

Jorge A. Villalobos S. *Introducción a las Estructuras de Datos. Aprendizaje Activo Basado en Casos*, Pearson Educación de Colombia Ltda., 2008. ISBN 978-958-699-104-9 (e-book)

Fecha de recepción: 27/3/2020

Fecha de aprobación: 30/3/2020

Introducción a las Estructuras de Datos. Aprendizaje Activo Basado en Casos es un libro dirigido a estudiantes que posean conocimientos básicos de programación orientada a objetos y del lenguaje Java. Forma parte del proyecto Cupi2 de la Universidad Nacional de los Andes, cuyo objetivo es la búsqueda de nuevas maneras de enfrentar el problema de enseñar a programar.

Utiliza una estrategia pedagógica basada en cinco pilares: aprendizaje activo, desarrollo incremental de habilidades, equilibrio en los ejes temáticos, aprendizaje basado en problemas y actualidad tecnológica. Hace una clasificación en siete dominios conceptuales (ejes temáticos) de los conocimientos y habilidades necesarias para resolver un problema utilizando un lenguaje de programación: Modelado y solución de problemas, Algorítmica, Tecnología y programación, Herramientas de programación, Procesos de *software*, Técnicas de programación y metodologías, Elementos estructuradores y arquitecturas.

El Modelado y solución de problemas está relacionado con la capacidad de abstraer la información importante y expresarla en términos de un

lenguaje de programación. Algorítmica consiste en utilizar las instrucciones del lenguaje de programación de la forma adecuada para resolver el problema. Tecnología y programación se refiere a los elementos tecnológicos que se necesitan y la habilidad de usarlos. Las Herramientas de programación son las utilizadas para poder desarrollar un programa. Los Procesos de *software* determinan las etapas, la calidad, grupo de trabajo, estimación de tiempos, documentación de cada etapa, *testing*, entre otros. Las técnicas de programación y metodologías incluyen las estrategias de cómo hacer las cosas, la experiencia de otros. Los elementos estructuradores y arquitecturas definen los componentes que forman parte del producto resultante y la forma en que se comunican.

El libro desarrolla los contenidos agrupándolos de acuerdo a su dificultad en seis niveles y en cada uno se tienen en cuenta los siete ejes temáticos necesarios para su desarrollo. Cada nivel organiza los contenidos de acuerdo a la siguiente clasificación: Objetivos Pedagógicos (capacidades que el lector obtendrá al completar el Nivel), Motivación (qué es lo que nos anima en este nivel, qué

dificultades se van a sortear), Casos de Estudio (problemas sobre los cuales se aplicarán los conceptos a aprender). Para cada Caso de Estudio se consideran: Objetivos de Aprendizaje, Comprensión de los Requerimientos, Comprensión del Mundo del Problema, Precondiciones, Postcondiciones e Invariantes en la etapa de desarrollo y Pruebas Unitarias Automáticas en la etapa de *testing*.

En cuanto a la Comprensión de los Requerimientos, el libro plantea una metodología para obtener la lista de requerimientos funcionales y no funcionales, a partir de la lectura de una narrativa y la confección de una planilla en la que se especifica cada Requerimiento. Los requerimientos funcionales son los servicios u operaciones que el programa debe proveer al usuario, los no funcionales representan características generales y restricciones de la aplicación o sistema que se esté desarrollando.

El Mundo del Problema hace referencia al contexto en el cual se desarrolla el problema, básicamente consiste en identificar las Clases y sus relaciones. Se utilizan diagramas en lenguaje UML.

Las Precondiciones, Postcondiciones e Invariantes establecen una forma de “programación defensiva” y forman parte del contrato de la clase, constructor y métodos. Se dejan determinados los valores de entrada aceptados por un método

(precondiciones); el resultado esperado, cambios en parámetros de entrada/salida y el nuevo estado al que pasará el objeto (postcondiciones); y lo que debe permanecer inalterable (invariante). Se utilizan en la etapa de desarrollo y son una herramienta para el programador que le permite controlar hasta un cierto punto, si el algoritmo de un determinado método cumple con lo establecido, si se ha realizado una invocación correcta respetando el contrato, etc. Al momento de realizar las Pruebas Unitarias y luego en el despliegue, son desactivados.

Las Pruebas Unitarias Automáticas son herramientas para mejorar la calidad de un programa y facilitar el proceso de desarrollo. Se implementan mediante nuevas Clases cuyos métodos responden a los casos de prueba diseñados para probar cada método de cada Clase del programa en desarrollo. Se confeccionan escenarios (objeto de la clase que se quiere probar con un estado conocido) y para cada escenario un conjunto de métodos de prueba. Cada método de prueba compara el valor obtenido al invocar el método probado con el valor esperado (debe ser conocido previamente). Para implementar el código de las Pruebas Unitarias Automáticas se explica el uso del *framework* de construcción de pruebas *Junit*.

Cada uno de los seis niveles presenta sus temas de desarrollo a través de los cuales se introducen nuevos conceptos o se incrementa la complejidad

del nivel anterior. Se hace sobre la resolución de los diferentes Casos de Estudio, en donde se analiza una narrativa que lo define, se plantean los pasos a seguir para su resolución y si hubiera un concepto o herramienta no explicada hasta ese momento, se introduce una explicación teórica con ejemplos. Al completar cada Caso de Estudio se obtiene la solución del problema planteado a través de una participación activa del lector, quien va completando planillas y resolviendo incrementalmente partes de diagramas o códigos fuente escritos en java. Al final del desarrollo de cada nivel se deben completar las Hojas de Trabajo con ejercitación para el lector.

El Nivel 1 “Búsqueda, Ordenamiento y Pruebas Automáticas” trata los temas: Requerimientos; Comprensión del Mundo del Problema; Invariantes de clase, Precondiciones, Postcondiciones; Implementación de métodos; Pruebas Unitarias; Interfaz de Usuario; Algoritmos clásicos de búsqueda; Algoritmos no recursivos de ordenamiento; Arquitectura de la solución. Se desarrollan los Casos de Estudio: 1) Un Traductor de Idiomas, 2) Un manejador de Muestras, 3) Una Exposición Canina.

El Nivel 2 “Archivos, Serialización y Tipos de Excepción” trata los temas: Persistencia; Excepciones; Pruebas Unitarias aplicadas al manejo de archivos, persistencia y excepciones; Uso básico de un depurador (entorno Eclipse). Se

desarrollan los Casos de Estudio: 1) Un Explorador de Archivos, 2) Una Tienda Virtual de Discos.

El Nivel 3 “Estructuras Lineales Enlazadas” trata los temas: Estructuras enlazadas; Algoritmos de manipulación de estructuras lineales; Interfaces de usuario complejas con nuevos componentes gráficos; Análisis y Diseño; Patrones de Algoritmo; Diagrama de Casos de Uso. Se desarrollan los Casos de Estudio: 1) Una Central de Pacientes, 2) Reservas en una Aerolínea

El Nivel 4 “Mecanismos de Reutilización y Desacoplamiento” trata los temas: Importancia de desacoplar las clases; Utilización de las interfaces para independizar contratos funcionales de las implementaciones; Aplicaciones de la Herencia en las aplicaciones en general y en particular en el uso de interfaces de usuario y excepciones; Gráficas simples en dos dimensiones. Se desarrolla el caso de estudio: 1) Un Editor de Dibujos

El Nivel 5 “Estructuras y algoritmos recursivos” trata los temas: Utilizar estructuras recursivas para representar la información del modelo del mundo; Algoritmos recursivos; Árboles binarios ordenados; Árboles n-arios. Se desarrollan los Casos de Estudio: 1) Un Directorio de Contactos, 2) Organigrama de una Empresa.

El Nivel 6 “Bases de datos y distribución básica” trata los temas: Utilización de socket para intercomunicar

programas sobre diferentes computadoras en red; Manejo de concurrencia; Almacenamiento sobre una base de datos elemental; Papel del programa servidor; Integración de todos los conceptos aprendidos en todos los Niveles; Lenguaje SQL y *Framework* JDBC. Se desarrolla el caso de estudio: 1) Juego Batalla Naval.

Puede decirse que el libro es sumamente didáctico. Introduce de manera estratégica los temas nuevos acompañándolos con ejemplos y ejercicios para el alumno. Desarrolla aspectos no

siempre considerados en los cursos de programación, que parte de la comprensión de los requerimientos, siguiendo con la asignación de responsabilidades a las clases, diseño de la firma de los métodos, especificación de los contratos (identificando precondiciones, postcondiciones), codificación del cuerpo de cada método y por último *test* de unidad. Altamente recomendable para perfeccionar la tarea de programación.

Leonel Guccione

Universidad Nacional de Mar del Plata