

**Instituto Politécnico Nacional**

Escuela Superior de Cómputo

## DOCUMENTACIÓN PRÁCTICA 1

Autor:

Contreras Barrita José Roberto

Materia:

Ingeniería de Software

Grupo:

3CV17

Profesor:

Vélez Saldaña Ulises

# Índice

<b>1. Diseño</b>	<b>2</b>
1.1. Tecnologías . . . . .	2
1.2. Diseño y estructura del proyecto . . . . .	2
1.3. Estructura de los datos . . . . .	4
<b>2. Descarga y ejecución</b>	<b>5</b>
2.1. Requisitos del sistema . . . . .	5
2.2. Descarga . . . . .	5
2.3. Ejecución . . . . .	5
<b>3. Funcionamiento y pruebas</b>	<b>7</b>
3.1. Agregar cliente . . . . .	8
3.2. Modificar cliente . . . . .	8
3.3. Eliminar cliente . . . . .	9
3.4. Listar cliente . . . . .	9
3.5. Salir del programa . . . . .	9
<b>4. Consideraciones</b>	<b>10</b>

# 1. Diseño

Para la creación de esta práctica se tuvieron que realizar suposiciones y decisiones para poder realizar el desarrollo. A continuación se describirán.

## 1.1. Tecnologías

Se decidió crear una aplicación que se ejecute en consola en un entorno Linux por lo cual se empleo el lenguaje de programación **Python** para la creación del código debido a que es un lenguaje de aprendizaje sencillo, de código abierto y que su interprete se encuentra instalado en todas las distribuciones Linux por lo que su ejecución no implicará ningún problema.

Una funcionalidad del programa es poder almacenar los datos de los clientes por lo cual se optó por utilizar bases de datos para la conservación y fácil manejo de la información, en este caso se utilizó el gestor de bases de datos **SQLite** el cual tiene su implementación en Python y no necesita la instalación de algún otro software extra como en el caso de otros gestores de bases de datos.

## 1.2. Diseño y estructura del proyecto

El manejo del programa se hará mediante un menú de opciones en el que el usuario dará como entrada mediante la consola el número de la opción a desear, este menú aparecerá después de realizar cada operación y se quitará una vez el usuario seleccione la opción de salir, para este programa solo es necesario el uso del teclado. En cuestión de seguridad los usuarios solo podrán ingresar al sistema si conocen la clave de acceso. Para el manejo de

la información se creó una base de datos con dos tablas, la tabla `clients` donde se almacena la información de los clientes que registre el usuario y la tabla `sec` donde se almacena la contraseña de acceso al programa. El proyecto contará con una estructura definida por los siguientes elementos:

- **docs:** Directorio que contendrá la documentación del proyecto en un archivo PDF.
- **res:** Directorio en el que se guardarán todos los recursos necesarios para que el programa funcione correctamente. Aquí estará el archivo necesario para el funcionamiento de la base de datos, este archivo es portable por si se desea usar el programa en otro ordenador con la información ya creada.
- **src:** Directorio donde se encuentra el código fuente del programa y tiene los archivos a interpretar, aquí se decidió codificar en base a módulos donde se crearon los siguientes dos archivos:
  - Archivo **main.py**: Contiene la lógica principal y las instrucciones para interactuar con el usuario. Hace uso del módulo `dbmanager`.
  - Archivo **dbmanager.py**: Módulo que contiene todas las operaciones para el manejo de la base de datos.
- **README.txt:** Archivo que da una breve descripción del programa, su estructura y su instalación.
- **run.sh:** Script en bash que se encarga de correr el programa.

### **1.3. Estructura de los datos**

Se tomaron ciertas decisiones al momento de definir los tipos de datos. Para el alias se considera un texto de máximo 10 dígitos considerando espacios. El nombre deberá tener máximo una longitud de 30 caracteres y los apellidos una longitud máxima de 20 caracteres cada uno. La razón social no tiene un límite de caracteres y acepta la mayoría de los caracteres.

El RFC tiene una longitud máxima de 13 caracteres ya que es la longitud máxima que puede tener un RFC en México. El número telefónico deberá tener una longitud de 10 dígitos debido a que es el nuevo estándar en México y el correo tendrá una longitud máxima de 50 caracteres considerando el dominio. En el apartado de seguridad la clave de accesos deberá tener máximo 50 caracteres y no debe tener espacios de preferencia.

## 2. Descarga y ejecución

Para este apartado se necesitara noción de comandos en Linux.

### 2.1. Requisitos del sistema

Para la instalación y ejecución de este programa deberá tener instalado el software de control de versiones **git** y el interprete de Python en este caso en su versión **python3**.

### 2.2. Descarga

Abriremos la terminal y nos dirigiremos a la ruta donde queremos guardar los archivos del programa. Una vez ubicados ahí ejecutaremos el siguiente comando para descargar todo lo necesario.

```
git clone https://github.com/jcontrerasroberto/isw-practical.git
```

### 2.3. Ejecución

Después de descargar el repositorio ingresaremos a la carpeta que contiene los archivos del programa:

```
cd isw-practical
```

Otorgamos permisos de ejecución al script `run.sh` y lo corremos:

```
chmod +x run.sh  
./run.sh
```

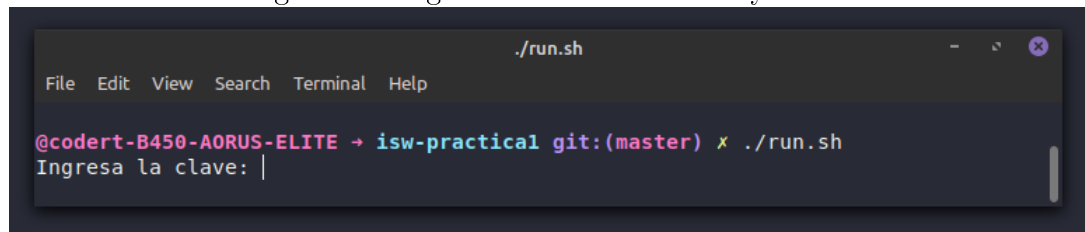
O en su defecto corremos el programa directamente con el comando:

```
python3 src/main.py
```

De ahora en adelante solo tendrás que entrar a la carpeta `isw-practical1` y correr el script `run.sh` para iniciar el programa

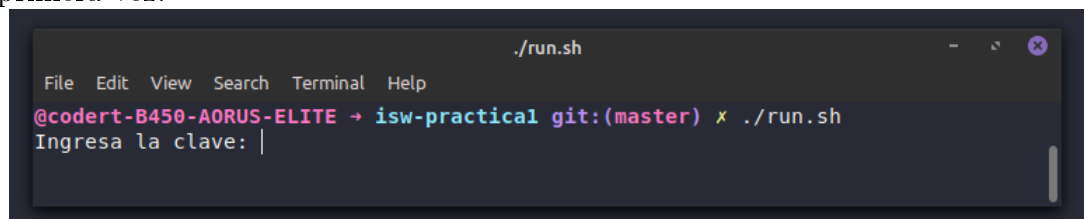
### 3. Funcionamiento y pruebas

La primera vez que ejecutemos el programa pedirá que ingresemos una nueva contraseña de seguridad. Ingresamos la contraseña y damos enter.

A terminal window titled './run.sh' with a menu bar (File, Edit, View, Search, Terminal, Help). The prompt is '@codert-B450-AORUS-ELITE → isw-practical git:(master) x ./run.sh'. The output is 'Ingresa la clave: |' with a cursor at the end.

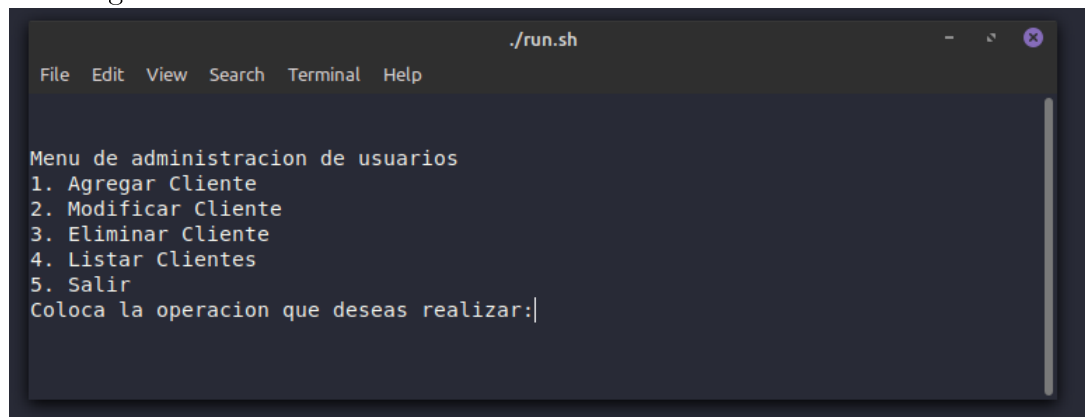
```
./run.sh
File Edit View Search Terminal Help
@codert-B450-AORUS-ELITE → isw-practical git:(master) x ./run.sh
Ingresa la clave: |
```

Las siguientes veces que iniciemos el programa pedirá la contraseña creada la primera vez.

A terminal window titled './run.sh' with a menu bar (File, Edit, View, Search, Terminal, Help). The prompt is '@codert-B450-AORUS-ELITE → isw-practical git:(master) x ./run.sh'. The output is 'Ingresa la clave: |' with a cursor at the end.

```
./run.sh
File Edit View Search Terminal Help
@codert-B450-AORUS-ELITE → isw-practical git:(master) x ./run.sh
Ingresa la clave: |
```

Después de ingresar la contraseña observaremos un menú donde el usuario puede escoger.

A terminal window titled './run.sh' with a menu bar (File, Edit, View, Search, Terminal, Help). The output is a menu titled 'Menu de administracion de usuarios' with five options: 1. Agregar Cliente, 2. Modificar Cliente, 3. Eliminar Cliente, 4. Listar Clientes, 5. Salir. The prompt is 'Coloca la operacion que deseas realizar:|' with a cursor at the end.

```
./run.sh
File Edit View Search Terminal Help
Menu de administracion de usuarios
1. Agregar Cliente
2. Modificar Cliente
3. Eliminar Cliente
4. Listar Clientes
5. Salir
Coloca la operacion que deseas realizar:|
```



### 3.1. Agregar cliente

Para usar esta función ingresaremos **1** en el menú. El programa nos pedirá los datos, iremos ingresándolos y apretando enter hasta que se registre al usuario.



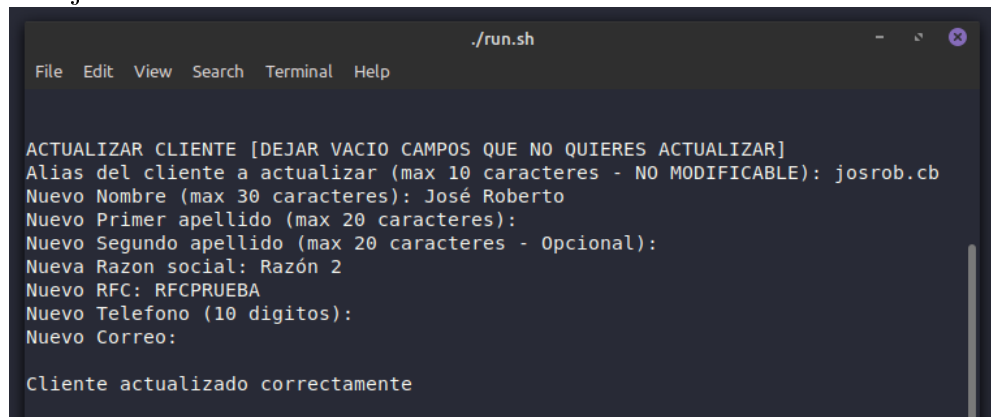
```
./run.sh
File Edit View Search Terminal Help

AGREGAR CLIENTE
Alias (max 10 caracteres): josrob.cb
Nombre (max 30 caracteres): Roberto
Primer apellido (max 20 caracteres): Contreras
Segundo apellido (max 20 caracteres - Opcional): Barrita
Razon social: Mi razón social
RFC: RFC12345
Telefono (10 digitos): 5516897854
Correo: contreras.jrobert@gmail.com

Cliente agregado correctamente
```

### 3.2. Modificar cliente

Para usar esta función ingresaremos **2** en el menú. El programa nos pedirá el alias del cliente a modificar y una vez que lo encuentre empezará a pedir los datos actualizados. Nota: Si no se quiere actualizar un campo solo dar enter dejando la entrada vacía.



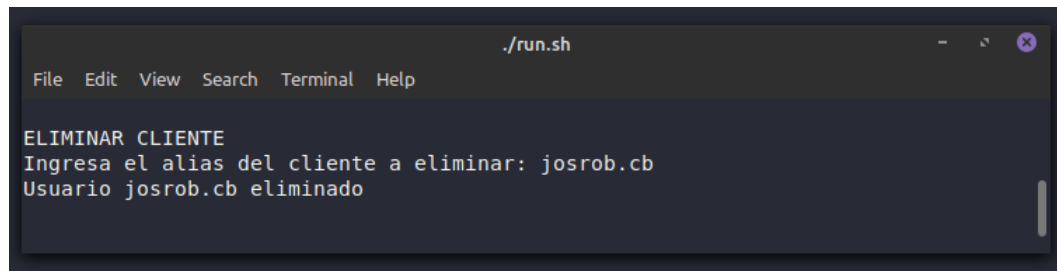
```
./run.sh
File Edit View Search Terminal Help

ACTUALIZAR CLIENTE [DEJAR VACIO CAMPOS QUE NO QUIERES ACTUALIZAR]
Alias del cliente a actualizar (max 10 caracteres - NO MODIFICABLE): josrob.cb
Nuevo Nombre (max 30 caracteres): José Roberto
Nuevo Primer apellido (max 20 caracteres):
Nuevo Segundo apellido (max 20 caracteres - Opcional):
Nueva Razon social: Razón 2
Nuevo RFC: RFCPRUEBA
Nuevo Telefono (10 digitos):
Nuevo Correo:

Cliente actualizado correctamente
```

### 3.3. Eliminar cliente

Para usar esta función ingresaremos **3** en el menú. El programa nos pedirá el alias del cliente a eliminar, solo se ingresa el alias y se da enter para eliminar al cliente.

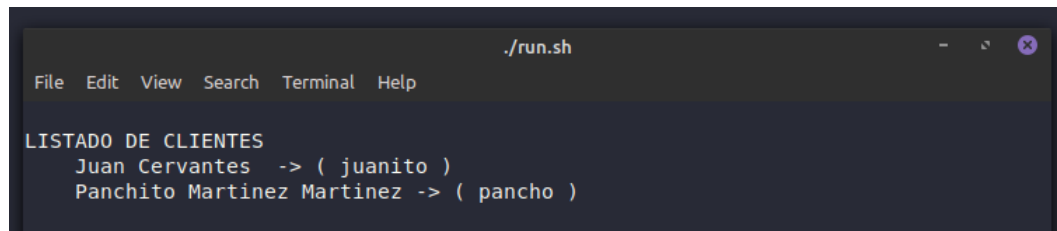


```
./run.sh
File Edit View Search Terminal Help

ELIMINAR CLIENTE
Ingresa el alias del cliente a eliminar: josrob.cb
Usuario josrob.cb eliminado
```

### 3.4. Listar cliente

Para usar esta función ingresaremos **4** en el menú. Se nos mostrará un listado mostrándonos el nombre y el alias de los usuarios agregados en el sistema.



```
./run.sh
File Edit View Search Terminal Help

LISTADO DE CLIENTES
Juan Cervantes -> ( juanito )
Panchito Martinez Martinez -> ( pancho )
```

### 3.5. Salir del programa

Para salir del programa solamente seleccionamos la opción **5** en el menú.

## 4. Consideraciones

Este sistema no necesita mucha potencia de hardware por lo que puede funcionar correctamente en computadores antiguos con distribución Linux ya sean con o sin entorno gráfico. Es un programa útil para el manejo de registros y que puede ser escalable añadiendo mejoras y nuevas funciones como la de mostrar todos los datos del cliente.

Dentro de los aspectos que faltan por programar es la validación de los campos, por ejemplo que en correo solo acepte entradas que tengan un formato *correo@domain.com* o que valide el RFC. Así mismo falta el manejo de errores con bloques try/catch para evitar un cierre inesperado del sistema.

De igual manera es posible que la lógica de programación se pueda mejorar o se puedan utilizar mejores prácticas al momento de programar en Python y finalmente se podría mejorar el instalador o se podría buscar la manera de empaquetar el programa en un archivo binario.