

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

Lista de Figuras.....	ii
1 Introdução.....	1
2 Método / Metodologia	2
3 Fichas Laboratoriais	3
3.1 Parte 1.....	3
3.2 Parte 2.....	3
3.3 Modelação de sistemas	7
3.4 Utilização de ferramentas de planeamento	10
4 Discussão	11
5 Conclusão.....	12
6 Referências.....	13

Lista de Figuras

FIGURA 1 - DIAGRAMA UML DE CASOS DE USO.....	7
FIGURA 2 - C4 MODEL	8
FIGURA 3 - DIAGRAMA GRL.....	9
FIGURA 4 - TRELLO	10
FIGURA 5 - <i>BURNDOWN CHART</i>	10

1 Introdução

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 trabalho está dividido em quatro partes principais, cada uma focada em diferentes aspetos 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

2 Método / Metodologia

Neste trabalho laboratorial, foram aplicados diversos conceitos para o planeamento e estruturação de uma aplicação de gestão de reservas e inventário para um restaurante. Esta secção detalha as abordagens utilizadas em cada fase do trabalho.

A atividade foi dividida em 5 partes:

- Escolher e justificar a metodologia ágil mais adequada (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* e 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 e 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, propondo melhorias para otimizar o processo de desenvolvimento de *software*.

3 Fichas Laboratoriais

3.1 Parte 1

Para o cenário apresentado a utilização do Scrum seria mais adequada, 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 continuam em desenvolvimento.

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

O Scrum poderia 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 o *feedback*.
5. **Retrospectiva 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

3.2 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 <i>status</i> das mesas (ex: 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 bebé, 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 <i>desktops</i> .
RNF06	O sistema deve fazer <i>backup</i> automático dos dados diariamente, mantendo os últimos 30 dias de <i>backups</i> .

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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • Ver uma lista de todas as reservas organizadas por data e hora • Poder filtrar reservas por <i>status</i> (confirmada, pendente, cancelada) • Editar detalhes de reservas existentes quando necessário

RF03	
User story	Como funcionário, quero poder atualizar o <i>status</i> das mesas em tempo real, para manter o sistema preciso e atualizado.
Critério de aceitação	<ul style="list-style-type: none"> • Mudar o <i>status</i> da mesa (livre, ocupada, reservada) com poucos cliques • Ver um mapa visual do restaurante com o <i>status</i> atual de cada mesa • Atualizar o número de pessoas a ocupar cada mesa

RF04	
User story	Como gerente, quero registrar a entrada e saída de itens do inventário, para manter um controle preciso dos suprimentos do restaurante.
Critério de aceitação	<ul style="list-style-type: none"> • Adicionar novos itens ao inventário com detalhes como quantidade, data de validade e fornecedor • Registrar 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	<ul style="list-style-type: none"> • 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, quero poder criar e modificar o menu do restaurante, para manter as ofertas atualizadas e atraentes para os clientes.
Critério de aceitação	<ul style="list-style-type: none"> • 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 a minha reserva existente, para adaptar meus planos conforme o necessário.
Critério de aceitação	<ul style="list-style-type: none"> • Aceder às 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	<ul style="list-style-type: none"> • 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 a minha reserva, para garantir que as minhas necessidades sejam atendidas.
Critério de aceitação	<ul style="list-style-type: none"> • 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 o compromisso dos clientes e melhorar o fluxo da caixa.
Critério de aceitação	<ul style="list-style-type: none"> • Opção para clientes fazerem um depósito ao realizar a reserva • Integração segura com o <i>gateway</i> 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

3.3 Modelação de sistemas

Diagrama de casos de uso para o sistema de reservas e gestão de inventário:

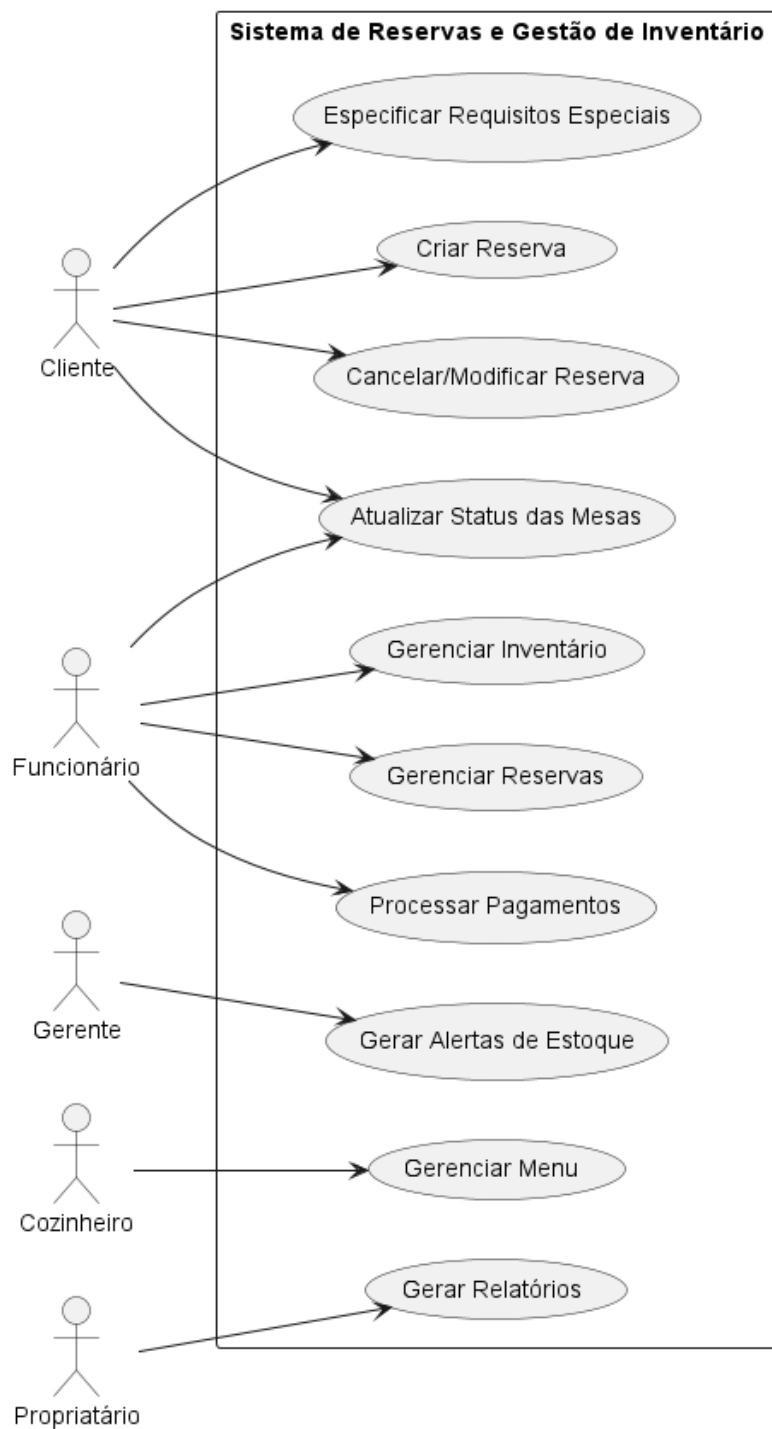


Figura 1 - Diagrama UML de casos de uso

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

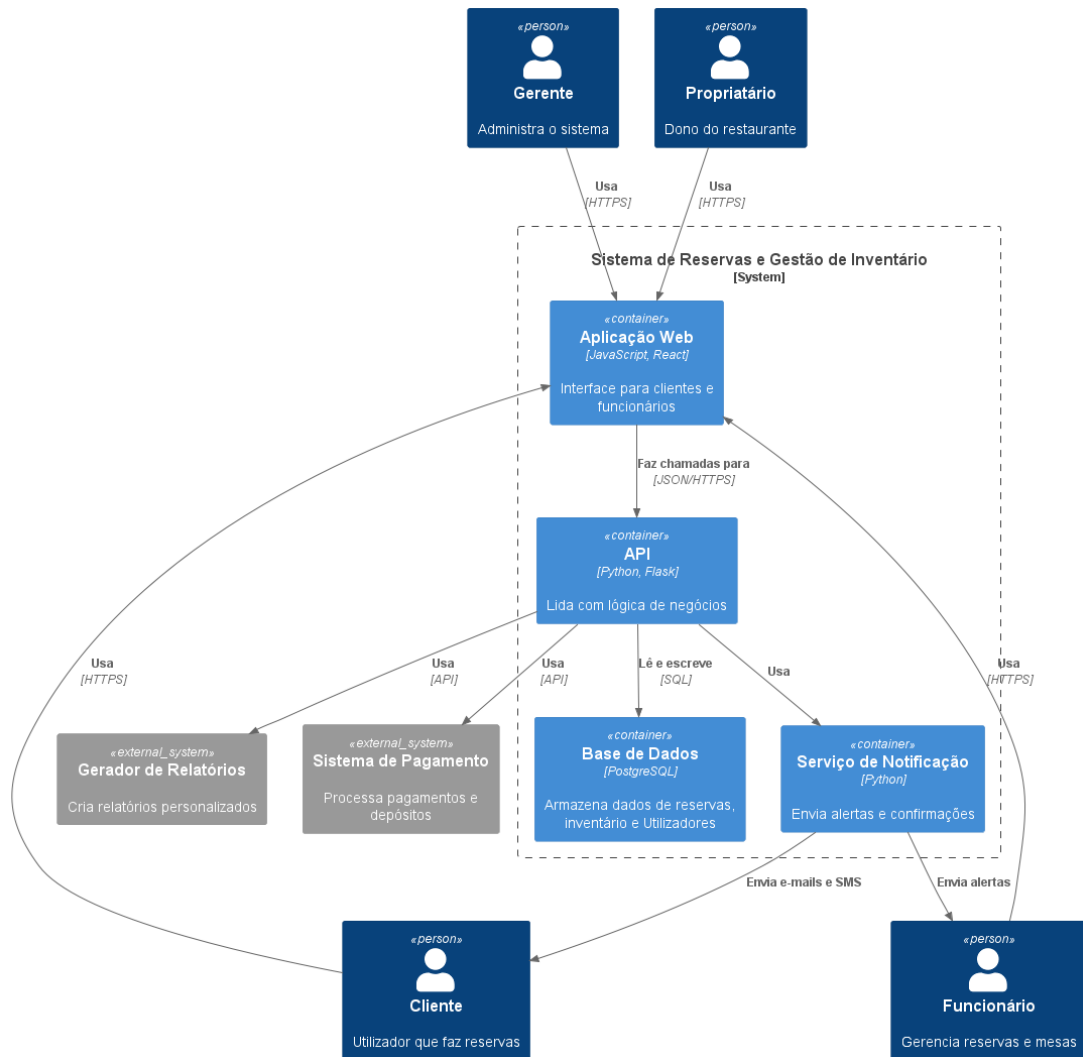


Figura 2 - C4 Model

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*:

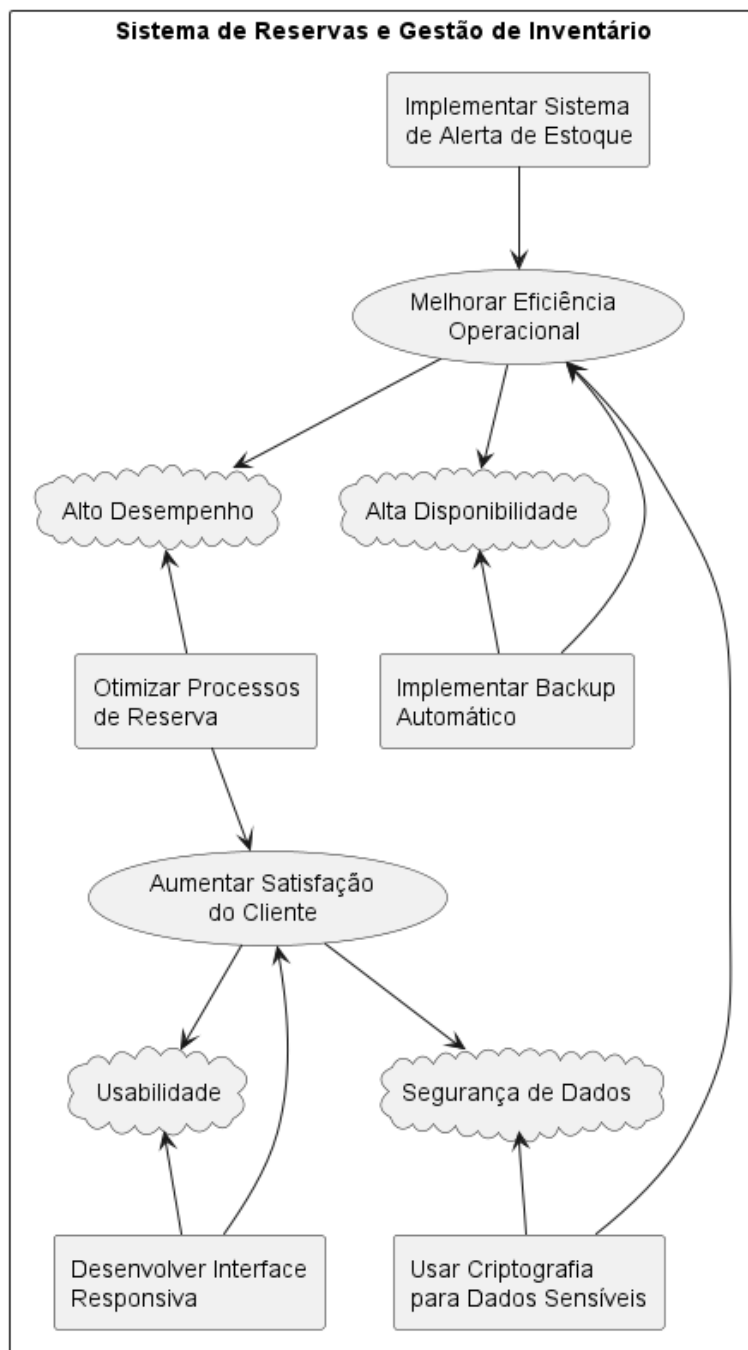


Figura 3 - Diagrama GRL

3.4 Utilização de ferramentas de planeamento

Nesta parte utilizámos o Trello para:

- 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).

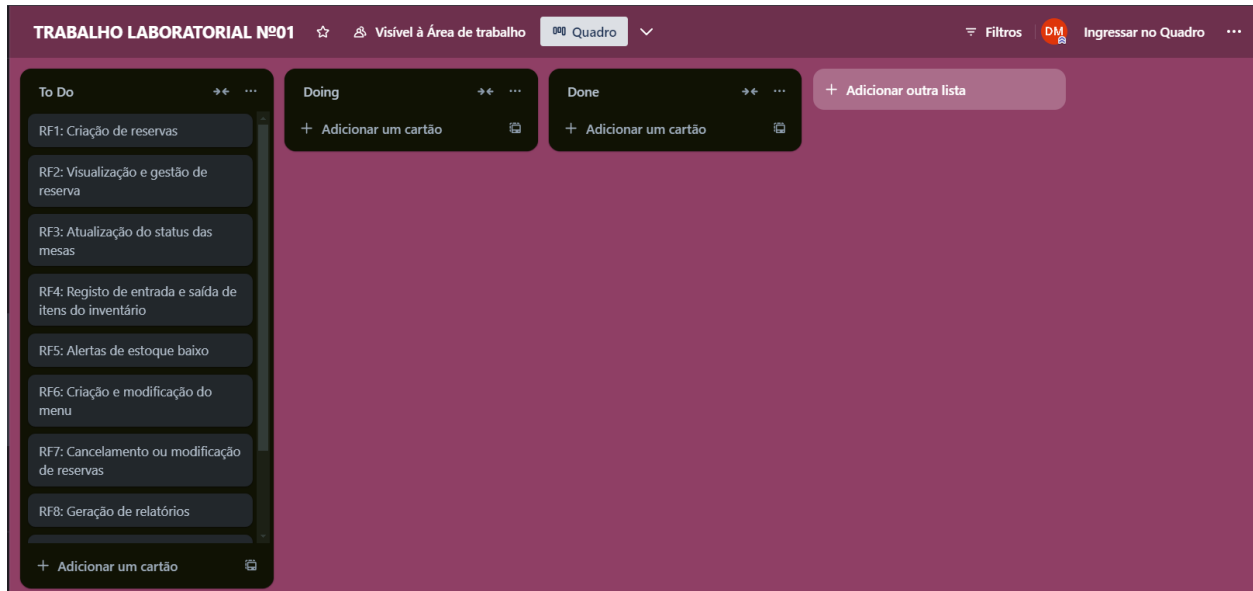


Figura 4 - Trello

Seguidamente, utilizámos 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 5 - Burndown chart

4 Discussão

A escolha da metodologia Scrum para este projeto foi relativamente simples, dadas as características do sistema de reservas e gestão de inventário para um restaurante. O Scrum oferece a flexibilidade necessária para lidar com possíveis mudanças nos requisitos e permite entregas incrementais, o que é ideal para este tipo de aplicação.

A especificação de requisitos foi desafiante, principalmente pela falta de contacto direto com potenciais utilizadores do sistema. Tivemos de nos colocar no lugar tanto dos clientes como dos funcionários do restaurante para identificar as necessidades mais prováveis. Esta abordagem, embora não ideal, permitiu-nos criar uma lista abrangente de requisitos funcionais e não funcionais.

Na fase de modelação, os diagramas UML, C4 Model e GRL ajudaram-nos a visualizar a estrutura e as interações do sistema hipotético. Este exercício permitiu-nos compreender melhor como diferentes técnicas de modelação podem ser aplicadas para representar diversos aspetos de um sistema de *software*.

A introdução ao Trello e ao Jira, ainda que superficial, deu-nos uma ideia de como estas ferramentas podem ser utilizadas para organizar e acompanhar o desenvolvimento de um projeto. O Trello mostrou-se intuitivo para a organização básica de tarefas, enquanto o Jira apresentou funcionalidades mais avançadas para a gestão de projetos ágeis.

Vale referir que a configuração inicial e a criação do *burndown chart* no Jira apresentou algumas dificuldades visto que nunca tínhamos utilizado a ferramenta. No gráfico em questão:

- 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, necessitando possivelmente de uma reavaliação da distribuição de tarefas.
- Uma linha muito abaixo da ideal pode indicar que o trabalho está a progredir mais rápido do que o previsto, permitindo potencialmente a inclusão de mais tarefas no *sprint*.

No nosso caso específico, notámos que o gráfico não apresentava nenhuma linha de progresso, o que é esperado dado que o *sprint* foi iniciado recentemente e nenhuma tarefa foi concluída.

5 Conclusão

Este exercício proporcionou-nos uma visão geral do processo de planeamento e estruturação de uma aplicação informática. Através da aplicação da metodologia Scrum, da definição de requisitos, da modelação do sistema e da exploração de ferramentas de gestão de projetos, pudemos compreender melhor como estes elementos se interligam no desenvolvimento de *software*.

Aprendemos a importância de escolher uma metodologia adequada, de definir requisitos claros e de utilizar diferentes técnicas de modelação para representar um sistema. Além disso, a introdução às ferramentas de gestão de projetos mostrou-nos como o trabalho pode ser organizado e acompanhado numa equipa de desenvolvimento.

Embora este tenha sido um exercício simulado, as competências e conhecimentos adquiridos serão valiosos para futuros projetos reais de desenvolvimento de *software*.

6 Referências

- Atlassian. “Jira.” *Atlassian*, 2019, www.atlassian.com/software/jira.
- iZenBridge Consultancy Pvt Ltd. “Burndown Charts with Jira - Managing Projects with Jira.” *YouTube*, 31 May 2022, www.youtube.com/watch?v=hxDWRD5ZxyA.
- “Open-Source Tool That Uses Simple Textual Descriptions to Draw Beautiful UML Diagrams.” *PlantUML.com*, plantuml.com/.
- Trello. “Trello.” *Trello*, 2023, trello.com/.