

Desenvolvimento de Sistemas de Software - Trabalho prático

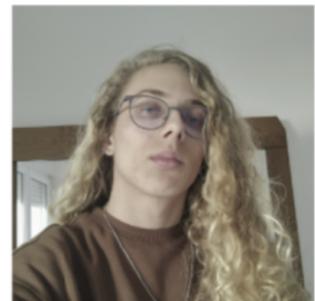
bUMger Kings



Gabriel Dantas
a107291



Simão Oliveira
a107322



José Fernandes
a106937

11.01.2026

Grupo 6: Gabriel Dantas (a107291), José Fernandes (a106937), Simão Oliveira (a107322)

Universidade do Minho - Licenciatura em Engenharia Informática, Desenvolvimento de Sistema de Software

Índice

1. Introdução	1
2. Modelação de Domínio	1
2.1. Considerações iniciais	1
2.2. Processo de construção do Modelo de Domínio	1
2.2.1. Identificação e seleção de entidades	1
2.2.2. Criação do dicionário de entidades	2
2.2.3. Identificação e seleção de associações	3
2.3. Modelo de Domínio obtido	5
3. Modelação de Use Cases	6
3.1. Análise dos cenários	6
3.2. Diagramas de Use cases	7
3.3. Use Cases	8
4. Identificação de responsabilidades da Lógica de Negócio	17
4.1. Introdução	17
4.2. Análise dos Casos de Uso	18
4.2.1. Gestão de Indicadores	18
4.2.1.1. Consultar Indicadores de Desempenho da Cadeia de Restaurantes	18
4.2.2. Gestão de Pedidos	18
4.2.2.1. Efetuar Pedido	19
4.2.3. Confeção de Pedidos	20
4.2.3.1. Reagendar Pedido	21
4.2.3.2. Marcar Tarefa como Concluída	21
4.2.4. Entrega de Pedidos	21
4.2.4.1. Entregar Pedido	22
5. Interface da Lógica de Negócio	22
5.1. Identificação dos Subsistemas	23
5.1.1. Motivação	23
5.1.2. Subsistemas Identificados	23
5.2. Diagrama de componentes da lógica de negócio	24
6. Diagrama de classes dos subsistemas da lógica de negócio	24
6.1. Diagrama de classes do subsistema de Pedidos	25
6.2. Diagrama de classes do subsistema de Tarefas	26

6.3. Diagrama Classes do subsistema de Restaurantes	26
7. Implementação das DAOs	27
7.1. Criação das tabelas	27
7.2. Diagrama de componentes e de classes atualizados	30
8. Diagramas de Sequência	32
8.1. Adicionar delay a uma tarefa	32
8.2. Adicionar proposta a um pedido	33
8.3. Atualizar o estado de um pedido	33
8.4. Calcular o tempo estimado de um pedido	34
8.5. Decompor um pedido em tarefas	34
8.6. Ver a faturação de um restaurante	35
8.7. Ver a faturação de todos os restaurantes	36
8.8. Finalizar um pedido	36
8.9. Obter a lista das propostas disponíveis	37
8.10. Iniciar um novo pedido	37
8.11. Listar pedidos com um determinado estado	37
8.12. Listar pedidos prontos	38
8.13. Listar as personalizações possíveis de uma proposta.	39
8.14. Listar todos os restaurantes	39
8.15. Listar todas as tarefas de um posto	40
8.16. Marcar uma tarefa como concluída	40
8.17. Recalcular ETA de tarefas	41
8.18. Calcular o tempo médio de atendimento de um restaurante	41
8.19. Calcular o tempo médio de atendimento da cadeia de restaurantes	42
9. Diagrama de Packages	42
10. Implementação e Interface	44
10.1. Menu Principal	44
10.2. Menu do Cliente	44
10.3. Menu do funcionário	46
10.4. Menu de Entregador	49
10.5. Trocar de restaurante	50
10.6. Menu do COO	50
11. Observações Finais	51
12. Enunciado do trabalho prático	52

1. Introdução

O presente documento descreve o processo de desenvolvimento da primeira fase do trabalho prático da unidade curricular de Desenvolvimento de Sistemas de Software. Nesta etapa foi realizada a modelação do domínio do problema, com o objetivo de compreender e representar os principais conceitos, entidades e relações que compõem o sistema a ser desenvolvido, o bUMger Kings.

Além disso, procedeu-se à identificação e especificação dos casos de uso, definindo as interações entre os atores e o sistema, bem como as funcionalidades que este deverá oferecer. Esta fase é essencial para estabelecer uma visão clara e partilhada do funcionamento do sistema, servindo de base para o desenho e implementação das fases seguintes do projeto.

2. Modelação de Domínio

2.1. Considerações iniciais

Para a elaboração do modelo de domínio da aplicação requisitada, foi realizada uma leitura e análise detalhada do enunciado fornecido pela equipa docente. Com base nesta análise, foram identificadas as entidades e relações do modelo de domínio. Devido à relativa ambiguidade de alguns aspectos, foi necessário formular determinadas assunções e efetuar pequenas adaptações de modo a construir um modelo que, no nosso entender, melhor representa a realidade do problema proposto.

Efetuou-se a leitura do capítulo 12 do livro *Object-Oriented Modeling and Design with UML* [1], que apresenta uma metodologia para o processo de desenvolvimento de modelos de domínios de classes. Apesar de não seguir tal método à risca, usaram-se algumas noções e práticas adquiridas.

Ao longo desta seção será discutido o processo de construção do modelo de domínio, incluindo a identificação das entidades e associações e eliminação de ambiguidades. De ressaltar que o processo iterativo ao longo desta seção foi um fator decisivo, pelo que se irá apresentar uma versão simplificada da evolução do modelo e seleção das entidades e relações, de modo a evitar fadiga na leitura, mas ilustrando os principais desafios passados.

2.2. Processo de construção do Modelo de Domínio

2.2.1. Identificação e seleção de entidades

Começou-se por extrair diretamente do enunciado do trabalho as entidades candidatas – conceitos identificados no domínio do problema que poderão vir a ser modelados como entidades, caso se revelem relevantes para representar informação essencial. Listam-se, de forma extensiva, abaixo.

- | | | |
|----------------|---------------|---------------|
| • Restaurante | • Empregado | • Nota |
| • Pedido | • Função | • Menu |
| • Cliente | • Alimento | • Ingrediente |
| • Stock | • Estágio | • Elemento |
| • Ecrãs táteis | • Posto | • Bebida |
| • Prato | • Funcionário | • Alergénio |
| • Confecção | • Display | |
| • Refeição | • Proposta | |

Durante este processo começou-se a escrever um dicionário de entidades, que falaremos na sub-seção seguinte, porém é importante ressaltar neste momento que foi uma ação fundamental para eliminar entidades candidatas que representavam informação repetida e irrelevante. Durante a escrita deste dicionário foi possível perceber que as entidades:

- Funcionário e Empregado são redundantes, ou seja, representam os mesmos conceitos do sistema;
- Pedido, Prato e Refeição são redundantes;
- Função e confecção aproximam-se mais de ações do que representam informação relevante;
- Estágio e Posto são redundantes;
- Ingrediente, Alimento, Elemento e Bebida tratam de graus de abstração diferentes, porém que se cruzam comumente.

Dado estas considerações, eliminaram-se as entidades candidatas que apresentavam redundâncias e que apresentam irrelevância.

Pensou-se em formas diferentes de organizar as entidades candidatas Ingrediente, Alimento, Elemento e Bebida de modo a representar mais fielmente e conter um mais adequado nível de abstração dos conceitos desejados. Com tal intuito manteve-se a entidade Ingrediente e criou-se a entidade Produto que se situa num nível de abstração superior a Alimento, Elemento e Bebida, contendo-os a todos estes.

Da descrição dos cenários de uso foi ainda possível retirar algumas novas entidades que apresentam semântica importante no domínio do problema. Tais entidades são:

- Gerente e COO (Chief Operating Officer);
- Tempo de Espera ;
- Pagamento, Método de Pagamento, Multibanco, Numerário e MB Way. Estas últimas três relacionar-se-ão pela associação “é um” com Método de Pagamento;
- Estado, Em Preparação, Pronto e Entregue. Estas três últimas relacionar-se-ão pela associação “é um” com Estado.

Assim, estão estabelecidas todas as entidades pertinentes no contexto do problema.

2.2.2. Criação do dicionário de entidades

A criação do dicionário de entidades foi um processo iterativo e evolutivo. O intuito era percecionar com uma maior clareza falhas de persistência no modelo. Foi, através deste que conseguimos identificar e resolver tautologias entre entidades, detetar entidades desprezáveis no contexto do domínio e definir entidades com um mais adequado grau de abstração.

Abaixo, na Tabela 1, apresenta-se o resultado final do desenvolvimento deste dicionário de entidades, que padroniza os termos usados no contexto do problema.

Tabela 1: Dicionário das entidades do sistema.

Entidade	Descrição
Restaurante	Representa o estabelecimento onde o sistema é utilizado. Pode conter múltiplos postos de trabalho, tal como ecrãs táteis associados.
Ecrã Tátil	Dispositivo através do qual o cliente interage com o sistema, realizando pedidos.

Pedido	Conjunto de produtos solicitados por um cliente. Está associado a um estado (ex.: “em preparação”, “pronto”, “entregue”) e tem um pagamento associado.
Cliente	Pessoa que realiza pedidos no restaurante. Faz pagamentos. Pode ter preferências associadas (ex.: alergénios a evitar).
Proposta	Unidade elementar de um pedido. Pode ser um produto ou um menu.
Produto	Unidade individual de consumo incluída num pedido (ex.: alimento, bebida, sobremesa). Utiliza ingredientes e compõe um menu. É uma proposta.
Menu	Conjuntos predefinidos de produtos disponíveis no restaurante. É uma proposta do pedido.
Ingrediente	Elemento básico utilizado na preparação dos produtos (ex.: ovo, alface, tomate). Cada ingrediente tem uma quantidade em stock. Possui ou não componentes alergénios.
Stock	Quantidade de um tipo de ingrediente disponível no restaurante. Permite controlar quantidades, entradas e saídas, garantindo que os produtos do menu podem ser preparados.
Alergénio	Substância presente em determinados ingredientes que pode causar reações adversas. Serve para informar o cliente e filtrar opções durante o pedido.
Tarefa	Representa uma ação ou conjunto de ações necessárias para preparar um determinado produto de um pedido (ex.: confeccionar ingredientes ou ir buscar bebida). Uma tarefa realiza-se em um posto.
Posto	Local físico onde se realizam tarefas, utilizando ingredientes. Pode estar associado a um funcionário e a um ecrã tátil.
Funcionário	Trabalhador do restaurante responsável por realizar tarefas e entregar pedidos.
Display	Dispositivo através do qual o funcionário interage com o sistema, consultando e manipulando pedidos.
Nota	Registo feito por um cliente associado a um pedido.
Tempo de Espera	Período estimado entre a submissão do pedido e a sua conclusão. Pode ser utilizado para gestão de filas e previsão de serviço.
Gerente	Trabalhador com permissões administrativas, responsável pela gestão global do restaurante, incluindo stock e supervisionar funcionários.
COO (Chief Operating Officer)	Responsável pela supervisão operacional de múltiplos restaurantes ou unidades.
Pagamento	Transação associada a um pedido. Contém informação sobre o valor pago, o método de pagamento utilizado e o estado da transação.
Método de Pagamento	Forma utilizada para liquidar um pedido (ex.: numerário, cartão, MB Way). Cada pagamento está associado a um método.
Estado do Pedido	Condição atual de um pedido (ex.: “em preparação”, “pronto”, “entregue”).

2.2.3. Identificação e seleção de associações

De seguida analisou-se o enunciado, como primeiro passo para obter as associações – relações ou ligações entre duas entidades que precisam de interagir ou conhecer-se mutuamente no sistema. Dado que já existe um dicionário bem definido de entidades, estas associações foram transpostas e levemente modificadas para se justificarem no contexto. Usar-se-á uma notação de “Entidade1 associação Entidade2 (cardinalidade)” para representar as associações neste relatório.

As associações que puderam ser retiradas diretamente do enunciado são:

- Cliente interage com Ecrã Tátil ($N : 1$)
- Cliente efetua Pedido ($1 : N$)
- Pedido é entregue a Cliente ($N : 1$)
- Pedido inclui Nota ($1 : 0..1$)
- Funcionário consulta Display ($1 : 1$)

O facto da associação entre Pedido e Nota ter cardinalidade $1 : 0..1$, representa que a nota tem apenas o estado de existência ou inexistência. Portanto, assumimos que todas as anotações feitas por um cliente são salvadas num mesmo local.

Algumas associações que consideramos triviais de inferir dado a estruturação do modelo de domínio feita e o contexto do problema são:

- Pedido é composto por Proposta ($N : N$)
- Produto é uma Proposta
- Menu é uma Proposta
- Menu contém Produto ($0..N : N$)
- Produto utiliza Ingrediente ($N : N$)
- Ingrediente possui Alergénio ($N : 0..1$)
- Pedido é exibido em Display ($0..N : N$)

Novamente, a decisão da cardinalidade $N : 0..1$ na associação entre Ingrediente e Alergénio deriva de estarmos apenas a denotar que o ingrediente tem ou não substâncias alergénias.

Com tais associações é possível pensar num fluxo de decisão para o pedido. Um cliente pode tanto selecionar produtos individuais como menus para o seu pedido através da proposta. O menu é apenas um agrupamento de produtos. O produto é, por sua vez, um agrupamento de ingredientes.

Das descrições do enunciado que se seguem do lado esquerdo, conseguimos inferir outras adaptadas à nossa estruturação do lado direito:

- | | |
|---|--|
| <ul style="list-style-type: none"> • Confecção passa por estágios • Alimento é Confeccionado • Funcionário prepara Produto | <ul style="list-style-type: none"> • Produto requer Tarefa ($N : N$) • Tarefa realiza-se em Posto ($1 : 1$) • Funcionário está alocado a Posto ($1 : 1$) • Posto utiliza Ingrediente ($N : N$) |
|---|--|

A confecção passa assim a ser pensada como uma linha de produção, em que existem tarefas para cada produto, que se realizam em determinados postos, onde estão alocados funcionários para confeccionar os ingredientes.

Algumas associações que considerámos desprezáveis no contexto do domínio, apesar de virem a ser úteis no futuro, quando se implementar o programa, são:

- Cliente retira produtos (que possam conter alergénios)
- Cliente acrescenta/retira ingredientes (ou cliente personaliza pedido)
- Pedido é empratado
- Pedido é embalado

Ambas as duas primeiras são facilmente pensáveis como um filtro a implementar na prática, não uma associação bem estabelecida. Ambas as duas últimas são insignificantes no contexto da modelação do problema, resumindo-se a Pedido é entregue a Cliente.

Dos cenários disponibilizados conseguimos definir as seguintes associações, que se enumeram exaustivamente abaixo:

- Cliente faz Pagamento ($1 : N$)
- Pedido inclui Pagamento ($1 : 1$)
- Pagamento utiliza Método de Pagamento ($N : 1$)
- Pedido possui Tempo de Espera ($1 : 1$)
- Tempo de Espera é exibido em Display ($0..N : N$)
- Stock possui Ingrediente ($1 : 0..N$)
- Pedido possui Estado do Pedido ($N : 1$)
- Restaurante possui Ecrã Tátil ($1 : N$)
- COO chefia Restaurante ($1 : N$)
- COO supervisiona Gerente ($1 : N$)
- Gerente supervisiona Funcionário ($1 : N$)

Uma cardinalidade pertinente de justificar é a da associação Stock possui Ingrediente. Apesar de o stock físico ter todos os ingredientes juntamente, faz sentido modelar o problema com stocks individualizados para cada ingrediente. Assim é mais simples interpretar casos de falta de algum ingrediente no posto, sendo necessário ir ao stock daquele ingrediente específico para o repôr no posto.

Como referido ainda na seção da definição e seleção das entidades, surgiram associações do tipo “é um”, que dispensam cardinalidades e demonstram uma relação hierárquica. Retirando as já referidas Menu é uma Proposta e Produto é um Proposta, usaram-se:

- Multibanco é um Método de Pagamento
- Numerário é um Método de Pagamento
- MB Way é um Método de Pagamento
- Em Preparação é um Estado
- Pronto é um Estado
- Entregue é um Estado

Algumas associações um pouco mais periféricas e talvez com menor relevância, mas que nos fizeram sentido incorporarem o modelo, são:

- Posto contém Display ($1 : 1$)
- Nota apresentada em Display ($0..N : N$)
- Gerente repõe Stock ($N : N$)

Assim, estão definidas todas as associações no nosso modelo de domínio.

2.3. Modelo de Domínio obtido

Dado este desenvolvimento, chegámos a uma versão do modelo de domínio que consideramos apropriada. Apresenta-se na Figura I.

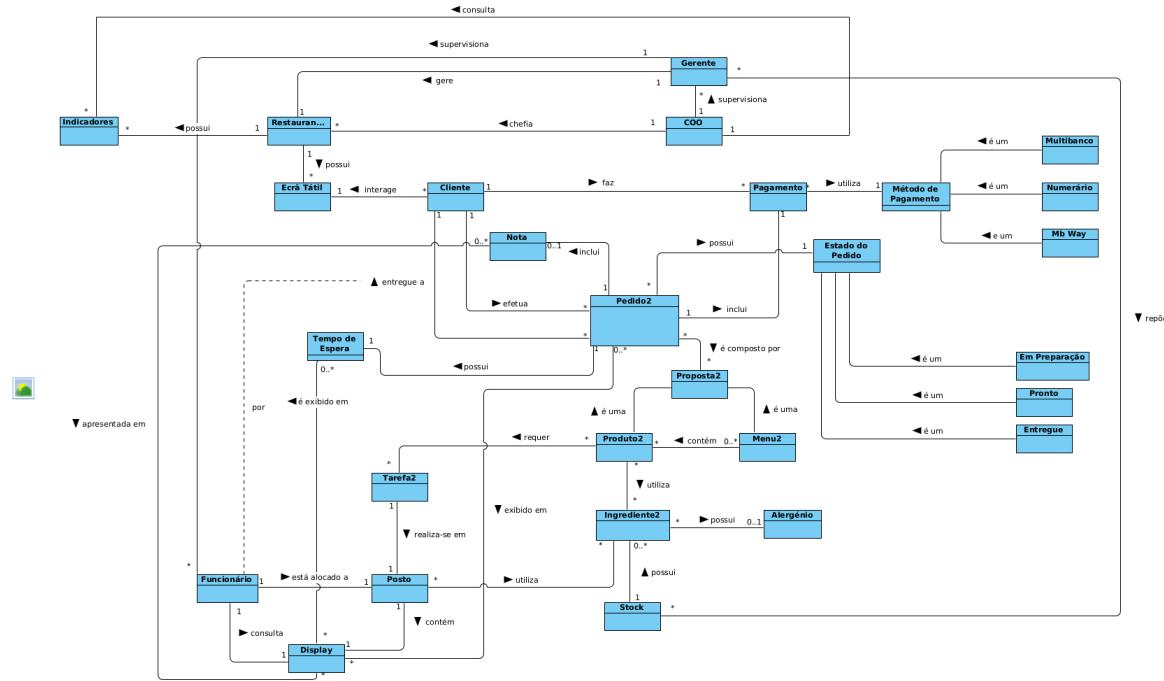


Figura 1: Modelo de Domínio obtido.

3. Modelação de Use Cases

3.1. Análise dos cenários

Da análise dos cenários disponibilizados, foi possível observar ações efetuadas de forma frequente por certos atores do sistema. Conseguimos concluir que terão que existir os seguintes use cases:

- Efetuar Pedido pelo Cliente, dos cenários 1 e 2;
- Realizar Pagamento pelo Cliente, dos cenários 1 e 2;
- Realizar Entrega pelo Funcionário, dos cenários 1 e 2;
- Reagendar Pedido pelo Funcionário, do cenário 3;
- Pedir Ingrediente do Armazém pelo Funcionário, do cenário 3;
- Repor Stock do Armazém pelo Gerente, por inferência do cenário 3 e da nossa modelação de domínio;
- Consultar indicadores de desempenho da cadeia de restaurantes pelo COO, do cenário 4;
- Consultar indicadores de desempenho de restaurante pelo Gerente, do cenário 4;
- Enviar mensagem para funcionários de qualquer restaurante pelo COO, do cenário 4;
- Enviar Mensagem para Funcionários do restaurante pelo Gerente, por generalização de funções, do cenário 4;
- Iniciar Sessão pelo Funcionário, COO e Gerente, do cenário 4 e por generalização de casos;
- Terminar Sessão pelo Funcionário COO e Gerente, por inferência através do cenário 4;
- Marcar Tarefa como concluída pelo Funcionário, pela nossa modelação de domínio.
- Associar funcionário a um posto, pelo COO e Gerente, pela nossa modelação de domínio.

De salientar, tal como na seção anterior, ao longo de toda esta o processo foi iterativo progressivo. Foram feitos aprimoramentos a cada passo de revisão, análise e proposta de alterações.

3.2. Diagramas de Use cases

Com base nos use cases diagnosticados na seção anterior, construiram-se 4 modelos diferentes, para melhor clarificar as vidas sobre o sistema. Na Figura 2 encontram-se os use cases relativos ao processo de efetuar pedidos e as consequências que isso envolve. De notar que o use case Efetuar Pedido depende de Realizar Pagamento, pelo que o inclui. Utilizou-se também a especialização para definir diferentes use cases com base em formas diferentes de Realizar Pagamento pelo Cliente, com Multibanco, com MB Way ou com Numerário.

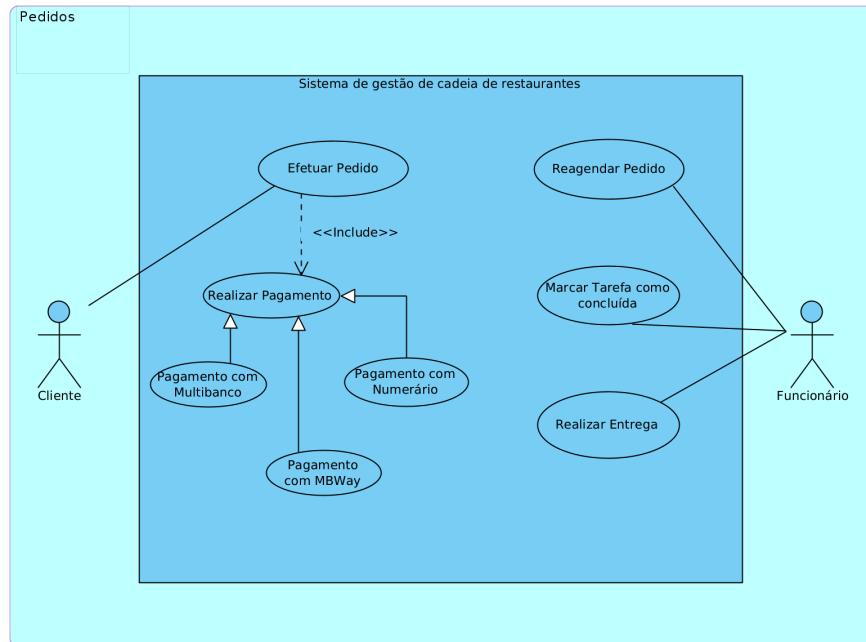


Figura 2: Diagrama de use cases para operações sobre pedidos e a sua manutenção.

Já na Figura 3, estão disponíveis os use cases relativos à gestão de stock.

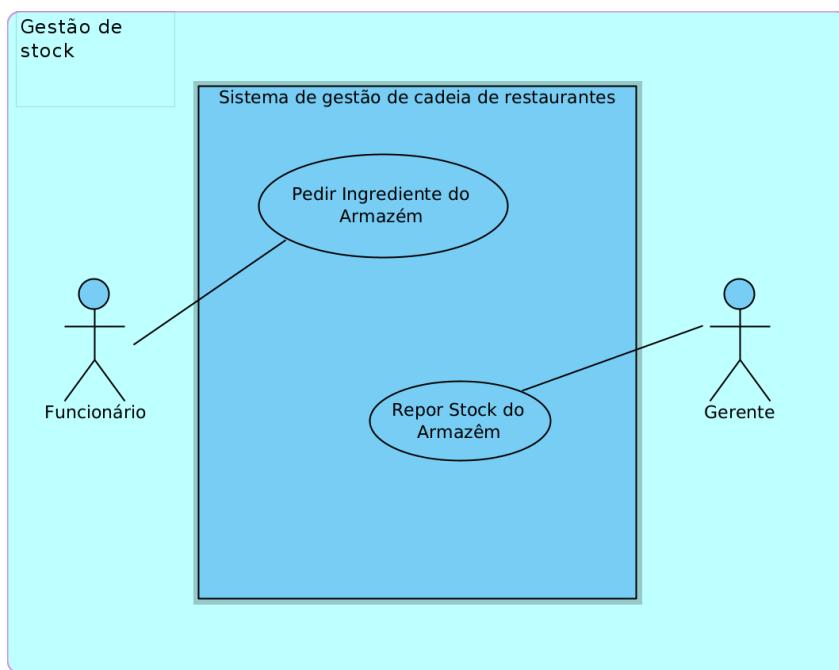


Figura 3: Diagrama de use cases para operações sobre gestão de stocks.

Na Figura 4 apresentam-se os diferentes use cases relacionados com a gestão do restaurante pelo Gerente e pelo COO.

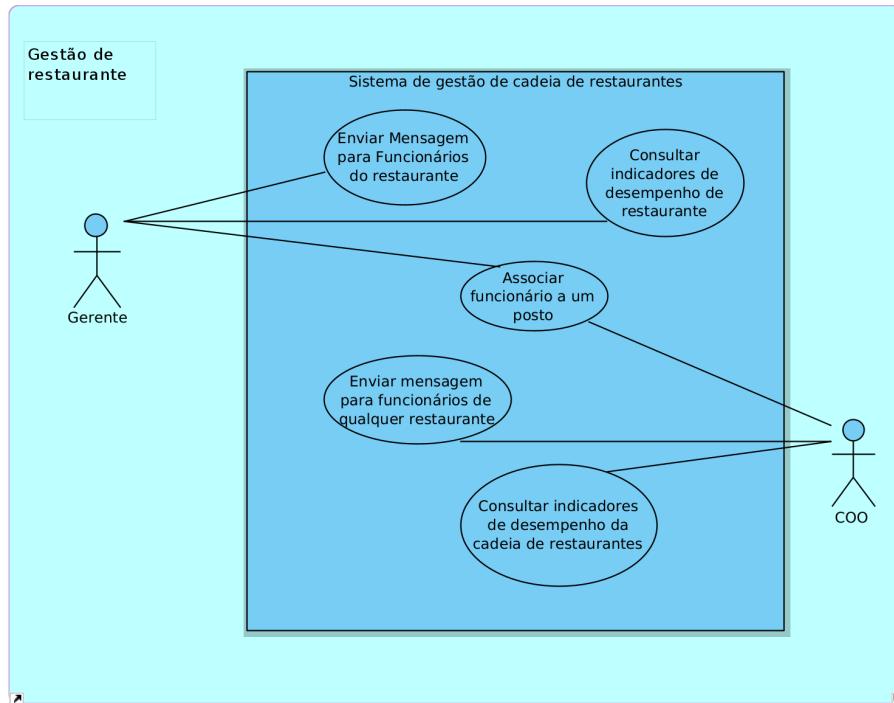


Figura 4: Diagrama de use cases para operações sobre gestão de restaurantes.

Por fim, a Figura 5 a contém os use cases relativos à autenticação no sistema pelos diferentes atores internos ao sistema.

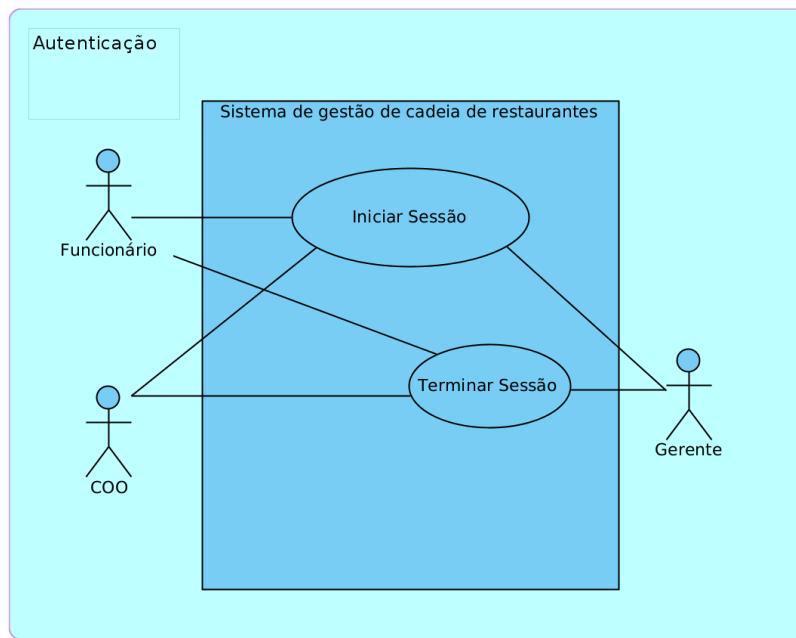


Figura 5: Diagrama de use cases para operações sobre autenticação.

3.3. Use Cases

Tendo todos os use cases bem identificados, passemos a fazer a especificação de todos eles. A parte que se segue exige uma leitura exaustiva.

Tabela 2: Use Case Efetuar Pedido.

Caso de uso	Efetuar pedido
Descrição	O ator seleciona os produtos que pretende e o sistema envia o pedido para produção.
Atores	Cliente
Pré-condição	
Pós-condição	<ul style="list-style-type: none"> • O pedido é enviado para a cozinha e começa a ser preparado. • O estado do pedido é atualizado para “Em preparação”. • O cliente tem o talão com o número do pedido e o tempo estimado de preparação.
Fluxo normal	<ol style="list-style-type: none"> 1. O Ator indica que pretende iniciar o pedido. 2. O sistema inicia o pedido. 3. O sistema procura todas as propostas disponíveis. 4. O sistema apresenta as propostas disponíveis. 5. O ator adiciona as propostas desejadas ao carrinho. 6. O ator indica que pretende finalizar o pedido. 7. O sistema pergunta se o ator pretende adicionar uma nota para a cozinha. 8. O ator responde que não. 9. O sistema calcula o valor total das propostas selecionadas. 10. O sistema apresenta o resumo e o valor total. 11. O ator confirma. 12. O ator realiza o pagamento com sucesso. 13. O sistema calcula o tempo estimado de preparação. 14. O sistema pergunta se a refeição é para consumir no restaurante. 15. O ator responde que sim. 16. O sistema cria e regista o pedido. 17. O sistema imprime o talão com o número do pedido e o tempo estimado de preparação. 18. O sistema envia o pedido para a cozinha (decompõe em tarefas e atribui aos postos).
Fluxo Alternativo 1	<p>[Personalizar propostas] (passo 5)</p> <ol style="list-style-type: none"> 5.1. O ator indica que quer personalizar uma proposta no carrinho. 5.2. O sistema apresenta as opções de personalização disponíveis para a proposta selecionada. 5.3. O ator define as personalizações pretendidas. 5.4. O sistema atualiza a proposta no carrinho. 5.5. O fluxo retoma no passo 5 do fluxo normal.
Fluxo Alternativo 2	<p>[Adicionar nota para a cozinha] (passo 8)</p> <ol style="list-style-type: none"> 8.1. O ator responde que sim. 8.2. O sistema apresenta uma caixa de texto para o ator introduzir a nota. 8.3. O ator escreve e confirma a nota. 8.4. O sistema guarda a nota temporariamente. 8.5. O fluxo retoma no passo 9 do fluxo normal.
Fluxo Alternativo 3	<p>[Pedido para take-away] (passo 14)</p> <ol style="list-style-type: none"> 14.1. O ator responde que não (take-away). 14.2. O sistema cria e regista o pedido. 14.2. O sistema imprime o talão com o número do pedido, o tempo estimado e o balcão de levantamento. 14.3. O fluxo retoma no passo 18 do fluxo normal.

Fluxo de Exceção 1	[Falha no pagamento] (passo 12) 12.1. O sistema avisa que houve uma falha no pagamento. 12.2. O sistema cancela o pedido.
---------------------------	---

Tabela 3: Use Case Reagendar Pedido.

Caso de uso	Reagendar pedido
Descrição	O ator altera o tempo previsto de execução de um pedido quando não o pode confeccionar de imediato, informando o sistema sobre o novo tempo estimado. O sistema atualiza os tempos de execução nos restantes postos e notifica o cliente.
Atores	Funcionário
Pré-condição	<ul style="list-style-type: none"> • O ator está autenticado no sistema.
Pós-condição	<ul style="list-style-type: none"> • O pedido é reagendado com novo tempo de execução. • Os restantes postos afetados são atualizados. • O cliente é informado da alteração do tempo estimado.
Fluxo normal	<ol style="list-style-type: none"> 1. O ator solicita que um pedido seja reagendado. 2. O sistema solicita o novo tempo previsto. 3. O ator introduz o novo tempo. 4. O sistema atualiza o tempo estimado do pedido e recalcula automaticamente os tempos dos pedidos seguintes afetados. 5. O sistema propaga a atualização para os restantes postos dependentes. 6. O sistema notifica o cliente, através do display do restaurante, da alteração no tempo de confeção.
Fluxo de Exceção 1	<p>[Tempo inválido] (passo 4)</p> <ol style="list-style-type: none"> 4.1. O sistema notifica o ator que o tempo introduzido não é válido. 4.2. O sistema cancela a operação.

Tabela 4: Use Case Marcar Tarefa como Concluída.

Caso de uso	Marcar tarefa como concluída
Descrição	O ator marca uma tarefa como concluída quando termina a sua execução no posto correspondente.
Atores	Funcionário
Pré-condição	<ul style="list-style-type: none"> • O ator está autenticado no sistema. • O ator tem uma tarefa atribuída.
Pós-condição	<ul style="list-style-type: none"> • Os postos seguintes ou o funcionário de entrega de que a tarefa foi concluída são notificados da conclusão da tarefa.
Fluxo normal	<ol style="list-style-type: none"> 1. O ator solicita aceder à lista de tarefas atribuídas. 2. O sistema apresenta as tarefas atribuídas ao funcionário. 3. O ator seleciona a tarefa que acabou de realizar. 4. O ator indica no sistema que a tarefa foi concluída. 5. O sistema regista a conclusão da tarefa e atualiza o seu estado. 6. O sistema verifica se todas as tarefas do pedido estão concluídas e, se sim, atualiza o estado do pedido para “Pronto”.

Tabela 5: Use Case Realizar Entrega.

Caso de uso	Realizar entrega
Descrição	O ator entrega o pedido completo ao cliente, seja para consumo no restaurante ou para take-away.
Atores	Funcionário
Pré-condição	<ul style="list-style-type: none"> • O ator está autenticado no sistema. • O pedido encontra-se marcado como pronto para entrega.
Pós-condição	<ul style="list-style-type: none"> • O estado do pedido é atualizado para “entregue”.
Fluxo normal	<ol style="list-style-type: none"> 1. O ator solicita aceder à lista de pedidos prontos para entrega. 2. O sistema apresenta os pedidos prontos para entrega (estado = “Pronto”). 3. O ator seleciona o pedido que irá entregar. 4. O sistema apresenta os detalhes do pedido (número, produtos, tipo de entrega). 5. O ator confirma a entrega ao cliente. 6. O sistema atualiza o estado do pedido para “Entregue” e regista o horário da entrega.

Tabela 6: Use Case Pedir Ingrediente do Armazém.

Caso de uso	Pedir ingrediente do armazém
Descrição	O ator solicita através do seu display o envio de um ingrediente em falta para o seu posto.
Atores	Funcionário
Pré-condição	<ul style="list-style-type: none"> • O ator está autenticado no sistema.
Pós-condição	<ul style="list-style-type: none"> • O armazém é notificado. • O tempo estimado de chegada do ingrediente requisitado pelo ator é-lhe apresentado.
Fluxo normal	<ol style="list-style-type: none"> 1. O ator solicita que um ingrediente lhe seja entregue no seu posto. 2. O sistema pergunta qual dos ingredientes tem que ser trazido do armazém. 3. O ator seleciona o ingrediente que pretende solicitar. 4. O sistema valida a disponibilidade em stock. 5. O ator confirma o pedido de envio. 6. O sistema apresenta o tempo estimado de entrega do ingrediente ao posto. 7. O sistema regista o pedido e notifica o armazém.
Fluxo de Exceção 1	<p>[Ingrediente não existe em stock] (passo 4)</p> <ol style="list-style-type: none"> 4.1. O sistema notifica o ator que o ingrediente não está disponível em stock. 4.2. O sistema regista a necessidade de encomendar o ingrediente do fornecedor. 4.3. O sistema cancela o pedido.

Tabela 7: Use Case Repor Stock do Armažem.

Caso de uso	Rapor stock do armazém
Descrição	O ator regista a entrada de novos ingredientes no sistema, atualizando as quantidades disponíveis em stock.
Atores	Gerente
Pré-condição	<ul style="list-style-type: none"> • O ator está autenticado no sistema.
Pós-condição	<ul style="list-style-type: none"> • As quantidades de stock dos ingredientes são atualizadas. • Os pratos que estavam indisponíveis por falta de stock ficam disponíveis.
Fluxo normal	<ol style="list-style-type: none"> 1. O ator solicita que seja atualizado um ingrediente que pretende repor. 2. O sistema apresenta a lista de ingredientes com as quantidades atuais de stock. 3. O ator seleciona o ingrediente a repor. 4. O sistema pergunta qual é a quantidade reposta. 5. O ator introduz a quantidade a adicionar. 6. O sistema valida a quantidade introduzida. 7. O sistema atualiza a quantidade em stock do ingrediente. 8. O sistema atualiza a disponibilidade dos pratos afetados pela falta do ingrediente.
Fluxo de Exceção 1	<p>[Quantidade inválida] (passo 6)</p> <ol style="list-style-type: none"> 6.1. O sistema deteta que a quantidade é negativa, zero ou num formato inválido. 6.2. O sistema apresenta mensagem de erro e cancela a operação.

Tabela 8: Use Case Associar Funcionário a um Posto.

Caso de uso	Associar funcionário a um posto
Descrição	O ator associa um funcionário a um posto de trabalho específico dentro de um restaurante, definindo as suas responsabilidades.
Atores	Gerente, COO
Pré-condição	<ul style="list-style-type: none"> • O ator está autenticado no sistema. • O ator possui as permissões necessárias para consultar os indicadores do restaurante selecionado.
Pós-condição	<ul style="list-style-type: none"> • O funcionário fica associado ao posto selecionado.
Fluxo normal	<ol style="list-style-type: none"> 1. O ator acede à área de gestão de funcionários. 2. O sistema apresenta a lista de funcionários e postos disponíveis. 3. O ator seleciona um funcionário e um posto de trabalho. 4. O ator confirma a associação. 5. O sistema valida a associação. 6. O sistema regista a associação entre o funcionário e o posto. 7. O sistema confirma a operação com uma mensagem de sucesso.
Fluxo Alternativo 1	<p>[Posto já tem funcionário associado] (passo 5)</p> <ol style="list-style-type: none"> 5.1. O sistema verifica que o posto já tem um funcionário associado. 5.2. O sistema pergunta se o ator pretende desassociar o funcionário atual. 5.3. O ator responde que sim. 5.4. O sistema desassocia o funcionário atual. 5.5. O fluxo retoma no passo 6 do fluxo normal.
Fluxo de Excessão 1	<p>[Atror não pretende desassociar funcionário atual] (passo 5.3)</p> <ol style="list-style-type: none"> 5.3.1. O ator responde que não. 5.3.2. O sistema cancela a operação.

Tabela 9: Use Case Enviar Mensagem para Funcionários do Restaurante.

Caso de uso	Enviar mensagem para funcionários do restaurante
Descrição	O ator envia uma mensagem para os displays de todos os funcionários do restaurante onde trabalha.
Atores	Gerente
Pré-condição	<ul style="list-style-type: none"> • O ator está autenticado no sistema. • O ator possui as permissões necessárias para consultar os indicadores do restaurante selecionado.
Pós-condição	<ul style="list-style-type: none"> • A mensagem é exibida nos displays dos funcionários do restaurante.
Fluxo normal	<ol style="list-style-type: none"> 1. O ator solicita enviar uma mensagem para todos os funcionários do restaurante onde trabalha. 2. O ator escreve o conteúdo da mensagem. 3. O ator confirma o envio. 4. O sistema envia a mensagem para todos os funcionários. 5. O sistema regista o envio e exibe confirmação ao ator.

Tabela 10: Use Case Enviar Mensagem para Funcionários de qualquer Restaurante.

Caso de uso	Enviar mensagem para funcionários de qualquer restaurante
Descrição	O ator envia uma mensagem para os displays de todos os funcionários de um restaurante específico da cadeia, escolhido pelo próprio.
Atores	COO
Pré-condição	<ul style="list-style-type: none"> • O ator está autenticado no sistema. • O ator possui as permissões necessárias para consultar os indicadores do restaurante selecionado.
Pós-condição	<ul style="list-style-type: none"> • A mensagem é exibida nos displays dos funcionários do restaurante selecionado.
Fluxo normal	<ol style="list-style-type: none"> 1. O ator solicita enviar uma mensagem para todos os funcionários de um restaurante. 2. O sistema apresenta a lista de restaurantes da cadeia. 3. O ator seleciona o restaurante para o qual quer enviar a mensagem. 4. O ator escreve o conteúdo da mensagem. 5. O ator confirma o envio. 6. O sistema envia a mensagem para todos os funcionários do restaurante selecionado. 7. O sistema regista o envio e exibe confirmação ao ator.

Tabela 11: Use Case Consultar Indicadores de Desempenho de Restaurante

Caso de uso	Consultar indicadores de desempenho de restaurante
Descrição	O ator consulta indicadores relativos à faturação e ao desempenho de atendimento do restaurante onde trabalha.
Atores	Gerente
Pré-condição	<ul style="list-style-type: none"> • O ator está autenticado no sistema. • O ator possui as permissões necessárias para consultar os indicadores do restaurante selecionado.
Pós-condição	<ul style="list-style-type: none"> • Os indicadores solicitados pelo ator são-lhe apresentados.
Fluxo normal	<ol style="list-style-type: none"> 1. O ator solicita a consulta dos indicadores de um restaurante. 2. O sistema apresenta a lista de indicadores disponíveis. 3. O ator seleciona o indicador que pretende visualizar. 4. O sistema processa o pedido e apresenta os dados correspondentes.
Fluxo de Exceção I	<p>[Indicadores indisponíveis] (passo 4)</p> <ol style="list-style-type: none"> 4.1. O sistema deteta a ausência de dados solicitados. 4.2. O sistema informa o ator de que não existem dados disponíveis.

Tabela 12: Use Case Consultar Indicadores de Desempenho da Cadeia de Restaurantes.

Caso de uso	Consultar indicadores de desempenho da cadeia de restaurantes
Descrição	O ator consulta indicadores agregados relativos à faturação e ao desempenho de atendimento de toda a cadeia de restaurantes.
Atores	COO
Pré-condição	<ul style="list-style-type: none"> • O ator está autenticado no sistema. • O ator possui as permissões necessárias para consultar os indicadores da cadeia de restaurantes.
Pós-condição	<ul style="list-style-type: none"> • Os indicadores solicitados pelo ator são-lhe apresentados.
Fluxo normal	<ol style="list-style-type: none"> 1. O ator solicita a consulta dos indicadores da cadeia de restaurantes. 2. O sistema pergunta se o ator pretende visualizar um indicador de toda a cadeia. 3. O ator responde que sim. 3. O sistema apresenta a lista de indicadores disponíveis. 4. O ator seleciona o indicador que pretende visualizar. 6. O sistema processa o pedido e calcula o indicador requisitado agregado da cadeia. 7. O sistema apresenta os dados correspondentes.
Fluxo Alternativo I	<p>[Ator pretende consultar indicadores de um restaurante específico] (passo 3)</p> <ol style="list-style-type: none"> 3.1. O ator responde que não. 3.2. O sistema apresenta a lista de restaurantes disponíveis. 3.3. O ator seleciona o restaurante pretendido. 3.4. O fluxo retoma no passo 4 do fluxo normal (calculando indicadores para o restaurante selecionado).
Fluxo de Exceção I	<p>[Indicadores indisponíveis] (passo 4)</p> <ol style="list-style-type: none"> 4.1. O sistema deteta a ausência de dados agregados. 4.2. O sistema informa o ator de que não existem dados disponíveis.

, caption: [Use Case: Consultar Indicadores de Desempenho da Cadeia])

Tabela 13: Use Case Autenticar Ator no Sistema.

Caso de uso	Autenticar ator no sistema
Descrição	O ator autentica-se no sistema.
Atores	Funcionário, COO, Gerente
Pré-condição	<ul style="list-style-type: none"> • Nenhum ator autenticado.
Pós-condição	<ul style="list-style-type: none"> • Ator fica com sessão iniciada no sistema e obtém acesso às funcionalidades permitidas segundo o seu perfil.
Fluxo normal	<ol style="list-style-type: none"> 1. Ator indica o seu endereço de email e palavra-passe. 2. O sistema valida o endereço de email e a palavra-passe 3. O sistema inicia uma sessão para o ator 4. Sistema informa que a autenticação teve sucesso.
Fluxo de Exceção I	<p>[Credenciais inválidas] (passo 2)</p> <ol style="list-style-type: none"> 2.1. O sistema informa que a autenticação não teve sucesso.

Tabela 14: Use Case Terminar Sessão no Sistema.

Caso de uso	Terminar sessão no sistema
Descrição	O ator fecha a sessão previamente iniciada.
Atores	Funcionário, COO, Gerente
Pré-condição	• O ator tem sessão iniciada.
Pós-condição	• A sessão iniciada pelo ator é terminada.
Fluxo normal	<ol style="list-style-type: none"> 1. O ator solicita o término da sua sessão. 2. O sistema fecha a sessão previamente iniciada. 3. O sistema informa o ator de que a sessão foi fechada com sucesso.

Temos, portanto, todos os casos de uso bem identificados e especificados.

4. Identificação de responsabilidades da Lógica de Negócio

4.1. Introdução

Esta seção corresponde ao início da segunda fase do projeto e dá continuidade ao trabalho desenvolvido anteriormente, nomeadamente à identificação dos casos de uso e à modelação do domínio do sistema. Na fase inicial, foi definida uma visão global e abrangente do sistema, contemplando um conjunto alargado de funcionalidades e entidades, com o objetivo de compreender o problema e estruturar corretamente o domínio da aplicação.

Nesta fase, o foco incide especificamente na análise detalhada dos casos de uso indicados pelo docente para implementação: Efetuar Pedido, Entregar Pedido e Consultar Indicadores. Estes casos de uso são aprofundados com o objetivo de identificar as transações e responsabilidades da lógica de negócios (camada de domínio) necessárias para suportar o funcionamento do sistema.

É importante salientar que, apesar da modelação inicial contemplar mais funcionalidades, apenas um subconjunto é analisado em detalhe nesta etapa. Os restantes casos de uso e conceitos previamente identificados mantêm-se válidos do ponto de vista conceptual, mas encontram-se fora do âmbito de implementação definido para esta fase do projeto.

Relativamente ao caso de uso Entregar Pedido, torna-se necessário considerar todo o processo de confeção do pedido. Por esse motivo, são também analisados os casos de uso Marcar Tarefa como Concluída e Reagendar Pedido, uma vez que estes são essenciais para compreender o fluxo completo desde a criação do pedido até à sua entrega ao cliente.

Para cada caso de uso, são identificados os passos que envolvem processamento de dados, cálculos, validações ou alterações de estado — ações que correspondem a operações do sistema e que deverão ser implementadas na lógica de negócios. A análise realizada servirá de base à definição dos métodos da interface da camada de lógica de negócios.

A metodologia seguida nesta seção consiste em:

- Apresentar o caso de uso com os respetivos fluxos;
- Destacar os passos que requerem operações do sistema;
- Extrair as responsabilidades identificadas para a lógica de negócios.

4.2. Análise dos Casos de Uso

Prossiga-se com a análise de cada caso de uso, identificando as responsabilidades do sistema.

4.2.1. Gestão de Indicadores

Segue-se o processo para os casos de uso relevantes na responsabilidade de gestão de indicadores.

4.2.1.1. Consultar Indicadores de Desempenho da Cadeia de Restaurantes

Caso de uso	Consultar indicadores de desempenho da cadeia de restaurantes
Fluxo normal	<ol style="list-style-type: none"> 1. O ator solicita a consulta dos indicadores da cadeia de restaurantes. 2. O sistema pergunta se o ator pretende visualizar um indicador de toda a cadeia. 3. O ator responde que sim. 3. O sistema apresenta a lista de indicadores disponíveis. 4. O ator seleciona o indicador que pretende visualizar. 6. O sistema processa o pedido e calcula o indicador requisitado agregado da cadeia. 7. O sistema apresenta os dados correspondentes.
Fluxo Alternativo 1	<p>[Aтор pretende consultar indicadores de um restaurante específico] (passo 3)</p> <ol style="list-style-type: none"> 3.1. O ator responde que não. 3.2. O sistema apresenta a lista de restaurantes disponíveis. 3.3. O ator seleciona o restaurante pretendido. 3.4. O fluxo retoma no passo 4 do fluxo normal (calculando indicadores para o restaurante selecionado).
Fluxo de Exceção 1	<p>[Indicadores indisponíveis] (passo 4)</p> <ol style="list-style-type: none"> 4.1. O sistema deteta a ausência de dados agregados. 4.2. O sistema informa o ator de que não existem dados disponíveis.

Tabela 15: Use Case: Consultar Indicadores de Desempenho da Cadeia.

Responsabilidades identificadas:

- 3.2 -> Listar todos os restaurantes da cadeia;
- 4 -> Calcular indicador agregado para toda a cadeia;
- 4 -> Calcular indicadores específicos para um restaurante individual.

4.2.2. Gestão de Pedidos

Segue-se o processo para os casos de uso relevantes na responsabilidade de gestão de pedidos.

4.2.2.1. Efetuar Pedido

Caso de uso	Efetuar pedido
Fluxo normal	<p>1. O Ator indica que pretende iniciar o pedido.</p> <p>2. O sistema inicia o pedido.</p> <p>3. O sistema procura todas as propostas disponíveis.</p> <p>4. O sistema apresenta as propostas disponíveis.</p> <p>5. O ator adiciona as propostas desejadas ao carrinho.</p> <p>6. O ator indica que pretende finalizar o pedido.</p> <p>7. O sistema pergunta se o ator pretende adicionar uma nota para a cozinha.</p> <p>8. O ator responde que não.</p> <p>9. O sistema calcula o valor total das propostas selecionadas.</p> <p>10. O sistema apresenta o resumo e o valor total.</p> <p>11. O ator confirma.</p> <p>12. O ator realiza o pagamento com sucesso.</p> <p>13. O sistema calcula o tempo estimado de preparação.</p> <p>14. O sistema pergunta se a refeição é para consumir no restaurante.</p> <p>15. O ator responde que sim.</p> <p>16. O sistema cria e regista o pedido.</p> <p>17. O sistema imprime o talão com o número do pedido e o tempo estimado de preparação.</p> <p>18. O sistema envia o pedido para a cozinha (decompõe em tarefas e atribui aos postos).</p>
Fluxo Alternativo I	<p>[Personalizar propostas] (passo 5)</p> <p>5.1. O ator indica que quer personalizar uma proposta no carrinho.</p> <p>5.2. O sistema apresenta as opções de personalização disponíveis para a proposta selecionada.</p> <p>5.3. O ator define as personalizações pretendidas.</p> <p>5.4. O sistema atualiza a proposta no carrinho.</p> <p>5.5. O fluxo retoma no passo 5 do fluxo normal.</p>

Fluxo Alternativo 2	[Adicionar nota para a cozinha] (passo 8) 8.1. O ator responde que sim. 8.2. O sistema apresenta uma caixa de texto para o ator introduzir a nota. 8.3. O ator escreve e confirma a nota. 8.4. O sistema guarda a nota temporariamente. 8.5. O fluxo retoma no passo 9 do fluxo normal.
Fluxo Alternativo 3	[Pedido para take-away] (passo 14) 14.1. O ator responde que não (take-away). 14.2 O sistema cria e regista o pedido. 14.2. O sistema imprime o talão com o número do pedido, o tempo estimado e o balcão de levantamento. 14.3. O fluxo retoma no passo 18 do fluxo normal.
Fluxo de Exceção 1	[Falha no pagamento] (passo 12) 12.1. O sistema avisa que houve uma falha no pagamento. 12.2. O sistema cancela o pedido.

Tabela 17: Use Case: Efetuar pedido.

Responsabilidades identificadas:

- 2 -> Iniciar o pedido;
- 3 -> Procurar e listar todas as propostas disponíveis (produtos e menus);
- 5.2 -> Procurar opções de personalização para uma proposta específica;
- 5.4 -> Adicionar uma proposta personalizada ao pedido;
- 9 -> Calcular valor total baseado nas propostas selecionadas;
- 13 -> Calcular tempo estimado de preparação do pedido;
- 14.2 -> Criar e registrar pedido no sistema (após pagamento bem-sucedido);
- 18 -> Decompor pedido em tarefas e atribuí-las aos postos apropriados.

4.2.3. Confeção de Pedidos

Segue-se o processo para os casos de uso relevantes na responsabilidade de confeccionar pedidos.

4.2.3.1. Reagendar Pedido

Caso de uso	Reagendar pedido
Fluxo normal	<ol style="list-style-type: none"> 1. O ator solicita que um pedido seja reagendado. 2. O sistema solicita o novo tempo previsto. 3. O ator introduz o novo tempo. 4. O sistema atualiza o tempo estimado do pedido e recalcula automaticamente os tempos dos pedidos subsequentes afetados. 5. O sistema propaga a atualização para os restantes postos dependentes. 6. O sistema notifica o cliente, através do display do restaurante, da alteração no tempo de confeção.
Fluxo de Exceção I	<p>[Tempo inválido] (passo 4)</p> <ol style="list-style-type: none"> 4.1. O sistema notifica o ator que o tempo introduzido não é válido. 4.2. O sistema cancela a operação.

Tabela 18: Use Case: Reagendar Pedido.

Responsabilidades identificadas:

- 5 -> Atualizar tempo estimado de um pedido;
- 5 -> Recalcular tempos de pedidos subsequentes afetados pela alteração;
- 6 -> Propagar atualizações de tempo para todos os postos envolvidos no pedido.

4.2.3.2. Marcar Tarefa como Concluída

Caso de uso	Marcar tarefa como concluída
Fluxo normal	<ol style="list-style-type: none"> 1. O ator solicita aceder à lista de tarefas atribuídas. 2. O sistema apresenta as tarefas atribuídas ao funcionário. 3. O ator seleciona a tarefa que acabou de realizar. 4. O ator indica no sistema que a tarefa foi concluída. 5. O sistema regista a conclusão da tarefa e atualiza o seu estado. 6. O sistema verifica se todas as tarefas do pedido estão concluídas e, se sim, atualiza o estado do pedido para “Pronto”.

Tabela 19: Use Case: Marcar Tarefa como Concluída.

Responsabilidades identificadas:

- 2 -> Procurar e listar tarefas atribuídas a um funcionário específico;
- 5 -> Registar conclusão de uma tarefa;
- 6 -> Atualizar estado da tarefa para “concluída”;
- 6 -> Verificar estado geral do pedido (se todas as tarefas estão concluídas);
- 6 -> Atualizar estado do pedido para “Pronto” quando todas as tarefas estiverem concluídas.

4.2.4. Entrega de Pedidos

Segue-se o processo para os casos de uso relevantes na responsabilidade de entrega de pedidos.

4.2.4.1. Entregar Pedido

Caso de uso	Entregar pedido
Fluxo normal	<ol style="list-style-type: none"> 1. O ator solicita aceder à lista de pedidos prontos para entrega. 2. O sistema apresenta os pedidos prontos para entrega (estado = “Pronto”). 3. O ator seleciona o pedido que irá entregar. 4. O sistema apresenta os detalhes do pedido (número, produtos, tipo de entrega). 5. O ator confirma a entrega ao cliente. 6. O sistema atualiza o estado do pedido para “Entregue” e regista o horário da entrega.

Tabela 20: Use Case: Entregar Pedido.

Responsabilidades identificadas:

- 2 -> Listar pedidos com estado “Pronto” (prontos para entrega);
- 4 -> Procurar detalhes completos de um pedido pelo seu número;
- 6 -> Atualizar estado do pedido para “Entregue”;
- 6 -> Registar data e hora da entrega.

5. Interface da Lógica de Negócio

Após a análise exaustiva dos casos de uso, identificaram-se todas as responsabilidades que a lógica de negócio terá de cumprir para satisfazer os requisitos funcionais. Apresenta-se, de seguida, uma pseudo-interface que agrupa os métodos identificados por área funcional.

ICadeiaLN — Interface da Lógica de Negócio
<p>Gestão de Restaurantes</p> <p><code>listarTiposIndicadores()</code> — devolve a lista de tipos de indicadores disponíveis. <code>calcularIndicador(tipo, idRest?)</code> — calcula o valor de um indicador, opcionalmente para um restaurante específico. <code>listarRestaurantes()</code> — devolve a lista de restaurantes registados no sistema.</p> <p>Gestão de Propostas e Pedidos</p> <p><code>iniciarNovoPedido()</code> - Cria um novo pedido vazio no sistema e retorna o seu id. <code>getPropostasDisponiveis()</code> — devolve as propostas que o cliente pode escolher. <code>listarPersonalizacoesPossiveis(nomeProposta)</code> — devolve as personalizações associadas a uma proposta. <code>adicionarPropostaAoPedido(idPedido, nomeProposta)</code> - adiciona uma proposta ao pedido e incrementa o seu preço total. <code>finalizarPedido(idPedido)</code> - altera o estado do pedido para “Pendente” e define o tempo estimado para a sua entrega <code>listarPedidosEstado(estado)</code>- lista todos os pedidos que estejam num dado estado (Ex: Concluido, Entregue, etc...) <code>atualizarEstadoPedido(idPed, estado)</code> — atualiza o pedido para estado para um dado</p>

estado.

getPedido(idPed) — devolve um pedido.

Gestão de Tarefas e Confeção

adicionarDelayATarefa(idTarefa, delay) — atualiza o tempo estimado de preparação de um pedido.

recalcularAgendamento() — reajusta os tempos das tarefas de outros postos, dado um reagendamento de uma tarefa pela parte de uma posto

calculaETAPedido(idPedido) — estima o tempo total necessário para preparar um pedido.

decomporPedido(Pedido) — gera as tarefas de confeção necessárias para o pedido.

listarTarefasDoPosto(tipoPosto) — devolve as tarefas atribuídas a um posto.

marcarTarefaComoConcluidaNoPosto(tipoPosto, idTarefa) — regista a conclusão de uma tarefa.

5.1. Identificação dos Subsistemas

5.1.1. Motivação

Embora a tabela fornecida acima forneça uma visão completa das responsabilidades da lógica de negócio, torna-se evidente que existe uma clara separação de responsabilidades entre diferentes áreas funcionais do sistema.

Pelo princípio da separação de responsabilidades e para melhorar a coesão e manutenibilidade do sistema, procedeu-se à criação de três subsistemas distintos, cada um com responsabilidades bem definidas.

5.1.2. Subsistemas Identificados

Identificaram-se três subsistemas principais:

- **SubsistemaRestaurantes** — Gestão de restaurantes e indicadores de desempenho
 - Responsável por operações relacionadas com a gestão da cadeia de restaurantes
 - Cálculo e consulta de indicadores de performance
- **SubsistemaPedidos** — Gestão de pedidos e propostas
 - Responsável pelo ciclo de vida completo do pedido
- **SubsistemaTarefas** - Gestão de tarefas e postos
 - Atribuição de tarefas aos funcionários e postos
 - Acompanhamento do progresso de confeção dos pedidos

É relevante referir que, nesta implementação, não aplicamos uma noção explícita de “Cliente” como entidade gerida pelo sistema. O cliente interage com o sistema através dos ecrãs táteis para efetuar pedidos, mas não é necessário manter informação persistente sobre o mesmo (não se guardam registos, históricos de compras, ou perfis de utilizadores). O foco está na gestão operacional dos pedidos e não na gestão de clientes.

5.2. Diagrama de componentes da lógica de negócios

Tendo todas as componentes do sistema devidamente definidas, podemos estruturar de forma clara a camada responsável pela lógica de negócios. Esta camada central coordena os vários subsistemas especializados, garantindo a correta execução das operações do sistema.

Quanto à camada de dados, esta permanece abstrata, por enquanto. A implementação concreta dos mecanismos de persistência não é necessária nesta seção e não influencia o desenho da lógica de negócios, podendo ser desenvolvida autonomamente numa etapa posterior.

No que diz respeito à interface com o utilizador (UI), optou-se por uma solução simples e funcional, suficiente para demonstrar e validar os fluxos do sistema. Esta abordagem minimalista será mantida ao longo de todo o projeto, dado que o foco principal reside na modelação e implementação da lógica de negócios, e não no desenvolvimento de uma interface gráfica complexa.

Dado o âmbito desta aplicação, o facto de existirem vários restaurantes implica que cada restaurante deva possuir, internamente, um Gestor de Pedidos e um Gestor de Tarefas. Esta decisão justifica-se pelo facto de estas componentes estarem naturalmente associadas a um restaurante individual, e não à cadeia como um todo. Assim sendo, o diagrama de componentes do sistema foi estruturado de acordo com esta lógica.

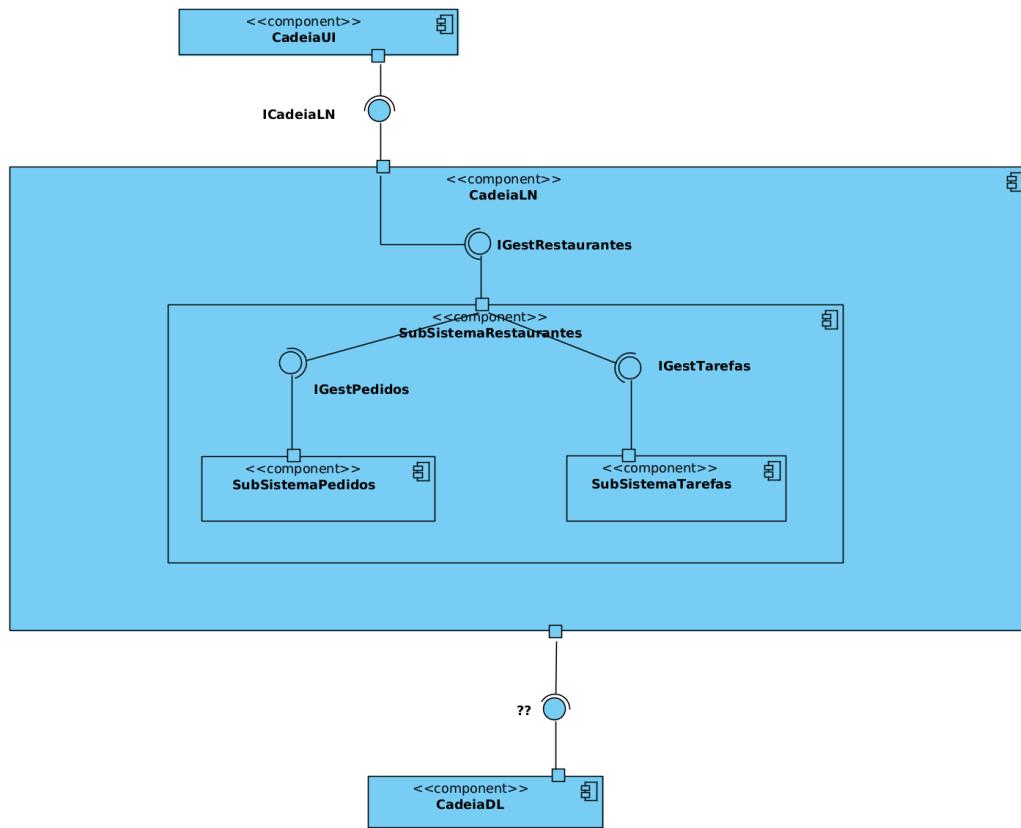


Figura 6: Diagrama de componentes obtido.

6. Diagrama de classes dos subsistemas da lógica de negócios

Após estabelecer a arquitetura de componentes do sistema, torna-se necessário detalhar a estrutura interna de cada subsistema através de diagramas de classes. Esta modelação permite

especificar as entidades que compõem cada componente, bem como as relações e responsabilidades que estabelecem entre si.

Esta análise detalhada constitui a base para a posterior implementação dos diagramas de sequência dos métodos do sistema, garantindo que todas as funcionalidades necessárias estão adequadamente representadas e que a estrutura do código reflete fielmente os requisitos do domínio do problema.

6.1. Diagrama de classes do subsistema de Pedidos

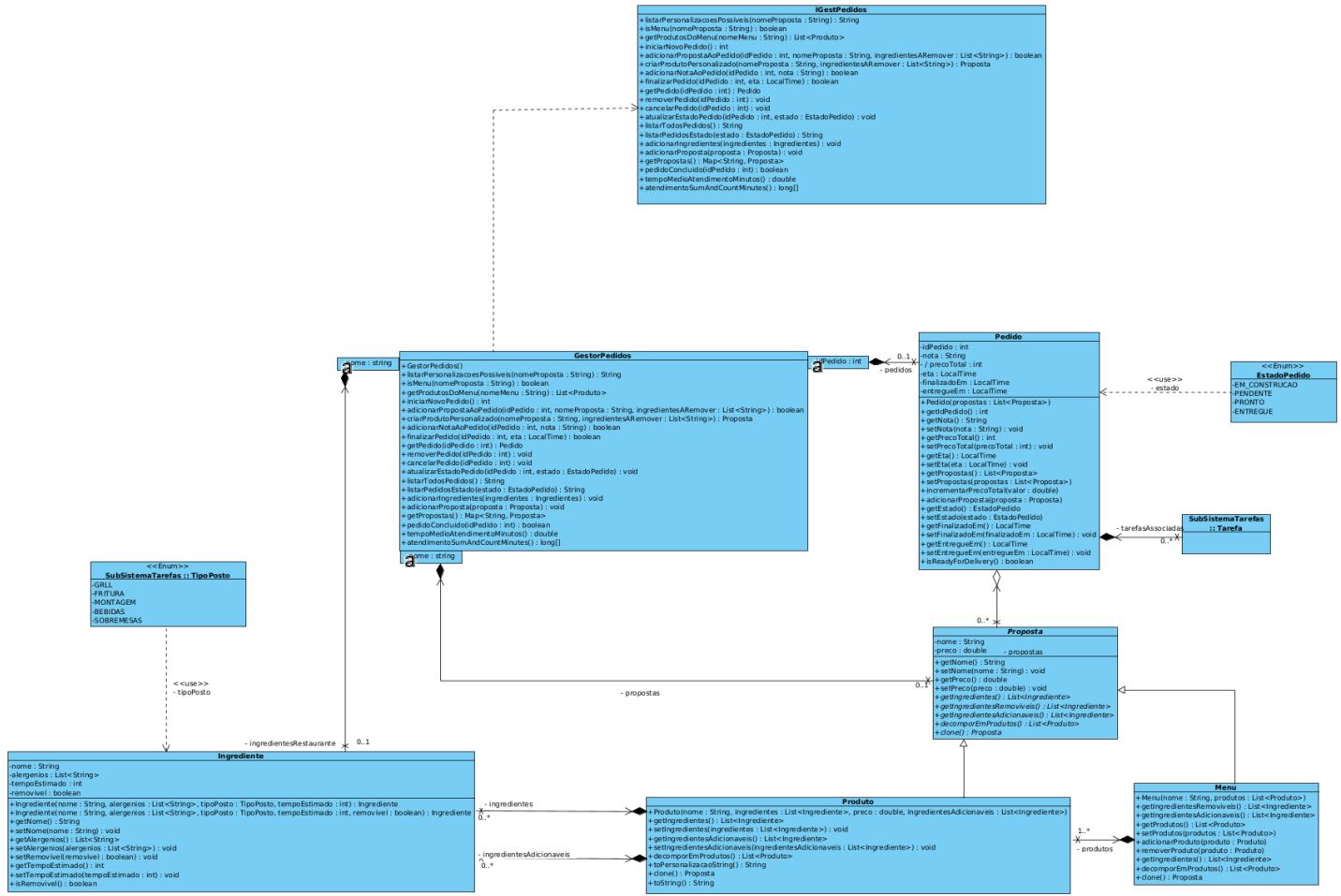


Figura 7: Modelo de Domínio obtido.

6.2. Diagrama de classes do subsistema de Tarefas

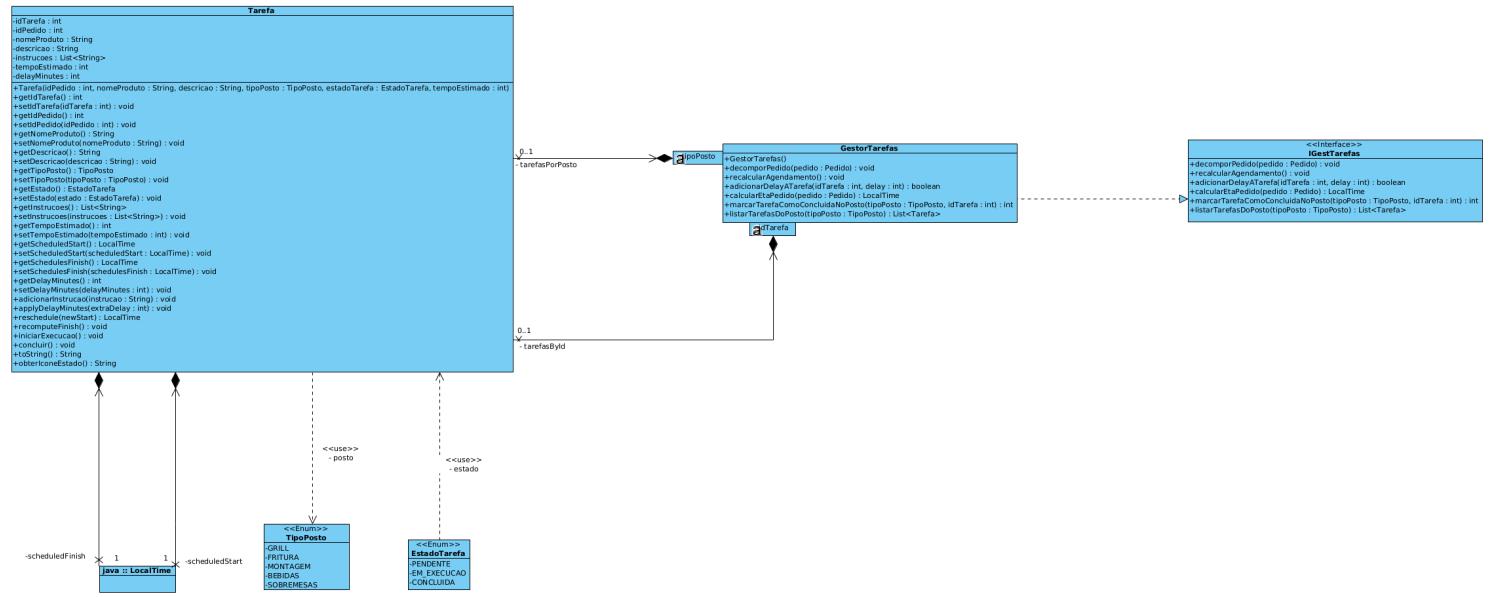


Figura 8: Modelo de Domínio obtido.

6.3. Diagrama Classes do subsistema de Restaurantes

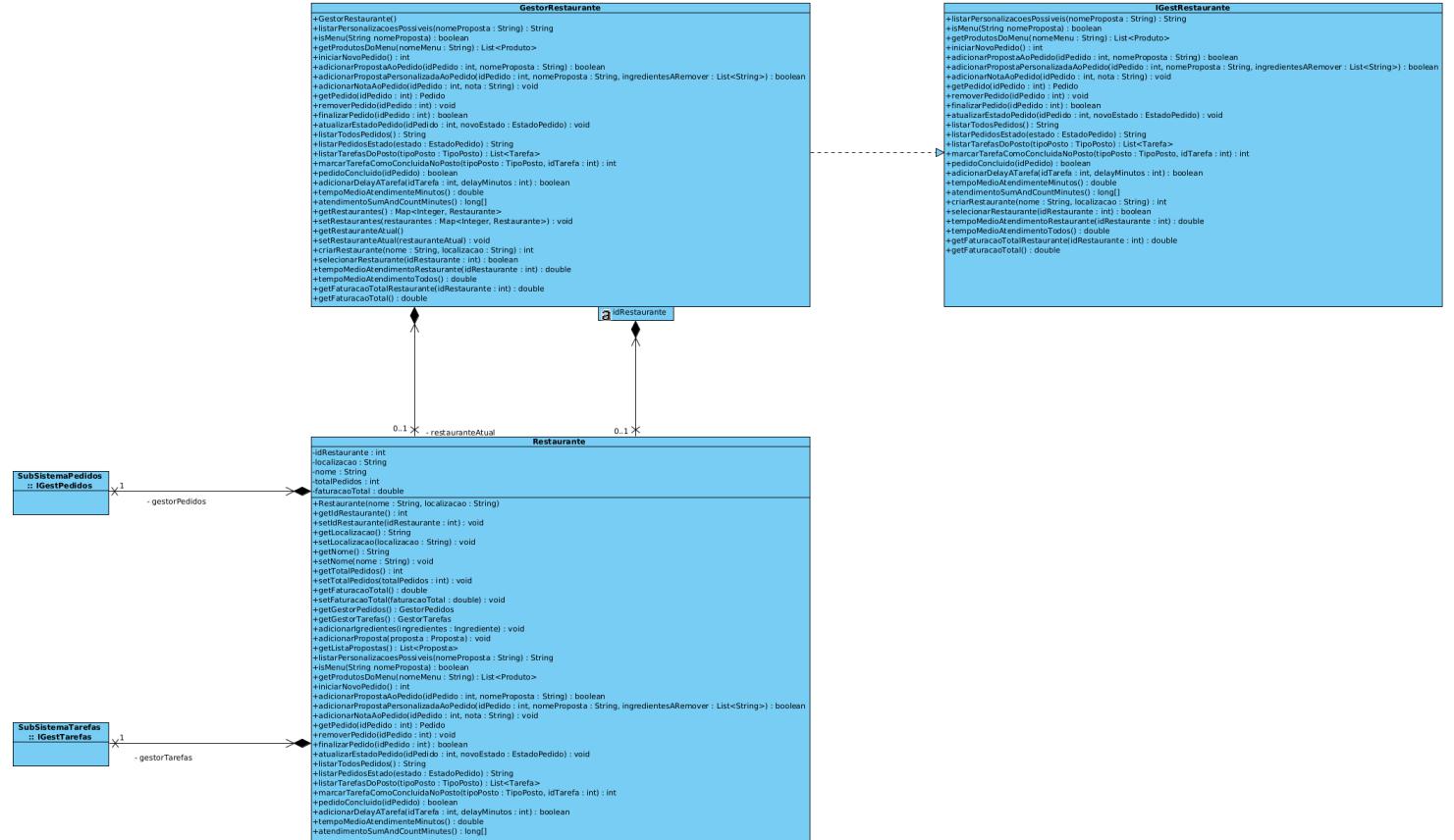


Figura 9: Modelo de Domínio obtido.

7. Implementação das DAOs

O modelo realizado até ao momento não contempla persistência de dados, pelo que é necessário implementar DAOs (*Data Access Objects*) que permitam o armazenamento e recuperação de informação entre execuções do sistema.

Para simplificar a integração com a modelação já definida, optou-se por implementar DAOs que respeitam a interface Map<K, V> do Java. Esta decisão obriga os DAOs a implementarem métodos como put(), get(), remove(), entre outros, facilitando a implementação dos DAOs na estrutura já modelada.

Esta abordagem reduz consideravelmente as alterações necessárias ao modelo: basta substituir as instâncias de Map previstas no *design* por instâncias dos respetivos DAOs. Assim, garante-se persistência sem comprometer a arquitetura já definida, sendo boa prática de seguir.

7.1. Criação das tabelas

Para realizar o mapeamento das classes para tabelas relacionais, optámos pela abordagem simples de criar uma tabela por classe, conforme estudado nas aulas teóricas.

Ficamos com as seguintes tabelas:

```

CREATE TABLE IF NOT EXISTS Restaurante (
    idRestaurante INT PRIMARY KEY AUTO_INCREMENT,
    nome VARCHAR(100) NOT NULL,
    localizacao VARCHAR(200) NOT NULL,
    totalPedidos INT DEFAULT 0,
    faturacaoTotal DOUBLE DEFAULT 0.0
);

CREATE TABLE IF NOT EXISTS Ingrediente (
    nome          VARCHAR(100),
    idRestaurante INT,
    tipoPosto     VARCHAR(50) NOT NULL, -- enum TipoPosto transformado em string
    tempoEstimado INT NOT NULL DEFAULT 1,
    removivel    BOOLEAN DEFAULT TRUE,
    PRIMARY KEY (nome, idRestaurante),
    FOREIGN KEY (idRestaurante) REFERENCES Restaurante(idRestaurante) ON DELETE CASCADE
);

-- Alergénios de um ingrediente (relação 1:N)
CREATE TABLE IF NOT EXISTS Ingrediente_Alergenio (
    nomeIngrediente VARCHAR(100),
    idRestaurante   INT,
    alergenio       VARCHAR(100),
    PRIMARY KEY (nomeIngrediente, idRestaurante, alergenio),
    FOREIGN KEY (nomeIngrediente, idRestaurante) REFERENCES Ingrediente(nome,
    idRestaurante) ON DELETE CASCADE
);

CREATE TABLE IF NOT EXISTS Proposta (
    nome          VARCHAR(100),
    idRestaurante INT,
    preco         DOUBLE NOT NULL,
    tipo          VARCHAR(10) NOT NULL, -- 'PRODUTO' ou 'MENU'

```

```
        PRIMARY KEY (nome, idRestaurante),
        FOREIGN KEY (idRestaurante) REFERENCES Restaurante(idRestaurante) ON DELETE
CASCADE
);

CREATE TABLE IF NOT EXISTS Produto (
    nome          VARCHAR(100),
    idRestaurante INT,
    PRIMARY KEY (nome, idRestaurante),
    FOREIGN KEY (nome, idRestaurante) REFERENCES Proposta(nome, idRestaurante) ON
DELETE CASCADE
);

-- Ingredientes de um produto (relação N:M)
CREATE TABLE IF NOT EXISTS Produto_Ingrediente (
    nomeProduto      VARCHAR(100),
    idRestaurante    INT,
    nomeIngrediente  VARCHAR(100),
    PRIMARY KEY (nomeProduto, idRestaurante, nomeIngrediente),
    FOREIGN KEY (nomeProduto, idRestaurante) REFERENCES Produto(nome, idRestaurante)
ON DELETE CASCADE,
    FOREIGN KEY (nomeIngrediente, idRestaurante) REFERENCES Ingrediente(nome,
idRestaurante) ON DELETE CASCADE
);

-- Ingredientes adicionáveis a um produto (relação N:M)
CREATE TABLE IF NOT EXISTS Produto_IngredienteAdicionavel (
    nomeProduto      VARCHAR(100),
    idRestaurante    INT,
    nomeIngrediente  VARCHAR(100),
    PRIMARY KEY (nomeProduto, idRestaurante, nomeIngrediente),
    FOREIGN KEY (nomeProduto, idRestaurante) REFERENCES Produto(nome, idRestaurante)
ON DELETE CASCADE,
    FOREIGN KEY (nomeIngrediente, idRestaurante) REFERENCES Ingrediente(nome,
idRestaurante) ON DELETE CASCADE
);

CREATE TABLE IF NOT EXISTS Menu (
    nome          VARCHAR(100),
    idRestaurante INT,
    PRIMARY KEY (nome, idRestaurante),
    FOREIGN KEY (nome, idRestaurante) REFERENCES Proposta(nome, idRestaurante) ON
DELETE CASCADE
);

-- Produtos de um menu (relação N:M)
CREATE TABLE IF NOT EXISTS Menu_Produto (
    nomeMenu        VARCHAR(100),
    idRestaurante   INT,
    nomeProduto     VARCHAR(100),
    PRIMARY KEY (nomeMenu, idRestaurante, nomeProduto),
    FOREIGN KEY (nomeMenu, idRestaurante) REFERENCES Menu(nome, idRestaurante) ON
DELETE CASCADE,
    FOREIGN KEY (nomeProduto, idRestaurante) REFERENCES Produto(nome, idRestaurante)
ON DELETE CASCADE
);
```

```

CREATE TABLE IF NOT EXISTS Pedido (
    idPedido      INT PRIMARY KEY AUTO_INCREMENT,
    idRestaurante INT NOT NULL,
    nota          VARCHAR(500),
    precoTotal    DOUBLE DEFAULT 0.0,
    estado        VARCHAR(20) NOT NULL, -- enum EstadoPedido como string
    eta           TIME,
    finalizadoEm  TIME,
    entregueEm    TIME,
    FOREIGN KEY (idRestaurante) REFERENCES Restaurante(idRestaurante) ON DELETE
CASCADE
);

-- Ingredientes da proposta num pedido (para produtos personalizados)
CREATE TABLE IF NOT EXISTS Pedido_Proposta_Ingrediente (
    idPedido      INT,
    ordemProposta INT,
    nomeIngrediente VARCHAR(100),
    quantidade    INT DEFAULT 1,
    PRIMARY KEY (idPedido, ordemProposta, nomeIngrediente),
    FOREIGN KEY (idPedido, ordemProposta) REFERENCES Pedido_Proposta(idPedido,
ordemProposta) ON DELETE CASCADE
);

CREATE TABLE IF NOT EXISTS Tarefa (
    idTarefa      INT PRIMARY KEY,
    idRestaurante INT NOT NULL,
    idPedido      INT NOT NULL,
    nomeProduto   VARCHAR(100),
    descricao     VARCHAR(500),
    tipoPosto     VARCHAR(50) NOT NULL, -- enum TipoPosto como string
    estado        VARCHAR(20) NOT NULL, -- enum EstadoTarefa como string
    tempoEstimado INT DEFAULT 1,
    scheduledStart TIME,
    scheduledFinish TIME,
    delayMinutes   INT DEFAULT 0,
    FOREIGN KEY (idRestaurante) REFERENCES Restaurante(idRestaurante) ON DELETE
CASCADE,
    FOREIGN KEY (idPedido) REFERENCES Pedido(idPedido) ON DELETE CASCADE
);

-- Instruções de uma tarefa (relação 1:N)
CREATE TABLE IF NOT EXISTS Tarefa_Instrucao (
    idTarefa      INT,
    ordemInstrucao INT,
    instrucao     VARCHAR(200),
    PRIMARY KEY (idTarefa, ordemInstrucao),
    FOREIGN KEY (idTarefa) REFERENCES Tarefa(idTarefa) ON DELETE CASCADE
);

-- Propostas de um pedido (cópia da proposta no momento do pedido)
-- Como as propostas podem ser personalizadas, guardamos uma cópia serializada
CREATE TABLE IF NOT EXISTS Pedido_Proposta (
    idPedido      INT,
    ordemProposta INT, -- ordem da proposta no pedido

```

```

nomeProposta      VARCHAR(100),
tipoProposta      VARCHAR(10), -- 'PRODUTO' ou 'MENU'
precoUnitario     DOUBLE,
PRIMARY KEY (idPedido, ordemProposta),
FOREIGN KEY (idPedido) REFERENCES Pedido(idPedido) ON DELETE CASCADE
);

```

7.2. Diagrama de componentes e de classes atualizados

A introdução das classes de acesso a dados (DAOs) no sistema implica a necessidade de atualizar os diagramas de classes e de componentes previamente definidos, refletindo com rigor a nova arquitetura de persistência implementada.

Os diagramas seguintes ilustram estas atualizações, mantendo a coerência com o modelo de domínio e os casos de uso estabelecidos anteriormente.

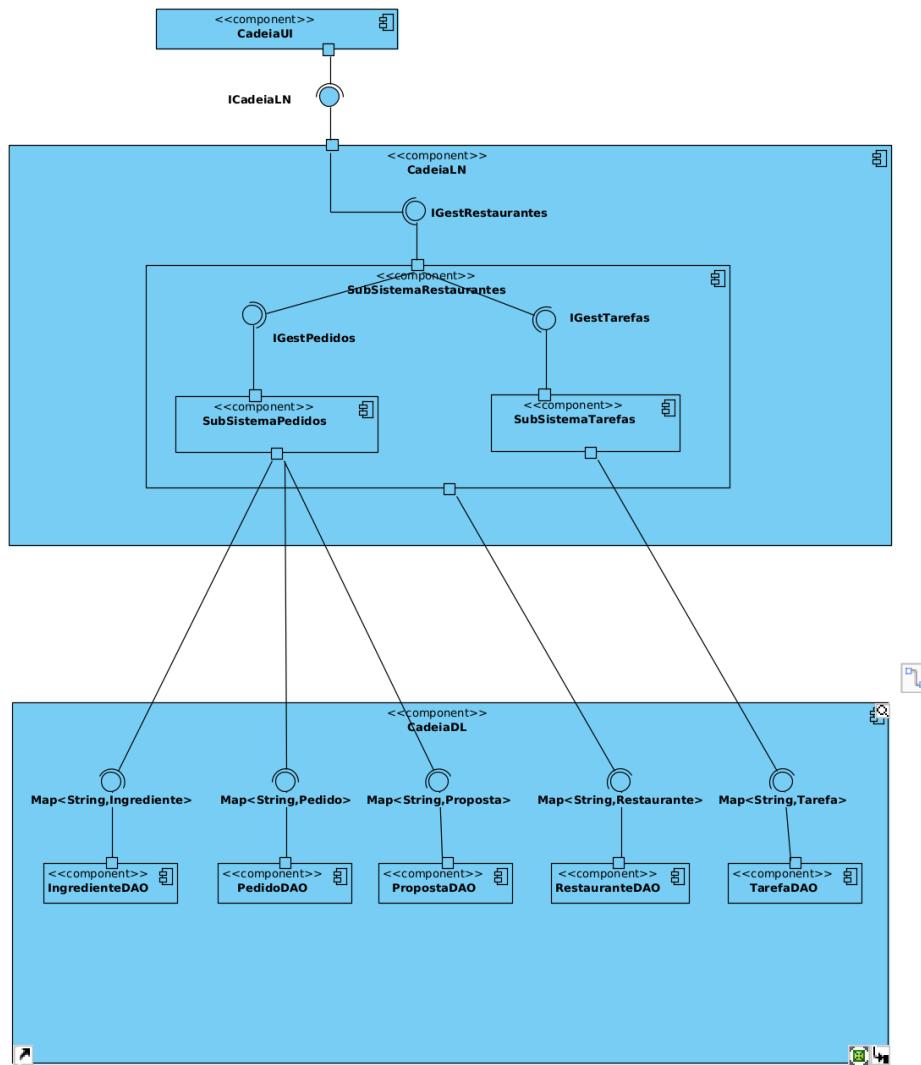


Figura 10: Diagrama de componentes com implementação DAO.

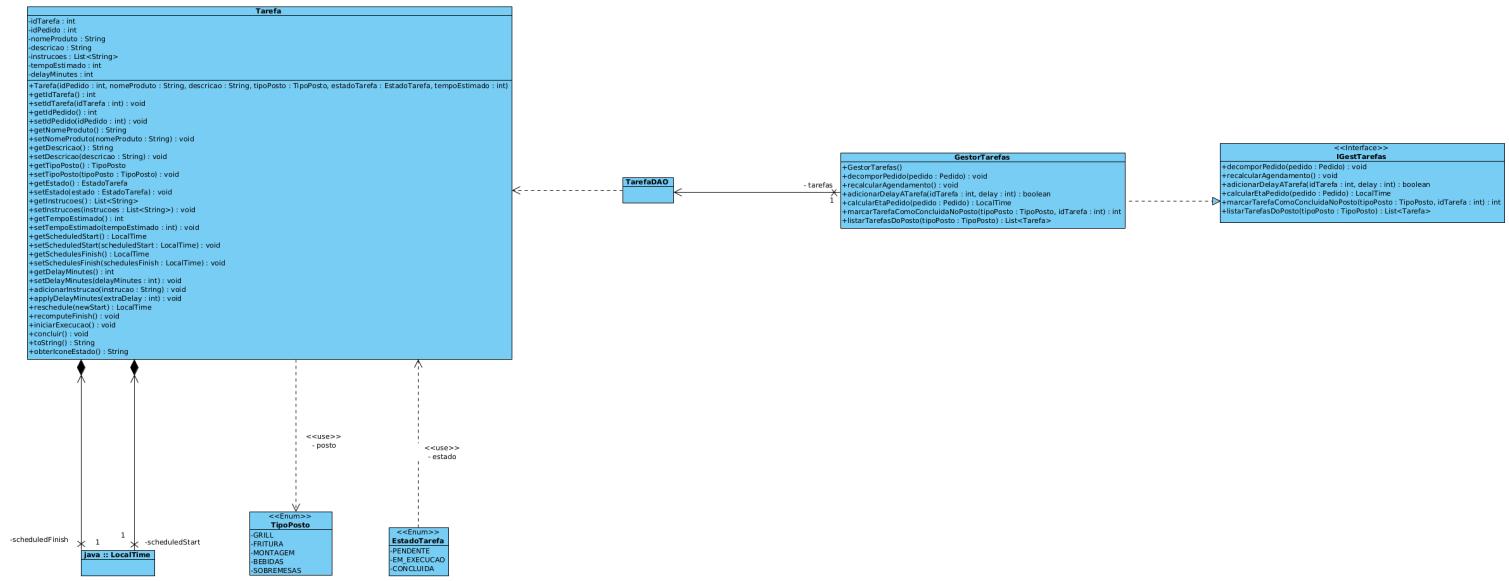


Figura 11: Diagrama de classes do módulo de Tarefas com implementação DAO.

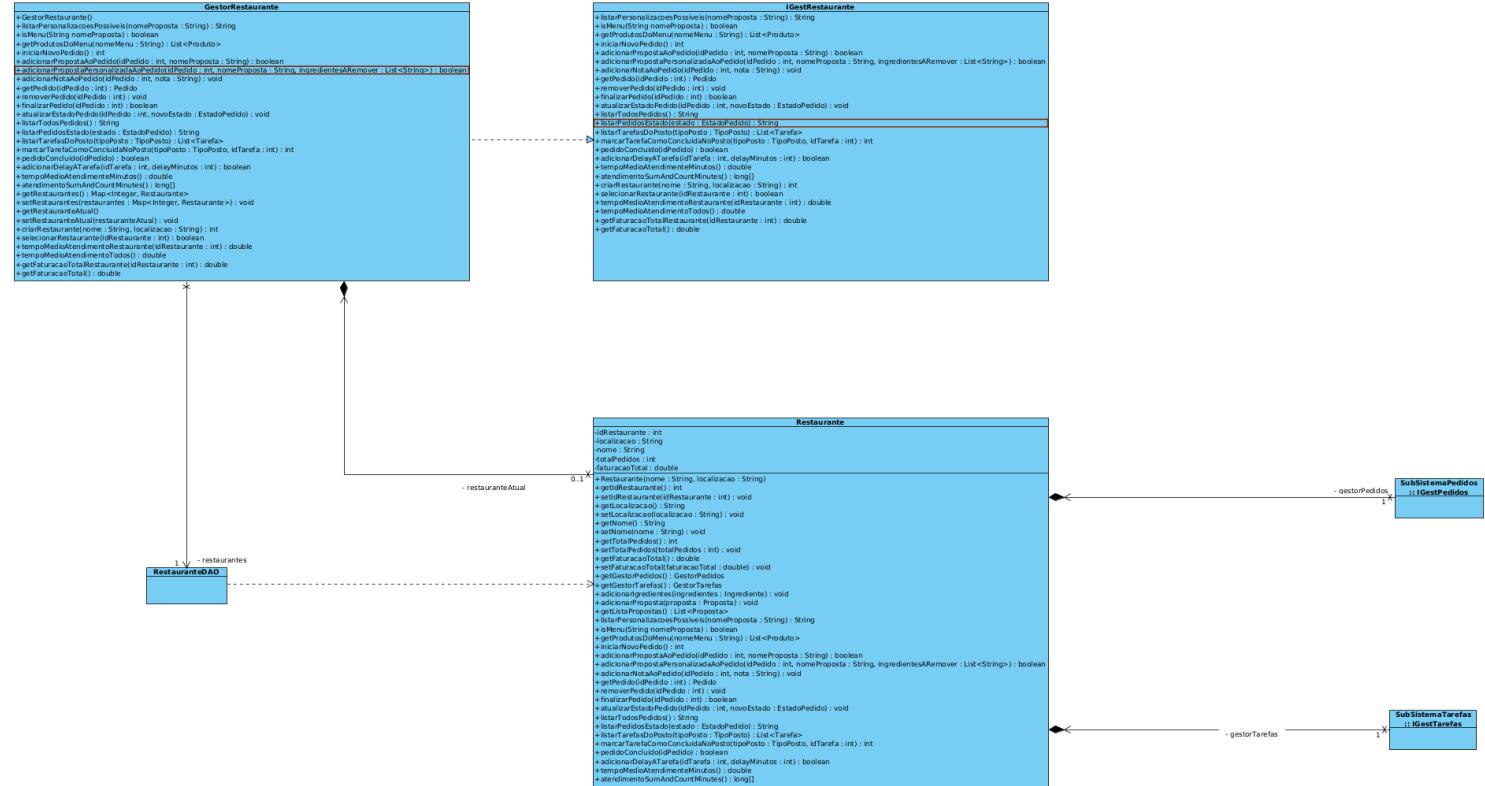


Figura 12: Diagrama de classes do módulo de Restaurantes com implementação DAO.

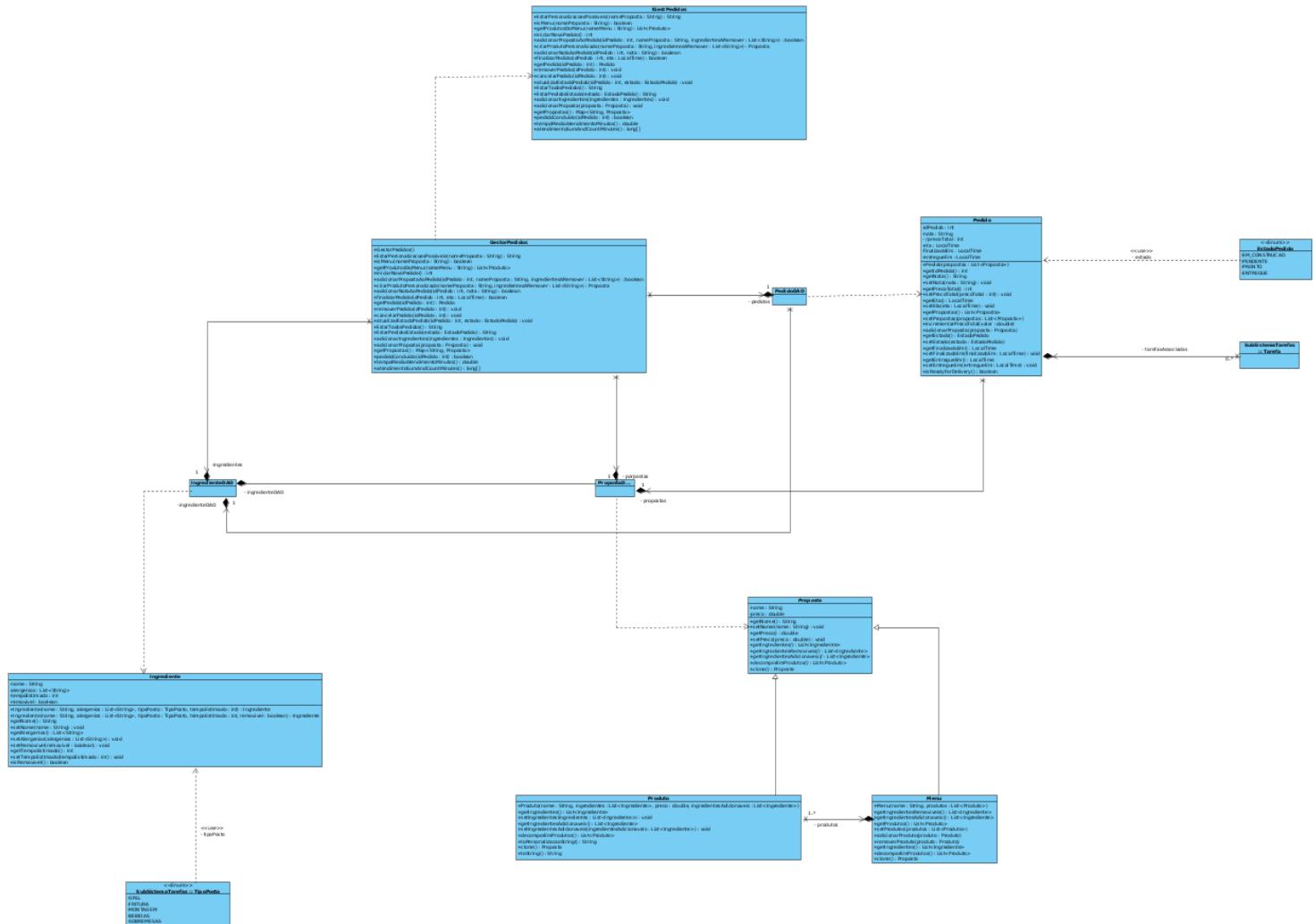


Figura 13: Diagrama de classes do módulo de Pedidos com implementação DAO.

Com a arquitetura de persistência devidamente estruturada através dos DAOs, torna-se possível avançar para a modelação do comportamento dinâmico do sistema através de diagramas de sequência.

8. Diagramas de Sequência

Os diagramas de sequência desta seção ilustram o fluxo de execução das principais funcionalidades do sistema. Por questões de clareza e concisão, optou-se por omitir operações triviais e amplamente conhecidas, tais como métodos de acesso (*getters* e *setters*), focando-nos exclusivamente nos fluxos de interação mais relevantes e complexos do sistema.

8.1. Adicionar delay a uma tarefa

Esta operação tem como objetivo reagendar o tempo estimado de realização de uma tarefa. A operação devolve falso caso a tarefa não exista ou já esteja concluída.

Se a operação for válida, o delay é aplicado à tarefa em questão e o tempo estimado das restantes tarefas associadas é recalculado.

Por motivos de simplificação, assumimos que um posto apenas pode executar tarefas de forma sequencial. Assim, o tempo estimado de cada tarefa depende da execução de todas as tarefas anteriores. Deste modo, se uma tarefa *A* for executada antes de uma tarefa *B*, então o tempo estimado de *B* será dado por tempoEstimado(*A*) + tempoExecução(*A*).

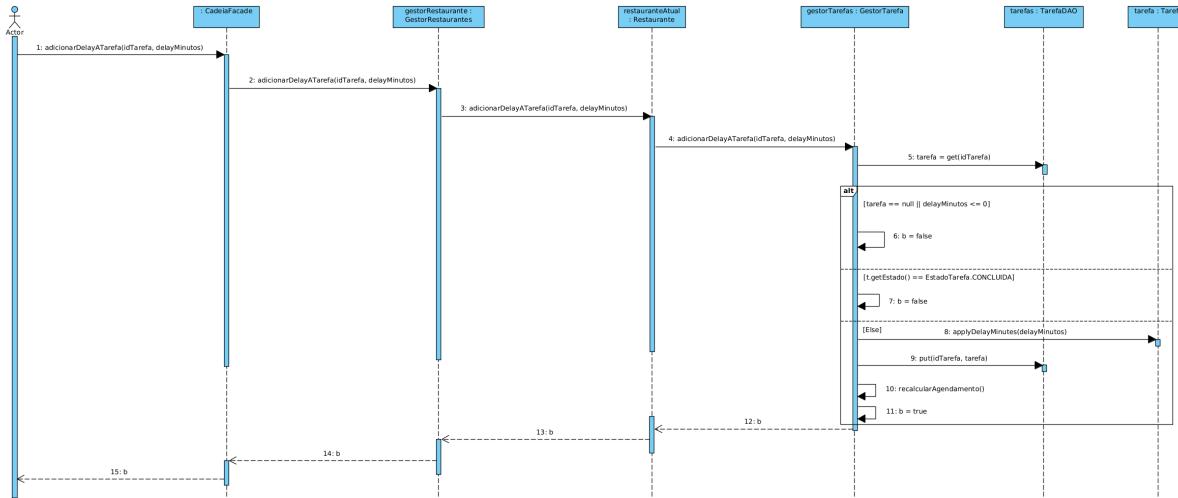


Figura 14: Diagrama de sequência: Adicionar atraso a uma tarefa.

8.2. Adicionar proposta a um pedido

Esta operação adiciona uma nova proposta a um pedido que se encontra em construção. A proposta pode corresponder a um menu, a um produto ou a um produto personalizado.

A proposta adicionada ao pedido é realizada através de composição, uma vez que uma proposta representa apenas uma abstração. Isto significa que a classe Proposta não define um objeto concreto por si só, mas sim um conceito genérico que é concretizado por diferentes tipos de propostas (menu, produto ou produto personalizado). Assim, o pedido não depende de implementações específicas, mas apenas da abstração Proposta, permitindo maior flexibilidade e extensibilidade do sistema.

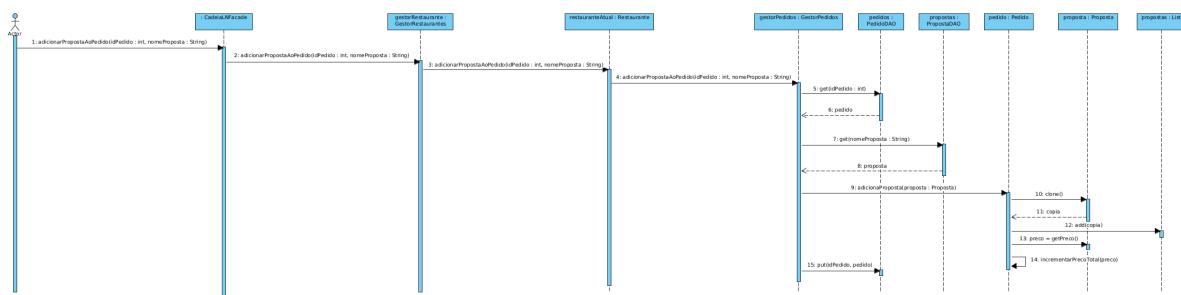


Figura 15: Diagrama de sequência: Adicionar proposta a um pedido.

8.3. Atualizar o estado de um pedido

O nosso sistema possui diversos estados, pelo que considerámos pertinente implementar uma função responsável por atualizar o estado de um pedido.

Esta função é simples, consistindo apenas em obter o pedido a partir da base de dados e atualizar o seu estado para o novo estado pretendido.

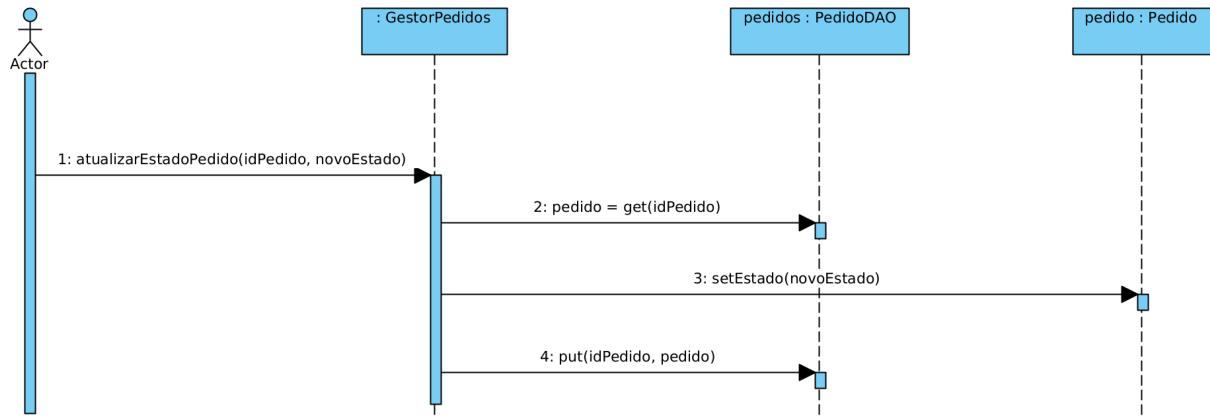


Figura 16: Diagrama de sequência: Atualizar estado de um pedido.

8.4. Calcular o tempo estimado de um pedido

Esta operação tem como objetivo fornecer ao cliente o tempo estimado para o seu pedido ficar pronto. Para tal, é realizada a iteração por todas as tarefas associadas ao pedido, sendo determinado o tempo restante de cada uma. O maior desses tempos corresponde ao ETA (Estimated Time of Arrival) do pedido.

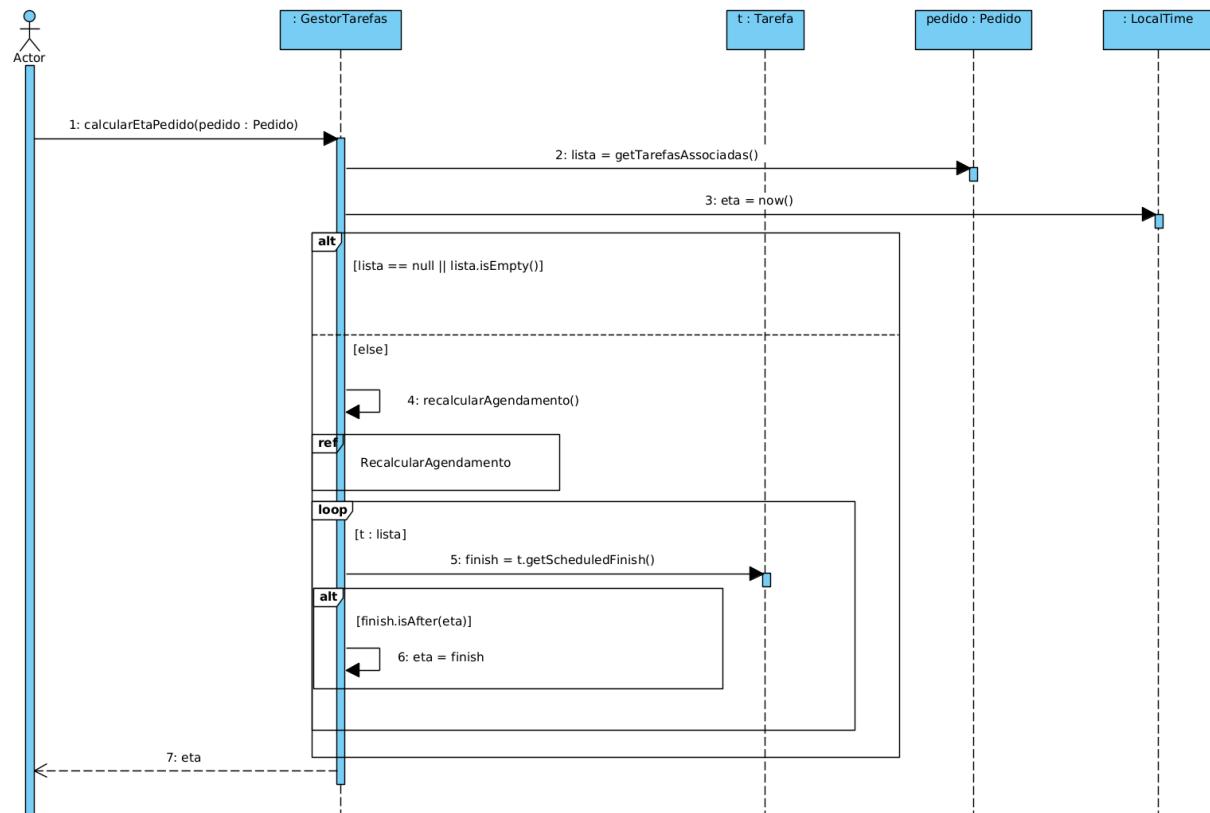


Figura 17: Diagrama de sequência: Calcular eta de um pedido.

8.5. Decompor um pedido em tarefas

Esta operação é uma das mais complexas do sistema, sendo executada no momento em que o cliente finaliza o seu pedido.

No nosso sistema, cada Produto possui uma lista de ingredientes associada, e cada ingrediente corresponde a uma tarefa individual. Por exemplo, o ingrediente carne origina uma tarefa associada à grelha, enquanto o ingrediente pickle origina uma tarefa associada à montagem. Assim, para um único produto podem ser criadas várias tarefas distintas.

Esta operação percorre todos os produtos presentes no pedido e, para cada produto, itera sobre a sua lista de ingredientes, criando uma nova tarefa para cada ingrediente. Cada tarefa criada é posteriormente adicionada à lista de tarefas do sistema, permitindo a correta gestão e execução do pedido.

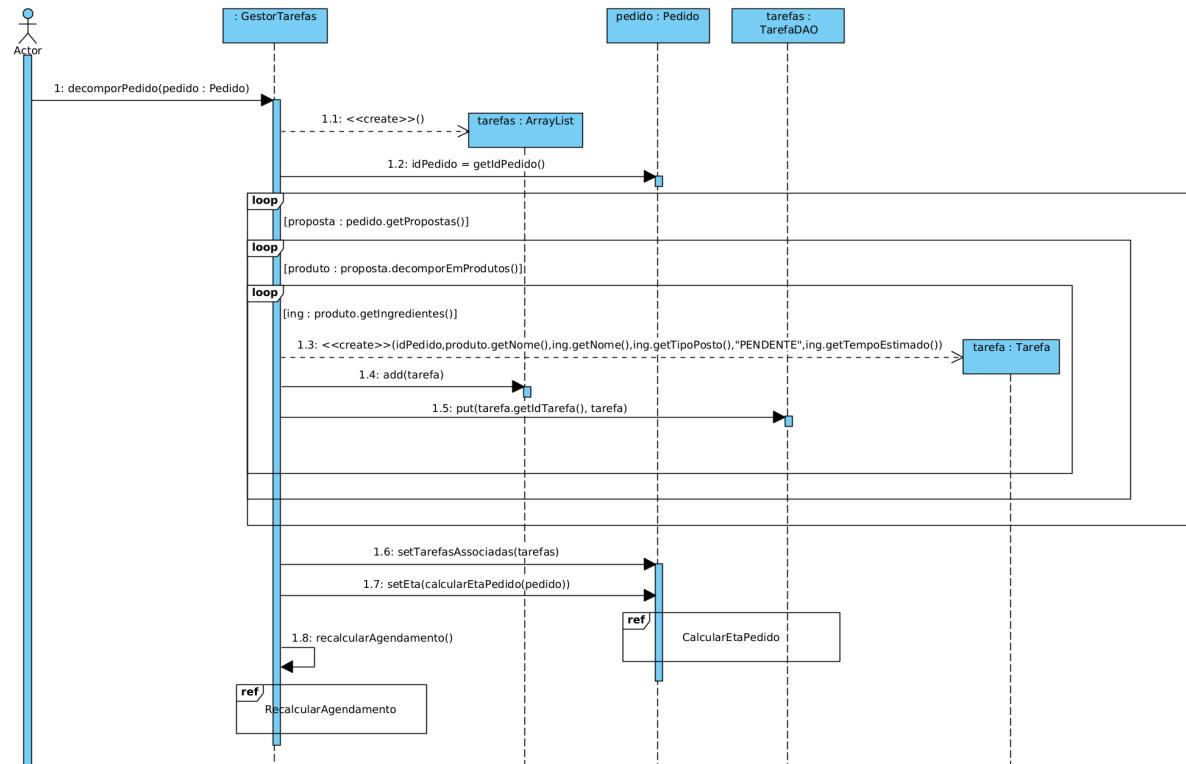


Figura 18: Diagrama de sequência: Decompor pedido em tarefas.

8.6. Ver a faturação de um restaurante

Esta operação é trivial: dado o identificador de um restaurante existente, o sistema retorna a sua faturação total, calculada como a soma do custo de todos os pedidos já realizados.

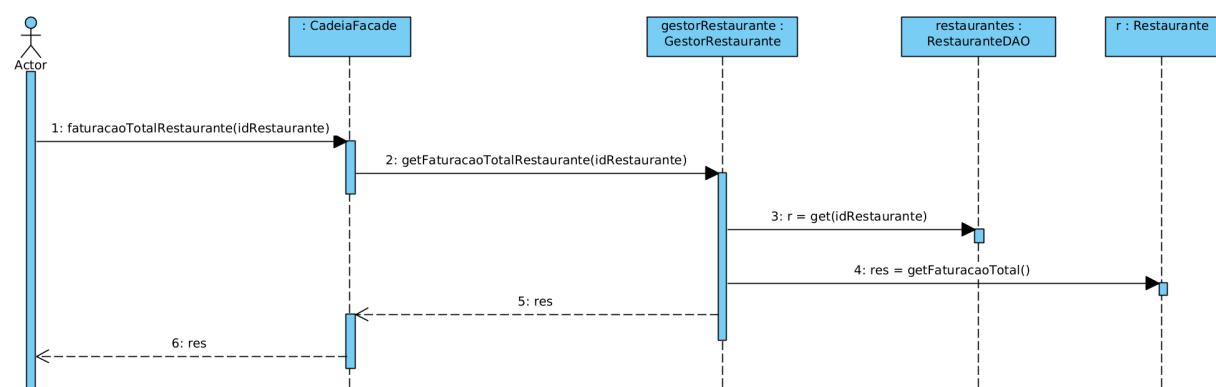


Figura 19: Diagrama de sequência: Consultar faturação total de um restaurante.

8.7. Ver a faturação de todos os restaurantes

Semelhante à operação anterior, nesta operação é feita a iteração por todos os restaurantes, mantendo um valor acumulado da faturação de cada um. Desta forma, obtém-se a faturação total da cadeia.

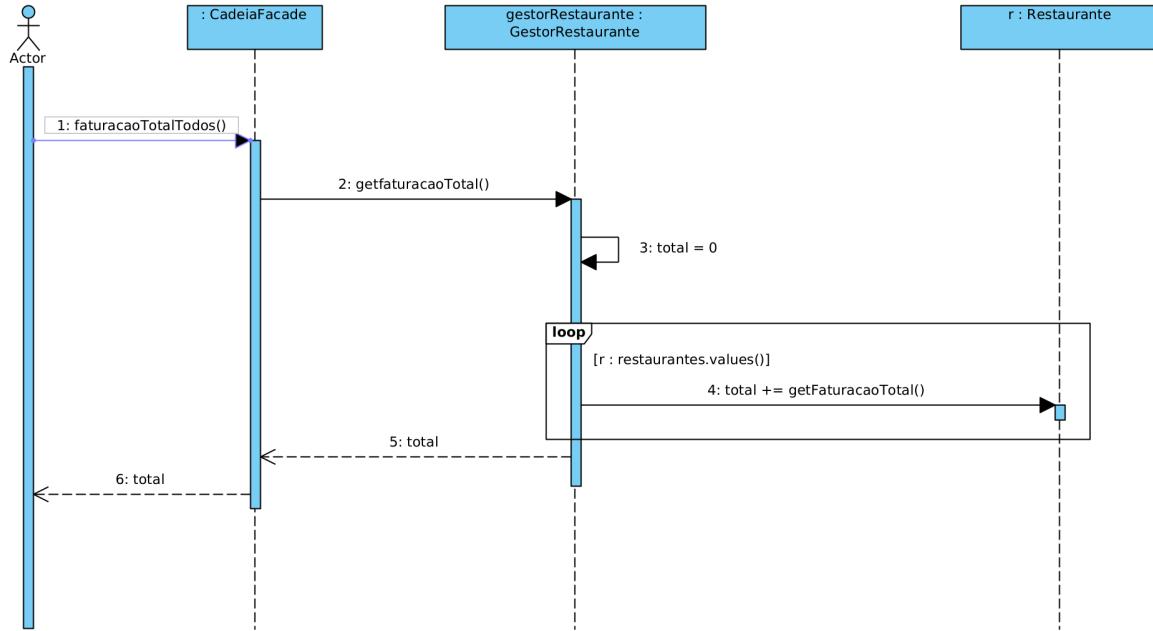


Figura 20: Diagrama de sequência: Consultar faturação total de todos os restaurantes.

8.8. Finalizar um pedido

Esta operação verifica se um pedido pode ser finalizado, isto é, se o mesmo não é nulo nem se encontra vazio. Caso o pedido seja válido, este é decomposto através da chamada da operação descrita anteriormente. No final, é retornado um valor booleano que indica a validade do pedido.

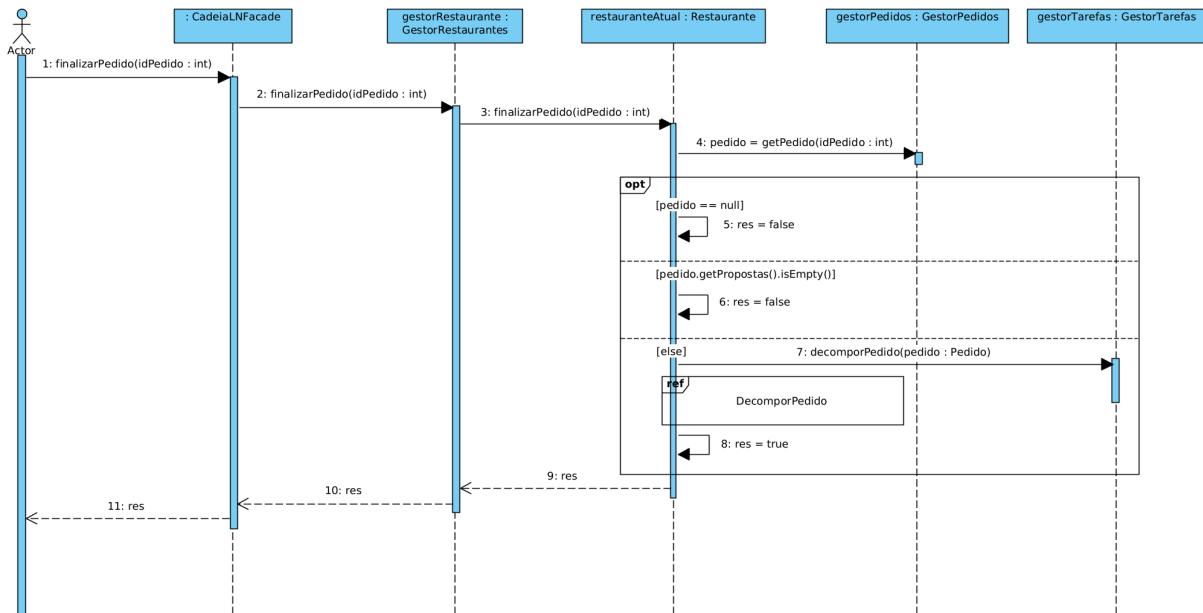


Figura 21: Diagrama de sequência: Finalizar pedido.

8.9. Obter a lista das propostas disponíveis

Esta operação tem como objetivo obter a lista de todas as propostas disponíveis no sistema, de forma a apresentá-las ao cliente, permitindo-lhe selecionar a proposta pretendida. Para este efeito, apenas é fornecido o nome de cada proposta, que, no contexto do sistema, funciona como identificador da mesma.

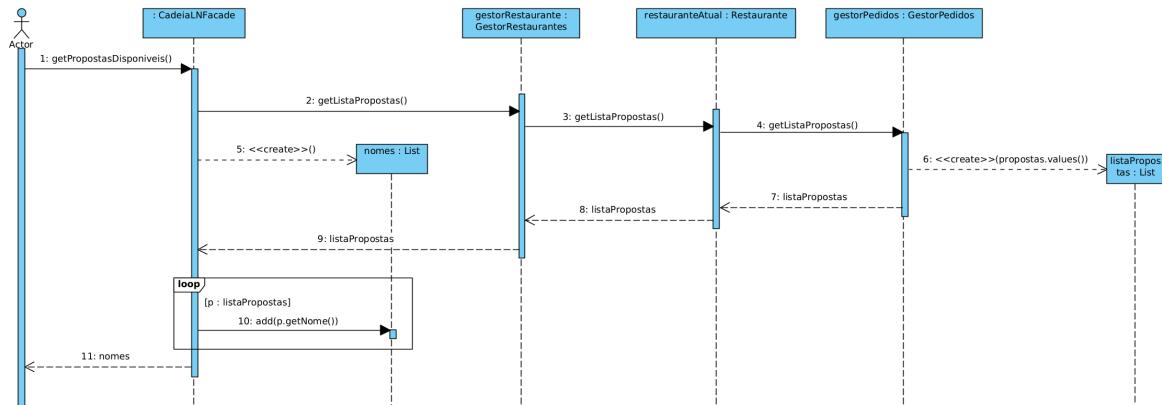


Figura 22: Diagrama de sequência: Obter lista de propostas disponíveis.

8.10. Iniciar um novo pedido

Quando um cliente indica que pretende iniciar um pedido, é criado no sistema um novo pedido vazio. É devolvido ao cliente o identificador desse pedido, permitindo-lhe, posteriormente, adicionar as propostas pretendidas através da operação *Adicionar Proposta* (Seção 8.2).

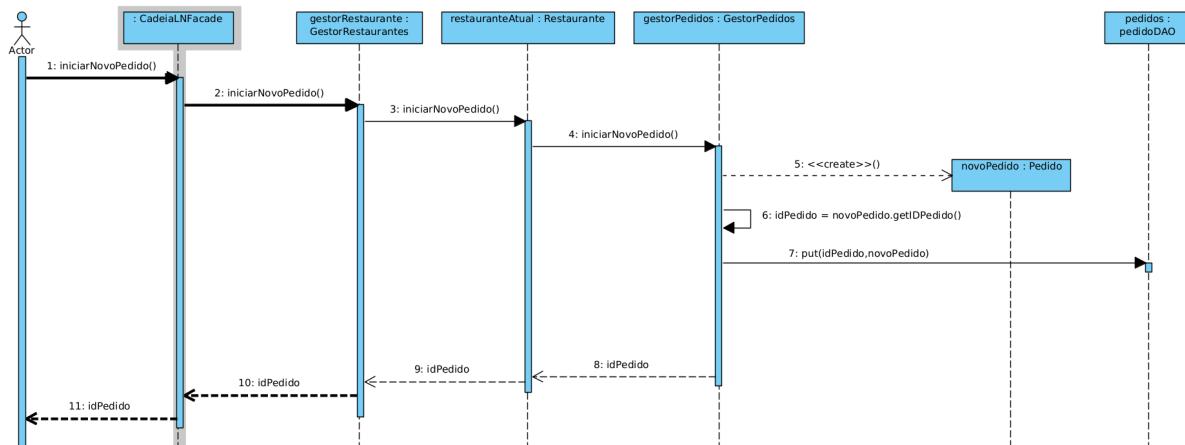


Figura 23: Diagrama de sequência: Iniciar novo pedido.

8.11. Listar pedidos com um determinado estado

Tal como referido anteriormente, os pedidos podem assumir diversos estados ao longo do seu ciclo de vida. Assim, considerou-se pertinente disponibilizar uma operação que permita obter todos os pedidos que se encontram num determinado estado, passado como argumento.

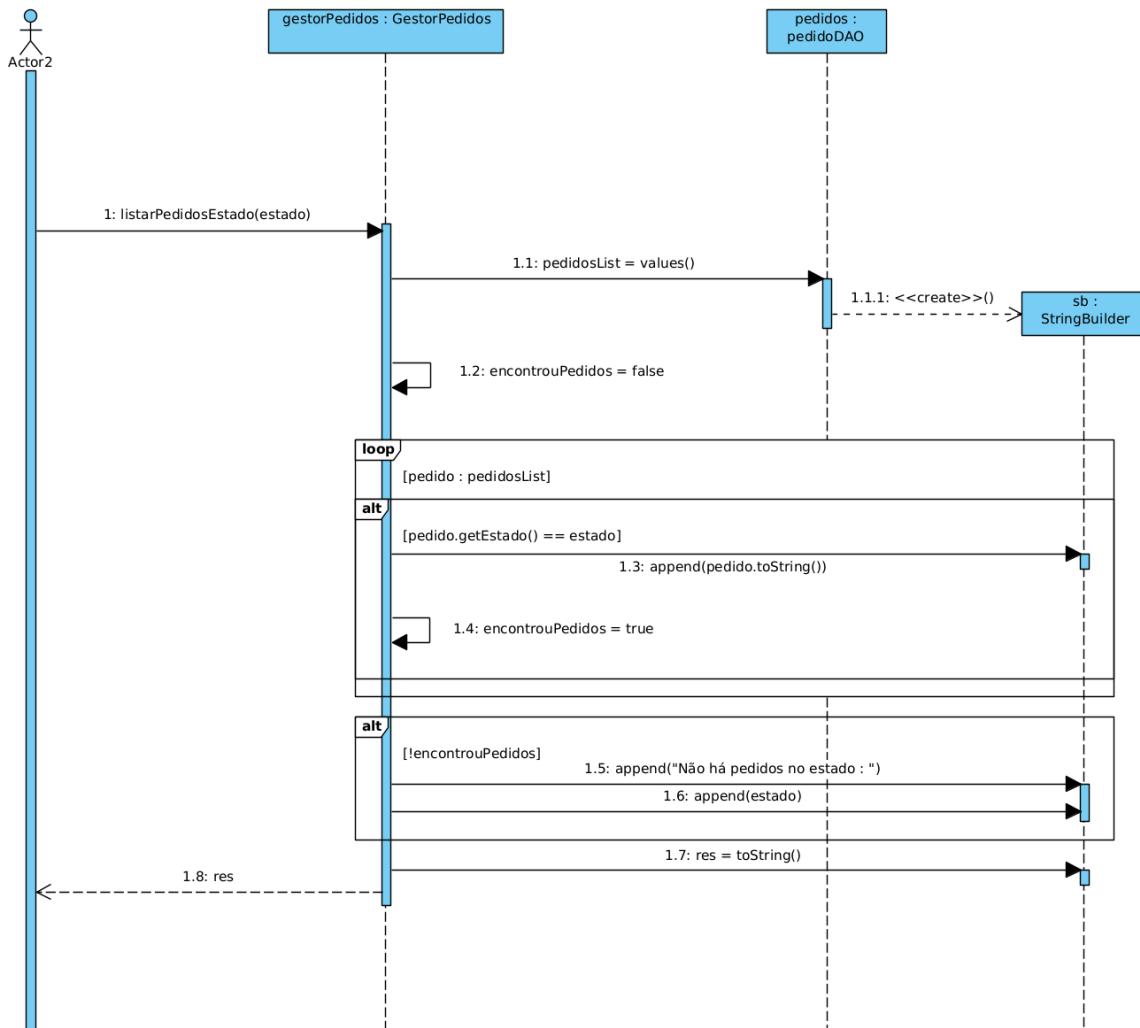


Figura 24: Diagrama de sequência: Listar pedidos por estado.

8.12. Listar pedidos prontos

Como exemplo de utilização da operação mencionada anteriormente, apresenta-se a forma como esta funciona no sistema. Esta funcionalidade é utilizada pelo entregador, sendo este abstraído da necessidade de indicar explicitamente o estado do pedido. Assim, o sistema disponibiliza a operação `listarPedidosProntos()`, que internamente invoca a operação `listarPedidosEstado(PRONTO)`.

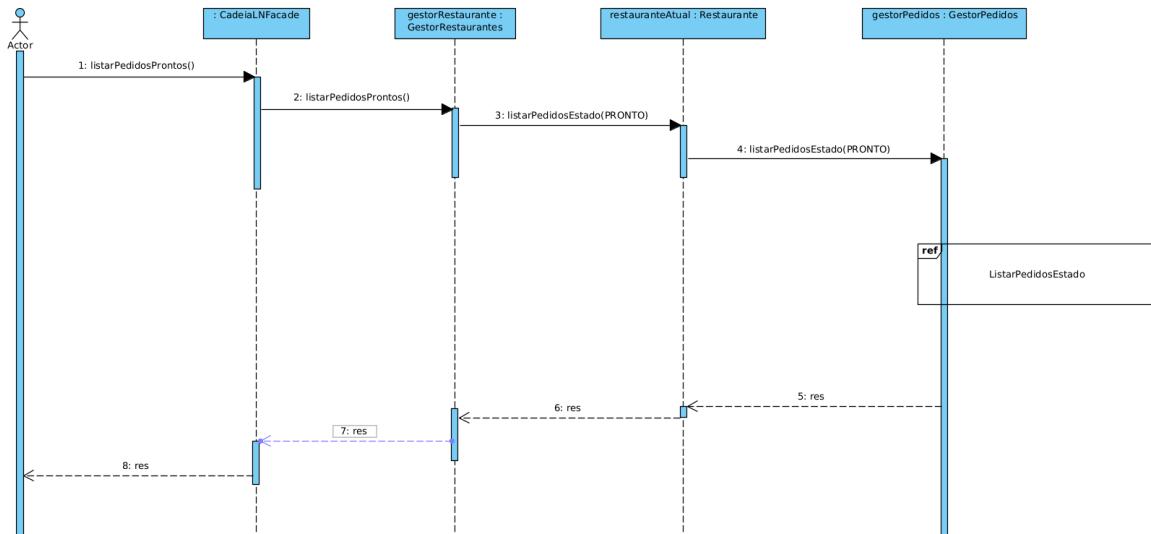


Figura 25: Diagrama de sequência: Listar pedidos prontos.

8.13. Listar as personalizações possíveis de uma proposta.

Cada proposta possui uma lista interna de ingredientes que podem ser removidos, correspondendo normalmente aos ingredientes que a constituem, bem como uma lista de ingredientes que podem ser adicionados. Esta restrição é importante, uma vez que permitir a adição de qualquer ingrediente a qualquer proposta poderia originar combinações incoerentes do ponto de vista do domínio do problema.

Assim, foi definida uma operação que, dado o nome de uma proposta, retorna todas as personalizações possíveis associadas à mesma.

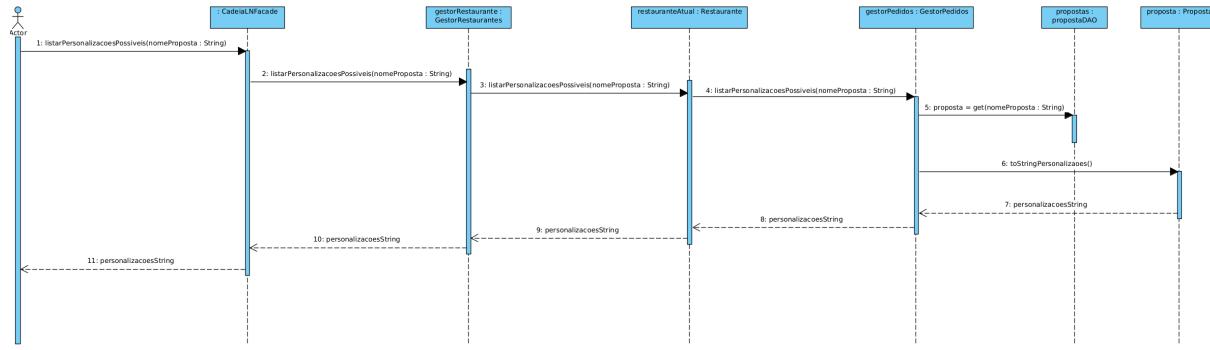


Figura 26: Diagrama de sequência: Listar personalizações possíveis.

8.14. Listar todos os restaurantes

Esta operação é de natureza trivial, uma vez que se limita a retornar a lista de todos os restaurantes registados no sistema.

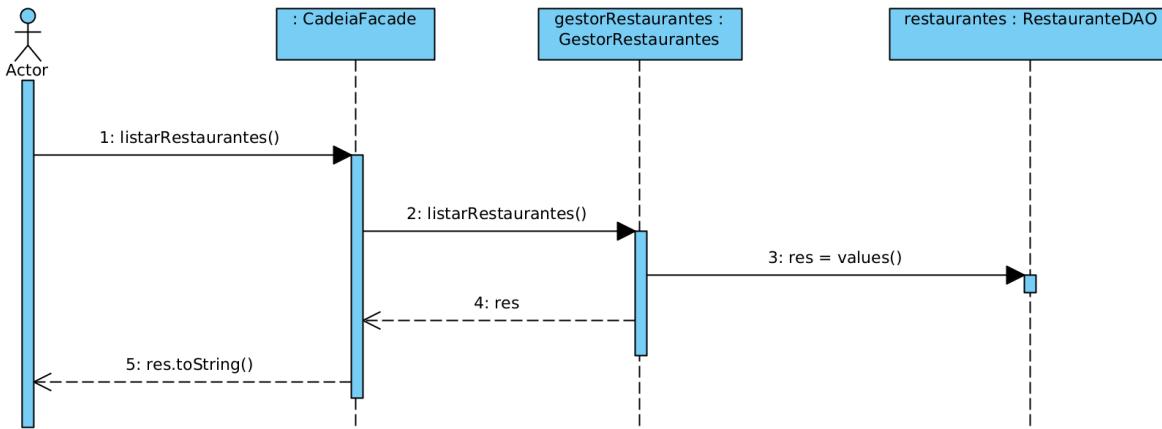


Figura 27: Diagrama de sequência: Listar restaurantes.

8.15. Listar todas as atarefas de um posto

Cada posto possui um conjunto de tarefas pendentes. Assim, é importante que o funcionário a ele associado consiga consultar essas tarefas de forma a poder realizá-las. Nesse sentido, foi definida uma operação que retorna a lista de todas as tarefas pendentes de um posto específico.

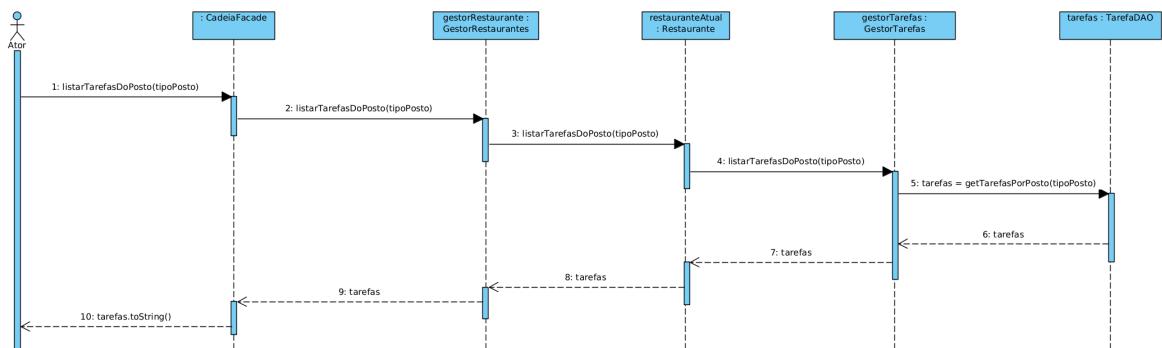


Figura 28: Diagrama de sequência: Listar tarefas de um posto de trabalho.

8.16. Marcar uma tarefa como concluída

Naturalmente, um posto necessita de uma operação que permita concluir uma tarefa. Assim, foi definida uma operação que, dado o identificador de uma tarefa, a marca como concluída. Após essa atualização, é recalculado o ETA do pedido associado, uma vez que a conclusão da tarefa pode influenciar o tempo estimado de conclusão do mesmo.

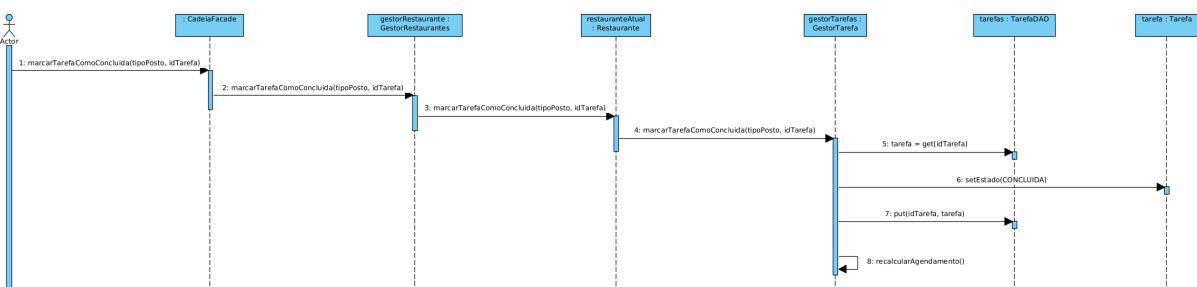


Figura 29: Diagrama de sequência: Marcar tarefa como concluída num posto.

8.17. Recalcular ETA de tarefas

Esta operação recalcula o agendamento das tarefas de forma sequencial por posto. Para cada tipo de posto, começa no instante atual e percorre a lista de tarefas associadas, ignorando as que já se encontram concluídas. Cada tarefa pendente é reagendada para começar no momento disponível (cursor), e o cursor é atualizado para o instante de fim dessa tarefa, garantindo que as tarefas seguintes têm os seus tempos estimados ajustados em função das anteriores.

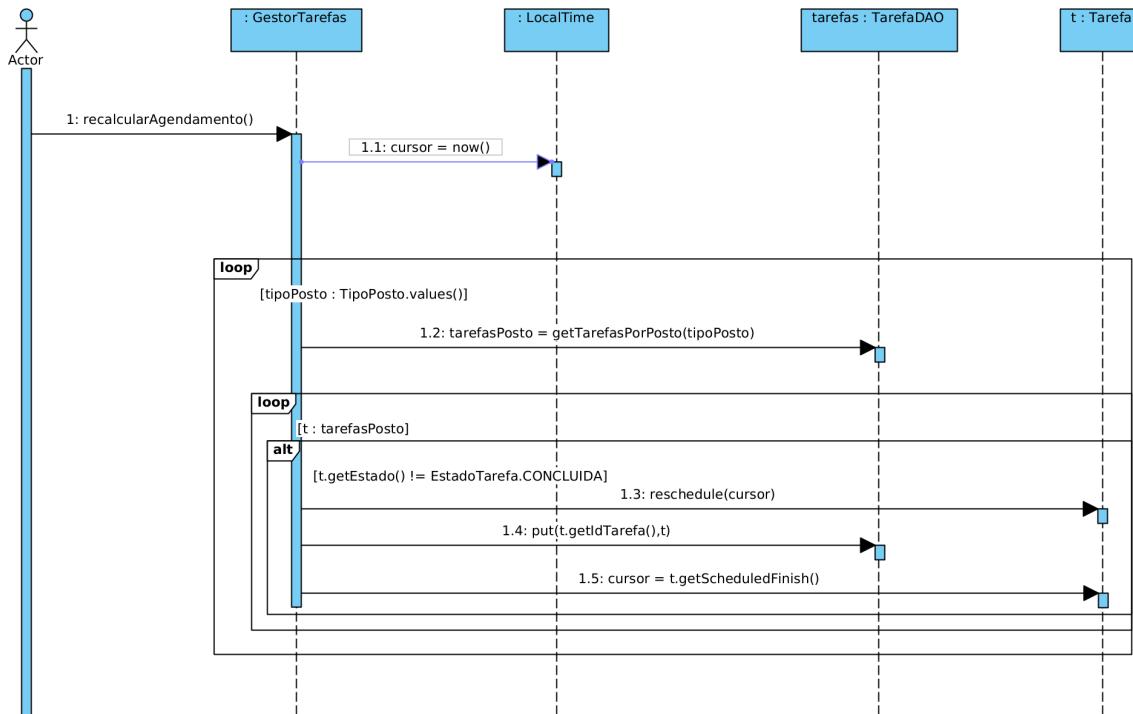


Figura 30: Diagrama de sequência: Recalcular agendamento de tarefas.

8.18. Calcular o tempo médio de atendimento de um restaurante

Esta operação tem como objetivo calcular o tempo médio de atendimento de um restaurante. Para tal, é efetuada uma iteração por todos os pedidos associados ao restaurante, calculando-se, para cada pedido, o intervalo de tempo entre a sua finalização pelo cliente e a sua entrega. Esses valores são acumulados num somatório e, no final, é retornado o valor obtido pela divisão do total acumulado pelo número de pedidos considerados.

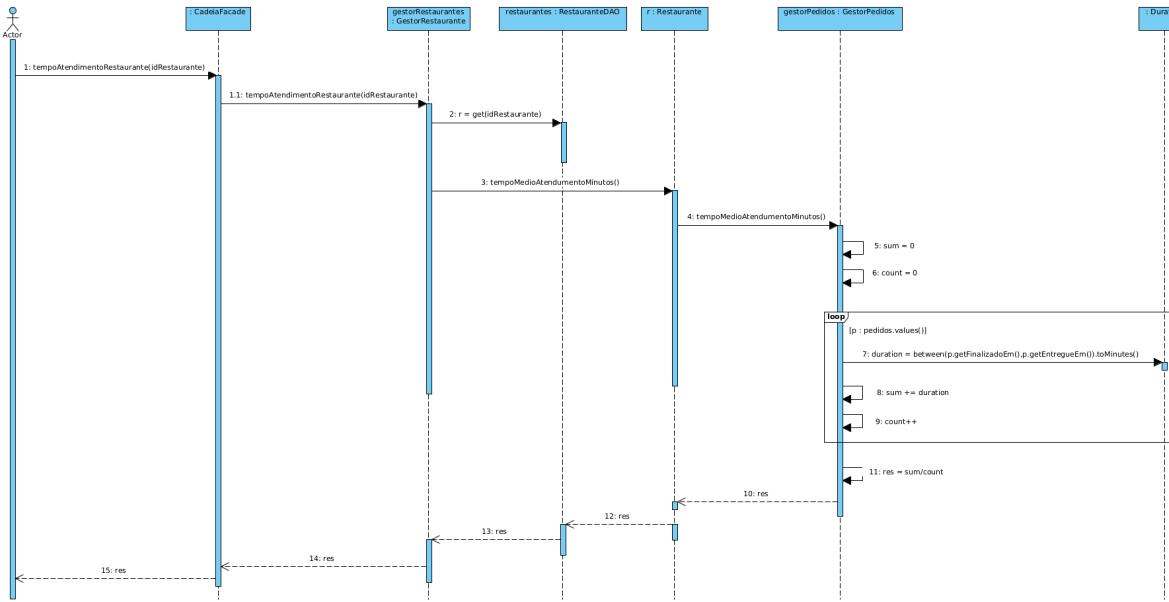


Figura 31: Diagrama de sequência: Calcular tempo médio de atendimento de um restaurante.

8.19. Calcular o tempo médio de atendimento da cadeia de restaurantes

Por fim, é importante que o COO tenha conhecimento do tempo médio de atendimento de todos os restaurantes da cadeia. Para tal, é efetuada uma iteração sobre todos os restaurantes, calculando-se o tempo médio de atendimento de cada um através da operação Seção 8.18. O resultado final corresponde à soma desses valores, fornecendo uma visão global do desempenho da cadeia.

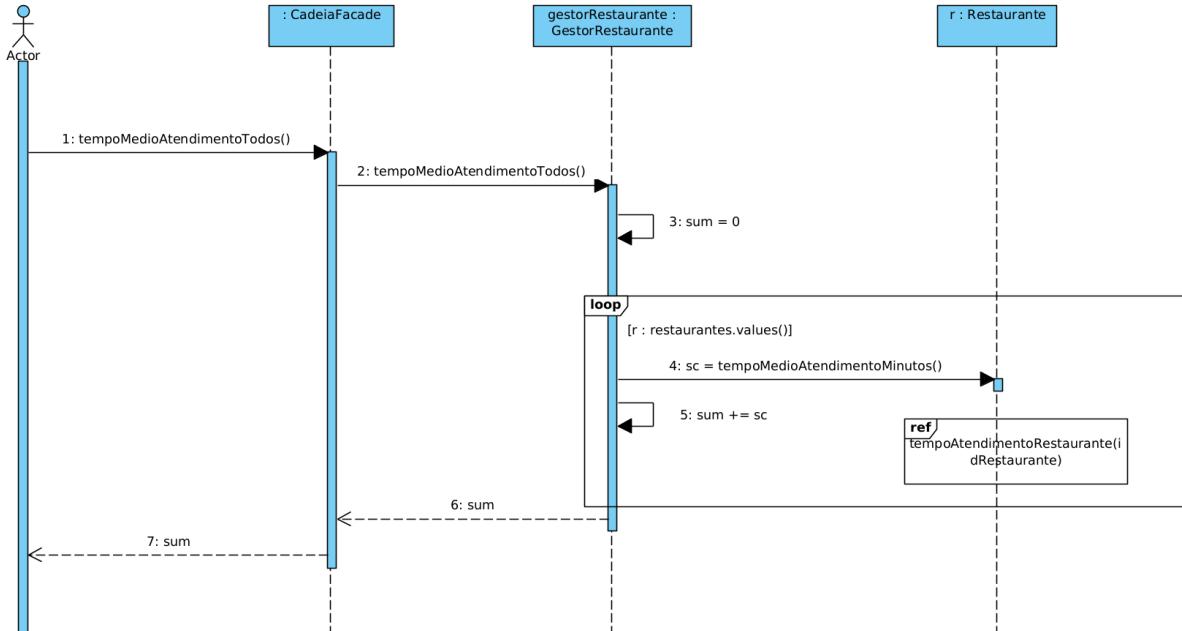


Figura 32: Diagrama de sequência: Calcular tempo médio de atendimento de todos os restaurantes.

9. Diagrama de Packages

O diagrama de packages permite visualizar a organização estrutural do sistema em módulos lógicos, evidenciando as dependências entre os diferentes componentes e promovendo uma

arquitetura modular e de fácil manutenção. Esta representação facilita a compreensão da separação de responsabilidades e da comunicação entre as diversas camadas do sistema.

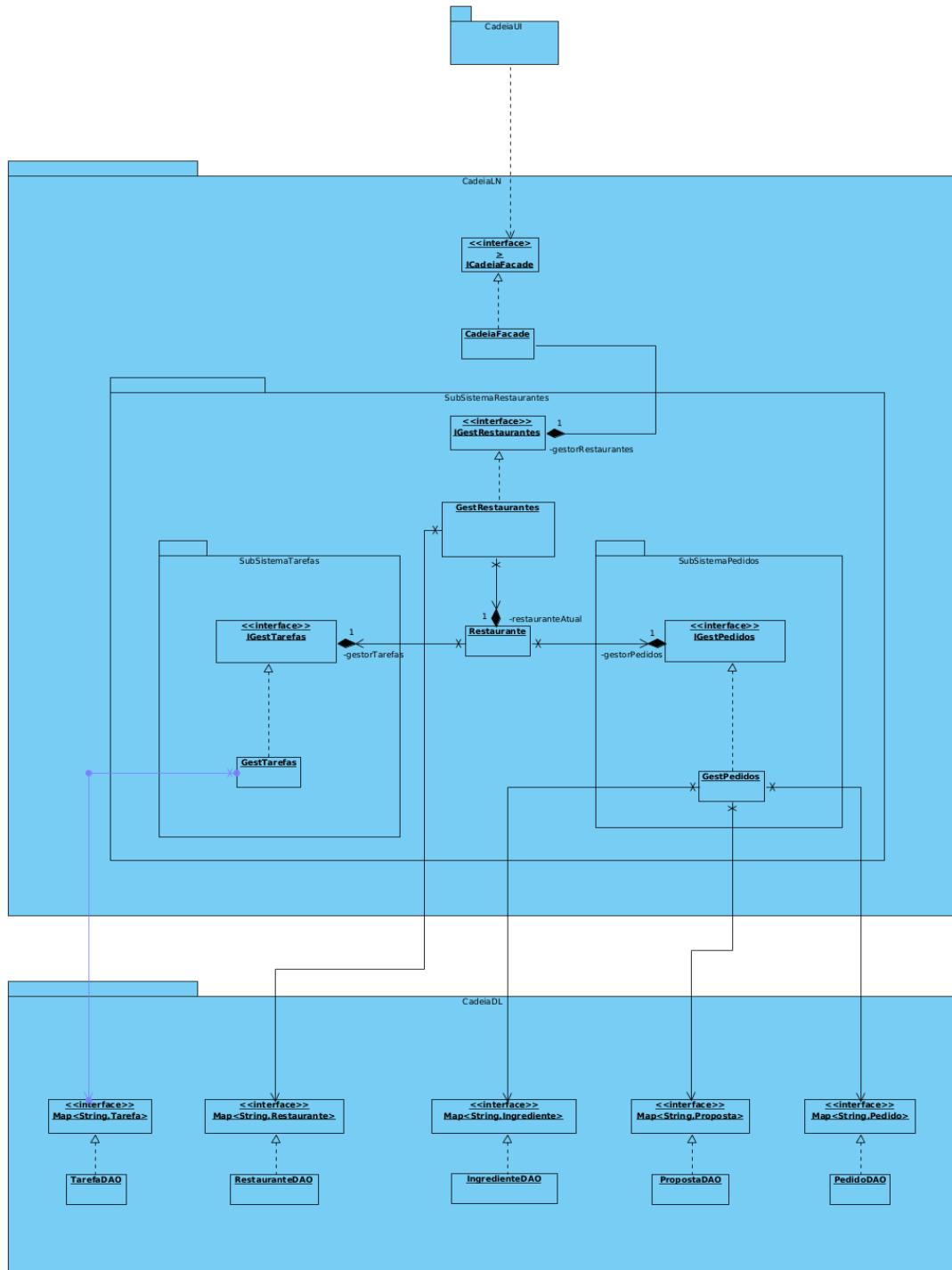


Figura 33: Diagrama de packages do sistema.

Com a conclusão deste diagrama, encerra-se a fase de modelação do sistema. Tendo definido todos os modelos considerados pertinentes, dispõe-se agora de uma base sólida e bem fundamentada para avançar para a fase de implementação do sistema.

10. Implementação e Interface

Graças à componente de modelação desenvolvida neste trabalho prático, tanto a nível conceptual como estrutural, a implementação do código revelou-se relativamente direta, exigindo apenas alguns ajustes mínimos.

Nesta seção pretende-se demonstrar os menus do sistema e as funcionalidades consideradas mais relevantes. Toda a estrutura do código foi implementada de acordo com a modelação previamente definida, garantindo a coerência entre o design e a execução do sistema.

10.1. Menu Principal

Quando um utilizador inicia o programa, é apresentado o menu principal, a partir do qual pode aceder aos diferentes menus disponíveis no sistema.

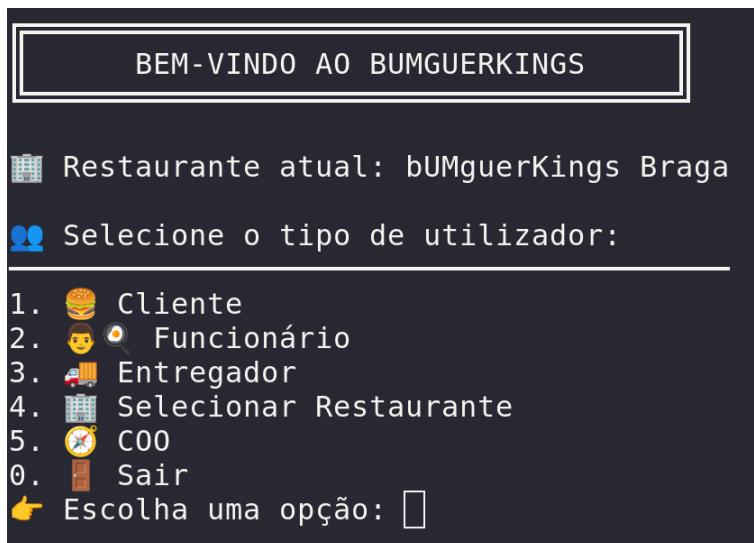


Figura 34: Menu Principal do sistema, onde o utilizador escolhe o menu pretendido.

10.2. Menu do Cliente

O menu do cliente disponibiliza duas operações principais: criar um novo pedido ou consultar todos os pedidos existentes no sistema.

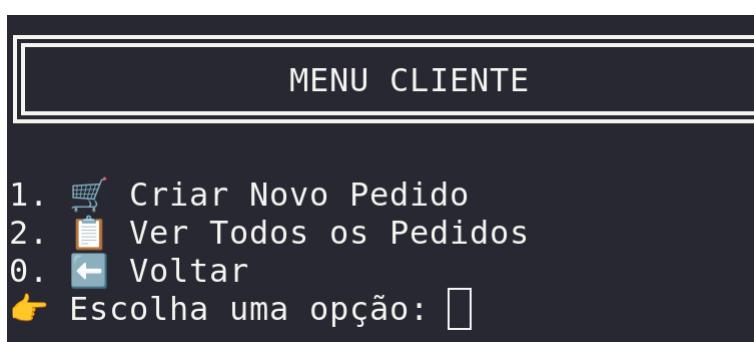


Figura 35: Menu do Cliente, permitindo criar um novo pedido ou consultar pedidos existentes.

Caso o cliente pretenda iniciar um novo pedido, este é criado no sistema. Em seguida, são apresentadas todas as propostas disponíveis no restaurante onde o cliente se encontra. Quando o cliente desejar finalizar o pedido, basta selecionar a opção correspondente.

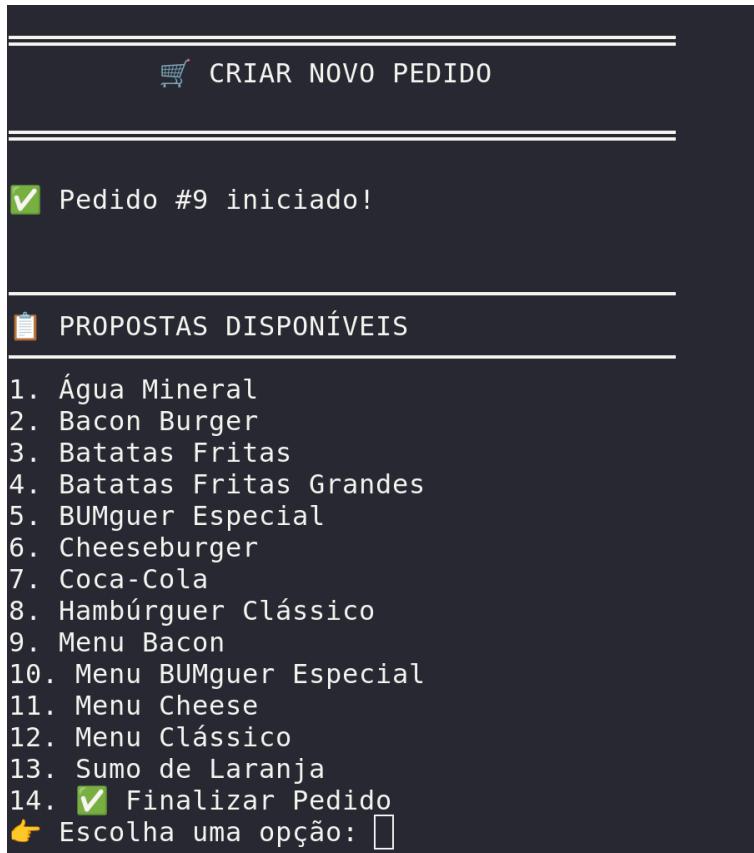


Figura 36: Fluxo de criação de um novo pedido pelo cliente, incluindo seleção de propostas.

Ao adicionar uma nova proposta, é questionado ao cliente se pretende personalizá-la. Caso a resposta seja afirmativa, são disponibilizadas todas as opções de personalização possíveis para essa proposta. No caso de o cliente escolher um menu, cada item pode ser personalizado individualmente.

```

👉 Escolha uma opção: 5
⚡ Deseja personalizar 'BUMguer Especial'? (s/n): s
🕒 Produto: BUMguer Especial
—————
— Ingredientes removíveis:
  Alface, Bacon, Carne de Vaca, Carne de Vaca, Cebola, Molho Especial, Pão de Hambúrguer, Pickles, Queijo Cheddar, Tomate
+ Ingredientes adicionáveis:
  Bacon, Carne de Vaca, Ketchup, Maionese, Mostarda, Queijo Cheddar
—————
— Ingredientes a remover (separados por vírgula, ou ENTER para nenhum): □

```

Figura 37: Personalização de uma proposta ou menu, permitindo ajustes individuais por item.

Quando o cliente estiver satisfeito com o pedido, pode finalizá-lo. O sistema pergunta então se o cliente deseja adicionar uma nota ao pedido, que poderá ser consultada pelos funcionários dos postos durante a preparação.

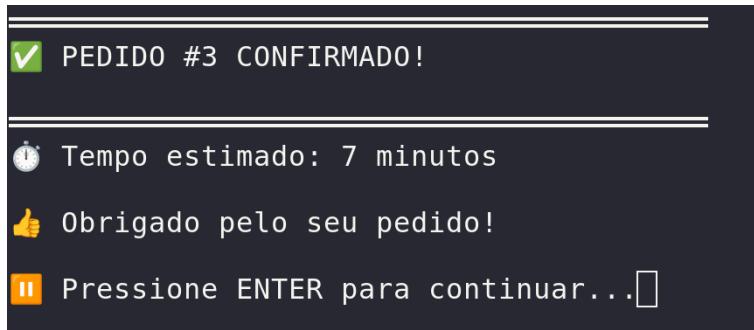


Figura 38: Finalização de pedido pelo cliente, incluindo opção de adicionar nota para os funcionários.

Por fim, o cliente também pode consultar todos os pedidos do sistema, visualizando o preço de cada um e o seu estado atual.

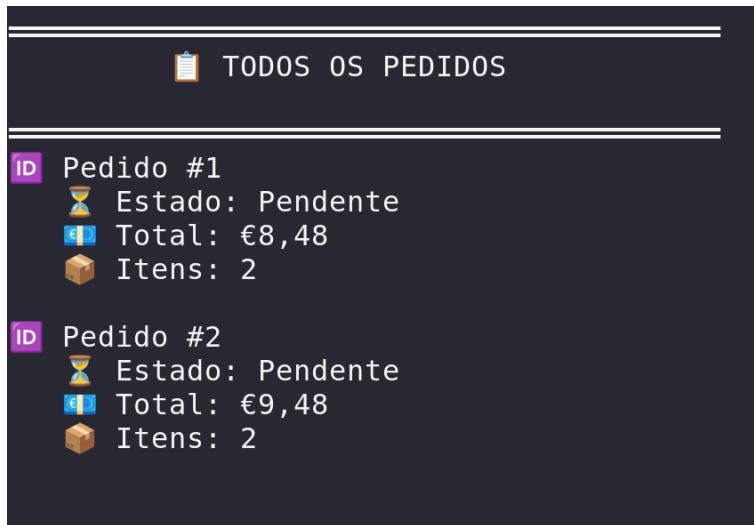


Figura 39: Visualização de todos os pedidos do cliente, incluindo preço e estado.

10.3. Menu do funcionário

Existem vários postos no sistema. Assim, ao aceder ao menu do funcionário, é solicitado que seja selecionado o posto correspondente.

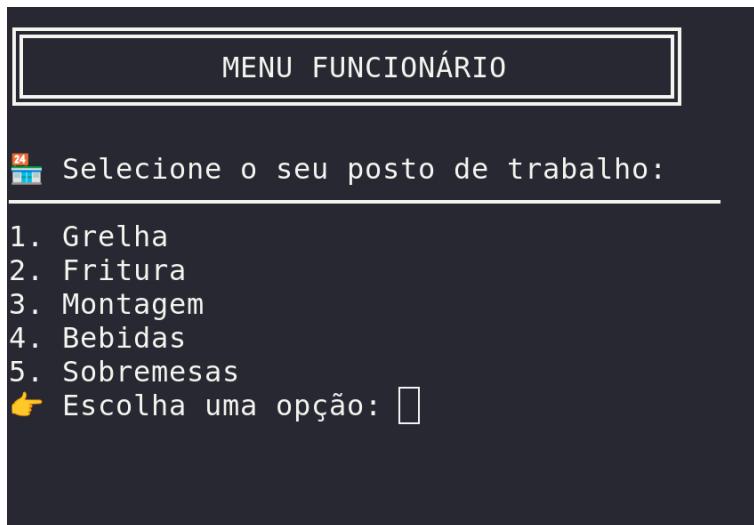


Figura 40: Seleção do posto pelo funcionário antes de aceder ao menu específico.

Ao selecionar o seu posto, são apresentadas diversas operações ao funcionário: consultar tarefas pendentes, concluir uma tarefa ou adicionar atraso a uma tarefa.

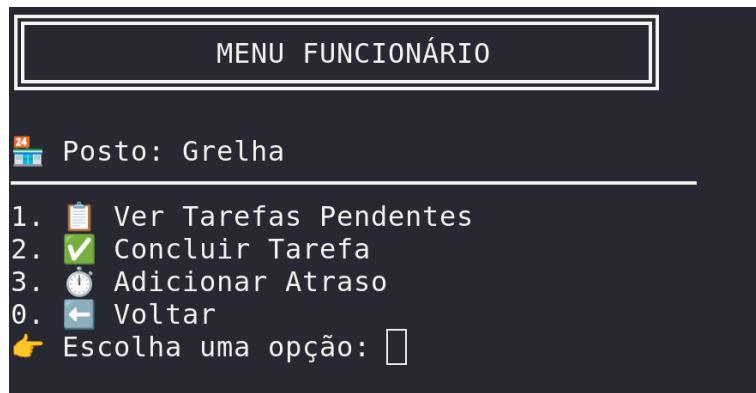


Figura 41: Menu do funcionário, com as opções principais para gestão das tarefas do posto.

Caso o funcionário pretenda apenas consultar as tarefas, o sistema apresenta, para cada tarefa do posto, o identificador da tarefa, o pedido associado, a descrição da ação a realizar, o estado da tarefa e o tempo estimado restante para conclusão. Caso o pedido possua uma nota ou personalizações, estas informações também são exibidas.

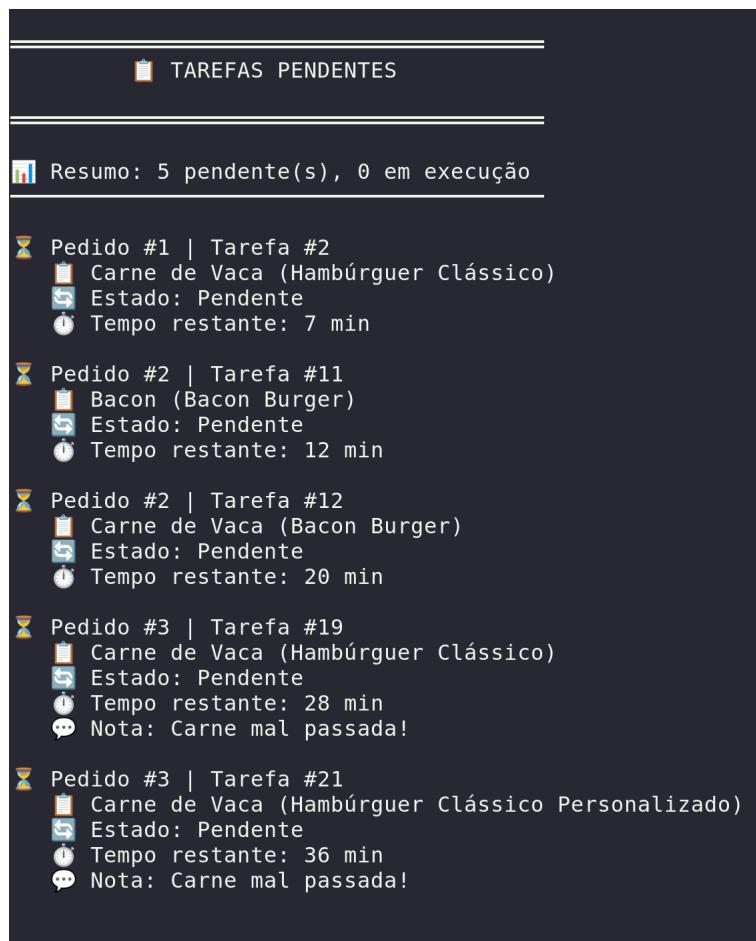


Figura 42: Visualização das tarefas pendentes do posto, incluindo informações detalhadas de cada tarefa.

Para concluir uma tarefa, o funcionário apenas precisa de fornecer o identificador da mesma. A tarefa é então marcada como CONCLUÍDA. Se todas as tarefas do pedido estiverem concluídas, o estado do pedido é atualizado para PRONTO.

```
Resumo: 5 pendente(s), 0 em execução

🕒 Pedido #1 | Tarefa #2
  🍔 Carne de Vaca (Hambúrguer Clássico)
  🚧 Estado: Pendente
  ⏳ Tempo restante: 7 min

🕒 Pedido #2 | Tarefa #11
  🍔 Bacon (Bacon Burger)
  🚧 Estado: Pendente
  ⏳ Tempo restante: 12 min

🕒 Pedido #2 | Tarefa #12
  🍔 Carne de Vaca (Bacon Burger)
  🚧 Estado: Pendente
  ⏳ Tempo restante: 20 min

🕒 Pedido #3 | Tarefa #19
  🍔 Carne de Vaca (Hambúrguer Clássico)
  🚧 Estado: Pendente
  ⏳ Tempo restante: 28 min
  💬 Nota: Carne mal passada!

🕒 Pedido #3 | Tarefa #21
  🍔 Carne de Vaca (Hambúrguer Clássico Personalizado)
  🚧 Estado: Pendente
  ⏳ Tempo restante: 36 min
  💬 Nota: Carne mal passada!

👉 Digite o id da tarefa a concluir (ou ENTER para cancelar): 11
✓ Tarefa concluída com sucesso!
```

Figura 43: Conclusão de uma tarefa pelo funcionário, com atualização do estado do pedido quando todas as tarefas estão concluídas.

Caso seja necessário alterar o tempo estimado de uma tarefa, o funcionário seleciona a opção de adicionar atraso e define o número de minutos a acrescentar.

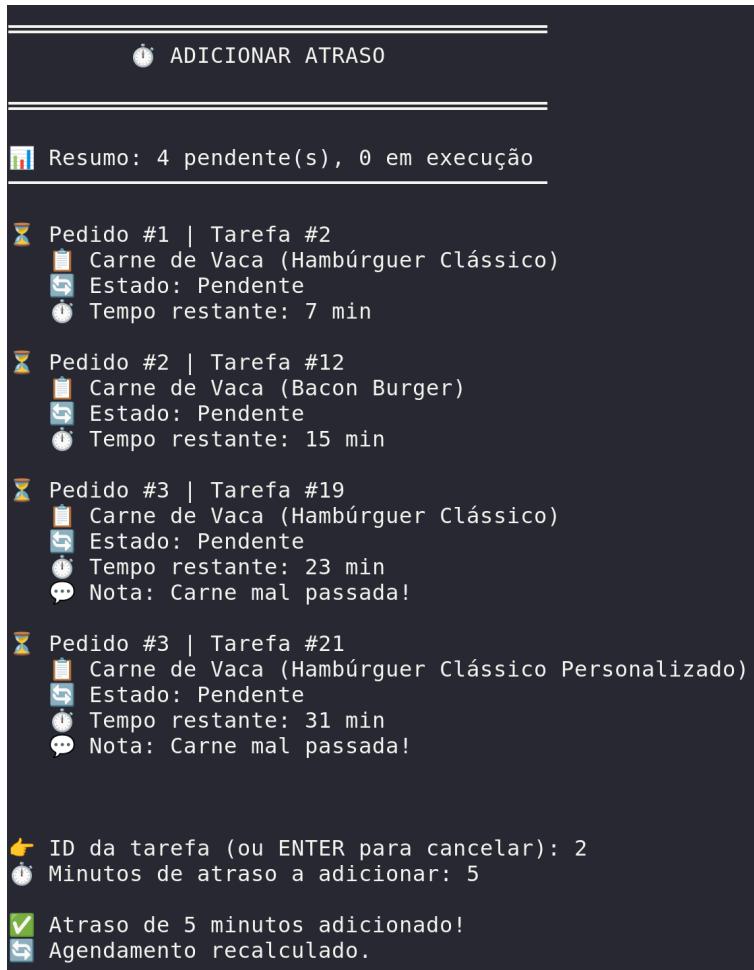


Figura 44: Adição de atraso a uma tarefa, permitindo ao funcionário ajustar o tempo estimado de conclusão.

10.4. Menu de Entregador

Quando todos os pedidos estão concluídos (ou seja, todas as tarefas associadas foram completadas), estes podem ser entregues. O menu do entregador permite realizar exatamente esta operação, possibilitando ao entregador consultar os pedidos prontos para entrega e efetuar a entrega de um pedido.

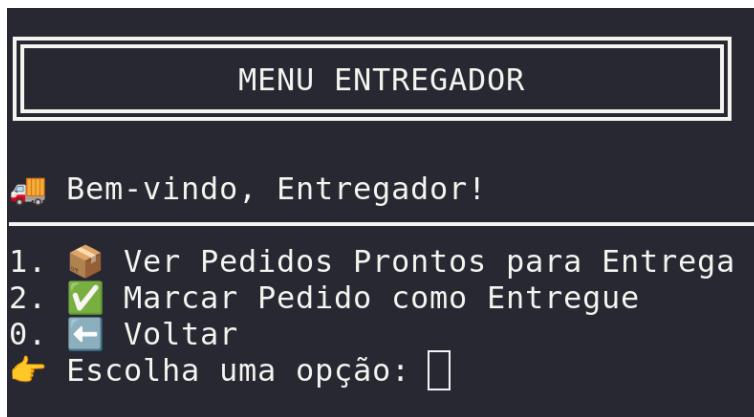


Figura 45: Menu do entregador, com opções para visualizar pedidos prontos e entregar pedidos.

Caso o entregador pretenda apenas consultar os pedidos prontos para entrega, o sistema apresenta uma lista contendo o identificador de cada pedido e o seu preço.

PEDIDOS PRONTOS PARA ENTREGA			
ID	Pedido #2		✓ Pronto € 9,48
ID	Pedido #3		✓ Pronto € 7,48
ID	Pedido #4		✓ Pronto € 6,99

Figura 46: Visualização de todos os pedidos prontos para entrega, incluindo o id do pedido e o preço.

Para efetuar a entrega de um pedido, o entregador apenas precisa de fornecer o identificador do pedido desejado. O sistema atualiza automaticamente o estado do pedido para refletir a entrega.

✓ MARCAR COMO ENTREGUE			
ID	Pedido #1		✓ Pronto € 8,48
ID	Pedido #2		✓ Pronto € 9,48
ID	Pedido #3		✓ Pronto € 7,48
ID	Pedido #4		✓ Pronto € 6,99

👉 Dige o ID do pedido a entregar (ou ENTER para cancelar): 1
 ✓ Pedido #1 marcado como entregue!
 !! Pressione ENTER para continuar...

Figura 47: Entrega de um pedido pelo entregador, efetuada através da introdução do id do pedido.

10.5. Trocar de restaurante

Para simplificar a interação, caso o utilizador pretenda alterar o restaurante no qual se encontra, basta selecionar a opção Escolher Restaurante. O sistema apresenta então todos os restaurantes da cadeia disponíveis.

É importante notar que cada restaurante possui o seu próprio gestor de pedidos e tarefas, ou seja, os pedidos e tarefas são geridos de forma independente em cada restaurante. Consequentemente, os menus e as tarefas podem variar de restaurante para restaurante.

📋	Restaurantes disponíveis:
#1	- bUMquerKings Braga (Campus de Gualtar)
#2	- bUMquerKings Ponte de Lima (Praça da República)
👉	Escolha uma opção: □

Figura 48: Seleção de restaurante pelo utilizador, permitindo trocar o contexto de pedidos e tarefas entre diferentes restaurantes da cadeia.

10.6. Menu do COO

O utilizador tem a opção de aceder ao menu do COO, onde pode consultar indicadores agregados de todos os restaurantes ou de um restaurante específico.

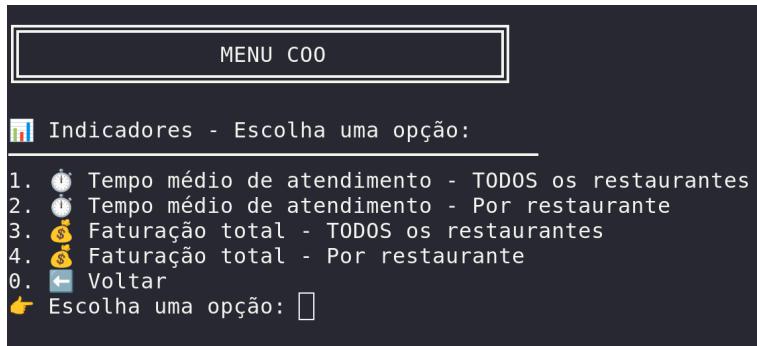


Figura 49: Menu do COO, permitindo aceder a indicadores agregados de todos os restaurantes ou de restaurantes individuais.

O COO pode então selecionar o indicador pretendido e, se aplicável, escolher o restaurante do qual deseja visualizar os dados.

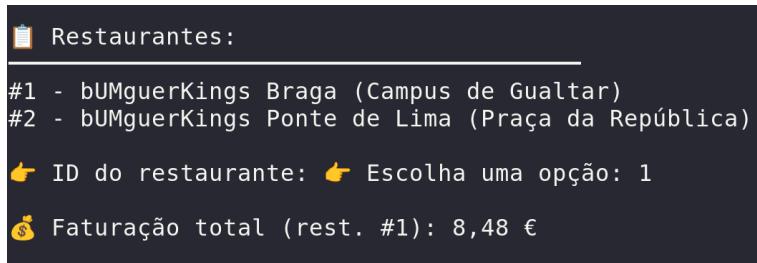


Figura 50: Visualização da faturação total de um restaurante selecionado, disponível no menu do COO.

11. Observações Finais

Ao longo deste trabalho, foi possível aplicar de forma prática os conceitos de modelação e desenvolvimento de sistemas de software estudados na unidade curricular.

A fase inicial de modelação de domínio revelou-se a mais desafiante, particularmente devido ao elevado nível de abstração necessário para capturar adequadamente os conceitos do problema. Apesar das dificuldades, consideramos ter alcançado um modelo de domínio robusto e coerente, que serviu de base sólida para todas as fases subsequentes do desenvolvimento.

Os casos de uso sofreram várias iterações e refinamentos ao longo do projeto, à medida que a compreensão do sistema se aprofundava. Este processo iterativo permitiu ajustar os requisitos e garantir uma maior precisão na especificação das funcionalidades.

Relativamente à arquitetura do sistema, a decisão inicial previa a criação de três subsistemas distintos: Restaurantes, Pedidos e Tarefas. Contudo, após análise mais aprofundada, concluímos que a solução mais adequada passaria por ter um subsistema de Restaurantes, onde cada restaurante possui os seus próprios subsistemas de Pedidos e Tarefas. Esta abordagem promove uma maior independência operacional entre restaurantes, refletindo de forma mais fiel a realidade do domínio.

Os diagramas de sequência revelaram-se relativamente simples de desenvolver, resultado direto da decisão de manter a simplicidade no design do código que pretendíamos implementar. Esta simplicidade intencional facilitou significativamente a compreensão do fluxo de execução do sistema.

Por fim, a implementação do código beneficiou enormemente da modelação rigorosa realizada nas fases anteriores. A transição dos modelos UML para código Java foi praticamente direta, demonstrando o valor de um trabalho de modelação bem estruturado e detalhado.

12. Enunciado do trabalho prático

Desenvolvimento de Sistemas de Software

Licenciatura em Engenharia Informática
Departamento de Informática
Universidade do Minho

2025/2026

Enunciado do Trabalho

António Nestor Ribeiro, Tiago Oliveira, Afonso Sousa

(disponibilizado em 26/09/2025)

Conteúdo

1	Introdução	1
2	Objectivo do trabalho	1
2.1	O pedido	2
2.2	Informação para a gestão	2
3	Realização do trabalho	2
3.1	Entrega intermédia	3
3.2	Entrega final	3
4	Apresentação e discussão do trabalho	4
5	Avaliação	4
6	Grupos de Trabalho	5

1 Introdução

Este documento apresenta o enunciado do trabalho prático da Unidade Curricular (UC) de Desenvolvimento de Sistemas Software para o ano lectivo 2025/2026. **Leia-o com atenção**, já que descreve, não só o sistema a desenvolver, como o processo que deve seguir para a realização do trabalho. Quaisquer dúvidas devem ser esclarecidas junto dos docentes da UC.

2 Objectivo do trabalho

Uma cadeia de restaurantes de fast-food pretende ter um sistema integrado que permita automatizar todos os aspectos do seu funcionamento nos restaurantes, desde os pedidos dos clientes até ao processo de elaboração dos pratos e à entrega dos mesmos aos clientes. Pretende-se também obter informação de funcionamento para a gestão da cadeia de restaurantes, no que concerne a volume de pedidos, stocks dos produtos, tempo médio de espera por uma refeição, entre outros.

Os restaurantes têm todos o mesmo modelo de funcionamento: os clientes escolhem o que pretendem consumir em ecrãs tácteis que estão à entrada do espaço e após finalizarem de compor o pedido, este é passado para a produção e empurramento (ou embalamento no caso de ser para take away). A confecção da refeição passa por diversos estágios e tem diversos empregados que assumem funções diferentes: um grelha a carne, outro frita as batatas, outro trata dos ovos, outro dos legumes, etc. Nem todos os pratos precisam de passar por todos estes estágios e é objectivo do sistema a construir que as tarefas estejam devidamente organizadas para que as refeições não fiquem com alimentos à espera da confeção de outros. Por exemplo, não faz sentido para um pedido de um prego no pão com ovo, ter o bife grelhado e o pão aquecido, mas ficar à espera de que o ovo seja estrelado.

Cada posto de um funcionário terá um display em que sabe o que tem de fazer e tem uma visão dos pedidos que se seguem e de algumas notas que tenham sido enviadas pelo cliente na altura em que fez o pedido. Poderá também reordenar os pedidos caso exista algo imprevisto (ex: falta momentânea de um ingrediente que terá de ser trazido do armazém), sendo que tal poderá ter impacto nas tarefas de outros funcionários.

Existem também funcionários que estão encarregados de efectuar a entrega dos pratos aos clientes que fazem a refeição dentro do restaurante ou então de embalar os pratos para os recipientes de take-away. Em ambas as situações, todos os componentes de um pedido deverão estar confeccionados na altura da entrega ao cliente.

2.1 O pedido

Quando o cliente faz o pedido nos ecrãs que existem para o efeito no restaurante, escolhe de uma série de propostas previamente existentes. Algumas dessas propostas admitem que se possam acrescentar ou retirar ingredientes. Em função dos ingredientes que são adicionados, é possível que surjam opções relativas à sua confecção e seja necessário perguntar ao cliente. Algumas das propostas aparecem sob a forma de um menu já composto. Mesmo nessas circunstâncias é possível escolher várias opções para os diferentes elementos do menu (por exemplo, na bebida podemos ter a opção de água, refrigerante, limonada, etc.).

Alguns alimentos podem ter a indicação de alergénios e o cliente pode decidir que não pretende ter algum deles ou a sua totalidade. Nessas circunstâncias a aplicação deve validar as propostas de refeição e os menus que ainda é possível oferecer sem que esses alergénios estejam presentes. Poderá também ser necessário alterar o processo de confecção, no sentido de pedir aos postos que não incluam os referidos alergénios na execução do pedido.

2.2 Informação para a gestão

O objectivo da construção de sistema integrado atrás referido deriva do facto de que a gestão pretende ter indicadores sobre os pedidos que os clientes fazem, sobre os produtos/refeições mais vendidos e poder controlar as necessidades de stock de produtos e eventuais necessidades de encomendar esses mesmos produtos. Pretende-se também ter indicadores sobre o tempo médio de atendimento dos pedidos e sobre as funções mais requisitadas na elaboração dos pedidos, para eventualmente abrir mais postos daquele tipo. O sistema deverá ter um conjunto de funcionalidades que permita à gestão da cadeia de restaurante obter este tipo de informação.

Para esclarecimento de eventuais dúvidas, contacte os docentes de DSS.

3 Realização do trabalho

A concepção e desenvolvimento da aplicação deverá seguir uma abordagem baseada em modelos (suportada por UML), de acordo com o processo de entregas faseadas descrito nas aulas teóricas. A aplicação deverá ser desenvolvida utilizando uma arquitectura multi-camada e tecnologias orientadas a objectos (preferencialmente, Java). Irá ser criado um repositório no GitHub¹ para cada grupo, onde deverá ser mantida a versão actualizada do trabalho.

¹<https://github.com>

Para facilitar o processo de concepção e desenvolvimento, o trabalho será realizado em duas fases.

3.1 Entrega intermédia

Análise de requisitos – a entregar até às 23h59 de 19 de outubro.

Objectivos:

- Um Modelo de Domínio com as entidades relevantes
- Um Modelo de Use Case (diagramas mais especificações do Use Case) com as funcionalidades propostas para o sistema

O resultado desta fase será sujeito a avaliação qualitativa.

3.2 Entrega final

Modelação conceptual e implementação da solução – a entregar até às 23h59 de 9 de janeiro.

Objectivos:

- Uma arquitectura conceptual do sistema, capaz de suportar os requisitos identificados.
- Os modelos comportamentais necessários para descrever o comportamento pretendido para o sistema
- Os modelos que considere necessários à descrição da implementação do sistema
- A implementação do sistema
- Documento técnico com todos os modelos desenvolvidos (em PDF).

Pretende-se que o documento técnico sirva de apoio à análise do trabalho, pelo que **deverá ter a seguinte estrutura**:

- **Capa com identificação** da Unidade Curricular, **do grupo (com fotos dos elementos)** e o URL do **repositório do trabalho**.
- Descrição dos resultados obtidos (máximo uma página).
- Diagramas relativos à **análise de requisitos** (Modelação de Domínio, Diagramas de Casos de Uso e correspondentes descrições dos casos de uso).

- Diagramas relativos à **modelação conceptual da solução** proposta (Diagramas de Classe e de Sequência).
- Diagramas com a descrição da **solução efectivamente implementada** (Diagramas de Classe, de Sequência, de Componentes e de packages).
- Manual de utilização do sistema desenvolvido.
- Em anexo, este enunciado.

Os diagramas mencionados acima podem ser complementados com outros que considerem relevante incluir.

4 Apresentação e discussão do trabalho

Para a apresentação do trabalho deverão preparar uma apresentação com a duração máxima de 15 minutos. Esta apresentação deverá descrever a solução e a abordagem seguida para a atingir, desde a análise dos cenários até a implementação e demonstração da solução final. A apresentação deverá terminar com uma análise crítica dos resultados obtidos.

Após essa apresentação, seguir-se-á um período de análise e discussão do trabalho de até 30 minutos.

5 Avaliação

A apresentação e discussão final do trabalho será realizada na semana de 12 de janeiro de 2026, em horários a combinar. A **presença** na discussão do trabalho é **obrigatória**.

Os pesos relativos de cada componente do trabalho serão os seguintes:

- Modelo de domínio e análise de requisitos: 25%
- Modelação conceptual: 25%
- Modelação final e implementação: 35%
- Apresentação e discussão: 15%

A nota de cada elemento do grupo será individual, tendo em consideração a nota do trabalho e a avaliação por pares. A equipa docente reserva-se a possibilidade de ajustar as notas, em função da sua avaliação de cada elemento durante a discussão do trabalho.

6 Grupos de Trabalho

Os grupos de trabalho deverão obrigatoriamente ser constituídos por de 3 a 5 elementos. A definição dos grupos de trabalho será realizada no Blackboard, **terminando a 8 de outubro**.

Bibliografia

- [1] M. Blaha e J. Rumbaugh, *Object-Oriented Modeling and Design with UML*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 2005.