

Contexto

La siguiente prueba se enmarca dentro de las actividades de repaso de la 2.ª evaluación del módulo “Entornos de Desarrollo”. A través de esta prueba se evaluará el diseño y realización de pruebas y la documentación de código.

Herramientas necesarias

- *Eclipse IDE*. Se utilizará Eclipse/IntelliJ/NetBeans y otro IDE de vuestra elección como entorno de desarrollo para realizar las pruebas.
<https://www.eclipse.org/eclipseide/>
- *GIT*. Se utilizará GIT como herramienta para el control de versiones.
<https://git-scm.com>
- *JavaDoc*. Se utilizará la herramienta JavaDoc para la documentación de código Java.
<https://www.oracle.com/technetwork/java/javase/documentation/index-137868.html>
- *JUnit*. Se utilizará el framework JUnit para la implementación de pruebas unitarias.
<https://junit.org/junit5/>
- *Github*. Repositorio remoto de código fuente.
<https://github.com>

Pasos previos para la realización de la práctica

1. Descargar fichero comprimido (.zip)
2. Crear un proyecto Java en vuestro IDE con dicho código.
3. Comprobar que es posible su ejecución. Solo si se ejecuta, no hace falta comprobar su correcta ejecución.
4. Crear una cuenta en GitHub y posteriormente un repositorio llamado “PruebaED_2a”.
Asegúrate de que ese repositorio es público. Si ya tenías una cuenta creada, no es necesario crear una nueva.

Ejercicios

Los ejercicios se dividirán en tres bloques principalmente, donde tendrás que (1) preparar el entorno de trabajo con el gestor de versiones GIT; (2) implementar un conjunto de pruebas unitarias, y (3) documentar el código utilizando javadoc.

Bloque 1. Preparar el entorno de trabajo

- Debes convertir la carpeta del proyecto en un repositorio local del git.
- Debes asociar tu repositorio remoto “PruebaED_2a” a dicho repositorio local.
- Sube el código de tu proyecto (sin modificar) al repositorio remoto. El mensaje del commit debe ser “codigo inicial”.

Bloque 2. Implementar pruebas unitarias

- Deberás implementar tests unitarios utilizando JUnit para **algunos** de los métodos de las clases Curso y Persona. Los métodos para los que tienes que crear tests están indicados con un comentario. Por ejemplo:

```
/*
 * Documentar y crear test unitario
 */
public void aniadirAlumno(Persona p) {
    listaAlumnos.add(p);
}
```

- Cada vez que implementes un test unitario deberás hacer un commit con el mensaje “implementacion test <nombreMetodo>”
- Cuando termines de implementar todos los test, sube el código a tu repositorio remoto.

Bloque 3. Documentación de código

- Deberás documentar los métodos indicados con un comentario de las clases Curso y Persona. También debes documentar la clase añadiendo información sobre el autor, versión, ...

```
/*
 * Documentar y crear test unitario
 */
public void aniadirAlumno(Persona p) {
    listaAlumnos.add(p);
}
```

- Cada vez que documentes una clase deberás hacer un commit con el mensaje “documentada <nombreClase>”
- Genera la documentación html en la misma carpeta del proyecto.
- Cuando termines de documentar todas las clases, sube el código junto con la documentación a tu repositorio remoto.

Entrega

Escribir la URL de tu repositorio online en github en la tarea del aula virtual.

Calificación

- Las tareas relacionadas con el control de versiones serán el 40% de la nota.
- Las tareas relacionadas con la documentación del código serán el 40% de la nota.
- Las tareas relacionadas con los tests unitarios serán el 20% de la nota.