



IES JAROSO

UNIDAD 3 : OPTIMIZACIÓN Y DOCUMENTACIÓN



Nombre y apellidos:

Curso:1º DAW

Fecha:21/02/2024

RESULTADO DE APRENDIZAJE

RA4.-Optimiza código empleando las herramientas disponibles en el entorno de desarrollo.

CRITERIOS DE EVALUACIÓN

- a) Se han identificado los patrones de **refactorización** más usuales.
- b) Se han elaborado las pruebas asociadas a la **refactorización**.
- c) Se ha revisado el código fuente usando un analizador de código.
- d) Se han identificado las posibilidades de configuración de un analizador de código.
- e) Se han aplicado patrones de refactorización con las herramientas que proporciona el entorno de desarrollo.
- f) Se ha realizado el **control de versiones** integrado en el entorno de desarrollo.
- g) Se ha **documentado** el código fuente mediante comentarios.
- h) Se han utilizado herramientas del entorno de desarrollo para **documentarlos** procesos,datos y eventos.
- i) Se han utilizado herramientas del entorno de desarrollo para **documentar** las clases.

INSTRUCCIONES

PARTE DE REFACTORIZACIÓN

- Se realiza en Eclipse oIntelliJ
- Se proporciona el código necesario en el proyecto **RefacTablero2**.
- **Exportar el proyecto** con las refactorizaciones realizadas.

PARTE GIT

- Debe estar GIT instalado y configurado.
- Debe estar SSH instalado.
- La solución se entrega con el enunciado de la pregunta y la(s) captura(s) de pantalla correspondiente, en un documento pdf.
- Cada pregunta tiene un valor de 1 punto.
- Tiempo de realización: 1,5 horas.

Entregar un fichero comprimido con el proyecto de refactorización y el fichero pdf de GIT.

REFACTORIZACIÓN

1. Importa el proyecto **RefacTablero2**. Modifica el método **dibujarTablero()** para que NO HAYA CÓDIGO DUPLICADO, mediante las siguientes operaciones:
 1. Extrae variables locales necesarias para las filas 1-7.
 2. Extrae métodos para las filas 1-7.
 1. Un método para imprimir la fila.
 2. Un método para el contenido de la fila.
 3. Métodos adicionales si son necesarios.
 3. Para la fila 0 y el pie del tablero genera los métodos que abrevien el código.

GIT REMOTO

1. En Ubuntu, crea un repositorio tipo bare en un directorio llamado **examen_tuNombre**.

```
alumno1@pc12:~/Documentos/github1/2trimestre/Las demas asignaturas/Entornos/Refactorización$ git init --bare examen_MartinGarbiso/
ayuda: Usando 'master' como el nombre de la rama inicial. Este nombre de rama predeterminado
ayuda: está sujeto a cambios. Para configurar el nombre de la rama inicial para usar en todos
ayuda: de sus nuevos repositorios, reprimiendo esta advertencia, llama a:
ayuda:
ayuda: git config --global init.defaultBranch <nombre>
ayuda:
ayuda: Los nombres comúnmente elegidos en lugar de 'master' son 'main', 'trunk' y
ayuda: 'development'. Se puede cambiar el nombre de la rama recién creada mediante este comando:
ayuda:
ayuda: git branch -m <nombre>
Iniciado repositorio Git vacío en /home/alumno1/Documentos/github1/2trimestre/Las demas asignaturas/Entornos/Refactorización/examen_MartinGarbiso/
```

```
alumno1@pc12:~/Documentos/github1/2trimestre/Las demas asignaturas/Entornos/Refactorización/examen_MartinGarbiso$ ls
branches  config  description  HEAD  hooks  info  objects  refs
```

2. En Ubuntu clona el repositorio anterior en un directorio llamado **proyecto_examen1_tuNombre**.

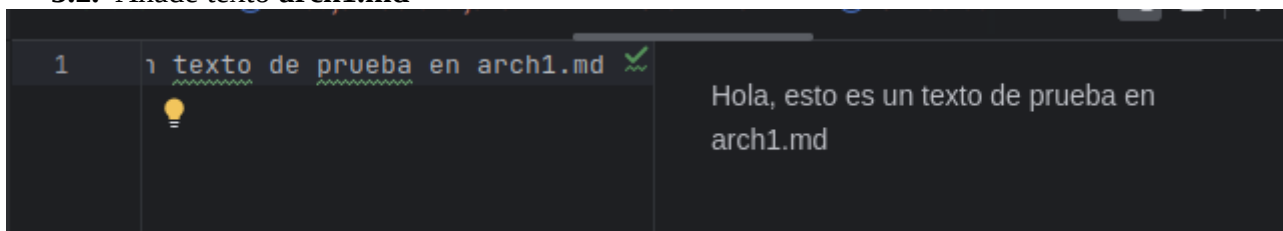
```
alumno1@pc12:~/Documentos/github1/2trimestre/Las demas asignaturas/Entornos/Refactorización$ git clone --bare examen_MartinGarbiso proyecto_examen1_MartinGarbiso/
Clonando en un repositorio vacío 'proyecto_examen1_MartinGarbiso'...
warning: Pareces haber clonado un repositorio sin contenido.
hecho.
```

3. En **proyecto_examen1_tuNombre** realiza y confirma las siguientes operaciones:

3.1. Añade 2 archivos llamados **arch1.md** y **arch2.md** con el contenido que desees.

```
alumno1@pc12:~/Documentos/github1/2trimestre/Las demas asignaturas/Entornos/Refactorización/proyecto_examen1_MartinGarbiso$ touch arch1.md arch2.md
alumno1@pc12:~/Documentos/github1/2trimestre/Las demas asignaturas/Entornos/Refactorización/proyecto_examen1_MartinGarbiso$
```

3.2. Añade texto **arch1.md**



```
alumno1@pc12:~/Documentos/github1/2trimestre/Las demas asignaturas/Entornos/Refactorización/proyecto_examen1_MartinGarbiso$ git add arch1.md
```

3.3. Elimina **arch2.md**

```
alumno1@pc12:~/Documentos/github1/2trimestre/Las demas asignaturas/Entornos/Refactorización/proyecto_examen1_MartinGarbiso$ git rm arch2.md
```

3.4. Renombra **arch1.md** por **fich1.md**

```
alumno1@pc12:~/Documentos/github1/2trimestre/Las demas asignaturas/Entornos/Refactorización/proyecto_examen1_MartinGarbiso$ git mv arch1.md fich1.md
```

3.5. Envía los datos a `examen_tuNombre`.

```
alumno1@pc12:~/Documentos/github1/2trimestre/Las demas asignaturas/Entornos/Refactorización/proyecto_examen1_MartinGarbiso$ git commit -m "Cambio de arch1 a fich1 y eliminar arch2"
[master (commit-raíz) 2cc16a8] Cambio de arch1 a fich1 y eliminar arch2
1 file changed, 1 insertion(+)
create mode 100644 fich1.md
```

```
alumno1@pc12:~/Documentos/github1/2trimestre/Las demas asignaturas/Entornos/Refactorización/proyecto_examen1_MartinGarbiso$ git push origin master
Enumerando objetos: 3, listo.
Contando objetos: 100% (3/3), listo.
Escribiendo objetos: 100% (3/3), 249 bytes | 249.00 KiB/s, listo.
Total 3 (delta 0), reusados 0 (delta 0), pack-reusados 0
To /home/alumno1/Documentos/github1/2trimestre/Las demas asignaturas/Entornos/Refactorización/examen_MartinGarbiso
* [new branch]      master -> master
```

4. En Windows o Ubuntu trae los datos del repositorio `examen_tuNombre` al directorio `proyecto_examen2_tuNombre`.

```
alumno1@pc12:~/Documentos/github1/2trimestre/Las demas asignaturas/Entornos/Refactorización$ git clone examen_MartinGarbiso proyecto_examen2_MartinGarbiso/
Clonando en 'proyecto_examen2_MartinGarbiso'...
hecho.
```

4.1. Visualiza los archivos.

```
ización$ cd proyecto_examen2_MartinGarbiso/
alumno1@pc12:~/Documentos/github1/2trimestre/Las demas asignaturas/Entornos/Refactorización/proyecto_examen2_MartinGarbiso$ ls
fich1.md
```

4.2. Realiza y confirma las siguientes operaciones:

4.2.1. Añade un directorio llamado **prueba** y dentro de él crea un fichero llamado **fich3.md** con el contenido que desees.

```
alumno1@pc12:~/Documentos/github1/2trimestre/Las demas asignaturas/Entornos/Refactorización/proyecto_examen2_MartinGarbiso$ mkdir prueba
alumno1@pc12:~/Documentos/github1/2trimestre/Las demas asignaturas/Entornos/Refactorización/proyecto_examen2_MartinGarbiso$ cd prueba/
alumno1@pc12:~/Documentos/github1/2trimestre/Las demas asignaturas/Entornos/Refactorización/proyecto_examen2_MartinGarbiso/prueba$ touch fich3.md
alumno1@pc12:~/Documentos/github1/2trimestre/Las demas asignaturas/Entornos/Refactorización/proyecto_examen2_MartinGarbiso/prueba$ cd ..
```

4.2.2. Elimina el **commit** que corresponde a la eliminación de `arch2.md`

```
alumno1@pc12:~/Documentos/github1/2trimestre/Las demas asignaturas/Entornos/Refactorización/proyecto_examen2_MartinGarbiso$ git revert "Cambio de arch1 a fich1 y eliminar arch2"
```

4.3. Envía los datos a `examen_tuNombre`.

```
alumno1@pc12:~/Documentos/github1/2trimestre/Las demas asignaturas/Entornos/Refactorización/proyecto_examen2_MartinGarbiso$ git commit -m "fich3 y revert de arch2"
[master 52b2c23] fich3 y revert de arch2
1 file changed, 2 insertions(+)
create mode 100644 prueba/fich3.md
```

5. En **proyecto_examen1_tuNombre** visualiza el contenido del repositorio actualizado.

```
alumno1@pc12:~/Documentos/github1/2trimestre/Las demas asignaturas/Entornos/Refactorización/proyecto_examen1_MartinGarbiso$ ls
arch2.md  fich1.md
alumno1@pc12:~/Documentos/github1/2trimestre/Las demas asignaturas/Entornos/Refactorización/proyecto_examen1_MartinGarbiso$
```