

Test Document

Objecte prova

És un test unitari per testejar si la classe Document funciona correctament.

Aquesta és una classe clau en el nostre gestor de documents. Aquest test comprova que amb aquesta classe podem tractar amb Documents aplicant funcions que en canviïn el seu estat dins del gestor.

Altres elements integrats a la prova

En aquesta prova s'utilitzen bàsicament estructures de dades dels atributs del Document (`ArrayList<String>` contingut, `HashMap<String, Double>`) tot i que també apareix el `CtrlDocument` ja que aquest és qui té les stopwords que necessita la creadora d'un Document. Hem fet els tests pertinents d'aquesta classe, i per tant si trobem un error en la realització d'aquests tests assumirem que l'error es troba en aquesta classe.

Valors estudiats

Totes les proves realitzades a aquest test són de caixa blanca. És a dir, la persona que ha programat la classe i la que fa els tests són la mateixa. Per tant, el programador coneix l'estructura interna del codi i pot testar tots els casos límits que trobi per comprovar si el codi funciona correctament.

- `creadora_Document()`

L'objectiu d'aquesta funció és crear un nou objecte Document amb diferents combinacions pels paràmetres i observar que es creen bé i respecten les restriccions establertes.

En aquesta funció comparem cada atribut del document amb el valor que hauria de tindre, observant així la correctesa de la funció.

- `creadora_formatInvalid()`

L'objectiu d'aquesta funció és crear un nou objecte Document amb valors null per al format i comprovar que salta l'excepció *ExceptionFormatNoValid*. Per fer-ho es crea un document amb el format null i es comprova que aquest comportament succeeixi.

- *creadora_notprimaryKey()*

L'objectiu d'aquesta funció és crear un nou objecte Document amb valors null per a les claus primàries *títol* i *autor* i comprovar que salta l'excepció *ExceptionNotPrimaryKeys*. Per fer-ho es crea un document amb el *títol* i l'*autor* amb valor null i es comprova que aquest comportament succeeixi.

- *ModificarInformacioDocument()*

L'objectiu d'aquesta funció és crear modificar un objecte Document existent. El que es fa es passar diferents combinacions pels paràmetres i observar que el comportament de la funció és correcte i es conserva la integritat.

S'ha creat un objecte Document, amb paràmetres vàlids, i a aquest se li apliquen diferents canvis.

Tenim quatre casos: modificar només el *títol*, modificar només l'*autor*, modificar només el contingut, modificar tots tres i modificar posant valors nulls.

En els quatre primers casos, veiem com l'objecte modificat té els mateixos valors per als paràmetres que els valors desitjats i per tant, que els objectes comparats són exactament iguals.

- *modificar_notprimaryKey()*

L'objectiu d'aquesta funció es comprovar un comportament específic de la funció *ModificarInformacioDocument()*. Volem comprovar que si posem valors null als paràmetres, fem saltar l'excepció *ExceptionNotPrimaryKeys*, ja que estem violant la propietat de les claus primàries.

El que fem es crear un document amb paràmetres vàlids i cridar a la funció de modificar amb valors null per a les claus primàries. Finalment, per comprovar la correctesa d'aquest cas comprovem que salta l'excepció *ExceptionNotPrimaryKeys*.

- *actualitzar_pesos()*

En aquesta funció comprovem que el valor dels pesos que tenim per a les paraules del contingut és coherent amb el contingut que tenim després d'aplicar certes modificacions.

En el primer cas, comprovem que al inicialitzar un Document, és realitza el càlcul d'aquests pesos de manera correcte. Per comprovar-ho, creem un document amb paràmetres vàlids i cridem a la funció. Finalment, comprovem el *HashMap* obtingut, amb el *HashMap* que esperem obtenir

En el segon cas, afegim paraules (tant de repetides com noves) en el contingut del document. Per comprovar que el comportament de la funció és correcte i que s'ha actualitzat el map, cridem a la funció. Finalment, comprovem el *HashMap* obtingut, amb el *HashMap* que esperem obtenir.

El tercer cas, és com el segon però en comptes d'afegir paraules, en treiem. La resta de comprovacions són idèntiques.

En el quart cas, creem un contingut buit i comprovem que al cridar la funció, aquesta ens retorna un *HashMap* buit. Per a saber si aquest comportament és correcte comparem el *HashMap* que té un document després de la crida a la funció amb un *HashMap* buit.

Finalment, en el cinquè i últim cas, comprovem que si tenim un document amb totes les paraules iguals, s'obté un *Hashmap* amb un únic atribut amb un pes de 1. Per comprovar aquest comportament, inicialitzem un contingut amb totes les paraules iguals i repetim el procés anterior. Un cop contingut el map ja modificat, el comparem amb un *HashMap* d'una única entrada amb valor 1 per a la clau determinada.