



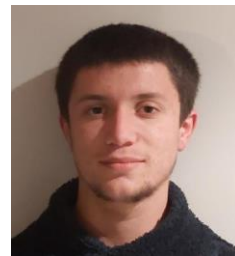
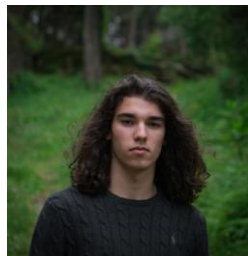
Universidade do Minho
Escola de Engenharia

Desenvolvimento de Sistemas de Software

Departamento de Informática

Trabalho Prático - Fase 2

Grupo 60



A93212 - Afonso Franco Ferreira

A93247 - André Lucas Silva Verdelho

A93248 - João Tiago Fidalgo Alves de Sousa

A93184 - Diogo Camacho Barbosa

A93188 - Nelson Miranda Ribeiro

Índice

Introdução	3
Objetivos	3
Alterações da primeira fase	4
Modelo de Use Case	7
Especificações de Use Cases	8
Diagrama de Componentes	15
Diagramas de Classes	16
Diagramas de Sequência	21
Análise Crítica Global dos Resultados	23

Introdução

Para este projeto foi-nos pedido para conceber e, posteriormente, desenvolver e implementar um Sistema de Gestão para Centros de Reparação de Equipamentos Eletrónicos.

O projeto foi realizado ao longo deste semestre e foi evoluindo à medida que expandíamos o nosso conhecimento sobre Desenvolvimento de Sistemas de Software.

Este Sistema de Gestão foi desenvolvido faseadamente: numa parte inicial, visualizamos o problema em questão; de seguida, estruturamos uma solução e refinamos a mesma; por fim, depois desta análise inicial, procedemos a implementar o código, acelerando esta última etapa uma vez que não era necessário estruturar o código à medida que o íamos escrevendo.

Deste modo foi possível apresentar uma solução substancialmente melhor do que a inicialmente planeada e implementar a mesma com uma rapidez sem precedentes.

Na primeira fase do projeto desenvolvemos o Modelo de Domínio e o Modelo de Use Cases, contudo, estes foram alterados na segunda fase e serão detalhadamente justificadas estas alterações mais à frente.

Por fim, queríamos reforçar que fizemos os possíveis para atingir e respeitar os requisitos pedidos e tentamos desenvolver o trabalho da forma mais realista alcançável.

Objetivos

Para este projeto tínhamos como principal objetivo garantir a gestão de todo o processo da reparação de equipamentos desde a criação de um orçamento para a reparação até à entrega do equipamento. Para além dos requisitos definidos no enunciado do projeto, pretendemos também desenvolver um sistema bem encapsulado, modular e expansível, sendo moldável para se ajustar a mudanças de requisitos futuras e fácil de se desenvolver.

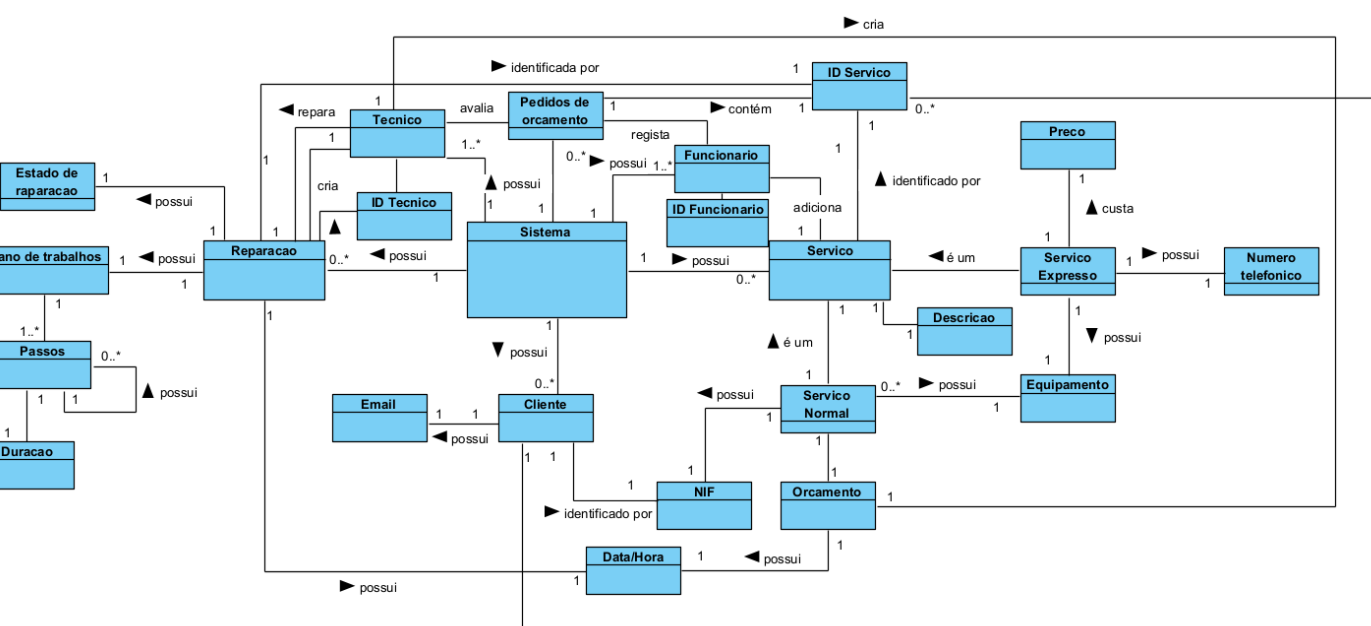
Alterações da primeira fase

Como já referimos, à medida que íamos ganhando uma perspetiva nova em relação aos conceitos de planeamento de sistemas de software, tivemos que realizar alterações ao Modelo de Domínio que inicialmente tínhamos desenvolvido.

Este modelo encontrava-se bastante incompleto e não estava corretamente organizado, deste modo, expandimos o modelo e obtivemos uma perspetiva muito mais específica e completa do sistema a desenvolver.

O modelo de Use Cases também sofreu algumas alterações, contudo, foram apenas melhorias e ajustes na qualidade das explicações lógicas dos mesmos e outros use cases foram removidos na sua totalidade.

Alteramos também o use case “Registar Detalhes de Reparação” , dividindo este em 3 Use Cases diferentes, mais representativos e específicos e também mais coerentes com o sistema.



Depois de reanalisarmos todos os cenários, chegamos a um novo consenso, tendo alcançado este modelo de Domínio de acordo com a nossa interpretação dos requisitos propostos e possíveis futuras expansões aos mesmos.

Começando pelo Sistema, que é a peça central deste modelo, observamos que este possui 6 associações: Serviço, Reparação, Cliente, Funcionário, Técnico e Pedidos de Orçamento. Um Serviço é identificado por um ID Serviço, pode ser Expresso ou Normal e possui uma Descrição.

O Serviço Expresso tem um Preço fixo que depende do serviço prestado, possui um Número Telefónico para poder ser contactado quando a reparação estiver concluída e também contém um Equipamento a ser reparado. O Serviço Normal contém o NIF do cliente, um Orçamento feito pelo Técnico que pode ou não ser aceite até uma certa data – Data/Hora – e também possui um equipamento a ser reparado.

O Sistema contém também uma listagem com todas as reparações identificadas pelo ID Serviço e com toda a informação em relação ao Estado de Reparação e o Plano de Trabalhos para a mesma.

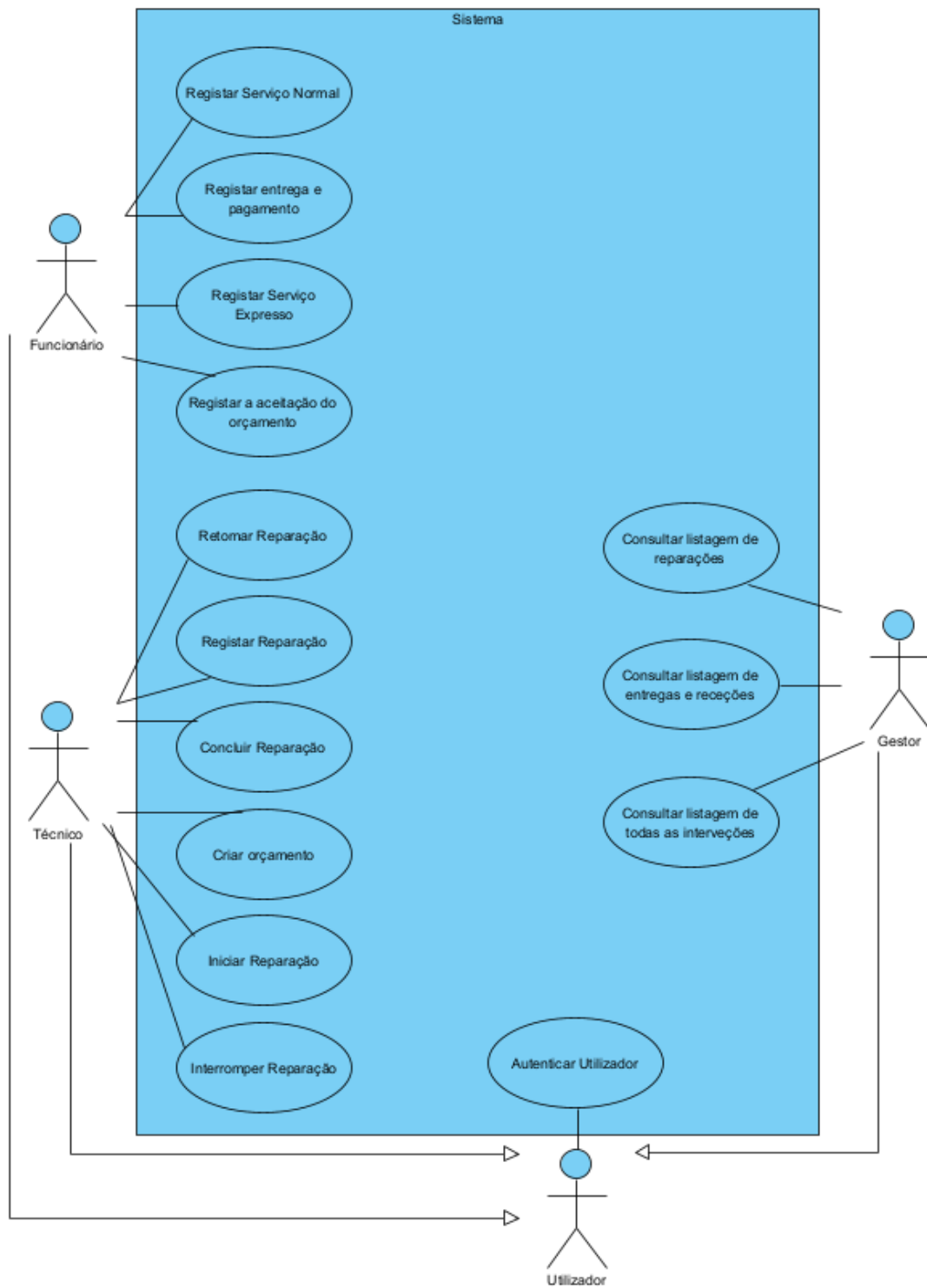
Uma Reparação dispõe de um Plano de Trabalhos que por sua vez contém uma lista de Passos que poderão ter Sub-Passos, e cada um destes tem um Custo e Duração a eles associado.

O Sistema possui também a listagem com todos os Clientes que possuem um Email e são identificados pelo seu NIF e o seu pedido é registado através do ID Serviço.

Os Pedidos de Orçamento são avaliados por um Técnico e registados por um Funcionário e contêm um ID Serviço e são correspondentes a uma Reparação.

Os Técnicos serão identificados por um ID Técnico e serão responsáveis por fazer as reparações e avaliar os Pedidos de Orçamento. Por outro lado, os Funcionários registam os Pedidos de Orçamento e os Serviços e são identificados por um ID Funcionário.

Modelo de Use Case



Especificações de Use Cases

Apos uma revalidação dos requisitos e uma análise detalhada do trabalho a desenvolver, decidimos reformular os Use Cases estipulados na 1 Fase.

Autenticar Utilizador

- Descrição: Utilizador autentica-se
- Cenários: Cenários 1,2,3,4 e 5
- Ator: Utilizador
- Pré-condição: True
- Pós-condição: O Utilizador fica autenticado
- Fluxo Normal:
 1. Utilizador introduz nome e palavra-passe
 2. Sistema dá acesso ao Utilizador
- Fluxo de Exceção: [Nome ou palavra-passe inválidos] (Passo 2)
 - 2.1 Acesso negado ao Sistema

Registar pedido de orçamento

- Descrição: Funcionário regista orçamento de reparação
- Cenários: Cenário 1
- Ator: Funcionário
- Pré-condição: Funcionário está autenticado
- Pós-condição: O pedido de orçamento é registado no sistema e enviado ao técnico
- Fluxo Normal:
 1. Funcionário fornece NIF do cliente e data de entrega
 2. Sistema cria pedido de reparação
 3. Sistema regista pedido de orçamento
 4. Sistema envia pedido de orçamento ao técnico

Criar Orçamento

- Descrição: Técnico cria o orçamento para uma reparação para ser enviado para o cliente que o pediu
- Cenários: Cenário 1
- Ator: Técnico
- Pré-condição: Técnico está autenticado
- Pós-condição: O pedido de orçamento é registado no sistema e enviado ao técnico
- Fluxo Normal:
 1. Técnico cria e regista o orçamento no sistema
 2. Sistema envia email ao Cliente com o orçamento

Registar a aceitação do orçamento

- Descrição: Funcionário regista a aceitação do orçamento
- Cenários: Cenário 1
- Ator: Funcionário
- Pré-condição: Funcionário está autenticado
- Pós-condição: O orçamento fica registado no sistema como aceite
- Fluxo Normal:
 1. Sistema recebe confirmação do orçamento
 2. Funcionário regista que orçamento foi aceite
- Fluxo de exceção (1): [Cliente recusa orçamento] (passo 1)
 - 1.1 Sistema elimina pedido de reparação
 - 1.2 Sistema adiciona equipamento à lista de equipamentos para serem levantados
- Fluxo de exceção (2): [Cliente não confirma reparação] (passo 2)
 - 2.1 Sistema deteta que o orçamento não foi confirmado no prazo de 30 dias
 - 2.2 Sistema arquiva orçamento

Registar Entrega e Pagamento

- Descrição: Funcionário regista a entrega e pagamento do equipamento
- Cenários: Cenário 1
- Ator: Funcionário
- Pré-condição: Funcionário está autenticado
- Pós-condição: No sistema fica registado a entrega e pagamento do equipamento
- Fluxo Normal:
 1. Funcionário fornece NIF ao sistema
 2. Sistema procura pedido através do NIF
 3. Sistema mostra o pedido
 4. Sistema regista o pagamento do serviço
 5. Sistema regista a entrega do equipamento

Registar Serviço Expresso

- Descrição: Funcionário confirma disponibilidade e efetua o seu registo
- Cenário: Cenário 2
- Ator: Funcionário
- Pré-Condição: O funcionário está autenticado
- Pós-Condição: O sistema fica com um novo registo de reparação
- Fluxo Normal:
 1. Sistema confirma disponibilidade de serviço
 2. Sistema regista o serviço
 3. Sistema notifica cliente por SMS

Registar reparação

- Descrição: Técnico regista a reparação no sistema, incluindo o plano de trabalhos para a reparação
- Cenários: Cenário 3
- Ator: Técnico
- Pré-condição: Técnico está autenticado
- Pós-condição: A reparação fica registada no sistema.
- Fluxo Normal:
 1. Sistema fornece a lista de pedidos de orçamento.
 2. Sistema inicia o registo do plano de Trabalhos para a Reparação.
 3. Sistema adiciona os passos introduzidos.
 4. Técnico introduz o número de horas de trabalho estimadas e o custo das peças necessárias.
 5. Sistema cria o registo da reparação e envia o orçamento ao cliente
- Fluxo de exceção (1): [Equipamento não pode ser reparado] (passo 2)
 - 2.1 Sistema notifica cliente que o equipamento não pode ser reparado

Iniciar reparação

- Descrição: O Técnico inicia a reparação
- Cenário: Cenário 1 e 4
- Ator: Técnico
- Pré-condição: O técnico está autenticado
- Pós-condição: A reparação é iniciada
- Fluxo normal:
 1. Técnico acede à lista de reparações programadas
 2. Técnico escolhe o pedido mais antigo
 3. Técnico seleciona a opção de iniciar a reparação
 4. Sistema regista o início da reparação

Interromper Reparação

- Descrição: O Técnico interrompe a reparação, podendo ser retomada posteriormente
- Cenário: Cenário 4
- Ator: Técnico
- Pré-condição: O técnico está autenticado
- Pós-condição: A reparação é interrompida
- Fluxo normal:
 1. Técnico acede à reparação em que está a trabalhar
 2. Técnico seleciona a opção para interromper a reparação
 3. Sistema regista a interrupção da reparação

Concluir reparação

- Descrição: O Técnico conclui a reparação
- Cenário: Cenário 4
- Ator: Técnico
- Pré-condição: O técnico está autenticado
- Pós-condição: A reparação é concluída
- Fluxo normal:
 1. Técnico acede à reparação em que está a trabalhar
 2. Técnico seleciona a opção de concluir a reparação
 3. Sistema regista a conclusão da reparação

Retomar Reparação

- Descrição: O Técnico retoma a reparação previamente interrompida
- Cenário: Cenário 4
- Ator: Técnico
- Pré-condição: O técnico está autenticado
- Pós-condição: A reparação é retomada
- Fluxo normal:
 1. Técnico acede à reparação em que está a trabalhar
 2. Técnico seleciona a opção para retomar a reparação
 3. Sistema retoma a reparação

Consultar Listagem de Reparações

- Descrição: Gestor consulta a listagem de reparações
- Cenário: Cenário 5
- Ator: Gestor
- Pré-condição: O gestor está autenticado
- Pós-condição: Os dados são consultados pelo sistema
- Fluxo normal:
 1. O Gestor requisita a listagem de reparações feitas por cada técnico.
 2. Sistema fornece a listagem.

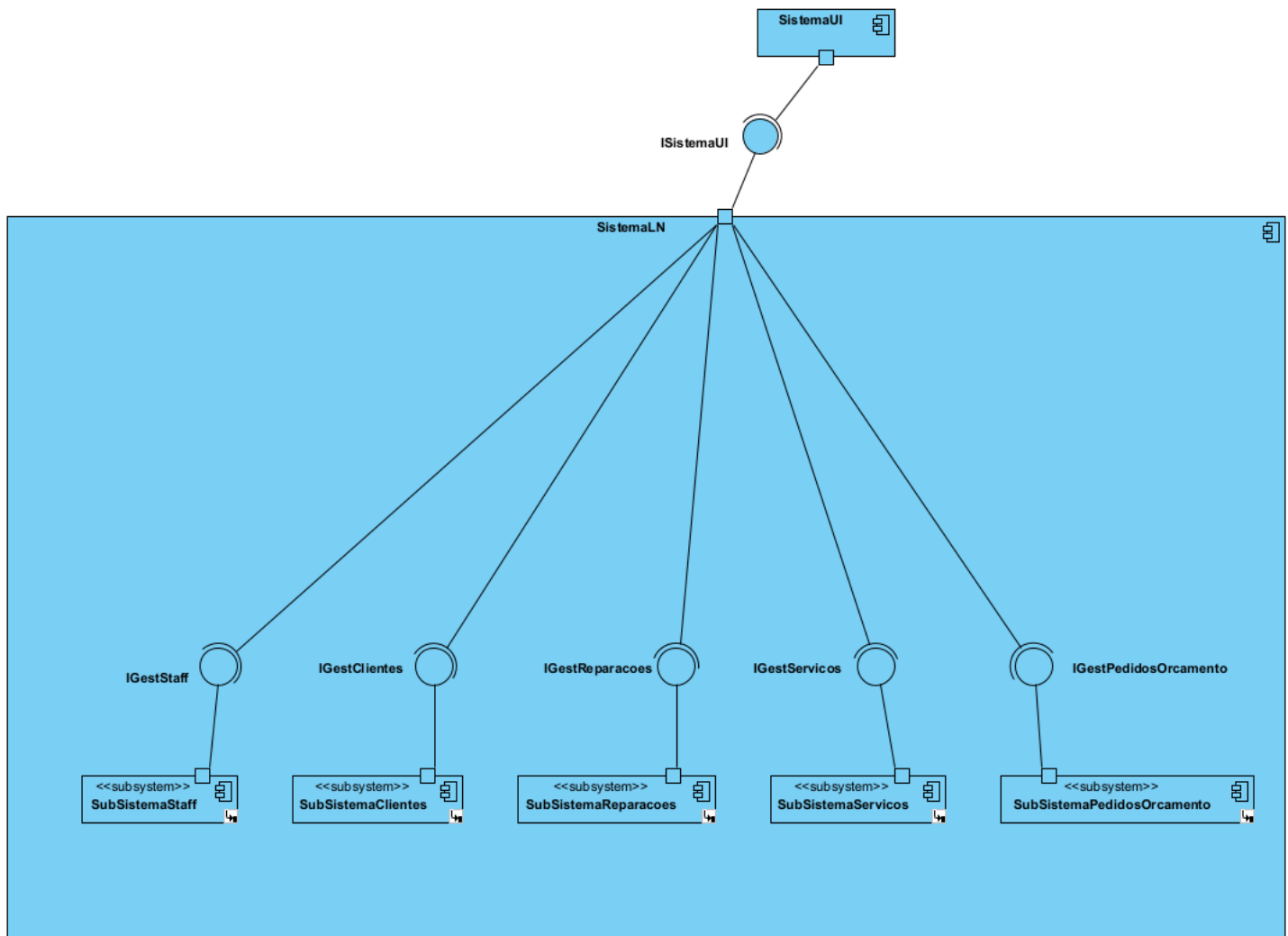
Consultar Listagem de Entregas e Receções

- Descrição: Gestor consulta a listagem de entregas e receções
- Cenário: Cenário 5
- Ator: Gestor
- Pré-condição: O gestor está autenticado
- Pós-condição: Os dados são consultados pelo sistema
- Fluxo normal:
 1. O Gestor requisita a listagem das receções e entregas de equipamentos feitas por cada funcionário.
 2. Sistema fornece a listagem.

Consultar Listagem de Todas as Intervenções

- Use case: Consultar listagem de todas as intervenções
- Descrição: Gestor consulta a listagem de todas as intervenções
- Cenário: Cenário 5
- Ator: Gestor
- Pré-condição: O gestor está autenticado
- Pós-condição: Os dados são consultados pelo sistema
- Fluxo normal:
 1. O Gestor requisita a listagem de todas as intervenções feitas por cada técnico.
 2. Sistema fornece a listagem.

Diagrama de Componentes



Através do Diagrama de Componentes podemos descrever os componentes do sistema e as diferenças que existem entre eles, identificando o que é necessário para construir o sistema.

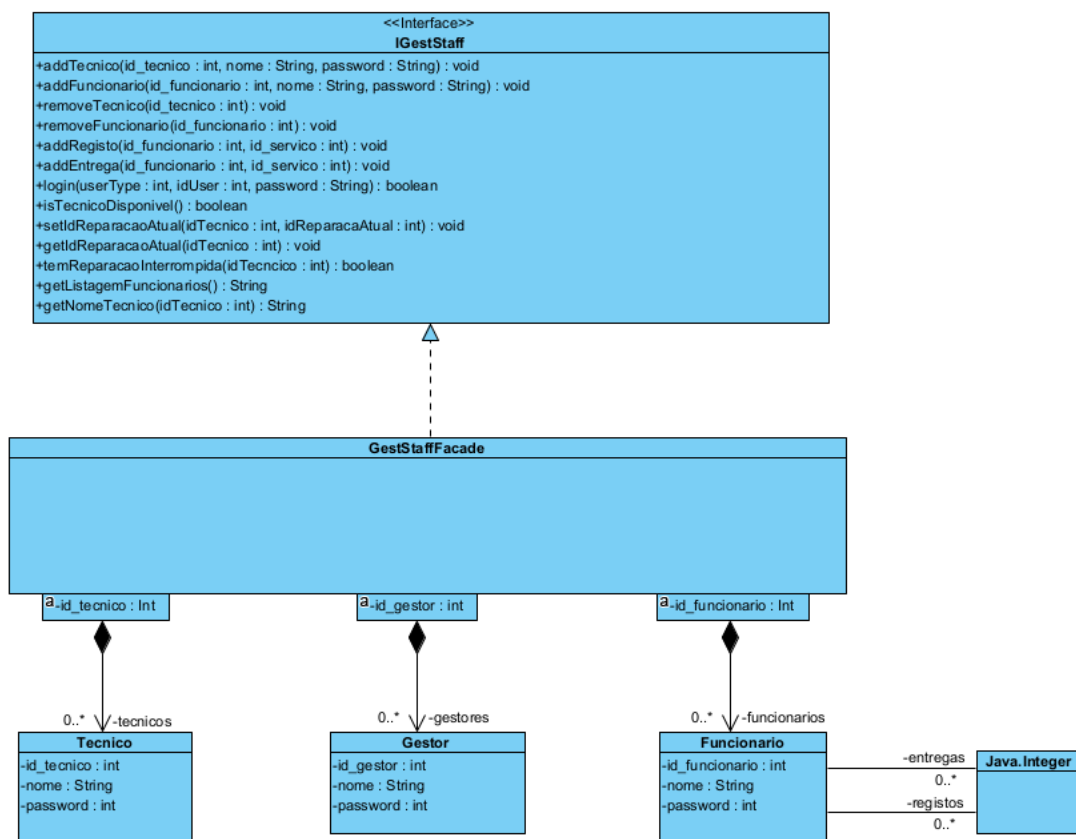
A criação de subsistemas permite uma melhor organização, uma vez que haverá o agrupamento dos métodos pelos subsistemas onde melhor se adequam. Trata-se de uma prática fundamental no que diz respeito ao encapsulamento, uma vez que cada subsistema implementa uma interface com os métodos que lhes foram atribuídos.

Deste modo, ficaram definidos os seguintes subsistemas: SubSistemaStaff, SubSistemaClientes, SubSistemaReparações, SubSistemaServicos e SubSistemaPedidosOrcamento, implementado as interfaces: IGestStaff, IGestClientes, IGestReparacoes, IGestServicos e IGestPedidosOrcamento, respetivamente.

Diagramas de Classes

Staff

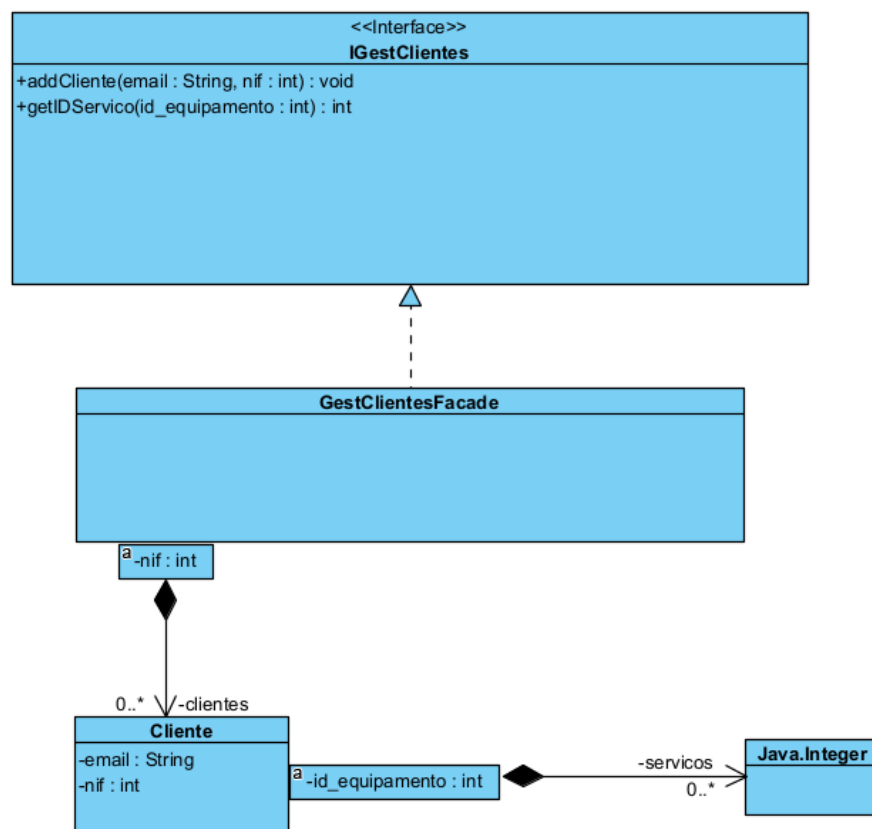
O diagrama de classes do subsistema Staff é constituído pela interface IGestStaff e pela classe GestStaffFacade que implementa os seus métodos. Existem também as classes Tecnico, Funcionario e Gestor, podendo a sua informação ser armazenada na classe GestStaffFacade através de *maps*, em que a sua chave seria o identificador do Staff correspondente. Contém também na classe Funcionario duas listas: uma correspondente aos identificadores dos serviços registados no sistema pelo funcionário em questão e a outra corresponde aos identificadores dos serviços que foram entregues pelo funcionário.



Cientes

Este diagrama de classes corresponde ao subsistema Clientes, que é constituído pela interface IGestClientes e pela classe GestClientesFacade que implementa os seus métodos.

No facade, existe uma estrutura map, denominada clientes, através do qual é possível aceder às informações de um cliente, através do seu NIF. Dentro da classe Cliente é possível encontrar outro map que contém todos os serviços que o cliente já requisitou do nosso sistema, cuja chave seria o identificador do equipamento do cliente, representando assim a ficha de um cliente.

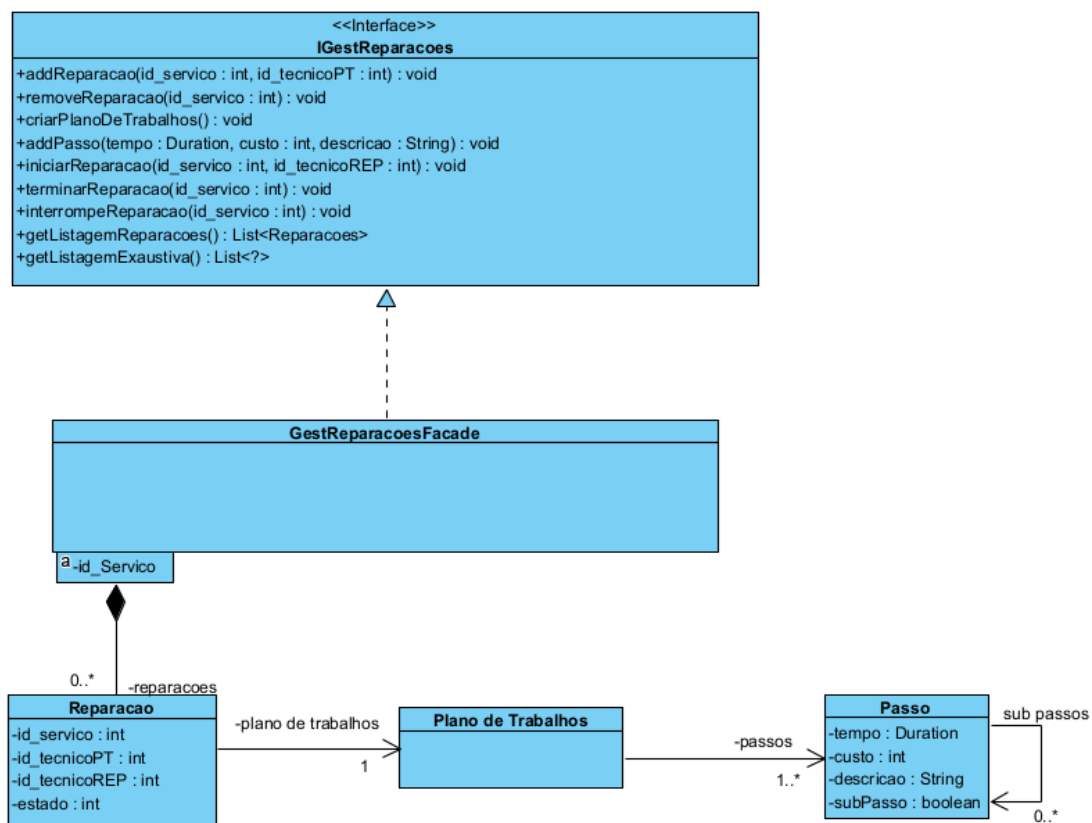


Reparações

O diagrama de classes do subsistema Reparações é constituído pela interface IGestReparacoes e pela classe GestReparacoesFacade que implementa os seus métodos.

No facade existe uma estrutura map, denominada reparações, através da qual é possível aceder às informações de uma reparação, identificadas pelo identificador de serviço. A classe Reparação contém o identificador do serviço, do técnico que fez o registo do plano de trabalho e do técnico que fez a reparação. Possui também um estado representado por um número inteiro e este pode ser: 0, quando a reparação não está completa, 1, quando o equipamento já se encontra reparado e -1, quando a reparação está interrompida. Para além destes quatro atributos já referidos, a classe Reparação possui um plano de trabalho, que por sua vez, contém uma lista de passos.

A classe Passo possui o tempo estimado para conclusão desse passo, o custo, a descrição e uma lista de passos. Essa lista de passos representa os sub-passos associados ao passo em questão.

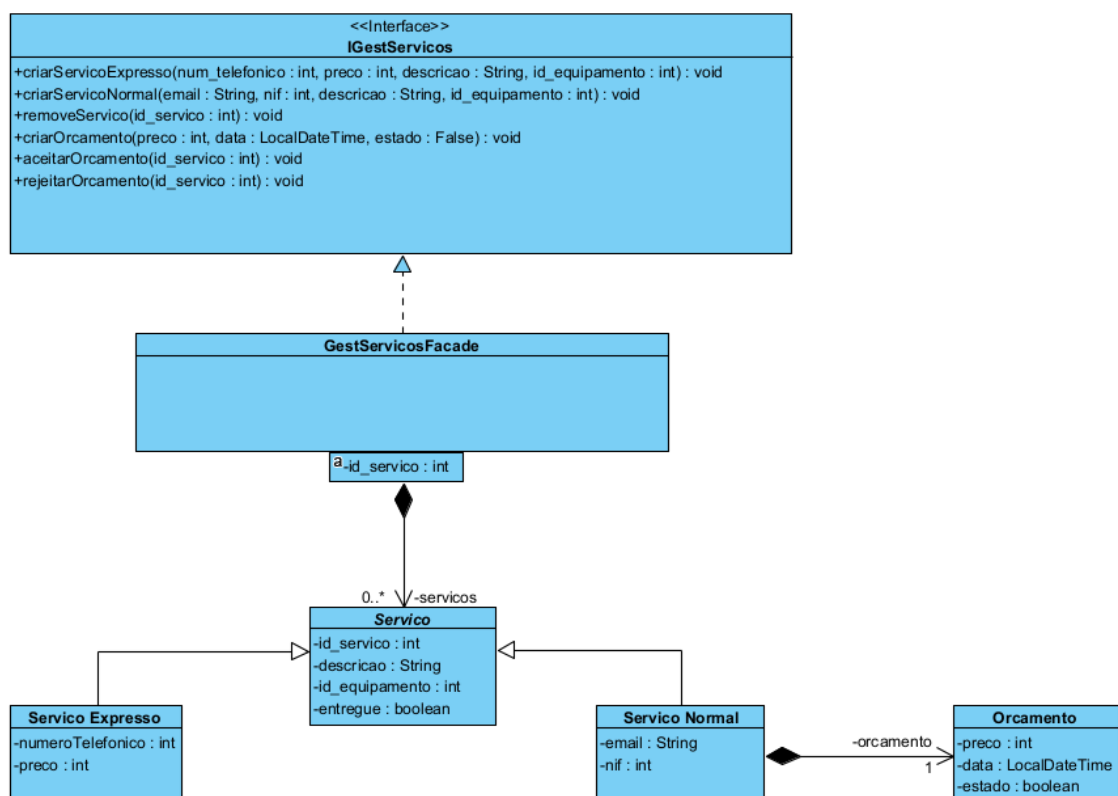


Serviços

O diagrama de classes do subsistema Serviços é constituído pela interface IGestServicos e pela classe GestServicosFacade que implementa os seus métodos.

No *facade* existe uma estrutura *map*, denominada *servicos*, através da qual é possível aceder às informações de um serviço, identificadas pelo identificador do serviço. A classe *Servico* contém o identificador do serviço, uma descrição, o identificador do equipamento a ser reparado e um booleano que representa se o equipamento já foi entregue.

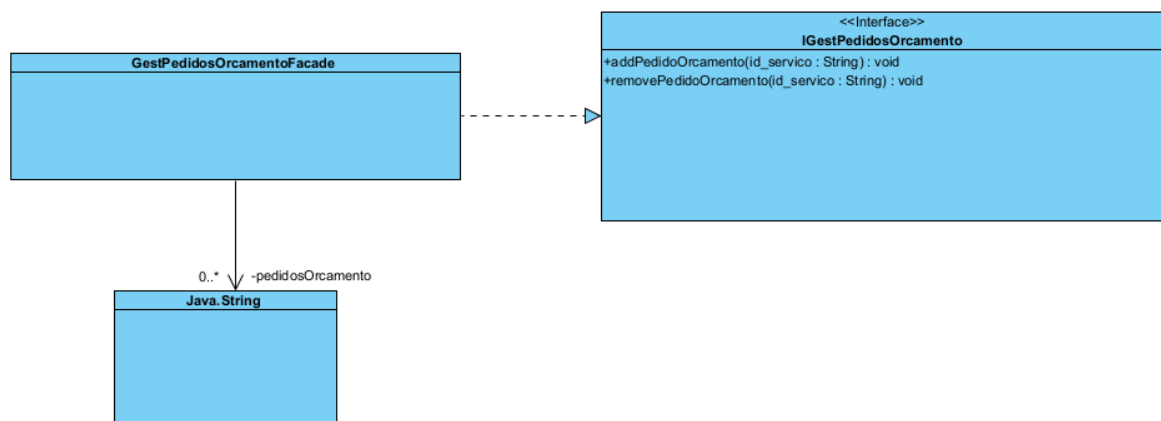
Uma vez que existem dois tipos de serviços disponíveis, tivemos de criar duas classes que dão *extend* à superclasse *Servico*. A subclasse *Servico Expresso*, contém o número telefónico do cliente e o preço já estipulado do serviço expresso em questão. A subclasse *Servico Normal* contém o email do cliente, o NIF do mesmo e um Orçamento.



Pedidos de Orçamento

O diagrama de classes do subsistema Pedidos de Orçamento é constituído pela interface `IGestPedidosOrcamento` e pela classe `GestPedidosOrcamentoFacade` que implementa os seus métodos.

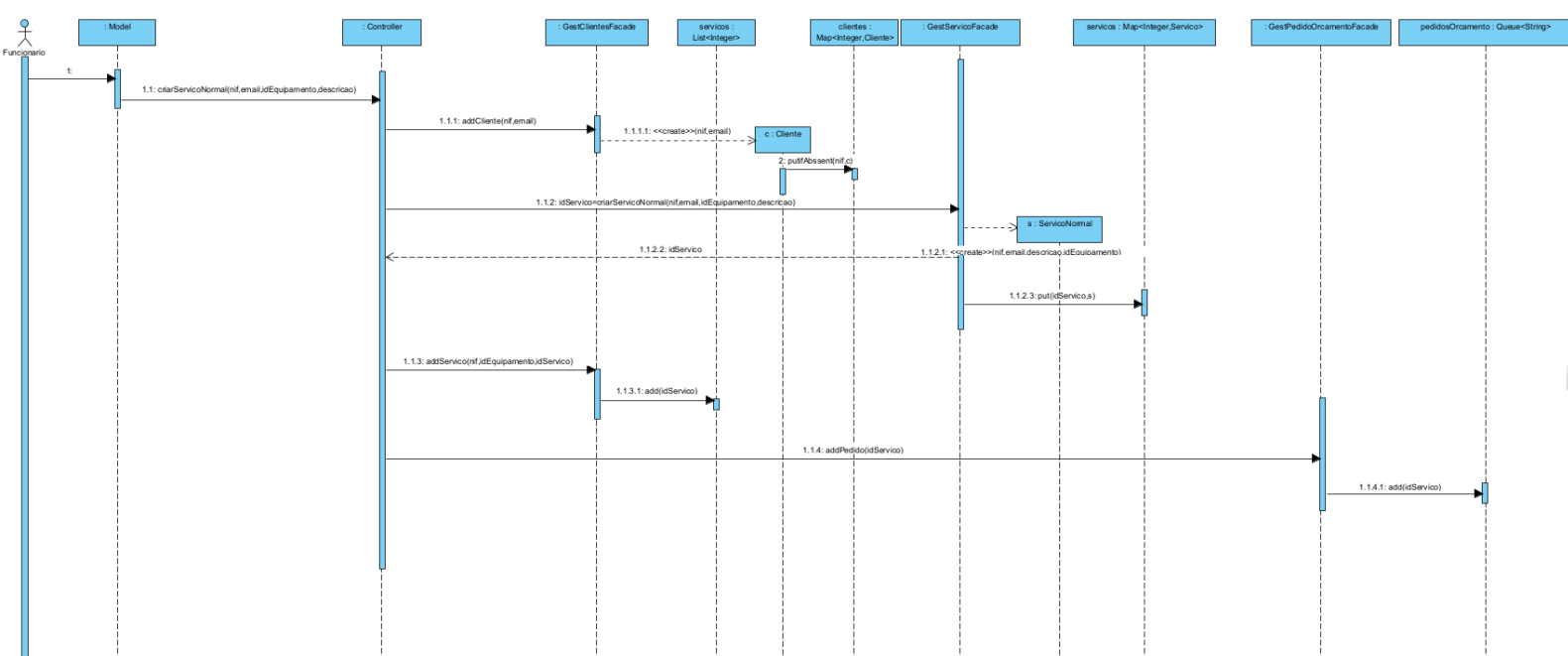
No *facade*, existe uma lista de pedidos de orçamento, cujos elementos são os identificadores dos serviços normais que fizeram um pedido de orçamento.



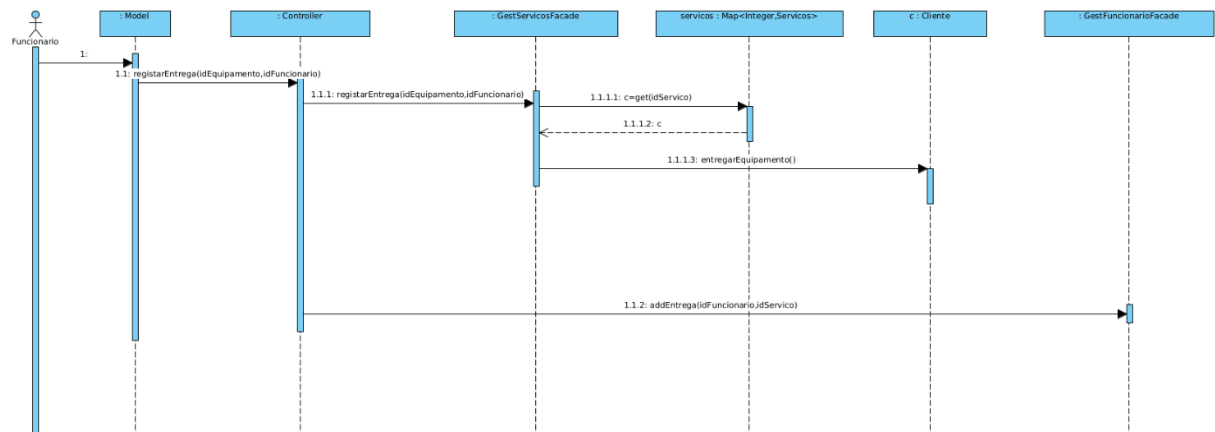
Diagramas de Sequência

Registrar Serviço Normal

O método de `criarServicoNormal` mostra os passos necessários para um funcionário registrar um serviço normal no sistema.

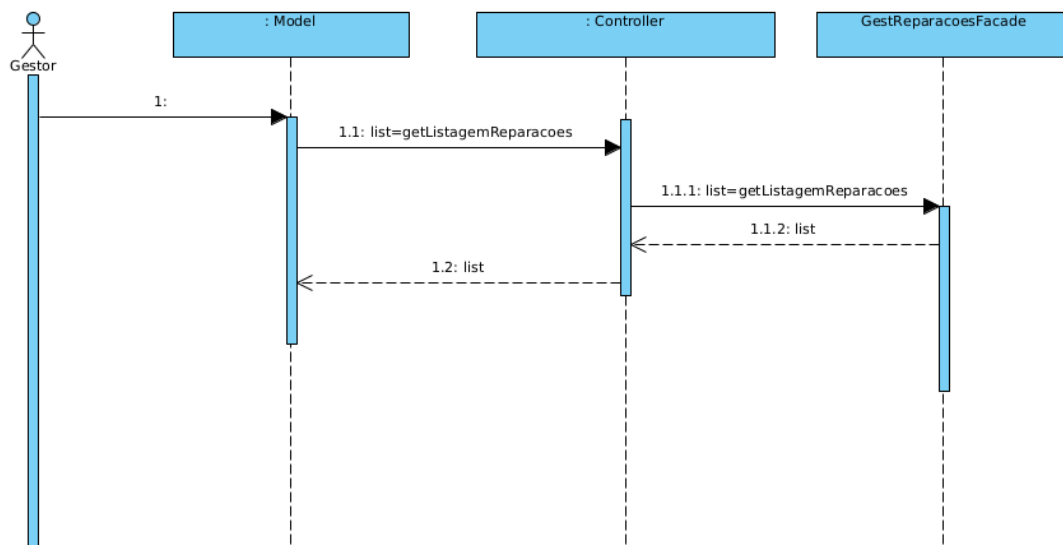


Registrar a Entrega e pagamento



O método registrarEntrega mostra os passos necessários para um funcionário registrar a entrega e o pagamento de um serviço.

Consultar Listagem de Reparações



O método getListagemReparacoes mostra os passos necessários para o gestor obter a listagens das reparações para cada técnico.

Análise Crítica Global dos Resultados

Durante a implementação do código tentámos sempre recorrer aos métodos de programação mais eficientes e corretos para cada problema em questão. Inicialmente ponderámos realizar o código ao mesmo tempo que estruturávamos o projeto através da criação dos diagramas, contudo, rapidamente compreendemos que esta metodologia não iria ser eficaz, nem seria nada correta tendo em conta o paradigma da cadeira de Desenvolvimento de Sistemas de Software. Esta metodologia não estava a ser inteligente visto que, à medida que íamos criando as classes e métodos, íamos também alterando os modelos constantemente, não estando a planear o trabalho de forma metódica e correta.

Deste modo, excluímos logo este método e decidimos realizar o projeto de maneira faseada, como anteriormente referido. No princípio do planeamento dos modelos de Domínio e Use Cases fomos bastante limitados na nossa visão e, consequentemente, estes modelos apresentaram inúmeras falhas, contrariedades e também redundâncias. Contudo, enquanto íamos debatendo soluções em grupo, as resoluções que consideramos mais adequadas começaram a ficar cada vez mais específicas e estruturadas, facilitando o futuro trabalho de implementar em código os modelos criados.

Consideramos que os diagramas que concebemos foram adequados e auxiliaram satisfatoriamente a nossa compreensão dos problemas em questão e ofereceram uma visão ampla e rápida do trabalho que iríamos ter que fazer. Sentimos também que o desenvolvimento deste projeto foi o mais constante que já tivemos, uma vez que também foi o primeiro projeto que estruturamos conforme as necessidades e requisitos impostos, levando a um fluxo menos turbulento do desenvolvimento em grupo, sendo que conseguimos facilmente dividir e atribuir tarefas a cada membro.

É importante notar que, apesar de nos terem auxiliado no desenvolvimento, os diagramas que criamos sofreram alterações durante a implementação, na medida em que alterámos um pouco a estratégia inicialmente pensada, afastando-nos dos modelos que nos estavam a guiar, tendo que, de seguida, corrigir os mesmos para serem corretamente ilustrativos do código escrito.

Conclusão

O projeto, para além de ter sido um excelente e inovador desafio, possibilitou-nos uma aprendizagem prática e eficaz e ajudou-nos a consolidar não só as matérias lecionadas nas aulas teóricas, mas também de temas abordados noutras cadeiras. Reconhecemos que o projeto foi enriquecedor e essencial para o nosso grupo, uma vez que nos permitiu enraizar e refinar todas as competências envolvidas.

Este trabalho prático consistiu no Planeamento e Desenvolvimento de um Software de Gestão de um serviço, sendo, deste modo, um excelente ponto de começo para a aprendizagem desta área, tendo-nos proporcionado uma melhor compreensão da estruturação de sistemas de software e de como esta é implementada.

Tivemos dificuldades inicialmente, uma vez que se tratavam de conceitos que nunca tinham sido lecionados, mas rapidamente assimilamos os mesmos e fizemos as devidas alterações e melhorias aos modelos e consideramos que o Sistema final foi bem especificado e implementado.

Em suma, foi uma ótima consolidação dos conceitos aprendidos e acreditamos que acabamos o projeto com um conhecimento muito mais profundo do que quando o iniciamos, pelo que sentimos que o próximo projeto que realizaremos será mais substancial e com menos falhas na estrutura. Deste modo, consideramos que o trabalho foi um sucesso.