**PARTE 2: Ejercicios de seudocódigo**

**Duración: 60 minutos**

**CONSIDERACIONES IMPORTANTES**

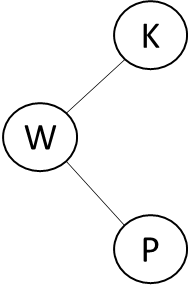
* Se deben respetar las normas publicadas sobre la escritura de seudocódigo (lenguaje natural, pre y post condiciones, seudocódigo detallado sobre esta implementación del TDA.
* Se recomienda escribir métodos auxiliares para simplificar el método principal.
* Se deben desarrollar completamente todas las operaciones invocadas.
* Se deben calcular los órdenes del tiempo de ejecución de los algoritmos desarrollados.

**EJERCICIO 1**

Dado un árbol binario, escribir un algoritmo que devuelva la cantidad de nodos internos (los nodos que no son hoja) no completos (es decir, que les falte un hijo), de acuerdo a la siguiente firma:

TArbolBinario.internosNoCompletos() //devuelve un entero

TNodoArbolBinario.internosNoCompletos() //devuelve un entero

Ejemplo

Si el árbol es:

El algoritmo devuelve: **2** (nodo “K” y nodo “W”)

**EJERCICIO 2**

**ESCENARIO**

Un hospital de nuestro país mantiene una **plantilla** de **médicos** cuyos elementos contienen información relacionada con cada médico que trabaja en la institución: **Cédula**, **nombre**, **especialidad**, **año de ingreso**. La plantilla es almacenada utilizando un Árbol Binario de Búsqueda, que se utiliza habitualmente para emitir listados de todos los médicos **ordenados por número de Cédula**.

Una consulta habitual es solicitar un directorio de todos los médicos de una cierta **especialidad** (ej: cardiología, pediatría, cirugía, etc.), ordenado **alfabéticamente**.

Por ejemplo, si, al iniciar, la **plantilla de médicos** del **Hospital** contiene (por su listado en inorden)

1, (1, “Pedro”, “Cirugía”, 2005);

2, (2, “María”, “Cardiología” , 2004);

3, (3, “José”, “Pediatría” , 2008);

4, (4, “Florencia“, “Pediatría” , 2004);

5, (5, “Juan”, “Cardiología” , 2007);

6, (6, “Ana”, “Cirugía” , 2002);

7, (7, “Martín”, “Pediatría” , 2009);

**luego de ejecutado el algoritmo** que construye el directorio para **“Cardiología”** (en un nuevo ABB), el listado del mismo en inorden es:

Juan, (5, “Juan”, “Cardiología”, 2007);

María, (2, “María”, “Cardiología”, 2004);

**SE PIDE:**

1. Desarrollar un algoritmo del TDA Árbol Binario de Búsqueda (utilizado para una “***plantilla de médicos***”, que extiende el TDA ABB estándar), cuyos elementos son del tipo **TElementoAB** con referencia a objeto del tipo “***Médico***” -con atributos ***cédula, nombre, especialidad y año de ingreso*)**,– y etiqueta correspondiente al campo ***cédula***, que, recibiendo como parámetro un **nombre de especialidad** (ej, “Cardiología”), construya un nuevo ABB que contendrá los médicos de esta especialidad, utilizando como clave de los nodos (clave de inserción / búsqueda) el **nombre** del médico.

**Entidades involucradas**

* **Medico** atributos: Cedula, Nombre, Especialidad, año de ingreso
* **PlantillaMedicos** ABB que agrega el método requerido. Los nodos son del tipo **TDA TElementoAB** (con campo etiqueta = **medico.cedula**, campo Dato referencia un Medico).
* **DirectorioEpecialidad, ABB** de especialidad con los médicos de la especialidad. Los nodos son del tipo **TDA TElementoAB** (con campo etiqueta = **medico.nombre**, campo Dato referencia un Médico)

**Firma de los métodos a desarrollar:**

**De TDA PlantillaMedicos** (que es o desciende de ABB)

**obtenerDirectorio (de tipo string especialidad) : DEVUELVE un TABB**

**CONSIDERACIONES IMPORTANTES**

* Se deben respetar las normas publicadas sobre la escritura de seudocódigo (lenguaje natural, pre y post condiciones, seudocódigo detallado sobre esta implementación del TDA.
* la lista original de médicos no se ve alterada.
* No se deben duplicar instancias de ningún TDA
* Se deben desarrollar completamente todas las operaciones invocadas.