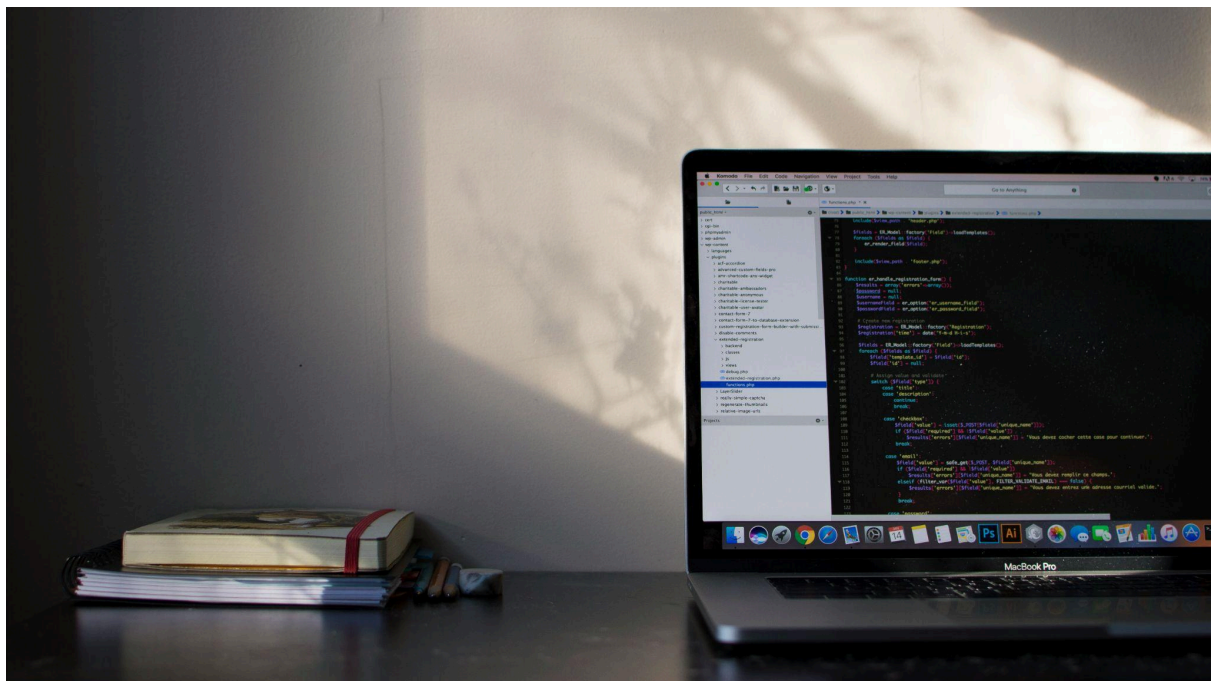


Hito 2

Sistemas de Gestión Empresarial

20 /11/2024

Sergio Vadillo Sánchez



Índice.

1. Introducción	2.
2. Descripción de la Interfaz y conexión	3 a 6.
3. Operaciones CRUD	7 a 12.
4. Consultas y Ordenación	13 y 14.
5. Exportación de datos a Excel	15 a 18.
6. Visualización de datos en gráficos	19 a 23.
7. Webgrafía	24.

1.Introducción

Este trabajo se enfoca en el desarrollo de un sistema para analizar datos sobre el consumo de alcohol y su impacto en la salud. Surge de la necesidad de gestionar, analizar y visualizar la información obtenida de encuestas a individuos sobre sus hábitos de consumo de bebidas alcohólicas y los problemas de salud relacionados. El objetivo principal es facilitar la toma de decisiones informadas mediante el uso de tecnologías modernas para el análisis y la representación de datos.

El problema radica en la falta de herramientas específicas que permitan en este caso al Dr. Fernando, manejar grandes volúmenes de datos de encuestas, analizar tendencias de consumo y su relación con diferentes indicadores de salud, y tomar decisiones basadas en evidencia. Este sistema busca abordar esta necesidad ofreciendo una solución integral que incluye almacenamiento de datos(CRUD), consultas dinámicas, generación de reportes y visualización gráfica.

Contexto

El consumo de alcohol es un tema de gran relevancia en el ámbito de la salud pública debido a su conexión con diversas afecciones, como problemas digestivos, dolores de cabeza, entre otros. Este trabajo se sitúa en el campo del análisis de datos aplicado a la salud, donde las herramientas tecnológicas no solo almacenan información, sino que también la transforman en conocimiento útil para identificar patrones de consumo peligrosos, evaluar la prevalencia de ciertos problemas de salud y diseñar estrategias de intervención más efectivas.

A través del desarrollo de este sistema, se busca integrar funciones como:

- Gestión de encuestas: Crear, actualizar y eliminar registros de individuos encuestados.
- Consultas y filtros personalizados: Ordenar los datos y aplicar filtros según necesidades específicas.
- Generación de reportes: Exportar datos en formatos accesibles como Excel.
- Visualización gráfica: Representar datos de manera clara y comprensible.

2.Descripción de la Interfaz y Conexión.

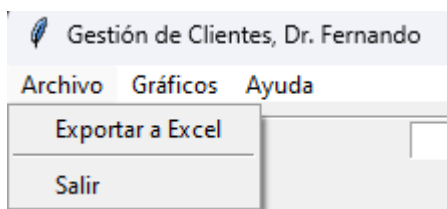
La interfaz de usuario ha sido desarrollada utilizando Tkinter, una biblioteca gráfica de Python que permite crear aplicaciones de escritorio interactivas y amigables. El sistema ofrece una experiencia intuitiva para los usuarios, quienes pueden gestionar encuestas relacionadas con el consumo de alcohol y problemas de salud de manera eficiente. Además, la aplicación está conectada a una base de datos MySQL, donde se almacenan y recuperan los datos para su análisis.

Interfaz de Usuario

Componentes Principales:

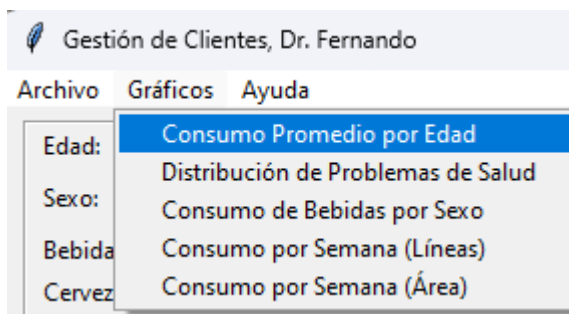
1. Menú Principal:

- Archivo:



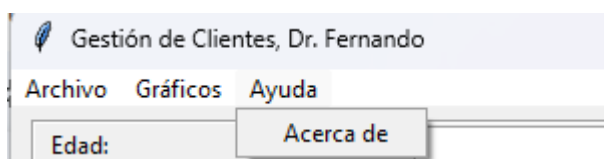
- Exportar datos a Excel: Permite descargar los datos visibles en el Treeview en formato Excel.
- Salir: Cierra la aplicación.

- Gráficos:



- Permite generar gráficos personalizados como consumo promedio por edad, distribución de problemas de salud, etc.

- Ayuda:



- Muestra información acerca del sistema y su propósito.

2. Formulario de Entrada, tabla de resultados y barra de estado:

The application window has a title bar 'Gestión de Clientes, Dr. Fernando' and a menu bar with 'Archivo', 'Gráficos', and 'Ayuda'. The main area is divided into a form and a table.

Formulario de Entrada:

- Edad:
- Sexo:
- Bebidas Semana:
- Cervezas Semana:
- Bebidas Fin de Semana:
- Bebidas Destiladas Semana:
- Vinos Semana:
- Pérdidas de Control:
- Diversión Dependencia Alcohol:
- Problemas Digestivos:
- Tensión Alta:
- Dolor de Cabeza:
- ID para Eliminar:
-

Barra de Acción:

-
-
-

Tabla de Resultados:

	ID	Edad	Sexo	Bebidas Semana	Cervezas Semana
1		57	Mujer	5	5
2		55	Mujer	6	4
3		19	Hombre	0	0
4		20	Hombre	0	0
5		21	Hombre	20	15
6		20	Hombre	1	12
7		19	Hombre	2	0
8		19	Hombre	3	5
9		22	Hombre	0	0
10		19	Hombre	2	10

Barra de Estado:

Listo - Haga clic en los encabezados de columna para ordenar.

- Sección para ingresar y editar datos de las encuestas, como edad, género, consumo semanal de bebidas, y problemas de salud.
- Botón para aplicar filtros dinámicos según criterios específicos (frecuencia de consumo, problemas de salud, etc.).
- Muestra los datos almacenados en la base de datos.
- Permite ordenar los registros al hacer clic en los encabezados de las columnas.
- Compatible con operaciones CRUD (Crear, Leer, Actualizar, Eliminar).
- Proporciona información en tiempo real sobre las acciones realizadas, como exportación de datos o éxito en una operación.

Conexión a la Base de datos

El sistema utiliza la biblioteca `mysql.connector` para conectar con una base de datos MySQL. Esta conexión permite realizar operaciones CRUD de manera eficiente.

Configuración de la Conexión:

El archivo `bdd.py` contiene el código encargado de gestionar la conexión:

```
import mysql.connector
from mysql.connector import Error

def conectar_bd():
    try:
        conexion = mysql.connector.connect(
            host="localhost",
            user="root",
            password="campusfp",
            database="ENCUESTAS",
            auth_plugin='mysql_native_password'
        )
        return conexion
    except Error as e:
        print(f"Error al conectar a MySQL: {e}")
        return None
```

Descripción:

1. Parámetros de conexión:

- `host`: Dirección del servidor (generalmente `localhost` para pruebas locales).
- `user` y `password`: Credenciales de acceso a la base de datos.
- `database`: Nombre de la base de datos que contiene la tabla de encuestas.
- `auth_plugin`: Asegura compatibilidad con el método de autenticación del servidor.

2. Gestión de errores: Si la conexión falla, se muestra un mensaje de error en la consola.

Captura de workbench

	idEncuesta	edad	Sexo	BebidasSemana	CervezasSemana	BebidasFinSemana	BebidasDestiladasSemana	VinosSemana	PerdidasControl	DiversionDependenciaAlcohol	ProblemasDigestivos	TensionAlta	DolorCabeza
▶	1	57	Mujer	5	5	5	0	3	1	No	Sí	No lo se	Alguna vez
	2	55	Mujer	6	4	5	1	2	3	Sí	Sí	No	Alguna vez
	3	19	Hombre	0	0	0	0	0	0	No	No	No lo se	Alguna vez
	4	20	Hombre	0	0	0	0	0	0	No	No	No	Nunca
	5	21	Hombre	20	15	10	5	0	12	Sí	No	No	Alguna vez
	6	20	Hombre	1	12	45	12	23	45	No	No	No	Alguna vez
	7	19	Hombre	2	0	2	2	0	3	No	No	No	Nunca
	8	19	Hombre	3	5	5	5	0	1	Sí	No	No lo se	Muy a menudo
	9	22	Hombre	0	0	0	0	0	0	No	No	No lo se	Alguna vez
	10	19	Hombre	2	10	5	1	5	2	No	No	No	Alguna vez
	11	19	Hombre	83	0	75	27	56	33	Sí	Sí	Sí	Muy a menudo
	12	21	Hombre	0	0	0	0	0	0	No	No	No	Alguna vez
	13	19	Hombre	5	5	5	5	0	1	Sí	No	No lo se	Alguna vez
	14	24	Hombre	25	18	10	7	0	365	No	No	No	Alguna vez
	15	38	Mujer	41	5	25	10	10	365	Sí	Sí	Sí	Muy a menudo
	16	53	Mujer	0	0	0	0	0	0	No	No	Sí	Alguna vez
	17	19	Hombre	4	0	1	1	1	1	No	No	No	Nunca
	18	20	Hombre	3	7	7	2	0	0	Sí	Sí	No lo se	Nunca
	19	20	Hombre	3	2	5	3	0	1	No	No	No	Alguna vez
	20	21	Mujer	2	0	2	0	2	0	No	No	No	A menudo
	21	19	Hombre	20	20	10	7	0	3	No	No	No lo se	Alguna vez
	22	20	Hombre	1	0	0	1	1	0	No	No	No	Nunca

3. Operaciones CRUD.

1. Crear:

- El usuario completa el formulario ya que hay excepciones para que se tengan que completar todos los campos si o si.
- Haces clic en “Crear” y los datos se validan para asegurar que sean correctos y completos.
- La información es enviada a la función crear_encuesta, que ejecuta una sentencia INSERT INTO.

Captura:

The screenshot shows a web application titled "Gestión de Clientes, Dr. Fernando". The interface includes a menu bar with "Archivo", "Gráficos", and "Ayuda". The main form contains the following fields and controls:

- Edad: 23
- Sexo: Hombre
- Bebidas Semana: 11
- Cervezas Semana: 5
- Bebidas Fin de Semana: 4
- Bebidas Destiladas Semana: 4
- Vinos Semana: 2
- Pérdidas de Control: 1
- Diversión Dependencia Alcohol: Sí
- Problemas Digestivos: No
- Tensión Alta: No
- Dolor de Cabeza: A menudo
- ID para Eliminar: (empty)

At the bottom of the form are buttons for "Crear", "Actualizar", and "Eliminar". A modal dialog box titled "Éxito" is overlaid on the form, displaying the message "Encuesta añadida exitosamente" and an "Aceptar" button.

Código:

```
# Crear una nueva encuesta en la base de datos
def crear_encuesta(encuesta):
    """
    Inserta una nueva encuesta en la base de datos.
    """
    conexion = conectar_bd()
    if conexion:
        cursor = conexion.cursor()
        sql = '''
            INSERT INTO ENCUESTA (edad, Sexo, BebidasSemana,
CervezasSemana,
                                BebidasFinSemana,
BebidasDestiladasSemana, VinosSemana,
                                PerdidasControl,
DiversiónDependenciaAlcohol,
                                ProblemasDigestivos, TensionAlta,
DolorCabeza)
            VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)
        '''
        valores = (encuesta.edad, encuesta.Sexo,
encuesta.BebidasSemana,
                                encuesta.CervezasSemana, encuesta.BebidasFinSemana,
encuesta.BebidasDestiladasSemana,
                                encuesta.VinosSemana, encuesta.PerdidasControl,
encuesta.DiversiónDependenciaAlcohol,
                                encuesta.ProblemasDigestivos, encuesta.TensionAlta,
encuesta.DolorCabeza)
        cursor.execute(sql, valores)
        conexion.commit()
        cursor.close()
        conexion.close()
```

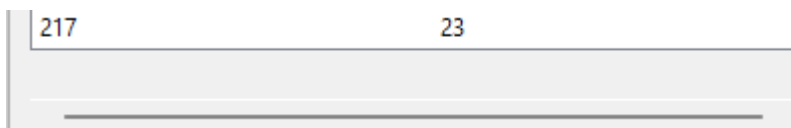
2. Leer:

- Al cargar la interfaz, la función leer_encuestas ejecuta una sentencia SQL SELECT * FROM.
- Los registros recuperados son convertidos en objetos Encuesta.
- La tabla de resultados en la interfaz se actualiza dinámicamente con los datos obtenidos.

Captura:

ID	Edad	Sexo	Bebidas Semana	Cervezas Semana
208	20	Hombre	0	3
209	21	Hombre	0	0
210	19	Hombre	5	2
211	20	Hombre	3	4
212	20	Hombre	20	15
213	20	Hombre	2	10
214	19	Hombre	69	40
215	21	Hombre	5	1
216	20	Hombre	0	0
217	23	Hombre	11	5

Con la barra se puede acceder al resto de datos



The image shows a horizontal scrollbar with a vertical line indicating the current position. Above the scrollbar, the values '217' and '23' are displayed, corresponding to the row and column selected in the table above.

Código:

```
# Leer todas las encuestas de la base de datos
def leer_encuestas():
    """
    Obtiene todas las encuestas almacenadas en la base de datos.
    """
    conexion = conectar_bd()
    encuestas = []
    if conexion:
        cursor = conexion.cursor()
        cursor.execute("SELECT * FROM ENCUESTA")
        resultados = cursor.fetchall()
        for fila in resultados:
            encuestas.append(Encuesta(*fila)) # Crear objetos Encuesta
a partir de los datos
        cursor.close()
        conexion.close()
    return encuestas
```

3. Actualizar:

- El usuario selecciona un registro en la tabla y edita los campos del formulario.
- Al hacer clic en el botón “Actualizar”, los cambios se envían a la función actualizar_encuesta, que ejecuta una sentencia SQL UPDATE.

Captura:

- Antes de actualizar:

218	23	Hombre	11	5
-----	----	--------	----	---

- Después de actualizar:

218	25	Mujer	1	1
-----	----	-------	---	---

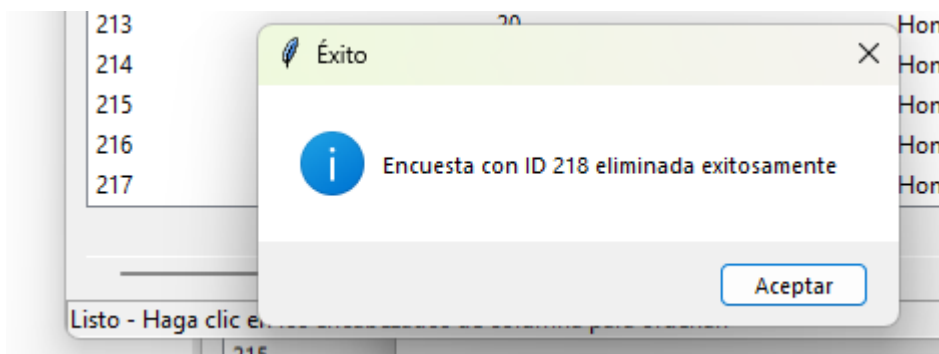
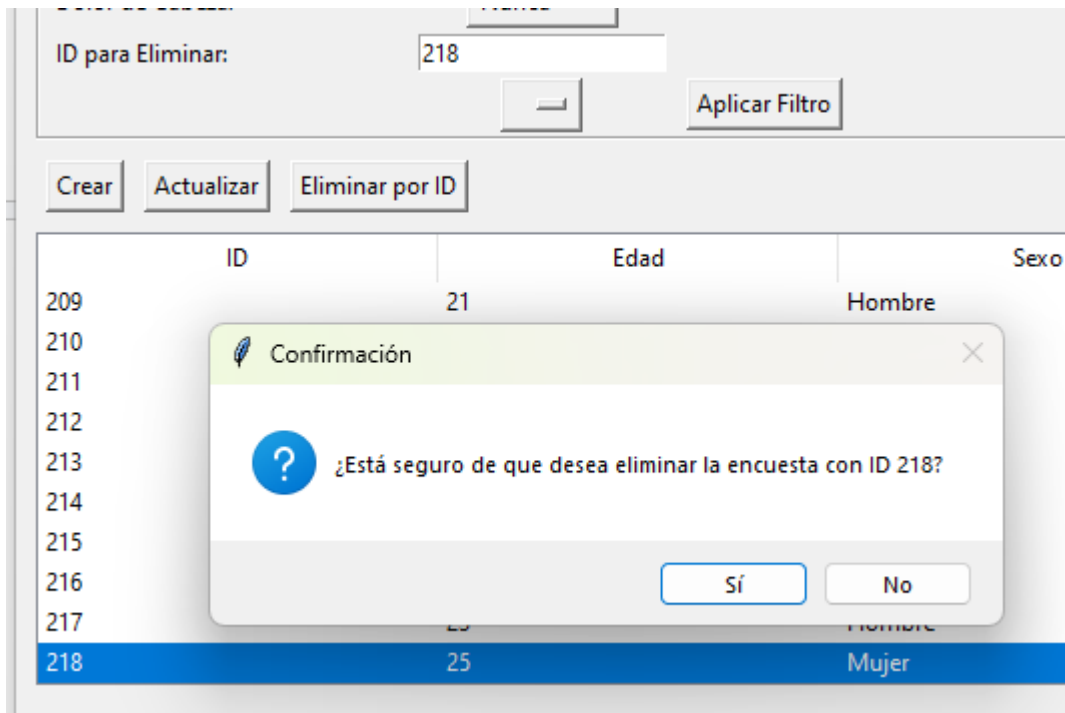
Código:

```
# Actualizar una encuesta existente
def actualizar_encuesta(encuesta):
    """
    Actualiza los datos de una encuesta existente en la base de datos.
    """
    conexion = conectar_bd()
    if conexion:
        cursor = conexion.cursor()
        sql = '''
            UPDATE ENCUESTA
            SET edad=%s, Sexo=%s, BebidasSemana=%s, CervezasSemana=%s,
            BebidasFinSemana=%s,
            BebidasDestiladasSemana=%s, VinosSemana=%s,
            PerdidasControl=%s,
            DiversionDependenciaAlcohol=%s, ProblemasDigestivos=%s,
            TensionAlta=%s, DolorCabeza=%s
            WHERE idEncuesta=%s
        '''
        valores = (encuesta.edad, encuesta.Sexo,
            encuesta.BebidasSemana, encuesta.CervezasSemana,
            encuesta.BebidasFinSemana,
            encuesta.BebidasDestiladasSemana, encuesta.VinosSemana,
            encuesta.PerdidasControl,
            encuesta.DiversionDependenciaAlcohol, encuesta.ProblemasDigestivos,
            encuesta.TensionAlta, encuesta.DolorCabeza,
            encuesta.idEncuesta)
        cursor.execute(sql, valores)
        conexion.commit()
        cursor.close()
        conexion.close()
```

4. Eliminar:

- El usuario ingresa el ID que desea eliminar
- Al hacer click en “Eliminar”, se solicita confirmación.
- Si el usuario confirma, se llama a la función eliminar_encuesta y ejecuta una sentencia SQL DELETE.

Captura:



Código:

```
# Eliminar una encuesta de la base de datos
def eliminar_encuesta(id_encuesta):
    """
    Elimina una encuesta de la base de datos utilizando su ID.
    """
    conexion = conectar_bd()
    if conexion:
        cursor = conexion.cursor()
        sql = "DELETE FROM ENCUESTA WHERE idEncuesta=%s"
        cursor.execute(sql, (id_encuesta,))
        conexion.commit()
        cursor.close()
        conexion.close()
```

4.Consultas y Ordenación de datos.

Consultas

Estas consultas permiten al usuario explorar los datos almacenados en la base de datos según diferentes criterios.

Estas consultas se realizan directamente sobre los registros cargados en el sistema y se reflejan en tiempo real en la interfaz gráfica.

Filtros Implementados

Se pueden aplicar los siguientes filtros mediante un menú desplegable en la interfaz gráfica:

- **Sin filtro:** Muestra todos los registros disponibles.
- **Alta frecuencia de consumo:** Filtra las encuestas con más de 10 bebidas consumidas semanalmente.
- **Perdió control más de 3 veces:** Muestra pacientes que reportaron pérdida de control en más de tres ocasiones.
- **Problema de salud específico:** Presenta registros de pacientes que sufrieron dolores de cabeza o presión alta.

Ejemplo de Uso de Filtros

- Selecciona "Perdió control más de 3 veces" en el menú de filtros.
- Presiona el botón "Aplicar Filtro".
- El sistema mostrará solo los registros correspondientes a pacientes que indicaron que perdieron el control +3 veces.

Perdió control +3 veces ▾		Aplicar Filtro
ID		
Vinos Semana	Pérdidas de Control	
	12	
	45	
	33	
	365	
	365	
	5	
	100	
	69	
	10	
	4	

Ordenación de datos:

La ordenación permite al usuario reorganizar los registros mostrados en el Treeview (tabla) según cualquier columna, como edad, sexo o cantidad de bebidas consumidas. Esto ayuda a identificar patrones o realizar análisis más detallados.

Funcionamiento

- El usuario puede hacer clic en el encabezado de cualquier columna del Treeview.
- Al hacer clic, los datos de esa columna se ordenan de forma ascendente.
- Si se hace clic nuevamente, el orden cambia a descendente.
- El estado de ordenación se muestra en la barra de estado al final de la ventana.

Ejemplo

Haz clic en el encabezado de la columna "Sexo".

Observa cómo los registros se reordenan, primero de un sexo y luego de otro.

Sexo	Sexo
Hombre	Mujer
Hombre	Mujer
Hombre	Mujer
Hombre	Mujer
Hombre	Mujer
Hombre	Mujer
Hombre	Mujer
Hombre	Mujer
Hombre	Mujer
Hombre	Mujer
Hombre	Mujer

Ordenando por Sexo en orden ascendente

Ordenando por Sexo en orden descendente

5. Exportación de datos a Excel.

Descripción del Proceso

El sistema incluye una funcionalidad que permite a los usuarios exportar los datos actualmente visibles en el Treeview (tabla) de la interfaz gráfica a un archivo en formato Excel. Este archivo contiene todos los registros filtrados o sin filtrar que se muestran en la tabla al momento de la exportación.

Detalles Técnicos

Implementación del Botón de Exportación

- Se implementó un botón y una opción en el menú superior llamado "Exportar a Excel".
- La exportación genera un archivo Excel con un nombre único, basado en la fecha y hora actual, asegurando que no se sobrescriben archivos existentes.

Ruta de Exportación

- Los archivos Excel generados se guardan automáticamente en una carpeta llamada exportaciones, ubicada en el directorio raíz del proyecto.
- Si esta carpeta no existe, el sistema la crea automáticamente antes de guardar el archivo.

Lógica de Exportación

- Los datos del Treeview se recopilan en forma de una lista de diccionarios.
- Se utiliza la biblioteca pandas para convertir estos datos en un DataFrame y escribirlos en un archivo Excel.

Código Clave

La función de exportación está definida de la siguiente manera:

```
# Función para exportar los datos visibles en el Treeview a un archivo Excel

def exportar_a_excel(status):
    encuestas = [tree.item(item)["values"] for item in tree.get_children()]
    df = pd.DataFrame(encuestas, columns=columns)

    # Crear la ruta de la carpeta de exportación
    carpeta_exportacion = os.path.join(os.path.abspath(os.path.dirname(__file__)), '..', 'exportaciones')

    # Imprimir la ruta para verificar si es correcta
    print(f"Ruta de exportación: {carpeta_exportacion}")

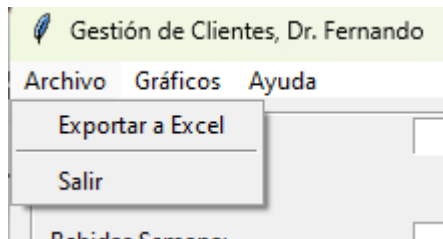
    # Verificar si la carpeta existe, si no, crearla
    try:
        if not os.path.exists(carpeta_exportacion):
            os.makedirs(carpeta_exportacion)
            print("Carpeta de exportación creada.")
    except Exception as e:
        print(f"Error al crear la carpeta de exportación: {e}")
        status.set("Error al crear la carpeta de exportación")
        messagebox.showerror("Error", f"No se pudo crear la carpeta de exportación: {str(e)}")
        return

    # Especificar la ruta completa del archivo
    archivo = os.path.join(carpeta_exportacion, f"consultas_exportadas_{datetime.now().strftime('%Y%m%d%H%M%S')}.xlsx")

    try:
        df.to_excel(archivo, index=False)
        status.set(f"Datos exportados exitosamente a {archivo}")
        messagebox.showinfo("Éxito", f"Datos exportados exitosamente a {archivo}")
    except Exception as e:
        print(f"Error al exportar a Excel: {e}")
        status.set("Error al exportar a Excel")
        messagebox.showerror("Error", f"No se pudo exportar a Excel: {str(e)}")
```

Evidencia de Funcionamiento

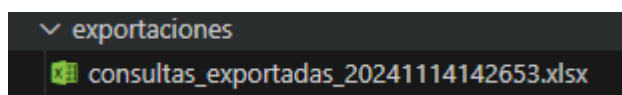
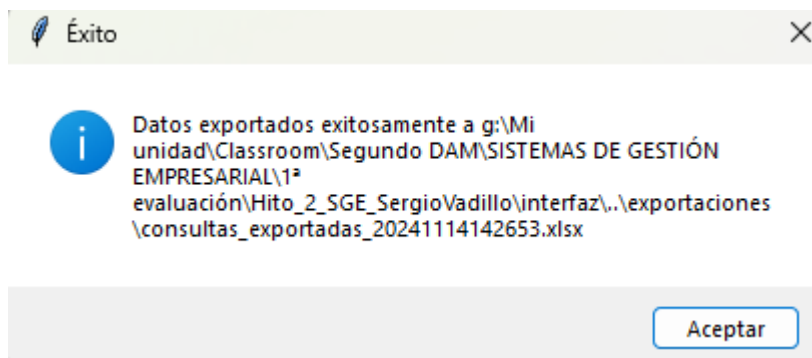
Captura del Menú de Exportación



Proceso de Exportación

Se selecciona la opción de exportación.

Aparece un mensaje de confirmación indicando que los datos se han exportado exitosamente.



Archivo Generado

consultas_exportac													
Archivo Inicio Insertar Diseño de página Fórmulas Datos Revisar Vista ¿Qué desea hacer?													
Pegar		Cortar		Copiar		Copiar formato		Portapapeles		Fuente		Alineación	
Calibri		11		A		Ajustar texto		General		Formato		Dar formato	
N		K		S		Combinar y centrar		%		000		Formato	
A1		ID											
A	B	C	D	E	F	G	H	I	J	K	L	M	N
ID	Edad	Sexo	vidas Sem	vezas Sem	Fin de S	Destiladas	nos Sem	idas de Co	Dependene	mas Dige	ensión Alt	or de Cabeza	
1	189	18 Mujer	0	0	0	0	0	0	0	No	No	No lo se	Alguna vez
2	183	19 Mujer	1	4	0	0	0	0	2	No	Sí	No	Muy a menudo
3	180	18 Mujer	1	1	1	0	0	0	0	No	No	No	Nunca
4	175	34 Mujer	0	0	0	0	0	0	0	No	Sí	No	Muy a menudo
5	161	21 Mujer	1	1	0	0	0	0	0	No	Sí	No	Alguna vez
6	147	21 Mujer	0	0	0	0	0	0	6	Sí	No	No	Nunca
7	144	62 Mujer	18	5	8	4	3	4	No	Sí	Sí	Muy a menudo	
8	143	58 Mujer	12	7	6	2	3	5	Sí	Sí	Sí	Muy a menudo	
9	142	19 Mujer	0	0	0	0	0	0	No	No	No lo se	Alguna vez	
10	136	28 Mujer	1	0	3	0	3	0	No	Sí	No	Muy a menudo	
11	133	18 Mujer	0	0	0	0	0	0	No	No	No	Alguna vez	
12	131	15 Mujer	0	0	0	0	0	0	No	No	No lo se	Alguna vez	
13	130	33 Mujer	10	6	15	10	1	2	Sí	Sí	No lo se	Alguna vez	
14	128	19 Mujer	0	5	5	0	0	0	No	No	No lo se	Nunca	
15	121	38 Mujer	0	0	0	0	2	2	No	No	No lo se	Alguna vez	
16	116	25 Mujer	5	2	5	2	0	1	No	No	No	A menudo	
17	107	15 Mujer	0	0	0	0	0	0	No	Sí	No lo se	Muy a menudo	
18	106	17 Mujer	0	0	0	0	0	0	No	Sí	No lo se	Alguna vez	
19	105	15 Mujer	0	0	0	0	0	1	No	No	No lo se	Muy a menudo	
20	104	17 Mujer	0	0	0	0	0	0	No	No	No lo se	Alguna vez	
21	99	17 Mujer	0	0	0	0	0	0	No	No	No lo se	Nunca	
22	93	17 Mujer	0	0	0	0	0	0	No	No	No	Alguna vez	
23	89	15 Mujer	0	0	0	0	0	2	No	No	No lo se	Muy a menudo	
24	86	16 Mujer	0	0	0	0	0	0	No	No	No lo se	Alguna vez	
25	84	16 Mujer	0	0	0	0	0	0	No	No	No	Alguna vez	
26	83	17 Mujer	0	0	0	0	0	0	No	No	No lo se	A menudo	
27	82	19 Mujer	0	0	0	0	0	0	No	No	No	Muy a menudo	
28	81	16 Mujer	0	0	0	0	0	1	No	No	No lo se	Muy a menudo	
29	80	16 Mujer	5	30	15	5	6	4	No	No	No lo se	Muy a menudo	
30	79	16 Mujer	0	0	0	0	0	0	No	No	No	A menudo	
31	76	47 Mujer	0	0	0	0	0	0	No	No	No lo se	Alguna vez	
32	73	38 Mujer	0	0	0	0	0	0	No	No	No	Nunca	
33	72	43 Mujer	0	0	0	0	0	0	No	No	No	Nunca	
34	71	44 Mujer	1	1	1	0	0	0	No	No	No	Alguna vez	
35	70	45 Mujer	0	0	0	0	0	0	No	No	No	Alguna vez	
36	69	44 Mujer	0	0	2	0	0	1	No	Sí	No	Alguna vez	
37	68	30 Mujer	2	0	2	0	2	1	No	No	No lo se	Alguna vez	

6. Visualización de Datos en Gráficos.

Visualización de Datos en Gráficos

Descripción de la Funcionalidad

El sistema incorpora una funcionalidad de visualización gráfica que permite a los usuarios analizar los datos recopilados mediante representaciones visuales. Los gráficos se generan en base a los datos disponibles en el sistema y están diseñados para ofrecer información clara y directa sobre los patrones y tendencias.

Tipos de Gráficos Implementados

Consumo Promedio por Edad

- Tipo de Gráfico: Gráfico de Barras.
- Descripción: Muestra el consumo promedio de bebidas alcohólicas por grupos de edad. Ideal para identificar tendencias de consumo relacionadas con la edad.
- Ejemplo de Código:

```
# Generar el gráfico según el tipo seleccionado
if tipo_grafico == "Consumo Promedio por Edad":
    df['Edad'] = df['Edad'].astype(int)
    df.groupby('Edad')['Bebidas
Semana'].mean().plot(kind='bar')
    plt.title("Consumo Promedio de Bebidas por Grupo de Edad")
    plt.xlabel("Edad")
    plt.ylabel("Consumo Promedio de Bebidas")
```

Distribución de Problemas de Salud

- Tipo de Gráfico: Gráfico de Pastel.
- Descripción: Representa la distribución de problemas de salud reportados por los encuestados, como dolores de cabeza o presión alta.
- Ejemplo de Código:

```
elif tipo_grafico == "Distribución de Problemas de Salud":
    problemas_salud = ['Problemas Digestivos', 'Tensión Alta',
'Dolor de Cabeza']
    problemas = df[problemas_salud].apply(lambda x: (x ==
"Sí").sum())
    problemas.plot(kind='pie', autopct='%1.1f%%')
    plt.title("Distribución de Problemas de Salud")
    plt.ylabel("")
```

Consumo de Bebidas por Sexo

- Tipo de Gráfico: Gráfico de Barras.
- Descripción: Compara el consumo promedio de bebidas alcohólicas entre hombres y mujeres.
- Ejemplo de Código:

```
elif tipo_grafico == "Consumo de Bebidas por Sexo":  
    df.groupby('Sexo')['Bebidas  
Semana'].mean().plot(kind='bar', color=['blue', 'orange'])  
    plt.title("Consumo Promedio de Bebidas por Sexo")  
    plt.xlabel("Sexo")  
    plt.ylabel("Consumo Promedio de Bebidas")
```

Consumo por Semana (Líneas)

Tipo de Gráfico: Gráfico de Líneas.

Descripción: Representa las tendencias de consumo de bebidas a lo largo del tiempo o por encuesta.

Ejemplo de Código:

```
elif tipo_grafico == "Consumo por Semana (Líneas)":  
    df['Bebidas Semana'] = df['Bebidas Semana'].astype(int)  
    df['Cervezas Semana'] = df['Cervezas Semana'].astype(int)  
    df[['Bebidas Semana', 'Cervezas Semana']].plot(kind='line')  
    plt.title("Consumo por Semana (Líneas)")  
    plt.xlabel("Encuestas")  
    plt.ylabel("Consumo")
```

Consumo por Semana (Área)

Tipo de Gráfico: Gráfico de Área.

Descripción: Similar al gráfico de líneas, pero con un área sombreada debajo para resaltar el volumen de datos.

Ejemplo de Código:

```
elif tipo_grafico == "Consumo por Semana (Área)":  
    df['Bebidas Semana'] = df['Bebidas Semana'].astype(int)  
    df['Cervezas Semana'] = df['Cervezas Semana'].astype(int)  
    df[['Bebidas Semana', 'Cervezas Semana']].plot(kind='area',  
alpha=0.4)  
    plt.title("Consumo por Semana (Área)")  
    plt.xlabel("Encuestas")  
    plt.ylabel("Consumo")
```

Personalización de los Gráficos

Títulos y Etiquetas: Cada gráfico incluye un título descriptivo, etiquetas para los ejes, y unidades que facilitan la interpretación de los datos.

Colores y Estilos: Se han seleccionado colores específicos para mejorar la distinción visual. Por ejemplo, tonos diferenciados para comparar entre categorías como "Hombres" y "Mujeres".

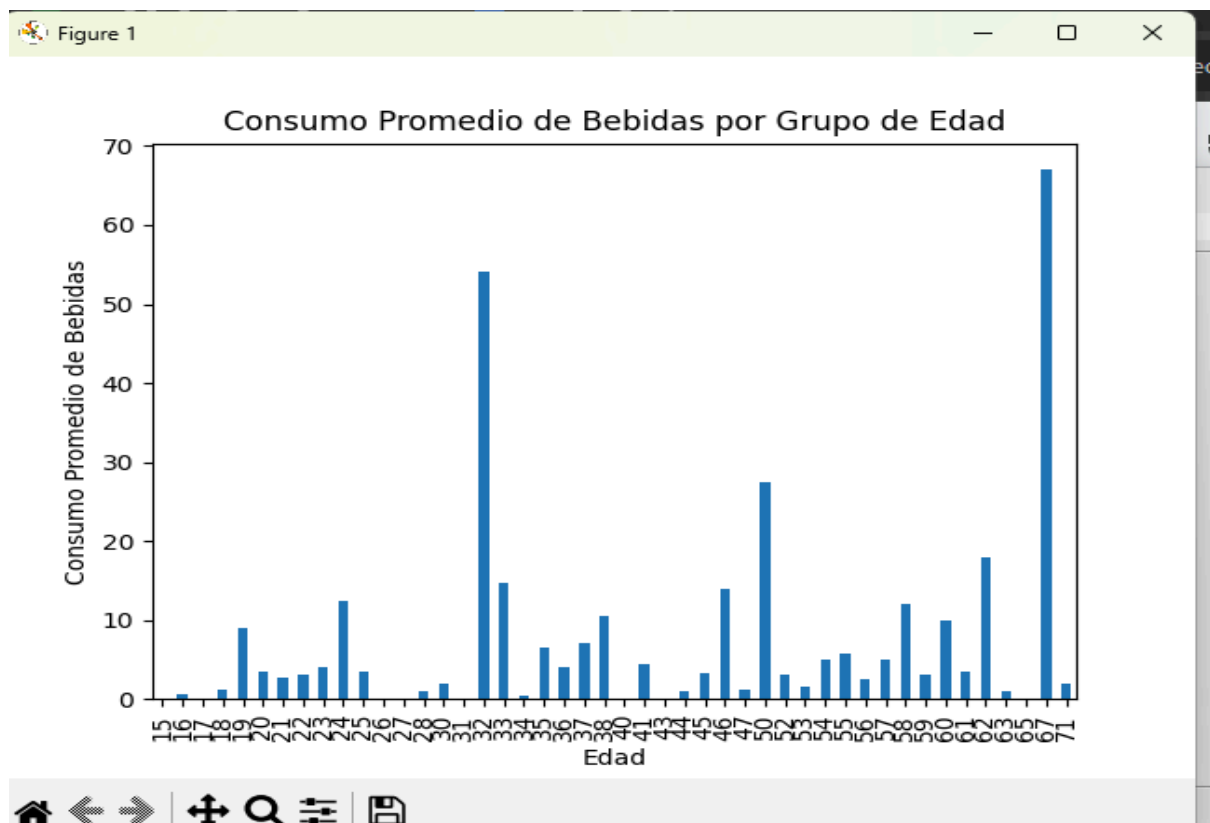
Interactividad: Los gráficos se generan dinámicamente en función de los datos seleccionados, permitiendo análisis personalizados.

Cómo Generar los Gráficos

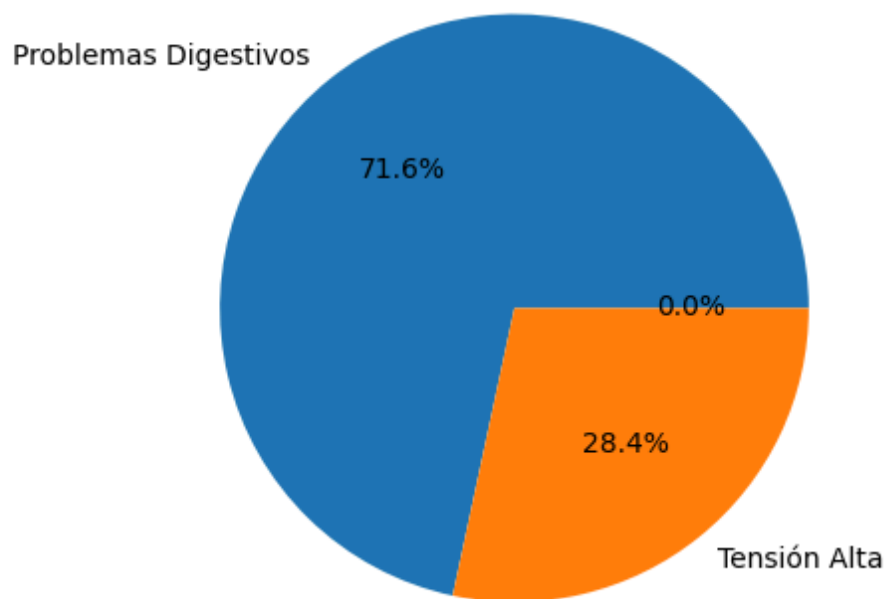
- Seleccione el tipo de gráfico deseado desde el menú superior o desde el menú contextual.
- La aplicación procesará los datos del Treeview y generará el gráfico solicitado.
- El gráfico aparecerá en una ventana separada utilizando la biblioteca matplotlib.

Evidencia de Funcionamiento

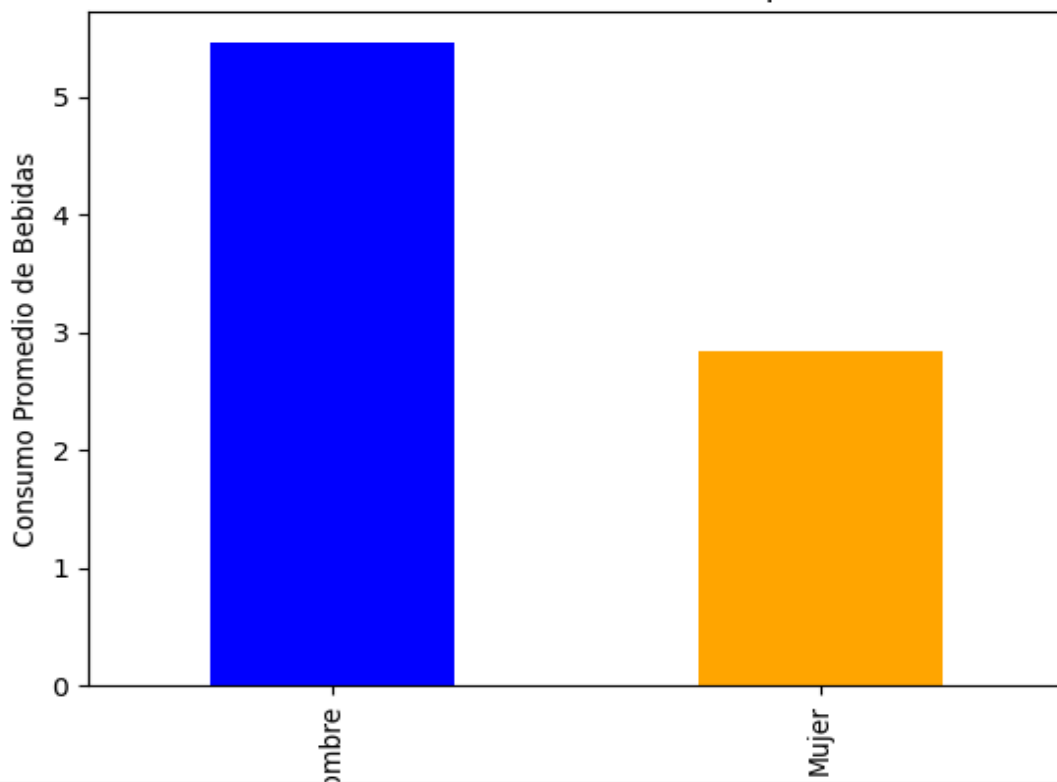
Capturas:

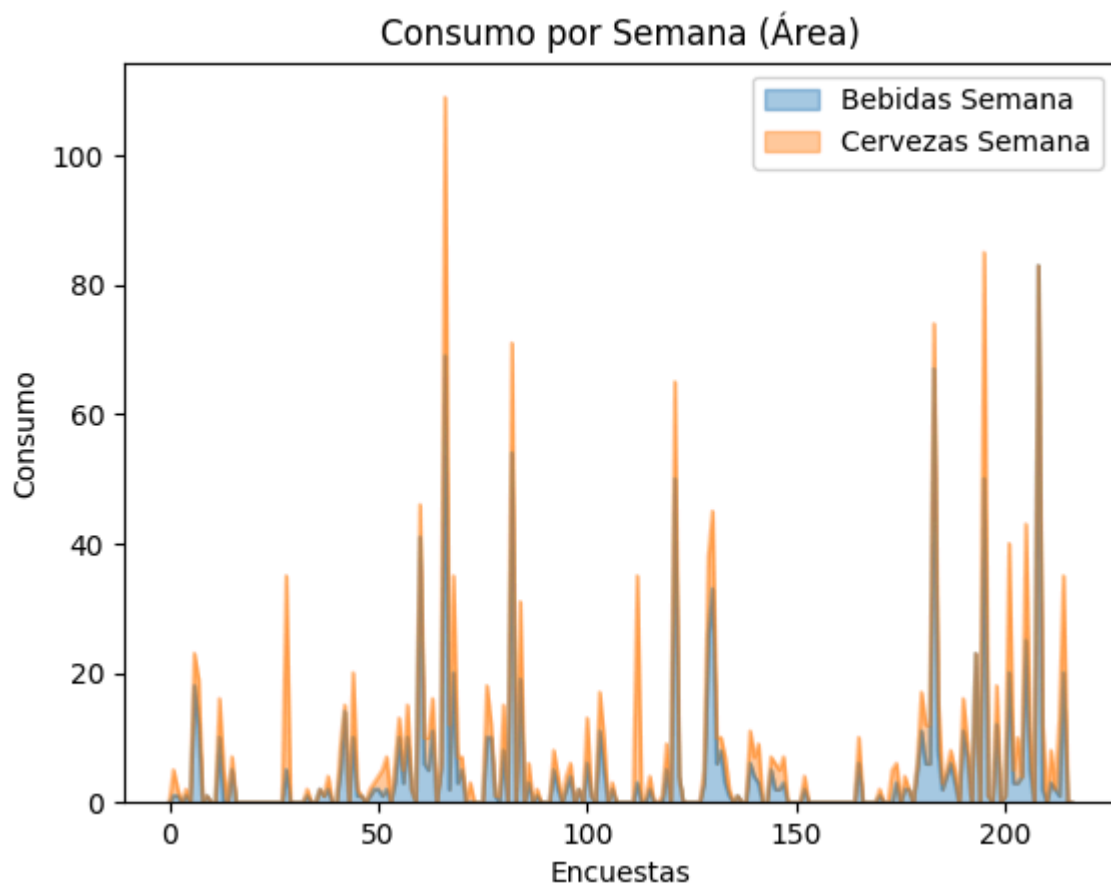
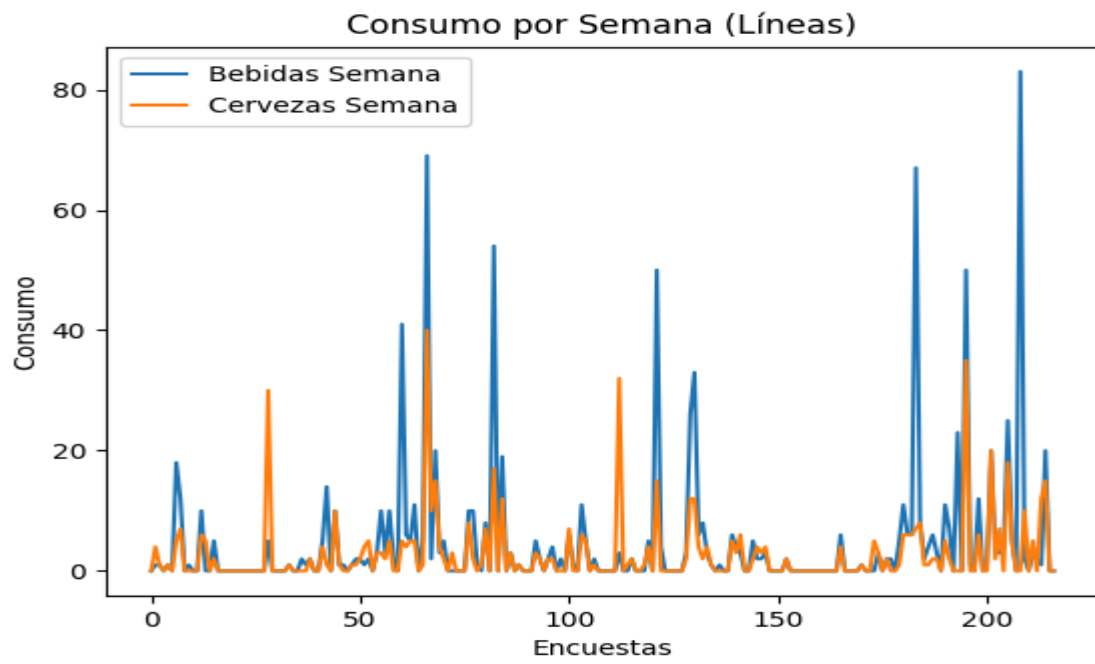


Distribución de Problemas de Salud



Consumo Promedio de Bebidas por Sexo





7.Webgrafía

References

Authentication plugin “caching_sha2_password” is not supported. (n.d.).

Stack Overflow. Retrieved November 14, 2024, from

<https://stackoverflow.com/questions/50557234/authentication-plugin-caching-sha2-password-is-not-supported>

Buscar un ítem en un treeview (tkinter). (n.d.). Recursospython.com.

Retrieved November 14, 2024, from

<https://foro.rekursospython.com/Thread-Buscar-un-%C3%ADtem-en-un-treeview-tkinter?pid=2984>

Matplotlib error - no module named tkinter. (n.d.). Stack Overflow. Retrieved

November 14, 2024, from

<https://stackoverflow.com/questions/36327134/matplotlib-error-no-module-named-tkinter>

Mysql-connector-python. (n.d.). PyPI. Retrieved November 14, 2024, from

<https://pypi.org/project/mysql-connector-python/>

Welcome to. (n.d.). Python.org. Retrieved November 14, 2024, from

<https://www.python.org/doc/>