



FACULTAD DE INGENIERÍA
DEPARTAMENTO DE TECNOLOGIAS
DE LA INFORMACIÓN

Tare integradora 2.

Algoritmos y estructuras de datos

Andres Cuellar Duvan
Garcia Gustavo
Villada

2020

Requerimientos

Rq1: Para la generación de nombres se debe usar la base de datos “ [dataset de nombres de data.world](#)”

Rq2: Para la generación de apellidos se debe usar la base de datos “ [dataset de apellidos de data.world](#)”

Rq3: Para la generación de fechas de nacimiento se debe usar la base de datos “ [dataset de apellidos de data.world](#)”, tomando en cuenta la distribución

Rq4: Para la generación de estatura se debe usar la base de datos “ [estos datos de población por países](#)”, tomando en cuenta la distribución

Rq5: Se debe poder eliminar a una persona del programa

Rq6: Se debe poder registrar a una persona del programa

Rq7: Se debe poder modificar los datos de una persona del programa

Rq8: Se debe poder buscar en el programa de acuerdo a los siguientes criterios:

1. Nombre
2. Apellido
3. Nombre Completo (Nombre + “ ” + Apellido)
4. Código

Rq9: El programa debe tener predictor de lo que se esté digitando en la búsqueda 1 y 3 del anterior requerimiento

Rnf1: El programa debe tener un árbol binario de búsquedas autobalanceado para el nombre

Rnf2: El programa debe tener un árbol binario de búsquedas autobalanceado para el apellido

Rnf3: El programa debe tener un árbol binario de búsquedas autobalanceado para el nombre completo

Rnf4: El programa debe tener un árbol binario de búsquedas autobalanceado para el código.

Diseño de las pruebas.

Objetivo de la prueba: obtener la key del nodo más pequeño del arbol				
Clase	Método	Escenario	Entradas	Resultado
AVLTree	getMin()	setup1()	El valor de la key que puede variar el criterio por el cual este construido el arbol	La key del Nodo avl que fue agregado dentro del arbol

Objetivo de la prueba: obtener la key del nodo más grande del arbol				
Clase	Método	Escenario	Entradas	Resultado
AVLTree	getMax()	Setup2()	El valor de la key que puede variar el criterio por el cual este construido el arbol	La key del Nodo avl que fue agregado dentro del arbol

Objetivo de la prueba: obtener el nodo predecesor del indicado				
Clase	Método	Escenario	Entradas	Resultado
AVLTree	getPredecessor()	Setup3()	El valor de la key que puede variar el criterio por el cual este construido el arbol	La key del Nodo avl que es predecesor del indicado

Objetivo de la prueba: obtener el nodo sucesor del indicado				
Clase	Método	Escenario	Entradas	Resultado
AVLTree	getSucessor()	Setup4()	El valor de la key que puede variar el criterio por el cual este construido el arbol	La key del Nodo avl que es sucesor del indicado

Objetivo de la prueba: obtener el peso del arbol				
---	--	--	--	--

Clase	Método	Escenario	Entradas	Resultado
AVLTree	getWeight()	Setup5()		El peso del árbol , ósea el número de nodos agregados

Objetivo de la prueba: obtener el nodo por medio de una key

Clase	Método	Escenario	Entradas	Resultado
AVLTree	Search()	Setup5()	El valor de la key que puede variar el criterio por el cual este construido el arbol	El Nodo avl que es busacado por su propia key

Objetivo de la prueba: se inserta o añade un nodo al arbol

Clase	Método	Escenario	Entradas	Resultado
AVLTree	Insert()	Setup5()	El valor de la key que puede variar el criterio por el cual este construido el árbol, el valor del nodo	Se añade un nodo dentro del árbol avl

Objetivo de la prueba: se borra o elimina un nodo del arbol

Clase	Método	Escenario	Entradas	Resultado
AVLTree	Delete ()	Setup5()	El valor de la key que puede variar el criterio por el cual este construido el árbol	Se rotorna el nodo que ha sido eliminado del arbol

Diagrama de clases:

