

# Crafting Stable: Project Report

Trabalho realizado por:

*Filipe Sousa [114196]*

*Gonçalo Calvo [120131]*

*Daniel Simbe [110235]*

## Table of Contents

<b>Crafting Stable: Project Report</b>	<b>1</b>
1 Introdução	3
1.1 Visão geral do projeto	3
2 Conceito do Produto	4
2.1 Personas	4
Persona 1 – Proprietária de Ferramentas	4
Persona 2 – Utilizador / Renter	4
Persona 3 – Administradora da Plataforma	5
2.2 Scenarios	5
Cenário 1 – Proprietária adiciona uma ferramenta	5
Cenário 2 – Arrendatário pesquisa e reserva uma ferramenta	5
Cenário 3 – Pagamento e devolução da ferramenta	5
Cenário 4 – Administradora monitoriza a plataforma	6
2.3 Epics	6
1 – Procura e Descoberta de Ferramentas	6
2 – Reserva e Agendamento	6
3 – Pagamentos e Transações	6
4 – Gestão de Ferramentas	7
5 – Perfis de Utilizador e Histórico	7
6 – Gestão da Plataforma e Monitorização	7
3 Domain Model	8
4 Arquitetura	9
4.1 Requisitos Funcionais e Não-Funcionais	9
4.1.1 Requisitos Funcionais	9
4.1.2 Requisitos Não-Funcionais	10
4.2 Vista da arquitetura	11
5 API para Desenvolvedores	12
6 Referências e Recursos	13

# 1 Introdução

## 1.1 Visão geral do projeto

O **Crafting Stable** é uma plataforma digital que visa promover a **economia de partilha** através do aluguer de ferramentas entre utilizadores.

Num contexto em que muitas ferramentas são adquiridas para utilizações pontuais e permanecem subutilizadas, o acesso a equipamentos adequados continua a ser dispendioso e pouco eficiente.

Atualmente, quem necessita de realizar pequenos trabalhos domésticos, reparações ou projetos de bricolagem enfrenta frequentemente a necessidade de comprar ferramentas caras para um uso ocasional. Por outro lado, muitos utilizadores possuem ferramentas que raramente utilizam.

O **Crafting Stable** surge para resolver este problema, oferecendo uma solução **simples, acessível e segura**, baseada num modelo *peer-to-peer*, que liga proprietários de ferramentas a utilizadores que delas necessitam.

Posto isto, será desenvolvida uma **aplicação web** que permita:

- **Aos utilizadores comuns (renters):**

- Procurar ferramentas disponíveis por categoria e localização;
- Reservar ferramentas por períodos específicos;
- Efetuar pagamentos de forma segura;
- Consultar históricos de aluguer.

- **Aos proprietários de ferramentas (owners):**

- Publicar e gerir anúncios de ferramentas;
- Definir preços e disponibilidade;
- Aceitar ou recusar pedidos de aluguer;

O desenvolvimento do projeto será orientado por **personas, cenários e epics**, derivados de *user stories*, garantindo uma abordagem centrada no utilizador. Adicionalmente, serão integradas práticas de **qualidade de software**, nomeadamente **Continuous Integration (CI)** e **Continuous Delivery (CD)**.

## 2 Conceito do Produto

Para garantir que a aplicação responde às necessidades reais dos seus utilizadores, foram definidas **personas** representativas dos principais perfis da plataforma: utilizadores que alugam ferramentas e utilizadores que as disponibilizam.

A análise dos objetivos, motivações e contextos associados a cada persona permitiu a construção de **cenários de utilização**, que ajudaram a identificar os principais fluxos funcionais do sistema.

Com base nesses cenários, foram extraídas **user stories**, posteriormente agrupadas em **epics**, servindo como base para o planeamento e desenvolvimento estruturado do produto.

O **Crafting Stable** distingue-se ainda pela sua **arquitetura flexível e extensível**, preparada para integração com serviços externos, como **gateways de pagamento, seguros** ou funcionalidades adicionais, permitindo a evolução contínua da plataforma.

### 2.1 Personas

#### Persona 1 – Proprietária de Ferramentas

- **Nome:** Ana Rodrigues
- **Idade:** 36
- **Profissão:** Arquiteta freelancer
- **Contexto:** Possui várias ferramentas de jardinagem e bricolagem que utiliza raramente, ocupando espaço e ficando longos períodos sem uso.
- **Objetivos:** Rentabilizar equipamentos; libertar espaço; reduzir desperdício através da economia de partilha.



#### Persona 2 – Utilizador / Renter

- **Nome:** Carlos Nunes
- **Idade:** 29
- **Profissão:** Técnico de manutenção
- **Contexto:** Necessita de ferramentas específicas para pequenas obras e reparações ocasionais, não justificando a compra dos equipamentos.
- **Objetivos:** Encontrar rapidamente ferramentas adequadas; alugar equipamentos de qualidade; reduzir custos associados a compras pontuais.



## Persona 3 – Administradora da Plataforma



- **Nome:** Sofia Ferreira
- **Idade:** 41
- **Profissão:** Gestora de Produto Digital
- **Contexto:** Responsável por supervisionar as operações da plataforma, analisar métricas de utilização e garantir a conformidade e segurança do sistema.
- **Objetivos:** Garantir uma operação estável e segura; monitorizar o desempenho da plataforma; promover o crescimento sustentável da comunidade.

## 2.2 Scenarios

### Cenário 1 – Proprietária adiciona uma ferramenta

A Ana (proprietária), após fazer login na plataforma, decide disponibilizar uma ferramenta que raramente utiliza.

Acede à opção “*Adicionar ferramenta*”, onde introduz o título, a categoria, o preço por dia, o preço da caução, localização e fotografia do equipamento.

Após a submissão, a ferramenta fica imediatamente visível no catálogo público da plataforma, podendo ser pesquisada por outros utilizadores.

### Cenário 2 – Arrendatário pesquisa e reserva uma ferramenta

O Carlos (arrendatário) necessita de uma máquina de lavar a pressão para um trabalho pontual durante o fim de semana.

Acede à plataforma, utiliza a pesquisa e aplica filtros como localização, preço e disponibilidade.

Após tal pesquisa seleciona a ferramenta pretendida e efetua a reserva e pagamento para as datas desejadas.

A seguir, receberá uma notificação de confirmação da reserva e na aba do utilizador poderá atualizar se no estado da reserva.

### Cenário 3 – Pagamento e devolução da ferramenta

No momento da reserva, o pagamento é processado de forma segura por um serviço externo integrado (por exemplo, PayPal em ambiente sandbox).

O proprietário entrega a ferramenta ao arrendatário na data acordada.

## Cenário 4 – Administradora monitoriza a plataforma

A Sofia (administradora da plataforma) acede ao painel de administração para acompanhar o funcionamento global do sistema.

Consulta métricas como o número de utilizadores registados, reservas ativas.

Com base nesses dados, identifica possíveis irregularidades e toma decisões para garantir a estabilidade, segurança e crescimento sustentável da plataforma.

## 2.3 Epics

### 1 – Procura e Descoberta de Ferramentas

Permitir que os utilizadores pesquisem e visualizem ferramentas disponíveis na plataforma, aplicando filtros relevantes.

- **US1:** Como Carlos (arrendatário), quero pesquisar ferramentas por palavra-chave para encontrar rapidamente o que preciso.
- **US2:** Como Carlos, quero filtrar ferramentas por categoria e localização para encontrar opções adequadas na minha área.
- **US3:** Como Carlos, quero visualizar detalhes da ferramenta (descrição, fotos e avaliações) para decidir com confiança.

### 2 – Reserva e Agendamento

Permitir que os utilizadores reservem ferramentas para períodos específicos, evitando, conflitos de disponibilidade.

- **US4:** Como Carlos, quero reservar uma ferramenta para datas específicas para garantir a sua disponibilidade.
- **US5:** Como Carlos, quero receber uma notificação de confirmação após a reserva para ter certeza de que foi efetuada.
- **US6:** Como Carlos, quero cancelar uma reserva num prazo definido para evitar penalizações.

### 3 – Pagamentos e Transações

Suportar pagamentos seguros e a correta transferência de valores entre utilizadores.

- **US7:** Como Carlos, quero efetuar o pagamento online de forma segura no momento da reserva..

## 4 – Gestão de Ferramentas

Permitir aos proprietários gerir as suas ferramentas e respetiva disponibilidade.

- **US8:** Como Ana, quero adicionar novas ferramentas com fotos, preço e disponibilidade.
- **US9:** Como Ana, quero editar ou remover ferramentas do catálogo sempre que necessário.
- **US10:** Como Ana, quero definir períodos de indisponibilidade para manutenção ou uso pessoal.

## 5 – Perfis de Utilizador e Histórico

Disponibilizar informação consolidada sobre a atividade dos utilizadores na plataforma.

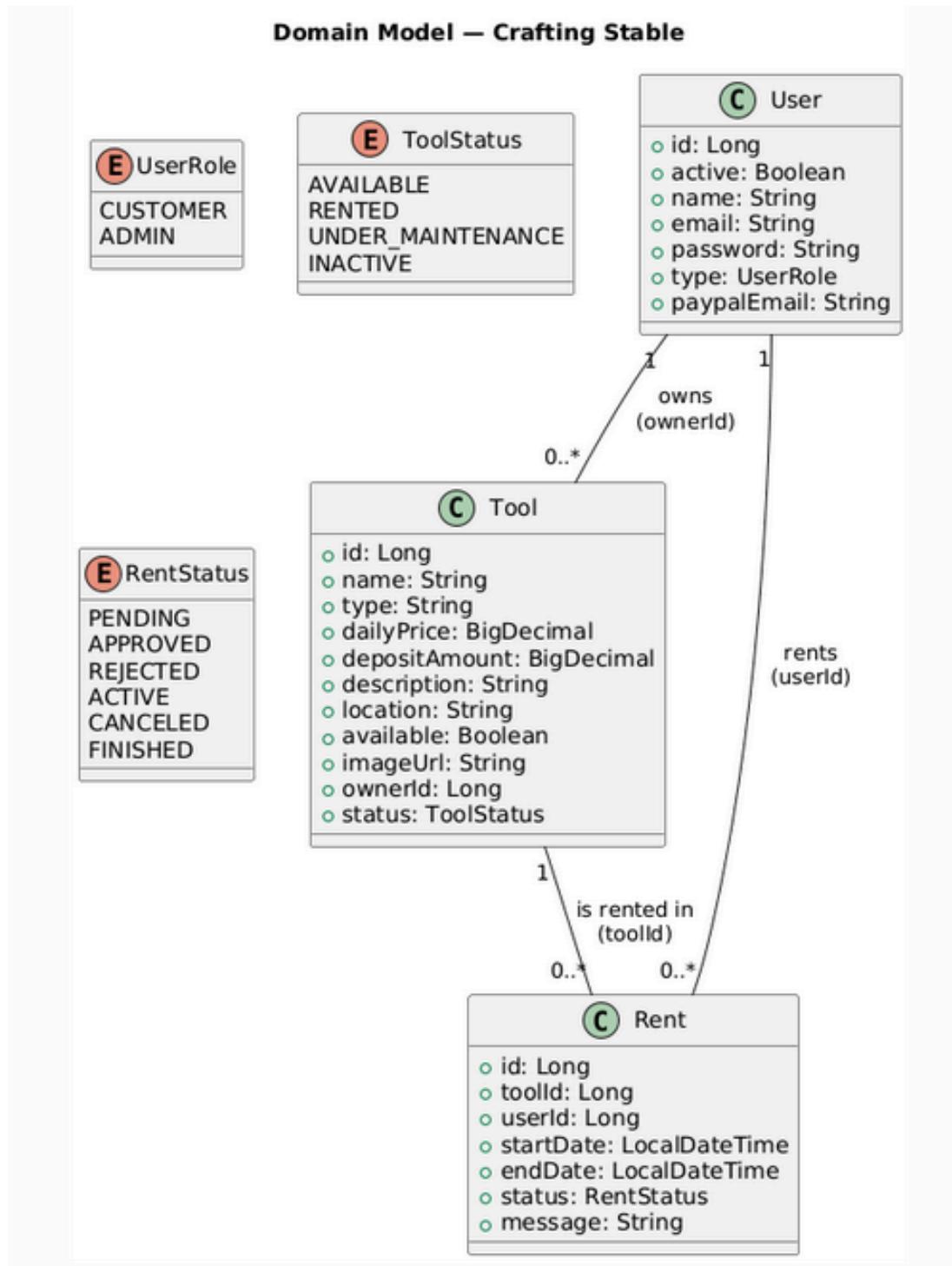
- **US11:** Como Carlos, quero consultar o meu histórico de reservas para acompanhar alugueres passados.

## 6 – Gestão da Plataforma e Monitorização

Fornecer ferramentas administrativas para garantir a estabilidade, segurança e crescimento do sistema.

- **US12:** Como Sofia (administradora), quero consultar métricas de utilização da plataforma para analisar o seu desempenho.
- **US13:** Como Sofia, quero identificar atividades suspeitas para garantir a segurança do sistema.

## 3 Domain Model



## 4 Arquitetura

### 4.1 Requisitos Funcionais e Não-Funcionais

#### 4.1.1 Requisitos Funcionais

##### 1. Gestão de Contas e Autenticação

- **RF01:** O sistema deve permitir o registo de novos utilizadores (Renter e Owner).
- **RF02:** O sistema deve permitir o login e logout de utilizadores.
- **RF03:** O sistema deve distinguir perfis de Utilizador Comum e Administrador com permissões distintas.

##### 2. Gestão de Ferramentas (Owners)

- **RF04:** O proprietário deve conseguir criar um anúncio de ferramenta inserindo: título, descrição, categoria, preço/dia, valor da caução e localização.
- **RF05:** O sistema deve permitir a inserção do link da fotografia associada à ferramenta.
- **RF06:** O proprietário deve poder editar ou remover os seus anúncios.

##### 3. Procura e Descoberta (Renters)

- **RF07:** O sistema deve permitir a pesquisa de ferramentas por palavras-chave.
- **RF08:** O sistema deve permitir filtrar resultados por: Categoria, Localização e Preço.
- **RF09:** O sistema deve apresentar uma página de detalhe da ferramenta com todas as informações e disponibilidade.

##### 4. Reservas e Pagamentos

- **RF10:** O Arrendatário deve conseguir selecionar datas de início e fim para a reserva.
- **RF11:** O sistema deve calcular automaticamente o custo total (dias × preço).
- **RF12:** O sistema deve impedir a reserva de ferramentas em datas já ocupadas.
- **RF13:** O sistema deve processar pagamentos (integração com Gateway externo: PayPal Sandbox).
- **RF14:** O sistema deve gerar um comprovativo de pagamento/reserva após o sucesso da transação.

##### 5. Administração (Backoffice)

- **RF15:** O Administrador deve ter acesso a um dashboard com métricas (nº utilizadores, reservas ativas).
- **RF16:** O Administrador deve conseguir remover anúncios que violem as políticas da plataforma.

#### 4.1.2 Requisitos Não-Funcionais

##### Desempenho (Performance)

- **RNF01 - Tempo de Resposta:** O sistema deve garantir um tempo de resposta inferior a 2 segundos para operações críticas (pesquisa, carregamento da home page) em condições normais.
- **RNF02 - Capacidade:** O sistema deve suportar múltiplos utilizadores concorrentes sem degradação perceptível do serviço (validado via testes de carga com k6).

##### Segurança (Security)

- **RNF03 - Autenticação:** A autenticação deve ser gerida via tokens JWT (JSON Web Tokens) para garantir sessões seguras e stateless.
- **RNF04 - Dados Sensíveis:** As palavras-passe devem ser armazenadas com hash robusto (ex: bcrypt). Dados de pagamento não devem ser guardados na base de dados local, sendo delegados ao gateway de pagamentos.

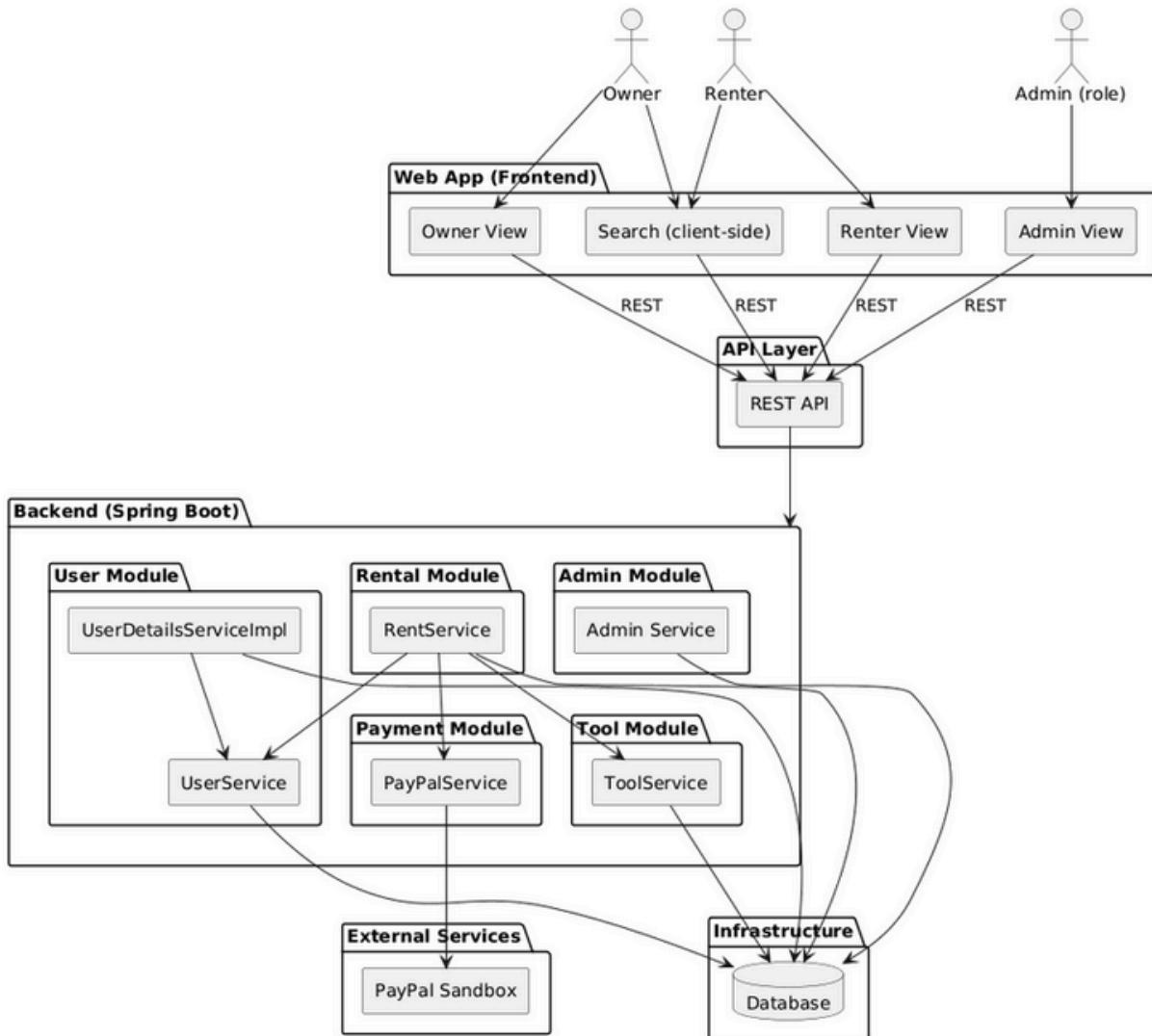
##### Disponibilidade e Confiabilidade (Availability & Reliability)

- **RNF05 - Uptime:** A aplicação deve manter uma disponibilidade de 99.9% durante o horário comercial.

##### Escalabilidade e Arquitetura (Scalability & Architecture)

- **RNF6 - Escalabilidade Horizontal:** O backend deve ser desenvolvido para permitir escalabilidade horizontal utilizando orquestração de contentores (Docker).
- **RNF7 - Portabilidade:** O Frontend (React + TypeScript + Vite) deve ser compatível com as versões mais recentes dos principais browsers (Chrome, Firefox, Safari, Edge).
- **RNF8 - Integração:** A arquitetura deve ser desacoplada (API RESTful) para facilitar a integração futura com outros clientes (ex: App Mobile).

## 4.2 Vista da arquitetura



# 5 API para Desenvolvedores

PayPal PayPal payment integration endpoints		
POST	/api/paypal/orders	Create a PayPal order
POST	/api/paypal/orders/{orderId}/capture	Capture a PayPal order
GET	/api/paypal/orders/{orderId}	Get order details
user-controller		
PUT	/api/users/{id}/role	
PUT	/api/users/{id}/deactivate	
PUT	/api/users/{id}/activate	
GET	/api/users	
POST	/api/users	
GET	/api/users/{id}	
DELETE	/api/users/{id}	
GET	/api/users/{id}/stats	
GET	/api/users/total-users	
GET	/api/users/stats/admin	
GET	/api/users/count	
tool-controller		
GET	/api/tools/{id}	
PUT	/api/tools/{id}	
DELETE	/api/tools/{id}	
PUT	/api/tools/{id}/status	
GET	/api/tools	
POST	/api/tools	
GET	/api/tools/{id}/check-availability	
GET	/api/tools/{id}/blocked-dates	
GET	/api/tools/{type}/{type}	
GET	/api/tools/available	
rent-controller		
PUT	/api/rents/{id}/reject	
PUT	/api/rents/{id}/cancel	
PUT	/api/rents/{id}/approve	
GET	/api/rents	
POST	/api/rents	
GET	/api/rents/{id}	
DELETE	/api/rents/{id}	
GET	/api/rents/interval	
auth-controller		
POST	/api/auth/register	
POST	/api/auth/login	
GET	/api/auth/me	
analytics-controller		
POST	/api/analytics/track	
GET	/api/analytics/user/{userId}	
GET	/api/analytics/tool/{toolId}	
GET	/api/analytics/summary	
GET	/api/analytics/events/{eventType}	

## 6 Referências e Recursos

### Recursos do Projeto

Resource:	URL/Location
Git Repository	<a href="#">Git Repository</a>
QA dashboard(online)	<a href="#">SonarCloud</a>
CI/CD pipeline	<a href="#">GitHub Actions</a>
Deployment ready to use	<a href="#">Crafting Stable</a>