



Alumno	Ignacio Javier Martínez Sánchez
Asignatura	Entornos de desarrollo
Curso	DAM
Año	2021-2022
Título de la práctica	Resumen examen 2T
Nombre del documento	Act_nombreactividad_nombreyprimerapellido.pdf

Enviar en formato PDF



## 1. Apartado

### Refactorización

---

La refactorización consiste en la reestructuración de un programa para hacerlo más eficiente, buscando que un mismo programa siga haciendo su función correspondiente pero llegando a una solución más eficaz, esto es, que requiera menos recursos del ordenador.

Ventajas del uso de la refactorización:

- Ayuda a encontrar errores
- Ayuda a programar más rápido
- Los programas se interpretan de forma más fácil
- Los diseños son más robustos
- Los programas tienen más calidad
- Se evita duplicar la lógica de los programas

Refactorizaciones más frecuentes:

- Extract method (reducción lógica): reducción de una función grande en varios métodos cortos que realizan tareas más específicas y crean métodos más complejos
- Métodos Inline (código embebido): sistema directamente opuesto al anterior
- Uso de variables autoaplicativas: consiste en el uso de variables para escribir líneas de código más simples en vez largas y complejas líneas difíciles de seguir.
- Uso correcto de variables temporales
- Cambio de algoritmo
- Mover métodos entre clases: busca balancear y repartir la carga de trabajo entre clases
- Cambio de atributo a otra clase que vaya a usarlo más
- Autoencapsular campos con getter y setter



- Extraer clases: delegar parte de la funcionalidad de una clase a otra subclase
- Clases delegadas
- Uso de constantes
- Encapsulamiento de arrays
- Reemplazar tipos de objeto con subclases: en ocasiones se usan variables internas para clasificar objetos, pero se recomienda cambiar este código y usar polimorfismo, creando clases derivadas
- Evitar los code smells

## 2. Apartado

### Documentación y Javadocs

---

La documentación es el proceso por el que el programador añade anotaciones con explicaciones del código, que facilitan la comprensión de este. Javadocs es una función que permite generar documentos HTML con la documentación del programa en Java, mediante el uso de `/** @ **/`

## 3. Apartado

### Patrones de diseño

---

Un patrón de diseño es un pseudocódigo de un determinado problema de diseño que al ser un problema frecuente tiene previamente diseñada una “solución” que no es más que un esquema a salir. Siguen el principio SOLID:

- Single responsibility (S): los objetos deben tener una única responsabilidad
- Open-Closed (O): cualquier entidad debe estar abierta para su extensión, pero cerrada a su modificación
- Liskov substitution (L): si se sustituye un objeto por una subclase suya, el funcionamiento debería ser idéntico



- Interface segregation (I): es mejor crear muchas interfaces específicas separadas antes de una interfaz genérica
- Dependency inversion (D): se debe depender de abstracciones, no de implementaciones concretas

Características:

- Pueden usarse bajo distintas circunstancias (reusabilidad)
- Han resuelto problemas parecidos en otras ocasiones (efectividad)
- Permiten comunicación fluida y común entre ingenieros y desarrolladores
- Estandarización
- Facilitan la comprensión de proyectos o estructuras más complejas
- No impiden que se use otra estrategia de diseño

Ejemplo de patrón de diseño:

<https://refactoring.guru/design-patterns/chain-of-responsibility>

## 4. Apartado GIT

---

Git es un sistema de control de versiones que permite hacer seguimiento de uno o varios directorios y los archivos contenidos en este con el fin de tener presentes los cambios en cada una de las modificaciones que hacemos en nuestro trabajo. Hay dos tipos de sistemas de control de versiones:

Sistemas centralizados:

- Tienen una gestión más sencilla
- El repositorio es único
- Tienen un mayor control y existe un número único de versión



- Las ramas o versiones alternativas se crean únicamente en el servidor
- Sólo existe una única ubicación de la información

#### Sistemas distribuidos:

- Cada programador tiene una copia del repositorio (cada desarrollador tiene una copia completa)
- En caso de fallo hay múltiples copias para poder restaurar
- En el servidor principal reside la copia principal (repositorio principal)
- Los desarrolladores pueden crear ramas locales para hacer ciertas pruebas o modificaciones
- El trabajo es mucho más ágil
- Se reduce la carga de trabajo del servidor

GIT es un sistema de control de versiones distribuido

GITHUB tienen una interfaz gráfica pero con GIT se trabaja mediante comandos en la CMD con la que haremos nuestro seguimiento y subiremos nuestro trabajo al repositorio de GITHUB.

Comandos básicos de la CMD de GIT:

