

Prova de Aptidão Profissional

Curso Profissional de Técnico de Informática – Sistemas



Plataforma de publicação e procura de tarefas

Daniel Kucheryavenko

Ano Letivo 2025/2026

Índice

Índice	2
Índice de Imagens	4
Índice de Tabelas	4
Agradecimentos	7
Resumo	7
Introdução	8
1. Enquadramento do Projeto	9
1.1. Identificação do Projeto	9
1.2. Problema a Resolver	9
1.3. Objetivos	9
1.4. Público-Alvo	10
1.5. Metodologia de Desenvolvimento	10
2. Desenvolvimento do Projeto	11
2.1. Planeamento	11
2.2. Evolução da DB	11
Primeira Versão	11
Segunda Versão	12
Terceira Versão	12
2.3. Páginas de login, registo e perfil.	13
2.4. Estrutura de Ficheiros	14
2.5. Fluxo da Aplicação	15
2.6. Arquitetura	16
2.7. Requisitos Funcionais e Não Funcionais	16
2.8. Instalação e Execução	17
2.9. Evolução do Projeto por Fases	17
3. Ferramentas Utilizadas	34
4. Linguagens e Tecnologias Utilizadas	35
5. Base de Dados	36
5.1. Modelo Relacional	36
5.2. Descrição das Tabelas	36
5.3. Relações Entre Tabelas	37
5.4. Integridade e Consistência dos Dados	39
5.5. Conexão a base de dados	39
6. Interface e Identidade Visual	40
7. Funcionalidades do Utilizador	42
7.1. Iniciar Sessão	42

7.2. Criar Conta	42
7.3. Página Inicial	43
7.4. Detalhe da Tarefa	44
7.5. Publicar Tarefa	44
7.6. Perfil	45
7.7. Minhas Tarefas	45
7.8. Tarefas a Fazer	46
7.9. Mensagens e Negociação.....	46
7.10. Concluir Tarefas e Avaliações	47
7.11. Avaliações e Confiança entre Utilizadores	49
8. Ferramentas de Administrador	50
8.1. Acesso ao Painel	50
8.2. Gestão de Categorias	50
8.3. Gestão de Tarefas	50
8.4. Gestão de Utilizadores.....	51
8.5. Operações e Organização do Painel	51
9. Segurança e Validação	52
10. Testes Realizados.....	53
10.1. Ambiente de Teste e Resultados	54
11. Dificuldades Encontradas.....	55
12. Melhorias Futuras.....	56
13. Conclusão	57
14. Manual Prático de Utilização	58
14.1. Criar uma conta	58
15.2. Iniciar sessão.....	59
15.3. Explorar tarefas	60
15.4. Utilizar pesquisa e filtros	61
15.5. Publicar uma tarefa.....	62
15.6. Consultar o detalhe e aceitar uma tarefa	63
15.7. Comunicar e negociar	64
15.8. Atualizar o perfil	65
15.9. Concluir e avaliar uma tarefa	66
15.10. Utilizar o painel de administração	66
16. Validação e Testes Detalhados	66
16.1. Testes de autenticação e sessão.....	66
16.2. Testes de formulários e validações.....	67
16.3. Testes de tarefas, estados e atribuições.....	67
16.4. Testes de mensagens, propostas e avaliações.....	67
16.5. Testes de administração, API e base de dados	67

17. Decisões Técnicas e Justificação	68
17.1. Organização por páginas e helpers	68
17.2. Base de dados relacional	68
17.3. Proteção de palavras-passe	68
17.4. Interface e responsividade	68
17.5. API e reutilização de dados	68
Anexos	69
Anexo A — Ligação à base de dados	69
Anexo B — Login e criação da sessão	70
Anexo C — Navegação modular no index.php	70
Anexo D — Controlo de permissões de administrador	71
Anexo E — Métricas do painel de administração	71
Anexo F — Seletor compacto de cidades	72
Anexo G — Segurança de palavras-passe	72
Anexo H — API de cidades	73
Anexo I — Controlo de permissões administrativas	73
Anexo J - Documentação dos Chats	74
Webgrafia	76

Índice de Imagens

Ilustração 1- Logotipo Tarefy	9
Ilustração 2- Diagrama ER da 1. ^a versão da base de dados	11
Ilustração 3- Diagrama ER da 2. ^a versão da base de dados	12
Ilustração 4- Diagrama ER da 3. ^a versão da base de dados	13
Ilustração 5- Fluxo geral de utilização da aplicação	15
Ilustração 6- Arquitetura geral do projeto	16
Ilustração 7- Template usado para o formulário de registo	18
Ilustração 8- Captura da navegação inicial	18
Ilustração 9- Captura da primeira página de publicação de tarefas	18
Ilustração 10- Troca de design da página de login	19
Ilustração 11- Página de login final	20
Ilustração 12- Captura do registo de contas da primeira versão	20
Ilustração 13- Captura do registo com espaço de confirmação da password	21
Ilustração 14- Novo design do formulário da criação de contas	21
Ilustração 15- Página de registo final	22
Ilustração 16- Template inicial a ganhar forma	22
Ilustração 17- Primeira página inicial com navbar e tarefas	23
Ilustração 18- Página principal final com cartões de tarefas e filtros	23
Ilustração 19- Primeira área de perfil do utilizador	24
Ilustração 20- Versão final da área de perfil	24
Ilustração 21- Primeira versão do formulário de publicação	25
Ilustração 22- Última versão do formulário de criação	25

Ilustração 23- Início do desenvolvimento dos detalhes das tarefas	26
Ilustração 24- Versão final dos detalhes das tarefas	26
Ilustração 25- Secção "Minhas tarefas"	27
Ilustração 26- Secção "Tarefas a fazer"	27
Ilustração 27- Secção "Concluir tarefas"	28
Ilustração 28- Primeira introdução da área de chat	28
Ilustração 29- Página da listagem dos "chats"	28
Ilustração 30- Página do "chat"	29
Ilustração 31- Desenvolvimento do painel de administração	30
Ilustração 32- Administração final.....	30
Ilustração 33- Administração final de utilizadores.....	31
Ilustração 34- Desenvolvimento interface do sistema de avaliações	31
Ilustração 35- Sistema final de avaliações.....	32
Ilustração 36- Visualização de perfis de terceiros.....	32
Ilustração 37- Demonstração do modo escuro	33
Ilustração 38- Footer final.....	33
Ilustração 39- Diagrama relacional da base de dados	36
Ilustração 40- Ficheiro .env	39
Ilustração 41- Página de início de sessão	42
Ilustração 42- Página de criação de conta.....	43
Ilustração 43- Página inicial e pesquisa de tarefas.....	43
Ilustração 44- Página de detalhe de uma tarefa	44
Ilustração 45- Formulário de publicação de tarefas	45
Ilustração 46- Página de perfil do utilizador	45
Ilustração 47- Área de tarefas publicadas pelo utilizador.....	46
Ilustração 48- Área de tarefas aceites para executar.....	46
Ilustração 49- Página de mensagens e negociação	47
Ilustração 50- Página de conclusão de tarefas	47
Ilustração 51- "Avaliações Pendentes"	48
Ilustração 52- Página de avaliações.....	48
Ilustração 53- Criação de conta na plataforma Tarefy	58
Ilustração 54- Início de sessão na aplicação	59
Ilustração 55- Exploração de tarefas na página principal	60
Ilustração 56- Pesquisa e filtros disponíveis na página inicial.....	61
Ilustração 57- Formulário atual de publicação de tarefa	62
Ilustração 58- Detalhe de tarefa com ações de aceitação e contacto	63
Ilustração 59- Mensagens e negociação associadas a uma tarefa.....	64
Ilustração 60- Atualização de dados do perfil	65
Ilustração 61- Ligação à base de dados e leitura da configuração.	69
Ilustração 62- Validação de credenciais e criação da sessão.....	70
Ilustração 63- Navegação modular para as páginas da aplicação.....	70
Ilustração 64- Verificação de permissões e token de administração.....	71
Ilustração 65- Consultas usadas para calcular as métricas administrativas.....	71
Ilustração 66- Excerto do código do seletor compacto de cidades	72
Ilustração 67- Excerto do código de proteção de palavras-passe.....	72
Ilustração 68- Excerto do código da API de cidades.....	73
Ilustração 69- Excerto do código de controlo administrativo	73
Ilustração 70- Excerto de código da funcionalidade da conversa	75
Ilustração 71- Continuação do excerto de código da funcionalidade da conversa	75

Índice de Tabelas

Tabela 1- Estrutura de ficheiros do projeto.....	15
Tabela 3 - Ferramentas utilizadas	34

Tabela 4 - Linguagens e tecnologias usadas	35
Tabela 5 - Descrição das tabelas da base de dados	37
Tabela 6 - Relações principais da base de dados	38
Tabela 7 - Testes funcionais e validações realizadas.....	54

Agradecimentos

A realização desta Prova de Aptidão Profissional representa a conclusão de uma etapa importante do meu percurso escolar e técnico. Quero agradecer à escola por ter aberto este excelente curso e pelas grandes oportunidades que me ofereceu. Ao longo deste projeto tive oportunidade de aplicar conhecimentos adquiridos durante o curso, principalmente nas áreas de programação, bases de dados, estruturação de interfaces e resolução de problemas.

Agradeço especialmente aos professores Carlos Ramos, Nuno Veloso, Henrique Teixeira, Ana Rodrigues, Ana Mateus e Carla Silva que me acompanharam no desenvolvimento da PAP, pelo apoio, pelas orientações, sugestões e correções realizadas ao longo do processo, bem como pelos valiosos conhecimentos que adquiri. O apoio recebido ajudou-me a melhorar a organização do projeto, a identificar falhas e a procurar soluções mais adequadas.

Agradeço aos meus colegas, amigos e familiares que testaram a minha PAP, dando o seu feedback sobre a sua experiência de utilização, perceber se as funcionalidades eram claras para um utilizador normal e deteção de bugs.

Por fim quero agradecer à empresa “Softi9” e ao meu orientador de estágio do 11.º ano Gibson Santos que me deu preciosos conhecimentos que me ajudaram não só no desenvolvimento da PAP como em outros projetos.

Resumo

Este relatório descreve o desenvolvimento do Tarefy, uma plataforma web criada para aproximar pessoas que precisam de ajuda em tarefas do dia a dia de utilizadores disponíveis para as realizar. A aplicação permite criar conta, iniciar sessão, publicar tarefas, pesquisar serviços, aceitar trabalhos, trocar mensagens, negociar preços, concluir tarefas e deixar avaliações.

O projeto foi desenvolvido com PHP, MySQL/MariaDB, HTML, CSS e JavaScript, seguindo uma organização modular de páginas, helpers, estilos e endpoints. Ao longo do desenvolvimento, a base de dados evoluiu de um modelo simples para uma estrutura relacional com tarefas, atribuições, conversas, mensagens, reservas, avaliações e mecanismos de recuperação de palavra-passe.

Palavras-chave: Tarefy, PHP, MySQL, tarefas, serviços, base de dados relacional, avaliações e administração.

Introdução

O projeto desenvolvido chama-se Tarefy. Trata-se de um website pensado para facilitar a publicação, procura e realização de tarefas do dia a dia em troca de dinheiro. A ideia principal é criar uma plataforma onde uma pessoa possa publicar uma tarefa que precisa de ser feita e onde outro utilizador possa encontrar essa tarefa, aceitar a sua realização ou iniciar uma conversa para combinar detalhes.

A aplicação foi desenvolvida em PHP, com uma base de dados MySQL/MariaDB, e utiliza HTML, CSS e JavaScript no lado do cliente. A interface recorre a Bootstrap e Bootstrap Icons para garantir uma apresentação coerente e responsiva. A interatividade do browser, incluindo o seletor pesquisável de cidades e a alternância entre temas, foi implementada em JavaScript puro.

O Tarefy não é apenas uma página estática. O projeto tem autenticação, registo de utilizadores, pesquisa de tarefas, filtros por categoria e preço, publicação de tarefas, carregamento de imagem, perfil de utilizador, tarefas publicadas, tarefas aceites, conclusão de tarefas, mensagens, negociação de preço, avaliações entre utilizadores, perfis públicos, utilizadores verificados, endpoints móveis e uma área de administração.

1. Enquadramento do Projeto

1.1. Identificação do Projeto

Nome do projeto: Tarefy.

Tipo de projeto: website com ligação a base de dados.

Área: publicação, procura e gestão de tarefas/serviços.

Utilizadores principais: pessoas que precisam de ajuda numa tarefa e pessoas disponíveis para realizar tarefas.



Ilustração 1- Logotipo Tarefy

1.2. Problema a Resolver

No dia a dia existem muitas pequenas tarefas que algumas pessoas não conseguem realizar sozinhas, seja por falta de tempo, ferramentas, conhecimento técnico ou disponibilidade. Ao mesmo tempo, existem pessoas que podem realizar esses serviços e que precisam de uma forma simples de encontrar oportunidades.

O problema que o Tarefy procura resolver é a falta de uma plataforma simples e direta onde se possa publicar uma tarefa, indicar uma descrição, preço base e categoria, e depois permitir que outro utilizador aceite ou contacte o publicador. A plataforma funciona como ponto de ligação entre necessidade e disponibilidade.

1.3. Objetivos

- Criar uma plataforma funcional para publicar e procurar tarefas.
- Permitir o registo e início de sessão de utilizadores.
- Guardar informação numa base de dados relacional.
- Permitir pesquisa, filtros e consulta de detalhes das tarefas.
- Permitir que utilizadores aceitem tarefas para executar.
- Criar uma área pessoal para acompanhar tarefas publicadas e aceites.
- Implementar mensagens e negociação de preço.
- Permitir a conclusão de tarefas e a avaliação entre utilizadores.
- Disponibilizar endpoints para consulta de dados por uma aplicação móvel.
- Criar uma área de administração para gestão do site.
- Construir uma interface simples, organizada e responsiva.

1.4. Público-Alvo

O público-alvo do Tarefy é composto por utilizadores que querem publicar uma tarefa — como montar móveis, resolver problemas informáticos, transportar objetos, limpar espaços ou pedir explicações, etc...— e por utilizadores que querem realizar essas tarefas em troca de pagamento.

A aplicação foi pensada para ser simples e acessível, com menus diretos e formulários claros, para que possa ser usada por pessoas com diferentes níveis de experiência tecnológica.

1.5. Metodologia de Desenvolvimento

O desenvolvimento foi realizado de forma incremental. Em vez de tentar implementar todas as funcionalidades de uma vez, o projeto foi **dividido em etapas**: primeiro foi criada a **estrutura de dados e a autenticação**, depois foram desenvolvidas as **tarefas**, a **comunicação entre utilizadores**, as **avaliações** e, por fim, a **área de administração** e os **testes**. Esta forma de trabalhar permitiu testar cada parte antes de avançar para a seguinte.

Sempre que foi adicionada uma funcionalidade importante, foram verificados os respetivos fluxos de utilização e as relações na base de dados. Este processo ajudou a detetar erros de validação, problemas de estados das tarefas e inconsistências entre tabelas antes da fase final do projeto.

2. Desenvolvimento do Projeto

2.1. Planeamento

O desenvolvimento do projeto começou no dia 24 de maio de 2025 com a entrega do modelo ER e do script SQL de criação da base de dados. A ideia inicial estava mais próxima de um site de vendas, mas evoluiu para uma plataforma de tarefas, por ser uma solução mais útil para necessidades reais do dia a dia e para a possibilidade de ganhar algum dinheiro através de pequenos serviços. Foi definido um plano cronológico dividido em várias fases:

Fase 1: análise inicial, entrega do modelo ER e criação do primeiro script SQL.

Fase 2: criação da base de dados e ligação PHP/MySQL.

Fase 3: criação das páginas de login, registo e perfil.

Fase 4: escolha e adaptação do template visual.

Fase 5: publicação, listagem, pesquisa, filtros e detalhe das tarefas.

Fase 6: aceitação, conclusão de tarefas e áreas pessoais do utilizador.

Fase 7: sistema de mensagens e negociação de preço.

Fase 8: avaliações, perfis públicos e utilizadores verificados.

Fase 9: painel de administração, testes, organização final e melhoria do relatório.

2.2. Evolução da DB

Primeira Versão

A primeira versão da base de dados era bastante simples e não estava adaptada à evolução prevista para o projeto. Apesar de cumprir os requisitos iniciais, apresentava algumas limitações que dificultavam a implementação de novas funcionalidades e a organização da informação. Assim, foi necessário proceder à reformulação da sua estrutura, tornando-a mais eficiente, escalável e adequada às necessidades da PAP.

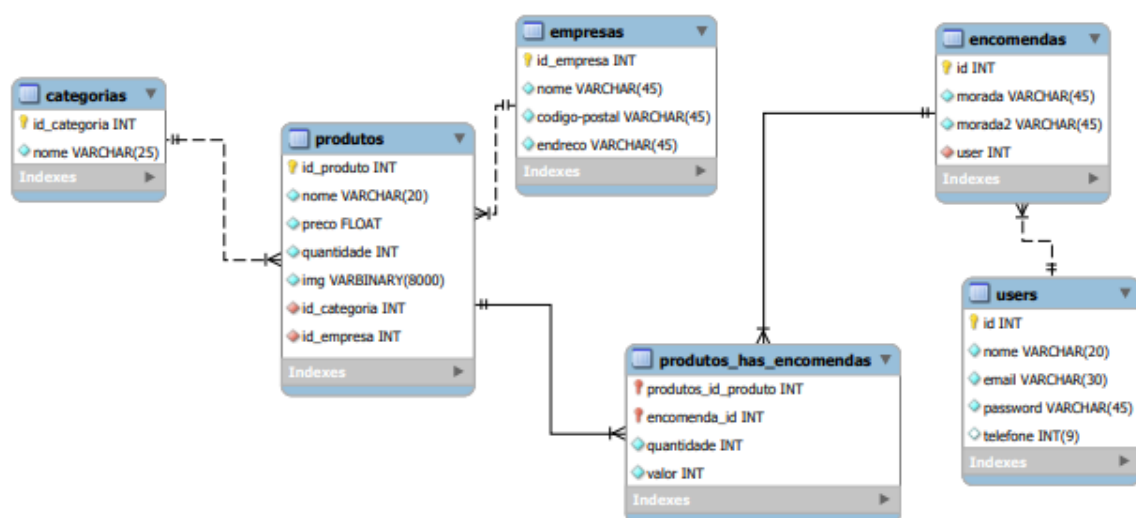


Ilustração 2- Diagrama ER da 1.ª versão da base de dados

Segunda Versão

A segunda versão da base de dados surgiu após uma reformulação da estrutura inicial, com o objetivo de adaptar melhor o sistema às necessidades reais da plataforma. Nesta versão foram substituídas as tabelas antigas por entidades mais adequadas ao funcionamento do projeto, como users, jobs, orders, reviews, categories e cities.

Uma das principais melhorias foi a organização das relações entre as tabelas através de chaves primárias e chaves estrangeiras. Desta forma, cada tarefa passou a estar associada a um utilizador responsável, cada utilizador passou a poder estar associado a uma cidade, e cada tarefa passou a pertencer a uma categoria. Também foi adicionada a possibilidade de criar “reservas” e avaliações, permitindo registar a conclusão de serviços e a opinião dos utilizadores envolvidos.

Esta versão tornou a base de dados mais coerente, escalável e preparada para novas funcionalidades. Ao contrário da primeira versão, que estava mais próxima de uma lógica de produtos e encomendas simples, a segunda versão passou a representar melhor o funcionamento da Tarefy, centrada na publicação de tarefas, contratação de serviços e avaliação entre utilizadores.

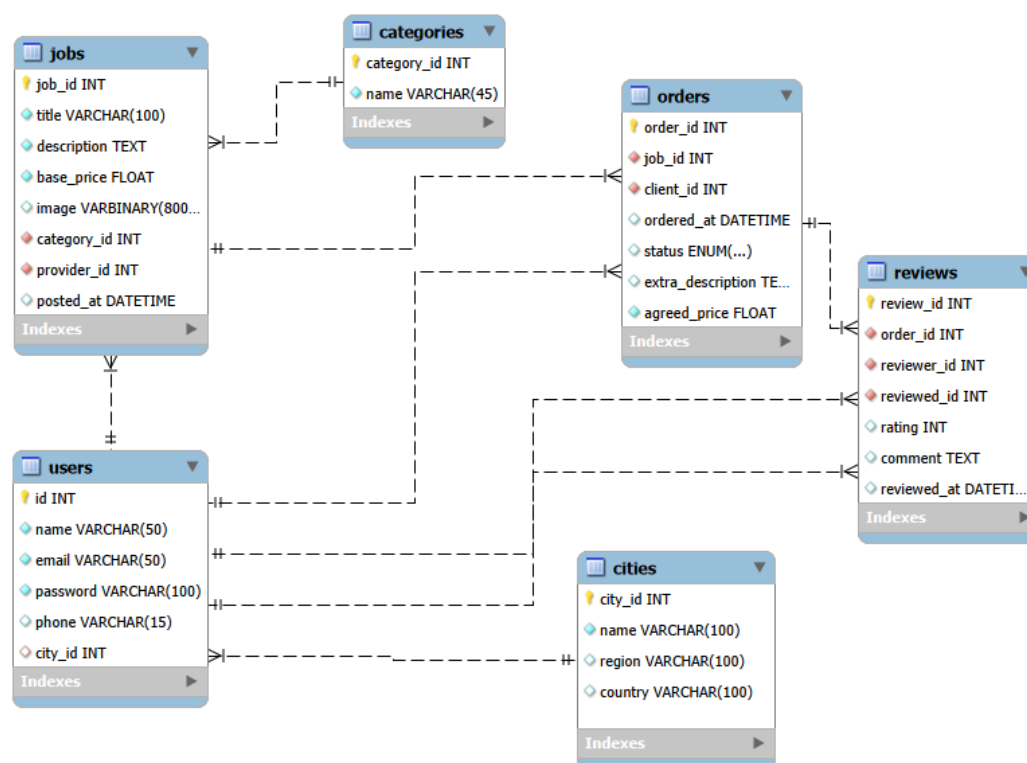


Ilustração 3- Diagrama ER da 2.ª versão da base de dados

Terceira Versão

Na terceira e última versão, a base de dados já está completa e funcional para a versão atual do Tarefy. Nesta fase, a estrutura deixou de estar focada apenas em utilizadores,

tarefas, encomendas e avaliações, e passou a incluir tabelas de apoio às conversas, às atribuições de tarefas, à recuperação de palavra-passe e ao funcionamento das avaliações.

Uma das principais alterações foi a criação das tabelas relacionadas com as mensagens. Com as tabelas `task_conversations` e `task_messages`, passou a ser possível guardar conversas entre utilizadores sobre uma determinada tarefa. Isto permite que quem publica uma tarefa e quem está interessado nela possam comunicar dentro da própria plataforma.

Também foi adicionada a tabela `task_assignments`, que guarda qual o utilizador que aceitou uma tarefa e o estado dessa execução. Assim, o sistema consegue distinguir tarefas ativas, tarefas em execução e tarefas concluídas.

Outra melhoria foi a criação da tabela `password_resets`, pensada para suportar uma futura recuperação de palavra-passe por token. Para as avaliações, as tabelas `orders` e `reviews` registam a relação entre quem publicou a tarefa, quem a executou e as classificações dadas por ambos.

Com estas alterações, a base de dados tornou-se mais organizada, mais completa e mais preparada para suportar o funcionamento real da aplicação.

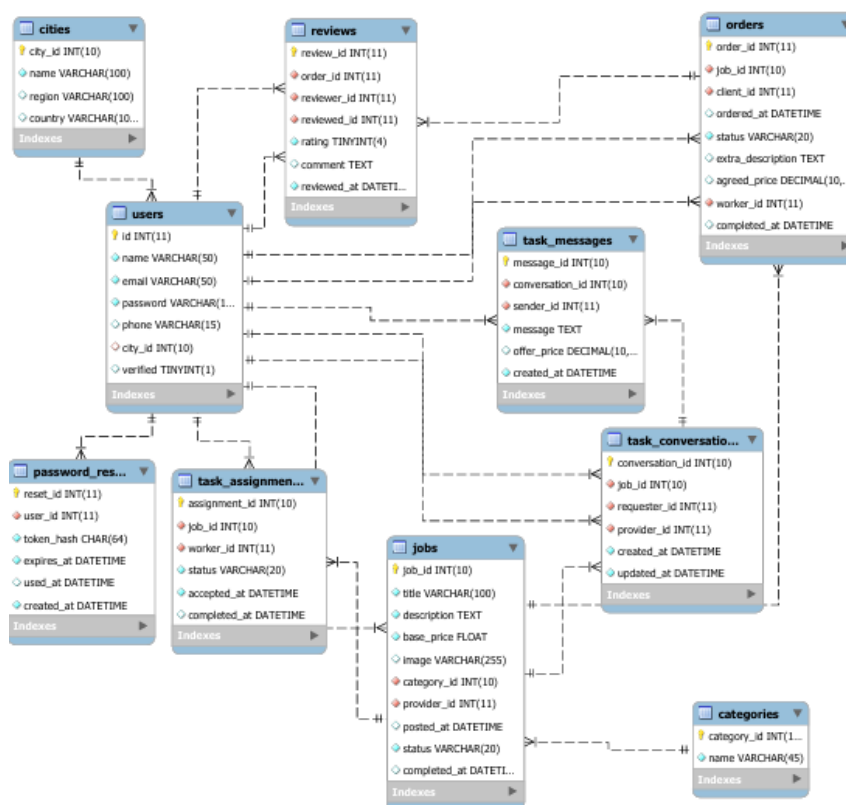


Ilustração 4- Diagrama ER da 3.ª versão da base de dados

2.3. Páginas de login, registo e perfil.

Esta etapa foi dedicada às páginas de autenticação e gestão de conta. A página `login.php` permite iniciar sessão através do nome de utilizador e da palavra-passe. A

validação é feita no servidor com prepared statements e com o helper `tarefyVerifyPassword()`, que suporta hashes atuais criados com `password_hash()` e ainda reconhece hashes SHA1 antigos para permitir migração gradual. Quando o login é bem-sucedido, o sistema guarda na sessão o identificador do utilizador, o nome, o email e a cidade. Caso a palavra-passe esteja num formato antigo ou precise de atualização, o sistema cria automaticamente um novo hash com `password_hash()`, sem obrigar o utilizador a alterar a palavra-passe naquele momento. A página de criação de conta valida campos obrigatórios, nome, email, telefone, cidade, confirmação de palavra-passe e regras mínimas de segurança. As cidades são carregadas a partir da base de dados e o frontend apresenta requisitos da palavra-passe em tempo real. A página de perfil permite alterar dados da conta e palavra-passe, enquanto os perfis públicos mostram estatísticas, avaliações e o estado de verificação do utilizador.

2.4. Estrutura de Ficheiros

O projeto está organizado por pastas, separando páginas PHP, estilos CSS, scripts JavaScript, imagens e ficheiros enviados pelo utilizador. Esta organização facilita a leitura e a manutenção do código.

Pasta / Ficheiro	Função
index.php	Entrada principal da aplicação autenticada. Carrega o layout, processa ações gerais e inclui as páginas através do parâmetro m.
login.php	Página e lógica de início de sessão. Usa prepared statements e verificação segura da palavra-passe.
logon.php	Página e lógica de criação de conta. Valida duplicações, cidade, telefone, email e palavra-passe.
config/db.php	Criação da ligação MySQL/MariaDB, leitura de variáveis de ambiente e definição de charset utf8mb4.
config/paths.php	Definição de caminhos partilhados, incluindo storage/uploads e compatibilidade com uploads antigos.
inc/pages/	Páginas internas: home, perfil, tarefas, chat, administração, avaliações e perfis públicos.
inc/helpers/	Funções reutilizáveis para chat, tarefas, avaliações, administração, passwords, imagens e utilizadores.
assets/css/	Folhas de estilo organizadas por base, layout, componentes, páginas, tema escuro e responsividade.
assets/js/	Scripts de interatividade, validação visual, seleção de

Pasta / Ficheiro	Função
	cidades, pré-visualização e tema.
assets/img/	Logótipos, favicon, imagem de fundo e placeholder de tarefa.
storage/uploads/	Imagens enviadas pelos utilizadores nas tarefas.
api/ e api/mobile/	Endpoints auxiliares, incluindo consulta de cidades e dados para aplicação móvel.
tests/	Testes e validações auxiliares, incluindo fluxo de avaliações.
docs/report/	Imagens, diagramas e recursos usados no relatório.

Tabela 1- Estrutura de ficheiros do projeto

2.5. Fluxo da Aplicação

O fluxo normal começa com a autenticação. Depois de iniciar sessão, o utilizador entra na página inicial, onde pode pesquisar tarefas, aplicar filtros, abrir detalhes, publicar uma tarefa nova ou aceder às áreas pessoais. Quando aceita uma tarefa, essa tarefa passa para a área Tarefas a Fazer e o estado muda para em execução. Depois de concluída, a tarefa pode gerar uma encomenda associada e ficar disponível para avaliação pelos utilizadores envolvidos. Quando existe necessidade de combinar preço ou detalhes, o utilizador pode iniciar uma conversa ligada à tarefa.



Ilustração 5- Fluxo geral de utilização da aplicação

2.6. Arquitetura

A arquitetura do projeto é composta por uma aplicação web em PHP, uma base de dados MySQL/MariaDB, ficheiros estáticos de interface e endpoints auxiliares. O browser envia pedidos HTTP para o servidor PHP, que processa as ações, consulta ou altera a base de dados e devolve páginas HTML ao utilizador. O código foi organizado em config, inc/pages, inc/helpers, assets, storage/uploads, api/mobile e tests, facilitando a manutenção e a separação de responsabilidades.

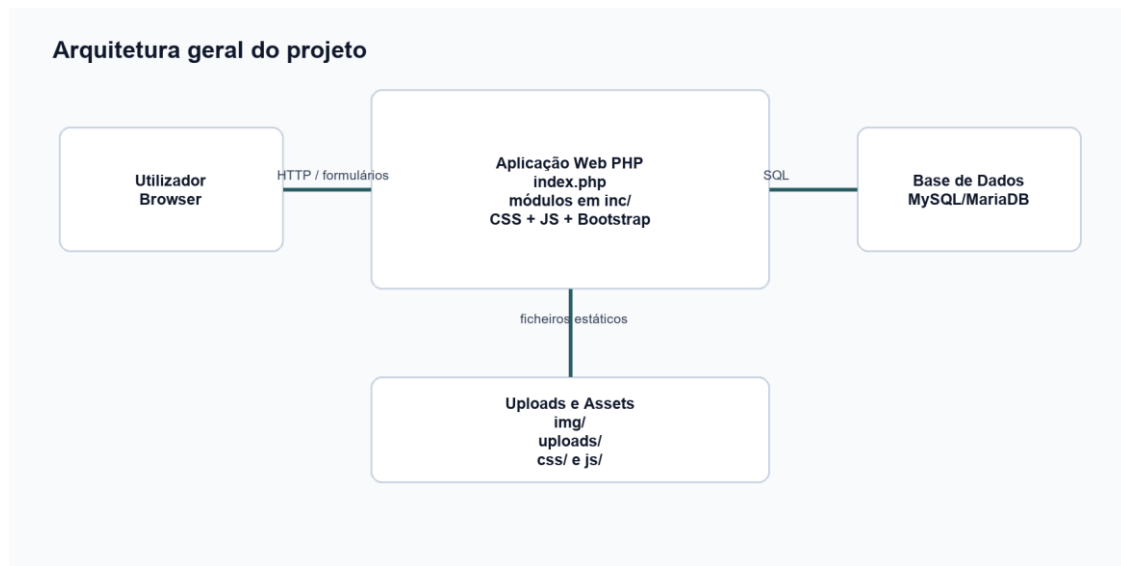


Ilustração 6- Arquitetura geral do projeto

2.7. Requisitos Funcionais e Não Funcionais

Os requisitos funcionais definem aquilo que a aplicação deve permitir fazer. No Tarefy, os principais requisitos implementados são:

- Registo, autenticação e edição do perfil de utilizador.
- Publicação, pesquisa, filtragem, consulta e aceitação de tarefas.
- Atribuição de uma tarefa a um executante e acompanhamento do respetivo estado.
- Mensagens entre utilizadores e proposta de preços durante a negociação.
- Conclusão de tarefas e avaliações mútuas após a realização do serviço.
- Gestão administrativa de categorias, tarefas, utilizadores e avaliações.
- Consulta de cidades através de endpoint JSON e suporte a endpoints para a aplicação móvel.

Para além das funcionalidades, foram considerados requisitos não funcionais: uma interface responsiva, validação de dados no servidor, proteção de páginas através de sessão, utilização de consultas preparadas, organização do código por responsabilidades e manutenção da consistência entre tabelas relacionadas.

2.8. Instalação e Execução

Para executar o projeto é necessário um servidor com PHP e uma base de dados MySQL ou MariaDB. Em ambiente local, pode ser utilizado Laragon ou XAMPP. Depois de importar o script SQL da base de dados, devem ser configuradas as variáveis de ambiente da ligação, ou os valores definidos no ficheiro config/db.php. A aplicação é acedida através do browser e o servidor PHP processa os pedidos recebidos.

A estrutura do projeto permite também iniciar um servidor de desenvolvimento com o PHP. Durante os testes foram verificados o carregamento das páginas, a validação de sintaxe e os fluxos principais de autenticação, tarefas, mensagens, avaliações, administração e API.

2.9. Evolução do Projeto por Fases

Esta secção foi reorganizada por áreas funcionais, para mostrar a evolução real do projeto e não apenas uma sequência de ficheiros. As versões analisadas foram pap_pc, PAP, PAP2.0, PAP2.0-projeto, PAP3.0, PAP4.0, PAP5.0, PAP6.0, PAP7.0 e PAPFinal, sendo esta última considerada a versão final apresentada na PAP.

A leitura da evolução é feita da versão mais simples para a versão final. Em cada área são indicadas as alterações principais, as decisões tomadas e o motivo pelo qual a solução final ficou mais adequada ao objetivo do Tarefy.

Evolução 1 - Conceito inicial, estrutura base e primeiras experiências: Nas primeiras versões, o projeto ainda não tinha a forma completa da plataforma Tarefy. Existiam páginas PHP simples, formulários de autenticação, experiências de navegação e uma ligação inicial à base de dados. Esta fase serviu para testar a criação de utilizadores, a separação entre ficheiros PHP, CSS e imagens, e a forma como a informação podia ser apresentada no browser.

Numa fase inicial foram usados modelos visuais e estruturas encontradas como referência na internet. Esses modelos ajudaram a perceber como montar um formulário e uma página com navegação, mas ainda não correspondiam ao resultado pretendido. A partir daí, o projeto foi sendo alterado para deixar de parecer um exemplo genérico e passar a ter uma identidade própria, centrada numa plataforma de tarefas.

A ideia também mudou tecnicamente. Antes de existir o fluxo completo de publicação e execução de tarefas, o projeto testava estruturas mais simples, próximas de produtos, locais, equipamentos ou registos básicos. Com a evolução da PAP, essas experiências foram substituídas por uma aplicação com utilizadores, categorias, cidades, tarefas, mensagens, avaliações e administração.

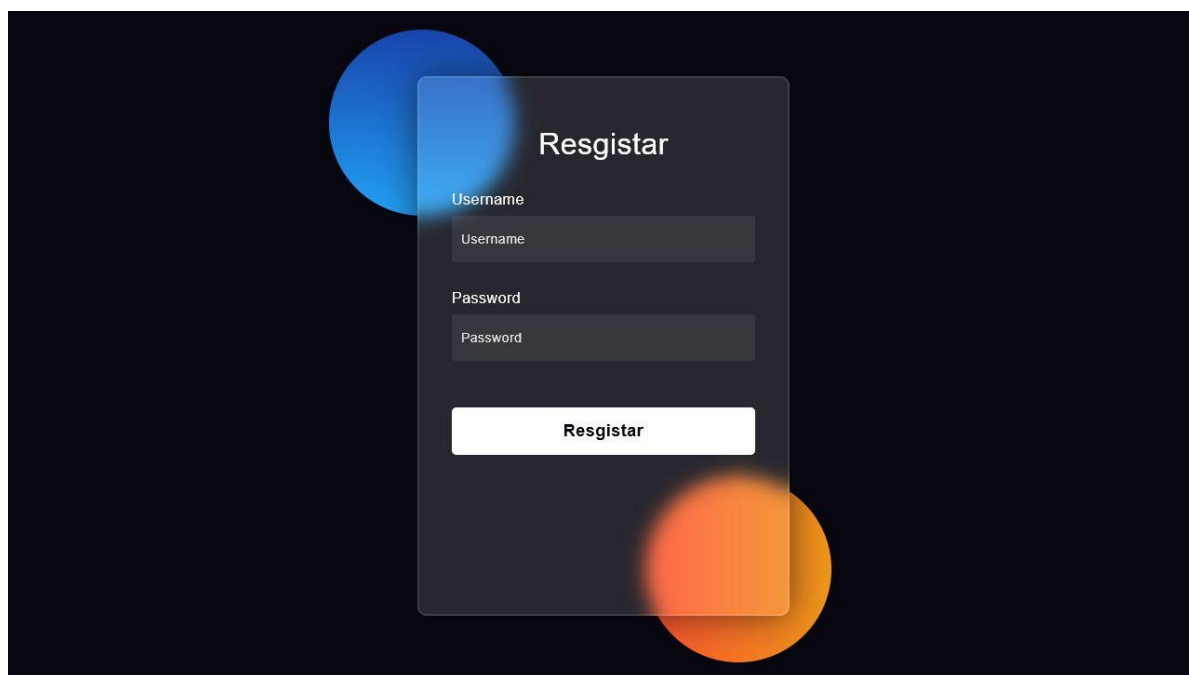


Ilustração 7- Template usado para o formulário de registo

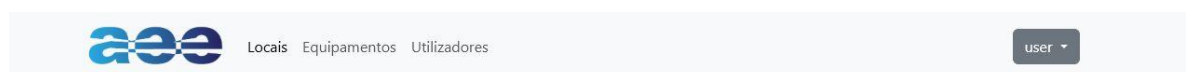


Ilustração 8- Captura da navegação inicial

Criar Tarefa

Nova Tarefa

Título

Descrição

Preço Base (€)

Categoria

Selecione a categoria

Imagem

Escolher ficheiro Nenhum ficheiro selecionado

Criar Tarefa

Ilustração 9- Captura da primeira página de publicação de tarefas

Evolução 2 - Login e autenticação: A página de login começou por seguir uma estrutura simples, próxima de um modelo visual já existente. Essa primeira solução permitia testar o formulário, mas ainda não transmitia a identidade. Nas versões seguintes, o login passou a ter uma composição mais cuidada, com separação entre a área de entrada e a área visual de boas-vindas.

Na versão final, o login ficou 100% funcional e integrado com a base de dados. O utilizador introduz o nome e a palavra-passe, o PHP valida os dados no servidor, cria a sessão e encaminha para a página principal. O visual final usa fundo com imagem, painel claro para o formulário, zona escura de destaque, cores preto/cinza/branco, sombras suaves, cantos arredondados e tipografia moderna definida nos ficheiros CSS. A autenticação passou ainda a suportar `password_hash()`, `password_verify()` e compatibilidade com hashes antigos.

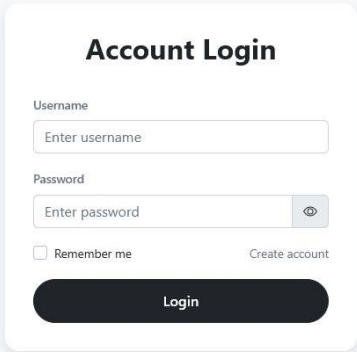
A screenshot of a simple login form titled "Account Login". It features two input fields: "Username" with the placeholder "Enter username" and "Password" with the placeholder "Enter password" and a toggle icon. Below the password field is a checkbox labeled "Remember me" and a link "Create account". At the bottom is a dark "Login" button.

Figura 1 - Login inicial da primeira versão

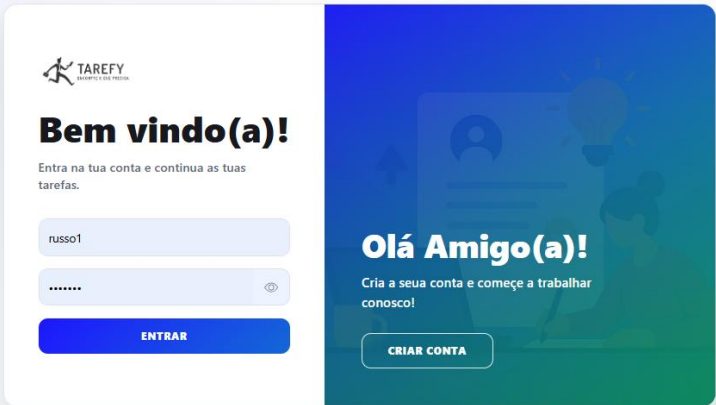
A screenshot of a modern login page for "TAREFY". The page is split into two sections. The left section is white and contains the TAREFY logo, the heading "Bem vindo(a)!", the subtext "Entra na tua conta e continua as tuas tarefas.", input fields for "russo1" and a masked password, and a blue "ENTRAR" button. The right section has a blue-to-green gradient background with a faint image of people working. It features the heading "Olá Amigo(a)!", the subtext "Cria a sua conta e comece a trabalhar conosco!", and a white "CRIAR CONTA" button.

Ilustração 10- Troca de design da página de login

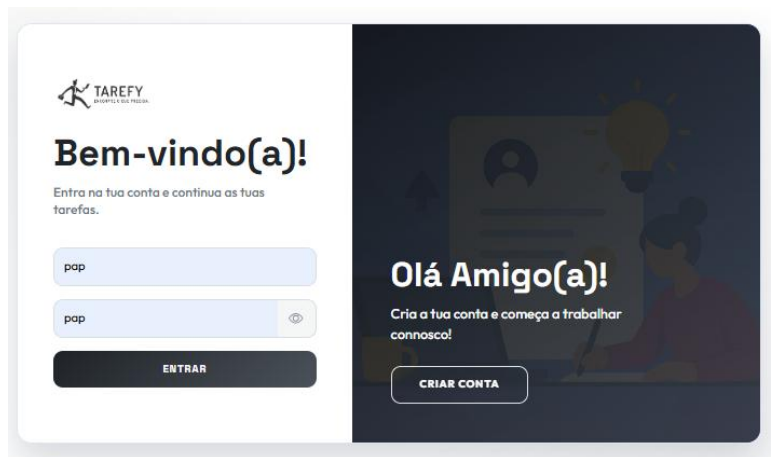


Ilustração 11- Página de login final

Evolução 3 - Criação de conta / login: A criação de conta também evoluiu por etapas. A primeira versão tinha poucos campos e servia apenas para confirmar que era possível guardar um utilizador. Depois foram acrescentados email, telefone, cidade e confirmação de palavra-passe, tornando o registo mais completo e seguro. Na versão final, o registo valida campos obrigatórios, confirma se a palavra-passe cumpre as regras mínimas, impede duplicação de nome, email e telefone, guarda a palavra-passe como hash e utiliza um seletor compacto de cidades. Esta solução tornou o formulário mais profissional, porque evita erros comuns e reduz o risco de contas incompletas ou repetidas.

Ilustração 12- Captura do registo de contas da primeira versão

Create Account

Username
Create username

Email
Insert your email

Phone number
Insert your number

City
Insert your city

Password
Create password

Confirm Password
Confirm password

Login

Create

Ilustração 13- Captura do registo com espaço de confirmação da password

Bem vindo(a) de volta!

Faça login para começar agora!

FAZER LOGIN

TAREFY

Crie uma conta

Nome de utilizador

Email

Número de telefone

Insira a sua cidade

Palavra-passe

- Mínimo 6 caracteres
- Letra maiúscula e minúscula
- Pelo menos um número
- Carácter especial (@\$!%*?&)

Confirmar senha

Ilustração 14- Novo design do formulário da criação de contas

Ilustração 15- Página de registo final

Evolução 4 - Página inicial, navbar, pesquisa e filtros: A página inicial passou de uma navegação simples para uma área principal onde o utilizador consegue descobrir tarefas disponíveis. As primeiras versões testavam cartões com imagem, título e preço, mas a organização visual ainda era limitada e os controlos ocupavam mais espaço do que o necessário.

Nas versões intermédias foram melhorados os cartões, a barra de navegação e a forma de apresentar tarefas. A pesquisa passou a permitir encontrar tarefas por texto, enquanto os filtros por categoria e preço ajudaram a reduzir a lista de resultados. Na versão final, a navbar dá acesso rápido ao início, perfil, mensagens, tarefas, avaliações, administração e criação de tarefas. A página inicial ficou mais limpa, com cartões mais legíveis, imagem de tarefa, preço, categoria e ligação para o detalhe.

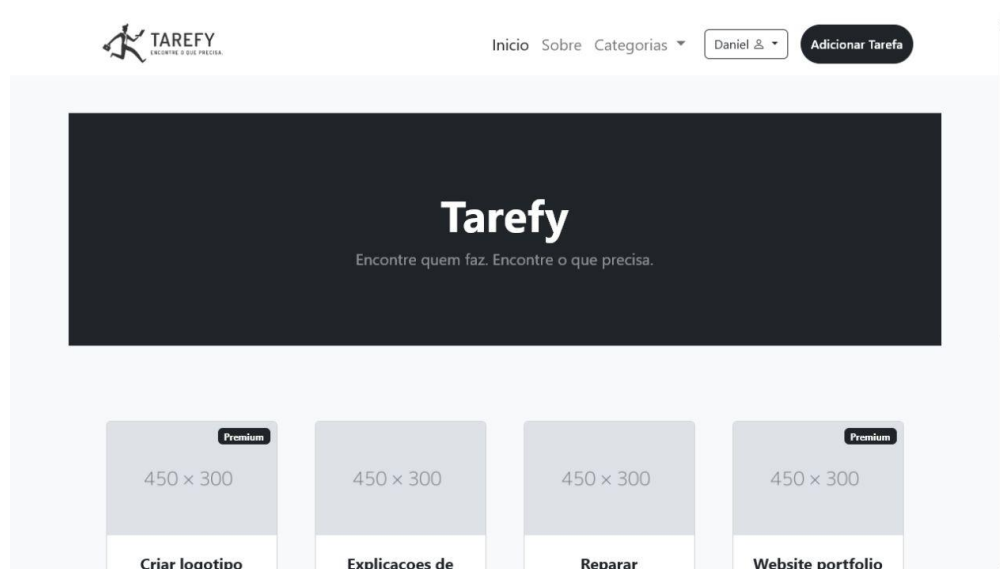


Ilustração 16- Template inicial a ganhar forma

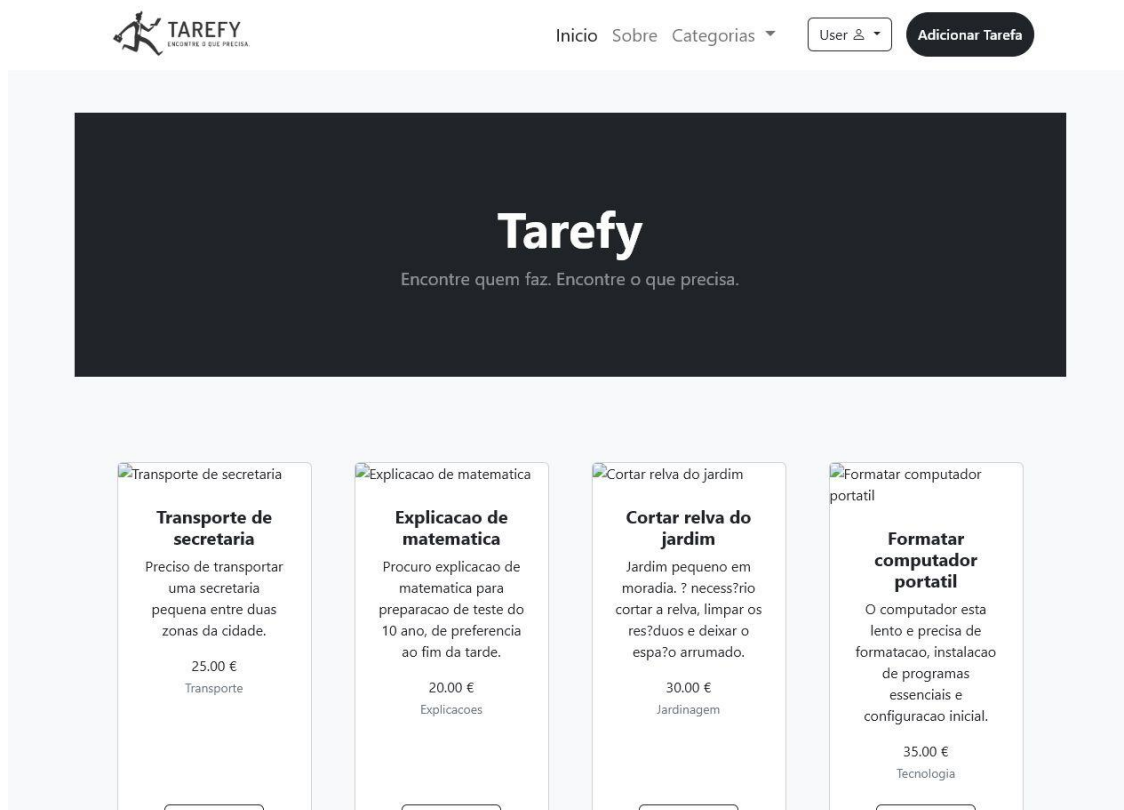


Ilustração 17- Primeira página inicial com navbar e tarefas

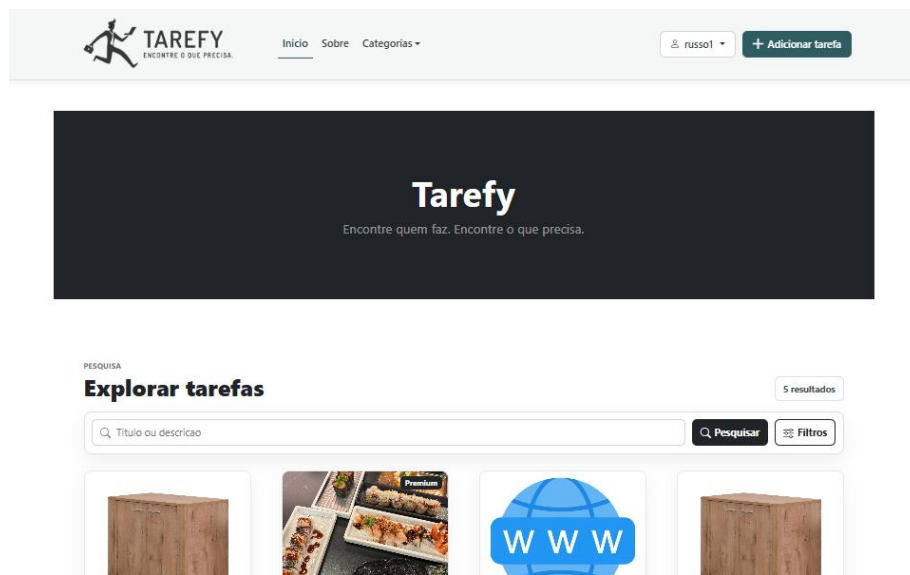


Ilustração 18- Página principal final com cartões de tarefas e filtros

Evolução 5 - Perfil do utilizador: O perfil começou por ser uma página simples para apresentar e editar dados básicos da conta. Ao longo das versões, foi melhorado para permitir atualizar nome, email e cidade sem obrigar o utilizador a alterar a palavra-passe.

Na versão final, a alteração da palavra-passe só acontece quando os campos próprios são preenchidos e confirmados. Isto evita alterações acidentais e melhora a segurança. O perfil ficou ligado ao restante sistema, porque a cidade, o nome e os dados do utilizador aparecem associados às tarefas, mensagens e avaliações.

The screenshot shows the 'Perfil' (Profile) page of the TAREFY application. At the top, there is a navigation bar with the TAREFY logo, links for 'Início', 'Sobre', and 'Categorias', a user dropdown menu showing 'Daniel', and a button 'Adicionar Tarefa'. The main content area is titled 'Perfil' and contains a section 'Dados do utilizador' (User data). This section includes input fields for 'Utilizador' (containing 'Daniel'), 'Email' (containing 'daniel@example.com'), 'City' (containing 'Porto'), 'Senha' (containing 'Insira a nova senha'), and 'Confirmação da Senha' (containing 'Confirmação'). A blue 'Guardar' (Save) button is at the bottom of the form. The footer of the page displays 'Copyright © Daniel Kucheryavenko'.

Ilustração 19- Primeira área de perfil do utilizador

The screenshot shows the updated 'Perfil' (Profile) page. The navigation bar now includes 'Início', 'Sobre', 'Categorias', and 'Admin', along with a dark theme toggle, a user dropdown menu showing 'pap', and a '+ Adicionar tarefa' button. The main content area is titled 'CONTA Perfil' and includes a subtitle 'Atualiza os teus dados de utilizador e define a tua localização.' and a 'Minhas tarefas' button. On the left, there is a user profile card for 'pap' with the email 'pap@email.com'. The main form area is divided into two sections: 'Dados pessoais' (Personal data) and 'Segurança' (Security). The 'Dados pessoais' section includes fields for 'Utilizador' (containing 'pap'), 'Email' (containing 'pap@email.com'), and a 'Cidade' dropdown menu (set to 'Aveiro'). The 'Segurança' section includes fields for 'Nova palavra-passe' (containing 'Nova palavra-passe') and 'Confirmar palavra-passe' (containing 'Confirmar palavra-passe'). A 'Guardar alterações' (Save changes) button is at the bottom right of the form.

Ilustração 20- Versão final da área de perfil

Evolução 6 - Criar tarefa: O formulário de criação de tarefa evoluiu de uma página inicial com campos essenciais para um formulário mais completo e validado. No início, o

objetivo era apenas guardar uma tarefa com título, descrição, preço e imagem. Depois foram acrescentadas categorias, validações e melhor organização visual. Na versão final, a tarefa fica associada ao utilizador autenticado, a imagem é validada antes de ser guardada e a categoria vem da base de dados. Esta evolução tornou o processo mais fiável, porque a tarefa deixa de ser apenas um registo isolado e passa a fazer parte de um fluxo completo da plataforma.

TAREFY
ENCONTRE O QUE PRECISA

Início Sobre Categorias Daniel Adicionar Tarefa

Criar Nova Tarefa

Título
Insira o título da tarefa

Descrição
Descreva a tarefa...

Preço Base (€)

Categoria
Escolha uma categoria

Imagem
Escolher ficheiro Nenhum ficheiro seleccionado

Criar Tarefa

Ilustração 21- Primeira versão do formulário de publicação

TAREFY
ENCONTRE O QUE PRECISA

Início Sobre Categorias Admin

Admin Adicionar tarefa

PUBLICAÇÃO

Publicar tarefa

Descreve o serviço, escolhe uma categoria e define um preço base.

Minhas tarefas

Detalhes da tarefa
Um título claro e uma descrição completa ajudam a tarefa a receber respostas melhores.

Título
Ex: Montar um armário

Descrição
Explica o que precisas, onde é e qualquer detalhe importante.

Preço base
€ 25

Categoria
Selecione uma categoria

Imagem
Opcional, mas ajuda a mostrar melhor o tipo de tarefa.

Escolher Imagem JPG
A imagem aparece em baixo antes de publicar.

+ Publicar tarefa

Boa publicação, resposta melhor.
Inclui o local, horário preferido, material necessário e o resultado esperado.

- ✓ Título específico
- ✓ Preço realista
- ✓ Descrição completa

Ilustração 22- Última versão do formulário de criação

Evolução 7 - Ver tarefa / detalhe da tarefa: A página de detalhe surgiu quando a listagem de tarefas deixou de ser suficiente. O utilizador precisava de consultar a descrição completa, o preço, a imagem, a categoria, o publicador e as ações disponíveis.

A versão final apresenta ações diferentes conforme o contexto: o dono não aceita a própria tarefa, outro utilizador pode aceitar ou contactar, e tarefas em execução ou concluídas apresentam estados adequados. Esta lógica tornou a página de detalhe numa peça central do funcionamento do Tarefy.

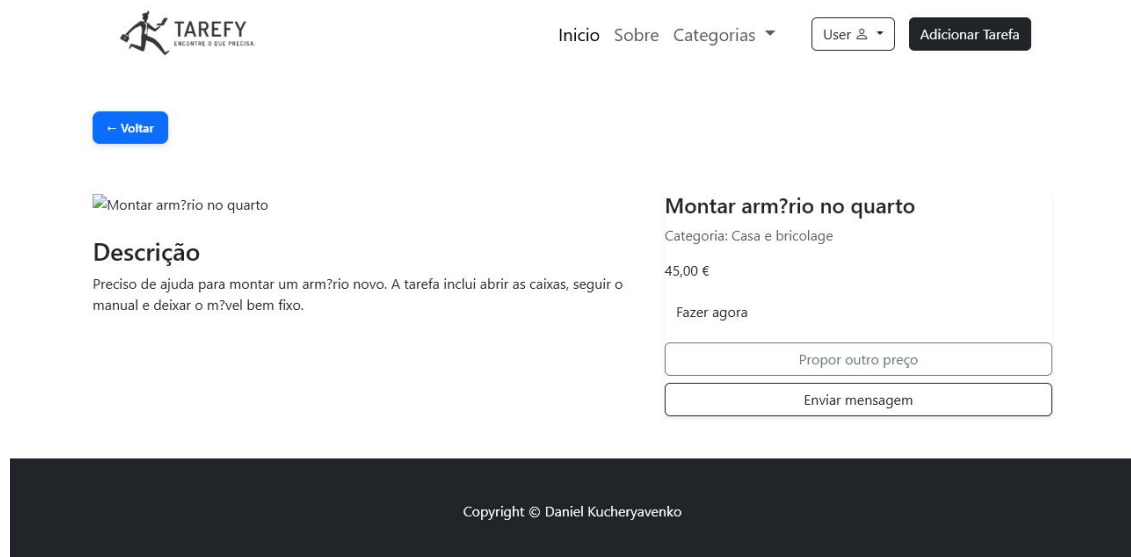


Ilustração 23- Início do desenvolvimento dos detalhes das tarefas

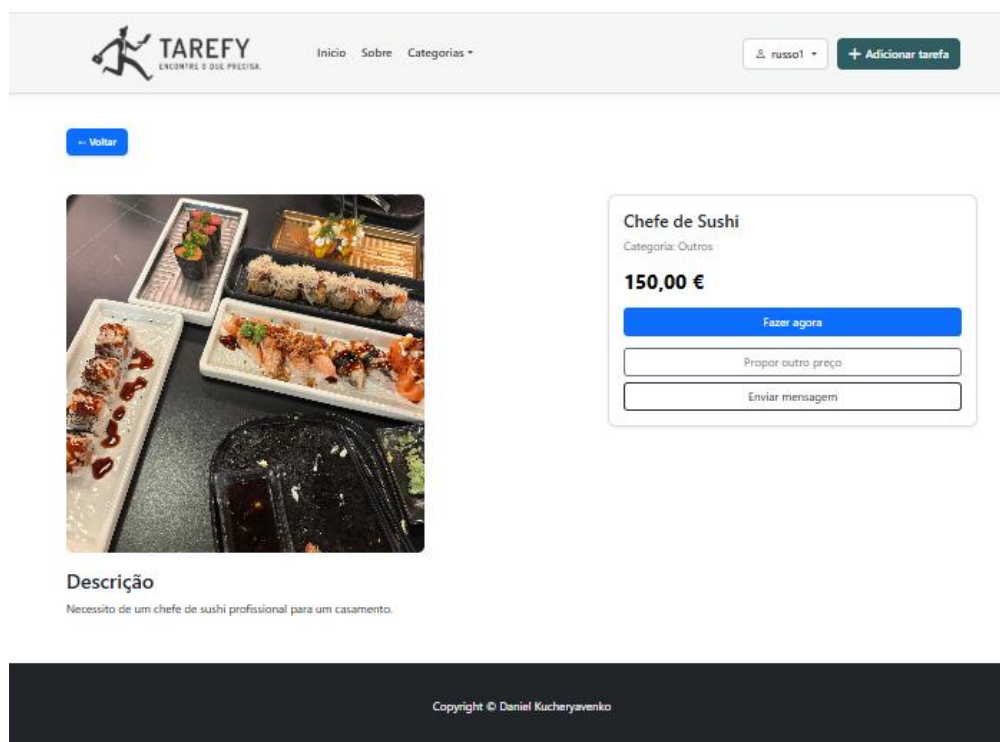
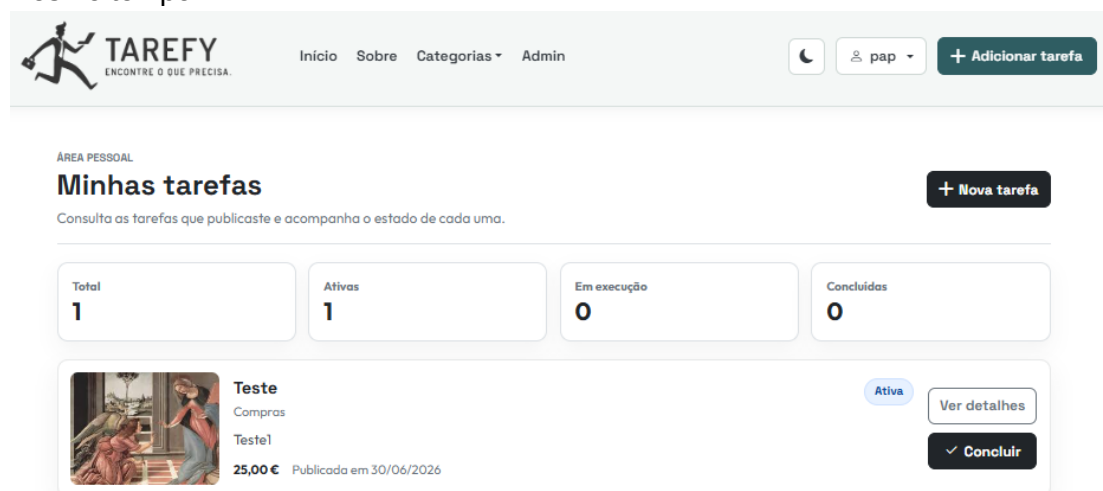


Ilustração 24- Versão final dos detalhes das tarefas

Evolução 8 - Minhas tarefas, tarefas a fazer e conclusão: Quando as tarefas passaram a ser aceites, foi necessário separar o que cada utilizador publica daquilo que aceita executar. Por isso surgiram as áreas de minhas tarefas, tarefas a fazer e tarefas concluídas.

Nas versões anteriores, a tarefa era sobretudo uma publicação. Nas versões mais avançadas, passou a ter ciclo de vida: ativa, em execução e concluída. A versão 4.0 introduziu ficheiros dedicados a minhas tarefas e conclusão, e as versões seguintes melhoraram o acompanhamento do estado. Na versão final, a aplicação atualiza as tabelas relacionadas e impede que a mesma tarefa seja aceite por várias pessoas ao mesmo tempo.



TAREFY
ENCONTRE O QUE PRECISA.

Início Sobre Categorias Admin

+ Adicionar tarefa

ÁREA PESSOAL

Minhas tarefas

+ Nova tarefa

Consulta as tarefas que publicaste e acompanha o estado de cada uma.

Total	Ativas	Em execução	Concluídas
1	1	0	0



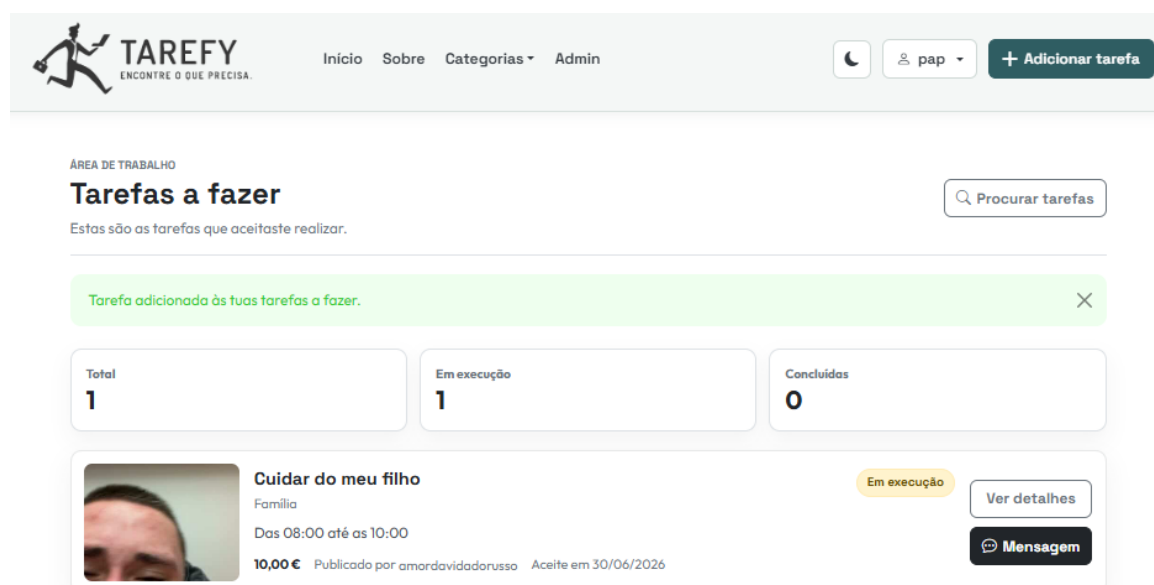
Teste
Compras
Teste1
25,00 € Publicada em 30/06/2026

Ativa

Ver detalhes

Concluir

Ilustração 25- Secção "Minhas tarefas"



TAREFY
ENCONTRE O QUE PRECISA.

Início Sobre Categorias Admin

+ Adicionar tarefa

ÁREA DE TRABALHO

Tarefas a fazer

Estas são as tarefas que aceitaste realizar.

Procurar tarefas

Tarefa adicionada às tuas tarefas a fazer.

Total	Em execução	Concluídas
1	1	0



Cuidar do meu filho
Família
Das 08:00 até as 10:00
10,00 € Publicada por amordavidadorusso Aceite em 30/06/2026

Em execução

Ver detalhes

Mensagem

Ilustração 26- Secção "Tarefas a fazer"

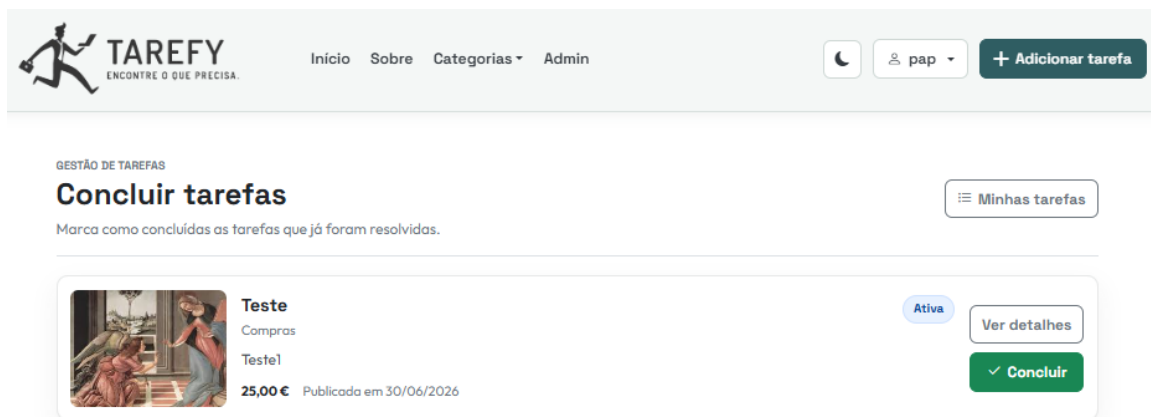


Ilustração 27- Secção "Concluir tarefas"

Evolução 9 - Mensagens e negociação: A comunicação foi acrescentada para que a plataforma não dependesse de contactos externos. A partir do detalhe de uma tarefa, os utilizadores podem criar uma conversa associada ao serviço e trocar mensagens. Nas versões intermédias surgiu o chat e a lógica de propostas. Na versão final, as mensagens ficam associadas à tarefa, ao remetente e ao destinatário, e podem incluir uma proposta de preço. Esta evolução tornou o Tarefy mais completo, porque permite combinar detalhes antes de concluir o trabalho.

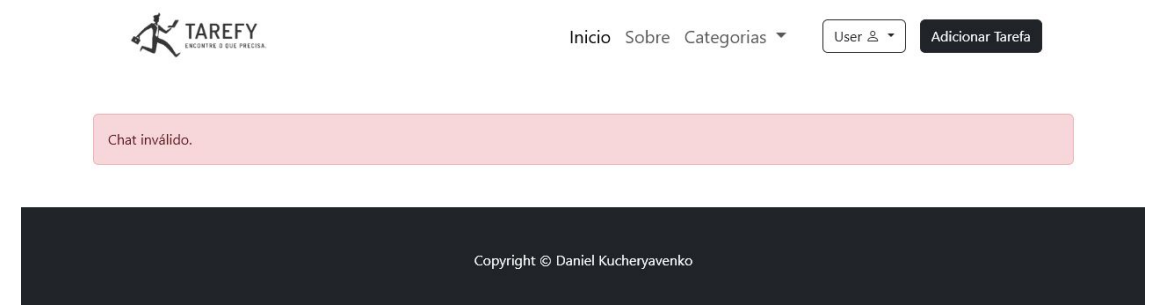


Ilustração 28- Primeira introdução da área de chat

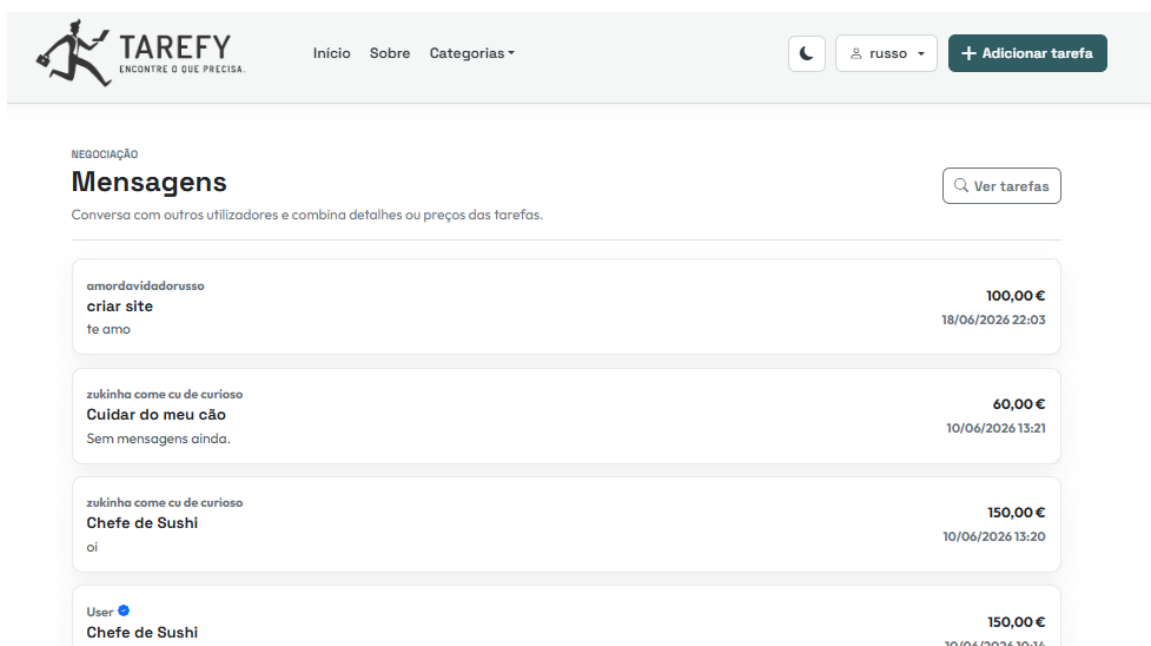


Ilustração 29- Página da listagem dos "chats"

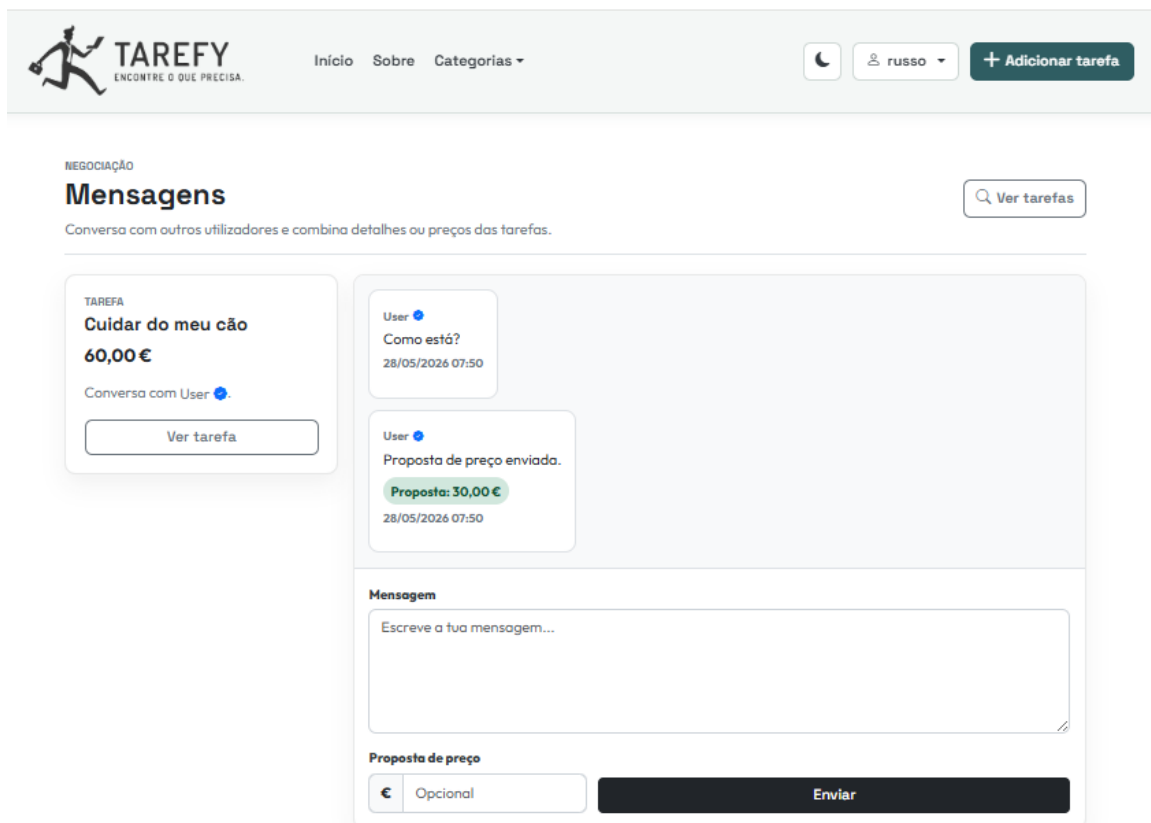


Ilustração 30- Página do "chat"

Evolução 10 - Administração do site: A administração foi introduzida quando a aplicação já tinha informação suficiente para precisar de gestão. A primeira versão do painel reunia contadores e ações sobre categorias, tarefas e utilizadores, mas ainda estava muito dependente de listas grandes e de uma organização inicial. Na versão final, a administração ficou separada das páginas normais e protegida por verificação de permissões e token de sessão. O administrador consegue consultar métricas, gerir categorias, alterar estados de tarefas, verificar utilizadores, remover avaliações e repor palavras-passe. Esta área tornou o projeto mais profissional, porque permite gerir a plataforma sem editar diretamente a base de dados.

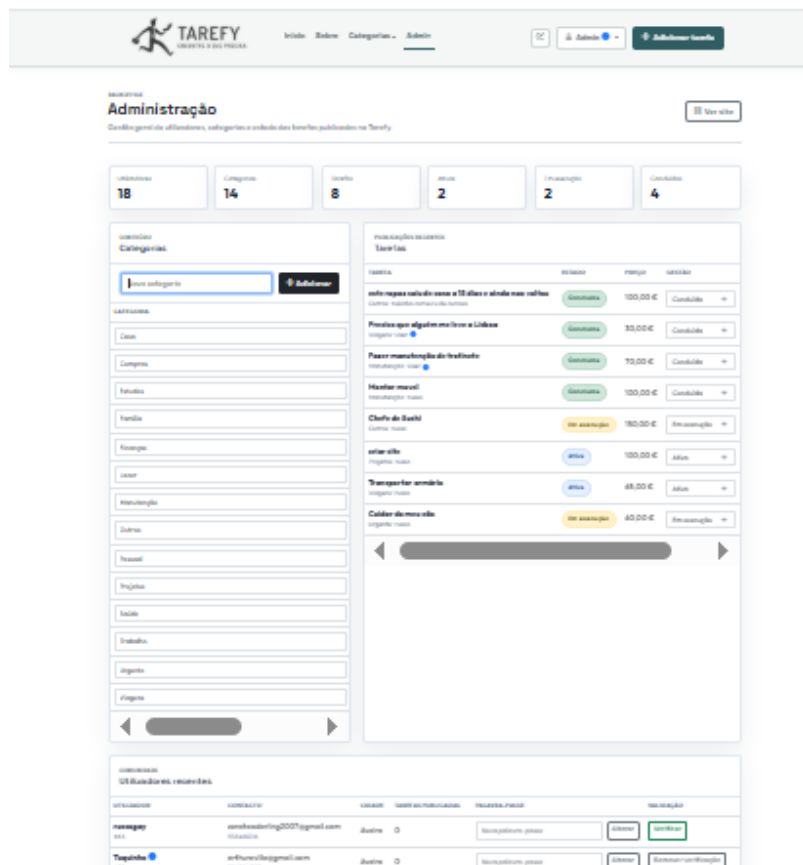


Ilustração 31- Desenvolvimento do painel de administração

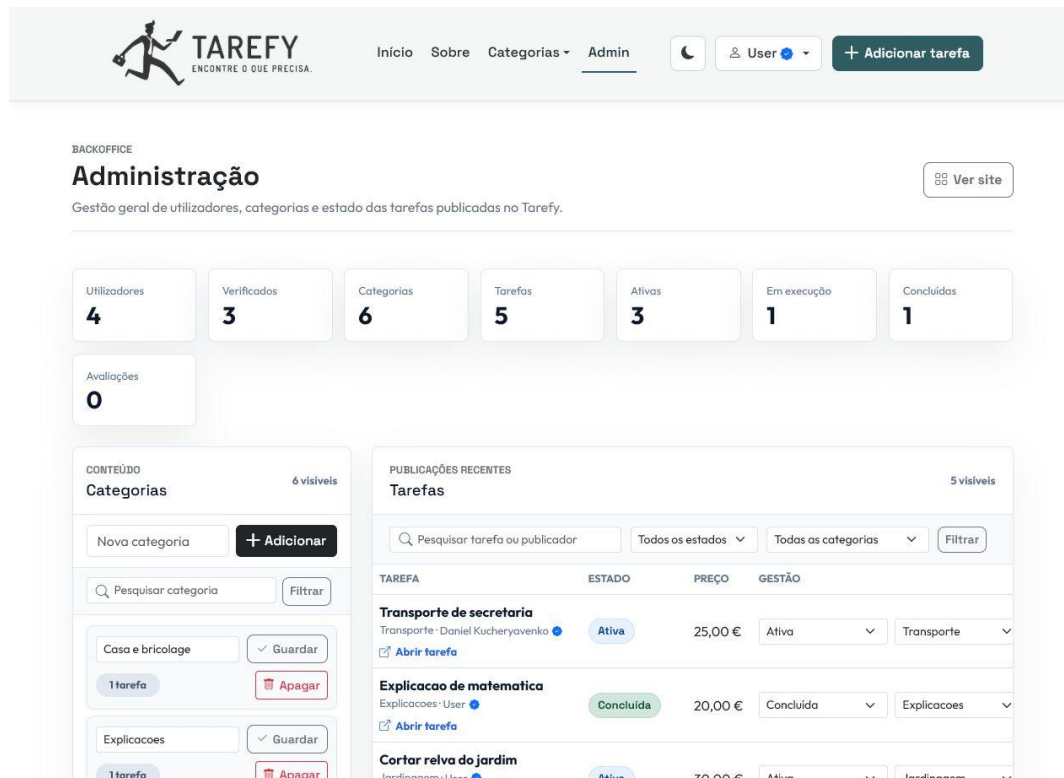


Ilustração 32- Administração final

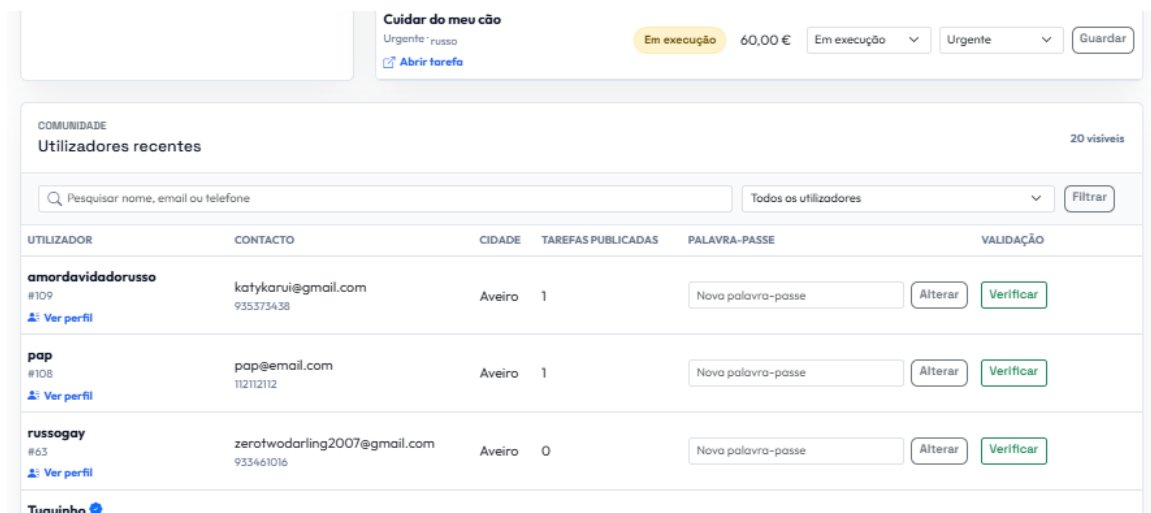


Ilustração 33- Administração final de utilizadores

Evolução 11 - Avaliações, perfis públicos e confiança: As avaliações foram acrescentadas numa fase avançada, quando o fluxo de tarefas já permitia aceitar e concluir serviços. Esta funcionalidade surgiu para criar confiança entre utilizadores, tal como acontece em plataformas reais de prestação de serviços.

Na versão final, a avaliação só fica disponível depois de a tarefa estar concluída. Cada participante pode avaliar a outra pessoa com estrelas e comentário, e essa informação é apresentada no perfil público. A introdução dos perfis públicos e das avaliações tornou o Tarefy mais transparente, porque um utilizador pode consultar a reputação de outra pessoa antes de aceitar ou publicar um serviço.

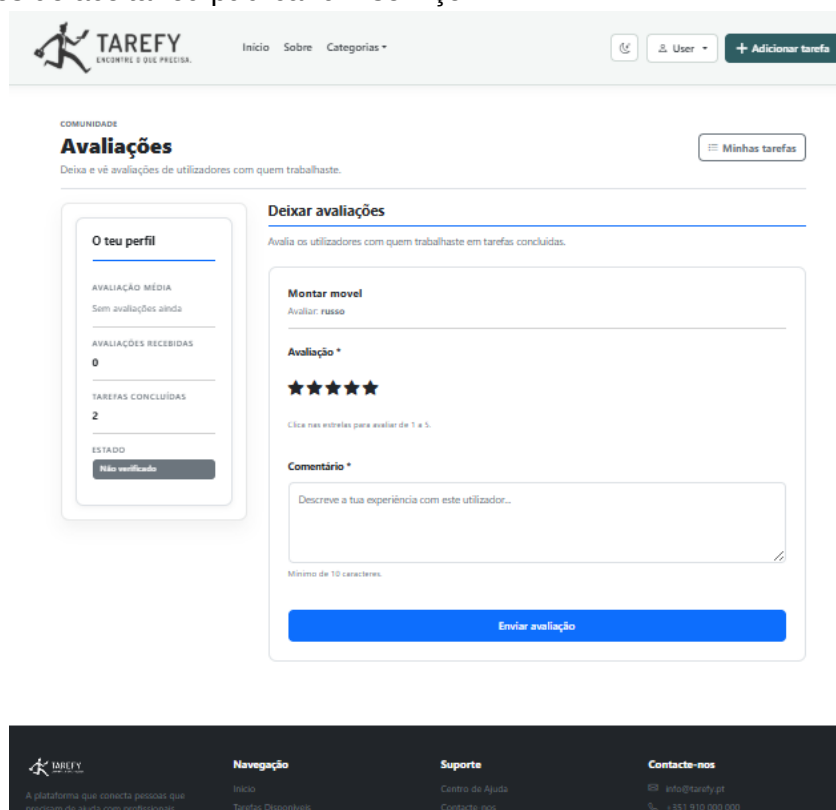


Ilustração 34- Desenvolvimento interface do sistema de avaliações

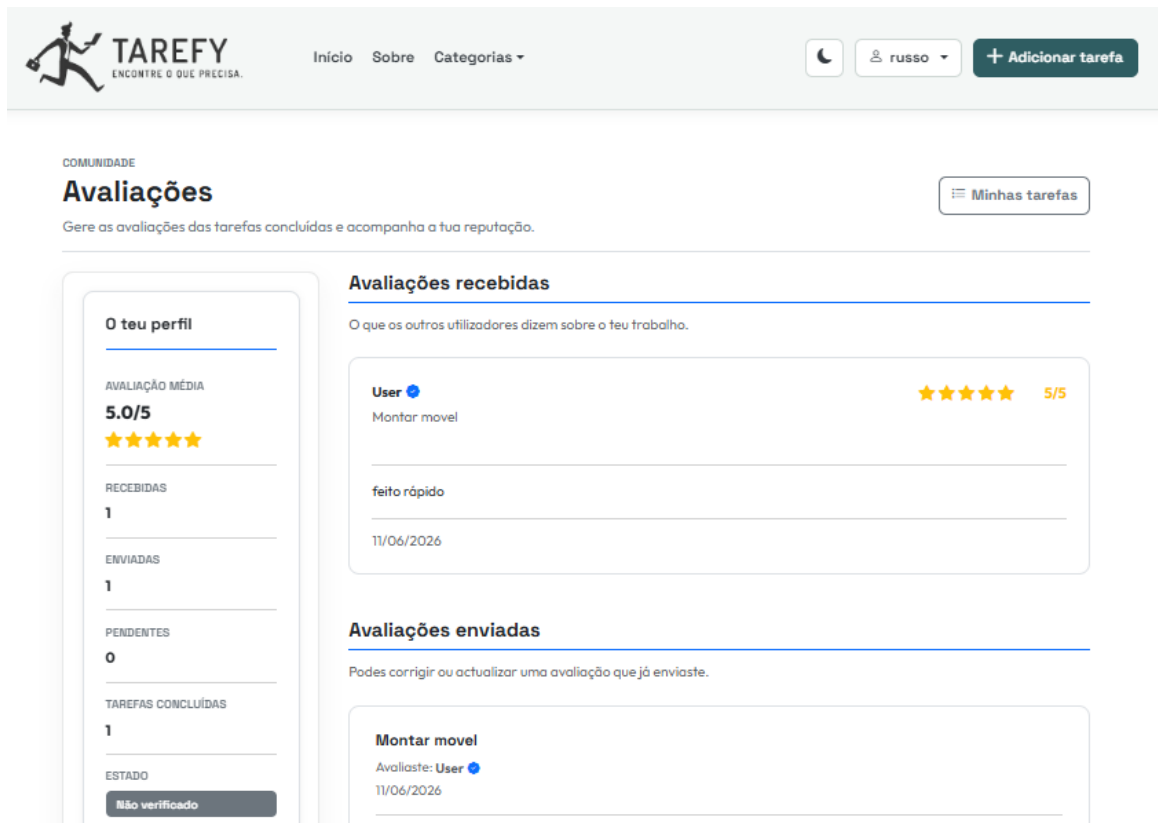


Ilustração 35- Sistema final de avaliações

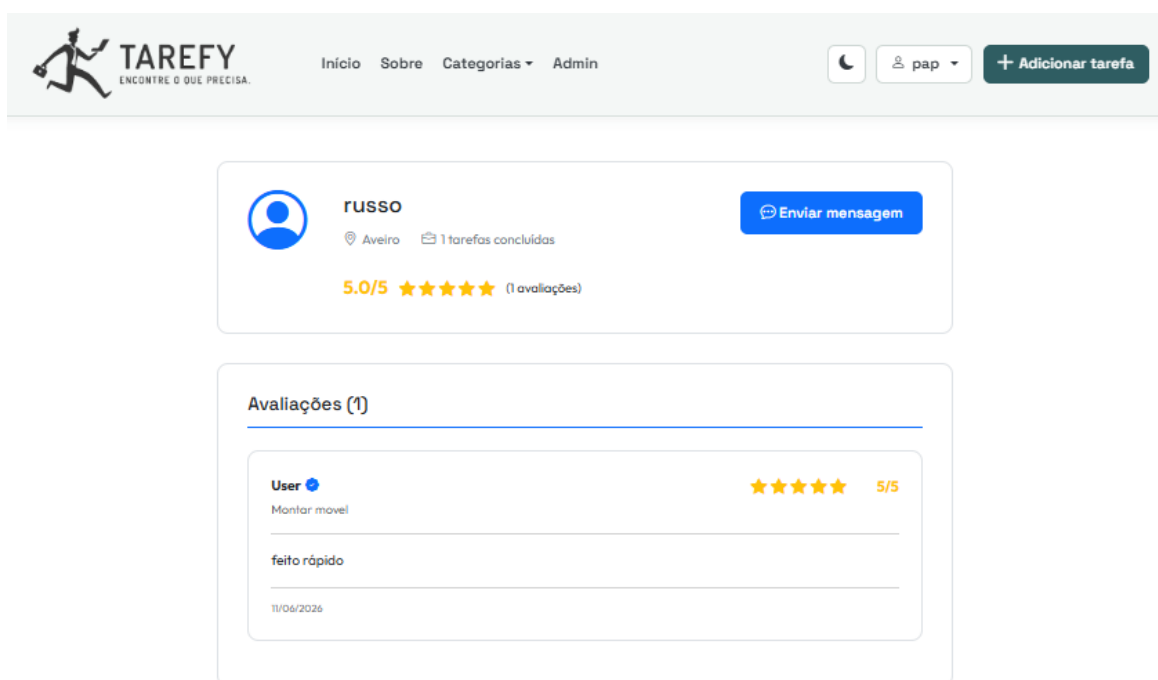


Ilustração 36- Visualização de perfis de terceiros

Evolução 12 - Organização final, API, estilos e testes: Nas últimas versões, o projeto deixou de estar organizado apenas por ficheiros soltos e passou a ter uma estrutura mais modular. A versão 7.0 já mostra a separação em assets, config, api/mobile,

inc/helpers e inc/pages. A versão final PAPFinal consolida esta organização e junta as funcionalidades principais num projeto mais estável.

Foram separados ficheiros de configuração, páginas, helpers, estilos, scripts e recursos. Esta organização facilita a manutenção, porque cada área tem responsabilidade própria: autenticação, tarefas, chat, avaliações, administração, imagens, cidades e base de dados. Também foram mantidos endpoints para a aplicação móvel e uma API de cidades em JSON.

A versão final inclui ainda modo escuro, seletor compacto de cidades, validação de imagens, prepared statements, proteção por sessão, tokens em ações sensíveis, testes manuais aos fluxos principais e teste automatizado ao fluxo de avaliações. No conjunto, a evolução demonstra a passagem de um protótipo simples para uma aplicação web funcional, organizada e adequada à apresentação final da PAP.

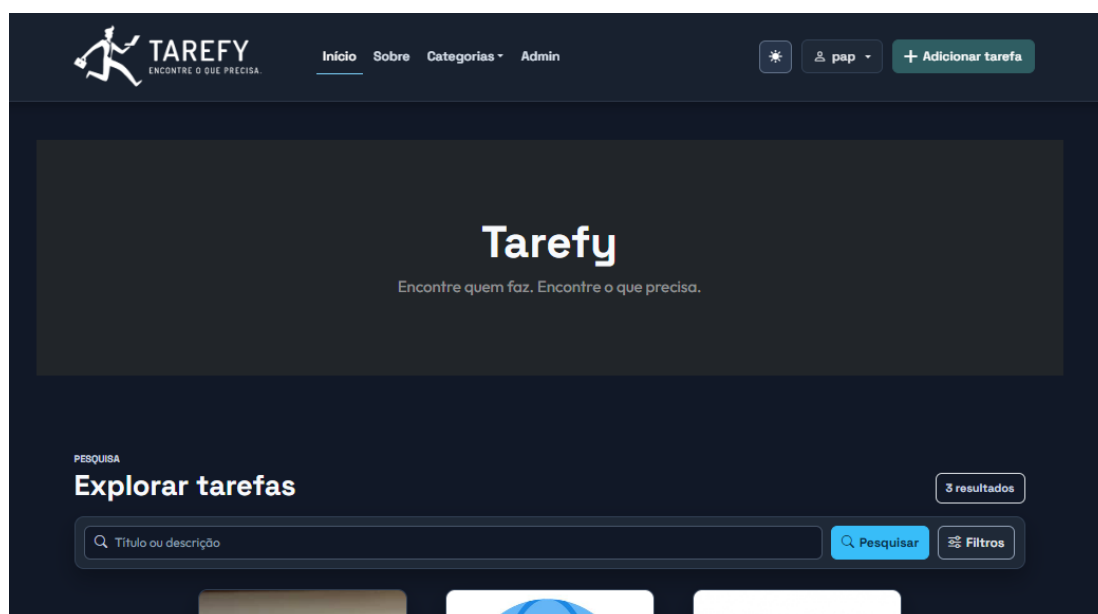


Ilustração 37- Demonstração do modo escuro

Evolução 13- Melhoria do rodapé: Numa fase posterior foi introduzido um rodapé mais completo, para dar um acabamento mais profissional ao Tarefy. Esta área passou a incluir o logótipo, uma breve descrição da plataforma, ícones de redes sociais, links de navegação, suporte e contactos.

O rodapé utiliza fundo escuro, texto claro e uma organização por colunas, facilitando a leitura e o acesso rápido a páginas importantes. Também inclui direitos de autor e a identificação do autor do projeto.

Esta melhoria tornou a interface mais completa, organizada e próxima de uma plataforma real.

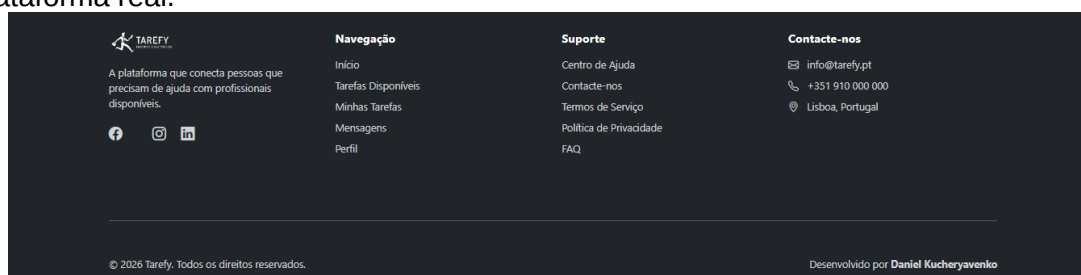


Ilustração 38- Footer final

3. Ferramentas Utilizadas

Ferramenta	Tipo	Utilização
Visual Studio Code	Editor de código	Usado para escrever, organizar e testar os ficheiros PHP, CSS, JS e SQL.
Laragon/XAMPP	Ambiente local	Usado para executar PHP, servidor local e MySQL/MariaDB durante os testes.
MySQL/MariaDB	Base de dados	Guarda utilizadores, cidades, categorias, tarefas, mensagens, encomendas e avaliações.
phpMyAdmin/SQL	Gestão de BD	Apoio à consulta, importação, alteração e verificação das tabelas.
Chrome	Browser	Usado para testar a aplicação, responsividade e gerar capturas das páginas.
PHP CLI	Validação técnica	Usado para validar sintaxe dos ficheiros com php -l e apoiar testes automatizados.
MySQL Workbench	Modelação de base de dados	Utilizado para criar diagramas ER, consultar relações e validar a evolução do modelo relacional.

Tabela 2 - Ferramentas utilizadas

4. Linguagens e Tecnologias Utilizadas

Tecnologia	Função no projeto
PHP	Back-end, sessões, formulários, uploads, consultas SQL, validações e regras da aplicação.
MySQL/MariaDB	Armazenamento relacional dos dados do projeto.
HTML5	Estrutura das páginas, formulários, navegação e conteúdos.
CSS3	Estilização, layout, responsividade, cartões, formulários e modo escuro.
JavaScript	Validações visuais, pré-visualização de imagens, seleção de cidades e alternância de tema.
Bootstrap	Componentes visuais, grelha, botões, navbar, modais e alertas.
Bootstrap Icons	Ícones usados em botões, navegação, estados e utilizadores verificados.
JavaScript (DOM e Fetch API)	Interatividade da interface, seletor compacto de cidades, validações visuais e comunicação com endpoints JSON.
MySQLi prepared statements	Execução de consultas parametrizadas para reduzir riscos de injeção SQL.

Tabela 3 - Linguagens e tecnologias usadas

5. Base de Dados

5.1. Modelo Relacional

A base de dados do Tarefy foi criada em MySQL/MariaDB. O modelo relacional é importante porque permite separar a informação por entidades e relacionar essas entidades através de chaves primárias, chaves estrangeiras e índices. A tabela jobs é a tabela central, pois representa as tarefas publicadas. À sua volta existem tabelas para utilizadores, cidades, categorias, atribuições de tarefas, conversas, mensagens, encomendas, avaliações e recuperação de palavra-passe.

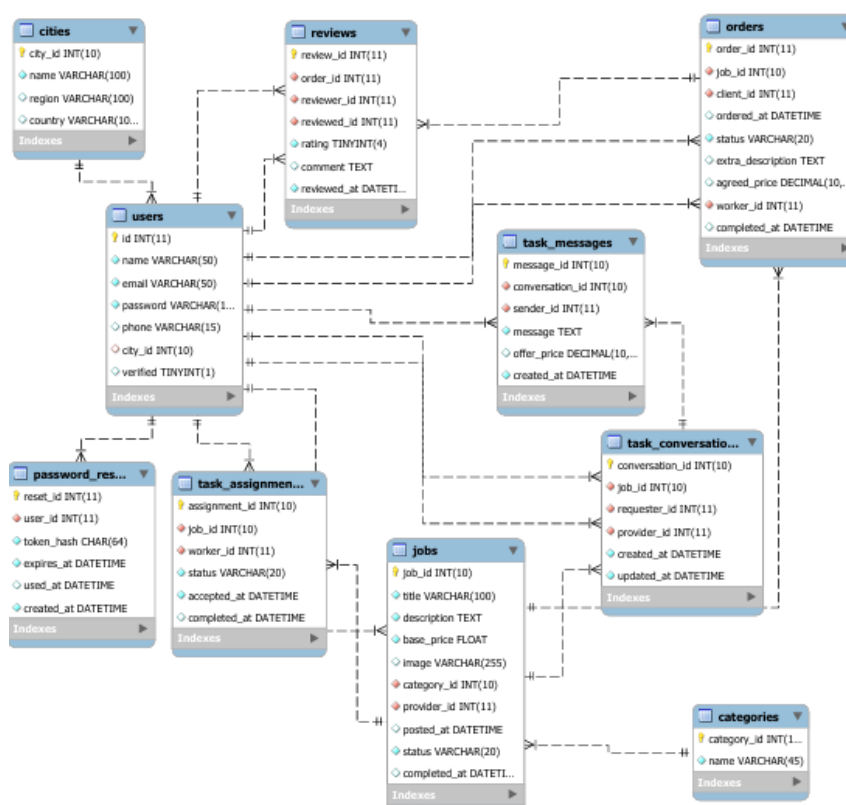


Ilustração 39- Diagrama relacional da base de dados

5.2. Descrição das Tabelas

Tabela	Objetivo	Campos principais
users	Utilizadores registados na plataforma.	id, name, email, password, phone, city_id, verified
cities	Cidades disponíveis para associar ao perfil.	city_id, name, region, country
categories	Categorias utilizadas para organizar tarefas.	category_id, name
jobs	Tabela principal das tarefas publicadas.	job_id, title, description, base_price, image, category_id, provider_id,

Tabela	Objetivo	Campos principais
		posted_at, status, completed_at
task_assignments	Registro do utilizador que aceitou executar uma tarefa.	assignment_id, job_id, worker_id, status, accepted_at, completed_at
task_conversations	Conversas associadas a uma tarefa entre publicador e interessado/executante.	conversation_id, job_id, requester_id, provider_id, created_at, updated_at
task_messages	Mensagens enviadas dentro de uma conversa.	message_id, conversation_id, sender_id, message, offer_price, created_at
orders	Registro da contratação usada para acompanhar conclusão e avaliações.	order_id, job_id, client_id, worker_id, ordered_at, status, agreed_price, completed_at
reviews	Avaliações feitas entre utilizadores depois da conclusão de uma tarefa.	review_id, order_id, reviewer_id, reviewed_id, rating, comment, reviewed_at
password_resets	Estrutura preparada para recuperação de palavra-passe por token.	reset_id, user_id, token_hash, expires_at, used_at, created_at

Tabela 4 - Descrição das tabelas da base de dados

5.3. Relações Entre Tabelas

Origem	Destino	Descrição
users.city_id	cities.city_id	Cada utilizador pode ter uma cidade associada.
jobs.category_id	categories.category_id	Cada tarefa pertence a uma categoria.
jobs.provider_id	users.id	Cada tarefa é publicada por um utilizador.
task_assignments.job_id	jobs.job_id	Cada atribuição está ligada a uma tarefa.
task_assignments.worker_id	users.id	Cada atribuição pertence ao utilizador que aceitou executar a tarefa.

Origem	Destino	Descrição
task_conversations.job_id	jobs.job_id	Cada conversa pertence a uma tarefa.
task_conversations.requester_id	users.id	Identifica o utilizador que iniciou a conversa.
task_conversations.provider_id	users.id	Identifica o utilizador que publicou a tarefa.
task_messages.conversation_id	task_conversations.conversation_id	Cada mensagem pertence a uma conversa.
task_messages.sender_id	users.id	Cada mensagem é enviada por um utilizador.
orders.job_id	jobs.job_id	Cada encomenda/contratação está ligada a uma tarefa.
orders.client_id	users.id	Identifica o utilizador que publicou ou contratou o serviço.
orders.worker_id	users.id	Identifica o utilizador que executou a tarefa.
reviews.order_id	orders.order_id	Cada avaliação pertence a uma contratação concluída.
reviews.reviewer_id	users.id	Identifica o utilizador que fez a avaliação.
reviews.reviewed_id	users.id	Identifica o utilizador que recebeu a avaliação.
password_resets.user_id	users.id	Cada pedido de recuperação pertence a um utilizador.

Tabela 5 - Relações principais da base de dados

5.4. Integridade e Consistência dos Dados

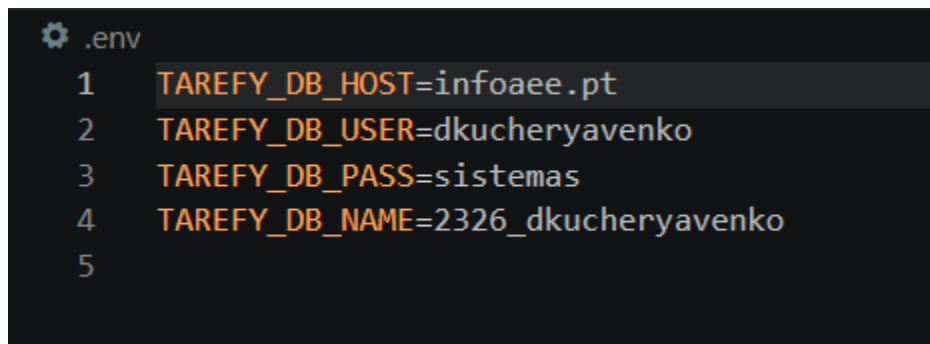
A integridade da base de dados é garantida através de chaves primárias, índices, chaves estrangeiras e regras de unicidade. Por exemplo, uma tarefa só pode referenciar uma categoria e um utilizador existentes, e uma mensagem só pode pertencer a uma conversa válida. Estas relações evitam a criação de registos sem ligação ao resto da aplicação.

Foram também definidas ações de eliminação em cascata nas tabelas de apoio às tarefas e conversas. Assim, quando uma tarefa ou conversa deixa de existir, os respetivos registos dependentes não ficam isolados na base de dados. No estado atual, o modelo contém dez tabelas e dezassete relações principais, representadas no diagrama relacional apresentado anteriormente.

5.5. Conexão a base de dados

Para melhorar a organização e evitar que os dados de acesso à base de dados fiquem escritos diretamente no código PHP, foi criado um ficheiro `.env`. Neste ficheiro são guardadas as informações necessárias para a ligação, como o servidor, o utilizador, a palavra-passe e o nome da base de dados.

Desta forma, o ficheiro `config/db.php` apenas lê estas variáveis e cria a ligação com MySQL, tornando o projeto mais organizado e fácil de configurar noutros ambientes.



```
1 TAREFY_DB_HOST=infoaee.pt
2 TAREFY_DB_USER=dkucheryavenko
3 TAREFY_DB_PASS=sistemas
4 TAREFY_DB_NAME=2326_dkucheryavenko
5
```

Ilustração 40- Ficheiro .env

6. Interface e Identidade Visual

A interface foi criada com uma aparência limpa e organizada. A navegação principal fica no topo e permite acesso rápido ao início, categorias, administração, perfil, mensagens, tarefas a fazer, tarefas publicadas, avaliações e criação de tarefas. A aplicação usa cartões, listas, formulários, badges de estado e ícones para apresentar a informação de forma clara.

O projeto também inclui modo escuro, ativado por JavaScript e guardado no localStorage. Esta funcionalidade melhora a personalização da experiência do utilizador e troca automaticamente o logótipo para manter boa leitura nos dois temas.

6.1. Estilos utilizados no projeto

- **Tipografia:** A versão final utiliza as fontes Outfit para o texto geral e Space Grotesk para títulos, elementos de interface e zonas de maior destaque. Estas fontes foram definidas no ficheiro `assets/css/base/typography.css`, com alternativas de sistema para garantir compatibilidade caso as fontes externas não carreguem.
- **Paleta principal:** A identidade visual usa uma base clara com branco, cinzas suaves e tons escuros. As cores mais presentes são `#ffffff`, `#f4f7f6`, `#d9dee3`, `#212529`, `#343a40`, `#1f2328`, `#2b3035` e `#4b5259`. Em zonas de destaque aparecem também tons verde-azulados como `#2f5d62` e `#203a43`, usados sobretudo na navegação e em elementos de identidade.
- **Cores de estado:** Para ações e feedback foram usados padrões familiares ao utilizador: azul para ações principais ou ligações (`#0d6efd`), verde para sucesso (`#198754`), vermelho para erro ou perigo (`#dc3545`), amarelo para avisos e estrelas de avaliação (`#ffc107`), e cinza para ações secundárias (`#6c757d`).
- **Login e criação de conta:** Os ficheiros `assets/css/login.css` e `assets/css/logon.css` definem painéis divididos, fundo visual, caixas de formulário claras, botões escuros, inputs com foco destacado, mensagens de erro/sucesso e comportamento responsivo.
- **Navbar e footer:** A navegação foi separada em `assets/css/layout/navbar.css` e o rodapé em `assets/css/layout/footer.css`. Estes estilos controlam o logótipo, espaçamento, links, botões, estado ativo, sombras, limites e adaptação a diferentes larguras de ecrã.
- **Formulários e tarefas:** Os ficheiros `assets/css/pages/forms.css`, `assets/css/pages/tasks.css` e `assets/css/taskDetail.css` controlam formulários, cartões de tarefas, áreas vazias, tabelas, botões, detalhe de tarefa, imagens e organização das páginas de criação, consulta e acompanhamento de tarefas.
- **Administração, chat e perfil:** A administração usa `assets/css/pages/admin.css`, as mensagens usam `assets/css/pages/chat.css` e os perfis públicos usam `assets/css/pages/user-profile.css`. Cada ficheiro concentra os estilos da respetiva área, evitando que o CSS principal fique demasiado grande.
- **Avaliações:** O ficheiro `assets/css/reviews.css` define cartões, estrelas, formulários de avaliação, histórico de classificações e adaptação ao modo escuro. A cor amarela das estrelas ajuda o utilizador a reconhecer rapidamente o sistema de classificação.

- **Componentes reutilizáveis:** Foram criados estilos separados para componentes como o seletor de cidades, links de utilizador, modal de informação e bloco do publicador da tarefa. Estes ficheiros ficam em `assets/css/components` e tornam o código visual mais fácil de reutilizar.
- **Modo escuro e responsividade:** O ficheiro `assets/css/themes/dark.css` define a versão escura da interface com fundos `#111827` e `#1f2937`, texto claro e realces azulados. O ficheiro `assets/css/responsive.css` ajusta logótipo, navegação, grelhas, formulários e cartões para ecrãs menores.
- **Biblioteca visual:** O projeto usa Bootstrap para estrutura, grelha, botões, tabelas e componentes base, e Bootstrap Icons para ícones visuais. O CSS próprio foi usado por cima dessas bases para adaptar o estilo à identidade do Tarefy.

7. Funcionalidades do Utilizador

7.1. Iniciar Sessão

A página de início de sessão permite ao utilizador entrar na aplicação através do nome de utilizador e palavra-passe. A validação é feita em PHP, consultando a tabela users através de prepared statements e verificando a palavra-passe com `tarefyVerifyPassword()`. Quando os dados estão corretos, o sistema guarda na sessão o id, nome, email e cidade do utilizador.

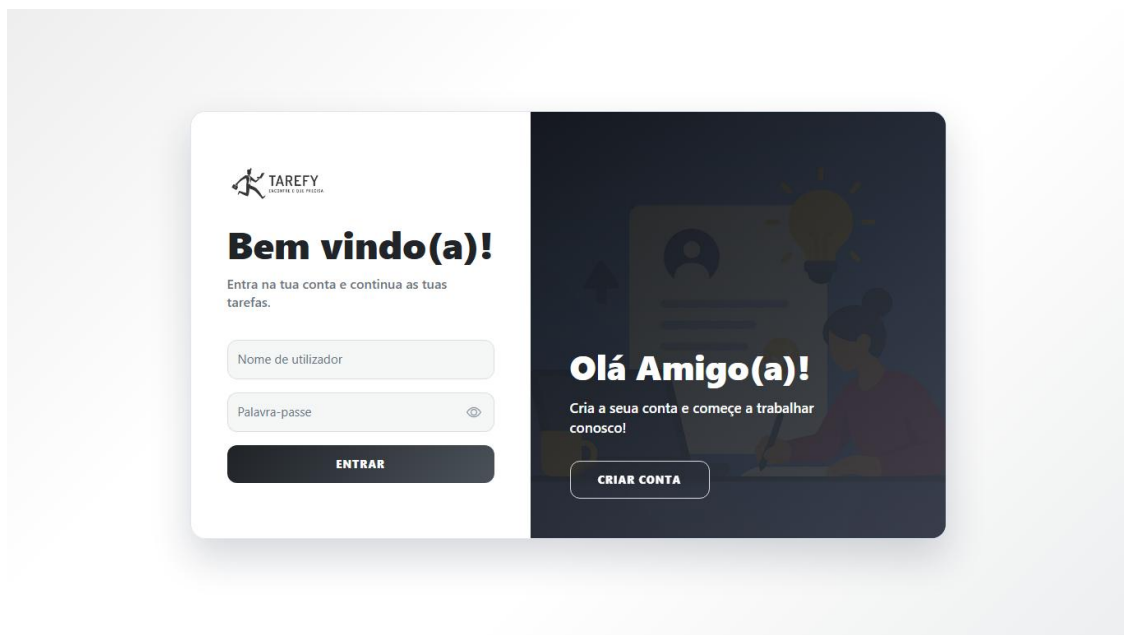


Ilustração 41- Página de início de sessão

7.2. Criar Conta

A criação de conta pede nome, email, telefone, cidade, palavra-passe e confirmação. O sistema verifica duplicações de nome, email e telefone, valida regras de segurança da palavra-passe e guarda o hash com `password_hash()`. A interface mostra os requisitos da palavra-passe em tempo real, ajudando o utilizador a perceber o que falta preencher corretamente.



Crie uma conta

Nome de utilizador

Email

Número de telefone

Insira a sua cidade

Palavra-passe

- Mínimo 6 caracteres
- Letra maiúscula e minúscula
- Pelo menos um número
- Carácter especial (@\$%?&)

Confirmar senha

Ilustração 42- Página de criação de conta

7.3. Página Inicial

A página inicial apresenta tarefas ativas e permite pesquisa por título ou descrição. Também existem filtros por categoria, preço mínimo, preço máximo e ordenação. Cada tarefa é apresentada num cartão com imagem, título, preço, categoria e botão para abrir os detalhes.



TAREFY
ENCONTRE O QUE PRECISA.

[Início](#) [Sobre](#) [Categorias](#) [Admin](#)

  pap [+ Adicionar tarefa](#)

Tarefy

Encontre quem faz. Encontre o que precisa.

PESQUISA

Explorar tarefas

3 resultados

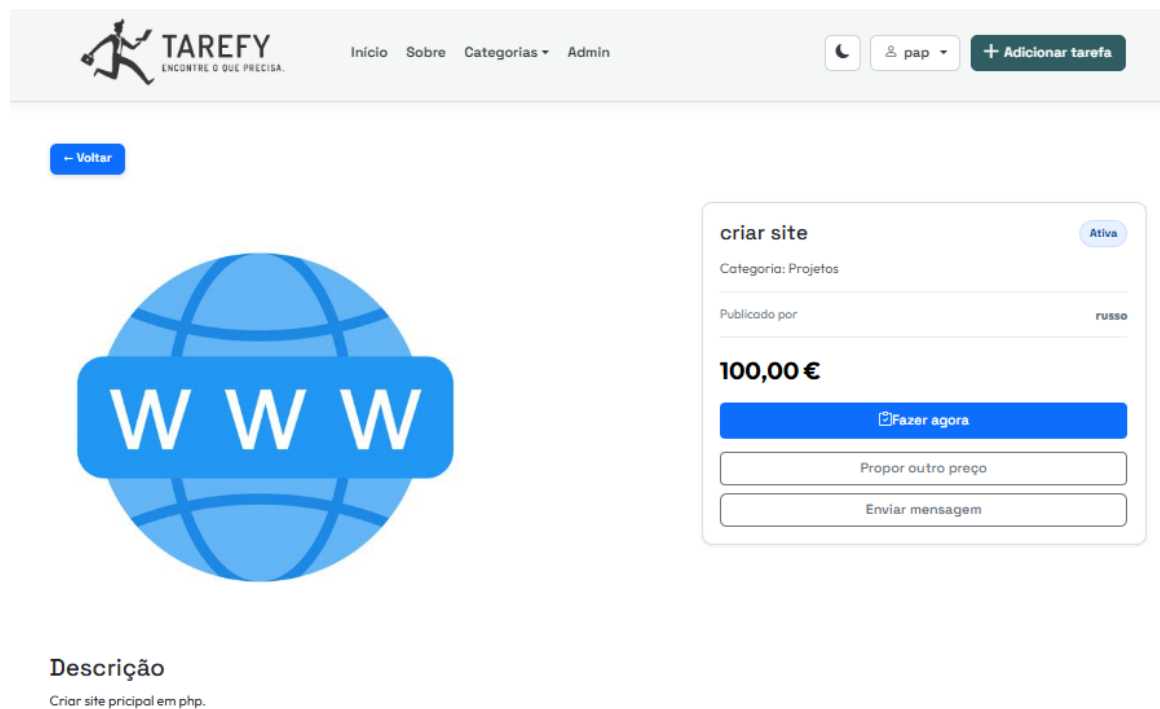
[Pesquisar](#) [Filtros](#)



Ilustração 43- Página inicial e pesquisa de tarefas

7.4. Detalhe da Tarefa

A página de detalhe apresenta a descrição completa, imagem, preço, categoria e estado da tarefa. Quando a tarefa pertence a outro utilizador, aparecem botões para aceitar a tarefa, propor outro preço ou enviar mensagem.



The screenshot shows the TAREFY web application interface. At the top is a navigation bar with the TAREFY logo (a person running with a briefcase) and the tagline 'ENCONTRE O QUE PRECISA.'. Navigation links include 'Início', 'Sobre', 'Categorias', and 'Admin'. On the right of the header are icons for a moon (theme toggle), a user profile labeled 'pap', and a '+ Adicionar tarefa' button. Below the header, on the left, is a blue button labeled '← Voltar'. The main content area features a large blue globe icon with 'WWW' in white. To the right of the globe is a task card for 'criar site', marked as 'Ativa'. The card displays 'Categoria: Projetos', 'Publicado por russo', and a price of '100,00 €'. Below the price are three buttons: 'Fazer agora' (highlighted in blue), 'Propor outro preço', and 'Enviar mensagem'. Below the task card, the section 'Descrição' is shown with the text 'Criar site principal em php.'

Ilustração 44- Página de detalhe de uma tarefa

7.5. Publicar Tarefa

O formulário de publicação permite criar uma tarefa com título, descrição, preço base, categoria e imagem opcional. O servidor valida o carregamento, aceitando apenas imagens JPG/JPEG.

TAREFY
ENCONTRE O QUE PRECISA

Início Sobre Categorias Admin

+ Adicionar tarefa

Minhas tarefas

Publicar tarefa

Descreve o serviço, escolhe uma categoria e define um preço base.

Detalhes da tarefa
Um título claro e uma descrição completa ajudam a tarefa a receber respostas melhores.

Título
Ex: Montar um armário

Descrição
Explica o que precisas, onde é e qualquer detalhe importante.

Preço base
€ 25

Categoria
Selecione uma categoria

Boa publicação, resposta melhor.
Inclui o local, horário preferido, material necessário e o resultado esperado.

- ✓ Título específico
- ✓ Preço realista
- ✓ Descrição completa

Ilustração 45- Formulário de publicação de tarefas

7.6. Perfil

A página de perfil permite alterar dados da conta. O utilizador pode mudar nome, email, cidade e palavra-passe. O sistema verifica se o nome ou email já pertencem a outra conta antes de gravar alterações.

TAREFY
ENCONTRE O QUE PRECISA

Início Sobre Categorias Admin

+ Adicionar tarefa

Minhas tarefas

Perfil

Atualiza os teus dados de utilizador e define a tua localização.

Dados pessoais
Estas informações aparecem associadas às tuas tarefas.

Utilizador
User

Email
user@tarefy.pt

Cidade
x Braga

Segurança
Deixa estes campos vazios para manter a palavra-passe atual.

Nova palavra-passe
Nova palavra-passe

Confirmar palavra-passe
Confirmar palavra-passe

Ilustração 46- Página de perfil do utilizador

7.7. Minhas Tarefas

A área Minhas Tarefas mostra as tarefas publicadas pelo utilizador autenticado e os respetivos estados. A página apresenta contadores de tarefas totais, ativas, em execução e concluídas.



[Início](#)
[Sobre](#)
[Categorias ▾](#)
[Admin](#)



 pap ▾

+ Adicionar tarefa

ÁREA PESSOAL

Minhas tarefas

+ Nova tarefa

Consulta as tarefas que publicaste e acompanha o estado de cada uma.

Total

1

Ativas

1

Em execução

0

Concluídas

0



Teste

Compras

Teste1

25,00 € Publicada em 30/06/2026

Ativa

Ver detalhes

Concluir

Ilustração 47- Área de tarefas publicadas pelo utilizador

7.8. Tarefas a Fazer

A área Tarefas a Fazer lista tarefas que o utilizador aceitou executar. Esta informação é guardada na tabela `task_assignments`. A partir desta página o utilizador pode consultar o detalhe da tarefa, abrir uma mensagem relacionada ou acompanhar se a tarefa ainda está em execução ou já foi concluída.



[Início](#)
[Sobre](#)
[Categorias ▾](#)
[Admin](#)



 pap ▾

+ Adicionar tarefa

ÁREA DE TRABALHO

Tarefas a fazer

Procurar tarefas

Estas são as tarefas que aceitaste realizar.

Total

1

Em execução

1

Concluídas

0



Cuidar do meu filho

Família

Das 08:00 até as 10:00

10,00 € Publicado por amordavidadorusso Aceite em 30/06/2026

Em execução

Ver detalhes

Mensagem

Ilustração 48- Área de tarefas aceites para executar

7.9. Mensagens e Negociação

O sistema de mensagens permite combinar detalhes e enviar propostas de preço. Cada conversa está associada a uma tarefa. As mensagens guardam remetente, texto, data e preço opcional, tornando possível negociar antes da execução do serviço.

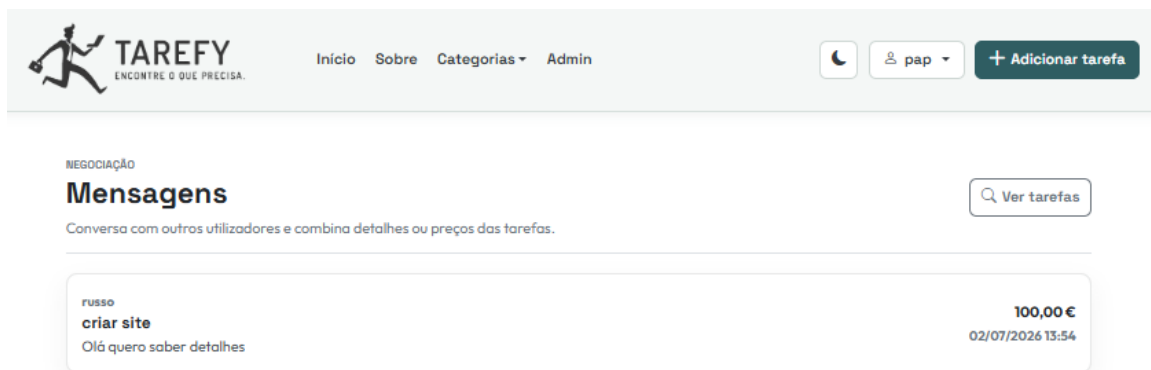


Ilustração 49- Página de mensagens e negociação

7.10. Concluir Tarefas e Avaliações

A página Concluir Tarefas permite ao utilizador que publicou a tarefa marcar o serviço como concluído. Quando isto acontece, o sistema atualiza o estado da tarefa, regista a data de conclusão, atualiza a atribuição correspondente e sincroniza a encomenda usada pelas avaliações.

O módulo de avaliações permite que os utilizadores envolvidos numa tarefa concluída se avaliem mutuamente. As avaliações ficam guardadas na tabela reviews, são associadas a uma order e aparecem nos perfis públicos, contribuindo para a confiança dentro da plataforma.

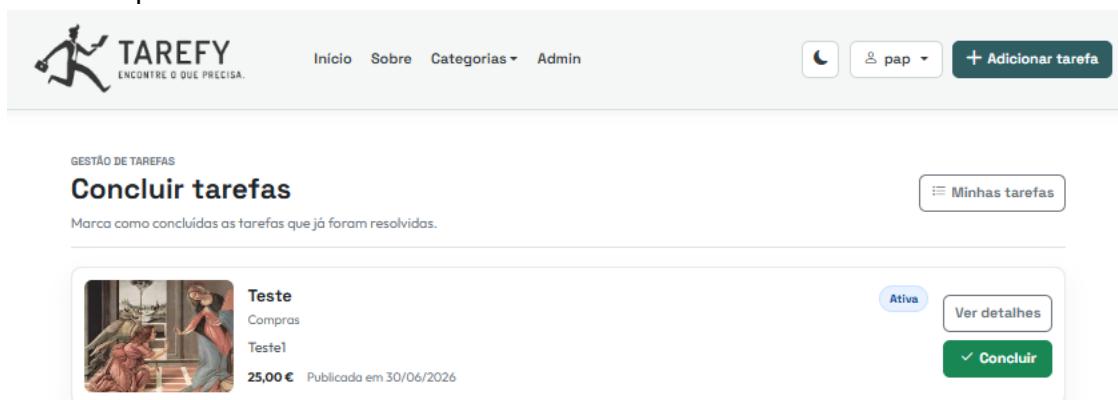


Ilustração 50- Página de conclusão de tarefas



TAREFY

ENCONTRE O QUE PRECISA.

Início

Sobre

Categorias ▾



 russo ▾

+ Adicionar tarefa

O teu perfil

AValiação Média

5.0/5



RECEBIDAS

1

ENVIADAS

1

PENDENTES

1

TAREFAS CONCLUÍDAS

2

ESTADO

Não verificado

Avaliações pendentes

Avalia os utilizadores com quem trabalhaste em tarefas concluídas.

Chefe de Sushi

Avaliar: zukinha come cu de curioso

Avaliação *



Clica nas estrelas para avaliar de 1 a 5.

Comentário *

Pouca qualidade

Minimo de 10 caracteres.

Enviar avaliação

Avaliações recebidas

Ilustração 51- "Avaliações Pendentes"



TAREFY

ENCONTRE O QUE PRECISA.

Início

Sobre

Categorias ▾



 russo ▾

+ Adicionar tarefa

COMUNIDADE

Avaliações

Gere as avaliações das tarefas concluídas e acompanha a tua reputação.

Minhas tarefas

O teu perfil

AValiação Média

5.0/5



RECEBIDAS

1

ENVIADAS

1

PENDENTES

0

TAREFAS CONCLUÍDAS

1

ESTADO

Não verificado

Avaliações recebidas

O que os outros utilizadores dizem sobre o teu trabalho.

User

Montar movei

 5/5

feito rápido

11/06/2026

Avaliações enviadas

Podes corrigir ou actualizar uma avaliação que já enviaste.

Montar movei

Avaliaste: User

11/06/2026

Ilustração 52- Página de avaliações

Página 48

7.11. Avaliações e Confiança entre Utilizadores

Depois de uma tarefa ser marcada como concluída, o publicador e o executante podem deixar uma avaliação um ao outro. A classificação é registada numa escala de uma a cinco estrelas e pode incluir um comentário. As avaliações recebidas ficam visíveis no perfil público do utilizador, ajudando outras pessoas a perceber a sua experiência dentro da plataforma.

The screenshot shows the TAREFY web interface. At the top is a navigation bar with the TAREFY logo (a running person icon and the text 'TAREFY ENCONTRE O QUE PRECISA'), links for 'Inicio', 'Sobre', 'Categorias', and 'Admin', a dark mode toggle, a user profile dropdown labeled 'pap', and a '+ Adicionar tarefa' button. Below the navigation bar is the user profile for 'RUSSO', located in 'Aveiro' with '1 tarefas concluídas'. The profile has a 5.0/5 rating from 1 review. A blue button 'Enviar mensagem' is next to the profile. Below the profile is a section titled 'Avaliações (1)' containing a single review from a user named 'User' (with a verified badge). The review shows a 5/5 star rating, the text 'Monitor movei' and 'feito rápido', and the date '11/06/2026'.

TAREFY
ENCONTRE O QUE PRECISA.

Inicio Sobre Categorias Admin

🌙 pap + Adicionar tarefa

RUSSO
Aveiro 1 tarefas concluídas
5.0/5 ★★★★★ (1 avaliações)

Enviar mensagem

Avaliações (1)

User
Monitor movei
feito rápido
11/06/2026

★★★★★ 5/5

8. Ferramentas de Administrador

Foi adicionada uma área de administração ao projeto. Como a base de dados original não tinha uma coluna própria para permissões, foi criada uma verificação por lista de utilizadores administradores no ficheiro `inc/helpers/adminAuth.php`. A lista pode ser configurada através da variável de ambiente `TAREFY_ADMIN_USERS`; por defeito, os nomes `Admin` e `admin` têm acesso à administração.

A administração foi integrada na navegação principal. Quando o utilizador tem permissões, aparece a aba `Admin` no menu. O painel mostra métricas, categorias, tarefas, utilizadores, utilizadores verificados e avaliações.

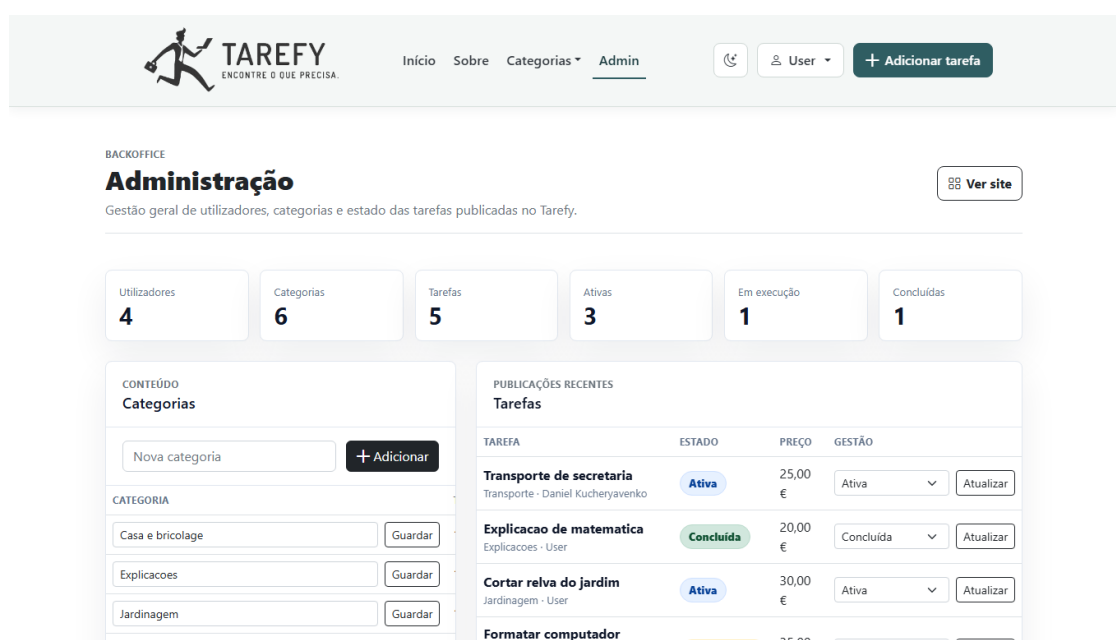


Figura 2 - Área de administração do site

8.1. Acesso ao Painel

O acesso é validado pela função `isCurrentUserAdmin()`. Caso o utilizador não esteja autorizado, a página apresenta uma mensagem de erro e não mostra os dados administrativos. As ações do painel usam token de sessão para impedir submissões acidentais ou não autorizadas.

8.2. Gestão de Categorias

- Adicionar novas categorias.
- Editar o nome de categorias existentes.
- Apagar categorias que ainda não tenham tarefas associadas.

8.3. Gestão de Tarefas

- Consultar tarefas recentes.
- Ver o estado atual de cada tarefa.

- Alterar estado para ativa, em execução ou concluída.
- Atualizar também a tabela de atribuições quando necessário.

8.4. Gestão de Utilizadores

- Consultar utilizadores recentes.
- Ver email, telefone, cidade e número de tarefas publicadas.
- Marcar utilizadores como verificados.
- Remover a verificação quando necessário.

Repor a palavra-passe de um utilizador com validação de segurança.

8.5. Operações e Organização do Painel

O painel foi organizado para permitir uma consulta rápida do estado geral da plataforma. Os contadores mostram utilizadores, categorias, tarefas, estados de execução e avaliações. Nas áreas de gestão, o administrador pode pesquisar informação, atualizar estados, gerir categorias e verificar utilizadores, mantendo as operações separadas por secções.

9. Segurança e Validação

- A segurança foi considerada em várias partes do projeto. A aplicação usa sessões PHP para proteger páginas internas, tokens de sessão para ações sensíveis e prepared statements nas consultas principais, reduzindo o risco de injeção SQL.
- As palavras-passe novas são guardadas com `password_hash()` e verificadas com `password_verify()`. Para não bloquear contas antigas, o helper de autenticação também reconhece hashes SHA1 legados e faz rehash automático depois de um login bem-sucedido.
- Na criação de conta existe validação de campos obrigatórios, duplicação de nome, email e telefone, confirmação de palavra-passe e regras mínimas de complexidade. No carregamento de imagem existe validação do ficheiro com `getimagesize()`, MIME type e extensão, aceitando apenas imagens JPG/JPEG.
- Proteção de páginas com sessão.
- Prepared statements em consultas SQL.
- Tokens de sessão para ações de tarefas, chat, avaliações e administração.
- `password_hash()`, `password_verify()` e rehash automático de hashes antigos.
- Validação de palavra-passe no registo e na reposição feita pelo administrador.
- Validação de imagens enviadas pelo utilizador.
- Separação de permissões para administração.

10. Testes Realizados

Além dos testes manuais, foi feita validação de sintaxe aos ficheiros PHP com php -l. A validação foi executada nos 34 ficheiros PHP do projeto e não apresentou erros.

Teste	Resultado
Login com credenciais válidas	Sessão criada e redirecionamento para index.php.
Login com dados inválidos	Mensagem de erro apresentada sem criar sessão.
Criação de conta	Validação de campos, duplicações e regras de palavra-passe.
Acesso à página inicial	Página carregada com tarefas, pesquisa, filtros e ordenação.
Publicação de tarefa	Formulário apresentado, imagem JPG validada e tarefa guardada na base de dados.
Detalhe da tarefa	Dados completos apresentados com ações adequadas ao utilizador.
Aceitar tarefa	Registo em task_assignments criado e estado atualizado para em execução.
Concluir tarefa	Estado atualizado para concluída e encomenda sincronizada para avaliações.
Sistema de mensagens	Conversa criada, mensagem guardada e proposta de preço registada.
Sistema de avaliações	Avaliações associadas a orders e apresentadas nos perfis.
Painel Admin	Métricas, categorias, tarefas, utilizadores, verificações e reposição de palavra-passe carregadas.
php -l em 34 ficheiros PHP	Sem erros de sintaxe.
Seleção de cidades	Pesquisa apresentada num seletor compacto; a escolha de Almada guardou corretamente o city_id 24.
API de cidades	Endpoint testado com resposta JSON e sincronização de 308 municípios portugueses na tabela cities.
Perfil sem mudança de password	Alterações de nome, email e cidade guardadas sem obrigar o utilizador a alterar

Teste	Resultado
	a palavra-passe.
Relações da base de dados	Validadas 17 foreign keys entre utilizadores, tarefas, mensagens, encomendas, avaliações e recuperação de palavra-passe.
Teste automático de avaliações	Fluxo de aceitação, conclusão e avaliações mútuas executado com sucesso através do ficheiro tests/reviews_flow_test.php.
Validação JavaScript	Ficheiros de interface validados com node -check, sem erros de sintaxe.

Tabela 6 - Testes funcionais e validações realizadas

10.1. Ambiente de Teste e Resultados

Os testes foram realizados num ambiente local com PHP e MySQL/MariaDB, recorrendo ao browser para validar a interface e ao PHP CLI para verificar a sintaxe dos ficheiros. Para os fluxos mais sensíveis foram usados dados temporários, removidos no final dos testes, de forma a confirmar a criação de registos, a atualização de estados e a consistência das relações sem alterar os dados principais da aplicação.

Além dos testes manuais, foi executado um teste automático do sistema de avaliações. Este teste confirmou a criação de uma tarefa, a aceitação por outro utilizador, a conclusão, a criação da encomenda associada e o registo de avaliações mútuas. Os resultados detalhados foram acrescentados à Tabela 6.

11. Dificuldades Encontradas

Uma das dificuldades foi gerir diferentes estados das tarefas. Uma tarefa pode estar ativa, em execução ou concluída, e os botões apresentados mudam conforme o utilizador seja o dono, o executante ou apenas um visitante autenticado.

Outra dificuldade foi criar uma área de administração sem alterar demasiado a lógica existente. A solução encontrada foi usar uma lista de administradores no código, permitir configuração por variável de ambiente e proteger as ações com token de sessão.

Também foi necessário organizar melhor o projeto, separando funcionalidades por ficheiros dentro da pasta `inc/pages` e por helpers dentro de `inc/helpers`, para evitar que o `index.php` ficasse demasiado grande e difícil de manter.

A evolução da base de dados exigiu cuidado especial, porque foi preciso manter compatibilidade com tabelas antigas, adicionar atribuições de tarefas, conversas e avaliações, e garantir que os estados ficassem sincronizados entre `jobs`, `task_assignments` e `orders`.

Outra dificuldade foi a normalização de passwords: o projeto passou a usar `password_hash()`, mas manteve suporte para SHA1 legado para que contas já existentes continuassem a conseguir iniciar sessão.

A escolha de cidades foi outra situação que exigiu vários ajustes. Uma lista simples com muitos municípios podia tornar o formulário pouco prático, enquanto uma caixa de pesquisa separada criava demasiado espaço visual. A solução final foi um seletor compacto: a pesquisa só aparece quando o utilizador abre a lista, mantendo o formulário limpo e permitindo encontrar rapidamente a cidade pretendida.

12. Melhorias Futuras

- Criar uma coluna role ou is_admin na tabela users para substituir a lista de administradores no código.
- Implementar recuperação de palavra-passe completa usando a tabela password_resets e envio de email.
- Criar notificações de novas mensagens, propostas e avaliações pendentes.
- Melhorar paginação, pesquisa e filtros na administração.
- Expandir a aplicação móvel ligada aos endpoints da pasta api/mobile.
- Adicionar sistema de denúncia ou moderação de tarefas e utilizadores.
- Melhorar tratamento de acentuação e codificação em todos os ficheiros PHP e templates.
- Criar mais testes automatizados para login, tarefas, mensagens, administração e API móvel.
- Adicionar suporte a mais formatos de imagem, mantendo validação segura no servidor.

13. Conclusão

O projeto Tarefy permitiu desenvolver uma aplicação web funcional, com ligação a base de dados e várias funcionalidades úteis. A plataforma permite criar contas, publicar tarefas, pesquisar serviços, aceitar tarefas, concluir tarefas, enviar mensagens, negociar preços, avaliar utilizadores e gerir informação através de uma área administrativa.

Durante o desenvolvimento foram aplicados conhecimentos de PHP, MySQL, HTML, CSS e JavaScript, bem como ferramentas e bibliotecas de apoio como Bootstrap e Bootstrap Icons. A organização modular do projeto, por páginas, helpers, configurações e recursos estáticos, tornou a aplicação mais simples de manter e de evoluir.

Apesar de ainda existirem melhorias possíveis, o resultado final corresponde ao objetivo principal: criar uma plataforma simples para aproximar pessoas que precisam de ajuda em tarefas de utilizadores disponíveis para as realizar, com autenticação, comunicação, gestão de estados, administração e mecanismos de confiança através de avaliações e perfis públicos.

14. Manual Prático de Utilização

Esta secção descreve a utilização normal do Tarefy do ponto de vista de quem entra na plataforma. Não substitui o código nem os testes técnicos, mas mostra como cada funcionalidade aparece no ecrã e qual o resultado esperado em cada passo.

14.1. Criar uma conta

O primeiro passo é abrir a página de criação de conta. O utilizador preenche os dados pessoais, escolhe a cidade e define uma palavra-passe que respeite os requisitos de segurança apresentados no formulário.

Antes de guardar, o sistema verifica se já existe um utilizador com o mesmo nome, email ou telefone. Quando os dados são válidos, é criada uma conta e a palavra-passe é guardada como hash.

Resultado esperado: O formulário valida os campos obrigatórios.

A conta fica pronta para iniciar sessão.

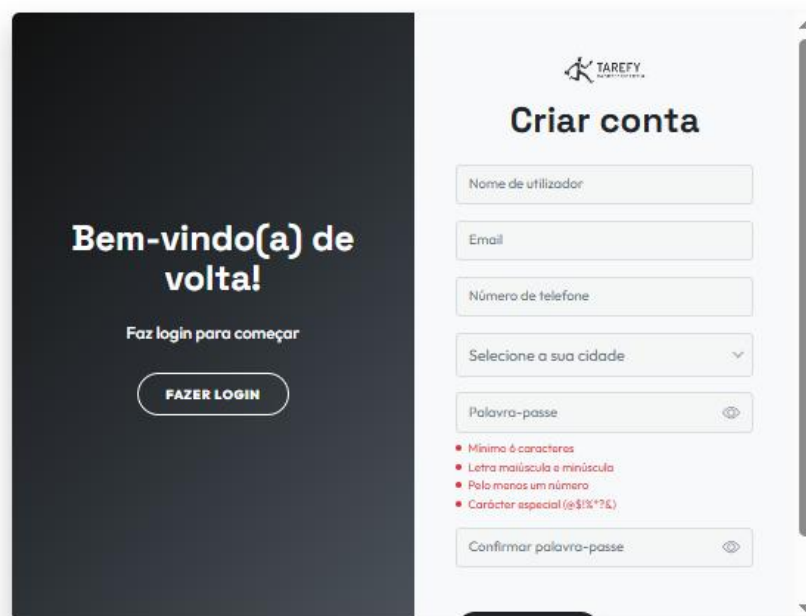


Ilustração 53- Criação de conta na plataforma Tarefy

15.2. Iniciar sessão

Depois de criar a conta, o utilizador inicia sessão com o nome de utilizador e a palavra-passe. A aplicação valida as credenciais e cria uma sessão apenas quando a autenticação é bem-sucedida.

Se os dados estiverem incorretos, é mostrada uma mensagem de erro sem revelar informação sobre a existência de outras contas. Se estiverem corretos, o utilizador é encaminhado para a área principal da plataforma.

Resultado esperado: As credenciais são verificadas no servidor.

O utilizador autenticado é encaminhado para o início.

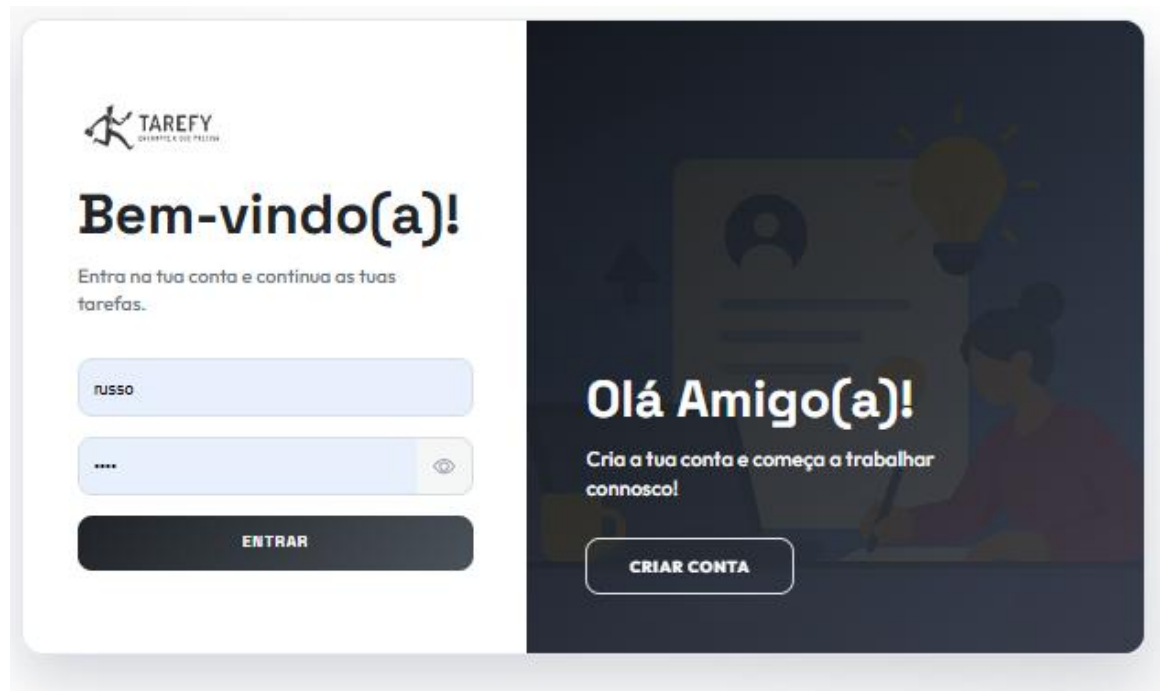


Ilustração 54- Início de sessão na aplicação

15.3. Explorar tarefas

Na página inicial é possível consultar as tarefas ativas publicadas por outros utilizadores. Cada cartão apresenta a informação essencial e permite abrir o detalhe quando é necessário perceber melhor o serviço pedido.

A pesquisa por texto facilita a procura de tarefas específicas e os filtros permitem limitar os resultados por categoria ou intervalo de preço.

Resultado esperado: As tarefas ativas são apresentadas em cartões.

A pesquisa reduz a lista de resultados.

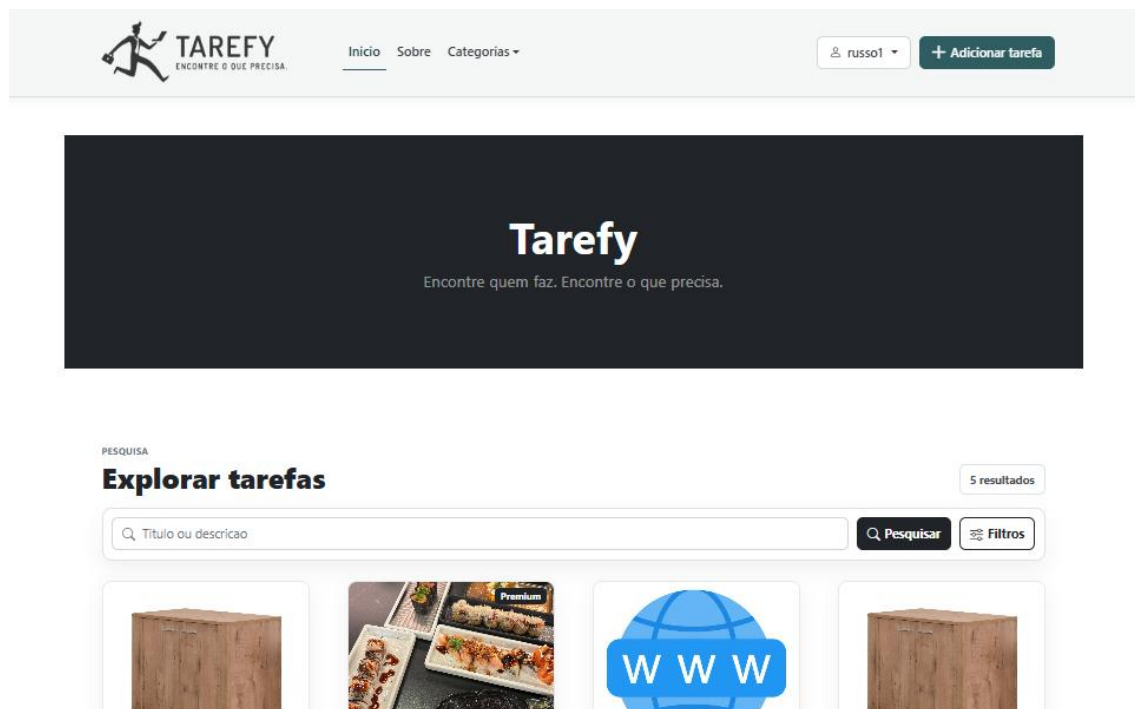


Ilustração 55- Exploração de tarefas na página principal

15.4. Utilizar pesquisa e filtros

A pesquisa aceita palavras relacionadas com o título ou a descrição da tarefa. Os filtros complementam a pesquisa e permitem selecionar uma categoria ou definir valores mínimos e máximos de preço.

Estas opções não alteram os dados guardados na base de dados. Servem apenas para adaptar a lista de tarefas ao que o utilizador procura naquele momento.

Resultado esperado: O utilizador pode pesquisar por texto.

Os filtros ajudam a encontrar tarefas adequadas.

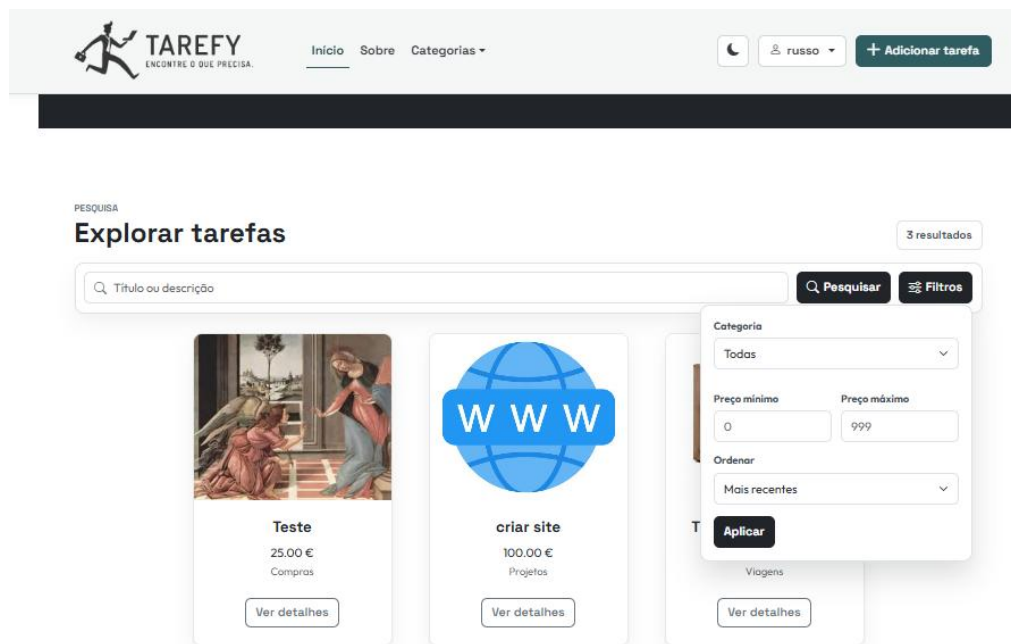


Ilustração 56- Pesquisa e filtros disponíveis na página inicial

15.5. Publicar uma tarefa

Para publicar uma tarefa, o utilizador indica um título claro, uma descrição completa, o preço base e a categoria. Pode também anexar uma imagem que ajude a explicar o serviço que pretende.

Quando o formulário é submetido, o servidor valida os dados e guarda a tarefa associada ao utilizador autenticado. A nova publicação fica disponível para consulta na área pessoal e na página inicial.

Resultado esperado: A tarefa fica associada ao utilizador que a publicou.

A imagem é opcional e é validada antes de ser guardada.

TAREFY
ENCONTRE O QUE PRECISA.

Inicio Sobre Categorias Admin

Admin + Adicionar tarefa

PUBLICAÇÃO

Publicar tarefa [Minhas tarefas](#)

Descreve o serviço, escolhe uma categoria e define um preço base.

Detalhes da tarefa
Um título claro e uma descrição completa ajudam a tarefa a receber respostas melhores.

Título
Ex: Montar um armário

Descrição
Explica o que precisas, onde é e qualquer detalhe importante.

Preço base € 25 **Categoria** Selecciona uma categoria

Imagem
Opcional, mas ajuda a mostrar melhor o tipo de tarefa.

Escolher imagem JPG
A imagem aparece em baixo antes de publicar.

Boa publicação, resposta melhor.
Inclui o local, horário preferido, material necessário e o resultado esperado.

- ✓ Título específico
- ✓ Preço realista
- ✓ Descrição completa

+ Publicar tarefa

Ilustração 57- Formulário atual de publicação de tarefa

15.6. Consultar o detalhe e aceitar uma tarefa

Ao abrir uma tarefa, o utilizador consegue consultar a descrição completa, a categoria, o preço e a imagem publicada. Se a tarefa pertencer a outra pessoa e ainda estiver ativa, ficam disponíveis ações relacionadas com a execução ou a comunicação.

Quando a tarefa é aceite, o sistema cria uma atribuição e altera o estado para indicar que o serviço está em execução. A tarefa deixa de estar disponível para ser aceite por outros utilizadores.

Resultado esperado: Apenas tarefas de outros utilizadores podem ser aceites.

A aceitação cria um registo na tabela `task_assignments`.

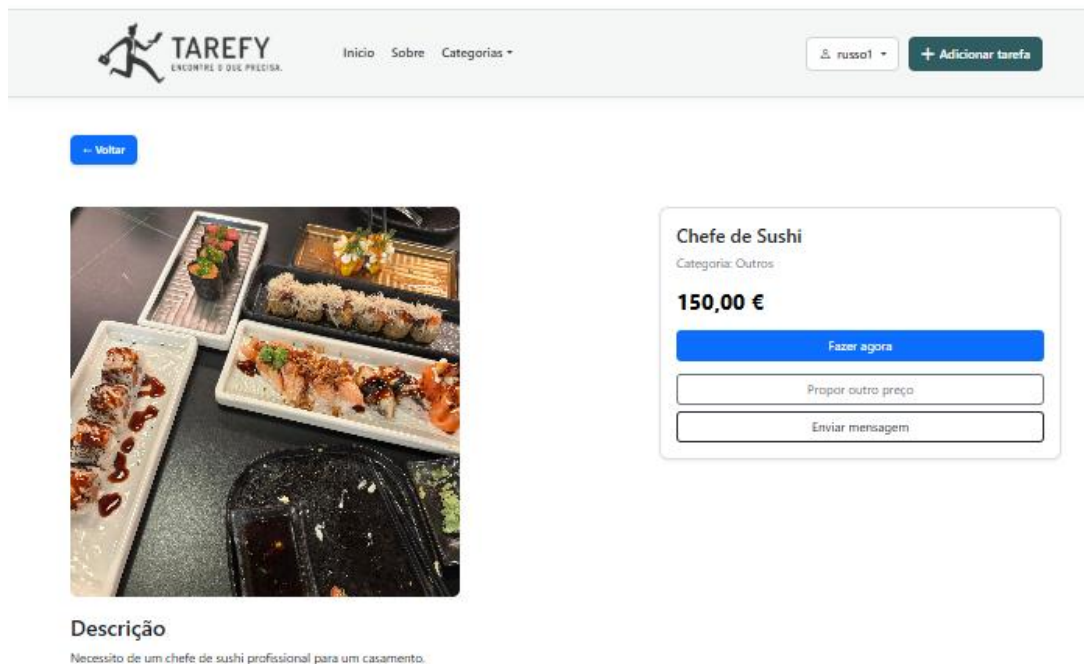


Ilustração 58- Detalhe de tarefa com ações de aceitação e contacto

15.7. Comunicar e negociar

A área de mensagens permite trocar informação sem sair da plataforma. Cada conversa está associada a uma tarefa, evitando que os assuntos de serviços diferentes fiquem misturados.

Quando necessário, uma mensagem pode incluir uma proposta de preço. Desta forma, as duas partes podem chegar a um acordo antes de concluir a tarefa.

Resultado esperado: Cada conversa pertence a uma tarefa concreta.

As mensagens podem conter propostas de preço.

TAREFY
ENCONTRE O QUE PRECISA

Início Sobre Categorias Admin

User + Adicionar tarefa

NEGOCIAÇÃO

Mensagens

Conversa com outros utilizadores e combina detalhes ou preços das tarefas.

Ver tarefas

Miguel Costa
Formatar computador portatil
Perfeito. Trago o portátil e confirmamos os programas necessários.
35,00 €
01/06/2026 10:00

Ana Silva
Montar armário no quarto
Posso montar o armário por 40 euros se tiver o manual disponível.
45,00 €
01/06/2026 10:00

Ilustração 59- Mensagens e negociação associadas a uma tarefa

15.8. Atualizar o perfil

No perfil, o utilizador pode alterar o nome, email ou cidade. A palavra-passe só é alterada quando os dois campos de segurança são preenchidos e coincidem, evitando mudanças acidentais.

A cidade é escolhida através de um seletor compacto. Ao abrir o campo, é possível pesquisar na lista sem ocupar espaço permanente no formulário.

Resultado esperado: Nome, email e cidade podem ser atualizados sem alterar a palavra-passe.

A cidade escolhida é guardada pelo respetivo identificador.

TAREFY
ENCONTRE O QUE PRECISA.

Início Sobre Categorias ▾

🌙 russo + Adicionar tarefa

CONTA

Perfil

Atualiza os teus dados de utilizador e define a tua localização.

Minhas tarefas

russo
daniel.cruglik@gmail.com

Dados pessoais

Estas informações aparecem associadas às tuas tarefas.

Utilizador

russo

Email

daniel.cruglik@gmail.com

Cidade

Aveiro ▾

Segurança

Deixa estes campos vazios para manter a palavra-passe atual.

Nova palavra-passe

Nova palavra-passe

Confirmar palavra-passe

Confirmar palavra-passe

✓ Guardar alterações

Ilustração 60- Atualização de dados do perfil

15.9. Concluir e avaliar uma tarefa

Depois de o serviço estar terminado, a tarefa pode ser marcada como concluída. Esta ação atualiza os estados relacionados e permite que os participantes deixem uma avaliação.

A avaliação inclui estrelas e comentário. Ao guardar, fica disponível no perfil público do utilizador que recebeu a classificação.

Resultado esperado: As avaliações só estão disponíveis em tarefas concluídas.

Cada participante pode avaliar a outra pessoa uma vez por contratação.

15.10. Utilizar o painel de administração

O painel de administração está disponível apenas para utilizadores autorizados. A partir desta área é possível observar métricas da plataforma e gerir categorias, tarefas, utilizadores verificados e avaliações.

As operações administrativas são protegidas por validação de sessão e token. Assim, ações como mudar um estado ou apagar uma categoria não ficam acessíveis a utilizadores comuns.

Resultado esperado: O acesso é limitado a utilizadores autorizados.

O painel organiza ações de gestão por secções.

16. Validação e Testes Detalhados

A validação foi tratada como parte do desenvolvimento e não apenas como uma tarefa final. Cada fluxo importante foi testado com dados válidos e inválidos, verificando o que aparece no ecrã, o que é guardado na base de dados e se a aplicação mantém o comportamento esperado depois de uma alteração.

16.1. Testes de autenticação e sessão

Foram testados logins com credenciais corretas, credenciais incorretas e páginas internas sem sessão ativa. O objetivo foi confirmar que o sistema cria sessão apenas depois de validar a palavra-passe e bloqueia o acesso a áreas privadas quando o utilizador não está autenticado.

Também foi verificado o comportamento de palavras-passe antigas. Quando uma conta ainda tem um hash legado, o sistema aceita a autenticação e atualiza o hash depois de um login bem-sucedido.

Resultado: os cenários principais foram executados sem erros bloqueadores.

16.2. Testes de formulários e validações

Os formulários de criação de conta, perfil e publicação de tarefas foram testados com campos vazios, emails inválidos, palavras-passe diferentes, nomes duplicados e imagens não permitidas. Estes testes garantem que a validação no servidor não depende apenas do que aparece no browser.

No perfil foi confirmado que a palavra-passe é opcional. O utilizador consegue alterar dados pessoais sem preencher a área de segurança, mas se decidir alterar a palavra-passe tem de confirmar o novo valor.

Resultado: os cenários principais foram executados sem erros bloqueadores.

16.3. Testes de tarefas, estados e atribuições

Foram testadas a publicação de tarefas, a aceitação por outro utilizador, a atualização do estado e a visualização nas áreas pessoais. Uma tarefa ativa passa a estar em execução quando é aceite e deixa de poder ser atribuída a outro utilizador.

A conclusão foi validada para garantir que os estados das tabelas jobs, task_assignments e orders ficam sincronizados. Este ponto foi importante porque envolve várias tabelas relacionadas.

Resultado: os cenários principais foram executados sem erros bloqueadores.

16.4. Testes de mensagens, propostas e avaliações

As conversas foram testadas a partir do detalhe de uma tarefa. Foi confirmada a criação da conversa, o envio de mensagens, o registo da data e a associação correta ao remetente e à tarefa respetiva.

O fluxo de avaliações foi validado de ponta a ponta através de um teste automático. O teste cria dados temporários, simula a aceitação e conclusão da tarefa, regista as duas avaliações e remove os registos de teste no final.

Resultado: os cenários principais foram executados sem erros bloqueadores.

16.5. Testes de administração, API e base de dados

No painel administrativo foram testadas ações de categorias, atualização de estado de tarefas, verificação de utilizadores e reposição de palavra-passe. Todas as ações confirmam a permissão do utilizador e validam o token de sessão antes de alterar dados.

A API de cidades foi testada através de pedidos JSON. A resposta devolve os municípios disponíveis e a aplicação mantém a lista sincronizada com a tabela cities. Também foram verificadas as relações da base de dados para confirmar que não existem referências a utilizadores ou tarefas inexistentes.

Resultado: os cenários principais foram executados sem erros bloqueadores.

17. Decisões Técnicas e Justificação

Ao longo do projeto foram tomadas várias decisões técnicas para manter a aplicação simples de utilizar e, ao mesmo tempo, mais fácil de manter. As decisões seguintes explicam algumas das opções que tiveram mais impacto na organização final do Tarefy.

17.1. Organização por páginas e helpers

Em vez de concentrar toda a lógica no `index.php`, as páginas foram separadas na pasta `inc/pages` e as funções reutilizáveis foram colocadas em `inc/helpers`. Esta divisão permite encontrar mais facilmente o código relacionado com autenticação, chat, avaliações, tarefas ou administração.

Esta decisão foi mantida porque reduz duplicação de código, melhora a consistência do sistema e facilita a evolução futura da plataforma.

17.2. Base de dados relacional

A utilização de chaves estrangeiras garante que as ligações entre tabelas fazem sentido. Uma mensagem não pode existir sem conversa, uma avaliação não pode existir sem encomenda e uma tarefa tem de estar associada a um utilizador e a uma categoria válidos.

Esta decisão foi mantida porque reduz duplicação de código, melhora a consistência do sistema e facilita a evolução futura da plataforma.

17.3. Proteção de palavras-passe

As palavras-passe não são guardadas em texto simples. A aplicação usa `password_hash()` para criar o hash e `password_verify()` para validar o acesso. O suporte temporário a hashes antigos foi mantido apenas para não bloquear utilizadores que já existiam antes da atualização de segurança.

Esta decisão foi mantida porque reduz duplicação de código, melhora a consistência do sistema e facilita a evolução futura da plataforma.

17.4. Interface e responsividade

A interface utiliza Bootstrap como base e estilos próprios organizados por componentes e páginas. A preocupação principal foi manter os formulários legíveis, os botões claros e a navegação utilizável em ecrãs mais pequenos, sem perder informação importante.

Esta decisão foi mantida porque reduz duplicação de código, melhora a consistência do sistema e facilita a evolução futura da plataforma.

17.5. API e reutilização de dados

A criação de endpoints JSON permite reutilizar dados fora da página HTML tradicional. O caso das cidades mostra esta vantagem: a mesma informação pode servir o formulário web e futuros clientes, como a aplicação móvel prevista no projeto.

Esta decisão foi mantida porque reduz duplicação de código, melhora a consistência do sistema e facilita a evolução futura da plataforma.

Anexos

Nesta secção são apresentados excertos de código importantes para compreender o funcionamento do projeto. Os excertos não substituem o código completo, mas mostram partes relevantes da ligação à base de dados, autenticação, navegação modular, permissões, administração e métricas do painel.

Anexo A — Ligação à base de dados

A ligação à base de dados está concentrada num único ficheiro. A configuração lê variáveis de ambiente e define o charset UTF-8, evitando que as restantes páginas tenham de repetir a mesma lógica.

```
function createDbConnection(bool $stopOnError = true): ?mysqli
{
    loadEnvFile(__DIR__ . "/../.env");

    $servername = dbEnv("TAREFY_DB_HOST", "localhost");
    $username = dbEnv("TAREFY_DB_USER", "root");
    $password = dbEnv("TAREFY_DB_PASS", "");
    $dbname = dbEnv("TAREFY_DB_NAME", "tarefy");

    try {
        $connection = @new mysqli($servername, $username, $password, $dbname);
    } catch (Throwable $exception) {
        if ($stopOnError) {
            die("Connection failed: " . $exception->getMessage());
        }

        return null;
    }

    if ($connection->connect_error) {
        if ($stopOnError) {
            die("Connection failed: " . $connection->connect_error);
        }
    }
}
```

Ilustração 61- Ligação à base de dados e leitura da configuração.

```

1  TAREFY_DB_HOST=infoaee.pt
2  TAREFY_DB_USER=dkucheryavenko
3  TAREFY_DB_PASS=sistemas
4  TAREFY_DB_NAME=2326_dkucheryavenko
5
```

Ilustração 62 - Ficheiro .env

Anexo B — Login e criação da sessão

O início de sessão consulta o utilizador através de uma query preparada, valida a palavra-passe e cria a sessão usada nas áreas protegidas da plataforma.

```
login.php

1  <?php
2  session_start();
3  require_once __DIR__ . '/config/db.php';
4  require_once __DIR__ . '/inc/helpers/passwordHelpers.php';
5
6  $user = filter_input(INPUT_POST, 'user', FILTER_SANITIZE_SPECIAL_CHARS);
7  $pass = filter_input(INPUT_POST, 'pass', FILTER_SANITIZE_SPECIAL_CHARS);
8
9  if ($_SERVER['REQUEST_METHOD'] === 'POST') {
10     $stmt = $conn->prepare('SELECT id, name, email, city_id, password FROM users WHERE name = ? LIMIT 1');
11     $stmt->bind_param('s', $user);
12     $stmt->execute();
13     $result = $stmt->get_result();
14
15     if ($result->num_rows === 1 && tarefyVerifyPassword($pass, $row['password'])) {
16         $_SESSION['user_id'] = $row['id'];
17         $_SESSION['user'] = $row['name'];
18         $_SESSION['email'] = $row['email'];
19         $_SESSION['city'] = $row['city_id'];
20         header('Location: index.php');
21         exit();
22     }
23 }
```

Ilustração 62- Validação de credenciais e criação da sessão

Anexo C — Navegação modular no index.php

O ficheiro index.php funciona como ponto central da aplicação. O módulo pedido na navegação determina qual a página interna que é carregada.

```
index.php

1  switch ($module) {
2      case 'perfil':
3          include __DIR__ . '/inc/pages/perfil.php';
4          break;
5      case 'home':
6          include __DIR__ . '/inc/pages/home.php';
7          break;
8      case 'newTask':
9          include __DIR__ . '/inc/pages/newTask.php';
10         break;
11     case 'minhasTarefas':
12         include __DIR__ . '/inc/pages/minhasTarefas.php';
13         break;
14     case 'chat':
15         include __DIR__ . '/inc/pages/chat.php';
16         break;
17     case 'admin':
18         include __DIR__ . '/inc/pages/admin.php';
19         break;
20     case 'reviews':
21         include __DIR__ . '/inc/pages/reviews.php';
22         break;
23 }
```

Ilustração 63- Navegação modular para as páginas da aplicação

Anexo D — Controlo de permissões de administrador

As operações administrativas verificam primeiro a autorização do utilizador. Além desta confirmação, é utilizado um token de sessão para validar cada ação enviada pelo painel.

```
inc/helpers/adminAuth.php

1 function tarefyAdminUsers(): array
2 {
3     $configuredUsers = getenv('TAREFY_ADMIN_USERS');
4     if ($configuredUsers !== false && trim($configuredUsers) !== '') {
5         return array_values(array_filter(array_map('trim', explode(',', $configuredUsers))));
6     }
7     return ['Admin', 'admin', 'pap'];
8 }
9
10 function isCurrentUserAdmin(): bool
11 {
12     if (!isset($_SESSION['user'])) {
13         return false;
14     }
15     return in_array((string) $_SESSION['user'], tarefyAdminUsers(), true);
16 }
17
18 function ensureAdminToken(): void
19 {
20     if (empty($_SESSION['admin_action_token'])) {
21         $_SESSION['admin_action_token'] = bin2hex(random_bytes(16));
22     }
23 }
```

Ilustração 64- Verificação de permissões e token de administração

Anexo E — Métricas do painel de administração

As métricas do painel são calculadas através de consultas simples à base de dados. Estes valores permitem perceber rapidamente a quantidade de tarefas, utilizadores e avaliações existentes.

```
inc/pages/admin.php

1 $stats = [
2     'users' => adminCount($conn, 'SELECT COUNT(*) FROM users'),
3     'verifiedUsers' => adminCount($conn, 'SELECT COUNT(*) FROM users WHERE verified = 1'),
4     'categories' => adminCount($conn, 'SELECT COUNT(*) FROM categories'),
5     'tasks' => adminCount($conn, 'SELECT COUNT(*) FROM jobs'),
6     'activeTasks' => adminCount($conn, "SELECT COUNT(*) FROM jobs WHERE status = 'ativa'"),
7     'runningTasks' => adminCount($conn, "SELECT COUNT(*) FROM jobs WHERE status = 'em_execucao'"),
8     'completedTasks' => adminCount($conn, "SELECT COUNT(*) FROM jobs WHERE status = 'concluida'"),
9     'reviews' => adminCount($conn, 'SELECT COUNT(*) FROM reviews'),
10 ];
```

Ilustração 65- Consultas usadas para calcular as métricas administrativas.

Anexo F — Seletor compacto de cidades

O código seguinte mostra a criação do seletor pesquisável utilizado nos formulários de registo e perfil. A lista é apresentada apenas quando o utilizador abre o campo, evitando ocupar espaço permanente na página.

```
assets/js/cityPicker.js

1  (function () {
2      function normalizeText(value) {
3          return String(value || '')
4              .trim()
5              .toLocaleLowerCase('pt-PT')
6              .normalize('NFD')
7              .replace(/[\u0300-\u036f]/g, '');
8      }
9
10     function enhanceCitySelect(select) {
11         var options = getOptions(select);
12         var trigger = document.createElement('button');
13         var menu = document.createElement('div');
14         var search = document.createElement('input');
15         var list = document.createElement('div');
16
17         trigger.className = 'city-picker-trigger';
18         search.placeholder = 'Pesquisar cidade';
19         list.setAttribute('role', 'listbox');
20
21         select.classList.add('city-picker-native');
22         select.required = false;
23     }
24 }());
```

Ilustração 66- Excerto do código do seletor compacto de cidades

Anexo G — Segurança de palavras-passe

Este excerto apresenta as funções usadas para criar hashes de palavras-passe, validar credenciais e identificar hashes antigos que precisam de atualização após o login.

```
inc/helpers/passwordHelpers.php

1  <?php
2  function tarefyHashPassword(string $password): string
3  {
4      return password_hash($password, PASSWORD_DEFAULT);
5  }
6
7  function tarefyVerifyPassword(string $password, string $storedHash): bool
8  {
9      if ($storedHash === '') {
10         return false;
11     }
12
13     $info = password_get_info($storedHash);
14     if (($info['algo'] ?? 0) !== 0) {
15         return password_verify($password, $storedHash);
16     }
17
18     return preg_match('/^[a-f0-9]{40}$/i', $storedHash) === 1
19         && hash_equals(strtolower($storedHash), sha1($password));
20 }
```

Ilustração 67- Excerto do código de proteção de palavras-passe

Anexo H — API de cidades

O endpoint devolve dados em JSON e permite manter a lista de cidades sincronizada com os municípios portugueses. A resposta é preparada para poder ser utilizada tanto pelo site como por outros clientes.

```
api/cities.php

1  <?php
2  define('TAREFY_DB_RETURN_NULL_ON_ERROR', true);
3  require_once __DIR__ . '/../config/db.php';
4
5  header('Content-Type: application/json; charset=utf-8');
6
7  function sendCitiesResponse(int $statusCode, array $payload): void
8  {
9      http_response_code($statusCode);
10     echo json_encode($payload, JSON_UNESCAPED_UNICODE);
11     exit();
12 }
13
14 function getLocalCityCount(mysqli $conn): int
15 {
16     $result = $conn->query('SELECT COUNT(*) AS total FROM cities');
17     $row = $result ? $result->fetch_assoc() : null;
18     return (int) ($row['total'] ?? 0);
19 }
20
21 // A lista de municípios é sincronizada com a tabela cities.
```

Ilustração 68- Excerto do código da API de cidades

Anexo I — Controlo de permissões administrativas

O painel de administração verifica se o utilizador tem autorização e protege as ações com um token de sessão. Esta validação impede que operações administrativas sejam executadas por utilizadores não autorizados.

```
inc/helpers/adminAuth.php

1  <?php
2  function tarefyAdminUsers(): array
3  {
4      $configuredUsers = getenv('TAREFY_ADMIN_USERS');
5      if ($configuredUsers !== false && trim($configuredUsers) !== '') {
6          return array_values(array_filter(array_map('trim', explode(',', $configuredUsers))));
7      }
8      return ['Admin', 'admin', 'pap'];
9  }
10
11 function isCurrentUserAdmin(): bool
12 {
13     if (isset($_SESSION['user'])) {
14         return false;
15     }
16     return in_array((string) $_SESSION['user'], tarefyAdminUsers(), true);
17 }
18
19 function ensureAdminToken(): void
20 {
21     if (empty($_SESSION['admin_action_token'])) {
22         $_SESSION['admin_action_token'] = bin2hex(random_bytes(16));
23     }
24 }
```

Ilustração 69- Excerto do código de controlo administrativo

Anexo J - Documentação dos Chats

A funcionalidade de chat permite que os utilizadores comuniquem entre si sobre uma determinada tarefa. Através deste sistema, é possível esclarecer dúvidas, combinar detalhes do serviço e negociar preços de forma simples e organizada.

Cada conversa está associada a uma tarefa específica. Quando um utilizador decide enviar uma mensagem sobre uma tarefa, o sistema verifica se já existe uma conversa entre esse utilizador e o criador da tarefa. Se já existir, essa conversa é aberta. Caso contrário, é criada uma nova conversa. Assim, evita-se que existam várias conversas repetidas sobre a mesma tarefa entre os mesmos utilizadores.

As conversas e mensagens são guardadas na base de dados. A tabela `task_conversations` guarda as informações principais da conversa, como a tarefa associada, o utilizador interessado, o dono da tarefa e as datas de criação e atualização. Já a tabela `task_messages` guarda as mensagens enviadas, o utilizador que enviou cada mensagem, a data de envio e, caso exista, uma proposta de preço. O sistema também garante que apenas os utilizadores envolvidos na conversa podem aceder ao chat. Antes de mostrar ou enviar mensagens, é feita uma verificação para confirmar se o utilizador autenticado faz parte daquela conversa. Se não tiver permissão, o acesso é recusado.

Na página das mensagens, o utilizador consegue ver todas as suas conversas organizadas pela mais recente. Ao abrir uma conversa, são apresentadas todas as mensagens por ordem cronológica, permitindo acompanhar facilmente o histórico da negociação. As mensagens enviadas pelo próprio utilizador aparecem de forma diferente das mensagens recebidas, tornando a leitura mais clara.

Para além das mensagens de texto, o chat também permite enviar propostas de preço. Esta funcionalidade é útil quando o utilizador quer sugerir um valor diferente para realizar a tarefa. Sempre que uma nova mensagem é enviada, a conversa é atualizada automaticamente e passa a aparecer no topo da lista de mensagens.

Desta forma, o sistema de chats torna a comunicação entre os utilizadores mais prática, segura e organizada, ajudando no processo de negociação e realização das tarefas.

```

function getOrCreateTaskConversation(mysqli $conn, int $jobId, int $requesterId, string &$error): ?int
{
    if (!ensureChatTables($conn)) {
        $error = 'error';
        return null;
    }

    if (function_exists('ensureWorkTables') && !ensureWorkTables($conn)) {
        $error = 'error';
        return null;
    }

    $stmt = $conn->prepare("
        SELECT j.job_id, j.provider_id, j.status, a.worker_id
        FROM jobs j
        LEFT JOIN task_assignments a ON a.job_id = j.job_id AND a.worker_id = ?
        WHERE j.job_id = ?
        AND (j.status = 'ativa' OR a.worker_id IS NOT NULL)
        LIMIT 1
    ");

    if (!$stmt) {
        $error = 'error';
        return null;
    }
}

```

Ilustração 70- Excerto de código da funcionalidade da conversa

```

$stmt->bind_param("ii", $requesterId, $jobId);
$stmt->execute();
$result = $stmt->get_result();
$job = $result->fetch_assoc();
$stmt->close();

if (!$job) {
    $error = 'invalid';
    return null;
}

$providerId = (int) $job['provider_id'];
if ($providerId === $requesterId) {
    $error = 'own_task';
    return null;
}

$stmt = $conn->prepare("
    INSERT INTO task_conversations (job_id, requester_id, provider_id, created_at, updated_at)
    VALUES (?, ?, ?, NOW(), NOW())
    ON DUPLICATE KEY UPDATE updated_at = updated_at
");

if (!$stmt) {
    $error = 'error';
}

```

Ilustração 71- Continuação do excerto de código da funcionalidade da conversa

Webgrafia

- PHP. Documentação oficial do PHP. Disponível em:
<https://www.php.net/docs.php>
PHP. Documentação oficial de password_hash() e password_verify().
Disponível em:
https://www.php.net/manual/pt_BR/ref.password.php
- MySQL. Documentação oficial do MySQL. Disponível em:
<https://dev.mysql.com/doc/>
- MariaDB. Documentação oficial do MariaDB. Disponível em:
<https://mariadb.com/kb/en/documentation/>
- Bootstrap. Documentação oficial do Bootstrap. Disponível em:
<https://getbootstrap.com/docs/>
- Bootstrap Icons. Documentação oficial dos Bootstrap Icons. Disponível em:
<https://icons.getbootstrap.com/>
- MDN Web Docs. Documentação de HTML, CSS e JavaScript. Disponível em:
<https://developer.mozilla.org/>
- GeoAPI Portugal. Lista de municípios usada na sincronização das cidades. Disponível em: <https://geoapi.pt/>
- Visual Studio Code. Documentação oficial. Disponível em:
<https://code.visualstudio.com/docs>
- Laragon. Site oficial. Disponível em:
<https://laragon.org/>