



# Prácticas Java

## T04 CrowdFernan

### Introducción

En este tema vas a realizar:

- Biblioteca de funciones para comprobación de cadenas (fortaleza de contraseñas, rangos, formatos...)
- Almacenamiento de usuarios y proyectos en arrays estáticos
- Registro de nuevos usuarios enviando token aleatorio al correo
- Generación del archivo .jar que empaquete el proyecto y 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”).

### Biblioteca de funciones de cadenas

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

- Que 2 contraseñas sean iguales (para utilizar en el registro de usuarios)
- Fortaleza de la contraseña (mínimo 8 caracteres, con minúsculas, mayúsculas, número y algún símbolo como `_,*,+@`)
- Comprobación de mínimos y máximos en textos (título y descripción de proyectos...)
- Comprobación de formatos en textos (`'%` en campos de interés y similares)

### Almacenamiento en arrays estáticos

Refactorizar el código utilizando arrays para almacenar los usuarios (10) y proyectos (20). Piensa bien cómo realizar las inversiones de cada usuario en varios proyectos. Para simplificar las pruebas, crea a mano 1 usuario de cada tipo y 1 proyecto.

Puedes utilizar arrays de 1 o 2 dimensiones.

### Registro de nuevos usuarios

Crea una nueva opción para que un usuario se pueda registrar a sí mismo. Sólo es posible esta opción para usuarios *inversores* y *gestores*.



# Prácticas Java

Al registrar un nuevo usuario, antes de poder iniciar sesión se tiene que validar su correo enviándole un token aleatorio. Algo similar a lo que se hizo con la autenticación 2FA mediante el código OTP.

Integra esta opción en la *biblioteca funcionesCorreos* con el envío de correos, generación del token... Alojrarlas en un paquete llamado *utilidades*.

## Empaquetar el proyecto

Crear archivo .jar que empaquete todo el proyecto. Ver en la Moodle (apartado “Herramientas y utilidades”) instrucciones para generarlo.

## 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. Explica cómo ejecutar el jar de la aplicación.

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 4** 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 4** 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:

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