# TreeQuery

Un documento web o una porción de documento pueden ser representados como un árbol con nodos de tipo NodoDocumento. Estos nodos tienen tres propiedades textuales:

* TipoNodo
* Id
* Clase

Los valores de estas propiedades serán siempre cadenas de texto constituidas por caracteres alfanuméricos (solo dígitos y letras). La propiedades Id y Clase pueden tener valor null, pero TipoNodo nunca será null ni vacía.

Los nodos tienen además una colección de nodos hijos, accesible mediante la propiedad pública Hijos.

Un ***selector simple*** es un texto que describe a uno o más nodos de un árbol. Los selectores pueden ser

* Selectores por identificador: comienzan con el carácter ´#´ seguido de un texto alfanumérico.
* Selectores por clase: comienzan con el carácter ´.´ seguido de un texto alfanumérico.
* Selectores por tipo de nodo: solo contienen caracteres alfanuméricos, sin ningún signo precedente.

Por ejemplo:

* El selector simple “div” corresponde a todos los nodos cuyo tipo de nodo sea “div”
* El selector simple “.resaltado” corresponde a todos los nodos cuya clase sea “resaltado”
* El selector simple “#tabla1” corresponde a todos los nodos cuyo identificador sea “tabla1”

Una ***consulta*** a un árbol está constituida por uno o varios selectores simples separados por espacios. La consulta “#seccion1 tr td” corresponde a todos los nodos de tipo “td” que sean descendientes (directos o indirectos) de algún nodo de tipo “tr” que a su vez descienda de algún nodo con id “seccion1”. En general, la consulta “sel1 sel2 sel3… seln-1 seln” denota todos los elementos que cumplan con el selector simple seln y que desciendan de un nodo que cumpla con seln-1, a su vez descendiente de un nodo que cumpla con seln-2, etc.

Implemente el método EjecutarConsulta, que a partir de un árbol como los anteriores y una consulta, devuelva un IEnumerable<NodoDocumento> con todos los nodos del árbol (en cualquier orden) que correspondan con la consulta dada.

## Aclaraciones

* Se garantiza que la consulta pasada a su método será válida y que ni ella ni el árbol tendrán valor null.
* El método EjecutarConsulta deberá poder ser ejecutado varias veces sin tener que cerrar la aplicación (preste especial atención a la inicialización de las variables static si es que decide usarlas).
* Las instancias de NodoDocumento se crean mediante la sintaxis new NodoDocumento(“tipoNodo”, “id”, “clase”, hijo1, hijo2, …), como puede observarse en la plantilla de solución.
* Para que un selector simple concuerde con un nodo, el valor de la propiedad en cuestión (TipoNodo, Id o Clase) debe coincidir exactamente con el texto indicado en el selector, incluso tomando en cuenta el uso de mayúsculas y minúsculas.

## Ejemplos

En el siguiente árbol, los nodos en que no se ha indicado el valor de las propiedades Clase o Id es porque tienen valor null.

Mostramos a continuación algunas consultas y los nodos que se obtienen como resultado.

|  |  |
| --- | --- |
| “.resaltado” |  |
| “div” |  |
| “.resaltado a” |  |
| “div span” |  |
| “#contenedor span” |  |
| “div .resaltado .atenuado” |  |

Las siguientes consultas producen como resultado secuencias vacías de nodos:

* “#idInexistente”
* “div div div”
* “.atenuado #link1”
* “#LINK1” (no coincide con la propiedad Id de ningún nodo debido a diferencias en el uso de mayúsculas/minúsculas)
* “#resaltado” (el texto “resaltado” no corresponde al valor de la propiedad Id de ningún nodo, sino a la propiedad Clase de algunos de ellos)