UNIVERSIDAD DEL VALLE DE GUATEMALA
CAMPUS CENTRAL
FACULTAD DE INGENIERÍA
Departamento de Ciencia de la Computación
Programación de plataformas móviles

Laboratorio #2

Objetivo

El objetivo de este laboratorio es que el alumno aplique los conceptos aprendidos sobre el lenguaje Kotlin. Familiarizarse con el entorno de desarrollo IntelliJ Community Edition.

Requisitos

Se desea almacenar la información del perfil de usuario de las personas que utilizan una aplicación.

Para esto debe crear una clase en Kotlin llamada **PerfilUsuario** que represente esta información con las siguientes propiedades:

- ID. Tipo entero. Almacena un identificador único.
- Nombres. Tipo cadena. Almacena los nombres del usuario.
- Apellidos. Tipo cadena. Almacena los apellidos del usuario.
- UrlPhoto. Tipo cadena. Almacena la url de una foto de perfil. Puede ser nulo.
- Edad. Tipo entero.
- Correo. Tipo cadena. Almacena el correo electrónico del perfil.
- Biografía. Tipo cadena. Puede ser nulo.
- Estado. Debe asegurar que esta propiedad solo pueda tener uno de estos 3 valores:
    - Activo
    - Pendiente
    - Inactivo
- Hobbies. Tipo Lista que contiene elementos de tipo **Hobby** (esta clase se define a continuación)

La clase **Hobby** debe tener los siguientes campos:

- Título. Tipo cadena.
- Descripción. Tipo cadena.
- UrlPhoto. Tipo cadena. Almacena la url de una foto asociada al hobby. Puede ser nulo.

La clase **PerfilUsuario** debe tener las siguientes funciones:

- AgregarHobby: recibir un objeto de tipo Hobby y agregarlo a la lista de Hobbies.

El programa debe iniciar la ejecución con un listado de usuarios precargados, los cuales debe definir y crear usted mismo.

Debe desarrollar estas opciones para trabajar con el listado de perfiles:

1. Crear perfil: Solicitar la información de un perfil de usuario y agregarlo a la lista.
2. Buscar perfil de usuario(s): Buscar perfiles por medio del ID o mediante nombres y/o apellidos. Debe mostrar toda la información del perfil en caso de ser encontrado, incluyendo todos los hobbies. En caso de no encontrar un perfil con la información ingresada debe dar un mensaje de respuesta indicando este caso.
3. Eliminar perfil: Eliminar un perfil de usuario específico por medio del ID. Debe dar un mensaje de respuesta al usuario indicando el resultado de esta operación.
4. Agregar Hobby: Buscar un perfil de usuario por ID o nombres y/o apellidos y agregar Hobby. En caso de no encontrar un perfil con la información ingresada debe dar un mensaje de respuesta indicando este caso.

Para solicitar la información al usuario puede hacerlo a través de la función **readln()** y ejecutar la aplicación mediante la consola, o puede utilizar cualquier interfaz gráfica si así lo desea. Este punto queda abierto a la creatividad del estudiante.

El programa debe manejar los errores y excepciones correspondientes, dando al usuario mensajes que faciliten su el control del error y recuperación.

Criterio de evaluación y entregables

- Subir los archivos del proyecto en IntelliJ a un repositorio de Github y compartir el link en la entrega de Canvas.
- Grabar un video de menos de 2 minutos explicando cómo funciona su programa y el código que desarrollaron. Subirlo a Youtube y compartir el link.
- La ponderación será la siguiente:
    - Modelos y propiedades = 20 puntos
    - Perfiles precargados = 10 puntos
    - Crear perfil = 10 puntos
    - Buscar perfil = 10 puntos
    - Eliminar perfil = 10 puntos
    - Agregar Hobby = 10 puntos
    - Video = 20 puntos
    - Se debe evitar el uso de código innecesario y mantener un estilo consistente en el código. – 10 puntos.
- Si no se puede acceder el link al momento de la calificación se descontarán 20 puntos por cada día que pase hasta que se pueda ingresar.
- Debe subir todos los archivos del proyecto de IntellyJ (principalmente los .kt). NO archivos comprimidos como .zip o cualquier otro tipo. Si no se puede revisar en el repositorio de Github se colocará 0 en la nota de este laboratorio.