**Escola Superior de Tecnologia e Gestão**

**Licenciatura em Engenharia Informática**

**Laboratório de programação**

**Ano Letivo 2024/2025**

**Trabalhos Laboratorial nº1**

**Elaborado em: 2024/09/16**

**Mariana Magalhães a2022147454**

**António Dinis a2021157297**

**Índice**

[List of Figures ii](#_Toc177409302)

[1 Introduction 1](#_Toc177409303)

[2 Método / Metodologia 2](#_Toc177409304)

[3 Fichas Laboratoriais 3](#_Toc177409305)

[3.1 Parte 1 3](#_Toc177409306)

[3.2 Parte 2 3](#_Toc177409307)

[3.3 Modelação de sistemas 6](#_Toc177409308)

[3.4 Utilização de ferramentas de planeamento 9](#_Toc177409309)

[4 Discussão 11](#_Toc177409310)

[5 Conclusion 12](#_Toc177409311)

[6 References 13](#_Toc177409312)

# List of Figures

[Figura 1 - diagrama uml de casos de uso 7](#_Toc177409296)

[Figura 2 - C4 Model 8](#_Toc177409297)

[Figura 3 - diagrama GRL 9](#_Toc177409298)

[Figura 4 – Trello 10](#_Toc177409299)

[Figura 5 - burndown chart 10](#_Toc177409300)

# Introduction

O objetivo principal deste trabalho é proporcionar uma experiência prática na aplicação dos conceitos fundamentais de planeamento e estruturação de aplicações informáticas. Tendo como base uma aplicação de gestão de reservas e inventário para um restaurante. Este projeto servirá como base para a aplicação prática das metodologias ágeis, técnicas de análise de requisitos, modelação de sistemas e utilização de ferramentas de planeamento.

Este trabalho está dividido em cinco partes principais, cada uma focada em aspetos cruciais do desenvolvimento de software:

1. Escolha da metodologia de desenvolvimento
2. Análise e especificação de requisitos
3. Modelação de sistemas
4. Utilização de ferramentas de planeamento

# Método / Metodologia

Neste trabalho laboratorial, foram aplicados diversos métodos e metodologias para o desenvolvimento da aplicação de gestão de reservas e inventário para um restaurante. Esta seção detalha as abordagens utilizadas em cada fase do trabalho.

Neste trabalho tivemos que:

* Escolher e justificar a metodologia ágil adequada (Agile, Scrum ou Kanban) para o desenvolvimento de uma aplicação informática, tendo em conta as características e necessidades do projeto.
* Realizar uma análise e especificação de requisitos funcionais e não funcionais, utilizando técnicas como user stories ou casos de uso, de forma a descrever com clareza as funcionalidades esperadas do sistema.
* Modelar o sistema de forma eficaz, criando diagramas de casos de uso (UML), de containers (C4 Model) e representações de objetivos estratégicos (GRL), assegurando a coerência entre os requisitos e a arquitetura proposta.
* Utilizar ferramentas de planeamento como o Jira e o Trello para organizar o monitorizar o desenvolvimento do projeto, criando um plano de trabalho eficiente e gerindo o backlog e sprints de forma estruturada.
* Refletir sobre as metodologias e ferramentas utilizadas, identificando os principais desafios enfrentados e propondo melhorias para otimizar o processo de desenvolvimento de software.

# Fichas Laboratoriais

## Parte 1

Para o cenário apresentado a utilização do scrum seria o mais adequado pois permite entregas incrementais e adaptação contínua.

O Scrum funciona bem com equipas pequenas (entre 5 e 9 pessoas), o que facilita a comunicação e a colaboração interna.

O Scrum utiliza sprints, o que permite entregas regulares de incrementos funcionais do produto. Isto é particularmente útil para este caso, pois podem começar a utilizar partes do sistema enquanto outras estão em desenvolvimento.

O Scrum permite a adaptação rápida a mudanças através de sprints curtos e revisões regulares, facilitando ajustes nos requisitos ou prioridades.

O scrum podia ser aplicado da seguinte forma:

1. Início do projeto: Criar um Product Backlog com todas as funcionalidades desejadas para o sistema de reservas e gestão de inventário.
2. Planeamento do Sprint: A cada 2-4 semanas, selecionar itens do Product Backlog para serem desenvolvidos no próximo sprint.
3. Desenvolvimento: Durante cada sprint, a equipa trabalha nas tarefas selecionadas, realizando reuniões diárias (Daily Scrum) para acompanhar o progresso e identificar obstáculos.
4. Revisão do Sprint: No final de cada sprint, apresentar as funcionalidades concluídas ao cliente (restaurante) e recolher feedback.
5. Retrospetiva do Sprint: A equipa reflete sobre o processo e identifica melhorias para o próximo sprint.
6. Repetição: O processo repete-se até que todas as funcionalidades sejam desenvolvidas e o cliente esteja satisfeito com o produto

## Parte 2

2.2.1. Requisitos funcionais

|  |  |
| --- | --- |
| RF | Descrição |
| RF01 | O sistema deve permitir aos clientes criar reservas para um determinado horário e número de pessoas. |
| RF02 | O sistema deve permitir aos funcionários visualizar e gerenciar todas as reservas existentes. |
| RF03 | O sistema deve permitir a atualização do status das mesas (livre, ocupada, reservada). |
| RF04 | O sistema deve permitir o registo de entrada e saída de itens do inventário. |
| RF05 | O sistema deve gerar alertas quando o estoque de um item estiver abaixo de um limite predefinido. |
| RF06 | O sistema deve permitir a criação e modificação do menu do restaurante. |
| RF07 | O sistema deve permitir aos funcionários cancelar ou modificar reservas existentes. |
| RF08 | O sistema deve gerar relatórios diários de ocupação das mesas e uso do inventário. |
| RF09 | O sistema deve permitir aos clientes especificar requisitos especiais para suas reservas (ex: cadeira de bebê, acessibilidade). |
| RF10 | O sistema deve permitir a integração com um sistema de pagamento para processamento de depósitos de reserva. |

2.2.2 Requisitos não funcionais

|  |  |
| --- | --- |
| RNF | Descrição |
| RNF01 | O sistema deve ser capaz de processar até 200 reservas por dia sem perda de desempenho. |
| RNF02 | O tempo de resposta para qualquer operação do sistema não deve exceder 5 segundos. |
| RNF03 | O sistema deve estar disponível 99,99% do tempo, permitindo manutenções programadas. |
| RNF04 | Todas as transações que envolvam dados pessoais dos clientes devem ser criptografadas. |
| RNF05 | A interface do utilizador deve ser responsiva, adaptando-se a dispositivos móveis e desktops. |
| RNF06 | O sistema deve fazer backup automático dos dados diariamente, mantendo os últimos 30 dias de backups. |

2.2.3 Especificação de requisitos

|  |  |
| --- | --- |
| RF01 | |
| User story | Como cliente, quero poder fazer uma reserva online para uma data e hora específicas, para garantir uma mesa no restaurante. |
| Critério de aceitação | Poder selecionar data, hora e número de pessoas  Receber uma confirmação imediata da reserva  Visualizar um resumo da reserva após a conclusão |

|  |  |
| --- | --- |
| RF02 | |
| User story | Como funcionário do restaurante, quero poder visualizar e gerenciar todas as reservas existentes, para organizar eficientemente o atendimento aos clientes. |
| Critério de aceitação | Ver uma lista de todas as reservas organizadas por data e hora  Poder filtrar reservas por status (confirmada, pendente, cancelada)  Editar detalhes de reservas existentes quando necessário |

|  |  |
| --- | --- |
| RF03 | |
| User story | Como garçom, quero poder atualizar o status das mesas em tempo real, para manter o sistema preciso e atualizado. |
| Critério de aceitação | Mudar o status da mesa (livre, ocupada, reservada) com poucos cliques  Ver um mapa visual do restaurante com o status atual de cada mesa  Atualizar o número de pessoas ocupando cada mesa |

|  |  |
| --- | --- |
| RF04 | |
| User story | Como gerente de estoque, quero registar a entrada e saída de itens do inventário, para manter um controle preciso dos suprimentos do restaurante. |
| Critério de aceitação | Adicionar novos itens ao inventário com detalhes como quantidade, data de validade e fornecedor  Registar a saída de itens do estoque quando utilizados na cozinha  Visualizar um histórico de transações de entrada e saída para cada item |

|  |  |
| --- | --- |
| RF05 | |
| User story | Como gerente, quero receber alertas quando o estoque de um item estiver abaixo de um limite predefinido, para poder fazer pedidos de reabastecimento a tempo. |
| Critério de aceitação | Configurar níveis mínimos de estoque para cada item  Receber notificações automáticas quando um item atingir o nível mínimo  Ver uma lista de todos os itens com estoque baixo em um painel de controle |

|  |  |
| --- | --- |
| RF06 | |
| User story | Como cozinheiro do restaurante, quero poder criar e modificar o menu do restaurante, para manter as ofertas atualizadas e atraentes para os clientes. |
| Critério de aceitação | Adicionar novos pratos ao menu com descrição, preço e ingredientes  Editar detalhes de pratos existentes  Organizar pratos em categorias (entradas, pratos principais, sobremesas, etc.)  Marcar pratos como indisponíveis temporariamente |

|  |  |
| --- | --- |
| RF07 | |
| User story | Como cliente, quero poder cancelar ou modificar minha reserva existente, para adaptar meus planos conforme necessário. |
| Critério de aceitação | Acessar minhas reservas existentes através de um número de confirmação ou login  Opção para cancelar a reserva com confirmação  Poder modificar data, hora ou número de pessoas da reserva  Receber uma confirmação das alterações feitas |

|  |  |
| --- | --- |
| RF08 | |
| User story | Como proprietário do restaurante, quero gerar relatórios diários de ocupação das mesas e uso do inventário, para analisar o desempenho do negócio. |
| Critério de aceitação | Gerar relatórios para períodos específicos (diário, semanal, mensal)  Visualizar gráficos de ocupação por hora e dia da semana  Ver estatísticas de uso de ingredientes e rotatividade de estoque  Exportar relatórios em formatos comuns (PDF, Excel) |

|  |  |
| --- | --- |
| RF09 | |
| User story | Como cliente com necessidades específicas, quero poder especificar requisitos especiais ao fazer minha reserva, para garantir que minhas necessidades sejam atendidas. |
| Critério de aceitação | Opção para adicionar requisitos especiais durante o processo de reserva  Lista pré-definida de requisitos comuns (cadeira de bebê, acessibilidade, alergias)  Campo de texto livre para requisitos não listados  Visualização dos requisitos especiais na confirmação da reserva |

|  |  |
| --- | --- |
| RF10 | |
| User story | Como funcionário, quero que o sistema se integre com um sistema de pagamento para processar depósitos de reserva, para garantir compromisso dos clientes e melhorar o fluxo de caixa. |
| Critério de aceitação | Opção para clientes fazerem um depósito ao realizar a reserva  Integração segura com gateway de pagamento para processar transações  Registo automático dos depósitos no sistema financeiro do restaurante  Geração de recibos para os clientes após o pagamento do depósito |

## Modelação de sistemas

Diagrama de casos de uso para a criação de reservas

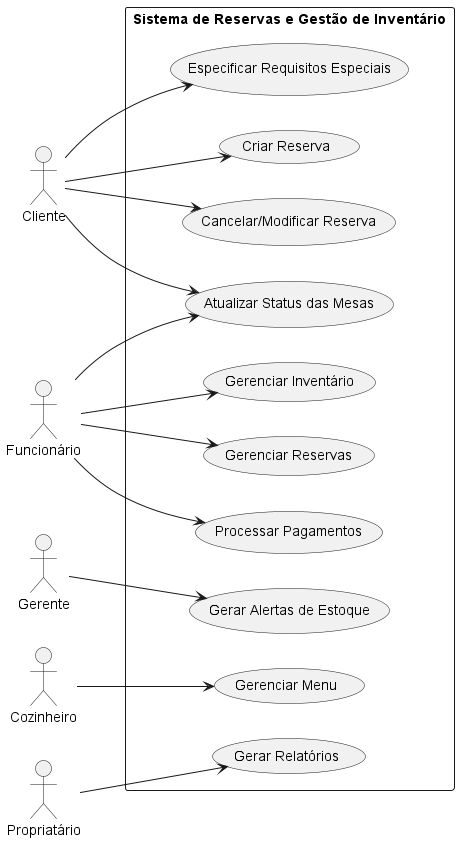


Figura - diagrama uml de casos de uso

Diagrama de containers C4 Model para descrever a arquitetura do sistema, incluindo frontend, backend e base de dados.

Uma imagem com texto, captura de ecrã, diagrama, Tipo de letra

Descrição gerada automaticamente

Figura - C4 Model

Diagrama GRL para modelar os principais objetivos não funcionais do sistema, como desempenho, segurança e escalabilidade, e identificar as decisões estratégicas que influenciam o design.

Uma imagem com texto, diagrama, desenho, file

Descrição gerada automaticamente

Figura - diagrama GRL

## Utilização de ferramentas de planeamento

Nesta parte vamos utilizar o Tello:

* Criar um quadro Kanban para o projeto, com listas como “Por Fazer”, “Em Progresso”, e “Concluído”.
* Adicionar pelo menos 10 cartões representando tarefas principais do sistema (baseadas nos requisitos identificados na Parte 2).

Uma imagem com texto, captura de ecrã, software, Software de multimédia

Descrição gerada automaticamente

Figura – Trello

E agora vamos utilizar o jira para:

* Configurar um sprint de 2 semanas para o desenvolvimento de parte do sistema.
* Adicionar tarefas ao backlog e planeá-las para o sprint.
* Gerar um burndown chart para monitorizar o progresso do sprint e discutir como o gráfico pode ser utilizado para ajustar o trabalho ao longo do ciclo.

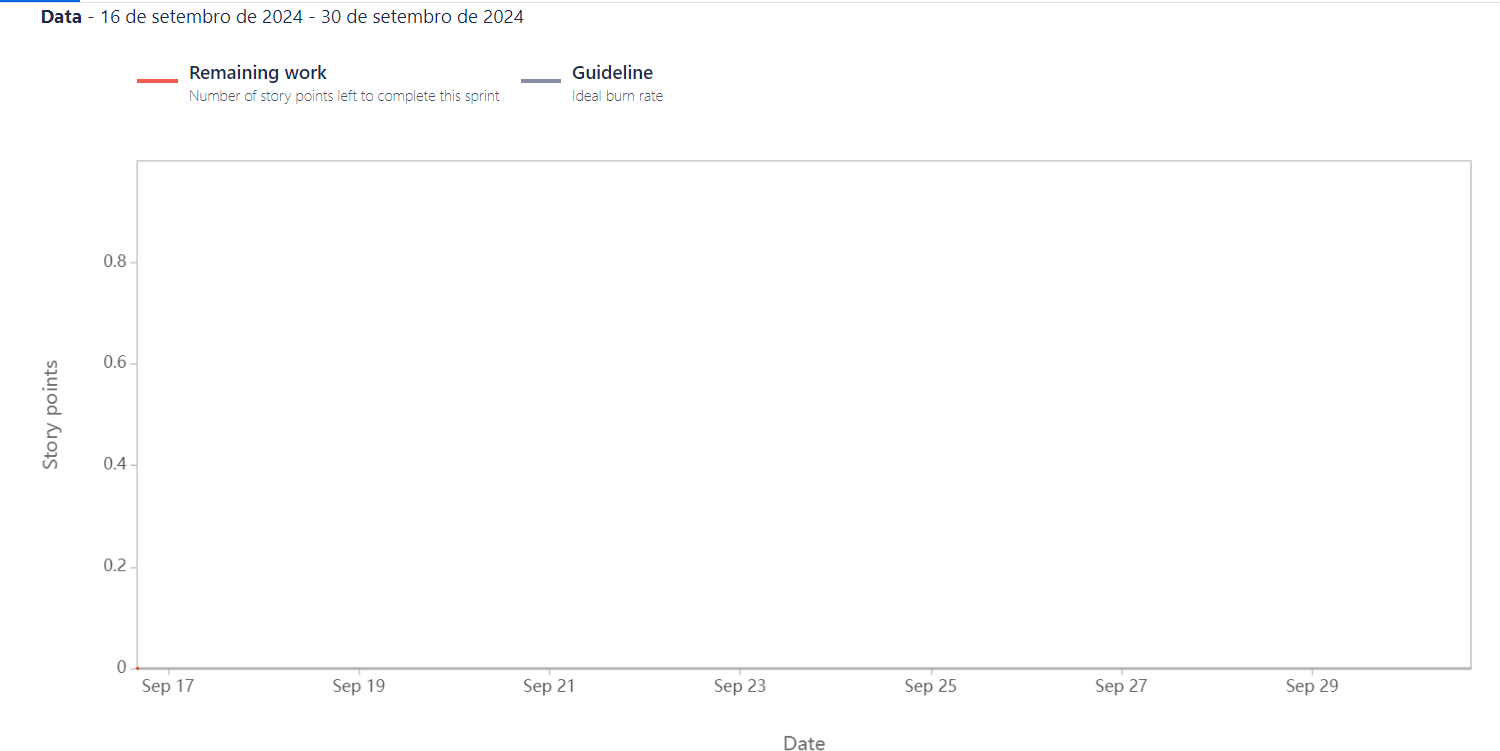


Figura - burndown chart

# Discussão

A seleção da metodologia adequada para o projeto não apresentou grandes dificuldades. O processo envolveu uma análise cuidadosa dos prós e contras de cada abordagem (Agile, Scrum, Kanban), considerando como cada uma poderia influenciar a produtividade e a qualidade do projeto. Esta análise comparativa foi crucial para garantir que a metodologia escolhida se alinhasse com as necessidades específicas do desenvolvimento da aplicação de gestão de reservas e inventário para o restaurante.

A fase de análise e especificação de requisitos apresentou desafios significativos, principalmente devido à ausência de entrevistas reais com funcionários ou gerentes de restaurantes. Esta limitação impactou a aquisição de requisitos, pois não tínhamos acesso direto às perspetivas e necessidades dos utilizadores finais. Para superar esta barreira, adotamos uma abordagem de empatia, colocando-nos nas posições tanto de clientes quanto de funcionários do restaurante. Este exercício de perspetiva permitiu-nos identificar requisitos que buscassem uma sintonia entre as necessidades de ambas as partes, embora reconheçamos que esta abordagem pode não capturar todas as nuances de um cenário real.

A construção dos diagramas UML, C4 Model e GRL foi uma fase crítica para a compreensão e comunicação do design do sistema. Esta etapa exigiu decisões importantes sobre a arquitetura do sistema, incluindo a escolha do sistema de gerenciamento de banco de dados (postgresql) e da linguagem de programação (python (serviço de notificações) e javascript (aplicação web)). Optamos por soluções que consideramos mais simplificadas, visando não complicar desnecessariamente o desenvolvimento.

O diagrama de casos de uso ajudou a visualizar as principais interações entre os utilizadores e o sistema. Identificamos casos de uso como " Criar Reserva", "Gerenciar Reservas" e " Gerenciar Inventário".

Utilizamos o diagrama de containers do C4 Model para representar a arquitetura de alto nível do sistema, o que permitiu visualizar claramente a separação entre o frontend (aplicação web), o backend (servidor de aplicação) e o base de dados. Este modelo nos ajudou a tomar decisões sobre como estruturar nossa aplicação e identificar possíveis pontos de integração.

Utilizamos o diagrama GRL para modelar os objetivos não funcionais do sistema, como desempenho, usabilidade e segurança. Através do GRL, pudemos visualizar como esses objetivos se relacionavam entre si e com os componentes do sistema, o que ajuda a priorizar certas características de design.

Por ultimo, na ultima parte da utilização de ferramentas de planeamento, o Trello foi utilizadp para o planeamento das tarefas foi relativamente direta, com o principal desafio sendo a identificação e priorização das tarefas mais importantes. Esta etapa forçou-nos a refletir criticamente sobre quais aspetos do projeto eram verdadeiramente cruciais para o seu sucesso.

O uso do Jira apresentou maiores dificuldades, uma vez que nunca tínhamos utilizado esta ferramenta, principalmente na configuração inicial e na criação do burndown chart. Aprendemos que este gráfico é uma ferramenta valiosa para monitorizar o progresso do sprint, permitindo-nos visualizar se estamos no caminho certo para concluir as tarefas dentro do prazo estipulado. Neste gráfico:

* Uma linha de progresso próxima à linha ideal indica um bom ritmo de trabalho.
* Uma linha significativamente acima da ideal sugere um progresso mais lento, possivelmente requerendo uma reavaliação da distribuição de tarefas.
* Uma linha muito abaixo da ideal pode indicar que estamos concluindo o trabalho mais rápido que o previsto, potencialmente permitindo a inclusão de mais tarefas no sprint.

No nosso caso específico, notamos que o gráfico não apresentava nenhuma linha de progresso inicialmente, o que é esperado dado que o sprint foi iniciado recentemente.

# Conclusion

Este trabalho proporcionou-nos uma valiosa experiência prática na aplicação de metodologias ágeis, técnicas de modelação e ferramentas de planeamento. Aprendemos a importância de adaptar nossas abordagens às limitações do projeto e a valor de uma boa comunicação e visualização no processo de desenvolvimento de software. As dificuldades encontradas, especialmente na especificação de requisitos sem acesso direto aos stakeholders, nos ensinaram a importância da criatividade e empatia no design de sistemas. Para projetos futuros, buscaremos incorporar mais feedback direto dos usuários finais, mesmo em cenários simulados, para enriquecer nossa compreensão dos requisitos e necessidades do sistema.

# References

Atlassian. “Jira.” *Atlassian*, 2019, www.atlassian.com/software/jira. Accessed 16 Sept. 2024.

iZenBridge Consultancy Pvt Ltd. “Burndown Charts with Jira - Managing Projects with Jira.” *YouTube*, 31 May 2022, www.youtube.com/watch?v=hxDWRD5ZxyA. Accessed 16 Sept. 2024.

“Open-Source Tool That Uses Simple Textual Descriptions to Draw Beautiful UML Diagrams.” *PlantUML.com*, plantuml.com/. Accessed 16 Sept. 2024.

Trello. “Trello.” *Trello*, 2023, trello.com/. Accessed 16 Sept. 2024.