

Informe de proyecto [Sistema de registro de Clientes para un Hotel]

Joel Molina^{L00385791}

Universidad de las Fuerzas Armadas
jamolina13@espe.edu.ec

Abstract

El propósito de este proyecto del primer parcial es dar a conocer un programa sobre la reservación de una hotelería, mostrando su funcionabilidad, aplicativo y a su vez, abarcando el uso del lenguaje de programación en Python conjunto con el editor de código Visual Code Studio. Cabe resaltar que, para el desarrollo del proyecto se logró implementar la base de datos en MongoDB Server, por medio de Python se pudo elaborar la interfaz gráfica, conexión con la base de datos y los métodos englobando las validaciones y funciones para el uso y manejo de CRUD, esto con el fin de poder llevar a cabo las operaciones de crear, leer, modificar y eliminar datos de nuestra base de datos.

1 Introducción

Es necesario saber sobre el gestor de base de datos denominado MongoDB el cual se utilizó para el desarrollo de este proyecto final de unidad 1, abarcando sus elementos y funciones principales. Por lo que toca tener en claro que, MongoDB es un SGBD particular debido a que guarda los datos en documentos haciendo que la transferencia de documentos sea más eficientes y eficaces. Adicional, nos da la posibilidad de manejar a grande escala los volúmenes de datos siendo así uno de los aspectos primordiales de este gestor de base de datos. Cabe recalcar que, se debe tener en claro que MongoDB no usa las bases de datos relaciones, más bien da uso a las bases de datos no relacionales para su aplicativo.

Como se viene mencionando CRUD es la implementación de cuatro accionares diferente siendo estos: crear, leer, actualizar y eliminar. El conjunto de estas 4 acciones crea un sistema de procesos que por sí solo logra mejorar de forma notable el desempeño de las bases de datos para bien. El objetivo de implementar CRUD en una BDD aparte de mejorar el desempeño de esta, también son facilitar de forma eficiente la manipulación de datos por medio de los usuarios, para una mejor gestión y corrección de errores en las BDD y por último poder mostrar al usuario si la base de datos que ha realizado o en la que se encuentra trabajando está bien diseñada. Podemos definir que la acción crear como su nombre nos dice, permite crear o agregar datos a las tablas, mientras leer nos permite revisar la información de una tabla, con el accionar actualizar podremos corregir o modificar información en la BDD y por último con la acción eliminar se puede borrar datos de las tablas pertenecientes a la BDD.

A lo largo de estos años la tecnología ha ido evolucionando para el beneficio del ser humano, puesto que han mejorado cada apartado tecnológico de manera eficiente, concisa e intuitiva.

Dentro de la evolución de la tecnología existe un papel fundamental que es la base de datos, siendo esta el propulsor de poder almacenar información ya sea localmente o en la nube, permitiéndonos tener un libre manejo de información acorde a las necesidades que demanda el uso personal o también el de una o varias empresas. Así pues, surgió la idea de realizar nuestro proyecto sobre un registro de usuarios que realizan una reserva de hotelería, con el fin de satisfacer las necesidades del usuario a la hora de manipular una base de datos. Conjunto con el gestor de datos no relacionales MongoDB Server, se logró llevar a cabo la creación, modificación, lectura, y eliminación de documentos que se encuentran en la base de datos.

Contexto y motivación Con el surgimiento y posicionamiento en el mercado del cloud computing día a día surgen sistemas basados en el modelado de bases de datos que requieren hacer uso de grandes cantidades de información. Actualmente si se requiere manipular grandes cantidades de datos se lo puede hacer tanto por el lenguaje SQL o NoSQL, para procesar toda esta gran cantidad de información es recomendable hacerlo por medio de bases de datos NoSQL ya que hacer uso de SQL presenta muchas trabas como lo es la lentitud del sistema, errores o fallos de hardware. Para mejorar el rendimiento de estos sistemas se hace uso del lenguaje NoSQL manipulado a través del gestor MongoDB en colaboración con Python.

Problema de estudio Centrándose en las bases de datos estas son usadas para la implementación de sistemas o almacenamiento de información para su posterior uso, pero actualmente la evolución de los sistemas y la gran cantidad de tráfico de datos nos lleva a dejar a un lado los sistemas tradicionales y buscar mejores opciones como implementar sistemas basados en NoSQL.

Trabajo relacionado Como primer ejemplo tenemos el sistema el “Sistema de Control de Reserva y Cobros en un Hotel” se presenta como un software muy completo con una programación orientada a objetos teniendo una funcionalidad decente, pero presenta la desventaja de aplicarse directamente en una pagina web por medio de PHP y MySQL esto genera problemas como falta de manejo de internacionalización, su optimización es pésima y además resulta difícil optimizar este software y por último no posee abstracción. Con todas las problemáticas mencionadas anteriormente se deduce de forma simple que el software desarrollado en este proyecto se presenta como una alternativa muy buena ya que su optimización es mejor, cuenta con una interfaz simple y amigable y por último tiene un fácil acceso a la base de datos todo esto hace denotar que el software desarrollado es mejor opción ante el “sistema de control de reserva y cobros de un hotel” [1].

El sistema Property Management System es un sistema hotelero muy completo, este sistema hace uso de una estructura general de un PMS, esto es un problema ya que cuando un sistema se basa en dicha estructura tiene un nivel alto de complejidad. La complejidad siempre es una barrera entre el usuario y sistema es por eso que aquí el sistema desarrollado en este proyecto destaca sobre los demás ya que su fácil uso motiva al usuario hacer uso del aplicativo frente a la competencia [2].

El siguiente sistema está desarrollado por estudiantes de la Universidad Cooperativa de Colombia, tiene el nombre de Hotel Rain y su objetivo principal es lograr agilizar el proceso de registro de usuarios, basado en un funcionamiento similar a el proyecto desarrollado en este informe. Aunque a priores este sistema cumple con el objetivo de tener un sistema optimizado presenta una gran barrera frente a usuarios sin programación ya que este sistema si o si te obliga a programar sentencias en SQL para poder hacer uso de sus funcionalidades, es en este apartado donde el sistema desarrollado en este proyecto destaca frente al anterior mencionado ya que este sistema no requiere de conocimientos en programación para hacer uso de este puesto que su fun-

cionamiento ya esta implementado y lo único necesario es hacer uso de los botones y el sistema realiza todo el proceso posterior [3].

Resumen de contribución Por medio de este proyecto se desarrolla un software pensado para gestionar los clientes de un hotel de forma eficiente, este aplicativo cumple con características que lo llevan a destacar frente a otros, su fácil uso, rapidez y efectividad son complementos que provocan que sistema gestor del hotel sea llamativo para empresas pequeñas y medias y sus usuarios.

2 Antecedentes

Para la ejecución del presente proyecto de unidad hemos usado numerosas tecnologías que, en relación entre sí, tienen la posibilidad de responder las necesidades de las organizaciones que comúnmente realizan uso de esta clase de programa para su trabajo del día a día.

Bases de datos NoSQL

Las bases de datos NoSQL nacen por el cambio que surge en el manejo de los datos e información, actualmente el manejo de grandes cantidades de información y el poco tiempo requerido para hacerlo provoca que las bases de datos no relacionales sean mejor opción que las bases de datos relacionales. Las bases de datos NoSQL maneja tres aspectos principales que lo destacan frente a SQL, estos son tamaño y cantidad de información, velocidades y falta de cambio [4].

MongoDB

Es necesario saber sobre el gestor de base de datos denominado MongoDB el cual se utilizó para el desarrollo de esta práctica, abarcando sus elementos y funciones principales. Por lo que toca tener claro que MongoDB es un SGBD particular debido a que guarda los datos en documentos haciendo que la transferencia de documentos sea más eficientes y eficaces. Adicional, nos da la posibilidad de manejar a grande escala los volúmenes de datos siendo así uno de los aspectos primordiales de este gestor de base de datos. Aún así, se debe tener en claro que MongoDB no usa las bases de datos relaciones, más bien da uso a las bases de datos no relacionales para su aplicativo.

Lenguaje de programación Python

En cuanto al lenguaje de programación escogido, nos decidimos por Python por arriba de Node.js. Python es un lenguaje de programación el cual tiene una sintaxis y tipificación fácil y fácil, que inclusive una persona con conocimientos básicos de programación podría comprender. Este lenguaje además está orientado a objetos, por lo que tenemos la posibilidad de hacer uso de los pilares de la programación dirigida a objetos para el desarrollo de programa. Por otro lado, este lenguaje igual aguanta multiparadigma, por consiguiente, se puede desarrollar orientado a objetos, de manera servible y además organizada. Por otro lado, Node.js es un ámbito de ejecución que su primordial objetivo es la ejecución de JavaScript fuera del navegador. El motivo primordial por la que usamos Python ha sido pues Python cuenta con muchas bibliotecas para el desarrollo, por lo cual poseemos más posibilidades para añadir. Otra razón ha sido la función de Python de poder desarrollar Interfaces Gráficas de Usuarios fácilmente [5].

Visual Studio Code

Finalmente, poseemos el editor de escrito en el que hemos decidido desarrollar el plan. En esta situación nos hemos decantado por la implementación de Visual Studio Code. En primera instancia, poseemos que es un editor de código fuente cuya primordial característica es que aguanta trabajo con diversos idiomas de programación paralelamente. En otras palabras una virtud bastante enorme por la cual seleccionamos este programa. Por otro lado, permanecen las extensiones hechas por la sociedad para la utilización en Visual Studio Code. Estas nos

proporcionan casi infinita diversidad de posibilidades para el desarrollo de cualquier tipo. Tienen la posibilidad de ofrecer ventajas como colorear ciertas piezas del código para que nos demos cuenta de algo, y tienen la posibilidad de llegar además a ayudarnos con recomendaciones de código para el lenguaje de programación en el cual estamos haciendo un trabajo. Por lo anterior dicho, es que hemos seleccionado este editor.

3 Método

Paso 1: Descarga de Python en el sitio web oficial.

Primero que nada, debemos tener en claro el lenguaje de programación por cuál vamos a trabajar, en nuestro caso como grupo vamos a desarrollar este proyecto por medio del lenguaje de programación Python. Así que, ingresamos al navegador web para posteriormente redireccionarnos al sitio web de Python y descargar para el sistema operativo Windows tal y como se puede observar en la Figura 1.

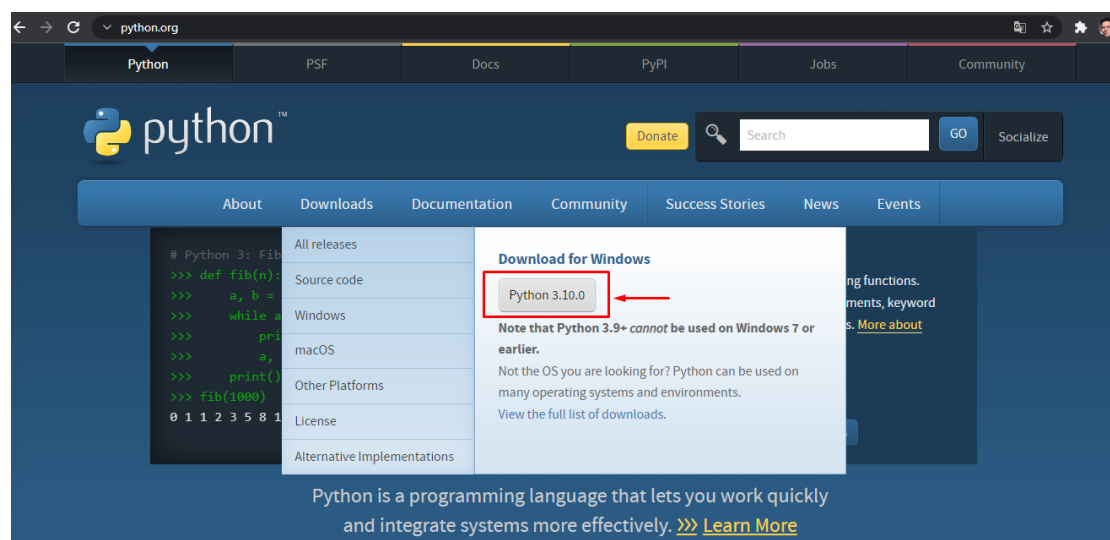


Figure 1: Descarga de Python [6].

Paso 2: Ejecución del programa Python.

Una vez descargado el ejecutable con extensión “.exe” procederemos a ejecutarlo como administrador.

Nombre	Fecha de modificación	Tipo	Tamaño
mongodb-windows-x86_64-5.0.4-signed....	23/11/2021 9:43	Paquete de Windo...	284.208 KB
python-3.10.0-amd64.exe	28/11/2021 19:10	Aplicación	27.653 KB

Figure 2: Ejecución del programa Python.

Paso 3: Instalación de Python 3.10.0.

Una vez ejecutado el programa Python procederemos a instalarlo marcando la opción que se muestra en la Figura 3.

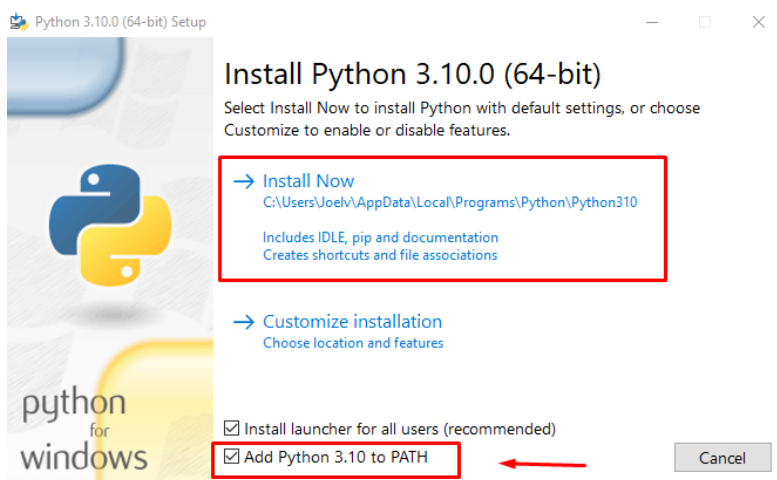


Figure 3: Instalación de Python.

Paso 4: Progreso de la configuración finalizado.

Finalmente se llegó al apartado en donde nos menciona que la instalación fue exitosa.

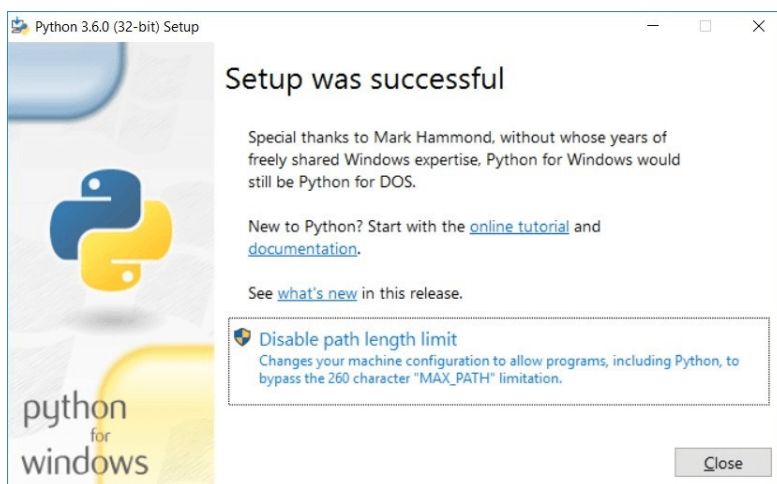


Figure 4: Instalación de Python finalizado.

Paso 5: Instalación de la extensión Python en Visual Code Studio.

En esta sección instalamos la extensión del lenguaje de programación Python en nuestro editor de código llamado Visual Code Studio. Cabe recalcar que para instalar la extensión debemos ingresar al Marketplace de Visual Code Studio para posteriormente proceder a buscarlo e instalarlo.

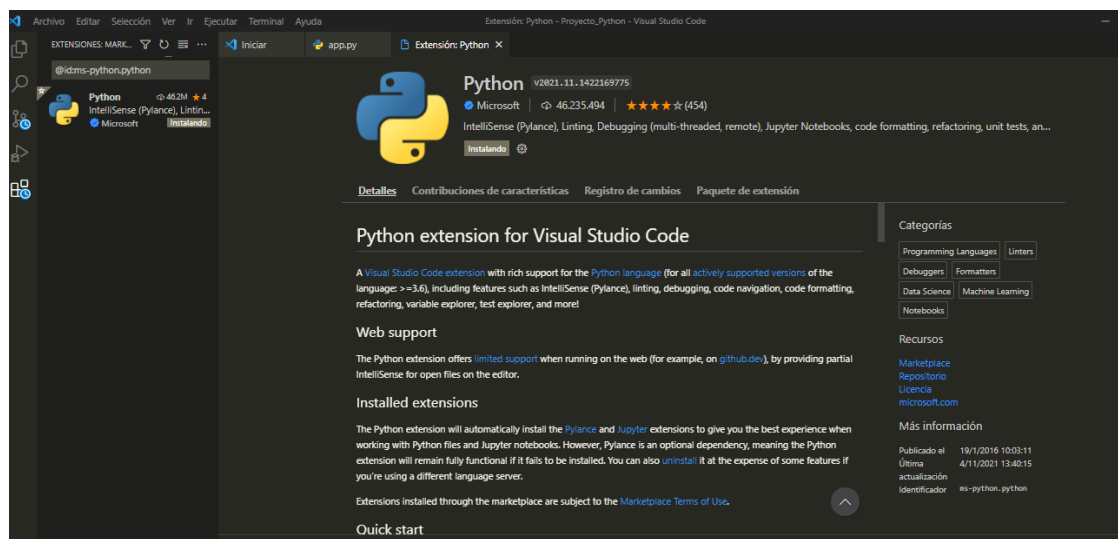


Figure 5: Instalación de la extensión de Python en VSC.

Paso 6: Verificación del funcionamiento de Python en Visual Code Studio.

Por medio de este apartado vamos a crear un archivo con extensión “py” para posteriormente abrir el terminal y verificar que todos los parámetros estén correctamente funcionales.

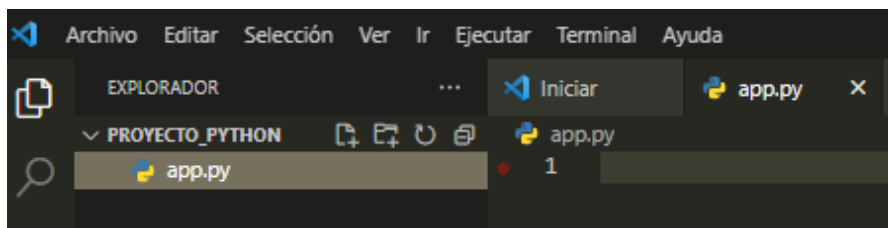


Figure 6: Verificación del funcionamiento de python en VSC.

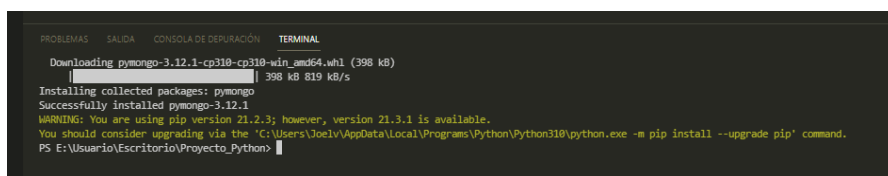


Figure 7: Terminal de Python con funcionabilidad correcta.

Paso 7: Abrir la GUI de MongoDB Compass para crear la base de datos.

Llenamos los campos que se muestran en la Figura 8. para la creación de nuestra nueva base de datos.

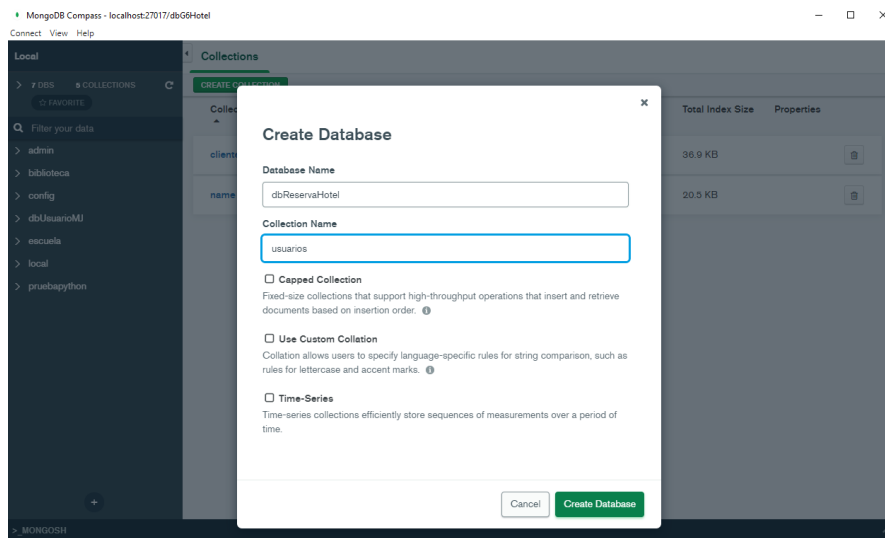


Figure 8: Inicio del GUI.

Paso 8: Ingreso de documentos dentro de la colección usuarios.

Se ingresa a la base de datos para luego proceder al ingreso de documentos dentro de la colección con el nombre usuarios.

```
--
> show dbs
admin            0.000GB
biblioteca       0.000GB
config           0.000GB
dbReservaHotel   0.000GB
dbUsuarioMJ      0.000GB
escuela          0.000GB
local            0.000GB
pruebapython     0.000GB
> use dbReservaHotel
switched to db dbReservaHotel
> db.usuarios.insert({'foto':'aqui va la foto', 'nombre':'Joel', 'apellido':'Molina', 'cedula':'2350632952', 'telefono':'0
979592898', 'correo':'jamolina13@espe.edu.ec', 'direccion':['Via Principal', 'Via Secundaria', 'Codigo Postal', 'Numero Casa
'], 'tipoAbitacion':'Individual', 'fechaI':new Date('2021,12,12'), 'fechaS': new Date('2021,12,15'), 'tarifa':'20'});
WriteResult({ "nInserted" : 1 })
>
```

Figure 9: Terminal de Python con funcionalidad correcta.

Paso 9: Verificación de documentos insertados a nuestra base de datos.

Una vez agregado documentos dentro de nuestra colección usuarios, nos redireccionaremos al GUI MongoDB Compass para observar que todos los pasos realizados se hayan concretado correctamente.

<code>_id: ObjectId("61a8732405805f839b8f7262")</code>	ObjectId
<code>foto: "aqui una foto"</code>	String
<code>nombre: "Joel"</code>	String
<code>apellido: "Molina"</code>	String
<code>cedula: "2350632952"</code>	String
<code>telefono: "0979592898"</code>	String
<code>correo: "jamolina13@espe.edu.ec"</code>	String
<code>direccion: "Via Chone"</code>	String
<code>tipoAbitacion: "Individual"</code>	String
<code>fechaI: 2021-11-11T05:00:00.000+00:00</code>	Date
<code>fechaS: 2021-11-12T05:00:00.000+00:00</code>	Date
<code>tarifa: "120"</code>	String

Figure 10: Terminal de Python con funcionalidad correcta.

Paso 10: Instalación de la librería NumPy, Matplotlib y Openvc-Python en Python.

La librería llamada NumPy principalmente nos posibilita hacer el cálculo numérico y estudio de datos para una gigantesca proporción de volúmenes de datos.

```
C:\Users\pined>pip install numpy
Collecting numpy
  Downloading numpy-1.21.4-cp310-cp310-win_amd64.whl (14.0 MB)
    | 14.0 MB 3.3 MB/s
Installing collected packages: numpy
Successfully installed numpy-1.21.4
WARNING: You are using pip version 21.2.3; however, version 21.3.1 is available.
You should consider upgrading via the 'C:\Users\pined\AppData\Local\Programs\Python\Python310\python.exe -m pip install --upgrade pip' command.
```

Figure 11: Terminal de Python con funcionabilidad correcta.

La librería denominada Matplotlib nos permite realizar gráficos en 2D.

```
C:\Users\pined>python -m pip install matplotlib
Collecting matplotlib
  Downloading matplotlib-3.5.0-cp310-cp310-win_amd64.whl (7.2 MB)
    | 7.2 MB 1.7 MB/s
Requirement already satisfied: numpy>=1.17 in c:\users\pined\AppData\Local\Programs\Python\Python310\lib\site-packages (from matplotlib) (1.21.4)
Collecting pillow<6.2.0
  Downloading Pillow-9.4.0-cp310-cp310-win_amd64.whl (3.2 MB)
    | 3.2 MB
Collecting pyparsing>=2.2.1
  Downloading pyparsing-3.0.9-py3-none-any.whl (97 kB)
    | 97 kB 6.8 MB/s
Collecting setuptools<=6.3.2-py3-none-any.whl (33 kB)
  Downloading setuptools-6.3.2-py3-none-any.whl (33 kB)
Collecting kiwisolver<=1.0.1
  Downloading kiwisolver-1.3.2-cp310-cp310-win_amd64.whl (52 kB)
    | 52 kB
Collecting fonttools<=4.22.0
  Downloading fonttools-4.28.2-py3-none-any.whl (680 kB)
    | 680 kB ...
Collecting packaging>=20.0
  Downloading packaging-21.3-py3-none-any.whl (40 kB)
    | 40 kB ...
Collecting python-dateutil<=2.7
  Downloading python-dateutil-2.8.2-py3-none-any.whl (247 kB)
    | 247 kB ...
Collecting cycler<=0.49
  Downloading cycler-0.11.0-py3-none-any.whl (6.4 kB)
Collecting six>=1.5
  Downloading six-1.16.0-py3-none-any.whl (11 kB)
Requirement already satisfied: setuptools in c:\users\pined\AppData\Local\Programs\Python\Python310\lib\site-packages (from setuptools<=6.3.2-py3-none-any.whl) (57.4.0)
Collecting toml<=1.0.0
  Downloading toml-1.2.2-py3-none-any.whl (12 kB)
Installing collected packages: pyparsing, toml, six, packaging, setuptools, python-dateutil, pillow, kiwisolver, fonttools, cycler, matplotlib
Successfully installed cycler-0.11.0 fonttools-4.28.2 kiwisolver-1.3.2 matplotlib-3.5.0 packaging-21.3 pillow-9.4.0 pyparsing-3.0.9 python-dateutil-2.8.2 setuptools-6.3.2 six-1.16.0 toml-1.2.2
WARNING: You are using pip version 21.2.3; however, version 21.3.1 is available.
You should consider upgrading via the 'C:\Users\pined\AppData\Local\Programs\Python\Python310\python.exe -m pip install --upgrade pip' command.
```

Figure 12: Terminal de Python con funcionabilidad correcta.

Sin embargo, con la librería Openvc-python vamos a poder hacer la detección de desplazamiento, reconocimiento de objetos y recomposición en tercera dimensión desde imágenes.

```
C:\Users\pined>pip install opencv-python
Collecting opencv-python
  Downloading opencv-python-4.5.4.60-cp310-cp310-win_amd64.whl (35.1 MB)
    | 35.1 MB 6.8 MB/s
Requirement already satisfied: numpy>=1.21.2 in c:\users\pined\AppData\Local\Programs\Python\Python310\lib\site-packages (from opencv-python) (1.21.4)
Installing collected packages: opencv-python
Successfully installed opencv-python-4.5.4.60
WARNING: You are using pip version 21.2.3; however, version 21.3.1 is available.
You should consider upgrading via the 'C:\Users\pined\AppData\Local\Programs\Python\Python310\python.exe -m pip install --upgrade pip' command.
```

Figure 13: Terminal de Python con funcionabilidad correcta.

3.1 Diseño

Para el desarrollo del diseño del aplicativo, primero se planteo cuales eran los factores determinantes a tener en cuenta para que el software sea amigable con el usuario e intuitivo, es por eso que se pensó en presentar un diseño con el cual este conformado por 3 frames, así cada uno de estos podrá representarse de forma autónoma para mejorar la presentación del aplicativo.

Para iniciar primero se crea el frame principal, este frame es de color verde con un tamaño de 1300 x 75, en este frame se puede visualizar el título del sitio web, así también se crea el segundo frame el cual contendrá lo que es la tabla donde se visualiza la tabla de la base de datos y por ultimo podemos apreciar el ultimo frame en el cual esta el formulario y botones que le dan funcionalidad al software.


```

framePrincipal = Frame(raiz)
framePrincipal.pack()
framePrincipal.config(width="1300", height="75")
framePrincipal.config(bg='green')

frameTabla = Frame(raiz)
frameTabla.pack()
frameTabla.config(width="1300", height="525")

frame2 = Frame(raiz)
frame2.pack()
frame2.config(width="1300", height="525")
frame2.config()

```

Figure 14: Codificación de la interfaz gráfica.

Entonces como resultado obtenemos un software que presenta en la parte superior el título o nombre, en la segunda parte esta ubicada la tabla que representa de forma grafica la información que contiene la base de datos alojada en MongoDB y para la parte inferior se aprecia lo que es el formulario con cada uno de los labels y cada uno de los botones que le da funcionalidad al software para registro de clientes de un hotel.

ID	FOTO	NOMBRE	APELLIDO	CEDULA	TELEFONO	CORREO	DIRECCION	TIPO HABITACION	FECHA INGRESO	FECHA SALIDA	TARIFA
61a8c8	aquí una foto	Jorge	Pineda	1234567891	1234567891	japinda@espe.edu.ec	Via Principal	Matrimonial	2021-12-14	2021-10-14	300
61a8c9	aquí una foto	Sebas	Calahorra	1719593981	0987628299	jcalahorra@espe.edu.ec	Santa Martha	Individual	2021-10-14	2021-10-18	125
61a87c	aquí una foto	Fabian	Carguendo	2350033391	0989892268	afcarguendo@espe.edu.ec	15 de septiembre	Matrimonial	2021-10-12	2021-10-14	300
61a873	aquí una foto	Joel	Molina	2350632952	0970920988	jmolina13@espe.edu.ec	Via Chone	Individual	2021-12-11	2021-12-16	125

Configuración de los usuarios

FOTO
 NOMBRE
 APELLIDO
 CEDULA
 TELEFONO
 CORREO
 DIRECCION
 TIPO HABITACION
 FECHA INGRESO
 FECHA SALIDA
 TARIFA

Crear usuario Editar usuario Eliminar usuario

Buscar por cedula

Buscar usuario

Figure 15: Interfaz gráfica ejecutada.

3.2 Análisis

El sistema de registro de clientes para un hotel desarrollado por el grupo Empresaurios 2.0 es un software con fines de lucro que busca por medio de la implementación de lenguaje NoSQL un funcionamiento rápido y óptimo sin fallas durante el procesamiento de información. Este sistema tiene como objetivo competir directamente con los sistemas de registros tradicionales buscando impulsar su introducción al mercado por medio de una interfaz sencilla pero funcional, se la implementa de esta forma ya que se busca persuadir a las cadenas hoteleras de implementar dicho software en sus páginas web ya que este tiene como característica presentar un entorno sencillo y amigable con el usuario. Es verdad que actualmente existen una gran cantidad de sistemas de registro de usuarios para hoteles, pero este software se presenta como una alternativa frente a los complejos sistemas actuales.

3.3 Optimización

En esta sección es importante resaltar que la optimización es fundamental a la hora de entregar un proyecto establecido, independientemente a la temática siempre es esencial tomar en cuenta que, al realizar un proyecto lo que se busca es que realice sus tareas de una manera rápida a comparación a los demás productos que se encuentran en el mercado. Así pues, nació la idea de crear nuestro proyecto con el fin de buscar mejorar los diferentes softwares que se encuentran en el mercado, ideando la manera de que el programa no requiera de mucha memoria para el funcionamiento puesto que sus recursos se aprovechan de manera competente.

Más que nada lo que tratamos de realizar con nuestro programa es que el usuario pueda utilizar el sistema de registro de clientes para un hotel, y a su vez poder aprovechar los demás componentes del sistema, agobiando recursos como memoria, procesamiento, la velocidad de lectura de la información y entre otros factores que afectan el rendimiento del ordenador que se encuentra ejecutando el programa.

3.4 Implementation

En esta sección daremos a conocer la implementación de código más relevante de nuestro proyecto de la unidad 1 que es sobre un sistema de registro de clientes para un hotel.

En este apartado creamos una variable “mostrarDatos” para mostrar datos la cual nos va a permitir verificar las variables almacenadas.

```
def mostrarDatos():
    try:
        registros = tabla.get_children()
        for registro in registros:
            tabla.delete(registro)
        for documento in coleccion.find():
            tabla.insert('', 0, text=documento["_id"], values=(documento["foto"], documento["nombre"],
                                                                documento["apellido"], documento["cedula"], documento["telef
        cliente.close()
    except pymongo.errors.ServerSelectionTimeoutError as errorTiempo:
        print("Tiempo excedido "+errorTiempo)
    except pymongo.errors.ConnectionFailure as errorConexion:
        print("Fallo al conectarse a mongodb "+errorConexion)
```

Figure 16: Variable que permite mostrar datos.

Por consiguiente, creamos una variable “crearRegistro” en la cual se procederá a crear un registro de los datos de nuestra tabla.

```
def crearRegistro():
    if len(foto.get()) != 0 and len(nombre.get()) != 0 and len(apellido.get()) != 0 and len(cedula.get()) != 0 and len(tele
    try:
        documento = {"foto": foto.get(), "nombre": nombre.get(), "apellido": apellido.get(), "cedula": cedula.get(), "t
        ), "direccion": direccion.get(), "tipoHabitacion": tipoA.get(), "fechaI": fechaI.get(), "fechaF": fechaF.get(),
        coleccion.insert(documento)
        foto.delete(0, END)
        apellido.delete(0, END)
        nombre.delete(0, END)
        cedula.delete(0, END)
        telefono.delete(0, END)
        correo.delete(0, END)
        direccion.delete(0, END)
        tipoA.delete(0, END)
        fechaI.delete(0, END)
        fechaF.delete(0, END)
        tarifa.delete(0, END)
    except pymongo.errors.ConnectionFailure as error:
        print(error)
    else:
        messagebox.showerror(message="Los campos no pueden estar vacios")
    mostrarDatos()
```

Figure 17: Variable que permite crear registro.

Por otro lado, la variable “editarRegistro” es aquello que nos permitirá lograr editar una o varios usuarios.

```
def editarRegistro():
    global ID_USUARIO
    if len(foto.get()) != 0 and len(nombre.get()) != 0 and len(apellido.get()) != 0 and len(cedula.get()) != 0 and len(tele
    try:
        idBuscar = {"_id": ObjectId(ID_USUARIO)}
        nuevosValores = {"foto": foto.get(), "nombre": nombre.get(), "apellido": apellido.get(), "cedula": cedula.get()
        ), "correo": correo.get(), "direccion": direccion.get(), "tipoHabitacion": tipoA.get(), "fechaI": fechaI.get(),
        coleccion.update(idBuscar, nuevosValores)
        foto.delete(0, END)
        nombre.delete(0, END)
        apellido.delete(0, END)
        cedula.delete(0, END)
        telefono.delete(0, END)
        correo.delete(0, END)
        direccion.delete(0, END)
        tipoA.delete(0, END)
        fechaI.delete(0, END)
        fechaF.delete(0, END)
        tarifa.delete(0, END)
    except pymongo.errors.ConnectionFailure as error:
        print(error)
    else:
        messagebox.showerror("Los campos no pueden estar vacios")
    mostrarDatos()
    crear["state"] = "normal"
    editar["state"] = "disabled"
```

Figure 18: Variable que permite editar registro.

Luego, al crear la variable “borrarRegistro” se procederá a borrar todos los datos almacenados de un registro.

```
def borrarRegistro():
    global ID_USUARIO
    try:
        idBuscar = {"_id": ObjectId(ID_USUARIO)}
        coleccion.delete_one(idBuscar)
        foto.delete(0, END)
        nombre.delete(0, END)
        apellido.delete(0, END)
        cedula.delete(0, END)
        telefono.delete(0, END)
        correo.delete(0, END)
        direccion.delete(0, END)
        tipoA.delete(0, END)
        fechaI.delete(0, END)
        fechaF.delete(0, END)
        tarifa.delete(0, END)
    except pymongo.errors.ConnectionFailure as error:
        print(error)
    crear["state"] = "normal"
    editar["state"] = "disabled"
    borrar["state"] = "disabled"
    mostrarDatos()
```

Figure 19: Variable que permite borrar registro.

De igual manera se creo 3 botones los cuales son: crear, editar y borrar.

```
# Boton crear
crear = Button(frame2, text="Crear alumno", width=30, font=(
    'Times Roman', 10), command=crearRegistro, bg="green", fg="white")
crear.grid(row=15, column=0)
# Boton editar
editar = Button(frame2, text="Editar alumno", width=30, font=(
    'Times Roman', 10), command=editarRegistro, bg="blue")
editar.grid(row=15, column=1)
editar["state"] = "disabled"

borrar = Button(frame2, text="Borrar alumno", width=30, font=(
    'Times Roman', 10), command=borrarRegistro, bg="red", fg="white")
borrar.grid(row=15, column=2)
borrar["state"] = "disabled"
```

Figure 20: Botones crear, editar y borrar.

Por consiguiente, enlazamos a una base de datos los registros de nuestra base de datos con MongoDB.

```
import pymongo
MONGO_HOST = "localhost"
MONGO_PUERTO = "27017"
MONGO_TIEMPO_FUERA = 10000

MONGO_URI = "mongodb://" + MONGO_HOST + ":" + MONGO_PUERTO + "/"

MONGO_BASEDATOS = "dbReservaHotel"
MONGO_COLECCION = "usuarios"
```

Figure 21: Conexión a la base de datos de MongoDB.

Finalmente, en esta sección únicamente es para dar a conocer que si en tal caso no hay conexión a la base de datos nos mencione que existe tal fallo.

```
try:
    cliente = pymongo.MongoClient(
        MONGO_URI, serverSelectionTimeoutMS=MONGO_TIEMPO_FUERA)
    baseDatos = cliente[MONGO_BASEDATOS]
    coleccion = baseDatos[MONGO_COLECCION]

    cliente.close()
except pymongo.errors.ServerSelectionTimeoutError as errorTiempo:
    print("Tiempo exedido "+errorTiempo)
except pymongo.errors.ConnectionFailure as errorConexion:
    print("Fallo al conectarse a mongodb "+errorConexion)
```

Figure 22: Mensaje si en tal caso no se conecta a la base de datos.

4 Evaluación Experimental

Mediante el uso de visual code como herramienta de programación, MongoDB como gestor de la base de datos y Python como lenguaje de programación orientada a objetos, se desarrolla la interfaz y funcionamiento del sistema de gestión hotelera.

4.1 Configuración experimental

En base a los objetivos a cumplir se aplican las operaciones CRUD ya que estas permiten implementar funcionalidades al software para que pueda funcionar de la forma deseada, estas acciones

que se aplican son la lectura de datos, modificación de datos, creación de datos y eliminar datos. Cada una de las operaciones anteriormente nombradas están implementadas en Python teniendo así una clase distinta para cada una, se debe tener en cuenta que también ese de carácter casi obligatorio implementar validaciones a cada una de las operaciones. La interfaz grafica también se encuentra implementada por medio de Python es así que podemos decir que MongoDB es quien aloja el sistema y Python es quien le da funcionalidad al sistema.

Cada operación implementada es llamada ejecutada por medio de los botones que se encuentran en la interfaz gráfica y así mismo en caso de hacer uso incorrecto de los botones se desarrollo por medio de las validaciones ventanas emergentes que nombren cual es el error al momento de querer implementar cierta operación.

4.2 Resultados

En esta sección de resultado se procederá a resaltar la interfaz final del proyecto del sistema de registro de clientes para un Hotel. A continuación, mediante la Figura 23 observaremos la interfaz principal del proyecto.

NOMBRE	APELLIDO	CEDULA	TELEFONO	CORREO	CIUDAD	TIPO HABITACIÓN	FECHA INGRESO	FECHA SALIDA	TARIFA
Fabian	Carguacundo	2341234567	0982134556	fcarguacundo@espe.edu.ec	Santo Domingo	Individual	2021-12-15	2021-12-16	130
Sebas	Calahorrano	9876543219	0934554321	jcalahorrano@espe.edu.ec	Santo Domingo	Individual	2021-12-13	2021-12-14	120
Jorge	Pineda	1718506601	0934567890	jpineda@espe.edu.ec	Santo Domingo	Individual	2021-12-11	2021-12-12	120
Joel	Molina	2350632952	0979552898	jemolina13@espe.edu.ec	Santo Domingo	Individual	2021-12-11	2021-12-16	125

Configuración de los usuarios

NOMBRE
 APELLIDO
 CEDULA
 TELEFONO
 CORREO
 CIUDAD
 TIPO HABITACIÓN
 FECHA INGRESO
 FECHA SALIDA
 TARIFA

Crear usuario Editar usuario Borrar usuario

Buscar por cedula

Buscar usuario

Figure 23: Interfaz general del proyecto.

Luego se realizó una búsqueda por cédula ingresando los datos acorde a un usuario ya registrado.

Sistema de Registro de Clientes - Hotel Empresarios 2.0

NOMBRE	APELLIDO	CEDULA	TELEFONO	CORREO	CIUDAD	TIPO HABITACIÓN	FECHA INGRESO	FECHA SALIDA	TARIFA
Jorge	Pineda	1718506601	0934567898	japineda@espe.edu.ec	Santo Domingo	Individual	2021-12-11	2021-12-12	120

Configuración de los usuarios

NOMBRE

APELLIDO

CEDULA

TELEFONO

CORREO

CIUDAD

TIPO HABITACIÓN

FECHA INGRESO

FECHA SALIDA

TARIFA

Crear usuario
Editar usuario
Borrar usuario

Buscar por cedula

Buscar usuario

Figure 24: Búsqueda por cédula del usuario.

Después de realizar el paso anterior, vamos a modificar a un usuario cambiando los datos que se encuentran ya establecidos.

Sistema de Registro de Clientes - Hotel Empresarios 2.0

NOMBRE	APELLIDO	CEDULA	TELEFONO	CORREO	CIUDAD	TIPO HABITACIÓN	FECHA INGRESO	FECHA SALIDA	TARIFA
Fabian	Carguacundo	2341234567	0982134556	fcarguacundo@espe.edu.ec	Santo Domingo	Individual	2021-12-15	2021-12-16	130
Sebas	Calahorrano	9876543219	0934554321	jscalahorrano@espe.edu.ec	Santo Domingo	Individual	2021-12-13	2021-12-14	120
Jorge	Pineda	1718506601	0934567898	japineda@espe.edu.ec	Santo Domingo	Individual	2021-12-11	2021-12-12	120
Joel	Molina	2350632952	0979592898	jamolina13@espe.edu.ec	Santo Domingo	Individual	2021-12-11	2021-12-16	125

Configuración de los usuarios

NOMBRE

APELLIDO

CEDULA

TELEFONO

CORREO

CIUDAD

TIPO HABITACIÓN

FECHA INGRESO

FECHA SALIDA

TARIFA

Crear usuario
Editar usuario
Borrar usuario

Buscar por cedula

Buscar usuario

Figure 25: Modificación de datos de un usuario.

Por consiguiente, mediante la Figura 26 observaremos la modificación realizada de manera exitosa, conjunto con los datos modificados.

NOMBRE	APELLIDO	CEDULA	TELEFONO	CORREO	CIUDAD	TIPO HABITACIÓN	FECHA INGRESO	FECHA SALIDA	TARIFA
Fabian	CaCaguano	1827496435	0986969944	fcarguacundo@espe.edu.ec	Santo Domingo	Individual	2021-12-15	2021-12-16	130
Sebas	Calahorrano	9876543219	0934354321	jcalahorrano@espe.edu.ec	Santo Domingo	Individual	2021-12-13	2021-12-14	120
Jorge	Pineda	1718506601	0934367898	japineda@espe.edu.ec	Santo Domingo	Individual	2021-12-11	2021-12-12	120
Joel	Molina	2350632952	0979592898	jamolina13@espe.edu.ec	Santo Domingo	Individual	2021-12-11	2021-12-16	125

Figure 26: Resultado final de los datos modificados.

Posteriormente, procedemos a agregar un nuevo usuario completando los campos que se pueden visualizar en la Figura 27.

Configuración de los usuarios

NOMBRE

APELLIDO

CEDULA

TELEFONO

CORREO

CIUDAD

TIPO HABITACIÓN

FECHA INGRESO

FECHA SALIDA

TARIFA

Crear usuario

Editar usuario

Borrar usuario

Buscar por cedula

Buscar usuario

Figure 27: Ingreso de datos para nuevo usuario.

Una vez ya ingresado los datos para el registro del nuevo usuario, nos saldrá una ventana emergente en donde mencionará que los datos del usuario creado fueron exitosos.

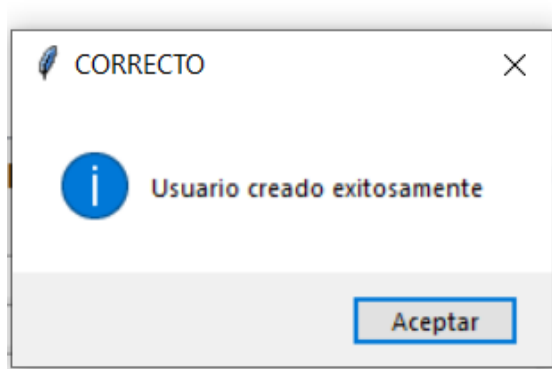


Figure 28: Mensaje emergente que menciona al usuario registrado.

Verificamos que el usuario se registró exitosamente en la base de datos.

	NOMBRE	APELLIDO	CEDULA	TELEFONO	CORREO	CIUDAD	TIPO HABITACIÓN	FECHA INGRESO	FECHA SALIDA	TARIFA
	Nuevousuaric	Nuevoapellid	1234123423	0981235123	nuevo@espe.edu.ec	NuevaCiudad	Habitacion	2021-12-20	2021-12-29	100
	Fabian	CaCaguano	1827496435	0986999944	fcarguacundo@espe.edu.e	Santo Domingo	Individual	2021-12-15	2021-12-16	130
	Sebas	Calahorrano	9876543219	0934554321	jscalahorrano@espe.edu.ec	Santo Domingo	Individual	2021-12-13	2021-12-14	120
	Jorge	Pineda	1718506601	0934567898	japineda@espe.edu.ec	Santo Domingo	Individual	2021-12-11	2021-12-12	120
	Joel	Molina	2350632952	0979592898	jamolina13@espe.edu.ec	Santo Domingo	Individual	2021-12-11	2021-12-16	125

Figure 29: Verificación de usuario registrado.

Así pues, procedemos a eliminar al usuario nuevo registrado.

	NOMBRE	APELLIDO	CEDULA	TELEFONO	CORREO	CIUDAD	TIPO HABITACIÓN	FECHA INGRESO	FECHA SALIDA	TARIFA
	Fabian	CaCaguano	1827496435	0986999944	fcarguacundo@espe.edu.e	Santo Domingo	Individual	2021-12-15	2021-12-16	130
	Sebas	Calahorrano	9876543219	0934554321	jscalahorrano@espe.edu.ec	Santo Domingo	Individual	2021-12-13	2021-12-14	120
	Jorge	Pineda	1718506601	0934567898	japineda@espe.edu.ec	Santo Domingo	Individual	2021-12-11	2021-12-12	120
	Joel	Molina	2350632952	0979592898	jamolina13@espe.edu.ec	Santo Domingo	Individual	2021-12-11	2021-12-16	125

Configuración de los usuarios

NOMBRE
APELLIDO
CEDULA
TELEFONO
CORREO
CIUDAD
TIPO HABITACIÓN
FECHA INGRESO
FECHA SALIDA
TARIFA

Crear usuario

Editar usuario

Borrar usuario

Buscar por cedula

Buscar usuario

Figure 30: Eliminación de usuario registrado.

Después de revisar las funciones de nuestro proyecto, procedemos a mostrar las validaciones que cuenta cada apartado de los campos. Por consecuente, observamos que en la Figura 31 que existe error cuando los campos están vacíos, cuando no se ha ingresado una cédula, cuando no se ha ingresado un nombre y cuando no se a ingresado un correo electrónico asociado al usuario.

ERROR

Los campos no pueden estar vacios

Aceptar

ERROR

Solo se admiten numeros la cédula

Aceptar

ERROR

Solo se admiten letras en el nombre

Aceptar

ERROR

Debe contener un @: example@.com

Aceptar

Figure 31: Validación de campos sobre el usuario.

Finalmente, las últimas validaciones son sobre las fechas, tanto de ingreso como de salida y,

a su vez, sobre no aceptar letras en el tipo de habitación.

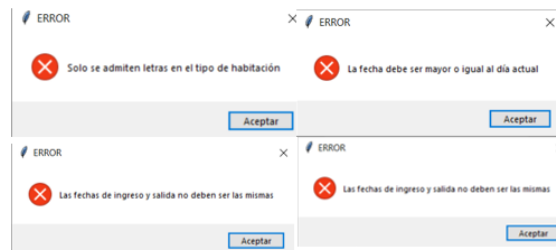


Figure 32: Validación final sobre los campos del usuario.

5 Conclusión

Ultimadamente, con el pasar de los tiempos hemos percatado que las bases de datos no relacionales han tomado mucho impacto con respecto al almacenamiento a grandes escalas de volúmenes de datos, tanto así que son más eficaces y rápidas con respecto a las bases de datos tradicionales que conocemos actualmente (locales). Entonces, lo que queremos dar a conocer es que, nos permiten trabajar por medio de una base de datos que contiene colecciones y cada una de las colecciones se encuentran con documentos que forman parte de la base de datos que, en este caso tiene el nombre de “ReservaHotel”.

A lo largo del desarrollo del informe del proyecto, hemos abarcado cada temática acorde a la implementación de la: guía de implementación del lenguaje de programación Python, creación, modificación, eliminación y manejo de lectura por medio de una base de datos con una colección y, documentos insertados por medio de la Shell de MongoDB Server. Asimismo, con el uso del editor de texto Visual Studio Code implementamos la instalación de las extensiones de Python junto con las librerías de NumPy, Matplotlib y Openvc-Python los cuales sirven para realizar cálculo numérico, graficar en 2D y el desplazamiento, reconocimiento de objetos y recomposición en tercera dimensión desde imágenes. Si bien no está demás mencionar que cada uno de estos apartados son fundamentales para implementar el programa de registro de hotelería, también debemos conocer que la programación en Python cumple el rol principal para poder llevar a cabo la GUI y las funciones que queremos implementar para así poder registrar ya sea nombre, apellido, foto, dirección, cédula y entre otros datos personales de un cliente.

Generalmente, nuestro sistema de registro de usuarios de hotelería está pensando para ser la mejor opción entre hoteles pequeños. Esto se debe a que tiene un balance perfecto entre el costo y las necesidades de un hotel pequeño. Los hoteles pequeños normalmente se manejan con un sistema de registros manual, en el cual el recepcionista encargado tiene que hacer uso de este registro para asentar a los usuarios que se están hospedando en el hotel, y así llevar a cabo un repertorio de los clientes que han visitado el hotel. Este proceso tiende a ser muy tedioso cuantos más registros se lleven a cabo cada mes, ya que si alguna vez tienen que buscar algún cliente que se ha hospedado durante el mes, tienen que revisar todo el registro. Así pues, la situación anterior descrita se puede evitar mediante el uso de un software el cual registre a los clientes en una base de datos y, los almacene de tal manera que podemos buscar a un usuario en específico mediante su carnet de identidad y así obtener los datos de este usuario sin tener que realizarlo manualmente con el libro donde se registran los clientes. Por ende, tomando los puntos mencionados anteriormente, lo que se busca es tratar de optimizar el tiempo y sutilmente el dinero que invierte el hotel en el registro de sus clientes manualmente, esto se debe a que la

búsqueda del software siempre es más veloz que la búsqueda manual que un empleado realiza y, las adquisición del software siempre va a rentar de mejor manera en tema de costos, puesto que sólo sería una adquisición, mientras que en el libro de registros se puede llegar a gastar mensualmente.

References

- [1] L. Festinger and J. Carlsmith, “Sistema de control de reserva y cobros en un hotel.,” *UNIVERSIDAD DEL BIOBIO*, 2011.
- [2] —, “Sistemas hoteleros para optimizar tu hotel o posada.,” *asksuite*, 2019.
- [3] J. Rada and L. Lozano, “Sistema de gestión hotelera.,” *UNIVERSIDAD COOPERATIVA DE COLOMBIA*, 2017.
- [4] R. Castro and A. González, “Utilidad y funcionamiento de las bases de datos nosql,” *Universidad Pedagógica y Tecnológica de Colombia*, 2012.
- [5] K. Mohit, “Node.js vs python: Elige la mejor tecnología para tu aplicación web,” *Kinsta Inc.*, 2021.
- [6] I. Python, “Download python 3.10.0,” *Python Software Foundation*, 2021.