Funciones de “LibreriaMenuPrincipal”

void mostrarLogo()

\*muestra el logo de la compañía por consola: “auto elite”

\*3A

1. int validarPatente(char archivoAuto[], AutoArchivo vehiculo);

\*Valida si la patente pasada por parametro está en el archivo autos, pero se le pasa la struct autoArchivo. Retorna flag=1 si existe.

1. int validarPatente2(char archVentas[], char letras[], char numeros[]);

\*Valida si la patente pasada por parametro está en el archivo de ventas, Retorna flag=1 si existe.

1. int validarPatente3(char archivoAuto[], char letras[], char numeros[]);

\*Valida si la patente pasada por parametro está en el archivo de autos, pero la patente se pasa directamente, sin pasar structs. Retorna flag=1 si existe.

1. void agregarAuto (char archivoAuto[]);

\*Agrega un auto al archivo de auto, por consola.

1. void modificarAuto(char archivoAuto[]);

\*esta función, llama a todas las funciones que modifican un dato individualmente, y da la opción de elegir de que modificar, depende que quiera el usuario.

1. void modificarMarca(char archivoAuto[], AutoArchivo patentee);

\*modifica la marca del archivo de auto mediante la patente.

1. void modificarModelo(char archivoAuto[], AutoArchivo patentee);

\*modifica el modelo del archivo de auto mediante la patente.

1. void modificarPatente(char archivoAuto[], AutoArchivo patentee);

\*modifica la patente del archivo de auto mediante la patente.

1. void modificarAnio(char archivoAuto[], AutoArchivo patentee);

\*modifica el anio del archivo de auto mediante la patente.

1. void modificarKms(char archivoAuto[], AutoArchivo patentee);

\*modifica los kms del archivo de auto mediante la patente.

1. void modificarDNI (char archivoAuto[], AutoArchivo patentee);

\*modifica el dni del archivo de auto mediante la patente.

1. void modificarPrecio (char archivoAuto[], AutoArchivo patentee);

\*modifica el precio del archivo de auto mediante la patente.

1. void modificarTodoElRegistro (char archivoAuto[], AutoArchivo patentee);

\*modifica todos los datos en orden del archivo de auto mediante la patente.

\*3B

1. int validarTelefono (char archPersonas[], Persona doc);

\*valida si el teléfono ingresado por parámetro existe en el archivo, retorna flag=1 si existe

1. int validarDni2(char archPersonas[], Persona usuarioAvalidar);

\*valida si el dni ingresado por parámetro existe en el archivo de personas, retorna flag=1 si existe

1. void agregarPersona (char archPersonas[]);

\*agrega una persona al archivo de personas, se ingresan los datos por consola

1. void modificarPersona(char archPersonas[]);

\*esta función llama a todas las demás y da la opción de poder modificar individualmente cada dato de la persona o modificar todo.

1. void modificarPersonaDni (char archPersonas[], Persona doc);

\*esta función modifica solo el dni de la persona, se busca mediante el dni.

1. void modificarPersonaNombre( char archPersonas[], Persona doc);

\*esta función modifica solo el nombre de la persona, se busca mediante el dni.

1. void modificarPersonaTelefono (char archPersonas[], Persona doc);

\*esta función modifica solo el teléfono de la persona, se busca mediante el dni.

1. void modificarPersonaDomicilio(char archPersonas[], Persona doc);

\*esta función modifica solo el domicilio de la persona, se busca mediante el dni.

1. void modificarPersonaRol(char archPersonas[], Persona doc);

\*esta función modifica solo el rol de la persona, se busca mediante el dni.

1. void modificarPersCompleta(char archPersonas[], Persona doc);

\*esta función modifica todos los datos de la persona, se busca mediante el dni.

\*3C

1. void mostrarPersonas(char archPersonas[]);

\*muestra todas las personas del archivo de personas.

\*3D

1. void mostrarUNApersona(char archPersonas[], Persona doc);

\*muestra solo una persona del archivo de personas, se busca por dni.

\*3E

1. void mostrarAutos(char archivoAuto[]);

\*muestra todos los autos del archivo de autos

\*3F

1. void mostrarAutosEnVenta (char archivoAuto[], char archPersonas[]);

\*muestra todos los autos en venta del arhivo Auto.

\*3G

1. void mostrarUNauto(char archivoAuto[], AutoArchivo patentee);

\*muestra un solo auto mediante la patente ingresada por parámetros

\*3H

1. void verVentas(char archVentas[]);

\*muestra todas las ventas registradas

\*3I

1. void verUnaVenta(char archVentas[]);

\*muestra solo una venta, mediante la patente pedida dentro de la función.

\*3J

1. void registrarUnaVenta(char archivoAuto[], char archVentas[]);

\*se utiliza para registrar una venta y guardarla en el archivo de ventas

\*3K

1. float gananciaDeAuto(char autosEnVenta[], float precioVenta, char letras[], char numeros[]);

\*calcula la ganancia que genero un auto ya vendido, retorna la ganancia como float

1. float verRecaudacion(char archVentas[]);

\*es para ver la ganancia en un mes y año ingresado por consola, el total de la ganancia del mes y año ingresado

\*3L

1. void mostrarUnaVentaLibre(Venta autito);

\*muestra una venta ingresada por parámetro.

1. Venta verMayorVenta(char archVentas[]);

\*muestra la venta con mayor ganancia hasta la fecha

\*3M

1. int contAutosMenoresDiezAnios(char archivoAuto[]);

\*Cuenta la cantidad de autos menores a 10 años, retorna la cantidad de autos del archivo.

1. void autosMenoresDiezAnios (char archivoAuto[], AutoArchivo \*\* autosMenor, int tamanio);

\* esta función guarda todos los autos menores a 10 años dentro de un arreglo dinamico.

1. int encontrarMenor(AutoArchivo autosMenor[], int validos, int pos);

\*esta función encuentra el menor auto por año, retorna la posición del menor

1. void ordenarPorSeleccion(AutoArchivo autosMenor[], int validos);

\*esta función ordena por selección el archivo de autos, por año de antiguedad

1. void mostrarMenoresDiezAnios(AutoArchivo autosMenor[], int tamanio);

\*muestra todos los autos menores de 10 años por el arreglo dinamico y ordenado

Funciones de “LibreríaInicio”

1. int menuInicio();

\*es el primer menú del programa, retorna la opción que elija el usuario

1. void menuRegistro(char usuarios[]);

\*esta función permite crear un usuario para ingresar al programa y lo guarda en el archivo de usuarios

1. int contarInt(int num);

\*esta función cuenta la cantidad de dígitos de un numero, retorna la cantidad de dígitos

1. int validarCorreo(char archivo[], stRegistro usuarioAValidar);

\*valida si el correo electrónico ya fue utilizado para registrarse y no repetir, retorna flag=1 si ya fue utilizado.

1. void mostrarUsuarios(char archivo[]);

\*muestra todos los usuarios creados

1. int validarDni(char archivo[], stRegistro usuarioAValidar);

\*validad si el dni ingresado ya existe en el archivo de usuarios, retorna flag=1 si ya fue utilizado

1. int validarUsuario(char archivo[], stRegistro usuarioAValidar);

\*valida si el nombre del usurio ya fue ingresado antes, retorna flag=1 si ya existe

1. int menuInicioSesion(char registro[]);

\*esta función compara lo ingresado por consola con los usuarios registrados, y si coinciden retorna flag=1.