

# **Projeto e Desenvolvimento Informático**

**3.º Ano da Licenciatura em Informática de Gestão**

**Ano Letivo  
2023-2024**

Atividade 3

## **Relatório do Projeto DevLancer**



Autores

**Joana Esteves (2021143113)**

**Tomás Figueiredo (2021141066)**

Docentes

**Jorge Henriques**

**Paulo Soares**

Elaborado em

**05/2024**



## ÍNDICE

1	Introdução .....	4
2	Tecnologias utilizadas.....	5
3	Implementação .....	7
3.1	<i>Estrutura Modular do Projeto</i> .....	10
3.2	<i>Implementações de Segurança</i> .....	10
4	Dificuldades e desvios .....	12
5	Melhorias a Implementar .....	13
6	Conclusão .....	14
7	Links do projeto.....	15
8	Referências .....	16

## **1 Introdução**

Este projeto foi desenvolvido no âmbito da unidade curricular de Projeto e Desenvolvimento Informático da licenciatura em Informática de Gestão do Instituto Superior de Contabilidade e Administração de Coimbra.

O presente relatório aborda o desenvolvimento de um website que consiste num Marketplace destinado à prestação de serviços freelancer na área da tecnologia. Este projeto surge da necessidade de criar uma plataforma que facilite a interação entre prestadores de serviços e clientes na área de tecnologia, proporcionando um ambiente onde seja possível encontrar, solicitar e fornecer serviços de forma eficiente e segura.

Este relatório visa fornecer uma visão geral do trabalho realizado, descrevendo as tecnologias utilizadas, as principais funcionalidades implementadas, desafios enfrentados e possíveis melhorias a serem consideradas no futuro.

## 2 Tecnologias utilizadas

Para o desenvolvimento da nossa plataforma online para a prestação e aquisição de serviços freelancer na área de tecnologia, utilizámos diversas tecnologias que permitiram a criação de uma aplicação robusta e eficiente. Em seguida listamos as principais tecnologias utilizamos e explicamos qual o proveito retirado das mesmas:

### Backend

- **Django:** Utilizámos Django como a framework principal para o desenvolvimento do projeto. O Django facilitou a manipulação da base de dados, dado que permitiu o uso de objetos Python graças ao seu ORM (Object-Relational Mapping). Para além disso, oferece um sistema de autenticação integrado e administração automática que acelerou o desenvolvimento deste projeto.
- **Python:** Como linguagem de programação, e derivado da escolha da Framework Django, utilizámos Python. Consideramos que foi uma escolha adequada pela sua simplicidade que permitiu desenvolver as funcionalidades necessárias de forma eficiente.

### Frontend

- **HTML:** Utilizámos HTML para estruturar as páginas web da aplicação, garantindo uma disposição clara do conteúdo.
- **CSS:** Utilizámos CSS para estilizar a interface, assegurando que o website fosse visualmente apelativo e responsivo em diferentes ecrãs.
- **JavaScript:** Incorporámos JavaScript para fazer uso de AJAX (Asynchronous JavaScript and XML) e tornar o website mais interativo.
- **Bootstrap:** Incorporámos o Bootstrap para agilizar o desenvolvimento do frontend, utilizando os seus componentes para criar uma interface moderna e responsiva.

### Ambiente de Desenvolvimento

- **Visual Studio Code (VSCode):** Optámos pelo Visual Studio Code como o editor de código principal devido às suas extensões que suportam Python e Django, bem como ferramentas que facilitaram o desenvolvimento.

### Pagamentos

- **PayPal Standard:** Implementámos a versão Standard do PayPal para simular os pagamentos com contas sandbox, através do API (Application Programming Interface) fornecido pelo PayPal. Esta ferramenta permitiu

testar e validar o fluxo de pagamentos sem transações reais, garantindo segurança e eficácia no processo de pagamento.

### Gestão de Projeto

- **GitHub:** Utilizámos o GitHub para o controlo de versão do código, permitindo a colaboração entre os membros da equipa.
- **Trello:** Utilizámos a plataforma Trello para a gestão de tarefas, organizando o progresso do projeto em quadros e checklists, o que facilitou a coordenação e a distribuição de tarefas entre os membros da equipa.

### Pacotes adicionais

Para este projeto fizemos uma seleção de pacotes adicionais que foram fundamentais para implementar as funcionalidades do DevLancer. Em seguida, listamos os mesmos e explicamos de que forma foram úteis ao desenvolvimento:

- **django-bootstrap5:** Biblioteca que integra o Bootstrap 5 com o Django, e permitiu uma fácil implementação de componentes Bootstrap na aplicação.
- **crispy-bootstrap5 e django-crispy-forms:** Bibliotecas que facilitaram a criação de formulários estilizados e eficientes no Django, integrando o Bootstrap 5 e melhorando a renderização e gestão dos formulários.
- **GitPython:** Biblioteca Python que facilitou a interação com repositórios Git diretamente por código Python.
- **pillow:** Biblioteca Python que adiciona funcionalidades de manipulação de imagens ao projeto Django.
- **django-bootstrap-datepicker-plus** e **django-datetime-widget:** Bibliotecas que fornecem widgets de data e hora baseados em Bootstrap para formulários Django, facilitando a entrada de datas e horas pelos utilizadores.
- **django-paypal** e **paypalrestsdk:** Biblioteca que facilita a integração do sistema de pagamentos PayPal com a aplicação Django e SDK (Software Development Kit) do PayPal para Python que oferece uma interface para interagir com a API REST (Application Programming Interface - Representational State Transfer) do PayPal, respetivamente.

### 3 Implementação

O website que desenvolvemos é um Marketplace dedicado à prestação de serviços freelancer na área de tecnologia. A plataforma foi projetada para ligar clientes que precisam de serviços específicos a prestadores de serviços de tecnologia com vários níveis de experiência. Em seguida, apresentamos as principais funcionalidades agrupadas por categorias:

#### Autenticação e Gestão de Conta

- **Registo de Utilizador:** Permite que os utilizadores se registem na plataforma como prestador de serviços ou cliente, adquirindo as permissões do tipo de utilizador com o qual se registaram.
- **Login, Logout e recuperação de palavra-passe:** Permite que os utilizadores registados façam login e logout e que recuperem a sua palavra-passe, se necessário.
- **Definições de utilizador:** Permite que os utilizadores registados alterem as definições de conta, como: palavra-passe, email de conta e nome de utilizador.

#### Gestão de Perfil e Serviços

- **Definições de Perfil:** Permite aos utilizadores gerir as suas informações de perfil, dependendo do tipo de utilizador.
  - **Cliente:** Pode alterar o e-mail para contacto.
  - **Prestador:** Pode fazer a alteração do e-mail para contacto, descrição de perfil, imagem de perfil e link para o portefólio.
- **Publicação de Serviços:** Possibilita que os prestadores de serviços publiquem os serviços que desejam oferecer na plataforma a partir do seu perfil, incluindo: título, descrição do serviço, categoria em que se insere, tecnologias que irá utilizar, preço médio por hora e imagem ilustrativa.

#### Pesquisa e Recomendações

- **Pesquisa e Navegação:** Oferece uma pesquisa por categorias, com opções de filtragem com base no preço máximo e tecnologias utilizadas e ainda ordenação dos resultados segundo a classificação dos prestadores de serviço e ordem crescente ou decrescente de preços. Qualquer utilizador, mesmo que não esteja registado, pode efetuar esta pesquisa, mas para proceder com o pedido de serviço terá de ter uma conta de cliente.
- **Recomendações:** Os utilizadores podem visualizar na página inicial recomendações de serviços com base na popularidade (número de pedidos).

## Gestão de Pedidos e Transações

- **Informações e comunicação:**
  - **Cliente:** Pode visualizar na página do serviço as informações do mesmo como: imagem exemplificativa, descrição do serviço e preço médio por hora, assim como as informações do prestador de serviço, nomeadamente: nome, imagem, descrição de perfil, e-mail para contacto, link para portefólio e avaliações de clientes anteriores. O cliente deve analisar o perfil do prestador e contactar o mesmo caso necessite de mais alguma informação antes de realizar o pedido.
  - **Prestador:** Quando o cliente envia um pedido, o prestador de serviços tem acesso ao nome e e-mail de contacto do cliente para o contactar em caso de dúvida sobre o pedido, antes de proceder a aceitar ou rejeitar.
- **Requisição de serviços:** Permite aos clientes enviarem um pedido ao prestador de serviços, que inclui: serviço pretendido, uma descrição do projeto que pretende, o orçamento que o prestador teria para a realização do projeto e o prazo máximo para a sua conclusão.
- **Gestão de serviços:** O prestador de serviços pode gerir os serviços que publicou no DevLancer: editar ou eliminar o mesmo e visualizar as páginas onde os pedidos estão publicados.
- **Gestão de pedidos:**
  - **Cliente:** Pode fazer a gestão dos pedidos efetuados: seguir o estado do pedido: pendente, aceite ou rejeitado, cancelar enquanto estão pendentes de resposta do prestador, efetuar o pagamento após ter sido aceite e dar o pedido como concluído quando o prestador terminar o serviço.
  - **Prestador:** Pode gerir os pedidos recebidos, incluindo aceitar ou rejeitar o mesmo, e seguir o estado do pedido: cancelado, pago e concluído, consoante as ações do cliente.
- **Transações e Pagamentos:** Facilita transações financeiras seguras com o método de pagamento PayPal. O cliente poderá realizar o pagamento caso o prestador de serviços tenha aceite o seu pedido, e após um pagamento bem-sucedido ambos poderão visualizar as informações da transação, como: data em que foi realizada e o valor.
- **Avaliações e Comentários:** Após os clientes confirmarem que o prestador concluiu o serviço, podem deixar uma avaliação e comentário sobre o prestador, proporcionando feedback útil para os restantes utilizadores.
- **Notificações:**
  - **Cliente:** Recebe notificações quando o seu pedido é aceite ou rejeitado.



- **Prestador:** Recebe notificações quando há um novo pedido de serviço, se o mesmo for cancelado, quando o cliente efetua o pagamento, confirma a conclusão e deixa uma avaliação.

### 3.1 Estrutura Modular do Projeto

No desenvolvimento do DevLancer, utilizámos uma estrutura modular, dividindo o código em diferentes aplicações, conforme recomendado pela arquitetura do Django. Esta abordagem facilita a manutenção, escalabilidade e colaboração entre os membros do grupo. As principais aplicações do projeto são:

- **Devlapp:** Contém a lógica principal da aplicação, incluindo a gestão dos serviços, pedidos e perfis.
- **Notificações:** Responsável pela gestão e envio de notificações aos utilizadores, informando sobre o estado dos pedidos e outras atualizações relevantes.
- **Pagamento:** Trata de toda a lógica relacionada com transações financeiras, com recurso ao PayPal para facilitar os pagamentos seguros.
- **Registo:** Gere a autenticação e registo de utilizadores.

### 3.2 Implementações de Segurança

Durante o desenvolvimento do DevLancer, utilizámos medidas para garantir a proteção dos dados e transações dos utilizadores, aproveitando os recursos da framework Django e do PayPal:

1. **Proteção CSRF:** Implementámos tokens CSRF (Cross-Site Request Forgery) em todos os formulários do website para prevenir ataques deste tipo.
2. **Proteção XSS:** O Django oferece proteção contra ataques de Cross-Site Scripting (XSS), pelo que processa automaticamente os dados inseridos pelos utilizadores antes de os renderizar nos templates HTML.
3. **Autenticação e Autorização:** A framework Django oferece um sistema de autenticação integrado do qual fizemos recurso para gerir as sessões dos utilizadores de forma segura. Além disso, configurámos permissões específicas para garantir que apenas utilizadores autorizados possam aceder e modificar determinados recursos.

4. **Encriptação de Dados Sensíveis:** Todas as palavras-passe dos utilizadores estão armazenadas de forma encriptada utilizando o algoritmo PBKDF2, fornecido pelo Django. Esta prática garante que, mesmo que a base de dados seja comprometida, as palavras-passe dos utilizadores continuam protegidas. Para a alteração de palavra-passe pelos utilizadores recorremos sempre ao sistema de reposição de palavra-passe do Django para garantir segurança no processo.
5. **Segurança nas Transações:** Integrámos o PayPal como meio de pagamento, aproveitando a segurança e confiabilidade desta plataforma para realizar transações financeiras. As comunicações entre o website DevLancer e o PayPal são encriptadas usando SSL/TLS, garantindo que os dados transmitidos são seguros. Além disto, a API REST do PayPal utiliza OAuth 2.0 para autenticação, garantindo que apenas utilizadores autorizados podem realizar transações.

## 4 Dificuldades e desvios

Durante o desenvolvimento do nosso website, enfrentámos algumas dificuldades e realizámos pequenos desvios em relação ao plano inicialmente estabelecido na 2ª entrega. Apesar de termos implementado a grande maioria das funcionalidades previstas, houve duas delas que não foram concluídas conforme planeado: a comunicação via chat integrado e o sistema de denúncias.

### Funcionalidades Não Implementadas

- **Denunciar Utilizadores:** Esta funcionalidade permitiria aos utilizadores denunciarem outros utilizadores caso houvesse indícios de que não estavam a cumprir as regras de utilização da plataforma. Embora tenha sido considerada inicialmente, concluímos que a sua relevância era limitada no contexto atual do projeto. O sistema de denúncias consistiria principalmente num formulário que os utilizadores preencheriam, o qual seria então revisto pelo administrador. Dada a simplicidade desta implementação e a perceção de que não traria um valor significativo imediato, decidimos não priorizar esta funcionalidade. Esta funcionalidade estava também relacionada ao Chat, que como veremos em seguida não foi implementado, dado que seria uma das páginas onde os utilizados poderiam fazer as denúncias com base nas conversas do mesmo.
- **Comunicações:** Planeámos inicialmente facilitar a comunicação entre prestadores de serviços e clientes através de um sistema de mensagens privadas num chat integrado. No entanto, devido a restrições de tempo e à necessidade de priorizar outras funcionalidades essenciais, esta funcionalidade não foi implementada. Além disso, considerámos que, dado que os utilizadores já têm acesso aos e-mails de contacto uns dos outros, poderiam comunicar externamente, o que mitigava a necessidade imediata de um sistema de chat integrado.

### Dificuldades Enfrentadas

Uma das principais dificuldades que enfrentámos foi a falta de experiência prévia com as tecnologias utilizadas. Nenhum membro da equipa tinha trabalhado anteriormente com Python, HTML, CSS, JavaScript ou Django. Este desafio inicial exigiu um período de aprendizagem significativa para adquirir as habilidades necessárias para o desenvolvimento do projeto. Apesar das dificuldades, conseguimos superar esta barreira através de bastante pesquisa e colaboração entre os membros do grupo.

Outro desafio foi a implementação do sistema de pagamentos com o PayPal. Esta funcionalidade envolveu a integração com a API do PayPal, o que exigiu testar várias implementações, pesquisa e testes para garantir que as transações estivessem funcionais. Embora complexa, conseguimos implementar esta funcionalidade com sucesso, permitindo que os pagamentos entre os utilizadores fossem processados de forma eficiente e segura.

## 5 Melhorias a Implementar

Para futuras iterações do projeto, há algumas melhorias em consideração:

1. **Implementação do Sistema de Chat:** Desenvolver a funcionalidade de comunicação direta entre clientes e prestadores de serviços, proporcionando uma forma mais eficiente de discutir detalhes de projetos, orçamentos e prazos dentro da plataforma DevLancer.
2. **Sistema de Denúncias:** Introduzir um formulário de denúncias para que os utilizadores possam reportar comportamentos inadequados ou violações das regras da plataforma. Isto ajudará a manter um ambiente mais seguro e regulado.
3. **Melhorias na Interface do Utilizador:** Continuar a refinar e melhorar a interface do utilizador para garantir que a experiência de navegação e utilização seja o mais intuitivo e agradável possível.

Embora tenhamos enfrentado desafios significativos durante o desenvolvimento, estamos satisfeitos com o progresso alcançado e consideramos que o projeto tem potencial para ser ainda mais trabalhado no futuro.

## **6 Conclusão**

Em conclusão, o desenvolvimento deste projeto proporcionou uma experiência enriquecedora de aprendizagem e aplicação de conhecimentos técnicos. Ao longo do processo, enfrentamos desafios significativos, como a familiarização com tecnologias completamente novas e o primeiro contacto com o desenvolvimento web. No entanto, estes obstáculos foram superados através de pesquisa, empenho e colaboração de todos os membros.

Consideramos que tivemos um bom desempenho, dado que alcançamos a grande maioria dos objetivos estabelecidos inicialmente, no entanto, reconhecemos que ainda há espaço para melhorias e expansões futuras, conforme abordado no relatório.

No geral, este projeto foi uma oportunidade relevante para o nosso percurso como estudantes de Informática de Gestão, que nos permitiu aplicar os conhecimentos que adquirimos em sala de aula em diversas unidades curriculares, permitindo-nos desenvolver habilidades práticas e enfrentar desafios e problemas característicos da área de Informática. Estamos satisfeitos com o resultado alcançado até o momento e entusiasmados para aprimorar e expandir este projeto no futuro.

Por fim, agradecemos ao Professor Paulo Soares pela sua orientação e suporte ao longo deste projeto.

## 7 Links do projeto

- **Github:** <https://github.com/tomas02f/DevLancer.git>
- **Trello:** <https://trello.com/b/oGyxUf3T/devlancer>

## 8 Referências

Bootstrap. (2024). Documentação do Bootstrap (versão 5.3) [Manual técnico]. <https://getbootstrap.com/docs/5.3/getting-started/introduction/>

Django Software Foundation. (2024). Django documentation (versão 5.0) [Manual técnico]. <https://docs.djangoproject.com/en/5.0/>

Django Software Foundation. (2024). Django security topics [Manual técnico]. <https://docs.djangoproject.com/en/5.0/topics/security/>

PayPal. (2024). PayPal sandbox testing guide [Manual técnico]. <https://developer.paypal.com/api/rest/sandbox/>

PayPal. (2024). Documentação da API do PayPal [Manual técnico]. django-crispy-forms. (2024). <https://developer.paypal.com/api/rest/>

Documentação do django-crispy-forms [Manual técnico]. <https://django-crispy-forms.readthedocs.io/en/latest/>

Pillow. (2024). Documentação do Pillow [Manual técnico]. <https://pillow.readthedocs.io/en/stable/>

Django-bootstrap-datepicker-plus. (2024). Documentação do django-bootstrap-datepicker-plus [Manual técnico]. <https://pypi.org/project/django-bootstrap-datepicker-plus/>