Universidade do Minho - Mestrado em Sistemas de Informação

UC - Engenharia de Software para Sistemas de Informação 2023/2024 – M3,

Modelação de um Sistema de Reserva de Campos de Padel

1.Introdução ao problema

O objetivo deste trabalho passa por desenvolver um sistema de informação para um problema que se tenha identificado no ecossistema social e que seja complexo o suficiente para merecer atenção e justificação de desenvolvimento de um sistema de informação para o mesmo.

Com esse intuito, identificamos um clube de padel, onde neste momento, o sistema de reservas funciona através de email ou chamada telefónica, significando isso que esse funcionamento pode ser otimizado através de um sistema de informação.

O clube Bpadel, em Braga, oferece uma variedade de campos distribuídos por duas áreas distintas da cidade e disponibiliza não só campos para jogos ocasionais, mas também para aulas. Essa diversidade na oferta requer a participação de diversos utilizadores/jogadores para que o negócio se mantenha viável, enquanto que envolve uma complexa interação entre outros intervenientes/atores. Apesar desta variedade de interações, o sistema de reserva dos campos e tipos de utilização ainda é precário devido a estar dependente do tempo e disponibilidade de um terceiro (secretário da empresa) que atenda ou esteja atento ao email para assim efetuar a reserva. Essas características tornam o clube Bpadel num local onde é possível equacionar o desenvolvimento e otimização de um sistema de informação capaz de ajudar a solucionar os problemas enumerados.

Neste contexto surgiu então a necessidade de desenvolver uma aplicação que simplifique o processo de reservas para todos os utilizadores (jogadores e administradores dos campos), enquanto, oferece uma experiência acessível e fácil de utilizar, disponível a qualquer hora e em qualquer local através de qualquer dispositivo móvel.

A opção recaiu por elaborar um sistema adaptado à realidade desta empresa, contudo o processo de reserva como é transversal a outros cenários (outros desportos, espetáculos, imóveis), faz com o sistema possa ser escalável e adaptado a outras realidades com processos semelhantes de reserva.

De forma a focar mais o desenvolvimento do sistema e não o tornar muito abrangente para que este seja exequível, o sistema será analisado e elaborado exclusivamente em relação ao processo de reserva de campos; posteriormente, a reserva de aulas pode ser adicionada, contudo essa evolução vi certamente introduzir uma complexidade adicional ao exercício.

Este relatório foi elaborado para explicar as etapas da análise do problema até as intenções de concepção da aplicação/software. Inicia com a identificação do problema, seguido pelo método de desconstrução desse problema, através da definição de requisitos, osprocessos envolvidos e os modelos derivados. Em seguida, aborda o desafio tecnológico, ou seja, qual software que melhor atende aos requisitos do sistema, e, por último, os objetivos pretendidos.

2. Enquadramento e contextualização com o SWEBOK

O desenvolvimento deste sistema tem como base o enunciado no SWEBOK (Software Engineering Body of Knowledge), como tal é fundamental que se inicie por uma análise de requisitos de utilizador de forma a conseguir modelar e desenvolver um sistema capaz de atender esses requisitos de forma eficaz. Além dessa análise de requisitos (funcionais e não funcionais), é indicado também a elaboração de diferentes casos de uso de forma a equacionar diferentes possibilidades e definir os diferentes rumos que cada ação poderá tomar.

É portanto fundamental que se recolham os requisitos funcionais e não funcionais do sistema. Os requisitos funcionais de forma a ficarem claras as funções que pretende que o sistema execute, e as características necessários do sistema para que este seja capaz de executar essas funções, assim como uma definição clara das respostas que o sistema deve dar perante os diversos "inputs" que lhe podem ser dados. São requisitos fundamentais para que se consiga entregar um sistema que cumpre o propósito para o qual foi solicitado, uma vez que são a orientação para o desenho, o desenvolvimento e a validação final.

Não menos importantes são também os requisitos não funcionais que não são requisitos de acções que o sistema deve fazer, mas sim propriedades que o sistema deve ter para conseguir executar de forma correta os requisitos funcionais, como por exemplo o desempenho, a robustez, a segurança, a escalabilidade entre outros, que no fundo não representam ações mas sim propriedades para que as ações sejam devidamente desenvolvidas.

Por fim, o desenvolvimento de casos de uso é também importante para simular e descrever a interação do utilizador com o sistema, assim como definir os caminhos que o sistema irá tomar mediante cada ação que lhe seja feita.

O SWEBOK como base serve exatamente para que seja possível entregar um sistema que vai de acordo ao solicitado quer de ponto de vista de requisitos funcionais como de requisitos não funcionais, assim como é um auxiliar que tenta mitigar falhar e prever o máximo de possibilidades de forma a evitar falhas, gralhas ou erros graves.

3. Conceptualização do software

3.1 Identificação dos atores

2 grupos de atores:

- Jogadores (vários)
- Administração/secretariado

3.2 Identificação e análise de requisitos

É importante para uma boa análise dos requisitos que estes sejam detalhados e completos (não vagos), que sejam coerentes e que sejam exequíveis tendo em conta o que é solicitado,

Neste caso enumerar os seguintes requisitos:

- Requisito 1: O sistema deve permitir que os utilizadores façam o "login" com a sua conta de utilizador, ou então que criem uma conta no caso de não terem (Funcional).

É importante que o processo de criação de conta de utilizador seja rápido, com o mínimo de campos possível para preenchimento e intuitivo (Não funcional).

- Requisito 2: O sistema deve ser capaz de apresentar um calendário com os horários livres e os horários ocupados, assim como o preço do horário livre (Funcional).

Esta revelação do calendário deve ser atualizada ao momento de forma a que enquanto um jogador pesquisa um horário, se outro reservar, esse mesmo horário passe a ocupado e não permita sobreposição de reservas (Não funcional).

- Requisito 3: O sistema deve ser capaz de permitir que o utilizador escolha um horário livre e reservá-lo (Funcional).

O tempo de resposta para iniciar o processo de reserva assim que selecionado o horário deve ser curto (Não funcional).

- Requisito 4: O sistema deve permitir que após a reserva o utilizador faça o pagamento da sua reserva e este passe ao estado confirmado (Funcional).

É imperioso garantir a segurança das transações monetárias e dos dados recolhidos (Não funcional).

Os requisitos avançados passaram pela elaboração de uma versão base do sistemas que fossemos capazes de implementar, ou seja são os requisitos mínimos para fazer sentido desenvolver o sistema, porém é claro que este sistema pode ser muito mais poderoso, integrar muito mais requisitos funcionais e muito adaptativo a muitos negócios.

- Requisito 5: O sistema deve apresentar um separador com os dados de contacto e morada dos locais onde se encontra a empresa (Funcional).

É importante o sistema ser escalável de forma a no futuro a empresa poder ter mais campos e os conseguir incluir no sistema (Não funcional).

3.3 Priorização dos requisitos

Com base no SWEBOK, é também importante priorizar requisitos de forma a que o desenvolvimento do sistema seja focado nos requisitos considerados mais importantes para que seja entregue mais valor a quem solicitou o sistema. É importante considerar quais os fatores mais relevantes face a outros e seguir essa hierarquia.

1. **Priorização dos requisitos funcionais**

Requisito	Importância para cliente	Dificuldade de implementação	Prioridade
O sistema deve permitir que os utilizadores façam o "login" com a sua conta de utilizador, ou então que criem uma conta no caso de não terem.	Alta	Média	1
O sistema deve ser capaz de apresentar um calendário com os horários livres e os horários ocupados, assim como o preço do horário livre	Alta	Alta	2
O sistema deve permitir que após a reserva o utilizador faça o pagamento da sua reserva e este passe ao estado confirmado	Alta	Alta	3
O sistema deve ser capaz de permitir que o utilizador escolha um horário livre e reservá-lo	Alta	Média	4
O sistema deve apresentar um separador com os dados de contacto e morada dos locais onde se encontra a empresa	Média	Baixa	6

3.3.2 Priorização dos requisitos não funcionais

Requisito	Importância para cliente	Dificuldade de implementação	Prioridade
Segurança dos dados	Alta	Alta	1
Atualização imediata	Alta	Média	2
Rapidez e simplicidade	Média	Média	3
Responsividade	Alta	Alta	4

4. Diagramas

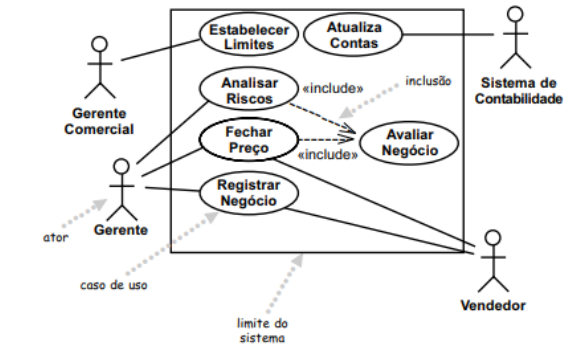
Os diagramas desenvolvidos dentro deste capitulo, seguem o padrão e linguagem de UML, principalmente consultado no livro 'UML-essecials' contido na bibliografia, de forma a estabelecer uma leitura e representação mais coerente e modular para qualquer profissional ou conhecedor da matéria.

4.1 Diagramas de Casos de Uso

No intuito de corresponde aos requisitos identificados e de iniciar a estruturação do software, em primeiro, recorremos a diagrama de casos de uso. Estes são uma representação gráfica da interação entre atores externos ao sistema (usuários ou entidades) e as funcionalidades (casos de uso) associadas ao próprio sistema. Essa ferramenta desempenha um papel fundamental na modelação de sistemas na engenharia de software e está normalmente vinculada à representação UML (Unified Modeling Language). Esta linguagem de representação, não só os permite representar o sistema, mas também comunicá-lo de uma forma padronizada, para haver uma maior compreensão dos códigos de representação existem imagens de associação com o livro 'UML-essencials', colocadas em anexo, que também acontece para os outros diagramas.

O principal propósito dos diagramas de casos de uso é oferecer uma visão abrangente das funcionalidades do sistema a partir da perspetiva dos utilizadores/jogadores/atores. Contudo, este concentra-se, apenas, nas interações representativas. Estes diagramas são ferramentas no processo de desenvolvimento de software, auxiliando no equilíbrio e funcionalismo da estrutura do sistema no que remete à relação entre os requisitos dos atores e das funcionalidades do sistema. Esta representação, destaca-se principalmente na fase iniciais do projeto, permitindo a utilizar dos requisitos a estrutura do sistema e consequentemente a definição da arquitetura do sistema.

Visando compreender e comunicar os requisitos, avançamos para a criação de alguns diagramas de casos de uso, apresentando as funcionalidades da plataforma e das soluções geradas. Estes diagramas, são desenvolvidos perante dois atores: os jogadores e o administrador do clube de padel e explicam a própria utilização, formal, do sistema, nos seus vários separadores e funcionalidades. Para facilitar o paralelismo entre o que foi aplicado no protótipo e a linguagem UML, estes vão ser apresentados lado a lado.



GURA 9.2 Diagrama de casos de uso.

Imagem 1 – diagrama de usos em 'UML-essencials'

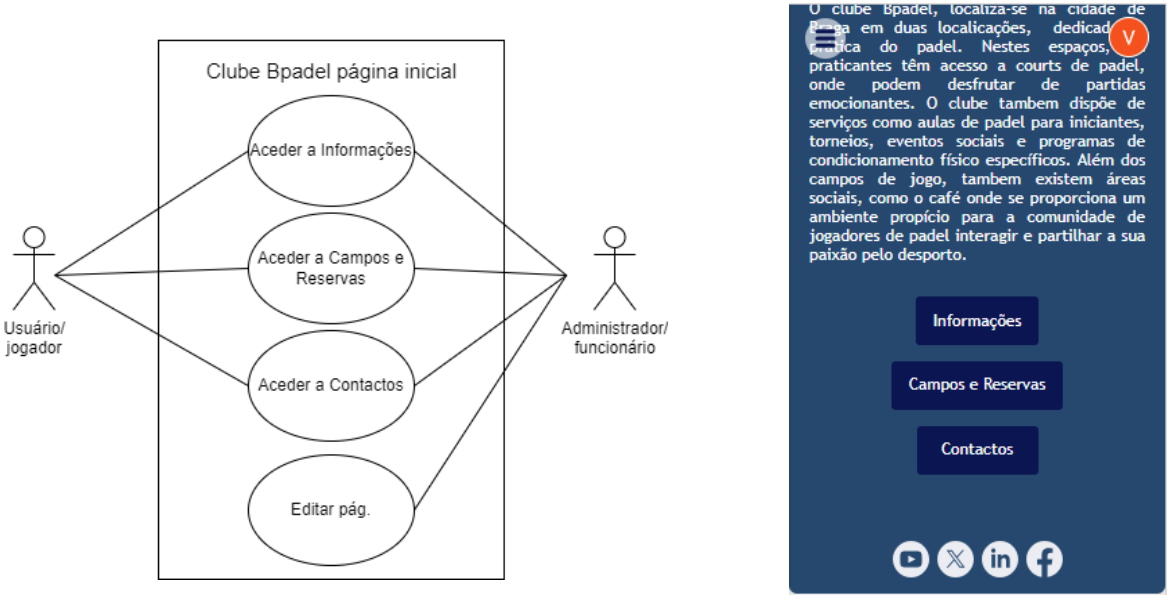


Figura 1: Diagrama de Usos da página principal

Nesta primeira instância, separador inicial, podemos então separar os casos de uso por:

1. Casos de uso do jogador:

- Ver Informações do Clube: Permite que aos jogadores acederem aos detalhes sobre o clube
- Ir para Campos e Reservas: Permite que aos jogadores irem para o menu dos campos e opções de reserva.
- Aceder menu de navegação: Permite aos jogadores acederem ao menu que lher permite seguir para outra página

2. Casos de uso do Administrador:

- Ir para campos e reservas: Permite que o administrador aceder ao separador dos campos e reservas e de informações e contactos
- Editar página: O administrador pode assim editar o aspeto e funcionalidades da página inicial.

Este diagrama expõe a primeira representação de como a app se apresenta para qualquer utilizador e a partir da qual podemos navegar no sistema. A partir desta página podemos ir para as informações, para obter as informações básicas, de funcionamento e localização do clube. Podemos também aceder a campos e reservas, para consultar os campos do clube e depois efetuar as reservas e por fim consultar também os contactos do clube, caso se queira efetuar algum pedido ou questão que não esteja determinada na própria aplicação.

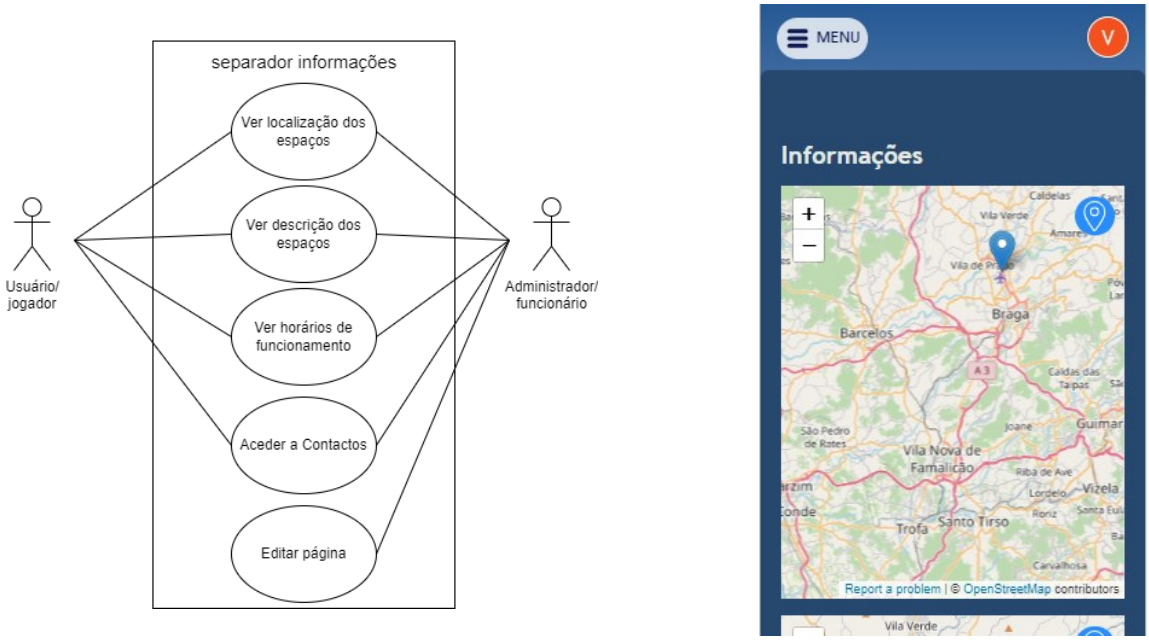


Figura 2: Diagrama de Usos do separador 'Informações'

Nesta segunda instância, após o acesso do menu principal, podemos ver as informações sobre o clube distribuídas por:

1. Casos de uso do jogador:

- Ver localização dos espaços: localização dos dois polos do clube na cidade através de mapa e as moradas em texto.
- Ver descrição dos espaços: existe uma descrição mais detalhada das instalações dos dois espaços, desde os campos que cada um contem e com um atalho (botão) para o separador dos campos e reservas, balneários e cafetaria.
- Ver horários de funcionamento dos espaços
- Ver tipos de serviços do clube: esta parte descreve que tipo de serviços é que os clubes dispõem (reservas individuais ou de grupo, aulas, aluguer de materiais e outros eventos,…)
- Aceder menu de navegação: Permite aos jogadores acederem ao menu que lhe permite seguir para outra página

2. Casos de uso do Administrador:

- Ver qualquer uma das opções dos jogadores
- Editar as informações.

Este diagrama apresenta assim como é que o clube através da aplicação passa algumas informações gerais para os jogadores. Para isso foram selecionadas informações que se ache pertinente para qualquer jogador que não conheça o clube ou que possa necessitar para saber a localização de cada um dos complexos do clube, com mapa interativo ou a morada escrita. Por outro lado, os horários de funcionamento do clube, em relação às especificidades dos dias da semana e feriados. Depois também consultar o tipo de serviços do clube, desde as reservas, as aulas, o aluguer de material e pormenores das instalações ou outros eventos. No caso do administrador tem possibilidade de verificar esta informação e fazer a sua edição.

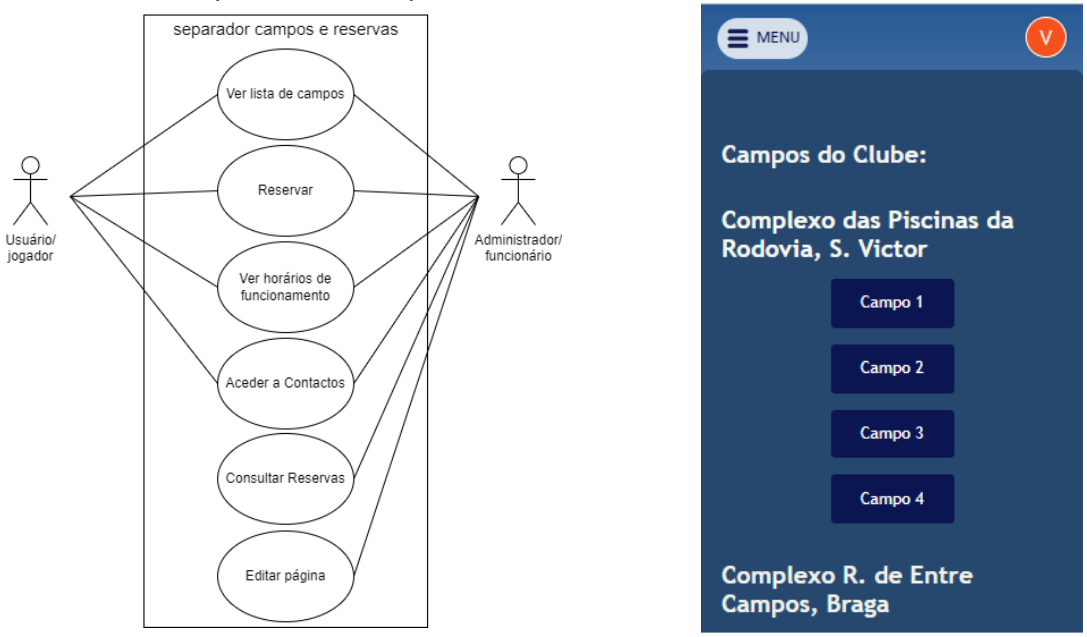


Figura 3 – Diagrama de Usos de 'Campos e Reservas'

No terceiro separador da app, apo acesso da página inicial, podemos ver a lista de Campos do clube e os horários de funcionamento:

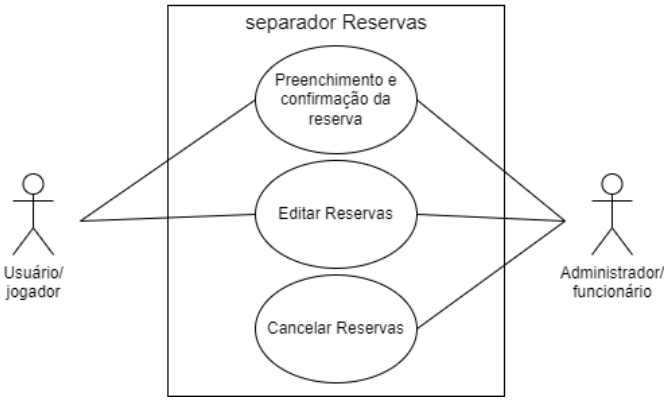
1. Casos de uso do jogador:

- Ver os campos – lista dos campos por localização/espaço com botão em cada um para reservar
- Ver horários de funcionamento dos campos
- Aceder menu de navegação: Permite aos jogadores acederem ao menu que lhe permite seguir para outra página

2. Casos de uso do Administrador:

- Consultar reservas
- Editar as informações dos campos ou reservas.

Este diagrama apresenta o separador dos 'campos e reservas' de onde podemos verificar a lista de campos que o clube contém separados pelos dois complexos, para facilitar a consulta. Mais uma vez, existe a referência ao horário de funcionamento, neste caso das reservas, visto que a última só vai até à hora anterior ao encerramento, para que se possa alugar o campo por uma hora e terminar quando encerra o clube.



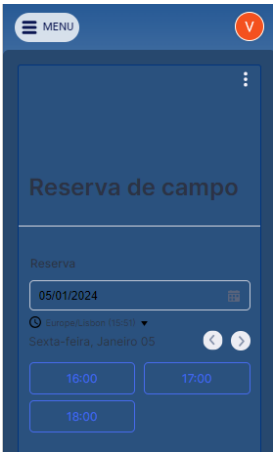


Figura 4 - Diagrama de Usos do separador 'Reservas'

Neste separador, acedido a partir do menu campos e reservas, poderemos efetuar a reserva do campo selecionado, anteriormente.

1. Casos de uso do jogador:

- Preenchimento das informações para completar a reserva (nome, morada, email, contacto e data e hora a reservar)
- Editar reserva – fazer alteração da reserva feita

2. Casos de uso do administrador:

- Preenchimento das informações para testar sistema
- Editar reserva – fazer alteração das reservas feitas comunicando com os jogadores
- Cancelar reservas – efetuar cancelamento em casos de necessidade

Este separador conectado a um formulário de preenchimento, obriga assim a que o utilizador preencha alguns dados básicos para que assim a reserva possa ser efetuada. Há uma premissa de segurança que obriga a que os dados sejam todos completos e nos formatos standard, para que a reserva seja completa. Por outro lado, a partir do momento que a reserva fique feita, já não é possível nos horários de seleção escolher essa hora de jogo, nesse campo – um sistema que depois é representado também nos diagramas de estado dos campos e das reservas.

Figura 5 – Diagrama de usos do separador 'Contactos'

No quinto separador do sistema, acedido por qualquer um dos outros (exceto o reservar) podemos ver a lista de Contactos do clube de padel segundo:

1. Casos de uso do jogador:

- Ver lista dr contactos: lista dos contactos gerais do clube de padel
- Aceder menu de navegação: Permite aos jogadores acederem ao menu que lher permite seguir para outra página

2. Casos de uso do Administrador:

- Ver lista dr contactos: lista dos contactos gerais do clube de padel
- Edição dos contactos

Este diagrama permite apresentar a lista de contactos do clube, que em caso de necessidade, por algum motivo que ultrapasse as funcionalidades diretas da aplicação possa servir para os jogadores contactarem a administração.

Apesar do software estar estruturado desta forma, devido ao conhecimento técnico de software que temos de momento, este deveria apresentar outras sequencia de passos por maior segurança. Estas diferenças foram consultadas na bibliografia (O'Docherty, M. 2005), (Anderson, R. 2008) e vídeos técnicos associados que assim destacariam pela existência de uma fase de login que assim serviria para proteção de ambas as partes, não só para proteção das informações pessoais, mas também para segurança do processo de reserva, quer para jogadores quer para a administração. Para finalizar esse processo deveria existir também uma fase de pagamento, de um sinal, para assegurar a reserva. Esse processo será comparado com a opção tomada nos diagramas de uso e na solução tecnologica e mais desenvolvido nos diagramas de atividades.

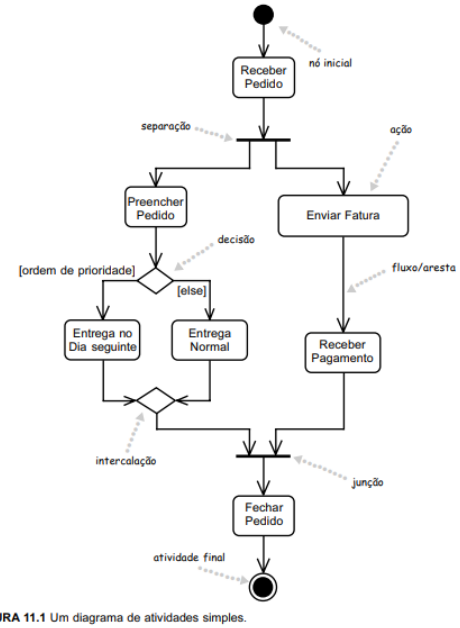
4.2 Diagramas de Atividade

Os diagramas de atividades constituem uma representação visual que possibilita a observação do fluxo de atividades em diferentes contextos, como processos, sistemas ou procedimentos, neste caso, relacionados com a reserva de campos de padel. Esses diagramas desempenham um papel fundamental na engenharia de software, oferecendo uma visão dinâmica do comportamento de um sistema e os passos necessários para efetuar uma função.

Esses diagramas são, principalmente, ferramentas visuais de representação que conseguem apresentar simplificadamente o fluxo de controle do sistema. A função base é descrever as sequências de ações, evidenciando as relações entre as partes. Quando aplicados a estruturação dos processos, esse tipo de diagrama oferece uma leitura entre as ações iniciais e as finais e a perceção das etapas que são necessárias para que o sistema fique completo e funcional. Depois deste será apresentado o de sequencias para ter uma leitura temporal mais imediata de um dos processos.

Apesar da sua simplicidade os diagramas de atividades mostram-se altamente transversais podendo ser aplicados em diversos sistemas de software com inúmeros propósitos.

No contexto da estruturação de sistemas, os diagramas de atividades desempenham um papel fundamental, tanto para declararem os requisitos como para os comunicarem e representarem para todas as partes envolvidas, eficientemente. Consequentemente, foram elaborados diagramas de atividades para representar as funcionalidades e estrutura do sistema solução para a reserva de campos de padel. Estes partem também da base do livro'UML-essencials'.



URA 11.1 Um diagrama de atividades simples.

Imagem 2 – diagrama de actividades em 'UML-essencials'

Neste primeiro momento, desenvolveu-se um diagrama de atividades baseado no processo de consulta das informações do clube Bpadel, apresentando o fluxo da consulta de informações, até se chegar aos horários. Este processo começa quando o jogador entra na página inicial da app depois se dirige às informações e posteriormente consulta os horários. Este processo representa o ato simples de visita à aplicação para ficar a conhecer o clube, que posteriormente pode voltar à pagina inicial e começar um processo de reserva, ou apenas a saída do utilizador.

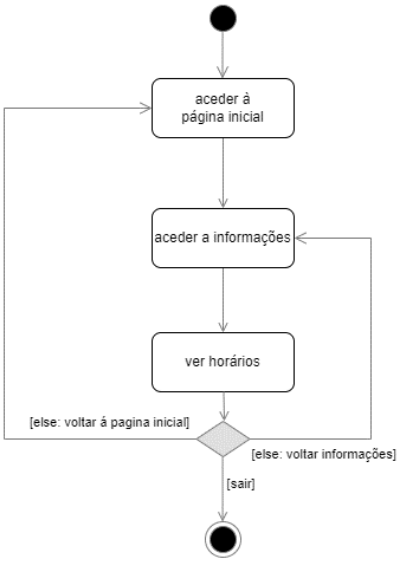
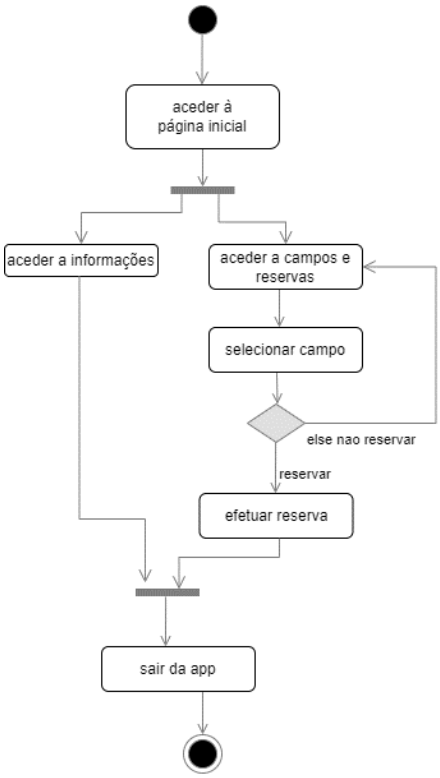


Figura 6 - Diagrama de atividades Consulta de Informações

De seguida, avançamos com o diagrama para efetuar reservas, que é de facto o propósito principal da solução. Este é o que determina a maior cadeia de ações na app e que de facto soluciona o problema base identificado. Este processo, inicia-se pala página principal ou pela de informações e termina com um preenchimento de um formulário com os dados do jogador/utilizador para completar a reserva.



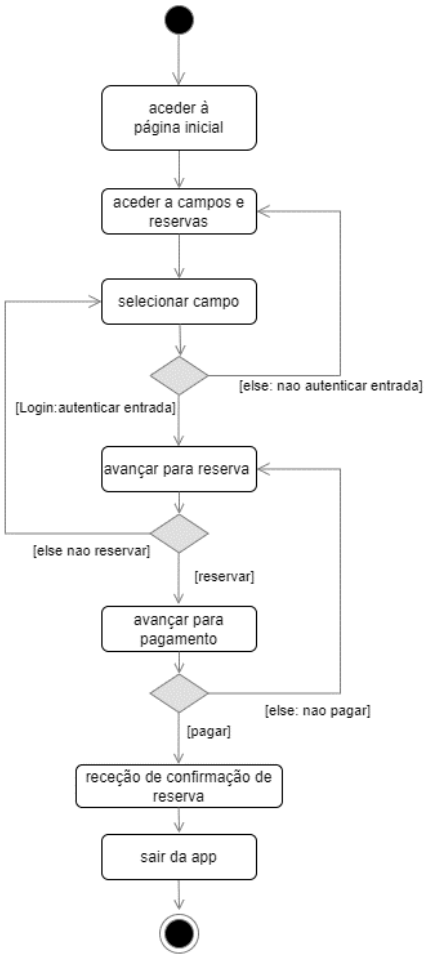


Figura 7 e 8 - Diagramas de atividades Efetuar Reserva (direita - como está a aplicação atualmente, esquerda - como otimizar em termos de segurança)

O caso da esquerda demonstra como funciona a app de momento e que a partir do menu inicial é possível seguir para várias páginas: informações contactos e campos e reservas.
Seguindo a parte dos campos e reservas, que era o que se pretendia explorar, podemos verificar a lista de campos a reservar, e a partir do momento que selecionamos um deles podermos avançar para a reserva, através do preenchimento de um formulário de dados e a submissão dos mesmos.

Como o tempo associado ao desenvolvimento tecnológico não nos permitiu aprimorar o artefacto da aplicação e os nossos conhecimentos antes deste projeto, em termos de programação eram quase nulos, apresentamos outro diagrama de atividades com a solução de reserva otimizada que poderá ser implementada no futuro. A diferença neste caso está relacionada com a adição de alguns sistemas de segurança que assim trazem uma maior integridade para o processo de reserva, não só para os jogadores, mas também para a administração. Estes melhoramentos passam por um processo de registo do utilizador e consequente login e também a adição de uma ação de pagamento para assegurar o verdadeiro interesse do processo de reserva.

Por último o diagrama de atividades relacionado com o processo de cancelamento de reserva, assim este permite perceber como se pode efetuar este processo e quais as diferenças para a reserva.

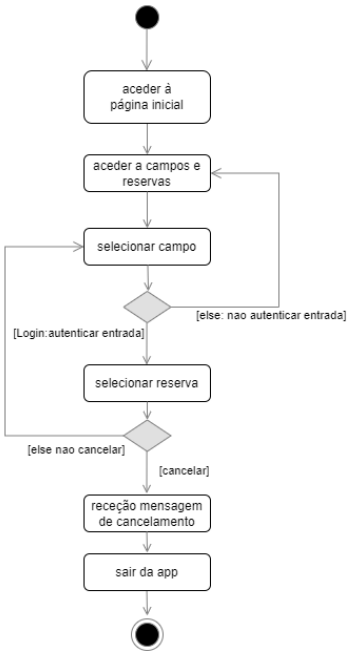


Figura 9 - Diagrama de atividades de cancelar reservas

Este diagrama permite-nos ver que o processo inicial ao cancelamento da reserva é semelhante ao da reserva mas a diferença acontece quando, após o login, temos acesso ás reservas feitas e selecionando uma delas, podemos efetuar alterações na mesma. Este processo é muito semelhante ao que acontece no caso do administrador, só que enquanto o utilizador so tem acesso ás reservas do seu próprio login, o administrador tem acesso a todas. No protótipo tecnológico este processo não acontece desta forma, visto que não conseguimos a interação do utilizador com a base de dados, apenas o administrador as pode cancelar, a partir da base de dados.

4.3 Diagrama de Sequencia

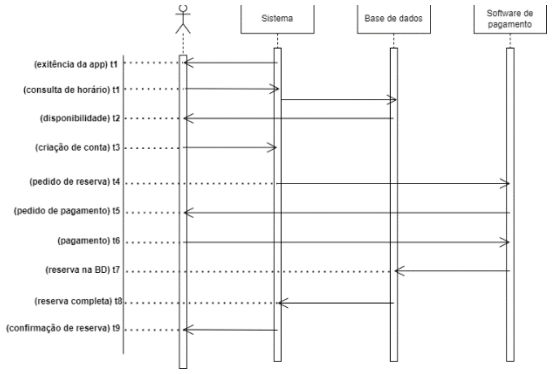


Figura 10 - Diagrama de sequência do processo de reserva (versão inicial)

Um diagrama de sequência, representa sequência de ações para fazer uma reserva e a interação entre objetos ao longo do tempo. Este é útil para nos demonstrar o fluxo de ações entre objetos que permite efetuar uma reserva, desde a entrada do utilizador no sistema/aplicação até a confirmação a alteração da reserva.

A partir dos passos delineados no diagrama de casos de uso, que se inicia com o acesso à aplicação e consulta de horários, avançamos para o processo de reserva. Este começa no t3 com o registo do utilizador no sistema, seguido pelo pedido de reserva no t4, que percorre o sistema (identificação de campo e horário), a base de dados (verificação de disponibilidade) e chega ao software de pagamento. No t5, o software de pagamento solicita o pagamento ao utilizador, que realiza ou não o pagamento no t6. Este processo de pagamento é implementado para assegurar a integridade da intenção de reserva.

Após esta etapa, o software de pagamento desencadeia uma série de ações para confirmar e bloquear a reserva nos outros elementos do sistema: para a base de dados no t7 e desta para o sistema no t8. Finalmente, a confirmação é enviada ao utilizador através do sistema no t9. Contudo, foi adicionada uma ação adicional que reinicia o processo: alterar ou cancelar a reserva no t10, desencadeando consequentemente um novo ciclo.

4.4 Diagrama de Classes:

O diagrama de classe do UML (Unified Modeling Language) é amplamente utilizado para modelar a estrutura estática de um sistema, incluindo suas classes, atributos, métodos e as relações entre essas classes. Embora o diagrama de classe não seja especificamente destinado a representar a base de dados, ele desempenha um papel fundamental na modelagem da estrutura de dados que será utilizada pelo sistema, o que, por consequência, pode incluir a representação da base de dados.

No contexto em que se deseja identificar a base de dados acessada para ser mostrada na interface, o diagrama de classe pode ser estendido ou complementado com outras ferramentas do UML, como o diagrama de componente ou o diagrama de pacote. Esses diagramas auxiliam na representação de módulos ou componentes do sistema, incluindo conexões com a base de dados.

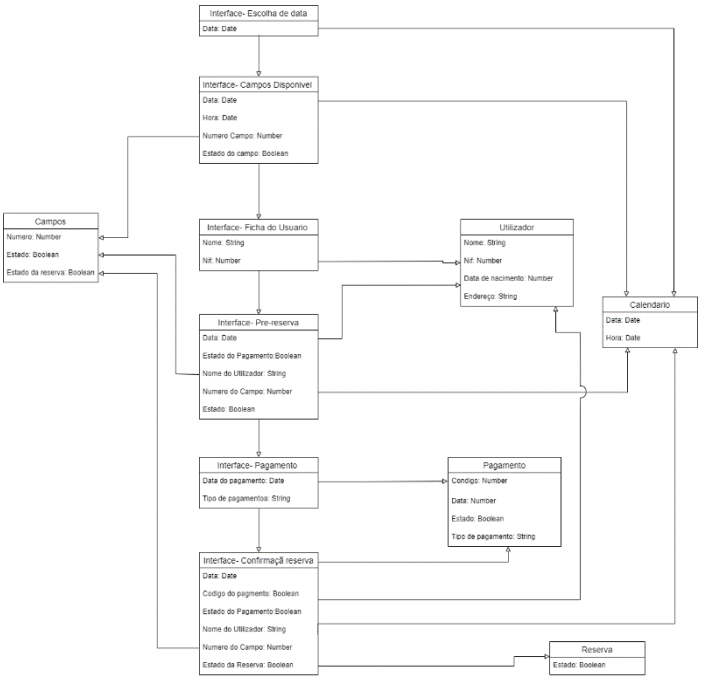


Figura 11 - diagrama de classes interface

Em um cenário prático, cada classe no diagrama de classe pode representar uma entidade de dados, e os relacionamentos entre essas classes refletem as relações na base de dados. Atributos de uma classe podem corresponder aos campos na tabela do banco de dados. Associar a classe ao componente ou módulo que interage com a base de dados ajuda a visualizar a conexão entre a lógica de negócios, representada pelas classes, e a persistência de dados na base de dados.

Portanto, enquanto o diagrama de classe destaca principalmente a estrutura de classes e suas relações, a integração com outros diagramas do UML permite uma representação mais abrangente do sistema, incluindo a identificação da base de dados e sua relação com a interface do usuário.

4.5 Fluxo de Informação

A integração cuidadosa de um fluxo de informação desempenha um papel crucial no processo de concepção de software. Em sua essência, essa prática proporciona uma representação visual minuciosa da manipulação e movimentação de dados dentro do sistema. Neste contexto, sua importância é substancial, e isso pode ser atribuído a diversas razões.

Primeiramente, ao facilitar a compreensão contextual, o fluxo de informação destaca-se como um guia eficaz para visualizar o cenário de uso do software. Isso implica na identificação dos atores envolvidos, suas interações e as funcionalidades-chave necessárias para atender às demandas do ambiente.

Além disso, a prática do fluxo de informação contribui significativamente para a elucidação precisa de requisitos. Definir claramente entradas, processos e saídas de cada componente do sistema resulta em requisitos mais claros e compreensíveis, estabelecendo uma base sólida para o desenvolvimento subsequente.

A visualização nítida do funcionamento do sistema é outra vantagem essencial proporcionada por essa prática. Esse aspecto facilita a compreensão das interações entre diferentes partes do software, promovendo uma visão abrangente do processo.

A capacidade de identificar condições e decisões no sistema é uma característica fundamental do fluxo de informação, permitindo modelar comportamentos variáveis com base em circunstâncias específicas. Isso contribui para a flexibilidade e adaptabilidade do software diante de diferentes situações.

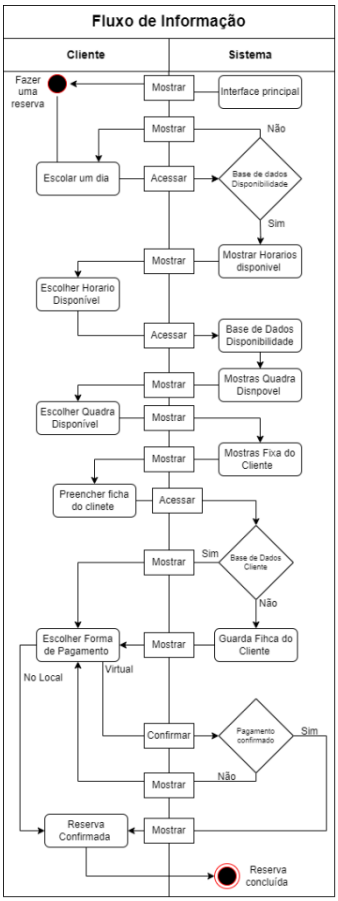


Figura 12 - Fluxo de Informação

Adicionalmente, a detecção antecipada de problemas ou lacunas nos requisitos é um benefício crítico. Isso propicia a criação de soluções mais robustas, evitando possíveis obstáculos durante o desenvolvimento e garantindo uma implementação mais eficaz.

A eficácia do fluxo de informação como uma ferramenta de comunicação entre membros da equipe, stakeholders e clientes é inegável. Sua utilidade transcende a simples representação visual, promovendo uma compreensão compartilhada e colaborativa do sistema em desenvolvimento.

Por fim, a prática do fluxo de informação não apenas orienta o desenvolvimento, mas também estabelece uma base sólida para os testes. Essa abordagem sistemática auxilia programadores na implementação eficiente da lógica do software e facilita a criação de casos de teste abrangentes para garantir a qualidade do sistema.

Assim, a incorporação cuidadosa de um fluxo de informação emerge como uma prática indispensável no processo inicial de idealização de software, fornecendo benefícios substanciais que transcendem as fases subsequentes de desenvolvimento e teste.

4.6 Diagramas de Estado

O diagrama de estados é uma ferramenta de representação que permite a leitura do comportamento dinâmico dum sistema, software ou outro tipo de processo. Este desempenha a função de apresentar as transições de estados que ocorrem em resposta a eventos externos, nos sistemas de software.

Estes diagramas proporcionam o entendimento de como o sistema, internamente, se adapta ou responde a estímulos externos, alterando os seus estados ao longo do tempo quer de forma cíclica ou linear. Assim, estes diagramas ajudam a perceber a reatividade do sistema e a mudanças de estado desencadeadas por diferentes eventos.

Para esta solução desenvolveu-se a representação gráfica das transições de estados, melhorando a compreensão articulação entre os diferentes atores e os elementos do sistema.

Por conseguinte avança-se para o desenvolvimento de quatro diagramas de estado distintos para se exemplificar as transições de estado das reservas, dos campos e da autenticação do utilizador e do administrador.

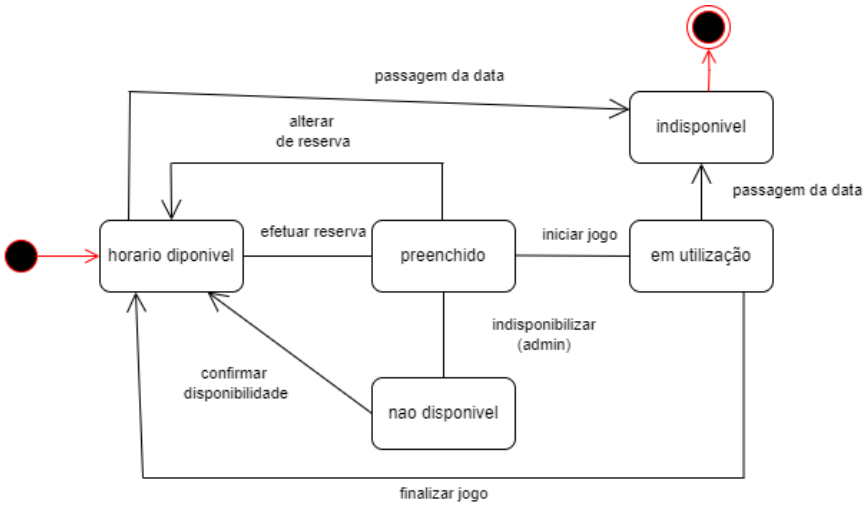


Figura 13 - Diagrama de estados das reservas

Este diagrama esclarece os estados vinculados à reserva. Inicialmente, a reserva está incompleta, ainda não foi paga. Somente após esta etapa é possível reconhecer-se como paga, ou se não for do interesse, a reserva fica em espera e passado algum tempo não é confirmada e o processo termina. No caso de desejar a confirmação, esta é paga e só então confirmada. Após esses passos, se não houver a intenção de realizar mais ações, é feito o logout.

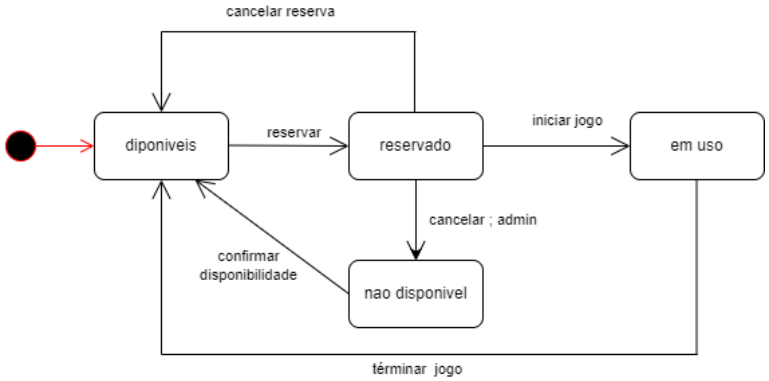


Figura 14 - Diagrama de estados dos campos

Este diagrama mostrando os estados possíveis dos campos desde reservados, disponíveis em uso ou não disponíveis. Este é muito semelhante ao das reservas, visto que ambos estão interligados, mas neste o estado de partida, sem nenhuma alteração, é o campo estar disponível.

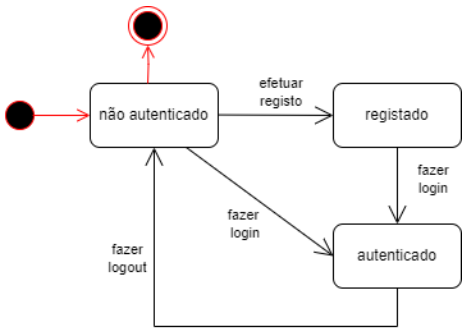


Figura 15 - Diagrama de estados do utilizador

Este diagrama mostra os diferentes estados do utilizador, quer seja jogador ou cliente ou por outro lado administrador. Estes também se iniciam por estarem não autenticados e depois têm de se registar, caso nunca o tenham feito ou então avançar para o login, caso queiram fazer processos mais complexos e que interaja com a base de dados. Esta diferenciação de estado, relaciona-se com processos de segurança, para haver uma conta registada e dados de quem interage com o sistema.

5. Requisitos não funcionais

Além dos diagramas e os requisitos neles representados, existem outros requisitos importantes que devem ser referidos que são requisitos não funcionais.

Estes requisitos não funcionais estão diretamente ligados aos requisitos funcionais e são os requisitos onde é solicitado além do conteúdo, a forma como o sistema se deve comportar, no caso específico são requisitos como a rapidez de desempenho do sistema, o facto do sistema ser interoperacional, ou seja ser capaz de comunicar com diferentes sistemas sendo portanto adaptativo, assim como ser expansível e evolutivo, ficando preparado para que no futuro se consigo incorporar no mesmo mais funções (requisitos funcionais), e ir crescendo à medida das necessidades apresentadas. O sistema deverá também ser de interface móvel, ou seja, capaz de correr em dispositivos móveis, e também "user friendly" de forma a ser de fácil aprendizagem e utilização para que a sua adoção seja um processo simples e mais rápido.

Para solucionar este desafio metodológico, estamos à procura de uma solução de software mobile, que permite configurar e implementar as funcionalidades e requisitos mencionados no problema metodológico. Nesse sentido, como um primeiro passo, ponderamos o desenvolvimento de um software para smartphones que capacite os utilizadores/jogadores a, inicialmente, explorar as condições e os serviços oferecidos pelo clube. Em seguida, o software exibirá a disponibilidade de campos com base num período ou calendário específico. Subsequentemente, possibilitará o registo do utilizador, garantindo a segurança dos dados pessoais, tanto dos utilizadores quanto daqueles fornecidos pela administração, a fim de assegurar a confiabilidade nas reservas. Após o registo, o utilizador poderá efetuar a reserva, que pode envolver uma percentagem do valor final para garantir a integridade do negócio. Por fim, o sistema permitirá que as reservas sejam consultadas e, sob certas condições, modificadas, tanto pelo utilizador quanto pela administração.

6. Objetivos

O objetivo principal deste projeto consiste no desenvolvimento de um software de sistema de informação para abordar um problema identificado no num negócio específico. O problema foi identificado num sistema de reservas feito até agora por chamada, presencial ou email, de um clube de padel chamado Bpadel, e será resolvido com recurso aos passos anteriormente enumerados.

Essa solução proporcionará uma compreensão clara da conceptualização e identificação do problema passível de resolução por meio de software, destacando os requisitos, a estratégia, métodos e processos empregados para sua solução, culminando na prototipagem de um modelo tecnológico que pode ser aplicado em futuras instâncias.

De forma mais detalhada podemos definir diferentes objetivos micro detalhando as problemáticas detectadas e de que forma o sistema se propõe resolver:

Reservas de forma manual: As reservas dos campos de padel são atualmente feitas com recurso ao telefone, email ou presencialmente ao balcão, fazendo com que este processo seja mais difícil do que deveria quer para os utilizadores como para os administradores dos campos. A utilização da aplicação desmaterializa este processo tornando o processo de reserva mais assim como o processo de gestão de reservas.

Pesquisa de disponibilidades: Mais um processo dependente de chamada, email ou presença ao balcão, ao qual acresce o problema de que o bloqueio de horários de reserva não ocorre de forma instantânea o que faz com que por lapso humano possam acontecer situações de "overbooking".

Gestão de reservas: Atualmente feita com total dependência aos recursos humanos, é um processo pouco ágil e muito vulnerável a potenciais erros. Com o auxílio da aplicação passa a existir um único local onde todas as reservas são colocadas, o que facilita a gestão e mitiga os possíveis erros humanos.

Pagamentos: Atualmente os pagamentos são feitos ao balcão, ou esporadicamente por transferência bancária ou MBWay, contudo sendo combinado após conversa com os administradores do campo. Com a ferramenta passa a ser possível integrar o sistema de pagamentos na ferramenta e passa a ficar também este processo centralizado na plataforma.

A adoção deste sistema propõe-se então atingir estes objetivos de forma que em conjunto, o processo global de reservas se torne num processo mais ágil, mais centralizado, e mais simples de gerir. Permitirá fazer concentrar a informação num só local e também com isso diminuir o risco de erros humanos que podem acontecer devido à grande dispersão de informação.

6. Prontidão da tecnologia do protótipo

Neste momento, os requisitos foram identificados e delineados e criados os diagramas que estruturam as definições e funções tanto do sistema quanto do software. As etapas subsequentes compreendem a prototipagem e teste desses requisitos utilizando softwares e outras ferramentas tecnológicas, visando tornar operacionais e funcionais as soluções desejadas, ou seja, alcançar o pleno funcionamento de um sistema de reserva.

Esse processo inclui a procura entre vários softwares lowcode desde OutSystems, Appian, Mendix e Jotform, proporcionando comparações que podem conduzir a reestruturações em alguns dos diagramas desenvolvidos anteriormente. Consequentemente a plataforma selecionada foi o Jotform devido a esta ser direcionada a softwares mobile, apesar de também funcionar com outros. Esta também foi selecionada em confronto com o nosso baixo conhecimento de programação, que assim a justificou coo escolha, devido a ser de rápida adequação e ter vários plugins para acrescentar funções e ate se adequar a outros softwares e plataformas.

Assim foi desenvolvido o protótipo o qual enquadramos no nível de prontidão TRL 4, com base em (https://en.wikipedia.org/wiki/Technology_readiness_level).

A plataforma encontra-se assim em funcionamento apesar de não conseguir corresponder a todos os requisitos, mas cumpre as funções e requisitos primários de permitir um processo de reserva e consultar informações sobre o clube.

7. Conclusão

7.2.Dificuldades

Com o propósito de ilustrar as adversidades enfrentadas pelo grupo na busca por soluções durante a execução do projeto ou trabalho, este estudo está integrado à unidade curricular do Departamento de Sistemas de Informação. Apesar de ser ministrado de maneira mais abrangente, visando proporcionar conhecimentos básicos a todos os alunos, é observável que o grupo possui uma formação de base diversificada em relação aos sistemas de informação.

Ao longo do desenvolvimento do projeto, foram identificadas dificuldades decorrentes da heterogeneidade nas formações dos membros, que podem influenciar a abordagem e compreensão do conteúdo. Este cenário demandou estratégias específicas para contornar ou resolver os desafios enfrentados. Este trabalho visa destacar essas experiências, fornecendo uma análise reflexiva sobre as soluções adotadas para superar as disparidades educacionais e maximizar a eficácia na condução do projeto.

A formação fundamental em Engenharia e Gestão Industrial proporcionou conhecimentos básicos em programação e lógica de programação, constituindo uma base técnica que, no entanto, foi desafiada a enfrentar a tarefa principal do projeto. O foco não estava estritamente na elaboração de códigos, mas sim na concepção lógica do processo de construção de um sistema. Esse desafio tornou-se particularmente desconfortável para o aluno, que, até então, não havia experienciado demandas dessa natureza.

Além disso, a complexidade do trabalho aumentou devido à necessidade de evitar a sobreposição ou confusão entre os conhecimentos adquiridos na formação básica em engenharia e gestão industrial e os conceitos apresentados na unidade curricular. Este contexto dificultou a elaboração de diagramas, uma vez que a clareza na diferenciação de conceitos tornou-se uma prioridade para assegurar a qualidade do projeto.

Outro desafio notável surgiu na fase de desenvolvimento da interface do sistema, onde a falta de experiência anterior tornou complexa a escolha de um design adequado. A compreensão de como cores e formas influenciam a experiência do usuário gerou incertezas na definição dos princípios estéticos ideais. Esta situação representou a primeira incursão do aluno na tomada de decisões desse tipo, o que acrescentou uma camada adicional de complexidade ao processo.

No caso do Mestrado de Sistemas de informação foi um caso de primeira abordagem è estruturação e desenvolvimento de sistemas de informação de uma forma prática e também de desenvolvimento de um artefacto tecnológico, refletindo-se assim na inexperiência, mas que foi de interesse explorar.

7.2 Aprendizado

O principal objetivo deste curso reside na avaliação do conhecimento adquirido pelos membros da equipe ao longo do processo, destacando os aprendizados individuais resultantes do esforço e do tempo dedicados ao estudo. Este empreendimento foi particularmente desafiador devido à natureza técnica da unidade curricular, exigindo não apenas a compreensão, mas também a aplicação prática dos conceitos ministrados.

No decorrer do projeto, a equipe reconhece que o professor buscou proporcionar uma visão mais realista da realidade enfrentada por profissionais envolvidos no desenvolvimento de projetos de software. Ao trazer para o contexto teórico a experiência prática, o grupo percebe que foi desafiado a compreender não apenas os aspectos conceituais, mas também as dificuldades cotidianas enfrentadas por aqueles que atuam diretamente nesse campo.

A avaliação geral é de que o desafio proposto foi concluído com sucesso, uma vez que o grupo conseguiu, em diversos momentos, vivenciar e superar as dificuldades diárias enfrentadas por profissionais dessa área. Isso reforça a percepção de que o curso proporcionou uma experiência mais próxima da realidade profissional, enriquecendo significativamente o entendimento teórico com uma abordagem prática e contextualizada.

Os alunos destacam a aplicabilidade dos ensinamentos adquiridos na unidade curricular não apenas no contexto acadêmico, mas também na vida prática, ressaltando a importância de atenção aos pequenos detalhes. A percepção de que pequenas nuances, apesar de terem impactos aparentemente mínimos, podem resultar em mudanças substanciais no desenvolvimento de software é ressaltada como uma lição valiosa.

A experiência profissional do aluno no chão de fábrica, onde a semântica das palavras têm impacto limitado, contrasta com a compreensão adquirida no mundo do desenvolvimento de software, onde a escolha e consistência dos termos utilizados são cruciais. A percepção de que a semântica das palavras influencia significativamente o resultado final destaca a importância de conhecer os termos específicos utilizados na área.

No que diz respeito à UML, o aluno reconhece ter tido um contato superficial, mas expressa entusiasmo em desenvolver mais competências nesta área, percebendo o UML não apenas como um diferencial curricular, mas também como uma ferramenta de gerenciamento de dados e informações.

Os alunos destacam a relevância do feedback do cliente, representado pelo professor neste contexto, no processo de desenvolvimento de software. Compreende que diferentes fases de entrega e feedback são essenciais para moldar o projeto de acordo com as expectativas e requisitos do cliente. A compreensão de que o desenvolvimento de tecnologia é fortemente vinculado à modelação e estruturação do projeto é reconhecida como fundamental, uma vez que eventuais falhas nesses pilares podem impactar significativamente a qualidade do produto final.

Concluindo este trabalho foi um importante passo para a compreensão dos passos necessários para o planeamento e conceptualização de um software de sistemas de informação, passando por todas as etapas, que nunca se tinha experimentado antes no nosso ciclo de estudos. Este permitiu-nos conhecer e aperfeiçoar uma linguagem de comunicação e representação de sistemas de software, como o UML, levantar os requisitos necessários para os mesmos e traduzir essa informação para um software e torna-lo funcional.

Bibliografia:

Aybüke Aurum · Claes Wohlin. Engineering and Managing Software Requirements. ISBN-10 3-540-25043-3

Albert Endres, Dieter Rombach, Addison Wesley, (1993) A Handbook of Software and Systems Engineering: Empirical Observations, Laws and Theories

Craig Larman 2004. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, , 3rd edition. Pearson, 2004, ISBN13: 9780131489066

Dafydd Stuttard, Marcus Pinto. 2008. The Web Application Hacker's Handbook: Discovering and Exploiting Security Flaws. SBN: 978-0-470-17077-9

D.M. Berry, E. Kamsties, and M.M. Krieger. 2003. From Contract Drafting to Software Specification: Linguistic Sources of Ambiguity

Karl E. Kurbel, Springer-Verlag, 2008, The Making of Information Systems: Software Engineering and Management in a Globalized World, , ISBN: 978-3-540-79260-4

Martin Fowler (2005) UML Essencial:Um breve guia para a linguagem-padrão de modelagem de objetos. SBN 0-321-19368-7

Mike O'Docherty. 2005. Object-Oriented Analysis and Design Understanding System Development with UML 2.0. ISBN-13 978-0-470-09240-8

Ross J. Anderson. 2008 Security Engineering.A Guide to BuildingDependable Distributed Systems Second Edition. ISBN: 978-0-470-06852-6

Scott Ambler, Mark Lines. 2012 Disciplined Agile Delivery: A Practioner's Guide to Agile Software Delivery in the Enterprise. IBM Press, 2012 , ISBN: 978-0-13-281013-5