

Como montar el proyecto

Para poder ver el proyecto en un entorno local se necesitan lo siguiente:

- Python 3.8.9
- Flask
- mysql-connector-python 8.0.29

La base de datos fue hecha con MySQL 8.0.31.

Una vez instalado todo lo necesario, se correrá el script **database_creation** que se encuentra dentro de la carpeta **Base de datos**, dicho script hará:

- Creará una base de datos llamada “distributors”
- Creará un usuario llamado “dp_test” con contraseña “DpPythonTest1*_”
- Dará los privilegios al usuario “dp_test” sobre “distributors” en un entorno local
- Creará las siguientes tablas:
 - distributors
 - persons
 - addresses
 - phone_numbers
 - persons_phone_numbers
- Insertará 5 registros en cada tabla nombrada anteriormente.
- Creará los procedimientos almacenados utilizados en el proyecto.

Una vez instalado lo necesario y con la base de datos creada, desde una terminal ingresamos a la carpeta Proyecto y usamos el siguiente comando:

```
python app.py
```

En caso de no encontrar el comando python la posible solución sería usar:

```
python3 app.py
```

Tenga en cuenta que la base de datos se abre con el puerto por default que es el 3306 y el Proyecto en python se abre en el puerto 4000

El puerto de Python se podría cambiar en el archivo app.py, en la parte de abajo en la siguiente línea de código:

```
app.run(debug=True, port=4000)
```

Documentación del proyecto

En el siguiente proyecto se crea una aplicación web en la cual se implementa un CRUD para la base de datos de distribuidores, en ella se puede observar la información de los distribuidores ingresados en la base de datos, así como dar de alta uno nuevo, hacer un borrado lógico para que no sea visible y también editar los campos de dirección y teléfono.

Funciones del proyecto

Inicio

Para la implementación del proyecto se ejecutaron diferentes funciones, la primera es en el home, es la que imprime los distribuidores en una tabla, la cual cuenta con identificador, nombre completo, dirección y el teléfono que tiene activo actualmente.

Para el desarrollo de esta pantalla se creó un procedimiento almacenado el cual incluye una selección de todas las tablas del proyecto, la consulta regresa los datos diseñados para mostrarse y usando métodos de limpieza.

Distribuidores

Identificador	Nombre completo	Dirección	Teléfono	Acciones
A123	Jorge Cárdenas Galicia	Calle: Juan Escutia, #211, Colonia: Juárez	6699115566	<div>Editar</div> <div>Eliminar</div>
B456	Arturo Ramirez Lopez	Calle: Rosales, #15, Colonia: Jabañes	1234567890	<div>Editar</div> <div>Eliminar</div>
C789	Brenda Jimenez Islas	Calle: Colorada, #3, Colonia: Toledo	6789012345	<div>Editar</div> <div>Eliminar</div>
D234	Samantha Lopez Flores	Calle: Luis Zuñiga, #16, Colonia: Centro	0987654312	<div>Editar</div> <div>Eliminar</div>
E567	Amanda Cadenas Hernández	Calle: Rosales, #20, Colonia: Vila Galaxia	1234	<div>Editar</div> <div>Eliminar</div>

Dar de alta un distribuidor (/user)

La implementación de la pantalla user se cuenta con varios pasos donde primero se toma la información del distribuidor y se verifica que sea la necesaria para dar de alta, en caso contrario se mostrará el error.

Después llamamos un procedimiento almacenado llamado add_distribuidor en el cual mandamos identificador, nombre, segundo nombre (en caso de haber), apellido paterno y apellido materno, este procedimiento almacenado envía un error si el identificador ya está dado de alta.

Si el identificador está dado de alta se verifica si tiene un borrado lógico y se pregunta si se requiere reactivar, de lo contrario sólo se muestra el error.

Si add_distribuidor no muestra ningún error, se llama al siguiente procedimiento almacenado que es add_address_tel, al cual se le envía el identificador, calle, numero, colonia y teléfono, aquí se inserta la información en su respectiva tabla y campo.

Agregar distribuidor

Identificador
Nombre
Segundo nombre
Apellido paterno
Apellido materno
Calle
Numero
Colonia
Teléfono
Agregar

Eliminar un distribuidor (/delete/id)

Aquí se toma el distribuidor y se le asigna el estado en 0, esto para no mostrarlo en la pantalla principal y para no tomarse en cuenta en futuros procesos.

Reactivar un distribuidor (/reactivate/id)

Aquí se toma el distribuidor y se le asigna el estado en 1, para volver a ser visible en la pantalla principal.

Editar distribuidor (/update/id)

Esta pantalla inicia tomando la información del formulario de editar, primero se crea una consulta para actualizar la dirección del distribuidor, la consulta se hace con una serie de if para revisar cuáles datos son los que se van a actualizar.

Después verifica si se necesita actualizar el teléfono, en el cual se llama a un procedimiento almacenado llamado update_tel, dicho procedimiento verifica si el distribuidor ya ingreso ese teléfono a la base de datos y cuál es su estatus.

Si ya está ingresado se verifica el estado, si el estado es 1 significa que está activo y no se requiere ninguna acción. Si el estado es 0, se pone en 0 el teléfono que esté activo y a continuación se pone en 1 el ingresado.

Si no está ingresado primeramente ponemos en estado 0 el teléfono que esté activo, a continuación ingresamos el teléfono nuevo con estado 1.



Editar Jorge Cárdenas Galicia

X

Calle

Numero

Colonia

Teléfono

Guardar cambios

Notas

Los mensajes de error, warning o success se muestran en la parte de arriba de la pantalla.

El distribuidor ya existe ¿desea reactivarlo?

Reactivar

Extras

Dentro de la carpeta Base de datos, además del archivo maestro llamado `database_creation`, también se encuentran los siguientes archivos:

- **procedimientos_almacenados:** Aquí están por separado los procedimientos almacenados que se usaron en el proyecto.
- **limpieza_datos:** Este archivo no se usa en ninguna parte del proyecto, es un extra en el cual se toma la opción de que en los campos de solo texto, los números se puedan cambiar por la letra parecida (Ej: '3' se cambiaría por la letra 'e'), al igual que eliminar los caracteres especiales.
- **database_model:** Este archivo está en formato mwb, fue creado con MySQL Workbench y se puede abrir ahí, es el modelo de la base de datos.
- **database_model_img:** Si no sería posible abrir el archivo anterior también se agrega una imagen con el modelo.

Limpieza de datos

Para imprimir los datos se requirió hacer una limpieza ya que al inicio se ingresó la información con errores.

Se crearon dos regex, los cuales son:

1. **regex_str** = "[0-9]|[\\|\\|!@#\$.&*~^_{};;<>,'?\\|()-]+"

Fue creada para eliminar todos los dígitos y los caracteres especiales.

Se implementó en los campos nombre, segundo_nombre, apellido_paterno y apellido_materno de la tabla persons, además de usarse en los campos calle y colonia de la tabla addresses.

2. **regex_digits** = "[a-z]|[\\|\\|!@#\$.&*~^_{};;<>,'?\\|()-]+";

Fue creada para eliminar todas las letras y los caracteres especiales.

Se implementó en los campos numero y tel de las tablas addresses y phone_numbers respectivamente.

Conclusión

El proyecto es bueno para ver el tipo de consultas que se hacen en una base de datos, desde la creación de una base de datos, de tablas, diferentes tipos de relaciones y sobre todo como mostrar la información afrontándose a relaciones de uno a uno y de uno a muchos.

Modelar la información en las consultas es algo primordial para el entendimiento de la información, utilizar funciones para concatenar, para sustituir o eliminar caracteres son indispensables para que la información sea de mejor calidad.

Python es una herramienta buena herramienta para el tratado de los datos, teniendo frameworks poderosos que facilitan el trabajo.