



CENTRO UNIVERSITÁRIO CATÓLICO SALESIANO AUXILIUM

RECREDENCIADO PELA PORTARIA Nº 358 DE 01/06/2021 - D.O.U. 02/06/2021 - SEÇÃO 1 - PÁG. 59

Sistema Marketplace para Freelancers



Thales Yuji Nishikawa Talhacoli

Bernardo Zorzeto de Silos

Sistema Marketplace para Freelancers

Relatório técnico apresentado como requisito parcial para obtenção de aprovação na disciplina de Projeto Final de Curso, no Curso de Tecnologia em Desenvolvimento de Sistemas, no Centro Universitário Católico Salesiano Auxilium.

Orientador: Alexandre Marcelino da Silva

Araçatuba, 2025



RESUMO

O presente relatório descreve o desenvolvimento de uma plataforma web chamada Marketplace para Freelancers, cujo objetivo é facilitar a contratação de serviços entre clientes e profissionais autônomos. A proposta surgiu da necessidade de oferecer um ambiente seguro, funcional e completo para freelancers divulgarem seus trabalhos e para contratantes encontrarem mão de obra especializada. O sistema permite o cadastro de usuários, publicação de demandas, envio e recebimento de propostas, assinatura digital de contratos, controle financeiro com métodos de pagamento integrados (PIX, boleto e cartão), além de avaliações mútuas e sistema de denúncias. O desenvolvimento foi realizado com base em metodologias ágeis e utilizou tecnologias como Django, ReactJS, MySQL e autenticação JWT, garantindo confiabilidade, rastreabilidade e escalabilidade para futuras evoluções da aplicação.

Palavras-chave: freelancers, marketplace digital, contratos digitais, pagamentos online, autenticação JWT, metodologias ágeis.



CENTRO UNIVERSITÁRIO CATÓLICO SALESIANO AUXILIUM

RECREDENCIADO PELA PORTARIA Nº 358 DE 01/06/2021 - D.O.U. 02/06/2021 - SEÇÃO 1 - PÁG. 59



SUMÁRIO

1. Introdução	4
1.1. Contexto	4
1.2. Descrição da Solução.....	4
1.3. Objetivos	4
1.4. Metodologia de Desenvolvimento	5
2. Modelagem de Negócio	5
2.1. Business Model Canvas	5
2.2. Product Backlog Priorizado	6
3. Modelagem de Sistemas	8
3.1. Diagrama de Casos de Uso	8
3.2. Especificação dos Casos de Uso	9
3.2.1 Cadastro de Usuário	9
3.3. Diagrama de Classes	11
3.4. Diagrama do Banco de Dados (DER, MER etc.)	13
3.5. Diagrama de Sequência.....	15
3.5.1 Diagrama de Sequência – Cadastro de Trabalho	15
3.6. Diagrama de Componentes / Implantação	15
4. Conclusão.....	16
5. Referências Bibliográficas	17



LISTA DE FIGURAS

Figura 1 – Business Model Canvas do sistema Marketplace para Freelancers.....	6
Figura 2 – Product Backlog priorizado do sistema Marketplace para Freelancers.....	7
Figura 3 – Diagrama de Casos de Uso do sistema Marketplace para Freelancers.....	8
Figura 4 – Mockup da tela de cadastro de usuário	9
Figura 5 – Ficha Técnica do Curso de Ação – Cadastro de Usuário.....	10
Figura 6 – Ficha Técnica do Curso de Ação – Assinatura de Contrato	10
Figura 7 – Diagrama de Classes do sistema Marketplace para Freelancers	11
Figura 8 – Diagrama de Banco de Dados Relacional do sistema Marketplace para Freelancers.....	13
Figura 9 – Diagrama de Sequência Cadastro de Trabalho no sistema Marketplace para Freelancers.	15
Figura 10 – Diagrama de Componentes do sistema Marketplace para Freelancers.	16



1. Introdução

1.1. Contexto

Com o crescimento das atividades profissionais autônomas e da prestação de serviços digitais, muitas pessoas passaram a buscar formas de divulgar seu trabalho e conquistar novos clientes de forma independente. No entanto, ainda é comum que essas relações ocorram de maneira informal, sem contratos claros, controle de pagamentos ou mecanismos de avaliação. Ao mesmo tempo, empresas e contratantes também enfrentam dificuldades para encontrar profissionais confiáveis, especialmente em plataformas que ofereçam segurança, organização e suporte completo.

Observando esse cenário, percebe-se a ausência de soluções locais e acessíveis que reúnam todos os recursos necessários para garantir uma contratação estruturada e segura. É nesse contexto que surgiu a proposta de desenvolvimento do sistema Marketplace para Freelancers.

1.2. Descrição da Solução

A solução desenvolvida é uma plataforma web chamada Marketplace para Freelancers, projetada para organizar todas as etapas do processo de contratação de serviços entre clientes e profissionais autônomos. O sistema foi construído utilizando o framework Django no backend, com Django REST Framework para criação de APIs RESTful. No frontend, foi adotado o ReactJS, proporcionando uma interface dinâmica e responsiva. O banco de dados utilizado é o MySQL, escolhido por sua robustez e compatibilidade com integrações web. A autenticação é baseada em JWT (JSON Web Token), permitindo sessões seguras com controle de acesso por tipo de usuário.

Entre os principais recursos já implementados estão: cadastro e login de usuários, publicação de trabalhos, envio e gestão de propostas, geração e assinatura de contratos digitais, gerenciamento de pagamentos com múltiplos métodos (PIX, boleto, cartão), comunicação via mensagens internas, envio de notificações, denúncias e sistema de avaliações mútuas.

Além disso, o sistema oferece funcionalidades de gerenciamento de habilidades, permitindo que os administradores cadastrem áreas de atuação organizadas por categoria e subcategoria. Cada trabalho pode ser vinculado a uma ou mais habilidades, o que torna a plataforma mais inteligente e precisa, possibilitando a melhoria nas buscas e maior compatibilidade entre perfis de usuários (clientes e freelancers).

Os testes de API foram amplamente realizados com o Postman, garantindo a validação das funcionalidades e a simulação de diferentes fluxos de uso. A aplicação foi desenvolvida seguindo boas práticas de organização, segurança e estruturação de código, com foco na clareza e manutenção do sistema.

1.3. Objetivos

Desenvolver uma plataforma web funcional e segura que permita a contratação de serviços entre freelancers e clientes, com controle de propostas, contratos, pagamentos e comunicação integrada.



Integrar soluções externas para validação de CPF/CNPJ, geração de boletos, pagamentos via PIX, cartão e assinatura digital de contratos, garantindo maior confiabilidade e valor jurídico às transações.

Implementar um sistema de categorização com base em habilidades e áreas de atuação, permitindo maior precisão na oferta e busca por trabalhos, favorecendo o match ideal entre clientes e freelancers.

Proporcionar uma experiência de uso completa, com usabilidade, rastreabilidade das ações e um ambiente profissional que promova confiança entre as partes envolvidas.

1.4. Metodologia de Desenvolvimento

Para o desenvolvimento deste projeto, foi adotada a metodologia ágil, com foco em ciclos de entrega curtos (sprints) e melhorias incrementais. A divisão por etapas permitiu validar cada funcionalidade separadamente, garantindo maior controle sobre o progresso e a correção de eventuais ajustes durante o processo.

A estrutura do sistema foi organizada em múltiplos módulos no framework Django, separando funcionalidades por áreas como usuários, trabalhos, propostas, contratos, pagamentos, avaliações, mensagens, denúncias, notificações e habilidades. O backend foi desenvolvido com Django REST Framework, permitindo a criação de APIs RESTful com autenticação JWT (JSON Web Token), o que garante segurança e controle de acesso entre usuários do tipo cliente, freelancer e administrador. O banco de dados utilizado foi o MySQL, pela sua estabilidade, compatibilidade e facilidade de integração com a stack Python.

As requisições e respostas da API foram testadas com o Postman, simulando os principais fluxos do sistema, como cadastro de usuários, envio de propostas, contratos, pagamentos e associação de habilidades a trabalhos. O projeto foi desenvolvido integralmente em ambiente local, com todo o controle de versões, testes e ajustes realizados diretamente no sistema de arquivos do desenvolvedor.

Os protótipos de interface das telas principais foram planejados com base no escopo funcional definido durante as fases iniciais do projeto, e gerados com apoio assistido por inteligência artificial. Essa abordagem permitiu visualizar a navegação e layout das telas com agilidade, garantindo coerência com os fluxos reais e as regras de negócio do sistema.

2. Modelagem de Negócio

2.1. Business Model Canvas

O Business Model Canvas é uma ferramenta visual amplamente utilizada para modelar e entender o modelo de negócios de um projeto. Ele permite mapear, em uma única página, os principais elementos que compõem o funcionamento de um produto ou serviço, auxiliando na organização estratégica e na comunicação entre os envolvidos. O objetivo do Canvas é oferecer uma visão clara



das propostas de valor, segmentos de clientes, canais de distribuição, fontes de receita, estrutura de custos, atividades-chave, recursos-chave e parcerias estratégicas.

No projeto Marketplace para Freelancers, o Canvas foi utilizado desde as fases iniciais para estruturar a proposta do sistema, alinhando os objetivos técnicos às necessidades de negócio. Durante o desenvolvimento, o modelo foi sendo ajustado para refletir avanços funcionais relevantes, como a categorização e associação de habilidades aos trabalhos. Essa funcionalidade amplia a organização das oportunidades disponíveis, melhora a experiência de busca e reflete diretamente na proposta de valor da plataforma — beneficiando tanto freelancers, que ganham visibilidade em suas áreas de atuação, quanto clientes, que localizam perfis mais alinhados com suas demandas.

Parceiros-chave	Atividades-chave	Proposta de Valor	Relacionamento com Cliente	Segmentos de Clientes
APIs externas para validação de CPF/CNPJ Gateways de pagamento (Gerencianet, Asaas, Mercado Pago ou similar) Serviços de assinatura digital de contratos (Clicksign, D4Sign, etc.)	Cadastro e gerenciamento de usuários Publicação e proposta de trabalhos Geração e assinatura de contratos Processamento de pagamentos Avaliação, denúncias e comunicação entre usuários Cadastro e gestão de habilidades Associação de habilidades aos trabalhos	Plataforma segura e confiável para contratação de serviços entre freelancers e clientes Formalização de acordos por meio de contratos digitais Integração com métodos reais de pagamento (PIX, boleto, cartão) Validação de CPF/CNPJ para maior segurança Ambiente profissional com comunicação, histórico e avaliações Organização dos trabalhos por área de atuação, com base em habilidades. Maior clareza para freelancers e clientes encontrarem perfis ou projetos relevantes.	Comunicação direta via chat integrado Sistema de avaliações mútuas Notificações internas (a serem implementadas) Suporte por e-mail (potencial futuro) Aprimoramento de busca com base em habilidades cadastradas (planejado)	Freelancers autônomos em busca de oportunidades de trabalho Clientes (pessoas físicas ou jurídicas) que desejam contratar serviços específicos Público-alvo geral (sem foco regional) Freelancers especializados por área de atuação Clientes em busca de perfis com habilidades específicas
Recursos-chave	Estrutura de Custos	Fontes de Receita	Canais	-
Tecnologias: Django, Django REST Framework, ReactJS, MySQL, JWT Infraestrutura: ambiente local com futura hospedagem Equipe de desenvolvimento composta por dois integrantes Sistema de categorização via tabela de habilidades e vínculos com trabalhos	Hospedagem do sistema para apresentação e operação Custos de uso das APIs externas (pagamento, contrato, validação) Tempo dedicado ao desenvolvimento e testes (não remunerado)	Modelo de negócio freemium Plano gratuito com limitações Plano premium com recursos e vantagens adicionais (ex: destaque nos trabalhos, acesso prioritário, etc.)	Plataforma web (versão desenvolvida para o TCC) Planejamento inicial inclui futura versão mobile Comunicação por e-mail e dentro do sistema (chat e notificações)	-

Figura 1 – Business Model Canvas do sistema Marketplace para Freelancers.

2.2. Product Backlog Priorizado

O backlog do projeto é uma lista organizada de tudo que será desenvolvido no sistema, incluindo funcionalidades, recursos técnicos e melhorias estruturais. Essa lista é construída de forma estratégica, com os itens ordenados por importância, complexidade e tempo estimado de implementação, o que contribui para o planejamento eficiente das etapas de desenvolvimento.

No caso do sistema Marketplace para Freelancers, o backlog foi montado com base nas funcionalidades reais que o sistema oferece, estruturadas de acordo com o fluxo de uso: iniciando pelo cadastro de usuários, passando pela publicação e proposta de trabalhos, geração de contratos, pagamentos, mensagens, avaliações, denúncias, notificações e, posteriormente, a categorização por habilidades, que aprimora a experiência de busca e organização do sistema.

Cada item do backlog foi descrito de forma objetiva e técnica, incluindo as seguintes informações:

- Tipo (funcional ou técnico),
- Peso de complexidade,



- Estimativa de horas para desenvolvimento,
- Prioridade de execução,
- Dependências.

O objetivo do backlog é garantir que todas as funcionalidades estejam bem documentadas, viáveis dentro do escopo e cronograma disponíveis, e alinhadas com a proposta real da plataforma.

ID	Nome do Requisito	Tipo	Descrição	Peso	Estimativa de Horas	Prioridade	Dependências	Crterios de Aceitação
1	Cadastro de usuário	Funcional	Permitir que clientes e freelancers criem contas com seus dados pessoais e tipo de perfil.	3	6 horas	Alta	-	Usuário deve se cadastrar e receber confirmação de criação com distinção de perfil.
2	Validação de CPF/CNPJ via API	Funcional	Validar CPF ou CNPJ durante o cadastro utilizando uma API externa para garantir autenticidade.	4	8 horas	Alta	REQUISITO 1	Cadastro só deve prosseguir com documento válido. Exibir erro claro em caso de falha.
3	Login e autenticação com sessão segura	Funcional	Permitir login com email e senha, utilizando JWT para autenticação e controle de sessão.	3	6 horas	Alta	REQUISITO 2	Usuário deve acessar sua conta e ser redirecionado ao painel de acordo com o tipo de perfil.
4	Recuperação de senha via email	Funcional	Permitir redefinição de senha por link seguro enviado ao email cadastrado.	2	4 horas	Média	REQUISITO 3	Usuário deve redefinir a senha e receber feedback de sucesso.
5	Conformidade com LGPD	Não funcional	Garantir que a aplicação esteja de acordo com a LGPD, permitindo controle de dados pelo usuário.	3	5 horas	Alta	REQUISITO 1	Usuário deve visualizar política de dados e solicitar exclusão de sua conta.
6	Acessibilidade e responsividade	Não funcional	Garantir que o sistema seja acessível e se adapte corretamente a desktop, tablet e mobile.	2	5 horas	Média	-	Todos os usuários devem acessar as funcionalidades em diferentes dispositivos.
7	Gerenciamento de plano gratuito e premium	Funcional	Permitir que usuários escolham entre plano gratuito e plano premium com benefícios exclusivos.	3	8 horas	Alta	REQUISITO 1	Usuário deve ver limites e vantagens, e conseguir alterar entre planos.
8	Chat em tempo real entre usuários	Funcional	Disponibilizar chat direto entre cliente e freelancer após proposta aceita.	3	10 horas	Alta	REQUISITO 13	Usuários devem se comunicar sem falhas após contrato firmado.
9	Notificações do sistema em tempo real	Funcional	Notificar usuários sobre mudanças de status, pagamentos, avaliações e mensagens.	3	6 horas	Alta	REQUISITO 11	Usuário deve visualizar alertas em tempo real no painel.
10	Busca de freelancers com filtros	Funcional	Permitir que clientes busquem freelancers por categoria, avaliação, valor e disponibilidade.	3	6 horas	Alta	REQUISITO 3	Cliente deve visualizar resultados com base em filtros definidos.
11	Envio de proposta de trabalho	Funcional	Permitir que clientes enviem propostas detalhadas, com valores e prazos.	3	8 horas	Alta	REQUISITO 10	Freelancer deve receber a proposta e poder aceitar ou recusar.
12	Visualização de status da proposta	Funcional	Permitir que clientes e freelancers vejam o status das propostas (pendente, aceita, recusada).	2	5 horas	Alta	REQUISITO 11	Status deve ser visível e atualizado em tempo real no painel.
13	Assinatura digital de contrato	Funcional	Permitir que o contrato seja assinado digitalmente pelas partes após proposta aceita.	4	10 horas	Alta	REQUISITO 12	Contrato deve ser assinado via API externa com validade jurídica.
14	Visualização do contrato assinado	Funcional	Permitir que ambas as partes acessem o contrato assinado a qualquer momento.	2	4 horas	Alta	REQUISITO 13	Contrato assinado deve ser exibido de forma clara e protegida.
15	Pagamento via PIX integrado	Funcional	Permitir que clientes realizem pagamento via PIX através de API externa.	4	8 horas	Alta	REQUISITO 13, REQUISITO 14	Cliente deve gerar QR Code PIX e ter confirmação automática no sistema.
16	Pagamento via boleto bancário	Funcional	Emitir boleto bancário com vencimento via integração com API externa.	3	6 horas	Média	REQUISITO 13, REQUISITO 14	Cliente deve conseguir emitir e pagar boleto. Status deve ser atualizado.
17	Pagamento com cartão de crédito e débito	Funcional	Integrar pagamento via cartão utilizando API como Asaas ou Gerencianet.	4	10 horas	Alta	REQUISITO 13, REQUISITO 14	Transações devem ser aprovadas e confirmadas com segurança.
18	Exportação de comprovante de pagamento	Funcional	Gerar comprovante de pagamento em PDF após confirmação.	2	5 horas	Média	REQUISITO 15, REQUISITO 16, REQUISITO 17	Usuário deve conseguir baixar o comprovante com dados da transação.
19	Histórico de transações com filtros	Funcional	Exibir todas as transações realizadas com opções de filtro por data e status.	3	6 horas	Média	REQUISITO 15, REQUISITO 16, REQUISITO 17	Usuário deve conseguir visualizar pagamentos por período e status.
20	Avaliação do freelancer após serviço	Funcional	Permitir que clientes avaliem freelancers com nota e comentário.	2	4 horas	Média	REQUISITO 14	Avaliação deve ser enviada após contrato concluído.
21	Avaliação do cliente pelo freelancer	Funcional	Permitir que freelancers também avaliem os clientes.	2	4 horas	Média	REQUISITO 20	Comentário e nota devem ser salvos corretamente no perfil do cliente.
22	Sistema de denúncias	Funcional	Permitir que usuários denunciem comportamentos abusivos.	2	4 horas	Média	REQUISITO 21	Usuário deve preencher justificativa e evidência e o admin deve ser notificado.
23	Painel administrativo do sistema	Funcional	Disponibilizar ao administrador ferramentas para visualizar denúncias, usuários e pagamentos.	3	8 horas	Alta	REQUISITO 22	Administrador deve acessar painel protegido com filtros e relatórios.
24	Cadastro de Habilidades	Funcional	Permitir que o sistema tenha uma base de habilidades categorizadas e subcategorizadas, utilizadas pelos clientes ao cadastrar trabalhos. As habilidades devem ser previamente registradas no sistema para associação posterior.	2	10 horas	Média	-	O administrador deve conseguir cadastrar habilidades com nome, categoria e subcategoria, e elas devem ser listadas corretamente ao cadastrar um trabalho.
25	Associação de Habilidades ao Trabalho	Funcional	Permitir que o cliente selecione habilidades relevantes ao cadastrar um trabalho. O sistema deve permitir múltiplas habilidades por trabalho, vinculando os dados à base de habilidades cadastradas.	3	12 horas	Alta	REQUISITO 24	O cliente deve conseguir selecionar uma ou mais habilidades disponíveis no momento do cadastro de um trabalho. As habilidades selecionadas devem ser armazenadas corretamente na tabela de associação

Figura 2 – Product Backlog priorizado do sistema Marketplace para Freelancers.

3. Modelagem de Sistemas

3.1. Diagrama de Casos de Uso

O diagrama de casos de uso apresenta de forma visual as principais interações entre os usuários do sistema e as funcionalidades que ele oferece.

No caso do sistema Marketplace para Freelancers, foram considerados três tipos principais de usuários: cliente, freelancer e administrador. Cada um deles tem acesso a funcionalidades específicas, conforme seu perfil e permissões.

Este diagrama foi utilizado como apoio na definição dos requisitos do sistema e também serviu de base para a criação do backlog priorizado.

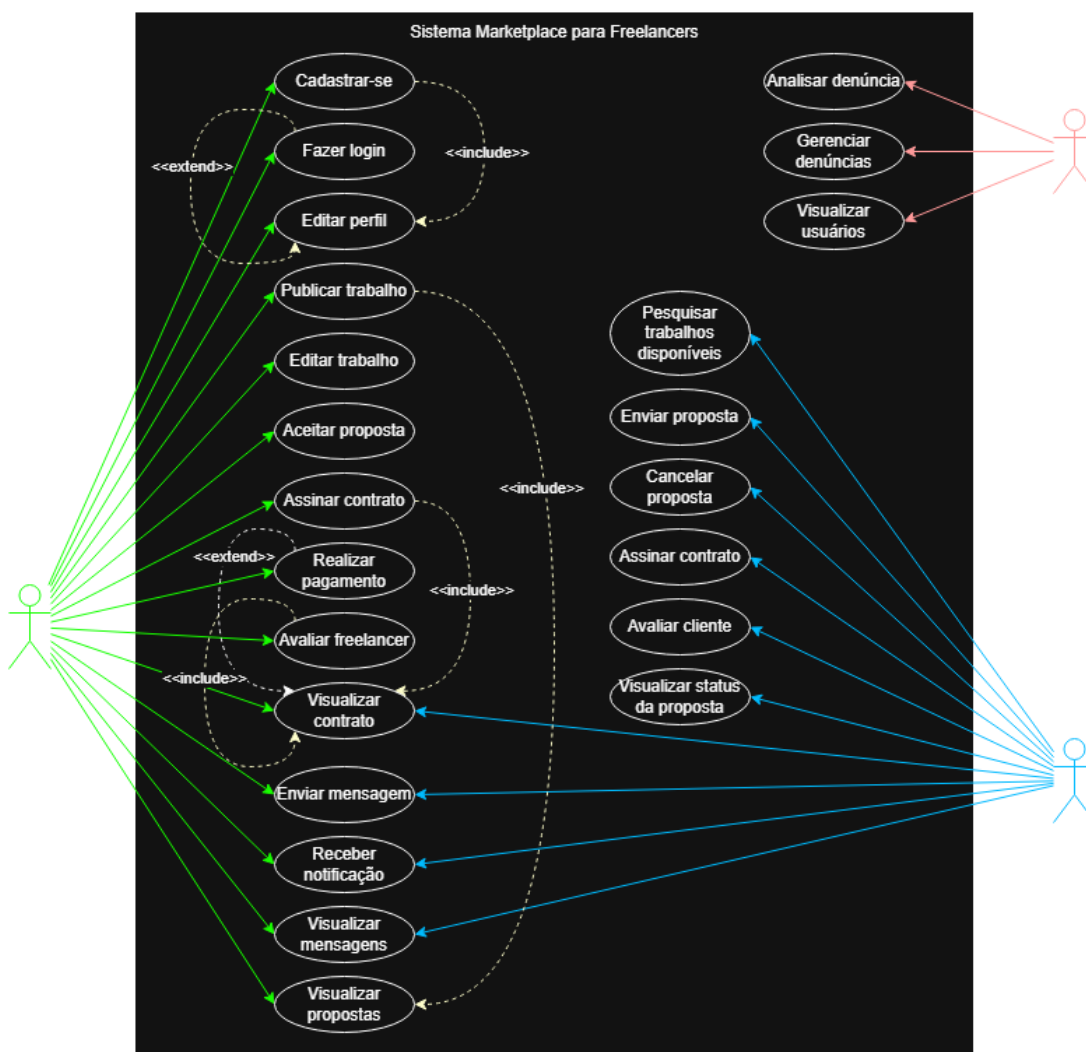


Figura 3 – Diagrama de Casos de Uso do sistema Marketplace para Freelancers.



3.2. Especificação dos Casos de Uso

Para cada evento, ou item do Backlog ou caso de uso (todos devem contém os mesmos elementos) apresentar o protótipo da tela que será/foi criada, bem como as condições de uso da interface

3.2.1 Cadastro de Usuário

Cadastro de Usuário

Nome completo

E-mail

Telefone

CPF

CNPJ

Senha

Confirmar Senha

Tipo de usuário

Cliente

Enviar foto

Cadastrar

Já tem uma conta? Fazer login

Figura 4 – Mockup da tela de cadastro de usuário



Criar a ficha de curso de ação para o caso de uso “Cadastro de Usuário”.

Objetivo	Pré-condições	Pós-condições	Regras de Negócio	Restrição/Risco
Permitir que clientes e freelancers criem uma conta no sistema para utilizar os recursos da plataforma.	O usuário não deve estar autenticado. Deve acessar a tela de cadastro a partir da página inicial ou de um link com a opção “Cadastre-se”.	A conta é criada com sucesso. O sistema armazena os dados do novo usuário e redireciona para a tela de login com uma mensagem de confirmação.	E-mail deve ser válido e único. CPF e CNPJ são campos separados com validação via API. Senha com no mínimo 6 caracteres. Tipo de usuário deve ser Cliente ou Freelancer. Upload de foto é opcional.	A falha na API de validação de CPF/CNPJ ou perda de conexão impede o cadastro.
Atores Envolvidos	Cenário Principal	Cenários Alternativos	Mensagem exibida	Ator Secundário
Cliente e Freelancer	O usuário acessa a tela de cadastro Preenche os campos: nome, e-mail, telefone, CPF, CNPJ Define e confirma a senha Seleciona o tipo de conta (Opcional) Envia uma foto de perfil Clica no botão “Cadastrar” O sistema valida os dados Em caso de sucesso, salva o usuário e redireciona para a tela de login com uma mensagem	Campos obrigatórios em branco CPF ou CNPJ inválido E-mail já cadastrado Senhas diferentes	“Cadastro realizado com sucesso. Faça login para continuar.”	Não se aplica. Somente o próprio usuário realiza o cadastro.

Figura 5 – Ficha Técnica do Curso de Ação – Cadastro de Usuário

Pode-se também adotar outro modelo de ficha de curso de ação, também conhecido como ficha técnica do curso de ação

Objetivo	Pré-condições	Pós-condições	Regras de Negócio	Restrição/Risco
Permitir que cliente e freelancer formalizem um acordo de prestação de serviço após a aceitação de uma proposta, garantindo o vínculo entre as partes.	A proposta deve estar com status “aceita”. Cliente e freelancer devem estar autenticados no sistema.	Um contrato é gerado, salvo no banco de dados e vinculado à proposta correspondente. Ambas as partes podem visualizar o contrato na tela de contratos do sistema.	Apenas o cliente pode iniciar a assinatura. O contrato deve conter data, nomes das partes, valor e ID da proposta. Um contrato só pode ser assinado uma vez por cada parte. A proposta deve obrigatoriamente estar em estado “aceita”	Caso a proposta não tenha sido aceita, o sistema deve bloquear a tentativa de assinar. Interrupções de rede podem impedir a geração ou salvamento do contrato.
Atores Envolvidos	Cenário Principal	Cenários Alternativos	Mensagem exibida	Ator Secundário
Cliente e Freelancer	Cliente acessa a proposta aceita Clica em “Assinar contrato” O sistema gera um contrato com dados da proposta Freelancer é notificado e acessa o contrato Freelancer assina Sistema confirma e registra o contrato assinado	Cliente tenta assinar proposta com status diferente de “aceita” Freelancer recusa a assinatura Interrupção durante a geração ou salvamento do contrato.	“Contrato assinado com sucesso. Ambas as partes podem visualizar o documento na área de contratos.”	Sistema (responsável por gerar, notificar e armazenar o contrato)

Figura 6 – Ficha Técnica do Curso de Ação – Assinatura de Contrato



3.3. Diagrama de Classes

O diagrama de classes é um dos principais diagramas da UML e tem como objetivo representar as classes que compõem o sistema, seus atributos, métodos e os relacionamentos entre elas.

No sistema Marketplace para Freelancers, o diagrama de classes foi utilizado para definir a estrutura lógica das entidades do sistema, servindo de base para a criação do banco de dados e para o desenvolvimento das funcionalidades previstas no backlog.

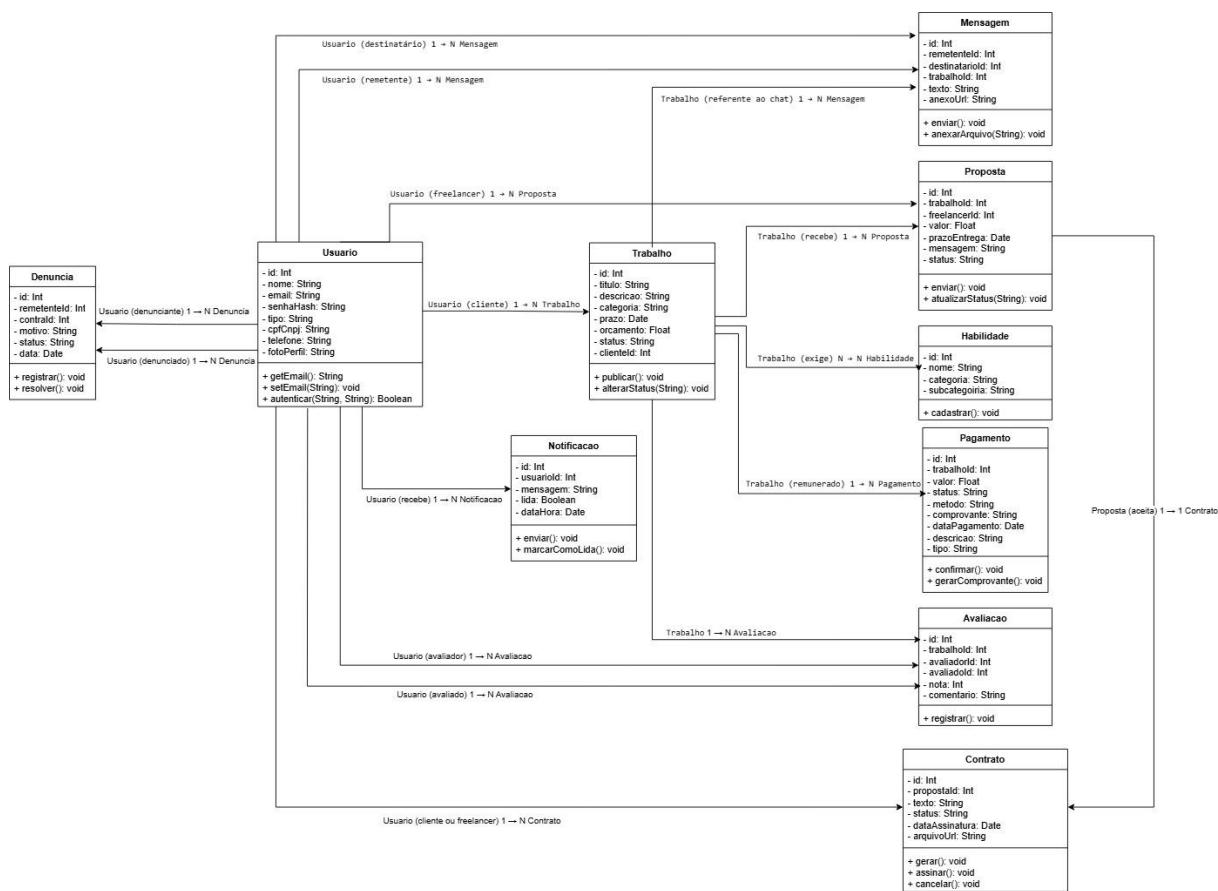


Figura 7 – Diagrama de Classes do sistema Marketplace para Freelancers

As principais classes do sistema incluem:

- **Usuário:** Representa todos os usuários do sistema, sejam clientes, freelancers ou administradores. Contém dados como nome, e-mail, senha, tipo, CPF/CNPJ, telefone e foto de perfil.
- **Trabalho:** Estrutura para serviços publicados por clientes, contendo título, descrição, categoria, subcategoria, prazo, orçamento e status.
- **Habilidade:** Representa conhecimentos ou competências específicas. É associada a trabalhos através de uma tabela intermediária (Trabalho_Habilidade), permitindo categorizar melhor os projetos.
- **Proposta:** Registra a oferta de freelancers para um trabalho específico, contendo valor, prazo de entrega, mensagem personalizada e status.



- Contrato: Formaliza a contratação após uma proposta ser aceita, com texto acordado, status, data de assinatura e anexo (arquivo).
- Pagamento: Está vinculado ao contrato e registra os dados de pagamento: valor, método (PIX, boleto, cartão), comprovante, data, tipo e status.
- Mensagem: Controla as comunicações privadas dentro da plataforma entre usuários, com suporte a texto, arquivos e vínculo ao trabalho.
- Avaliação: Permite que usuários avaliem uns aos outros após a conclusão de um trabalho, com nota e comentário, vinculados ao trabalho e às partes envolvidas.
- Notificação: Envia alertas internos para os usuários sobre eventos importantes no sistema (como nova proposta recebida, mensagem nova, etc).
- Denúncia: Registra comportamentos inadequados, vinculando o remetente, o denunciado, a proposta relacionada, motivo e status.

Todos esses elementos estão interligados para garantir a integridade dos dados, a rastreabilidade das ações e a fluidez da experiência do usuário dentro da plataforma.



3.4. Diagrama do Banco de Dados (DER, MER etc.)

ou Dicionário de Dados)

O Diagrama Entidade-Relacionamento (DER) foi utilizado para representar a estrutura lógica do banco de dados relacional do sistema Marketplace para Freelancers. Esse modelo visual descreve com precisão as entidades principais, seus atributos e todos os relacionamentos com integridade referencial, fundamentais para garantir o funcionamento correto do sistema.

A estrutura foi desenvolvida com base no planejamento técnico e funcional do projeto, refletindo fielmente os dados manipulados no sistema. Estão contempladas funcionalidades como cadastro e autenticação de usuários (clientes e freelancers), criação de trabalhos, propostas de freelancers, assinatura de contratos, pagamentos, troca de mensagens, avaliações mútuas, notificações internas, denúncias e a relação entre trabalhos e habilidades.

Abaixo está representado o Diagrama Entidade-Relacionamento completo:

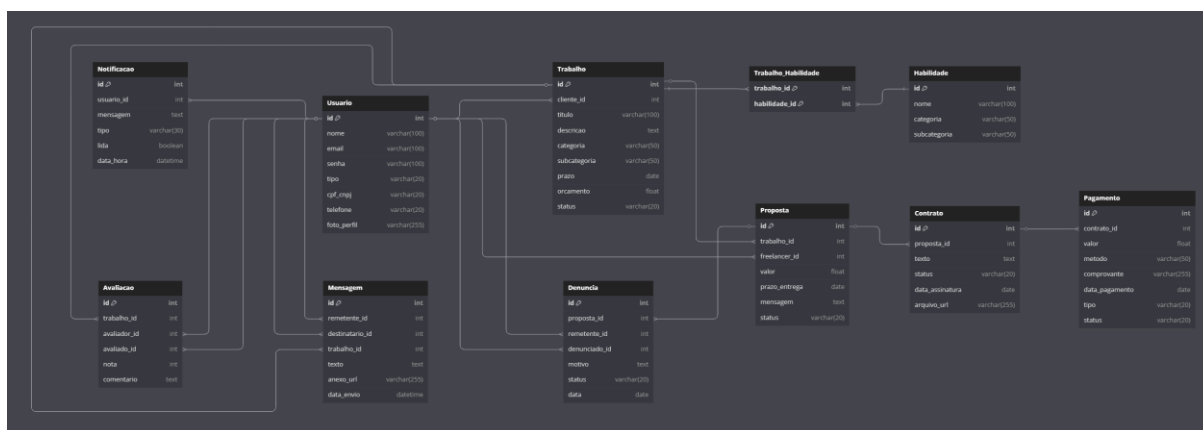


Figura 8 – Diagrama de Banco de Dados Relacional do sistema Marketplace para Freelancers.

Legenda dos relacionamentos:

Rel_01 – Um usuário (cliente) pode criar muitos trabalhos. Cada trabalho pertence a um único cliente.

Rel_02 – Um trabalho pode ter várias habilidades. Representado pela tabela de junção Trabalho_Habilidade.

Rel_03 – Um trabalho pode receber várias propostas de freelancers. Cada proposta pertence a um único freelancer.

Rel_04 – Uma proposta aceita gera um contrato. Cada contrato está vinculado a uma proposta.

Rel_05 – Um contrato possui um pagamento associado. Um pagamento pertence a um único contrato.

Rel_06 – Um usuário pode enviar e receber mensagens. Cada mensagem possui remetente e destinatário.



Rel_07 – Um usuário pode avaliar outro usuário ao final de um trabalho. A avaliação inclui nota e comentário, sendo vinculada a um trabalho específico e aos dois usuários (avaliador e avaliado).

Rel_08 – Um usuário pode receber diversas notificações com diferentes tipos, status de leitura e data.

Rel_09 – Um usuário pode denunciar outro usuário dentro do contexto de uma proposta. A denúncia inclui remetente, denunciado, motivo e status.

Rel_10 – Um contrato está vinculado a uma proposta aceita e contém o texto acordado, status, data de assinatura e anexo.

Rel_11 – Um pagamento está associado a um contrato e inclui valor, método, comprovante, tipo e status.

Esses relacionamentos garantem integridade referencial entre as entidades do sistema.



3.5. Diagrama de Sequência

O diagrama de sequência abaixo descreve o fluxo completo para o cadastro de um novo trabalho no sistema Marketplace para Freelancers. O fluxo envolve o usuário preenchendo os dados do trabalho, a tela de cadastro, validações feitas pelo Backend, persistência dos dados no banco e possível associação de habilidades ao trabalho.

3.5.1 Diagrama de Sequência – Cadastro de Trabalho

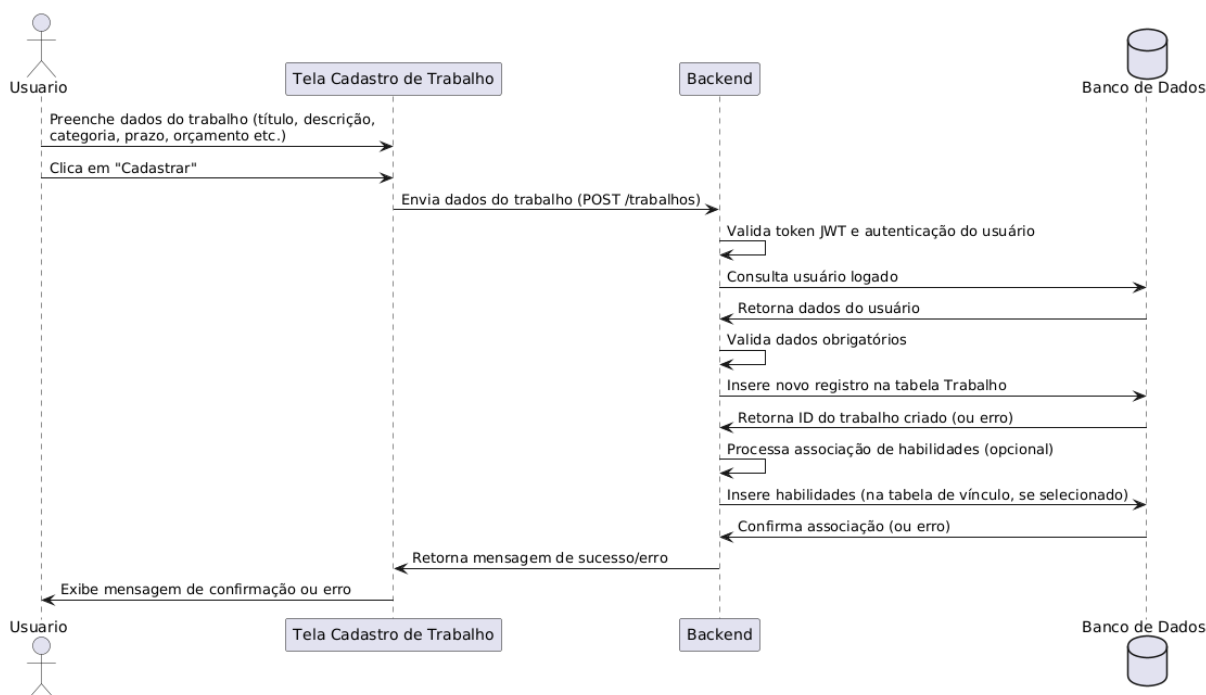


Figura 9 – Diagrama de Sequência Cadastro de Trabalho no sistema Marketplace para Freelancers.

Participantes:

- Usuário: Responsável por iniciar o cadastro do trabalho.
- Tela Cadastro de Trabalho: Interface onde o usuário insere os dados e recebe o feedback.
- Backend: Responsável pelas validações, lógica de negócio e persistência dos dados.
- Banco de Dados: Onde as informações são efetivamente armazenadas.

3.6. Diagrama de Componentes / Implantação

Objetivo do diagrama e principais componentes que o projeto possui

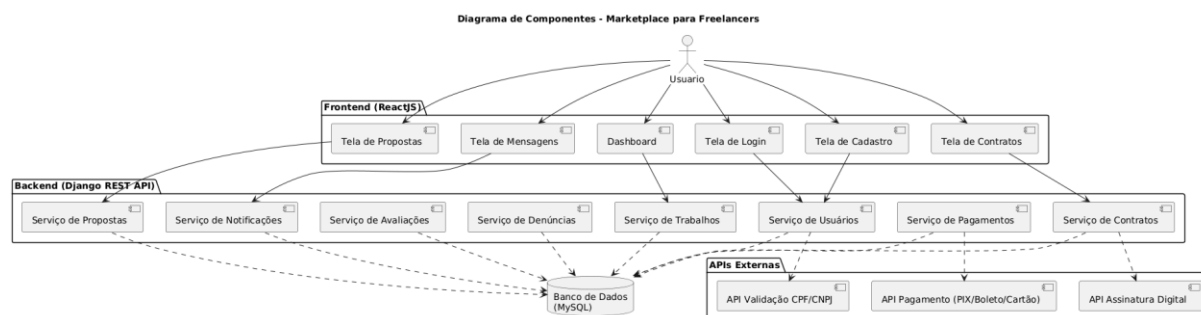


Figura 10 – Diagrama de Componentes do sistema Marketplace para Freelancers.

O diagrama de componentes apresentado acima evidencia a arquitetura do sistema Marketplace para Freelancers, segmentada nas principais camadas: Frontend (ReactJS), Backend (Django REST API), Banco de Dados (MySQL) e integrações com APIs externas (validação de CPF/CNPJ, pagamentos e assinatura digital). Os componentes destacados representam os módulos essenciais da plataforma, demonstrando como o usuário interage com as telas, que se comunicam com serviços do backend responsáveis pela lógica de negócio, persistência dos dados e integrações externas.

Essa estrutura garante a separação de responsabilidades, escalabilidade do sistema e facilita a manutenção e evolução futura da aplicação.

4. Conclusão

O desenvolvimento do sistema Marketplace para Freelancers resultou em uma plataforma web robusta, segura e funcional, que visa aproximar clientes e profissionais autônomos, facilitando a contratação de serviços de forma estruturada e confiável. Ao longo do projeto, foram implementadas funcionalidades essenciais como cadastro e autenticação de usuários, publicação e busca de trabalhos, propostas, contratos digitais com assinatura, processamento de pagamentos integrados, avaliações mútuas, notificações e sistema de denúncias.

Dentre os principais diferenciais, destaca-se a adoção de uma arquitetura modular, baseada em boas práticas de desenvolvimento ágil, segurança de dados, categorização de habilidades, além da integração com APIs externas para validação de documentos, métodos de pagamento e assinatura digital. Os testes intensivos realizados com o Postman garantiram a validação dos fluxos e contribuíram para a qualidade final do produto.

O projeto contribui significativamente para o cenário de plataformas digitais, trazendo benefícios como aumento da confiança nas relações de trabalho online, transparência nos processos e rastreabilidade de todas as interações. Além disso, proporciona um ambiente organizado, capaz de evoluir com facilidade, agregando valor tanto para clientes quanto para freelancers e servindo de base para futuras melhorias e expansões.



5. Referências Bibliográficas

DJANGO Software Foundation. Django Documentation. Disponível em:
<https://docs.djangoproject.com/>

REACTJS. React Documentation. Disponível em: <https://react.dev/>

MySQL. MySQL 8.0 Reference Manual. Disponível em: <https://dev.mysql.com/doc/>

PLANTUML. PlantUML Language Reference Guide. Disponível em: <https://plantuml.com/>