

Diccionario
+FB_NEGATIVO: const int +FB_POSITIVO: const int +FB_NEUTRO: const int -raiz: NodoAVL<T1, T2>* -cantidadNodos: int
+Diccionario() +buscar(nodo: NodoAVL<T1, T2>*, clave: T1): NodoAVL<T1, T2>* +buscar(clave: T1): bool +quitarNodo(clave: T1) +vacio(): bool +recorridoInOrden(nodo: NodoAVL<T1, T2>*) +recorridoInOrden() +recorridoAncho(nodo: NodoAVL<T1, T2>*, cont: int) +recorridoAncho() +cantidadNiveles(): int +obtenerCantidad(): int +imprimir(clave: int) +~Diccionario() -obtenerPredecesor(nodo: NodoAVL<T1, T2>*):NodoAVL<T1, T2>* -quitarRaiz() -quitarNodo(nodo: NodoAVL<T1, T2>*) -quitarHoja(nodo: NodoAVL<T1, T2>*) -quitarNodoConHijoDerecho(nodo: NodoAVL<T1, T2>*) -quitarNodoConHijoIzquierdo(nodo: NodoAVL<T1, T2>*) -quitarNodoConDosHijos(nodo: NodoAVL<T1, T2>*) -limpiar() -chequearBalance(nodo: NodoAVL<T1, T2>*) -balancear(padre: NodoAVL<T1, T2>*, hijo: NodoAVL<T1, T2>*) -rotacionSD(padre: NodoAVL<T1, T2>*, hijo: NodoAVL<T1, T2>*) -rotacionSI(padre: NodoAVL<T1, T2>*, hijo: NodoAVL<T1, T2>*) -rotacionDID(padre: NodoAVL<T1, T2>*, hijo: NodoAVL<T1, T2>*) -rotacionDDI(padre: NodoAVL<T1, T2>*, hijo: NodoAVL<T1, T2>*) -actualizarBalanceo(nodo: NodoAVL<T1, T2>*)

1

<<tiene>>

0...*

NodoAVL
-clave: T1 -valor: T2 -alturaDerecha: int -alturaIzquierda: int -factorBalanceo: int -izquierdo:NodoAVL<T1, T2>* -derecho: NodoAVL<T1, T2>* -padre:NodoAVL<T1, T2>*
+NodoAVL(clave: T1, valor: T2) +cambiarAlturalzq(nuevaAltura: int) +cambiarAlturaDer(nuevaAltura: int) +cambiarClave(clave: T1) +cambiarValor(valor: T2) +cambiarDerecho(derecho: NodoAVL<T1, T2>*) +cambiarIzquierdo(izquierdo: NodoAVL<T1, T2>*) +cambiarPadre(padre: NodoAVL<T1, T2>*) +cambiarHijoNuevo(padre: NodoAVL<T1, T2>, hijoNuevo:nodo: NodoAVL<T1, T2>*) +quitarPadre(nodo: NodoAVL<T1, T2>*) +obtenerAlturaDer(): int +obtenerAlturalzq()int: +obtenerClave(): T1 +obtenerValor(): T2 +obtenerDerecho():NodoAVL<T1, T2>* +obtenerIzquierdo():NodoAVL<T1, T2>* +obtenerPadre():NodoAVL<T1, T2>* +hijoIzquierdoUnico(): bool +hijoDerechoUnico(): bool +dosHijos(): bool +esHoja():bool +calcularAltura(nodo: NodoAVL<T1, T2>*): int +calcularFactorBalanceo(nodo: NodoAVL<T1, T2>*) +obtenerFB(): int +~NodoAVL()