dos pedidos de indenização, autorizar ou não a remuneração dos respectivos indivíduos da solicitação. |

| **User Story - Rebeca** |
| --- |
| 1. Eu, como participante, quero ter uma interface que permita aceitar um convite de entrada de adesão a um smart contract para fazer parte de um grupo de seguro mútuo peer-to-peer. |
|
|
| 1. Eu, como participante, quero ter uma interface que permita fazer um depósito no smart contract para manter o percentual mínimo da reserva de risco. |
| 1. Eu, como participante, quero ter uma interface que permita solicitar a indenização do seguro celular para receber o valor de cobertura do smart contract |

Tabelas critério de aceitação e testes de validação:

| Número | 1 |
| --- | --- |
| Título | Criação de smart contracts |
| Personas | Augusto - agente da seguradora |
| História | Eu, como agente da seguradora, quero ter uma interface que permita a criação de cláusulas de smart contracts para definir as regras de organização dos grupos. |
| Critérios de aceitação | CR-01 - deve ser fornecido o percentual mínimo do valor protegido que será mantido nas reservas de risco (ex: 5%)  CR-02 - deve ser fornecida a taxa administrativa da Seguradora P2P no momento da ativação do grupo (ex: 10%).  CR-03 - deve ser fornecido o endereço da carteira da seguradora para recebimento de taxas administrativas e posse do smart contract. |
| Testes de aceitação | Critério de aceitação: CR-01   1. Seguradora não forneceu o valor mínimo    1. Aceito = errado.    2. Recusado = correto. 2. Seguradora fornece o valor mínimo    1. Aceito = correto.    2. Recusado = errado.   Critério de aceitação: CR-02   1. Seguradora não forneceu o valor da taxa    1. Aceito = errado.    2. Recusado = correto. 2. Seguradora fornece o valor da taxa    1. Aceito= correto.    2. Recusado = errado.   Critério de aceitação: CR-03   1. Seguradora não forneceu o endereço da carteira    1. Aceito= errado.    2. Recusado= correto. 2. Seguradora forneceu a carteira    1. Aceito = correto.    2. Recusado = errado. |

Gera link

| Número | 2 |
| --- | --- |
| Título | Compartilhamento do smart contract |
| Personas | Augusto - agente da seguradora |
| História | Eu, como agente da seguradora, quero compartilhar o contrato para possíveis clientes interessados no serviço de seguro mútuo de smartphone, a fim de que os seguros sejam cotados e haja menos burocracia para o processo. |
| Critérios de aceitação | CR-01 - deve ser possível compartilhar o link de adesão referente àquele contrato por email. |
| Testes de aceitação | Critério de aceitação: CR-01   1. Email invalido:    1. Aceito: errado.    2. Recusado: correto. 2. Email Válido:    1. Aceito: correto.    2. Recusado: errado. |

Após execução user story 2 o cliente terá como acessar o smart contract

| Número | 6 |
| --- | --- |
| Título | Entrar no smart contract |
| Personas | Rebeca - Futura cliente |
| História | Eu, como participante, quero ter uma interface que permita aceitar um convite de entrada de adesão a um smart contract para fazer parte de um grupo de seguro mútuo peer-to-peer. |
| Critérios de aceitação | CR-01 - participante deve fornecer sua carteira Ethereum  CR-02 - participante deve fornecer informações do dispositivo  Informações do dispositivo: Marca/Modelo, Valor da cobertura e IMEI  CR-03 - somente cadastradas carteiras com saldo disponível acima da taxa administrativa da Seguradora P2P no momento da ativação do grupo (ex: 10%). |
| Testes de aceitação | Critério de aceitação: CR-01   1. Carteira do participante não possui saldo mínimo    1. Aplicação web aceita: errado, deve ser redirecionado.    2. Aplicação web recusa: correto. 2. Carteira do participante possui saldo mínimo    1. Aplicação web aceita: correto    2. Aplicação web recusa: errado, deve ser redirecionado. 3. Carteira do participante inexiste    1. Aplicação web aceita: errado, deve ser redirecionado.    2. Aplicação web recusa: correto. 4. Carteira do participante existe    1. Aplicação web aceita: correto.    2. Aplicação web recusa: errado.   Critério de aceitação: CR-02   1. Participante informa "358267715108 984" como IMEI    1. Aplicação web aceita: errado, não deve haver espaço na numeração.    2. Aplicação web recusa: correto. 2. Participante informa "358267715108984" como IMEI    1. Aplicação web aceita: correto.    2. Aplicação web recusa: errado, deve ser corrigido. 3. Participante não fornece informações do dispositivo    1. Aplicação web aceita: errado, deve ser corrigido    2. Aplicação web recusa: correto. 4. Participante fornece informações do dispositivo    1. Aplicação web aceita: correto.    2. Aplicação web recusa: errado, deve ser corrigido.   Critério de aceitação: CR-03   1. A carteira cadastrada do usuário não possui mais do que a quantia de 10% da taxa administrativa.    1. Entrou = errado, deve ser redirecionado    2. Recusou = correto. 2. A carteira cadastrada do usuário possui mais do que a quantia de 10% da taxa administrativa.    1. Entrou = correto.    2. Recusou = errado, deve ser corrigido |

A partir desse momento, com os cliente aprovados,

| Número | 3 |
| --- | --- |
| Título | Publicar smart contract |
| Personas | Augusto - agente da seguradora |
| História | Eu, como agente da seguradora, quero ter como publicar o contrato a fim de torná-lo válido e funcional |
| Critérios de aceitação | CR-01 - todos os pedidos de adesão (senders) devem ter sido de convidados ao smart contract  CR-02 - é preciso haver no mínimo um dono do smart contract  CR-03 - a taxa administrativa deverá ser contabilizada de cada participante  CR-04 - nenhum dado pessoal sensível foi publicado no smart contract |
| Testes de aceitação | Critério de aceitação: CR-01   1. Um dos participantes não foi convidado.    1. Entrou = errado.    2. Recusou = correto. 2. Todos os participantes foram convidados    1. Entrou = correto.    2. Recusou = errado.   Critério de aceitação: CR-02   1. O endereço da carteira da seguradora, fornecido na criação do smart contract, não é válido    1. Aceito= errado.    2. Recusado= correto. 2. O endereço da carteira da seguradora, fornecido na criação do smart contract, continua válido    1. Aceito= correto.    2. Recusado= errado.   Critério de aceitação: CR-03   1. O valor total das transações recebidas é diferente da soma das respectivas taxas administrativas    1. Aceito= errado.    2. Recusado= correto. 2. O valor total das transações recebidas é igual à soma das respectivas taxas administrativas    1. Aceito= correto.    2. Recusado= errado.   Critério de aceitação: CR-04   1. Não foi aplicada no IMEI fornecido pelos participantes uma função hash    1. Aceito= errado.    2. Recusado= correto. 2. Foi aplicada no IMEI fornecido pelos participantes uma função hash    1. Aceito= correto.    2. Recusado= errado. |

O smart contract foi publicado e agora os participantes fazem os depósitos mensais

| Número | 7 |
| --- | --- |
| Título | Pagamento mensal |
| Personas | Rebeca - Futura cliente |
| História | Eu, como participante, quero ter uma interface que permita fazer um aporte no smart contract para manter o percentual mínimo da reserva de risco. |
| Critérios de aceitação | CR-01 - O valor da cobertura deve ser fornecido em ETH para o smart contract |
| Testes de aceitação | Critério de aceitação: CR-01   1. O smart contract recebeu o aporte de R$100    1. Aceito = errado.    2. Recusado = correto. 2. O smart contract recebeu o aporte de 0,012ETH    1. Aceito = correto.    2. Recusado = errado. |

Depois dos aportes o participante pode realizar o sinistro quando for necessário

| Número | 8 |
| --- | --- |
| Título | Pedido de indenização |
| Personas | Rebeca - Futura cliente |
| História | Eu, como participante, quero ter uma interface que permita solicitar a indenização do seguro celular para receber o valor de cobertura do smart contract |
| Critérios de aceitação | CR-01 - O hash do IMEI e o valor da indenização devem ser fornecidos na solicitação  CR-02 - Devem ser fornecidos documentos que dão base para o pedido (Boletim de Ocorrência, …) |
| Testes de aceitação | Critério de aceitação: CR-01   1. o hash do IMEI não condiz com o do smart contract    1. Indenização aceita = errado.    2. Indenização negada = correto. 2. o hash do IMEI condiz com o do smart contract    1. Indenização aceita = correto.    2. Indenização negada = errado. |

Após o pedido de indenização, o agente da seguradora receberá uma fila de pedidos

| Número | 4 |
| --- | --- |
| Título | Administração do smart contract |
| Personas | Augusto - agente da seguradora |
| História | Eu, como agente da seguradora, quero ter como visualizar as solicitações de indenização realizadas pelos participantes a fim de avaliar o caso. |
| Critérios de aceitação | CR-01 - Somente o próprio participante pode solicitar indenização.  CR-02 - Solicitações de indenização devem respeitar ordem cronológica de pedido  Exemplo: Pedido A = 16:00, Pedido B = 14:00 |
| Testes de aceitação | Critério de aceitação: CR-01   1. sender é diferente do endereço do participante registrado no smart contract    1. Indenização aceita = errado.    2. Indenização negada = correto.   Critério de aceitação: CR-02   1. Pedido A foi autorizado antes do pedido B    1. Indenização aceita = errado.    2. Indenização negada = correto. |

O agente da seguradora avaliará os pedidos de sinistro

| Número | 5 |
| --- | --- |
| Título | Administração do smart contract |
| Personas | Augusto - agente da seguradora |
| História | Eu, como agente da seguradora, desejo ter uma interface para, de acordo com a veracidade dos pedidos de indenização, autorizar ou não a remuneração dos respectivos indivíduos da solicitação. |
| Critérios de aceitação | CR-01 - Todo o saldo atual do participante que solicitou indenização deve ser consumido.  CR-02 - O restante do valor X a ser indenizado será dividido por todos os demais participantes, proporcionalmente ao saldo de cada um |
| Testes de aceitação | Critério de aceitação: CR-01   1. O participante tem saldo de R$2000 restando R$0    1. Indenização aceita = correta.    2. Indenização negada = errado.   Critério de aceitação: CR-02 |

### 4.2.2. Versão 2.0

| **User Story - Augusto** |
| --- |
| 1. Eu, como agente da seguradora, quero ter uma interface que permita a criação de cláusulas de smart contracts para definir as regras de organização dos grupos. |
| 1. Eu, como agente da seguradora, quero ter como visualizar os grupos e seus integrantes a fim de monitorar se a quantidade de participantes está ideal ou administrá-lo. |
| 1. Eu, como agente da seguradora, quero ter como publicar o contrato a fim de torná-lo válido e funcional. |
| 1. Eu, como agente da seguradora, quero ter como visualizar as solicitações de indenização realizadas pelos participantes a fim de avaliar o caso e autorizar ou não a remuneração. |

| **User Story - Rebeca** |
| --- |
| 1. Eu, como futuro participante, quero fornecer as informações do meu celular a fim de possibilitar um futuro convite a um smart contract de seguro mútuo. |
| 1. Eu, como participante, quero ter uma interface que permita aceitar um convite de entrada de adesão a um smart contract para fazer parte de um grupo de seguro mútuo peer-to-peer. |
|
|
| 1. Eu, como participante, quero ter uma interface que permita fazer um aporte no smart contract para pagar a taxa administrativa e manter ao menos o percentual mínimo da reserva de risco. |
| 1. Eu, como participante, quero ter uma interface que permita solicitar a indenização do seguro celular para receber o valor de cobertura do smart contract. |

Tabelas critério de aceitação e testes de validação:

| **Número** | **5** |
| --- | --- |
| **Título** | **Participação no smart contract** |
| **Personas** | **Rebeca - Futura cliente** |
| **História** | Eu, como futuro participante, quero fornecer as informações do meu celular a fim de possibilitar um futuro convite a um smart contract de seguro mútuo. |
| **Critérios de aceitação** | **CR-01 -** participantedeve fornecer sua carteira Ethereum  **CR-02 -** participantedeve fornecer informações do dispositivo  **Informações do dispositivo:** Marca/Modelo, Valor da cobertura e IMEI |
| **Testes de aceitação** | Critério de aceitação: **CR-01**   1. Carteira do participante não possui saldo mínimo    1. Aplicação web aceita: errado, deve ser redirecionado.    2. Aplicação web recusa: correto. 2. Carteira do participante possui saldo mínimo    1. Aplicação web aceita: correto    2. Aplicação web recusa: errado, deve ser redirecionado. 3. Carteira do participante inexiste    1. Aplicação web aceita: errado, deve ser redirecionado.    2. Aplicação web recusa: correto. 4. Carteira do participante existe    1. Aplicação web aceita: correto.    2. Aplicação web recusa: errado.   Critério de aceitação: **CR-02**   1. Participante informa "358267715108 984" como IMEI    1. Aplicação web aceita: errado, não deve haver espaço na numeração.    2. Aplicação web recusa: correto. 2. Participante informa "358267715108984" como IMEI    1. Aplicação web aceita: correto.    2. Aplicação web recusa: errado, não deve haver espaço na numeração. 3. Participante não fornece informações do dispositivos    1. Aplicação web aceita: errado, deve ser corrigido    2. Aplicação web recusa: correto. 4. Participante fornece informações do dispositivo    1. Aplicação web aceita: correto.    2. Aplicação web recusa: errado, deve ser corrigido. |

| **Número** | **1** |
| --- | --- |
| **Título** | **Criação de smart contracts** |
| **Personas** | **Augusto - agente da seguradora** |
| **História** | Eu, como agente da seguradora, quero ter uma interface que permita a criação de cláusulas de smart contracts para definir as regras de organização dos grupos. |
| **Critérios de aceitação** | **CR-01 -** deve ser fornecido o percentual mínimo do valor protegido que será mantido nas reservas de risco (ex: 5%)  **CR-02 -** deve ser fornecida a taxa administrativa da Seguradora P2P no momento da ativação do grupo (ex: 10%).  **CR-03 -** deve ser fornecido o endereço da carteira da seguradora para recebimento de taxas administrativas e posse do smart contract. |
| **Testes de aceitação** | Critério de aceitação: **CR-01**   1. Seguradora não forneceu o valor mínimo    1. Aceito = errado.    2. Recusado = correto. 2. Seguradora fornece o valor mínimo    1. Aceito = correto.    2. Recusado = errado.   Critério de aceitação: **CR-02**   1. Seguradora não forneceu o valor da taxa    1. Aceito = errado.    2. Recusado = correto. 2. Seguradora fornece o valor da taxa    1. Aceito= correto.    2. Recusado = errado.   Critério de aceitação: **CR-03**   1. Seguradora não forneceu o endereço da carteira    1. Aceito= errado.    2. Recusado= correto. 2. Seguradora forneceu a carteira    1. Aceito = correto.    2. Recusado = errado. |

A partir desse momento, os clientes presentes nesta faixa de valor da franquia serão automaticamente convidados a participar do grupo.

| **Número** | **6** |
| --- | --- |
| **Título** | **Entrar no smart contract** |
| **Personas** | **Rebeca - Futura cliente** |
| **História** | Eu, como participante, quero ter uma interface que permita aceitar um convite de entrada de adesão a um smart contract para fazer parte de um grupo de seguro mútuo peer-to-peer. |
| **Critérios de aceitação** | **CR-03 -** somente cadastradas carteiras com saldo disponível acima da taxa administrativa da Seguradora P2P no momento da ativação do grupo (ex: 10%). |
| **Testes de aceitação** | Critério de aceitação: **CR-03**   1. A carteira cadastrada do usuário não possui mais do que a quantia de 10% da taxa administrativa.    1. Entrou = errado, deve ser redirecionado    2. Recusou = correto. 2. A carteira cadastrada do usuário possui mais do que a quantia de 10% da taxa administrativa.    1. Entrou = correto.    2. Recusou = errado, deve ser corrigido |

A partir desse momento, com os clientes aprovados, a seguradora pode monitorar os grupos

| **Número** | **2** |
| --- | --- |
| **Título** | **Monitorar grupos** |
| **Personas** | **Augusto - agente da seguradora** |
| **História** | Eu, como agente da seguradora, quero ter como visualizar os grupos e seus integrantes a fim de monitorar se a quantidade de participantes está ideal ou administrá-lo. |
| **Critérios de aceitação** | **CR-01 -** Caso o contrato do grupo não esteja publicado, deve ser possível visualizar quantos participantes há.  **CR-02 -** Caso o contrato do grupo esteja publicado, deve ser possível desativar o grupo. |
| **Testes de aceitação** | Critério de aceitação: **CR-01**   1. A plataforma indica um grupo sem a quantidade mínima de participantes atingida como pronto (verde).    1. Aceito= errado.    2. Recusado= correto. 2. A plataforma indica um grupo com a quantidade mínima de participantes atingida como pronto (verde).    1. Aceito= correto.    2. Recusado= errado. |

.

Posteriormente à monitoração do grupo, assim que possível, a seguradora fará a ativação do grupo.

| **Número** | **3** |
| --- | --- |
| **Título** | **Publicar smart contract** |
| **Personas** | **Augusto - agente da seguradora** |
| **História** | Eu, como agente da seguradora, quero ter como publicar o contrato a fim de torná-lo válido e funcional. |
| **Critérios de aceitação** | **CR-01 -** todos os pedidos de adesão (senders) devem ter sido de convidados ao smart contract  **CR-02 -** é preciso haver no mínimo um dono do smart contract  **CR-03 -** a taxa administrativa deverá ser contabilizada de cada participante  **CR-04 -** nenhum dado pessoal sensível foi publicado no smart contract |
| **Testes de aceitação** | Critério de aceitação: **CR-01**   1. Um dos participantes não foi convidado.    1. Entrou = errado.    2. Recusou = correto. 2. Todos os participantes foram convidados    1. Entrou = correto.    2. Recusou = errado.   Critério de aceitação: **CR-02**   1. O endereço da carteira da seguradora, fornecido na criação do smart contract, não é válido    1. Aceito= errado.    2. Recusado= correto. 2. O endereço da carteira da seguradora, fornecido na criação do smart contract, continua válido    1. Aceito= correto.    2. Recusado= errado.   Critério de aceitação: **CR-03**   1. O valor total das transações recebidas é diferente da soma das respectivas taxas administrativas    1. Aceito= errado.    2. Recusado= correto. 2. O valor total das transações recebidas é igual à soma das respectivas taxas administrativas    1. Aceito= correto.    2. Recusado= errado.   Critério de aceitação: **CR-04**   1. Não foi aplicada no IMEI fornecido pelos participantes uma função hash    1. Aceito= errado.    2. Recusado= correto. 2. Foi aplicada no IMEI fornecido pelos participantes uma função hash    1. Aceito= correto.    2. Recusado= errado. |

O smart contract foi publicado e agora os participantes podem fazer os aportes, tanto a taxa administrativa quanto o valor percentual mínimo.

| **Número** | **7** |
| --- | --- |
| **Título** | **Aporte dos valores** |
| **Personas** | **Rebeca - Futura cliente** |
| **História** | Eu, como participante, quero ter uma interface que permita fazer um aporte no smart contract para pagar a taxa administrativa e manter ao menos o percentual mínimo da reserva de risco. |
| **Critérios de aceitação** | **CR-01 -** O valor da cobertura deve ser fornecido em ETH para o smart contract  **CR-02 -** O primeiro aporte deve ser destinado ao pagamento da taxa administrativa. |
| **Testes de aceitação** | Critério de aceitação: **CR-01**   1. O smart contract recebeu o aporte de R$100    1. Aceito = errado.    2. Recusado = correto. 2. O smart contract recebeu o aporte de 0,012ETH    1. Aceito = correto.    2. Recusado = errado.   Critério de aceitação: **CR-02**   1. O primeiro aporte foi destinado à reserva de risco    1. Aceito = errado.    2. Recusado = correto. 2. O primeiro aporte foi destinado à taxa administrativa    1. Aceito = correto.    2. Recusado = errado. |

Depois dos aportes o participante pode realizar o acionamento do sinistro quando for necessário.

| **Número** | **8** |
| --- | --- |
| **Título** | **Pedido de indenização** |
| **Personas** | **Rebeca - Futura cliente** |
| **História** | Eu, como participante, quero ter uma interface que permita solicitar a indenização do seguro celular para receber o valor de cobertura do smart contract. |
| **Critérios de aceitação** | **CR-01 -** O hash do IMEI e o valor da indenização devem ser fornecidos na solicitação  **CR-02 -** Devem ser fornecidos documentos que dão base para o pedido (Boletim de Ocorrência, texto…) |
| **Testes de aceitação** | Critério de aceitação: **CR-01**   1. o hash do IMEI não condiz com o do smart contract    1. Indenização aceita = errado.    2. Indenização negada = correto. 2. o hash do IMEI condiz com o do smart contract    1. Indenização aceita = correto.    2. Indenização negada = errado.   Critério de aceitação: **CR-02**   1. Um arquivo de B.O. não foi anexado ao pedido.    1. Aceito = errado.    2. Recusado = correto. 2. Um arquivo de B.O. foi anexado ao pedido.    1. Aceito = correto.    2. Recusado = errado. |

Após o pedido de indenização, o agente da seguradora receberá uma fila de pedidos

| **Número** | **4** |
| --- | --- |
| **Título** | **Administração do smart contract** |
| **Personas** | **Augusto - agente da seguradora** |
| **História** | Eu, como agente da seguradora, quero ter como visualizar as solicitações de indenização realizadas pelos participantes a fim de avaliar o caso e autorizar ou não a remuneração. |
| **Critérios de aceitação** | **CR-01 -** Somente o próprio participante pode solicitar indenização.  **CR-02 -** Solicitações de indenização devem respeitar ordem cronológica de pedido  **Exemplo:** Pedido A = 16:00, Pedido B = 14:00  **CR-03 -** Em caso de aceite, todo o saldo atual do participante que solicitou a indenização deve ser consumido.  **CR-04 -** Em caso de aceite, o valor do ressarcimento deve respeitar o seu Limite Máximo Indenizável (LMI).  **Exemplo:** O valor coberto é de R$2000 e a reserva de LMI do participante é de 300.  **CR-05 -** Em caso de aceite, o restante do valor X a ser indenizado será dividido por todos os demais participantes, proporcionalmente ao saldo de cada um. |
| **Testes de aceitação** | Critério de aceitação: **CR-01**   1. sender é diferente do endereço do participante registrado no smart contract    1. Aceito = errado.    2. Recusado = correto. 2. sender é o mesmo do endereço do participante registrado no smart contract    1. Aceito = correto.    2. Recusado = errado.   Critério de aceitação: **CR-02**   1. Pedido A foi autorizado antes do pedido B    1. Indenização aceita = errado.    2. Indenização negada = correto. 2. Pedido A foi autorizado depois do pedido B    1. Indenização aceita = correto.    2. Indenização negada = errado.   Critério de aceitação: **CR-03**   1. O participante tem saldo de R$2000 restando R$0    1. Indenização aceita = correta.    2. Indenização negada = errado.   Critério de aceitação: **CR-04**   1. O participante tem saldo de R$150 (50%), sendo indenizado R$2000.    1. Aceito = errado.    2. Recusado = correto. 2. O participante tem saldo de R$150 (50%), sendo indenizado R$1000.    1. Aceito = correto.    2. Recusado = errado. |

## 

## 4.3. Wireframe

As páginas abaixo são uma representação da aplicação web, que terá dois módulos: a área da seguradora P2P e a área do cliente. Essa representação da diagramação e das estruturas macro do site são utilizadas para contribuir com o desenvolvimento do design do front end, as funcionalidades do sistema e o alinhamento das interações.

### 4.3.1. Área da Seguradora P2P

A página abaixo, se refere a configuração e criação de novo grupo vinculado a ela (com deploy de novo smart contract). A seguradora para criá-lo deve informar o endereço da sua wallet, o percentual da reserva de risco (percentual mínimo do valor protegido que deve ser mantido nas reservas de risco), a taxa administrativa (taxa que é cobrada pela segurada para gerenciar os grupos) e o convite dos membros do grupo mútuo

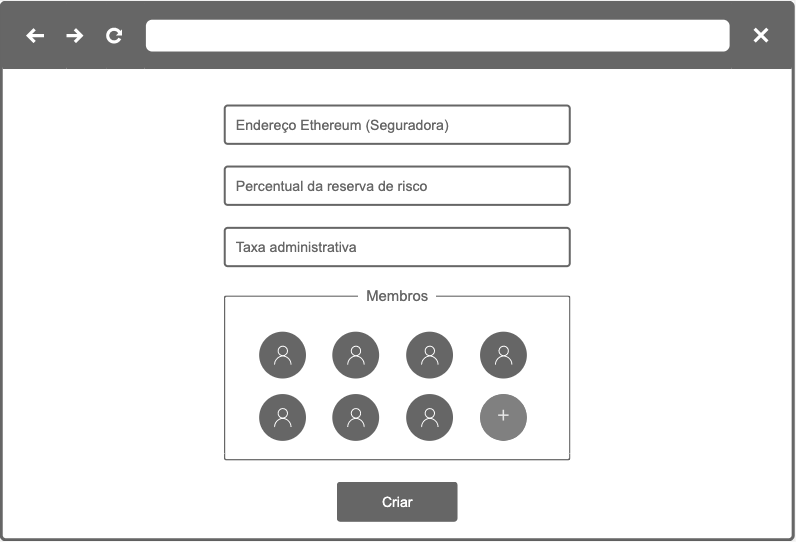


Imagem 8 - Página de criação de grupo mútuo.

A segunda página, se refere ao processo de regulação de sinistro da seguradora, no qual analisa o pedido de indenização do cliente. Através dela é possível visualizar o hash do IMEI, o valor da indenização solicitado e dados do boletim de ocorrência (meio de análise). Após analisar as informações a seguradora autoriza ou recusa a solicitação.



Imagem 9 - Página de Análise de Indenização.

### 4.3.2. Área do Cliente

Após a corretora enviar o convite para novos membros de um grupo mútuo, o cliente tem acesso a página abaixo. Para ingressar em um grupo, ele deve confirmar a marca/modelo do celular, o valor de cobertura (se trata do valor do aparelho), o hash do IMEI do aparelho, e por fim, o depósito do valor mínimo necessário para compor a reserva de risco.

-

Imagem 10 - Página de Adesão.

A página de pagamento, serve para o cliente fazer um depósito adicional/avulso ao smart contract para repor suas reservas de risco e restaurar a capacidade de cobrir o Limite Máximo de Indenização (LMI) original informado pelo Participante. Será possível visualizar o saldo que há no smart contract, e para concluir o depósito, é preciso informar o hash do IMEI, o valor do depósito, e por fim, concluir o pagamento.

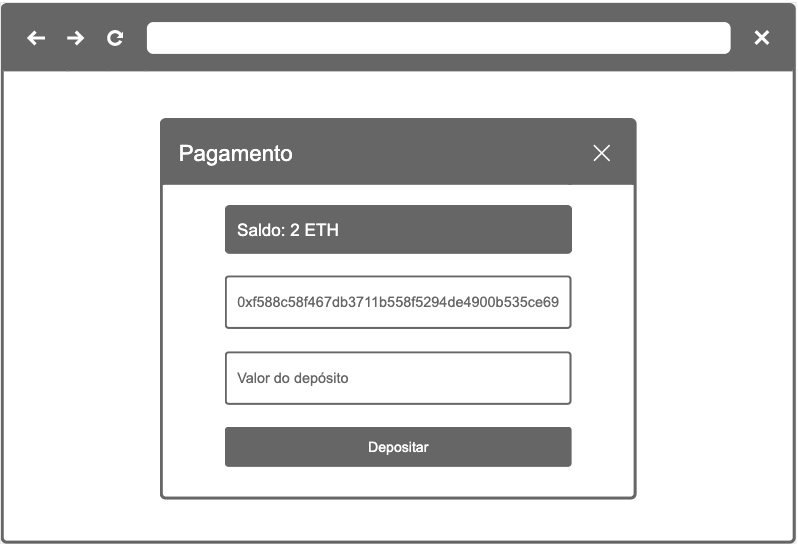


Imagem 11 - Página de Pagamento.

Por fim, o cliente tem acesso a página de pedido de indenização, no qual ele aciona o seguro do seu aparelho. Para isso, é necessário informar o hash do IMEI, o valor da indenização solicitado e dados do boletim de ocorrência (meio de análise da seguradora P2P).



Imagem 12 - Página de Pedido de Indenização.

## 4.4. Protótipo de interface com o usuário

A partir dos wireframes, foram criados mockups no Figma para Usuário e Administradora, separadamente. Para o usuário, foi utilizado o modelo mobile, pensando na usabilidade e a existência de um aplicativo da Coover, em contrapartida, a administradora possui uma plataforma desktop.

Para acessar os mockups das interfaces, acesse os links:

* Usuário: [Figma usuário](https://www.figma.com/file/sXgHqyUspfpGqlVSploFQ0/CooverUsu%C3%A1rio?node-id=0%3A1&t=8y66FPLOhj1UmDjA-1)
* Administrador: [Figma Adm Coover](https://www.figma.com/file/jSOqXMDy2Kepu2m9cnOlLm/CooverAdministradora?node-id=0%3A1&t=Juq3foXQW6EX3DH6-1)

### 4.4.1.Usuário

#### 4.4.1.1. Jornada entrada

Na interface do usuário ao entrar você seguirá a seguinte ordem das páginas, sendo elas: Saber mais, entrar, criar login, plugin da Metamask (sem essa última parte não é possível entrar e fazer parte da nossa aplicação)e login . A seguir imagens das páginas na sua respectiva ordem:

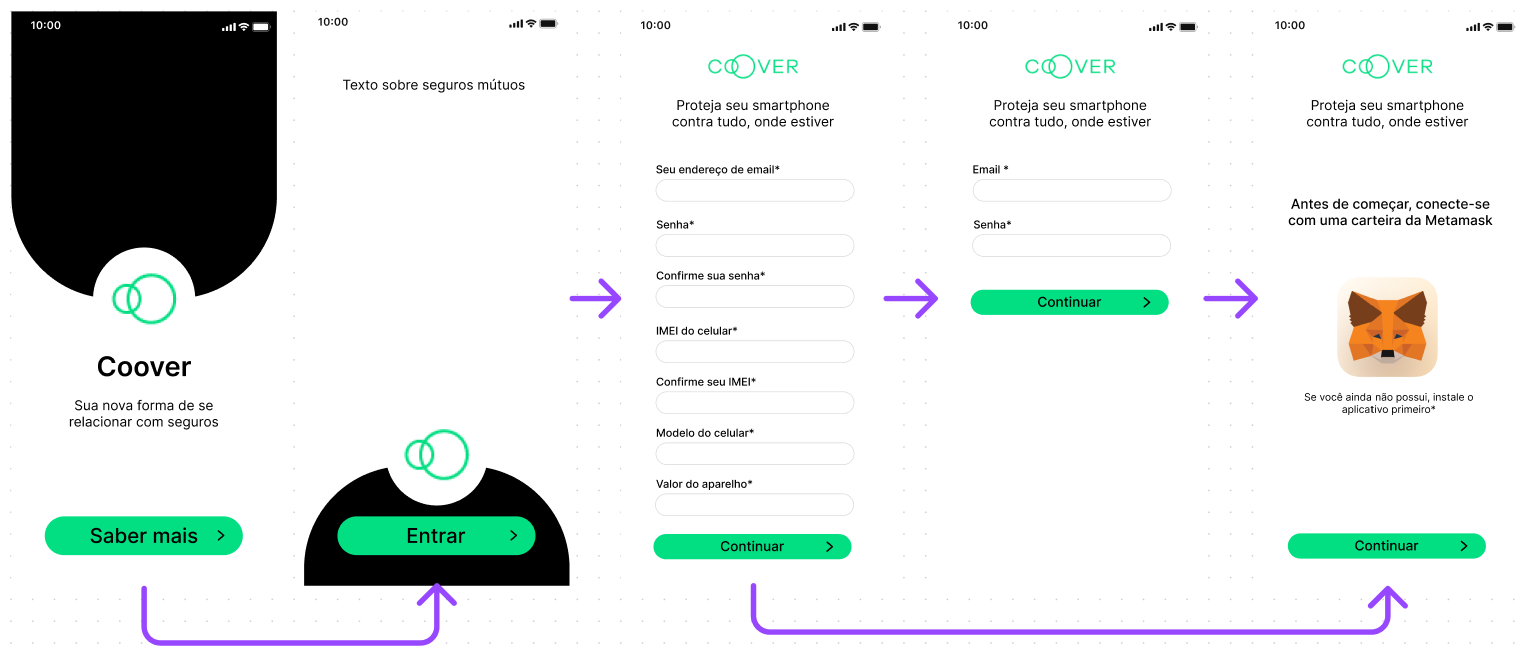


Imagem 13 - Jornada de entrada

#### 

#### 4.4.1.2.Jornada convite

A Primeiro momento, como o cliente não participa de nenhum grupo, na página Dashboard, caso não exista um grupo ainda não ativo e compatível com o usuário, ele não terá nada na página “grupos”. Após um tempo ele receberá um convite para participar. Ao clicar no convite você consegue visualizar todas as informações do seguro; A seguir imagens das páginas na sua respectiva ordem:

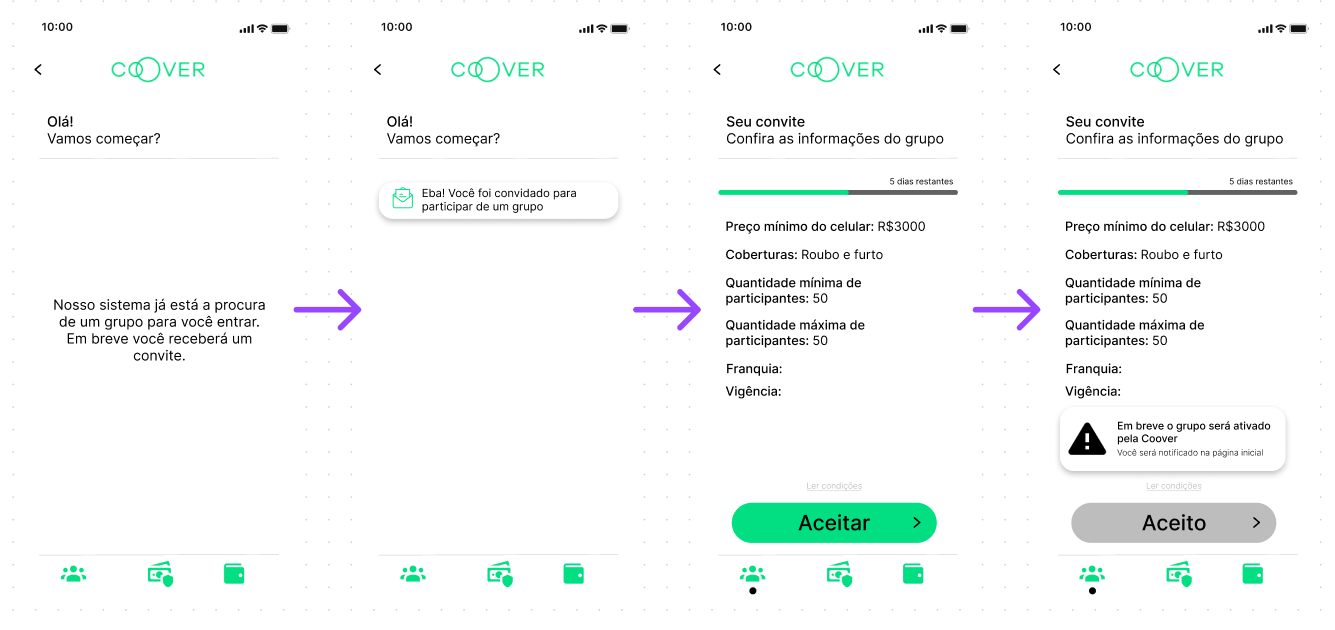


Imagem 14 - Jornada Convite

#### 

#### 4.4.1.3.Jornada Pagamento

Após aceitar o convite e a Administradora ativá-lo na rede blockchain, é necessário fazer o primeiro pagamento, em que há a retirada da taxa administrativa por fins de cobertura de gas fee, para, só assim, realmente poder realizar as demais transações do contrato. Além disso, na página “Fundos” o cliente pode visualizar seu saldo pessoal no smart contract e repor sua reserva para garantir o limite máximo indenizável total. A seguir imagens das páginas na sua respectiva ordem:

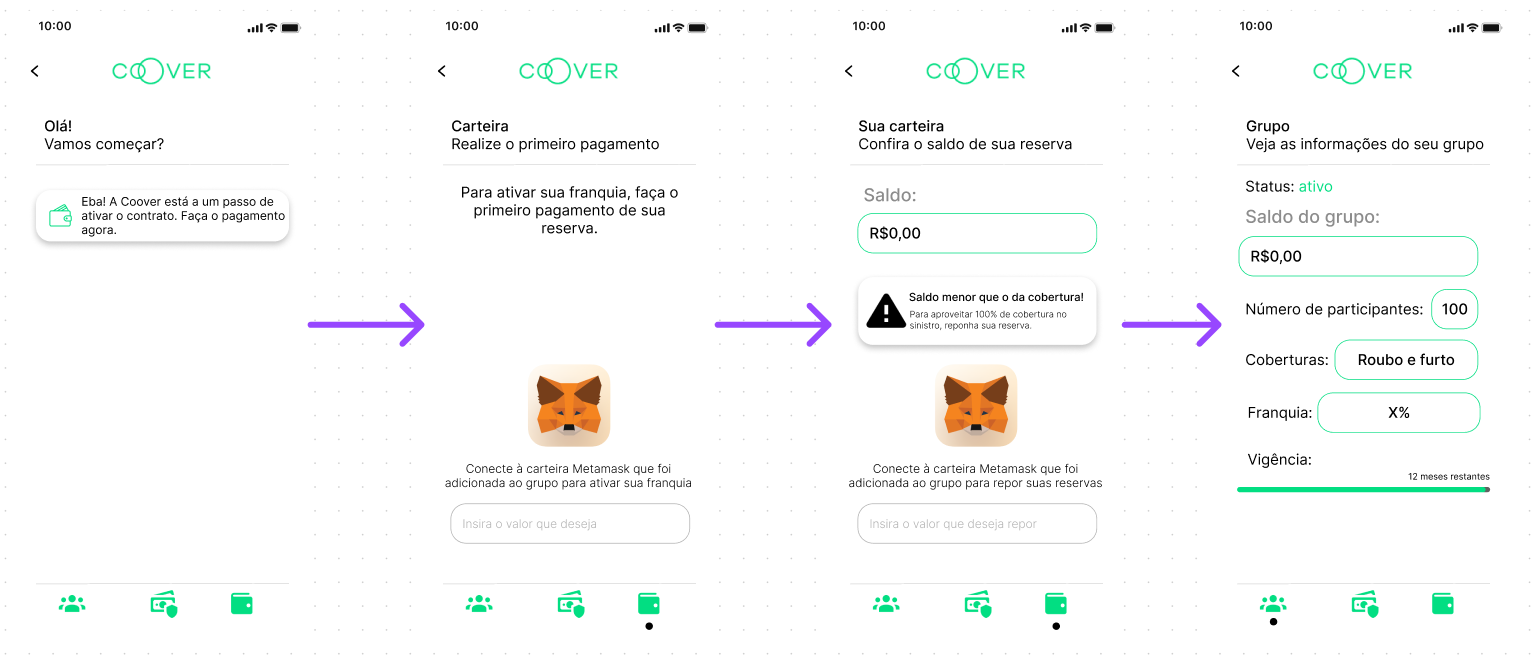


Imagem 15 - Jornada Pagamento

#### 4.4.1.4.Jornada pedido de indenização

Caso o cliente queira pedir uma indenização, ele entra na página “Indenização” e envia os dados necessários (Imei, valor requisitado e motivo). Após clicar em solicitar, o pedido ficará em análise até que a Coover aceite/recuse a proposta. A seguir imagens das páginas na sua respectiva ordem:

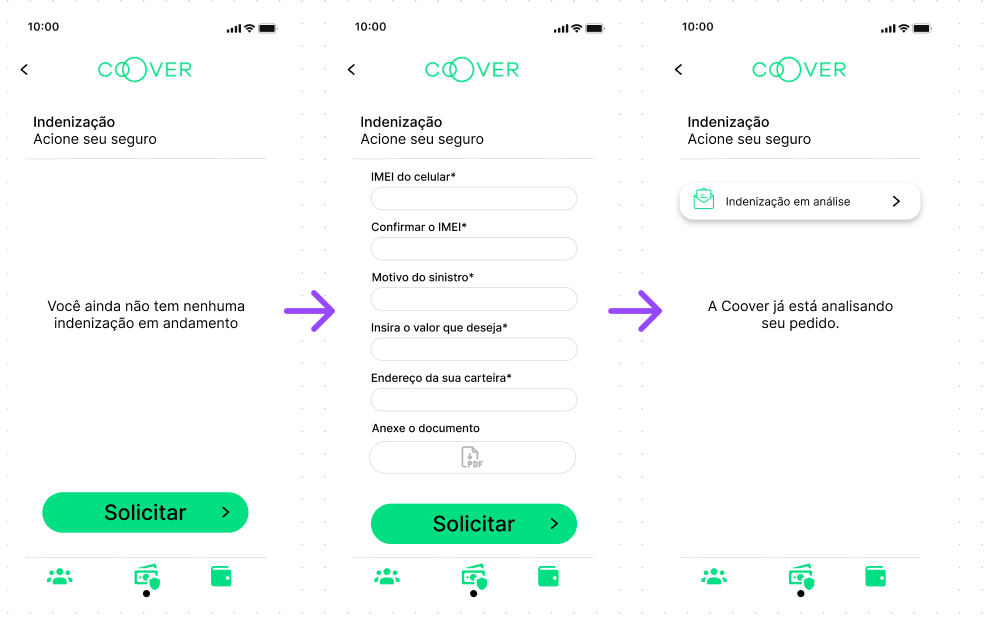


Imagem 16 - Jornada pedido de indenização

### 4.4.2.Administrador Coover

#### 

#### 4.4.2.1.Jornada Login

Ao entrar na plataforma admin Coover, a primeira página será a "Grupos ”, onde o usuário poderá ter uma visão mais geral de todos os grupos, porém no primeiro acesso ele não terá nenhum grupo. A seguir imagens das páginas na sua respectiva ordem:

#### 

Imagem 17 - Jornada Login Administrador

#### 4.4.2.2.Jornada Novo contrato

Clicando em “Criar grupo” ou no ícone “+” você será direcionado a página “Novo contrato” onde o administrador insere os dados referentes ao seguro a ser criado (Taxa Administrativa, Taxa de Limite Máximo Indenizável, Mínimo e máximo de pessoas, Valor mínimo do celular e data de expiração). A seguir imagens das páginas na sua respectiva ordem:

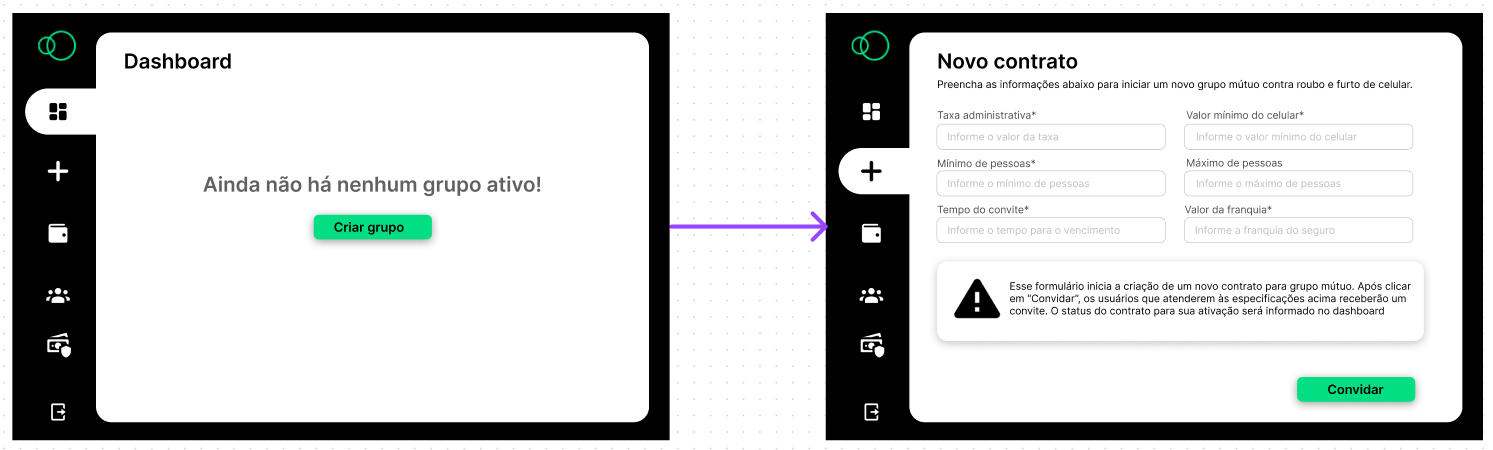


Imagem 18 - Jornada Novo Contrato

#### 4.4.2.3.Jornada Carteira Coover

Clicando no ícone de carteira na tabela de “Grupos”, você será direcionado a página “Carteira Coover” onde o administrador pode sacar o valor referente ao total das taxas administrativas do respectivo contrato. A seguir imagem da página:

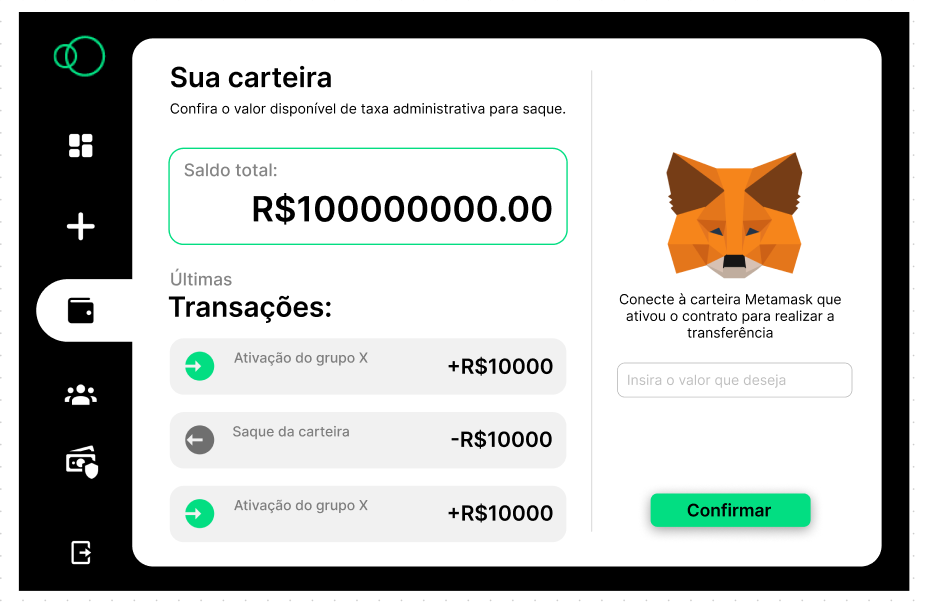


Imagem 19 - Jornada Carteira Coover

#### 4.4.2.4.Jornada Visualizar grupos

Clicando no ícone de “pessoas” você será direcionado a página “Grupos” onde o administrador tem uma tabela com todos os grupos, seus status e saldos. Além disso, ao selecionar um desses grupos, você pode ter mais dados sobre ele. A seguir imagens das páginas:

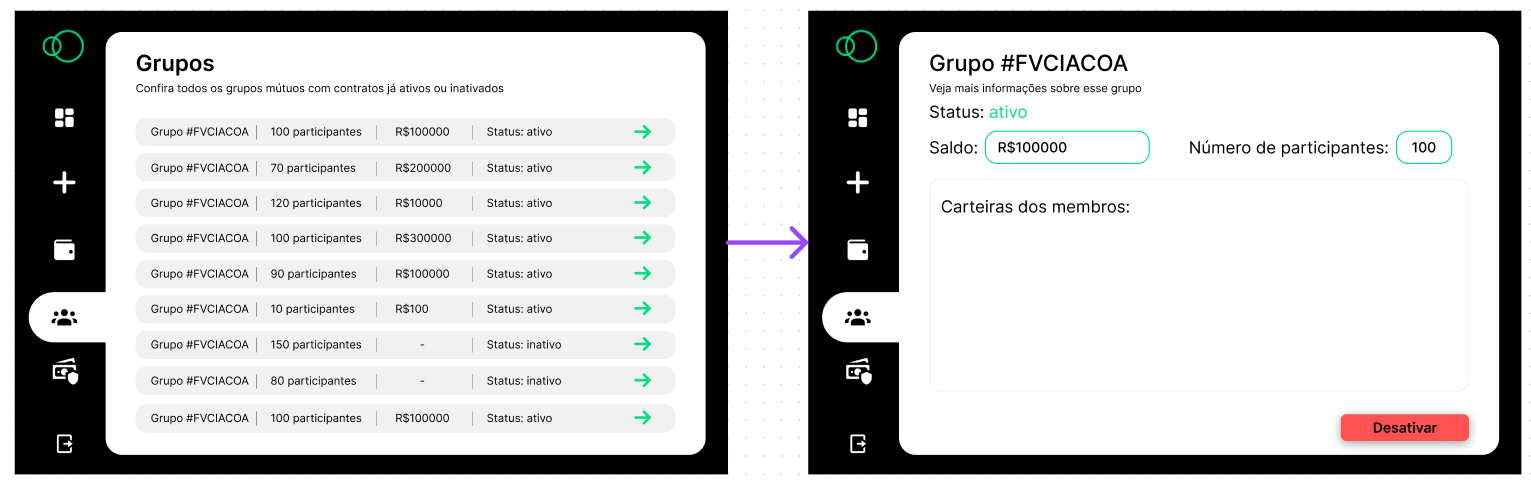


Imagem 20 - Jornada Visualizar Grupos

#### 

#### 4.4.2.5.Jornada Solicitações

Clicando no penúltimo ícone você será direcionado a página “Solicitações” onde o administrador tem uma visão de todos os pedidos de indenização e seus status. Além disso, ao selecionar uma dessas solicitações você pode ter mais dados sobre e aprová-la ou recusá-la, caso ainda esteja em análise. A seguir imagens das páginas:

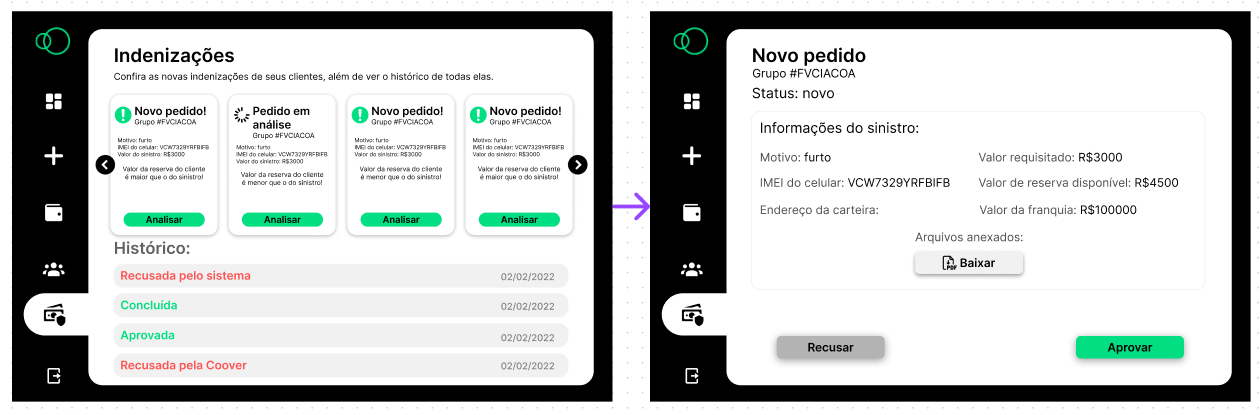


Imagem 21- Jornada Solicitações

### 4.4.5. Interfaces criadas

Além disso, criamos as interfaces web tanto do usuário (para dispositivos móveis) quanto para administradores da Coover (versão desktop). Você pode acessar o vídeo com a navegação das telas neste link : [Vídeo coover](https://drive.google.com/drive/folders/1ZevSa-5TJMhhQExamLFv2Z7LSGan_Yc6?usp=share_link)

# 5. Solução Proposta

## 5.1. Solução

Na busca por oferecer produtos securitários de maneira eficiente, barata e segura, foi proposto o desenvolvimento de uma solução em blockchain ethereum que permita a criação de grupos de seguro mútuos, nos quais são um conjunto de pessoas que se unem e formam grupos, auto administrando-se em alguma medida, para se protegerem contra as consequências econômicas advindas da concretização de riscos predeterminados.

A administração do grupo será realizada através de smart contracts, que são programas que se executam de forma automática assim que certas condições acordadas previamente pelas partes são atendidas. Nesse sentido, todas as regras de negócio e as reservas financeiras serão mantidas neste contrato inteligente.

Os usuários irão interagir com o sistema através de uma aplicação híbrida em Web 3, que conterá as seguintes áreas:

1. Área do cliente (participante): processo de adesão, pagamento (reposição de reservas de risco), pedido de indenização;
2. Área da Seguradora P2P: configuração e criação de novo grupo vinculado a ela (com deploy de um novo smart contract), atuação no processo de indenização.

Além disso, é válido ressaltar que a seguradora terá o papel de criar grupos e gerenciá-los. Sendo assim, não há transferência de risco para a seguradora como é feito no seguro privado, ela apenas opera os grupos mútuos.

## 5.2. Arquitetura Proposta

A primeira arquitetura proposta, após o kick off com o Stakeholder Coover segue o diagrama abaixo, que inclui a aplicação web da Coover, com host em navegadores (Chrome) e plugin dos usuários com a Metamask ou outra wallet, que guarde as chaves públicas e privadas que permitem acesso aos criptoativos e o deploy dos smart contracts na EVM (Ethereum Virtual Machine) pela Coover.

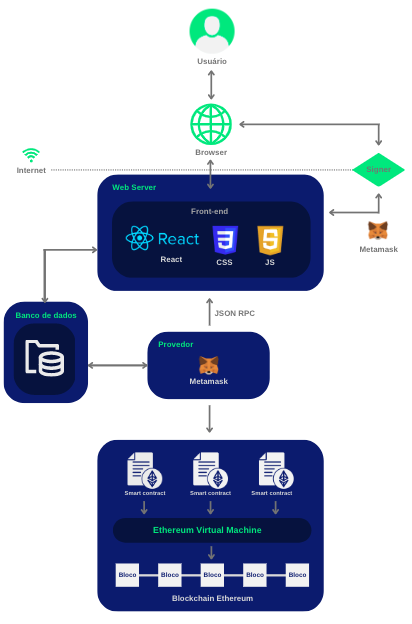
**

Imagem 21 - 2° Versão da Arquitetura

## 

## 5.3. Smart Contract

### 5.3.1. Regras de negócio

A fim de que nossa persona consiga realizar todas as user stories e as funções de forma mais estruturada estabelecemos as seguintes regras de negócio em nosso código:

#### Regra de negócio 1:

A Coover é definida como dona e administradora do contrato, devendo estabelecer os membros deste e seus IMEIs dos aparelhos protegidos, assim como a taxa administrativa.

#### Regra de negócio 2:

Os membros no deploy do contrato, devem realizar um pagamento inicial . Todo e qualquer valor deve ser depositado em ETH. O primeiro aporte deve ser referente ao percentual mínimo do valor protegido(definido pela Coover e específico de cada grupo, desse valor será separado uma porcentagem, também definida pelo criador do contrato, relativo à taxa administrativa); sem esse pagamento não é possível entrar no grupo.

#### Regra de negócio 3:

A Coover deve aprovar a indenização de um segurado e, a partir disso, é informado a carteira dele. O hash do segurado em que o sinistro ocorreu deve ser igual ao que está armazenado no contrato de sua carteira para que seja concretizado. Além disso, o cliente deve anexar um B.O sobre o “por quê” do pedido, com isso o administrador da Coover analisa o documento e “aprova” ou “recusa” a proposta.

#### Regra de negócio 4:

A Coover pode retirar a taxa administrativa a qualquer momento.

#### Regra de negócio 5:

O cliente deve conseguir ver seu valor de reserva para conferir o valor protegido do seguro.

#### Regra de negócio 6:

O cliente deve conseguir repor sua reserva. Todo e qualquer valor deve ser depositado em ETH. A reserva de risco deve atingir o percentual mínimo do valor protegido(definido pela Coover e específico de cada grupo, desse valor será separado uma porcentagem, também definida pelo criador do contrato, relativo à taxa administrativa);

### 5.3.2. Diagrama de blocos

O diagrama de blocos da solução quebra o projeto em partes menores e mais gerenciáveis, representadas por blocos ou módulos. Isso ajuda na compreensão da estrutura e dos componentes da solução por meio de uma representação visual simplificada.

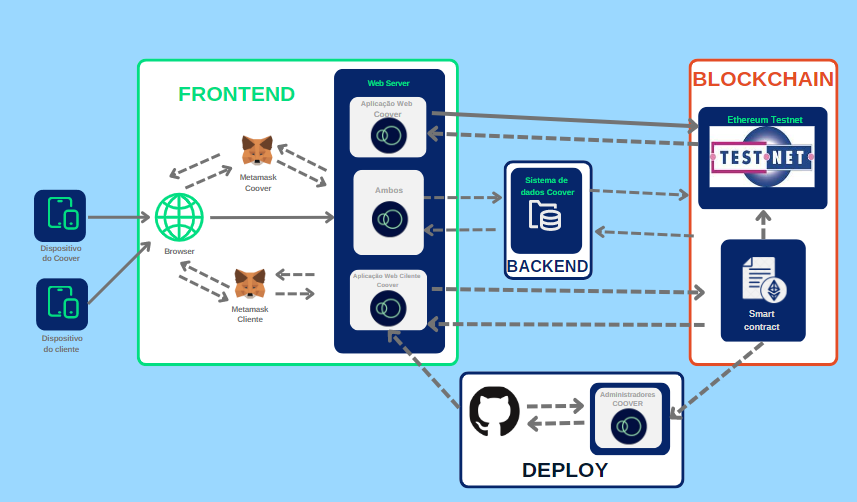


Imagem 22 - 1° Versão do diagrama de blocos

Na solução proposta há dois atores que interagem com o sistema, sendo estes, a seguradora P2P (representada pela Coover) e o participante (cliente do seguro). Essa interação dos usuários acontece através do Front end, onde se encontra a lógica da interface, além disso para acessar as funcionalidades da aplicação é necessário se conectar a Metamask, pois para realizar determinadas ações será necessário “assiná-la” (confirmar) usando a chave privada.

No bloco Backend, há o sistema de dados da Coover, onde são armazenados dados pessoais de seus clientes, que não podem ser publicados no smart contract, mas são necessários para definir a lógica de negócios do software. Por fim, a seguradora P2P (administradora) realiza o deploy do contrato (smart contract) na Ethereum Testenet, blockchain alternativa do Ethereum para testes globais. Nesse smar contract estará armazenado os fundos dos grupos mútuos e todas as regras de negócio, tais como a de pagamento de indenização.

### 5.3.3.UML

Os diagramas UML abaixo representam como o sistema funciona, retratando como os atores (Seguradora P2P e Participante) interagem com as funcionalidades da aplicação (evidenciando resultados ou respostas), tais como: criação de smart contract, pedido de indenização e reposição da reserva de risco. Os diagramas UML são importantes para alcançar maior clareza no comportamento esperado para o smart contract, definindo suas entradas e respostas.

#### 5.3.3.1. UML Indenização

Para a “indenização”, entendemos primeiro no âmbito de negócios. O usuário somente irá realizar essa ação quando ele for roubado e quiser requisitar a indenização. Nesse caso, teremos uma primeira fase de cadastro completo na MetaMask com login e autenticação e, posteriormente, a fim de validar que o celular roubado realmente está cadastrado no contrato, o usuário deverá fornecer o IMEI do celular através do nosso frontend. Esses são os passos necessários para o usuário encontrar e informar os dados pré requisitados.

Após esse processo inicial, pode-se seguir para a requisição em si, etapa em que o usuário enviará o B.O. pelo front-end, este que será repassado para a seguradora a fim de possibilitar a validação do pedido por uma instituição confiável. Em caso de aceite da indenização, o smart contract deve ressarcir o usuário através do endereço de carteira enviado.

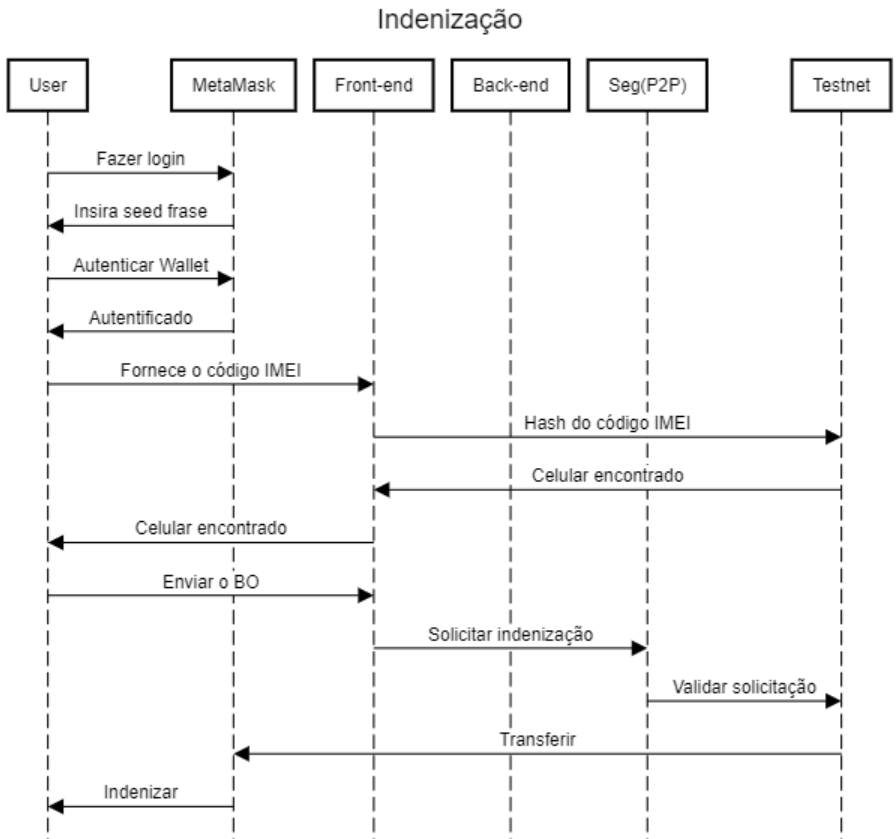


Imagem 23

#### 5.3.3.2. UML Indenização negada

Para a “indenização não aceita” temos o diagrama encurtado, nesse caso, teremos uma primeira fase de cadastro completo na MetaMask com login e autenticação. Seguindo os passos do UML anterior,o usuário deverá fornecer o IMEI do celular através do nosso frontend.

Posteriormente ao usuário enviar o B.O. pelo front-end, este que será repassado para a seguradora a fim de possibilitar a validação do pedido por uma instituição confiável, considerando que a seguradora recuse o pedido, uma notificação de pedido negado será retornada ao usuário.

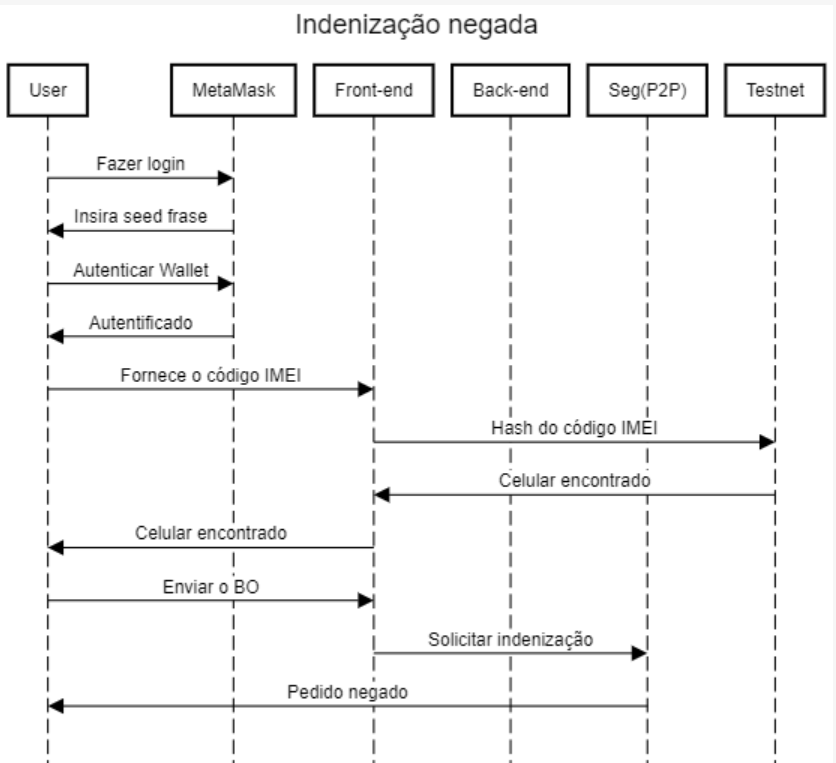


Imagem 24

#### 5.3.3.3. UML Repor reserva de risco

Para a reposição de reserva de risco, considerando o escopo limitado do projeto, o usuário poderá repor uma vez que um cliente do mesmo grupo do seguro mútuo for roubado e tiver sua indenização aceita, sendo assim, o usuário pode examinar através do front-end se a sua carteira possui uma porcentagem satisfatória do LMI (Limite máximo indenizável).

No caso do cliente estar insatisfeito com o seu saldo, ele poderá depositar dinheiro através de sua Meta Mask e, logo em seguida, transferir para o contrato, atualizando assim seu montante financeiro dentro das regras do smart contract. Então o front end irá informar ao cliente que o saldo foi atualizado.

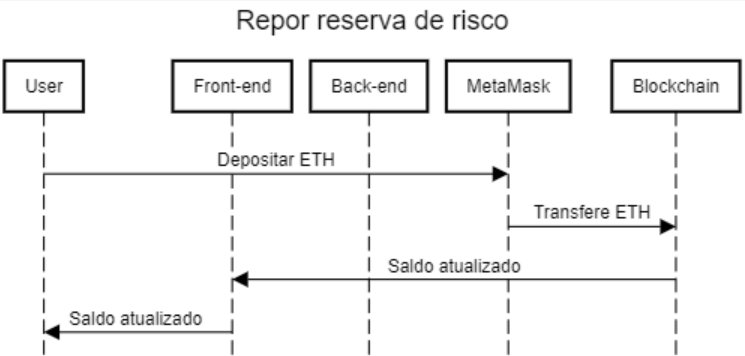


Imagem 25

#### 5.3.3.4. UML Criação do smart contract

Para a criação de smart contract, partindo do princípio que será uma atividade administrativa, a criadora e owner seria a Coover.

O processo de criação começa com a criadora informando como será o seguro, seja de maneira técnica exibindo o código do contrato, como também comercial, informando no front-end valores como taxa administrativa, faixa de preço, e valor percentual mínimo. A entidade “Pré-ativação” se refere à funcionalidade de construir o esqueleto de um contrato antes que ele seja publicado, possibilitando que o cliente se informe dos valores do contrato e a seguradora monitore a quantidade de clientes inseridos/interessados naquele contrato.

Após isso, ela parte para encontrar e atrair usuários para o grupo de seguro, alocando eles para o contrato, conseguindo assim termos a pré ativação do contrato, que é informado ao criador.

Depois desses pré-requisitos, o grupo P2P é confirmado pela seguradora, que ativa o contrato e logo em seguida faz o deploy na rede do ethereum. A fim de executar uma prova de conceito, o deploy será feito na testnet.

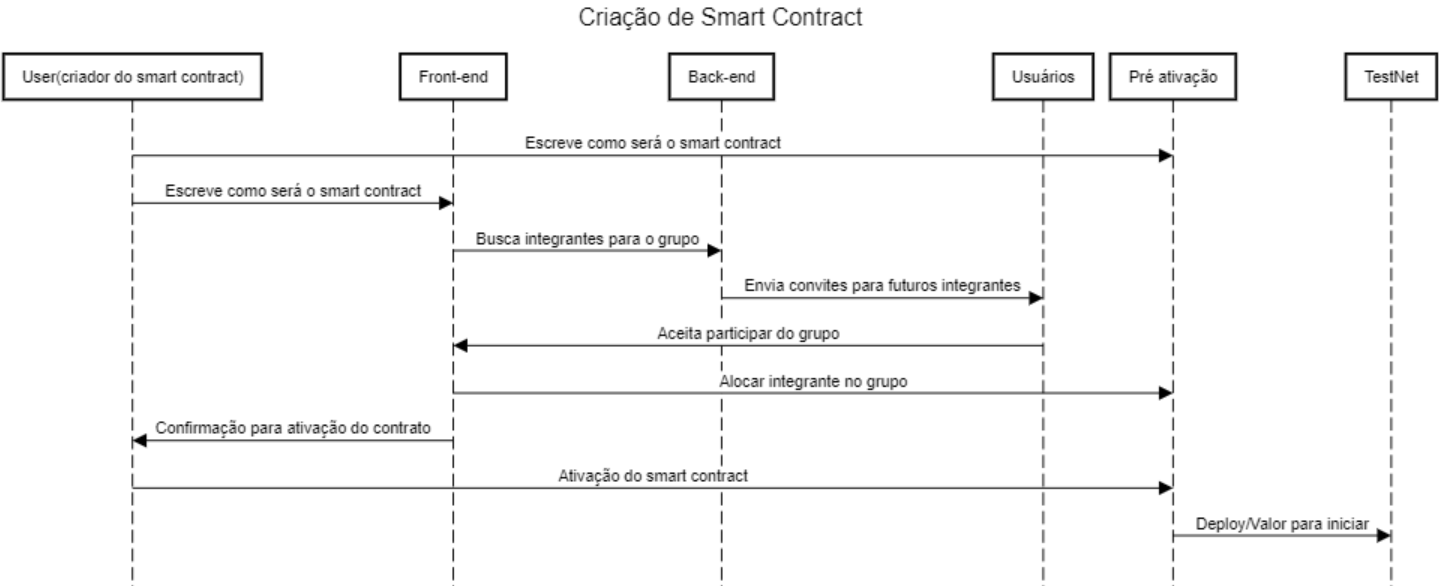


Imagem 26

#### 5.3.3.5. Diagrama UML de implementação

O diagrama de classes de implementação fornece uma visão geral da estrutura do código-fonte do sistema, permitindo uma melhor compreensão da organização das classes e sua interação e ajuda na escalabilidade de projetos. A seguir está o Diagrama de implementação do projeto:

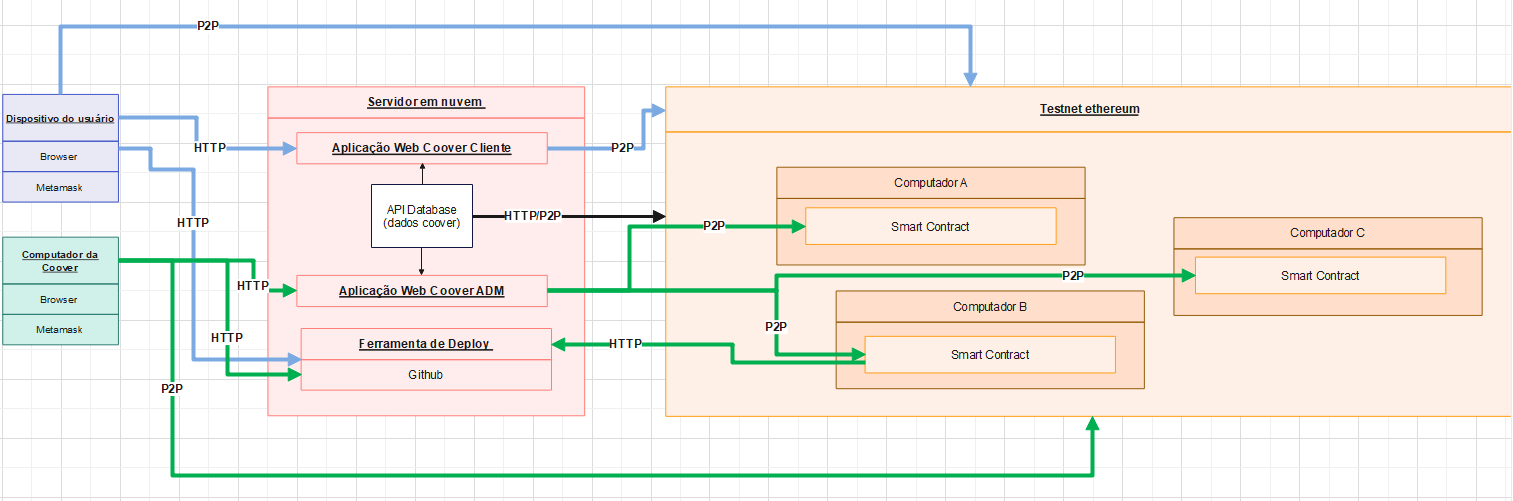


Imagem 27 - Diagrama de implementação.

Através do diagrama acima é possível visualizar a distribuição física do processamento do sistema. Onde é exposto como os hardwares (dispositivo do usuário e da seguradora) se relacionam com o software, em primeiro com a aplicação web (através do protocolo http), na qual está armazenada em um servidor em nuvem. E em segundo, com o smart contract (através do protocolo JSON-RPC, ou seja, de modo P2P), evidenciando como a comunicação é realizada com a Testnet Ethereum.

## 5.3.4. Integração

Na pasta "backend" do projeto, podem ser encontradas subpastas que contêm os arquivos que definem as ABIs do Smart Contract, o arquivo responsável por estabelecer a conexão com o banco de dados MongoDB, o arquivo que utiliza a biblioteca ethers.js para criar e conectar contratos inteligentes na rede Ethereum, além de códigos que exportam dois middlewares para autenticação dos usuários (cliente e administrador) da aplicação web. Também na pasta "backend", são encontrados códigos que definem o modelo de dados da aplicação utilizando o Mongoose, que abrangem informações do grupo mútuo, cadastro de usuários e indenizações. Além disso, há arquivos que criam as rotas da aplicação web, que são utilizadas para definir as respostas conforme as solicitações recebidas. Já na pasta "frontend", encontram-se as rotas que correspondem às páginas de login, logout, perfil do usuário, dashboard do usuário, criação de conta do usuário, pedido de nova indenização, visualização das indenizações, dentre outras.

Para funcionamento total da aplicação são necessários dois arquivos .env, tanto no backen, quanto no frontend. Para o backend é necessário passar as seguintes variáveis: DB\_URL (url do MongoDB), NODE\_ENV = development, JWT\_SECRET, SEGURO\_FACTORY\_ADDRESS, MNEMONIC e INFURA\_API\_KEY. Para o frontend: SEGURO\_FACTORY\_ADDRESS.

### 5.3.4.1. Diagrama UML da integração

Este diagrama UML representa um processo de integração envolvendo a criação de um smart contract por um usuário, a alocação de membros em grupos e a cobrança de uma taxa administrativa. O processo começa com o usuário criando um smart contract e inserindo informações relevantes no front-end usando React e Web3.js. O back-end, que utiliza a biblioteca Ethers, armazena dados sensíveis. O usuário cliente é convidado para se juntar ao smart contract e aceita o convite. Uma pop-up aparece para confirmar a pré-ativação do usuário, que então confirma. Em seguida, o front-end aloca os membros em grupos dentro do smart contract. Uma nova pop-up aparece para informar o usuário que ele deve pagar uma taxa administrativa. O usuário revisa os dados da transação da taxa, insere a quantidade necessária de ETH e confirma a transferência. O front-end, então, transfere a taxa para o smart contract. Uma nova pop-up informa o criador do smart contract que a taxa foi paga com sucesso. Em seguida, o usuário deve confirmar a conexão do MetaMask ao smart contract. Uma pop-up aparece para informar que a conexão foi estabelecida com sucesso. Por fim, o criador do smart contract ativa o smart contract.

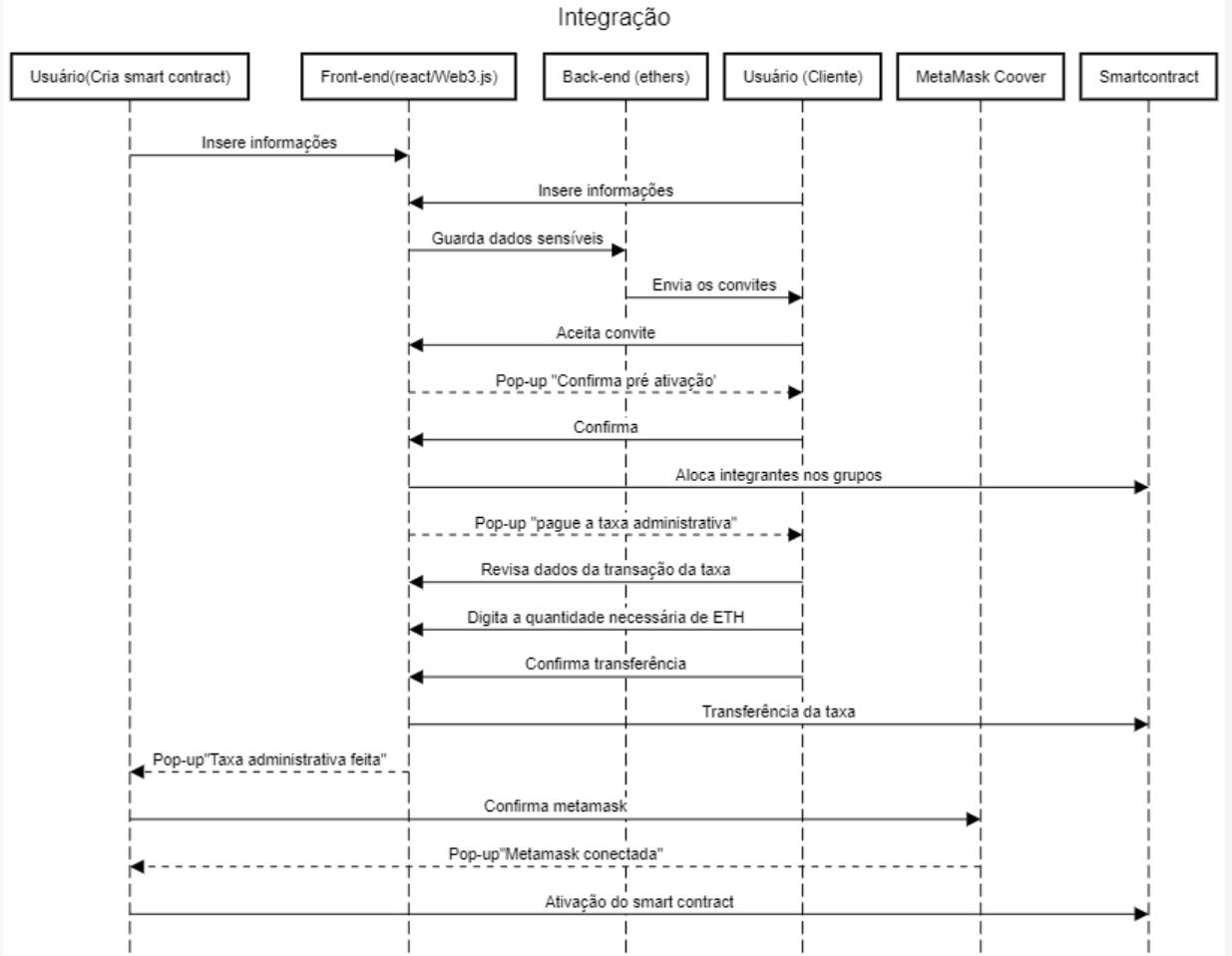
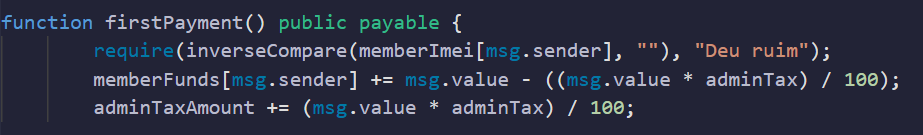


Imagem 28 - Diagrama de integração.

## 5.4. Algoritmos e Equações

A solução desenvolvida engloba distintas equações matemáticas, visto que envolvem diversas transações entre seguradora e cliente:

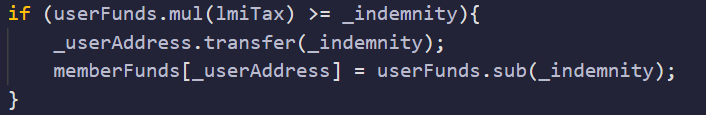
1. Primeiro pagamento do cliente que ativa seu estado no contrato:

## 

## 

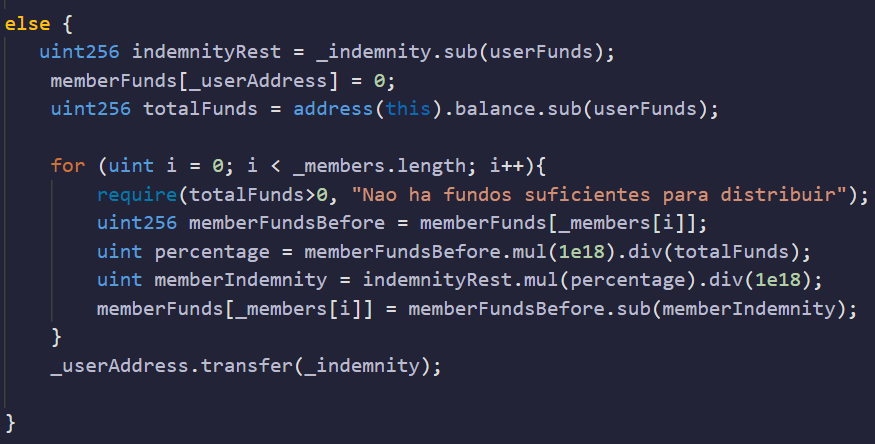
## O fundo do membro, por instância, é igual a zero. Quando realizado o pagamento, adiciona-se o valor ao fundo, retirando a porcentagem da taxa administrativa desse valor enviado pelo cliente.

Também adiciona-se esse valor retirado de taxa administrativa numa variável que guarda todas as taxas administrativas.

1. Pagamento de indenização:

Primeiramente, o fundo do usuário a ser indenizado é multiplicado pela taxa de Limite Máximo Indenizável e o resultado desse valor deve ser maior ou igual ao de indenização. Caso seja, o valor de indenização é retirado do fundo do usuário, caso não seja, a função continua. Abaixo seguem as equações:

* fundo do usuário \* taxa LMI >= indenização;
* fundo do usuário = fundo do usuário - indenização.

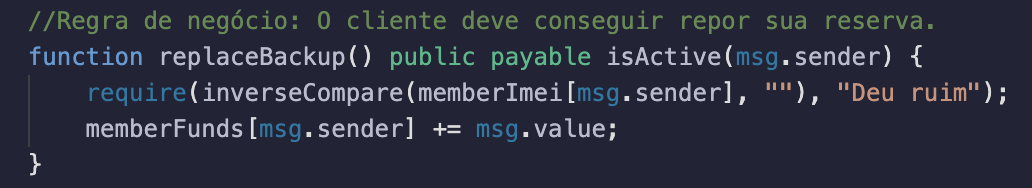


## Caso o valor de reserva do cliente indenizado seja menor ao da indenização, cria-se uma variável (indemnityRest), que refere-se ao valor da indenização menos ao de fundos do usuário (zerando sua reserva). Também é estabelecida uma variável (totalFunds), que refere-se a todo o valor de reserva do contrato menos ao de reserva do cliente indenizável (esse valor será utilizado para verificar se ainda restam fundos no contrato).

A partir disso, é calculada a porcentagem de participação de cada membro no pagamento do resto da indenização, dividindo os fundos do usuário pelo total de fundos (lembrando de que esse totalFunds já subtraiu as reservas do usuário indenizado). A seguir, é calculado o valor a ser pago pelo usuário, multiplicando o valor restante de indenização pela porcentagem. Por fim é retirado esse valor do fundo do usuário e é feita a transferência da indenização. Abaixo seguem as equações:

* restante da indenização = indenização total - reserva do indenizado;
* fundos totais = fundo do contrato - reserva do indenizado;
* porcentagem = fundo do usuário / fundos totais;
* participação da indenização = restante da indenização \* porcentagem;
* fundo do usuário = fundo do usuário - participação da indenização.

1. Reposição da reserva de risco



Esta função serve para repor a reserva de risco de algum indivíduo, primeiramente ela faz o requerimento de que exista um Imei existente da pessoa que está repondo, após isso, a partir do mapping “ memberFunds” conseguimos alocar o dinheiro para o indivíduo.

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

O projeto aborda uma tecnologia emergente e, por isso, apresenta riscos de ataques, tanto no quesito de contrato envolvendo a rede blockchain, quanto à plataforma web. Dessa forma, os requisitos abaixo analisados foram “Confidencialidade”, visto que todo código na blockchain é visto por qualquer pessoa no mundo, é importante garantir que os dados sensíveis não sejam vistos por pessoas não autorizadas, e “Disponibilidade”, para que o sistema possa ser acessado e utilizado a qualquer momento por clientes da Coover, por exemplo, para repor reserva de risco ou solicitar uma indenização a qualquer momento.

Dessa forma, os atacantes e possíveis ataques estão listados abaixo:

1. Confidencialidade:

No requisito de confidencialidade, foram identificados 2 principais atacantes: Atacantes financeiros ou Atacantes de Informações, de acordo com os possíveis riscos de ataques do sistema. Portanto, foram identificados ataques de phishing, caso dados privados de clientes sejam expostos, assim como ataques de engenharia social, seguindo os mesmos critérios.

1. Disponibilidade:

Neste requisito é importante que o sistema fique ativo aos seus usuários legítimos, para que possam aceitar convites, pedir indenizações, repor reservas, visualizar seu saldo e receber indenizações. Portanto, se um atacante controlar a maioria de nós na rede pode bloquear ações que envolvam a blockchain, sendo elas: pedir indenizações, repor reservas, visualizar seu saldo e receber indenizações.

A tabela abaixo apresenta de forma simplificada os ataques, probabilidade de acontecer, o impacto aos usuários e à Coover e o seu risco, sendo o último medido de acordo com a probabilidade e o impacto.

| Ataque | Probabilidade | Impacto | Risco |
| --- | --- | --- | --- |
| Ataque de confidencialidade | Médio(muitas pessoas podem acabar caindo em golpes como phishing). | Médio(isso faria com que as pessoas perdessem dinheiro porem nao teria um impacto tão grande para o negócio). | Médio |
| Ataque de disponibilidade | Baixo (é difícil alguém controlar a maior parte da rede). | Alto(faria um estrago para o negócio caso um tipo de ataque como esse acontecesse). | Baixo |
| Ataque de autenticidade | Baixo(não é comum que algum atacante possa influenciar na votação a partir de identificadores falsos). | Alto(um golpe como esse poderia destruir um negócio). | Médio |

Por fim, foi realizada uma modelagem de atacantes visando a motivação de tais ataques por eles:

* Atacantes financeiros: motivados pelo lucro, esses atacantes estão interessados em roubar fundos ou manipular as transações para seu próprio benefício financeiro.
* Atacantes de informação: motivados pelo acesso a informações confidenciais, esses atacantes estão interessados em controlar ou obter acesso a informações sensíveis.

# 6. Desenvolvimento e Resultados

## 6.1. Solidity

Foram utilizadas diferentes tecnologias e ferramentas para o desenvolvimento dos contratos e de deploy em uma testnet pública. Para a construção dos contratos e testes das transações, foi desenvolvido em Solidity, na IDE do Remix é utilizada a biblioteca SafeMath do Openzeppelin para o funcionamento correto das funções matemáticas. Para o deploy, foi utilizao o suite de desenvolvimento Truffle, a tesnet pública da Goerli e a Metamask para o pagamento do deploy.

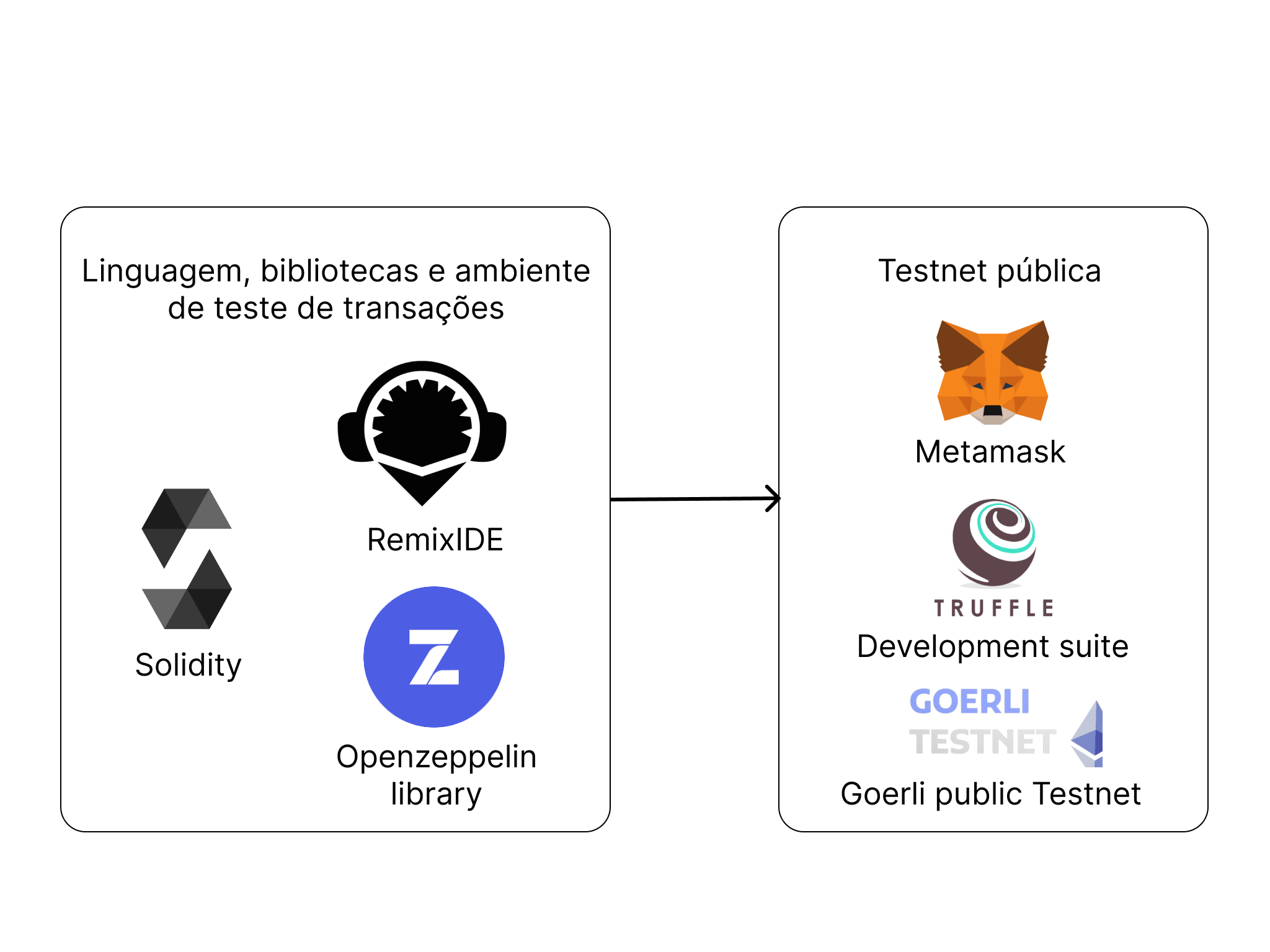


Imagem 29

## 6.2. Tecnologia adotada

Em nossa solução utilizamos as seguintes tecnologias e ferramentas:

* Wireframe/Prototipação/Mockup -> Figma, Canva;
* Front-end->React e bibliotecas “next.js” e “ ethers.js”;
* Back-end->Mongodb, Node js com Express, biblioteca “ethers”;
* Carteira-> Wallet Metamask;
* Smart contract-> RemixIDE, Solidity, biblioteca"Open Zeppelin Library(SafeMath)"
* Deploy-> Truffle e Sepolia Testnet.;

## 6.3. Solução interface Adm Coover

### 6.3.1. User stories

As user stories utilizadas nesse módulo são (referente ao tópico 4.2.2 do documento):

* User storie 1 ->Eu, como agente da seguradora, quero ter uma interface que permita a criação de cláusulas de smart contracts para definir as regras de organização dos grupos.
* User storie 2 -> Eu, como agente da seguradora, quero ter como publicar o contrato a fim de torná-lo válido e funcional.
* User storie 3 -> Eu, como agente da seguradora, quero ter como visualizar as solicitações de indenização realizadas pelos participantes a fim de avaliar o caso.
* User storie 4-> Eu, como agente da seguradora, desejo ter uma interface para, de acordo com a veracidade dos pedidos de indenização, autorizar ou não a remuneração dos respectivos indivíduos da solicitação.

### 6.3.2. Prototipação

Os frames relacionadas a interface Administrador da Coover são (referente ao tópico 4.4.2 do documento):

* Figura 17 - Jornada Login Administrador
* Figura 18 - Jornada Novo Contrato
* Figura 19 - Jornada Carteira Coover
* Figura 20 - Jornada Visualizar Grupos
* Figura 21 - Jornada Solicitações

## 6.4. Solução interface Cliente Coover

### 6.4.1. User stories

As user stories utilizadas nesse módulo são (referente ao tópico 4.2.2 do documento):

* User storie 5 ->Eu, como futuro participante, quero fornecer as informações do meu celular a fim de possibilitar um futuro convite a um smart contract de seguro mútuo.
* User storie 6 -> Eu, como participante, quero ter uma interface que permita aceitar um convite de entrada de adesão a um smart contract para fazer parte de um grupo de seguro mútuo peer-to-peer.
* User storie 7 -> Eu, como participante, quero ter uma interface que permita fazer um aporte no smart contract para pagar a taxa administrativa e manter ao menos o percentual mínimo da reserva de risco.
* User storie 8-> Eu, como participante, quero ter uma interface que permita solicitar a indenização do seguro celular para receber o valor de cobertura do smart contract.

### 6.4.1. Prototipação

Os frames relacionadas a interface Administrador da Coover são (referente ao tópico 4.4.2 do documento):

* Imagem 13 - Jornada de entrada
* Imagem 14 - Jornada Convite
* Imagem 15 - Jornada Pagamento
* Imagem 16 - Jornada pedido de indenização

## 6.2. Deploy

Para a realização do deploy seguimos o tutorial do [Inteli Blockchain](https://github.com/InteliBlockchain/InteliBlockchain/tree/main/tutoriais/Truffle%20-%20Deploy%20e%20compila%C3%A7%C3%A3o%20de%20contrato%20em%20Solidity) . Utilizamos a testnet pública (Sepolia), Development Suite Truffle e ETH da Faucet Sepolia. A Sepolia é uma rede de testes da Ethereum que usa Proof of Authority (PoA) ao invés de Proof of Work (PoW), permitindo que desenvolvedores possam testar contratos inteligentes sem gastar muito tempo e dinheiro validando transações. Além disso, a Sepolia disponibiliza uma faucet que fornece ETH de teste gratuitamente, tornando-a uma opção segura e econômica para testes antes de implementar na rede principal. Por outro lado, o Truffle é uma suíte de ferramentas para desenvolvimento de contratos inteligentes na Ethereum que auxilia os desenvolvedores a escrever, testar, implantar e gerenciar esses contratos de maneira mais simples e eficiente.

Esses foram os passos resumidamente:

1- Ter NodeJs, Git e VSCode;

2- Instalar e iniciar o Truffle no Terminal CMD;

3- Criar o contrato em Solidity na pasta "contracts";

4- Criar o arquivo que fará o deploy do contrato na pasta "migrations" (o arquivo deve ser .js);

5- Criar o arquivo "package.json" e baixar bibliotecas que o contrato esteja usando no CMD;

6- Editar as informações do contrato que deseja realizar o deploy no arquivo criado em "migrations", informando os parâmetros para deploy e o nome do contrato (não do arquivo). No nosso caso, como temos o contrato Factory que realiza o deploy dos demais contratos de seguro mútuo, não é necessário passar nenhum outro argumento para realizar o deploy;

7- Selecionar a rede no arquivo "truffle-config.js" (a rede utilizada foi Sepolia, então as linhas 93 a 99 foram descomentadas);

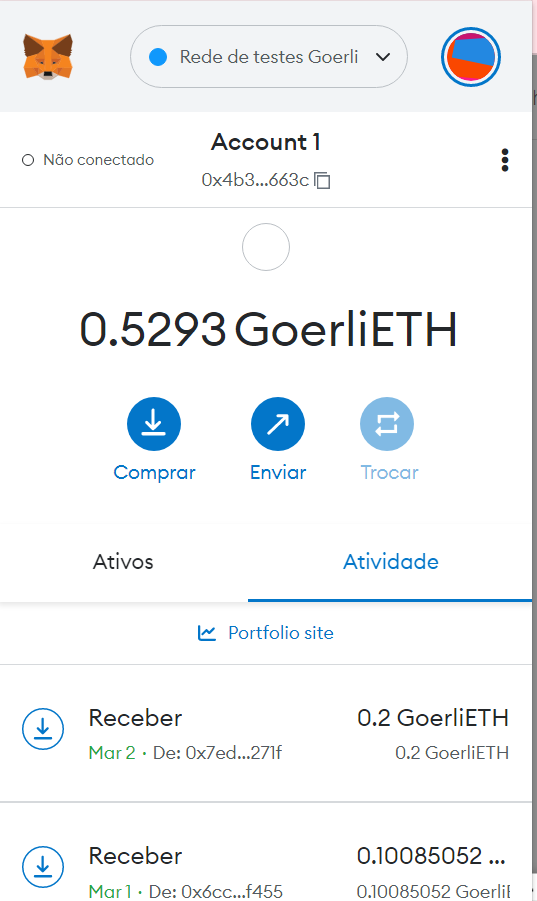
8-Instalar e criar o arquivo .env (descomentando as linhas 44 a 47 do arquivo "truffle-config.js") e adicioná-lo ao arquivo ".gitignore";

9- Adicionar a seed prhase da carteira Metamask que será o owner do contrato e a api key do Infura no arquivo ".env", além de alterar a variável, agora criada, nas linha 45 do "truffle-config.js";

10- Executar o comando "truffle deploy --network sepolia" no CMD (garanta que a carteira Metamask tenha fundos GOERLI para pagar o gas do contrato).

Como seguem as imagens abaixo, esses são os resultados do deploy:





## 6.5. Avaliação

### 6.5.1. Testes Automatizados

Os testes automatizados das principais funções de um contrato inteligente são essenciais para garantir sua segurança e confiabilidade.

No contexto da criação de um seguro P2P da Coover, essa importância é ainda mais relevante, já que a natureza descentralizada do processo exige uma validação rigorosa.

Neste estudo, utilizamos o Remix IDE para criar testes automatizados das principais funções do smart contract, a fim de garantir que ele funcione corretamente e que os usuários possam confiar plenamente em sua utilização.

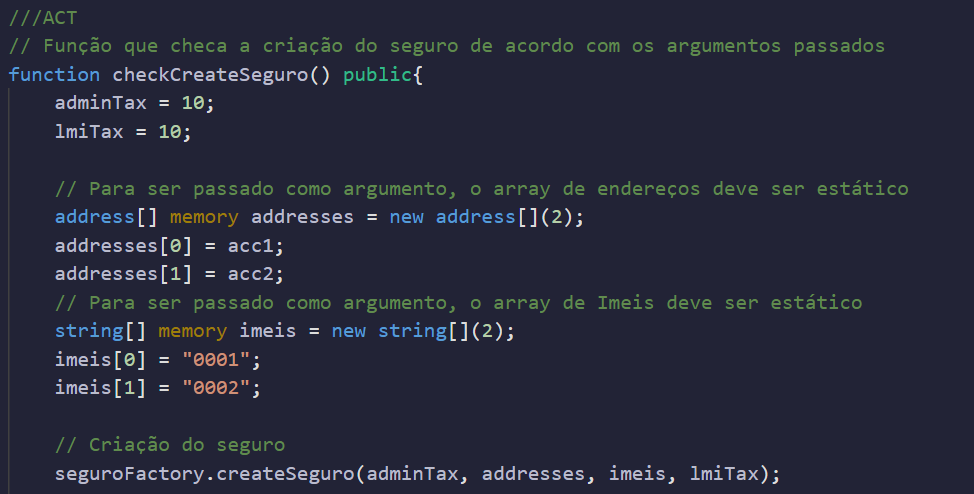
...

Um dos testes foi o arquivo ***“seguroFactory\_test.sol”*** que foi destinado ao contrato factory do seguro. O contrato factory instancia o contrato do seguro, então testamos se a criação do seguro ocorreu como desejado. Ele inicializa as pré-condições que são: valor percentual mínimo = 10 e Taxa Administrativa = 10. Depois executa a função "**checkCreateSeguro()**", que cria um novo seguro com alguns argumentos.

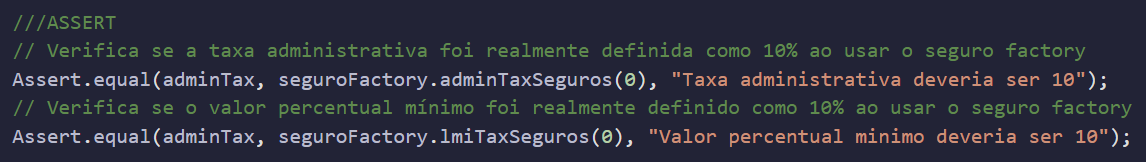
* **Pré-condições:**
  + Contas:
    - C01 -> 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb
    - C02 -> 0x4B20993Bc481177ec7E8f571ceCaE8A9e22C02db
  + Dono do contrato -> Deployer do seguro factory (msg.sender)
  + Taxa administrativa -> 10;
  + Valor percentual mínimo -> 10
  + Membros -> C01, C02

Imeis -> 0001, 0002

* **Procedimento de teste:**

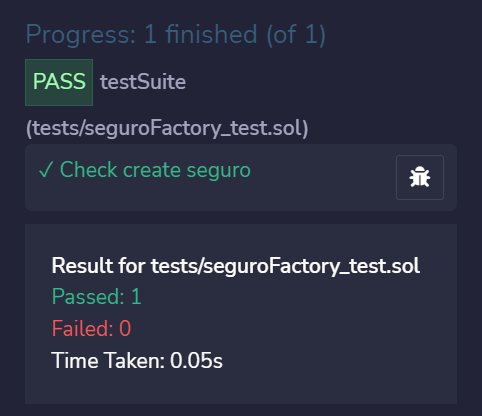
Depois de definidos todos as pré-condições e inputs, é instanciado o contrato do seguro:  


Depois verifica se a taxa administrativa e o valor mínimo percentual foram definidos corretamente usando a função "**Assert.equal()**". O objetivo do teste é garantir que a criação de novos seguros esteja funcionando corretamente através do seguro Factory.



* **Resultado esperado:**
  + Taxa administrativa -> 10;
  + Valor percentual mínimo -> 10
* **Pós condições:**
  + Contrato **Seguro Sprint 4** criado.
  + Contrato **Seguro Sprint 4** não criado.

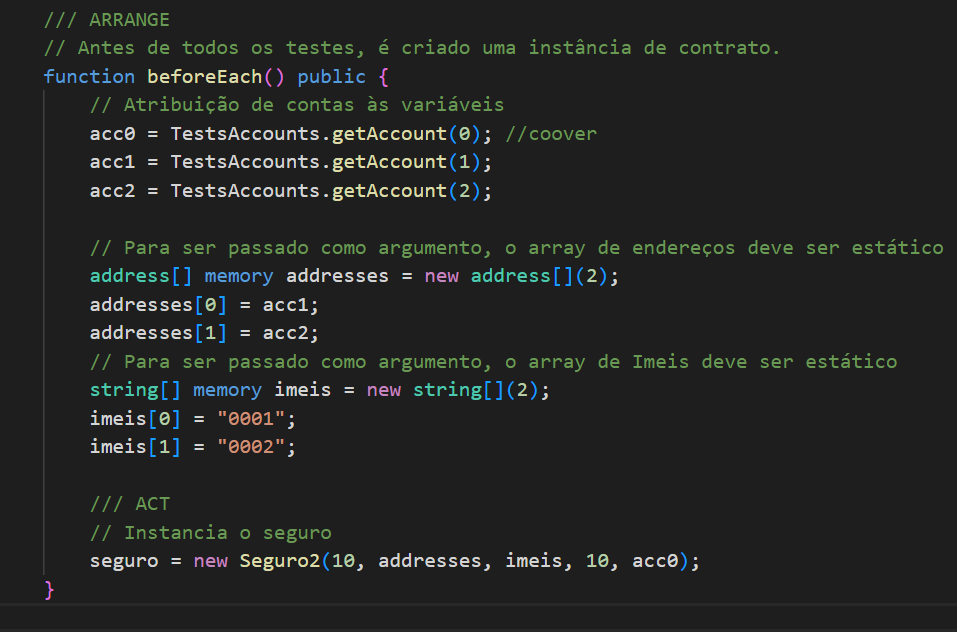
Executando, temos que o contrato foi criado de maneira correta, com o Valor percentual mínimo = 10 e Taxa administrativa = 10.



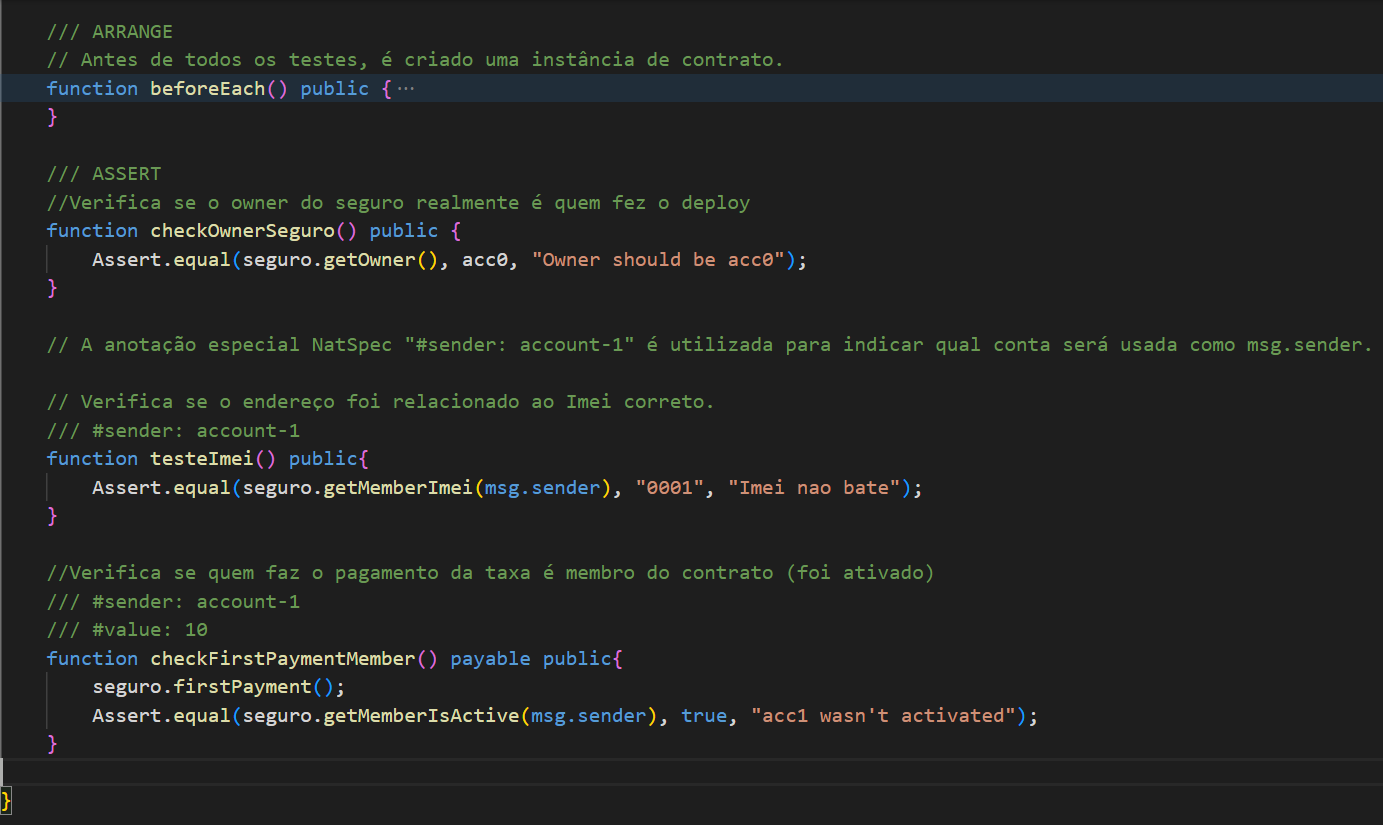
O outro arquivo de teste ***“seguroSprint4\_test.sol”*** foi direcionado ao smart contract do seguro. Nesse arquivo, testamos de forma automatizada se os argumentos passados no requisito de negócio de criação do seguro foram realmente atribuídos ao seguro.

* **Pré-condições:**
  + Contas:
    - C00 -> 0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
    - C01 -> 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb
    - C02 -> 0x4B20993Bc481177ec7E8f571ceCaE8A9e22C02db
  + Dono do contrato -> C00
  + Taxa administrativa -> 10;
  + Valor percentual mínimo -> 10
  + Membros -> C01, C02
  + Imeis -> 0001, 0002
* **Procedimento de teste:**

Depois de definidos todos as pré-condições e inputs, é instanciado o contrato do seguro:



Logo em seguida são feitos os seguintes testes:

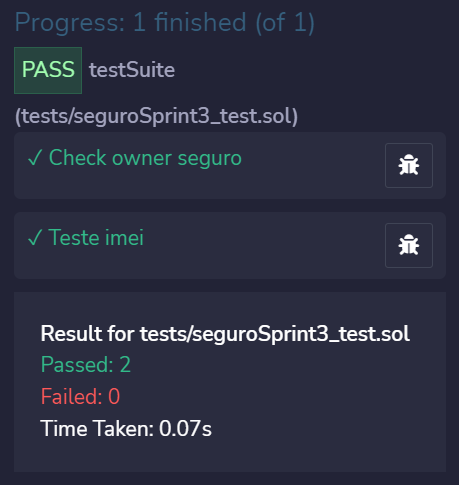


* **Resultado esperado:**
  + Dono do contrato -> C00
  + Taxa administrativa -> 10;
  + Valor percentual mínimo -> 10
  + Membros -> C01, C02
  + Imeis -> 0001, 0002
  + Relação Membro-Imei:

|  | C01 | C02 |
| --- | --- | --- |
| IMEI | 0001 | 0002 |

* **Pós condições:**
  + Contrato **Seguro Sprint 4** criado.
  + Contrato **Seguro Sprint 4** não criado.

Executando, temos que o contrato foi criado de maneira correta, com o Valor percentual mínimo = 10, Taxa administrativa = 10, Owner = C00 e Imei da C01 = 0001.



# 7. Conclusões e Recomendações

Em resumo, a solução desenvolvida foi planejada considerando as limitações que a utilização da Blockchain pode trazer em uma aplicação no mundo real, sendo a principal delas o custo (gas fee) para a realização dessas transações e, dessa forma, utiliza uma solução híbrida com Web 2.0, em que partes do fluxo da aplicação ocorrem fora da rede, o que ainda garante segurança e praticidade das transações de um contrato, mas fornece uma experiência mais fluida e, principalmente, escalável a longo prazo.

# Anexos

Capturas de tela das ferramentas utilizadas no deploy em Truffle do contrato:

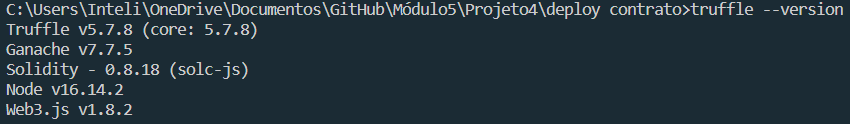


Imagem - Truffle version

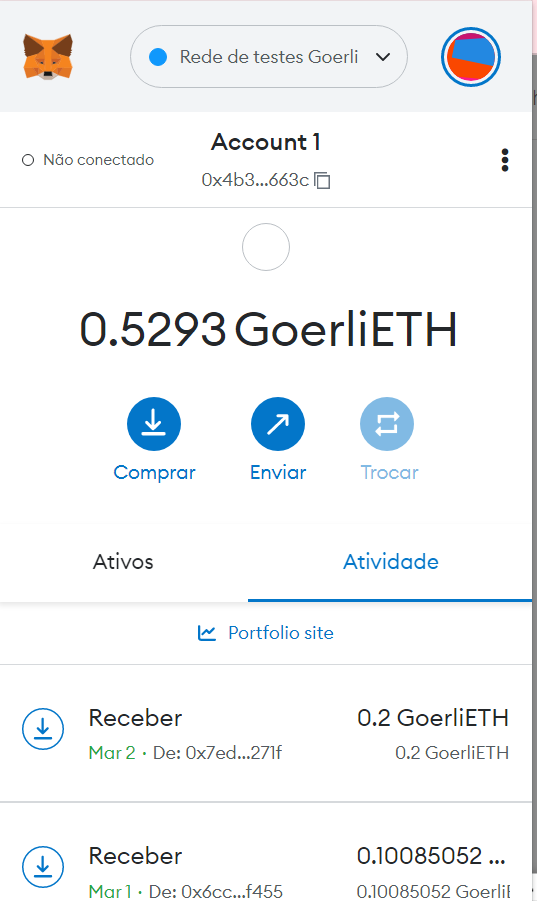


Imagem - Wallet Metamask

Utilize esta seção para anexar materiais como manuais de usuário, documentos complementares que ficaram grandes e não couberam no corpo do texto etc.

***Sugestão:***

*Documentos que são alterados por cada sprint, como a Matriz de Riscos, devem ser movidas para a seção de anexo.*

*No corpo do documento deve permanecer o documento atual.*

*Separar os documentos por sprints.*

***Sugestão de divisão da seção Anexo:***

## ANEXO I – Sprint 1

Mova para essa seção os documentos produzidos na Sprint 1 que sofreram alterações na Sprint 2.

## ANEXO II – Sprint 2

Mova para essa seção os documentos produzidos na Sprint 2 que sofreram alterações na Sprint 3.

## ANEXO III – Sprint 3

Mova para essa seção os documentos produzidos na Sprint 3 que sofreram alterações na Sprint 4.

## ANEXO IV – Sprint 4

Mova para essa seção os documentos produzidos na Sprint 4 que sofreram alterações na Sprint 5.