

# PROGRAMA INGENIERÍA DE SISTEMAS

## PRUEBAS VALIDACIÓN Y VERIFICACIÓN DE SOFTWARE

### PRÁCTICO 3-1

### 2020

#### 1. Aplicación

Se desea desarrollar una aplicación para monitorizar el estado de los bloques de disco duro y los ficheros o archivos. Se debe programar utilizando Java aplicando POO y el patrón de arquitectura MVC como las convenciones para programar en java.

- La aplicación debe mostrar por pantalla:
  - Espacio usado y espacio libre;
  - Espacio dañado;
  - Lista de sus ficheros con sus tamaños;
  - Monitoreo de actividad del CPU memoria o disco.

El ejemplo de este desarrollo es Administrador de Tareas de Windows, en data y en Interfaz Gráfica de Usuario.

#### 2. La aplicación debe ser multiplataforma

La aplicación debe monitorear dos (2) Sistemas Operativos.

- Linux
- Solaris
- Windows
- MacOS

#### 3. Arquitectura por capas

Definir el numero de capas a utilizar en el desarrollo de la aplicación.

#### 4. AspectJ

“The changes to AspectJ have been dramatic. It now includes two syntax possibilities: the traditional syntax and the annotation-based @AspectJ syntax, which lowers the barrier to begin using AspectJ. It supports many weaving possibilities, including load-time weaving, making the use of AspectJ for applications such as monitoring and tracing a simple experience. To top it off, Spring has adopted AspectJ’s programming model as its AOP solution. This caused the Spring community to be drawn to the power of AOP with AspectJ as the preferred programming model.” [1]

Utilizando AOP – AspectJ se debe monitorizar a las clases, métodos o sentencias de java que se invoquen en el transcurso de la aplicación.

- El llamado a las sentencias de las primitivas del sistema operativo o componente de software que suministre una interfaz común entre la aplicación y uno o más dispositivos de almacenamiento de datos, tales como una Base de datos o un archivo.

## TRABAJOS CITADOS

- [1] R. LADDAD, Enterprise AOP with Spring Applications AspectJ in Accion, Grand Forks: Manning Publications Co. , 2010.

AGP

# PROGRAMA INGENIERÍA DE SISTEMAS

## PRUEBAS VALIDACIÓN Y VERIFICACIÓN DE SOFTWARE

### PRÁCTICO AspectJ

#### 2021

### 1. Herramientas

En la práctica se evidenciará la utilización de la herramienta:

- AspectJ <https://www.eclipse.org/aspectj/>

### 2. AspectJ

Algunos aspectos de la implementación del sistema, como el registro, el manejo de errores, la aplicación de estándares y las variaciones de funciones, son notoriamente difíciles de implementar de forma modular. El resultado es que el código se enreda en un sistema y genera problemas de calidad, productividad y mantenimiento. El desarrollo de software orientado a aspectos permite la modularización limpia de estas preocupaciones transversales. El proyecto Eclipse AspectJ Development Tools (AJDT) proporciona soporte de herramientas basadas en la plataforma Eclipse para AOSD con AspectJ. Nuestro objetivo es ofrecer una experiencia de usuario que sea coherente con las herramientas de desarrollo de Java (JDT) al trabajar con proyectos y recursos de AspectJ. [1]

“The changes to AspectJ have been dramatic. It now includes two syntax possibilities: the traditional syntax and the annotation-based `@AspectJ` syntax, which lowers the barrier to begin using AspectJ. It supports many weaving possibilities, including load-time weaving, making the use of AspectJ for applications such as monitoring and tracing a simple experience. To top it off, Spring has adopted AspectJ’s programming model as its AOP solution. This caused the Spring community to be drawn to the power of AOP with AspectJ as the preferred programming model.” [2]

### 3. Entorno de la prueba

Utilizando AOP – AspectJ implementar un sistema de trazas con aspectos. Se tiene el típico CRUD (Create, Read, Update, Delete) de contactos, donde deseamos que cada vez que se inserte, modifique o borre un contacto se deje una traza con el tipo de acceso, utilizar un ejercicio de su autoría y crear los aspectos necesarios para hacer seguimiento al CRUD.

El documento debe estar organizado:

- a. Portada
- b. Introducción
- c. Entorno
- d. AOP (AOP (Aspect Oriented Programming).)
- e. Integración de AspectJ con Java.
- f. Taller realizado
- g. Pruebas de la ejecución del taller

- h. Conclusiones
- i. Referencias Bibliográficas

#### BIBLIOGRAFÍA

- [1] Eclipse Foundation, «Eclipse Foundation - Eclipse AJDT - AspectJ Development Tools Project,» Eclipse Foundation, 01 08 2020. [En línea]. Available: <https://projects.eclipse.org/projects/tools.ajdt>. [Último acceso: 19 10 2020].
- [2] R. LADDAD, Enterprise AOP with Spring Applications AspectJ in Accion, Grand Forks: Manning Publications Co. , 2010.

AGP