

Cómo crear una GUI Responsive con QT

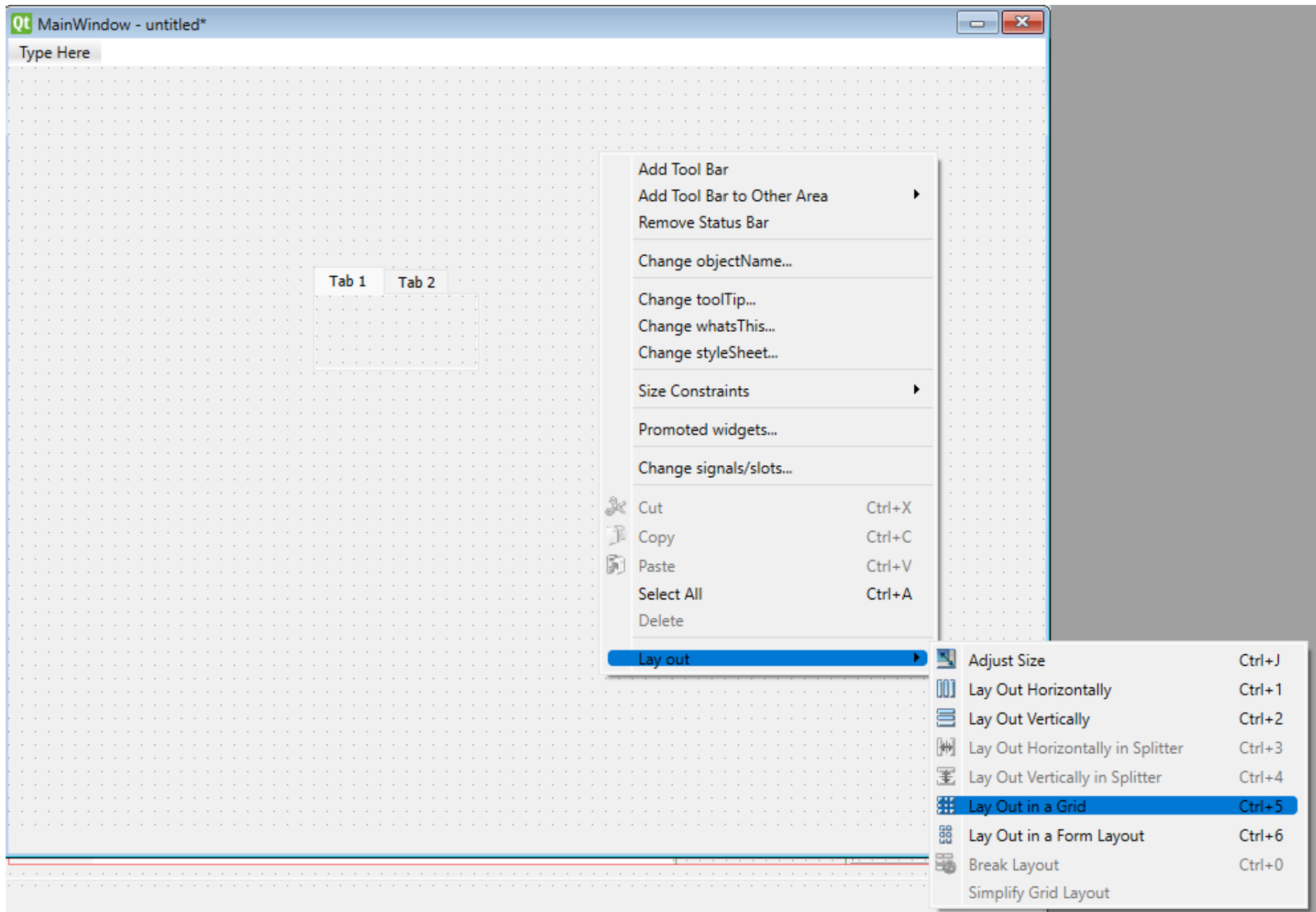
Autor: Salvador Jesús Megías Andreu

La herramienta que vamos a usar para crear un ejemplo sencillo de una GUI responsive es **QT designer**. Es super recomendable usar esta herramienta, puesto que aunque puedas programar en QT a pelo desde 0, esta herramienta te puede facilitar mucho la vida y ahorrarte gran parte de la construcción de una GUI.

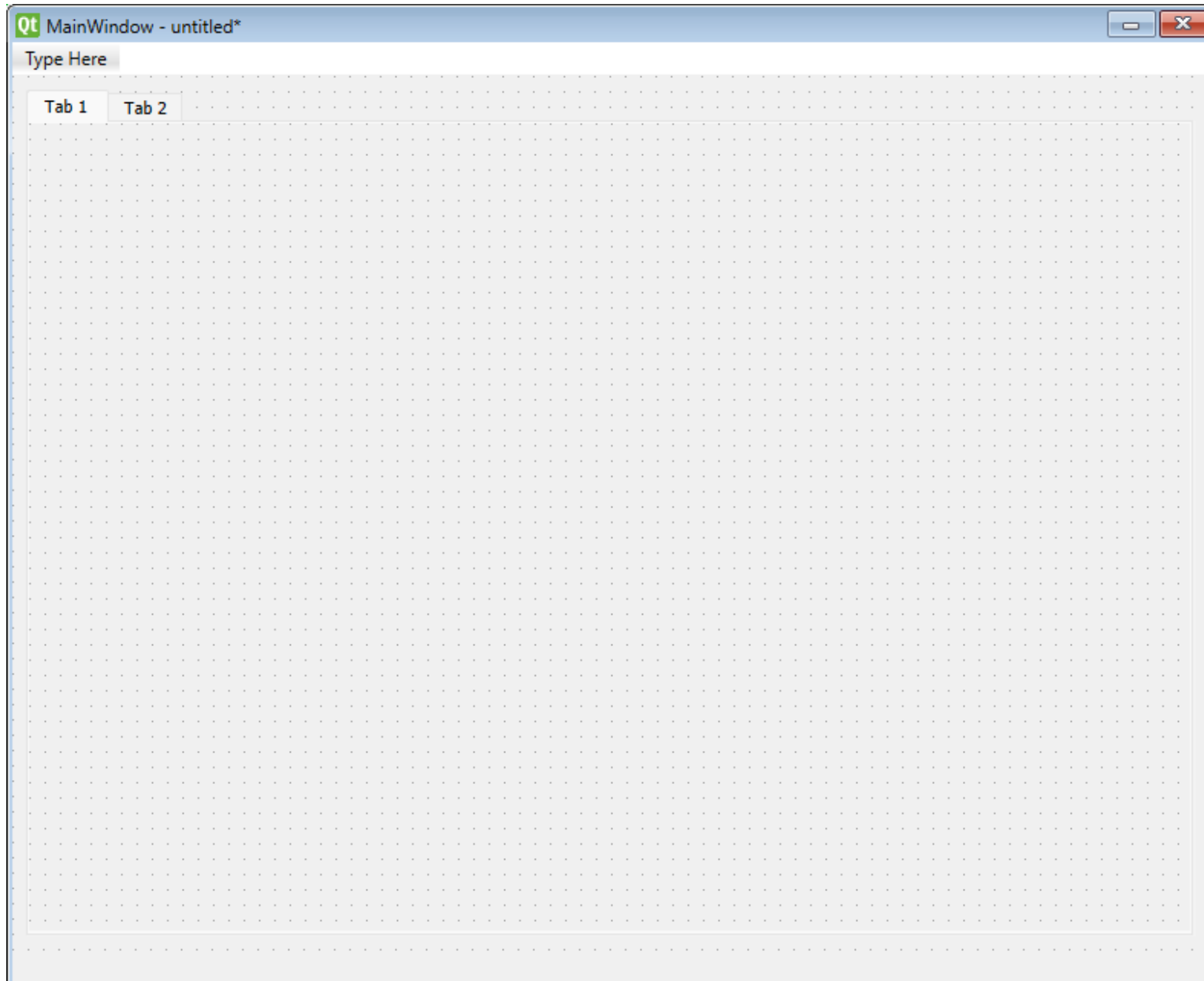
Los elementos clave para conseguir que nuestra GUI sea responsive son los **Layouts**. Los **layouts** son, por así decirlo, contenedores integrados en QT Designer con propiedad responsive intrínsecas. De tal forma que se colocas un **widget** (como puede ser un Dial, un Label ...etc) dentro de un layout, este adquirirá las propiedades responsive del contenedor (layout) en el que esté contenido.

Vamos a realizar un ejemplo sencillo para poner en práctica lo anteriormente dicho, para poder haceros una idea de como funciona y a raíz de ese ejemplo podréis extrapolar esta base a un proyecto de una GUI responsive más grande y compleja.

Lo primero de todo voy a crear un MainWindow en QT Designer y meter un **Tab** en el Mainwindow, luego le damos al botón derecho del ratón y: -> **Layout** -> **Lay Out in a Grid**. Hacemos esto para asignar a toda la ventana inicial un layout y asi hace que la ventana inicial pueda ser responsive (que pueda variar el tamaño del GUI), debido a esto, el **Tab se ajusta automáticamente al tamaño de la ventana** (como se puede observar):



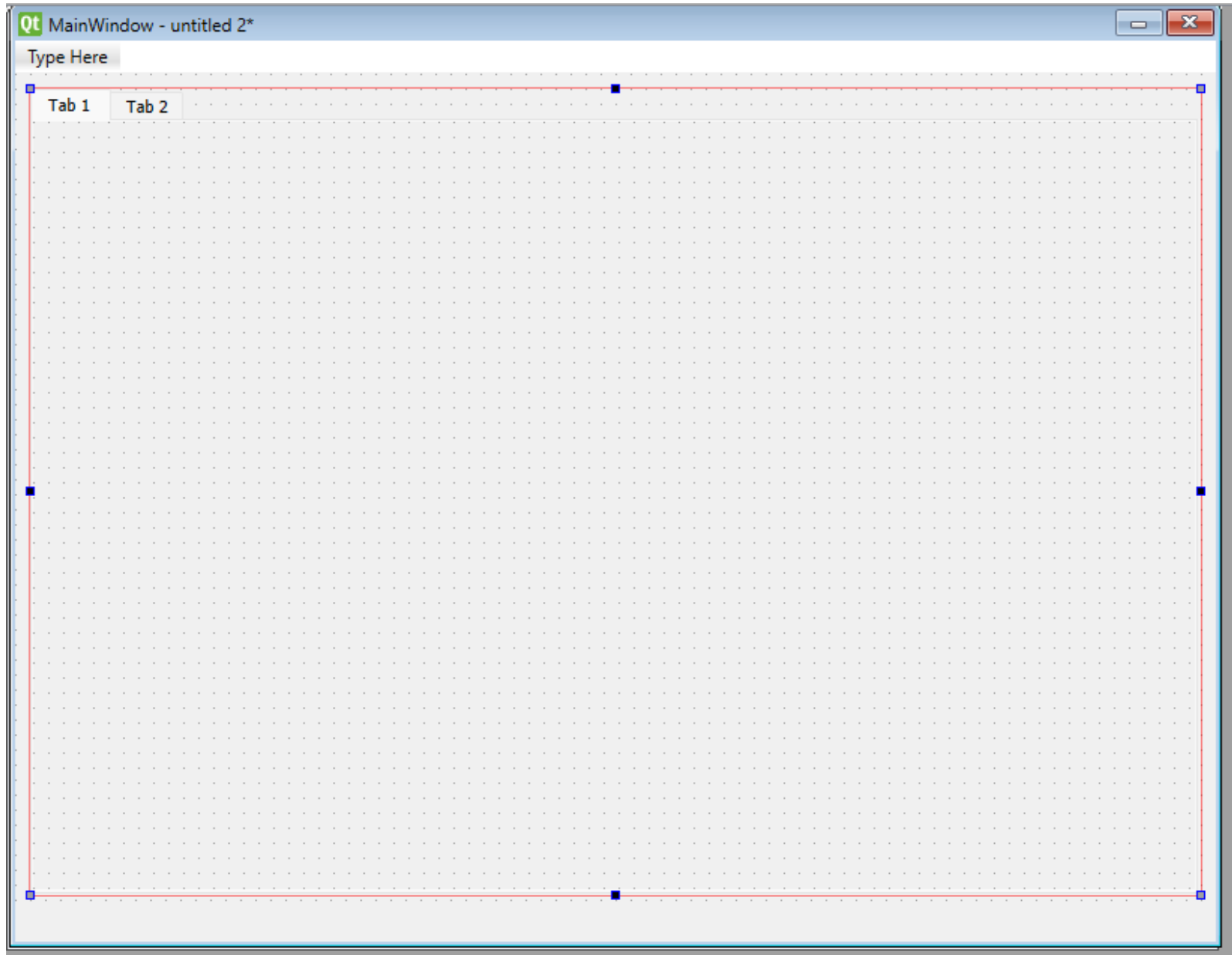
Resultado del paso anterior:



Si no te deja ponerle el **Lay Out in a Grid** es porque la ventana no debe estar vacía, por lo que debes meterle el **Tab** u otro **Widget** para poder hacerlo (o ponerle un Grid Layout al Main Window, también sirve)

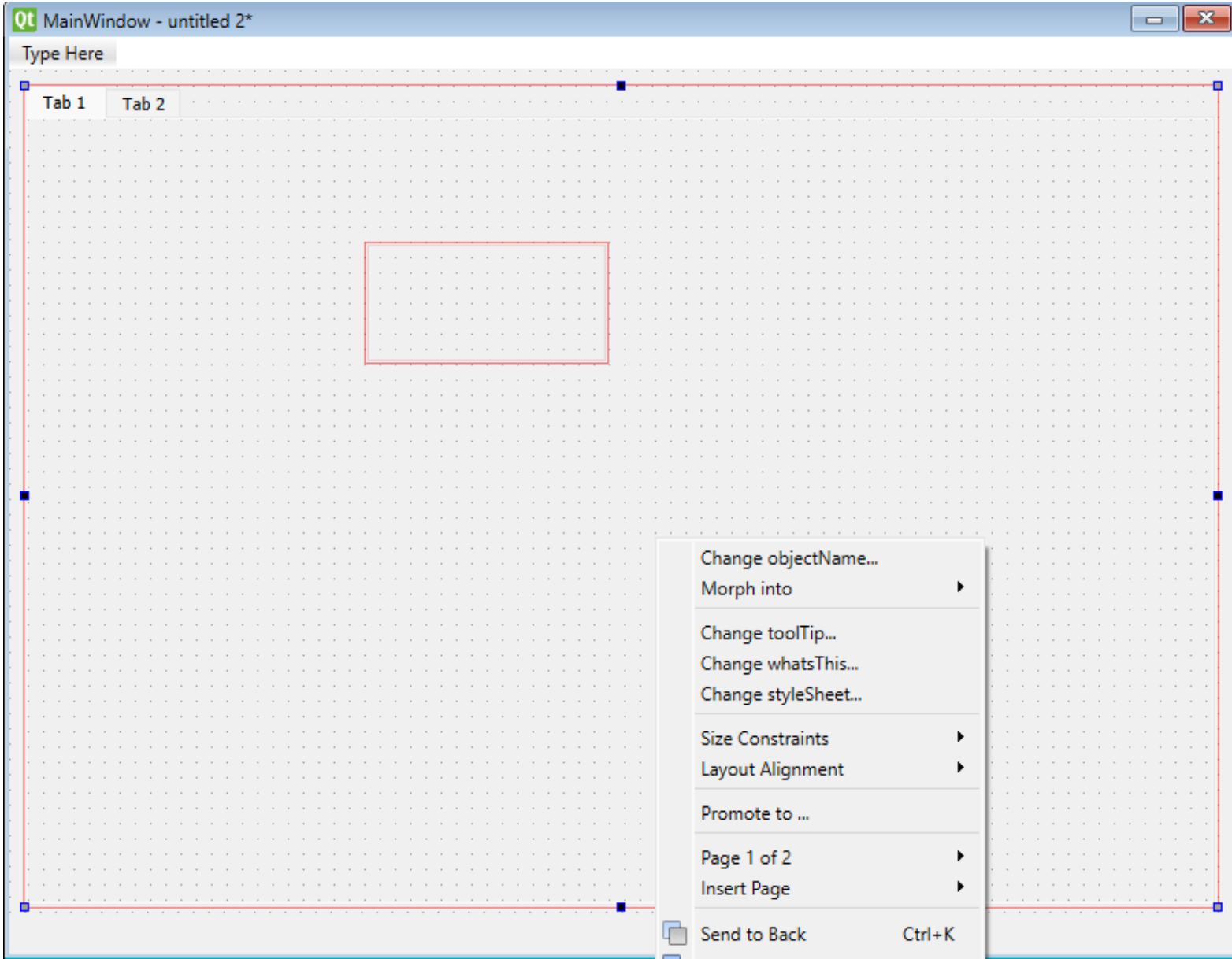
Ejemplo aplicando un Grid Layout en vez de lo dicho anteriormente (el resultado es el mismo):

Para este ejemplo vamos a usar un **Tab** puesto que para poner un layout a un Tab es un proceso un poco distinto al resto. Por lo que insertamos el widget **Tab** (Al soltar el Tab sobre el Grid Layout, el Tab se ajusta automáticamente al tamaño de la ventana):

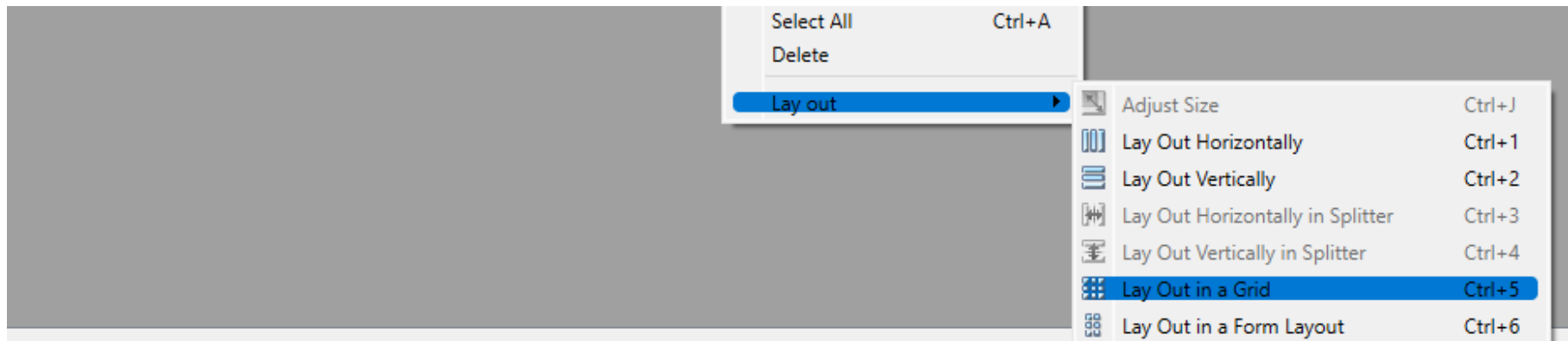


A partir de aquí, debe ser igual lo que se haga.

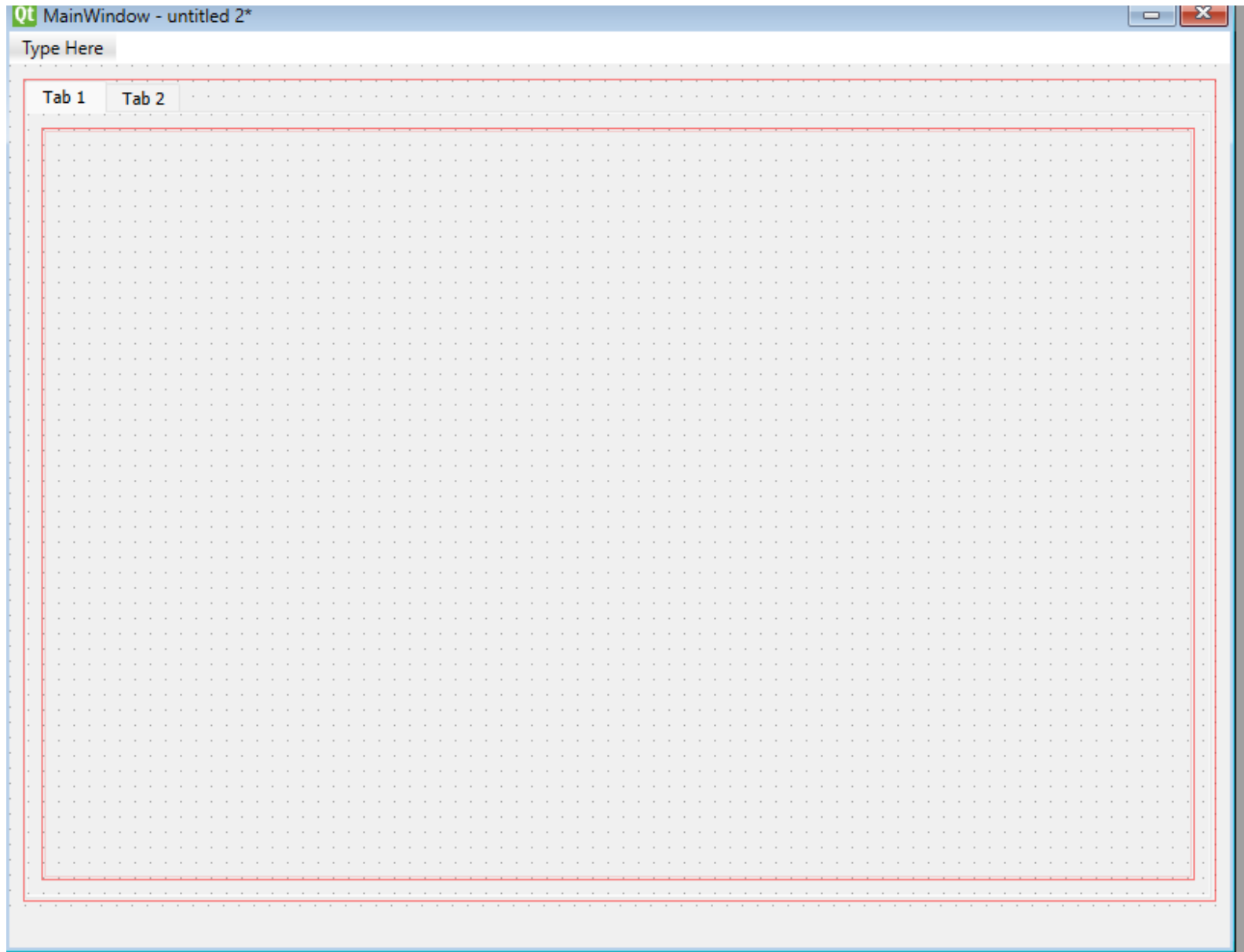
Luego metemos un **Grid Layout** en el Tab, seleccionamos el Tab con el botón derecho del ratón y de damos a: -> **Lay out** -> **Lay Out in a Grid**.
Obteniendo como resultado un Layout integrado en el Tab completo. Con esto conseguimos que todos los widgets que metamos en el Tab sean totalmente responsive con respecto al tamaño de la ventana del GUI.



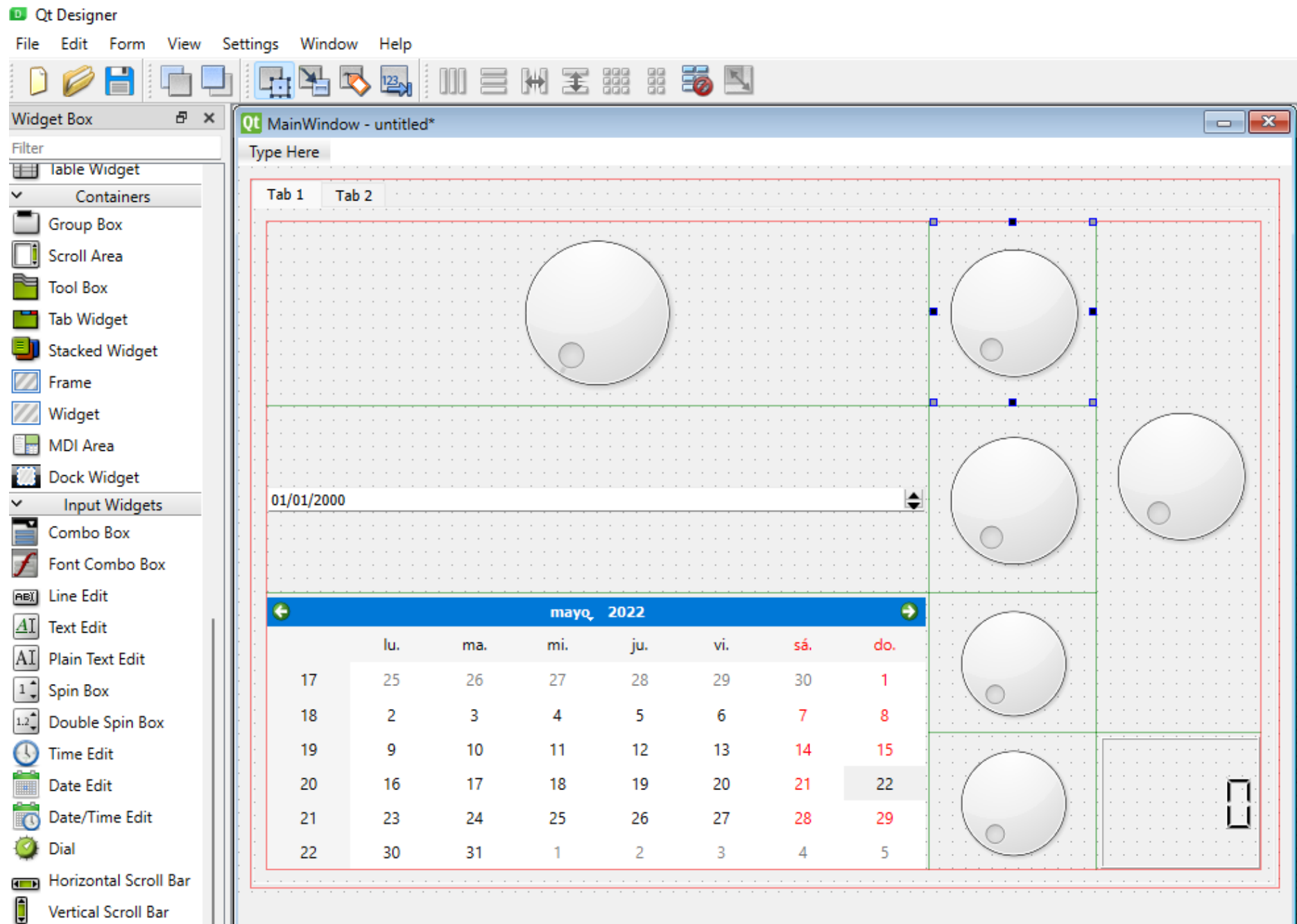
- Change objectName...
- Morph into ▶
- Change tooltip...
- Change whatsThis...
- Change styleSheet...
- Size Constraints ▶
- Layout Alignment ▶
- Promote to ...
- Page 1 of 2 ▶
- Insert Page ▶
- Send to Back Ctrl+K
- Bring to Front Ctrl+L
- Cut Ctrl+X
- Copy Ctrl+C
- Paste Ctrl+V
- Select Ancestor ▶



Resultado del paso anterior:



Luego vamos a incorporar elementos o widgets al Layout del Tab (Diales, calendario....etc):



No he podido grabar las demo de otro forma, capturando la pantalla me lo bloqueaba

A continuación veremos una demo de la GUI responsive anteriormente creada:

In [4]:

```
from IPython.display import HTML

HTML("""
<div align="middle">
<video width="80%" controls>
    <source src="./multimedia/demo1.mp4" type="video/mp4">
</video></div>""")
```

Out[4]:

También voy a mostrar otra demo de una GUI responsive que yo mismo creé para mi TFG (en su versión casi inicial):

In [5]:

```
from IPython.display import HTML

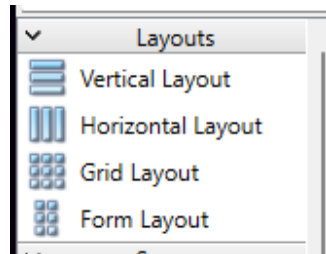
HTML("""
<div align="middle">
<video width="80%" controls>
```

```
<source src="./multimedia/demo2.mp4" type="video/mp4">
</video></div>""")
```

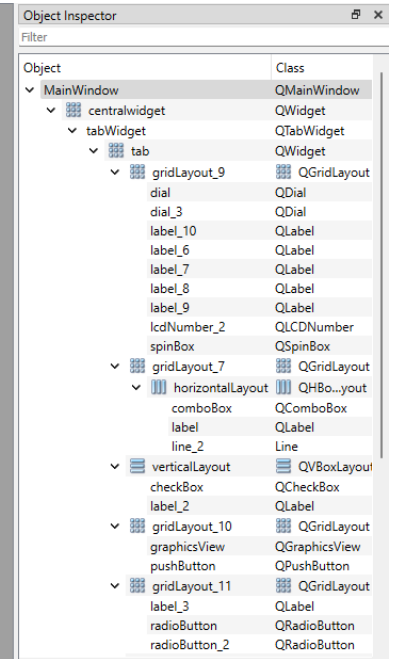
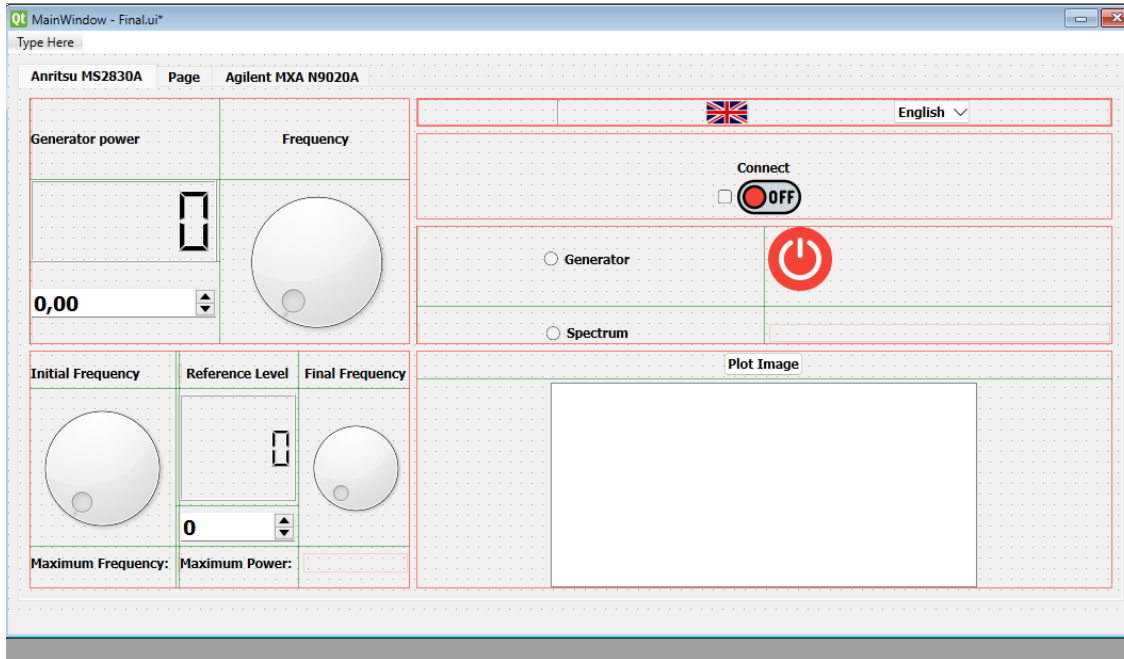
Out[5]:



También puedes hacer uso de otro tipo de Layouts (verticales, horizontales...) estos recomiendo que los utilicéis dentro de un **Grid Layout** para menos lío, estos sirven a la hora de colocar widgets, que se coloquen de forma vertical u horizontal, e ir jugando con todo esto para que los elementos que hay en la GUI estén colocados más o menos donde queráis que estén colocados y que sea de manera responsive:



En la última demo por ejemplo se puede observar que yo hice uso de layouts horizontales y verticales para colocar los elementos de la GUI más o menos donde yo quería que estuviesen para que se viese bien:



In []: