
DESENVOLVIMENTO DE *MARKETPLACE* DE SERVIÇOS *FREELANCERS* ESPECIALIZADOS

DO COUTO DIAS RAMOS, Caio Felipe
FERREIRA DA MOTA, Rafael
NEGRETTO, Diego Henrique

Centro Universitário Hermínio Ometto – FHO, Araras – SP, Brasil

1. Estratégia de Testes

Testes Estruturais

Objetivo: Verificar a estrutura interna do sistema, validando a lógica dos algoritmos, funções, conexões com banco de dados e contratos inteligentes.

Ferramentas: Jest (Node.js), Hardhat (Smart Contracts), MySQL Workbench.

Testes Funcionais

Objetivo: Garantir que cada funcionalidade do sistema cumpra os requisitos especificados.

Método: Testes simulando ações do usuário (login, cadastro, busca de projetos, envio de propostas, deploy de smart contracts).

Ferramentas: Cypress, Postman (para testes de API).

2. Estratégia de Testes Específicos

Teste de Integração

Objetivo: Verificar a interação entre os módulos do sistema (frontend, backend, banco de dados e blockchain).

Exemplos: Validação do envio de proposta do frontend até o registro no smart contract.

Integração da carteira digital com o Ethereum.

Teste de Sistema

Objetivo: Validar o sistema como um todo, simulando o comportamento real de usuários e fluxos de navegação.

Exemplos: Publicar um projeto, receber candidaturas, gerar smart contract, confirmar pagamento.

Teste de Aceitação

Objetivo: Verificar se o sistema atende às expectativas do usuário final.

Método: Feedback de usuários-teste (freelancers e clientes simulados).

3. Plano de Testes – Requisitos Funcionais

Requisito	Caso de Teste	Entrada	Resultado Esperado	Critério de Aceitação
RF01 - Cadastro de usuário	Preencher formulário	Nome, Email, senha	Conta criada com sucesso	Conta registrada e redirecionamento para login
RF02 - Publicação de projeto	Preencher título, descrição e habilidades	Dados válidos	Projeto listado na plataforma	Projeto visível na área de projetos
RF03 - Proposta de freelancer	Clicar 'Enviar Proposta'	Texto da proposta	Proposta vinculada ao projeto	Cliente visualiza proposta
RF04 - Geração de Smart Contract	Clicar 'Criar contrato'	Dados do projeto e aceite de ambas as partes	Contrato criado e registrado em blockchain	Hash do contrato exibido

RF05 - Login	Informar email e senha	Dados válidos	Redirecionamento para painel	Sessão iniciada com token
-----------------	------------------------------	------------------	---------------------------------	------------------------------

4. Plano de Testes – Requisitos Não-Funcionais

Requisito	Tipo	Plano de Teste	Critério de Aceitação
RNF01 - Segurança	Confidencialidade	Testar criptografia de senha e HTTPS	Dados protegidos com HTTPS e bcrypt
RNF02 - Desempenho	Tempo de resposta	Testar tempo de carregamento de páginas principais	< 2 segundos em 90% das requisições
RNF03 - Usabilidade	Navegação	Testar clareza dos menus, feedback de ações	Interface intuitiva e aprovada por usuários-teste
RNF04 - Confiabilidade	Smart Contracts	Simular diferentes cenários	Execução automática sem falhas ou erros
RNF05 - Compatibilidade	Multiplataforma	Acessar via Chrome, Firefox e mobile	Layout responsivo e funcionamento igualitário