**Documento de Especificação de Requisitos**

Time: CPM 22 Bugs  
  
Gabriel Camilo Silva Oliveira  
Gustavo Morais Medeiros  
Jonathan Felipe Viana Cardoso Prado  
Robert Cardoso Cantares Martins  
Wendel Márcio de Oliveira Filho  


## **Requisitos Funcionais**

**Cadastro de Usuários**

* RF001 - Usuários devem ser capazes de realizar seu cadastro pessoal na plataforma como “Hoteleiro” ou “Hóspede” ao preencher os campos com nome, e-mail, senha e CPF.
* RF002 - Usuários devem ser capazes de alterar as informações cadastrais (exceto o CPF).
* RF003 - Os hoteleiros devem ser capazes de disponibilizar quartos, que por sua vez podem ter suas informações consultadas por outros usuários.

**Cadastro de Hotéis e Quartos**

* RF004 - Caso esteja logado no sistema como “Hoteleiro”, o usuário deve ser capaz de cadastrar um hotel. Para isso, deve informar o nome, endereço, descrição, imagens, informações de contato, pontos turísticos e atrações próximas, comodidades, tipos de quartos, regras e políticas da propriedade e eventuais pormenores.
* RF005 - Caso esteja logado no sistema como “Hoteleiro” e possuir ao menos 1 hotel cadastrado, o usuário deve ser capaz de cadastrar um quarto. Para isso, deve informar o tipo do quarto, número de hóspedes que comporta, preço por noite, preço total da reserva (ao selecionar datas de check-in e check-out), disponibilidade, serviços, comodidades e imagens.

**Consultar e alterar informações**

* RF006 - Ao acessar as informações de um quarto, os usuários devem ser capazes de verificar a disponibilidade do quarto em um intervalo de datas e realizar a reserva caso esteja disponível.
* RF007 - Usuários devem conseguir cancelar as reservas que, de acordo com a data de cancelamento e políticas de cancelamento do hotel, resultarão em reembolso parcial ou completo.
* RF008 - Hoteleiros devem ser capazes de alterar informações dos quartos cadastrados, neste caso, usuários com reservas ativas devem receber uma notificação alertando sobre as mudanças.
* RF009 - Hoteleiros devem ser capazes de deletar quartos, caso existam hóspedes com reservas ativas, este processo resultará no reembolso completo.

**Contratos Inteligentes**

* RF010 - Cada reserva deve ser representada por um contrato inteligente individual a fim de proporcionar maior flexibilidade e controle sobre cada reserva.
* RF011 - Haverá um contrato principal responsável por gerenciar todas as reservas. Ele pode conter funções para criar novas reservas, editar reservas existentes, cancelar reservas, lidar com a lógica de reembolso e obter informações sobre as reservas. Além disso, ele deve conter a lógica para manipular os pagamentos e reembolsos. Este contrato será responsável por criar novos contratos de reserva individual.
* RF012 - Cada reserva feita pelos clientes será representada por um contrato inteligente individual. Este contrato conterá informações específicas da reserva, como datas, tipo de quarto, preço e estado da reserva. Ele também deve conter a lógica para manipular eventos relacionados à reserva, como cancelamento ou edição.
* RF013 - Os contratos inteligentes podem interagir diretamente com as carteiras digitais dos clientes e do hotel para processar pagamentos e reembolsos. Os clientes podem autorizar transações diretamente de sua carteira digital para reservar quartos, e o hotel pode processar reembolsos diretamente para a carteira digital dos clientes.
* RF014 - Uma vez que as informações armazenadas em uma rede distribuída Blockchain sejam imutáveis, os contratos individuais de reserva devem possuir métodos que permitam aos clientes revisar e confirmar os detalhes de sua reserva antes de confirmá-la definitivamente na Blockchain.

## **Requisitos Não-Funcionais Segurança**

* RNF001 - Os contratos inteligentes devem ser implementados seguindo práticas de segurança para proteger os fundos e informações dos usuários contra ataques e vulnerabilidades.
* RNF002 - Deve ser implementado um sistema de autenticação robusto para garantir que apenas usuários autorizados possam acessar e modificar as reservas.
* RNF003 - As informações sensíveis dos usuários (como dados de pagamento e informações pessoais) durante a realização das reservas deve ser mantido em criptografia após a transação do contrato ser validada. Dessa forma, a assinatura deve ser verificável por meio da chave pública do usuário e consultável por meio de sua chave privada.

**Eficiência**

* RNF004 - O código dos contratos inteligentes deve ser otimizado para minimizar o consumo de recursos computacionais e garantir uma boa experiência do usuário.
* RNF005 - O sistema deve ser capaz de lidar com um grande volume de transações sem comprometer o desempenho.

**Disponibilidade**

* RNF006 - Uma vez que está diretamente conectado com a rede Blockchain da Ethereum (MainNet), a aplicação deve estar disponível para os usuários durante a maior parte do tempo, com tempo de inatividade mínimo planejado para manutenção e atualizações.
* RNF007 - Deve ser implementado um sistema de monitoramento para detectar e responder rapidamente a falhas e interrupções no sistema.

**Escalabilidade**

* RNF008 - O sistema deve ser escalável para lidar com um aumento no número de usuários e transações sem comprometer o desempenho.
* RNF009 - Deve ser possível adicionar recursos adicionais, como mais contratos de reserva ou servidores de aplicativos, conforme necessário para atender à demanda crescente.

**Usabilidade**

* RNF010 - As mensagens de erro e feedback devem ser claras e informativas para ajudar os usuários a entender e resolver problemas facilmente.
* RNF011 - A interface do usuário deve ser intuitiva mesmo para usuários inexperientes em blockchain e estar de acordo com as melhores práticas de UX Design, como as diretrizes de Nielsen.

**Compatibilidade**

* RNF012 - Os usuários deverão possuir uma conta Metamask (carteira virtual de criptomoedas) para conectar-se à rede Ethereum.
* RNF013 - A aplicação deve estar disponível em qualquer navegador que suporte o plugin da carteira Metamask.