



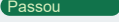



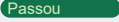
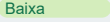


Funcionalidade	Descrição	Ações a executar	Resultado esperado	Estado	Prioridade	Observações
FC-01	Leitura do ficheiro CSV disponibilizado, com os dados relativos a Códigos Postais, e Concelho e Distrito incompletos	Utilizar a função <code>zipcode_infos_from_csv</code> , passando o caminho do ficheiro CSV	Criação de lista de ZipcodeInfo	<div>Passou</div>	<div>Baixa</div>	Utilização de classe ZipcodeInfo onde é guardado o Código Postal, com validação dos dados passados na criação de cada instância.
FC-02	Busca de informações no sistema de caching local	Utilização da função <code>get</code> , com o URL pretendida, e o caminho do ficheiro de cache	Devolve a informação relativa ao URL	<div>Passou</div>	<div>Baixa</div>	Os acessos a sites externos são guardados num cache local, para evitar que os mesmos neguem o serviço após um determinado número de pedidos, dentro de um determinado período. Se o ficheiro relativo ao URL passado não existir em cache local, o sistema descarrega as informações a partir do URL e guarda no caminho especificado. Posteriormente, devolve a informação em cache local.
FC-03	Busca com mecanismo de <i>fallback</i> , onde são utilizadas diferentes fontes de informação (ZipInfoExtractor)	Iterar sobre uma lista de ZipInfoExtractor , e executar a função <code>fetch_info</code> associada, passando um objecto do tipo ZipcodeInfo	A primeira instância a encontrar informações sobre o Código Postal pedido devolve essas informações na forma de um objecto do tipo LocaleInfo	<div>Passou</div>	<div>Baixa</div>	Se um elemento da lista não devolver um objecto do tipo LocaleInfo , será utilizado o elemento seguinte, e assim sucessivamente até que algum elemento encontre e devolva. Caso nenhum elemento devolva LocaleInfo , o processo avança para o próximo código postal.
FC-04	Busca de informação associada a cada Código Postal (Distrito e Concelho), no site externo https://www.codigo-postal.pt	Utilizar a função <code>fetch_info</code> , passando o código postal	Busca de informações de Concelho e Distrito	<div>Passou</div>	<div>Média</div>	As buscas neste site são feitas a partir de <i>parsing</i> do código HTML, utilizando a framework BeautifulSoup . A busca pode falhar por negação de serviço, devolver a informação pretendida, ou pode não encontrar informações sobre o Código Postal pretendido.
FC-05	Busca de informação associada a cada Código Postal (Distrito e Concelho), no site externo https://www.cttcodigopostal.pt	Utilizar a função <code>fetch_info</code> , passando o código postal	Busca de informações de Concelho e Distrito	<div>Passou</div>	<div>Média</div>	O site disponibiliza uma API que devolve objectos no formato JSON . O pedido pode falhar por negação de serviço, devolver a informação pretendida, ou pode resultar em informação nula, na forma de um array vazio, no caso de não existir informação relativa ao Código Postal pretendido.
FC-06	Inserção dos dados enriquecidos num Dictionary	Utilização do código <code>info_map[zip_info]=res</code> , <code>info_map</code> é um Dictionary , <code>zip_info</code> é um objecto do tipo ZipcodeInfo , e <code>res</code> é um objecto do tipo LocaleInfo	Inserção no Dictionary com sucesso	<div>Passou</div>	<div>Baixa</div>	
FC-07	Escrita do ficheiro CSV enriquecido, onde consta uma listagem de Códigos Postais, e o Concelhos e Distritos relativos.	Utilização da função <code>dump_infos_to_csv</code> , passando o caminho do ficheiro onde pretendemos guardar os dados, e um Dictionary com as informações relativas aos Códigos Postais, Concelhos e Distritos	Criação do ficheiro CSV	<div>Passou</div>	<div>Alta</div>	Apenas são guardados os dados relativos aos Códigos Postais sobre os quais foi possível obter a informação relativa ao Distrito e Concelho.
FC-08	Escrita de ficheiro de texto, onde são registados os Códigos Postais sobre os quais não foi possível obter a informação relativa ao Concelho e Distrito.	Utilização da função <code>io.write</code> , passando uma String onde consta a data e hora, e o Código Postal	Criação/escrita dos dados pretendidos no ficheiro	<div>Passou</div>	<div>Baixa</div>	A escrita no ficheiro é feita em modo append , de forma a manter os registos existentes.
FC-09	Limpeza da tabela da Base de Dados	Utilização da função <code>clear_db</code> , que faz uma query para truncar a tabela na Base de Dados	Eliminação dos dados pre-existent na tabela da Base de Dados	<div>Passou</div>	<div>Alta</div>	Utilizo a framework mysql-connector-python para fazer interações com a Base de Dados num servidor MySQL. Para evitar a inserção de dados duplicados, realizo a limpeza da tabela, onde posteriormente serão inseridos os dados enriquecidos.
FC-10	Inserção dos dados enriquecidos na Base de Dados a partir do código do programa	Utilização da função <code>insert_into_db</code> , passando um Dictionary com chave do tipo ZipcodeInfo e valor do tipo LocaleInfo	Os dados são inseridos na Base de Dados	<div>Passou</div>	<div>Alta</div>	
FC-11	Pesquisa de Código Postal na Base de Dados a partir do código do programa	Utilização da função <code>db_search_for_zipcode</code> , passando uma instância de ZipcodeInfo preenchida com Código Postal pretendido	Instância de LocaleInfo , onde estão as informações relativas ao Concelho e Distrito	<div>Passou</div>	<div>Baixa</div>	Caso não exista um registo associado ao Código Postal passado, a função devolve None

Funcionalidade	Descrição	Ações a executar	Resultado esperado	Estado	Prioridade	Observações
FC-12	Pesquisa de Código Postal através de pedido HTTP ao nosso programa, passando um Código Postal válido , e o qual consta na Base de Dados	<p>Aceder ao endereço/porta onde o programa está a ser executado, e fazer o pedido HTTP com o parâmetro codigo_postal válido.</p> <p>Por exemplo: http://localhost:5000/?codigo_postal=9675-069 </p>	Resposta do servidor HTTP com status code 200 , e o objecto no formato JSON , com as informações relativas ao Concelho e Distrito associadas ao Código Postal pedido			<p>Para criar o servidor HTTP, utilizei a framework Flask.</p> <p>No ambiente de desenvolvimento, o endereço/porta são localhost:5000</p> <p>O Código Postal deve ser enviado através do parametro codigo_postal, no formato <cp4>-<cp3>, em que cp4 é um número com 4 dígitos, e cp3 é um número com 3 dígitos.</p>
FC-13	Pesquisa de Código Postal através de pedido HTTP ao nosso programa, passando um Código Postal válido , mas o qual não consta na Base de Dados	<p>Aceder ao endereço/porta onde o programa está a ser executado, e fazer o pedido HTTP com o parâmetro codigo_postal válido.</p> <p>Por exemplo: http://localhost:5000/?codigo_postal=1234-123 </p>	Resposta do servidor HTTP com status code 404 , e um objecto no formato JSON , onde existe uma mensagem a informar que aquele Código Postal não consta na Base de Dados			
FC-14	Pesquisa de Código Postal através de pedido HTTP ao nosso programa, passando um Código Postal inválido	<p>Aceder ao endereço/porta onde o programa está a ser executado, e fazer o pedido HTTP com o parâmetro codigo_postal inválido.</p> <p>Por exemplo: http://localhost:5000/?codigo_postal=aaaa </p>	Resposta do servidor HTTP com status code 400 , e um objecto no formato JSON , com uma mensagem a informar que aquele Código Postal é inválido, e a informar qual o formato correcto			
FC-15	Pedido HTTP ao nosso programa através do endereço/porta, sem passar o parâmetro codigo_postal	<p>Aceder ao endereço/porta onde o programa está a ser executado, sem enviar o parâmetro codigo_postal</p> <p>Por exemplo: http://localhost:5000</p>	Resposta do servidor HTTP com status code 400 , e um objecto no formato JSON , com uma mensagem a informar acerca do parâmetro codigo_postal em falta, e o seu formato correcto.			
FC-16	Pedido HTTP ao nosso programa através do endereço/porta, passando alguns parâmetros, mas sem passar o parâmetro codigo_postal	<p>Aceder ao endereço/porta onde o programa está a ser executado, sem enviar o parâmetro codigo_postal</p> <p>Por exemplo: http://localhost:5000?aaa=bbb&ccc=ddd </p>	Resposta do servidor HTTP com status code 400 , e um objecto no formato JSON , com uma mensagem a informar acerca do parâmetro codigo_postal em falta, e o seu formato correcto.			O único parâmetro a ser utilizado é o codigo_postal , e apenas a sua falta ou invalidez resulta numa mensagem ao utilizador; qualquer parâmetro adicional é ignorado