

Introducción a Qt

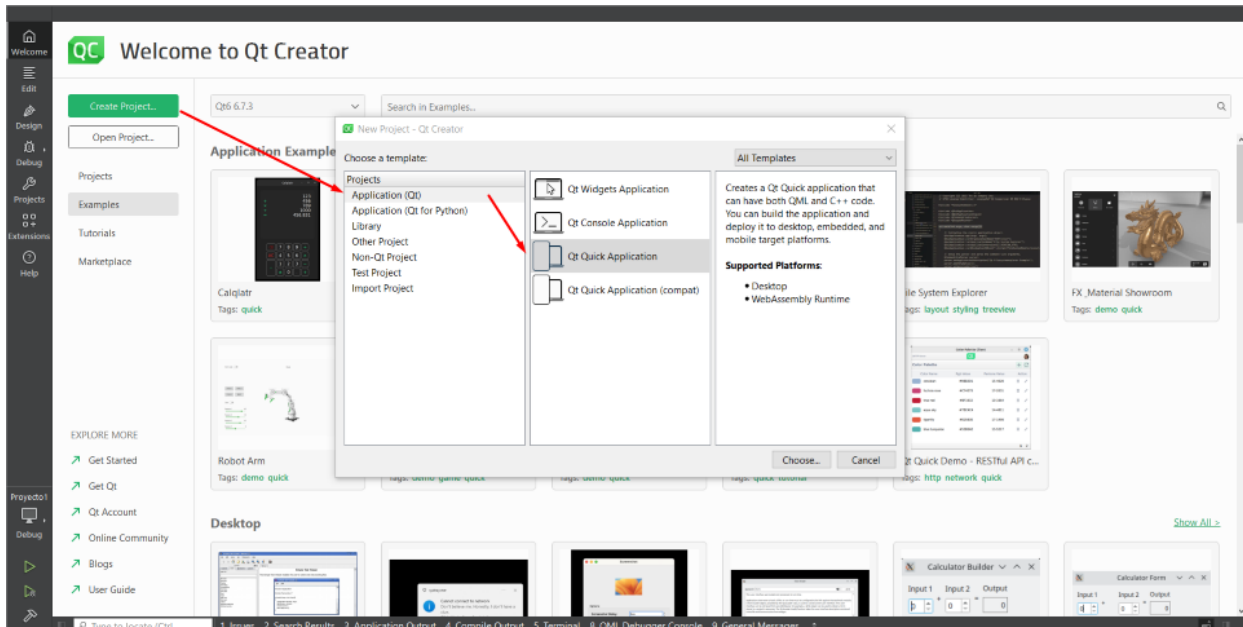
Índice

Introducción al diseño con Qt Quick y QML.....	3
Creación de un proyecto con Qt Quick	3
Primeros pasos con Qt Quick.....	6
Creación de componentes.....	7
Propiedades	10
Uso de componentes	11
Crear propiedades para un componente.....	12
Crear estados para un componente	13
Enlaces.....	15
Transiciones.....	18
Text Field y Button	19
Button.....	22
Connections	23

Introducción al diseño con Qt Quick y QML

Creación de un proyecto con Qt Quick

Para empezar a trabajar, el primer paso será crear un proyecto en blanco. El proyecto será de Aplicación (Qt) y de tipo Qt Quick.



Elegimos la ruta que queremos para almacenar nuestro proyecto

Qt Quick Application

Location

Details

Kits

Summary

Project Location

Creates a Qt Quick application that can have both QML and C++ code. You can build the application and deploy it to desktop, embedded, and mobile target platforms.

Name:

Ejemplo

Create in:

E:\QtProjects

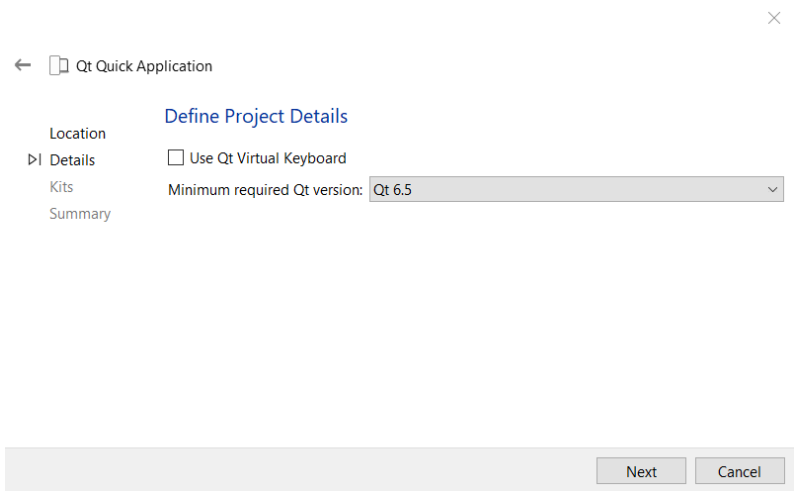
Browse...

☐ Use as default project location

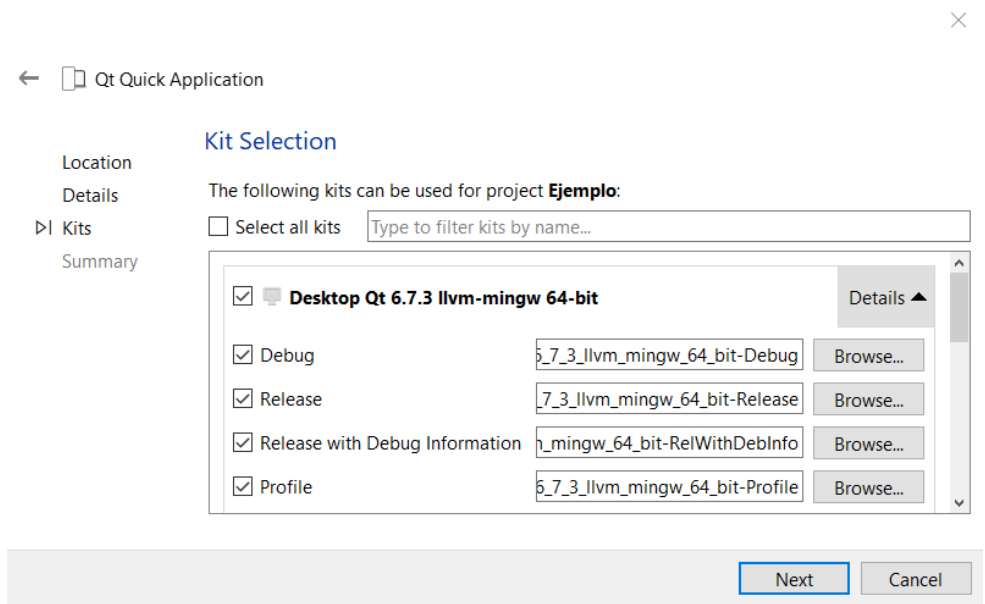
Next

Cancel

Seleccionamos la versión mínima de Qt requerida. Lo dejaremos como Qt 6.5



En el apartado de Kits, lo dejaremos por defecto.

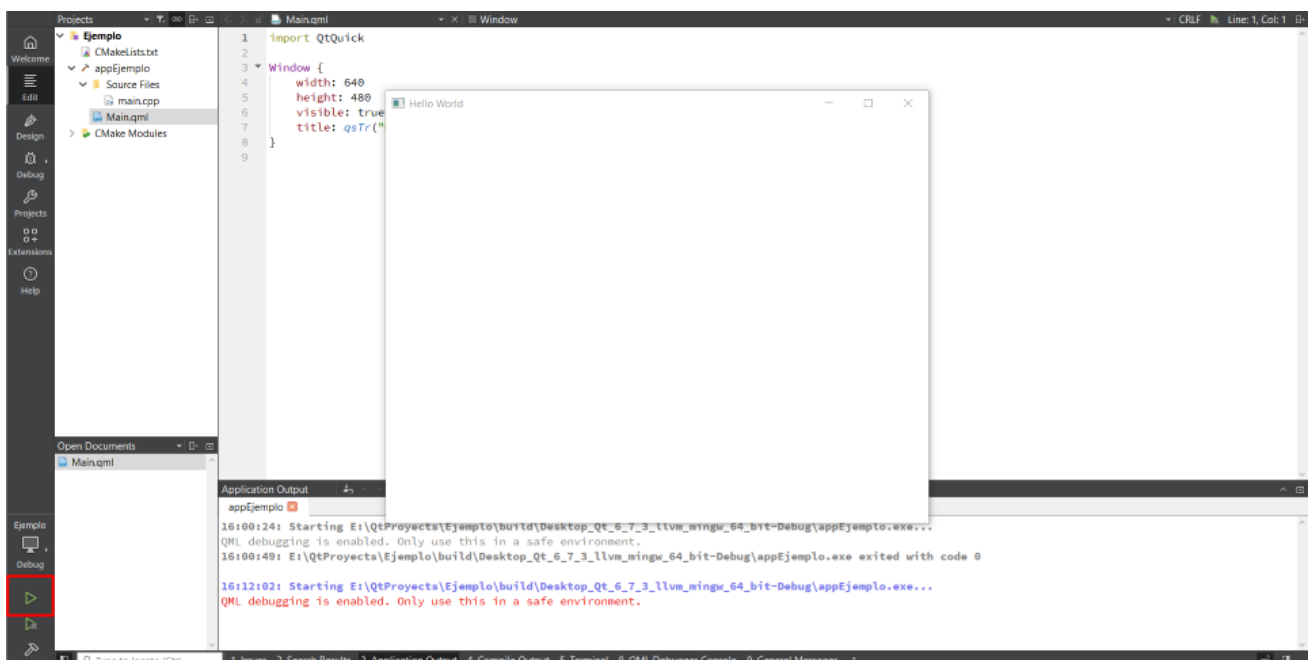


Una vez hecho el proceso de creación del proyecto, nos llevará directamente a su vista.

Vemos que nos encontramos en el apartado **Edit**. Aquí, podremos ver la estructura del proyecto y además se nos abre el editor de texto, donde podremos modificar el código (como en cualquier IDE).

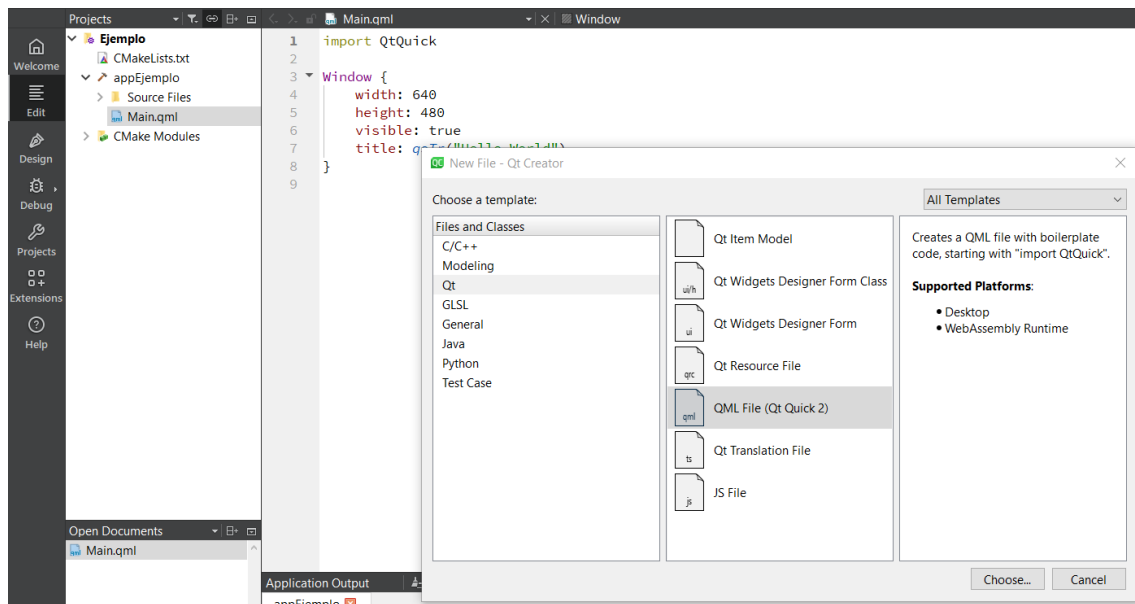
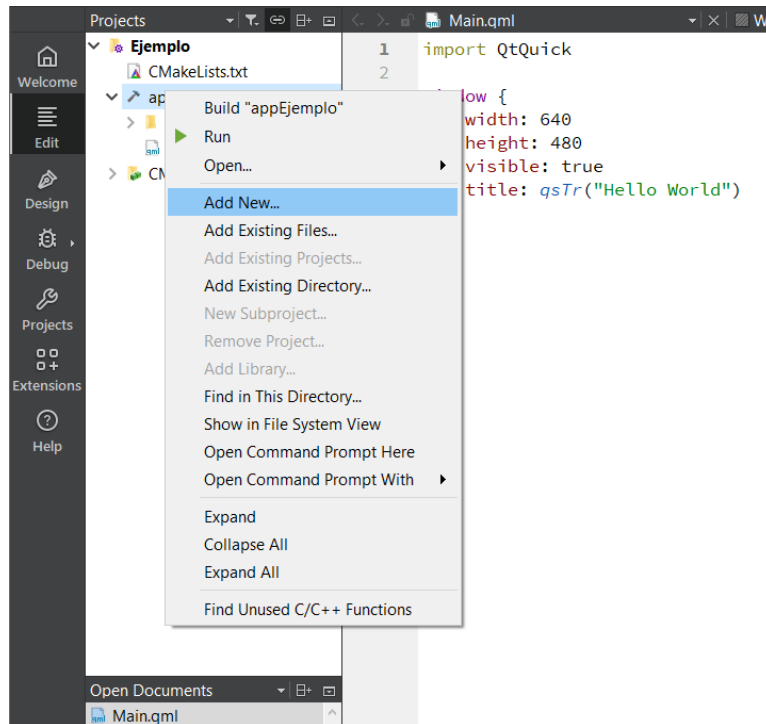


El proyecto por defecto incorpora el archivo **Main.qml**, con un componente **Window**. Este componente es básico en Qt y es para definir la ventana de la aplicación. En la parte inferior izquierda, pulsamos en **Run** para ejecutarla.



Primeros pasos con Qt Quick

Para crear nuevos ficheros .qml, donde crearemos nuestros componentes, haremos click derecho en la aplicación (appEjemplo en el caso de esta guía) y **Add New**. Crearemos MiRectangulo.qml.

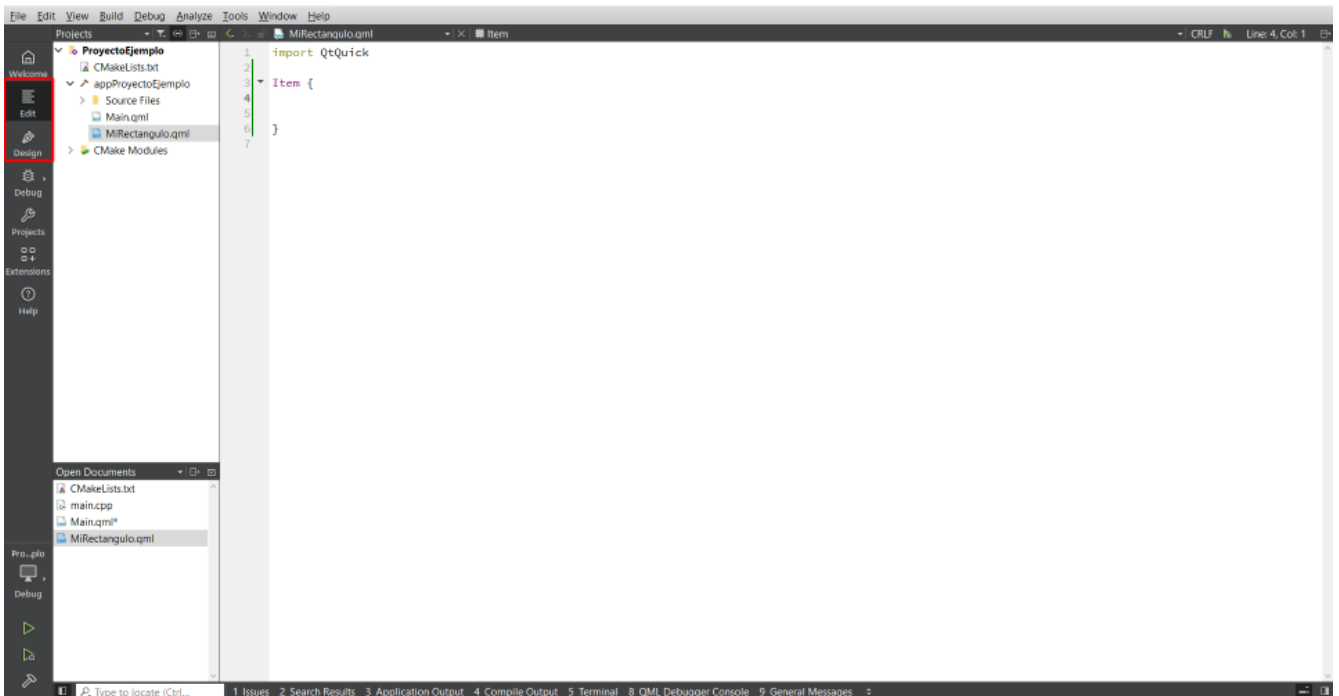


Nos aparecerá un nuevo fichero .qml donde habrá un código vacío con **Item {}**, que será un contenedor para nuestro componente.

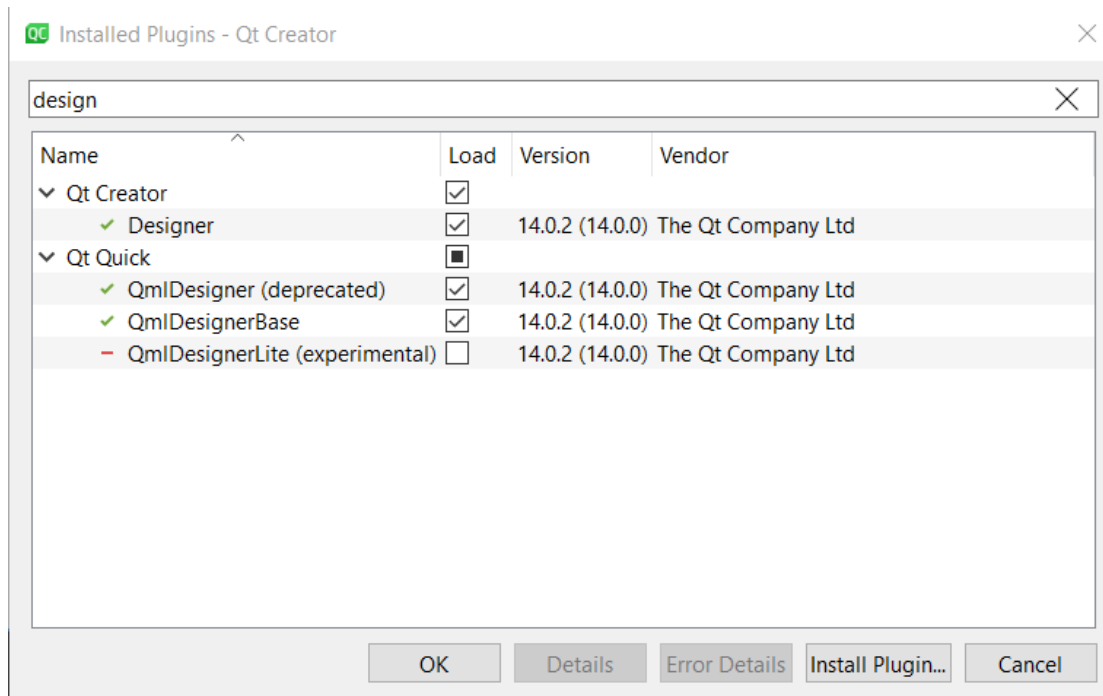
Creación de componentes

En Qt Creator tenemos dos vistas esenciales Edit y Design. Como se menciona anteriormente, con Edit podemos ver nuestro código. Con Design, se nos abre una vista donde podemos arrastrar y soltar componentes ya ofrecidos por QtQuick. Debemos importar en el .qml el siguiente módulo

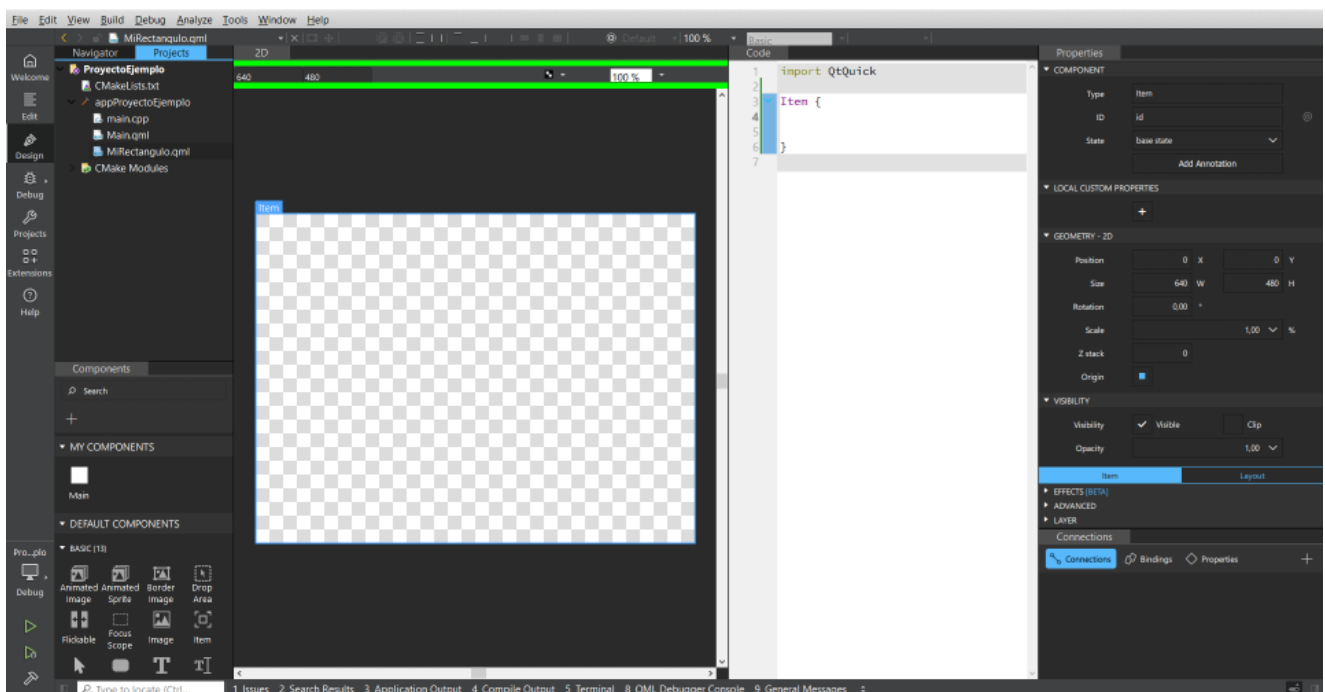
import QtQuick.Controls



Si la vista Design aparece desactivada al clickar sobre un archivo .qml, deberemos activarla en la barra superior, **Help > About Plugins**, buscamos QmlDesigner y lo activamos (aparecerá con un -)

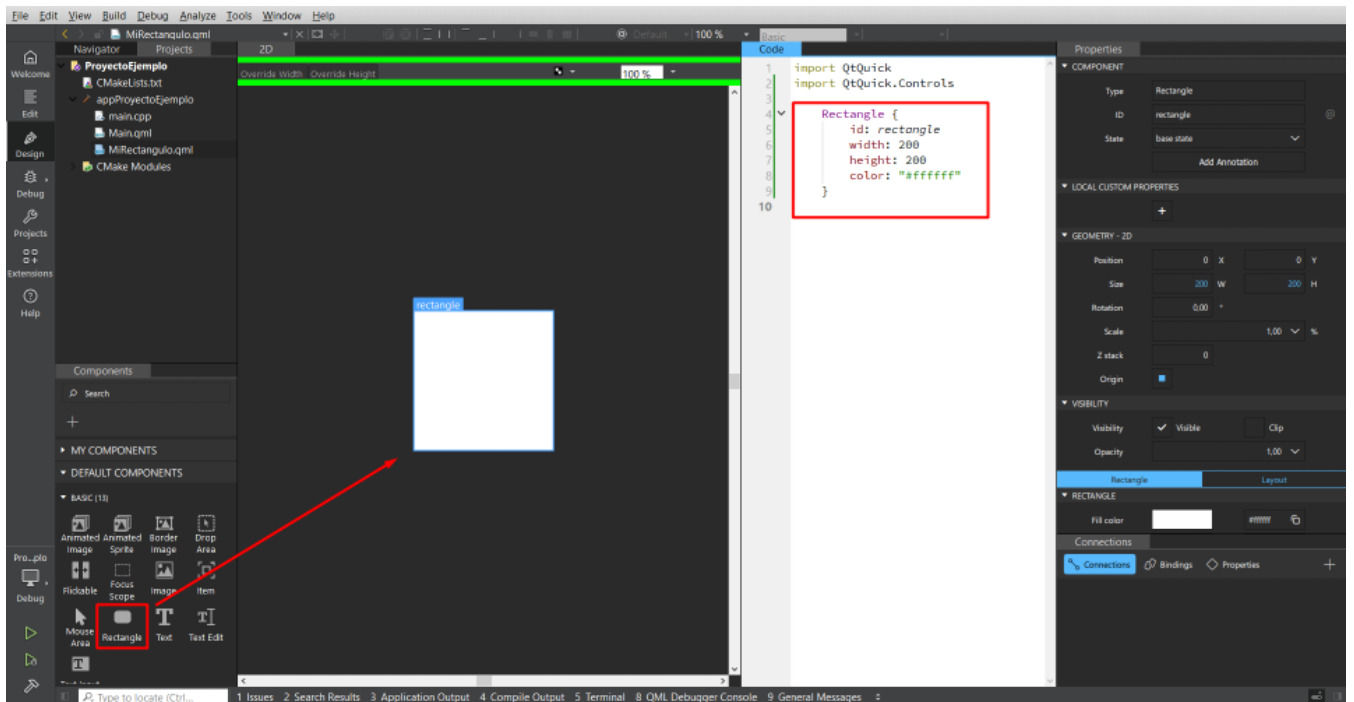


Una vez que lo activemos podremos acceder a la vista Design del componente creado.



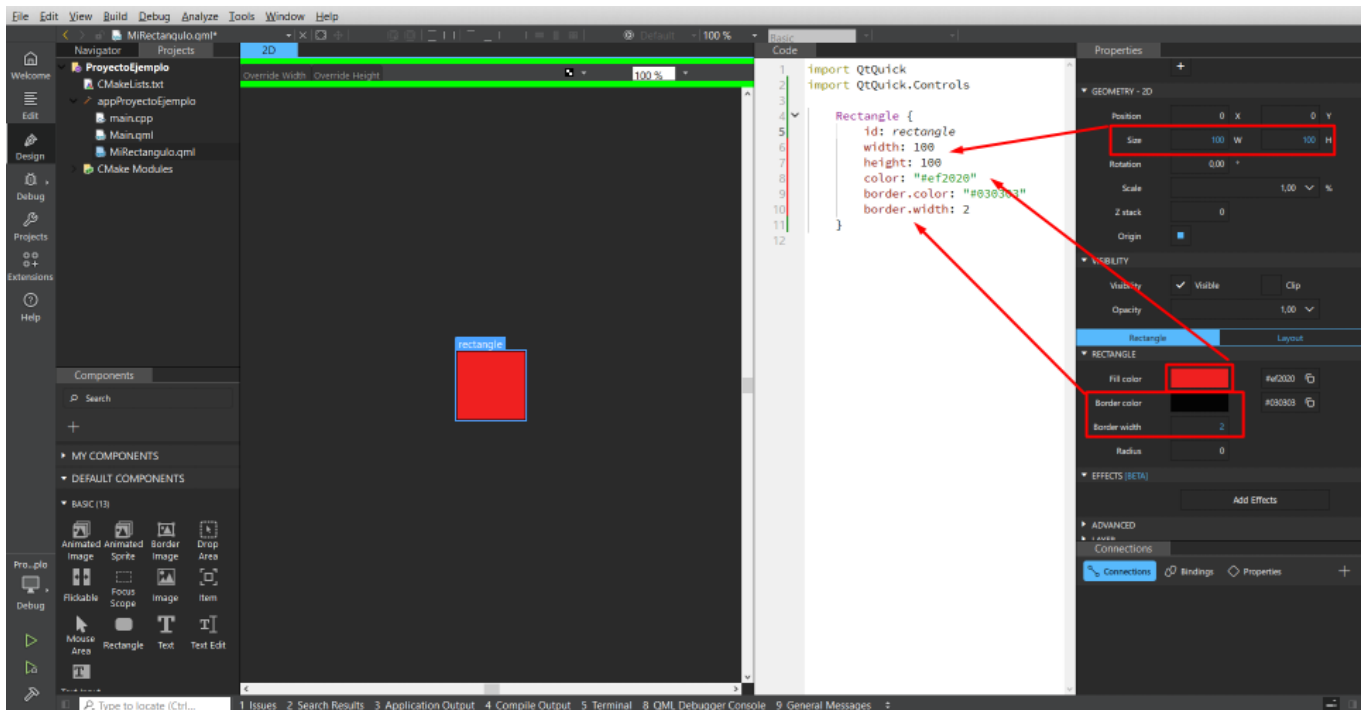
El componente **Item** no es más que un contenedor en el cual podemos añadir varios componentes con tal de que estén contenidos en un mismo padre.

En este caso, vamos a eliminar **Item**, ya que nuestro componente será únicamente un Rectángulo que arrastraremos de la parte inferior izquierda en la vista de **Components**.

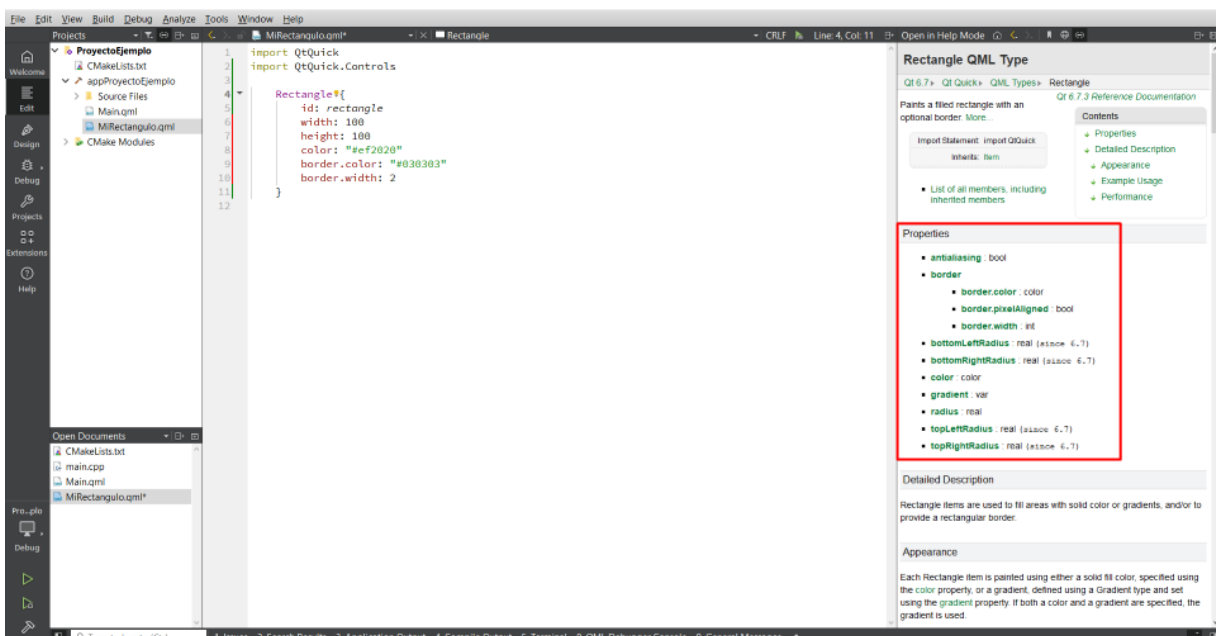


Propiedades

El componente rectángulo tiene unas propiedades ya definidas (que podemos modificar) que nos aparecen a la derecha de la vista. Entre ellas, su color, tamaño, borde... Una vez modificadas desde la vista de **Properties** se reflejarán en el código del rectángulo.

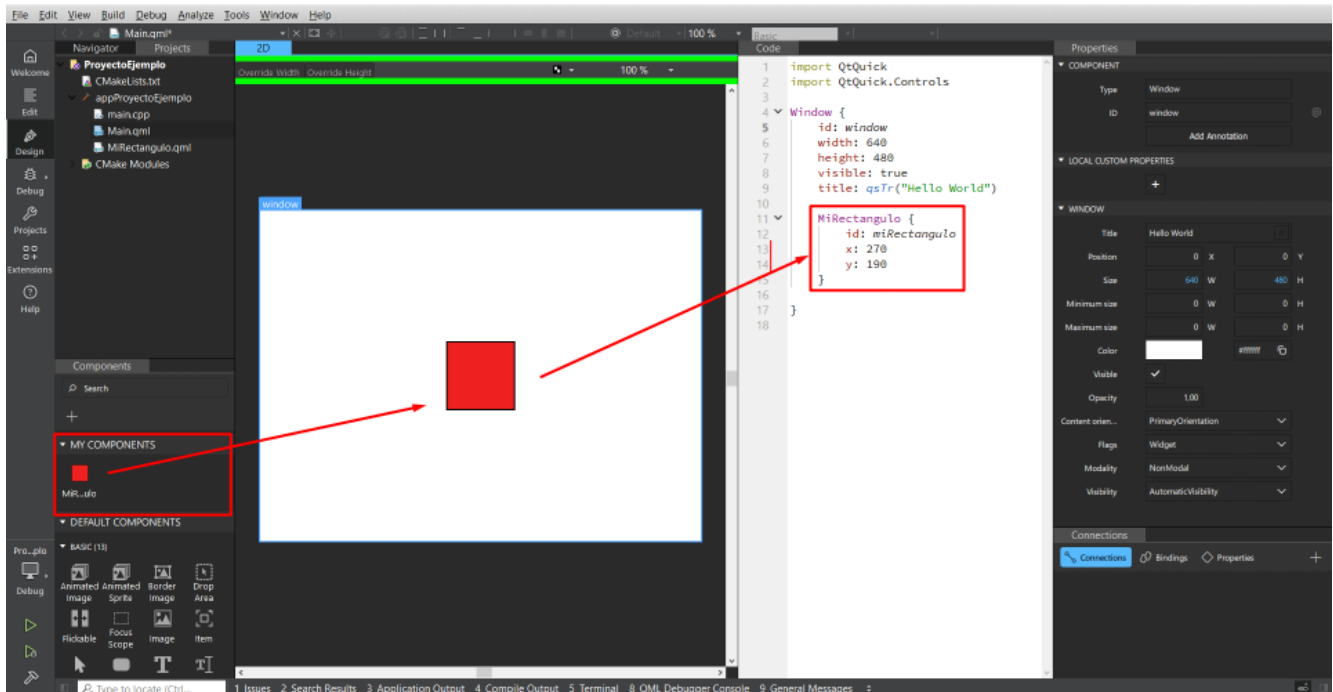


Las propiedades de los componentes podemos modificarlas como queramos y cambiarán su aspecto. Para poder ver todas las propiedades iremos a la vista de **Edit** (el código), clickaremos el componente Rectangle y pulsaremos F1.



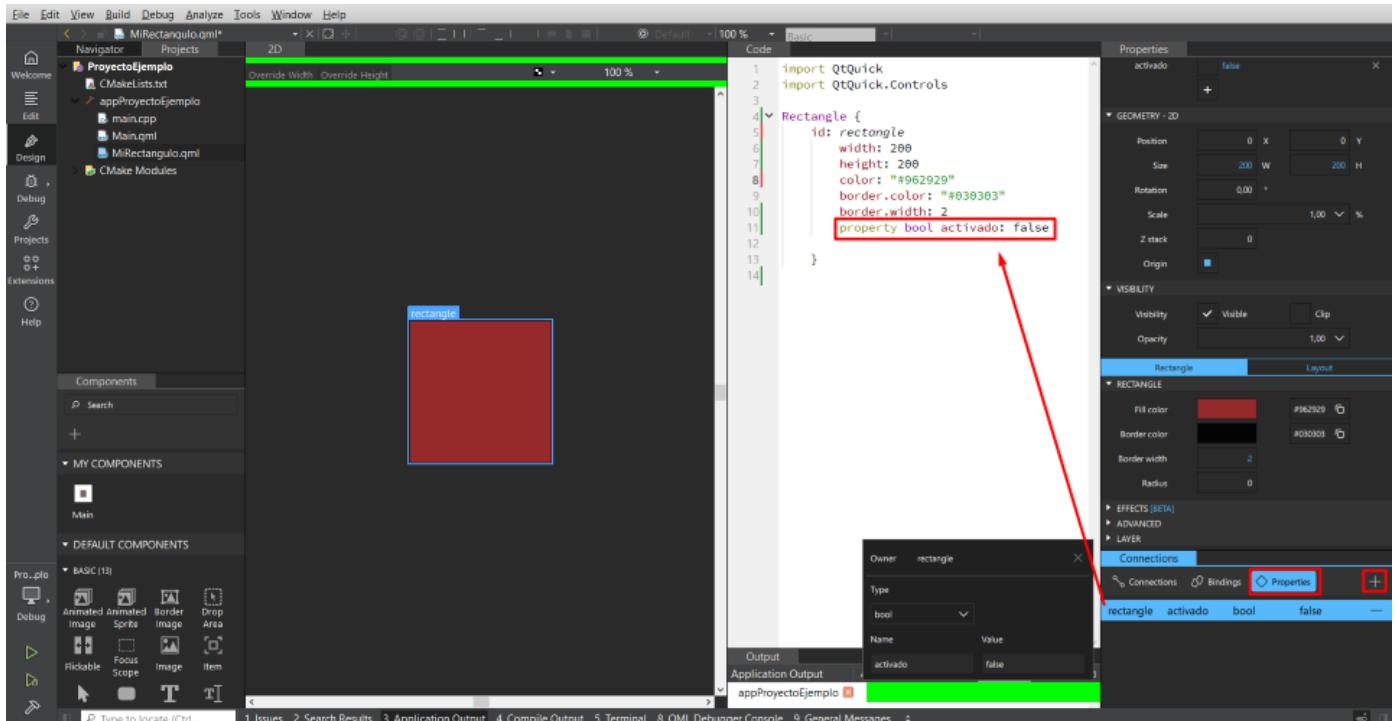
Uso de componentes

En este momento, tenemos una pantalla principal "Main.qml" y un componente "MiRectangulo.qml". Desde la vista Design, podemos arrastrar nuestros propios componentes (igual que hemos hecho con el rectángulo). Vamos a añadir MiRectangulo a Main.



Crear propiedades para un componente

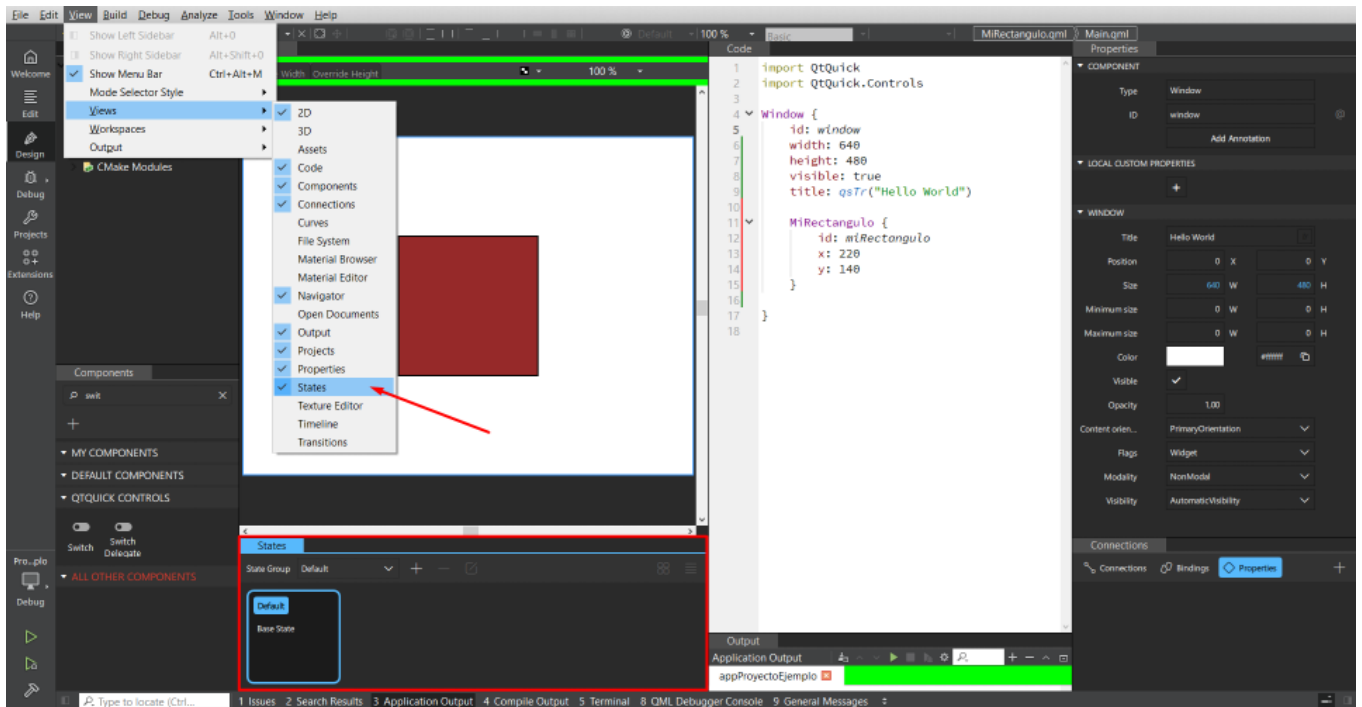
Además de las propiedades que nos ofrecen los componentes de Qt Quick, podemos añadir más con tal de añadir complejidad o funciones nuevas. En este caso, hemos añadido la propiedad **activado** que por defecto aparece como false.



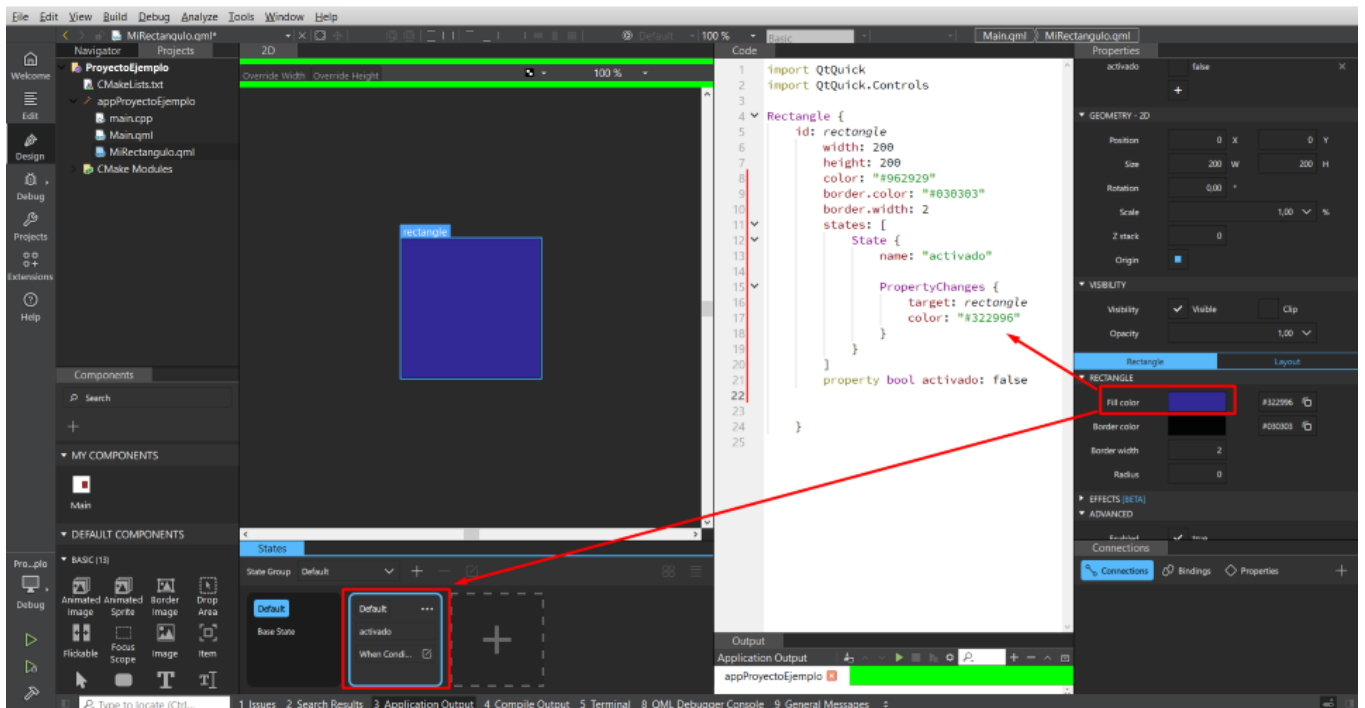
Crear estados para un componente

Una vez creada la propiedad “activado” para el componente, crearemos un estado, es decir, como se visualiza un componente en un momento determinado.

Primero, tendremos que activar la vista yendo a la barra superior **View** > **Views** > **States**

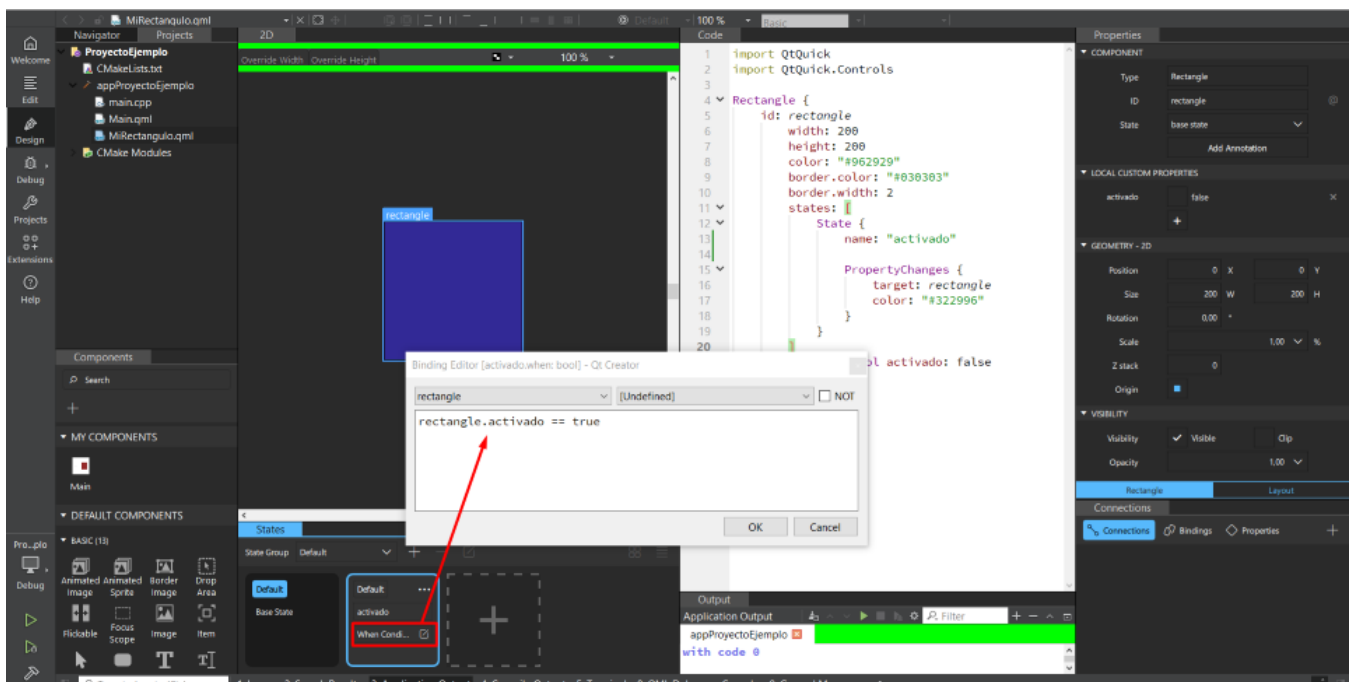


En esta vista podremos crear y gestionar todos los estados en los que se puede encontrar un componente. Dándole a '+', creamos un nuevo estado.



Teniendo seleccionado el estado nuevo que hemos creado, podemos modificarlo como queramos, en este caso cambiamos el color a azul.

Como podemos ver en el código, se ha creado una lista de estados con un solo estado identificado por name: "activo" y que hace referencia al componente *rectangle* y que lo que cambia es únicamente el color. Añadimos una condición para activar ese estado.

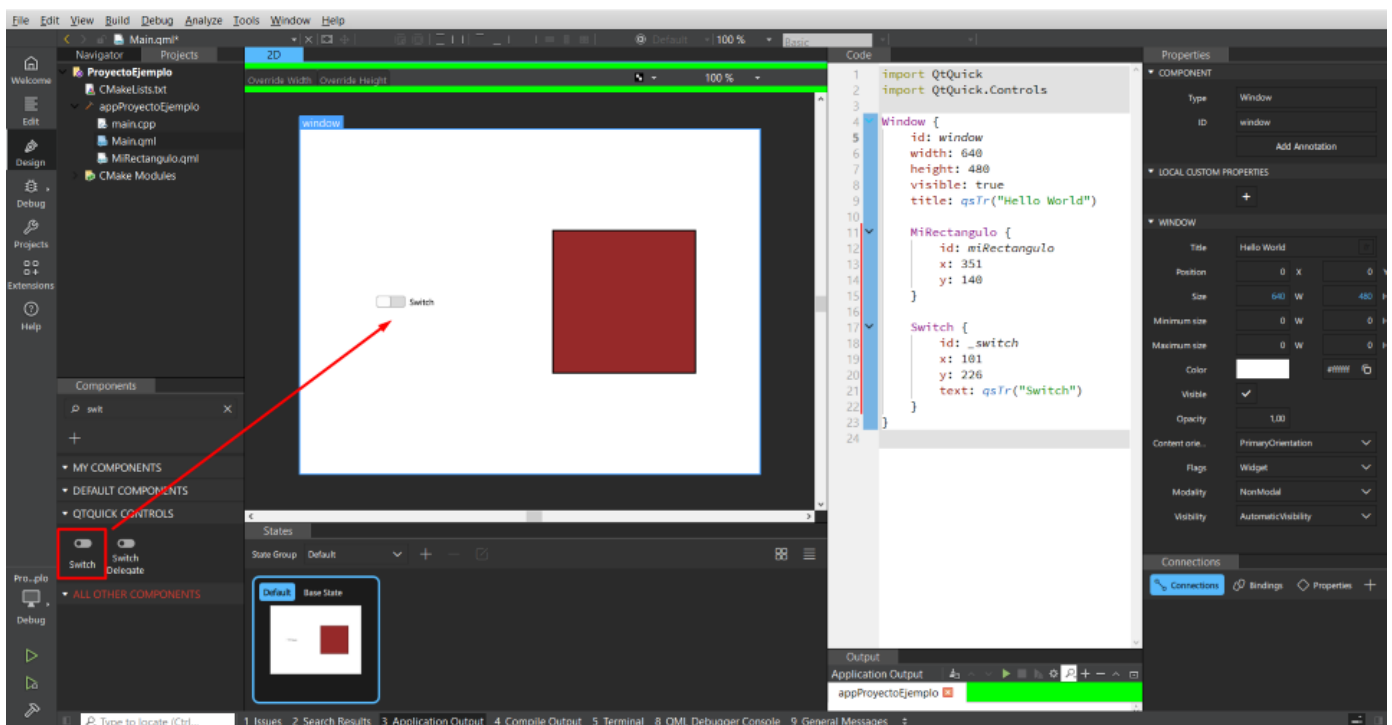


Enlaces

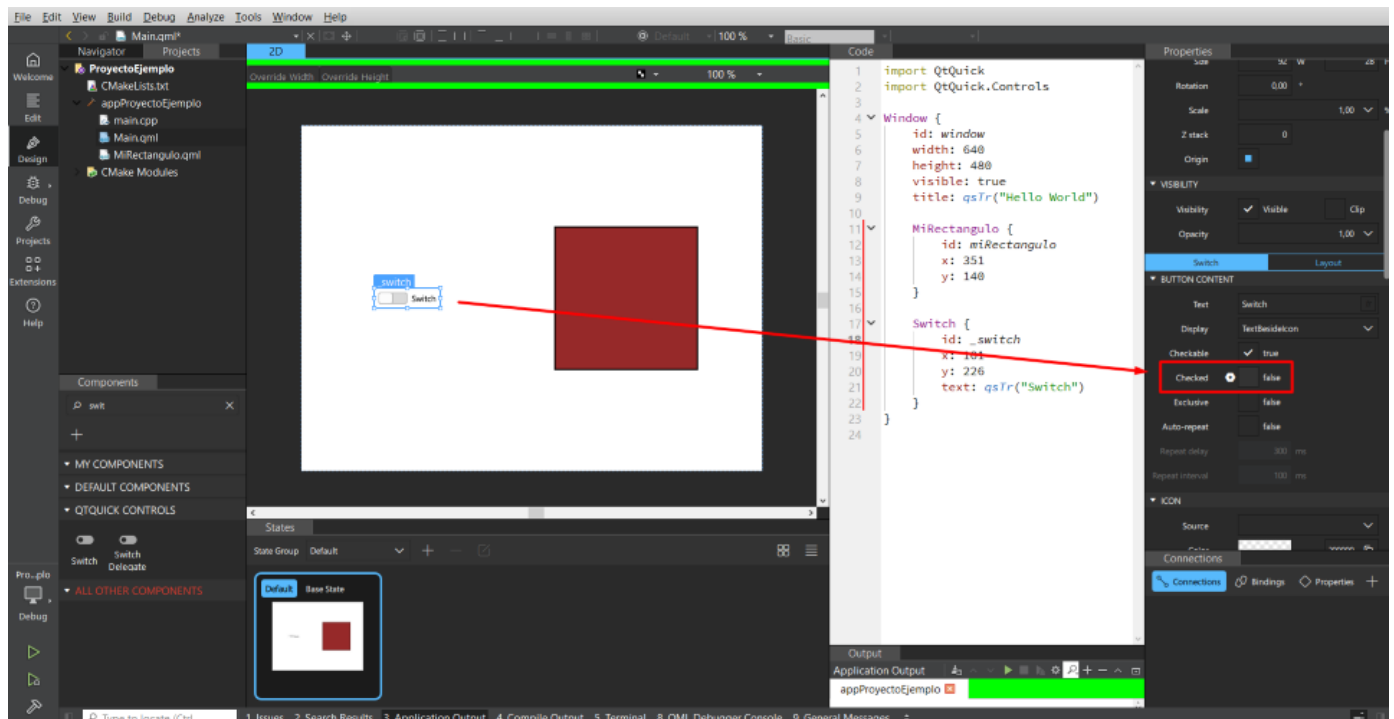
Los enlaces (o bindings) se utilizan para crear una relación entre las propiedades de dos componentes. Las propiedades deben ser del mismo tipo y el cambio de una supondrá el cambio de la otra.

Primero, añadimos un componente de tipo Switch a nuestra vista. El Switch tiene una propiedad de tipo boolean llamada *Checked* que controla si está activado o no.

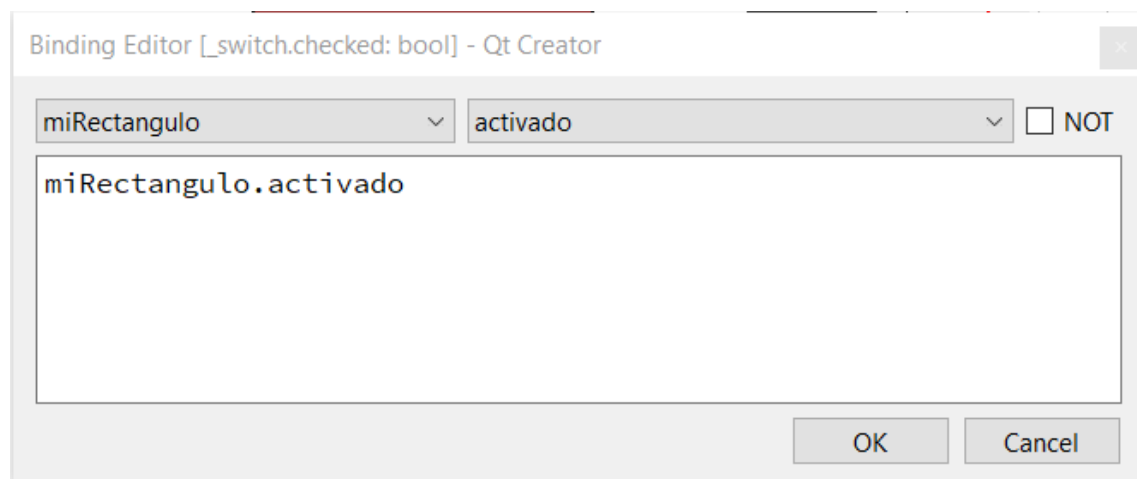
Mediante Bindings, vamos a hacer que esa propiedad *Checked* se enlace con nuestra propiedad creada anteriormente *activado*, en nuestro componente rectángulo.



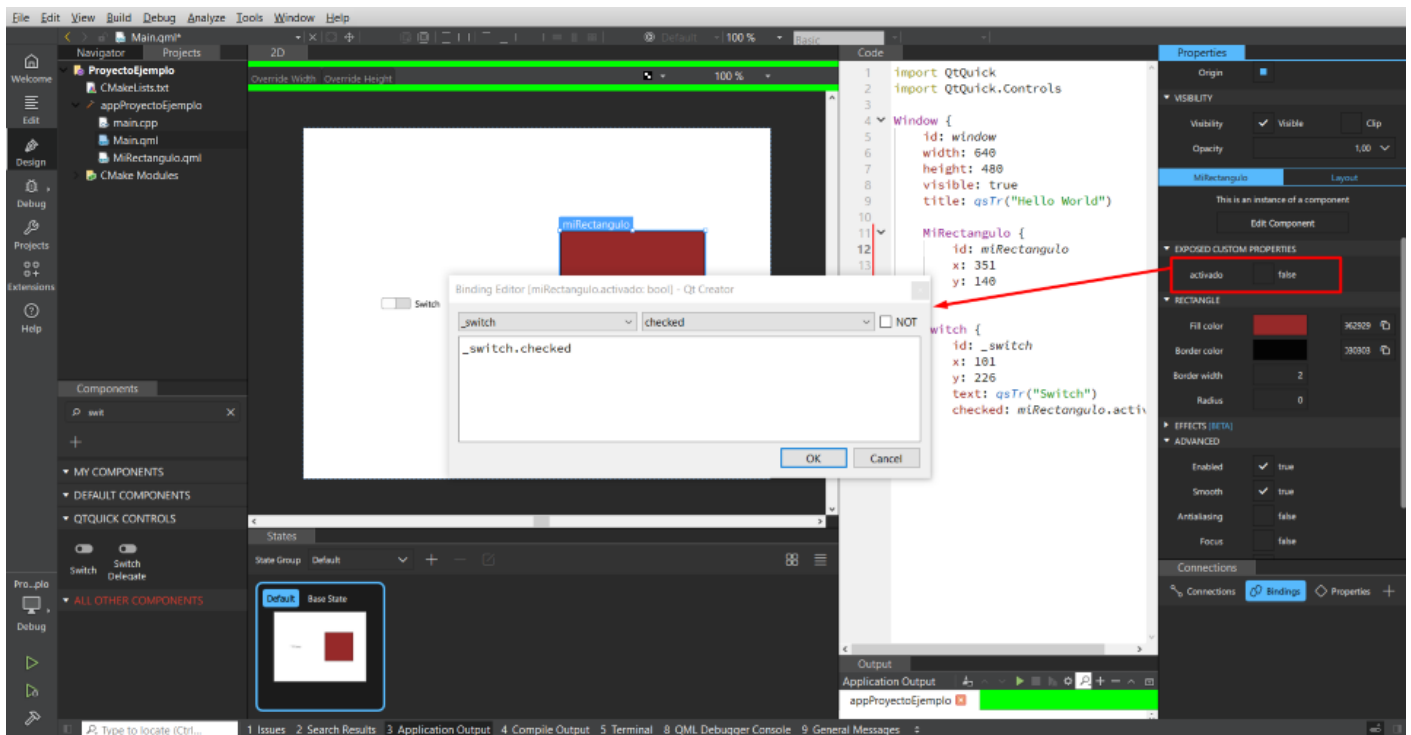
Al lado de algunas propiedades nos aparecerá una “ruedecita” con la que podremos crear el enlace con la propiedad de otro componente.



Clickamos en Set Binding y seleccionamos la propiedad de nuestro rectángulo.



De igual manera, seleccionamos el rectángulo y enlazamos la propiedad *activado* con *Checked* del Switch.

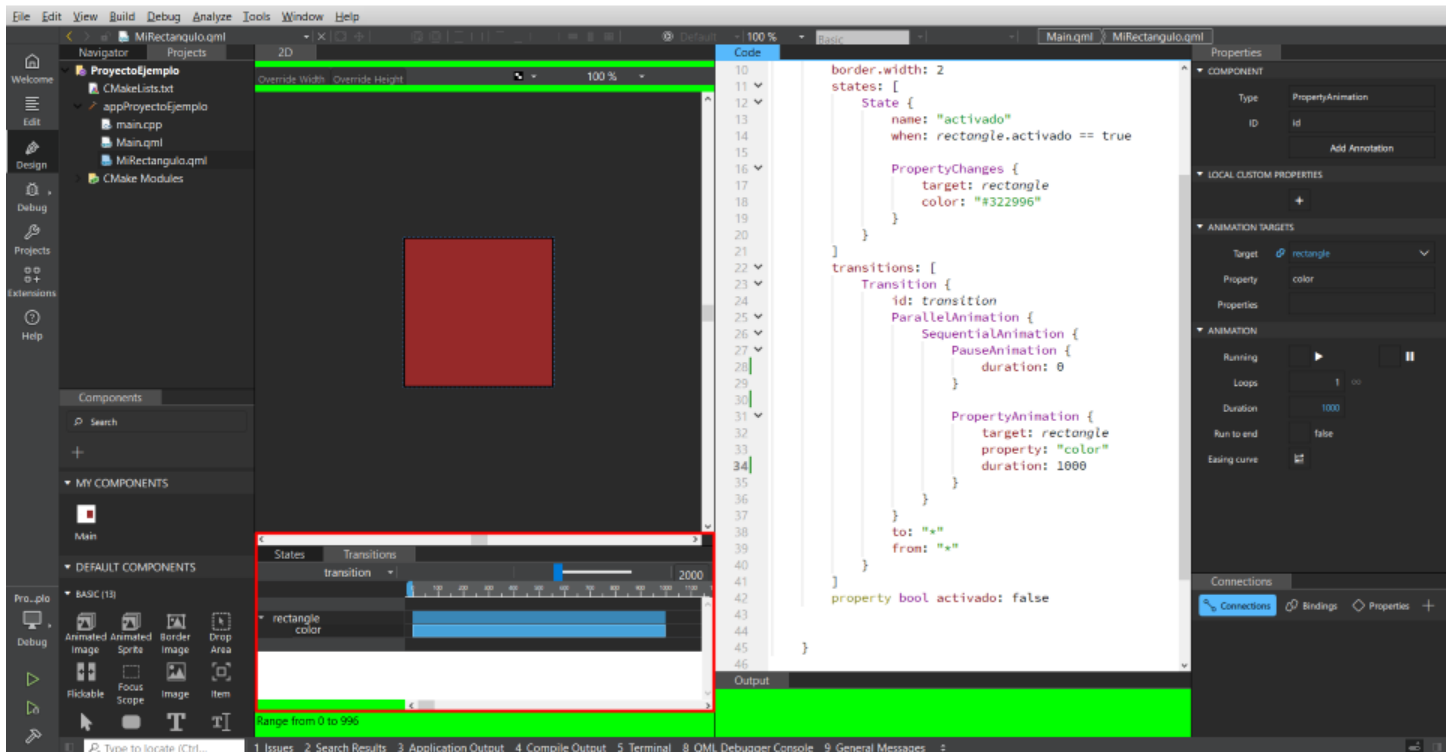


Este enlace hará que el cambio en el Switch suponga un cambio en la propiedad *activado* del rectángulo y a su vez, cambie de estado.

Transiciones

Las propiedades se pueden modificar de forma que su cambio en la vista no sea inmediata, por ejemplo cambiando el color de un componente en un lapso de un segundos.

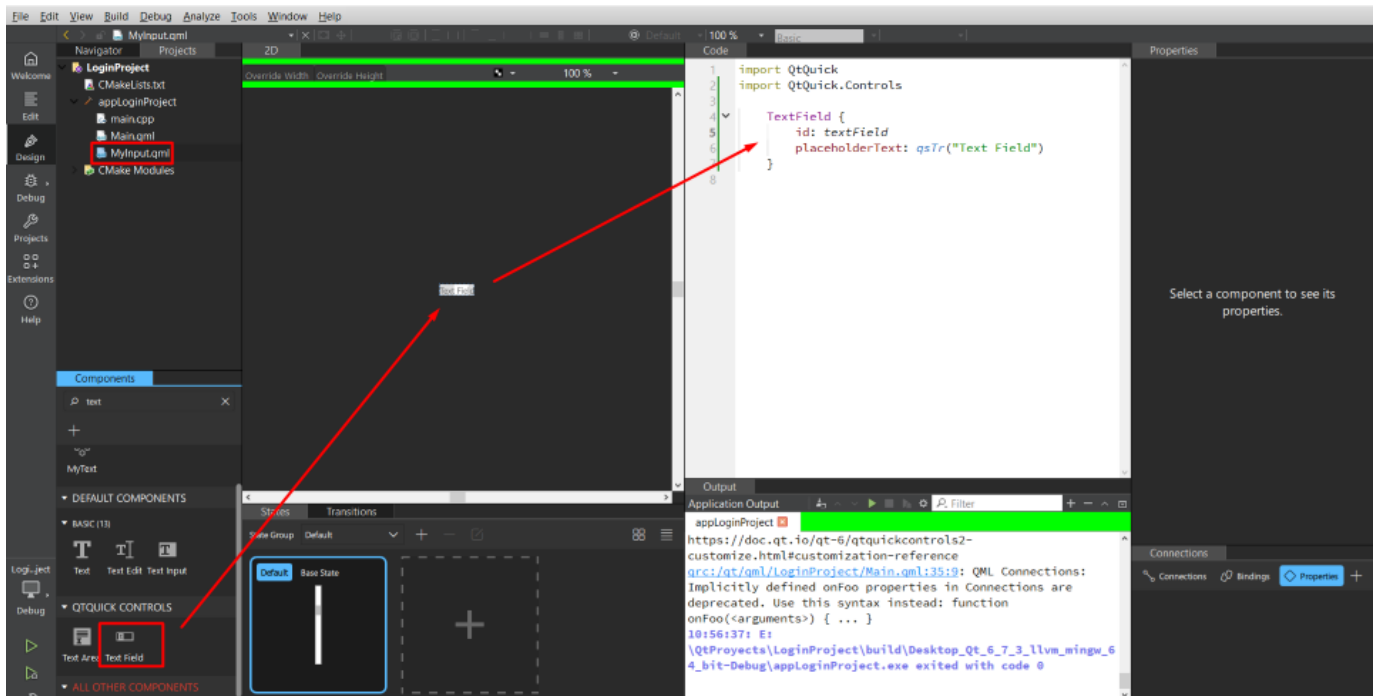
Para ver la vista de transiciones, iremos a la barra superior **View > Views > Transitions**



Text Field y Button

Vamos a crear una vista de un *Login* utilizando los componentes Text Field y Button. El primero será un campo de texto donde el usuario podrá escribir su usuario/contraseña y el segundo será un botón que al clickarlo compruebe si el usuario existe o no.

Primero, creamos los inputs arrastrando el componente Text Field al nuevo fichero .qml que hayamos creado. En este caso, MyInput.qml.

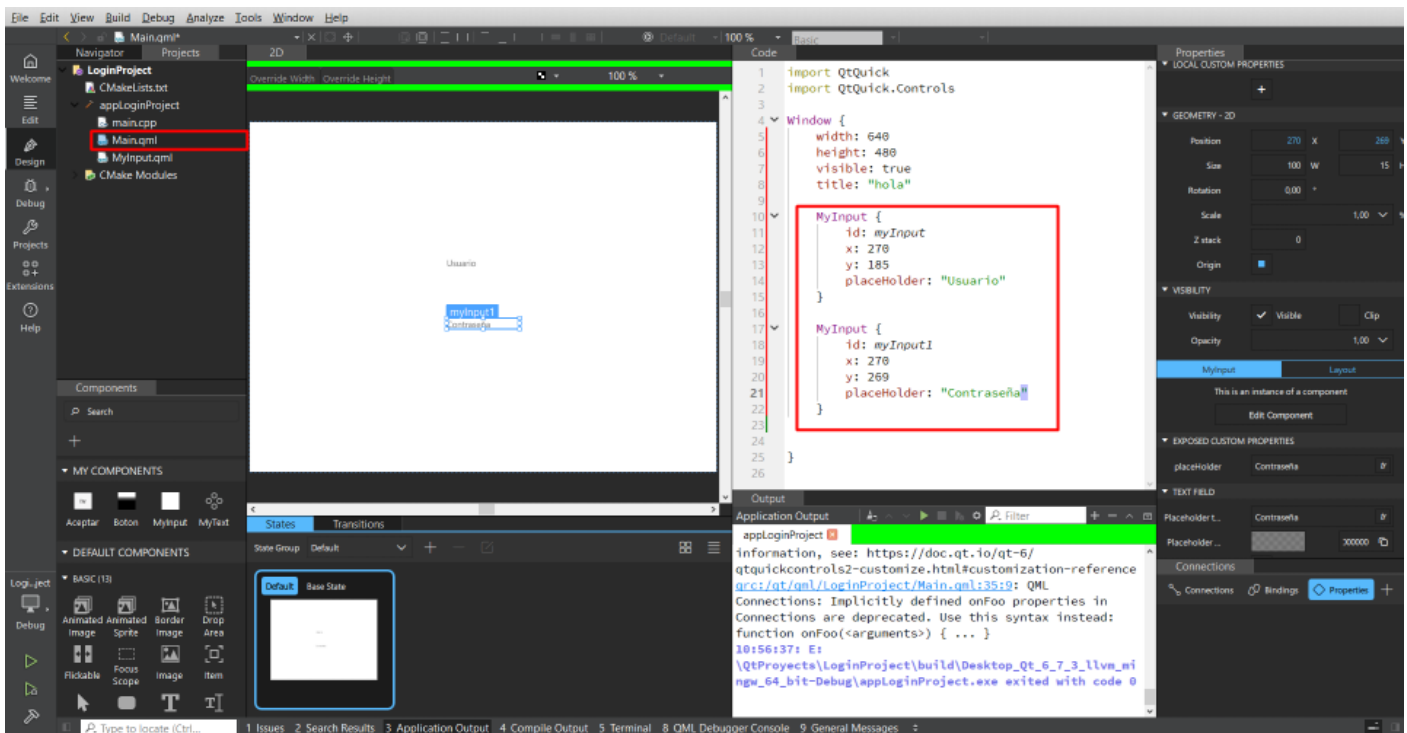


El placeholder de nuestro input queremos que sea dinámico, es decir, cuando estemos creando el componente en la vista decidamos que texto es el que mostrará. Para eso, crearemos una propiedad de tipo string y se la asignaremos a la propiedad (ya creada), placeholderText.

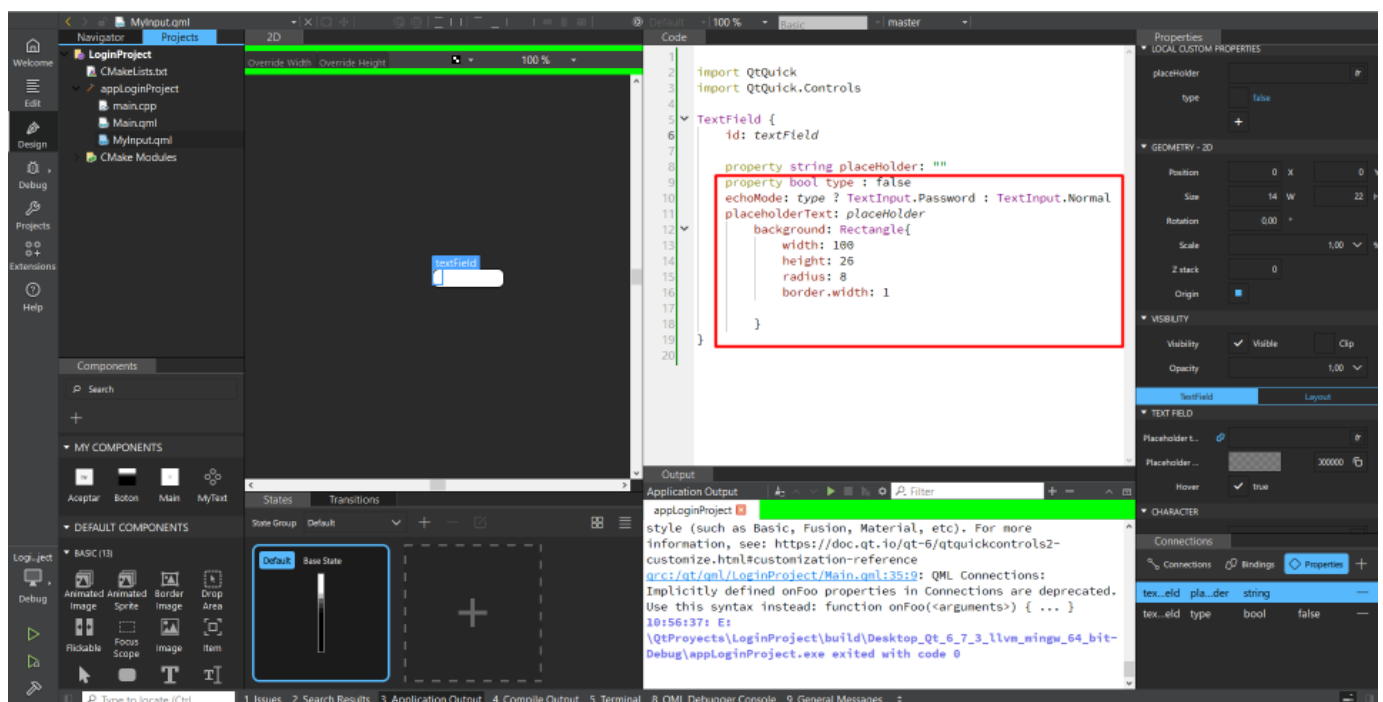
```
import QtQuick
import QtQuick.Controls

TextField {
    id: textField
    width: 100
    property string placeHolder: ""
    placeholderText: placeHolder
}
```

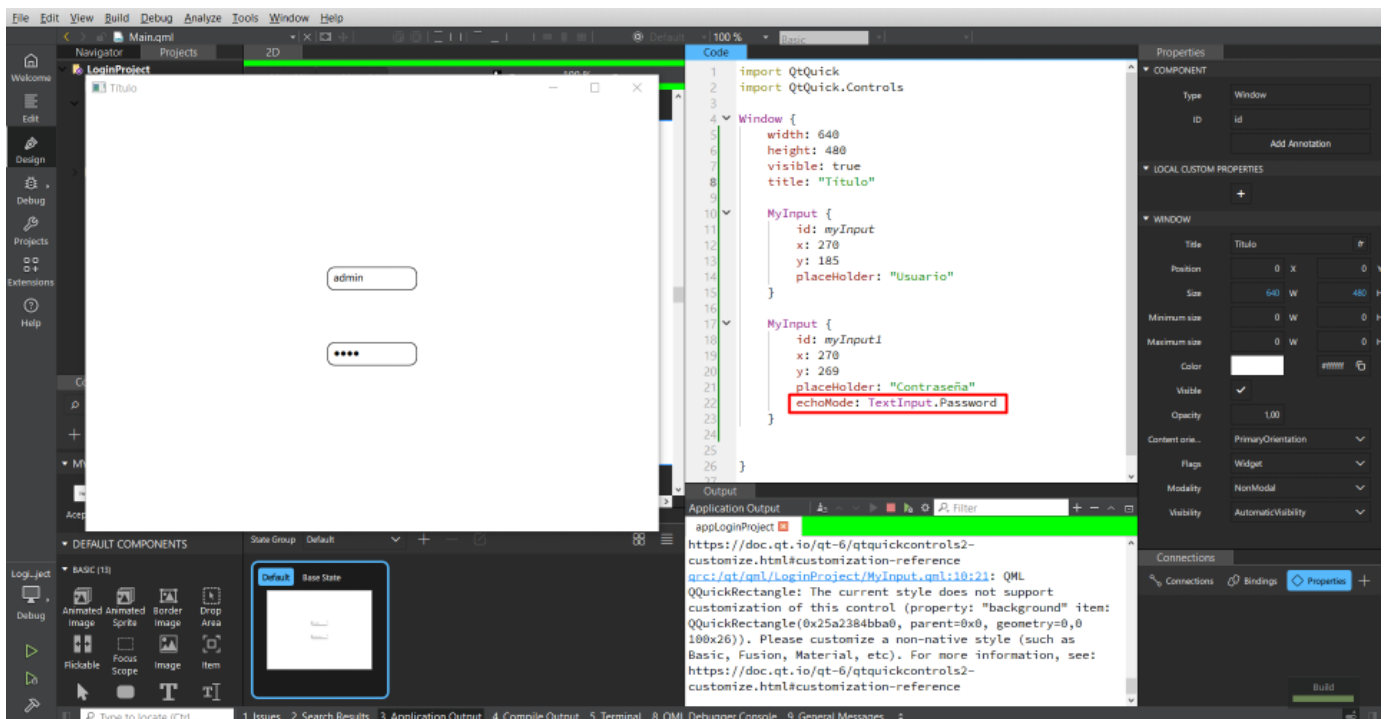
Ahora, en la vista Main.qml añadimos los dos inputs y le ponemos como propiedad "Usuario" y "Contraseña" respectivamente.



Si probamos, nos saldrán dos inputs donde nos dejará escribir texto. Pero queremos añadir dos cosas más, algo de estilo para nuestro input y que, el segundo input, al ser una contraseña, no se muestre el texto.



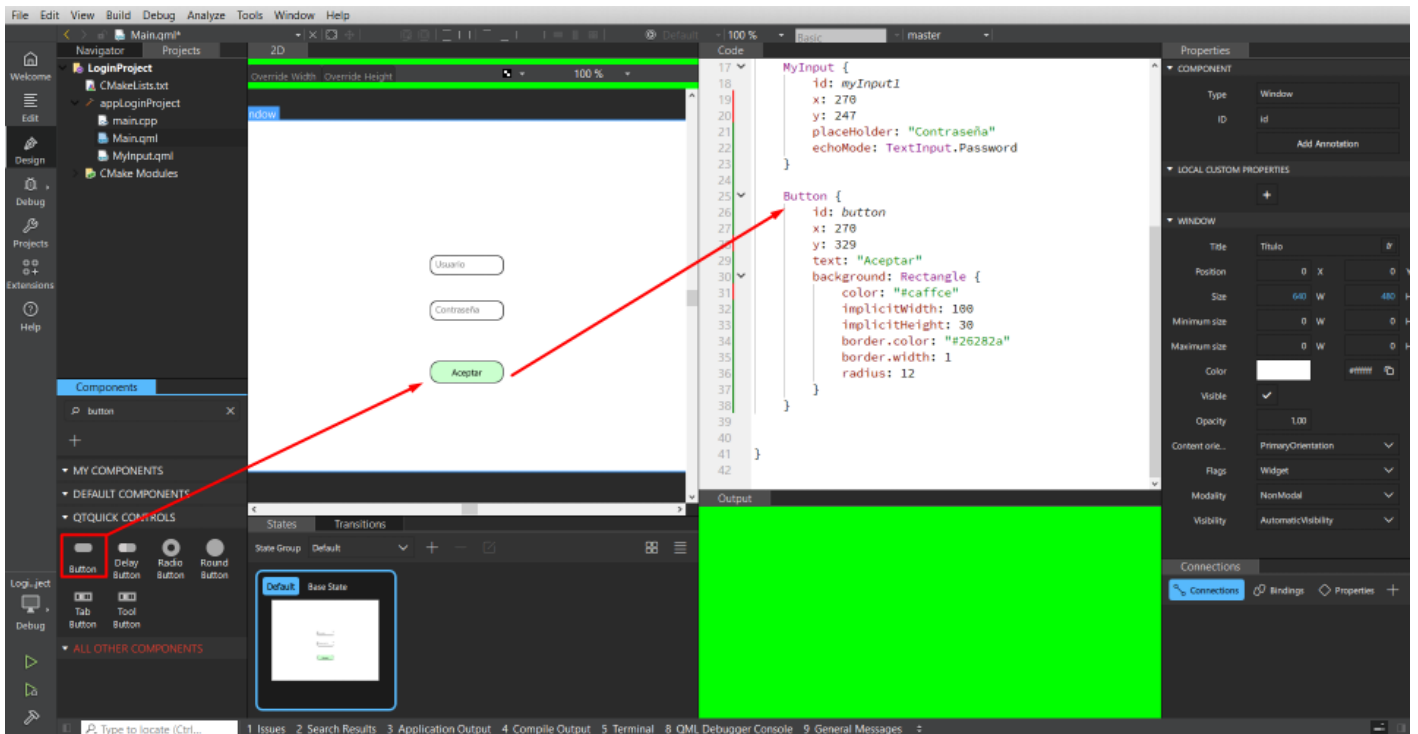
- Background.
 - Propiedad para definir el fondo del input, que será su estilo. Para añadir bordes, colores... Utilizamos el componente Rectangle.
- echoMode.
 - Propiedad para definir si el input es de tipo Password o no.



Button

Siguiendo con el ejemplo anterior, vamos a añadir un botón a nuestra vista *Login*, con tal de comprobar que el usuario que intenta iniciar sesión existe.

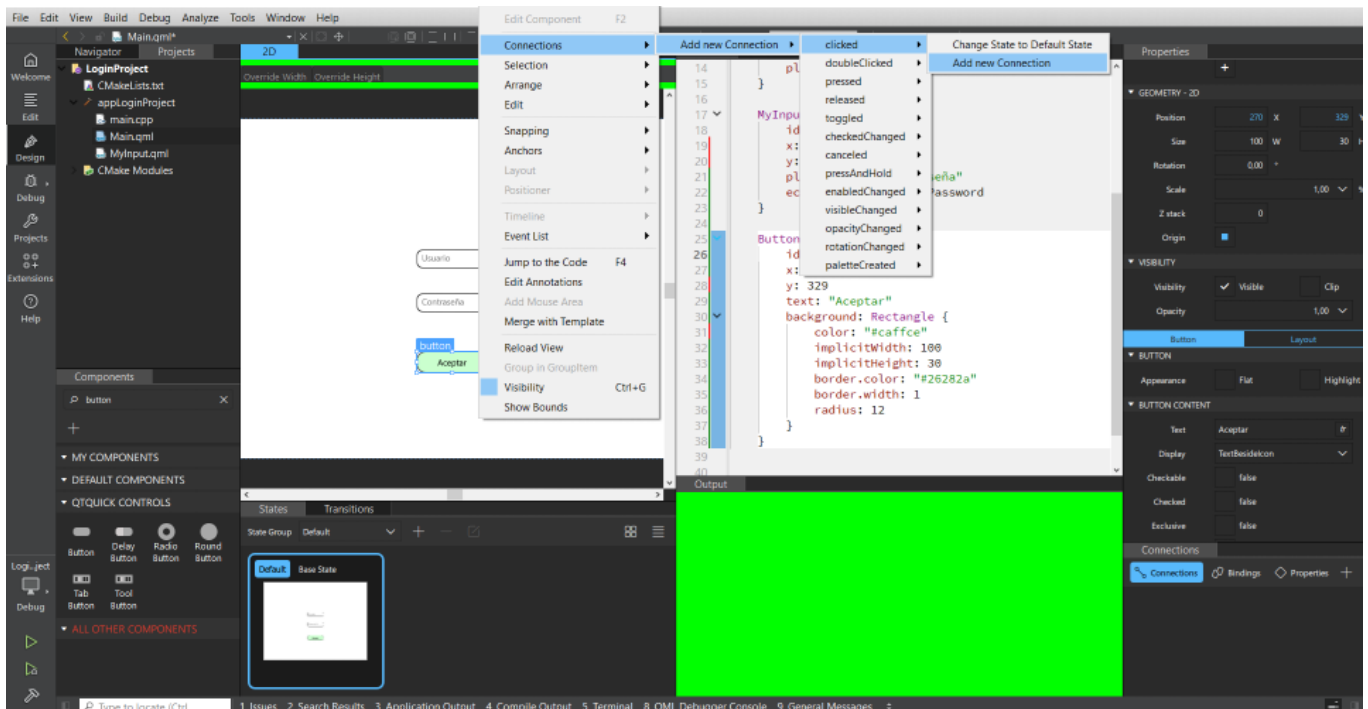
Añadimos el componente Button desde la vista Design y le hacemos algunos retoques de estilo mediante la propiedad *background*.



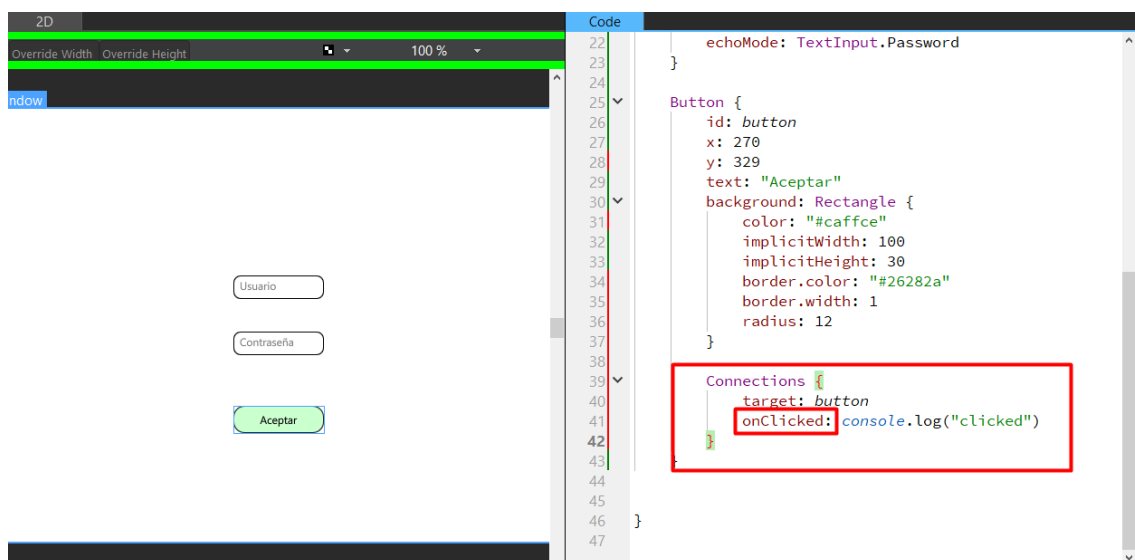
Connections

Las connections en Qt Quick son escuchadores que se pueden enlazar a componentes y según que señal (eventos) reciban estos escuchadores, podremos reaccionar.

Haciendo click derecho en el componente Buton, vamos a **Connections > Add new Connection > Clicked > Add new Connection**

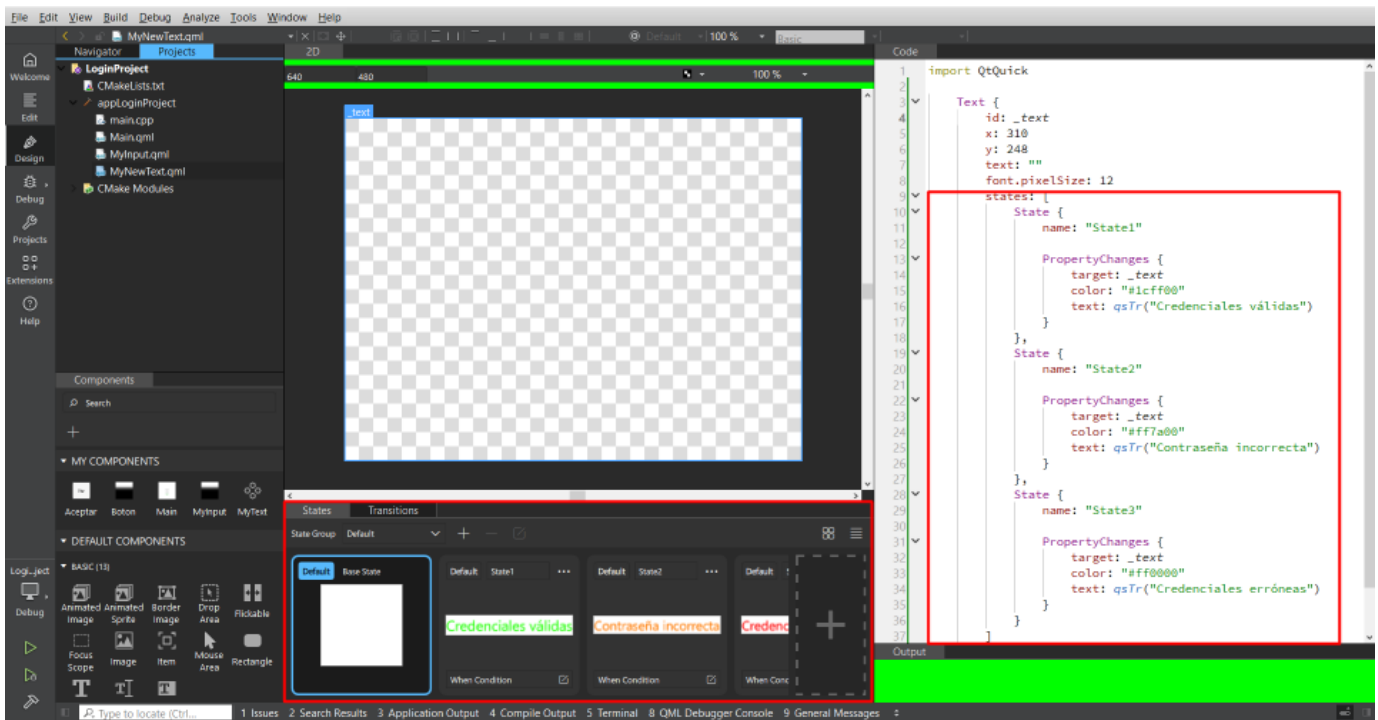


Aparecerá dentro del botón la nueva conexión creada.



Creamos un componente de tipo texto y le creamos tres estados.

- Credenciales válidas
- Credenciales erróneas
- Contraseña incorrecta



Finalmente, en la vista de main.qml, añadiremos nuestro componente de Texto y en la Señal de onClick comprobaremos si existe o no el usuario. Accederemos al estado del texto mediante su id myNewText.state y le asignaremos el que corresponda.

