

## TRABAJO DE APLICACIÓN 3

### Ejercicio #3

#### TODO EL EQUIPO:

- a) Tener abierto en una computadora el tutorial de JUNIT (<http://www.vogella.com/articles/JUnit/article.html>)
- a) Generar una **clase de test** para la clase **TArbolBB**,
- b) Escribir test cases para el método de listado en **inorden**.
- c) Ejecutar los test cases y corregir los errores que eventualmente se identifiquen.
- d) **Sincronizar las máquinas para los sub-equipos.**

### Ejercicio #4 - test cases para obtenerAltura y obtenerTamaño

#### PASO 1: TODO EL EQUIPO:

- a) Modificar las interfaces de TElementoAB y TArbolBB para incluir las operaciones “obtenerTamaño” y “obtenerAltura”
- b) Hacer que el entorno Netbeans genere los métodos correspondientes de TElementoAB y TArbolBB (sin implementación, con emisión de excepciones)
- c) Sincronizar las computadoras con el repositorio

#### PASO 2: EN SUBEQUIPOS:

##### Sub-Equipo A:

- a) Implementar los “**test cases**” para el método **obtenerTamaño** del árbol.

##### Sub-Equipo B:

- a) Implementar los “**test cases**” para el método **obtenerAltura** del árbol.

#### PASO 3: TODO EL EQUIPO: Integrar en repositorio GIT.

## TRABAJO DE APLICACIÓN 3

### Ejercicio #5

Agregar funcionalidades al TDA Arbol Binario de Búsqueda.

#### PASO 1: EN SUBEQUIPOS:

##### Sub-Equipo A:

- a) Implementar en JAVA el algoritmo para **obtener la altura** del árbol.
- b) Ejecutar los test cases correspondientes, previamente desarrollados
- c) Revisar el código implementado y resolver eventuales errores descubiertos.

##### Sub-Equipo B:

- b) Implementar en JAVA el algoritmo para **obtener el tamaño** del árbol.
- c) Ejecutar los test cases correspondientes, previamente desarrollados
- d) Revisar el código implementado y resolver eventuales errores descubiertos.

#### PASO 2: TODO EL EQUIPO

- a) Integrar los cambios en el repositorio
- d) Probar los algoritmos insertando en el árbol varias claves y verificar en papel que ambos algoritmos funcionen correctamente