

# Sistema ANACONDA para el análisis automático de la calidad del software

**Enrique Martín**, *Telefónica Móviles España*  
**Fernando Sáenz**, *Universidad Complutense de Madrid*

1

Sistema ANACONDA para el análisis automático de la calidad del software

## **Antecedentes.**

- Reingeniería Cobol -> Natural.
- Nuevas aplicaciones.
- Intenso incremento puntual de desarrolladores.

## **El problema.**

- Frecuentes fallos *en el paso* a producción.
- Frecuentes fallos *una vez en* producción.
- Entorno de desarrollo-producción-pruebas no exactamente iguales.
- Descontento, desconcierto, ...

2

### **La petición.**

- Se crea el Departamento de Calidad
- ¿Situación real?
- “Un sistema automático que examine el código para ver si está bien”.

### **La solución.**

- Un sistema de análisis de calidad del software: ANACONDA.

3

### **Aspectos a analizar:**

- Errores.
- Normativa.
- Calidad.
  - Mantenibilidad.
  - Documentación.
  - ...

### **Condicionantes:**

- Trabajar con las herramientas disponibles (Natural y ADABAS).
- Resultados inmediatos.

4

## ¿Análisis? ¿Qué se busca?

- Se realiza una lista de todos aquellos errores más comunes y de ella se seleccionan los que se pueden detectar automáticamente.

- |  |  |
|--|--|
| · Módulos guardados después de compilados.     | · Utilización de PFs.                              |
| · Módulos llamados no existentes.              | · Líneas de comentarios frente a líneas de código. |
| · Módulos no llamados.                         | · Nomenclatura de los módulos (Normativa).         |
| · Código muerto.                               | · Campos numéricos correctamente formateados.      |
| · Archivos/campos referenciados no existentes. | · Utilización de la rutina de errores estándar.    |
| · Cabeceras.                                   | · Variables definidas no utilizadas.               |
| · Documentación.                               | · Campos de entrada sin ‘_’.                       |
| · Nivel de anidamiento.                        | · etc.   |
| · Líneas por módulo.                           |  |

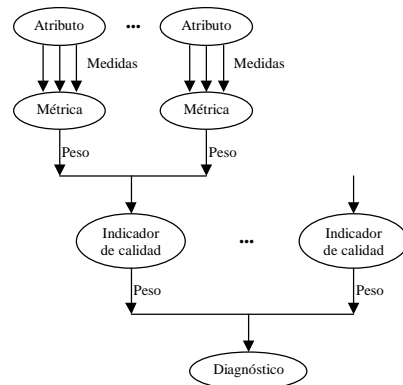
5

## Objetivo:

- Medir automáticamente todos los atributos que sea posible.
- De los indicadores de calidad obtenidos emitir un diagnóstico.
- Cuanto mayor mayor sea el número de atributos más acertado será el diagnóstico.
- Hay atributos que no se pueden medir automáticamente: se extrapola el diagnóstico obtenido de los atributos medibles a los no medibles.

6

## Elementos que intervienen en el análisis



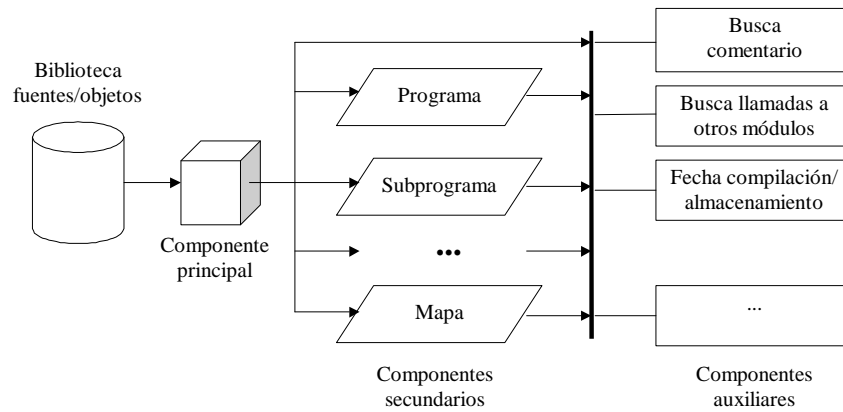
7

## Fase de medida:

- El componente principal, que lleva el control de todo el proceso, localiza uno a uno los módulos a analizar. Para cada módulo, identifica su tipo y pasa el control al componente secundario.
- Los componentes secundarios, especializados en el análisis de cada tipo de módulo, con la intervención de los componentes auxiliares, mide cada uno de los atributos correspondientes al tipo de módulo que se está analizando.
- Al devolver el control el componente secundario al componente principal, éste calcula su métrica, acumulando el resultado en el de los módulos ya analizados.
- Los componentes auxiliares están especializados en tareas concretas (determinar la fecha de almacenamiento, localizar comentarios en una línea, etc.)

8

## Fase de medida:



9

## Fase de análisis:

- Al terminar la fase de medida, el componente principal tiene las medidas globales de cada uno de los atributos.
- Para cada atributo se ha establecido una métrica basada en el conocimiento de los expertos que determina la calidad de ese atributo.
- Los indicadores de calidad se determinan al aplicar un peso a cada atributo que lo compone.
- Por último, cada indicador de calidad tiene un peso en el diagnóstico final.

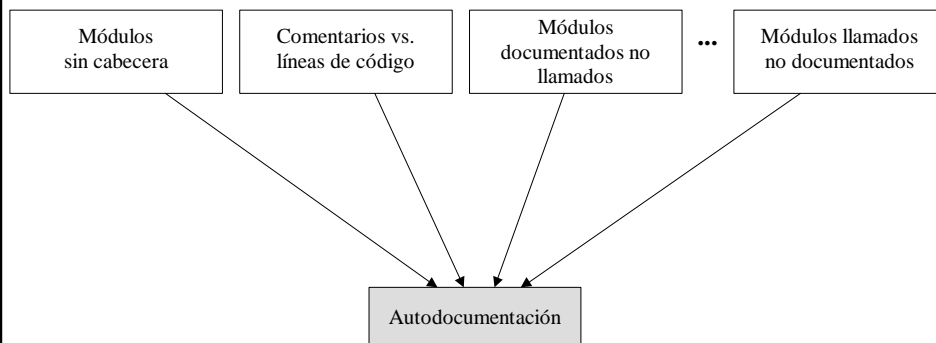
10

## Fase de diagnóstico:

- Existencia/ausencia de errores.
  - Informe de errores.
  - Autorización de paso a producción.
- Cumplimiento de la normativa.
  - Informe de incumplimientos.
  - Porcentaje de nivel de cumplimiento.
  - Recomendación de paso a producción.
- Calidad del software.
  - Informe de la calidad
  - Porcentaje del nivel de calidad.
  - Recomendación de paso a producción.

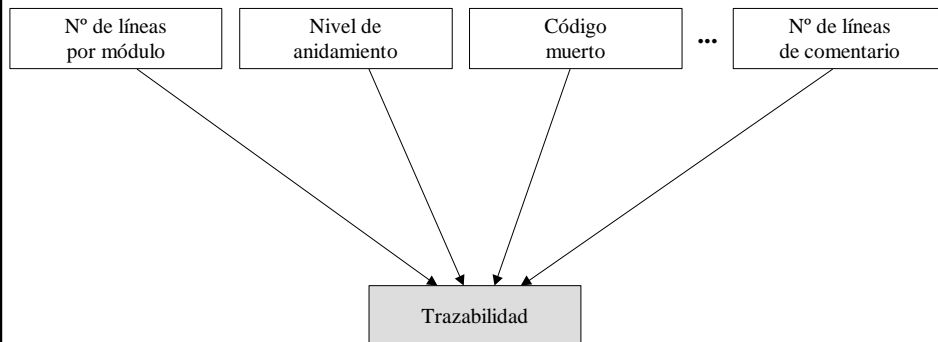
11

## Ejemplo 1: Indicador de calidad Autodocumentación



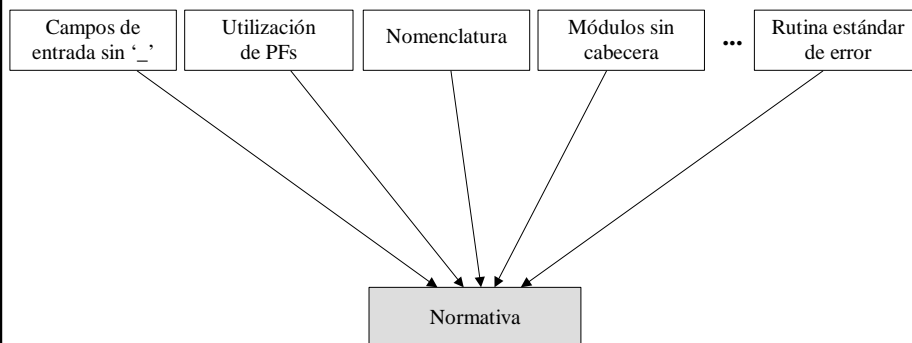
12

## Ejemplo 2: Indicador de calidad Trazabilidad



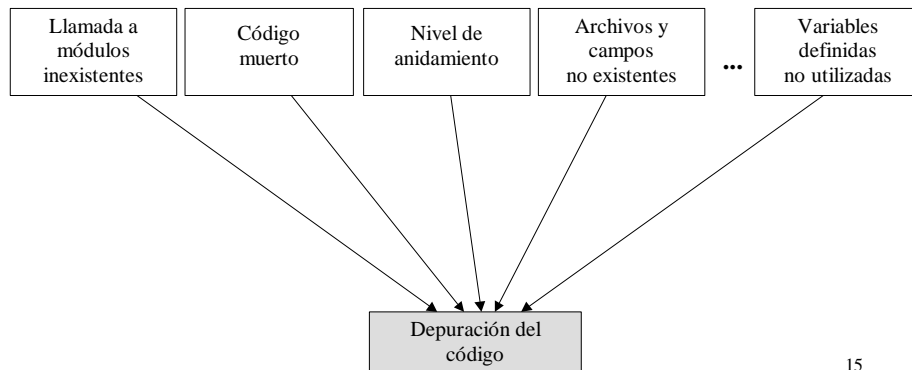
13

## Ejemplo 3: Indicador de calidad Normativa



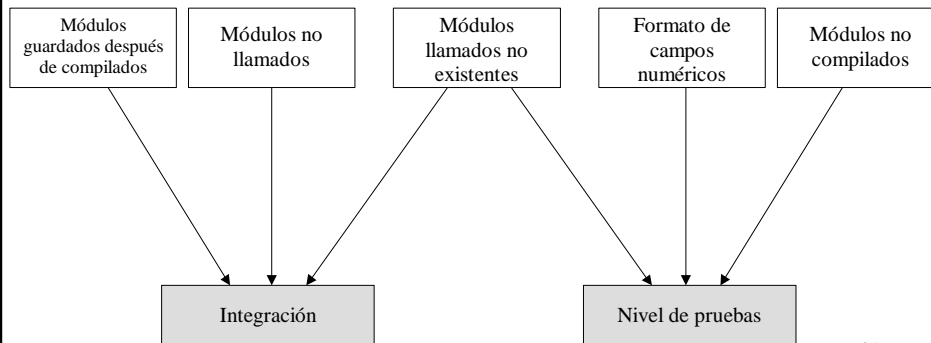
14

### Ejemplo 4: Indicador de calidad Depuración del código



15

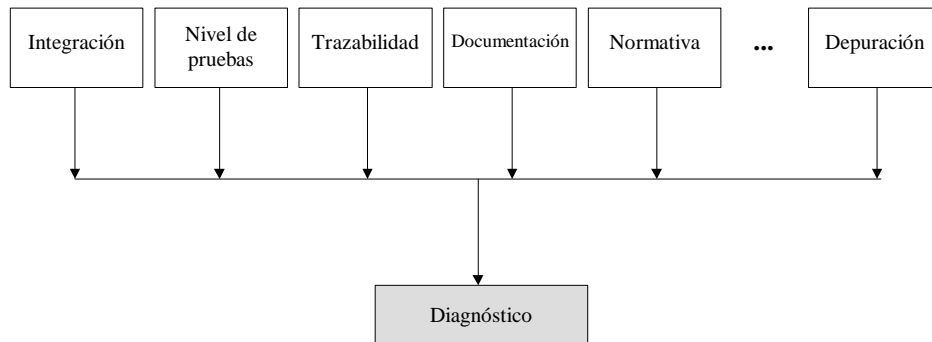
### Ejemplo 5: Indicadores de calidad Integración y Nivel de pruebas



16



## Ejemplo 6: Determinación del diagnóstico



17

## Coste

- No necesita ningún software adicional.
- No tiene costo significativo de proceso.
- Bajo costo de desarrollo.

18

## Conclusiones

- Facilita la implantación.
- Disminuye los errores.
- Facilita el mantenimiento.
- Ayuda a la toma de decisiones.
- Bajo coste de desarrollo y mantenimiento.
- Adaptable a las necesidades de cada instalación.
- Una sólida base para evolucionar.