

1. INTRODUCCIÓN

Los objetivos principales de la práctica son aprender a diseñar nuevos Tipos Abstractos de Datos (TAD) utilizando las estructuras de datos vistas en clase, a modularizar la organización de los datos y las operaciones sobre ellas en distintas clases, a cargar esas estructuras con información, utilizar las operaciones adecuadas de cada estructura de datos para resolver el problema de forma eficiente, y codificar algoritmos para el procesamiento de la información.

Otro de los objetivos es trabajar la competencia transversal de trabajo en grupo. Concretamente, aprender a especificar y planificar las tareas a realizar, repartir el trabajo y realizar un seguimiento, e integrar y probar las partes que se han realizado de forma independiente. Para ello, también aprenderá a definir casos de prueba implementando pruebas unitarias utilizando JUnit [1, 2].

Cualquier información relacionada con la práctica la encontraréis en el Tema 7 (apartado Práctica) de la asignatura “Estructuras de datos y Algoritmos” en eGela (<http://egela.ehu.es>).

2. FASES Y ENTREGABLES DE LA PRÁCTICA

La práctica se realizará **en grupos de 4**, y se llevará a cabo principalmente fuera del horario docente. Poco más o menos, requiere una dedicación de unas 50 horas de trabajo/alumno (2h en clase), distribuidas a lo largo de 5 semanas (segunda mitad del curso), con 10 horas de dedicación semanal.

La práctica se realizará en dos fases que se describen a continuación:

Fase	Entregables		Fecha de entrega	Puntuación (max.)
I	1	Diseño de las estructuras de datos	11 de noviembre de 2014	0,5 puntos
II	2	Implementación de los subprogramas	19 de diciembre de 2014	1,5 puntos
	3	Documentación final	19 de diciembre de 2014	0,5 puntos

La Fase I se iniciará, desarrollará y finalizará el **11 de noviembre de 2014** en horario lectivo. El documento final pertenecerá al grupo de informes recogidos durante el curso.

En la Fase II, el grupo recibirá el diseño y parte de la implementación de la práctica. La tarea del grupo será completar la implementación de los distintos métodos que están distribuidos en distintas clases. Además de implementar las pruebas unitarias en JUnit para comprobar el funcionamiento de los métodos implementados.

La documentación asociada a cada fase de la práctica deberá ser enviada antes de la fecha indicada, a la correspondiente tarea de la asignatura en eGela [3] o Web-CAT[4], para que se considere que se ha realizado la práctica.

Tarea: Cada grupo debe anotar el nombre del grupo y los nombres de sus miembros en la Wiki Grupos, dentro del apartado ‘Práctica’ en eGela. Fecha límite: **7 de noviembre de 2014**

3. DOCUMENTACIÓN A ENTREGAR

El grupo deberá enviar para la fecha límite, indicada para cada una de las fases de la práctica en el apartado 2, la documentación que se especifica en los siguientes apartados. Se creará una tarea para cada fase en eGela o Web-CAT para que el grupo pueda enviar la documentación solicitada. **Los**

ficheros que se envíen se deberán nombrar con el nombre del grupo y es suficiente con que los envíe uno de ellos.

3.1. FASE I: DISEÑO DE LAS ESTRUCTURAS DE DATOS

Se pide:

- Representar** gráficamente (utilizando el diagrama de clases o diagrama de objetos de UML [5]) la estructura de las clases (principalmente, nombre + atributos + constructores + operaciones) necesarias para representar las entidades y la información más importante de la práctica. En cada caso, el grupo deberá seleccionar alguna de las estructuras de datos vistas en clase para organizar los datos y **argumentar las decisiones** tomadas.
- Desarrollar** en Java[6] el apartado a), sin llegar a implementar las operaciones. En este caso, especificar mediante JavaDoc[7] la pre- y post-condiciones. Para ello, se tendrán en cuenta las características de la práctica así como las operaciones principales que se van a necesitar, detalladas en el capítulo 4. El grupo deberá documentar cada clase y sus atributos para facilitar la comprensión y la utilización de la misma.

3.2. FASE II: IMPLEMENTACIÓN DE LOS SUBPROGRAMAS

Como mínimo, **semanalmente** (cada viernes) el grupo deberá enviar un archivo *.zip* que contenga el proyecto (p.ej. de Eclipse [7]) con la implementación de la práctica y **la versión final antes de la fecha indicada** en el apartado 2. Además de los métodos solicitados en el apartado 4, se deberá comprobar el correcto funcionamiento de cada uno de los subprogramas implementando pruebas unitarias en JUnit. **Nota:** En la versión final, el proyecto no debe tener ningún error de compilación.

Además de lo anterior, se incluirá también un documento que describa cómo ha sido el desarrollo del proyecto (IMPORTANTE: No lo dejéis para el final, hacerlo en paralelo). Entre los puntos se incluirá:

- **Resumen** de lo que contiene el documento y la dedicación en el desarrollo de la práctica. El resumen de la dedicación incluirá dos tablas, las cuales desglosarán el número de horas empleadas (total y porcentaje) 1) por cada miembro y 2) por las tareas más importantes.
- **Planificación de tareas:** pequeña descripción (o nombre método), fecha inicio y fin, responsable, tiempo estimado, tiempo real y si se ha recibido alguna ayuda o se ha ayudado a otro grupo (quién y qué tipo de ayuda).
 - Las tareas pueden ser del tipo: planificar, analizar, discutir, tomar decisiones, diseñar o implementar clase, subprograma o prueba unitaria, integrar partes, probar, documentar, resolver problemas, ayudar, ...
- **Casos de prueba:** Describir brevemente los casos de prueba que habéis realizado para cada subprograma: descripción, inicialización, resultado esperado y resultado obtenido.
- **Actas de las reuniones realizadas:** fecha y hora, miembros reunidos o ausentes, orden del día, resumen de las decisiones tomadas y fecha de la próxima reunión.

IMPORTANTE: En las tareas de tipo diseño o implementación de clases, subprogramas o pruebas unitarias, todos los miembros del grupo deberían de realizar, por lo menos, una tarea de cada tipo.

Los alumnos pueden ayudarse entre ellos para solucionar aspectos técnicos y/o entender el diseño y el código facilitados en la práctica, pero **nunca** estará permitido facilitar a otro alumno la implementación de los subprogramas solicitados en el apartado 4. El alumno no debe olvidar que el objetivo es aprender practicando. Cualquier indicio de plagio, supondrá quedarse fuera de la evaluación continua y el único examen que podrá realizar este curso será la extraordinaria de Junio.

4. DESCRIPCIÓN BREVE DE LA PRÁCTICA

Al inicio de la Fase I, el grupo recibirá más información al respecto.

5. DISEÑO Y FUNCIONALIDAD

Después de haber finalizado la Fase I, el grupo recibirá el diseño y la especificación de la funcionalidad que tienen que implementar para realizar la Fase II.

6. BIBLIOGRAFÍA

1. JUnit. <http://www.junit.org>
2. Tutorial de JUnit, ver 'Material complementario' en eGela.
3. eGela. <http://egela.ehu.es>
4. Web-CAT. <http://lsi.vc.ehu.es/Web-CAT>
5. Resumen de UML, ver 'Material complementario' en eGela.
6. Java. <https://www.oracle.com/java/index.html>
7. JavaDoc, ver 'Material complementario' en eGela.
8. Eclipse. <http://www.eclipse.org>