

The background is a light blue grid with various geometric shapes. There are large triangles in shades of green, blue, and black, some with white outlines. Smaller triangles in similar colors are scattered throughout. A large, faint white triangle is also visible in the center.

HITO 2 1T Diego Huerta Arias

Fecha límite: 20/11/2024

Índice

| | |
|----------------------------------------------------|--------|
| <u>Introducción</u> | pág.2 |
| <u>Descripción de la Interfaz y Conexión</u> | pág.3 |
| <u>Operaciones CRUD</u> | pág.4 |
| <u>Exportación de Resultados a Excel</u> | pág.8 |
| <u>Visualización de Datos en Gráficos</u> | pág.10 |
| <u>Código en funcionamiento</u> | pág.11 |

Introducción

En el ámbito del análisis de datos, la recopilación y gestión de información relacionada con el consumo de alcohol y su impacto en la salud representan un desafío clave para instituciones médicas y sociales. El objetivo de esta aplicación es proporcionar una herramienta funcional que permita a profesionales, como médicos o investigadores, registrar, gestionar y analizar datos derivados de encuestas de consumo de alcohol y salud.

El desarrollo de esta solución desarrollada por mi considera varios aspectos esenciales:

- **Recopilación de datos estructurados:** Los datos de encuestas relacionadas con variables demográficas (edad, sexo) y de consumo (tipos y cantidad de bebidas) se capturan mediante una interfaz gráfica intuitiva.
- **Gestión eficiente de registros:** Incluye funcionalidades CRUD (Crear, Leer, Actualizar y Eliminar), lo que permite mantener una base de datos actualizada.
- **Visualización y análisis de resultados:** Herramientas para generar gráficos y exportar información a formatos como Excel facilitan el análisis de los datos recopilados, promoviendo decisiones basadas en evidencia.
- **Conexión con bases de datos relacionales:** La aplicación utiliza MySQL como sistema de almacenamiento, asegurando la persistencia y la integridad de los datos.

En términos de **programación**, el proyecto combina varios módulos y tecnologías clave:

1. **Interfaz gráfica de usuario (GUI):** Utilizando Tkinter para crear una aplicación visual y accesible.
2. **Gestión de base de datos:** La biblioteca pymysql maneja la conexión y las consultas a una base de datos MySQL.
3. **Exportación y análisis de datos:** openpyxl se emplea para manejar archivos Excel, mientras que matplotlib se utiliza para visualizar los datos a través de gráficos.
4. **Organización modular y orientación a objetos:** El código está estructurado en clases y métodos, facilitando su escalabilidad y mantenimiento.

Descripción de la Interfaz y Conexión

1. Ventana Principal:

- Contiene el menú principal con opciones para gestionar encuestas, visualizar resultados y exportar datos.
- Acceso a secciones como:
 - **Registro de personas que han realizado la encuesta:** Permite agregar nuevos individuos con datos relevantes (nombre, edad, género, etc.).
 - **Encuestas de Consumo:** Registro de las respuestas de encuestas relacionadas con el consumo de alcohol.

El código del menú principal:

```
# Menú
menu_barra = tk.Menu(raiz)
raiz.config(menu=menu_barra)

menu_archivo = tk.Menu(menu_barra, tearoff=0)
menu_barra.add_cascade(label="Archivo", menu=menu_archivo)
menu_archivo.add_command(label="Salir", command=raiz.quit)
```

2. Operaciones CRUD:

- **Crear:** Inserción de nuevos pacientes o encuestas con validaciones previas.
- **Leer:** Consultas filtradas según criterios como edad, consumo, o género.
- **Actualizar:** Modificación de registros existentes.
- **Eliminar:** Eliminación controlada de datos con confirmaciones.

El código del CRUD:

3. Conexión al Servidor:

- Esta línea de código lo que hace es recoger las variables para conectar a la base de datos y las ejecuta.

```
def obtener_conexion_db(): 6 usages  ⬆ diego huerta arias
    return pymysql.connect(host="localhost", user="root", password="curso", database="encuestas")
```

Operaciones CRUD

Create:

```
def agregar_registro_db(id, edad, sexo, bebidas, cervezas, bebidas_fs, bebidas_dest, vinos, perdidas, diversion, 1 usage, 1 diego huerta arias,
                        digestivos, tension, dolor):
    try:
        conn = obtener_conexion_db()
        cursor = conn.cursor()
        query = """
        INSERT INTO ENCUESTA (idEncuesta, edad, Sexo, BebidasSemana, CervezasSemana, BebidasFinSemana,
                               BebidasDestiladasSemana, VinosSemana, PerdidasControl, DiversionDependenciaAlcohol,
                               ProblemasDigestivos, TensionAlta, DolorCabeza)
        VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)
        """
        cursor.execute(query, args=(id, edad, sexo, bebidas, cervezas, bebidas_fs, bebidas_dest, vinos, perdidas, diversion,
                                    digestivos, tension, dolor))
        conn.commit()
    except Exception as e:
        print(f"Error al agregar el registro: {e}")
    finally:
        conn.close()
```

Conexión a la base de datos: El código establece una conexión con la base de datos mediante una función que no está definida en el fragmento, la cual devuelve una conexión activa.

Creación de un cursor: Una vez conectados a la base de datos, se crea un cursor. El cursor es un objeto que permite ejecutar sentencias SQL en la base de datos y manejar los resultados.

Definición de la consulta SQL: Este cacho de código prepara una consulta SQL de tipo INSERT para agregar un nuevo registro en la tabla ENCUESTA. Esta consulta tiene varios campos (como idEncuesta, edad, Sexo, etc.), y se espera que esos valores sean proporcionados como parámetros de la función para insertarlos en la base de datos.

Read:

```
def obtener_registros_db(): 3 usages  diego huerta arias
    try:
        conn = obtener_conexion_db()
        cursor = conn.cursor()
        cursor.execute("SELECT * FROM ENCUESTA")
        registros = cursor.fetchall()
        return registros
    except Exception as e:
        print(f"Error al obtener los registros: {e}")
        return []
    finally:
        conn.close()
```

La función empieza con un bloque try, que indica que el código dentro de este bloque intentará ejecutarse. Si ocurre algún error en el proceso, el control pasará al bloque except.

Se conecta a la base de datos mediante una llamada a la función de obtener_conexion_db().

Se crea un cursor con conn.cursor(). El cursor es un objeto que se utiliza para ejecutar consultas SQL en la base de datos y manejar los resultados. Con el cursor, se ejecuta la consulta SQL SELECT * FROM ENCUESTA. El comando SELECT * significa que se están obteniendo todas las filas (registros) de la tabla ENCUESTA y todas sus columnas. Todo esto es llamado en el boton

Update:

```
def actualizar_registro_db(id, edad, sexo, bebidas, cervezas, bebidas_fs, bebidas_dest, vinos, perdidas, diversion, 1 usage, 1 diego huerta arias,
    digestivos, tension, dolor):
    try:
        conn = obtener_conexion_db()
        cursor = conn.cursor()
        query = """
            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
        """
        cursor.execute(query, [args: (edad, sexo, bebidas, cervezas, bebidas_fs, bebidas_dest, vinos, perdidas, diversion,
            digestivos, tension, dolor, id)])
        conn.commit()
    except Exception as e:
        print(f"Error al actualizar el registro: {e}")
    finally:
        conn.close()
```

Conexión a la base de datos: Se establece una conexión a la base de datos mediante la función `obtener_conexion_db()`.

Ejecutar la consulta: Se ejecuta la consulta utilizando el método `cursor.execute()`, donde los valores que van a actualizarse son proporcionados como parámetros. Estos valores se toman de los argumentos de la función (por ejemplo, edad, sexo, bebidas, etc.). Además, se incluye id al final de la consulta para identificar qué registro debe ser actualizado (aquel con el idEncuesta que coincide con el valor de id).

Confirmación de la actualización: Después de ejecutar la consulta, `conn.commit()` asegura que los cambios sean guardados en la base de datos. Si no se llama a `commit()`, los cambios no se aplicarán.

Manejo de errores: Si ocurre algún error en el proceso (por ejemplo, problemas con la conexión o con la consulta), se captura con `except` y se imprime un mensaje de error. Esto permite identificar y solucionar problemas más fácilmente.

Delete:

```
def eliminar_registro_db(id): 1 usage  👤 diego huerta arias
    try:
        conn = obtener_conexion_db()
        cursor = conn.cursor()
        cursor.execute( query: "DELETE FROM ENCUESTA WHERE idEncuesta = %s", args: (id,))
        conn.commit()
    except Exception as e:
        print(f"Error al eliminar el registro: {e}")
    finally:
        conn.close()
```

Conexión a la base de datos: Se establece una conexión a la base de datos mediante la función `obtener_conexion_db()`, que se asume que devuelve una conexión válida a la base de datos.

Consulta SQL: Se ejecuta una consulta DELETE para eliminar un registro de la tabla ENCUESTA. La consulta utiliza una cláusula WHERE para especificar qué registro debe ser eliminado. En este caso, el registro se identifica por su campo `idEncuesta`, que debe coincidir con el valor de `id` proporcionado como argumento de la función.

Confirmación de la eliminación: Después de ejecutar la consulta, `conn.commit()` guarda los cambios realizados en la base de datos, asegurando que el registro se elimine de manera permanente.

Manejo de errores: Si ocurre algún error durante la ejecución de la consulta, se captura mediante el bloque `except`. En caso de error, se imprime un mensaje detallado que puede ayudar a diagnosticar el problema.

Exportación de Resultados a Excel

```
def crear_actualizar_excel(self): 1 usage  ▲ diegohuertaarias +1
    try:
        # Solicitar criterios de filtro
        filtro_edad = simpledialog.askstring( title: "Filtro", prompt: "Ingrese la edad máxima (o deje en blanco para todos):")
        filtro_sexo = simpledialog.askstring( title: "Filtro", prompt: "Ingrese el sexo (o deje en blanco para todos):")
        filtro_bebidas = simpledialog.askstring( title: "Filtro",
                                                prompt: "Ingrese el número máximo de bebidas semanales (o deje en blanco para todos):")

        registros = obtener_registros_db()
        if not registros:
            messagebox.showwarning( title: "Sin Registros", message: "No se encontraron registros en la base de datos.")
            return

        # Filtrar registros
        registros_filtrados = []
        for registro in registros:
            edad_valida = not filtro_edad or (registro[1] is not None and int(registro[1]) <= int(filtro_edad))
            sexo_valido = not filtro_sexo or (
                registro[2] is not None and registro[2].lower() == filtro_sexo.lower()
            )
            bebidas_validas = not filtro_bebidas or (
                registro[3] is not None and int(registro[3]) <= int(filtro_bebidas)
            )

            if edad_valida and sexo_valido and bebidas_validas:
                registros_filtrados.append(registro)

        if not registros_filtrados:
            messagebox.showwarning( title: "Sin Registros", message: "No se encontraron registros que coincidan con los filtros.")
            return

        # Crear o cargar el archivo Excel
        try:
            wb = openpyxl.load_workbook("registros.xlsx")
            ws = wb.active
        except FileNotFoundError:
```

```
            wb = openpyxl.Workbook()
            ws = wb.active
            ws.append(["ID", "Edad", "Sexo", "Bebidas Semanales", "Cervezas Semanales", "Bebidas Fin de Semana",
                    "Bebidas Destiladas Semana", "Vinos Semana", "Pérdidas Control", "Diversión Dependencia",
                    "Problemas Digestivos", "Tensión Alta", "Dolor Cabeza"])

        # Limpiar las filas existentes
        ws.delete_rows( idx: 2, ws.max_row)

        # Añadir los registros filtrados
        for registro in registros_filtrados:
            ws.append(registro)

        # Guardar el archivo
        wb.save("registros.xlsx")
        messagebox.showinfo( title: "Éxito", message: "Archivo Excel creado/actualizado exitosamente")

    except Exception as e:
        messagebox.showerror( title: "Error", message: f"Error al crear/actualizar el archivo Excel: {e}")
        print(f"Error al crear/actualizar el archivo Excel: {e}")
```

Obtener Registros de la Base de Datos:

Se obtienen todos los registros de la base de datos utilizando la función `obtener_registros_db()`. Si no se encuentran registros, se muestra una advertencia y se termina la ejecución de la función.

Preparar Datos para el Gráfico:

Se extraen las edades y el número de bebidas semanales de los registros obtenidos.

Crear y Mostrar el Gráfico:

Se crea un gráfico de dispersión utilizando `matplotlib`.

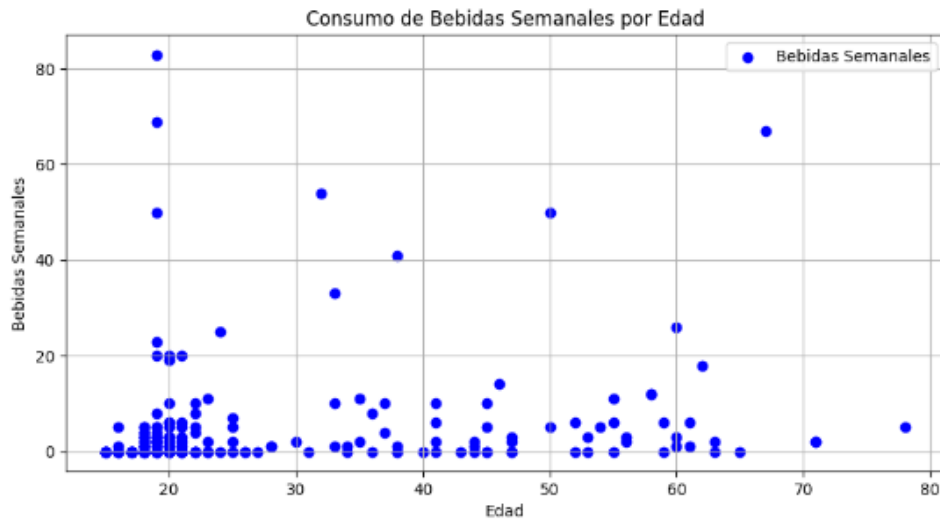
Se configuran las etiquetas de los ejes, el título del gráfico y la leyenda.

Se muestra el gráfico.

Manejo de Errores:

Si ocurre algún error durante el proceso, se muestra un mensaje de error y se imprime el error en la consola.

Visualización de Datos en Gráficos



Ejes del Gráfico:

Eje X: Edad de las personas.

Eje Y: Cantidad de bebidas consumidas semanalmente.

Concentración de Puntos:

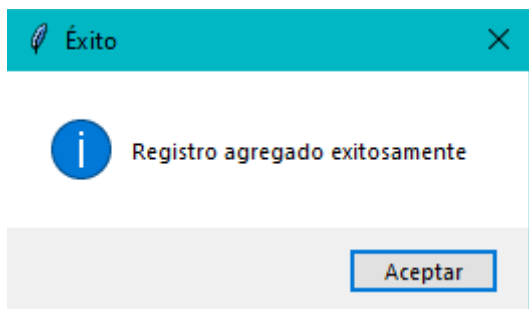
Se observa una mayor concentración de consumo en personas jóvenes, especialmente alrededor de los 20 años. Esto sugiere que el consumo de bebidas alcohólicas es más frecuente en este grupo de edad.

Valores Atípicos:

Algunos individuos en diferentes edades presentan un consumo de bebidas muy alto, lo que indica la presencia de consumidores con hábitos de bebida significativamente mayores que la media.

Este gráfico ayuda a identificar patrones de consumo de alcohol en diferentes personas de todo tipo de edades. En este podemos ver la concentración de consumo en jóvenes. En resumen este grafico podría ayudar a concienciar a gente que quizás no ve cuanto consume alcohol y ver que parte de la sociedad ingiere mas o menos.

Agregamos los datos por celda

[illegible]

Comprobación en la base de datos y en la app.

App:

| | | | | | | |
|-----|----|--------|---|---|---|---|
| 217 | 18 | ivujer | 3 | 3 | 3 | 0 |
| 218 | 50 | Hombre | 5 | 0 | 5 | 2 |

Base de datos:

| Result Grid | | | | | | | | | | | | |
|-------------|------------|---------|---------|---------------|----------------|------------------|-------------------------|-------------|-----------------|-----------|--------------------|---------------------|
| | idEncuesta | edad | Sexo | BebidasSemana | CervezasSemana | BebidasFinSemana | BebidasDestiladasSemana | VinosSemana | PerdidasControl | Diversion | DependenciaAlcohol | ProblemasDigestivos |
| 209 | 21 | Hombre | 0 | 0 | 0 | 0 | 2 | 0 | No | | | |
| 210 | 19 | Hombre | 5 | 2 | 3 | 2 | 0 | 2 | No | | | |
| 211 | 20 | Hombre | 3 | 4 | 3 | 2 | 0 | 0 | No | | | |
| 212 | 20 | Hombre | 20 | 15 | 10 | 8 | 0 | 100 | Sí | | | |
| 213 | 20 | Hombre | 2 | 10 | 15 | 20 | 5 | 4 | No | | | |
| 214 | 19 | Hombre | 69 | 40 | 30 | 8 | 12 | 69 | Sí | | | |
| 215 | 21 | Hombre | 5 | 1 | 1 | 1 | 0 | 2 | No | | | |
| 216 | 20 | Hombre | 0 | 0 | 0 | 0 | 0 | 0 | No | | | |
| 217 | 78 | Mujer | 5 | 5 | 5 | 0 | 3 | 1 | No | | | |
| 218 | 50 | Hombre | 5 | 0 | 5 | 2 | 2 | 0 | no | | | |
| * Hombres | Hombres | Hombres | Hombres | Hombres | Hombres | Hombres | Hombres | Hombres | Hombres | Hombres | Hombres | Hombres |

Actualizamos el registro que acabamos de crear:

Archivo

Edad:

45

Sexo:

Hombre

Bebidas Semanales:

5

Cervezas Semanales:

0

Bebidas Fin de Semana:

5

Bebidas Destiladas Semana:

2

Vinos Semana:

2

Pérdidas Control:

0

Diversión Dependencia:

no

Problemas Digestivos:

no

Tensión Alta:

si

Dolor Cabeza:

si

Agregar Registro

Ver Registros

Actualizar Registro

Eliminar Registro

Crear/Actualizar Excel

Mostrar Gráficos

| | Id | Edad | Sexo | Bebidas | Cervezas | Bebidas_fs | Bebidas_dest |
|-----|----|--------|------|---------|----------|------------|--------------|
| 209 | 21 | Hombre | 0 | 0 | 0 | 0 | |
| 210 | 19 | Hombre | 5 | 2 | 3 | 2 | |
| 211 | 20 | Hombre | 3 | 4 | 3 | 2 | |
| 212 | 20 | Hombre | 20 | 15 | 10 | 8 | |
| 213 | 20 | Hombre | 2 | 10 | 15 | 20 | |
| 214 | 19 | Hombre | 69 | 40 | 30 | 8 | |
| 215 | 21 | Hombre | 5 | 1 | 1 | 1 | |
| 216 | 20 | Hombre | 0 | 0 | 0 | 0 | |
| 217 | 78 | Mujer | 5 | 5 | 5 | 0 | |
| 218 | 50 | Hombre | 5 | 0 | 5 | 2 | |

Éxito

i

Registro actualizado exitosamente

Aceptar

Comprobación en la base de datos y en la app.

App:

| | | | | | | |
|-----|----|--------|---|---|---|---|
| 218 | 45 | Hombre | 5 | 0 | 5 | 2 |
|-----|----|--------|---|---|---|---|

Base de datos:

| Result Grid | | | | | | | | | | | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |
|-------------|------|--------|---------------|----------------|------------------|-------------------------|-------------|-----------------|-----------------------------|---------------------|--------------|-------|----------------|--------------------|
| idEncuesta | edad | Sexo | BebidasSemana | CervezasSemana | BebidasFinSemana | BebidasDestiladasSemana | VinosSemana | PerdidasControl | DiversiónDependenciaAlcohol | ProblemasDigestivos | | | | |
| 209 | 21 | Hombre | 0 | 0 | 0 | 0 | 2 | 0 | No | No | | | | |
| 210 | 19 | Hombre | 5 | 2 | 3 | 2 | 0 | 2 | No | No | | | | |
| 211 | 20 | Hombre | 3 | 4 | 3 | 2 | 0 | 0 | No | No | | | | |
| 212 | 20 | Hombre | 20 | 15 | 10 | 8 | 0 | 100 | Sí | Sí | | | | |
| 213 | 20 | Hombre | 2 | 10 | 15 | 20 | 5 | 4 | No | No | | | | |
| 214 | 19 | Hombre | 69 | 40 | 30 | 8 | 12 | 69 | Sí | Sí | | | | |
| 215 | 21 | Hombre | 5 | 1 | 1 | 1 | 0 | 2 | No | No | | | | |
| 216 | 20 | Hombre | 0 | 0 | 0 | 0 | 0 | 0 | No | No | | | | |
| 217 | 78 | Mujer | 5 | 5 | 5 | 0 | 3 | 1 | No | Sí | | | | |
| 218 | 45 | Hombre | 5 | 0 | 5 | 2 | 2 | 0 | no | no | | | | |
| 218 | 45 | Hombre | 5 | 0 | 5 | 2 | 2 | 0 | no | no | | | | |

Eliminamos el registro de la base de datos:

Edad:

Sexo:

Bebidas Semanales:

Cervezas Semanales:

Bebidas Fin de Semana:

Bebidas Destiladas Semana:

Vinos Semana:

Pérdidas Control:

Diversión Dependencia:

Problemas Digestivos:

Tensión Alta:

Dolor Cabeza:

45

Hombre

5

0

5

2

2

0

no

no

Agregar Registro

Actualizar Registro

Ver Registros

Eliminar Registro

Crear/Actualizar Excel

Mostrar Gráficos

| | Id | Edad | Sexo | Bebidas | Cervezas | Bebidas_fs | Bebidas_dest |
|-----|----|--------|------|---------|----------|------------|--------------|
| 209 | 21 | Hombre | 0 | 0 | 0 | 0 | |
| 210 | 19 | Hombre | 5 | 2 | 3 | 2 | |
| 211 | 20 | Hombre | 3 | 4 | 3 | 2 | |
| 212 | 20 | Hombre | 20 | 15 | 10 | 8 | |
| 213 | 20 | Hombre | 2 | 10 | 15 | 20 | |
| 214 | 19 | Hombre | 69 | 40 | 30 | 8 | |
| 215 | 21 | Hombre | 5 | 1 | 1 | 1 | |
| 216 | 20 | Hombre | 0 | 0 | 0 | 0 | |
| 217 | 78 | Mujer | 5 | 5 | 5 | 0 | |
| 218 | 45 | Hombre | 5 | 0 | 5 | 2 | |

Comprobación en la base de datos y en la app.

App:

Vemos que el scroll no baja mas abajo,

| | Id | Edad | Sexo | Bebidas | Cervezas | Bebidas_fs | Bebidas_dest |
|-----|----|--------|------|---------|----------|------------|--------------|
| 208 | 20 | Hombre | 0 | 3 | 0 | 0 | |
| 209 | 21 | Hombre | 0 | 0 | 0 | 0 | |
| 210 | 19 | Hombre | 5 | 2 | 3 | 2 | |
| 211 | 20 | Hombre | 3 | 4 | 3 | 2 | |
| 212 | 20 | Hombre | 20 | 15 | 10 | 8 | |
| 213 | 20 | Hombre | 2 | 10 | 15 | 20 | |
| 214 | 19 | Hombre | 69 | 40 | 30 | 8 | |
| 215 | 21 | Hombre | 5 | 1 | 1 | 1 | |
| 216 | 20 | Hombre | 0 | 0 | 0 | 0 | |
| 217 | 78 | Mujer | 5 | 5 | 5 | 0 | |

Base de datos:

E igual aquí.

| | idEncuesta | edad | Sexo | BebidasSemana | CervezasSemana | BebidasFinSemana | BebidasDestiladasSemana | VinosSemana | PerdidasControl | DiversiónDependenciaAlcohol | ProblemasDigestivos |
|--|------------|------|--------|---------------|----------------|------------------|-------------------------|-------------|-----------------|-----------------------------|---------------------|
| | 208 | 20 | Hombre | 0 | 3 | 0 | 0 | 3 | No | No | I |
| | 209 | 21 | Hombre | 0 | 0 | 0 | 0 | 0 | No | No | I |
| | 210 | 19 | Hombre | 5 | 2 | 3 | 2 | 0 | No | No | I |
| | 211 | 20 | Hombre | 3 | 4 | 3 | 2 | 0 | No | No | I |
| | 212 | 20 | Hombre | 20 | 15 | 10 | 8 | 0 | 100 | Sí | I |
| | 213 | 20 | Hombre | 2 | 10 | 15 | 20 | 5 | 4 | No | I |
| | 214 | 19 | Hombre | 69 | 40 | 30 | 8 | 12 | 69 | Sí | I |
| | 215 | 21 | Hombre | 5 | 1 | 1 | 1 | 0 | 2 | No | I |
| | 216 | 20 | Hombre | 0 | 0 | 0 | 0 | 0 | 0 | No | I |
| | 217 | 78 | Mujer | 5 | 5 | 5 | 0 | 3 | 1 | No | I |
| | 218 | 45 | Hombre | 5 | 0 | 5 | 2 | | | | |

Creamos el Excel de la consulta que hagamos

En mi caso yo pondré 50 años, hombre, 10 bebidas semanales

Filtro

Ingrese la edad máxima (o deje en blanco para todos):

OK

Cancel

Filtro

Ingrese el sexo (o deje en blanco para todos):

Hombre

OK

Cancel

Filtro

Ingrese el número máximo de bebidas semanales (o deje en blanco para todos):

10

OK

Cancel

Vemos que en el Excel no hay ningún dato que se pase de 50 años, que sea mujer, y que haya bebido mas de 10 bebidas semanales.

| C0 | C1 | C2 | C3 | C4 | C5 | C6 |
|----|------|--------|-------------------|--------------------|-----------------------|---------------------------|
| ID | Edad | Sexo | Bebidas Semanales | Cervezas Semanales | Bebidas Fin de Semana | Bebidas Destiladas Semana |
| 3 | 19 | Hombre | 0 | 0 | 0 | 0 |
| 4 | 20 | Hombre | 0 | 0 | 0 | 0 |
| 6 | 20 | Hombre | 1 | 12 | 45 | 12 |
| 7 | 19 | Hombre | 2 | 0 | 2 | 2 |
| 8 | 19 | Hombre | 3 | 5 | 5 | 5 |
| 9 | 22 | Hombre | 0 | 0 | 0 | 0 |
| 10 | 19 | Hombre | 2 | 10 | 5 | 1 |
| 12 | 21 | Hombre | 0 | 0 | 0 | 0 |
| 13 | 19 | Hombre | 5 | 5 | 5 | 5 |
| 17 | 19 | Hombre | 4 | 0 | 1 | 1 |
| 18 | 20 | Hombre | 3 | 7 | 7 | 2 |
| 19 | 20 | Hombre | 3 | 2 | 5 | 3 |
| 22 | 20 | Hombre | 1 | 0 | 0 | 1 |
| 23 | 19 | Hombre | 0 | 0 | 0 | 0 |

La generación del grafico es la base de datos:

