Trabajo Practico: Listas y Arboles

Participantes	Guzman, Hernan Cichello, Alan
Email	hernan23@gmail.com alancichello@gmail.com

NodoLista

```
def tieneSiguiente(self):
Función para consultar si el nodo tiene siguiente

def append(self, nuevoNodo):}
Función recursiva que añade un nodo a la lista

def len(self):
Función recursiva que suma la cantidad de nodos de la lista

def buscarDato(self, datoABuscar, actPos = 0):
Función que busca un dato pasado por parámetro

def eliminarDato(self, datoABuscar, actPos = 0):
Función que elimina un dato pasado por parametro
```

ListaEnlazada

Descripción de las funciones desarrolladas:

```
def estaVacia(self):
Consulta si la lista esta vacía

def appendOrd(self, dato):
Función que agrega los datos de forma ordenada de mayor a menor

def validarPosicion (self, pos):
Función que consulta si la posición es mayor o igual a cero, que la misma se menor
que el tamaño de la lista y que no esté vacía.

def getDato(self, pos):
Lee un dato según la posición solicitada.

def delete(self, pos):
Función que elimina un nodo en la lista en la posición enviada, si la posición a
eliminar es la priemra, el nodo raíz pasa a ser el siguiente, sino en la posición
cero se llama a la función recursiva de nodo Lista.

def buscarDato(self, datoABuscar):
Busca si un dato se encuentra en la lista

def eliminarDato(self, datoAEliminar):
Función que elimina un dato en la lista, pasando el dato que se desea eliminar por
parámetro.
```

NodoArbol

```
def grado(self):
```
Función que devuelve el grado del nodo, 0 si es una hoja, 1 si tiene izquierda o derecha, y 2 si tiene izquierda y derecha

```
def altura(self):
```
Función que devuelve la altura de un nodo

```
def buscaMaximo(self):
```
Función que busca si el nodo tiene derecho, en ese caso cambia el dato a Máximo.

```
def predecesor(self):
```
Busca el máximo valor de la parte izquierda del nodo, lo que dará el predecesor

```
def nivelAListaa(self, nivel, listaNivel, nivelNodo = 0):
```
Función que devuelve la lista de nodos que se encuentran en el nivel enviado por parámetro.

```
def eliminarPagina(self, dirWeb):
```
Función que elimina una página web en el árbol: 1- Busca si se encuentra la página en el nodo RAIZ, 2- Si se encuentra elimino la página llamando a la función eliminar dato, pasándole la dirección web, 3- Se consulta si el nodo tiene izquierda para llamar recursivamente a la función y continuar con la búsqueda, 4- Igual que el punto 3 pero consultando si tiene derecha.

```
def cantidadPalabras(self, cant, listaPalabras):
```
función que devuelve las palabras en el nodo que tengan igual o mayor cantidad de caracteres que los solicitados; 1- consulta si la cantidad de caracteres es mayor o igual a los solicitado; 2- si la palabra cumple con la condición se agrega a la lista que se devolverá; 3- se consulta si el nodo tiene izquierda para llamar recursivamente a la función y continuar con la búsqueda; 4- igual que el anterior pero consultando si tiene derecha

```
def cantidadPalabrasConPaginas(self, cant, listaPalabras):
```
función que devuelve las palabras en el nodo que tengan igual o mayor número de páginas que las solicitadas; 1- consulta si la cantidad de páginas es mayor o igual al solicitado; 2- si la cantidad de páginas cumple con la condición se agrega a la lista que se devolverá; 3- se consulta si el nodo tiene izquierda para llamar recursivamente a la función y continuar con la búsqueda; 4- igual que el anterior pero consultando si tiene derecha

```
def internasMayusculas(self, listaPalabras = ListaEnlazada()):
```
función que devuelve los subárboles en los que la palabra comienza con mayúscula; 1- consulta si el nodo no es una hoja; 2- si el nodo no es una hoja, tomo la primera letra y consulto si esta en mayúscula; 3- si esta en mayúscula lo agrego a la lista que se devolverá; 4- se consulta si el nodo tiene izquierda para llamar recursivamente a la función y continuar con la búsqueda; 5- igual que el anterior pero consultando si tiene derecha

ArbolBuscador

```
def altura(self):
Devuelve la altura de un árbol

def maximo(self):
Función que busca el máximo de una árbol

def existeNodo(self, palabraBuscar):
Si el árbol no esta vació, busca la palabra pasada por parámetro en todos los nodos.

def agregarDirWeb(self, lista, direccionWeb):
Función que agrega la dirección web a la lista del nodo

def insertarPalabra(self, palabraNueva, direccionWeb):
Función que busca insertar una palabra al árbol, previamente verificando: 1- Que la
palabra no exista en el árbol; 2-Si no existe la palabra, la agrego junto con la
lista de la Pagina web

def insertarPagina(self, listaDePalabras, paginaWeb, pos = 0):
Función que agrega una pagina web recibiendo por parámetros una lista de palabras

def buscarSimilitudes(self, listaPaginas, paginasEncontradas, listaSimilitud =
ListaEnlazada(), pos=0):
función que compara entre dos listas las similitudes. Esto se realiza para que en la
lista resultante no solo se muestren las páginas que se encuentran en las dos listas

def buscarPalabras(self, listaPalabras, paginasEncontradas = ListaEnlazada(), vacio =
False, pos = 0):
función que recibiendo una lista de palabras, devuelve las paginas donde se
encuentran estas palabras

def palabrasDePagina(self, dirWeb):
Función que devuelve las palabras que contiene esa página Web.

def eliminarPalabra(self, palabra):
Función que elimina una palabra, por lo que elimina el nodo del árbol que contenga la
palabra indicada

def eliminarPagina(self, pagina):
Función que elimina la página indicada en todo el árbol

def estaBalanceado(self):
Función que devuelve si el árbol esta balanceado: 1- consulta si tiene izquierdo y
derecho para consultar la altura de ambos; 2- calcula que la diferencia entre ambas
alturas no sea superior a 1

def cantidadTotalPalabras(self, cantidad):
función que devuelve una lista de palabras, estas deben tener igual o mayor cantidad
de caracteres que la cantidad recibida

def buscarPaginasNoRepetidas(self, listaPaginas, paginasEncontradas):
Función que compara que entre dos listas no se repitan valores

def buscarPaginas(self, listaPalabras, paginasEncontradas = None):
Función que recibe una lista de palabras y busca las páginas de cada una.

def paginasEnNivel(self, nivel):
Función que devuelve una lista con las páginas que se encuentran en el nivel recibido
def cantidadPalabrasMasUsadas(self, cantPaginas):
```


```
Función que devuelve una lista de palabras, donde la cantidad de páginas que contenga
sea mayor o igual que la cantidad recibida'''


def internasMayusculasAlfabetico(self):
Función que devuelve las palabras de los subnodos que comiencen con mayúscula'''
```