

Manual Técnico del Modelo de Detección de Datos Duplicados.

**QUITO – ECUADOR | 2023**

**AUTOR:**  Ing. María Doménica Gómez Sánchez

Ing. Darío Hernán Herrera Herrera

Ing. Cristina Elizabeth Pavón Domínguez

Ing. Jacqueline Vanessa Quispe Tirado

**TUTOR:** MSc. José Luis Pérez Galán

DISEÑO DE UN MODELO DE DETECCIÓN DE DATOS DUPLICADOS MEDIANTE PROCESAMIENTO DEL LENGUAJE NATURAL PARA OPTIMIZAR LA EFICIENCIA EN LA GESTIÓN DE DATOS EN UN LABORATORIO FARMACÉUTICO.

**QUITO – ECUADOR | 2023**

CONTENIDO

[CONTENIDO 2](#_Toc151149720)

[INTRODUCCIÓN 3](#_Toc151149721)

[OBJETIVO 3](#_Toc151149722)

[GLOSARIO DE TÉRMINOS 3](#_Toc151149723)

[ARQUITECTURA DEL PROYECTO 4](#_Toc151149724)

[FLUJO DEL PROCESO DEL PROYECTO 4](#_Toc151149725)

[DIAGRAMAS UML 6](#_Toc151149726)

[ARCHIVO DE RECOPILACIÓN DE DATOS 7](#_Toc151149727)

[PROGRAMACIÓN DEL PROYECTO 15](#_Toc151149728)

[DIAGRAMA ENTIDAD-RELACIÓN 32](#_Toc151149729)

[DICCIONARIO DE DATOS 33](#_Toc151149730)

[ESTRUCTURA BASE DE DATOS RELACIONAL 34](#_Toc151149731)

INTRODUCCIÓN

El presente documento es una guía para dar a conocer el funcionamiento del Algoritmo y del modelo utilizado para la detección de datos duplicados en un laboratorio farmacéutico, para lo cual se empleará diagramas UML como diagramas de casos de uso. También se describirá el código utilizado para el desarrollo del modelo. Adicional se describe la base de datos utilizada, así como también su diccionario de datos correspondiente.

OBJETIVO

Proporcionar al usuario una guía para que conozca el funcionamiento del modelo de detección de datos duplicados en un laboratorio farmacéutico.

GLOSARIO DE TÉRMINOS

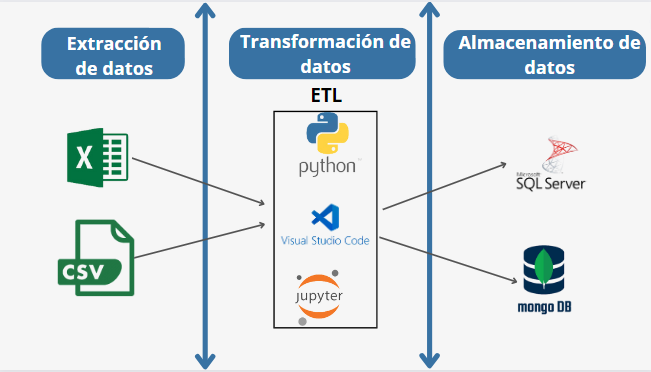
**Base de datos**: sistema donde se organiza información de manera estructurada, lista para ser leída, actualizada o eliminada según las exigencias de los usuarios.

**Diagrama**: es una representación gráfica que utiliza formas geométricas para expresar una proposición, resolver un problema o visualizar las relaciones entre diversas partes o elementos de un conjunto o sistema.

**UML**: Unified Modeling Language, es un lenguaje gráfico empleado para describir, representar, construir y documentar los componentes de un sistema.

ARQUITECTURA DEL PROYECTO

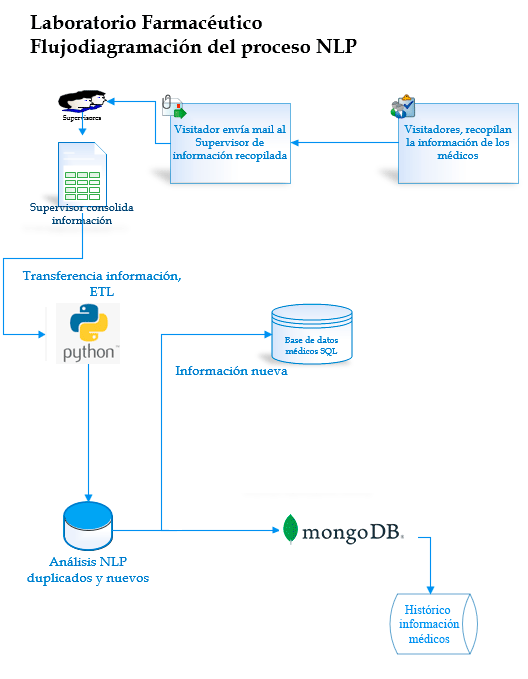
**Figura 1***Arquitectura del proyecto*



FLUJO DEL PROCESO DEL PROYECTO

Toda la información consolidada por los supervisores en el archivo Excel será cargada a un dataframe, luego de un pre procesamiento de la data contenida en el dataframe el modelo NLP leerá la data y realizará una comparación con la información que contiene la base de datos relacional, evitando así la duplicidad de la información e ingresos erróneos de nuevos médicos en el sistema.

**Figura 2***Diagrama de Flujo del proceso*



* Visitadores recopilan la información referente a los médicos
* Visitadores envían la información recopilada a los supervisores
* Los supervisores consolidan la información de sus visitadores mediante el documento Excel programado que se incorpora como parte de este proyecto para control de calidad de información.
* Se realiza la subida del archivo Excel a un dataframe.
* Se ejecuta el modelo NLP para contrastar la información del dataframe con la base de datos transaccional en cuanto a los nombres del médico.
* Inserción a la base de datos SQL de los médicos nuevos detectados mediante el proceso de NLP.
* Paso de información de los resultados NLP a las tablas de la base de datos NoSQL.

DIAGRAMAS UML

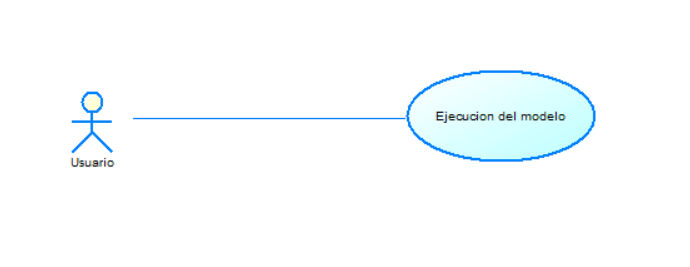
A continuación, se presentan algunos diagramas UML que facilitaran de una manera gráfica el entendimiento y comprensión del funcionamiento del modelo de detección de datos duplicados en un laboratorio farmacéutico.

***Diagramas de Casos de Uso***

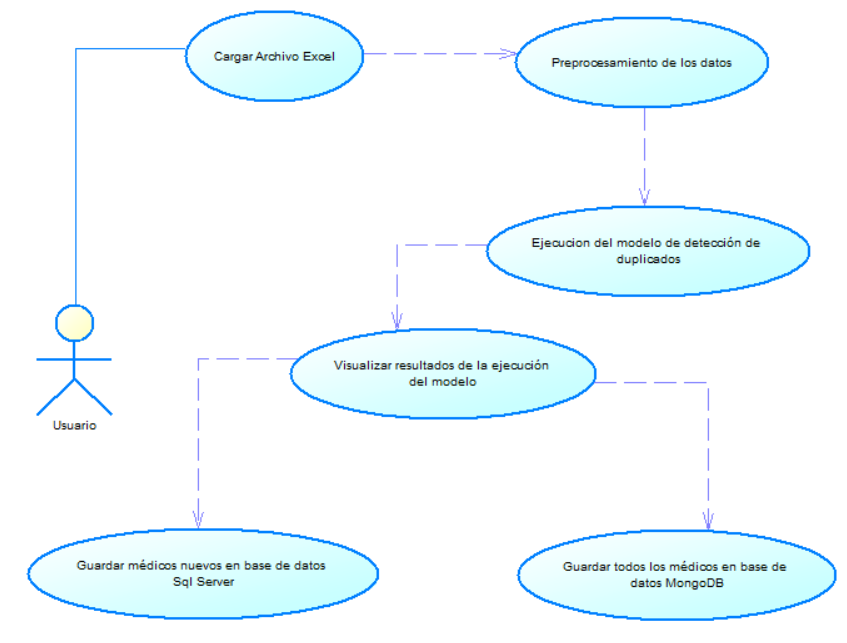
**Identificación de actores**: estos son los actores que participarían en el modelo.

**Usuario**: persona encargada de ejecutar el modelo.

**Figura 3***Diagrama General de los Casos de Uso del modelo de detección de datos duplicados en un laboratorio farmacéutico.*



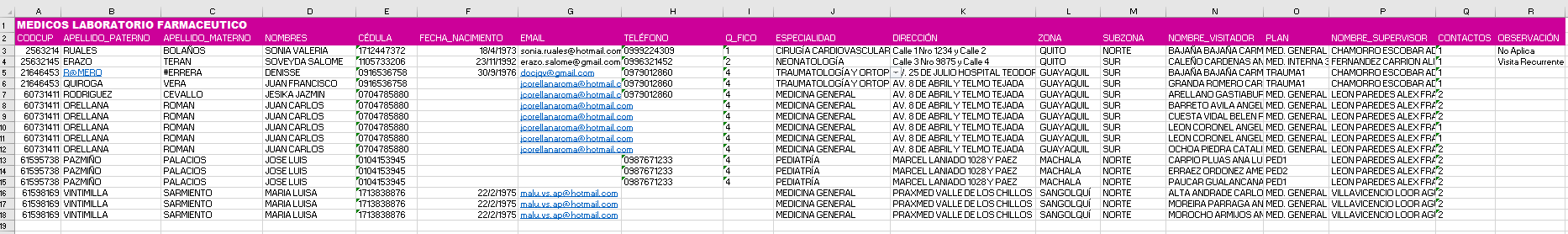
**Figura 4***Diagrama del Caso de Uso de la ejecución del modelo de detección de datos duplicados en un laboratorio farmacéutico.*



ARCHIVO DE RECOPILACIÓN DE DATOS

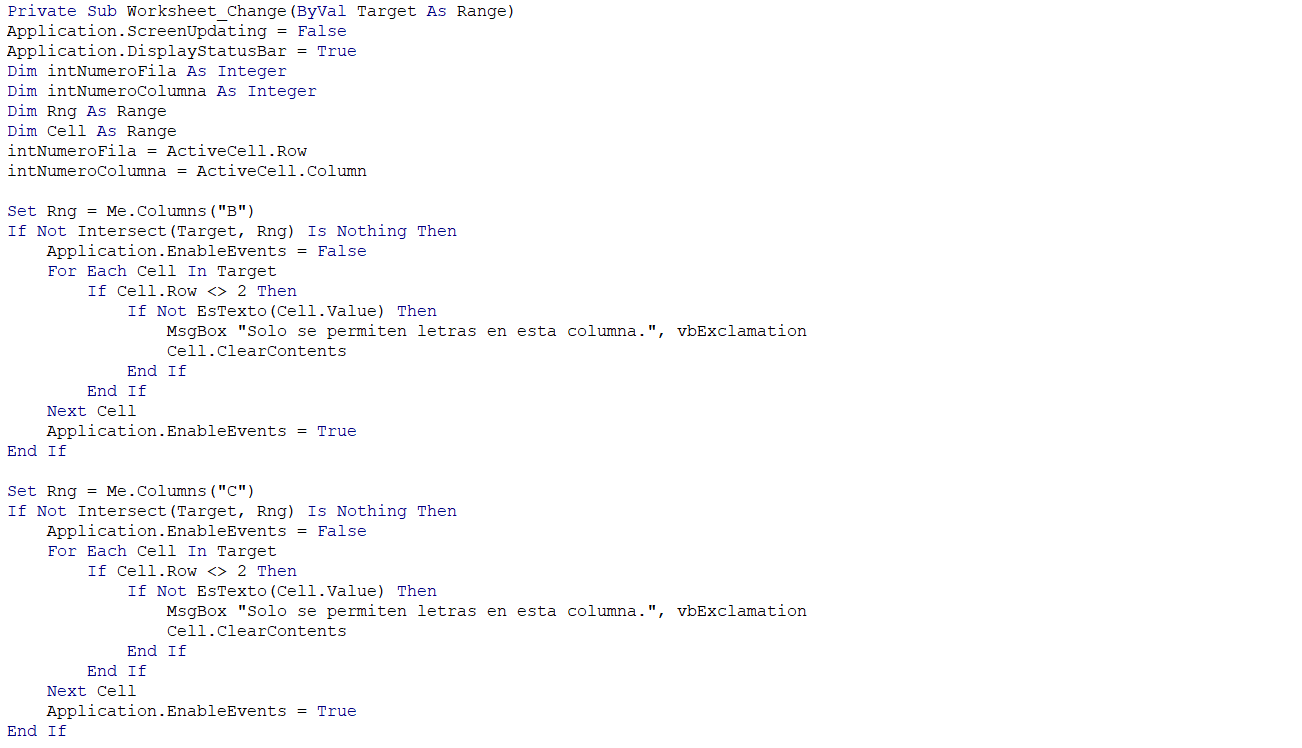
En este proyecto se implementó un archivo Excel para la fase de recopilación de datos. Para reducir los errores de digitación en cuanto a la información de las columnas que manejan catálogos, se creó una macro programada en Visual Basic 6.0, la cual nos permite tener combo box en cada celda de cada una de las siguientes columnas catálogos: Especialidad, Zona, Subzona, Visitador, Planes, Supervisor.

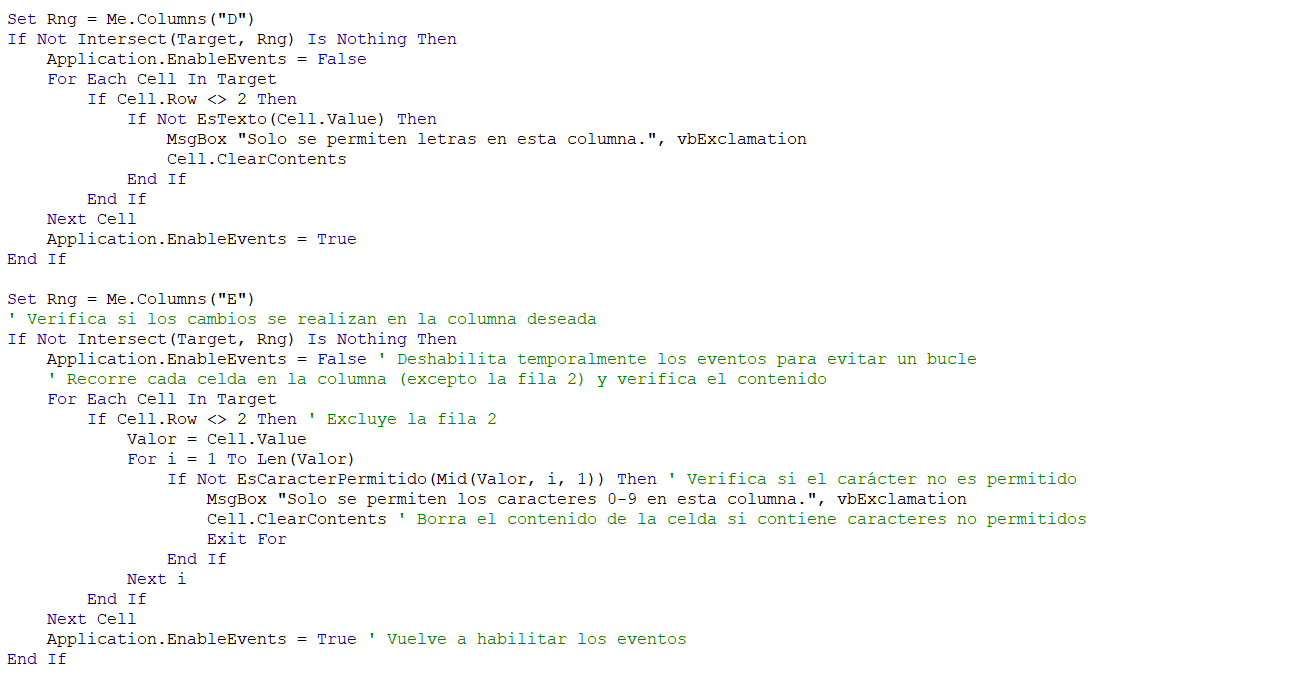
**Figura 5***Archivo de recopilación de datos*



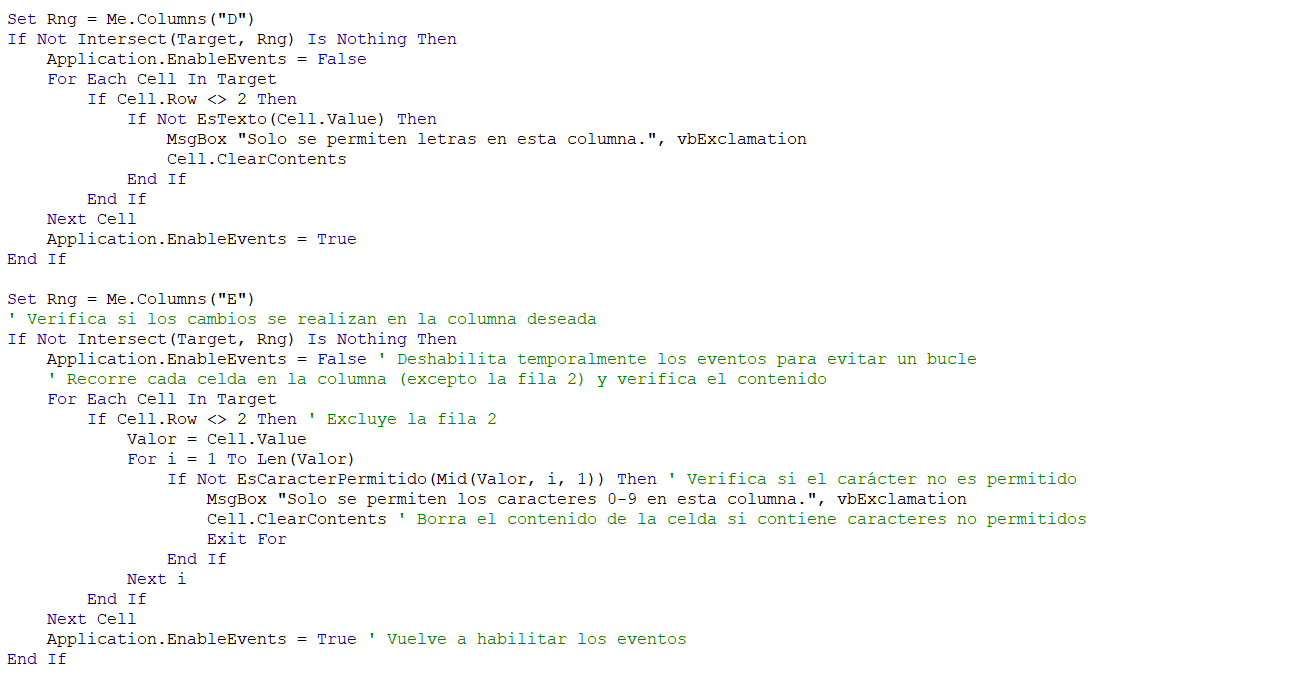
Las funciones y métodos programados en la macro son los siguientes:

**Figura 6***Código Visual Basic con la validación para ingresar en las columnas apellidos y nombres solo caracteres alfabéticos.*

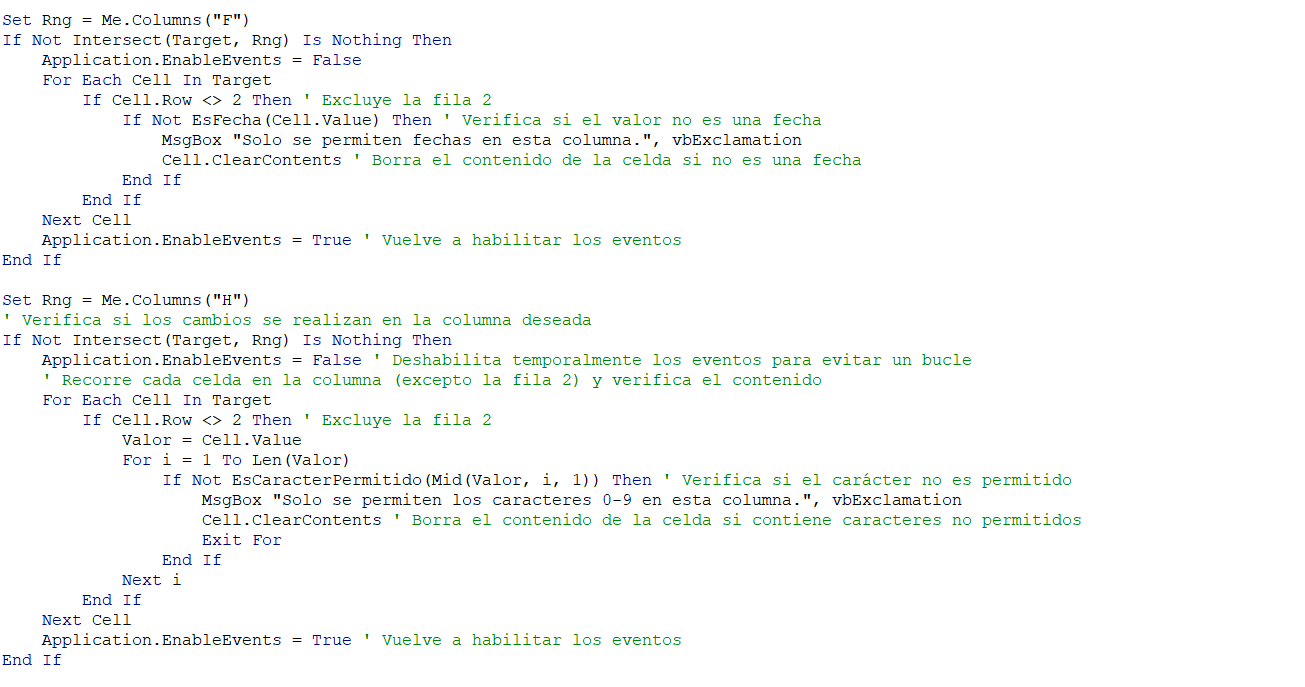




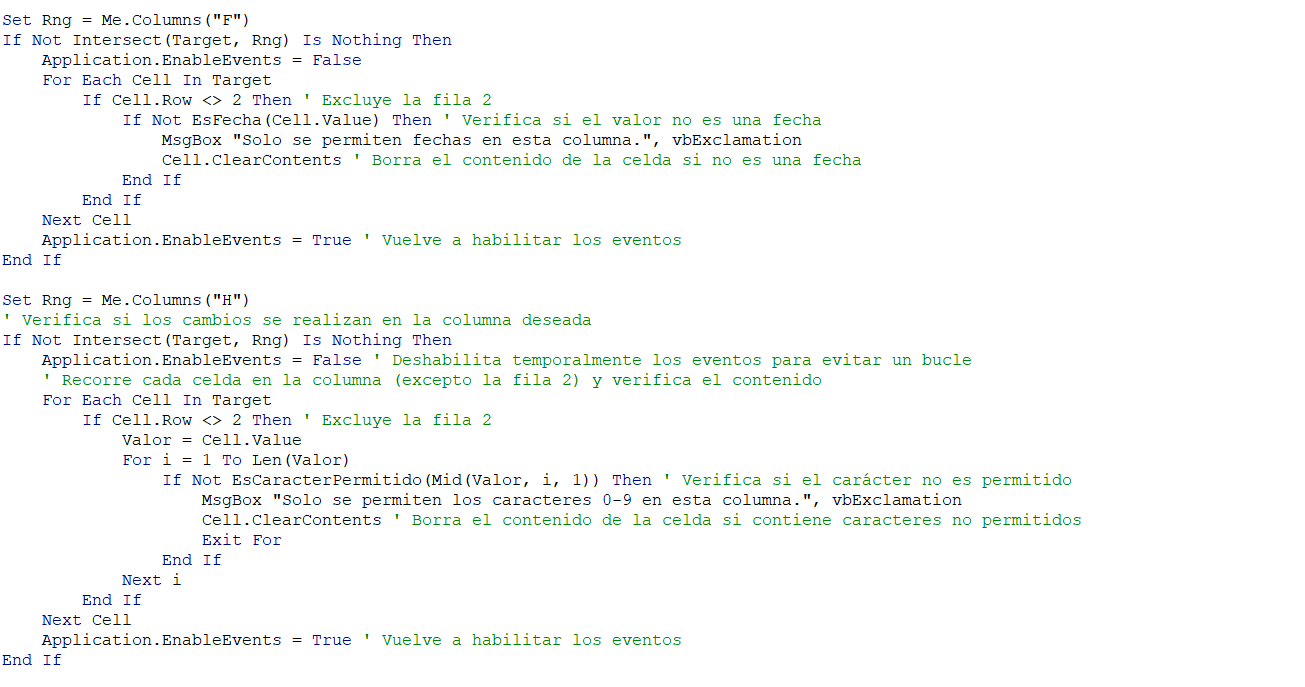
**Figura 7***Código Visual Basic para validar que en la columna cédula se ingrese únicamente caracteres del 0 – 9.*



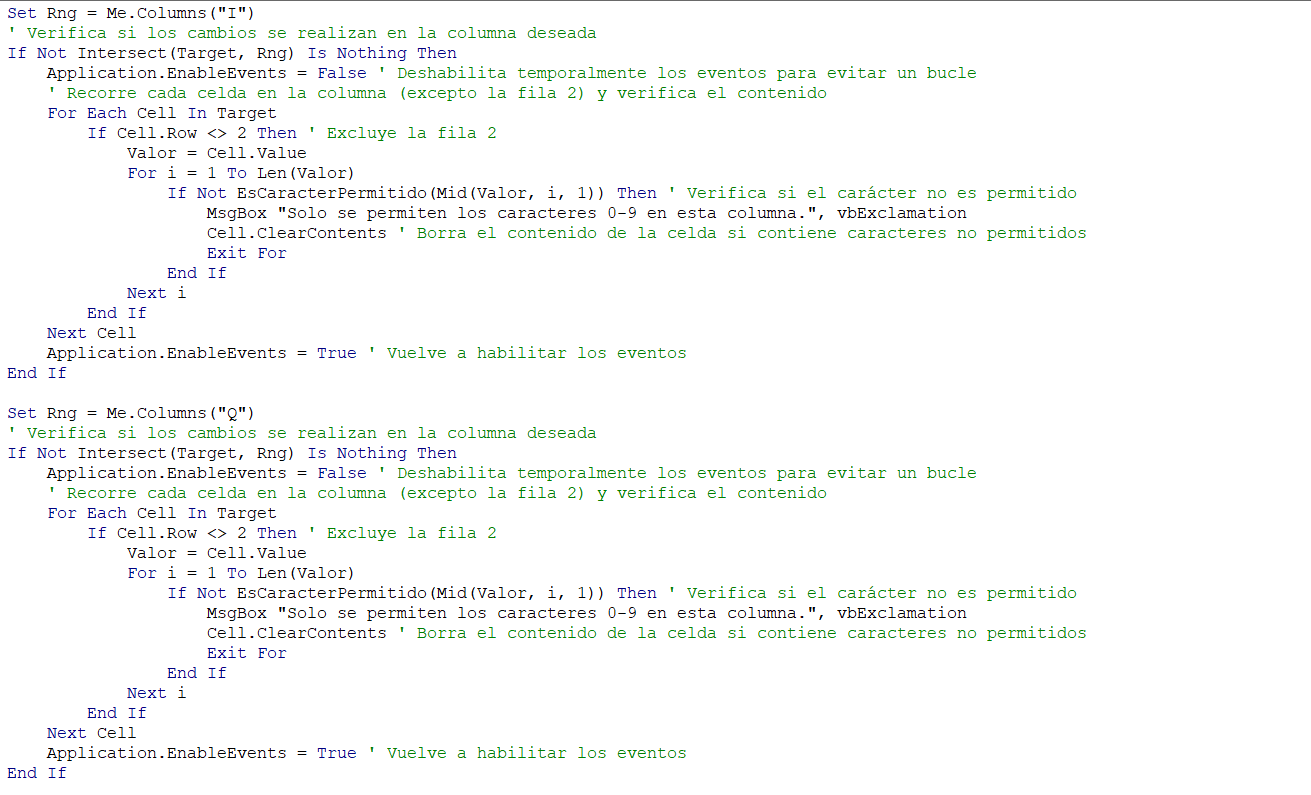
**Figura 8***Código Visual Basic para validar que en la columna fecha de nacimiento la información ingresada sean fechas validas.*



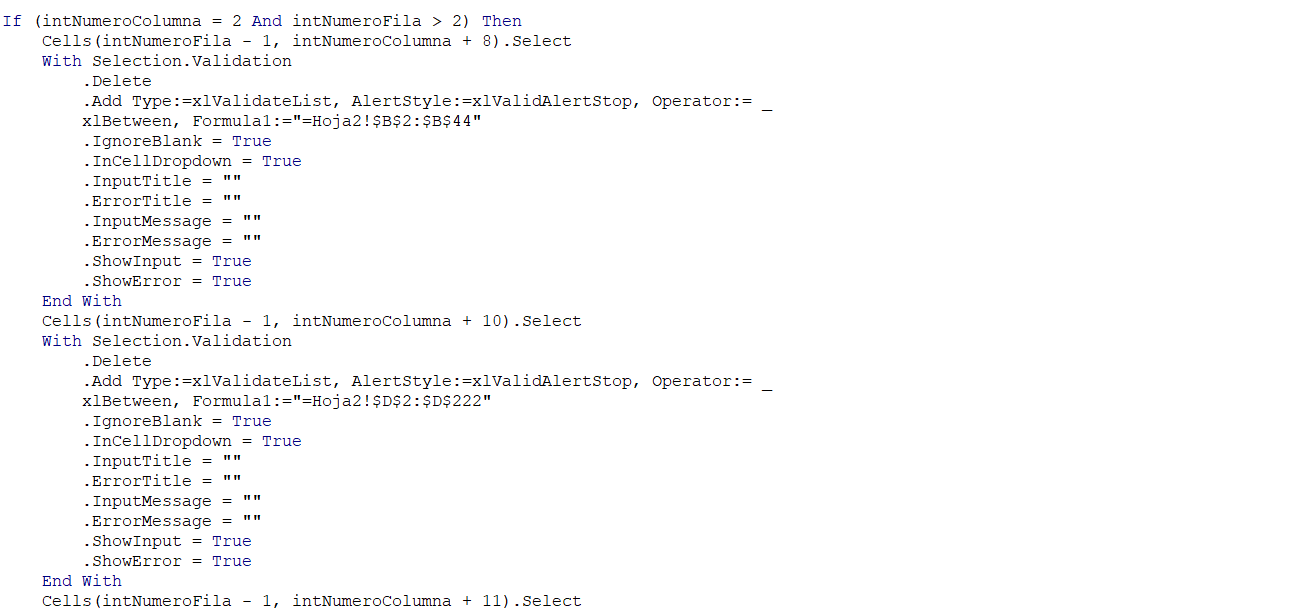
**Figura 9***Código Visual Basic para validar que en la columna celular se ingresen únicamente caracteres del 0 – 9.*



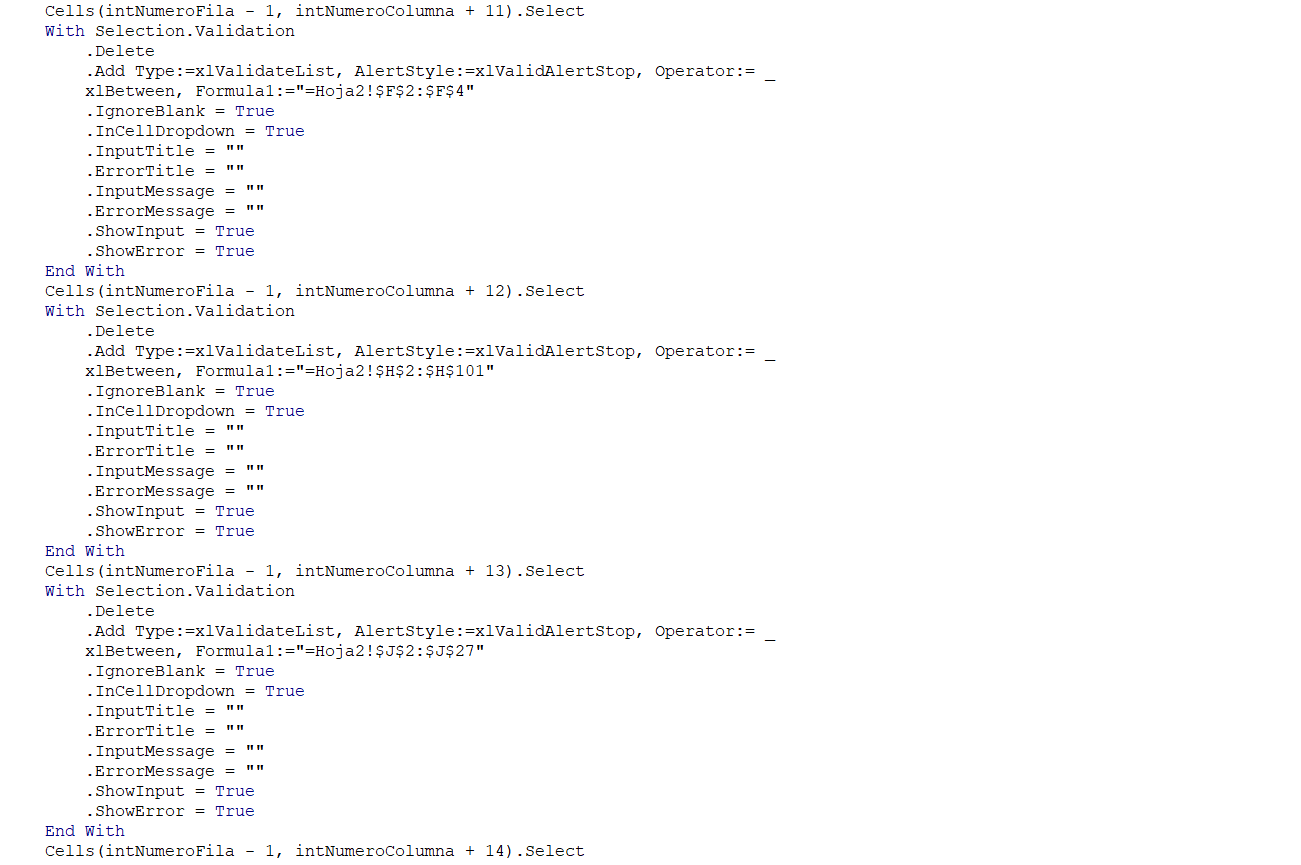
**Figura 10***Código Visual Basic para validar que en las columnas categoría y contactos solo se ingresen caracteres del 0 – 9.*

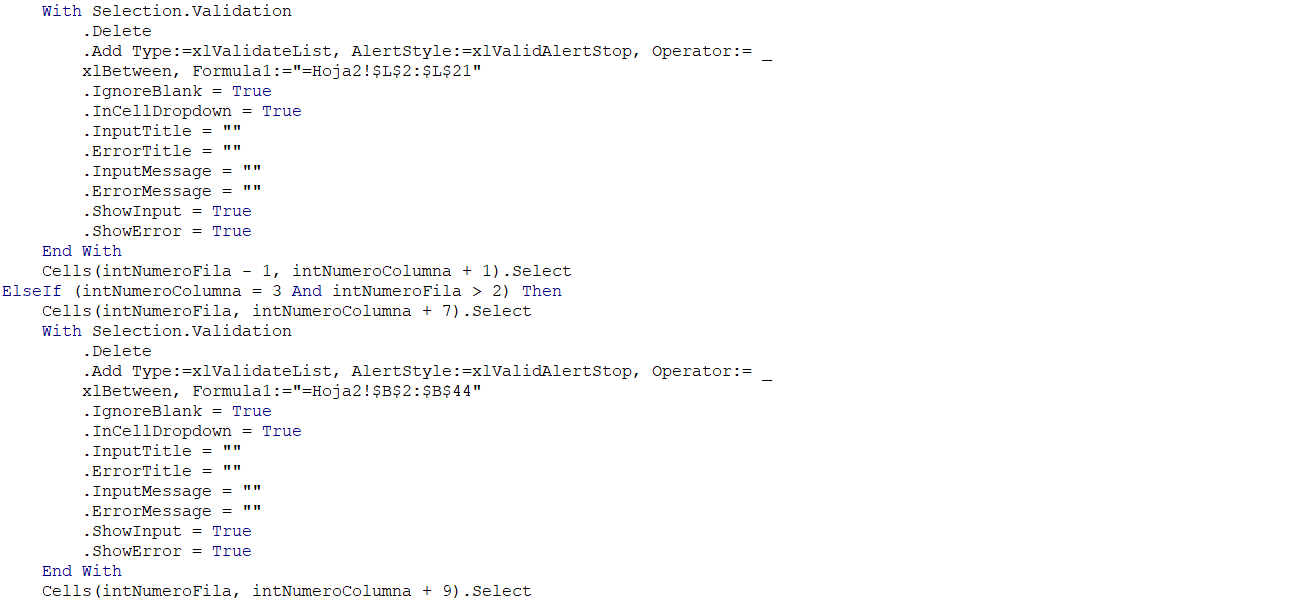


**Figura 11***Código Visual Basic para presentar en las celdas de la columna especialidad mediante un combo box.*

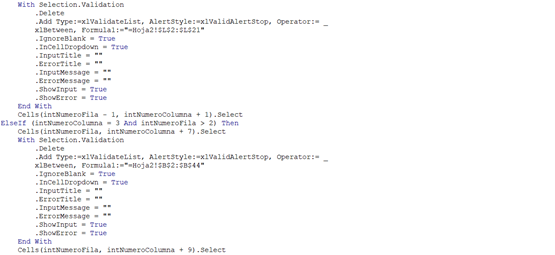


**Figura 12***Código Visual Basic para presentar en las celdas de las columnas zona, subzona, visitador un combo box con sus catálogos correspondientes.*

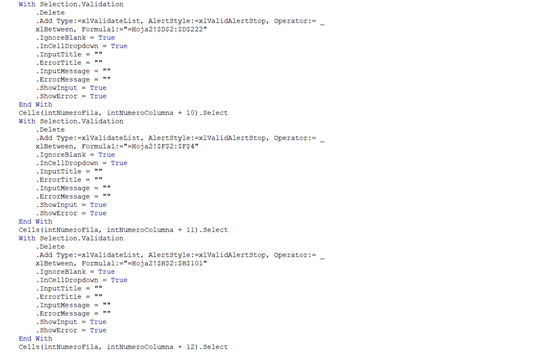




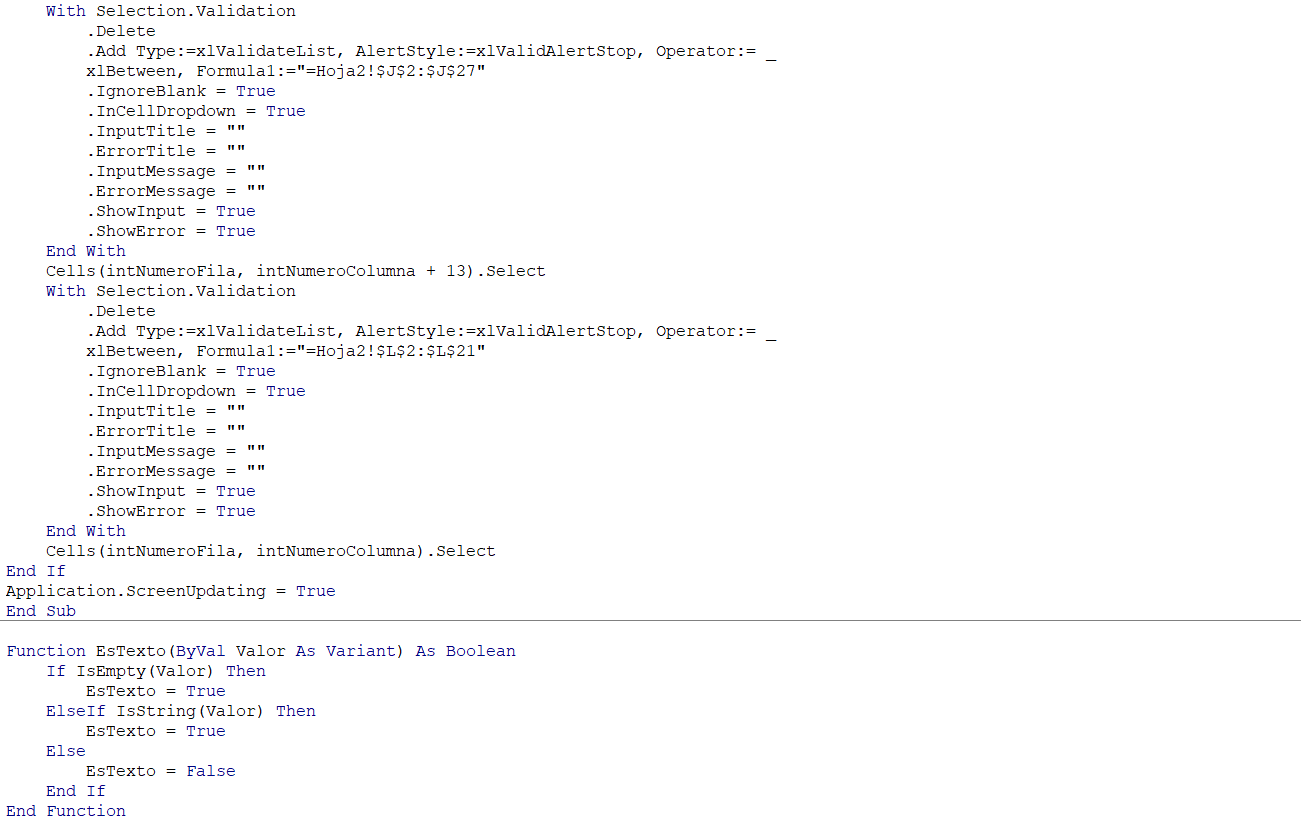
**Figura 13***Código Visual Basic para presentar en las celdas de la columna supervisor un combo box con los supervisores.*



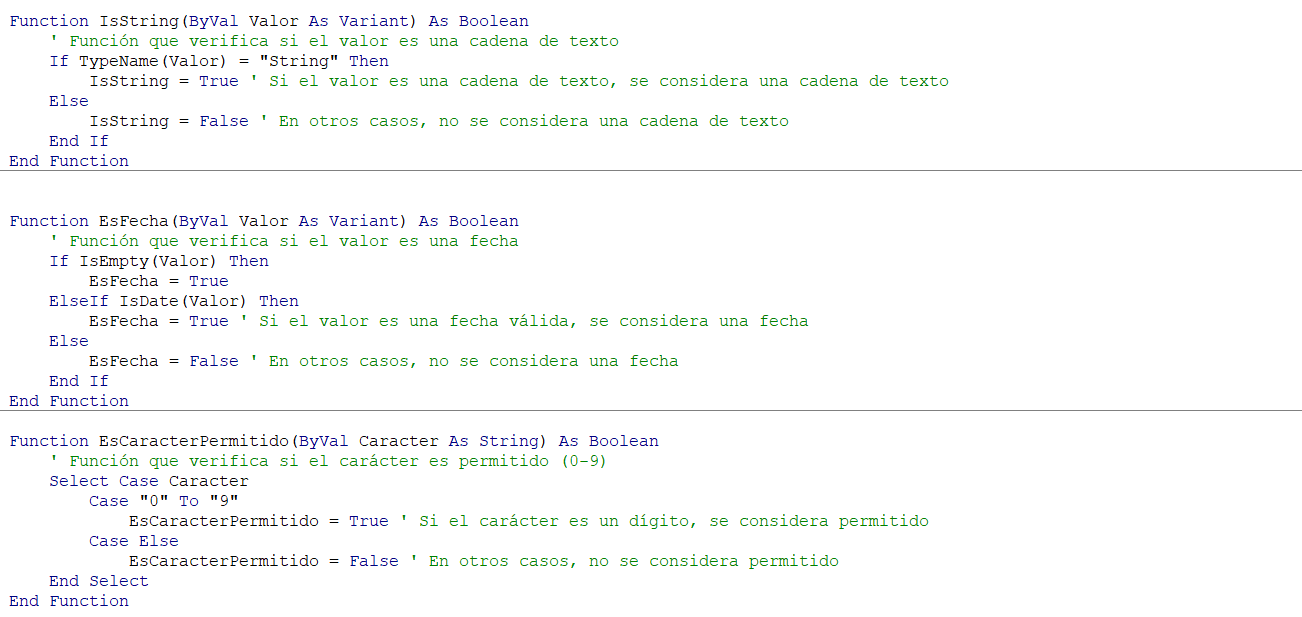
**Figura 14***Código Visual Basic para presentar en las celdas de la columna plan un combo box con los planes.*



**Figura 15***Código Visual Basic con la función que permite validar que lo ingresado en la celda sean solo letras.*



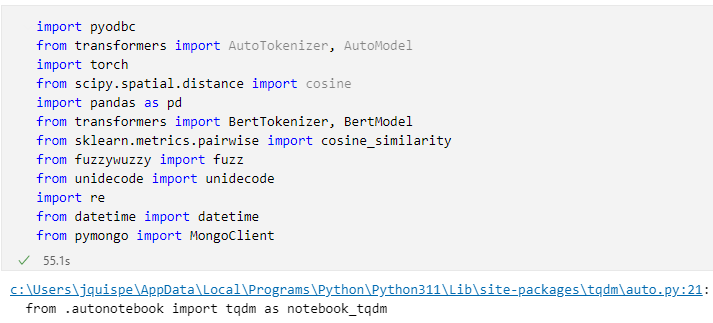
**Figura 16***Código Visual Basic con la función que permite validar que lo ingresado en la celda sean números y la función que permite validar que lo ingresado sea una fecha.*



PROGRAMACIÓN DEL PROYECTO

***Importación De Bibliotecas***

**Figura 17***Importación de bibliotecas*



***Comando ejecutado para la conexión con la BDD de SQL SERVER***

La conexión se establece mediante la biblioteca PYODBC. Los componentes utilizados son:

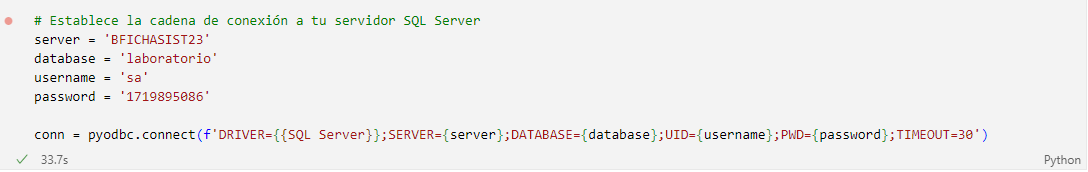
**Server**: especifica el nombre del servidor donde se encuentra alojada la base de datos.

**Database**: indica el nombre de la base de datos a la que se accederá.

**Username y Password**: credenciales de inicio de sesión necesarias para autenticar y acceder a la base de datos.

**pyodbc.connect**: establece la conexión con el servidor SQL Server mediante la especificación del controlador ('SQL Server') y la concatenación de los parámetros de conexión, incluidos el servidor, la base de datos, el nombre de usuario y la contraseña.

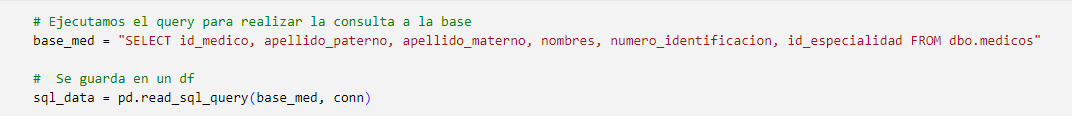
**Figura 18***Conexión BBD SQL Server*



***Consulta de base de datos de médicos***

En este código se ha realizado una consulta a la base de datos utilizando un query SQL para recuperar los siguientes campos de la tabla 'médicos': 'id\_medico', 'apellido\_paterno', 'apellido\_materno', 'nombres', 'numero\_identificacion' e 'id\_especialidad'. Los resultados de la consulta se almacenan en un DataFrame utilizando la función pd.read\_sql\_query. Esto facilita el manejo y el análisis de los datos recuperados en un formato tabular, lo que es esencial para la posterior comparación y detección de duplicados en nombres de médicos.

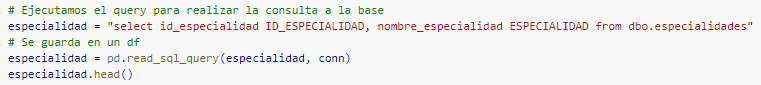
**Figura 19***Query base médicos*



***Consulta de Especialidades Médicas***

Esta consulta nos permite acceder a todos los campos de la tabla 'especialidades', lo que proporciona una visión integral de todas las especialidades médicas registradas en la base de datos del laboratorio farmacéutico.

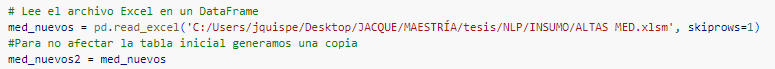
**Figura 20***Query base de especialidades*



***Carga Del Archivo Excel***

La carga de información nueva enviada por parte de los supervisores se cargará mediante el siguiente código, el cual nos permite cargar el archivo Excel con macros e iterar desde la segunda fila añadiendo al código skiprows = 1.

**Figura 21***Código para carga de archivo Excel*



Se selecciona las columnas con las cuales se va a trabajar en el modelo y tratamiento de datos.

**Figura 22***Código para seleccionar las columnas a trabajar*



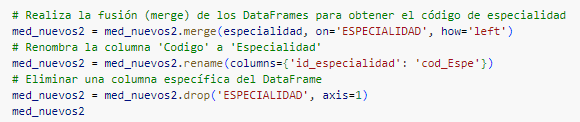
Se realiza la concatenacion de los campos de nombres y apellidos para poder ejecutar el modelo

**Figura 23***Código para concatenar los campos*



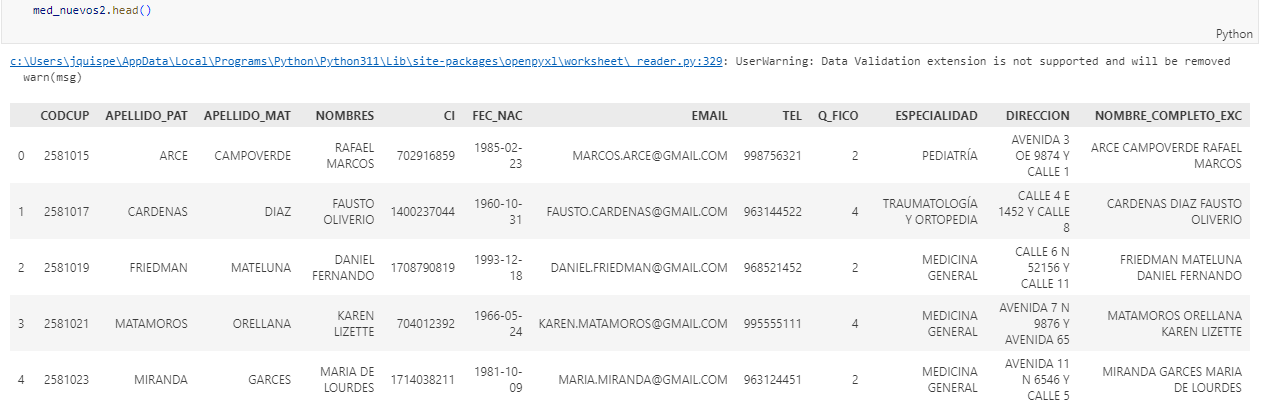
Con la finalidad de homologar la información mantenida en las bases vs la información nueva se procede a asignar el codigo de la especialidad.

**Figura 24***Merge con base de especialidades*



Finalmente se visualiza la información cargada

**Figura 25***Visualización del archivo cargado*

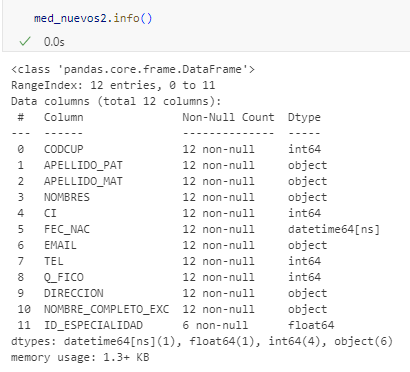


***Pre procesamiento de los datos***

La fase de pre procesamiento de los datos es bastante importante para garantizar la calidad, integridad y consistencia de los datos, por tal motivo se ejecutó una serie de comandos que nos permitió que los datos estén limpios, coherentes y listos para ser utilizados en la etapa de construcción del modelo de detección de datos duplicados basado en NLP, lo que contribuye a la eficiencia y la precisión del modelo.

Como primer paso se valida el tipo de datos que se tiene en la dataframe.

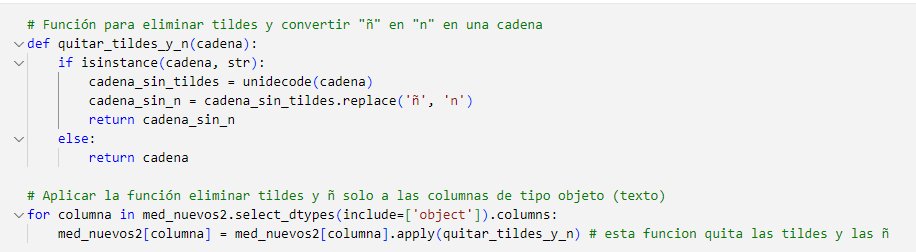
**Figura 26***Código de visualización de tipo de datos del archivo Excel*



***Eliminación de Tildes y Conversión de "ñ"***

Para asegurar que los datos de texto estén en un formato uniforme, se aplicó una función que eliminó las tildes de las letras acentuadas y convirtió la "ñ" en "n" en todas las columnas de tipo texto. Esto garantiza que no haya variaciones debidas a acentos o caracteres especiales que puedan dificultar la detección de duplicados, así como también asegura que la información que se cargue a la base no vaya con caracteres especiales difíciles de leer o que dificulten la eficiencia del modelo.

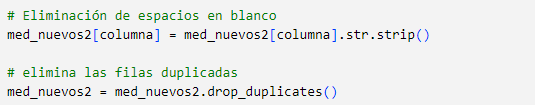
**Figura 27***Código para eliminación de tildes y Ñ*



***Eliminación de Espacios en Blanco***

Se procedió a eliminar espacios en blanco al principio y al final de los valores de las columnas de texto, lo que contribuye a la consistencia de los datos.

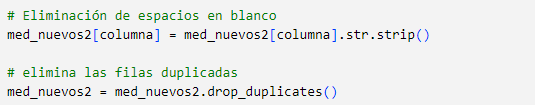
**Figura 28***Código para eliminación de espacios en blanco*



***Eliminación de Filas Duplicadas***

Con el objetivo de evitar redundancias, se eliminaron las filas duplicadas en el conjunto de datos.

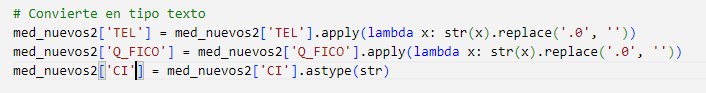
**Figura 29***Código para eliminación de filas duplicadas*



***Normalización de Datos Numéricos como Texto***

Algunas columnas que originalmente contenían datos numéricos, como "TEL" y "Q\_FICO," se convirtieron a tipo texto para asegurar que tengan la estructura que requiere la BBD del laboratorio farmacéutico. Adicional, se eliminaron los decimales ".0".

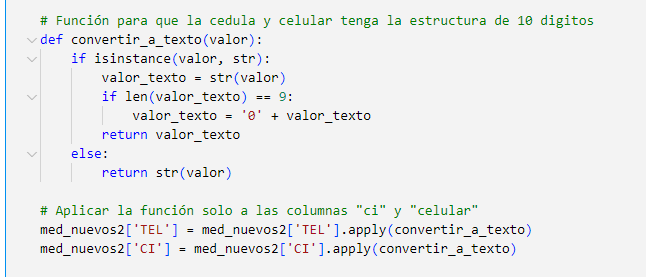
**Figura 30***Código para conversión a tipo texto*



***Estandarización de Formatos de Cédula y Número de Teléfono***

Se aplicó una función específica para que tanto los números de cédula como los números de teléfono tengan una estructura de 10 dígitos. Si el número de cédula o celular están compuesto por solo 9 dígitos, se le agregó un "0" al principio.

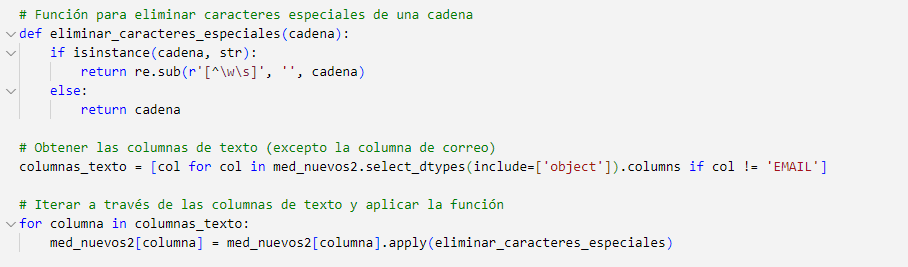
**Figura 31***Código para conversión a tipo texto*



***Eliminación de Caracteres Especiales***

Para garantizar que los datos de texto sean coherentes y puedan ser procesados de manera efectiva, se eliminaron todos los caracteres especiales de las columnas de texto, a excepción de la columna "EMAIL".

**Figura 32***Código para eliminación de caracteres especiales*



***Conversión a mayúsculas***

Con el fin de homogeneizar el formato, todo el texto presente en el conjunto de datos se convirtió a mayúsculas.

**Figura 33***Código para conversión a mayúsculas*



***Conversión de la fecha de nacimiento a texto con formato adecuado***

Para preparar los datos de fecha de nacimiento que se cargarán en la base de manera consistente y compatible, se transformó la columna "FEC\_NAC". Para que cumpla con los requisitos de la base y faciliten la gestión de datos.

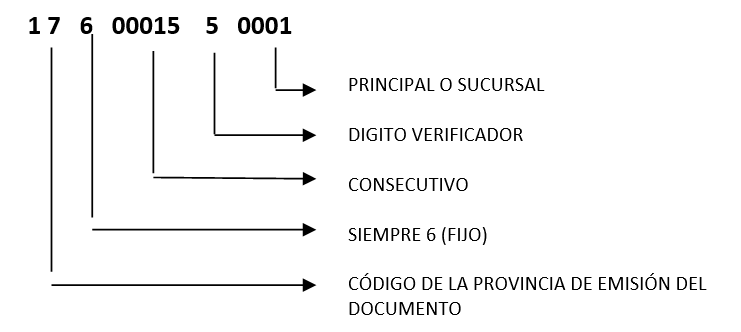
**Figura 34***Código para conversión a fecha*



***Algoritmo verificador de cédulas***

La cédula es un identificador importante en el contexto de la gestión de datos en un laboratorio farmacéutico, y es fundamental garantizar que los números de cédula sean válidos y cumplan con los estándares establecidos. Debido a ello se generó la siguiente función para verificar la validez de los números de cédula presentes en la columna "CI" del conjunto de datos.

**Figura 35***Descripción de algoritmo verificador de cédulas*



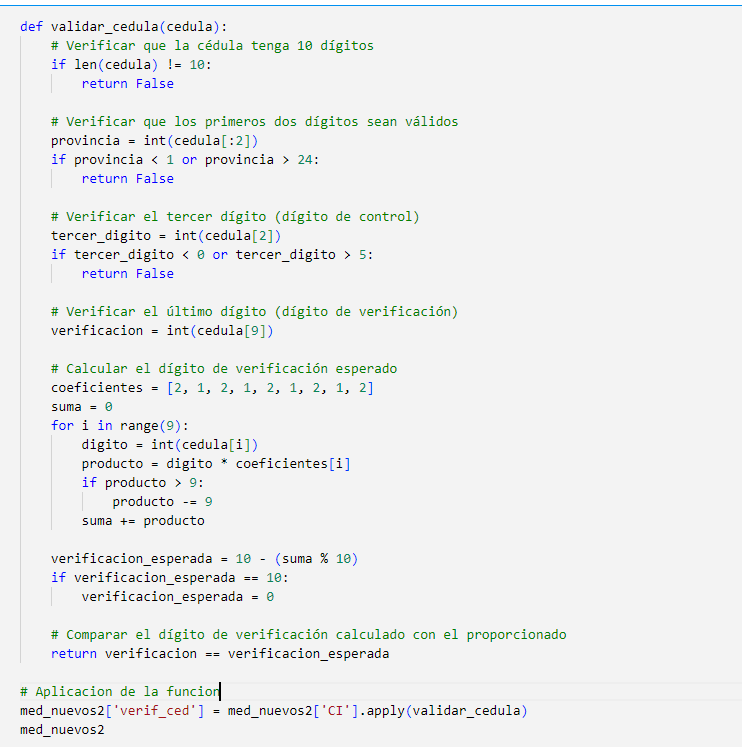
Esta función verifica primero que la cédula este compuesta por 10 dígitos, comprueba que los primeros dos dígitos representen una provincia válida en Ecuador (números del 01 al 24).

Valida el tercer dígito, conocido como "dígito de control," que debe estar en el rango de 0 a 5.

Posterior a ello calcula el "dígito de verificación" utilizando un algoritmo específico, que involucra la multiplicación de los dígitos de la cédula por coeficientes específicos y la suma de los productos resultantes sería el digito verificador.

Compara el "dígito verificador" calculado con el proporcionado en la cédula y verifica sí coinciden. En caso de coincidencia, la función devuelve True, lo que indica que la cédula es válida; de lo contrario, devuelve False.

**Figura 36***Código para validación de cédula de identidad*



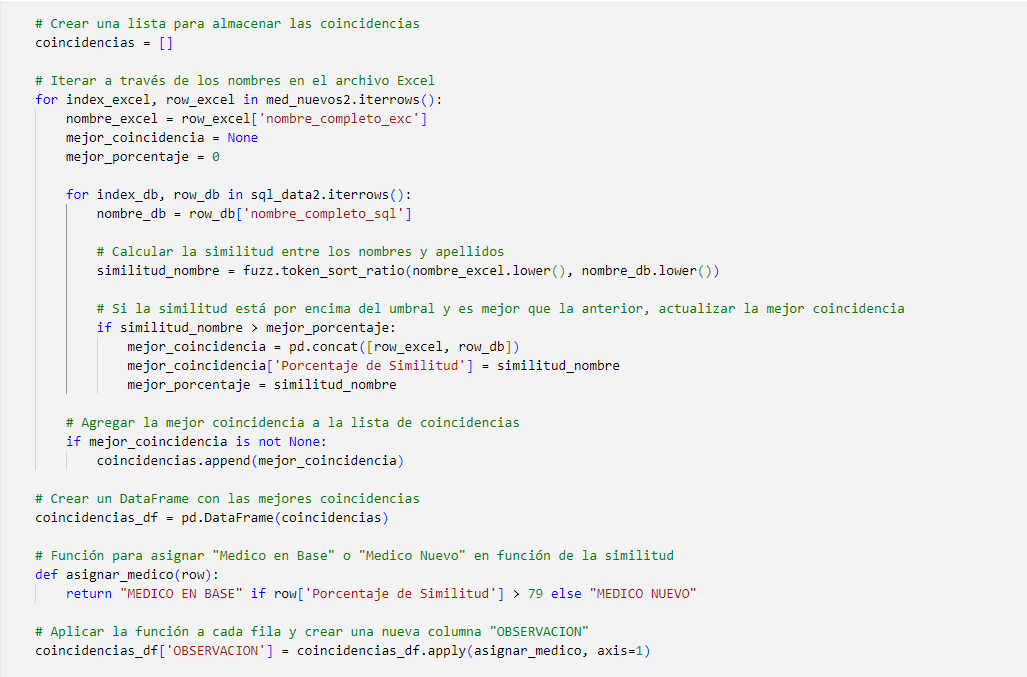
***Modelo de detección de duplicados***

Para cumplir con el objetivo del presente proyecto se lleva a cabo una comparación entre los nombres de médicos presentes en un archivo Excel y los nombres almacenados en la base de datos. Su finalidad es identificar posibles duplicados o similitudes significativas entre los nombres de médicos. A través de los siguientes modelos:

***Algoritmo Fuzzy – Token Sort Ratio***

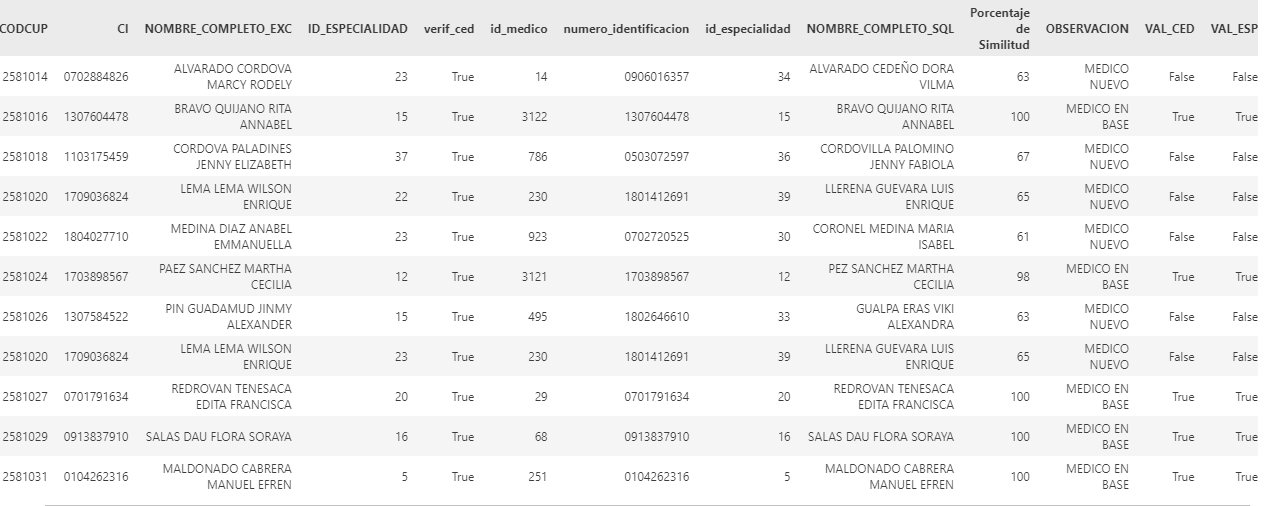
Para la ejecución del modelo fuzzywuzzy mediante algoritmo token sort ratio, el código es:

**Figura 37***Código para cálculo de similitud fuzzy-token sort ratio*



Obteniendo el siguiente resultado:

**Figura 38***Código para visualización de resultados fuzzy-token sort ratio*

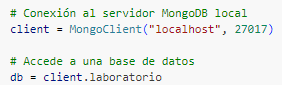


***Conexión y almacenamiento de registros históricos en MongoDB***

Utilizando la biblioteca Pymongo para Python se ha logrado establecer la conexión con el sistema de base de datos No SQL MongoDB, con la finalidad de almacenar los registros históricos, para ello:

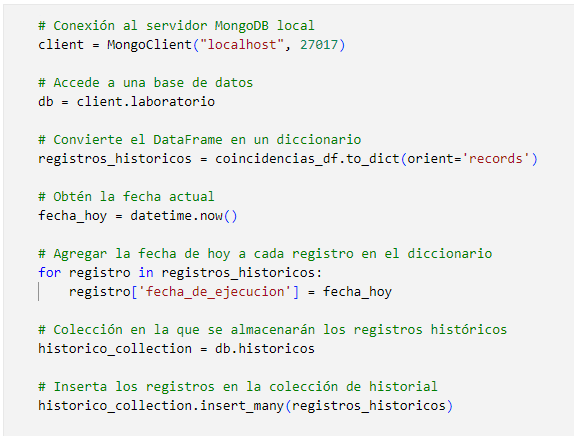
Se establece la conexión con MongoDB que se ejecuta localmente "localhost", en el puerto predeterminado 27017. Posterior a ello se ha accedido a la base de datos específica para el almacenamiento llamada "laboratorio".

**Figura 39***Conexión y acceso a la base Mongo DB*



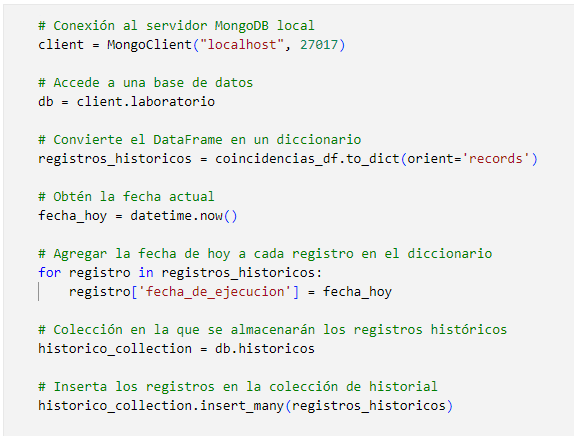
El DataFrame coincidencias\_df se ha convertido en un diccionario utilizando el método to\_dict con el parámetro orient='records'. Esto facilita la manipulación y la posterior inserción de los datos en MongoDB.

**Figura 40***Conversión en diccionario de datos*



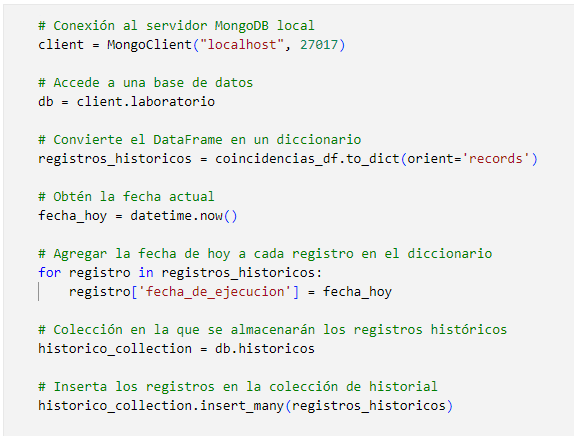
Se ha obtenido la fecha y hora actuales utilizando el módulo datetime para marcar los registros históricos con la marca de tiempo correspondiente y se añade este dato a cada registro del diccionario.

**Figura 41***Código para obtener la fecha de ejecución*



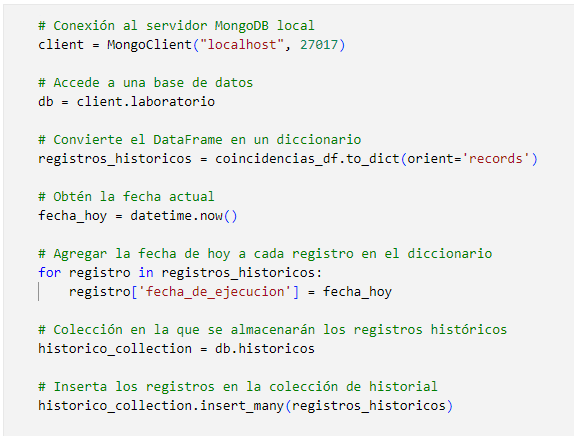
Para poder almacenar los registros históricos se ha creado la colección "históricos" en la base de datos mencionada anteriormente.

**Figura 42***Código para acceder a la colección históricos*



Finalmente, se realiza la inserción de los Registros en la Colección de Historial permitiendo así la visualización, consultas y seguimiento de los cambios realizados en la base de datos.

**Figura 43***Código para insertar los registros*

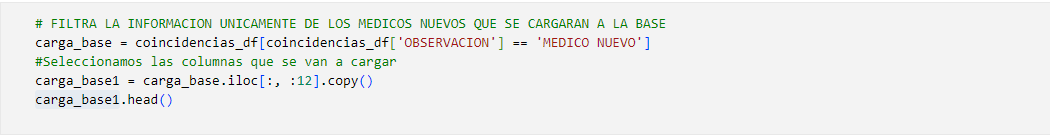


***Carga de médicos nuevos a la base de datos SQL Server***

Como último paso de este proceso, se incluye la inserción de los datos de médicos nuevos a la base de datos, para ello se ejecuta lo siguiente:

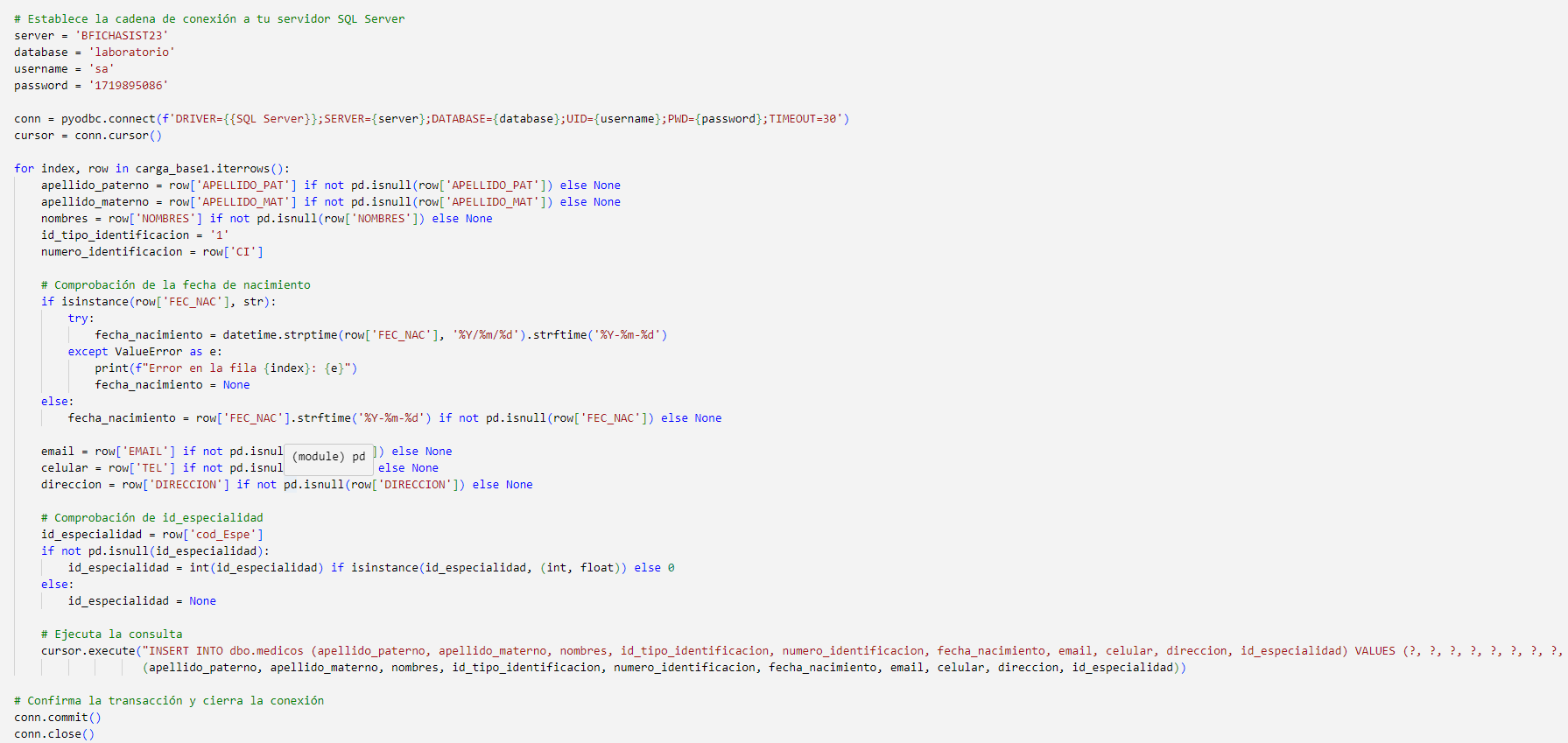
Se realiza el filtro para asegurarnos que la información insertada en la base sea únicamente de los médicos nuevos.

**Figura 44***Código para filtrar información de médicos nuevos*



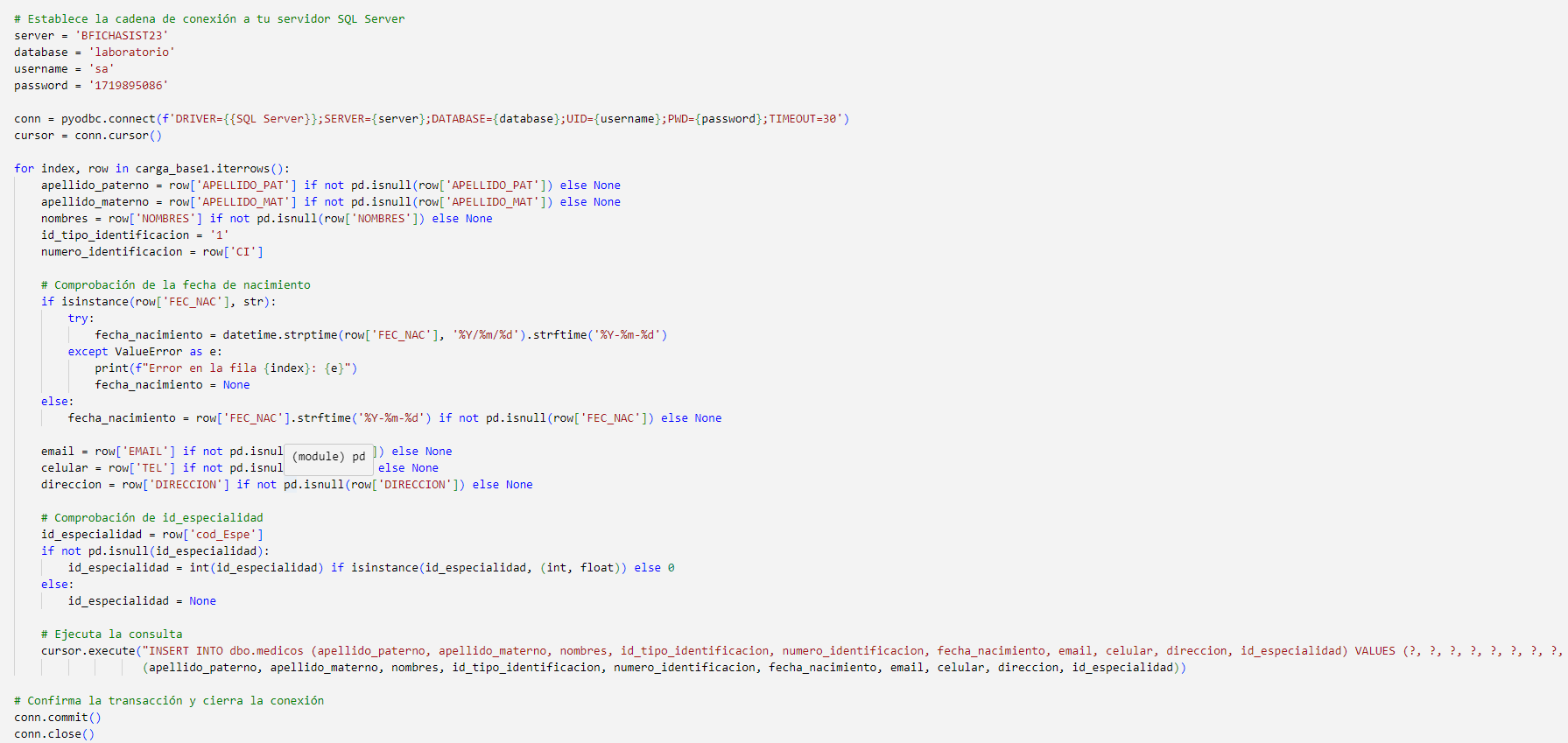
Al igual que con los anteriores sistemas de base de datos, se establece la conexión con la base de datos SQL Server utilizando la biblioteca PYODBC y los parámetros proporcionados, que incluyen el nombre del servidor, base de datos, usuario y contraseña.

**Figura 45***Código para establecer conexión con la base SQL Server*



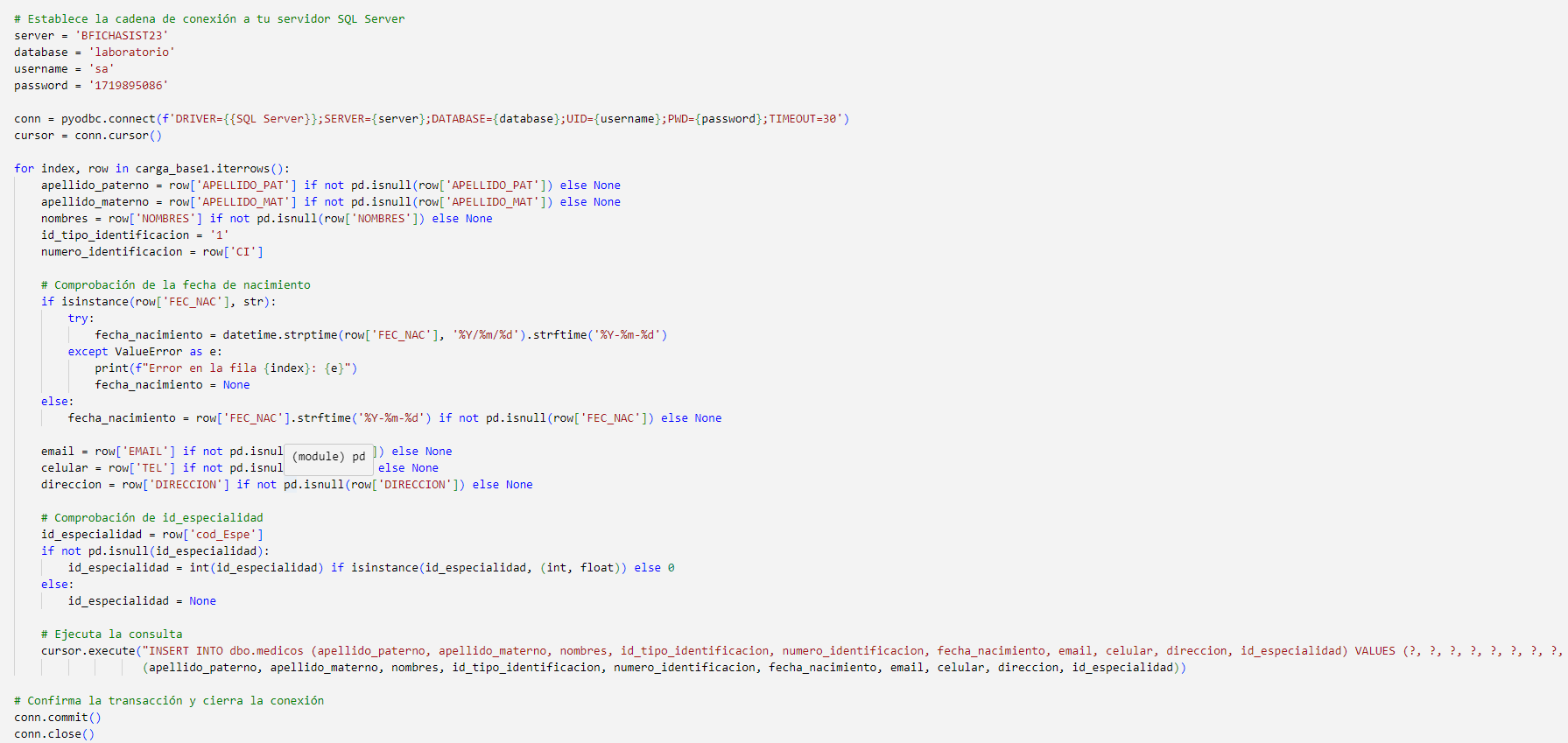
Se realiza una validación de la información contenida en las columnas para su inserción en la base de datos, teniendo en cuenta posibles valores nulos y el formato adecuado para campos como fechas y valores numéricos.

**Figura 46***Código para validar la información a cargar*



Se ejecuta una consulta SQL de inserción en la tabla 'dbo.medicos' de la base de datos, con los valores proporcionados para cada columna correspondiente. Posteriormente se aprueba la ejecución de la inserción y se cierra la conexión, lo que asegura que los datos se guarden de manera efectiva y que la conexión se cierre adecuadamente después de completar la inserción.

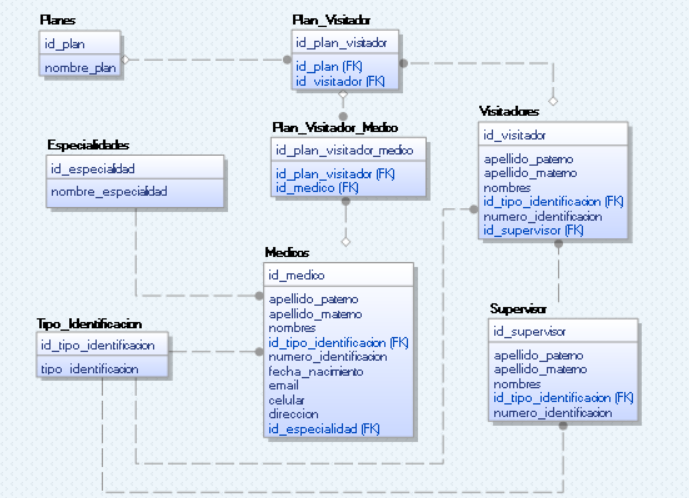
**Figura 47***Query para inserción de datos*



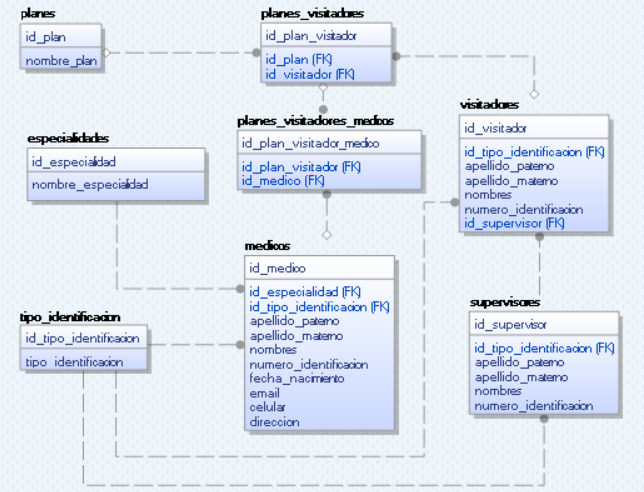
**BASE DE DATOS**

DIAGRAMA ENTIDAD-RELACIÓN

**Figura 48***Modelo Lógico de la Base de Datos*



**Figura 49***Modelo Físico de la Base de Datos*



DICCIONARIO DE DATOS

Tablas del Diagrama Médicos Laboratorio Farmacéutico

**Tabla 1***Diccionario de datos*

|  |  |  |
| --- | --- | --- |
| **N°** | **Nombre** | **Descripción** |
| 1 | Medicos | Tabla que almacena la información referente a los datos personales de los médicos. |
| 2 | Especialidades | Tabla que almacena la información de las especialidades que tiene cada medico |
| 3 | Tipo\_Identificacion | Tabla que almacena la información de los tipos de identificación que puede tener |
| 4 | Planes | Tabla que almacena la información acerca de los planes que va a tener cada médico. |
| 5 | Visitadores | Tabla que almacena la información referente a los datos de los visitadores médicos. |
| 6 | Supervisores | Tabla que almacena la información referente a los datos de los supervisores de los visitadores. |
| 7 | Planes\_Visitadores | Tabla de cruce que almacena la información referente a los datos de los planes asignados a los visitadores. |
| 8 | Planes\_Visitadores\_Medicos | Tabla de cruce que almacena la información referente a los datos de los planes de los visitadores asignados a los médicos. |

ESTRUCTURA BASE DE DATOS RELACIONAL

Las tablas que intervienen para este proyecto son:

**Tabla 2***Listado de tablas de la base de datos SQL*

|  |  |  |  |
| --- | --- | --- | --- |
| **Tablas** | **Listado de campos** | **Identificar cantidad de registros** | **Tipo de campos** |
| Especialidades | \* Código de la especialidad \* Nombre de la especialidad | 43 | \*Numérico \* Texto |
| Visitadores | \* Código del visitador \* Apellido Paterno \* Apellido Materno \* Nombres \* Cédula de identidad  \* Código del supervisor | 220 | \* Numérico \* Texto  \* Texto  \* Texto  \* Numérico  \* Numérico |
| Supervisor | \* Código del supervisor \* Apellido Paterno \* Apellido Materno \* Nombres \* Cédula de identidad | 20 | \* Numérico \* Texto  \* Texto  \* Texto  \* Texto |
| Planes | \* Código del plan \* Nombre del plan | 26 | \* Numérico \* Texto |
| Médico | \* Código del médico \* Apellido Paterno \* Apellido Materno \* Nombres  \* Número de identificación \* Fecha de nacimiento \* Correo \* Celular  \* Dirección \* Especialidad del médico | 8000 | \* Numérico \* Texto  \* Texto  \* Texto  \* Texto  \* Texto  \* Texto  \* Texto  \* Texto  \* Texto |
| Plan Visitador | \* Código plan visitador  \* Código plan  \* Código visitador | 330 | \* Numérico  \* Numérico  \* Numérico |
| Plan Visitador Médico | \* Código plan visitador médico  \* Código plan visitador  \* Código médico | 12500 | \* Numérico  \* Numérico  \* Numérico |

***Estructura Tabla Médicos***

Almacena la información referente a los datos personales de los médicos.

**Tabla 3***Estructura de la tabla Médicos*

|  |  |  |  |
| --- | --- | --- | --- |
| **Código** | **Descripción** | **Primary Key** | **Foreign Key** |
| id\_medico | Identificador único del medico | X |  |
| apellido\_paterno | Apellido paterno del medico |  |  |
| apellido\_materno | Apellido materno del medico |  |  |
| nombres | Nombres del medico |  |  |
| numero\_identificacion | Número de identificación del medico |  |  |
| fecha\_nacimiento | Fecha de nacimiento del medico |  |  |
| email | Correo electrónico del medico |  |  |
| celular | Numero celular del medico |  |  |
| dirección | Dirección del medico |  |  |
| id\_especialidad | Clave foránea de la tabla que representa al id de la especialidad del medico |  | X |

***Estructura Tabla Especialidades***

Almacena la información de las especialidades que tiene cada médico

**Tabla 4***Estructura de la tabla Especialidades*

|  |  |  |  |
| --- | --- | --- | --- |
| **Código** | **Descripción** | **Primary Key** | **Foreign Key** |
| id\_especialidad | Identificador único de la especialidad. | X |  |
| nombre\_especialidad | Nombre de la especialidad |  |  |

***Estructura Tabla Planes***

Almacena la información acerca de los planes que va a tener cada médico.

**Tabla 5***Estructura de la tabla Planes*

|  |  |  |  |
| --- | --- | --- | --- |
| **Código** | **Descripción** | **Primary Key** | **Foreign Key** |
| id\_plan | Identificador único del tipo de identificación. | X |  |
| nombre\_plan | Nombre del plan |  |  |

***Estructura Tabla Visitadores***

Almacena la información referente a los datos de los visitadores médicos.

**Tabla 6***Estructura de la tabla Visitadores*

|  |  |  |  |
| --- | --- | --- | --- |
| **Código** | **Descripción** | **Primary Key** | **Foreign Key** |
| id\_visitador | Identificador único del visitador. | X |  |
| apellido\_paterno | Apellido paterno del visitador. |  |  |
| apellido\_materno | Apellido materno del visitador. |  |  |
| nombres | Nombres del visitador |  |  |
| id\_tipo\_identificacion | Clave foránea de la tabla que representa al id del tipo de identificación del visitador |  | X |
| numero\_identificacion | Número de identificación del visitador |  |  |
| id\_supervisor | Clave foránea de la tabla que representa al id del supervisor |  | X |

***Estructura Tabla de Supervisores***

Almacena la información referente a los datos de los supervisores de los visitadores.

**Tabla 7***Estructura de la tabla Supervisor*

|  |  |  |  |
| --- | --- | --- | --- |
| **Código** | **Descripción** | **Primary Key** | **Foreign Key** |
| id\_supervisor | Identificador único del supervisor. | X |  |
| apellido\_paterno | Apellido paterno del supervisor |  |  |
| apellido\_materno | Apellido materno del supervisor |  |  |
| nombres | Nombres del supervisor |  |  |
| id\_tipo\_identificacion | Clave foránea de la tabla que representa al id del tipo de identificación del supervisor |  | X |
| numero\_identificacion | Número de identificación del supervisor |  |  |

***Estructura Tabla Planes\_Visitadores***

Tabla de cruce que almacena la información referente a los datos de los planes de que tienen asignados los visitadores.

**Tabla 8***Estructura de la tabla Planes\_Visitador*

|  |  |  |  |
| --- | --- | --- | --- |
| **Código** | **Descripción** | **Primary Key** | **Foreign Key** |
| id\_plan\_visitador | Identificador único de la tabla planes visitadores. | X |  |
| id\_plan | Clave foránea de la tabla que representa al id del plan |  | X |
| id\_visitador | Clave foránea de la tabla que representa al id del visitador |  | X |

***Estructura Tabla Planes\_Visitadores\_Médicos***

Tabla de cruce que almacena la información referente a los datos de los planes de los visitadores asignados a los médicos.

**Tabla 9***Estructura de la tabla Planes-Visitador-Médicos*

|  |  |  |  |
| --- | --- | --- | --- |
| **Código** | **Descripción** | **Primary Key** | **Foreign Key** |
| id\_plan\_visitador\_medico | Identificador único de la tabla planes visitadores médicos. | X |  |
| id\_plan\_visitador | Clave foránea de la tabla que representa al id del plan visitador. |  | X |
| id\_medico | Clave foránea de la tabla que representa al id del médico |  | X |