

Norberto Moreira

CLIENTE:

TPSI 1123

TURMA:

**RELATÓRIO INTERMÉDIO**

Afonso Ferreira – T0127531

Catarina Lopes – T0127533

Hélder Fernandes – T0127540

Simão Correia – T0127546

Rafael Costa – T0127545

ELABORADO POR:

**PADEL CONNECT**

**ÍNDICE**

[**INTRODUÇÃO** 5](#_Toc184730674)

[Enquadramento do projeto 5](#_Toc184730675)

[Organização do relatório 6](#_Toc184730676)

[Objetivos a cumprir 7](#_Toc184730677)

[Objetivos de valorização adicional 9](#_Toc184730678)

[Requisitos Funcionais 10](#_Toc184730679)

[Requisitos Não Funcionais 12](#_Toc184730680)

[Diagrama de Gantt 13](#_Toc184730681)

[**ENGENHARIA DE SOFTWARE** 14](#_Toc184730682)

[Diagrama Use Case e Sequência – Criar Campo Padel 14](#_Toc184730683)

[Diagrama Use Case e Sequência – Criar Reserva 16](#_Toc184730684)

[Diagrama Use Case e Sequência – Gerir Reserva 17](#_Toc184730685)

[Diagrama Use Case e Sequência – Notificações 17](#_Toc184730686)

[Diagrama Use Case e Sequência – Pesquisar Campo 17](#_Toc184730687)

[Diagrama Use Case e Sequência – Feedback Reserva 17](#_Toc184730688)

[Diagrama Use Case e Sequência – Criar Promoção 17](#_Toc184730689)

[Diagrama de classes 17](#_Toc184730690)

[Dicionário de dados 17](#_Toc184730691)

[**DESENVOLVIMENTO** 18](#_Toc184730692)

[Arquitetura da aplicação 18](#_Toc184730693)

[Tecnologias 19](#_Toc184730694)

[Processo de Desenvolvimento 20](#_Toc184730695)

[Sprint I – 21 outubro a 25 outubro 21](#_Toc184730696)

[Sprint II – 28 outubro a 1 novembro 21](#_Toc184730697)

[Sprint III – 4 novembro a 8 novembro 22](#_Toc184730698)

[Sprint IV – 11 novembro a 15 novembro 22](#_Toc184730699)

[Sprint V – 18 novembro a 22 novembro 22](#_Toc184730700)

[Sprint VI – 25 novembro a 29 novembro 23](#_Toc184730701)

[Sprint VII – 2 dezembro a 6 dezembro 23](#_Toc184730702)

[Sprint VIII – 9 dezembro a 11 dezembro 23](#_Toc184730703)

[Interfaces de Utilizador: Protótipo VS Produto Final 24](#_Toc184730704)

[**CONCLUSÃO** 25](#_Toc184730705)

[Objetivos Cumpridos 25](#_Toc184730706)

[Contratempos / Dificuldades 25](#_Toc184730707)

[Possíveis Melhorias 25](#_Toc184730708)

[Considerações Finais 26](#_Toc184730709)

# **INTRODUÇÃO**

O presente projeto desenvolve-se no âmbito da UFCD 5424 - Planeamento de Projetos e Sistemas de Informação, integrada no curso Técnico Especialista em Tecnologias e Programação de Sistemas de Informação.

## Enquadramento do projeto

O projeto visa desenvolver um website inovador para gestão de campos de padel, com o objetivo de transformar e otimizar os processos de reserva e administração destes espaços. A solução tecnológica ultrapassa o conceito tradicional de plataforma digital, apresentando-se como uma resposta estratégica e integrada para a gestão eficiente de campos de padel.

Mediante uma interface amigável, a plataforma oferece uma experiência personalizada que responde simultaneamente às necessidades operacionais de utilizadores, gestores e administradores.

A plataforma foi concebida para proporcionar uma experiência integrada e completa, abrangendo desde a reserva até ao pagamento, e acompanhamento dos campos de padel. Contempla um sistema de login e registo seguro, visualização de horários disponíveis, gestão de reservas e funcionalidades de pagamento, além de ferramentas avançadas para administradores realizarem análises e estatísticas.

Através de relatórios analíticos e estatísticas, os administradores poderão tomar decisões estratégicas com base em dados concretos sobre a utilização dos espaços.

O cliente responsável por esta iniciativa é Norberto Nogueira, que procura uma plataforma digital capaz de revolucionar a gestão de campos de padel, melhorando significativamente a experiência de todos os envolvidos.

O website será desenvolvido com recurso a tecnologias web modernas, garantindo escalabilidade, segurança e uma experiência de utilização intuitiva, preparando-o para futuras expansões e integrações.

## Organização do relatório

Este relatório encontra-se dividido em quatro capítulos:

**Capítulo 1 – Introdução**

Descreve o problema a ser desenvolvido, quais os objetivos gerais do website, os seus requisitos funcionais e não funcionais. É também apresentada uma previsão de execução das tarefas através de um diagrama de Gantt.

**Capítulo 2 – Engenharia de Software**

Este capítulo contextualiza a fundamentação técnica do projeto, alinhando-o com os princípios da engenharia de software.

Utiliza diagramas UML para modelar a plataforma com ênfase em gestão de reservas e campos, funcionalidades administrativas e de clientes, pagamentos, notificações email, feedback e promoções.

**Capítulo 3 – Desenvolvimento**

Neste capítulo abordaremos a solução técnica adotada para o desenvolvimento do website. São descritos os principais aspetos tecnológicos, incluindo o uso de frameworks, linguagens de programação e tecnologias envolvidas.

**Capítulo 4 – Conclusão**

Por fim serão apresentados os resultados alcançados com o desenvolvimento da plataforma, destacando os objetivos cumpridos e as funcionalidades implementadas. Além disso serão abordadas as dificuldades e contratempos enfrentados durante o desenvolvimento e como esses obstáculos foram superados.

## Objetivos a cumprir

Os principais objetivos deste projeto são desenvolver uma aplicação que permita a gestão eficiente de campos de padel, através de uma plataforma intuitiva tanto para utilizadores como para administradores. O sistema visa a facilitar a reserva online de campos, através de um website, garantindo uma experiência otimizada para todos os envolvidos.

As metas essenciais para este projeto são:

**Sistema de Login e Registo:** O sistema permitirá que os utilizadores se registem facilmente através do email ou rede social. Os administradores terão a capacidade de gerir esses utilizadores, garantindo segurança e integridade do acesso ao sistema.

**Implementar Sistema de Reservas:** A aplicação oferecerá uma interface intuitiva onde os utilizadores poderão escolher os horários disponíveis dos vários campos, permitindo que realizem as reservas com confirmação em tempo real. Além disso, os clientes receberão emails informativos acerca das reservas.

**Feedbacks e Avaliações:** Os utilizadores terão a possibilidade de avaliar uma reserva após a conclusão da mesma.

**Funcionalidades de administrador:** Os administradores terão a capacidade ainda de:

* **Gerir Reservas:** Os administradores terão um painel de controlo para visualizar, alterar ou cancelar reservas.
* **Gerir Campos:** Os administradores terão um painel de controlo para visualizar, alterar ou apagar campos
* **Gerir Horários:** Capacidade de definir horários de funcionamento, ajustar tempos de reserva
* **Gestão de Preços:** Implementação de preços dinâmicos, como descontos para horários menos concorridos ou pacotes promocionais.
* **Relatórios e Análises:** Visualização de estatísticas de uso dos campos, receitas, cancelamentos e feedbacks.

**Dashboard Intuitivo:** A dashboard proporcionará uma visão clara e organizada das estatísticas de uso de campos. Os utilizadores terão ainda a possibilidade de consultar o histórico de reservas passadas e futuras.

**Integração Pagamentos Online:** O sistema permitirá pagamentos através de múltiplos métodos, como cartão de crédito, paypal, entre outras opções populares, garantindo um processo de reserva completo.

**Plataforma de Acessibilidade:** A aplicação será desenvolvida para ser responsiva com uma interface otimizada para dispositivos desktop e móveis.

## Objetivos de valorização adicional

Para além dos objetivos principais, temos também algumas metas adicionais que visam a tornar a aplicação mais robusta e adaptável às necessidades futuras.

As metas adicionais são:

**Sistema de rating de campos:** Implementar um sistema de rating aos campos, para que os utilizadores possam comparar campos de acordo com o feedback de outros utilizadores.

**Exportação e Importação de dados:** Seria interessante implementar a capacidade de importar e exportar ficheiros CSV.

**Sistema de Espera:** Por vezes nem sempre temos alguém com quem partilhar a paixão do padel, pelo que seria interessante implementar uma funcionalidade onde os utilizadores poderiam reservar um campo e aguardar por outros jogadores.

**Arquivo de reservas:** Para aqueles utilizadores que fazem reservas com muita frequência, seria interessante implementar um sistema de arquivo para manter as reservas a decorrer mais organizadas e acessíveis.

**Bloqueio de conta:** Uma implementação de segurança interessante, seria criar uma funcionalidade de bloqueio de conta, que iria verificar a quantidade de tentativas de login.

**Registar com rede social:**  Por fim mas não menos importante seria a implementação de registo e inicio de sessão através de uma rede social ou google.

## Requisitos Funcionais

Os requisitos funcionais descrevem as funcionalidades que o sistema deve oferecer para cumprir com os seus objetivos. Os requisitos funcionais são os seguintes:

**Sistema de Login:**

* Capacidade de registo e login através de email
* Será necessária ativação de conta através de email.
* Os administradores terão a capacidade de gerir estes utilizadores.

**Reserva de Campos:**

* Visualização de horários disponíveis para os diferentes campos
* Capacidade de reservar um ou mais campos diretamente pela aplicação.
* Integração de sistemas de pagamento para finalizar uma reserva.
* Enviar um email de confirmação de reserva.
* Os clientes terão a capacidade de gerir as suas reservas.
* Os administradores terão a capacidade de gerir todas as reservas.

**Gestão de horários e preços:**

* Capacidade de definir horários de funcionamento para cada campo.
* Configurar preço hora de cada campo.

**Dashboard:**

* Visualização de estatísticas de uso de campos, reservas e receitas.
* Gestão de reservas.
* Capacidade de edição de informações pessoais.

Esses requisitos garantem que a aplicação cumpra as funções principais necessárias para uma gestão eficaz dos campos de padel, ao mesmo tempo que proporciona uma experiência fácil e intuitiva para os utilizadores e administradores.

## Requisitos Não Funcionais

Os requisitos não funcionais descrevem como o sistema deve ser:

**Segurança:**

* Autenticação segura dos utilizadores, através de protocolos como HTTPS e encriptação de dados sensíveis.
* Acesso exclusivo a dados administrativos.

**Usabilidade:**

* Interface intuitiva e fácil de usar, tanto para utilizadores que reservam campos como para administradores que gerem os gerem.
* Dashboard informativo e amigável, proporcionando uma visão clara e botões de fácil navegação.

**Desempenho:**

* Processamento rápido de reservas e confirmações em tempo real

**Escalabilidade:**

* Estrutura flexível que permita adicionar novas funcionalidades.

## Diagrama de Gantt

No desenvolvimento do projeto de gestão de campos de padel, a organização das tarefas foi planeada de forma estratégica através do Diagrama de Gantt, uma ferramenta visual essencial para o controlo do tempo e das atividades programadas. Este diagrama é complementado com a gestão ativa do backlog no Trello, onde cada tarefa é monitorizada e atualizada conforme o progresso do projeto.

O Diagrama de Gantt divide o projeto em vários sprints, sendo que cada uma delas é composta por tarefas específicas, atribuídas a membros da equipa, com datas de início e término claramente definidas.

A utilização do Diagrama de Gantt e a gestão contínua do backlog no Trello garantem uma execução estruturada e eficiente das tarefas. Este método de organização proporciona transparência e comunicação eficaz entre todos os membros da equipa envolvidos, assegurando que todos os requisitos, tanto funcionais como não funcionais, sejam atendidos dentro dos prazos estipulados, contribuindo para o sucesso do projeto.

# **ENGENHARIA DE SOFTWARE**

Este capítulo contextualiza o projeto de gestão de campos de padel através de UML (Unified Modeling Language), uma linguagem de modelagem amplamente utilizada no desenvolvimento de software. Através da utilização de UML, conseguimos representar visualmente os diversos aspetos do sistema, promovendo uma melhor compreensão da estrutura e das interações dos componentes do projeto.

## Diagrama Use Case e Sequência – Criar Campo Padel

O diagrama de use case ilustra o processo completo de criação de um campo de padel. O fluxo inicia-se quando um utilizador com perfil de editor ou administrador acede a página específica de criação, onde o utilizador preenche todos os dados necessários para a inserção do campo. Após o preenchimento, o sistema valida as informações inseridas, podendo resultar em dois cenários: dados válidos, que permitem prosseguir com a criação, ou dados inválidos, que requerem correções. Em ambos os casos, o sistema fornece feedback apropriado ao utilizador.

Os fluxos principais e alternativos mostram o tratamento dos dados em cada cenário. Enquanto o diagrama de sequência mapeia a interação entre os diferentes componentes do sistema.

|  |  |
| --- | --- |
|  | |
|  |  |
|  | |

## Diagrama Use Case e Sequência – Criar Reserva

O diagrama de use case ilustra o processo completo de criação de uma reserva. O fluxo inicia-se quando um utilizador autenticado acede ao catálogo de campos, seleciona um ou mais campos, seleciona as datas da reserva e procede para o preenchimento dos dados pessoais do utilizador, que serão carregados automaticamente caso o utilizador os tenha preenchido. Por fim é necessário escolher e efetuar o pagamento.

Os fluxos principais e alternativos mostram o tratamento dos dados em cada cenário. Enquanto o diagrama de sequência mapeia a interação entre os diferentes componentes do sistema.

|  |
| --- |
|  |
|  |
|  |

## Diagrama Use Case e Sequência – Gerir Reserva

O seguinte diagrama de use case ilustra o processo completo de gestão de uma reserva. O processo inicia-se com o utilizador a aceder à página de reservas. O sistema irá apresentar as reservas de acordo com a Role, caso seja um Administrador apresenta todas as reservas, caso seja Gestor apresenta as reservas dos seus campos e caso seja um Cliente apresenta apenas as suas reservas. Nesta aba o cliente pode ver ou cancelar a reserva.

Os fluxos principais e alternativos mostram o tratamento dos dados em cada cenário. Enquanto o diagrama de sequência mapeia a interação entre os diferentes componentes dos sistemas.

|  |
| --- |
|  |
|  |
|  |

## Diagrama Use Case e Sequência – Notificações

O diagrama de use case das notificações de email descreve os diferentes cenários em que o sistema envia e-mails aos utilizadores especificando os diversos cenários em que são enviados os emails. Após a reserva é enviado um email com os detalhes da reserva e confirmação de pagamento.

O diagrama de sequência mapeia a interação entre os diferentes componentes dos sistemas.

|  |
| --- |
|  |
|  |

## Diagrama Use Case e Sequência – Pesquisar Campo

O diagrama de use case das de pesquisa de campo representa as diferentes formas de realizar a pesquisa de campo entre localização e nome de campo.

Os fluxos principais e alternativos mostram o tratamento dos dados em cada cenário. Enquanto o diagrama de sequência mapeia a interação entre os diferentes componentes dos sistemas.

|  |
| --- |
|  |
|  |
|  |

## Diagrama Use Case e Sequência – Feedback Reserva

O diagrama de use case ilustra o processo completo de envio de feedback de reserva. O fluxo inicia-se quando o utilizador recebe o pedido de feedback após a reserva. O utilizador avalia a reserva e envia um comentário adicional.

Os fluxos principais e alternativos mostram o tratamento dos dados em cada cenário. Enquanto o diagrama de sequência mapeia a interação entre os diferentes componentes do sistema.

|  |
| --- |
|  |
|  |
|  |

## Diagrama Use Case e Sequência – Criar Promoção

O use case representa a forma como é criada uma promoção, pelo administrador. O administrador acede à página de criação de promoção, preenche os dados, define a visibilidade e é criada a promoção.

Os fluxos principais e alternativos mostram o tratamento dos dados em cada cenário. Enquanto o diagrama de sequência mapeia a interação entre os diferentes componentes do sistema.

|  |
| --- |
|  |
|  |
|  |

## Dicionário de dados

O dicionário de dados mostra com detalhe a estrutura organizacional da nossa empresa assim como as relações entre elas. Este dicionário apresenta a descrição das tabelas. As relações entre estas entidades são essenciais para assegurar a integridade e a eficiência dos dados.

**Tabela Roles**

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo | Tipo | Descrição | Restrições |
| id | Auto-increment | Identificador único | Chave primária |
| nome | Varchar (20) | Nome da role | Obrigatório, único |
| created\_at | Timestamp | Data de criação | Automática |
| updated\_at | Timestamp | Data de atualização | Nulo |
| deleted\_at | Timestamp | Data de eliminação | Nulo |

**Tabela Users**

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo | Tipo | Descrição | Restrições |
| id | Auto-increment | Identificador único | Chave primária |
| email | Varchar (100) | Nome da role | Obrigatório, único |
| role\_id | Integer | ID identificador da role | Obrigatório, chave estrangeira, único |
| username | Varchar (30) | Nome de utilizador | Obrigatório |
| password | Varchar (255) | Senha hash (bcrypt) | Obrigatório |
| nif | Integer (9) | Número identificação fiscal | Obrigatório |
| birthday | Date | Data nascimento | Obrigatório |
| last\_login\_at | Timestamp | Data último login | Automática |
| email\_verified\_at | Timestamp | Data de verificação | Nulo |
| new\_user | Boolean | Novo Utilizador | Padrão: 1 |
| user\_blocked | Boolean | Utilizador bloqueado | Padrão: 0 |
| blocked\_at | Timestamp | Data de bloqueio | Nulo |
| remember\_token | Boolean | Lembrar conta | Nulo |
| created\_at | Timestamp | Data de criação | Automática |
| updated\_at | Timestamp | Data de atualização | Nulo |
| deleted\_at | Timestamp | Data de eliminação | Nulo |

**Tabela Nacionalidades**

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo | Tipo | Descrição | Restrições |
| id | Auto-increment | Identificador único | Chave primária |
| name | Varchar (25) | Nome da nacionalidade | Obrigatório, único |
| continent | Varchar (255) | Nome do continente | Obrigatório |
| created\_at | Timestamp | Data de criação | Automática |
| updated\_at | Timestamp | Data de atualização | Nulo |
| deleted\_at | Timestamp | Data de eliminação | Nulo |

**Tabela Clientes**

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo | Tipo | Descrição | Restrições |
| id | Auto-increment | Identificador único | Chave primária |
| user\_id | Integer | Identificador do utilizador | Obrigatório, chave estrangeira, único |
| nationality\_id | Integer | Identificador da nacionalidade | Nulo, chave estrangeira |
| address | Varchar (100) | Morada completa | Nulo |
| first\_name | Varchar (50) | Primeiro nome do cliente | Nulo |
| last\_name | Varchar (50) | Último nome do cliente | Nulo |
| gender | Varchar (20) | Descrição do género | Nulo |
| contact | Integer (9) | Número de contacto | Nulo, único |
| newsletter | Boolean | Receber emails promocionais | Padrão: 0 |
| created\_at | Timestamp | Data de criação | Automática |
| updated\_at | Timestamp | Data de atualização | Nulo |
| deleted\_at | Timestamp | Data de eliminação | Nulo |

**Tabela Empresas**

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo | Tipo | Descrição | Restrições |
| id | Auto-increment | Identificador único | Chave primária |
| user\_id | Integer | Identificador do utilizador | Obrigatório, chave estrangeira |
| address | Varchar (100) | Morada completa | Nulo |
| name | Varchar (30) | Primeiro nome do cliente | Obrigatório, único |
| contact | Integer (9) | Número de contacto | Nulo, único |
| newsletter | Boolean | Receber emails promocionais | Padrão: 0 |
| created\_at | Timestamp | Data de criação | Automática |
| updated\_at | Timestamp | Data de atualização | Nulo |
| deleted\_at | Timestamp | Data de eliminação | Nulo |

**Tabela Campos**

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo | Tipo | Descrição | Restrições |
| id | Auto-increment | Identificador único | Chave primária |
| company\_id | Integer | Identificador da empresa | Obrigatório, chave estrangeira |
| name | Varchar (120) | Nome do campo | Obrigatório, único |
| price\_hour | Double (10, 2) | Preço do campo | Obrigatório |
| type\_floor | Varchar (50) | Qual é o tipo de piso do campo | Obrigatório |
| illumination | Boolean | Indica se o campo tem iluminação | Padrão: 0 |
| cover | Boolean | Indica se o campo tem cobertura | Padrão: 0 |
| shower\_room | Boolean | Indica se o campo tem balneário | Padrão: 0 |
| lockers | Boolean | Indica se o campo tem cacifos | Padrão: 0 |
| rent\_equipment | Boolean | Indica se o campo tem aluguer de equipamento | Padrão: 0 |
| status | Varchar (50) | Estado do campo |  |
| last\_maintenance | Date | Data da última manutenção | Nulo |
| file\_path | Varchar | Receber emails promocionais | Padrão: 0 |
| created\_at | Timestamp | Data de criação | Automática |
| updated\_at | Timestamp | Data de atualização | Nulo |
| deleted\_at | Timestamp | Data de eliminação | Nulo |

**Tabela Métodos Pagamento**

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo | Tipo | Descrição | Restrições |
| id | Auto-increment | Identificador único | Chave primária |
| description | Varchar (50) | Descrição do método de pagamento | Obrigatório |
| created\_at | Timestamp | Data de criação | Automática |
| updated\_at | Timestamp | Data de atualização | Nulo |
| deleted\_at | Timestamp | Data de eliminação | Nulo |

**Tabela Pagamentos**

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo | Tipo | Descrição | Restrições |
| id | Auto-increment | Identificador único | Chave primária |
| payment\_method\_id | Integer | Identificador do método de pagamento | Obrigatório, chave estrangeira |
| amount | Decimal (10, 2) | Total de pagamento | Obrigatório |
| status | Varchar (30) | Estado do pagamento | Padrão: Pendente |
| payment\_date | Date | Número de contacto | Nulo |
| created\_at | Timestamp | Data de criação | Automática |
| updated\_at | Timestamp | Data de atualização | Nulo |
| deleted\_at | Timestamp | Data de eliminação | Nulo |

**Tabela Reservas**

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo | Tipo | Descrição | Restrições |
| id | Auto-increment | Identificador único | Chave primária |
| user\_id | Integer | Identificador do utilizador | Obrigatório, chave estrangeira, única |
| status | Varchar (30) | Estado do pagamento | Padrão: Pendente |
| additional\_info | Varchar (255) | Informação acerca da reserva | Nulo |
| start\_date | DateTime | Data de início de reserva | Obrigatório |
| end\_date | DateTime | Data de fim de reserva | Obrigatório |
| total | Decimal (10, 2) | Total da reserva | Obrigatório |
| privacy\_policy | Boolean | Aceitar termos de privacidade | Nulo |
| created\_at | Timestamp | Data de criação | Automática |
| updated\_at | Timestamp | Data de atualização | Nulo |
| deleted\_at | Timestamp | Data de eliminação | Nulo |

**Tabela CamposReservas**

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo | Tipo | Descrição | Restrições |
| id | Auto-increment | Identificador único | Chave primária |
| field\_id | Integer | Identificador do campo | Obrigatório, chave estrangeira |
| reservation\_id | Integer | Identificador da reserva | Obrigatório, chave estrangeira |
| start\_date | DateTime | Data de início de reserva campo | Obrigatório |
| end\_date | DateTime | Data de fim de reserva do campo | Obrigatório |
| created\_at | Timestamp | Data de criação | Automática |
| updated\_at | Timestamp | Data de atualização | Nulo |

**Tabela Cancelamentos**

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo | Tipo | Descrição | Restrições |
| id | Auto-increment | Identificador único | Chave primária |
| reservation\_id | Integer | Identificador da reserva | Obrigatório, chave estrangeira |
| reason | Varchar (255) | Razão do cancelamento | Nulo |
| total\_refunded | Decimal (10, 2) | Total devolvido | Obrigatório |
| status | Varchar (255) | Estado do cancelamento | Obrigatório |
| cancellation\_date | Date | Data de cancelamento | Nulo |
| created\_at | Timestamp | Data de criação | Automática |
| updated\_at | Timestamp | Data de atualização | Nulo |

**Tabela Promoções**

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo | Tipo | Descrição | Restrições |
| id | Auto-increment | Identificador único | Chave primária |
| description | Varchar (100) | Descrição da promoção | Obrigatório |
| promo\_code | Varchar (20) | Código promocional | Obrigatório |
| usage\_limit | Integer | Quantidade de vezes que pode ser usado | Padrão: 1 |
| min\_spend | Decimal (10, 2) | Mínimo a ser gasto para aplicar a promoção | Padrão: 0.00 |
| discount | Decimal (10, 2) | Desconto a ser aplicado na reserva | Obrigatório |
| for\_inactive\_users | Boolean | Data de início de reserva | Padrão: 0 |
| for\_new\_users | Boolean | Data de fim de reserva | Padrão: 0 |
| additional\_info | Varchar (255) | Informações adicionais da reserva | Nulo |
| start\_date | Date | Data de início da promoção | Padrão: Hoje |
| end\_date | Date | Data de fim da reserva | Obrigatório |
| generic | Boolean | Anúncio genérico ou nao | Padrão: 1 |
| active | Boolean | Aceitar termos de privacidade | Padrão: 0 |
| created\_at | Timestamp | Data de criação | Automática |
| updated\_at | Timestamp | Data de atualização | Nulo |
| deleted\_at | Timestamp | Data de eliminação | Nulo |

**Tabela ClientesPromoções**

|  |  |  |  |
| --- | --- | --- | --- |
| Atributo | Tipo | Descrição | Restrições |
| id | Auto-increment | Identificador único | Chave primária |
| client\_id | Integer | Identificador do cliente | Obrigatório, chave estrangeira |
| promotion\_id | Integer | Identificador da promoção | Obrigatório, chave estrangeira |
| created\_at | Timestamp | Data de criação | Automática |
| updated\_at | Timestamp | Data de atualização | Nulo |

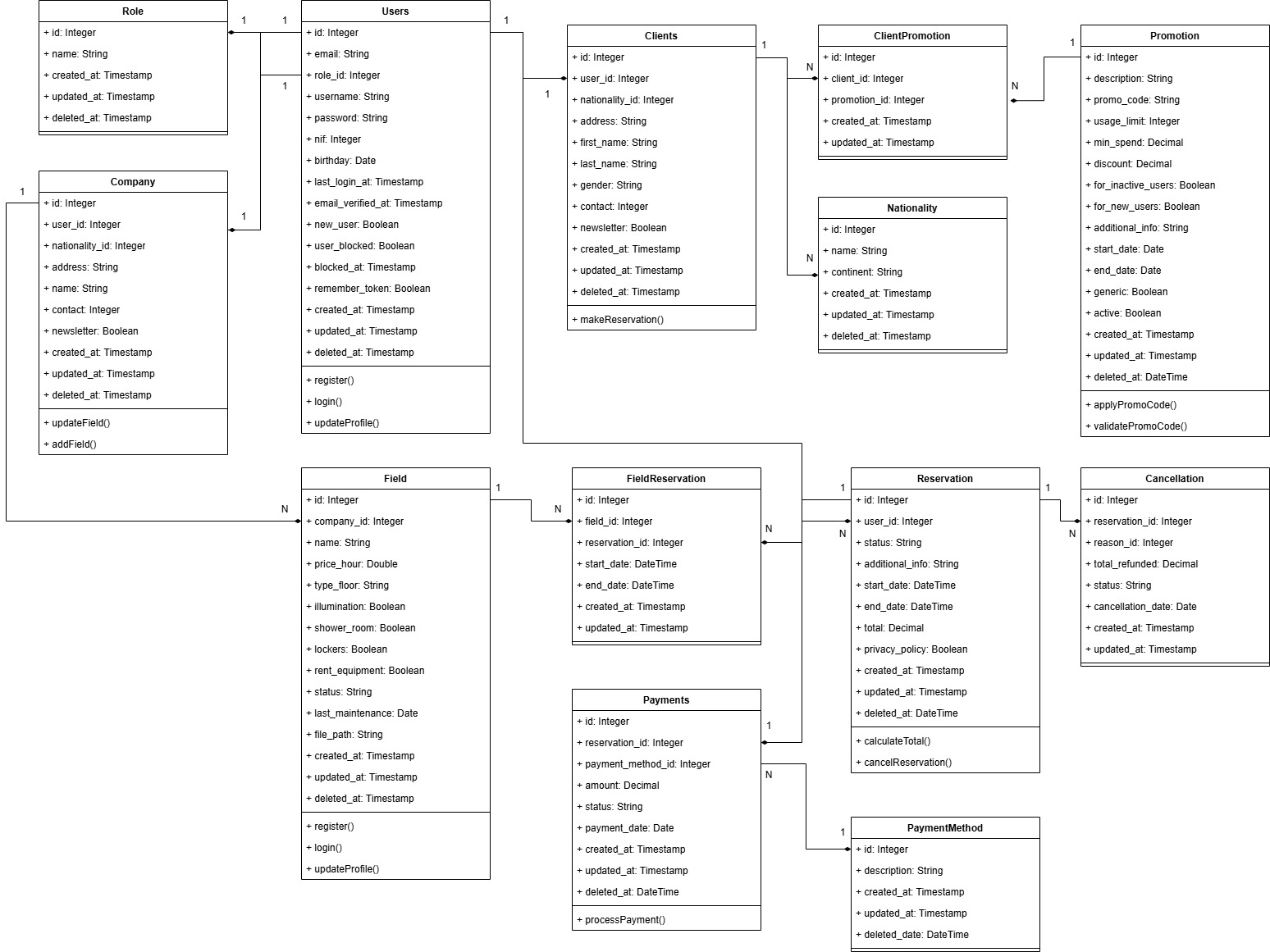
**Relações das Tabelas**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Primeira Tabela** |  | **Ligação** |  | **Segunda Tabela** |
| Utilizadores | 1 |  | 1 | Cliente |
| Utilizadores | 1 |  | 1 | Empresa |
| Roles | 1 |  | N | Utilizadores |
| Nacionalidades | 1 |  | N | Clientes |
| Empresa | 1 |  | N | Campos |
| Campo | N |  | N | Reservas |
| Pagamentos | N |  | 1 | Método Pagamentos |
| Promoções | N |  | N | Clientes |
| Reserva | 1 |  | 1 | Pagamentos |
| Cancelamentos | 1 |  | 1 | Reservas |

## Diagrama de classes

O diagrama de classes fornece uma visão detalhada da estrutura lógica do sistema, apresentando as entidades, seus atributos e métodos, além das relações entre elas.

Este diagrama é fundamental para compreender como os componentes do sistema interagem, facilitando a manutenção e expansão do software. As associações definidas no diagrama garantem a coesão e a integridade das operações, refletindo as regras de negócio de forma clara e eficiente.



# **DESENVOLVIMENTO**

## Arquitetura da aplicação

Para o desenvolvimento deste projeto, adotámos o padrão d e design MVC (Model-View-Controller), uma escolha fundamentada na sua eficácia comprovada e na experiência prévia da equipa. Esta arquitetura proporciona uma separação clara entre as camadas de apresentação (View), lógica de negócios (Controller) e acesso a dados (Model), facilitando a manutenção e escalabilidade do sistema.

As ferramentas tecnológicas foram cuidadosamente selecionadas, integrando o Laravel para implementação do padrão MVC no back-end, o Angular para desenvolvimento de uma interface dinâmica e responsiva no front-end, e o MySQL para uma gestão eficiente da base de dados. A estilização foi implementada com CSS, garantindo um design responsivo e uma experiência de utilizador otimizada. Por fim o Git foi utilizado como sistema de controlo de versão para gestão colaborativa do código.

Esta combinação permite-nos desenvolver uma aplicação robusta e escalável.. O design responsivo garante a compatibilidade com diversos dispositivos, enquanto a arquitetura modular facilita futuras expansões e manutenções.

## Tecnologias

Para o desenvolvimento deste projeto, utilizámos várias ferramentas e tecnologias que, em conjunto, garantiram um fluxo de trabalho eficiente e uma aplicação robusta:

**Desenvolvimento Web e Frameworks**

* Laravel: Framework PHP para implementação da arquitetura MVC e gestão do back-end.
* Angular: Framework JavaScript para desenvolvimento da interface web dinâmica e interativa
* HTML: Linguagem de marcação para estruturação do conteúdo
* CSS: Linguagem de estilização para design responsivo

**Ambiente de Desenvolvimento e Base de Dados**

* XAMPP: Aplicação que incluí Apache, MySQL e PHP para ambiente de desenvolvimento local.
* MySQL: Sistema de gestão de base de dados relacional.
* PHPMyAdmin: Interface gráfica para administração da base de dados

**Controlo de Versão e Colaboração**

* Git: Sistema de controlo de versão para gestão de código.
* GitHub: Plataforma de hospedagem e colaboração de código.

**Planeamento e Design**

* Drawio: Criação de diagramas e fluxogramas
* Figma: Design e prototipagem de interfaces
* Trello: Gestão e organização de tarefas

**Comunicação e Testes**

* Microsoft Teams: Plataforma principal para reuniões
* Discord: Plataforma de comunicação rápida através de texto, voz e vídeo e partilha de documentos.
* Postman: Ferramenta especializada parta testes e depuração de APIs

Este conjunto de ferramentas proporcionou uma base sólida para o desenvolvimento estruturado do projeto, facilitando a colaboração da equipa e garantindo a qualidade do produto final.

## Processo de Desenvolvimento

O processo de desenvolvimento do projeto foi estruturado utilizando a metodologia Ágil, com foco em sprints curtas e bem definidas.

Cada sprint tinha um conjunto específico de tarefas e objetivos, permitindo um desenvolvimento progressivo e iterativo da aplicação.

A seguir, apresentamos um resumo das atividades realizadas em cada sprint:

## Sprint I – 21 outubro a 25 outubro

O projeto teve início com a organização da equipa e a atribuição das primeiras tarefas. Catarina e o Rafael focaram-se no desenvolvimento da apresentação do site, enquanto o Simão, o Hélder e o Afonso trabalharam na lógica e funcionalidade da aplicação.

Durante esta fase, foram definidas as cores e a tipografia do site, criando uma identidade visual consistente. A Catarina e o Rafael iniciaram o desenvolvimento do website no Figma e construíram parte da estrutura base em Visual Studio Code. O Simão ficou responsável pela criação da base de dados e pelo design do logótipo. O Hélder elaborou a UML relativa à funcionalidade de pesquisa de campos, e o Afonso preparou a UML relativamente às promoções.

## Sprint II – 28 outubro a 1 novembro

Com a estrutura inicial definida, a Catarina e o Rafael avançaram na construção do site em Visual Studio Code. O Simão criou as UML’s para gerir reservas de clientes, criar novas reservas e adicionar campos de padel ao sistema. O Hélder dedicou-se à UML de notificações, enquanto Afonso desenvolveu a UML correspondente ao feedback das reservas. Esta sprint teve um foco maior no planeamento das funcionalidades essenciais para o funcionamento do sistema.

## Sprint III – 4 novembro a 8 novembro

Esta etapa foi marcada pela exploração da integração entre Angular e Laravel. O Hélder finalizou UML’s pendentes e iniciou o desenvolvimento do CRUD de pagamentos. O Simão e o Afonso testaram API Laravel e Postman, além de trabalharem na atualização de Laravel, PHP e Node.js. A Catarina e o Rafael começaram a estudar Angular e a realizar testes de integração com Laravel, enquanto o Afonso desenvolveu os CRUD’s de empresas e campos, e Simão implementou CRUD’s para clientes, roles e nacionalidades.

## Sprint IV – 11 novembro a 15 novembro

O foco principal foi a construção inicial do website em Angular. A Catarina e o Rafael começaram a desenvolver a página inicial e o dashboard, enquanto o Simão trabalhou no CRUD de utilizadores, registo e login, explorando o sistema de autenticação com Passport. O Hélder e o Afonso dedicaram-se à integração e revisão de código. Por fim, a equipa uniu esforços para testar os servidores online e assegurar o bom funcionamento das implementações, salientando a maior contribuição para o Rafael

## Sprint V – 18 novembro a 22 novembro

Neste sprint, o Hélder finalizou os CRUD’s de pagamentos e métodos de pagamento, enquanto a Catarina avançou na criação de páginas informativas

em Angular e iniciou o relatório intermédio. O Rafael completou a implementação de CRUD’s no dashboard Angular, o Afonso desenvolveu o CRUD de reservas, e o Simão adaptou o código Laravel para integração com Angular.

A equipa começou a solidificar o núcleo funcional do projeto.

## Sprint VI – 25 novembro a 29 novembro

O Hélder e o Afonso implementaram o envio de emails no back-end, enquanto o Simão prosseguiu com o CRUD de promoções e avançou no desenvolvimento do dashboard em Angular. O Afonso ainda desenvolveu o CRUD de cancelamentos. O Rafael dedicou-se à revisão do código Angular, e a Catarina trabalhou na criação de um catálogo de campos. Este sprint teve como objetivo preparar as bases para funcionalidades mais avançadas e garantir a coesão do código.

## Sprint VII – 2 dezembro a 6 dezembro

As atividades desta sprint envolveram a implementação de reservas no carrinho e o envio de emails pelo Afonso e Rafael. O Simão corrigiu erros de formulário e revisou o código do dashboard, e, juntamente com o Afonso e o Rafael trabalharam na implementação de roles. A Catarina desenvolveu a

página de pagamentos em Angular e preparou a apresentação final, enquanto o Hélder avançou com a integração de pagamentos.

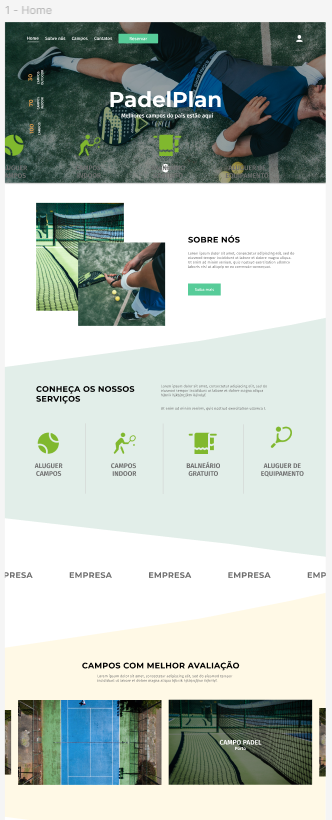
## Sprint VIII – 9 dezembro a 11 dezembro

O sprint final foi dedicado à revisão de código, à conclusão do relatório e à preparação da apresentação do projeto. A equipa focou-se em consolidar o trabalho realizado nas sprints anteriores, corrigindo possíveis erros e ajustando os detalhes para garantir a entrega de um produto final de qualidade.

## Interfaces de Utilizador: Protótipo VS Produto Final

O design das nossas interfaces foi desenvolvido com um foco duplo: atender aos requisitos do cliente e proporcionar uma experiência excecional ao utilizador final. Através de um UI/UX cuidadosamente planeado, conseguimos criar uma interface intuitiva e visualmente atrativa, que equilibra funcionalidade com simplicidade.

**Página Inicial**

****

**Login e Registo**

**Sobre Nós**

**Contactos**

**Dashboard**

# **CONCLUSÃO**

## Objetivos Cumpridos

## Contratempos / Dificuldades

Durante o desenvolvimento do projeto foram surgindo alguns desafios que nos conseguiram pôr à prova.

Inicialmente enfrentamos problemas com a integração do passport porque não estávamos a conseguir fazer a manipulação de tokens, mas conseguimos superar essa barreira.

Já no final do projeto enfrentámos alguns problemas com os pagamentos e abordámos o cliente sobre a possibilidade de colocar pagamento no ato, o que nos foi concebida luz verde para avançar.

## Possíveis Melhorias

O projeto atual do PadelConnect está bastante robusto, mas existem sempre áreas onde é possível melhorar, uma delas seria a paginação entre as páginas.

Inicialmente o volume de dados que tínhamos no site era mínimo, daí a ausência da paginação, no entanto, com o crescimento dos dados reparamos que seria necessário inserir uma forma de navegação mais organizada. Uma vez que esse reparo foi dado no final do projeto, e dado o seu impacto no projeto, decidimos manter sem paginação.

Outra melhoria que deveria ser implementada numa futura atualização seria a implementação de métodos de pagamento como MBWay ou Cartão.

## Considerações Finais

O projeto foi concluído com êxito, alcançando todos os objetivos planeados. A aplicação desenvolvida foi ao encontro das expectativas do cliente, possibilitando uma maior eficiência na gestão de campos de Padel.

A equipa trabalhou de forma organizada o que foi fundamental para o cumprimento das tarefas. A comunicação foi um fator importante entre a equipa embora não tenha sido o nosso ponto forte inicialmente, no entanto trabalhamos sempre com foco na colaboração. A integração de ideias permitiu que cada sprint fosse concluído dentro dos prazos.

Este projeto não só pode trazer uma contribuição significativa para o mercado de padel, como também é um importante passo no nosso desenvolvimento como profissionais. A satisfação do nosso cliente para com o projeto representa o impacto positivo e a importância de criar aplicações que sejam funcionais, acessíveis e criativas. Sentimo-nos realizados por entregar esta aplicação que faz a diferença e que reflete o nosso compromisso com a excelência e competência na área da programação.