Interfaz gráfica enfocada a base de datos hospitalaria Algoritmos Computacionales

De La Torre Morales Ma. Fernanda,Martínez Gutiérrez Berenice Anahí, Santos Sánchez Betsabe

UNAM
Facultad de Ciencias
Prof. Pedro Porras Flores

5 de junio de 2019

Resumen

Se diseñó una base de datos enfocada en un hospital con el objetivo de administrar, actualizar y guardar la información de los pacientes efectivamente y convenientemente para los distintos funcionarios en este establecimiento mediante los lenguajes Python y SQL. Posteriormente, se diseñó una interfaz gráfica con la implementación de la paquetería Tkinder enfocada a la programación orientada a objetos.

Figura 1: Procesos necesarios para una App,software o página de internet.

1. Introducción

La información hoy en día ya no puede ser administrada manualmente debido a la constante actualización de datos y tecnología; es por ello que existe una necesidad de crear algoritmos computacionales para computarizar y automatizar la información para un almacenamiento adecuado y fácil acceso. Para ello, existen las bases de datos; que pueden ser definidas com un almacén en el que se puede guardar una gran cantidad de información para su posterior consulta y utilización. Con ayuda de programas de aplicación, como las interfaces, las bases de datos pueden ayudar a la recaudación de información adecuada y efectiva, sin la necesidad para un usuario externo de tener conocimiento en programación. Tomando en cuenta lo anterior, se propuso desarollar un sistema que permitiera administrar datos de una clínica, y donde se le permite a los usuarios realizar diferentes actividades según su rol en la clínica (paciente, secretaria o doctor). Todo esto basandonos en el lenguaje de progrmación Python debido a que es un lenguaje multiparadigma de código abierto, relativamente fácil de utilizar y con el cual estabamos familiarizadas.

En lo que respecta a este proyecto, fueron necesarios los conocimientos básicos en programación en SQL, por ello, a continuación se explicarán algunos conceptos.

SQL significa: "lenguaje de consulta estructurado".Es un lenguaje estándar para acceder y manipular bases de datos, es decir, ejecutar consultas,recuperar datos,insertar registros,actualizar registros,eliminar registros, crear nuevas bases de datos, tablas,procedimientos almacenados, vistas en una base de datos, establecer permisos en tablas y procedimientos. Existen varias versiones de SQL, empero,todos admiten al menos los comandos principales (como SELECT, UPDATE, DELATE, INSERT, WHERE) de una manera similar.

RDBMS, - Sistema de Gestión de Base de Datos Relacional-, es la base de SQL y de todos los sistemas de bases de datos modernos, como MS SQL Server, IBM DB2, Oracle, MySQL y Microsoft Access. Los datos en RDBMS se almacenan en objetos de base de datos llamados tablas. Una tabla es una colección de entradas de datos relacionados y consta de columnas y filas, cada tabla se divide en entidades más pequeñas llamadas campos. Un campo una columna en una tabla que está diseñada para mantener información específica sobre cada registro en la tabla. Un registro, también llamado fila, es cada entrada individual que existe en una tabla.

La sintaxis de SQL requiere comillas simples alrededor de valores de texto (la mayoría de los sistemas de bases de datos también permitirán dobles comillas). Sin embargo, los campos numéricos no deben estar entre comillas, por otro lado, los comandos pueden estar escritos en minúsculas o mayúsculas y se ejecutan de la misma manera. Un ejemplo del empleo de un comando

es el siguiente con el comando WHERE que se puede combinar con los operadores booleanos AND, OR y NOT. Los operadores AND y OR se utilizan para filtrar registros según más de una condición. El operador AND muestra un registro si todas las condiciones separadas por AND son verdaderas. El operador OR muestra un registro si alguna de las condiciones separadas por OR es verdadera. El operador NOT muestra un registro si las condiciones no son verdaderas.

Un campo en el que vale la pena enfatizar, es aquel con un valor NULL, que es un campo sin valor. Si un campo en una tabla es opcional, es posible insertar un nuevo registro o actualizar un registro sin agregar un valor a este campo. Luego, el campo se guardará con un valor NULL, el cual es diferente de un valor cero o un campo que contiene espacios. Un campo con un valor NULL es uno que se ha dejado en blanco durante la creación del registro. Existen campos que deben contener valores únicos y no pueden contener valores NULL, estos son los que incluyen la restricción PRIMARY KEY. Una tabla solo puede tener una clave primaria; y en la tabla, esta clave principal puede constar de una o varias columnas (campos). Una clave que se utiliza para vincular dos tablas es la restricción FOREIGN KEY, un campo (o colección de campos) en una tabla que se refiere a la clave primaria en otra tabla. La tabla que contiene la clave externa se denomina tabla secundaria y la tabla que contiene la clave candidata se denomina tabla de referencia o principal.

Algunas veces existe una confusión entre SQL y MySQL. Sin embargo, como se mencionó anteriormente, MySQL es un RDBMS para almacenar, modificar y administrar una base de datos utilizando SQL.Es un software gratuito que se puede instalar y descargar desde su página oficial que ofrece también soporte de conectores con la herramienta MySQL Workbench para diseñar y desarrollar bases de datos. Una ventaja que ofrece este software, es el almacenamiento, manejo y modificación de datos de forma tabular por medio de Workbench, que son más sencillas de trabajar y cuenta con actualizaciones constantes,que en contraste con SQL, como es un lenguaje fijo, los comandos siguen siendo los mismos.

Un conector útil para el manejo de los datos con diversas paqueterías como Pandas de Python, es MySQL Connector / Python que permite que los programas de Python accedan a las bases de datos de MySQL, utilizando una API que cumple con la especificación de la API de la base de datos de Python v2.0 (PEP 249). Está escrito en Python puro y no tiene ninguna dependencia, excepto la biblioteca estándar de Python.

En conjunto con MySQL Workbench sería ideal un diseño de una interfaz gráfica la cuál facilitara lo más posible el manejo de los datos para una variedad de usuarios como doctores, secretarias y administradores. Para ello, como una primera aproximación, se utilizó el módulo Tkinter de Python que tiene como base la programación orientada a objetos; que se define como un paradigma de programación que proporciona un medio de estructuración de programas para que las propiedades y comportamientos se empaqueten en objetos individuales. Por ejemplo, un objeto podría representar a una persona con una propiedad de nombre, edad, dirección, etc., con comportamientos como caminar, hablar, respirar y correr. Dicho de otra manera, la programación orientada a objetos es un enfoque para modelar cosas del mundo real como los coches, así como las relaciones entre cosas como empresas y empleados, estudiantes y profesores, etc.

Otro paradigma de programación, es la programación procesal, que estructura un programa como una receta, ya que proporciona un conjunto de pasos, en forma de funciones y bloques de código, que fluyen secuencialmente para completar una tarea.

A diferencia de la programación procesal, en la programación orientada en objetos, los objetos están en el centro del paradigma de programación, no solo representando los datos, como en la programación procesal, sino también en la estructura general del programa. Python es un lenguaje de programación multiparadigma, puede elegir el paradigma que mejor se adapte al problema en cuestión, mezclar diferentes paradigmas en un programa y cambiar de un paradigma a otro a medida que su programa evoluciona.

En la programación orientada a objetos, cada objeto es una instancia de alguna clase. Las estructuras de datos como cadenas y listas están diseñadas para representar cosas con una característica en común, como nombres de pacientes.

Supongamos que se quiere rastrear varios pacientes diferentes. Si utilizó una lista, el primer elemento podría ser el nombre del paciente, mientras que el segundo elemento podría representar su edad. Este método conlleva a una serie de problemas, aunque no sean notorios a primera instancia, el primero, ¿Realmente se está respetando ese orden?, segundo, ¿Es fácil el proceso para una cantidad de 150 pacientes?, tercero, ¿Cómo añadir una propiedad extra como fecha de nacimiento en la lista? Esto carece de organización, y es la necesidad exacta de clases.

Las clases se utilizan para crear nuevas estructuras de datos definidas por el usuario que contengan información arbitraria sobre algo. En el caso de un paciente, podríamos crear una clase paciente () para rastrear propiedades sobre el paciente como el nombre y la edad.

Es importante tener en cuenta que una clase solo proporciona estructura, es un proyecto de cómo se debe definir algo, pero en realidad no proporciona ningún contenido real. La clase paciente () puede especificar que el nombre y la edad son necesarios para definir un paciente, pero en realidad no va a indicar el nombre o la edad de un animal específico. Para ello se crean las instancias, que son los objetos que perenecen a una clase y tienen valores reales. En sí, la clase es como un cuestionario que debe de llenar el usuario para poder crear un objeto, primero está la creación de una clase y luego la de un objeto.

Las características de un objeto, llevan por nombre:.ªtributo", existen aquellos atributos que pueden pertenecer a toda una clase y aquellos que pertenecen a un objeto en específico. Para comenzar con los atributos de istancia, se emplea el método: ïnit",

brindándole un estado inicial al objeto. Al menos debe llevar una variable, ésta puede ser una variable self que se refiere al objeto en sí mismo, cabe destacar que siempre es el primer argumento del método mencionado anteriormente. Para poder operar con los atributos de los objetos, se crean los métodos de instancia, que pertenecen a la misma clase, pueden ser funciones que le den una particularidad al objeto que se está trabajando. Finalmente, en este posicionamiento entre clase, método y atributo, se puede decir que todos atributos que forman a una clase, los puede heredar otra, anulándolos, modificándolos o incluso extendiendo la funcionalidad de la clase anterior la cúal se llama primaria y la que hereda,secundaria.

Como se hizo mención, la paquetería Tkinter tiene como base la programación orientada a objetos. Es la interfaz estándar de Python y tiene disponible distintos elementos gráficos estándar para el usuario, llamados wigets, que permiten que se pueda crear un primer bosquejo de una aplicación a través del mismo.

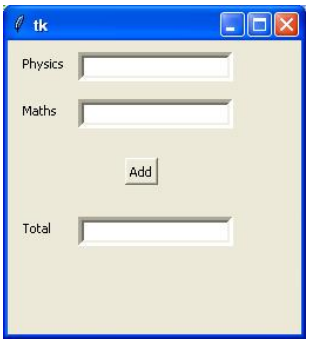

Figura 2: Ventana diseñada con Tkinter.

2. Metodología

Se estableció una secuencia de pasos para la realización del proyecto; dichos pasos serán descritos con detalle en las siguientes secciones:

2.1. Identificar las prioridades/características del sistema de administración de una clínica.

Fue necesario identificar las prioridades y características que debía tener nuestro sistema para administrar una clínica, asi, a partir de ellas poder organizar el trabajo. Las puntos identificados fueron:

- 1. Actualización de la base de datos en tiempo real.
- 2. Agilizar la consulta de datos.
- 3. Generar un historial de los pacientes,usuarios y citas.
- 4. Que el sistema sea fácil y eficaz para cualquier usuario.

2.2. Actualización de la base de datos.

Una parte fundamental en nuestro proyecto fue el asegurarnos que a traves de la interfaz gráfica pudieramos tener una actualización en nuestra base de datos automaticamente. En términos generales requeriamos utilizar un Sitema Gestor de Base de Datos, el cual se define como un sistema que nos permite almacenar, modificar y extraer información de una base de datos.

Este sistema, en nuestro caso, ademas contribuiría a facilitar y restringir el acceso a los usuarios tomando en cuenta el rol que desempeñaran en la clínica, lo cual nos permitiria controlar en cierta medida la seguridad de la información; lo cual, en todos los programas y aplicaciones de la actualidad es de suma importancia.

2.3. Realizar una base de datos con MYSQL.

Para la creación de la base de datos que almacenaría toda la información de la supuesta clínica, se decidió utilizar MYSQL, pues es un sistema de gestión de bases de datos relacional, considerada como la base datos de código abierto más popular del mundo. En cuanto a capacidad de almacenamiento, en MySQL 5.0, usando el motor de almacenamiento MyISAM , el máximo tamaño de las tablas es de 65536 terabytes. MySQL fue escrito en C y C++ y destaca por su gran adaptación a diferentes entornos de desarrollo, permitiendo su interactuación con los lenguajes de programación más utilizados como PHP, Perl, Java y Python.

Para nuestro proyecto, se creó una base de datos (hospitalprueba1), la cual consistía de cuatro tablas: Citas, Diab,Pacientes y Usuarios. La primera nos permitia tener un registro de las citas, la segunda un registro sobre la presencia de enfermedades como Diabetes e Hipertensión, la tercera tabla nos permitia guardar la información básica de cada paciente y la cuarta tabla nos servía para guardar los Usuarios y su tipo de rol, los cuales se usaría para ingresar en la interfaz gráfica.Dichas tablas podían ser visualizadas de manera gráfica (Figura 3.) a través de un programa que ofrece MYSQL llamado MYSQL Workbench.

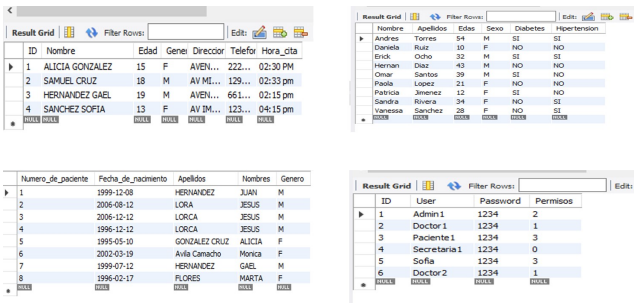

Figura 3: Tablas de la base de datos creada.

2.4. Backend.

Como nuestro propósito es crear una interfaz gráfica, era necesario primero desarrollar todo el código que

permitiera un correcto funcionamiento de la interfaz. A esto se le llama Backend.

El Backend es la capa de acceso a datos de un software (en nuestro caso nuestra interfaz) , que no es directamente accesible por los usuarios comunes, además contiene la lógica de la aplicación que maneja dichos datos. El Backend también accede al servidor; en este caso, se conecta con la base de datos de forma local.

Toda esta parte fue desarrollada con el lenguaje de Python, sin embargo, como se mencionó con anterioridad, para la manipulación de una base de datos se requiere usar el lenguaje SQL, por lo que fue necesario establecer una conexión entre MYSQL y Python; para ello se utilizó un controlador llamado mysql.connector. Una vez realizada la relación, pudimos utilizar los cursores, que son una herramienta que nos permite conectarnos y realizar operaciones con la base de datos. Nos permite realizar consultas, además de poder mostrar, insertar, actualizar y borrar datos en las diferentes tablas.

2.5. Ocupar la paquetería Pandas para analizar datos y manipulación de datos.

Pandas es una librería de Python para el analisis de datos que permite usar y/o crear tablas de datos (DataFrame) las cuales pueden ser manipuladas es decir, se pueden incrementar columnas o filas. Además permite corroborar que los datos esten completos asi evitando que la base de datos evitando inconvenientes en una clinica. Pandas nos permite seleccionar filas y con ellas se pueden realizar graficas, esto permitira al personal de aministracion generar estadisticas.

2.6. Dar diseño a las ventanas para el sistema de administración clínica.

En esta sección, nos plamteamos como queríamos que se desplegaran las ventanas tomando en consideración el rol que desempeñaba cada usuario. La idea general se puede ver plasmada en la Figura 4 Cabe mencionar que en la base de datos, a cada Usuario se le asociaba un número y gracias a este podiamos determinar que rol tenía (0 si era secretaria, 1 Doctor, 2 Admin y 3 Paciente)

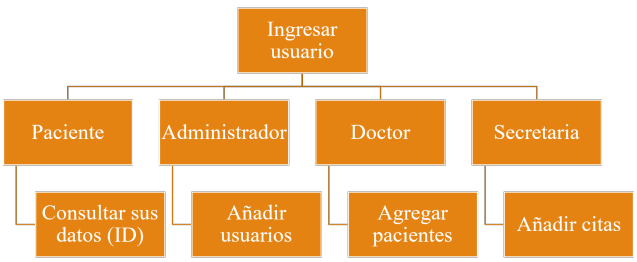

Figura 4: Diagrama de organización de las ventanas de la interfaz.

2.7. Interfaz gráfica

Para el diseño del frontend, se utilizó la paquetería Tkinter; el código se dividió en tres scripts los cuales son secuenciales. El primero ayuda a la conexión entre la base de datos en MySQL y Python mediante el conector mysql.connector. Hacer dicho código para la conexión con la base de datos fue primordial para que las clases secundarias heredaran dicha conexión y no se preocupara por la creación igualmente en Python cada vez que se ejecutaba un comando parecido al de MySQL, esto se puede encontrar en el archivo *conexion.py* en el repositorio indicado en las referencias.

Posteriormente, se propusó trabajar con cinco funciones base para la clase de manejo de datos que puede tener cada uno de los usuarios, como insertar, eliminar, modificar, buscar y poder crear usuarios con diversos permisos comandos que a su vez se encuentran en el lenguaje SQL, todos los argumentos se refieren a los tipos de datos que se insertan en cada uno de los tablas, así como la variable self que se definió en la introducción, estos fueron en realidad, los métodos de atributos [Véase *pr1.py* el cual contiene todas las funciones que cualquier usuario de la interfaz puede realizar].

Finalmente en el último script, se definieron a través de varios widgets, una nueva clase con subclases dentro de la misma que iban a contener el tipo de vetana que se le va a mostrar al usuario dependiendo su posición; en esta parte del proyecto la presentación fue el enfoque principal, los widgets más empleados fueron:

- Entry: Dio una entrada de texto.
- Button:Se empleó para ejecutar un comando o una operación
- Label:Mostró una etiqueta.
- LabelFrame: Dibujó un borde y un título.
- Message: Mostró un mensaje en que la misma ventana se ajusta.

Cabe mencionar que Tkinter es una aplicación que pasa la mayor parte del tiempo en un bucle de eventos, ingresado a través del método mainloop. Los eventos en la interfaz provenían de varias fuentes, como pulsaciones de teclas en la búsqueda de citas y operaciones del puntador por parte del usuario tal cuál como oprimir un botón del mismo, toda esta parte visual se puede consultar en *finalprog.py*.

3. Resultados

Se diseñó una interfaz gráfica con diferentes ventanas (Figura 5) que le permiten a usuarios de diversos roles (doctores, paciente, secretaria o administrador) manipular datos de una base de datos especifica, de una forma fácil y práctica, según los permisos que tenga otorgados.

Como se especificó en la sección 2.6,logramos implementar un programa en el cual al validar un Usuario el sistema reconocia su rol, y, por ejemplo, si era un

Paciente, entonces se le mandaba a una ventana especial para que pudiera consultar sus datos, por otro lado, si se trataba de un doctor, entonces se le dirigía a una ventana donde podía agregar nuevos pacientes. De esta manera se evitaba el uso de ciertas funciones a los usuarios, manteniendo un control en el acceso y de esta manera, asegurando infromación, tanto por confidencialidad hacia los pacientes como para evitar un desajuste en la base de datos original.

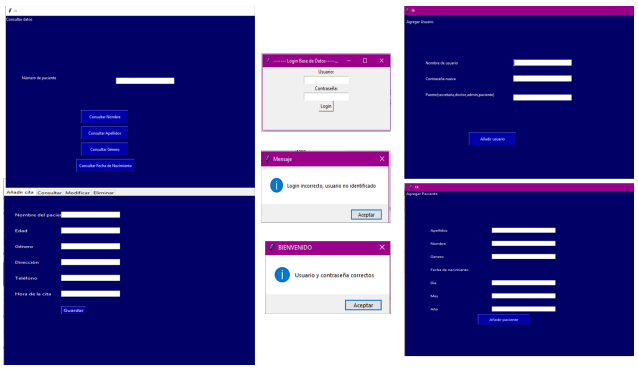

Figura 5: Ventanas de la interfaz.

Otra parte importante en nuestros resultados fue el uso de Pandas para poder analizar los datos almacenados en la base datos. En esta parte logramos conectar MYSQL con Pandas, de tal manera que las tablas de la base de datos se convirtieran en DataFrames y con eso, pode desde agragar nuevas columnas hasta poder interpretar datos a través de herremientas como gráficos que reflejaran la población de enfermos con diabetes del genero masculino y femenino. Por falta de tiempo, esta parte solo quedó desarrollada en un Notebook.

4. Conclusiones

- Se logró diseñar un sistema con Python para administrar una clinica.
- Se crearon usuarios según su rol en la clinica como lo es: doctor, paciente, secretaria o administrador.
- Se otorgaron permiso para que cada usuario manipule datos segun su rol.
- La base de datos se va actualizando en tiempo real.
- El sistema diseñado con Python es de fácil uso.
- Se utilizó Pandas para un breve análisis de la información de una tabla de la base de datos.

5. Trabajo a futuro

- Se pretende probar el sistema de administracion para una clinica en un servidor con el fin de que las personas que no cuentan con Python en sus computadoras lo puedan ocupar.
- Continuar desarollando el sistema de administracion con el fin de crear más ventanas para las necesidades de una clinica .
- En una de las ventanas de la interfaz grafica se pretende anexar documentos que el doctor pueda generar como son recetas medicas o resultados de analisis clinicos.
- Incrementar el uso de pandas para que el personal de administracion pueda generar estadisticas y graficos, que despues puedan ser incorporados a las ventas de la interfaz grafica.

6. Referencias

- Todo el material usado en este proyecto puede ser encontrado en el repositorio de Github: ⟨https://github.com/Berenice08/ProyectoAlgoritmos.git⟩
- Video de demostración de nuestro programa: ⟨https://www.youtube.com/watch?v=CcKT4lQvnU4⟩
- KLINE, K., GOULD, L. and ZANEVSKY, A. (1999). *Transact-SQL programming* Beijing [etc.]: O'Reilly.
- CUEVAS ÁLVAREZ, A. (2016). *Python 3.* Paracuellos del Jarama, Madrid: Ra-Ma.
- GRUS, J. (2019). *Data science from scratch.* [Place of publication not identified]: O'REILLY MEDIA.