

### Hoja de trabajo No. 1

**Realizar:** Programa de Radio.

**Realizarse:** en TRIOS.

**Objetivos:**

- Utilizar GIT para guardar las versiones del programa.
- Practicar la definición de interfaces y reutilización de código.
- Emplear JUnit para casos de prueba.
- Repasar el diagrama UML de clases.
- Repasar el lenguaje JAVA.**

**Programa a realizar:**

Realizar el programa del ejercicio 1.14 de libro de The Pragmatic Programmer<sup>1</sup>:

“Design a data structure to simulate the workings of a kitchen blender. It will eventually be a web-based, IoT-enabled blender, but for now we just need the class to control it. It has ten speed settings (0 means off). You can’t operate it empty, and you can change the speed only one unit at a time (that is, from 0 to 1, and from 1 to 2, not from 0 to 2).”

Se debe tener un programa principal que emplee la licuadora. Simplemente muestra un menú, o una interfaz GUI, con las opciones:

- Prende la licuadora
- Llena la licuadora con lo que se desea licuar
- Incrementa velocidad
- Consulta en que velocidad esta la licuadora
- Consulta si la licuadora esta llena.
- Vaciar la licuadora

**Tareas:**

- Diagrama UML de clases. Recuerde que la notación debe ser la correcta para las clases, interfaces y sus relaciones.
- Construir la interface y la clase que implemente la licuadora.
- Su programa principal debe permitir cambiar la clase que implementa la licuadora, sin que se vea afectada en su funcionamiento. Ver el requisito de calificación que se describe más adelante.
- Debe dejar evidencia de todo el desarrollo en el repositorio de GIT agregar a su catedrático y auxiliar para que tengan acceso al mismo. Los correos de los auxiliares le serán proporcionados más adelante.
- Incluya tres pruebas unitarias con JUnit. NO deben ser simplemente los tests generados automáticamente por el IDE, recuerde que tiene que modificarlos para que realmente prueben los métodos de su clase.

Debe subir al Canvas todos los productos elaborados en los incisos a, b, c, e y los enlaces a su repositorio de GIT (o del sistema seleccionado para control de versiones). **Asegúrese de permitir ingreso a su repositorio a su catedrático y auxiliares del curso o que sea un repositorio público.**

---

<sup>1</sup> Thomas, David; Hunt, Andrew. Pragmatic Programmer, The: Your journey to mastery, 20th Anniversary Edition (English Edition) (p. 209). Pearson Education. Edición de Kindle.

**Requisito para calificar esta hoja.**

Su programa principal debe funcionar, **sustituyendo la clase que implementa la licuadora** por una clase producida por otros dos o tres grupos como mínimo. El cambio debe ser lo más transparente posible en su programa principal, requiriendo que se **modifique una línea o menos** del mismo al momento de compilarse.

**NOTA:** para ejecutar este programa **NO SE USARÁ** ningún IDE. Debe poder iniciar su ejecución desde la línea de comandos del sistema operativo. No puede suponer que todos los desarrolladores emplearán un IDE específico, ni que se cuenta con clases que sean generadas automáticamente por el IDE.



**Calificación:** su programa debe funcionar y tener control de versiones para ser calificado.

Aspecto	Puntos
Estilo de codificación: comentarios, indentación, nombres de variables significativas. Consultar "Code Conventions for the Java Programming Language" <sup>2</sup> o Google Java Style <sup>3</sup>	10
Documentación generada con Javadoc	10
Uso del repositorio: existen más de tres versiones guardadas, la última versión es igual a la colocada en el Canvas. En el repositorio se encuentra evidencia que existen contribuciones en el código u otros documentos, realizadas por cada integrante del grupo.	10
Diagrama de clases: muestran la abstracción y encapsulación de las operaciones. Muestra correctamente las relaciones entre las clases. Se encuentra también guardado en el repositorio de GIT o del sistema empleado. Suba solamente la imagen en formato jpeg, jpg o similar.	5
Funcionamiento del programa. Usted adjunta también screenshots o videos para mostrar que el programa funciona.	15
Funcionamiento del programa usando la clase implementada por otros grupos. El auxiliar seleccionará estos grupos al azar, pero usted puede indicar con quienes ha probado que funciona su programa. Adjunte screenshots de funcionamiento de su programa funcionando con clases implementadas por otros grupos.	15
Su clase de licuadora funciona en el programa de otros grupos. El auxiliar utilizará la clase licuadora que su grupo elaboró y lo probará con el programa	15

<sup>2</sup> Ya están antiguas, pero aún tienen puntos útiles: <http://www.oracle.com/technetwork/java/codeconvtoc-136057.html>

<sup>3</sup> <http://google-styleguide.googlecode.com/svn/trunk/javaguide.html>



principal de otro grupo seleccionado al azar. Adjunte screenshots de funcionamiento del programa de otros grupos que emplean las clases desarrolladas por usted.	
Utiliza JUnit para probar por lo menos tres de las operaciones de la licuadora. Adjunte con su tarea las pruebas unitarias definidas. Adjunte screenshots de la ejecución de las pruebas. Debe existir por lo menos una muestra de que las pruebas fallan y una de que las pruebas si funcionan.	20
<b>TOTAL:</b>	<b>100</b>