**Ejercicios.**

1. Diseñar una interfaz de usuario, donde se debe solicitar los siguientes datos para obtener un registro, tener en cuenta que dicha interfaz no debe permitir insertar si un campo obligatorio no fue digitado por el usuario y se debe validar el tipo de dato que se necesita, por ejemplo, en la edad no se pueden insertar letras:

• Número documento. → Obligatorio.

• Primer Nombre → Obligatorio.

• Segundo Nombre → Opcional.

• Primer Apellido → Obligatorio.

• Segundo Apellido → Opcional.

• Teléfono → Opcional.

• Correo → Opcional.

• Dirección → Obligatorio.

• Edad → Obligatorio.

• Genero → Obligatorio.

RESPUESTA:

Interfaz de usuario gráfica

Descripción generada automáticamente

**Nota:** Se recomienda probarlo o interactuar con el para mirar las otras validaciones.

2. Terminado el primer punto, la información digitada por el usuario se debe almacenar en:

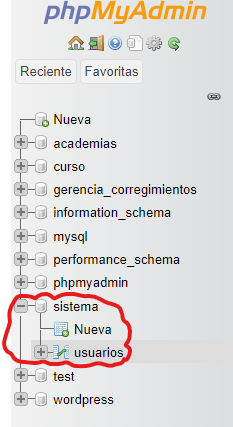
* Base de datos: se puede utilizar cualquier motor de base de datos [Sql server, MySQL , SQLite]. También se puede usar Entity FrameWork si la persona lo desea.

RESPUESTA:

En este caso se usó phpMyAdmin para manejar la administración de MySQL a través de páginas web.

Se creó la base de datos llamada: sistema.

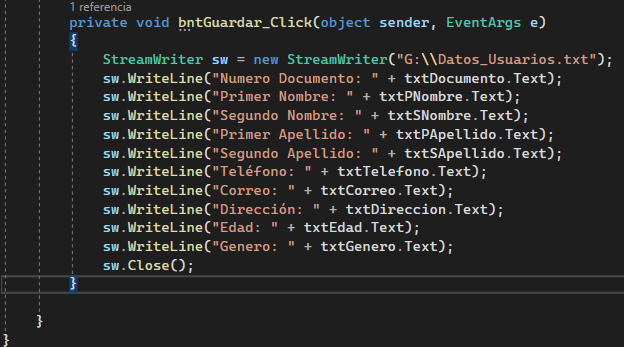
Con su tabla: usuarios.



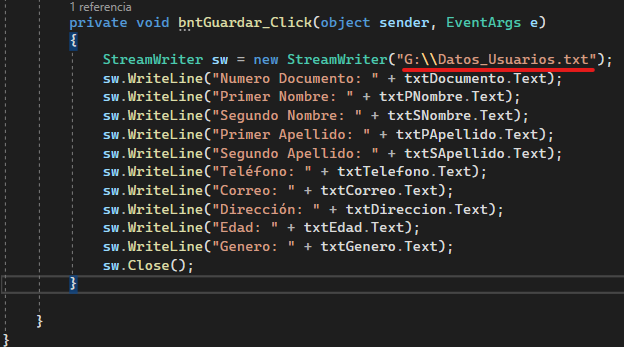
* Archivos Planos: Guardar la información en cualquier orden, tener en cuenta que el archivo se debe guardar donde el usuario lo requiera.

RESPUESTA:

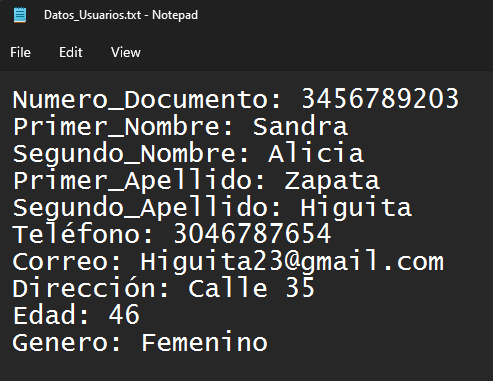
En este caso mediante el siguiente código, se logró exitosamente que Después de llenar los campos de la interfaz de usuario, al darle al botón guardar, se creara y se guardara un archivo con extensión .txt (de texto) en el equipo.



Si el usuario quiere guardarlo en otro lado, simplemente cambia la dirección:



Archivo de texto Datos\_Usuarios:



Dato importante: \*\* Definir la estructura de la tabla en la base de datos de manera libre. Tener en cuenta los campos, definir un tipo de dato lógico y su respectivo tamaño.

RESPUESTA:

CREATE TABLE `usuarios` (

`ID` int(11) NOT NULL AUTO\_INCREMENT ,

`Numero\_Documento` varchar(80) COLLATE utf8mb4\_bin NOT NULL,

`Primer\_Nombre` varchar(100) COLLATE utf8mb4\_bin NOT NULL,

`Segundo\_Nombre` varchar(100) COLLATE utf8mb4\_bin NOT NULL,

`Primer\_Apellido` varchar(100) COLLATE utf8mb4\_bin NOT NULL,

`Segundo\_Apellido` varchar(100) COLLATE utf8mb4\_bin NOT NULL,

`Telefono` varchar(50) COLLATE utf8mb4\_bin NOT NULL,

`Correo` varchar(255) COLLATE utf8mb4\_bin NOT NULL,

`Direccion` varchar(100) COLLATE utf8mb4\_bin NOT NULL,

`Edad` tinyint(4) NOT NULL,

`Genero` varchar(100) COLLATE utf8mb4\_bin NOT NULL,

PRIMARY KEY (`ID`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_bin;

3. Cuando se tengan mínimo 10 registros insertados, el usuario final debe poder realizar las siguientes interacciones:

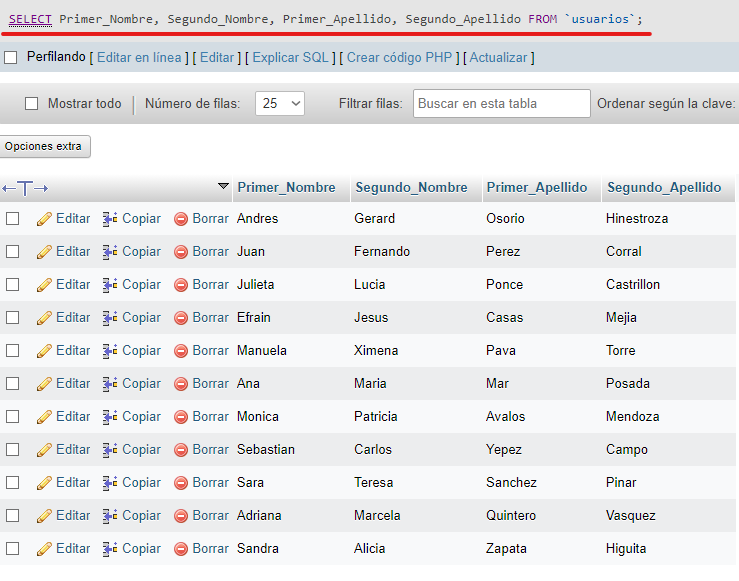
RESPUESTA:

Tabla llena de datos:



* Consultar el nombre completo de todas las personas que fueron insertadas.

R//:



* ¿Cuántas mujeres hay?

R//:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

* ¿Cuántos hombres hay?

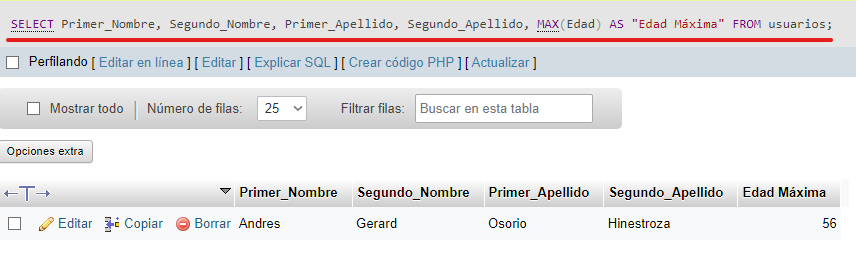
R//:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

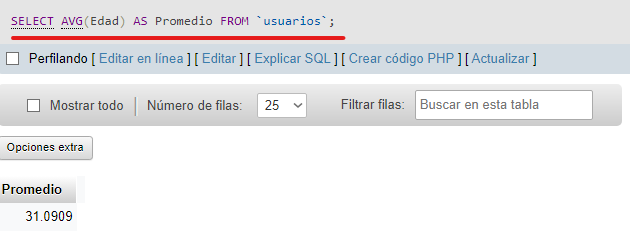
* El nombre completo de la persona con mayor edad.

R//:



* El promedio de la edad.

R//:



Dato importante: \*\* Para obtener las respuestas anteriores, se puede utilizar la base de datos creada o el archivo plano. También la parte de diseño se deja libre, debe ser intuitiva para el usuario.

4. Publicar proyecto en repositorio publico Git Hub. [Enviar Link].

RESPUESTA:

Link:

4. Diseñar una API REST que le de valor agregado al ejercicio planteado, debe ser lógico con lo que se solicita y debe tener las reglas bien definidas.

RESPUESTA:

