Estudiantes 2266919/2266916alejandrovidal.juanpereira

Requerimientos escogidos para el programa:

1. El sistema en el módulo de gestión recursos debe registrar recursos para que la coordinadora de bienestar universitario o el monitor pueda almacenar el tipo de recurso, el nombre del recurso y la cantidad que ingresa al área.
2. El sistema en el módulo de gestión de recursos debe modificar el nombre y la cantidad de un recurso para que la coordinadora o el monitor de bienestar pueda hacer el cambio de algún dato erróneo que se haya ingresado.
3. El sistema en el módulo de gestión de recursos debe registrar la salida de recursos por concepto de donación para que la coordinadora o el monitor pueda almacenar el nombre del recurso, la cantidad, la fecha y a quien se le realiza la donación.
4. El sistema en el módulo de gestión de recursos debe registrar un retiro (dar de baja) de recursos que se realice para que la coordinadora o el monitor pueda almacenar el nombre de recurso, la cantidad y la fecha en que se realiza el retiro.
5. El sistema en el módulo de gestión de préstamos debe registrar un préstamo que se realice para que la coordinadora o el monitor pueda almacenar la fecha y hora del préstamo, el código, nombre, apellido, teléfono, plan, jornada, estamento, de quien solicitó el préstamo, así como los recursos prestados y su cantidad.
6. Cuando se registre un préstamo el sistema debe asignar a ese préstamo un estado de “activo” para que la coordinadora o el monitor sepa que prestamos se encuentran activos en su jornada laboral.
7. El sistema en el módulo de gestión de préstamos debe finalizar un préstamo que esté en estado activo o mora para que la coordinadora o el monitor pueda cambiar el estado del préstamo a finalizado, almacenar la fecha y hora de la finalización del préstamo y el código de la persona que recibió en la oficina el recurso.
8. El sistema en el módulo de gestión de usuarios debe modificar el código, nombres, apellidos, teléfono, plan, jornada y estamento del usuario así la coordinadora o el monitor puede realizar el cambio de algún dato erróneo que se haya ingresado.
9. El sistema en el módulo de inicio debe permitir el inicio de sesión para que la coordinadora o el monitor pueda acceder a todas las gestiones del sistema y a sus funcionalidades.
10. Cuando se inicie sesión en el sistema este debe mostrar el nombre del usuario que se encuentre usando la aplicación, de esta manera se puede reconocer quién está usando el software en dicho momento

Receta:

\*Función ingresar:

Definición de datos:

user1=Variable que contiene un Entry con el nombre del usuario

user=Tipo StringVar que contiene lo obtenido en la entrada de texto

password1=Variable que contiene un Entry con la contraseña

password=Tipo StringVar que contiene la contraseña obtenida de la entrada de texto

contra=Variable que contiene un Label con el texto “Ingrese Contraseña”

lus=Variable que contiene un Label con el texto de “Ingrese Usuario”

b1=Variable que contiene el botón con nombre “Accéder” que llama a la función

Firma:

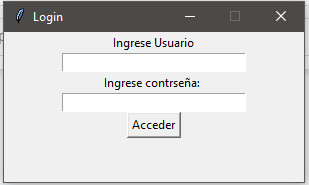
Entradas=user, password

Salidas=llamado a la función “principal” que contiene el resto del programa

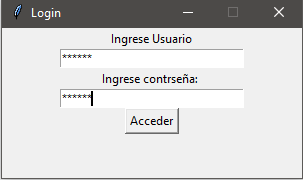
Propósito: Que esta primera función sea la encargada de preguntar un usuario y contraseña para poder dar paso al resto del programa.

Ejemplos funcionales:

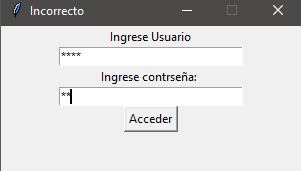
1. Al ejecutar el programa:



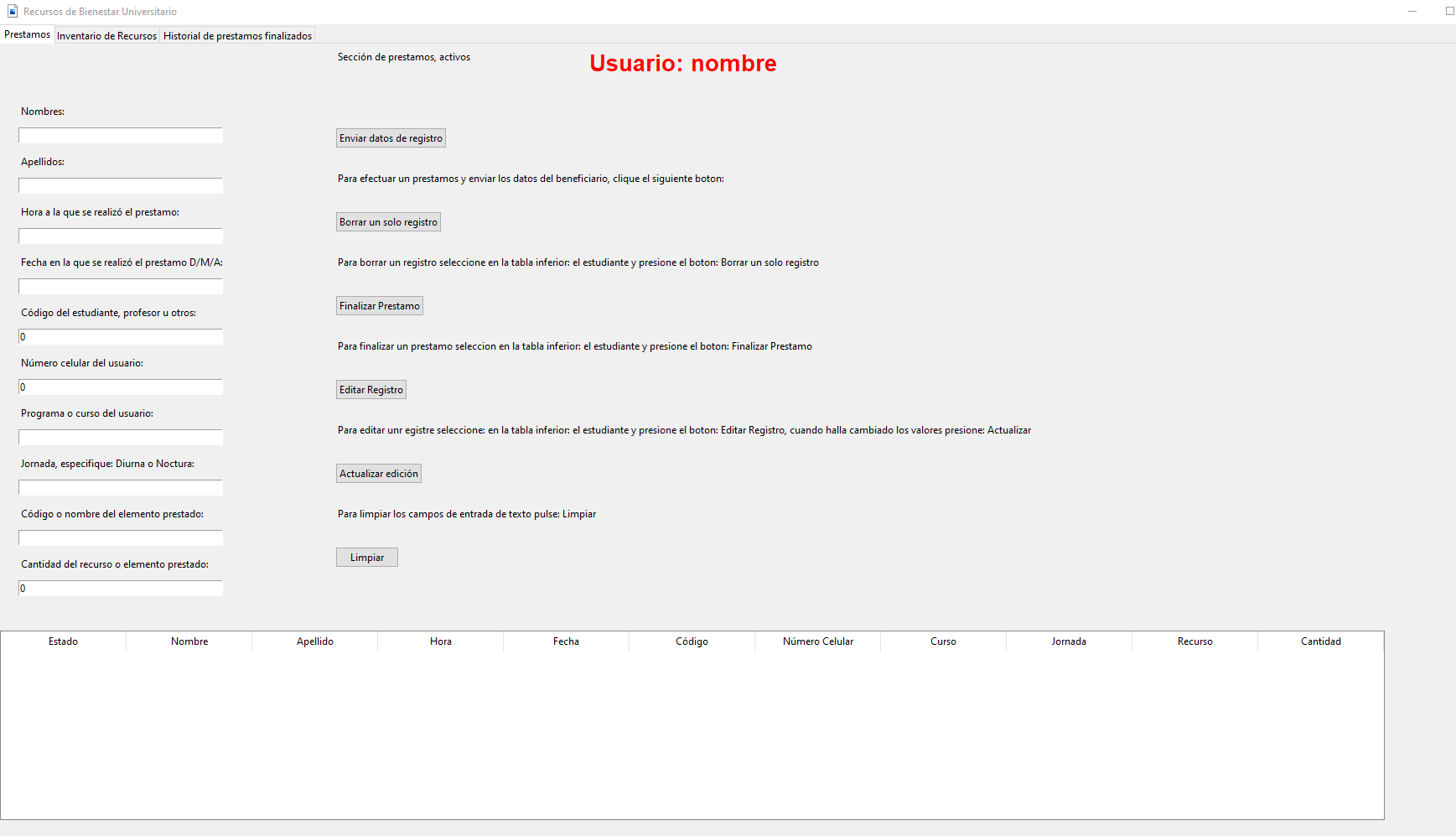
1. Ingreso de datos del usuario



1. Al ingresar contraseña o usuario incorrecto se mostrará que es “Incorrecto” en la parte superior izquierda



1. Caso de contraseña correcta. Paso al programa principal



\*Función principal:

Definición de datos:

PRIMER INTERFAZ

nombre, apellido, hora, fecha, codigo, curso, jornada, recurso, elemento1, codigo2, cantidad2A, tipo, Ele1, ti2,

donaobaja=Variables de tipo StringVar que contiene datos preguntados al usuario de tipo cadena.

codigo, numCelular, cantidad, codigo2, cantidad2A, cant2A, cod2=Variables de tipo IntVar

que contienen datos de tipo numérico preguntados al usuario

ventana2=Variable que abre la segunda ventana tras ingresar usuario y contraseña correctos.

panel=Variable que contiene un panel de pestañas del módulo ttk.Notebook

tab1=Indica el panel sobre el cual se trabajará

nombrA=Variable que contiene un Label que dice “Nombres: “

nombr=Variable que contiene un Entry para que el usuario ingrese su nombre

apellidA=Variable que contiene un Label que dice “Apellidos”

apellid=Variable que contiene un Entry para que el usuario ingrese su apellido

horA=Variable que contiene un Label que dice “Hora a la que se realizó el préstamo: ”

hor=Variable que contiene un Entry para que se ingrese la hora de préstamo

fechA=Variable que contiene un Label que dice “Fecha en la que se realizó el préstamo: “

fech=Variable que contiene un Entry para que se ingrese la fecha del préstamo

codigA=Variable que contiene un Label que dice “Código del estudiante, profesor u otros:” para tener una identificación del usuario

codig=Variable que contiene un Entry para que se ingrese el código o identificación del usuario

numCelulaA=Variable que contiene un Label que dice: “Número celular del usuario: ”

numCelula=Variable que contiene un Entry para que se ingrese el numero celular de usuario

cursA=Variable que contiene un Label que dice: “Programa o curso del usuario: “

curs=Variable que contiene un Entry para que se ingrese el curso al que pertenece el usuaro

jornadA=Variable que contiene un Label que dice: Jornada, especifique: Diurna o Nocturna:“

jornad=Entrry variable que contien un Entry para que se ingrese la jornada que cursa el usuario

recursA=Variable que contiene un Label que dice: “Código o nombre del elemento prestado:”

recurs=Variable que contiene un Entry para que se ingrese el recurso que fue prestado

cantidaA=Variable que contiene un Label que dice: “Cantidad o recurso o elemento prestado:”

etiqueta=Variable que contiene un Label que dice “Sección de préstamos activos”

datost=Variable que contiene un Label, que dice “"Para efectuar un prestamos y enviar los datos del beneficiario, clique el siguiente botón:"

botonenviar=Variable que contiene un botón para llamar a la función “enviardatos”

botonborrar=Variable que contiene un botón para llamar a la función “borrarregistro”

borrar1=Variable que contiene un Label que dice “Para borrar un registro selecciona en la tabla inferior: el estudiante y presione el botón: Finalizar Préstamo”

botonfinalizar=Variable que contiene un botón para llamar a la funcion “terminarprestamo”

finalizarpres=Variable que contiene un Label que dice “Para finalizar un préstamo en la selección en la tabla inferior; el estudiante y presiones el botón; Finalizar Préstamo”

botoneditar=Variable que contiene un botón para llamar a la función “seleccionado”

editartext=Variable que contiene un Label que dice “Para editar un registro seleccione : en la tabla inferior: el estudiante y presione el botón: Editar Registro, cuando haya cambiado los valores presione: Actualizar”

actualizar=Variable que contien un boton para llamar a la función “editarregistro”

borrarlabel=Variable que contiene un Label que dice “Para limpiar los campos de entrada de texto pulse: Limpiar”

botonlimpiar=Variable que contiene un botón para llamar a la función “eliminarcasillas”

labelusuario=Variable que contiene un Label que dice “Usuario” y es encargado de mostrar el usuario que inicio sesión

trev=Variable que contiene una tabla con los datos ingresados del usuario anteriormente, perteneciente al módulo ttk.Treeview

SEGUNDA INTERFAZ

tab2=Contiene un segundo panel o pestaña perteneciente al módulo ttk.Frame

ElementoA1=Variable que contiene un Label que dice “Nombre del elemento”

ElementoA2=Variable que contiene un Entry para registrar el tipo de elemento al inventario

cantidadA2=Variable que contiene un Label que dice: “Cantidad”

cantidad2Variable que contiene un Entry que captura la cantidad de elementos ingresados al inventario

codigA2=Variable que contiene un Label que dice: “Código/ID del elemento:”

codig2=Variable que contiene un Entry para capturar el código del elemento

textotipo=Variable que contiene un Label que dice “Tipo del recurso: Deportivo o Musical”

tip=Variable que contiene un Entry para capturar el tipo de elemento

botonenviar2=Variable que contiene un botón que llama a la función “enviarelementos”

botoneditar2=Variable que contiene un botón que llama a la función “seleccionado2”

editartext2=Variable que contiene un Label que dice: “Para editar un registro seleccione: en la tabla inferior: el estudiante y presione el botón: Editar Registro, cuando haya cambiado los valores presiones: Actualizar”

actualizar2=Variable que contiene un botón para llamar a la función “editaregistroelemento”

etiqueta2=Variable que contiene un Label que dice: “Sección para visualizar los elementos totales de bienestar disponibles, así como agregar nuevos elementos o darlos de baja”

arbol2=Contiene la segunda tabla donde se agregan o bajan elementos de bienestar, perteneciente al módulo ttk.Treeview

TERCER INTERFAZ

tab3=Contiene el tercer panel o pestaña perteneciente al módulo ttk.Frame

etiqueta3=Variable que contiene un Label que dice: “Sección para visualizar prestamos finalizados”

vistarbol3=Contiene la tercera tabla donde se agrega el historial de prestamos finalizados, perteneciente al modulo ttk.Treeview

etiqueta4=Variable que contiene un Label que dice: “Registro de elementos dados de baja y elementos adquiridos o donación”

tree4=Contiene la cuarta tabla donde se agregan dichos registros perteneciente al modulo ttk.Treeview

EA1=Variable que contiene un Label que dice : “Nombre del elemento”

EA2= Variable que contiene un Entry para agregar el elemento

cA2=Variable que contiene un Label que dice: “Cantidad”

c2=Variable que contiene un Entry para entrar la cantidad de elementos

coA2=Variable que contiene un Label que dice “Código/ID del elemento: ”

co2=Variable que contiene un Entry para agregar el código del elemento

textip=Variable que contiene un Label que dice: “Tipo del recurso: Deportivo o Musical”

ti=Variable que contiene un Entry para capturar el tipo de recurso

donaobajalabel=Variable que contiene un Label que dice: “Indique si es baja o donación”

donaobajaentry=Variable que contiene un Entry para preguntar estos parámetros

btenviar=Variable que contiene un botón que llama a la función “listo”

Firma:

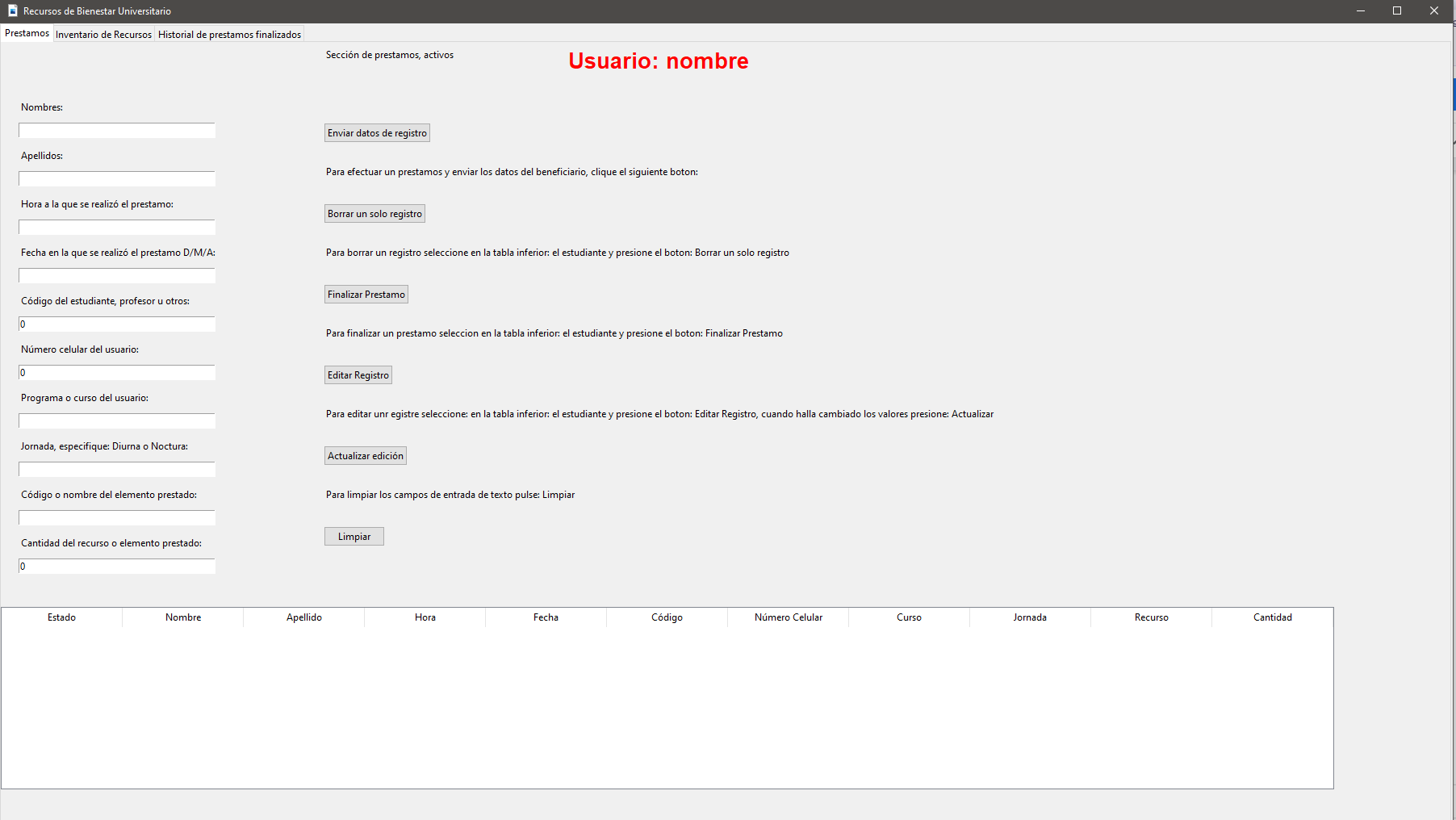
Entradas: numCelular, cantidad, codigo2, cantidad2A, cant2A, cod2, nombre, apellido, hora, fecha, codigo, curso, jornada, recurso, elemento1, codigo2, cantidad2A, tipo, Ele1, ti2,

Salida: Llamado a las distintas funciones que mencione en los botones anteriores.

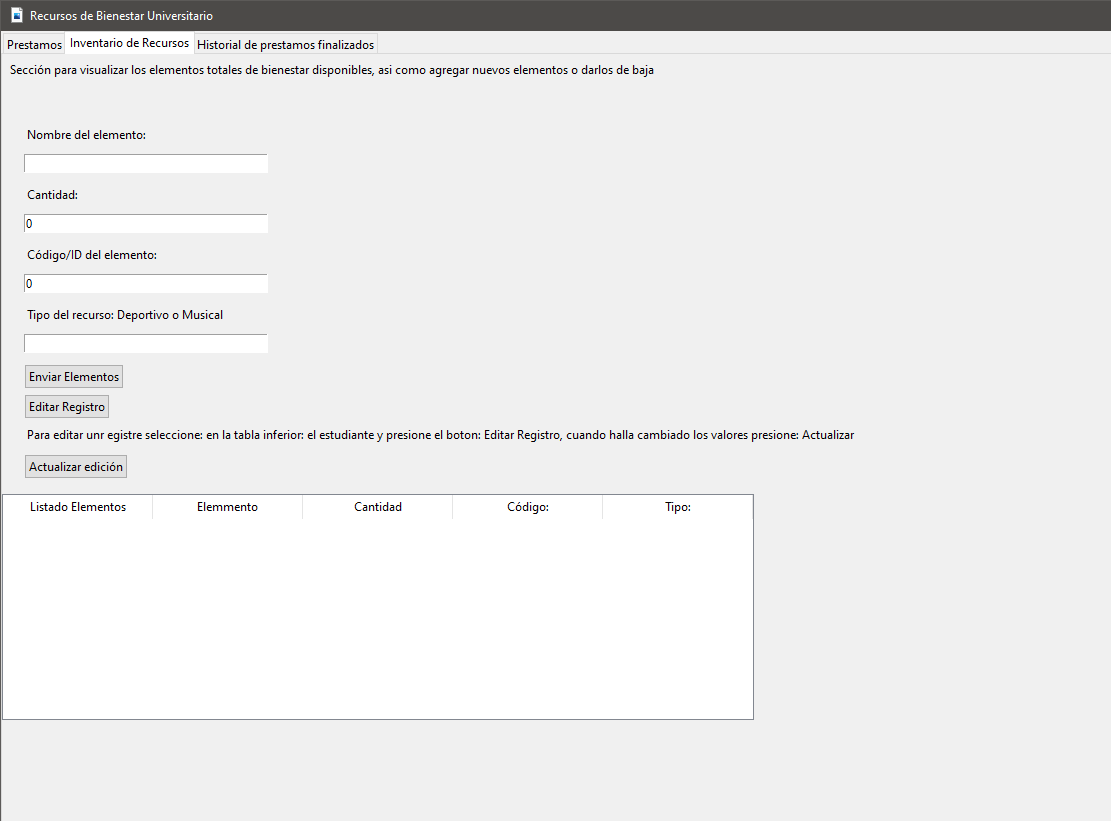
Propósito: Que esta función sea como su nombre lo dice, la principal puesto que esta tiene el cuerpo del programa así como es la encargada de mandar a llamar las demás funciones que realizan otras tareas.

Ejemplos funcionales:

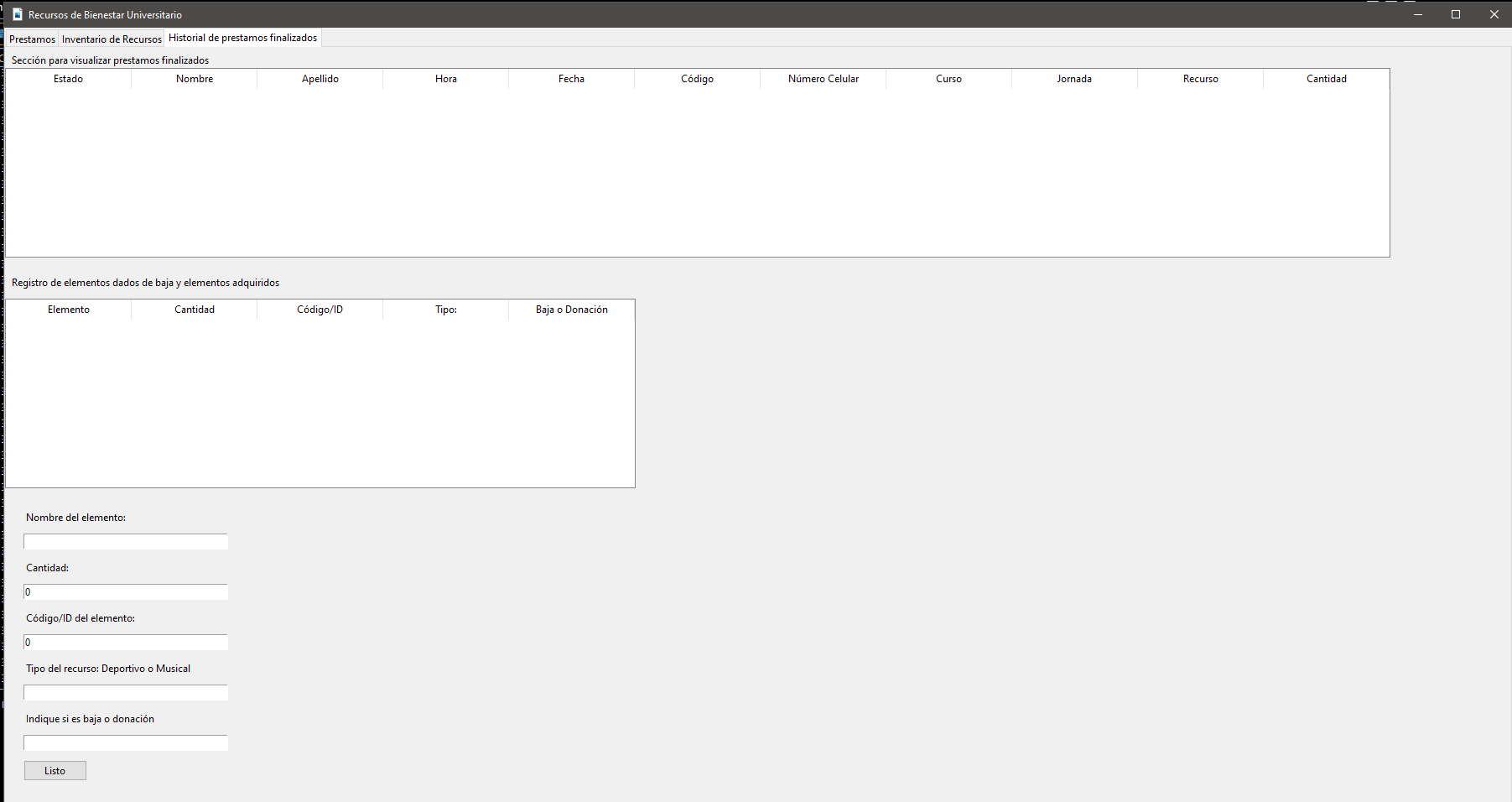
Página principal:



Segunda página:



Tercera página:



\*Función almacenaArchivo

Definición de datos:

nombreFile=Variable que contiene el nombre del archivo donde guardaremos la información

cadenaAlmacenar=Variable que contiene la cadena con los datos del usuario ingresados

file= Variable que me permite interactuar con el archivo de texto

Firma:

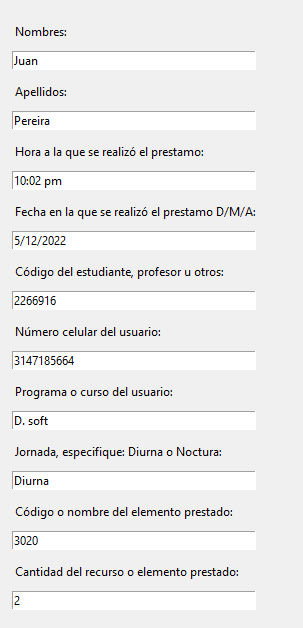
Entradas=nombreFile, CadenaAlmacenar, file

Salidas=Escritura en archivo de texto

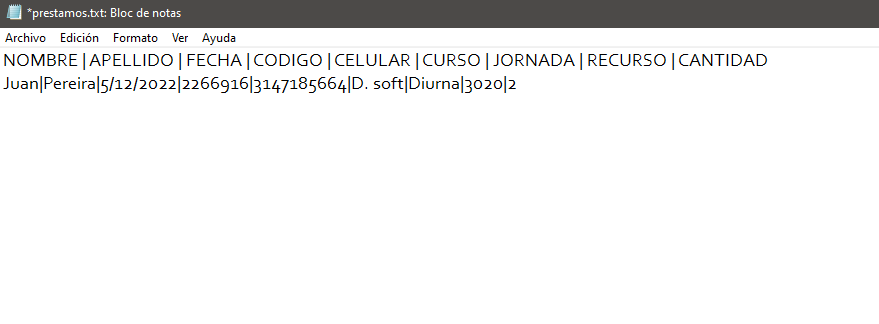
Propósito: El propósito de la variable es que me escriba un registro de los prestamos realizados en una hoja de texto.

Ejemplos:

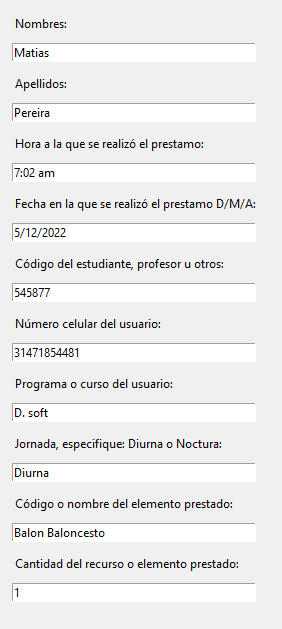
Ingresamos datos:



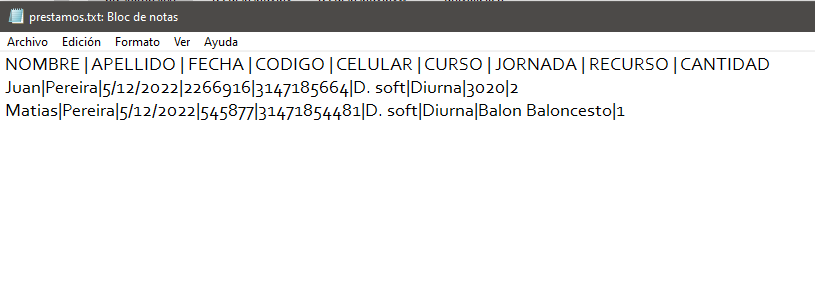
Resultado:



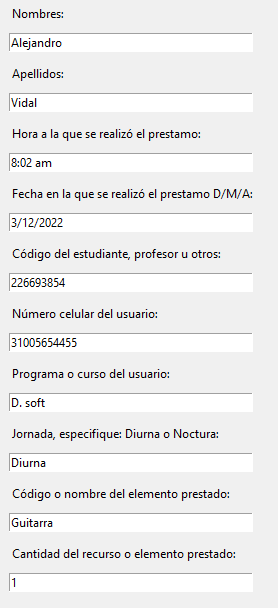
Ingresamos datos:



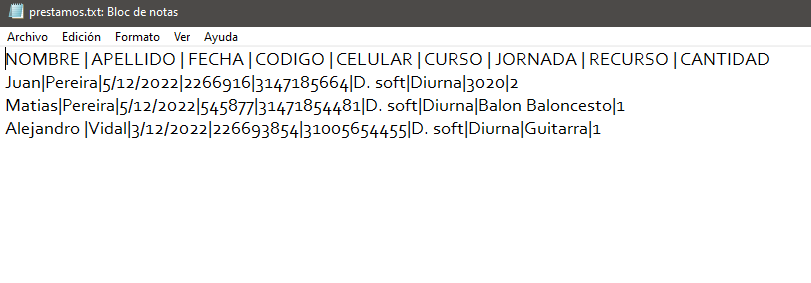
Resultado:



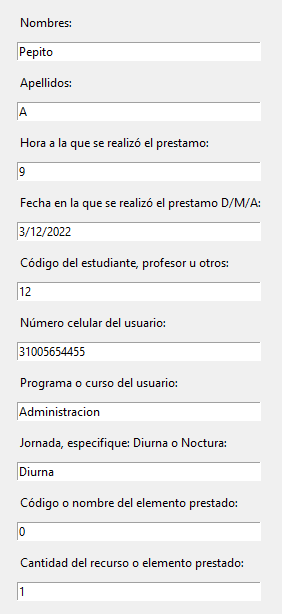
Ingresamos datos:



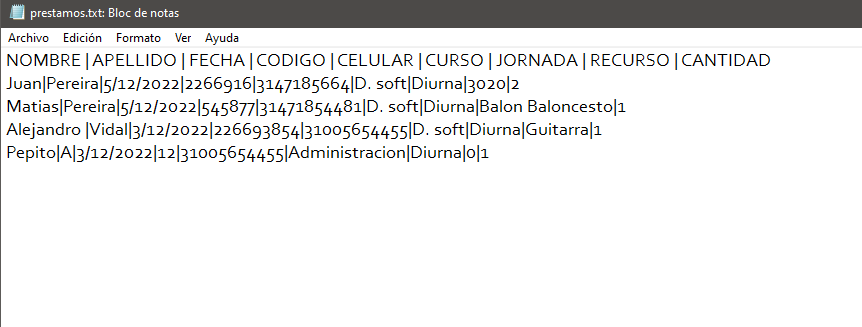
Resultado:



Ingresamos datos: Esta vez incompletos a ver que pasa:



Resultado:



\*Función enviar elementos:

Definición de datos:

ElementoA2=Variable StringVar que contiene el nombre del elemento

cantidad2=Variable IntVar que contiene la cantidad del elemento

codig2=Variable IntVar que contiene el código del elemento

tip=Variable StringVar que contiene el tipo del elemento, deportivo o musical

Firma:

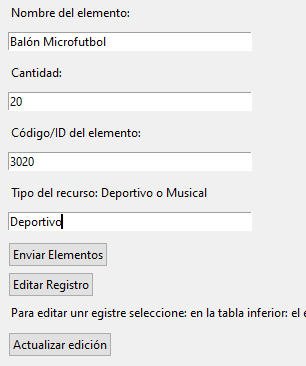
Entradas: ElementoA2, cantidad2, codig2, tip

Salidas: Ingreso de los elementos anteriores al arbol2

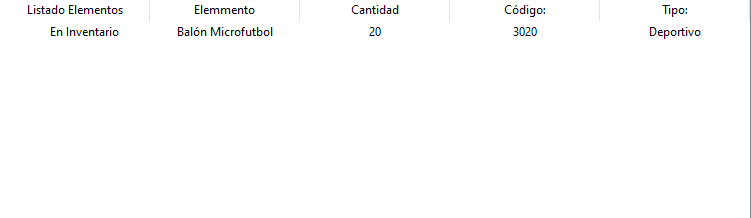
Propósito: Que la función envíe los datos al registro que hay en la aplicación de los elementos que tiene bienestar en total

Ejemplos funcionales:

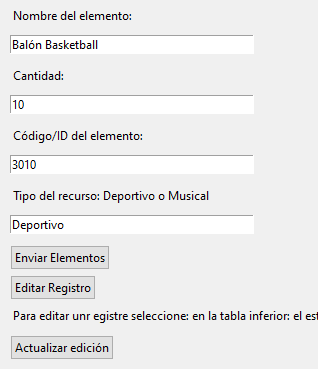
Ingresamos datos:



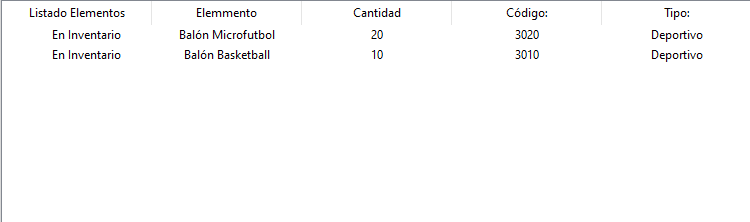
Resultado:



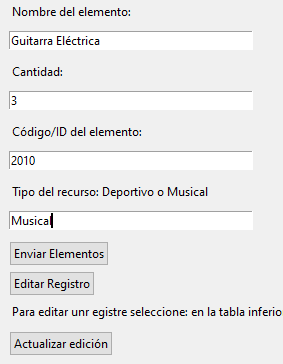
Ingresamos datos:



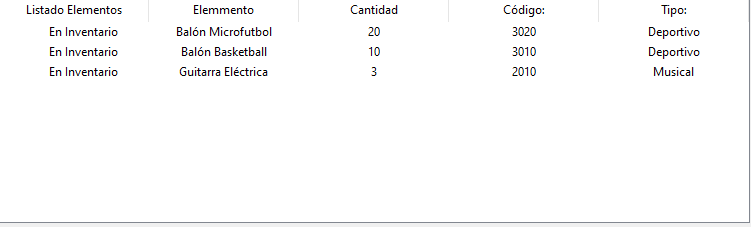
Resultado:



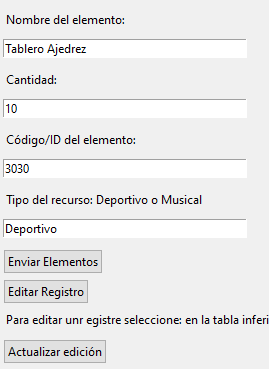
Ingresamos datos:



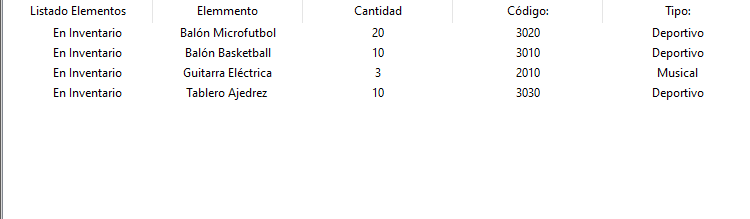
Resultado:



Ingresamos datos:



Resultado:



\*Función editarregistro:

Definición de datos:

selected=Variable que nos permite seleccionar un ítem en la tabla trev

nombr=Variable StringVar que contiene el nombre ingresado del el usuario

apellid=Variable StringVar que contiene el apellido ingresado del usuario

hor=Variable StringVar que contiene la hora del préstamo

fech=Variable StringVar que contiene la fecha del préstamo

codig=Variable IntVar que contiene el código ingresado del usuario

numCelula=Variable IntVar que contiene el número celuar ingresado del usuario

curs=Variable StringVar que contiene el curso ingresado del usuario

jornad=Variable StringVar que contiene la jornada académica ingresada del usuario

recurs=Variable StringVar que contiene el recurso del préstamo

cantida=Varibale IntVar que contiene la cantidad prestada de un recurso

Firma:

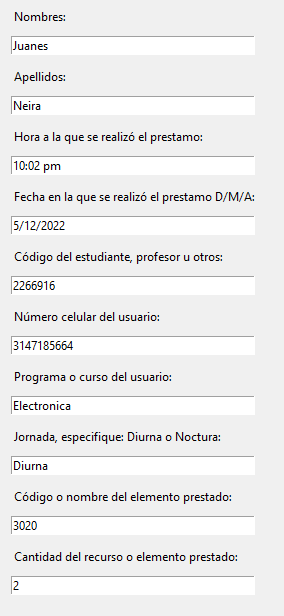
Entrada=nombr, apellid, hor, fech, codig, numCelula, curs, jornad, recurs, cantida

Salida= Nuevos datos editados.

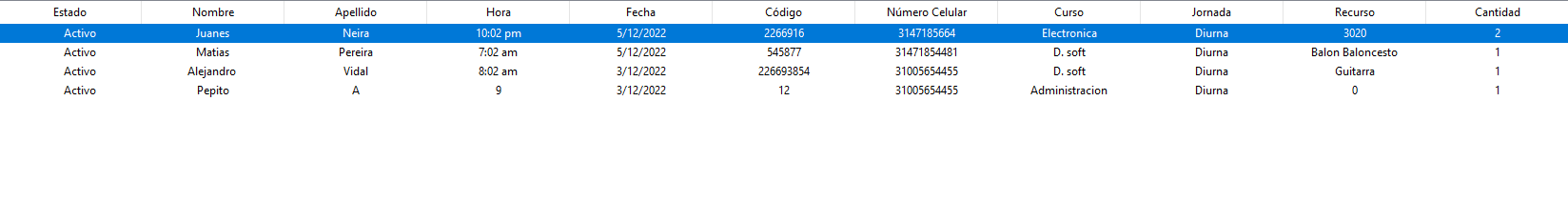
Propósito: Que la funcion me permita editar datos que queramos en la tabla

Ejemplos:

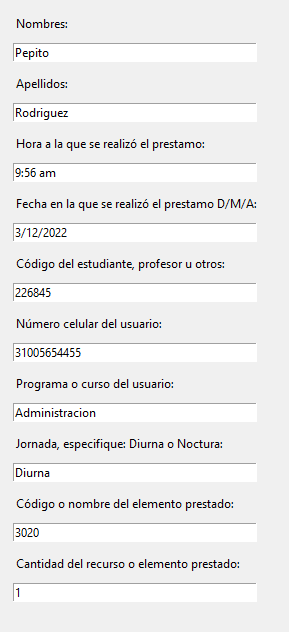
Editamos:



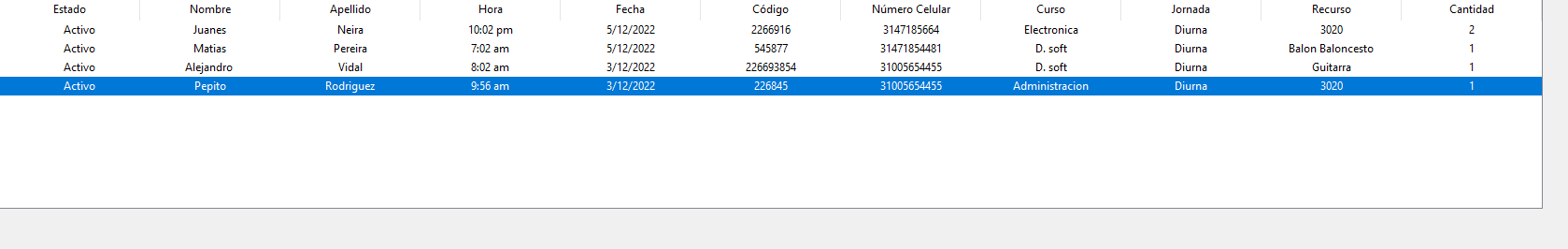
Resultado:



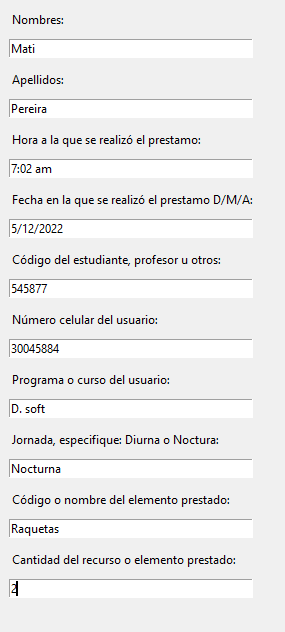
Editamos:



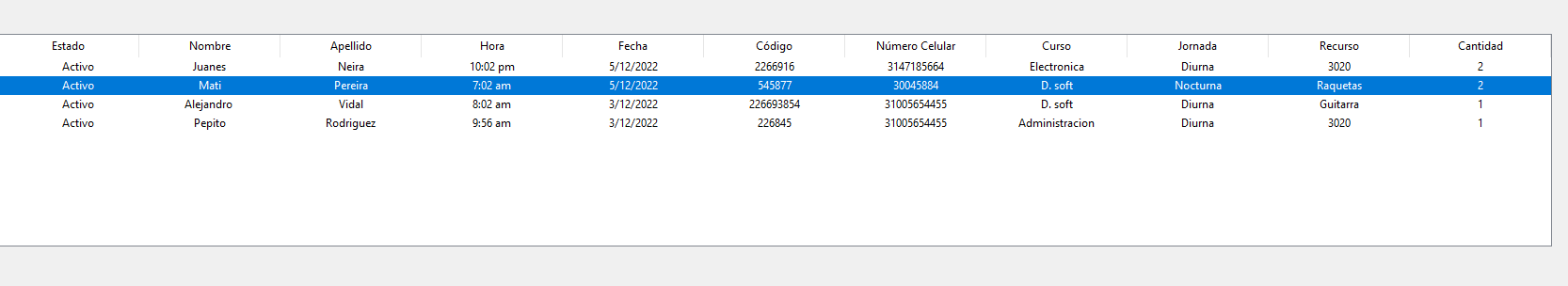
Resultado:



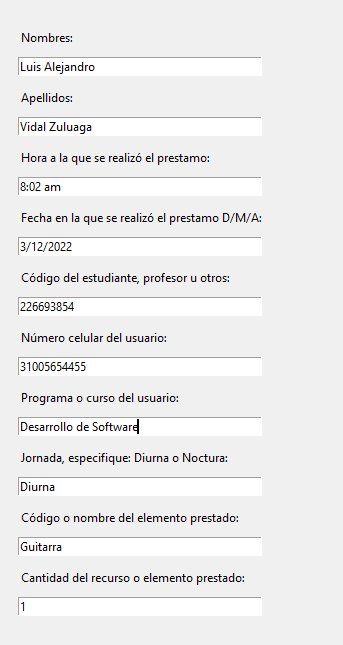
Editamos:



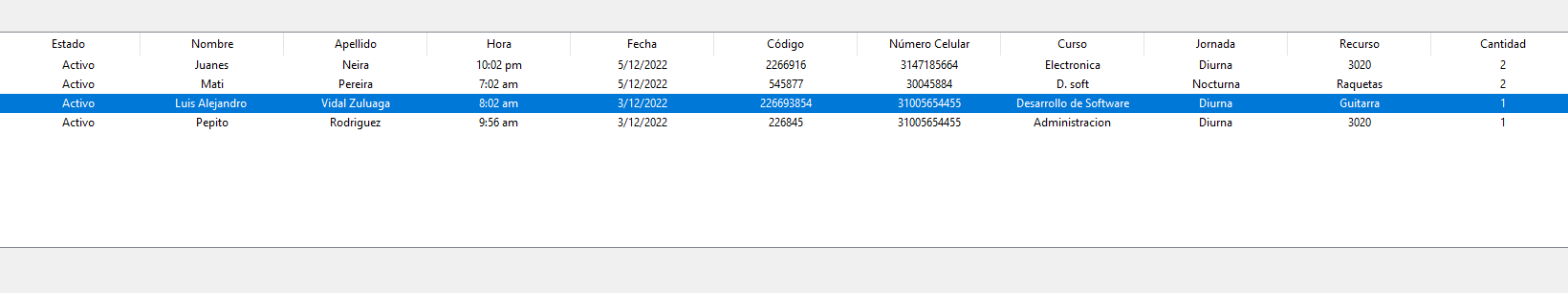
Resultado:



Editamos:



Resultado:



Función editregistroelemento

Definición de datos:

selected2=Variable que nos permite seleccionar alguna casilla en la tabla arbol2

ElementoA2=Variable StringVar que contiene el elemento

cantidad2=Varible IntVar que contiene la cantidad de un elemento

codig2=Variable IntVar que contiene el código de un elemento

tip=Variable StringVar que contiene el tipo de un elemento

Firma:

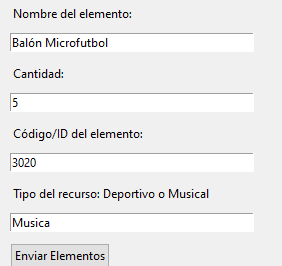
Entradas: ElementoA2, cantidad2, codig2, tip

Salida: Nuevos datos editados en la tabla

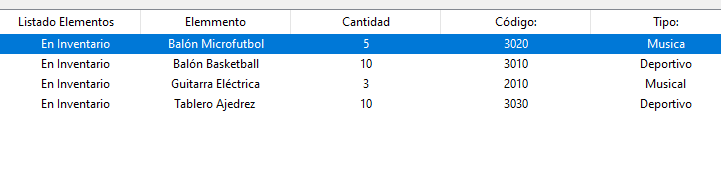
Propósito: Que la función me permite editar datos que ya habíamos agregado en la tabla de elementos que tiene bienestar en su inventario

Ejemplos:

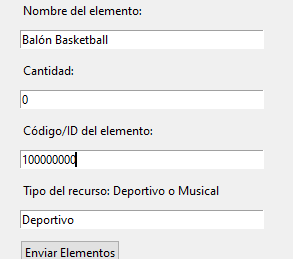
Editamos:



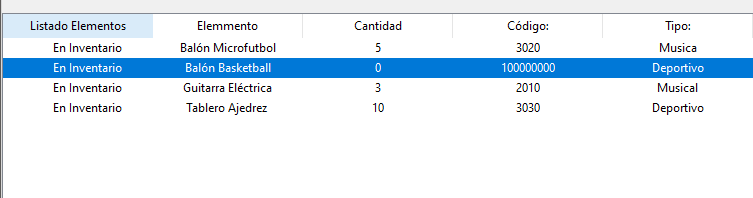
Resultado:



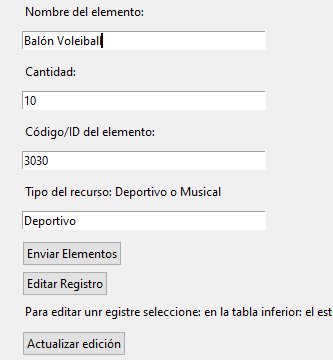
Editamos:



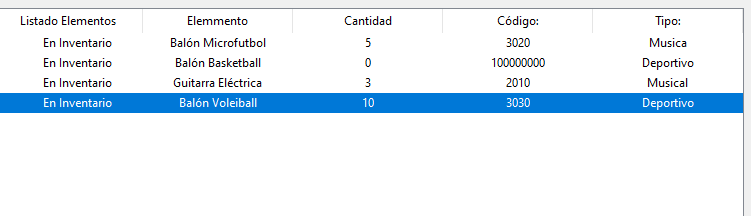
Resultado:



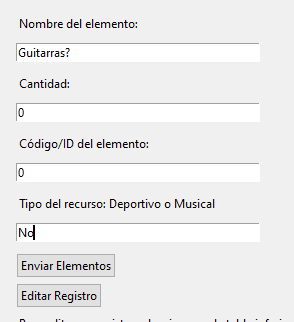
Editamos:



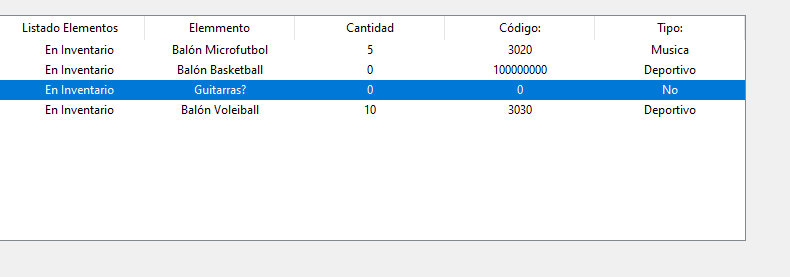
Resultado:



Editamos:



Resultado:



\*Función seleccionado:

Definición de datos:

nombr, apellid, hor, fech, curs, jornad, recurs=Variables tipo StringVar que contienen los datos obtenidos por el usuario

codig, numCelua, cantida=Variables tipo IntVar que contiene los datos obtenidos del usuario

selected=Variable que me permite seleccionar alguna fila de la tabla

valr=Variable que me permite interactuar con todos los valores de la fila en una tabla

Firma:

Entradas=nombr, apellid, hor, fech, curs, jornad, recurs, codig, numCelua, cantida

Salidas= Datos otra vez ingresados a los campos de texto

Propósito: Que la función me permite seleccionar una fila de una tabla (trev), permitiendo a su vez interactuar con los datos en compañía de otras funciones, como para rellenar las entradas de texto según la fila seleccionada para después ser editada por eso es más una función auxiliar que otra cosa

Ejemplos funcionales:

También fueron presentados en las imágenes anteriores

\*Función seleccionado2

Definición de datos:

selected2=Variable que nos permite seleccionar alguna casilla en la tabla arbol2

ElementoA2=Variable StringVar que contiene el elemento

cantidad2=Varible IntVar que contiene la cantidad de un elemento

codig2=Variable IntVar que contiene el código de un elemento

tip=Variable StringVar que contiene el tipo de un elemento

Firma:

Entradas: ElementoA2, cantidad2, codig2, tip

Salidas: Datos ingresados de nuevo a los campos de texto

Propósito: Que la función me permite seleccionar una fila de una tabla, permitiendo a su vez interactuar con los datos en compañía de otras funciones, permitiendo rellenar de nuevo en las entradas de texto los valores de la tabla, solo que para esta ocasión me tocó hacer otra funcion aparte por comodidad puesto que son valores diferentes a la tabla anterior.

\*Función borrarregistro

Definición de datos:

remover=variable que me permite seleccionar una fila de la tabla en este caso trev

trev=que contiene la tabla de la página 1 que para este caso está usando otras herramientas como delete

Firma:

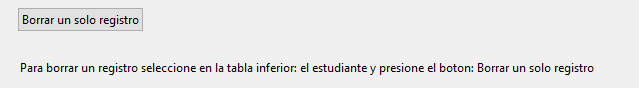
Entradas=trev que es la tabla que se selecciona una fila para borrar registro

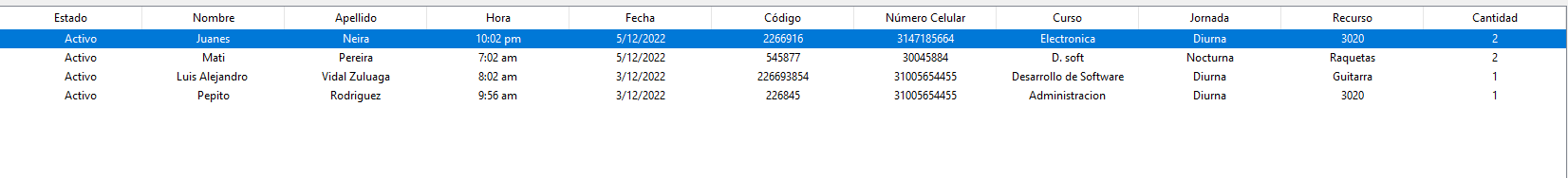
Salida=Sin la lista anterior en la tabla

Propósito: Que la función me permite borrar una fila entera de datos de la tabla en caso de que se necesite.

Ejemplos funcionales:

Presionamos el botón:

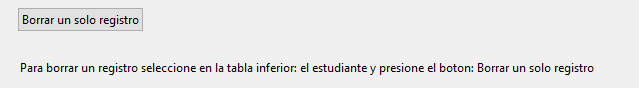


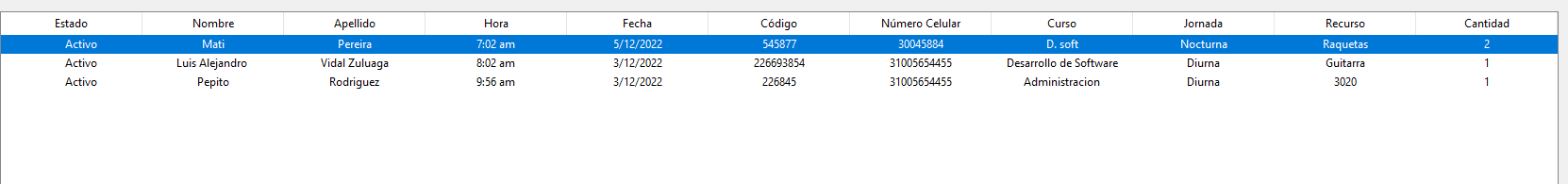


Resultado:

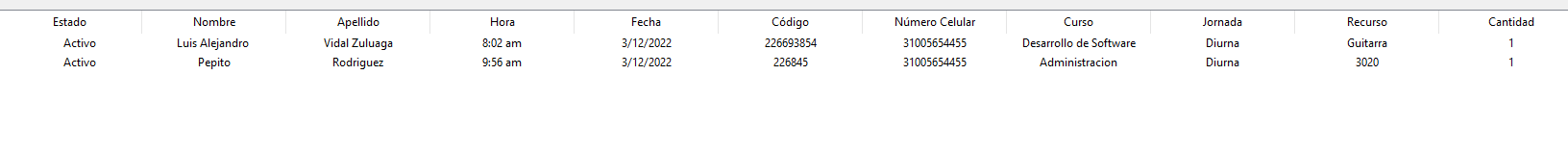


Presionamos el botón:

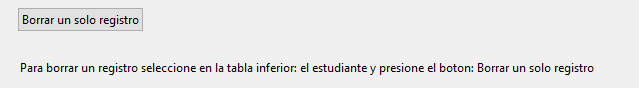


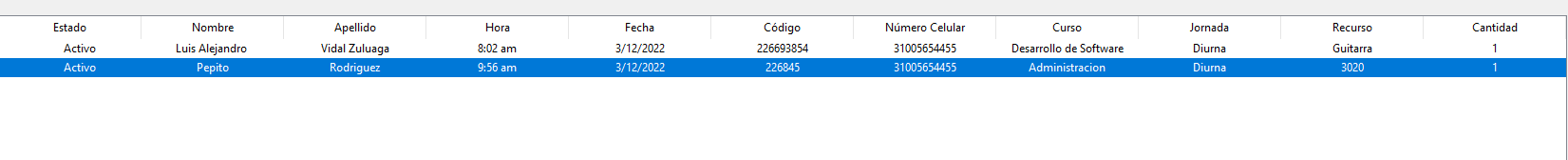


Resultado:

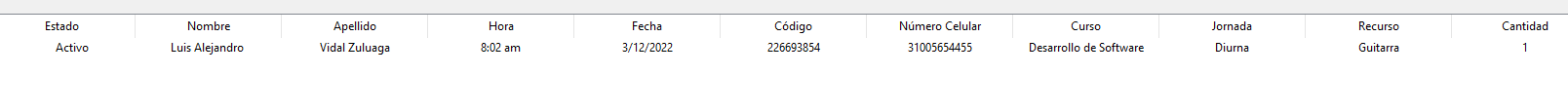


Presionamos el botón:

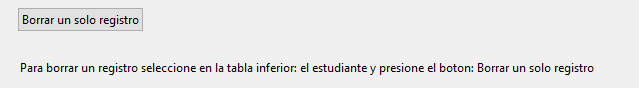


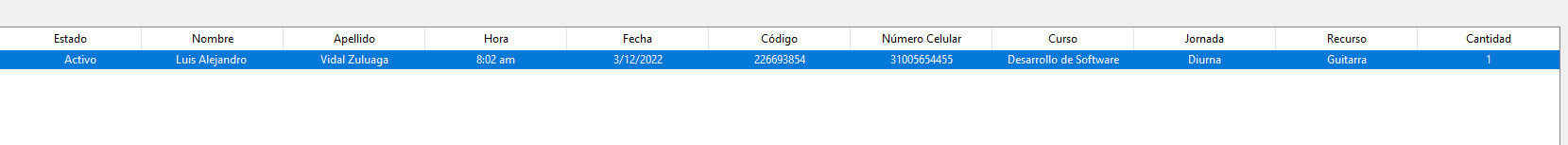


Resultado:

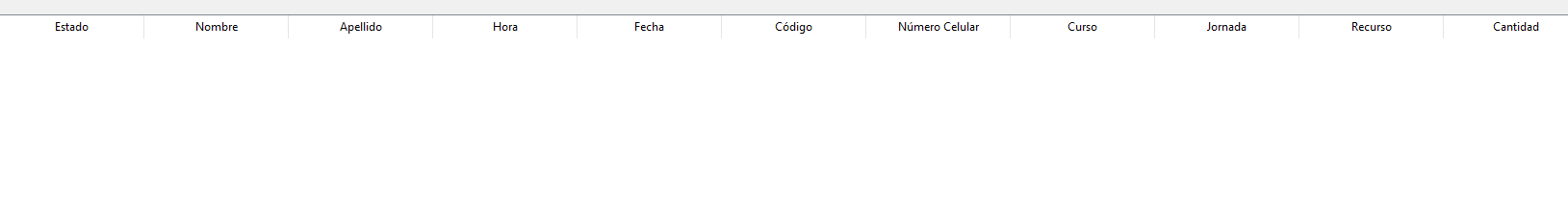


Presionamos el botón:





Resultado:



\*Función listo

Definición de datos:

EA, donaobajaentry, ti=Variables tipo StringVar que contienen los datos ingresados correspondientes a un elemento, como lo son su nombre, si es donación o baja y tipo de elemento

c2, co2=Variables tipo IntVar que contienen los datos ingresados correspondientes a un elemento, como lo son el código y cantidad

tree4=Corresponde a la tabla número 4 para hacer el correspondiente registro

Firma:

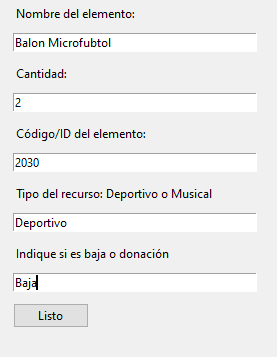
Entradas= EA, t, c2, co2

Salidas=Inserción de estos datos en tree4

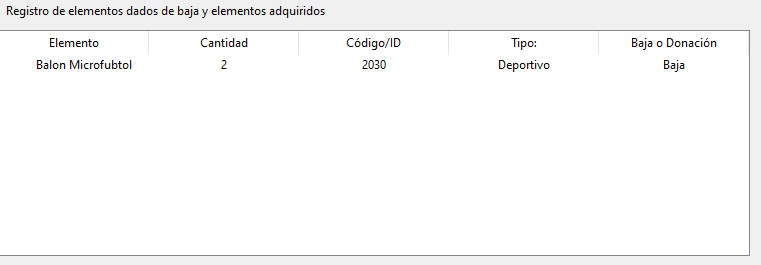
Propósito: Que la función me permita almacenar los datos que se nos preguntan en la tabla de elementos dados de baja o de donaciones

Ejemplos funcionales:

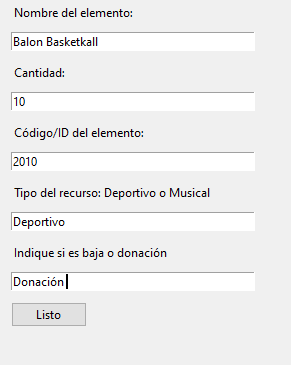
Ingresamos:



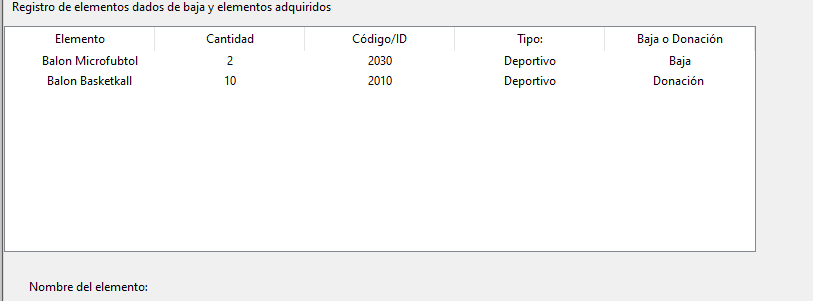
Resultado:



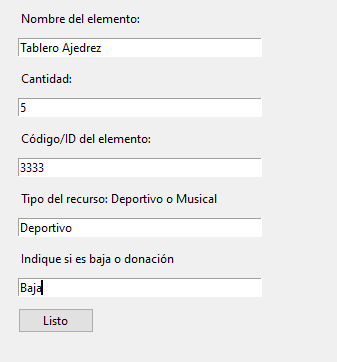
Ingresamos:



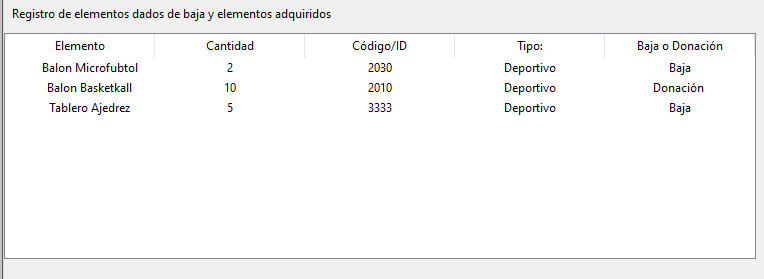
Resultado:



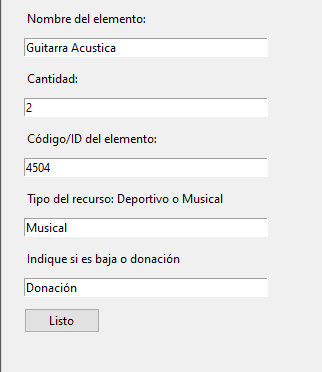
Ingresamos:



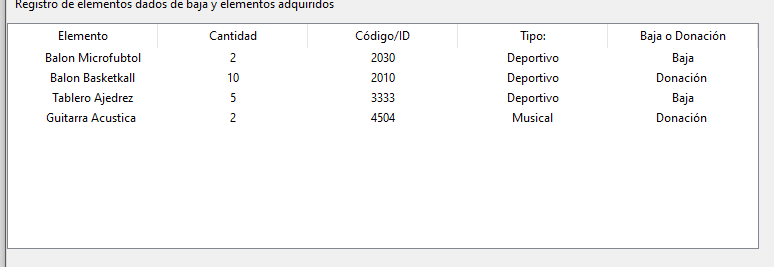
Resultado:



Ingresamos:



Resultado:



\*Función terminarprestamo

Definición de datos:

nombr, apellid, hor, fech, curs, jornad, recurs=Variables tipo StringVar que contienen los datos obtenidos por el usuario

codig, numCelua, cantida=Variables tipo IntVar que contiene los datos obtenidos del usuario

vistarbol3=Treeview correspondiente a la pestaña 4 que almacena el registro de prestamos finalizados

Firma:

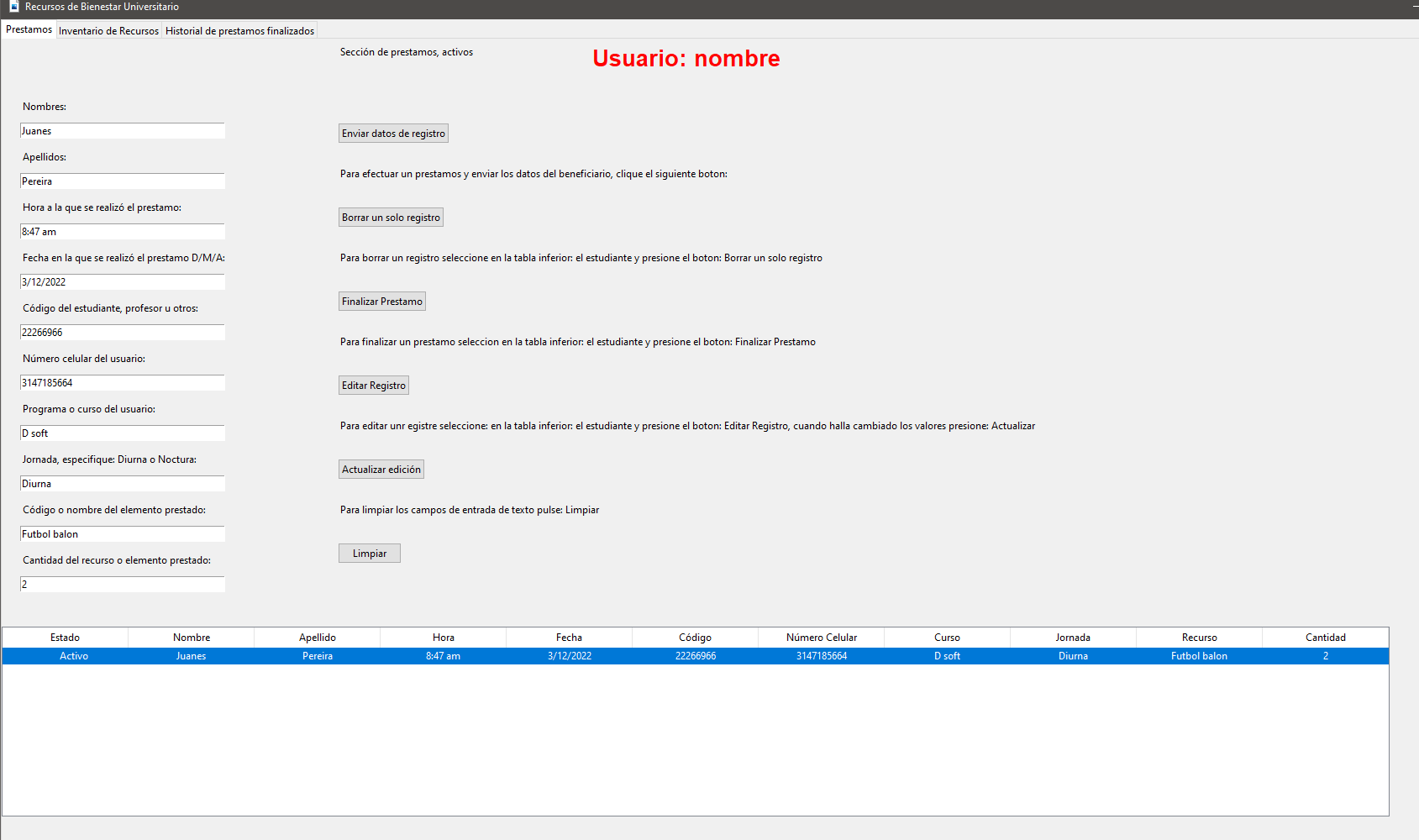
Entradas=nombr, apellid, hor, fech, curs, jornad, recurs, codig, numCelua, cantida

Salida=Inserción de dichos datos en vistarbol3

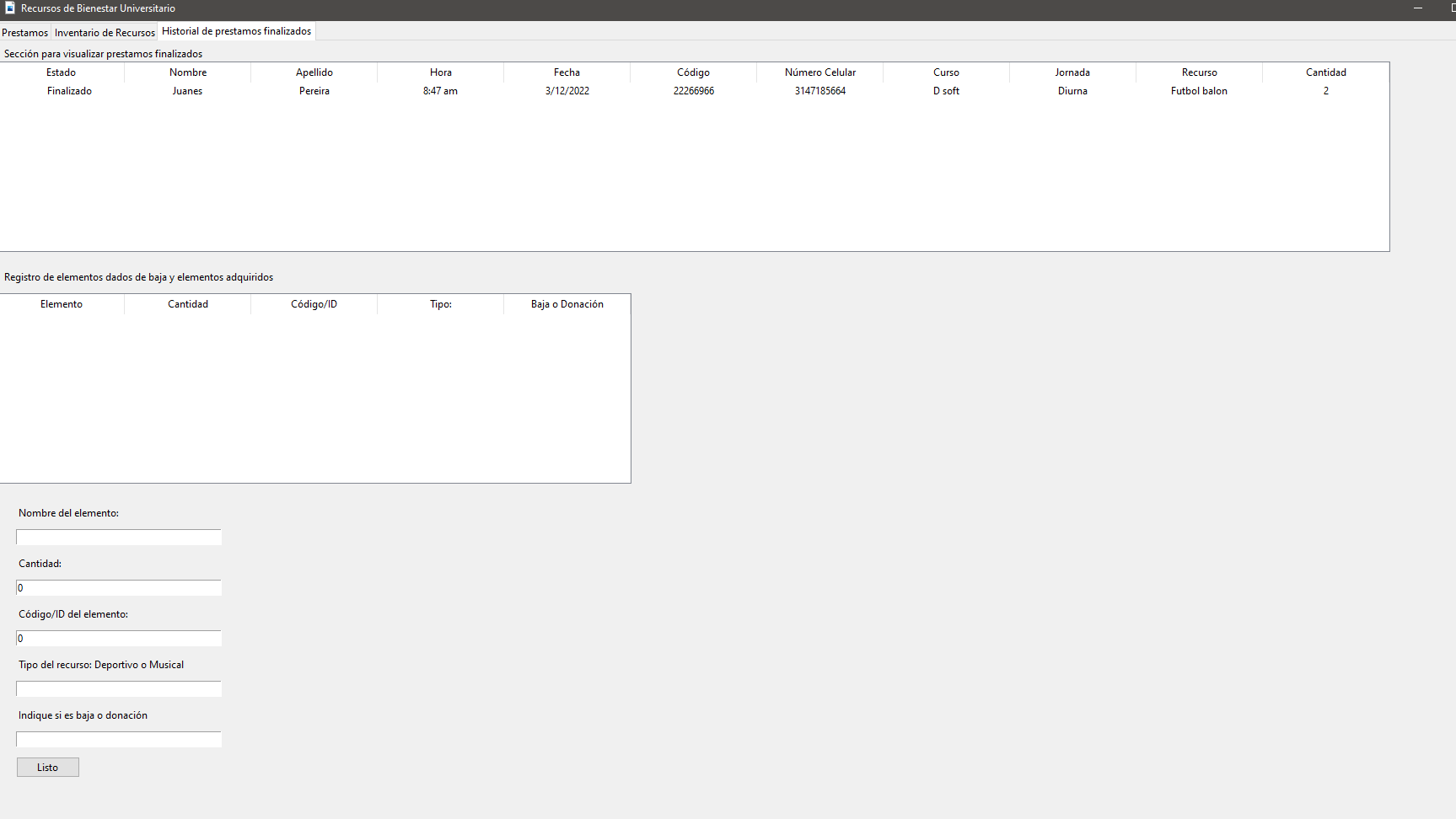
Propósito: Que la función me permita almacenar los datos de prestamos finalizados en una tabla que está en otra pestaña, para comodidad de la persona encargada, este complementa de mejor manera al archivo de texto.

Ejemplos funcionales:

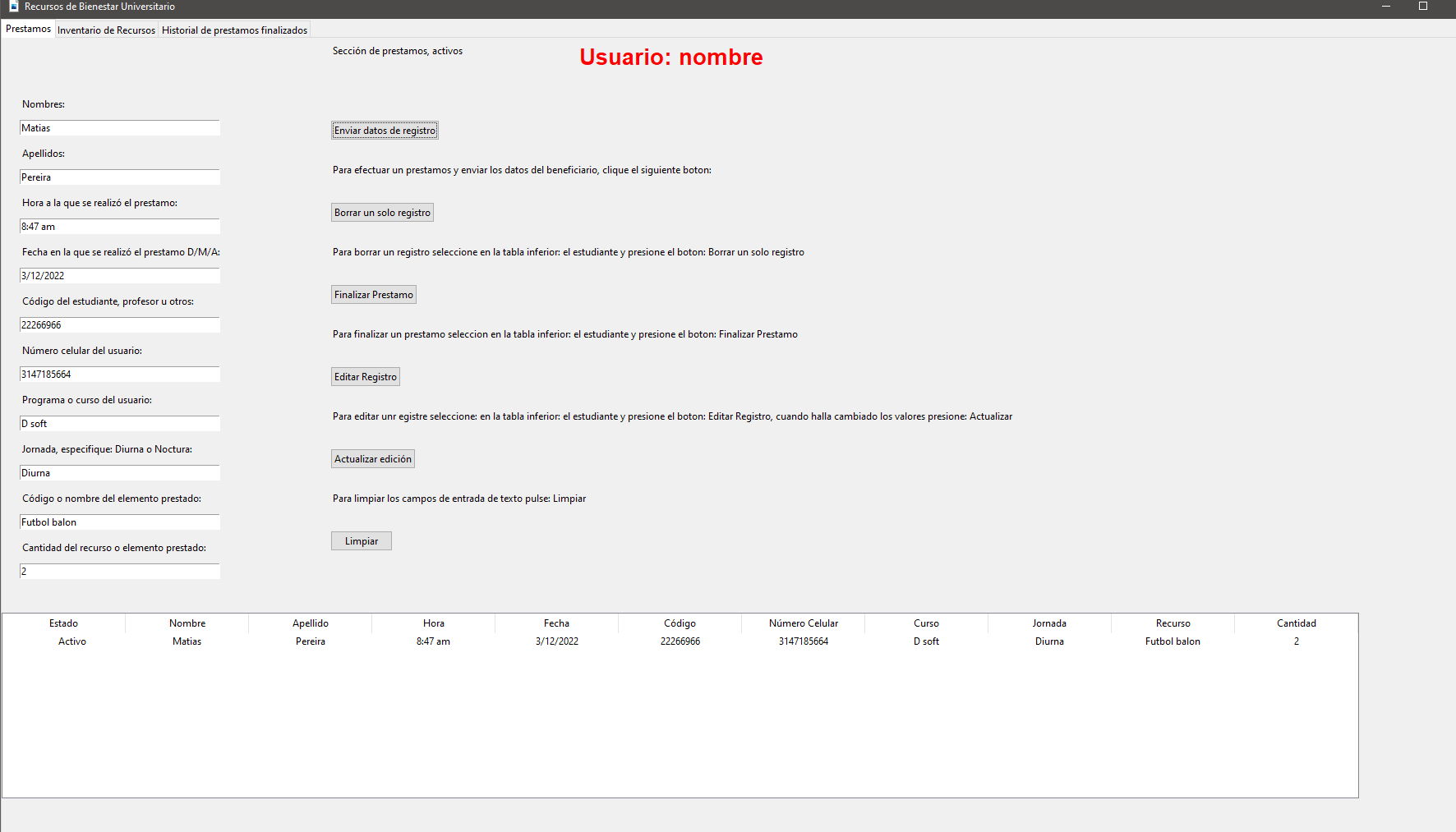
Antes:



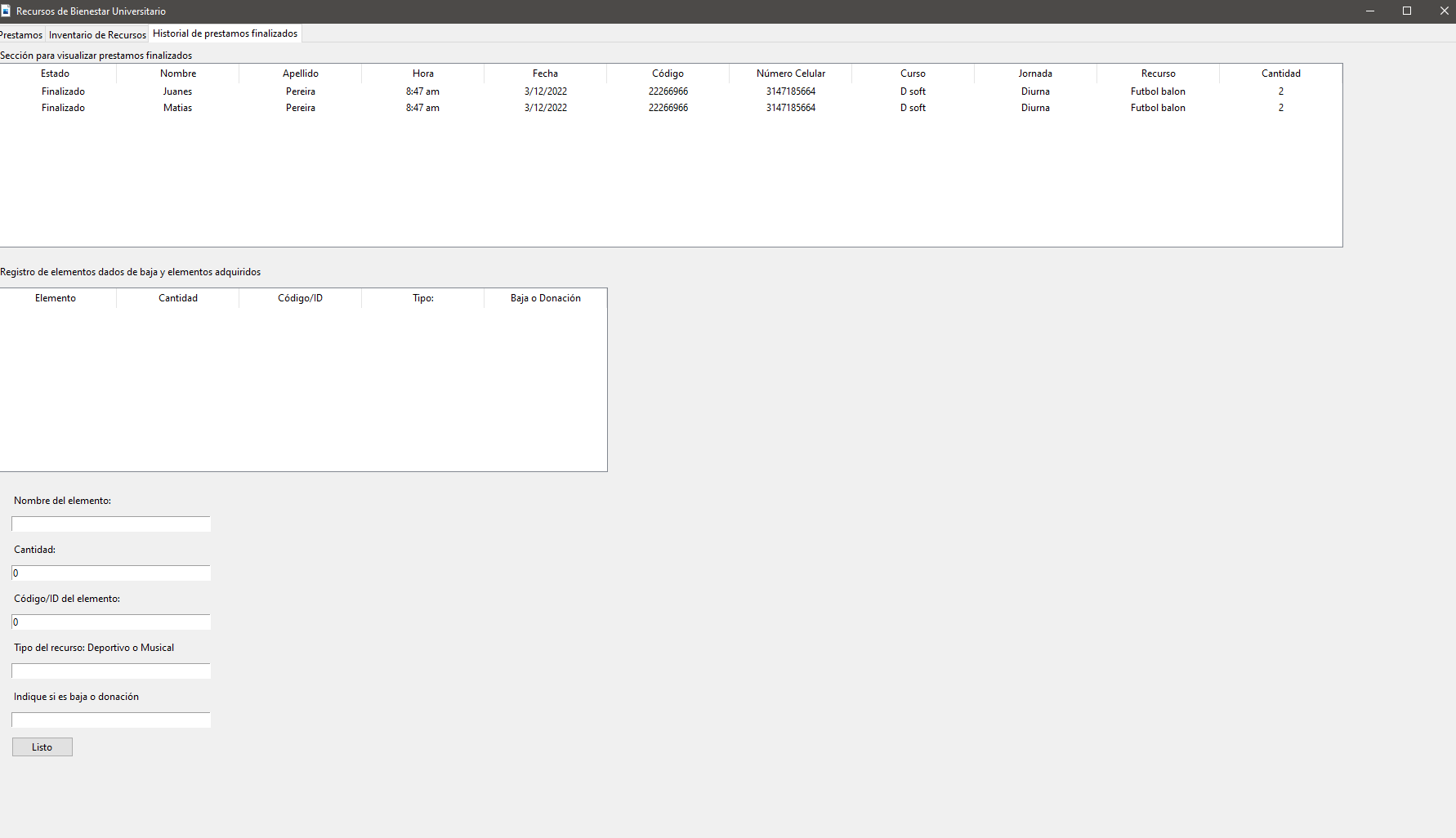
Luego de presionar el botón “Finalizar Préstamo”:



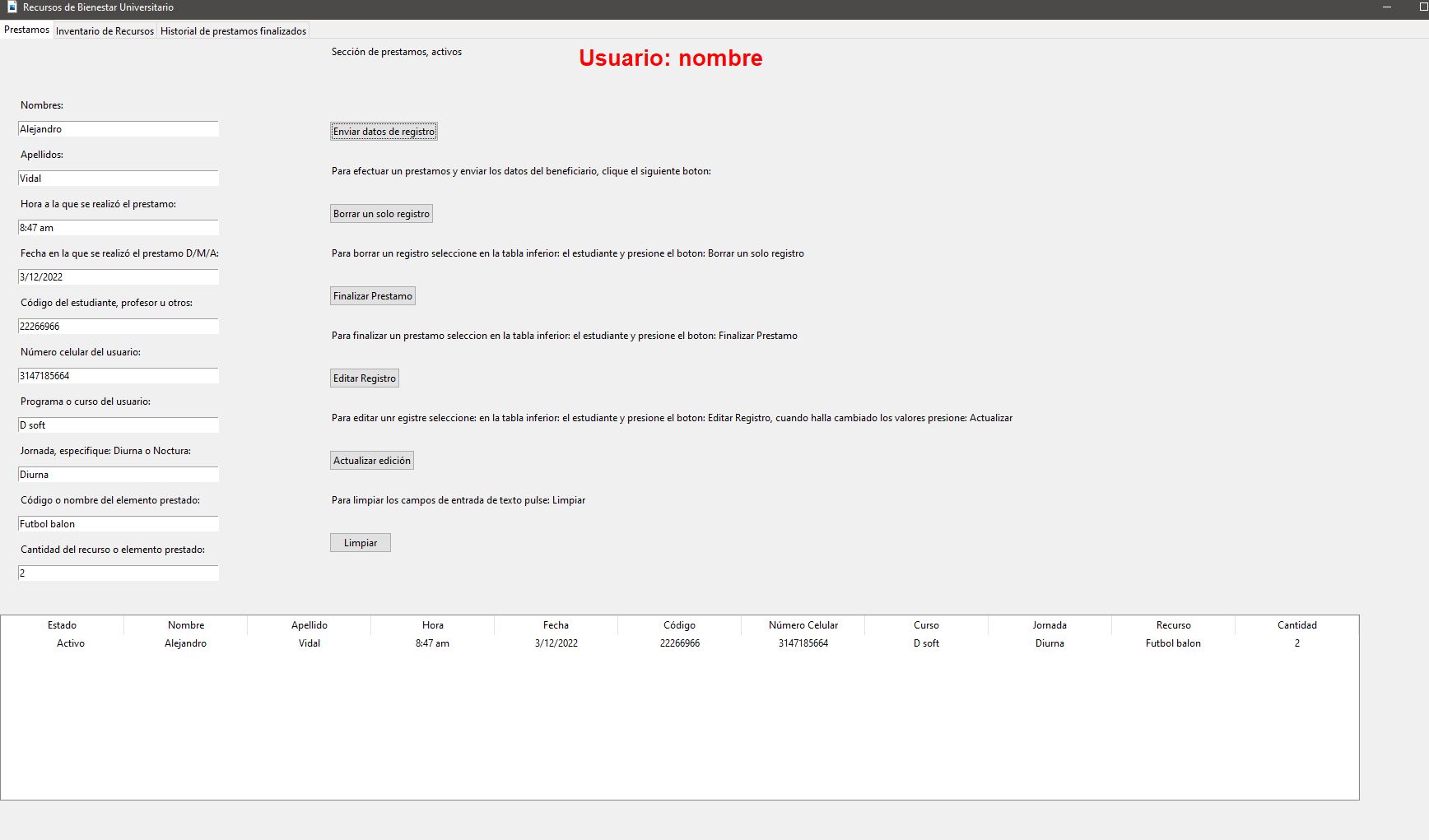
Antes:



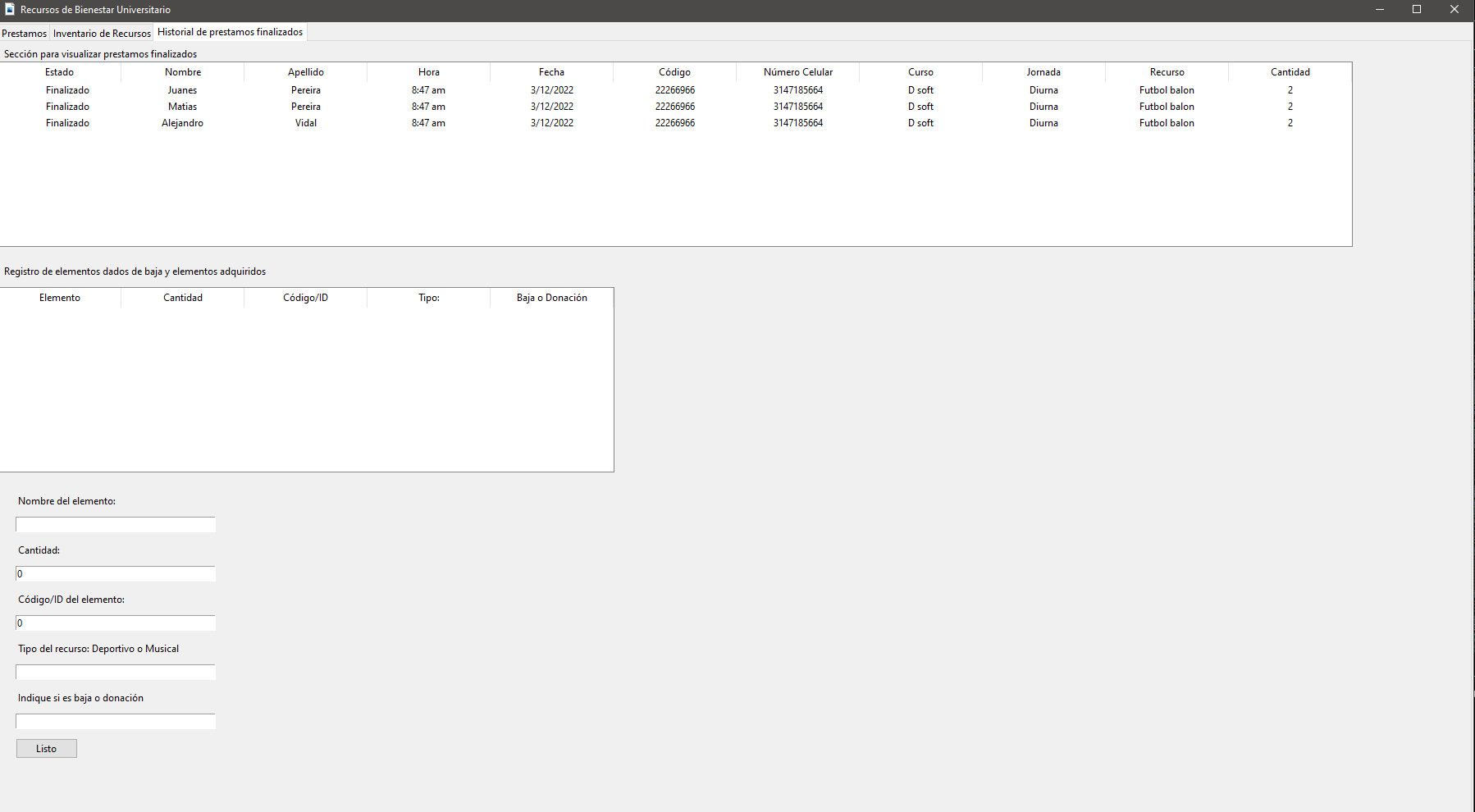
Después:



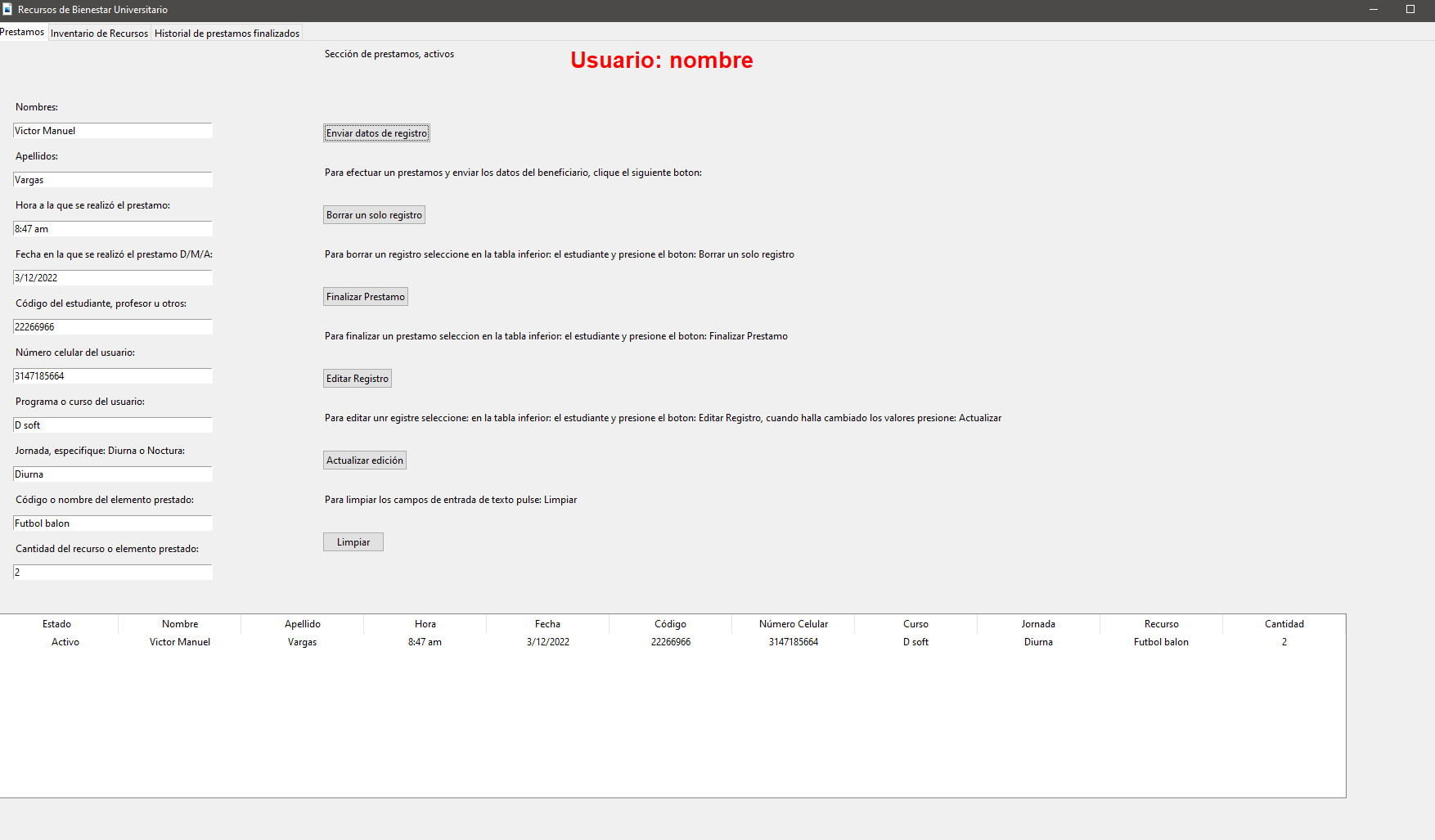
Antes:



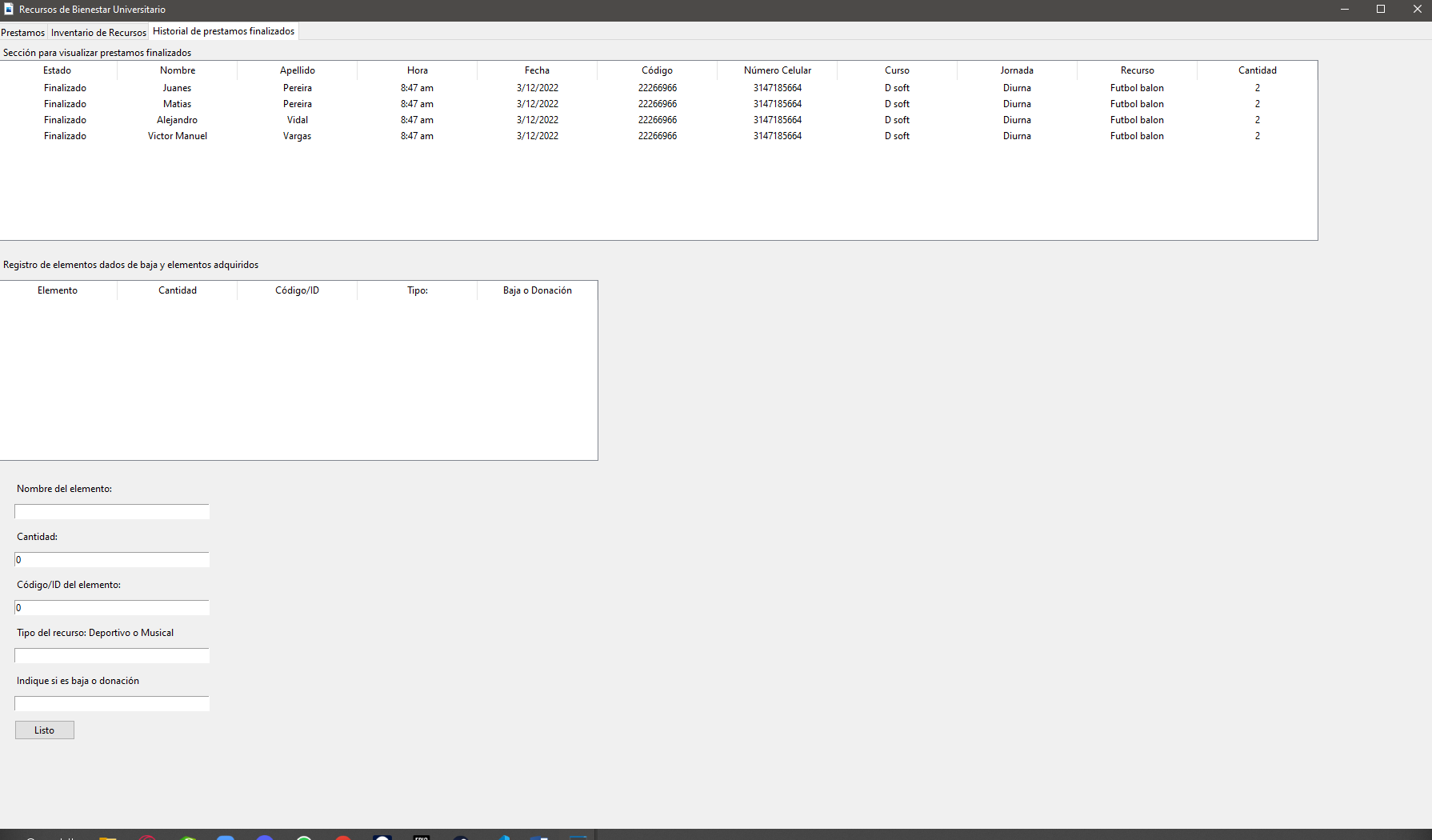
Después:



Antes:



Después:



\*Función eliminarcasillas

Definición de datos:

nombr, apellid, hor, fech, curs, jornad, recurs=Variables tipo StringVar en Entrys que contienen los datos obtenidos por el usuario

codig, numCelua, cantida=Variables tipo IntVar en Entrys que contiene los datos obtenidos del usuario

Firmas:

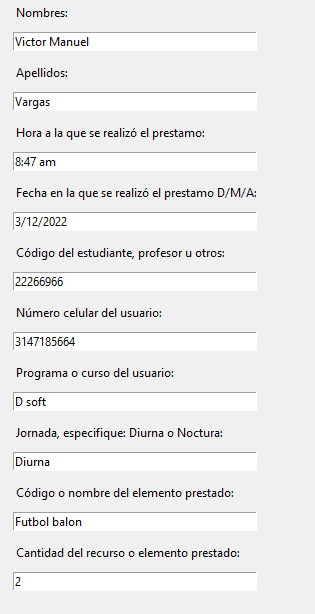
Entradas: Entrys de las variables nombr, apellid, hor, fech, curs, jornad, recurs, codig, numCelua, cantida

Salidas: Entrys de estas variables vacíos listos para ser usados de nuevo

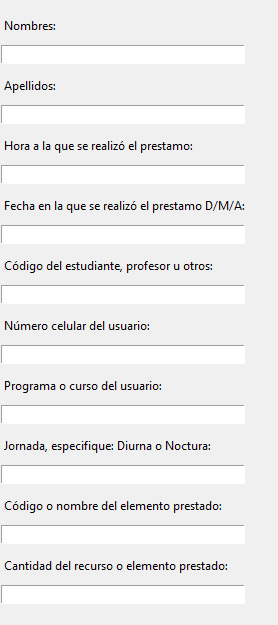
Propósito: Que la función me limpie los datos de los entrys una vez yo ya los halla mandado al registro

Ejemplos funcionales:

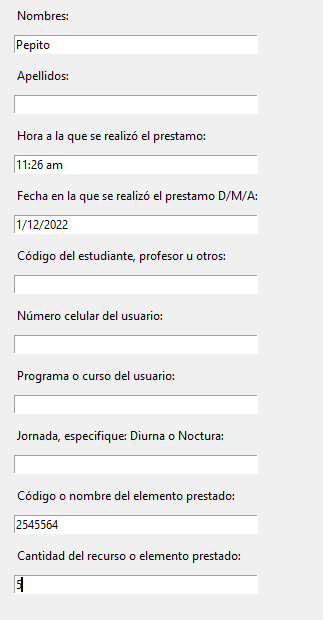
Antes:



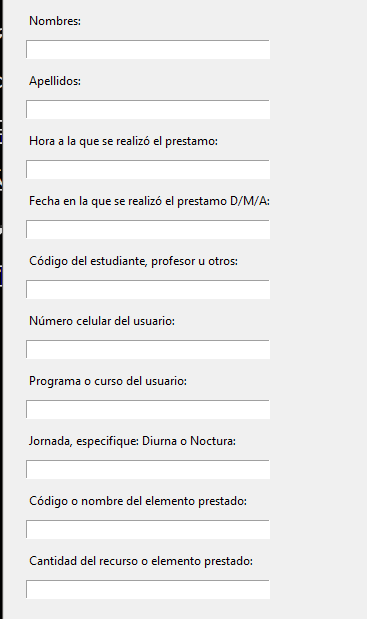
Después:



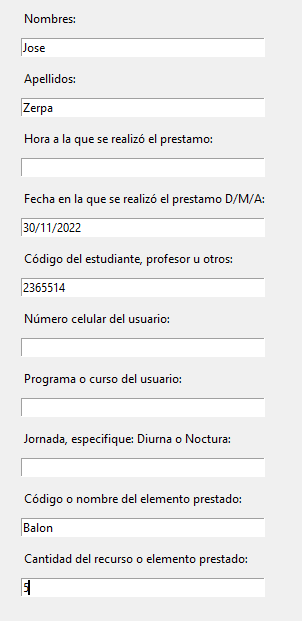
Antes:



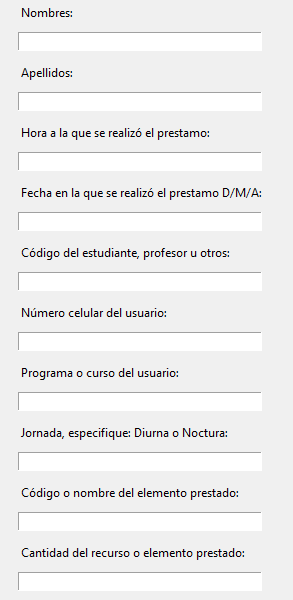
Después:



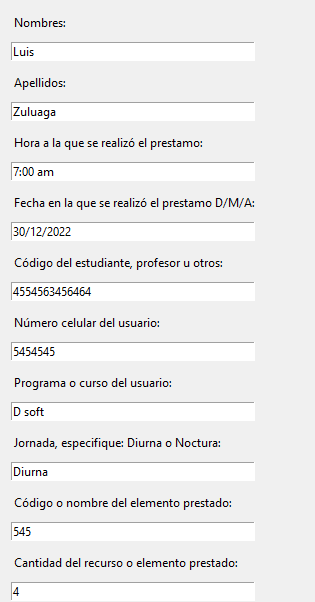
Antes:



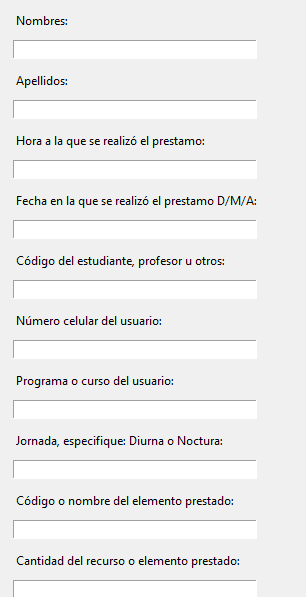
Después:



Antes:



Después:



Función enviardatos

Definición de datos

nombr, apellid, hor, fech, curs, jornad, recurs=Variables tipo StringVar en Entrys que contienen los datos obtenidos por el usuario

codig, numCelua, cantida=Variables tipo IntVar en Entrys que contiene los datos obtenidos del usuario

Firmas:

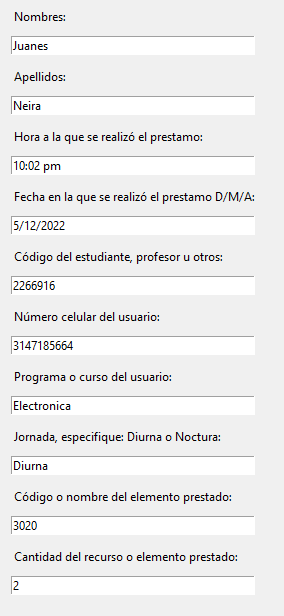
Entradas: Entrys de las variables nombr, apellid, hor, fech, curs, jornad, recurs, codig, numCelua, cantida

Salidas Introducción de estos datos a la tabla trev donde quedan registrados los préstamos

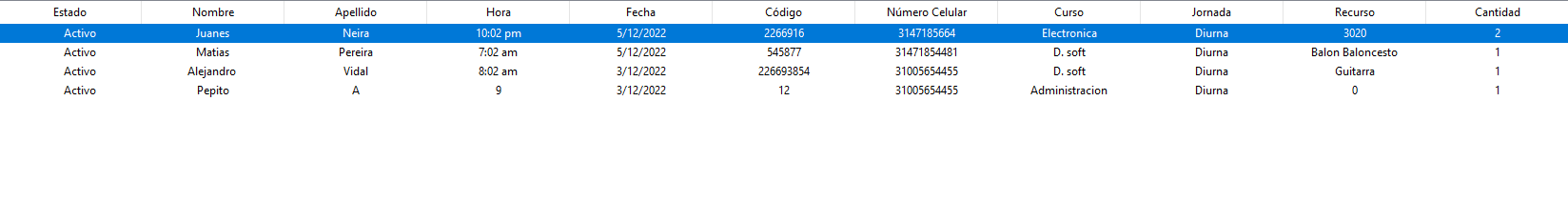
Propósito Que la función introduzca y capte los datos provenientes de entrys en la tabla trev

Ejemplos funcionales

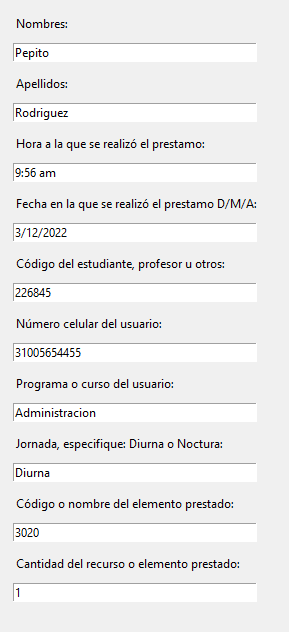
Entramos los datos



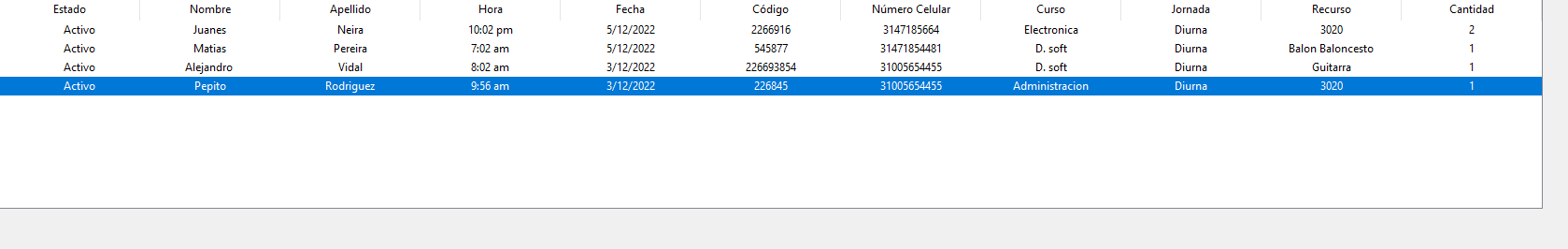
Después



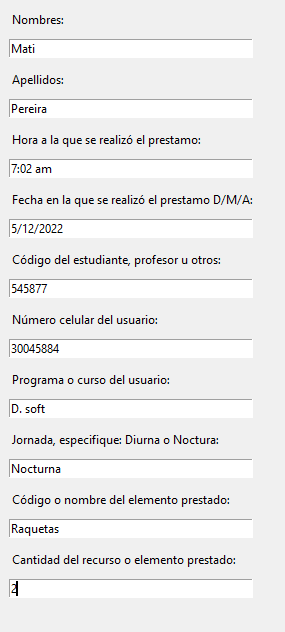
Antes



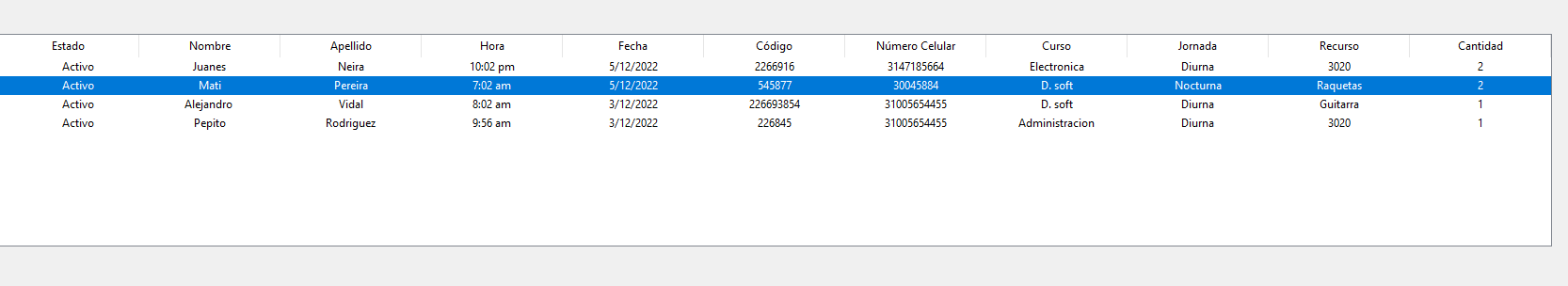
Después



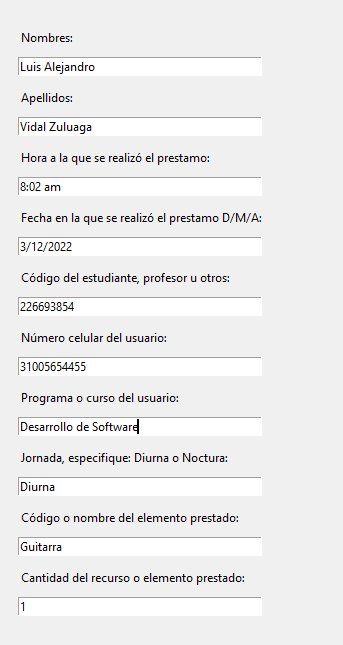
Antes



Después



Antes



Después

