

MÓDULO INNOVACIÓN EN GESTIÓN DE DATOS

EVIDENCIA DE APRENDIZAJE N°2

Programación I

Introducción a Programación Orientada a Objetos (POO) y al uso de Archivos en Python (archivos de texto y binarios).

- Crear las clases **Usuario** (id, username, password, email) y **Acceso** (id, fechaIngreso, fechaSalida, usuarioLogueado). Donde un usuario puede tener muchos accesos y un acceso tiene el dato de un usuario.
- La aplicación debe tener un menú principal que permita:
 - **Hacer el CRUD de los Usuarios en POO**
 - **Agregar** un nuevo usuario,
 - **Modificar** un usuario,
 - **Eliminar** un usuario (dado su username o email)
 - **Buscar** por email o username y mostrar los datos de ese usuario si existe o dejar un mensaje que no existe el usuario buscado,
 - y un **Mostrar todos** los usuarios
 - **Salir de la aplicación** y
 - La opción de **ingresar al sistema**, en esta opción se solicitará el username y password, y de existir, ingresa al sistema (mostrar un mensaje que ingreso, y dar opciones para salir del sistema y volver al menú principal o salir directamente de la aplicación).
- Al hacer el CRUD de Usuario en POO a los objetos, es decir, las instancias, almacenarlas en un archivo binario llamado **usuarios.ispc** (antes estaban en un diccionario).
- En cada acceso de un usuario a la aplicación, previo control (si los datos de ingreso del usuario, username y password existen y son correctos), además de mostrar un mensaje que ingresó a la aplicación (y dejar una opción de volver al menú principal o salir), dejar registro en **accesos.ispc** (otro

archivo binario) el objeto Acceso y en el caso contrario, si no existe el usuario o los datos son incorrectos dejar un registro en **logs.txt** de los intentos fallidos de ingreso, detallando fecha completa (día y hora) y los datos ingresados (usuario y clave).

- Que en el repositorio se encuentre a modo de evidencia de uso y prueba, los archivos usuarios.ispc y accesos.ispc (log.txt no es necesario dejarlo).
- En el repositorio se debe dejar lo requerido en una carpeta llamada **evidencia2**, dentro de ella además debe tener un archivo README.md con los datos referidos a esta evidencia, algunas preguntas disparadoras que ayuden a redactar este readme: ¿Cómo ejecutar y probar este programa? ¿Qué inconvenientes han tenido y si lograron superarlos? ¿Es necesario instalar algo además de python para ejecutar el programa?

Base de Datos II:

Creación de Base de datos relacional en MySQL, creación de tablas y agregación de relaciones (claves primarias y foráneas). Inyección de datos y creación de consultas simple y complejas realizando interacción entre las tablas.

- Crear la base de datos relacional y sus tablas (con sus columns y PK, FK)
- Insertar datos en cada una de las tablas para permitir el testeo de las consultas de SQL que se van a utilizar en el PF.
- Creación de consultas SQL para la extracción de la data que luego será utilizada por el programa realizado en Python. Entre estas consultas deberá incluir un CRUD (select, insert, update, delete) y al menos 2 consultas utilizando JOIN. Tener en cuentas las buenas prácticas en cuanto a la sintáxis SQL y a las relaciones.
- En el repositorio se debe dejar lo requerido en una carpeta llamada evidencia2, dentro de ella una subcarpeta llamada BaseDeDatos donde se encuentre, la base de datos exportada en extensión .sql, y un archivo adicional, que puede ser en .sql también, con las consultas SQL que van a ser



utilizadas en el backend del programa hecho en Python para la extracción y uso de la data.

ENTREGA: En repositorio que se encuentra registrado en la planilla de los grupos de trabajo

Fecha límite de Entrega: Domingo 03/10/24