

## Introducción a QML - I

En este tutorial introduciremos los conceptos básicos de desarrollo mediante **QML** y el framework **Qt Quick**.

### ¿Qué es QML?

QML es un Lenguaje declarativo de Meta-Objetos basado en JavaScript que permite diseñar interfaces de usuario. Es parte del Framework Qt de Nokia.

Se basa una idea básica; declarar elementos **QML** en un árbol de objetos que permiten visualizar bloques, textos, gráficos, imágenes y poder interactuar con ellos mediante estados, animaciones, transiciones... Conceptualmente es muy similar a **XAML**. Los elementos **QML**, además, pueden ser utilizados mediante el estándar JavaScript en la misma declaración QML o mediante la inclusión de ficheros JavaScript .js. Los elementos QML pueden también ser integrados en software desarrollado con el framework Qt en C++ .

### ¿Qué es Qt Quick?

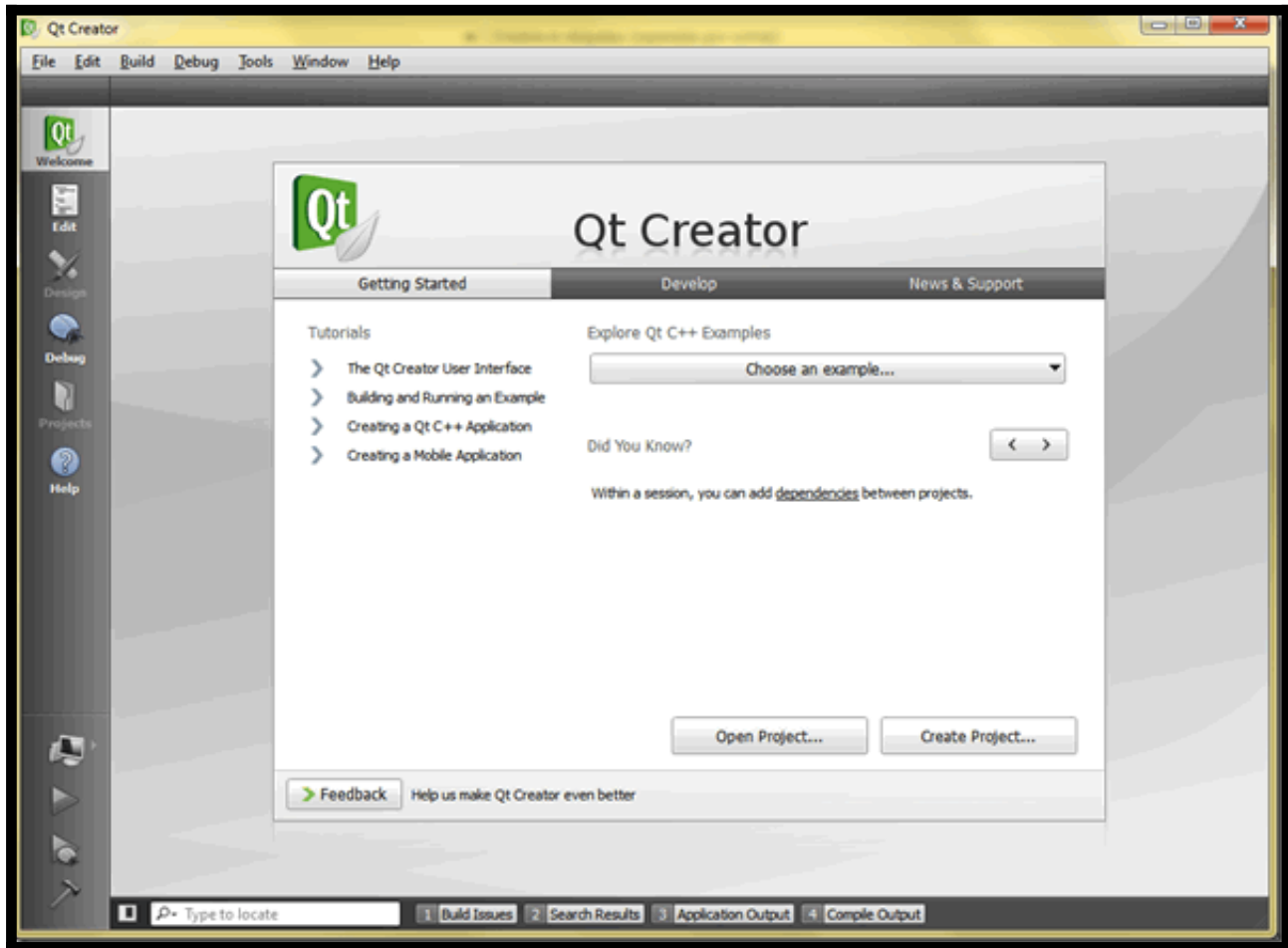
Es un conjunto de tecnologías asociadas a Qt

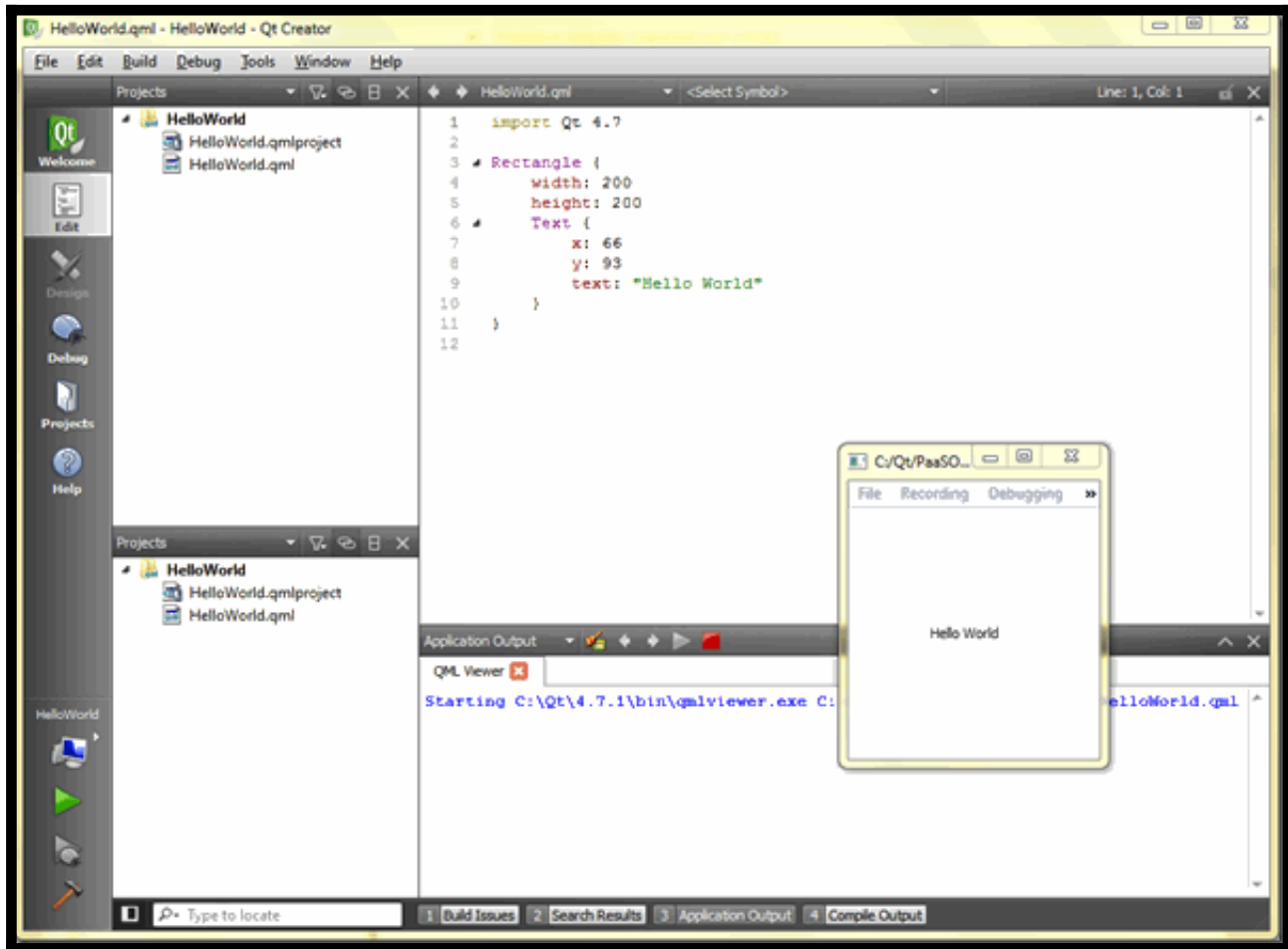
- **QML** – Un lenguaje para desarrollo y diseño de interface de usuario
- **Qt declarative** – Módulo de intergración de QML y las librerías Qt C++
- **Qt Creator tools** – Entorno de desarrollo

### Empecemos con el Hola mundo

Si no hemos nunca instalado **Qt**, descargamos e instalamos desde <http://www.qt.io/download-open-source>

Mediante Qt Creator creamos un proyecto de tipo QML Application.



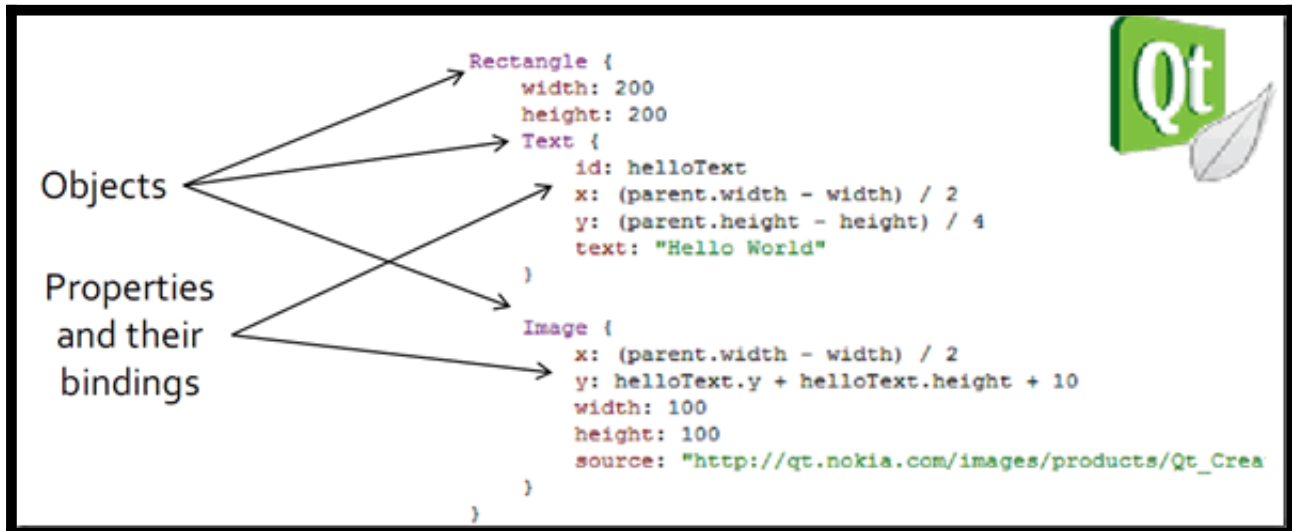


Compilamos y ejecutamos y ya tenemos un HolaMundo.

Este primer ejemplo nos muestra como es la declaración básica de un objeto **Rectangle** que contiene a su vez otro objeto **Text**.

### Otro ejemplo

Veamos un ejemplo algo más avanzado, fijaros en como es un lenguaje declarativo basado en JavaScript. Además como **Rectangle** es el contenedor de todo el árbol de objetos.



Las propiedades de los objetos se vinculan (binding) a expresiones dentro del árbol. **Cuando un binding cambia automáticamente se reevalúa la expresión** (en los lenguajes declarativos esto no sucede en todos los casos.)

## QML suma tendencias,

Tendencias en la Web:

- Amamos JavaScript y JSON.
- XML no nos gusta para ciertas cosas.
- Necesitamos lenguajes declarativos que separen UI de codificación

QML no es nada nuevo, más bien es una versión adaptada para **Qt** de una tendencia hacia lenguajes declarativos en los UI (Véanse MXML (Flash), XUL (Gecko), XAML (.Net)). La gran diferencia es la adopción de JSON en vez de XML.

También dispone de soporte XML, servicios XML y XPath.

## Qt Declarative

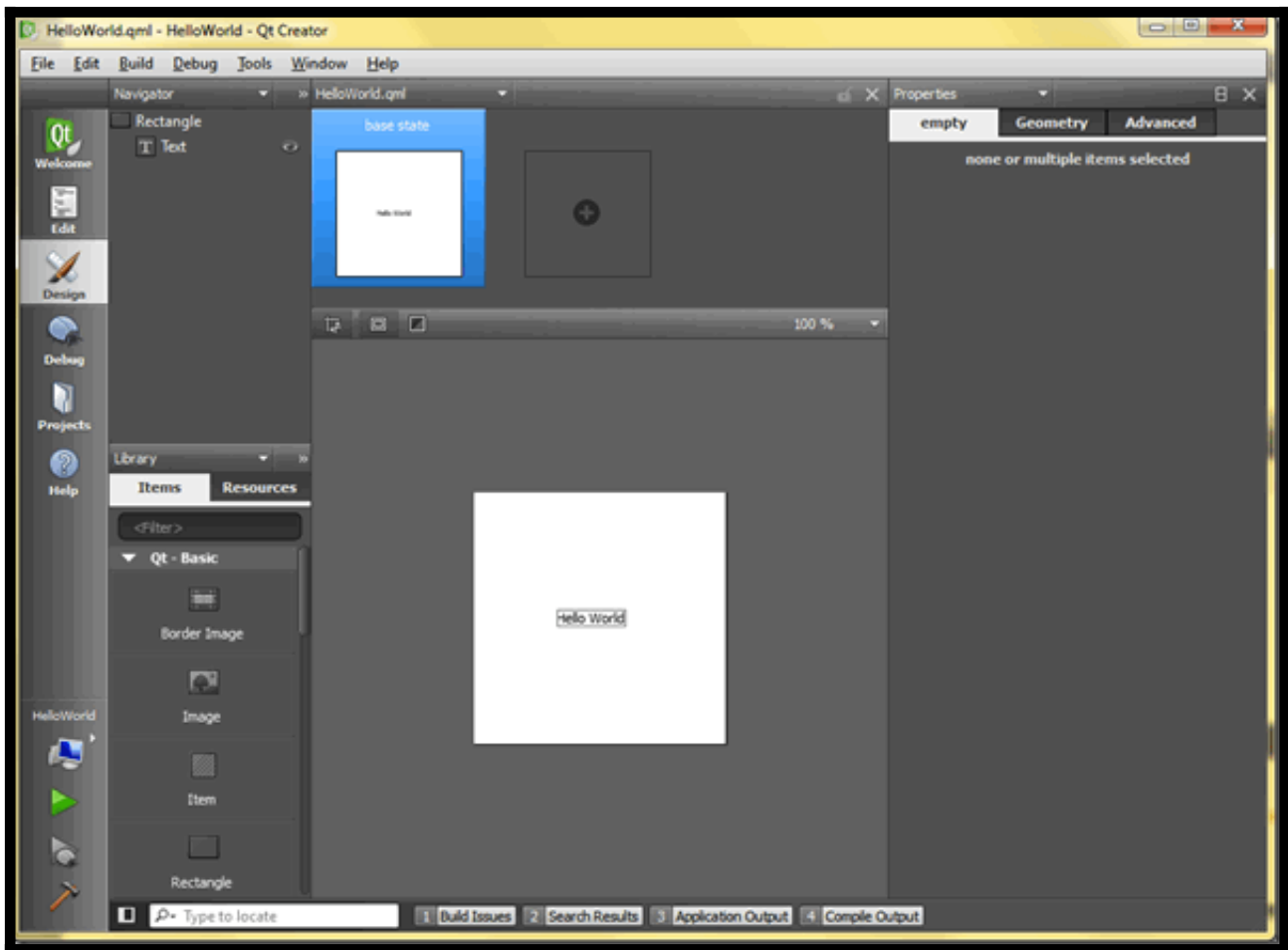
Es un Framework C++ que conecta QML y C++ para integrar “scripts” QML en aplicaciones C++ e integrar plug-in’s C++ en aplicaciones QML. Sigue en desarrollo.

## Qt Creator

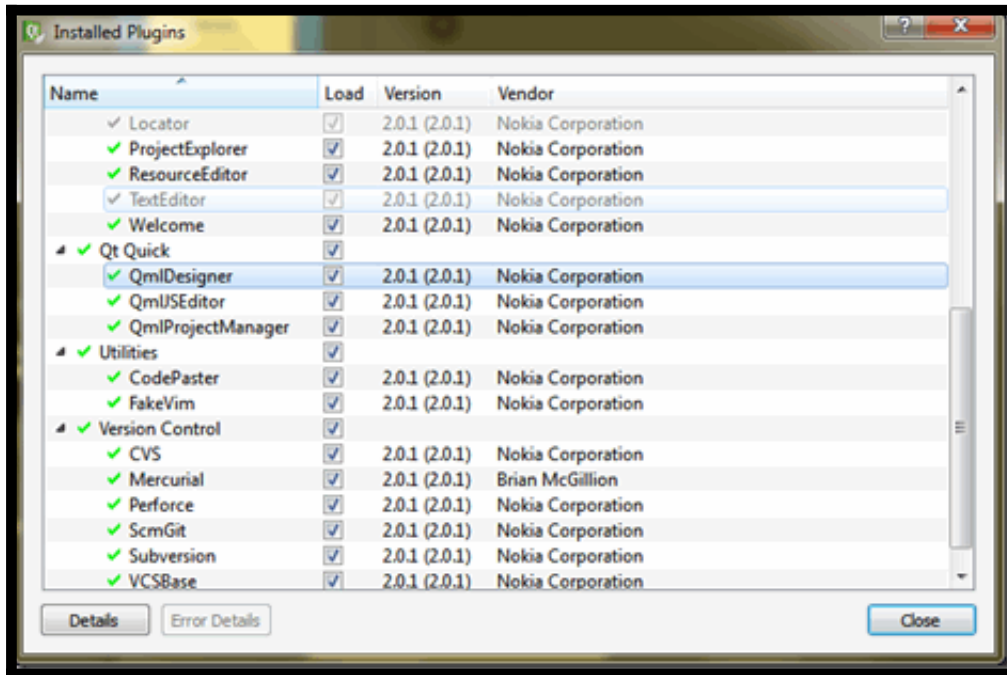
Qt Creator es IDE de desarrollo que integra C++ y QML. Consta de un diseñador QML (QML designer) para edición visual y de los editores de código asociados (QML y C++)

## QML designer

Sirve para la edición visual y de los editores de código asociados (QML y C++)



**NOTA:** Si por algún motivo no puedes editar mediante el Design revisa que el complemento está enable y prueba nuevamente



## Una última cuestión

La configuración del proyecto respecto a QML la tenéis en Projects

