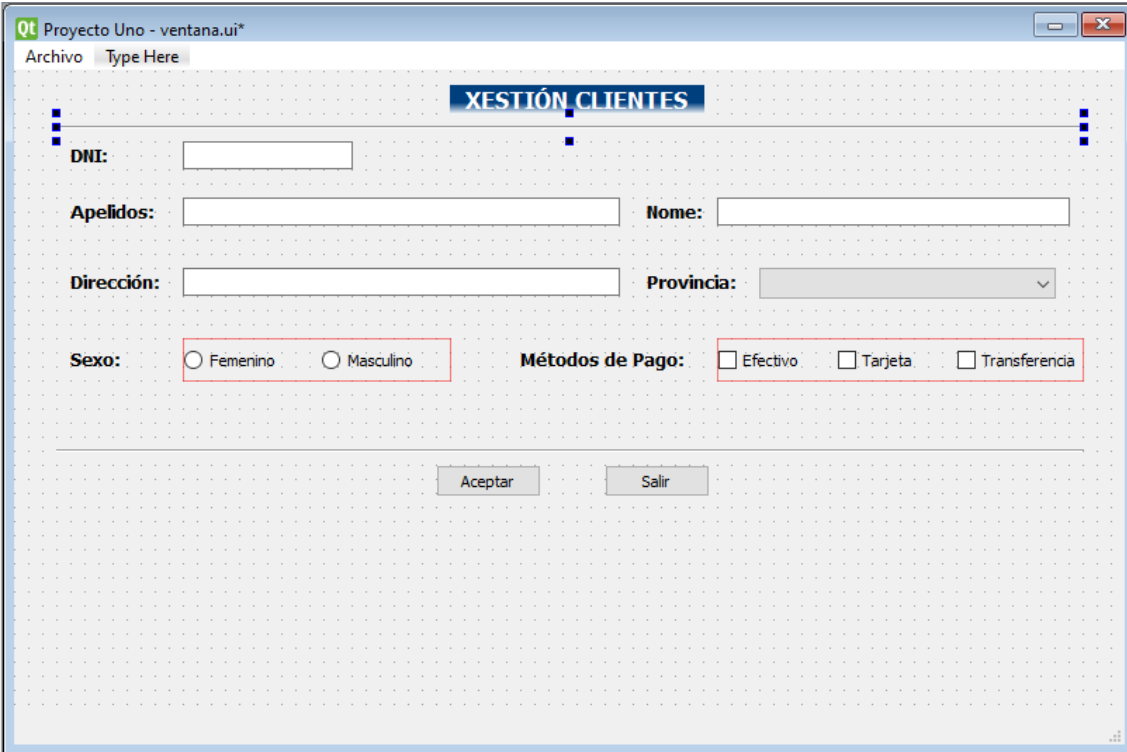


Combobox

El tercer widget más utilizado es el **combobox** o **cuadro combinado**.

Tradicionalmente, es una **combinación entre una lista desplegable y un cuadro de texto**, que permite al usuario introducir cualquier valor directamente en el teclado o, alternativamente, seleccionar un valor de la lista. El término "lista desplegable" se utiliza a veces para significar "cuadro combinado", aunque no todos los lenguajes lo consideran de la misma forma.

Modificar la IU para obtener el siguiente resultado:








The screenshot shows a Qt application window titled "Proyecto Uno - ventana.ui*" with a menu bar containing "Archivo" and "Type Here". The main content area has a title bar "XESTIÓN CLIENTES" and a form with the following fields:

- DNI:** A text input field.
- Apellidos:** A text input field.
- Nombre:** A text input field.
- Dirección:** A text input field.
- Provincia:** A combobox (dropdown menu).
- Sexo:** Two radio buttons labeled "Femenino" and "Masculino".
- Métodos de Pago:** Three checkboxes labeled "Efectivo", "Tarjeta", and "Transferencia".

At the bottom of the form are two buttons: "Aceptar" and "Salir".

Observamos los cambios en la interfaz gráfica: la inclusión de una caja de texto para introducir la dirección (por simplificar, obviamos cajas de texto para número de vivienda, piso, bloque, etc.). Además, supondremos que será en Galicia con lo que incluiremos en el cuadro combinado de la derecha solamente las provincias gallegas.

Esquema de los widgets introducidos hasta el momento. Obsérvese el nombre intuitivo dado en cada caso.

Object	Class
▼ MainWindow	QMainWindow
▼  centralwidget	QWidget
btnAceptar	QPushButton
btnSalir	QPushButton
cmbProv	QComboBox
editApel	QLineEdit
editDir	QLineEdit
editDni	QLineEdit
editNome	QLineEdit
▼  horlaySex 	QHBoxLayout
rbtFem	QRadioButton
rbtMasc	QRadioButton
▼  horlayPago 	QHBoxLayout
chkEfectivo	QCheckBox
chkTarjeta	QCheckBox
chkTransf	QCheckBox
lblApel	QLabel
lblDir	QLabel
lblDni	QLabel
lblMetpago	QLabel
lblNome	QLabel
lblProv	QLabel
lblSexo	QLabel
lblTitCli	QLabel
lblValido	QLabel
lineInf	Line
lineSup	Line
▼ menubar	QMenuBar
▼ menuArchivo	QMenu
actionSalir	QAction
statusBar	QStatusBar

Ahora trabajemos con el código del **combobox**:

```
i.stateChanged.connect(clients.Clients.selPago)

...
conexión eventos input widgets y otras funciones
...

clients.Clients.cargarProv()
var.ui.cmbProv.activated[str].connect(clients.Clients.selProv)
var.ui.editDni.editingFinished.connect(events.Eventos.validoDNI)

if __name__ == '__main__':
```

Líneas de código añadidos en **main.py**

```
def cargarProv():
    """
    esta solución es provisional en su momento lo haremos de otra forma
    cargando los registros desde una base de datos
    """
    try:
        prov = ['A Coruña', 'Lugo', 'Ourense', 'Pontevedra']
        for i in prov:
            var.ui.cmbProv.addItem(i)
    except Exception as error:
        print('Error: %s ' % str(error))
```

Función **cargarProv**

```
def selProv(prov):
    try:
        print('Has seleccionado la provincia de ', prov)
        return prov
    except Exception as error:
        print('Error: %s ' % str(error))
```

Función **selProv**

En el fichero **main.py** añadimos dos líneas:

- la primera ***clients.Clients.cargarProv()*** es una función que se lanza cuando arranca la aplicación. Su función es cargar el cuadro combinado o combobox.
- la segunda llama al evento **activated**. Este evento tiene como función cargar en una variable el valor seleccionado del combobox, de ahí que haya una tupla con un ***string***.

En el fichero **clients.py** tenemos dos funciones:

- la primera función ***cargarProv*** se encarga de cargar en el *combobox* el listado de provincias.
- la segunda función ***selProv*** nos devuelve la provincia seleccionada.