**Interface Layer**

1. **Input**
   1. Requisitos Funcionais

* Pesquisa por texto

Quando o utilizador pretender realizar a sua pesquisa textualmente, ser-lhe-á apresentada uma barra branca na janela da aplicação. Aqui o cliente deverá introduzir os parâmetros da sua procura. Dependendo do tipo de cliente, a introdução é feita de maneira diferente. Se a pesquisa for feita num dispositivo *desktop*, o teclado está naturalmente presente na máquina. Por outro lado, caso seja feita num dispositivo *mobile*, será utilizado o teclado que se tem por defeito nesses sistemas. Introduzidos os parâmetros, o cliente deve selecionar o botão apropriado para que a pesquisa seja iniciada.

* Pesquisa por voz

Caso o utilizador não queira ou não consiga fazer a pesquisa por texto, será dada a possibilidade de indicar o que pretende através da sua voz. Para dar início à pesquisa, deverá premir o botão apropriado. Ao selecionar este botão, a aplicação entra em modo de captura de voz, permanecendo neste estado por um período de tempo curto. Aqui, tudo que for dito ao dispositivo será convertido para texto que por sua vez é utilizado para efetuar a pesquisa. A captura de voz termina automaticamente após o utilizador não indicar nada por alguns instantes de tempo.

1. **Apresentação de resultados**
   1. Requisitos funcionais

* Apresentação dos resultados

Efetuada uma pesquisa, é apresentado ao utilizador uma lista de restaurantes que satisfazem os parâmetros introduzidos. A ordem dos resultados é influenciada pelos seguintes critérios: avaliação do estabelecimento, número de comentários feitos e distância ao local de onde foi efetuada a pesquisa.

Destes fatores, distinguem-se os que exercem uma influência positiva na ordem de apresentação: avaliação e número de comentários. Entre estes, a avaliação do estabelecimento é a que tem mais influência. Quanto maior for o número de comentários, pressupõe-se que a avaliação do estabelecimento é mais fidedigna. Assume-se que entre um restaurante com uma avaliação de 4/5 e 100 comentários e um restaurante com a mesma avaliação, mas apenas com 10 comentários, a avaliação do primeiro será a mais fiável. Por esta razão conjuga-se a avaliação com o número de comentários.

Por outro lado, quanto maior for a distância ao local pior será a sua posição na lista de resultados. Assim, evita-se que os restaurantes mais longínquos ocupem os primeiros lugares da lista.

* Histórico de pesquisa

Sempre que um utilizador selecionar a aba histórico, ser-lhe-á apresentada uma lista de pesquisas ordenadas da mais recente para a mais antiga. Ao selecionar uma pesquisa já feita, o cliente é encaminhado para o respetivo estabelecimento. Devido ao número elevado de procuras que poderão ocorrer, o utilizador terá de fazer *scroll down* de forma a visualizar o histórico mais antigo. Do histórico podem ser eliminados estabelecimentos que foram visitados pelo utilizador.

* Sugestões de restaurantes a visitar

É dado ao utilizador a hipótese de ser a aplicação a sugerir-lhe o próximo estabelecimento que deverá visitar. A opção *sugestões* apresenta os cinco melhores estabelecimentos que este ainda não tenha visitado. A ordem de apresentação é por avaliação.

* Filtragem de resultados

Apresentados os resultados da pesquisa, o utilizador tem a possibilidade de refiná-los, selecionando as opções de interesse. Para tal, o cliente tem de selecionar um botão que vai apresentar-lhe um menu para definir as opções do refinamento.

O aperfeiçoamento dos resultados pode ser feito na distância máxima ao local, na avaliação mínima do estabelecimento e/ou se encontra em horário de funcionamento no momento. Introduzidos os campos de personalização da pesquisa, o utilizador seleciona o botão designado para registá-los. Logo de seguida, são apresentados os resultados que cumprem os fatores pretendidos.

* Informações a exibir

A lista dos resultados contém informações mais genéricas para cada um dos restaurantes. Aqui apresenta-se: nome, foto e avaliação. Consideram-se estas as informações mais relevantes para que o utilizador possa formar uma primeira impressão sobre o estabelecimento.

Selecionado o restaurante, acrescentam-se às informações iniciais: descrição do estabelecimento, contactos, *website*, horário de funcionamento, comentários e trajeto ao destino.

Quantos aos comentários, o utilizador poderá navegar sobre a lista apresentada. O local é apresentado numa janela da *Windows Maps*, em que ao seleciona-la é calculado o percurso ao destino.

* 1. Requisitos não funcionais
* Facilidade de utilização

A *User Interface* é intuitiva ao ponto de ser possível a um utilizador, sem experiência na área da informática, conseguir efetuar uma pesquisa, navegar e refinar os resultados, selecionar um estabelecimento e inclusive partilhar numa rede social.

**Business Layer**

1. **Opinião**
   1. Requisitos Funcionais

* Partilha numa rede social

No *Chewie*, selecionado um estabelecimento, o utilizador pode escolher a opção de partilha, sendo-lhe fornecido um conjunto de redes sociais disponíveis. Depois, este é redirecionado para a aplicação respetiva da rede social ou para o *browser.*

* Comentários sobre o estabelecimento visitado

Sempre que um utilizador selecionar um estabelecimento, é-lhe dada a opção de efetuar um comentário sobre o prato gastronómico e/ou estabelecimento. É apresentada uma janela ao utilizador para que possa inserir o seu comentário. Um utilizador apenas pode comentar um estabelecimento uma única vez.

* Avaliação sobre o estabelecimento visitado

Sempre que um utilizador selecionar um estabelecimento, é-lhe dada a opção de avaliar o estabelecimento visitado. A avaliação é um número inteiro compreendido entre 1 e 5, que é guardada diretamente na Base de Dados da aplicação. Uma janela é apresentada para que o utilizador faça a sua avaliação, podendo apenas dar uma única avaliação a um estabelecimento.

1. **Preferências**
   1. Requisitos Funcionais

* Informação para as preferências do utilizador

Um utilizador indica numa pesquisa o prato pretendido e, caso queira, um conjunto de parâmetros adicionais que representam o seu gosto pessoal. Ao indicar que visitou um estabelecimento, é guardada na Base de Dados, para aquele prato, os parâmetros adicionais que inseriu anteriormente. Estes representam as suas preferências, no que diz respeito ao prato pesquisado. Quanto maior o número de vezes que pesquisar o mesmo prato com esses parâmetros, maior será a sua relevância numa pesquisa futura, para aquele prato.

1. **Histórico**
   1. Requisitos Funcionais

* Histórico de Pesquisa

Na aplicação existe uma aba chamada histórico que armazena todas as pesquisas efetuadas. Para cada utilizador são guardados os estabelecimentos que este indica como visitados, como também a data.

1. **Login**
   1. Requisitos Funcionais

* *Login* no sistema

O Cliente pode registar-se na aplicação ou, caso já o tenha feito, efetuar o *login*. Isto serve para que o Cliente possa trocar entre dispositivos (*mobile* ou *desktop*), sem perder as suas preferências de pesquisa ou histórico, como também não permitir que avalie/comente o mesmo estabelecimento mais que uma vez. Para que uma pesquisa possa ser feita, o Cliente tem que estar autenticado.

**Servidor**

1. Pesquisa
   1. Requisitos não funcionais

* Tempo de resposta da aplicação

Pretende-se que o tempo de resposta da aplicação seja o mais curto possível, uma vez que é um fator de grande importância na hora da refeição. Assim, a aplicação deverá apresentar uma lista de resultados num período de tempo inferior a cinco segundos. Desta forma, o tempo de resposta não interfere de maneira negativa na experiência que o utilizador tem com a aplicação.

* Limitação da região

A pesquisa de pratos gastronómicos é limitada à região de Braga.

1. Base de Dados
   1. Requisitos não funcionais

* Memória ocupada pelos dados

A aplicação tem uma arquitetura Cliente-Servidor. O espaço ocupado pelos dados no Cliente é mínimo, isto porque o volume de informação a ser acedido pela aplicação é muito elevado, ou seja, o Servidor é quem guarda e tem acesso direto aos dados. O Cliente terá guardado o seu *username* e a respetiva *password*, de forma a poder ser identificado no momento de *login*.