



# FoodFinder

Design e Desenvolvimento de uma Aplicação Web  
Etapa 1

David Ferreira, 68222  
André Mendes, 68228  
João Santos, 68943  
Diogo Mesquita, 69185

Licenciatura em Engenharia Informática  
Laboratório de Aplicações Web e Base de Dados

Vila Real, novembro de 2020

## Resumo

Atualmente, as aplicações *Web* estão em todo o lado e esta é uma tendência que continua a aumentar com cada vez mais plataformas, ferramentas e informação a transitar para serviços remotos. Desta maneira, este tipo de aplicação faz, cada vez mais, parte do nosso dia-a-dia e ocupam um espaço importante na vida das organizações.

Com o intuito de proceder ao *design* e implementação de uma aplicação *Web* que consiste, entre várias funcionalidades, num portal de restaurantes e dos seus pratos do dia, o presente documento dedica-se a explorar e apresentar o modelo conceptual de dados, a análise dos requisitos funcionais e a consequente especificação dos casos de uso do sistema. Este é um passo imprescindível no planeamento desta *web app* de modo a garantir que a solução final responda corretamente às necessidades do sistema a construir.

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>2</b>
<b>2</b>	<b>Classes de Utilizadores do Sistema</b>	<b>4</b>
<b>3</b>	<b>Modelo Conceptual dos Dados</b>	<b>5</b>
<b>4</b>	<b>Análise e Levantamento dos Requisitos Funcionais</b>	<b>7</b>
<b>5</b>	<b>Os Casos-de-Uso</b>	<b>9</b>
<b>6</b>	<b>Conclusão</b>	<b>12</b>
<b>7</b>	<b>Referências</b>	<b>13</b>

# Capítulo 1

## Introdução

Uma aplicação *web* permite que o processamento de informação seja instanciado remotamente através de um *browser* e executado parcialmente num servidor *web*, servidor aplicativo (*application server*) e/ou num servidor de base de dados (*database server*) (Bruno & Thom, 2005). O surgimento deste tipo de tecnologias permitiu que uma aplicação tivesse um maior número de utilizadores, que estes estivessem dispersos pelo mundo e que existisse uma forma mais fácil para partilhar informação entre diferentes utilizadores e diferentes dispositivos.

Desse modo, o estudo deste tipo de plataformas e tecnologias - tanto a nível de planeamento, *design* e implementação - afigura-se como um elemento crucial na atualidade.

Com esse intuito, vamos realizar o planeamento, *design* e implementação de uma aplicação *web* que se baseia num portal onde é possível a restaurantes registados inserirem informação sobre o seu estabelecimento e adicionarem os seus pratos do dia. Por outro lado, clientes autenticados deverão ter a capacidade de guardar restaurantes e pratos do dia na sua lista de favoritos.

Para isso, esse desenvolvimento será dividido em três fases distintas, mas cada uma necessária para a fase consequente. Este presente documento dedica-se a apresentar e a explorar a primeira fase deste projeto. Desse modo, em primeiro lugar, este começa por estabelecer as classes de utilizadores do sistema e, em seguida, o modelo conceptual dos dados, a análise e levantamento dos requisitos do sistema a construir e, por fim, os casos de

uso através da sua representação com o diagrama de casos-de-uso da família UML.

## Capítulo 2

# Classes de Utilizadores do Sistema

Neste capítulo, são apresentadas na tabela 2.1 as diferentes classes de utilizadores que irão interagir com o sistema em estudo, tal como uma breve descrição das funcionalidades às quais cada classe tem acesso.

Classe de Utilizador	Descrição
Administrador	Responsável por criar outros utilizadores administradores, aceitar pedidos de registo de restaurantes e fazer a gestão de utilizadores.
Visitante	É um utilizador não autenticado. Tem acesso a funcionalidades básicas como consultar informação dos restaurantes e consultar os seus pratos do dia.
Cliente	Para além de partilhar as funcionalidades do utilizador Visitante, ainda pode pesquisar pratos do dia por palavra-chave ou tipo, tal como adicionar pratos e restaurantes à sua lista de favoritos.
Restaurante	Para além de partilhar as funcionalidades do utilizador Visitante, pode, também, fazer a gestão dos seus pratos do dia.

*Tabela 2.1: As diferentes classes de utilizador do sistema.*

## Capítulo 3

# Modelo Conceptual dos Dados

O modelo entidade-relacionamento é utilizado para descrever níveis externos e conceituais de dados e são independentes dos aspetos físicos e internos. São úteis, pois para além de apresentarem o que deverá estar presente na base de dados, eles também fornecem vários outros detalhes como a representação explícita de objetos, atributos, relações, constrangimentos e outros (Ricardo & Urban, 2017).

No diagrama 3.1 é apresentado o diagrama entidade-relacionamento para o sistema FoodFinder. Analisando o mesmo, podemos, rapidamente, identificar quais são as entidades, e as suas propriedades, que irão estar patentes na nossa base de dados: numa primeira análise, estas serão Utilizador, Restaurante, Cliente, Administrador e Prato\_do\_Dia. Por outro lado, também podemos perceber as relações que estas entidades estabelecem entre si.

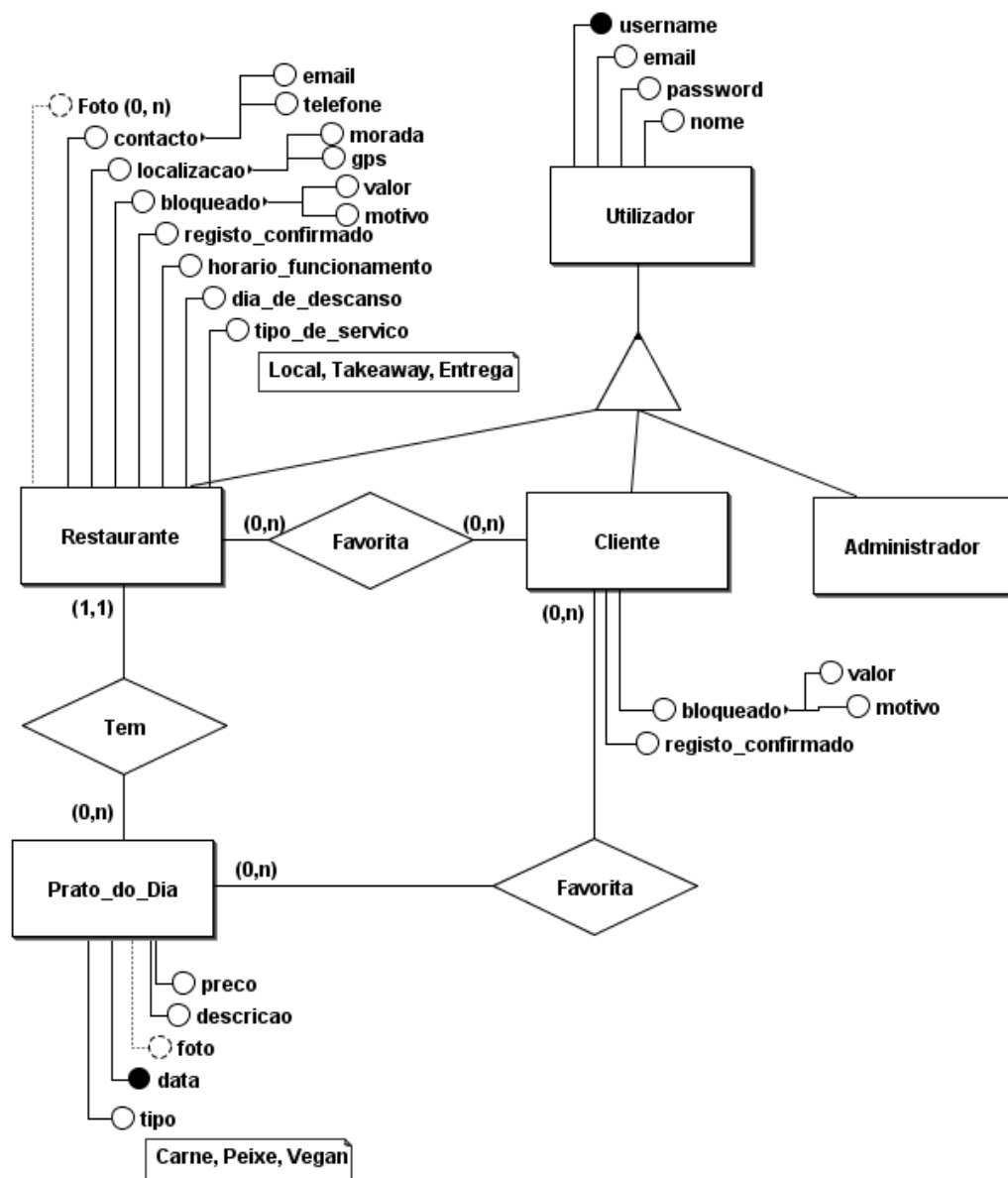


Figura 3.1: O modelo conceptual dos dados do sistema FoodFinder, através de um diagrama E-R.



## Capítulo 4

# Análise e Levantamento dos Requisitos Funcionais

Segundo Sommerville & Sawyer, os requisitos são a especificação do que deverá ser implementado (Sommerville & Sawyer, 2006). No caso dos requisitos funcionais, estes são descrições do que os desenvolvedores deverão implementar para permitir que os utilizadores executem as suas tarefas de acordo com os objetivos e o intuito do sistema.

Bell, Morrey & Pugh afirmam mesmo que o estabelecimento e a definição dos requisitos é a atividade isolada mais importante no desenvolvimento de *software*, pois se não fomos capaz de corretamente especificar o que é preciso, é inútil implementa-lo (Bell, Morrey & Pugh, 1992).

Desse modo, em seguida, são apresentados os requisitos funcionais que deverão ser implementados na construção do sistema em estudo.

1. O sistema FoodFinder deverá permitir o registo dos clientes.
2. O sistema FoodFinder deverá permitir o registo dos restaurantes.
3. O sistema FoodFinder deverá permitir a criação de Administradores.
4. O sistema FoodFinder deverá permitir a inserção de informação do utilizador.

5. O sistema FoodFinder deverá enviar uma mensagem para confirmação do registo do cliente.
6. O sistema FoodFinder deverá permitir aceitar pedidos de registo de restaurantes.
7. O sistema FoodFinder deverá permitir o bloqueio dos utilizadores.
8. O sistema FoodFinder deverá permitir a autenticação do utilizador.
9. O sistema FoodFinder deverá informar motivo do bloqueio a quando autenticação.
10. O sistema FoodFinder deverá permitir realizar a gestão dos utilizadores.
11. O sistema FoodFinder deverá permitir a gestão de pratos em destaque no sistema.
12. O sistema FoodFinder deverá permitir a inserção de um prato do dia.
13. O sistema FoodFinder deverá manter um histórico dos pratos inseridos.
14. O sistema FoodFinder deverá permitir reutilizar pratos anteriores.
15. O sistema FoodFinder deverá permitir a consulta de pratos do dia.
16. O sistema FoodFinder deverá permitir a consulta de restaurantes.
17. O sistema FoodFinder deverá permitir pesquisar um restaurante através de proximidades geográficas.
18. O sistema FoodFinder deverá permitir pesquisar um prato do dia por *keyword*.
19. O sistema FoodFinder deverá permitir adicionar à lista de favoritos um prato do dia.
20. O sistema FoodFinder deverá permitir adicionar à lista de favoritos um restaurante.
21. O sistema FoodFinder deverá enviar notificação quando prato do dia favorito encontra-se disponível.

# Capítulo 5

## Os Casos-de-Uso

Os casos de uso são as técnicas para capturar os requisitos funcionais de um sistema, descrevendo as interações típicas entre os utilizadores de um sistema e o próprio sistema (Fowler, 2004).

Assim, com base nos requisitos que foram anteriormente apresentados, no diagrama 5.1 estão representados os casos-de-uso que são comuns aos diferentes atores do sistema. Por outro lado, nos diagramas 5.3, 5.1, 5.2 são apresentados os casos-de-uso específicos e particulares a cada um dos atores.

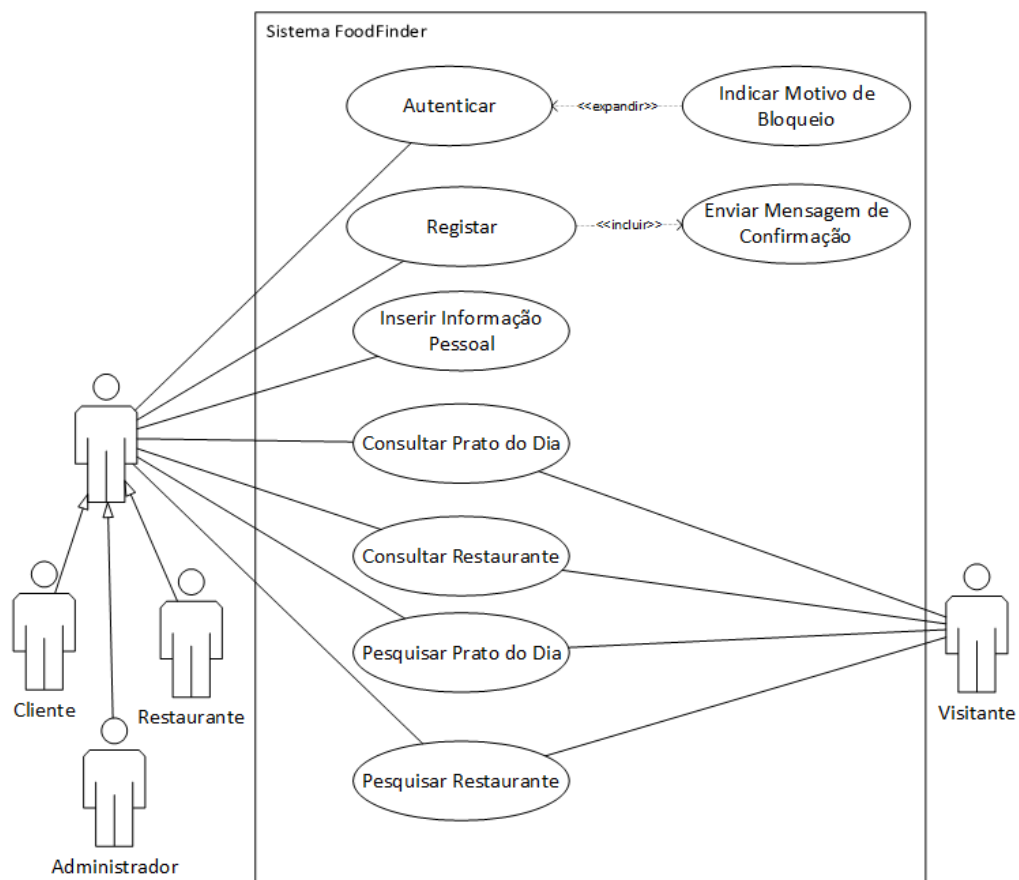


Figura 5.1: Diagrama que apresenta os casos-de-uso comuns aos vários atores do sistema FoodFinder.

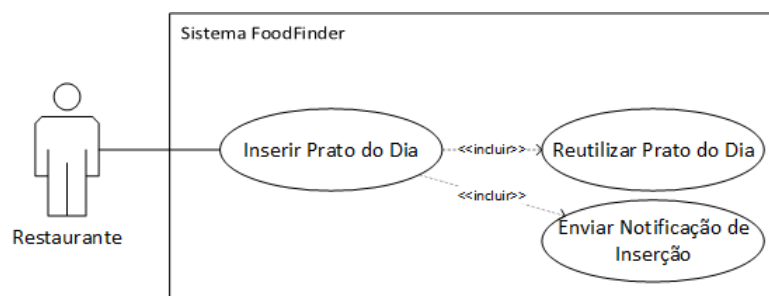


Figura 5.2: O diagrama dos casos-de-uso do sistema FoodFinder relativos ao ator Restaurante.

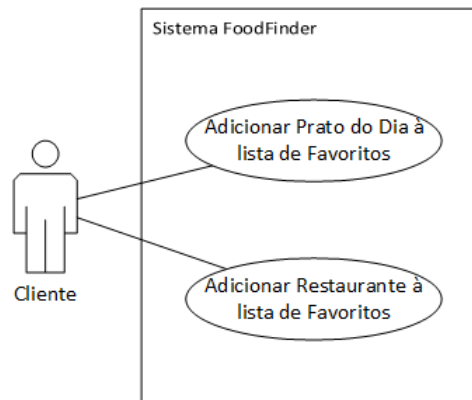


Figura 5.3: O diagrama dos casos-de-uso do sistema FoodFinder relativos ao ator Cliente.

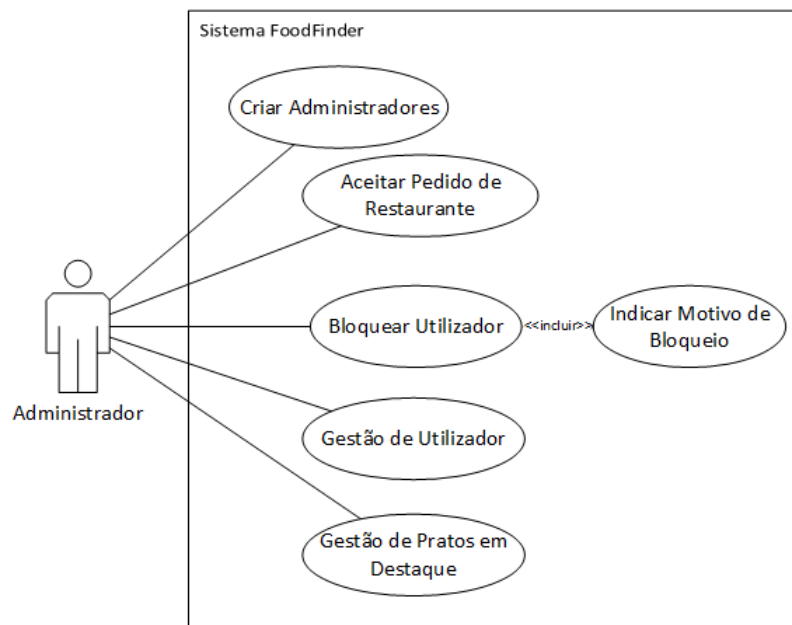


Figura 5.4: O diagrama dos casos-de-uso do sistema FoodFinder relativos ao ator Administrador.

# Capítulo 6

## Conclusão

O grande crescimento no uso de aplicações *web* e o aumento da sua complexidade exige que mais atenções sejam feitas no que toca ao seu planeamento e *design*.

O estudo desenvolvido nesta fase do trabalho prático, permitiu construir as bases nas quais sustentarão o planeamento adicional e a própria implementação do sistema a construir. Com o modelo conceptual dos dados temos uma primeira visão sobre a base de dados que suportará o sistema, incluindo as entidades e relações que dela farão parte. Com o levantamento e análise de requisitos, tal como a construção e definição dos casos-de-uso, temos uma boa descrição para os diferentes utilizadores que farão uso do sistema e de que forma eles interagirão com ele.

A próxima fase residirá no mapeamento do modelo concetual dos dados para o modelo relacional e a consequente implementação do modelo físico da base de dados. Da mesma forma, também serão especificadas as interfaces de *backoffice* e *frontoffice* recorrendo a *mockups*.

# Referências

Bell, D., Morrey, I. & Pugh, J. (2006). Software Engineering: A programming approach. Prentice Hall.

Bruno, V. & Thom, J. (2005). Characteristics of Web applications that affect usability: A review.

Fowler, M. (2004). UML Distilled: a brief guide to the standard object modeling language. Addison-Wesley Professional.

Ricardo, C. M., & Urban, S. D. (2017). Database Processing: Fundamentals, Design and Implementation. Pearson.

Sommerville, I. & Sawyer, P. (2006). Requirements Engineering: a good Practice Guide. John Wiley.