
APLICAÇÃO WEB “RISTORANTIS”

Universidade de Trás-os-Montes e Alto-Douro

Licenciatura em Engenharia Informática

Laboratório de Aplicações Web e Base de Dados

António Marques

Luís Barbosa

Ano Letivo 2020/2021

Autores

Ana Dias al69691

Diana Alves al68557

Diana Ferreira al68938

Rui Vaz al68565

Resumo

No âmbito da Unidade Curricular de Laboratório de Aplicações Web e Base de Dados foi proposta a criação de um projeto sobre uma aplicação web que permita a gestão dos pratos do dia de variados restaurantes. Para que tal seja atingido, é necessário a implementação de um software adequado, seguindo certos aspetos e normas. Neste caso específico iremos pôr em prática conhecimentos adquiridos em Unidades Curriculares passadas, tais como, Engenharia de Software e Base de Dados, bem como aplicar conhecimentos adquiridos ao longo do semestre de outras Unidades, como Sistemas de Informação e Engenharia Web.

Desse modo, este sistema não só permite que um cliente possa visualizar o seu restaurante de eleição, como também permite a criação de listas de pratos do dia preferidos ou até mesmo de restaurantes, para além de outras características associadas a outros intervenientes da aplicação web. Assim, o sistema criado deverá ser capaz de orientar e instruir um desenvolvedor de maneira a que o fim seja atingido de forma coerente e concisa.

Portanto, nesta primeira fase, fizemos o levantamento de requisitos funcionais e não funcionais, assim como os de casos-de-usos, que posteriormente levou-nos à execução de diagramas de casos-de-uso. Com os casos-de-uso criados e especificados, formamos os diagramas de casos-de-uso. Para além disso, usamos também diagramas de Entidade-Relacionamento para um melhor desenvolvimento do projeto.

Relativamente à segunda etapa do trabalho, o seu principal objetivo foi o mapeamento entre o modelo conceptual de dados e o modelo relacional, com uma implementação do modelo físico do mesmo recorrendo à linguagem SQL. A linguagem SQL é a linguagem de pesquisa declarativa padrão para banco de dados relacional (base de dados relacional).

Assim, através de diagramas adequados, e de outros instrumentos, será possível seguir um conjunto de normas que em simultâneo funcionarão de modo a que um sistema de funcionamento da aplicação web seja possível.

Índice

Introdução.....	4
Enquadramento Teórico	5
Objetivos da Etapa do Trabalho Prático	9
Desenvolvimento	10
Requisitos Funcionais	10
Requisitos Não Funcionais	11
Listagem de Casos-de-Uso	12
Diagramas de Casos-de-Uso	13
Especificação dos Casos-de-Uso	16
Diagrama E-R (Entidade-Relacionamento).....	31
Mapeamento e Normalização do diagrama E-R para o Modelo Relacional.....	32
Mockups de <i>backoffice</i> e <i>frontoffice</i>	40
Bibliografia	52

Introdução

Com o avanço da tecnologia e a implementação e uso da informação, foi possível que as pessoas realizassem a maioria de suas atividades de maneira direta, automatizada, eletrónica e eficientemente. Para acompanhar a necessidade de fornecer aos cidadãos de um país a capacidade de participar e se beneficiar dos serviços *online*, bem como reduzir outros problemas associados, o uso da *Internet* está cada vez mais presente. O uso de uma aplicação web torna o acesso à informação muito mais rápido e prático, uma vez que esta pode apresentar uma seleção variada de detalhes, concentrando-se apenas num só lugar.

A realização da primeira etapa tem como objetivo preparar os alunos de forma a que estes desenvolvam a sua capacidade de levantamento de requisitos, assim como, a sua distinção de requisito funcional e não funcional, uma vez que, esta é uma fase crucial e essencial no planeamento e desenvolvimento de um sistema de software. Para além disso, também tem como alvo o desenvolvimento da capacidade de interpretação de um diagrama Entidade-Relacionamento (E-R), assim como, fazer o mapeamento deste para o modelo relacional, e a sua normalização até à 3ª forma normal, compreendendo assim, o conceito de vários conteúdos. Depois, será implementada a especificação de casos-de-uso, assim como a sua descrição, permitindo dessa forma representar através de diagramas os requisitos.

A realização da segunda etapa tem como objetivo o aprofundamento da aprendizagem dos alunos ao nível da linguagem SQL, bem como a distinção entre o modelo conceptual de dados e o modelo relacional. Para além disso, será especificado as interfaces relativas ao utilizador em relação com o *backoffice*, o *frontoffice*, recorrendo a *mockups*.

Enquadramento Teórico

No âmbito da engenharia, um requisito consiste da definição documentada de uma propriedade ou comportamento que um produto ou serviço particular deve atender. Por outras palavras, um requisito é o termo utilizado para descrever as atividades envolvidas na identificação, documentação e manutenção de um sistema de software, especificando o que deverá ser implementado neste.

Como o sistema final de software é desenvolvido com base num conjunto de requisitos, é crucial que estes estejam bem definidos para atingir sucesso no processo de desenvolvimento do software. Existem dois tipos diferentes de requisitos: requisitos funcionais e não funcionais.

Um **requisito funcional** representa o que o software faz, em termos de tarefas e serviços. Estes podem ser cálculos, detalhes técnicos, manipulação de dados e de processamento e outras funcionalidades específicas que definem o que um sistema, idealmente, será capaz de realizar.

Os **requisitos não-funcionais** são os requisitos relacionados ao uso da aplicação em termos de desempenho, usabilidade, confiabilidade, segurança, disponibilidade, manutenção e tecnologias envolvidas. Estes requisitos dizem respeito a como as funcionalidades serão entregues ao usuário do software

Casos de Uso são uma especificação de um conjunto de ações executadas por um sistema, que contém um resultado observável. Caso de uso é representado por uma elipse, com o nome do caso de uso dentro ou abaixo. Se há limites do sistema no diagrama, o caso de uso deve ficar dentro.

Os **diagramas de casos de uso** descrevem a funcionalidade do sistema sob a perspetiva do utilizador e ilustram os requisitos do sistema, estando concentrados exclusivamente nos processos automatizados. São utilizados para representar os requisitos funcionais, assegurando que estes sejam bem compreensíveis, tendo como objetivo mostrar o que deve ou não fazer um sistema. Um caso de uso representa uma unidade discreta da interação entre um ator (humano, dispositivo ou outro software) e o sistema.

Os atores são sempre externos ao sistema, sendo um ator de um diagrama alguém ou algo que desempenha um papel no sistema. Este pode ter a mesma entidade a desempenhar vários papéis e ter várias entidades a desempenhar o mesmo papel no sistema (apenas um ator). Um ator pode ser de vários tipos:

- Pessoa.

- Sistema.
- Tempo.

Assim, dentro dos diagramas, existem relações entre os atores e ações, como relacionamento de Herança (relacionamento entre atores, utilizado quando queremos representar uma especialização/generalização), relacionamento de Extensão (quando se deseja traçar um relacionamento alternativo), relacionamento de Inclusão (utilizado quando um comportamento se repete em mais de um caso de uso) e relacionamento de Associação (representa a interação entre um ator e um caso de uso por meio de mensagens, sendo representado por uma linha sólida).

Na realização de um diagrama de casos-de-uso também existem regras para a sua construção:

- A possibilidade de criar tantos diagramas de casos-de-uso quanto necessário.
- Não devem ser definidos relacionamentos association entre atores, dado que tal está fora do âmbito da modelação do sistema.
- Não devem ser definidos relacionamentos association entre casos-de-uso, dado que não é a finalidade dos diagramas mostrar qualquer sequência.
- Todos os casos-de-uso devem ter relacionamentos association com atores, com a exceção daqueles que têm relacionamentos com outros casos-de-uso, que poderão ter ou não.

Um dos maiores benefícios dos diagramas de casos-de-uso é o facto de facilitar a comunicação entre os vários intervenientes no projeto.

Os **elementos básicos** de um diagrama E-R são as entidades, os relacionamentos e os atributos.

Uma **entidade** representa um conjunto de objetos (concretos ou abstratos) do mundo real que possuem características comuns. Os objetos em si são as ocorrências.

Um **relacionamento** representa uma associação entre entidades. Os relacionamentos também possuem ocorrências, que são associações entre as ocorrências das entidades. Este contém ainda duas propriedades, que são: a Cardinalidade e o Grau.

Um relacionamento pode possuir uma:

- Cardinalidade máxima (1 ou N): representa a quantidade máxima de ocorrências de uma entidade que pode estar associada a uma mesma ocorrência de outra entidade. Os relacionamentos podem ser do tipo:
 - 1:1 um para um;

- 1:N um para muitos;
- N:N muitos para muitos.
- Cardinalidade mínima (0 ou 1): indica se a participação de todas as ocorrências de uma entidade de um determinado relacionamento é obrigatória (1) ou opcional (0).

Cada relacionamento tem um **grau** que é definido pelo número de entidades ao qual estão associadas, isto é, se existir apenas uma entidade, o grau de relacionamento é nomeado por unário; se existir duas entidades é um relacionamento binário; etc.

Os atributos representam as propriedades elementares de uma entidade ou um relacionamento, podendo as entidades e os relacionamentos estar associados a um ou mais atributos.

Existem vários tipos de atributos:

- Atributos Derivados: São obtidos a partir de outros atributos.
- Atributos Opcionais: São atributos que podem ou não ter valores, dependendo do utilizador.
- Atributos Multivalor: São atributos que podem ter mais que um valor, por exemplo, números de telemóvel.
- Atributos Multivalor Opcionais.
- Atributos Compostos: São atributos que representam um conjunto de atributos, ou seja, um atributo que é dividido num conjunto de subatributos.

O **Modelo Relacional** foi originado de um desenvolvimento teórico, tendo a sua base num ramo da matemática que é, simultaneamente, simples e poderosa – a teoria de conjuntos.

Dentro do Modelo Relacional existe o conceito de Tabela (ou Relação), que é uma estrutura bidimensional com um determinado esquema e zero ou mais instâncias. Este esquema é constituído por um ou mais atributos (também nomeados por campos ou colunas). Cada atributo terá associado um tipo de dados de acordo com a informação que irá armazenar. A cada instância do esquema de uma relação designa-se por *tuplo* (ou linha).

No Modelo Relacional há vários tipos de chaves:

- Superchave: associação de um ou mais atributos que, em conjunto, identificam univocamente cada um dos tuplos;

- **Chave Candidata**: subconjunto dos atributos de um superchave que, sendo ainda superchave, não pode ser reduzido sem perder essa qualidade;
- **Chave Primária**: chave selecionada entre várias chaves candidatas para, efetivamente, identificar cada tuplo.
- **Chave Estrangeira**: Atributo ou conjunto de atributos de uma relação, que é chave primária noutra relação.

A existência de chave estrangeira numa relação prende-se com a necessidade de manter a interligação entre essa relação e a relação onde esse conjunto de atributos é chave primária.

A **Normalização** é um processo sistemático, definido por um conjunto de regras bem definidas, que visa a eliminar fontes de redundância nos dados. Este processo ocorre através de um conjunto de fases que conduzem a base de dados a estados onde a redundância se torna cada vez menor. A cada um destes estados dá-se o nome de forma normal (FN).

Uma relação diz-se que está na **Primeira Forma Normal (1FN)** quando não contém atributos multivalor e não contém grupos repetitivos;

Uma relação diz-se que está na **Segunda Forma Normal (2FN)** quando está na 1FN e todos os atributos não chave dependem funcionalmente da totalidade da chave.

Uma relação diz-se que está na **Terceira Forma Normal (3FN)** quando está na 2FN e todos os atributos não chave não dependem funcionalmente uns dos outros.

A **Linguagem SQL** é uma linguagem normalizada pelo ANSI. A maior parte dos SGBD relacionais usam a linguagem SQL, mas possuem as suas próprias extensões à SQL. No entanto, tipicamente a linguagem pode ser migrada de plataforma para plataforma sem mudanças estruturais principais.

Objetivos da Etapa do Trabalho Prático

Nesta primeira etapa do Trabalho Prático temos como objetivo o levantamento de requisitos, quer funcionais, quer não funcionais, bem como a representação através de especificação de casos-de-uso, usando diagramas. Assim, mais tarde será possível o desenvolvimento de um protótipo funcional da aplicação web “Ristorantis”, cumprindo todos as funções requeridas no protocolo.

Os objetivos da primeira fase são:

- Relatório detalhado da execução do sistema;
- Análise dos requisitos de dados do sistema;
- Especificação do modelo conceptual de dados do sistema, recorrendo a diagramas de Entidade-Relacionamento (E-R);
- Análise dos requisitos funcionais do sistema;
- Especificação do modelo funcional, recorrendo a diagramas de Casos-de-Uso.

Na segunda etapa do Trabalho Prático temos como objetivo a realização do mapeamento entre o modelo de dados conceptual e o modelo relacional, que a posteriori será implementado recorrendo à linguagem SQL.

Os objetivos da segunda fase são:

- Relatório detalhado da execução do trabalho;
- Mapeamento entre o modelo conceptual de dados e o modelo relacional;
- Implementação do modelo físico da base de dados recorrendo à linguagem SQL;
- Especificação das interfaces com o utilizador para o *backoffice* e o *frontoffice* recorrendo a *mockups*.

Desenvolvimento

Requisitos Funcionais

- RF1. A aplicação deverá mostrar todos os restaurantes registados.
- RF2. A aplicação deverá permitir a consulta da informação dos restaurantes.
- RF3. A aplicação deverá permitir a consulta do prato do dia dos restaurantes.
- RF4. A aplicação deverá permitir fazer o login.
- RF5. A aplicação deverá permitir fazer a autenticação.
- RF6. A aplicação deverá permitir fazer o logout.
- RF7. A aplicação deverá permitir fazer o registo de utilizadores.
- RF8. A aplicação deverá permitir fazer o registo de restaurantes.
- RF9. A aplicação deverá permitir a pesquisa de pratos do dia e sua informação.
- RF10. A aplicação deverá permitir as funções atualizar e consultar a lista de pratos preferidos.
- RF11. A aplicação deverá permitir as funções atualizar e consultar a lista de restaurantes preferidos.
- RF12. A aplicação deverá permitir o registo do prato do dia.
- RF13. A aplicação deverá manter um registo de pratos do dia.
- RF14. A aplicação deverá permitir a reutilização de dados do histórico de pratos.
- RF15. A aplicação deverá permitir a criação de outros administradores.
- RF16. A aplicação deverá permitir a aceitação dos pedidos de registo dos restaurantes.
- RF17. A aplicação deverá permitir a gestão de utilizadores.
- RF18. A aplicação deverá permitir efetuar a confirmação de registo por email.
- RF19. A aplicação deverá identificar o administrador que confirmou o registo de restaurantes.
- RF20. A aplicação deverá permitir efetuar login após a confirmação de email.
- RF21. A aplicação deverá permitir a confirmação do registo dos restaurantes.
- RF22. A aplicação deverá permitir a introdução de informação após a validação de registo dos restaurantes.
- RF23. A aplicação deverá permitir bloquear um utilizador.
- RF24. A aplicação deverá permitir desbloquear um utilizador.
- RF25. A aplicação deverá permitir o registo do motivo do bloqueio de acesso.
- RF26. A aplicação deverá notificar o cliente caso um restaurante preferido insira um prato do dia.
- RF27. A aplicação deverá notificar o cliente caso um prato preferido seja inserido como prato do dia.
- RF28. A aplicação deverá impedir a criação de uma conta caso o email inserido já exista na base de dados.

Requisitos Não Funcionais

- RNF1. A aplicação deverá permitir a pesquisa de pratos do dia por palavra chave e/ou por tipo.
- RNF2. O sistema deverá apresentar um menu com todas as funcionalidades.
- RNF3. O sistema deverá permitir aceder às várias funcionalidades existentes.
- RNF4. A aplicação deverá limitar o número de palavras do motivo do bloqueio.
- RNF5. A aplicação deve de permitir a utilizadores com login efetuado verificar as suas notificações.
- RNF6. A aplicação deve atualizar as páginas de pesquisa quando filtros forem alterados.
- RNF7. A aplicação deverá notificar o utilizador caso as credenciais estejam erradas.
- RNF8. O login pode ser efetuado através do username ou do email do utilizador.
- RNF9. No momento do registo a password deve conter no mínimo duas letras maiúsculas seguido de dois números e pelo menos um carácter especial.
- RNF10. As datas em todo o site terão o formato DD/MM/AAAA.

Listagem de Casos-de-Uso (por atores)

Visitante

- Login;
- Consultar Informação de Restaurantes;
- Consultar Pratos do Dia;
- Criar Conta.

Utilizador

- Logout;
- <visitante>.

Cliente

- Pesquisar Pratos do Dia;
- Consultar Lista de Favoritos de Pratos do Dia;
- Atualizar Lista de Favoritos de Pratos do Dia;
- Consultar Lista de Favoritos de Restaurantes;
- Atualizar Lista de Favoritos de Restaurantes;
- <utilizador>.

Restaurante

- Registrar Pratos do Dia;
- <utilizador>.

Administrador

- Criar Administrador;
- Aceitar pedido de Registo de Restaurante;
- Bloquear Utilizadores;
- Desbloquear Utilizadores;
- <utilizador>.

Diagramas de Casos-de-Uso

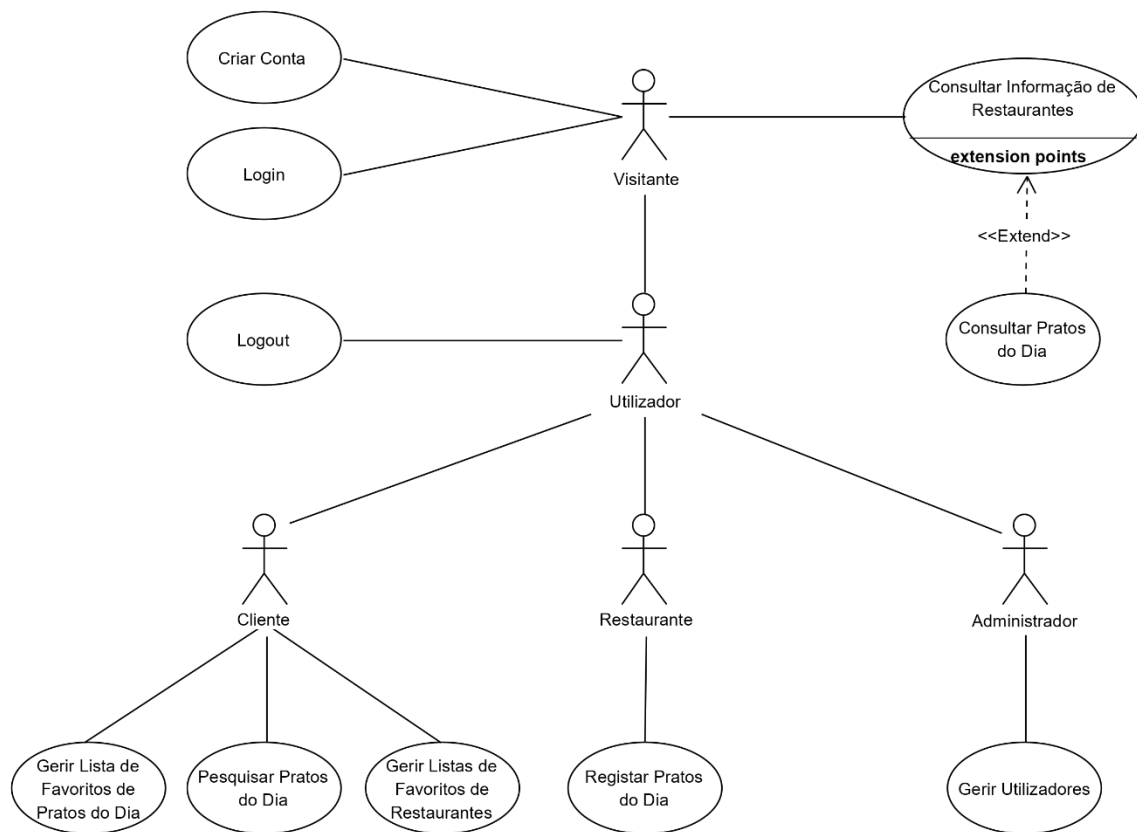


Figura 1: Diagrama de Atores e Casos-de-Uso

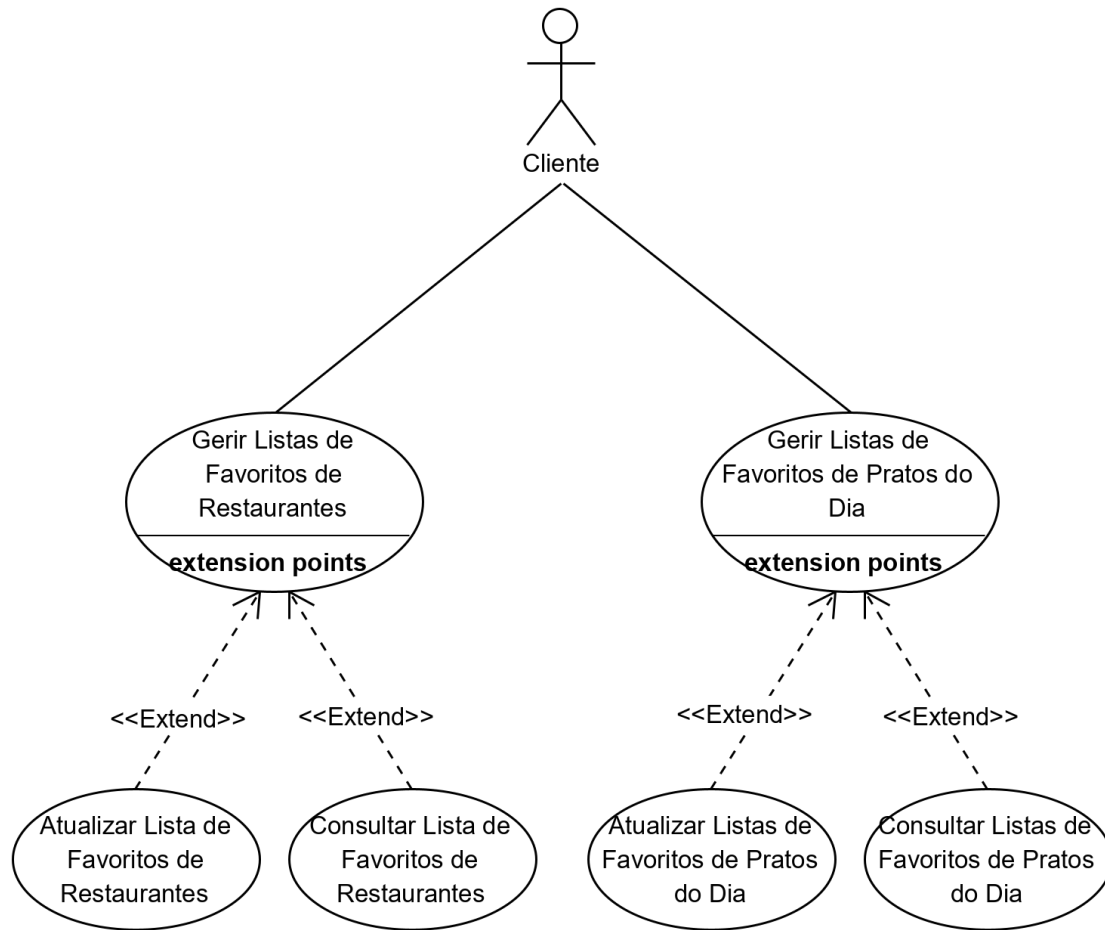


Figura 2: Diagrama de casos-de-uso para o ator Cliente

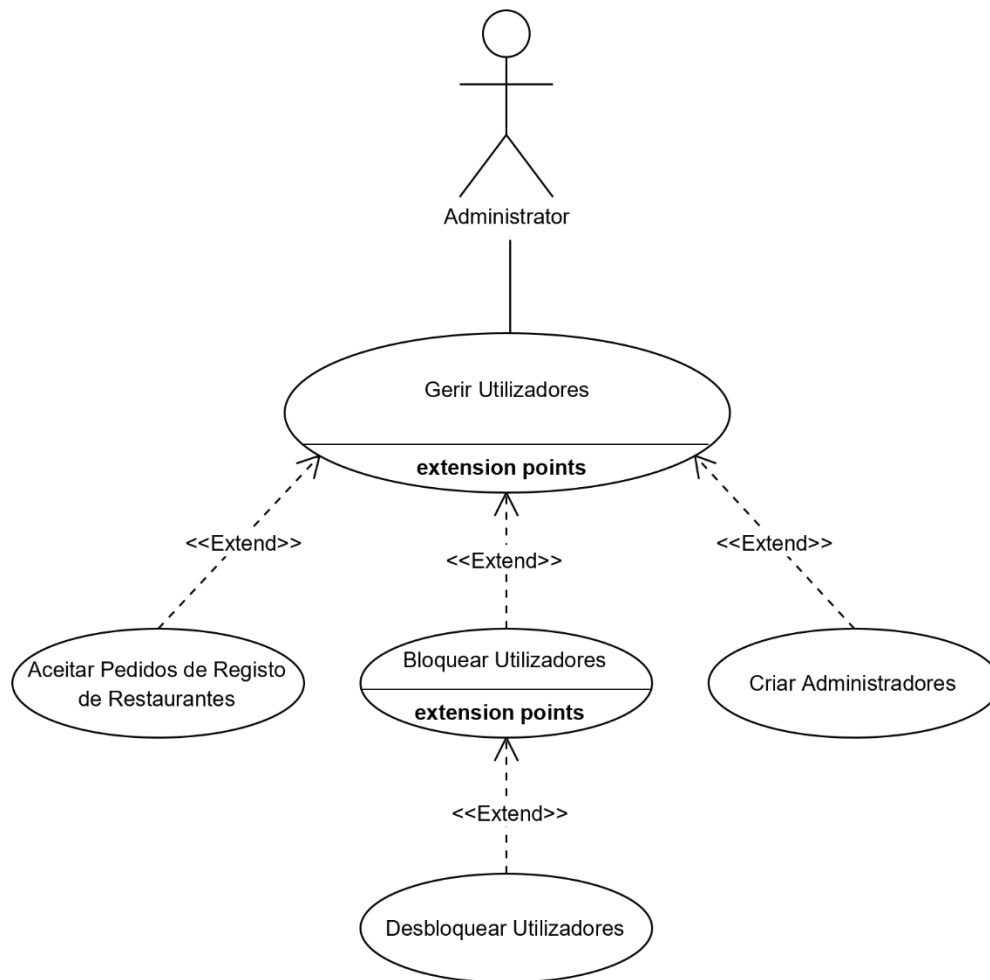


Figura 3: Diagrama de casos-de-uso para o ator Administrador.

Especificação dos Casos-de-Uso

Nome		Consultar Informação de Restaurantes
Tipo		Caso de Uso
ID		RF2
Ator		Visitante
Descrição		Permite consultar a informação relativa a cada restaurante registado no sistema.
C E N Á R I O	Pré-Condições	<ul style="list-style-type: none"> • Não tem.
	Fluxo Principal	<ol style="list-style-type: none"> 1. É apresentada uma lista com os restaurantes registados no sistema; 2. O visitante deverá selecionar um restaurante para aceder às informações do restaurante.
	Fluxo Alternativo	<ul style="list-style-type: none"> • Não tem.
	Pós-Condições	<ul style="list-style-type: none"> • Informação do Restaurante consultada com sucesso.
Notas		Não tem.

Nome		Consultar Pratos do Dia
Tipo		Caso de Uso
ID		RF3
Ator		Visitante
Descrição		Permite consultar o Prato do Dia bem como os Restaurantes em que este se encontra.
C E N Á R I O	Pré-Condições	<ul style="list-style-type: none"> • Não tem.
	Fluxo Principal	<ol style="list-style-type: none"> 1. É apresentada uma lista com os restaurantes registados no sistema; 2. O visitante deverá selecionar um restaurante para aceder às informações e a todos os pratos do dia relativos a esse restaurante.
	Fluxo Alternativo	<ul style="list-style-type: none"> • Não tem.
	Pós-Condições	<ul style="list-style-type: none"> • Informação sobre o Prato do Dia consultada com sucesso.
Notas		Não tem.

Nome		Login
Tipo		Caso de Uso
ID		RF4
Ator		Visitante
Descrição		Realizar o login dos utilizadores.
C E N Á R I O	Pré-Condições	<ul style="list-style-type: none"> O utilizador não efetuou o login.
	Fluxo Principal	<ol style="list-style-type: none"> O utilizador insere as suas credenciais (username ou e-mail, password). <ol style="list-style-type: none"> Uma, ou ambas, as credenciais inseridas pelo eleitor estão erradas.
	Fluxo Alternativo	<ol style="list-style-type: none"> Uma, ou ambas, as credenciais inseridas pelo utilizador estão erradas. <ol style="list-style-type: none"> O caso de uso inicia no primeiro passo sempre que o utilizador insira as credenciais erradas; Durante as tentativas de acesso: <ol style="list-style-type: none"> O sistema pede para introduzir novamente as credenciais do utilizador; O sistema informa o utilizador que a sua autenticação não foi realizada com sucesso.
	Pós-Condições	<ul style="list-style-type: none"> O utilizador efetuou o login com sucesso.
Notas		Permite ao utilizador aceder ao menu de funcionalidades.

Nome		Logout
Tipo		Caso de Uso
ID		RF6
Ator		Utilizador
Descrição		Permitir terminar sessão no sistema.
C E N Á R I O	Pré-Condições	<ul style="list-style-type: none"> O utilizador deverá ter efetuado o login.
	Fluxo Principal	1. O utilizador deverá selecionar "Logout".
	Fluxo Alternativo	<ul style="list-style-type: none"> Não tem.
	Pós-Condições	<ul style="list-style-type: none"> Sessão terminada com sucesso.
Notas		Não tem.

Nome		Criar Conta
Tipo		Caso de Uso
ID		RF7 e 8
Ator		Visitante
Descrição		Permite ao visitante fazer o registo no sistema.
C E N Á R I O	Pré-Condições	<ul style="list-style-type: none">• Não tem.
	Fluxo Principal	<ol style="list-style-type: none">1. O visitante deverá escolher o tipo de conta (cliente/restaurante);2. O visitante deverá preencher os campos obrigatórios;3. O visitante deverá submeter o registo.
	Fluxo Alternativo	<ul style="list-style-type: none">• Não tem.
	Pós-Condições	<ul style="list-style-type: none">• O registo só fica concluído após a confirmação da conta.
Notas		<p>Caso a conta seja de um cliente a confirmação é feita pelo próprio via email;</p> <p>Caso seja uma conta de um restaurante, este tem de esperar pela validação de um administrador.</p>

Nome		Pesquisar Pratos do Dia
Tipo		Caso de Uso
ID		RF9
Ator		Cliente
Descrição		Permite pesquisar o Prato do Dia.
C E N Á R I O	Pré-Condições	<ul style="list-style-type: none"> O utilizador deve estar registado, com login efetuado no sistema como “Cliente” e no menu de funcionalidades.
	Fluxo Principal	<ol style="list-style-type: none"> É apresentada uma search box onde o cliente poderá pesquisar o Prato do Dia que entender; É apresentada uma lista com os Pratos do Dia registados no sistema de acordo com a pesquisa efetuada.
	Fluxo Alternativo	<ul style="list-style-type: none"> Não tem.
	Pós-Condições	<ul style="list-style-type: none"> Informação sobre os Pratos do Dia consultada com sucesso.
Notas		Os clientes podem pesquisar pratos do dia por palavra chave e/ou por tipo.

Nome		Consultar Lista de Favoritos de Pratos do Dia
Tipo		Caso de Uso
ID		RF10
Ator		Cliente
Descrição		Permitir consultar informações sobre a Lista de Favoritos de Pratos do Dia.
C E N Á R I O	Pré-Condições	<ul style="list-style-type: none"> O cliente deverá ter o login efetuado.
	Fluxo Principal	<ol style="list-style-type: none"> O utilizador deverá encontrar-se no menu de funcionalidades; O utilizador deverá selecionar "Pratos do Dia"; O utilizador deverá escolher a opção "Favoritos".
	Fluxo Alternativo	<ul style="list-style-type: none"> Não tem.
	Pós-Condições	<ul style="list-style-type: none"> Lista de Favoritos consultada com sucesso.
Notas		Permite ao cliente aceder a outras funcionalidades, tais como: atualizar a lista.

Nome		Atualizar Lista de Favoritos de Pratos do Dia
Tipo		Caso de Uso
ID		RF10
Ator		Cliente
Descrição		Permitir atualizar informações sobre a Lista de Favoritos de Pratos do Dia.
C E N Á R I O	Pré-Condições	<ul style="list-style-type: none"> O cliente deverá ter o login efetuado.
	Fluxo Principal	<ol style="list-style-type: none"> O utilizador deverá encontrar-se no menu de funcionalidades; O utilizador deverá selecionar "Pratos do Dia"; Caso o utilizador pretenda adicionar um prato do dia à lista, este deverá pesquisá-lo e selecioná-lo; Caso o utilizador pretenda retirar um prato do dia da lista: <ol style="list-style-type: none"> Este pode desselecionar um prato do dia diretamente da Lista de Favoritos; Este pode desselecionar um prato do dia após a pesquisa do mesmo.
	Fluxo Alternativo	<ul style="list-style-type: none"> Não tem.
	Pós-Condições	<ul style="list-style-type: none"> Lista atualizada com sucesso.
Notas		Não tem.

Nome		Consultar Lista de Favoritos de Restaurantes
Tipo		Caso de Uso
ID		RF11
Ator		Cliente
Descrição		Permitir consultar informações sobre a Lista de Favoritos de Restaurantes.
C E N Á R I O	Pré-Condições	<ul style="list-style-type: none"> O cliente deverá ter o login efetuado.
	Fluxo Principal	<ol style="list-style-type: none"> O utilizador deverá encontrar-se no menu de funcionalidades; O utilizador deverá selecionar "Restaurantes"; O utilizador deverá escolher a opção "Favoritos".
	Fluxo Alternativo	<ul style="list-style-type: none"> Não tem.
	Pós-Condições	<ul style="list-style-type: none"> Lista de Favoritos consultada com sucesso.
Notas		Permite ao cliente aceder a outras funcionalidades, tais como: atualizar a lista.

Nome		Atualizar Lista de Favoritos de Restaurantes
Tipo		Caso de Uso
ID		RF11
Ator		Cliente
Descrição		Permitir atualizar informações sobre a Lista de Favoritos de Restaurantes.
C E N Á R I O	Pré-Condições	<ul style="list-style-type: none"> O cliente deverá ter o login efetuado.
	Fluxo Principal	<ol style="list-style-type: none"> O utilizador deverá encontrar-se no menu de funcionalidades; O utilizador deverá selecionar "Restaurantes"; Caso o utilizador pretenda adicionar um restaurante à lista, este deverá pesquisá-lo e selecioná-lo; Caso o utilizador pretenda retirar um restaurante da lista: <ol style="list-style-type: none"> Este pode desselecionar um restaurante diretamente da Lista de Favoritos; Este pode desselecionar um restaurante após a pesquisa do mesmo.
	Fluxo Alternativo	<ul style="list-style-type: none"> Não tem.
	Pós-Condições	<ul style="list-style-type: none"> Lista atualizada com sucesso.
Notas		Não tem.

Nome		Registar Pratos do Dia
Tipo		Caso de Uso
ID		RF12
Ator		Restaurante
Descrição		Permite ao restaurante registar um Prato do Dia.
C E N Á R I O	Pré-Condições	<ul style="list-style-type: none"> O utilizador deve estar registado, com login efetuado no sistema como “Restaurante” e no menu de funcionalidades.
	Fluxo Principal	<ol style="list-style-type: none"> O restaurante deverá encontrar-se no menu de funcionalidades; O restaurante deverá selecionar “Registar Prato do Dia”; O restaurante deverá preencher todos os campos obrigatórios de registo; O restaurante deverá submeter o registo do Prato do Dia.
	Fluxo Alternativo	<ul style="list-style-type: none"> Não tem.
	Pós-Condições	<ul style="list-style-type: none"> Prato do Dia registado com sucesso.
Notas		Não tem.

Nome		Criar Administrador
Tipo		Caso de Uso
ID		RF15
Ator		Administrador
Descrição		Permite a um administrador criar outro administrador.
C E N Á R I O	Pré-Condições	<ul style="list-style-type: none"> O utilizador deve estar registado, com login efetuado no sistema como “Administrador” e no menu de funcionalidades.
	Fluxo Principal	<ol style="list-style-type: none"> O Administrador escolhe a opção “Utilizadores” no menu individual do próprio; O sistema mostra a opção de "Criar Administrador"; O Administrador tem de preencher todos os campos obrigatórios; O Administrador tem de efetuar a submissão.
	Fluxo Alternativo	<ul style="list-style-type: none"> Não tem.
	Pós-Condições	<ul style="list-style-type: none"> Um novo Administrador é criado com sucesso.
Notas		Não tem.

Nome		Aceitar pedido de Registo de Restaurante
Tipo		Caso de Uso
ID		RF16
Ator		Administrador
Descrição		Permite ao administrador aceitar ou rejeitar o pedido de registo de um restaurante.
C E N Á R I O	Pré-Condições	<ul style="list-style-type: none"> O utilizador deve estar registado, com login efetuado no sistema como “Administrador” e no menu de funcionalidades.
	Fluxo Principal	<ol style="list-style-type: none"> O Administrador escolhe a opção “Utilizadores” no menu individual do próprio; O sistema mostra a opção de "Aceitar Pedidos de Registo de Restaurantes"; O sistema apresenta uma lista de pedidos pendentes; O Administrador seleciona um pedido e analisa-o; O Administrador toma uma decisão entre aceitar ou rejeitar o pedido; O Administrador deve submeter a decisão.
	Fluxo Alternativo	<ul style="list-style-type: none"> Não tem.
	Pós-Condições	<ul style="list-style-type: none"> O pedido é aceite ou rejeitado com sucesso.
Notas		<p>Ao aceitar o pedido deverá ficar registado qual o administrador que o aceitou;</p> <p>Só após aceitação do pedido é que o restaurante pode introduzir a informação.</p>

Nome		Bloquear Utilizadores
Tipo		Caso de Uso
ID		RF23
Ator		Administrador
Descrição		Permite ao administrador bloquear um utilizador.
C E N Á R I O	Pré-Condições	<ul style="list-style-type: none"> O utilizador deve estar registado, com login efetuado no sistema como “Administrador” e no menu de funcionalidades.
	Fluxo Principal	<ol style="list-style-type: none"> O Administrador escolhe a opção “Utilizadores” no menu individual do próprio; O sistema mostra um motor de pesquisa de utilizadores; O Administrador escolhe o utilizador, através do ID, username ou email, que pretende bloquear; O Administrador tem de indicar o motivo do bloqueio; O Administrador seleciona a opção “Bloquear Utilizador”.
	Fluxo Alternativo	<ul style="list-style-type: none"> Não tem.
	Pós-Condições	<ul style="list-style-type: none"> O Utilizador é bloqueado com sucesso.
Notas		Quando o Utilizador tenta efetuar login, este é informado que não o pode fazer devido a estar bloqueado, sendo apresentado o motivo.

Nome		Desbloquear Utilizadores
Tipo		Caso de Uso
ID		RF24
Ator		Administrador
Descrição		Permite ao administrador desbloquear um utilizador.
C E N Á R I O	Pré-Condições	<ul style="list-style-type: none"> O utilizador deve estar registado, com login efetuado no sistema como “Administrador” e no menu de funcionalidades.
	Fluxo Principal	<ol style="list-style-type: none"> O Administrador escolhe a opção “Utilizadores” no menu individual do próprio; O Administrador escolhe a opção “Utilizadores Bloqueados” no menu individual; O sistema mostra um motor de pesquisa de utilizadores; O Administrador escolhe o utilizador que pretende desbloquear, na lista de utilizadores bloqueados; O Administrador seleciona a opção “Desbloquear Utilizador”.
	Fluxo Alternativo	<ul style="list-style-type: none"> Não tem.
	Pós-Condições	<ul style="list-style-type: none"> O Utilizador é desbloqueado com sucesso.
Notas		Quando o Utilizador tenta efetuar login, este é informado que já se encontra desbloqueado.

Diagrama E-R (Entidade-Relacionamento)

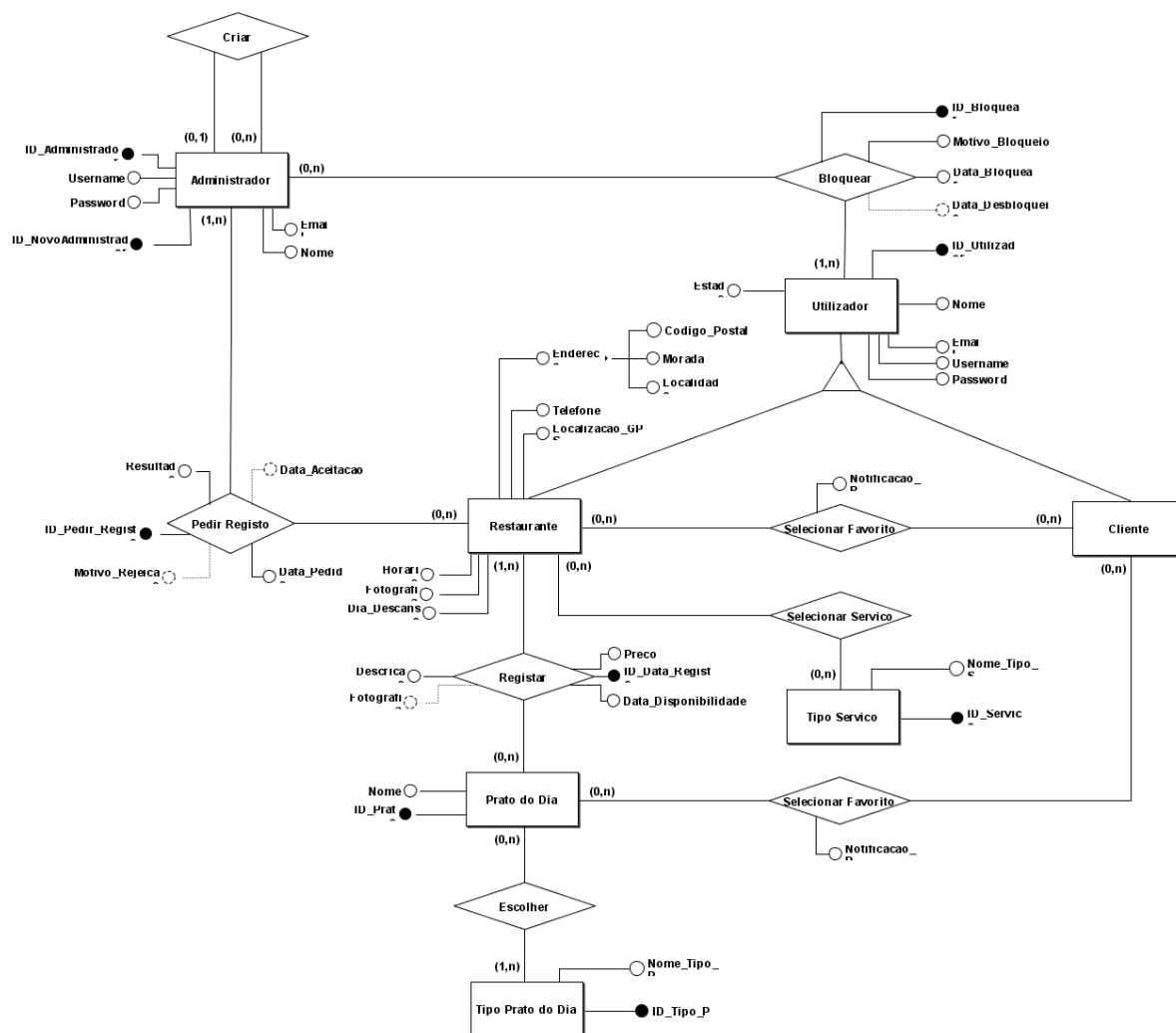


Figura 4: Diagrama E-R

Mapeamento e Normalização do diagrama E-R para o Modelo Relacional

Entidades

Administrador

Administrador (ID_Administrador, Username, Password, Email, Nome)

Utilizador

Utilizador (ID, Nome, Email, Username, Password, Estado)

Cliente

Cliente (ID_Cliente)

ID_Cliente referencia Utilizador

Restaurante

Restaurante (ID_Restaurante, Telefone, Localizacao_GPS,
Endereco_Codigo_Postal, Endereco_Morada, Endereco_Localidade, Horario,
Fotografia, Dia_Descanso)

ID_Restaurante referencia Utilizador

Prato do Dia

PratoDoDia (ID_Prato, Nome)

Tipo de Serviço

Tipo_Servico (ID_Servico, Nome_Tipo_S)

Tipo PratoDoDia

Tipo_PratoDoDia(ID_Tipo, Nome_tipo_P)

Relacionamentos

Selecionar Serviço

Selecionar_Servico (~~ID_Servico~~, ~~ID_Restaurante~~)

ID_Servico referencia Tipo_Servico

ID_Restaurante referencia Restaurante

Registrar

Registrar (ID_Data_Registo, Data_Disponibilidade, Fotografia, Descricao, Preco,
~~ID_Restaurante~~, ~~ID_PratoDoDia~~)

ID_Restaurante referencia Restaurante

ID_PratoDoDia referencia PratoDoDia

Pedir Registo

Pedir_Registo (Data_Pedido, Data_Aceitacao, Resultado, Motivo_Rejeicao,
~~ID_Restaurante~~, ~~ID_Administrador~~)

ID_Restaurante referencia Restaurante

ID_Administrador referencia Administrador

Selecionar Restaurantes Favoritos

Selecionar_R_Favoritos (Notificacao_R, ~~ID_Cliente~~, ~~ID_Restaurante~~)

ID_Cliente referencia Cliente

ID_Restaurante referencia Restaurante

Selecionar Pratos Favoritos

Selecionar_P_Favoritos (Notificacao_P, ~~ID_Cliente~~, ~~ID_Restaurante~~)

ID_Cliente referencia Cliente

ID_Restaurante referencia Restaurante

Bloquear

Bloquear (Data_Bloquear, Motivo_Bloqueio, Data_Desbloqueio,
~~ID_Administrador, ID~~)

ID_Adminitrador referencia Administrador

ID referencia Utilizador

Criar

Criar(~~ID_Administrador~~, ~~ID_NovoAdministrador~~)

ID_Administrador referencia Administrador

ID_NovoAdministrador referencia Administrador

Escolher

Escolher(~~ID_Prato~~, ~~ID_Tipo~~)

ID_Prato referencia PratoDoDia

ID_Tipo referencia Tipo _PratoDoDia

Implementação do modelo físico da base de dados recorrendo à linguagem SQL

```
USE MASTER
GO
```

```
CREATE DATABASE RISTORANTIS
GO
```

```
USE RISTORANTIS
GO
```

```
---ENTIDADES---
```

```
CREATE TABLE Administrador(
    ID_Administrador    INTEGER NOT NULL,
    Username             NVARCHAR(10) NOT NULL,
    Password             NVARCHAR(10) NOT NULL,
    Email                NVARCHAR NOT NULL,
    Nome                 NVARCHAR(150) NOT NULL,

    PRIMARY KEY (ID_Administrador)
)
```

```
CREATE TABLE Utilizador(
    ID_Utilizador       INTEGER NOT NULL,
    Nome                NVARCHAR(150) NOT NULL,
    Email               NVARCHAR NOT NULL,
    Username             NVARCHAR(10) NOT NULL,
    Password             NVARCHAR(10) NOT NULL,
    Estado              NVARCHAR NOT NULL,

    PRIMARY KEY (ID_Utilizador)
)
```

```
CREATE TABLE Cliente(
    ID_Cliente    INTEGER NOT NULL,

    PRIMARY KEY(ID_Cliente),
    FOREIGN KEY(ID_Cliente) REFERENCES
Utilizador(ID_Utilizador)
)
```

```
CREATE TABLE Restaurante(
    ID_Restaurante          INTEGER NOT NULL,
    Telefone                INTEGER NOT NULL,
    Localizacao_GPS         NVARCHAR NOT NULL,
    Endereco_Codigo_Postal NVARCHAR(8) NOT NULL,
    Endereco_Morada         NVARCHAR(50) NOT NULL,
    Endereco_Localidade    NVARCHAR(50) NOT NULL,
    Horario                 NVARCHAR NOT NULL,
    Fotografia              NVARCHAR NOT NULL,
    Dia_Descanso            NVARCHAR(50) NOT NULL,

    CHECK(Telefone LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),
    CHECK(Endereco_Codigo_Postal LIKE '[0-9][0-9][0-9][0-9]-[0-9][0-9][0-9]'),

    PRIMARY KEY(ID_Restaurante),
    FOREIGN KEY (ID_Restaurante) REFERENCES
Utilizador(ID_Utilizador)
)
```

```
CREATE TABLE PratoDoDia(
    ID_Prato          INTEGER NOT NULL,
    Nome              NVARCHAR (50) NOT NULL,

    PRIMARY KEY (ID_Prato)
)
```

```
CREATE TABLE Tipo_Servico(
    ID_Servico          INTEGER NOT NULL,
    Nome_Tipo_S         NVARCHAR(50) NOT NULL,

    PRIMARY KEY(ID_Servico)
)
```

```
CREATE TABLE Tipo_PratoDoDia(
    ID_Tipo_P          INTEGER NOT NULL,
    Nome_tipo_P        NVARCHAR(50) NOT NULL

    PRIMARY KEY (ID_Tipo_P)
)
```

---RELACIONAMENTOS---

```
CREATE TABLE Selecionar_Servico(
    ID_Servico          INTEGER NOT NULL,
    ID_Restaurante      INTEGER NOT NULL,
```

```

        FOREIGN KEY(ID_Servico) REFERENCES
Tipo_Servico(ID_Servico),
        FOREIGN KEY(ID_Restaurante) REFERENCES
Restaurante(ID_Restaurante)
)

CREATE TABLE Registrar(
    ID_Data_Registo          INTEGER NOT NULL,
    Data_Disponibilidade     DATE NOT NULL,
    Fotografia                NVARCHAR,    --opcional--
    Descricao                NVARCHAR(50) NOT NULL,
    Preco                    MONEY NOT NULL,
    ID_Restaurante           INTEGER NOT NULL,
    ID_PratoDoDia            INTEGER NOT NULL,

    PRIMARY KEY (ID_Data_Registo),
    FOREIGN KEY (ID_Restaurante) REFERENCES
Restaurante(ID_Restaurante),
    FOREIGN KEY (ID_PratoDoDia) REFERENCES
PratoDoDia(ID_Prato),
)

CREATE TABLE Pedir_Registo (
    ID_Pedir_Registo        INTEGER NOT NULL,
    Data_Pedido             DATE NOT NULL,
    Data_Aceitacao          DATE,    --opcional--
    Resultado               NVARCHAR(10) NOT NULL,
    Motivo_Rejeicao          NVARCHAR(100), --opcial--
    ID_Restaurante          INTEGER NOT NULL,
    ID_Administrador        INTEGER NOT NULL,

    PRIMARY KEY (ID_Pedir_Registo, ID_Restaurante,
ID_Administrador),
    FOREIGN KEY (ID_Restaurante) REFERENCES
Restaurante(ID_Restaurante),
    FOREIGN KEY (ID_Administrador) REFERENCES
Administrador(ID_Administrador),
)

CREATE TABLE Selecionar_R_Favoritos(
    ID_Cliente              INTEGER NOT NULL,
    ID_Restaurante          INTEGER NOT NULL,
    Notificacao_R           BIT NOT NULL,

    PRIMARY KEY(ID_Cliente, ID_Restaurante),
    FOREIGN KEY(ID_Cliente) REFERENCES Cliente(ID_Cliente),
    FOREIGN KEY (ID_Restaurante) REFERENCES
Restaurante(ID_Restaurante)
)

```

```
CREATE TABLE Selecionar_P_Favoritos(
    ID_Cliente          INTEGER NOT NULL,
    ID_Restaurante      INTEGER NOT NULL,
    Notificacao_P       BIT NOT NULL,

    PRIMARY KEY (ID_Cliente, ID_Restaurante),
    FOREIGN KEY(ID_Cliente) REFERENCES Cliente(ID_Cliente),
    FOREIGN KEY (ID_Restaurante) REFERENCES
Restaurante(ID_Restaurante)
)

CREATE TABLE Bloquear(
    ID_Bloquear         INTEGER NOT NULL,
    Data_Bloquear       DATE NOT NULL,
    Motivo_Bloqueio     NVARCHAR (100) NOT NULL,
    Data_Desbloqueio    DATE, --opcional--
    ID_Administrador    INTEGER NOT NULL,
    ID_Utilizador        INTEGER NOT NULL,

    PRIMARY KEY(ID_Bloquear, ID_Administrador, ID_Utilizador),
    FOREIGN KEY(ID_Administrador) REFERENCES
Administrador(ID_Administrador),
    FOREIGN KEY(ID_Utilizador) REFERENCES
Utilizador(ID_Utilizador)
)

CREATE TABLE Criar(
    ID_Administrador    INTEGER NOT NULL,
    ID_NovoAdministrador INTEGER NOT NULL,

    FOREIGN KEY(ID_Administrador) REFERENCES
Administrador(ID_Administrador),
    FOREIGN KEY(ID_NovoAdministrador) REFERENCES
Administrador(ID_Administrador)
)

create table Escolher(
    ID_Prato            INTEGER NOT NULL,
    ID_Tipo_P           INTEGER NOT NULL,

    FOREIGN KEY (ID_Prato) REFERENCES PratoDoDia(ID_Prato),
    FOREIGN KEY (ID_Tipo_P) REFERENCES
Tipo_PratoDoDia(ID_Tipo_P)
)
```

[illegible]

Mockups de backoffice e frontoffice

Ristorantis

[←](#)
[→](#)
[↺](#)

Ristorantis
[username](#) | [Terminar Sessão](#)

Alterar Dados Restaurante


* Telefone:

* Horário:

* Morada:

Dia Descanço:

* Código Postal:

* Fotografia: 

* Localidade:

* Localização GPS:

[Restaurantes](#)
[Registar](#)
[Sobre](#)
info@ristorantis.com

☐ **Ristorantis**

Ristorantis

[←](#)
[→](#)
[↺](#)

Ristorantis
[username](#) | [Terminar Sessão](#)

Alterar Dados da Conta

* Nome:

* Email:

* Nome de utilizador:

* Palavra-passe antiga:

[Restaurantes](#)
[Registar](#)
[Sobre](#)
info@ristorantis.com

☐ **Ristorantis**

Ristorantis

[←](#)
[→](#)
[↻](#)

Ristorantis

[username](#) | [Terminar Sessão](#)

Alterar Prato

Nome:

Tipo:

* Preço:

* Descrição:

* Dia Disponibilidade:

NOVEMBER 2020						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5
6	7	8	9	10	11	12

Fotografia:

Restaurantes
Registrar
Sobre

☐ Ristorantis
info@ristorantis.com

Ristorantis

[←](#)
[→](#)
[↻](#)

Ristorantis

[Entrar](#) | [Registrar](#)

Nome do Restaurante

Prato do Dia - 01/01/2020

[Voltar ao Início](#)

Restaurantes
Registrar
Sobre

☐ Ristorantis
info@ristorantis.com


Ristorantis

[←](#)
[→](#)
[↺](#)

☰

Ristorantis username | Terminar Sessão

Nome do Restaurante



☒ Aprovar Pedido
☒ Rejeitar Pedido
 Motivo Rejeição:

Voltar

Confirmar

Restaurantes Registrar Sobre

☐ Ristorantis

info@ristorantis.com

Ristorantis

[←](#)
[→](#)
[↺](#)

☰

Ristorantis username | Terminar Sessão

Detalhes Utilizador

Nome:

Email:

Nome de utilizador:

Motivo Bloqueio:

Desbloquear

Bloquear

Cancelar

Tornar Admin

Restaurantes Registrar Sobre

☐ Ristorantis

info@ristorantis.com

Ristorantis

[←](#)
[→](#)
[↻](#)

Ristorantis

username | [Terminar Sessão](#)

Alterar Preferência de Notificações

☒ Receber notificação por email sobre pratos do dia favoritos
☒ Receber notificação por email sobre restaurantes favoritos

[Restaurantes](#)
[Registrar](#)
[Sobre](#)
info@ristorantis.com

☐ Ristorantis

Ristorantis

[←](#)
[→](#)
[↻](#)

Ristorantis

username | [Terminar Sessão](#)

Lista de Utilizadores

Username	Nome	Tipo	Estado
username1	Nome	Cliente	Ativo
username2	Nome	Cliente	Ativo

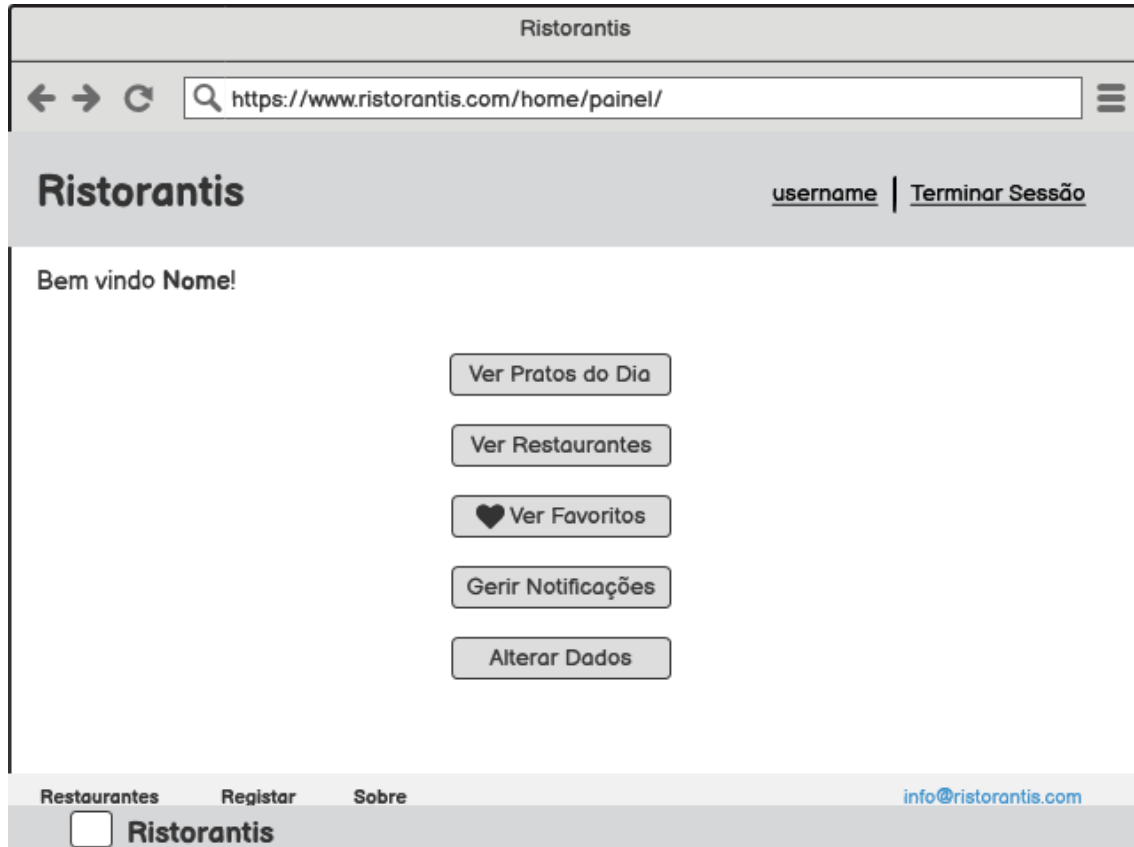
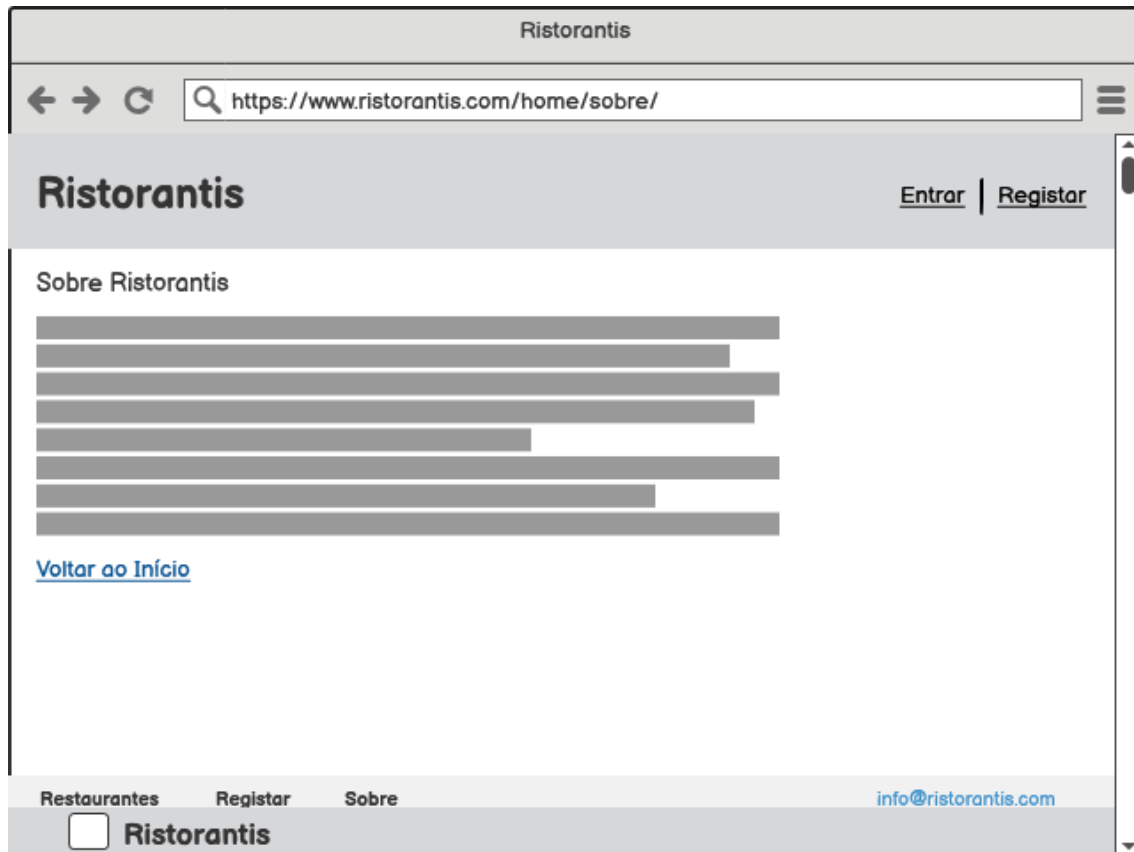
Ordenar por ▼

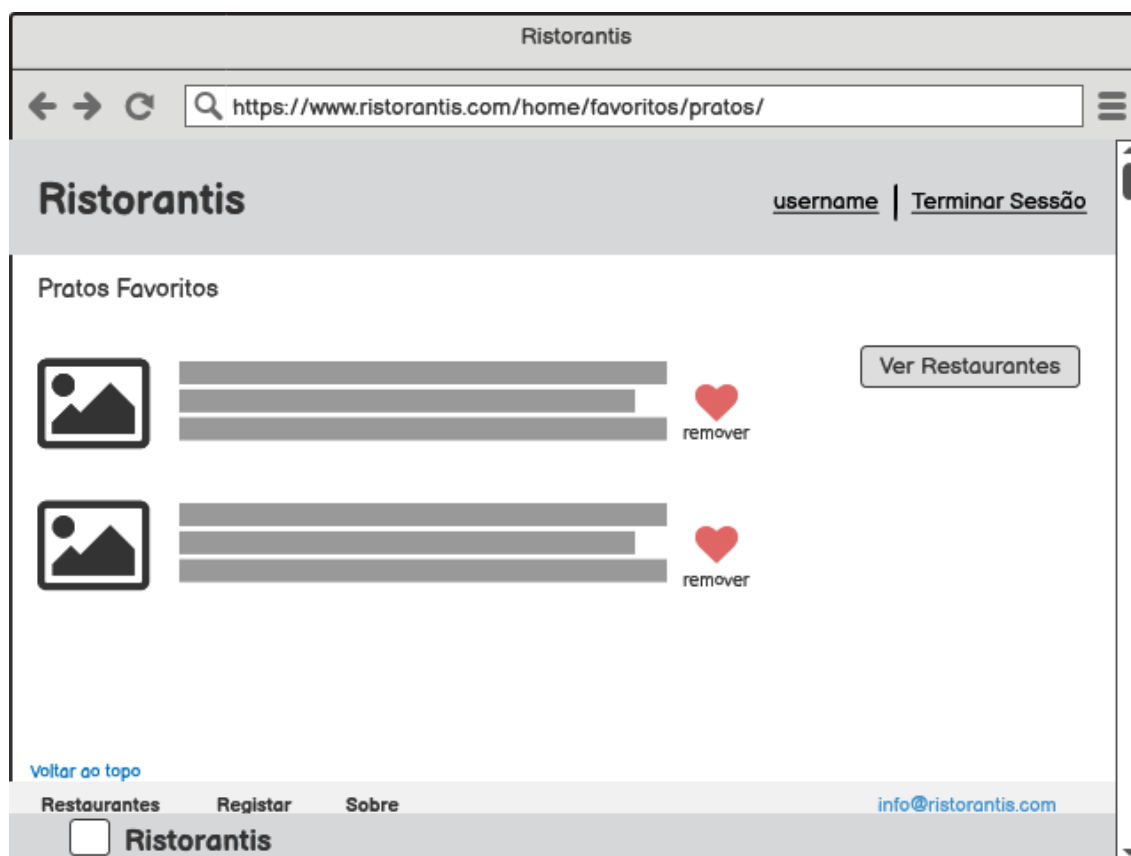
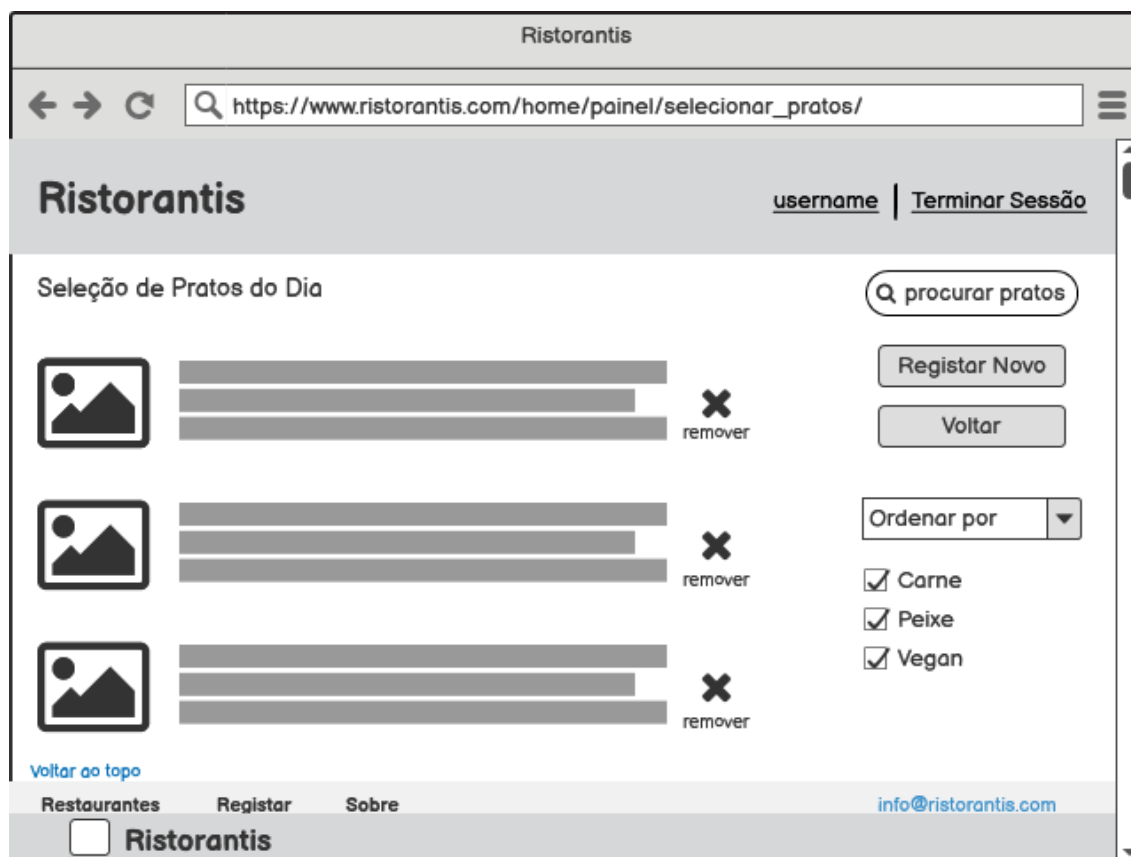
☒ Clientes
☐ Restaurantes
☐ Administradores
☐ Utilizadores Bloq.

[Voltar ao topo](#)

[Restaurantes](#)
[Registrar](#)
[Sobre](#)
info@ristorantis.com

☐ Ristorantis





Ristorantis

← → ↺

🔍 https://www.ristorantis.com/home/login/

☰

Ristorantis

Iniciar sessão

Email ou username:

Palavra-passe:

Iniciar sessão

[Esqueci a Palavra-passe](#)

Restaurantes

Registrar

Sobre

info@ristorantis.com

☐ Ristorantis

Ristorantis

← → ↺

🔍 https://www.ristorantis.com/home/

☰

Ristorantis


Entrar

Registrar


Sugestões de Restaurantes

🔍 procurar restaurante


Ver Pratos do Dia



🤍



🤍



🤍

[Voltar ao topo](#)

Restaurantes

Registrar

Sobre

info@ristorantis.com

☐ Ristorantis

Ristorantis

←
→
↺

☰

Ristorantis
username | Terminar Sessão

Bem vindo Nome!

Pedidos de Restaurantes

Gerir Utilizadores

Alterar Dados

Restaurantes
Registrar
Sobre
info@ristorantis.com

☐
Ristorantis

Ristorantis

←
→
↺

☰

Ristorantis
username | Terminar Sessão

Bem vindo Nome!

Selecionar Pratos do Dia

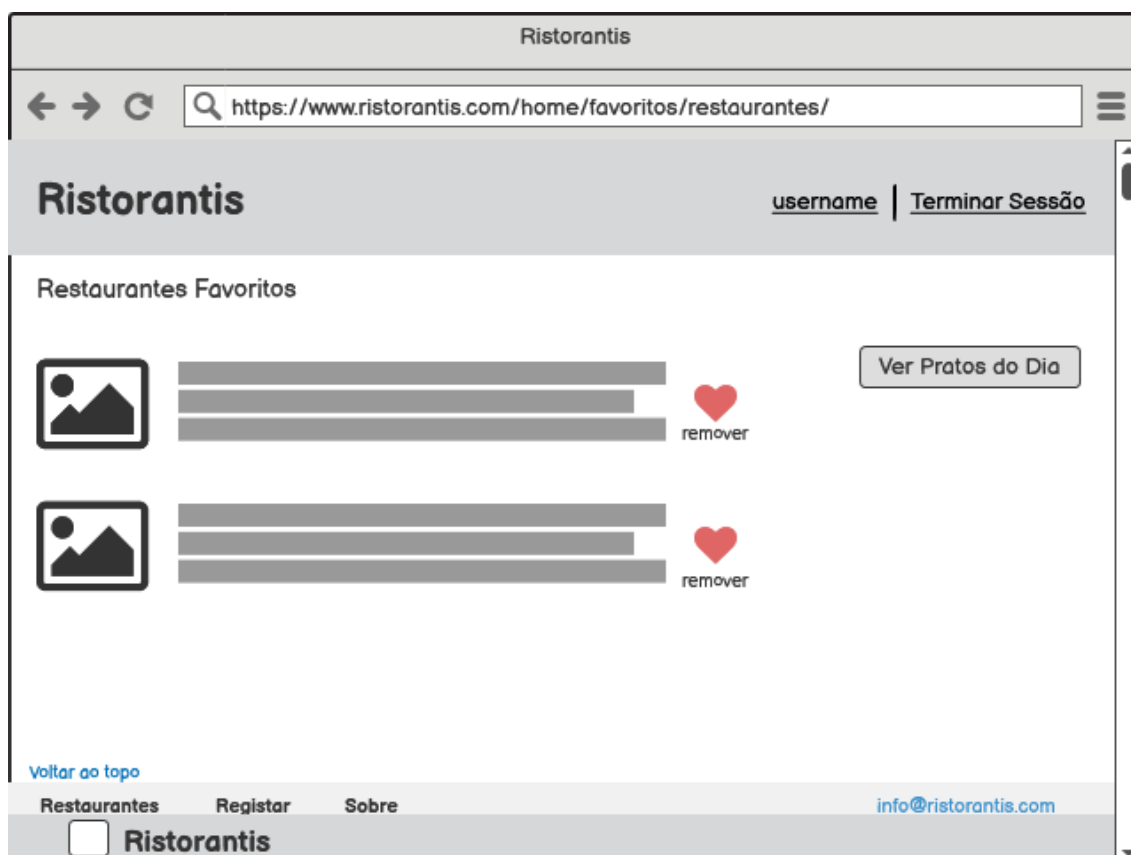
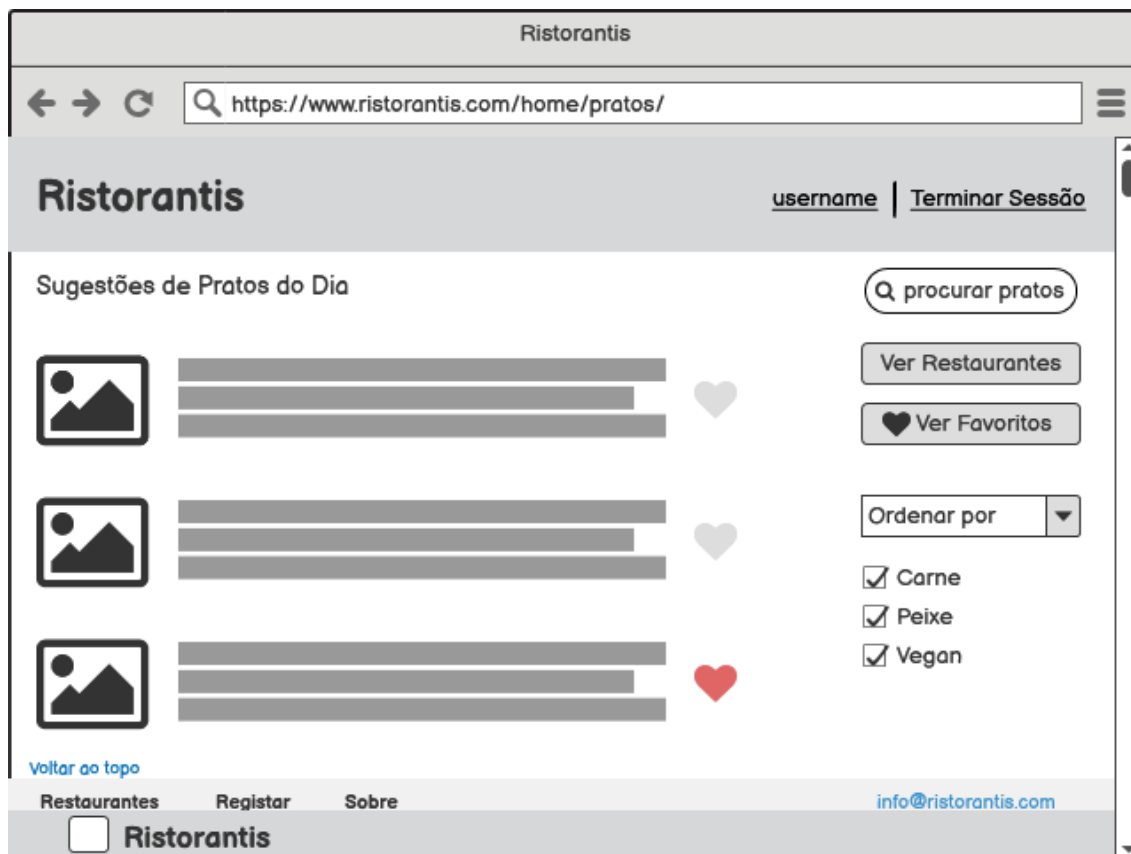
Registrar Novo Prato

Alterar Dados Conta

Alterar Dados Restaurante

Restaurantes
Registrar
Sobre
info@ristorantis.com

☐
Ristorantis



Ristorantis

[←](#)
[→](#)
[↻](#)

Ristorantis

Recuperar Palavra-passe

Email:

[Voltar atrás](#)

Restaurantes Registrar Sobre info@ristorantis.com

☐ Ristorantis

Ristorantis

[←](#)
[→](#)
[↻](#)

Ristorantis


[username](#) | [Terminar Sessão](#)

Registar Prato

* Nome:
* Tipo:
* Preço:
* Descrição:

* Dia Disponibilidade:

NOVEMBER 2020						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5
6	7	8	9	10	11	12

Fotografia: 

Restaurantes Registrar Sobre info@ristorantis.com

☐ Ristorantis

Ristorantis

[←](#)
[→](#)
[↻](#)

Ristorantis

Criar Conta

* Nome:

* Email:

* Nome de utilizador:

* Palavra-passe: ?

☐ Concordo com os [Termos de Uso](#) e [Política de Privacidade](#).

ou

[Restaurantes](#)
[Registrar](#)
[Sobre](#)
info@ristorantis.com

☐ Ristorantis

Ristorantis

[←](#)
[→](#)
[↻](#)

Ristorantis


Registrar Restaurante

* Telefone:

* Morada:

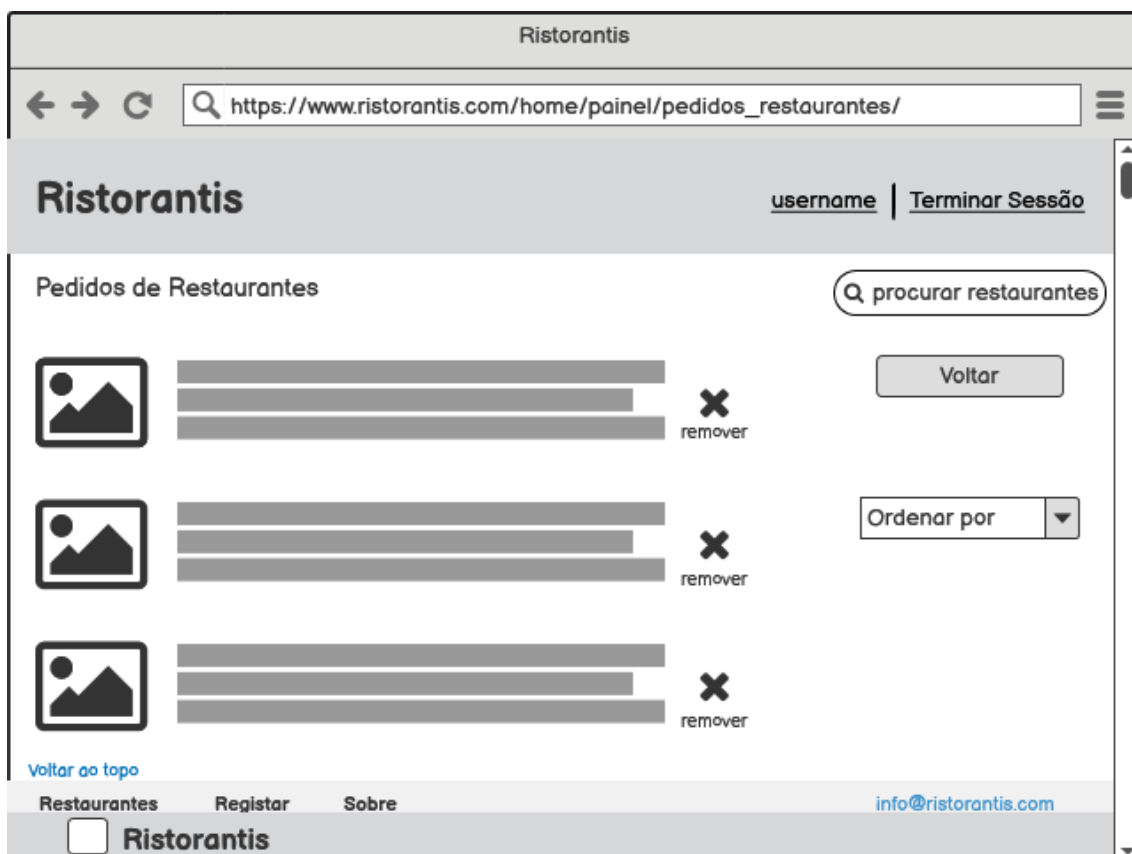
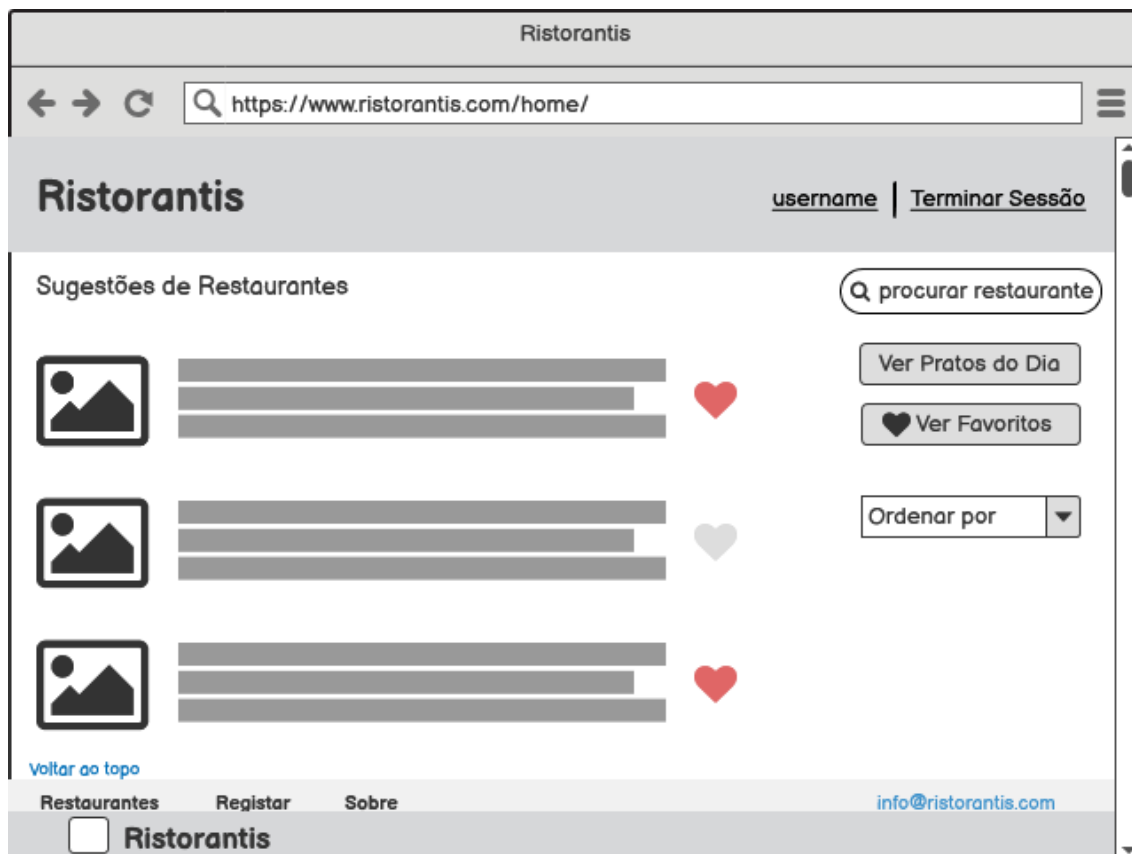
* Código Postal:

* Localidade:



[Restaurantes](#)
[Registrar](#)
[Sobre](#)
info@ristorantis.com

☐ Ristorantis



Bibliografia

- Gouveia, J., Acetatos das Aulas Teóricas. UTAD, Vila Real.
- Martins, P. (2020). Acetatos das Aulas Teóricas. UTAD, Vila Real.
- WIKIPÉDIA. Desenvolvido pela Wikimedia Foundation Diagrama E-R. Disponível em https://pt.wikipedia.org/wiki/Modelo_entidade_relacionamento. Acesso em 26 de outubro de 2020.
- LUCIDCHART. Diagrama Entidade - Relacionamento. Disponível em <https://www.lucidchart.com/pages/pt/o-que-e-diagrama-entidade-relacionamento>. Acesso em 26 de outubro de 2020.
- Wikipédia: Requisito. Disponível em <https://pt.wikipedia.org/wiki/Requisito>. Acesso em de outubro de 2020.