



INFORMATICA 01- BIOINGENIERÍA

EXAMEN 02- 2023-01

Docente: Verónica Henao Isaza

INDICACIONES GENERALES

El presente examen ha sido diseñado para ser resuelto de forma individual. Usted deberá dar solución a cada uno de los puntos que se encuentran en este documento en una carpeta de trabajo. Pueden usar todos los elementos vistos en clase incluyendo diapositivas, resúmenes, pero no está permitido la interacción con otros compañeros ni búsquedas en navegadores, en caso de detectarse dicha interacción, se considerará fraude y se anulará el examen. Una vez termine de resolver los ejercicios, deberá subir su solución al teams del curso. No se acepta ningún otro tipo de entrega:

- La solución del examen deberá estar en una carpeta comprimida, únicamente se califica este formato.
- Comprima la carpeta con la siguiente codificación: **P2_NOMBRE_APELLIDO**
- Todas las decisiones tomadas para la solución del ejercicio deben estar bien justificado.

Tiene 2 horas disponibles para resolverlo y montar toda la evidencia en la clase del teams. Exámenes entregados con retraso se penalizarán con 0.5 por cada 5 minutos de retraso.

EJERCICIO

Para el sistema de ingreso privado de la plataforma del Hospital San Vicente de Paul, existen ciertas credenciales de acceso que solo tienen algunos usuarios. Diseñe un sistema que valide que el usuario y contraseña entregado por teclado por el usuario corresponde a un usuario válido en la plataforma. Si este no se encuentra registrado en la base de datos, entonces deberá mostrarle un mensaje de alerta indicándole que el usuario o contraseña ingresado es errónea. Para identificar que el usuario y contraseña sea válida, deberá buscar la información en la base de datos entregada que se encuentra en el archivo 'usuarios.py' el cual deberá importar la información contenida al archivo 'main.py', el cual será el archivo principal que implementará los menús para la interacción con el usuario.

Una vez el usuario se haya logueado exitosamente, deberá imprimirse la información del usuario, como nombre del usuario, cédula, permisos y última fecha de ingreso. Tenga en cuenta, que deberá añadir a la base de datos la fecha de ingreso del sistema, cada vez que se inicie sesión.

Una vez imprima la información, muestre el siguiente menú al usuario.

1. Consultar a partir del número de identificación si un paciente tiene agendado citas médicas.

Observaciones:

- Imprima la información del paciente
 - Imprima fecha de agendamiento de la cita
 - Imprima hora de agendamiento de la cita
2. Consultar a partir del número de identificación si un paciente tiene exámenes de laboratorio.
- Observaciones:**
- Imprima la información del paciente
 - Imprima el o los exámenes de laboratorio en estado pendiente
 - Imprima el o los exámenes de laboratorio en estado aprobado
3. Consultar exámenes de laboratorio de un paciente en estado **pendiente**.
- Observaciones:**
- Imprima el o los exámenes de laboratorio en estado pendiente
4. Consultar exámenes de laboratorio de un paciente en estado **aprobado**
- Imprima el o los exámenes de laboratorio en estado aprobado
5. Ingresar paciente para agendar exámenes de laboratorio en donde debe mantener la misma estructura que se presenta en la base de datos entregada en el archivo *'exámenes_laboratorio.py'*
- Solicitar número de identificación
 - Solicitar Nombres
 - Solicitar Apellidos
 - Ingresar fecha de agendamiento del examen de laboratorio
 - Ingresar el examen en estado pendiente para el cual se agendo
6. Salir

Validaciones necesarias:

El sistema debe realizar las siguientes validaciones para su correcto funcionamiento:

- Fecha: debe tener formato yyyy-mm-dd, validar que los días correspondan con el mes correspondiente, incluso para años bisiestos.
- Nombre: Solo puede contener letras y espacios.
- Contraseña: Debe tener mínimo 4 caracteres.
- Hora: debe tener formato de 24h con la siguiente estructura hh:mm, validar que las horas, minutos y segundos tengan correspondencia.

El programa desarrollado debe cumplir con las siguientes:

- Utilizar funciones en el desarrollo del programa a criterio del programador con sus respectivas. Mínimo debe tener **3 funciones**.
- Debe validar las entradas por teclado a nivel de tipo y de rango (cuando así corresponda).
- Crear al menos dos módulos (archivos fuente), uno para el programa principal que implementará los menús para la interacción con el usuario, y otro para las funciones que se encargarán de la administración de la información de las bases de datos.

- Todos los métodos restringidos según se indicó en las clases de teoría, no pueden usarse en el desarrollo de la solución.
- Utilizar strings, listas y diccionarios, según sea conveniente. No se admiten otros tipos de datos o métodos avanzados de Python que no se hayan visto en clase