

TRABAJO FINAL - INFORMÁTICA I

BIOINGENIERÍA - UNIVERSIDAD DE ANTIOQUIA

1. HISTORIA DE USUARIO

El banco de sangre Bio Blood Bank está interesado en desarrollar una aplicación en Python que le permita almacenar, consultar, editar y eliminar resultados de pruebas cruzadas que se realizan a los hemocomponentes en el banco de sangre a las unidades entregadas en las jornadas de donación voluntaria, la información a gestionar es la información del donante (nombre, documento, edad y género) y las siete pruebas básicas que son: HCV – Hepatitis C, HBV – Hepatitis B, CMV – Citomegalovirus, VIH – Virus de Inmunodeficiencia Humana, HTLV - Virus Linfotrópico T Humano, Sífilis y Chagas.

La información de las pruebas anteriores es cualitativa, es decir, el resultado para cada una de las siete pruebas es Positivo – POS o Negativo – NEG y la única forma que una unidad de sangre y sus derivados sea apta para transfusión, es que las siete pruebas sean negativas, si por lo menos una de ellas es positiva, la unidad desde ser incinerada, ya que representa un riesgo de infección para los especialistas el banco de sangre o el paciente si se llega a transfundir por error.

Debido a que la información que se almacenará en el aplicativo es información sensible de los donantes y se debe proteger su privacidad, este aplicativo debe tener una protección de ingreso, es decir, un login, de tal forma que sea una barrera de acceso para quienes no tienen permisos para visualizar dicha información.

2. DESCRIPCIÓN DEL TRABAJO A REALIZAR

Con base en la información suministrada en el numeral 1, se debe desarrollar un script en Python, que permita realizar de forma automática la valoración de las unidades donadas y determinar si son apta o no para transfusión con base en lo descrito anteriormente.

El usuario debe suministrar cierta información al aplicativo, la cual será almacenada en una base de datos MySQL, dicha información será procesada, consultada y posteriormente presentada en pantalla por medio de uno o varios mensajes con los detalles de la unidad, su viabilidad para usar y la información del donante.

2.1 Para ingresar a la aplicación desarrollada, el algoritmo deberá pedir usuario y contraseña, hasta que estos valores no sean los correctos, el aplicativo no podrá avanzar de ahí.

2.2 La información de usuario y contraseña debe estar almacenada en un archivo llamado ***login.txt*** y guardado en una subcarpeta llamada ***admin***.

2.3 La estructura del archivo debe ser ***clave : valor***. Por ejemplo, los datos para ingresar al aplicativo son usuario: admin y la contraseña: Admin123, esta información se debe almacenar en el archivo como ***admin : Admin123***. Si tiene más usuario, cada renglón será una clave : valor con la información de acceso.

2.4 Deberá usar la gestión de archivos para leer la información de acceso que este contiene y usarla para permitir el acceso de los autorizados.

2.5 Una vez haya pasado el login, la aplicación llevará a un menú que tendrá tres opciones, así:

1. Gestión de usuarios.
2. Gestión de hemocomponentes.
3. Salir

2.6 Para el caso de la primera opción del menú **Gestión de usuario**, deberá llevar a otro menú que permita gestionar usuarios autorizados, así:

1. Insertar nuevo usuario.
2. Mostrar todos los usuarios.
3. Modificar contraseña.
4. Eliminar usuario
5. Volver al menú anterior.

** Al usuario al que se hace referencia en esta sección, es la persona que puede usar la aplicación desarrollada, no confundir con el donante.*

2.7 Para las opciones de modificar o eliminar, se debe pedir el usuario y buscar a través de este parámetro para hacer la edición o borrar según sea el caso.

2.8 Todas las operaciones con archivos; insertar, mostrar, modificar y eliminar información de los usuarios para el login, se debe hacer usando funciones.

2.9 Para cada una de las opciones del menú de gestión de usuarios (2.6) usar todos los recursos vistos en clase para gestionar el archivo ***login.txt*** y hacer todas las opciones que pide el menú.

2.10 Para el caso de la opción del menú 2 **Gestión de hemocomponentes**, este deberá llevar a otro submenú que tenga las siguientes opciones:

1. Ingresar una unidad en forma manual
2. Ingresar unidades sanguíneas desde archivos (automático)
3. Consultar todas las unidades.

4. Consultar la información de una unidad.
 5. Modificar información de una unidad.
 6. Eliminar un registro de unidad.
 7. Volver al menú anterior.
- 2.11 Lo descrito en el numeral anterior es lo que se conoce como un CRUD de una base de datos (BD), para lo cual usarán una BD de MySQL como las vistas en clase.
- 2.12 La base de datos MySQL deberá estar configurada con los siguientes parámetros:
Usuario: informatical
Contraseña: bio123
Nombre de la base de datos: bloodbank
- 2.13 La base de datos deberá contar con mínimo dos tablas, una con la información del donante y la otra con la información de la unidad de sangre. La clave común entre estas dos tablas será un código asignado a cada unidad que se ingrese al sistema. La estructura interna de cada tabla estará configurada según lo considerado por cada equipo.
- 2.14 Menú **Ingresar una unidad en forma manual**: Se debe captura por teclado la información del donante, que son los resultados de las siete pruebas cruzadas obligatorias HCV, HBV, CMV, VIH, HTLV, Sífilis y Chagas y respecto a la información del donante, nombre completo, número de documento de identidad, edad y género. Adicional a lo anterior, se debe pedir el código de la unidad. Este código será alfanumérico y el dato con el que se busque, modifique y/o elimine la información de la unidad sanguínea. La información relacionada anteriormente será guardada en la base de datos.
- 2.15 Adicional a la información que se lista anteriormente y con la cual se deben crear las tablas de la base de datos, el algoritmo debe determinar en forma automática si la unidad es apta o no para ser transfundida (usando estructuras condicionales) y al igual que la información anterior, debe ser almacenada en la base de datos. NOTA: no se puede preguntar si es apto o no, se debe determinar en forma automática con base en la información del numero 1 de este PDF.
- 2.16 Las opciones de POS y NEG para los resultados de las pruebas, se debe hacer con un menú, de tal forma que el usuario solo ingrese 1 o 2, A o B, etc. En caso contrario deberá alertar al usuario y volver a pedir que seleccione una de estas dos opciones. Con el objetivo de ahorra código, esta opción se debe hacer con una función.
- 2.17 Menú **Ingresar unidades sanguíneas desde archivos (automático)**: deberá leer la información de uno o varios archivos en forma simultánea desde una carpeta que se llama *donors*, extraer los mismos datos que se listan en el numeral 2.14 y almacenarlos en la base de datos de MySQL llamada *bloodbank* y de igual manera que en el ingreso manual, se debe determinar la viabilidad del componente con base en la información de las

pruebas. Vaya a los archivos proporcionados como muestra y analice cuál información es la que tiene que extraer.

- 2.18 Para el caso de la opción **3** del menú “**Consultar todas las unidades**”, el algoritmo deberá mostrar todos los registros de las unidades almacenadas en la base de datos con toda la información que le corresponde, es decir, información de las pruebas cruzadas y la información del donante. La información se debe mostrar de forma ordenada.
- 2.19 Si la opción escogida por el usuario es la **4. Consultar la información de una unidad**, el algoritmo deberá pedir el código del componente y devolver la información relacionada a este, es decir, los resultados POS o NEG de las pruebas cruzadas, la información del donante y la viabilidad de este para ser transfundido (APTO o NO APTO), en caso de que el código del componente no esté en la base de datos, el algoritmo deberá decir: “El componente no existe en la base de datos” y volver al menú anterior. TODA la información que se muestra en esta opción se debe mostrar en forma ordenada.
- 2.20 La validación del dato número de edad, se debe hacer utilizando la estructura try/except en una función.
- 2.21 Para la opción del menú **4. Modificar información de una unidad**, se debe pedir el código del componente y solo permitir la actualización de nombre, documento, edad y género del donante, en resto de la información no podrá ser editada.
- 2.22 La información a editar no pueden ser los tres campos editables al mismo tiempo, el usuario deberá escoger que campo va a modificar.
- 2.23 Para el caso del menú **5. Eliminar un registro de unidad**, se debe buscar y eliminar usando el código del componente.
- 2.24 En los dos casos anterior, modificar o eliminar, en caso de no estar el código en la base de datos, el algoritmo deberá decir que el componente que va a eliminar o editar (según sea el caso) no existe en la base de datos y volver al menú anterior.
- 2.25 Las funciones que utilice se deberán importar desde un archivo .py.

3 CONDICIONES DE ENTREGA

- 3.1 El trabajo se realizará en equipos de 3 o 4 integrantes del mismo grupo de clase (OJO, no se aceptan trabajos individuales o en parejas, ni conformados por integrantes de grupos de clase diferentes), los cuales se inscribirán en un archivo que les será compartido en Drive, es decir, ustedes mismos irán seleccionando los grupos y se irán inscribiendo.
- 3.2 Se deben entregar todos los archivos (.ipynb y/o .py) diseñados en Python (funciones y script), el archivo login.txt y el archivo de la base de datos de MySQL (bloodbank.sql), los cuales deben estar comprimidos (.zip, .rar, .7z, etc.) en una carpeta que los contenga y que tenga el nombre del equipo y el grupo de clase. Por ejemplo: equipo1_grupo1.zip o equipo3_grupo2.rar. OJO, la presentación de los trabajos cuenta y por eso les pido el favor de tener cuidado de cómo nombran los archivos, se les rebajará media unidad si envían archivos como final_copia1.py, final_final_esto_si_fue.py ... etc.
- 3.3 Deben asegurarse de entregar correctamente los archivos ya que la revisión se realizará única y exclusivamente con los archivos cargados en el Classroom, en la actividad diseñada para este propósito. NOTA: la entrega del trabajo la deberá hacer uno de los integrantes y en los comentarios en el momento de la entrega, deberá relacionar el nombre de los otros integrantes del equipo de trabajo.
- 3.4 En la cabecera del script se deben colocar los nombres de los autores como un comentario (integrantes del equipo) y el código debe estar debidamente documentado. Descripción de las variables, descripción general de los procesos y resultados a mostrar.
- 3.5 Este trabajo debe ser cargado a través de Classroom a más tardar el día **VIERNES 15 DE NOVIEMBRE DEL 2024 A LAS 11:59 PM. IMPORTANTE: EL TRABAJO NO SE RECIBIRÁ DESPUÉS DE ESTA HORA, ES DECIR A LAS 12:00 AM ESTE 20% SERÁ CERO (0.0)**. NOTA: La actividad creada para este propósito tendrá un temporizador que cerrará el proceso de entrega en esa fecha y hora.
- 3.6 Cualquier intento de fraude o copia de código entre los diferentes equipos implicará una calificación de cero (0.0) para todos los involucrados sin posibilidad de recuperación.
- 3.7 Si al ejecutar el script hay algún error, sin importar cual sea, el trabajo se calificará sobre 3.0.