

Trabajo Práctico Final

Complejidad Temporal, Estructuras de Datos y
Algoritmos

Alumno: Alan Lopez

Universidad Nacional Arturo Jauretche

Comisión 2



UML del sistema.

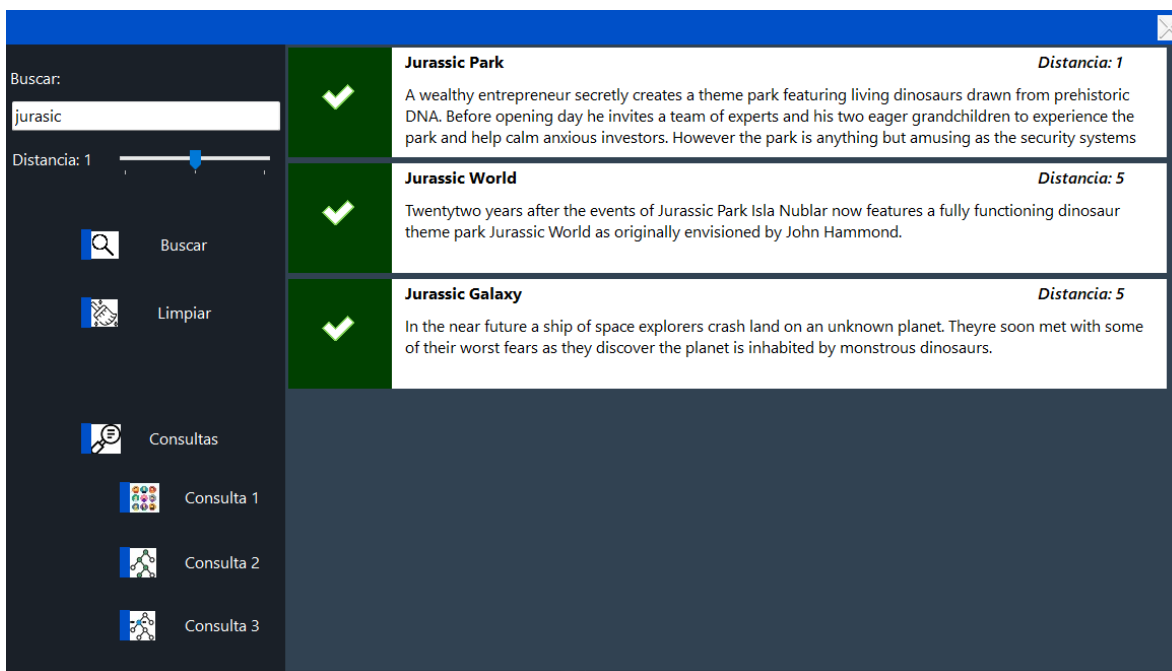
El código fue trabajado en Visual Studio.

El sistema inicia solicitando al usuario que seleccione el archivo csv donde se encuentran los recursos a indexar.

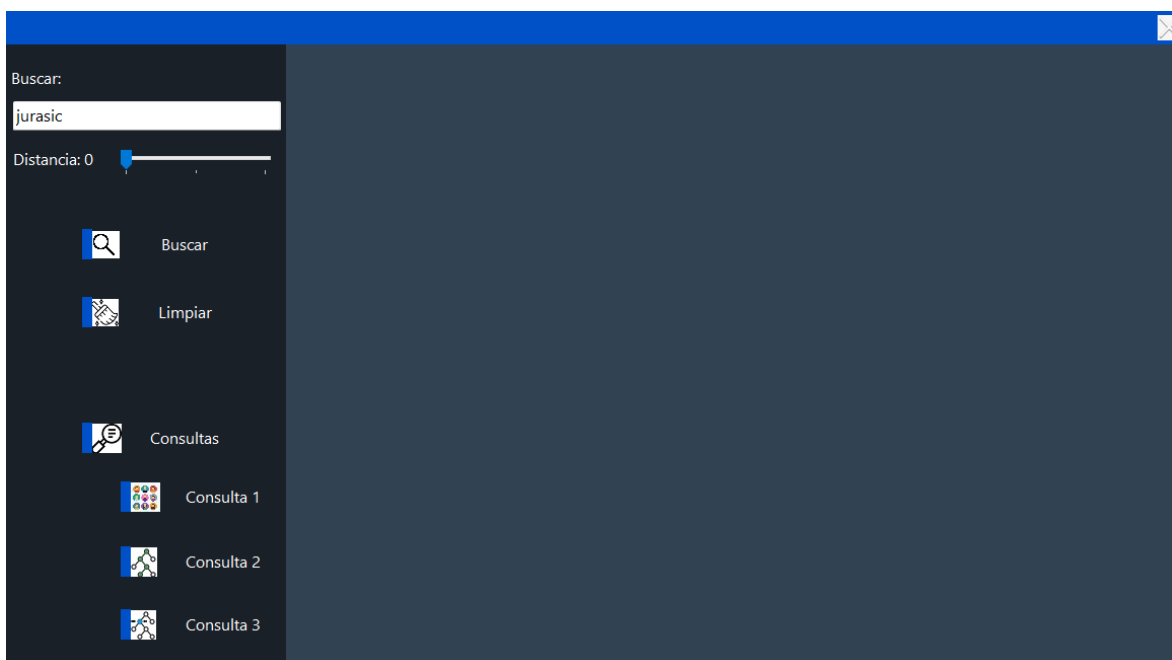
Como se indica en el UML, se nos pide crear los métodos dentro de la clase "ESTRATEGIA". Cada uno de estos métodos cumplen con una función ya indicada.

1. AgregarDato (ArbolGeneral arbol, DatoDistancia dato): Este método agrega un nuevo dato a un árbol BK, donde tanto el dato como el árbol son enviados como parámetro.
2. Buscar(ArbolGeneral arbol, string elementoABuscar, int umbral, List collected): Retorna en la lista llamada collected el resultado de realizar una búsqueda de la cadena almacenada en elementoABuscar sobre el árbol BK almacenado en el parámetro arbol y con un nivel de tolerancia indicado por el parámetro umbral.
3. Consulta1 (ArbolGeneral arbol): Retorna un texto con todas las hojas del árbol BK del sistema.
4. Consulta2 (ArbolGeneral arbol): Retorna un texto que contiene todos los caminos hasta cada hoja.
5. Consulta3 (ArbolGeneral arbol): Retorna un texto que

contiene los datos almacenados en los nodos del árbol diferenciados por el nivel en que se encuentran.

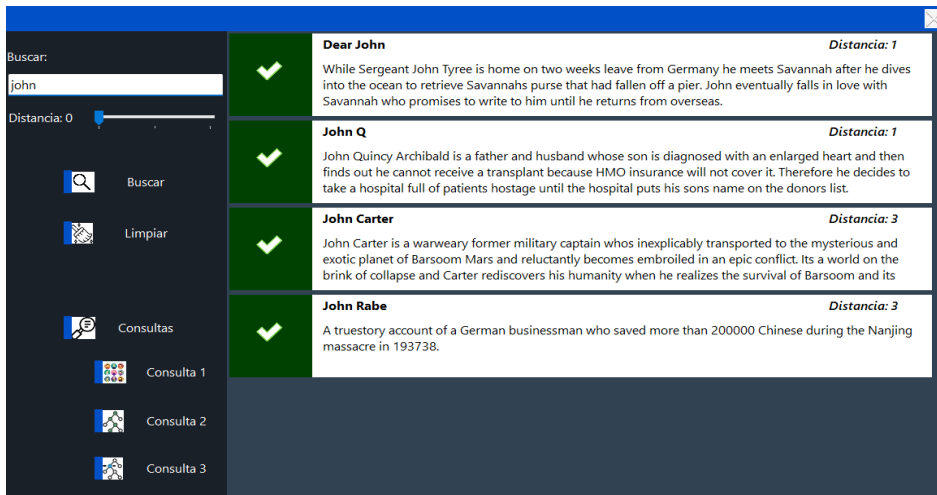


En este caso podemos observar cómo nos arroja una búsqueda con un nivel de distancia 1 dado a que escribimos JURASSIC con una sola "s", por ende, nos arroja los resultados. Ahora si decidimos disminuir la distancia a 0 debemos de agregar la S faltante porque si no lo hacemos sucede lo siguiente:



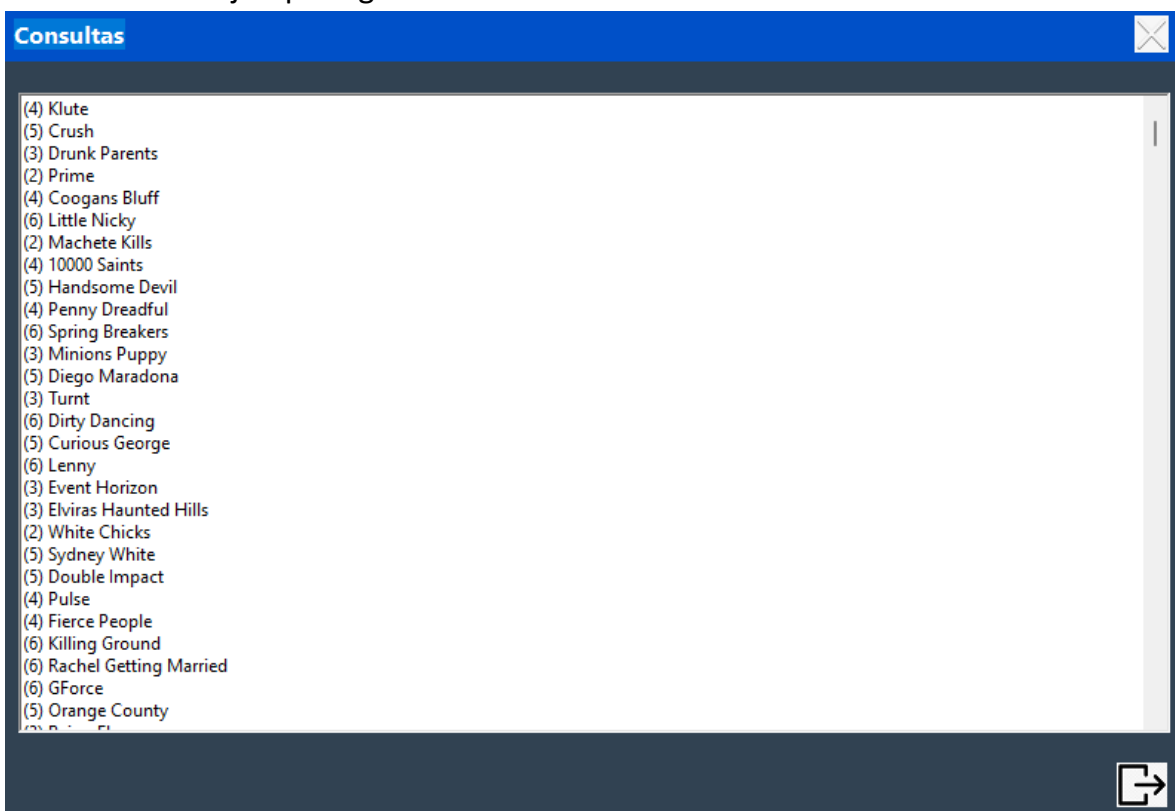
La búsqueda no reconoce ningún resultado textual y por ende no nos devuelve nada.

Ahora realizamos la búsqueda para la siguiente película con distancia 0 y observamos los resultados que nos arroja.

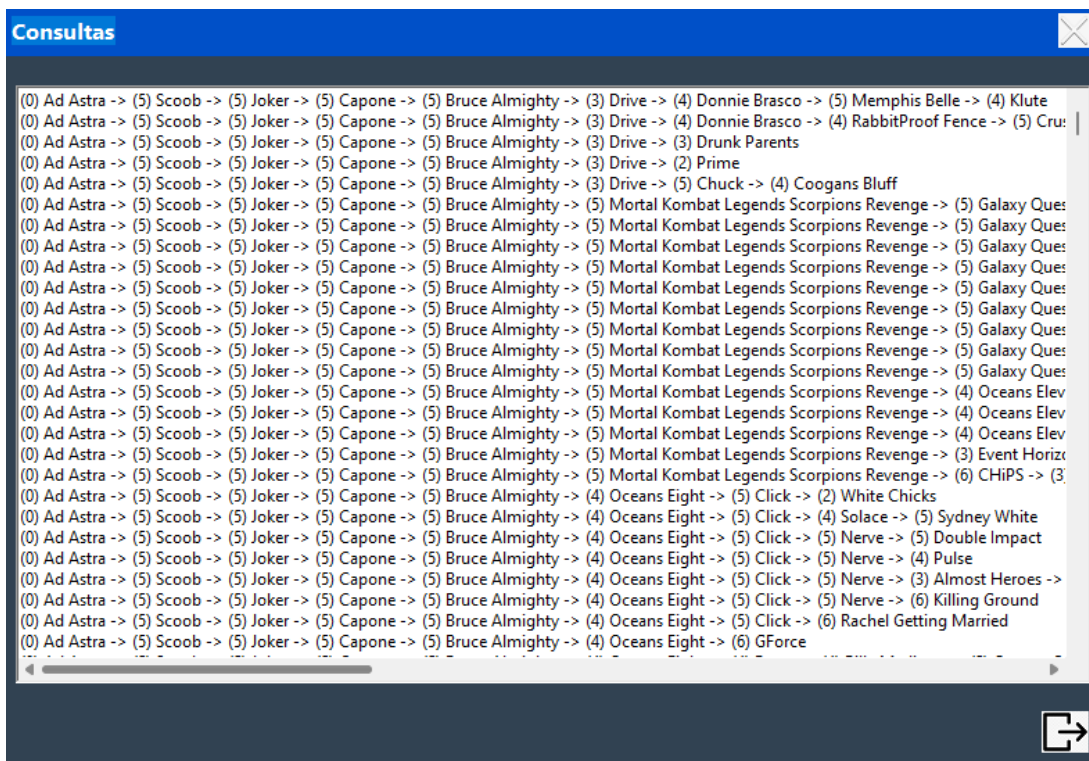


Podemos observar que la búsqueda fue exitosa y que nos arroja las películas.

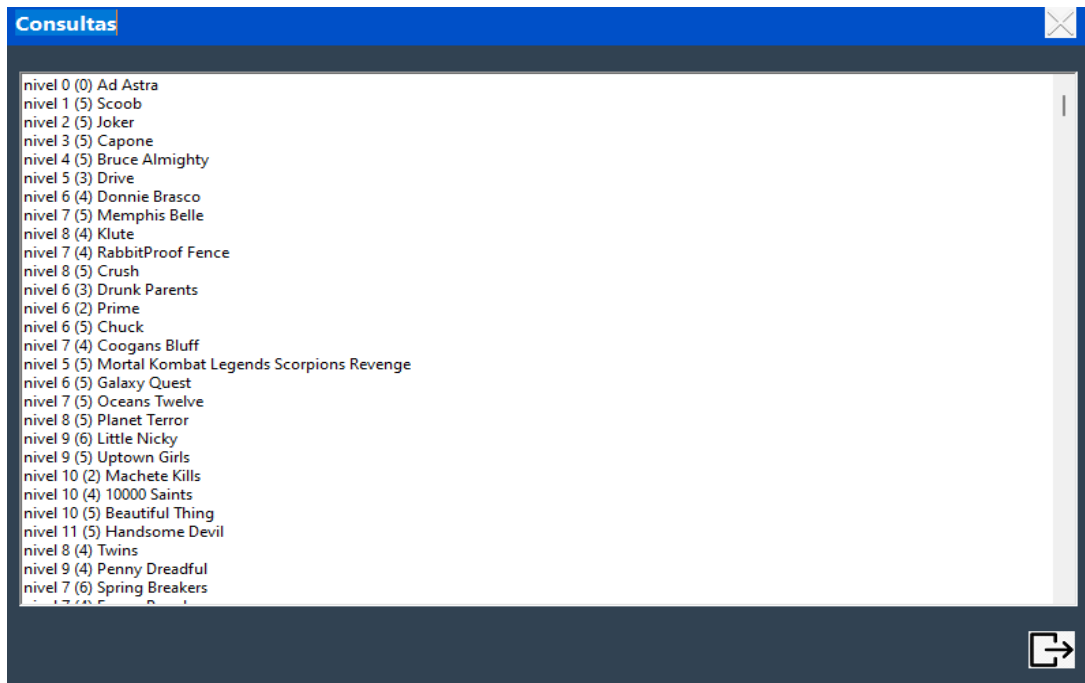
Cuando realizamos la consulta 1 podemos obtener un texto con todas las hojas pertenecientes al árbol del sistema y su distancia con el padre. Tal como se puede observar en el ejemplo siguiente:



En la consulta número 2 podemos observar cómo nos retorna una lista con todos caminos hasta cada hoja.



La consulta 3 Retorna un texto que contiene los datos almacenados en los nodos del árbol diferenciados por el nivel en que se encuentran y nuevamente la distancia en el padre:



El trabajo final resultó ser un gran desafío para mi pero me impulso a realizar múltiples búsquedas para expandir mis conocimientos sobre el lenguaje y el funcionamiento de los árboles.

Gracias a esa búsqueda pude resolver este ejercicio y aplicar esos conocimientos.

Mi principal problema fue entender el funcionamiento interno del programa sin haberlo creado yo, dado a que no había tenido experiencia hasta ahora de utilizar código ya creado por otro y tener que entender que es cada clase, como y para que funciona sin siquiera un UML.

Esta fue la primera puesta en contacto con una app un poco más semejante a lo que se encuentra en el día a día dado a que solo utilizábamos aplicaciones de consola sin interfaz gráfica y en ningún momento trabajamos con una aplicación ya diseñada, con este modelo de ejercicio similar al pedido de un trabajo real.