



Prácticas Java

T05 CrowdFERNAN

Introducción

En este tema vas a realizar:

- Creación de clases base *Usuario*, *Proyecto* e *Inversion* y de gestión.
- **Biblioteca de funciones para trabajar con fechas.**
- Creación del diagrama UML de clases.
- Documentación mediante JavaDoc
- Publicación en GitHub del nuevo código en una nueva rama

Todo ello, siguiendo las recomendaciones de código limpio y controlando todos los posibles errores.

Recuerda que tienes una explicación general de toda la práctica en su conjunto en el documento “Práctica obligatoria 23-24 Explicación general – CrowdFERNAN.docx” en la Moodle (apartado “General”).

Creación de clases base y de gestión

Crea las clases base *Usuario*, *Proyecto* e *Inversion*. Cada una con sus atributos, constructores y métodos que creas necesarios.

Crea las clases necesarias para gestionar las anteriores. Por ejemplo, *GestionUsuario* tendrá como atributo un array estático de usuarios y los correspondientes métodos para insertarlos, eliminarlos, modificarlos, buscarlos...

En este punto ya no estamos limitados por número de usuarios ni de proyectos. Establece una cantidad inicial para esos arrays estáticos que deberán crecer en el caso de que se llenen.

En una inversión tendrás que relacionar un usuario con un proyecto, piensa bien cómo hacerlo. Un dibujo antes de programar seguro que te ayudará...

Si lo haces bien, todo el trabajo previo de login, registro de usuarios mediante token... te seguirá funcionando.

Biblioteca de funciones de fechas

Crea una *biblioteca funcionesFechas* en un paquete llamado *utilidades* para trabajar con fechas. Las funciones deben resolver los siguientes aspectos:

- Decir si una fecha/hora es anterior a otra.
- Obtener los segundos/minutos/horas/días restantes entre una fecha pasada por parámetro y ahora.
- Convertir a nuestro formato dd/mm/aaaa.



Prácticas Java

- Cualquier otra función que creas conveniente para el funcionamiento de la aplicación...

Trabajar en las funciones con `LocalDate`, `LocalTime` y `LocalDateTime`.

En este punto, después de las prácticas anteriores, tu paquete *utilidades* debería tener *funcionesFechas*, *funcionesCadenas* y *funcionesCorreos*.

Diagrama de clases

Crea el diagrama UML de clases de tu aplicación. Puedes utilizar cualquier aplicación como, por ejemplo <https://app.diagrams.net/>

Guarda el diagrama resultante en png.

Documentación

Generar documentación de todas las funciones con *JavaDoc* y entregar el html resultante.

Añadir al documento que elaboraste en el tema anterior, la nueva funcionalidad que se ha desarrollado en esta práctica para que la documentación esté al día.

Sería recomendable añadir en el nombre la *versión del documento* (v2, v3, v4...) y dentro del mismo, la *fecha de la última actualización*.

Repositorio en GitHub

Añadir el nuevo código al repositorio del proyecto en GitHub, pero en una nueva rama.

Vamos a tener una rama por cada tema, así que, crea una rama con nombre **tema 5** que será la que apunte a los últimos cambios que vayas realizando en esta práctica.

Cuando vayas a entregar la práctica, fusiona la rama **tema 5** con la rama **master**. De esta forma, la rama master tendrá siempre los últimos cambios y mantendrás el resto de ramas que vayas creando para ver la evolución del código tema a tema.

Entrega de la práctica

La entrega de la práctica se hará en la Moodle en un archivo .zip con lo siguiente:

- a. Documentación del programa a modo de manual de usuario actualizado.
Especificar en el documento la URL del repositorio de GitHub.
- b. Diagrama UML de clases en png.
- c. Documentación en html generada con JavaDoc.
- d. Carpeta src/ con los ficheros .java con el código.
- e. Archivo .jar.