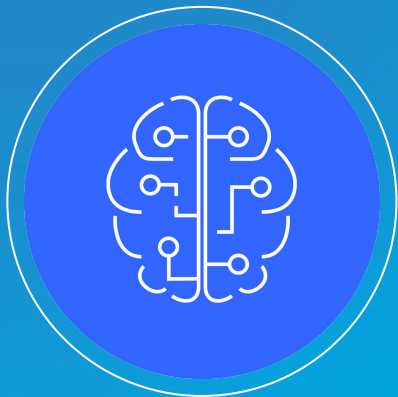




TECNICATURA SUPERIOR EN

Ciencia de Datos e Inteligencia Artificial



Materia

Innovación en Gestión de Datos – TSCDIA
2024

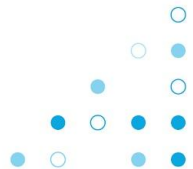
Equipo - Integrantes: *Grupo 8*

Nombre Apellido: Ian Denis Marini

Nombre Apellido: Cristhian Sebastian
Sosa

Nombre Apellido: Simon Azul Sanchez
Vottero

**“Es la ciencia del
pensamiento
racional”**

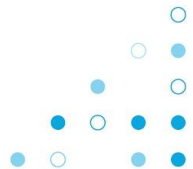


DESCRIPCIÓN

Este proyecto consiste en desarrollar un programa en Python que interactúe con una base de datos MySQL para gestionar y registrar datos de usuarios y precipitaciones. Incluyendo el uso de gráficas para visualizar datos registrados.

El proyecto se divide en varias capas:

- Desarrollo de funciones para el manejo de usuarios y accesos
- Creación de bases de datos y registro de precipitaciones
- Desarrollo de gráficos para visualizar datos pluviales
- Conexión a base de datos MySQL y pruebas

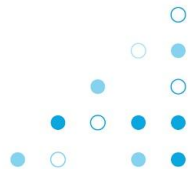


JUSTIFICACION

Este proyecto permite gestionar información de usuarios y precipitaciones, utilizando Python y MySQL. Es útil para instituciones que necesitan analizar datos climáticos o gestionar información de usuarios en bases de datos. La motivación es adquirir experiencia práctica en Python, SQL y en la generación de gráficos de datos.

OBJETIVOS

- Desarrollar un sistema de gestión de registros y visualización de datos en Python y MySQL.
- Implementar funciones para la gestión de usuarios y accesos.
- Crear bases de datos para almacenar información de usuarios y registros de precipitaciones.
- Generar gráficos para visualizar los datos de precipitaciones de forma clara y analítica.

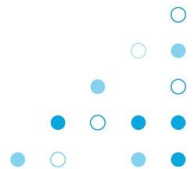


METODOLOGÍA

En este proyecto, se aplicaron metodologías de trabajo estructuradas y organizadas para asegurar un desarrollo eficaz y ordenado, además de una entrega completa y documentada.

Dividimos el proyecto en evidencias o entregas parciales. Cada evidencia correspondía a una fase específica del proyecto, con un conjunto de objetivos y tareas bien definidas.

- En **Evidencia 1**: Trabajamos en la base de datos, diseño inicial y requerimientos básicos.
- En **Evidencia 2**: Nos enfocamos en la Programación Orientada a Objetos (POO), manejo de archivos, y la creación de relaciones en la base de datos.
- En **Evidencia 3**: Añadimos opciones avanzadas de ordenamiento, gráficos, y optimizaciones en la estructura de la base de datos.
- **Entrega Final**: Consistió en consolidar todas las evidencias en una aplicación completa y bien documentada.



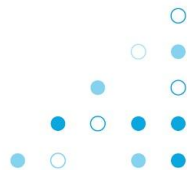
METODOLOGÍA

Cada fase del proyecto fue construida sobre la anterior, añadiendo complejidad y funcionalidad en cada etapa. Este método permitió la entrega de versiones parciales del sistema, cada una con nuevas funcionalidades y mejoras.

Al realizar desarrollos incrementales, pudimos realizar pruebas en cada versión, detectando y corrigiendo errores en etapas tempranas. Esto aseguró una construcción más estable y permitió un seguimiento claro de los avances.

- Empezamos con la creación de la **base de datos** y el código básico de **Python** para gestión de datos.
- Luego añadimos funciones de **POO**, manejo de archivos y gráficos.
- En la fase final, optimizamos la base de datos y añadimos consultas avanzadas en el backend.

Utilizamos **GitHub** como plataforma principal para la colaboración en el proyecto. Cada integrante del equipo trabajó en diferentes aspectos del sistema, y las tareas fueron divididas y organizadas a través del **tablero de proyectos** del repositorio de GitHub.



Registrar Usuarios e Iniciar Sesión

Propósito de registrar_usuario: Esta función permite registrar nuevos usuarios en el sistema con información clave como el DNI, correo, fecha de nacimiento, nombre de usuario y contraseña. La información se almacena en el sistema, verifica que los datos cumplan con ciertos criterios de validez (**Fig. 1**).

```
----- Menú Principal -----
0. Menú Fidelización de Clientes
-----
1. Registrar Usuario
2. Iniciar Sesión
3. Modificar Usuario
4. Eliminar Usuario
-----
Menú Ordenar y Buscar Usuarios
-----
5. Ordenar y Buscar Usuarios
6. Mostrar Todos los Usuarios
-----
Cargar / Graficar (Registros Pluviales)
-----
7. Cargar Registros Pluviales
8. Graficar Registros Pluviales
9. Salir
-----
Seleccione una opción: 1
Nombre: Alan
Apellido: Mariani
DNI: 34584698
Correo electrónico: AlanMar@gmail.com
Fecha de nacimiento (dd/mm/yyyy): 15/05/1996
Nombre de usuario (6-12 caracteres): AlanMar
Contraseña: Pacopedacito123*
Usuario creado correctamente.
```

Fig. 1

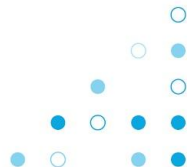
También se permite modificar a los Usuarios (**Fig. 2**) ya registrados en caso de que hubiera algún error, como también sería el caso de eliminar el usuario (**Fig.3**) en caso de que sea necesario.

```
----- Menú Principal -----
0. Menú Fidelización de Clientes
-----
1. Registrar Usuario
2. Iniciar Sesión
3. Modificar Usuario
4. Eliminar Usuario
-----
Menú Ordenar y Buscar Usuarios
-----
5. Ordenar y Buscar Usuarios
6. Mostrar Todos los Usuarios
-----
Cargar / Graficar (Registros Pluviales)
-----
7. Cargar Registros Pluviales
8. Graficar Registros Pluviales
9. Salir
-----
Seleccione una opción: 3
Ingrese el nombre de usuario o correo del usuario que desea modificar: Alanmar
Usuario encontrado. Ingrese los nuevos datos (deje en blanco si no desea modificar):
Nombre actual: Alan. Nuevo nombre: Darel
Apellido actual: Mariani. Nuevo apellido: Marini
Ingrese nueva contraseña (8 caracteres mínimo, mayúscula, número y caracter especial): AlanDarel582*
Usuario modificado correctamente.
```

Fig. 2

```
----- Menú Principal -----
0. Menú Fidelización de Clientes
-----
1. Registrar Usuario
2. Iniciar Sesión
3. Modificar Usuario
4. Eliminar Usuario
-----
Menú Ordenar y Buscar Usuarios
-----
5. Ordenar y Buscar Usuarios
6. Mostrar Todos los Usuarios
-----
Cargar / Graficar (Registros Pluviales)
-----
7. Cargar Registros Pluviales
8. Graficar Registros Pluviales
9. Salir
-----
Seleccione una opción: 4
Ingrese el nombre de usuario o correo del usuario que desea eliminar: AlanMariani@gmail.com
Usuario eliminado correctamente.
```

Fig. 3



Seguridad

Validación de Datos: En `registrar_usuario`, se utilizan múltiples validaciones para asegurar la calidad y unicidad de los datos. Por ejemplo:

Validación del DNI: Se comprueba que sea numérico y que no esté registrado previamente en el sistema.

Formato de Fecha: Se impone un formato `dd/mm/yyyy`, lo cual facilita su interpretación y análisis, especialmente en aplicaciones de bases de datos.

Nombre de Usuario: La longitud de 6-12 caracteres y la verificación de unicidad aseguran que cada usuario tenga una identidad clara y distinta.

Contraseña: La contraseña debe cumplir con ciertos criterios de seguridad (longitud mínima, uso de mayúsculas, minúsculas, números y caracteres especiales), lo que reduce el riesgo de contraseñas débiles y mejora la protección de cuentas.

Robustez y Flexibilidad del Script

Manejo de Errores: Cada función maneja errores específicos para que el usuario reciba mensajes detallados cuando algo falla.

Reutilización de Funciones: Muchas funciones (como la de conexión) se reutilizan en todo el script, lo que reduce la redundancia y permite una mejor organización.

Interactividad: Cada función guía al usuario con instrucciones claras y opciones que hacen que el script sea fácil de entender y operar.



Propósito de `iniciar_sesion`: La función permite a los usuarios acceder al sistema con un nombre de usuario y contraseña válidos, (Fig. 4) limitando los intentos de inicio de sesión para mejorar la seguridad. En caso de fallos, registra los intentos fallidos y, al alcanzar un límite, bloquea al usuario (Fig. 5).

```
----- Menú Principal -----
0. Menú Fidelización de Clientes
-----
1. Registrar Usuario
2. Iniciar Sesión
3. Modificar Usuario
4. Eliminar Usuario
-----
Menú Ordenar y Buscar Usuarios
-----
5. Ordenar y Buscar Usuarios
6. Mostrar Todos los Usuarios
-----
Cargar / Graficar (Registros Pluviales)
-----
7. Cargar Registros Pluviales
8. Graficar Registros Pluviales
9. Salir
-----
Seleccione una opción: 2
Usuario: Ianmarini
Contraseña: Mariniian123*

¡Bienvenido Ianmarini! Has ingresado a la Gestión de Base de Datos de Fidelización de Clientes.

--- Gestión de Base de Datos ---
1. Consultar Historial de Cuentas de Clientes
2. Contar Clientes por Área Geográfica
3. Listar Clientes Activos e Inactivos
4. Buscar Clientes por Mes de Nacimiento
5. Volver al Menú Principal
6. Salir de la Aplicación
Seleccione una opción: █
```

Fig. 4

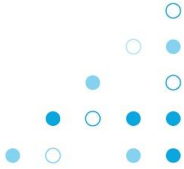
```
----- Menú Principal -----
0. Menú Fidelización de Clientes
-----
1. Registrar Usuario
2. Iniciar Sesión
3. Modificar Usuario
4. Eliminar Usuario
-----
Menú Ordenar y Buscar Usuarios
-----
5. Ordenar y Buscar Usuarios
6. Mostrar Todos los Usuarios
-----
Cargar / Graficar (Registros Pluviales)
-----
7. Cargar Registros Pluviales
8. Graficar Registros Pluviales
9. Salir
-----
Seleccione una opción: 2
Usuario: Ianmarini
Contraseña: Ianmar123*
Contraseña incorrecta. Intento 1/4
Contraseña: Ianmar123.
Contraseña incorrecta. Intento 2/4
Contraseña: Ianmar123-
Contraseña incorrecta. Intento 3/4
Contraseña: Ianmar123,
Contraseña incorrecta. Intento 4/4
Usuario bloqueado.
```

Fig. 5

Límites de Intentos en

`iniciar_sesion`: Para prevenir ataques de fuerza bruta, la función limita los intentos de inicio de sesión a cuatro. Después de eso, se registra el intento fallido y se bloquea al usuario, mitigando el riesgo de acceso no autorizado.

Registro de Intentos Fallidos: La función `iniciar_sesion` registra intentos fallidos en un log (presumiblemente para auditoría o análisis de seguridad), lo que proporciona información valiosa para identificar posibles intentos de acceso malintencionados.



```

Usuario: Ianmarini
Contraseña: Mariniian123*

¡Bienvenido Ianmarini! Has ingresado a la Gestión de Base de Datos de Fidelización de Clientes.

--- Gestión de Base de Datos ---
1. Consultar Historial de Cuentas de Clientes
2. Contar Clientes por Área Geográfica
3. Listar Clientes Activos e Inactivos
4. Buscar Clientes por Mes de Nacimiento
5. Volver al Menú Principal
6. Salir de la Aplicación
Seleccione una opción: █

```

Fig. 6

Conectar a la Base de Datos (**crear_conexion** y **conectar_db**)

crear_conexion: Permite establecer una conexión simple a la base de datos usando credenciales predeterminadas. Esta función es la base para realizar operaciones en la base de datos, y está estructurada para manejar errores de conexión, lo que garantiza que el usuario reciba un mensaje adecuado si algo falla (**Fig. 6**).

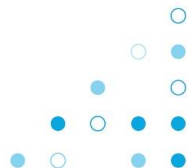
```

Intentando conectar a la base de datos...
Credenciales: {'host': 'localhost', 'user': 'root', 'password': '4466', 'database': 'd_BaseDatos_Fidelizacion_ISPC'}
Conectando al servidor MySQL...
Conexión inicial exitosa, intentando crear la base de datos...
Cursor creado, ejecutando creación de base de datos...
Base de datos 'd_BaseDatos_Fidelizacion_ISPC' creada o ya existente, usando base de datos.
Intentando reconectar a la base de datos...
Conexión exitosa a la base de datos 'd_BaseDatos_Fidelizacion_ISPC'
Cliente: Juan Pérez, Fecha Alta: 2020-01-01, Antigüedad: 48, Churn: 0
Cliente: María Gómez, Fecha Alta: 2019-06-15, Antigüedad: 54, Churn: 1

```

Fig. 7

conectar_db: Aumenta la flexibilidad de **crear_conexion** al permitir crear la base de datos si no existe y luego reconectar. Esto asegura que el script pueda inicializar la base de datos desde cero, reduciendo la dependencia de configuraciones previas (**Fig. 7**).



```

Intentando conectar a la base de datos...
Credenciales: {'host': 'localhost', 'user': 'root', 'password': '4466', 'database': 'd_BaseDatos_Fidelizacion_ISPC'}
Conectando al servidor MySQL...
Conexión inicial exitosa, intentando crear la base de datos...
Cursor creado, ejecutando creación de base de datos...
Base de datos 'd_BaseDatos_Fidelizacion_ISPC' creada o ya existente, usando base de datos.
Intentando reconectar a la base de datos...
Conexión exitosa a la base de datos 'd_BaseDatos_Fidelizacion_ISPC'
=== Crear Tabla en la Base de Datos ===
Ingresa el nombre de la tabla: fecha_cumpleaños
Ingresa las columnas en el siguiente formato:
nombre_columna1 tipo_columna1 restricciones, nombre_columna2 tipo_columna2 restricciones
Ejemplo: nombre VARCHAR(255) PRIMARY KEY, edad INT NOT NULL, comuna VARCHAR(100) DEFAULT 'Desconocida'

Ingresa las columnas separadas por comas: fecha DATE NOT NULL
Consulta SQL generada: CREATE TABLE IF NOT EXISTS fecha_cumpleaños (fecha DATE NOT NULL);
Tabla 'fecha_cumpleaños' creada exitosamente.

```

Fig. 8

```

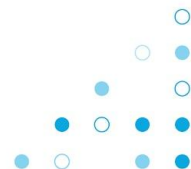
Intentando conectar a la base de datos...
Intentando conectar a la base de datos...
Credenciales: {'host': 'localhost', 'user': 'root', 'password': '4466', 'database': 'd_BaseDatos_Fidelizacion_ISPC'}
Conectando al servidor MySQL...
Conexión inicial exitosa, intentando crear la base de datos...
Cursor creado, ejecutando creación de base de datos...
Base de datos 'd_BaseDatos_Fidelizacion_ISPC' creada o ya existente, usando base de datos.
Intentando reconectar a la base de datos...
Conexión exitosa a la base de datos 'd_BaseDatos_Fidelizacion_ISPC'
=== Agregar Datos a la Tabla en la Base de Datos ===
Tablas disponibles:
areageografica
cargosllamadas
clientes
detallesllamadas
empleados
fecha_cumpleaños
historialcuentas
llamadasclientes
llamadaserviciocliente
tiposplanes
Ingresa el nombre de la tabla: fecha_cumpleaños
Columnas disponibles:
fecha (date)
¿Cuántas filas desea agregar?: 1
Ingresa dato para 'fecha' (deje vacío para NULL): 1994/09/02
Datos agregados a la tabla fecha_cumpleaños correctamente.

```

Fig. 9

crear_tabla_mysql es esencial para definir la estructura de datos. Permite crear tablas en la base de datos, asegurando que no haya duplicados y validando las columnas ingresadas por el usuario. Este enfoque garantiza que el usuario puede personalizar las tablas de acuerdo con sus necesidades (**Fig. 8**).

agregar_datos_mysql permite a los usuarios ingresar datos en una tabla específica. La función se asegura de que los datos coincidan con las columnas y permite la inserción de valores NULL, lo que facilita la entrada de datos flexibles (**Fig. 9**).



`eliminar_tabla_mysql` permite la eliminación segura de tablas que ya no son necesarias. Se verifica primero la existencia de la tabla y se informa al usuario, ayudando a prevenir errores que podrían resultar en la pérdida accidental de datos (**Fig. 10**).

```
Intentando conectar a la base de datos...
Credenciales: {'host': 'localhost', 'user': 'root', 'password': '4466', 'database': 'd_BaseDatos_Fidelizacion_ISPC'}
Conectando al servidor MySQL...
Conexión inicial exitosa, intentando crear la base de datos...
Cursor creado, ejecutando creación de base de datos...
Base de datos 'd_BaseDatos_Fidelizacion_ISPC' creada o ya existente, usando base de datos.
Intentando reconectar a la base de datos...
Conexión exitosa a la base de datos 'd_BaseDatos_Fidelizacion_ISPC'
=== Eliminar Tabla en la Base de Datos ===
Tablas disponibles:
areageografica
cargosllamadas
clientes
detallesllamadas
empleados
fecha_cumpleaños
historialcuentas
llamadasclientes
llamadasserviciocliente
tiposplanes
Ingresa el nombre de la tabla a eliminar: fecha_cumpleaños
Tabla 'fecha_cumpleaños' eliminada exitosamente.
```

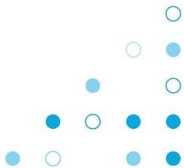
Fig. 10

`eliminar_datos_mysql` permite eliminar registros específicos de una tabla. Esta función es particularmente útil para la limpieza de datos o para eliminar información innecesaria, manteniendo la base de datos organizada (**Fig. 11**).

```
Formato: DELETE FROM fecha_cumpleaños WHERE: condicion

Ejemplos de condiciones para eliminar:
1. 'nombre_columna = 'valor'' para eliminar filas donde la columna tenga un valor específico.
2. 'edad > 30' para eliminar filas donde la edad sea mayor a 30.
3. 'comuna = 'Madrid' AND edad < 25' para eliminar filas donde la comuna sea 'Madrid' y la edad sea menor a 25.
4. 'nombre_columna IN ('valor1', 'valor2')' para eliminar filas donde la columna tenga uno de los valores especificados.
Ingresa únicamente la condición de eliminación (o deja vacío para salir): 1994/09/02
Datos eliminados exitosamente.
```

Fig. 11



Cargar Registros Pluviales

Esta función (**cargar_registros_pluviales**) se conecta al archivo CSV pertinente ("registroPluvial{año}.csv") y lista los registros de lluvias a lo largo de cada día de un mes y año especificados (**Fig. 12**).

En caso de no haber datos preexistentes a la fecha elegida, se generan de manera aleatoria (**Fig. 13**). De la misma forma, de no existir el archivo .CSV, se creará uno nuevo.

```
else:
    print("Archivo no encontrado. Generando datos aleatorios...")

    # Generar datos aleatorios para el año
    dias_por_mes = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]
    datos_pluviales = {}
    meses = [
        "Enero", "Febrero", "Marzo", "Abril", "Mayo", "Junio",
        "Julio", "Agosto", "Septiembre", "Octubre", "Noviembre", "Diciembre"
    ]

    # Generar registros pluviales aleatorios
    for i, mes in enumerate(meses):
        # Generar datos para el mes correspondiente
        registros = [random.randint(0, 100) for _ in range(dias_por_mes[i])]
        datos_pluviales[mes] = registros
```

Fig. 13

```
-----
Seleccione una opción: 7
Ingrese el año (por ejemplo, 2023): 2012
Archivo no encontrado. Generando datos aleatorios...
Datos pluviales aleatorios generados y guardados en 'registroPluvial2012.csv'.
Ingrese el mes que desea consultar (por ejemplo, Enero): Junio
Registros pluviales de Junio 2012:
0      23.0
1      61.0
2      22.0
3      78.0
4      69.0
5      100.0
6      85.0
7      59.0
8      92.0
9      42.0
10     66.0
11     56.0
12     91.0
13     51.0
14      4.0
15     16.0
16     52.0
17     66.0
18     49.0
19     15.0
20     64.0
21     33.0
22      9.0
23     64.0
24      2.0
25     56.0
26     67.0
27     41.0
28     45.0
29     38.0
30     NaN
Name: Junio, dtype: float64
-----
```

Fig. 12

Graficar Registros Pluviales

Al seleccionar la opción de graficar los registros pluviales (**Fig.14**). el menú proporcionará al usuario de tres opciones: Grafico de barras, gráfico de dispersión y gráfico circular o de “torta”.

A continuación el usuario ingresa el año a graficar y el programa procede a realizar el gráfico. En este caso se seleccionó la primera opción (**Fig. 15**) que corresponde a la función de gráfico de barras (**graficar_barras_lluvias_anuales**)

```
-----
Seleccione una opción: 8
Seleccione un tipo de gráfico:
1. Gráfico de Barras de Lluvias Anuales
2. Gráfico de Dispersión (Meses vs Días)
3. Gráfico Circular de Lluvias por Mes
Seleccione una opción de gráfico: 1
Ingrese el año para el que desea graficar las precipitaciones: 2012
-----
```

Fig. 14

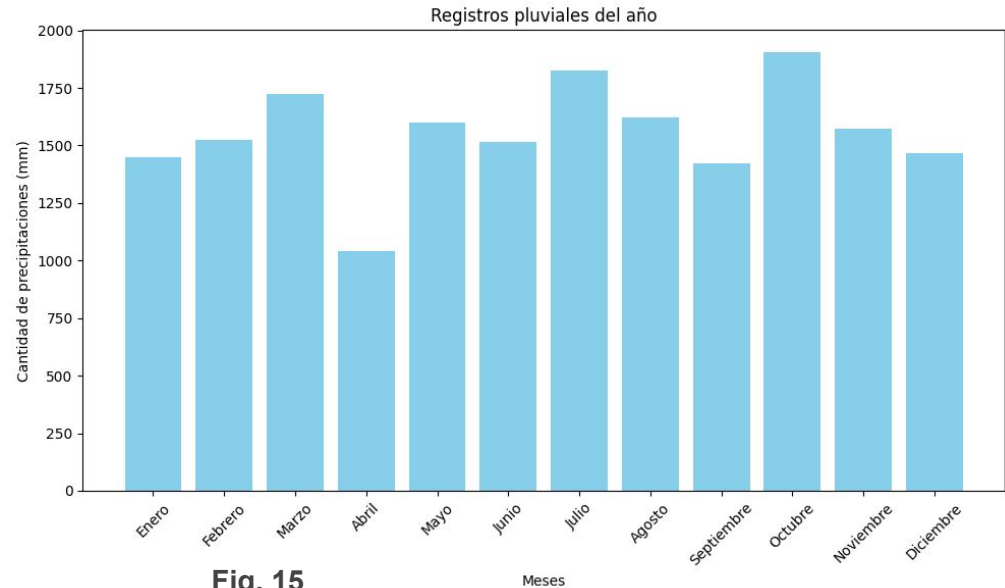
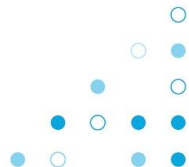


Fig. 15



Ordenamiento, Búsqueda y Muestra de Usuarios

La segunda sección del menú principal permite elegir entre mostrar todos los usuarios registrados (**mostrar_todos_los_usuarios**, Fig. 16), o acceder a un submenú (Fig. 17) en donde se podrá ordenar a los usuarios, mostrarlos o buscar a un usuario particular en base a diversos parámetros.

En este caso en particular se ejemplifica una búsqueda por DNI (**buscar_usuario_por_dni**)

```
--- Menú de Ordenamiento y Búsqueda de Usuarios ---
1. Ordenar Usuarios por Username y Guardar
2. Buscar Usuario por DNI
3. Buscar Usuario por Username
4. Buscar Usuario por Email
5. Mostrar Todos los Usuarios
6. Volver al Menú Principal
Seleccione una opción: 2
Ingrese el DNI del usuario: 123123
ID: 2, Nombre: josejoma maria, Usuario: josema, DNI: 123123
--- Menú de Ordenamiento y Búsqueda de Usuarios ---
```

Fig. 17

```
----- ¡Bienvenido a la aplicación de fidelización de clientes! -----
----- Por favor, seleccione una opción del menú para continuar. -----

----- Menú Principal -----
0. Menú Fidelización de Clientes

-----
1. Registrar Usuario
2. Iniciar Sesión
3. Modificar Usuario
4. Eliminar Usuario

-----
Menú Ordenar y Buscar Usuarios
-----
5. Ordenar y Buscar Usuarios
6. Mostrar Todos los Usuarios

-----
Cargar / Graficar (Registros Pluviales)
-----
7. Cargar Registros Pluviales
8. Graficar Registros Pluviales
9. Salir

-----
Seleccione una opción: 6

Usuarios en 'usuarios.ispc':
ID: 1, Nombre: pedre paredes, Usuario: perorades, DNI: 111112
ID: 2, Nombre: josejoma maria, Usuario: josema, DNI: 123123

Usuarios en 'usuariosOrdenadosPorUsername.ispc':
ID: 2, Nombre: josejoma maria, Usuario: josema, DNI: 123123
ID: 1, Nombre: pedre paredes, Usuario: perorades, DNI: 111112
```

Fig. 16

Base de Datos

```
40  -- Consultas con join
41  -- Listar clientes y su historial de cuentas
42  SELECT Clientes.Nombre, Clientes.Apellido, HistorialCuentas.FechaAlta, HistorialCuentas.AntiguedadCta, HistorialCuentas.Churn
43  FROM Clientes
44  INNER JOIN HistorialCuentas ON Clientes.IDCliente = HistorialCuentas.IDCliente;
45
46  -- Obtener el detalle de cargos de las llamadas de los clientes
47  SELECT Clientes.Nombre, Clientes.Apellido, CargosLlamadas.CargoDia, CargosLlamadas.CargoTarde, CargosLlamadas.CargoNoche, CargosLlamadas.CargoInternacional
48  FROM Clientes
49  INNER JOIN LlamadasClientes ON Clientes.IDCliente = LlamadasClientes.IDCliente
50  INNER JOIN CargosLlamadas ON LlamadasClientes.IDLlamada = CargosLlamadas.IDLlamada;
```

Fig. 18

Se realizaron distintas consultas avanzadas (**Fig. 18**) que fueron integradas en el backend de Python. En los scripts, trabajamos consultas complejas utilizando múltiples **JOIN**, **GROUP BY** y condiciones específicas para cada una de ellas. Estas consultas permiten listar clientes que realizaron llamadas internacionales, actualizar el estado de clientes en abandono, inactivar registros, y obtener detalles de cargos de llamadas. Además, a través de la agregación de datos, se calculan costos totales de llamadas por cliente en períodos específicos, optimizando la consulta y manipulación de datos en el sistema.

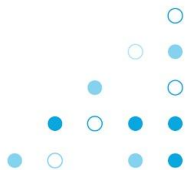
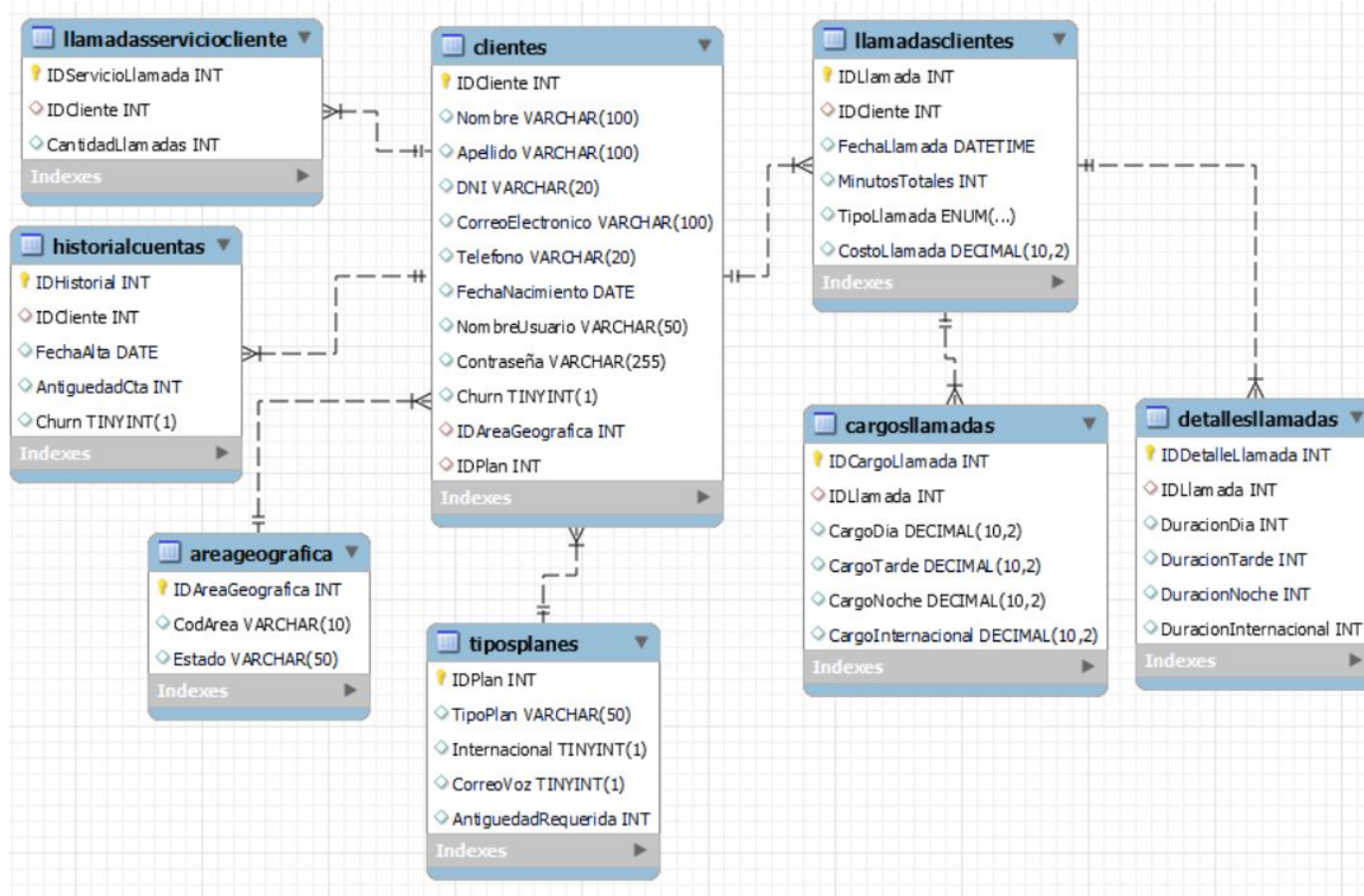


Diagrama CrowFoot



CONCLUSIONES

Este proyecto integra diferentes módulos y funcionalidades para la gestión de usuarios, datos de lluvia y una base de datos de fidelización de clientes. Aprovechando el uso de Python y sus bibliotecas, el sistema permite:

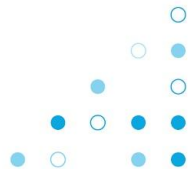
Almacenamiento y Persistencia: Manejo eficiente de datos mediante archivos binarios, MySQL.

Análisis y Visualización de Datos: Representación gráfica de registros pluviales, con gráficos de barras, de dispersión y circulares, que ayudan en el análisis de patrones de precipitación.

Gestión Integral de Usuarios: Implementación de un sistema CRUD completo que facilita la organización y búsqueda de usuarios a través de técnicas de búsqueda

Interfaz Interactiva: Un menú de navegación guiada que permite realizar consultas y operaciones en bases de datos.

En resumen, este sistema proporciona una base sólida para gestionar y analizar datos de usuarios y registros pluviales, destacándose por la eficiencia en el manejo de datos en múltiples formatos y bases de datos.



¡Muchas gracias!