

Aplicações Web 2024

Construção de uma aplicação Web



Projeto Final

Época Normal (PP1, PP2 e Exame) e de Recurso

Índice

1	Introdução	3
2	Fases e Tecnologia	3
3	A Aplicação Web	3
4	O Website (Extra)	4
5	Sugestões	5
6	Regras no desenvolvimento do projeto	5
6.1	Regras de implementação e codificação	5
6.2	Constituição de grupos	6
6.3	Submissões do projeto	6
7	Avaliação	7
7.1	Regras de Avaliação	7
7.2	Critérios de Avaliação	8
8	Anexos	9
8.1	Sistema de Gestão da Base de Dados (SGBD)	9
8.1.1	MySQL	9
8.2	Web Service	9
8.2.1	Abordagem RESTful	9
8.2.2	Comunicação entre a SPA e o serviço RESTful	10
8.2.3	Propostas dos Projetos	11

1 Introdução

O principal objetivo deste projeto é utilizar HTML, CSS e JavaScript, e uma linguagem de programação do lado do servidor (Php ou outra) para desenvolver uma solução Web que permita a criação de uma aplicação Web sobre os temas em anexo sorteados aos grupos. A aplicação Web deverá compreender todas as técnicas e seguir as boas práticas do desenvolvimento de código faladas nas aulas e ser possível realizar todas as funcionalidades principais indicadas neste enunciado e outras dependentes.

2 Fases e Tecnologia

A aplicação Web deve ser desenvolvida segundo uma **metodologia *Single-Page Application (SPA)*** para a vertente de Website de apresentação do serviço e o back-office para as funções administrativas, ou seja, ter uma página em HTML5 em que o conteúdo vai sendo gerado e atualizado dinamicamente mediante a interação do utilizador com o sistema. A aplicação deverá utilizar um sistema de navegação entre páginas mais comum (navegação através de *hyperlinks*).

Numa **primeira fase**, o principal foco do projeto é o desenvolvimento de uma interface HTML/CSS que possibilite uma interação limpa e simples com as principais funcionalidades.

Seguidamente, numa **segunda fase**, pretende-se que os alunos explorem as capacidades das linguagens de programação *Javascript* relativas à manipulação de elementos do DOM e que adicione as funcionalidades de CRUD (*Create, Read, Update, Delete*) para lidar com os dados na vertente *client-side*, guardando esses dados em tempo real e recorrendo a listas de objetos em tecnologia *JavaScript*.

Ficando para uma **última fase** a ligação deste sistema a uma base de dados, com manipulação dos dados através de “roteamento” dos diversos pedidos efetuados pelos clientes (*browsers*), mediante a adoção de uma tecnologia *server-side*. Pretende-se que os alunos construam uma camada de acesso a dados recorrendo à tecnologia de serviços Web (Web Services), neste caso implementando a popular norma RESTful em detrimento de protocolos mais antigos RPC (*Remote Procedure Call*) ou SOAP (*Simple Object Access Protocol*). O servidor Web e respetivos serviços deverão ser implementados em linguagem Node.js, sendo que o repositório de dados será criado recorrendo ao SGBD MySQL¹.

3 A Aplicação Web

Apresentam-se aqui, de forma resumida, as especificações e requisitos básicos que deverão ser consideradas pelos alunos de forma a implementarem a Aplicação Web.

Para uma gestão realista de um determinado tema, terá de existir um foco principal na adequação dos conceitos que o sistema pretende suportar. Como tal, as **possíveis entidades** a considerar aqui como exemplo serão as seguintes, **podendo existir outras**, dependendo das funcionalidades que se queira implementar:

- **Utilizadores**
- **Privilégios**

¹ Sugere-se o *software* MySQL Workbench como interface gráfica para se trabalhar com o repositório de dados.

- **CRUD das funções principais**
- **Segurança (Autenticação, controlo de sessão e criptografia)**

O sistema deverá permitir a criação, visualização, edição e eliminação de *Categorias*. Para cada uma dessas categorias deverá ser possível criar (com um título ou nome e uma descrição) e os respectivos *identificadores*, que vão estar sempre associados a uma determinado *Categorias*.

Na gestão dos *Utilizadores*, o sistema deverá permitir a criação, edição e remoção de utilizadores. A data de inscrição desse utilizador no sistema não deverá aparecer na listagem, aparecendo apenas o tempo ao qual já se encontra registado na plataforma (ex.: 15 minutos, 1 hora, 2 dias, 3 semanas, 5 meses, etc), que deverá ser calculada mediante a data de inscrição de cada utilizador. Devem existir diferentes tipos de utilizador com permissões distintas (privilégios). É necessário criar no mínimo 3 tipos de utilizadores, o administrador do sistema, o gestor de conteúdo e outros que se adequa com o seu tema. Cabe ao administrador as funções de gestão de utilizadores, moderação das categorias e controlo e configuração da plataforma. O gestor de conteúdos pode apenas criar e moderar os seus conteúdos e o visitante/clientes poderá apenas usar um recurso, comentar e reportar algum algo.

Como referido anteriormente, o intuito desta parte do trabalho é desenvolver um website de navegação comum e para tal será necessário fazer a gestão de qual o conteúdo a mostrar mediante as opções de um menu principal que deve responder às necessidades de base da aplicação. O clique em cada opção de menu proporcionará a criação de conteúdo através do DOM. Contudo, não é necessário que todo o conteúdo seja construído através do DOM, podendo existir algumas partes onde se poderá optar pela escrita do código HTML na página e gerir as visibilidades desses conteúdos através de *JavaScript*. Um exemplo simples dessa gestão de visibilidades poderá ocorrer com os formulários para inserção dos dados das diversas entidades.

Em relação ao Layout, o sistema deve seguir as diretrizes indicadas nas aulas teóricas e laboratoriais que permitam ter uma página *Responsive*. Em termos de interfaces gráficas, **cada grupo deverá optar pela opção gráfica que achar mais adequada, desde que se mantenha fiel aos objetivos e requisitos de base propostos para este projeto, sendo esta alvo do processo de avaliação.**

4 O Website (Extra)

O objectivo deste website é o de colocar os serviços disponíveis da aplicação a dispor de internautas na internet.

É pretendido recorrer à utilização da API HTML de *drag & drop* e de *Javascript* para que seja possível arrastar um elemento da página para um outro elemento existente. Quando essa acção de movimento terminar deverá ser executado um código *Javascript* de validação dos sobre a solicitação. Por exemplo, registo a página.

A componente visual desta aplicação ficará a cargo do aluno, mas será importante que não haja uma diferença estética muito acentuada em relação ao restante website. Não é permitido usar templates prontos, durante a defesa o professor irá se certificar disso, no entanto, podem usar alguns recursos do Framework bootstrap.

5 Sugestões

As interfaces de gestão das diversas entidades podem seguir uma metodologia como a utilizada nos laboratórios dedicados a *JavaScript*. Conseguindo ter um código mais versátil, e tendo em consideração as propriedades existentes em cada tipo de objeto, é possível minimizar o trabalho para a gestão dessas entidades.

Outro aspeto a ter em atenção é que sempre que se pretende criar uma entidade, os campos do formulário devem estar limpos para que não perdem dados inseridos numa criação/edição anterior (caso se opte por mostrar/esconder formulários pré-existent no HTML).

Poder-se-á optar, ou não, por usar uma solução de *web framework* JavaScript para servidor como o framework “Express”, explicado em aula dedicada ao tema.

O módulo MySQL para o Php ou Node.js e outras, terá de ser instalado com o intuito de criar conexões à base de dados MySQL previamente instalada. Aqui será necessário ter-se os dados de acesso ao servidor de base de dados.

Poder-se-á sentir a necessidade de usar extensões do Visual Studio Code para ajudar com JSON, SQL, XML, etc. Os alunos poderão usar o que acharem conveniente para melhorar a sua produtividade e a qualidade do trabalho final.

6 Regras no desenvolvimento do projeto

6.1 Regras de implementação e codificação

O projeto deverá ser desenvolvido segundo as seguintes etapas e tecnologias:

- **Etapas 1** - HTML, CSS e JavaScript para interatividade e gerir entidades com geração de objetos por omissão e a definição da IU (do website e da aplicação web)
 - — **Equivalente a Primeira Prova Parcelar.**
- **Etapas 2** - Construção de uma base de dados em SQL e das classes principais php para alimentar as páginas que apresentam as funcionalidades principais do projecto;
 - — **Equivalente a Segunda Prova Parcelar.**
- **Etapas 3** - Construção de uma base de dados completo em SQL, de todas as classes php e um Webservice RESTful em php ou Node.js, que irá interagir com a SPA através da tecnologia AJAX ou php para a implementação de todas as funcionalidades;
 - — **Equivalente ao Exame Normal, Recurso e Especial.**

Os grupos deverão colocar em prática os conceitos fundamentais da programação para a Web, base de dados e do paradigma de programação orientada por objetos.

É necessário que o projeto cumpra o que é pedido no seu enunciado, **sendo deixado ao critério dos alunos qualquer aspeto de implementação que não seja referido no mesmo, devendo ser apresentado ao respetivo docente de laboratório e devidamente documentado.**

Obviamente que se pretende que a aplicação utilize um modelo suficientemente genérico que possibilite a sua utilização com as mais variadas funcionalidades, **mas existindo algumas especificidades que teriam que ser consideradas, gerando uma possível complexidade adicional.**

6.2 Constituição de grupos

Cada projeto deverá ser elaborado por um grupo de dois alunos, podendo eventualmente ser elaborado individualmente (devendo ter aprovação prévia do respetivo docente de laboratório). Não serão permitidos grupos com mais do que dois alunos, ou grupos formados por alunos de turmas de laboratório que não tenham o mesmo docente.

6.3 Submissões do projeto

O projeto deverá ser entregue até à data limite especificada por via exclusivamente classroom.

Existem dois momentos de entrega:

- **Fase 1: até às 23:55 do dia 29 de Abril de 2024.**
 - Um documento resumo denominado AW_F2R_Grupo_Número que apresente:
 - as funcionalidades e tipos de utilizadores, prints dos ficheiro e estrutura das pastas;
 - os requisitos funcionais de base a implementar;
 - mockups correspondentes;
 - Templates interativo do website e da Aplicação web das funcionalidades principais numa pasta zip denominado AW_F2C_Grupo_Número;
- **Fase 2: até às 23:55 do dia 19 de Maio de 2024.**
 - Um documento resumo atualizado denominado AW_F2R_Grupo_Número que apresente:
 - as funcionalidades, suas estrutura e tipos de utilizadores;
 - modelo relacional;
 - diagrama de classes e de casos de usos;
 - lista de possíveis frameworks/APIs a usar e com que objetivo (não é obrigatório o uso de qualquer framework);
 - prints dos códigos php das funcionalidades principais e das telas alimentadas com dados;
 - Ficheiro zip denominado AW_F2C_Grupo_Número contendo:
 - Ficheiro readme: que apresenta as configurações de instalação e configuração do projecto;
 - Pasta contendo o projecto completo;
 - Ficheiro Sql com as estruturas das tabelas, dados e as constraints.
- **Fase 3: até às 23:30 16/06/2024.**
 - Esta será a entrega final e deverá incluir o projeto completo, englobando todas as etapas. O que foi entregue na fase 1 poderá ter sido retificado para esta fase final. Os materiais do projeto deverão incluir:

- Pasta do projeto pronta a ser aberta e executada em VSCode. Deverá ter na raiz dessa pasta:
 - Ficheiro `package.json` com a descrição do projeto, em particular as suas dependências em termos de módulos, permitindo o seu carregamento através de “`npm install`”.
 - Ficheiro `app.js` com a implementação do *web server*;
 - Pasta `.vscode` com os ficheiros `launch.json` e `tasks.json`;
 - Pasta `www` com os ficheiros de implementação do programa: ficheiros `*.html`, pasta `images` para conter as imagens/ícones, pasta `scripts` para conter os ficheiros JavaScript e pasta `styles` para conter os ficheiros CSS.
 - Ficheiro, na linguagem SQL, com todo o código necessário para criação de todas as tabelas necessárias à solução, modelo entidade-relação. **Deverá existir também a inserção de linhas com dados de exemplo que permitam o teste da aplicação.**
- Os ficheiros JavaScript deverão ser documentados através de JSDoc.

Todos os ficheiros que compõem o projeto deverão estar guardados num único ficheiro compactado no formato ZIP, cujo nome deverá seguir a seguinte estrutura:

`AW_F<numFase><C/R>_G<numGrupo>_<numAluno1>_<numAluno2>.zip`. **Nota: o não cumprimento desta regra implica a dedução de 3 valores na nota final do projecto que o grupo tirar.**

7 Avaliação

7.1 Regras de Avaliação

- A classificação do programa terá em conta a qualidade da programação (fatores de qualidade do *software*), a estrutura do código criado segundo os princípios da programação orientada por objetos, tendo em conta conceitos como a coesão de classes e métodos, o grau de acoplamento entre classes e o desenho de classes orientado pela responsabilidade, e a utilização/conhecimento das linguagens envolvidas.
- Serão premiadas a facilidade de utilização, a apresentação, a imaginação e a criatividade.
- Após a entrega haverá uma discussão oral para validação da nota final da componente Projeto.
- Os alunos que não comparecerem à discussão/defesa serão classificados com zero. Nesta discussão com os alunos poderá ser apurada a capacidade do aluno de produzir o código apresentado. A nota atribuída será zero nos casos em que essa capacidade não for demonstrada.
- Após a discussão, a nota do Projeto será atribuída individualmente a cada um dos elementos do grupo.
- A avaliação oral será realizada pelo respetivo docente de laboratório e irá ser feita uma marcação prévia para cada grupo de trabalho.
- A apresentação de relatórios ou implementações plagiadas leva à imediata atribuição de nota zero a todos os trabalhos nessas condições, quer tenham sido o original ou a cópia.

- No rosto do relatório e nos ficheiros de implementação deverá constar o número, nome e turma dos autores e o nome do docente a que se destina.

7.2 Critérios de Avaliação

1. Funcionalidades e operações essenciais	40%
2. Implementação técnica (utilização correta dos elementos HTML; definição e utilização de CSS; desenvolvimento do servidor, etc.)	35%
4. Documentação (bom estilo (identificadores, comentários, indentação); JSDoc, relatório; etc.)	10%
4. Segurança e Extra (autenticação, validação e utilização de bibliotecas, ferramentas ou tecnologia não lecionada e que faça sentido adicionando real valor ao projeto <u>desde que devidamente justificada pelo aluno</u> ; novos requisitos/funcionalidades, realização da etapa extra; etc.)	15%

8 Anexos

8.1 Sistema de Gestão da Base de Dados (SGBD)

Apesar de não ser o principal foco da Unidade Curricular de Aplicações Web, o sistema de gestão de base de dados irá ser uma componente da máxima importância no projeto e, como tal, terá o respetivo peso na nota final da segunda fase. E até porque em termos profissionais, a maior parte dos projetos de engenharia informática recorre naturalmente a um armazenamento de dados.

8.1.1 MySQL

Trata-se de um SGBD relacional que usa SQL (Structured Query Language), sendo o seu código fonte aberto aos utilizadores. Neste momento é dos sistemas mais usados do mercado e para isso contribui uma base de documentação relativamente completa. Outro dos motivos da escolha deste SGBD é o facto de ele ter sido utilizado nas anteriores unidades curriculares “Bases de Dados 1 e 2”. Poderá efetuar o download do servidor no seguinte URL: <https://dev.mysql.com/downloads/installer/>.

O procedimento de instalação é simples e intuitivo, contudo tenha atenção ao passo de escolha do “porto” de ligação ao SGBD, do nome do utilizador de acesso (por default costuma ser root) e a palavra-passe definida para esse mesmo utilizador. Esta informação será necessária para interagir com o SGBD.

Existe um aplicativo gráfico opcional na instalação (MySQL Workbench), que poderá ser usado pelos alunos de forma a tornar mais apelativa a criação das tabelas, campos, chaves primárias, chaves estrangeiras, índices, procedimentos e funções. Tendo em conta, os conhecimentos e experiência adquirida pelos alunos na utilização do MySQL Workbench na Unidade Curricular de Bases de Dados 1 e 2, a sua utilização pode revelar-se um meio simples e rápido para o desenvolvimento da base de dados de suporte necessária ao sistema da aplicação Web.

8.2 Web Service

Um *Web Service*² é um qualquer serviço que está disponível pela internet, que usa uma linguagem de comunicação standard para troca de mensagens, e não está dependente de nenhum sistema operativo ou linguagem de programação.

8.2.1 Abordagem RESTful

Para o desenvolvimento deste projeto deverá recorrer à solução arquitetural RESTful. Apesar de se usar principalmente JSON para a troca de mensagens, é possível recorrer a outros *standards*, como por exemplo XML. Esta abordagem utiliza os métodos (verbos, códigos, recursos) HTTP para efetuar as tradicionais operações de CRUD (ver Tabela 1), não guardando estado entre pedidos.

² Para mais informação: https://en.wikipedia.org/wiki/Web_service

OPERAÇÕES	COMANDOS SQL	REST
CREATE	INSERT	POST
READ (ou RETRIEVE)	SELECT	GET
UPDATE	UPDATE	PUT e/ou PATCH
DELETE (ou DESTROY)	DELETE	DELETE

Tabela 1 - Relação entre “Operações CRUD”, “Comandos SQL” e “Verbos HTTP”.

8.2.2 Comunicação entre a SPA e o serviço RESTful

A arquitetura que se pretende implementar poder ser visualizada na Figura 1.

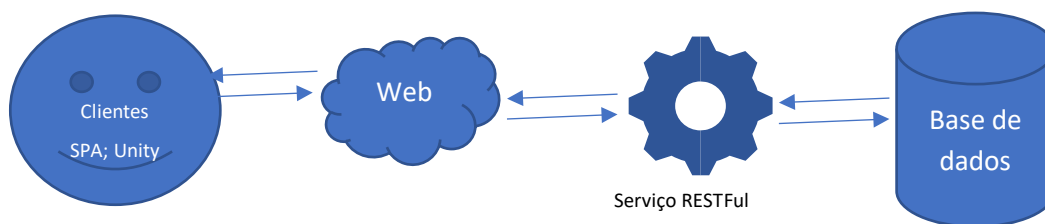


Figura 1 - Arquitetura.

O objetivo será que através do serviço RESTful se consiga a informação que irá iniciar os objetos das classes que já foram desenvolvidas anteriormente.

Caso tenha construído os objetos por omissão para testar a SPA numa fase anterior, terá de se remover a geração e criar-se as respetivas funções auxiliares recorrendo à tecnologia AJAX (*Asynchronous JavaScript and XML*). Serão efetuados os pedidos necessários por parte do cliente ao serviço RESTful. Abaixo pode-se observar um exemplo de teste para verificar se é possível carregar o recurso uma determinada entidade.

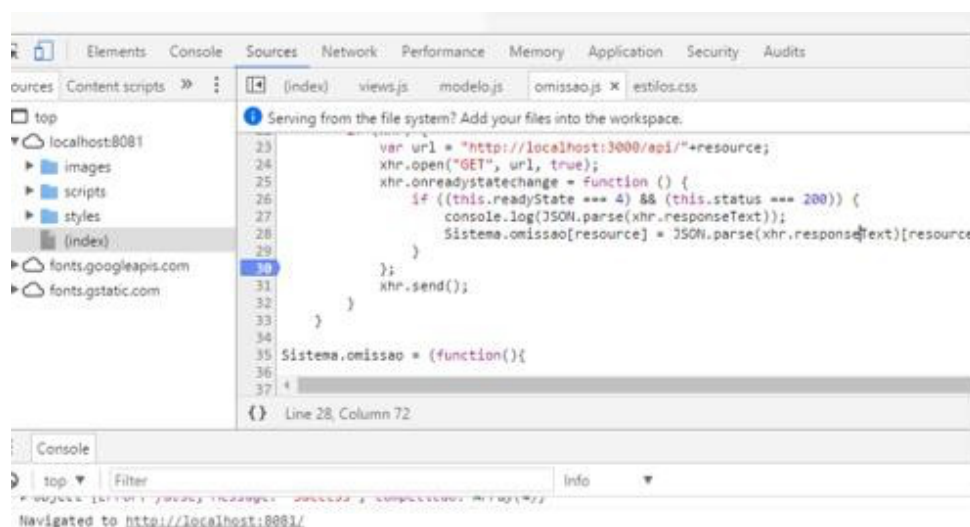


Figura 2 - Exemplo de consumo do serviço RESTful por parte do cliente (ambiente de debug do Chrome).

8.2.3 Propostas dos Projetos

	Descrição do sistema	Grupo
1	Controle de hospedagens em hotéis (quartos/hóspedes/reservas/facilidades) <ul style="list-style-type: none"> - Cadastro de quarto (Número, andar, qtidade de camas, tipo, facilidades, valor) - facilidade (nome, qtidade) - hóspede (nome, endereço, telefone, nif,...) - reservas (**hóspede, **quarto, situação, data inicio, data fim) 	Grupo 1
2	Sistema de controle de imobiliária (locador/locatário/imóveis) <ul style="list-style-type: none"> - imovel (identificacao, resumo, endereco, qtde_quartos,tamanho, finalidade, aluguel/venda), **locatario) - locador (Nome, endereco, telefone, nif) - locatário (Nome, endereco, telefone, nif) - contrato (tipo, **locador, **locatario, **imovel, duracao (sealuguel)) 	Grupo 6
3	Sistema de controle de transportadora (Caminhão/Motorista/Viagens) <ul style="list-style-type: none"> - caminhao (nome, ano fabricacao, Km rodados, marca, **pneus) - motorista (nome, endereco, telefone, numero cartMotorista,nif) - pneu (marca, duracao) - viagem (**caminhao, **motorista, origem, destino, Km, datainicio, data fim) 	Grupo 2
4	Sistema de controle de vestibular (cursos/candidatos/salas/prova) <ul style="list-style-type: none"> - curso (nome, qtde vagas, turno) - candidato (nome, endereco, telefone, mail, **curso, nif, data nascimento, **prova) - sala (numero, predio, capacidade) - prova (**aluno, questao, alternativa) - gabarito (questao, alternativa correta) 	Grupo 4
5	Sistema de controle de horários médicos (médicos/exames/pacientes/convenio) <ul style="list-style-type: none"> - medico (nome, CRM, telefone) - exames (nome, sigla) - pacientes (nome, endereco, telefone, **convenio) - convenio (nome, sigla) - consulta (**paciente, **exame, **medico, HORARIO, DIA); - exames (nome, sigla) - pacientes (nome, endereco, telefone, **convenio) - convenio (nome, sigla) - consulta (**paciente, **exame, **medico, HORARIO, DIA); 	Grupo 5

6	Sistema de locadora de veículos (veículos/opcionais/locadores) <ul style="list-style-type: none"> - veiculo (marca, nome, placa, anoFabr, Km, **opcionais, valor) - opcionais (nome, valorExtra) - locatario (nome, endereco, telefone, CPF, cart Motorista) - locacao (**carro, **locatario, data_inicio, data_fim, KmAndados) 	Grupo 11
7	Sistema de controle de trabalhos em faculdade (alunos/trabalhos/disciplinas) <ul style="list-style-type: none"> - disciplina (codigo, nome, ano) - aluno (nome, matricula) - disciplina_X_aluno (**aluno, **disciplina) - trabalhos (**aluno, **disciplina, data_entrega, local_trabalho(upload)) 	Grupo 8
8	Sistema de controle de recursos e projetos (Projetos/Clientes/Recursos) <ul style="list-style-type: none"> - cliente (nome, CNPJ/nif, endereco, telefone) - projeto (nome, **cliente, data_inicio, data_fim) - recurso (nome, telefone, horas_trabalho) - alocao (**recurso, **projeto, data_inicio, data_fim) 	Grupo 10
9	Sistema de tickets de promocoões shopping (Cliente/Loja/Ticket) <ul style="list-style-type: none"> - loja (nome, localizacao, **ramo_atividade) - ramo_atividade (nome, sigla) - cliente (nome, endereco, telefone, e-mail, CPF) - tickets (**loja, **cliente, valor, numero) 	Grupo 7
10	Sistema de controle de vagas/currículos (Vagas/currículos/empresa) <ul style="list-style-type: none"> - empresa (nome, endereco, telefone, mail) - qualificacoes (nome, sigla) - vaga (**empresa, **qualificacoes, salario, data_inicio, data_fim, ativo(S/N)) - curriculo (nome, endereco, telefone, e-mail, **qualificacoes) - solicitacaoVaga (**curriculum, **vaga) 	Grupo 3
11	Sistema de gestão de pontos turísticos (pontos turísticos/quartos/hóspedes/reservas) <ul style="list-style-type: none"> - Cadastro de quarto (Número, andar, qtidade de camas, tipo, facilidades, valor) - turistas (nome, endereço, telefone, nif, pais...) - reservas (**hóspede, **quarto, situação, data inicio, data fim) - pontos turisticos (nome, localização, estado, gastronomia, lingua) - localização (nome, descrição, coordenadas geograficas) - turistas (nome, pais, nif) - hoteis (nome, classificação, serviços, quartoDisponiveis) 	Grupo 9