

**Universidade do Minho**

Escola de Engenharia

Mestrado Integrado em Engenharia Informática

**Unidade Curricular de**

**Laboratórios de Informática IV**

Ano Lectivo de 2017/2018

**Smart Painting**

**Cesário Perneta – A73883**

**João Gomes – A74033**

**Nadine Oliveira – A75614**

**Ricardo Cardante – A75368**

**Tiago Fraga – A74092**

Março,2018

**LI41617**

|  |  |
| --- | --- |
| Data de Recepção |  |
| Responsável |  |
| Avaliação |  |
| Observações |  |

**Smart Painting**

**Cesário Perneta – A73883**

**João Gomes – A74033**

**Nadine Oliveira – A75614**

**Ricardo Cardante – A75368**

**Tiago Fraga – A74092**

Março,2018

# Resumo

O presente relatório descreve a primeira de três fases do projeto “*Smart Painting”* que surgiu da necessidade de implementar uma solução informática de marcação e seleção de variados serviços no ramo da pintura domiciliária.

Esta fase consiste na fundamentação do projecto, sendo iniciada pela contextualização do caso de estudo, da motivação e dos objetivos. Nestes tópicos, para além de descrevermos o contexto em que o projeto está inserido, o seu modo de funcionamento e o porquê de termos abraçado este projecto, abordaremos também as metas que definimos como pontos cruciais para levar este trabalho ao sucesso.

Após algumas reuniões e momentos de reflexão, conseguimos definir todos estes pontos permitindo assim, definir a identidade do nosso projecto, bem como os recursos e toda a informação que vamos precisar para o inicializar.

Em seguida, ultrapassada esta fase “mais teórica” do arranque do trabalho, colocando em prática os conhecimentos mais práticos elaboramos uma maquete do sistema, assim como um plano de desenvolvimento do trabalho futuro através da formulação do Diagrama de Gantt.

Por fim, apresentaremos algumas conclusões da elaboração desta primeira parte do trabalho e alguns aspetos que consideramos importantes para as fases posteriores.

**Área de Aplicação:** Desenvolvimento e Implementação de um sistema de marcação de serviços de pintura.

**Palavras-Chave:** Desenvolvimento de Software, Engenharia de Software, Projecto, Sistema, Diagrama de Gantt, Aplicação, Pintura, Pintores.

# Índice

[Resumo i](#_Toc511577072)

[Índice ii](#_Toc511577073)

[Índice de Figuras iv](#_Toc511577074)

[Índice de Tabelas v](#_Toc511577075)

[1.1. Contextualização 1](#_Toc511577076)

[1.2. Apresentação do Caso de Estudo 2](#_Toc511577077)

[1.3. Motivação e Objetivos 3](#_Toc511577078)

[1.3.1 Motivação 3](#_Toc511577079)

[1.3.2 Objetivos 3](#_Toc511577080)

[1.4. Justificação do Sistema 4](#_Toc511577081)

[1.5. Utilidade do Serviço 4](#_Toc511577082)

[1.6. Estabelecimento da Identidade do Projecto 5](#_Toc511577083)

[1.7. Identificação dos Recursos Necessários 5](#_Toc511577084)

[1.8. Maqueta do Sistema 6](#_Toc511577085)

[1.9. Medidas de Sucesso do Projecto 7](#_Toc511577086)

[1.10. Plano de Desenvolvimento (Diagrama de Gantt) 8](#_Toc511577087)

[2.1. Requisitos do Utilizador 10](#_Toc511577088)

[2.2. Requisitos do Sistema 11](#_Toc511577089)

[2.2.1 Requisitos Funcionais 11](#_Toc511577090)

[2.2.2 Requisitos Não Funcionais 13](#_Toc511577091)

[3.1. Modelo de Domínio 14](#_Toc511577092)

[3.2. Modelo de Use Case 15](#_Toc511577093)

[3.3. Especificação de Use Cases 17](#_Toc511577094)

[3.4. Diagrama de Sequência de Use Case 19](#_Toc511577095)

[3.4.1 Exemplo de Diagrama de Sequência – Marcar Serviço 19](#_Toc511577096)

[3.4.2 Exemplo de Diagrama de Sequência – Adicionar Pintor 21](#_Toc511577097)

[3.5. Diagrama de Sequência de Subsistemas 22](#_Toc511577098)

[3.5.1 Exemplo de Diagrama de Sequência de Subsistemas – Marcar Serviço 23](#_Toc511577099)

[3.5.2 Exemplo de Diagrama de Sequência de Subsistemas – Adicionar Pintor 24](#_Toc511577100)

[3.6. Diagrama de Classes 24](#_Toc511577101)

[4.1. Modelo Concetual 27](#_Toc511577102)

[4.1.1 Tipos de entidades 27](#_Toc511577103)

[4.1.2 Tipos de relacionamentos 27](#_Toc511577104)

[4.1.3 Associar atributos com os tipos de entidades e relacionamentos 28](#_Toc511577105)

[4.1.4 Determinar domínio dos atributos 31](#_Toc511577106)

[4.1.5 Determinar chaves primárias, candidatas e alternativas 32](#_Toc511577107)

[4.1.6 Desenho do Diagrama E-R 33](#_Toc511577108)

[4.1.7 Revisão do modelo de dados com o utilizador 33](#_Toc511577109)

[4.2. Modelo Lógico 34](#_Toc511577110)

[4.2.1 Entidades 34](#_Toc511577111)

[4.2.2 Relacionamentos 34](#_Toc511577112)

[4.2.3 Atributos Multi-Valor 36](#_Toc511577113)

[4.2.4 Normalização 36](#_Toc511577114)

[4.2.5 Elaboração e validação do esquema lógico da base de dados 37](#_Toc511577115)

[4.2.6 Análise do crescimento futuro 37](#_Toc511577116)

[4.2.7 Revisão do modelo lógico com o utilizador 38](#_Toc511577117)

[Anexos 44](#_Toc511577118)

**Anexos**

# Índice de Figuras

[Figura 1. Maqueta do Sistema. 7](#_Toc511577119)

[Figura 2. Diagrama de Gantt 9](file:////Users/tiagofraga/Desktop/LI4/Trabalho/Local/Fase2/Relatorio/Fase2_LI4.docx#_Toc511577120)

[Figura 3 - Modelo de Domínio 15](#_Toc511577121)

[Figura 4 - Diagrama de Use Case 17](#_Toc511577122)

[Figura 5 - Especificação de Use Case - Marcar Serviço 18](#_Toc511577123)

[Figura 6 - Especificação de Use Case - Adicionar Pintor 19](#_Toc511577124)

[Figura 7 - Diagrama de Sequência - Marcar Serviço 21](#_Toc511577125)

[Figura 8 - Diagrama de Sequencia - Adicionar Pintor 22](#_Toc511577126)

[Figura 9 - DSS - Marcar Serviço 23](#_Toc511577127)

[Figura 10 - DSS - Adicionar Pintor 24](#_Toc511577128)

[Figura 11 - Diagrama de Classes de Back Office 25](#_Toc511577129)

[Figura 12 - Diagrama de Classes de Front Office 26](#_Toc511577130)

[Figura 13 - Diagrama E-R 33](#_Toc511577131)

[Figura 14 - Transformação do relacionamento Solicita em tabela. 34](#_Toc511577132)

[Figura 15 - Transformação do relacionamento Realiza em tabela. 35](#_Toc511577133)

[Figura 16 - Transformação do relacionamento Utiliza em tabela. 35](#_Toc511577134)

[Figura 17 - Transformação do atributo Multivalor Lista de Favoritos em tabela. 36](#_Toc511577135)

[Figura 18 - Modelo Lógico 37](#_Toc511577136)

# Índice de Tabelas

[Tabela 1 - Tabela das Entidades 27](#_Toc511577137)

[Tabela 2 - Tabela dos Relacionamentos 28](#_Toc511577138)

[Tabela 3 - Associação dos atributos da entidade Cliente 29](#_Toc511577139)

[Tabela 4 - Associação dos atributos da entidade Serviço 30](#_Toc511577140)

[Tabela 5 - Associação dos atributos da entidade Pintor 30](#_Toc511577141)

[Tabela 6 - Associação dos atributos da entidade Material 31](#_Toc511577142)

1. Introdução
   1. Contextualização

De acordo com o ultimo inquérito “*Quality of life in european cities* “Braga foi considerada como a 12ª cidade com melhor qualidade de vida na europa, obtendo a melhor qualificação a nível nacional.

As justificações fornecidas pela população para corroborar tal distinção foram diversas, tais como melhoras nos serviços de saúde, uma maior satisfação com o serviço de educação, e ainda, dizem ser uma das cidades onde encontram habitação de qualidade a um preço razoável.

Braga é ainda uma cidade que tem apresentado um grande crescimento económico, sendo apontado como fator de relevo a grande oferta de mão-de-obra qualificada que leva a que grandes, médias e pequenas empresas optem por se sedearem ou expandirem-se para esta cidade, contribuindo assim, para o aumento do emprego na região. Ora, tal desenvolvimento, levou a um aumento na procura de imoveis, tanto para arrendamento, como para comprar e ainda para investimento.

Foi devido a esta ultima situação que decidimos sedear em Braga uma empresa de prestação de serviços de pintura de imoveis, que irá atuar na mesma. Com tamanha procura de habitações, principalmente para arrendamento, em que os contratos são de tempo limitado, é importante e necessário uma manutenção regular do imóvel, de forma a que este se mantenha apelativo e rentável. Assim sendo, verificamos que os serviços que disponibilizamos enquadram-se perfeitamente na cidade de Braga, oferecendo aos proprietários uma maior facilidade em requisitar o nosso tipo de serviços.

Para tornar a nossa empresa mais apelativa será desenvolvida uma aplicação móvel que permite contratar os nossos serviços de acordo com as preferências do cliente. Esta aplicação apresentará ao cliente um catálogo de cores e de tipos de tintas, fornecendo ao cliente a opção de este poder escolher com tranquilidade, e até de debater com amigos, familiares ou até mesmo com decoradoras, quais os tons mais indicados para o seu imóvel.

* 1. Apresentação do Caso de Estudo

O sistema a ser desenvolvido divide-se em três tipos de interface – uma para o utilizador, outra para o pintor e a ultima para os proprietários – em distintos tipos de plataforma.

Na interface do utilizador, será criada uma aplicação móvel que depois de descarregada das “*Apps Stores*” permitirá que o cliente através do seu smartphone:

* + Solicite um serviço de pintura a ser executado na hora. (Não pode escolher o pintor, ser-lhe-á destacado o que estiver mais próximo e disponível da habitação);
  + Agende um serviço de pintura para um determinado dia e hora, podendo assim, escolher o pintor que pretende que faça o serviço, conforme a disponibilidade do mesmo;
  + Avalie pintores;
  + Adicione fotos/comentários no perfil de cada pintor sobre o serviço;
  + Reporte pintores, caso se passe algo que não esteja de acordo com a conduta do serviço;
  + Partilhe perfis/comentários nas redes sociais;
  + Verifique faturas/serviços passados.
  + Efetue pagamento.

Com todas estas funcionalidades, a aplicação terá uma rica e poderosa capacidade para satisfazer todos os desejos e necessidades dos clientes, desde o agendamento, avaliação e pagamento do serviço/pintores até á própria critica, permitindo assim melhorar os trabalhos posteriores.

Na interface do trabalhador, que também será no formato móvel de maneira a possibilitar que leve o seu *smartphone/tablet* e aceda a toda a informação, no local, do serviço solicitado pelo cliente. Desta forma o pintor pode:

* + Verificar a quantidade de serviços que tem agendados;
  + Visualizar a localização do domicilio do cliente;
  + Alterar estado de disponibilidade;
  + Em cada serviço visualizar todos os pormenores solicitados pelo cliente;
  + Visualizar o seu perfil (Ver avaliações e comentários, no entanto, não o pode editar );

Por fim, os trabalhadores, catalogo de serviços, catalogo de cores e todas as características que o sistema pode oferecer serão guardadas numa base de dados online. Essa base de dados pode ser gerida – através do browser – pelo proprietário do negocio de forma que:

* + Adicione/Remova trabalhadores;
  + Altere catalogo de cores;
  + Modifique características de serviços;
  + Edite perfis de trabalhadores;
  1. Motivação e Objetivos
     1. Motivação

Como todos sabemos, para a realização de qualquer trabalho de pintura domiciliaria, o pintor tem de ir conhecer o que vai pintar, fazer as medições necessárias, informar-se acerca da cor que o cliente deseja, verificar se a parede apresenta danos, fazer o orçamento, etc. Todas estas situações retiram tempo quer ao cliente que tem de estar presente no momento da avaliação e ao trabalhador que tem de fazer a avaliação da parede.

O facto de o cliente conseguir explicar todos os problemas e tudo aquilo que deseja de uma forma simples, reduzindo assim o tempo desperdiçado em visitas do trabalhador só para reunir informação e se preparar para o trabalho a efetuar é uma das nossas motivações.

Durante este processo de pesquisa, reparamos que a única forma existente para contratar um pintor, era através de anúncios existentes em alguns sites ou pelos métodos tradicionais. Este facto motivou-nos a criar uma plataforma capaz de permitir contratar este tipo de trabalhador de forma fácil e simples para cada utilizador. Outro fator que também nos deparamos na nossa pesquisa, foi o do mercado imobiliário na cidade de Braga ter sofrido um grande crescimento ao longo dos últimos anos.

Este conjunto de factos motivou-nos a arranjar uma solução que acompanhasse o avanço tecnológico e imobiliário e assim apresentar uma alternativa aos meios tradicionais.

* + 1. Objetivos

O principal objetivo da solução informática apresentada é disponibilizar uma interface ao utilizador intuitiva e de fácil utilização, que simplifique o processo de marcação do serviço.

Da forma como a caracterização do tipo de serviço é implementada, reduzimos o tempo de trabalho bem como o numero de deslocações efetuadas pelo trabalhador e permitimos ao trabalhador ter os materiais necessários para o serviço logo à partida.

Além disso, a localização geográfica permite a que o trabalhado saiba o local onde vai realizar o serviço e ainda como chegar lá evitando trajetos demorosos.

O facto de todos os utilizadores poderem avaliar, comentar e reportar um determinado trabalhador, leva a que as informações presentes nas aplicações sejam as mais verídicas possíveis, assim sempre que um utilizador não conhece o trabalhador disponível para fazer o trabalho pode ter em conta a consideração as opiniões de outros utilizadores.

* 1. Justificação do Sistema

Após uma análise no vasto mercado de aplicações existente nos dias de hoje, deparamo-nos com a falta de uma aplicação que visasse fornecer um serviço de pintura mais rápido, cómodo e eficaz. Por exemplo, obriga a pessoa que requisitou o serviço a permanecer em sua casa, levando muitas à impossibilidade de comparência no seu emprego. Com este sistema, o cliente não precisa de abdicar de tanto tempo, visto que uma das etapas, o reconhecimento dos espaço e questões mais superficiais, são logo abordadas no pedido ao sistema.

Outra vantagem relação a lojas de pintura físicas, é que o cliente não tem de se deslocar à loja para escolher certos pormenores, como por exemplo a cor. Com este sistema, o cliente tem uma perceção imediata de como fica a cor que pretende através de um catálogo de cores, algo que agiliza o processo, economizando tempo e dinheiro.

Tendo em conta todas as dificuldades encontradas quando se pretende fazer remodelações a nível de pintura em casa, a nossa empresa desenvolveu um sistema, baseado numa aplicação e também num site, que pretende resolver todas as adversidades principalmente no início do processo.

* 1. Utilidade do Serviço

Dado a grande dinâmica existente no setor imobiliário presente em Braga, verificamos que o serviço por nós disponibilizado, preenche algumas lacunas existentes na disponibilização e prestação deste serviço na região, nomeadamente na maneira de "chegar" ao cliente.

É importante acompanhar o ritmo de desenvolvimento da cidade, proporcionando ao consumidor certas comodidades, e é exatamente isto que faltava neste tipo de serviço, uma maneira fácil de o cliente poder requisitar o serviço de forma rápida, em qualquer altura e para qualquer altura.

* 1. Estabelecimento da Identidade do Projecto

Nome: “Smart Painting”;

Categoria: “Serviços – Pintura”;

Descrição: A *Smart Painting* é uma aplicação que o vai ajudar quando precisar de dar outra alegria e cor ao seu lar; apenas precisa de ter o seu smartphone à mão e no momento a seguir poderá ter em sua casa um pintor certificado e preparado com as suas escolhas, de forma a dar a vivacidade e a mudança que pretende. No final, pode avaliar os nossos serviços e inclusive recomendar aos seus amigos através das suas redes sociais e efetuar o pagamento sem chatices, pois pode associar o seu cartão de crédito ao seu perfil de cliente. Não volte a aventurar-se no incerto e descarregue já a nossa aplicação.

Características:

* + Serviço imediato;
  + Agendamento de serviços;
  + Avaliação dos pintores;
  + Comentários e fotos dos serviços;
  + Partilha em redes sociais;
  1. Identificação dos Recursos Necessários

Uma parte importante do projeto será de que modo se irá obter recursos e como serão tratados e validados. Inicialmente vamos recorrer a sites de lojas existentes para recolha de dados.

Estes são *Robbialac, Cin e Sotinco*. Destes sítios pretendemos extrair uma lista com os materiais necessários a realização de trabalhos de pintura, bem como todos os tipos de cores de tintas, extraindo também os seus preços.

Pela observação realizada, chegamos à conclusão que conseguiríamos toda a informação necessária acerca de materiais e seus preços no catalogo disponibilizado por estas empresas. A nossa empresa por sua vez vai tentar estabelecer uma parceria com estas empresas, de modo a conseguir preços reduzidos em materiais, o que é uma excelente vantagem quer para nós que conseguimos apresentar melhores preços aos nossos clientes, quer para as empresas que nos fornecem os materiais, uma vez que utilizamos os seus produtos.

Os utilizadores só poderão realizar marcações, avaliações de trabalhadores e reportar informação incorreta ou uma situação de desagrado com o trabalhador.

Em caso de um trabalhador receber demasiadas denúncias por parte dos utilizadores, nós enquanto empresa removemos esse trabalhador da lista de escolhas e prosseguimos com o despedimento do mesmo.

Ainda relativamente a aplicação, iremos ainda recorrer a *API’s* do *Google+, Facebook e Twitter* para login, de modo a termos acesso logo ao perfil do utilizador, facilitando assim o registo por parte do utilizador.

Para além disto, vamos utilizar várias ferramentas ao longo deste projeto, tais como:

* *Microsoft Project* para planeamento do projeto;
* *Microsoft Word* para redação dos relatórios;
* *Microsoft PowerPoint* para realização das apresentações;
* *Visual Paradigm* para modelação do projeto;
* *Microsoft SQL Server* para criação da base de dados;
* *Microsoft Visual Studio* éo *IDE* utilizado;
* *Microsoft .NET C#* para o código da nossa aplicação;
* *Bing Maps* que será utilizado para mostrar num mapa a localização da casa do cliente.
  1. Maqueta do Sistema

A aplicação a desenvolver será baseada numa arquitetura com três camadas, camada de apresentação, negócio e de dados.

A nível da camada de apresentação, vamos possuir três interfaces distintas, uma para os clientes onde estes poderão requisitar os nossos serviços, outra para os trabalhadores para estes terem acesso ao tipo de serviço pretendido pelo cliente e os dados do mesmo, e ainda uma para o proprietário, onde este poderá gerir o seu negócio.

Na camada de negócio, é onde serão tratados todos os pedidos vindos dos utilizadores, estes pedidos vão ser tratados nesta camada, de modo a responder às necessidades pretendidas. Para dar resposta a estes pedidos, esta camada mantém uma ligação com a camada de dados para ir buscar informação necessária ao processamento destes.

Na base de dados, é onde será guardada toda a informação relevante para o funcionamento da aplicação, como por exemplo os catálogos disponíveis, a informação dos clientes e dos funcionários, os tipos de serviços, entre outros.

Em baixo é apresentada uma maquete geral da arquitetura do sistema a implementar, onde são apresentadas apenas algumas das funcionalidades que se pretende implementar.

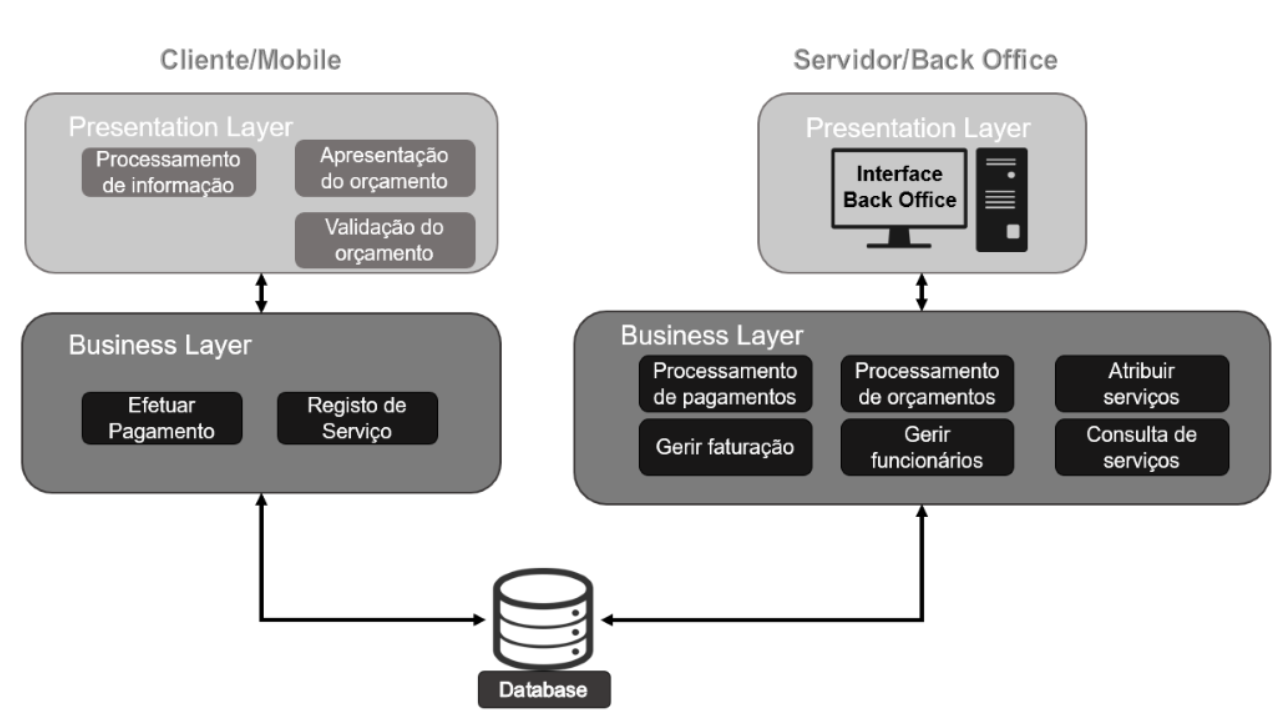


Figura 1. Maqueta do Sistema.

* 1. Medidas de Sucesso do Projecto

Torna-se fulcral na fase de fundamentação de projeto, organizar uma série de medidas que testem o êxito do mesmo.

Decidimos que a primeira vistoria do sucesso e crescimento do nosso projeto irá ser efetuado ao decimo segundo mês desde o dia em que aplicação comece a sua atividade. Dessa forma, é seguidamente apresentada uma lista com os tópicos que consideramos basilares para obter sucesso com a aplicação e com o site.

* Obter 1000 serviços contratados;
* Aumentar em 40% o número de pintores;
* Conseguir angariar, pelo menos, mais um parceiro de tintas.
* Atingir um lucro de 50 000 euros.

Após esta etapa inicial, temos como objetivo principal e a longo prazo, expandir os nossos serviços para fora do distrito de Braga e estar a operar nas duas grandes cidades nacionais, Porto e Lisboa.

* 1. Plano de Desenvolvimento (Diagrama de Gantt)

O diagrama de Gantt foi o gráfico usado para planearmos todas as etapas do nosso projeto.

O nosso projeto foi dividido em três partes fundamentais, com datas de finalização correspondentes as datas de entrega de cada parte do projeto. As etapas a que me refiro são as seguintes:

1. Fundamentação -> 12/02/2018 até 05/03/2018

2. Especificação -> 06/03/2018 até 16/04/2018

3. Implementação -> 17/04/2018 até 28/05/2018

Dentro destas fases fomos dividindo em várias etapas secundárias que achamos mais relevantes, neste resumo optamos por assinalar as que achamos que precisaríamos de mais tempo para a sua respetiva realização.

Na fase de Fundamentação achamos que a Pesquisa sobre o tema levava mais tempo visto corresponder a fase de coleta de informação/ dados sobre o tema escolhido.

Na fase de Especificação concordamos que a Implementação da Base de Dados consumiria bastante mais tempo que as outras tarefas desta fase, não esquecendo que a Analise de Requisitos, os Diagramas de Sequência e os Diagramas de Classe consumiriam bastante do nosso tempo daí a referenciar estas como as tarefas fundamentais da segunda fase.

Na terceira e ultima fase a Implementação achamos que a Implementação do Sistema de Localização ocupava grande parte do nosso tempo em relação as outras tarefas definidas nesta fase, visto ser a primeira vez que implementamos um sistema de localização nas nossas aplicações.

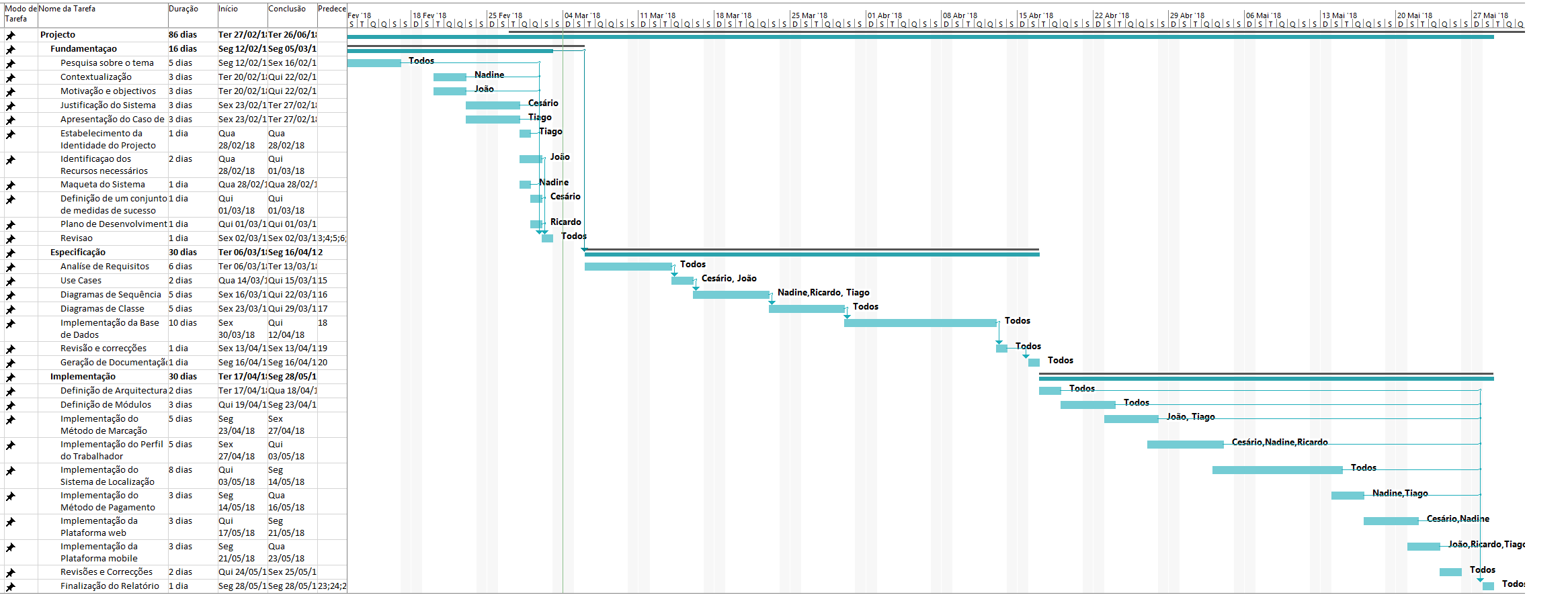


Figura 2. Diagrama de Gantt

1. Análise de Requisitos

A pesquisa e análise de requisitos trata-se duma importante e essencial fase do na especificação do projeto dada a sua função de descrever o funcionamento de um sistema. É fulcral existir uma listagem das pretendidas capacidades do *software* que reflitam ambas as ideias dos desenvolvedores e clientes. Para tal efetuou-se vários estudos na área da pintura imobiliária e as suas deficiências, bem como entrevistas a trabalhadores por conta própria e de outrem para saber o que mais se pretende duma aplicação do género.

Nas seguintes secções fazer-se à apresentação dos requisitos de utilizador bem como os de sistema, que se dividem em funcionais e não funcionais.

* 1. Requisitos do Utilizador

Aborda-se neste capítulo os requisitos de utilizador, no entanto é importante destacar dois aspetos que permitem uma compreensão básica do sistema:

1. Na aplicação móvel existem dois tipos de utilizador – clientes e proprietários de negócio – ambos com distintas funções.
2. O proprietário do negócio terá acesso a um site no browser de forma a controlar todo o negócio com funções completamente distintas dos visados em cima.

* Funções do Cliente:
  + Registo e Login;
  + Agendar serviço a ser executado no imediato;
  + Reserva de serviço para uma determinada data onde pode:
* Introduzir local (Interior ou Exterior)
* Introduzir tipo de superfície
* Introduzir Medidas
* Introduzir cor
* Introduzir observações
  + Escolher Pintor (Caso Reserve o serviço);
  + Escolher Método de Pagamento, no final de cada agendamento, podendo associar o cartão de crédito ou por referência bancária;
  + Classificar Serviço através de fotos, comentários e numeração de 1 a 5. Classificar pintor através de comentário e numeração;
  + Aceder ao Perfil pessoal de forma a visualizar faturas passadas;
  + Visualizar Perfil do Pintor;
  + Adicionar Pintores a lista de Favoritos;
  + Partilhar o serviço/avaliação nas redes sociais;
* Funções do Pintor:
  + Efetuar Login;
  + Aceder ao Perfil pessoal, podendo visualizar as avaliações;
  + Visualizar Registo de Serviços efetuados;
  + Visualizar as marcações agendadas;
  + Aceder ao serviço de GPS;
  + Alterar estado de disponibilidade, importante para a marcação de serviço na hora;
* Funções do Proprietário:
  + Efetuar Login;
  + Adicionar Trabalhadores;
  + Remover Trabalhadores;
  + Editar lista de Materiais;
  + Visualizar marcações agendadas;
  + Editar Preço dos Materiais;
  1. Requisitos do Sistema

Os requisitos do sistema classificam-se em requisitos funcionais - que corresponde ao que o sistema deve fornecer, responder ou comportar – e requisitos não funcionais – que dizem respeito a limitações nos serviços ou funções oferecidas pelo sistema.

* + 1. Requisitos Funcionais
* Registo:
  + De modo a manter a distinção entre cliente, pintor e proprietário, implica obrigatoriamente a criação de diferentes perfis para os diferentes tipos de utilizadores de modo a separar as diversas funcionalidades;
  + Como o sistema oferece a capacidade de efetuar o registo através das redes sociais, este será feito usando as API´s de Facebook, etc. Este ponto tem como objetivo extrair a foto principal, o nome, o email.
  + Todos os utilizadores têm de estar devidamente autenticados para exercer as respetivas funções.
  + As permissões serão geridas pelo sistema gestor de base de dados (SGBD) que será responsável pela atribuição das funções definidas, para cada tipo de utilizador.
* Marcação:
  + Consoante a escolha do local de serviço (Interior/Exterior) é retirada da base dados as tintas correspondentes ao pedido efetuado;
  + É efetuada uma filtragem ao catálogo acima descrito consoante o tipo de superfície;
  + Após ser indicado o tamanho é apresentando a lista de tintas disponíveis para o pedido selecionado.
  + É calculado o preço consoante as escolhas efetuadas e apresentado ao cliente.
* Pagamento e Serviço:
  + Após a marcação de serviço que fique concluída com o devido pagamento, o serviço fica definitivamente agendado, sendo adicionado ao perfil do trabalhador em causa e notificando o cliente com um aviso.
  + Após o pagamento a fatura é enviada por email e adicionada ao perfil do cliente, para ser visualizada mais tarde.
* Avaliação:
  + Como o sistema oferece a capacidade de efetuar partilha de comentários e avaliações através das redes sociais, este será feito usando as API´s de Facebook, etc., de modo a realizar uma partilha confiável e segura;
  + Após o utilizador aceder ao perfil do trabalhador, e indicar o número da avaliação, comentário e caso pretenda adicionar uma ou mais fotografias, estas são adicionadas ao perfil do pintor e consequentemente na base de dados;
  + Caso o cliente adicione um trabalhador aos seus favoritos, o sistema adiciona o nome do trabalhador a lista de favoritos presente no perfil do cliente;
* Proprietário:
  + De modo a apagar um trabalhador é necessário que este já esteja registado no sistema, eliminando, assim, todos os dados relativos a este trabalhador do sistema;
  + De modo a adicionar um trabalhador é necessário que este não esteja já registado;
  + De modo a alterar a lista de materiais o proprietário se pretender adicionar terá de indicar a marca, o nome e uma fotografia e preço;
    1. Requisitos Não Funcionais
  + O utilizador tem de ter equipamento capaz de suportar as funcionalidades de GPS;

1. Modelação em UML
   1. Modelo de Domínio

Neste capítulo vai ser representado o modelo de domínio da aplicação. Este modelo foi criado na fase inicial da modulação, com o intuito de identificar, logo à partida, todas as entidades que iram compor a aplicação. Após esta identificação, foi feita uma cuidada análise e identificados então todos os relacionamentos presentes no domínio da aplicação.

Foram identificadas logo de início as três entidades que vão interagir com a aplicação, **Pintor, Proprietário** e **Cliente.**

O **Pintor** tem associado um email, um nome, uma fotografia, a disponibilidade atual, isto é, se se encontra ocupado ou disponível para efetuar um serviço e ainda, poderá realizar serviços.

O **Cliente** tem associado um email, um nome, uma fotografia, uma morada, uma lista de favoritos, onde irá poder adicionar pintores que prefira, um histórico de marcações de todos os serviços já efetuados, e ainda, um histórico com todas as faturas dos mesmos.

O **Proprietário** tem associado um email e um nome e será quem efetuará a gestão dos pintores, podendo adicionar ou remover os mesmos.

Outra das entidades mais importantes identificada é o **Serviço.** O serviço pode ser requisitado de duas formas, ou imediato, caso o cliente pretenda o serviço na hora, ou agendado em que o cliente marca o serviço para uma data específica. Vai ser caracterizado por um local, isto é, se é um serviço interior ou ao ar livre, e por um tipo de superfície a pintar, onde irá ser possível escolher as tintas e as cores com que se pretende pintar a mesma. Tem ainda relacionado um pagamento e um preço, pagamento este que poderá ser efetuado com cartão de crédito, ou então, por referência de multibanco. Por fim, poderá também ser feita uma avaliação ao serviço quando este for concluído.

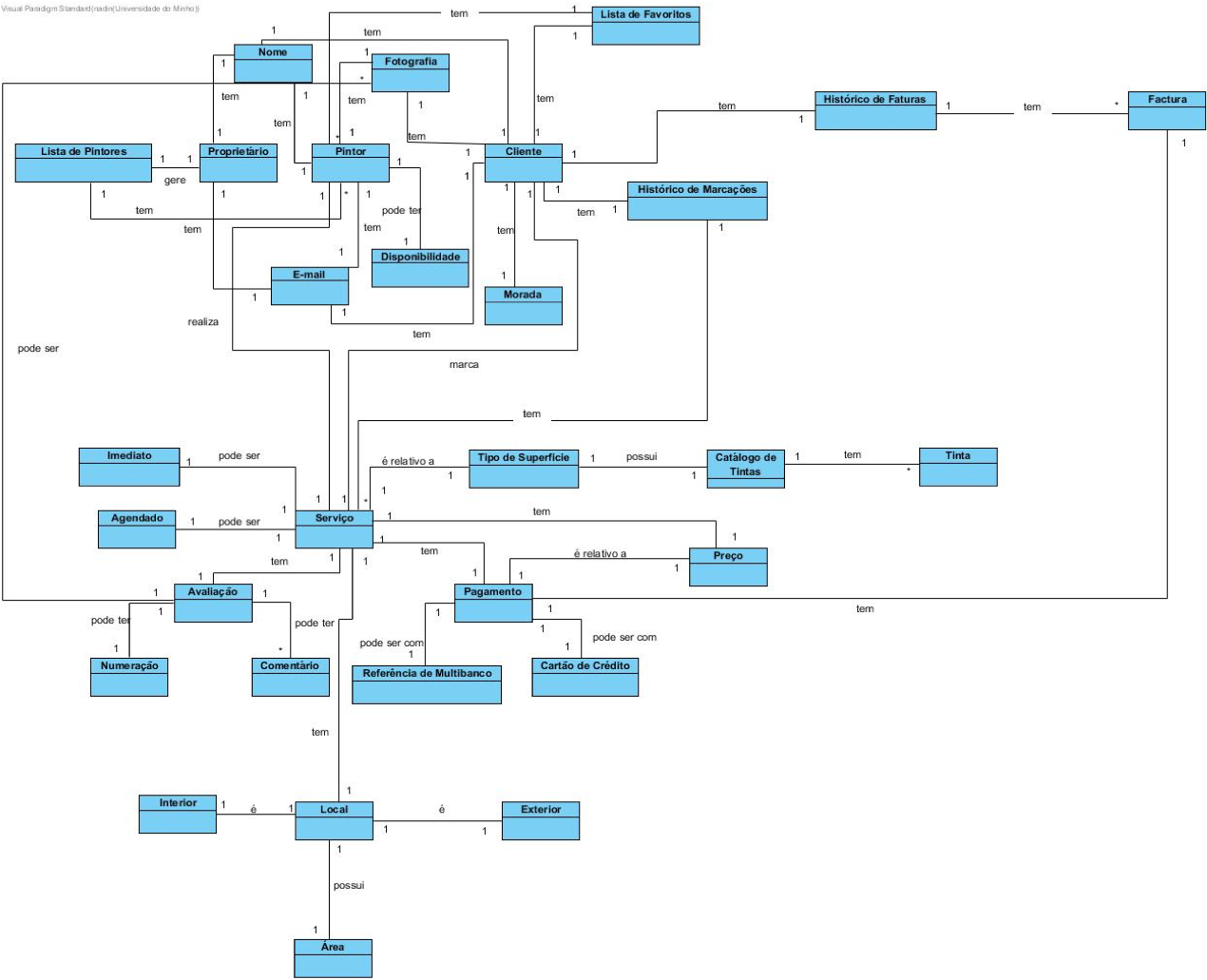


Figura 3 - Modelo de Domínio

* 1. Modelo de Use Case

Apresentado o modelo de domínio da aplicação, torna-se necessário identificar os utilizadores que vão interagir com um sistema, bem como as funcionalidades a ser implementadas, é, portanto, neste capítulo retratado o modelo de use case.

Este modelo, permite-nos representar as diferentes interações que cada ator vai poder realizar no sistema. É um modelo simples, com pouco detalhe técnico, que consegue proporcionar uma vista geral do resultado da aplicação final.

No contexto da nossa aplicação *Smart Painting* foram identificados três atores distintos:

* **Cliente -**  O cliente é o ator fulcral ao sistema, que utiliza a aplicação para requisitar os serviços disponibilizados pela mesma.
* **Proprietário –** O proprietário vai ser o ator que gere a aplicação, adicionando e removendo pintores e gerindo o stock de materiais.
* **Pintor –** O pintor é registado pelo proprietário, e é o ator que vai executar os serviços requisitados pelo cliente.

Após a identificação dos atores, foram analisadas todas as funcionalidades que eles poderiam aceder na aplicação:

* **Cliente:**
  + Registo;
* Solicitar Serviço;
* Marcar Serviço;
* Ver Faturas;
* Adicionar Favoritos;
* Partilhar nas Redes Sociais;
* Avaliar Serviço;
* **Proprietário**
* Adicionar Pintor;
* Remover Pintor;
* Editar Materiais;
* Editar perfil pintores;
* Ver Serviços Agendados;
* Ver Serviços Efetuados;
* **Pintor**
* Ver Serviços efetuados;
* Ver Serviços agendados;
* Alterar estado do Serviço;
* Ver avaliações;
* Alterar disponibilidade;

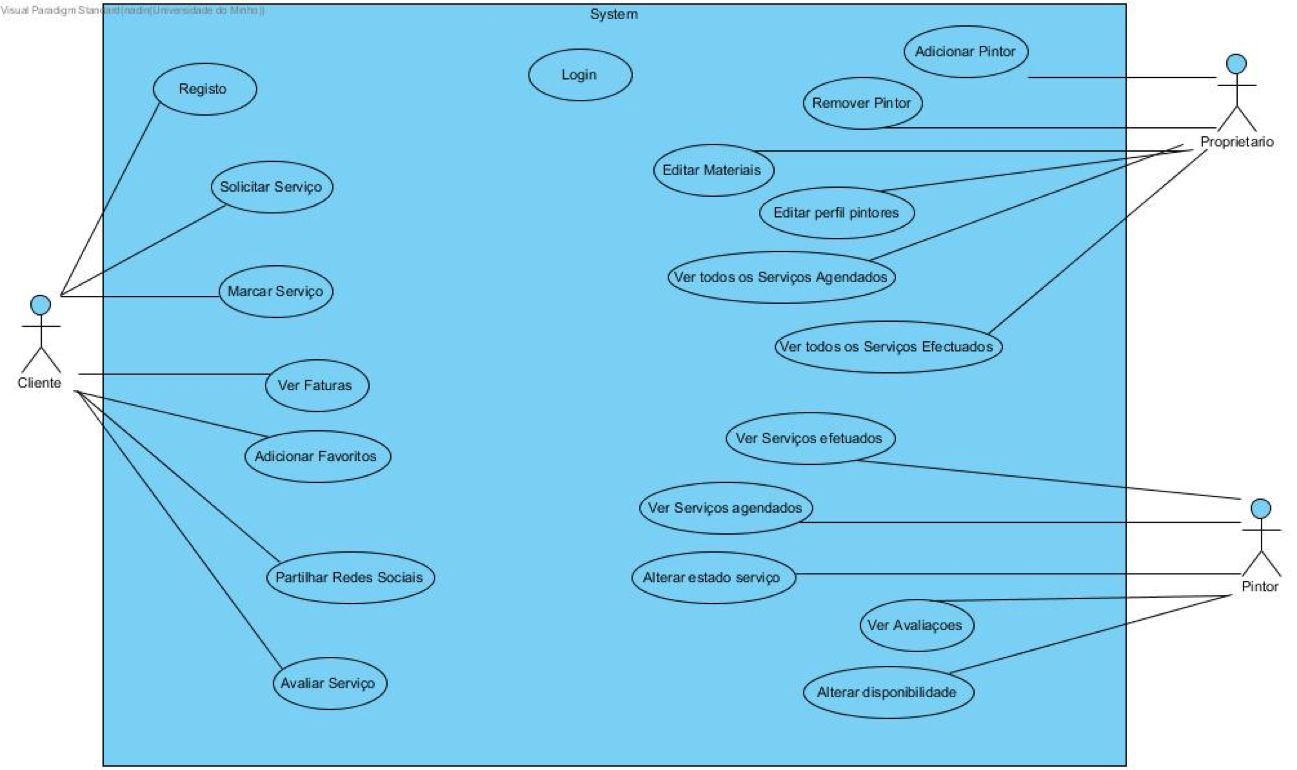


Figura 4 - Diagrama de Use Case

* 1. Especificação de Use Cases
     1. Especificação de Use Cases – Marcar Serviço

Na seguinte imagem é possível observar a descrição em formato tabular do Use Case “Marcar Serviço”, que tem como ator um Cliente. Na linha da breve descrição é descrito que esta funcionalidade permite realizar uma reserva de um serviço para um determinado dia e hora.

A pré-condição indicada é que o Cliente para utilizar esta funcionalidade tem de estar devidamente autenticado na aplicação, e como pós-condição temos que a marcação é realizada com sucesso.

De seguida é descrito todo o processo de marcação, que ocorre sequencialmente, mostrando todas as interações entre o ator e o sistema. No inicio o ator introduz as características do serviço que pretende, o sistema verifica tais características, em caso dessas características não estarem de acordo, é lançado uma exceção, obrigando o ator a reintroduzir as características.

Nesta funcionalidade é dada a liberdade ao ator de escolher uma data e um pintor para realizar o serviço, portanto o sistema verifica se o pintor escolhido está disponível para essa data, caso não esteja é lançada outra exceção, obrigando o ator a fazer nova escolha de pintor.

Posto isto o sistema apresenta os métodos de pagamento, o ator escolhe o método e realiza o pagamento, o sistema precisa de validar o pagamento, se o pagamento não tiver sido efetuado com sucesso, é lançada novamente uma exceção, obrigando o ator a fazer novo pagamento. Terminado o pagamento é emitida e enviada a fatura ao cliente, e por sua vez notifica o ator que tal se sucedeu.

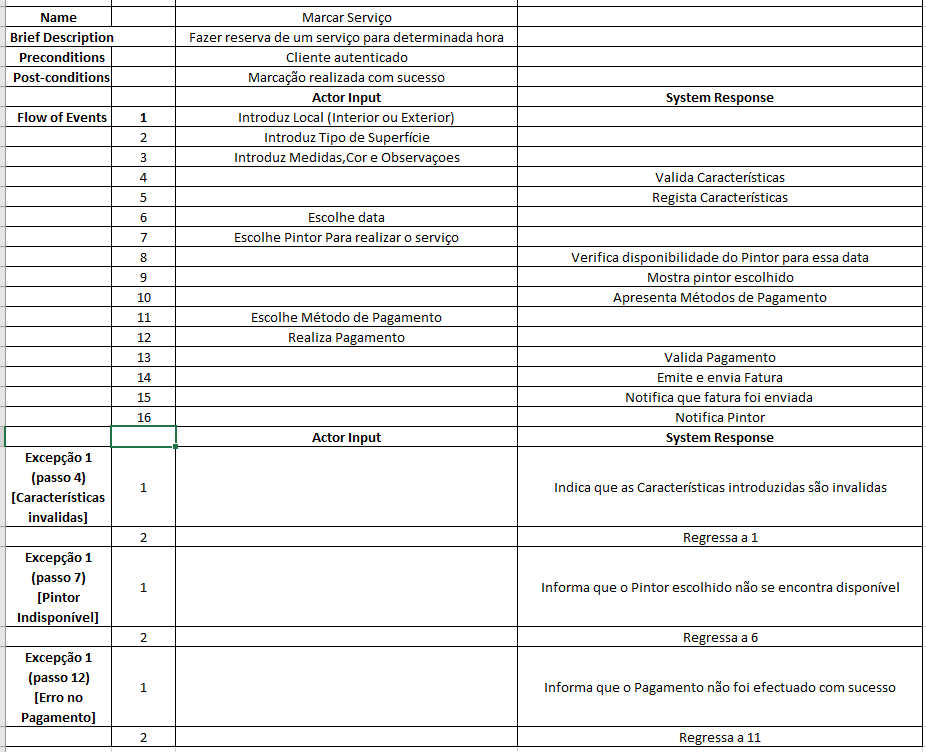


Figura 5 - Especificação de Use Case - Marcar Serviço

* + 1. Especificação de Use Cases – Adicionar Pintor

Como podemos observar na especificação da funcionalidade “Adicionar Pintor”, temos como ator o Proprietário. Na linha da breve descrição é descrito que esta funcionalidade permite ao proprietário adicionar um Pintor ao sistema.

Como pré-condição temos que o Proprietário para puder utilizar esta funcionalidade tenha de estar devidamente autenticado, e como pós-condição temos que o Pintor é adicionado com sucesso.

De seguida, temos a descrição sequencial do processo de adicionar um pintor, para que tal aconteça, o ator insere os dados do Pintor a adicionar, inseridos os dados o sistema valida os mesmos, em caso de os dados serem inválidos é lançado uma de duas exceções, a primeira quando os dados são referentes a um pintor já existente no sistema, e outra quando os dados não são inseridos corretamente.

Posto esta fase o sistema regista os dados do pintor, cria dados de autenticação, devolve os dados criados e adiciona o pintor a lista de pintores.

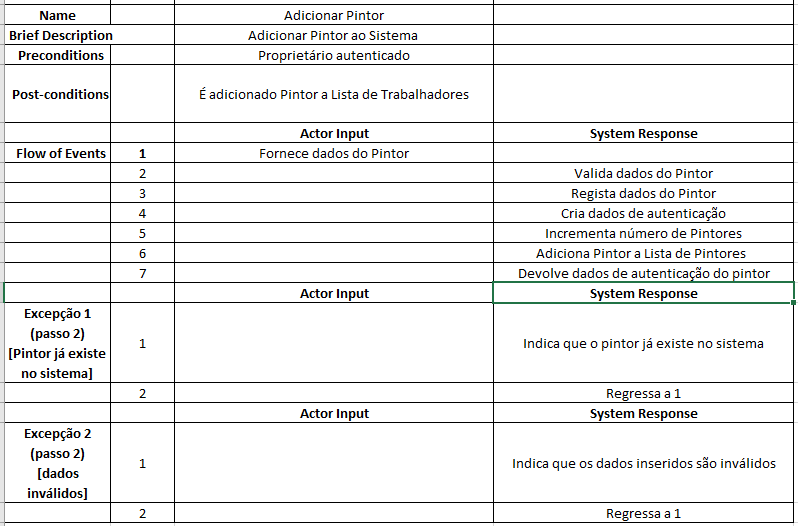


Figura 6 - Especificação de Use Case - Adicionar Pintor

* 1. Diagrama de Sequência de Use Case

Com o propósito de representar a informação descrita e detalhada acerca das funcionalidades do sistema, optamos por desenvolver, Diagramas de Sequência, permitindo assim uma melhor compreensão da iteração do utilizador e o sistema.

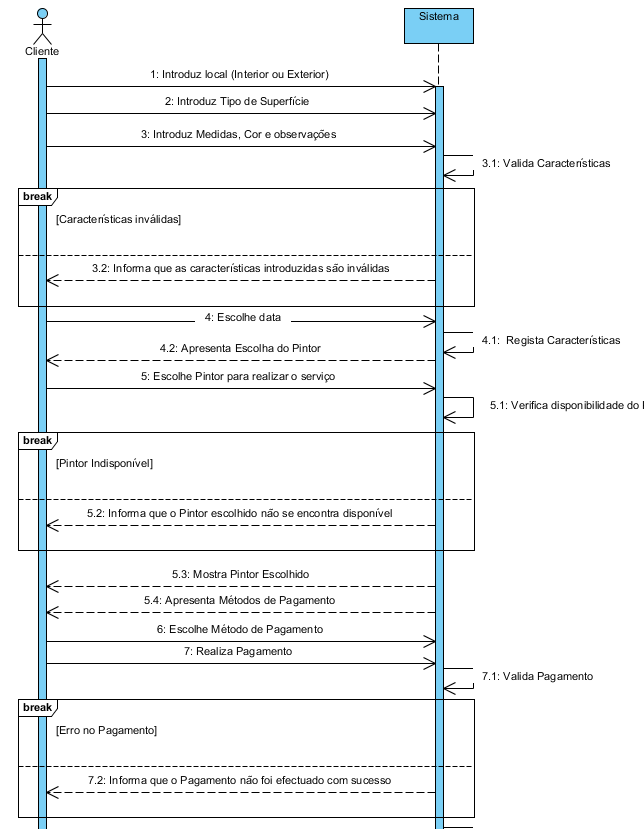
Desenvolvemos ainda Diagramas de Sequência de Subsistemas, de modo, a aproximarmos esta etapa do projeto com a próxima etapa que é o desenvolvimento da aplicação.

De forma a exemplificar o desenvolvimento deste tipo de modelo, serão apresentados alguns exemplos, bem como outros relevantes em anexo.

* + 1. Exemplo de Diagrama de Sequência – Marcar Serviço

De forma a completar o exemplo fornecido anteriormente sobre a descrição do Use Case “Marcar Serviço”, apresentamos o seu diagrama de Sequência.

Este Use Case é relativo ao Cliente logo, neste diagrama o ator será o cliente que vai usar a aplicação. Como podemos verificar um cliente para poder marcar um serviço tem de introduzir as características do serviço, na qual o sistema antes de avançar para o próximo passo verifica se as características introduzidas, estão de acordo. O cliente depois do sistema registar as características terá de escolher a data para o serviço ser efetuado, posto isto escolhe o pintor para realizar o serviço, o sistema nesta fase valida esta escolha verificando a disponibilidade do pintor escolhido. Na fase final o sistema apresenta os métodos de pagamento disponíveis, forçando o cliente a escolher e a realizar o pagamento, o sistema aguarda que o pagamento seja efetuado e valida o pagamento, posto isto emite e envia a fatura e notifica o pintor em causa, terminando assim o processo de marcação.



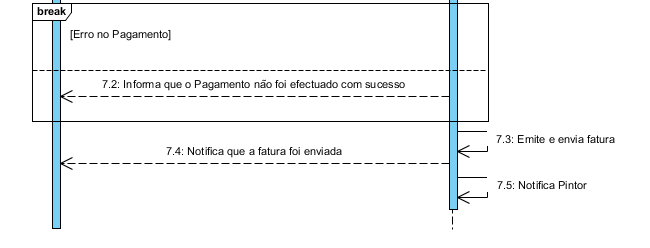


Figura 7 - Diagrama de Sequência - Marcar Serviço

* + 1. Exemplo de Diagrama de Sequência – Adicionar Pintor

Tal como anteriormente, pretendemos apresentar o diagrama de Sequência desta vez relativo ao Use Case “Adicionar Pintor”.

Este Use Case é relativo ao Proprietário, por isso, temos como ator o Proprietário. Para poder adicionar novo Pintor, o ator tem de fornecer os dados dele. Estes dados têm de ser validados, se os dados passarem na validação, o sistema regista os dados do Pintor, cria os dados de autenticação, adiciona o pintor a lista de Pintores e devolve os dados de autenticação ao ator.

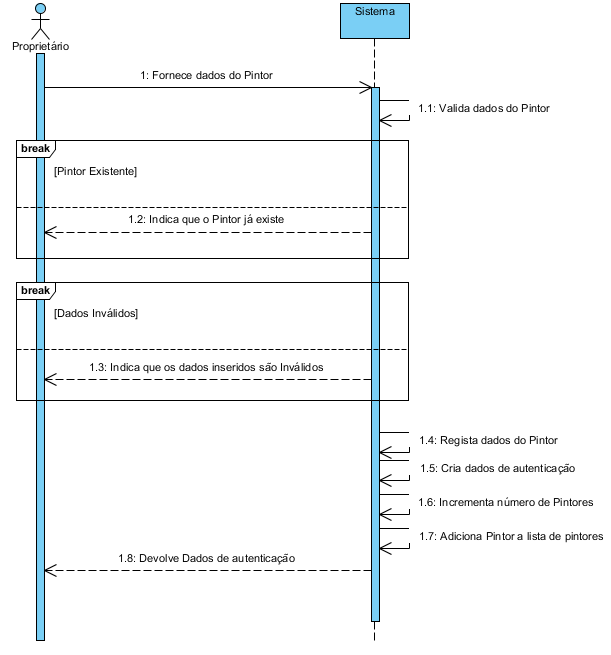


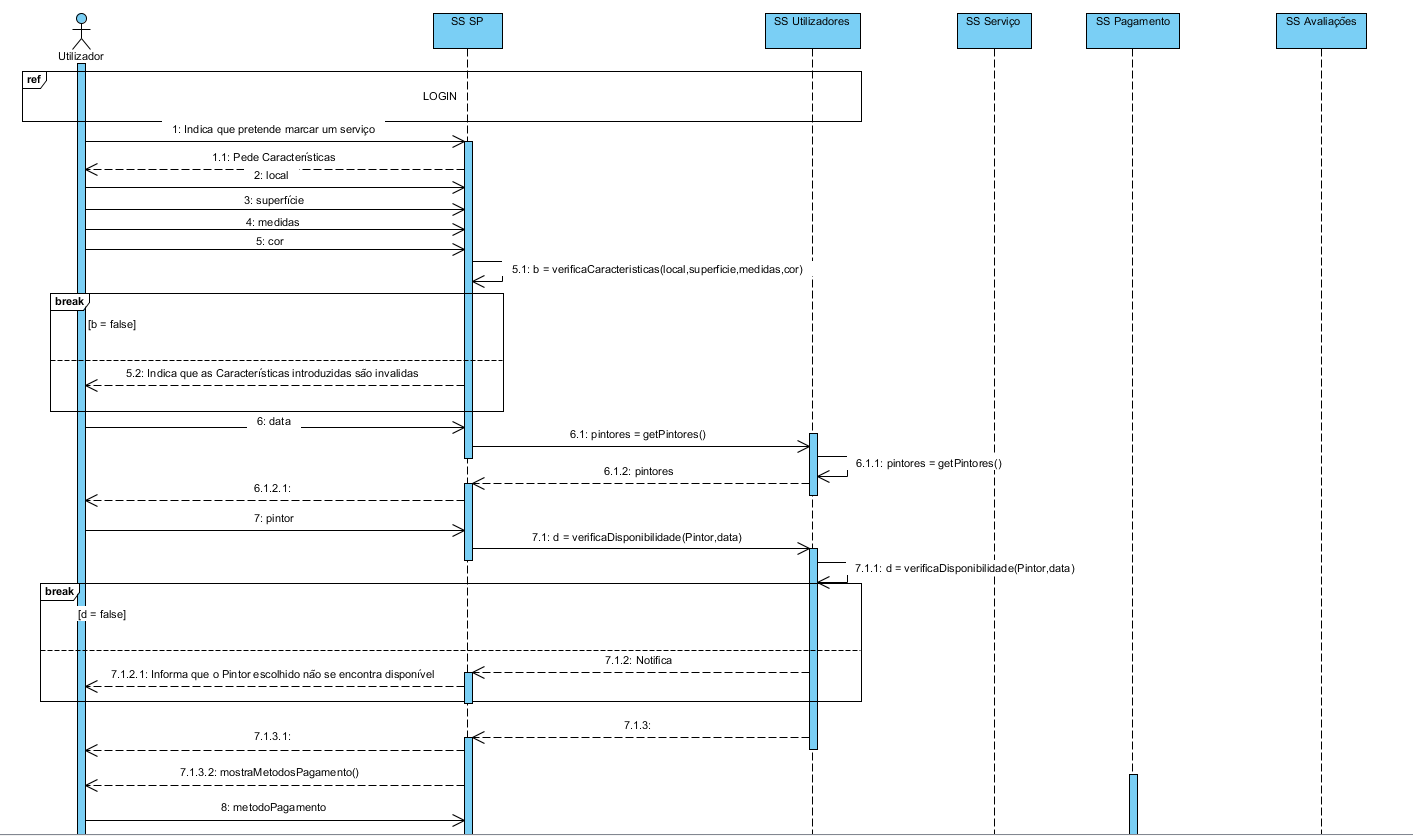
Figura 8 - Diagrama de Sequencia - Adicionar Pintor

* 1. Diagrama de Sequência de Subsistemas

Como explicado anteriormente, de modo, a aproximarmos esta fase de modulação com a fase de implementação desenvolvemos diagramas de Sequência de Subsistemas. Para tal tivemos que inicialmente decidir quais iriam ser os subsistemas a ter em conta. Portanto os subsistemas que consideramos foram os seguintes: subsistema **SP** relativo a App, vai receber todos os pedidos do utilizador e como tudo o que ele coloca na app; subsistema **Utilizadores**, possui toda a informação relativa aos utilizadores; subsistema **Serviço**, tudo o que é relativo aos serviços encontra-se neste subsistema; subsistema **Pagamento**, possui toda a informação sobre os pagamentos; e por último subsistema **Avaliações**, possui toda a informação acerca das avaliações efetuadas. Para exemplificarmos esta situação resolvemos exemplificar com dois Use Cases já apresentados, isto é, com o “Marcar Serviço” e o “Adicionar Pintor”.

* + 1. Exemplo de Diagrama de Sequência de Subsistemas – Marcar Serviço

No inicio do diagrama precisamos de ter a referência, ao “Login” devido a pré-condição desta funcionalidade que exige que o Cliente esteja devidamente autenticado. O processo de marcação começa com o pedido efetuado ao subsistema **SP**, este pede as características do serviço, faz uma verificação as características, indicando que estão erradas em caso de erro. Posto isto o cliente insere uma data, e o subsistema **SP** comunica com o subsistema de **Utilizadores**, de modo, a obter a lista de Pintores para apresentar ao Cliente e este escolher o Pintor, depois de escolhido o Pintor o subsistema **SP** tem de pedir ao subsistema **Utilizadores** que verifique a disponibilidade do Pintor para a data escolhido, em caso de sucesso é apresentado os métodos de pagamento ao Cliente, o Cliente escolhe e realiza o pagamento, o subsistema **SP** comunica com o subsistema **Pagamento**, para validar o pagamento. Este subsistema em caso de pagamento efetuado com sucesso emite e envia a fatura notificando **SP** que por sua vez notifica o cliente, terminando assim o processo de marcação.



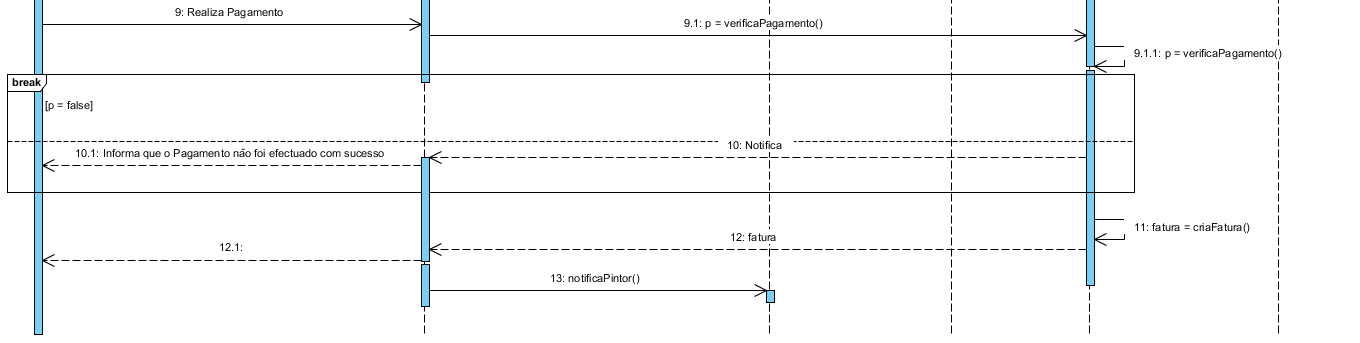


Figura 9 - DSS - Marcar Serviço

* + 1. Exemplo de Diagrama de Sequência de Subsistemas – Adicionar Pintor

Como foi dito anteriormente, esta funcionalidade exige que o Proprietário esteja devidamente autenticado, por isso é que como podemos observar temos uma referência ao “Login” no inicio do diagrama. O utilizador começa o procedimento pedindo para adicionar um pintor ao subsistema **SP**, este subsistema pede as informações do pintor, depois de dadas estas informações o subsistema **SP** recorre ao subsistema **Utilizadores** de modo a saber se os dados do utilizador são válidos, ou seja, se estão de acordo com o pretendido e se não são referentes a um pintor já existente. Posto isto, neste subsistema **Utilizadores** é adicionado Pintor à lista de Pintores e criado os dados de autenticação. Enviando estes dados para o subsistema **SP** que envia ao Proprietário terminando assim o processo.

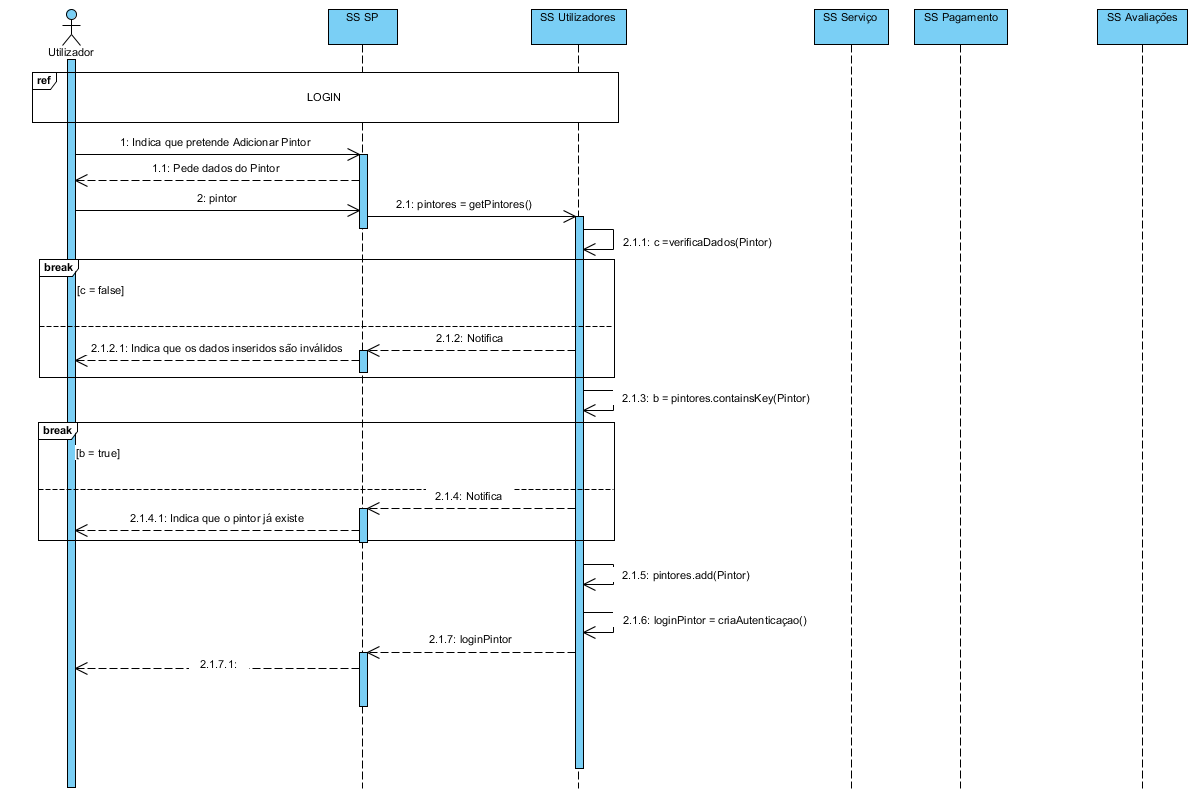


Figura 10 - DSS - Adicionar Pintor

* 1. Diagrama de Classes

Por fim, é apresentado o diagrama de classes do sistema. Este diagrama contém as classes que irão fazer parte da implementação da aplicação, bem como algumas variáveis de maior importância.

Foi modulado dois diagramas separadamente, um para o Front Office e outro para o Back Office, dado que, estes vão ter funcionalidades distintas, enquanto o Front Office permite a interação do pintor e do cliente com o sistema, o Back Office apenas permite ao proprietário realizar a gestão da aplicação.

Para cada modelo, com base nos requisitos fornecidos, foram ainda identificados alguns métodos que irão fazer parte do código de implementação da aplicação, de modo a que, o resultado final da aplicação seja o pretendido e cumpra todos os requisitos propostos inicialmente.

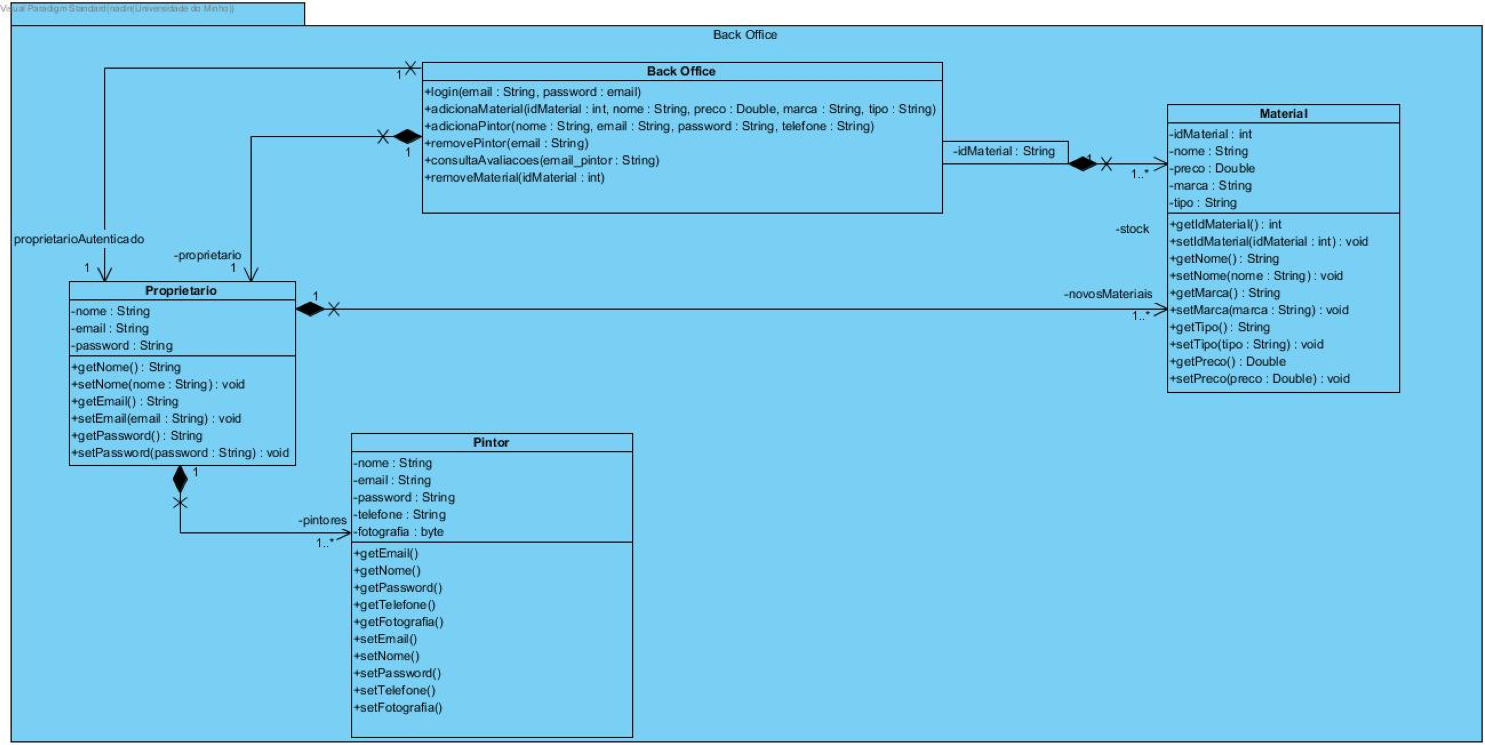


Figura 11 - Diagrama de Classes de Back Office

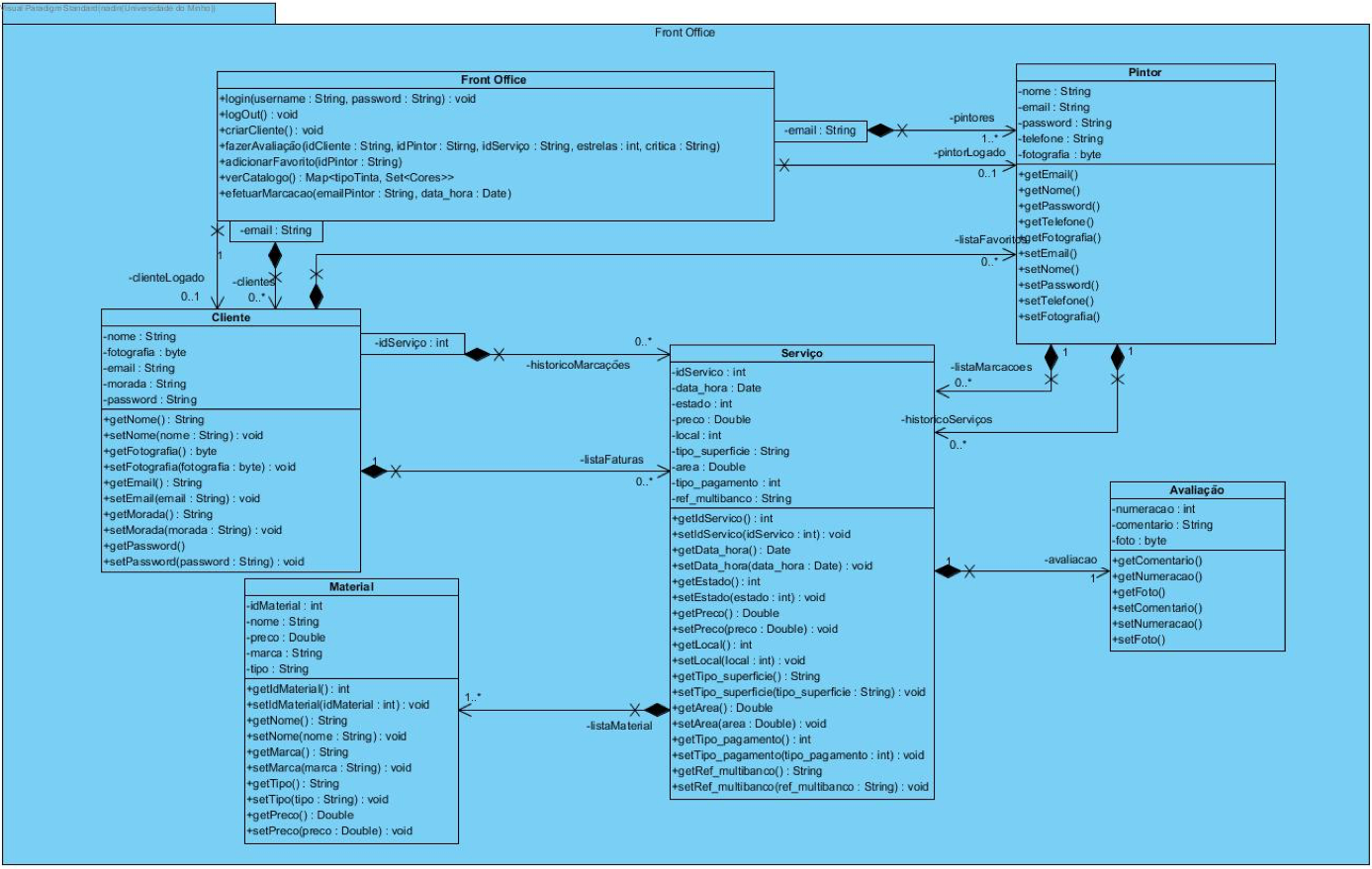


Figura 12 - Diagrama de Classes de Front Office

1. Base de Dados
   1. Modelo Concetual
      1. Tipos de entidades

De forma a conseguirmos identificar as diferentes entidades que o sistema engloba, foi necessário determinar quais os objetos que se enquadram nesta definição. Para isso, após a leitura e análise de requisitos, foram identificadas três entidades: cliente, avaliação e estabelecimento.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome da Entidade** | **Descrição** | **Sinónimos** | **Ocorrências** |
| **Cliente** | Cliente que pretende usar a aplicação para marcar um serviço desejado; | Utilizador, comprador | Um cliente pode solicitar ou marcar um serviço, introduzindo todos os dados necessários para a realização do mesmo; |
| **Serviço** | Serviço requisitado por um cliente, e efetuado por um pintor; | Trabalho | Qualquer cliente registado pode requerer um serviço; |
| **Pintor** | Pintor realiza os serviços que lhe são destacados; | Trabalhador | Um pintor pode visualizar todos os dados de um serviço; O proprietário pode adicionar pintores ao sistema; |
| **Material** | Material que é utilizado nos serviços; | Tinta, Cor; | O proprietário pode alterar os materiais do sistema, pode adicionar, remover ou alterar o preço; |

Tabela 1 - Tabela das Entidades

* + 1. Tipos de relacionamentos

Após identificadas as entidades do nosso SBD, é necessário detetar todos os relacionamentos existentes entre as mesmas. A leitura e análise dos requisitos permite identificar estes relacionamentos que as entidades estabelecem entre sim, bem como a sua respetiva cardinalidade.

Desta forma, apresentamos na seguinte tabela todos os relacionamentos entre entidades que foram identificados.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nome da Entidade** | **Multiplicidade** | **Relacionamento** | **Multiplicidade** | **Nome da Entidade** |
| **Cliente** | 1 | Solicita | N | Serviço |
| **Pintor** | 1 | Realiza | N | Serviço |
| **Serviço** | N | Utiliza | N | Material |

Tabela 2 - Tabela dos Relacionamentos

* + 1. Associar atributos com os tipos de entidades e relacionamentos

De seguida, será abordada a associação entre a informação que conseguimos reter sobre as diferentes entidades. Para isso, foi necessário detetar todos os atributos necessários para representar cada entidade, bem como identificar o seu respetivo tipo.

* **Atributos simples/compostos:**

Após a leitura dos requisitos, chegou-se à conclusão que existe um atributo da entidade serviço com a definição de atributo composto: Avaliações. Na entidade Cliente existe outro atributo composto, neste caso: Lista de Favoritos. No caso das avaliações, esta é composta por um número identificador, um comentário, uma numeração e uma fotografia. Por outro lado, a Lista de Favoritos é composta pelo e-mail do pintor que da entrada na lista.

* **Atributos derivados:**

Após o levantamento de requisitos e identificação dos atributos, chegou-se à conclusão que não existem atributos derivados no nosso sistema.

* **Atributos Multi-Valor:**

Da mesma análise referida nos pontos anteriores, é possível concluir que o atributo Lista de favoritos da entidade Cliente, já anteriormente referido como atributo composto, é também atributo multivalor, uma vez que pode existir mais do que um pintor na lista de favoritos de um cliente.

* **Associação entre atributos e entidades:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Nome da Entidade** | **Atributos** | **Descrição** | **Tipo e Tamanho** | **Nulo** | **M.V** | **Derivado** | **Composto** |
| **Cliente** | E-mail | E-mail com que o cliente está registado. É o identificador do cliente. | 45 Caracteres Variaveis. | Nao | Nao | Nao | Nao |
| Password | Password que o cliente utiliza para fazer o login. | 20 Caracteres Variaveis. | Nao | Nao | Nao | Nao |
| Nome | Nome do cliente. | 45 Caracteres Variaveis. | Nao | Nao | Nao | Nao |
| Morada | Morada do cliente | 45 Caracteres Variaveis. | Nao | Nao | Nao | Nao |
| Fotografia | Fotografia do cliente, caso ele pretenda. | Fotografia | Sim | Nao | Nao | Nao |
| Telefone | Numero de telemóvel do cliente | 9 Digitos | Nao | Nao | Nao | Nao |
| Lista de Favoritos | Lista de Pintores que o cliente guarda como sendo os seus favoritos. |  | Sim | Sim | Nao | Sim |
| Lista de Favoritos  -> E-mail | E-mail dos pintores que o cliente selecionou como sendo os seus favoritos | 45 Caracteres Variaveis. | Nao | Nao | Nao | Nao |

Tabela 3 - Associação dos atributos da entidade Cliente

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Nome da Entidade** | **Atributos** | **Descrição** | **Tipo e Tamanho** | **Nulo** | **M.V** | **Derivado** | **Composto** |
| **Serviço** | ID | Identificador unico de cada serviço do nosso sistema | Valor inteiro positivo | Nao | Nao | Nao | Nao |
| Estado | Indentificador do estado do serviço, ou seja, se o serviço está acabado, pendente, ou por iniciar. | 1 digito | Nao | Nao | Nao | Nao |
| Local | Localização do serviço dentro da casa, isto é, se é no interior ou exterior do local. | 1 digito | Nao | Nao | Nao | Nao |
| Referência Multibanco | Referência multibanco gerada após ser solicitado o serviço | 45 Caracteres Variaveis. | Nao | Nao | Nao | Nao |
| Preço | Preço total do serviço | Valor decimal positivo | Nao | Nao | Nao | Nao |
| Data | Data do serviço | Data | Nao | Nao | Nao | Nao |
| Superficie | Tipo de superfície onde irá ser feita a intervenção. | Texto | Nao | Nao | Nao | Nao |
| Area | Tamanho da zona a intervir | Valor decimal positivo | Nao | Nao | Nao | Nao |
| Avaliações | Avaliações que o cliente pode inserir sobre o serviço prestado |  | Sim | Nao | Nao | Sim |
| Avaliações  -> Comentário | Comentário da avaliação introduzida pelo cliente | Texto | Nao | Nao | Nao | Nao |
| Avaliações  -> Numeração | Numero de 1 a 5 com que o cliente avalia o serviço | 1 Digito | Nao | Nao | Nao | Nao |
| Avaliações  -> Fotografia | Fotografia do serviço, caso o cliente assim o pretenda | Fotografia | Sim | Nao | Nao | Nao |

Tabela 4 - Associação dos atributos da entidade Serviço

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Nome da Entidade** | **Atributos** | **Descrição** | **Tipo e Tamanho** | **Nulo** | **M.V** | **Derivado** | **Composto** |
| **Pintor** | E-Mail | E-mail com que o pintor está registado no sistema. É o identificador único de cada pintor. | 45 Caracteres Variaveis. | Nao | Nao | Nao | Nao |
| Nome | Nome do pintor | 45 Caracteres Variaveis. | Nao | Nao | Nao | Nao |
| Password | Password que o pintor utiliza para fazer o login no sistema | 20 Caracteres variáveis | Nao | Nao | Nao | Nao |
| Fotografia | Fotografia do pintor, ao contrario do cliente, neste caso, é obrigatório. | Fotografia | Nao | Nao | Nao | Nao |
| Telefone | Número de telemóvel do pintor. | 9 dígitos | Nao | Nao | Nao | Nao |

Tabela 5 - Associação dos atributos da entidade Pintor

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Nome da Entidade** | **Atributos** | **Descrição** | **Tipo e Tamanho** | **Nulo** | **M.V** | **Derivado** | **Composto** |
| **Material** | ID | Identificador unico da tinta | Valor inteiro positivo | Nao | Nao | Nao | Nao |
| Nome | Nome da tinta | 45 Caracteres Variaveis. | Nao | Nao | Nao | Nao |
| Tipo | Tipo de paredes onde é utilizada a tinta | 45 Caracteres Variaveis. | Nao | Nao | Nao | Nao |
| Marca | Marca da tinta | 20 Caracteres Variaveis. | Nao | Nao | Nao | Nao |
| Preço | Preço da tinta | Valor decimal positivo | Nao | Nao | Nao | Nao |

Tabela 6 - Associação dos atributos da entidade Material

* + 1. Determinar domínio dos atributos

Seguidamente, são descritos os domínios dos atributos referentes às diferentes entidades. O domínio consiste num conjunto de valores que pertencem a um determinado tipo, e que pode ser atribuído a cada atributo.

Entidade: Cliente

* Nome: Nome completo do cliente. É uma *string* com 45 caracteres variáveis;
* E-mail: E-mail com que o cliente está registado. É uma *string* com 45 caracteres variáveis.
* Password: Password com que o cliente está registado no sistema. Utiliza-a cada vez que pretende fazer login. É uma *string* com 20 caracteres variáveis;
* Morada: Morada completa do cliente. É uma *string* com 45 caracteres variáveis;
* Telemóvel: Número de telemóvel do cliente. É um número composto por 9 dígitos;
* Fotografia: Fotografia do cliente, se assim ele o pretender. É do tipo DATA;
* Lista de Favoritos: Lista de pintores que o cliente seleciona como favoritos. Cada elemento é composto pelo e-mail do pintor;

Entidade: Serviço

* ID\_Serviço: Número identificador do serviço. É um valor inteiro positivo;
* Estado: É o estado em que o serviço se encontra no momento, isto é, se esta pendente, por terminar, ou acabado. É um dígito que indicará esses três estados, sendo que serão, respetivamente, 0,1 ou 2.
* Local: Local da morada do cliente onde vai ser feito o serviço, isto é, se é no interior ou exterior da casa. Assim como no caso em cima, é também um dígito, sendo que o 0 representa o interior, e o 1 o exterior;
* Referência Multibanco: Referência Multibanco que é gerada assim que o serviço é confirmado. É uma *string* com 45 caracteres variáveis;
* Preço: Preço final do serviço. É um valor decimal positivo;
* Data: Data da realização do serviço. É um valor do tipo data;
* Superfície: Tipo da superfície onde vai ser feito o serviço. É representado por uma *string*.
* Área: Valor da área que vai ser intervencionada. É um valor decimal positivo.
* Avaliações: As avaliações efetuadas aos serviços são compostas por:
  + Comentário: É um valor do tipo texto de 140 caracteres.
  + Numeração: É um valor inteiro positivo de 1 a 5.
  + Fotografia: É um valor pertencente ao tipo *DATA.*

Entidade: Pintor

* E-Mail: E-mail com que o pintor está registado no sistema, neste caso, é o seu identificador único. É uma *string* com 45 caracteres variáveis;
* Nome: Nome completo do pintor. É uma *string* com 45 caracteres variáveis;
* Password: Conjunto de caracteres com que o pintor faz o login no sistema. É uma *string* com 20 caracteres variáveis;
* Fotografia: É um valor pertencente ao tipo *DATA.*
* Telefone: É um número composto por 9 dígitos;

Entidade: Material

* ID\_Material: Número que identifica o material. É um valor inteiro positivo;
* Nome: Nome completo da tinta. É uma *string* com 45 caracteres variáveis;
* Tipo: Tipo de superfície onde a tinta pode ser aplicada. É uma *string* com 45 caracteres variáveis;
* Marca: Nome da marca da tinta. É uma *string* com 20 caracteres variáveis;
* Preço: Preço da tinta. É um valor decimal positivo;
  + 1. Determinar chaves primárias, candidatas e alternativas

De forma a identificar unicamente cada uma das ocorrências, foi necessário determinar os atributos chave.

Deste modo, uma vez de cada uma das entidades contém um identificador ou um e-mail estes atributos foram considerado como chave primária. Assim, é possível identificar cada um dos registos de uma entidade através do seu identificador ou do seu e-mail, uma vez que este é único.

De seguida apresentamos as chaves primárias que consideramos para as entidades identificadas anteriormente:

* Cliente: E-Mail\_Cliente
* Serviço: ID\_Serviço
* Pintor: E-Mail\_Pintor
* Material: ID\_Material
  + 1. Desenho do Diagrama E-R

Apresentamos, de seguida, o desenho do diagrama E-R (Entidade-Relacionamento) de forma a representar conceptualmente as relações entre as entidades da base de dados.

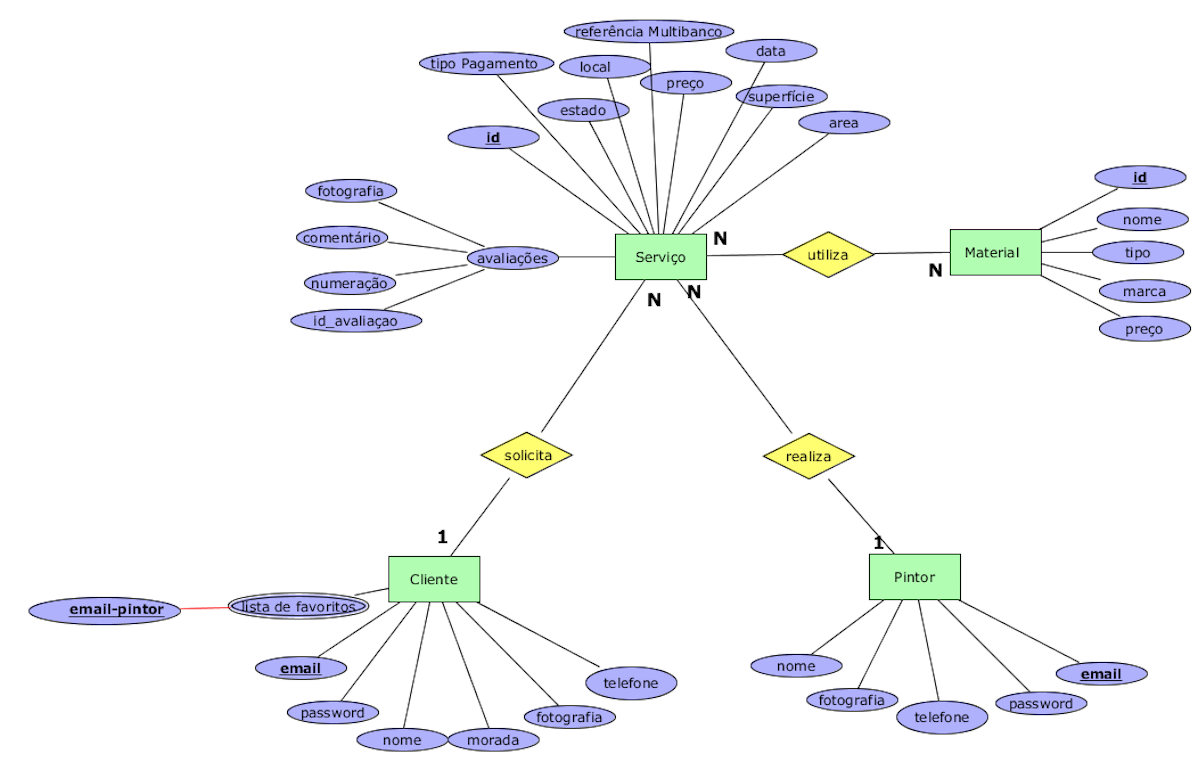


Figura 13 - Diagrama E-R

* + 1. Revisão do modelo de dados com o utilizador

Após a realização do modelo conceptual, este foi revisto pelo utilizador. Nenhum problema foi detetado e por isso o modelo de dados foi aceite.

* 1. Modelo Lógico

Nesta secção irá ser demonstrado como iremos transformar o modelo conceptual construído no modelo lógico, seguindo várias etapas até ao modelo final.

* + 1. Entidades

Sendo que uma entidade forte trata-se duma entidade em que a sua existência não depende de outra, todas as quatro entidades presentes no modelo conceptual – Cliente, Serviço, Pintor e Material– estão de acordo com esta norma e irão dar origem a tabelas.

* + 1. Relacionamentos

No modelo concetual estão representadas três relações, sendo elas todas de 1:N. Estas relações foram mantidas no modelo lógico através da inclusão de chaves estrangeiras em certas tabelas como vai ser mostrado a seguir.

* **1 Cliente solicita N Serviços:**

Através da inclusão da **chave estrangeira** **e-mail\_cliente** – que é chave primária na tabela cliente – na tabela **Serviços** é possível manter este relacionamento presente no modelo concetual.

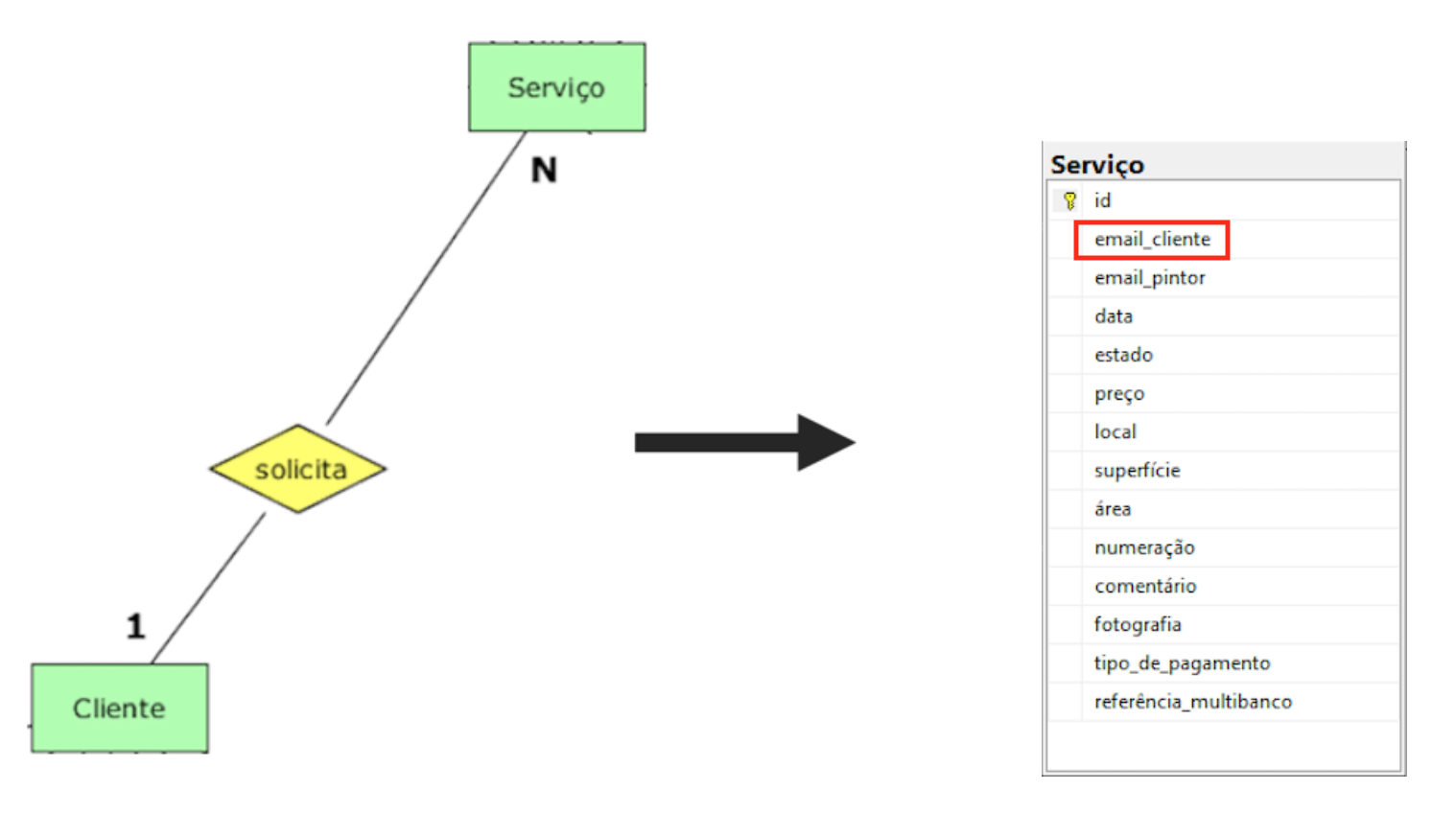


Figura 14 - Transformação do relacionamento Solicita em tabela.

* **1 Pintor Realiza N Serviços:**

Através da inclusão da **chave estrangeira** **e-mail\_pintor** – que é chave primária na tabela pintor – na tabela **Serviços** é possível manter este relacionamento presente no modelo concetual.

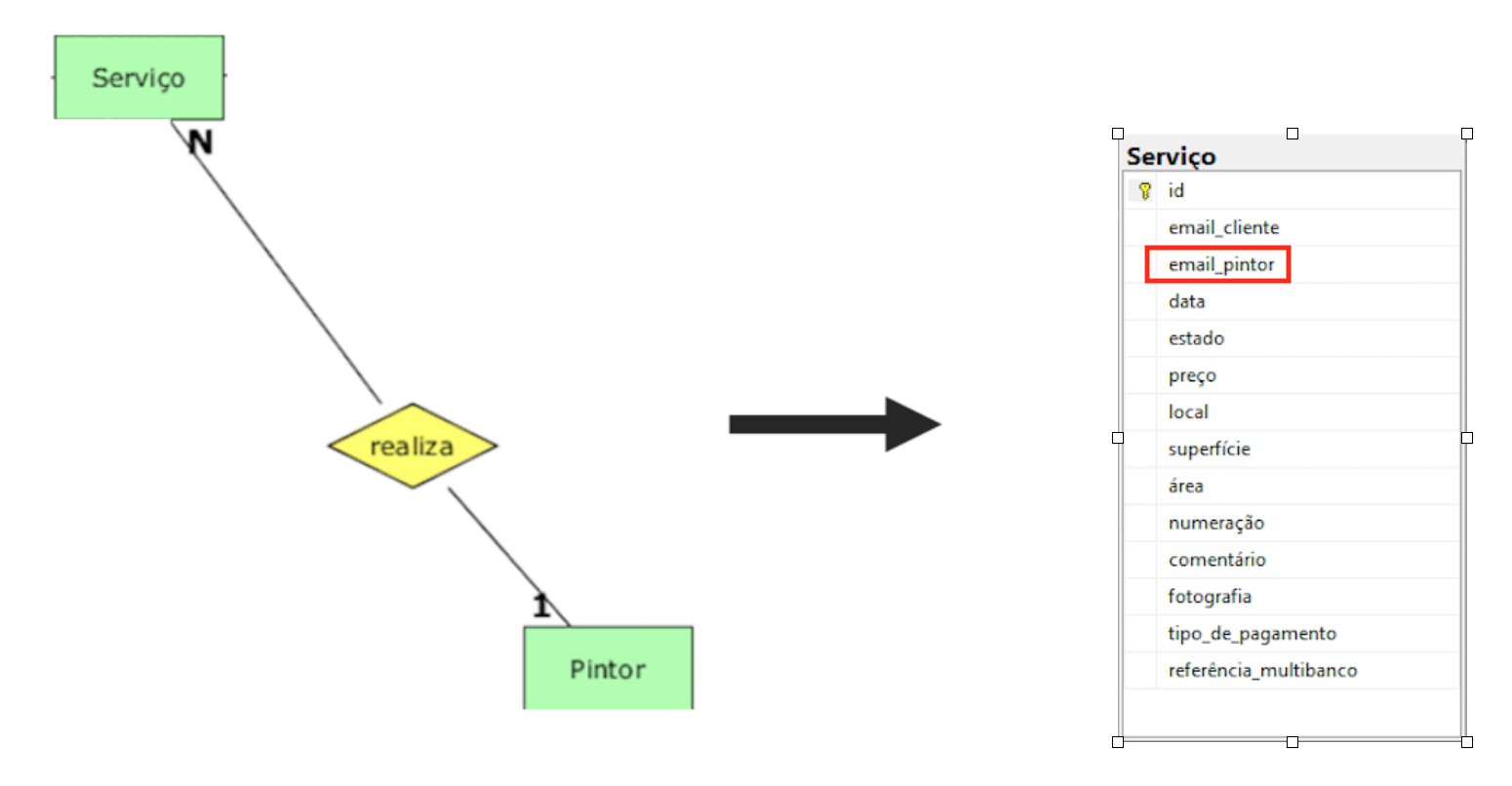


Figura 15 - Transformação do relacionamento Realiza em tabela.

* **N Material é utilizado em N Serviços:**

Através da inclusão das **chaves estrangeiras** **id\_serviço e id\_material** – que são chaves primárias na tabela serviço e na tabela material – criando a nova tabela **Serviço\_Material** é possível manter este relacionamento presente no modelo concetual.

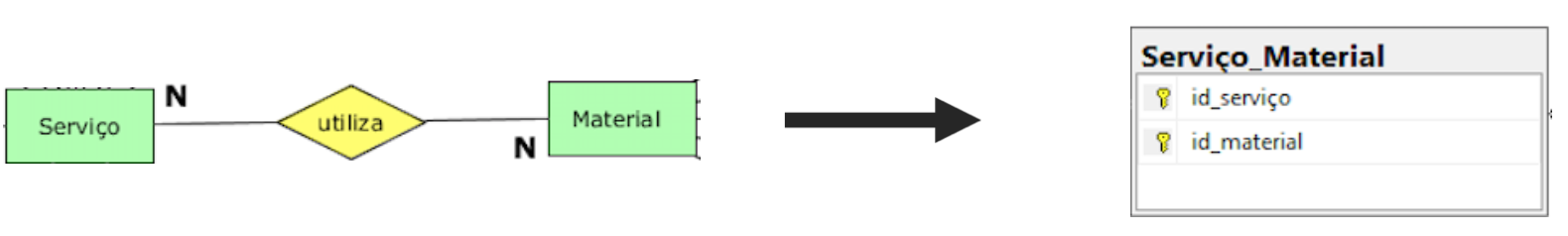


Figura 16 - Transformação do relacionamento Utiliza em tabela.

* + 1. Atributos Multi-Valor

No modelo conceptual está representado um atributo multivalor, que é ao mesmo tempo um atributo composto multivalor. De modo a representar no modelo lógico, foi criada uma nova tabela para esse atributo.

A entidade Cliente contem o atributo multivalor ‘Lista de Favoritos’ que por sua vez, é também composto pelo atributo ‘email\_Pintor’.



Figura 17 - Transformação do atributo Multivalor Lista de Favoritos em tabela.

* + 1. Normalização

1. **1ª Forma Normal**

De modo a uma tabela estar de acordo com a 1ª forma normal, é necessário que uma chave não contenha várias ocorrências do mesmo atributo.

Ora, de acordo com uma análise ao modelo atualmente contruído é possível constatar que em todas as tabelas não existe a ocorrência de grupos repetidos, estando assim, de acordo com a primeira forma normal.

1. **2ª Forma Normal**

Como anteriormente foi provado, o modelo encontra-se de acordo com 1ª forma normal, que é o primeiro requisito necessário para estar de cordo com a segunda.

Outra condição necessária a segunda forma normal é que todas as chaves não primárias sejam dependentes da chave primária – algo que, através da observação das tabelas existentes, é também possível de confirmar. Existem, no entanto, chaves que poderiam por si mesmas identificar uma tabela, mas que, no entanto, foram anteriormente consideradas chaves candidatas, o que as torna também dependentes.

1. **3º Forma Normal**

Finalmente, para uma tabela encontrar-se de acordo com a terceira normal, é necessário que cumpra também a segunda.

De seguida verifica-se se alguma das colunas existentes nessa tabela é dependente de outra(s), algo que não acontece em nenhuma das tabelas existentes no modelo lógico construído.

* + 1. Elaboração e validação do esquema lógico da base de dados

Após a realização do modelo lógico e da validação feita anteriormente, conclui-se que este se encontra corretamente elaborado e validado. Como resultado final obtemos o seguinte:

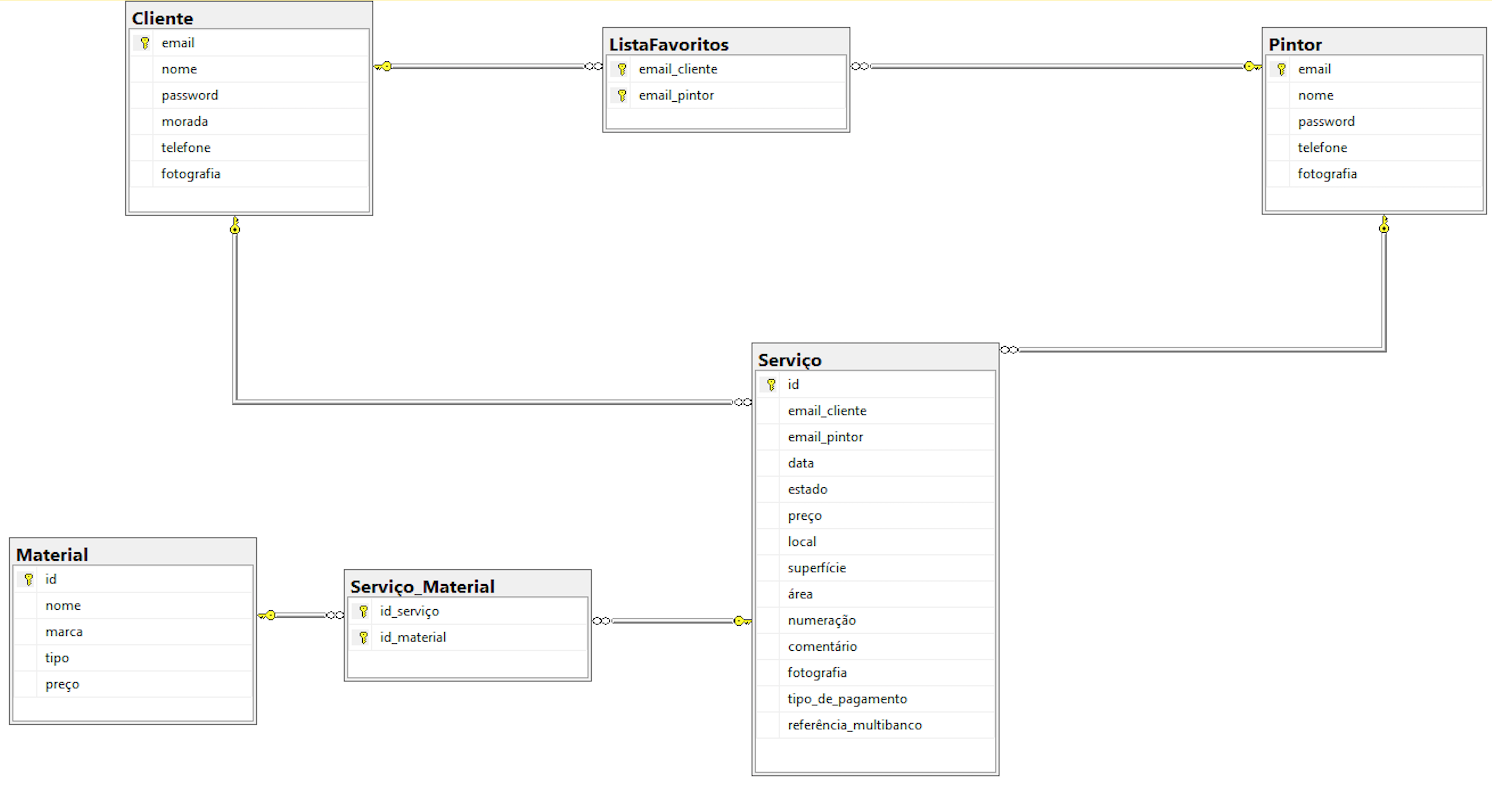


Figura 18 - Modelo Lógico

* + 1. Análise do crescimento futuro

Embora a atual base de dados tenha sido criada para satisfazer as necessidades anteriormente mencionadas, é bastante flexível e pronta para trabalhar com alterações que sejam pretendidas. Por exemplo, se o proprietário pretender adicionar outro tipo de trabalhador ou de material ao sistema, facilmente seriam criadas tabelas para esse mesmo efeito, não afetando de qualquer maneira os dados atualmente existentes.

* + 1. Revisão do modelo lógico com o utilizador

Após a realização do modelo lógico, este foi revisto pelo utilizador com o intuito de assegurar que este o considera uma representação fidedigna do que é pretendido. Nenhum problema foi detetado e por isso o modelo de dados foi aceite.

1. Mockups
2. Conclusões e Trabalho Futuro

Terminada a primeira etapa do projeto, damos por concluída a fase da fundamentação do projeto. Inicialmente, realizamos uma recolha e análise de informações onde identificamos e compreendemos as principais necessidades dos utilizadores. Esta recolha de informação teve por base as opiniões dos utilizadores e trabalhadores, onde identificamos as suas respetivas lacunas que o nosso sistema deve conseguir corrigir. Com este conhecimento fundamentamos o caso de estudo, as motivações e objetivos da aplicação a implementar. Esta compreensão detalhada de como se comporta o sistema permite uma idealização e implementação mais eficazes.

Com o planeamento das atividades, foi possível ter uma melhor perceção das tarefas a realizar no futuro. Por outro lado, a distribuição de tarefas pelo grupo de trabalho e a previsão de períodos de tempo para as várias tarefas permitiu uma melhor organização da construção do projeto.

O passo seguinte para a elaboração do projeto consistiu na especificação do sistema, onde foram levantados os vários tipos de requisitos e elaborada a modelação do sistema, através do uso de diagramas em UML.

De modo a uma melhor estruturação dos diagramas, foi realizada um levantamento de requisitos. Assim, procedeu-se à devida separação dos mesmos em requisitos de utilizador e de sistema, sendo estes últimos divididos ainda em requisitos funcionais e não funcionais. Esta organização permitiu compreender melhor as funcionalidades que o sistema deve implementar.

Em seguida, procedeu-se à elaboração dos modelos em UML onde foram desenvolvidos vários diagramas que permitiram interpretar aspetos fundamentais que o sistema deve implementar. Deste modo, foi elaborado um diagrama de *Use Case* que permite visualizar as funcionalidades a implementar e quais os utilizadores que irão usufruir das mesmas. Os diagramas de sequência permitiram ter uma perceção dos comportamentos entre os utilizadores e o sistema, bem como o ordenamento temporal das mensagens trocadas entre estes. Por fim, o diagrama de classes foi útil para obter uma perspetiva sobre a arquitetura do sistema a desenvolver.

Após desenvolvidos os modelos, procedeu-se ao desenvolvimento da base de dados de acordo com o levantamento de requisitos. Deste modo, o primeiro passo foi identificar as entidades, os atributos e os relacionamentos, que permitiram a construção do modelo conceptual. Em seguida, procedeu-se à passagem para o modelo lógico, onde verificamos se o modelo evitava redundâncias.

Por último, foram contruídos os *mockups* que permitem ter uma perceção da *interface* que o sistema irá respeitar.

A terceira fase do projeto consiste na implementação da aplicação nos testes que nos permitem verificar a sua operacionalidade.

Referências

**Jornal de Negócios -** http://www.jornaldenegocios.pt/economia/autarquias/detalhe/braga\_entre\_as\_cidades\_com\_mais\_qualidade\_de\_vida\_da\_europa;

**Blasting News** - https://pt.blastingnews.com/braga/2015/07/saiba-porque-economia-de-braga-e-o-exemplo-a-seguir-pelo-pais-00483109.html;

**Robbialac -** http://tintasrobbialac.pt/institucional.aspx;

**CIN -**http://www.cin.pt/portal/portal/user/anon/page/catalogos.psml?categoryOID=94908080808580GC&contentid=&nl=pt;

**Sotinco -** http://www.sotinco.pt/catalogos;

Lista de Siglas e Acrónimos

**API** **A**plication **P**rogram **I**nterface

**GPS** **G**lobal **P**ositioning **S**ystem

**UML U**nified **M**odeling **L**anguage

**IDE** On-Line Analytical Processing

**SBD****S**istema de **B**ase de **D**ados

# Anexos