ESTRUCTURA DE DATOS Y PROGRAMACIÓN ORIENTADA A OBJETOS

Tema Nº4:PyQt5 y QtDesigner

Indicador de logro Nº4:Instala y reconoce el entorno, documentación y librería de PyQt5 mediante el uso de formularios.

**TEMA 01 Teoría de los**

Imagen que contiene Icono

Descripción generada automáticamente

**TEMA Nº4:**

PyQt5 y QtDesigner

**Subtema 4.1:**

PyQt5 y QtDesigner

**PYQT5 INTERFACES GRÁFICAS CON PYTHON**

**¿Qué es PyQT5?**

Es un binding de la biblioteca gráfica QT para el lenguaje de programación Python. PyQT5 nos permite crear interfaces gráficas con python de manera rápida y sencilla, la legibilidad del código de Python hace que sea una tarea sumamente sencilla realizar interfaces gráficas, además que también posee una interfaz de diseño para crear nuestras interfaces gráficas.

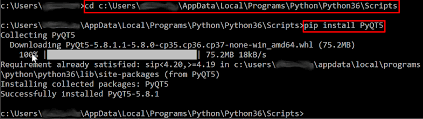
La flexibilidad que tiene esta biblioteca, es que podemos diseñar por completo nuestras interfaces y luego comenzar a programar, esto es un punto muy importante, ya que si has utilizado Tkinter para realizar interfaces gráficas con Python sabrás de lo forzoso que es tener que ir diseñando y programando a la vez.

Con Tkinter se puede diseñar primero y luego programar, pero el punto que contradice todo acá es que en Tkinter el diseño no lo hacemos a través de una interfaz de diseño como la posee PyQT5, si no que el diseño lo hacemos escribiendo más código.

**INSTALACIÓN DE PYQT5**

Para instalar PyQt5 debemos ejecutar desde la terminal el siguiente código:

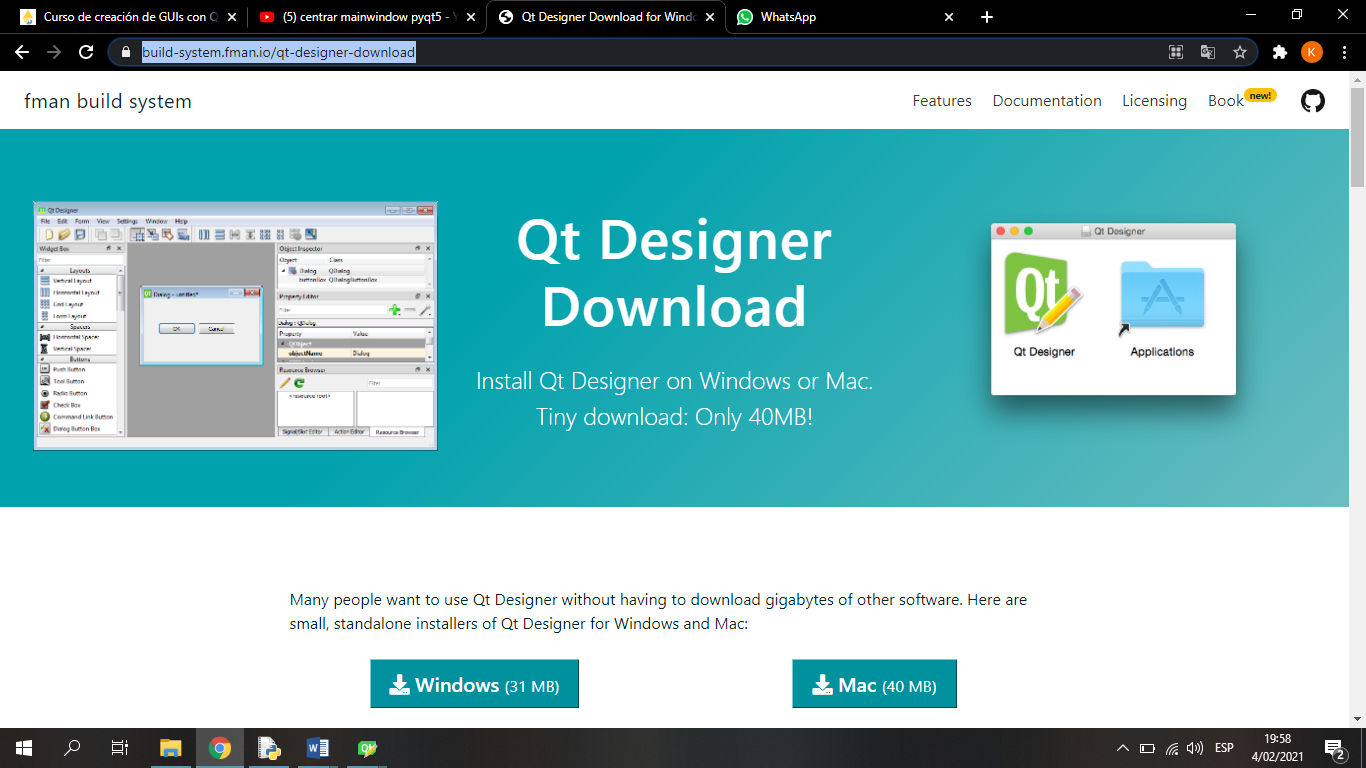
pip install PyQt5



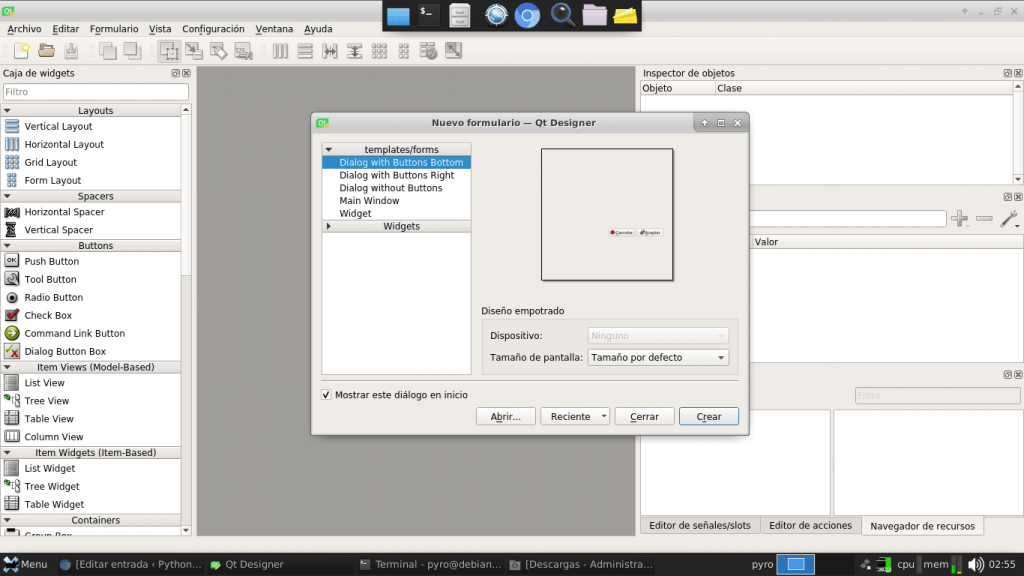
**INSTALACIÓN DE QTDESIGNER**

Podemos dirigirnos a la siguiente dirección: <https://build-system.fman.io/qt-designer-download>

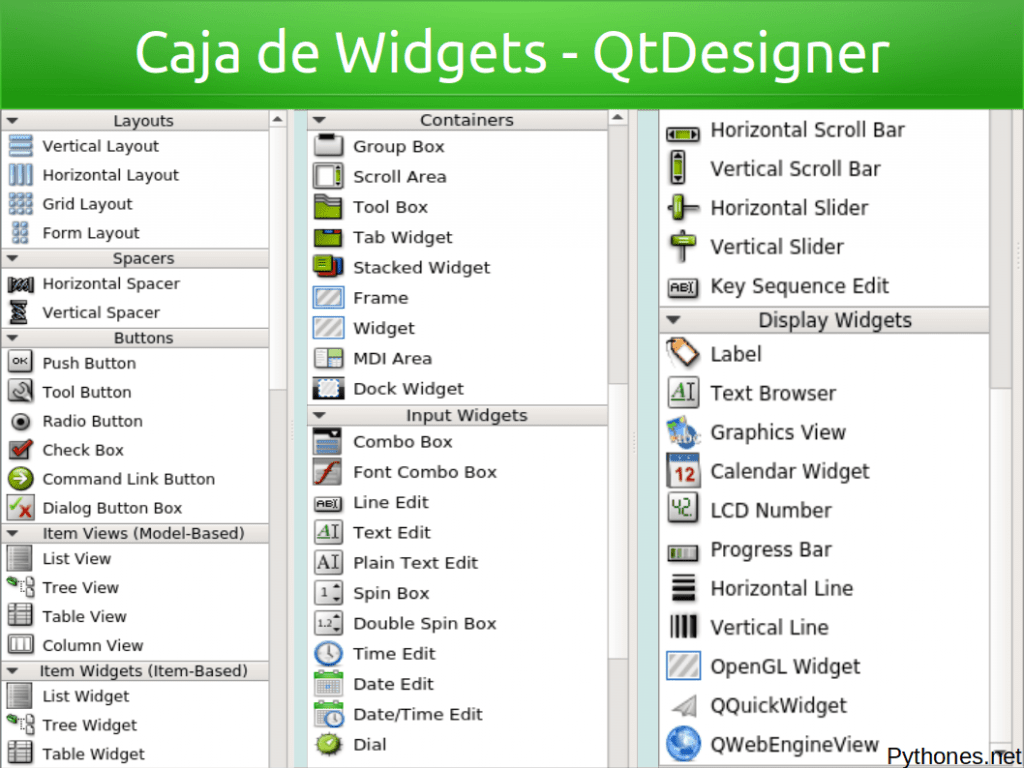
Descargar para Windows o Mac, ejecutarlo e iniciar la instalación.



Una vez instalada la aplicación, podemos ejecutarla.



**CONOCIENDO EL ENTORNO DE QTDESIGNER**



* + - 1. **LAYOUTS – DISEÑOS**

Comenzamos con el primer apartado de la lista correspondiente a Layouts o diseños en nuestro idioma. Este apartado contiene Widgets pyqt que podemos utilizar para cambiar la disposición de nuestro programa. Véase como una capa que nos permite alterar gráficamente el orden del espacio de cada Widget dentro de ella.

**USO EN DESIGNER:** Basta con agregar la capa y dimensionar, luego procedemos a ordenar dentro los demás Widgets.

|  |  |  |
| --- | --- | --- |
| WIDGET | MÉTODOS | VISUAL |
| Vertical Layout | QLayout | Vertical Layout |
| Horizontal Layout | QLayout | https://pythones.net/wp-content/uploads/2019/04/horizontal-layout-min.png |

|  |  |  |
| --- | --- | --- |
| Grid Layout (Grilla) | QGridLayout | https://pythones.net/wp-content/uploads/2019/04/grid-Layout-min.png |
| Form Layout | QFormLayout | Form Layout |

* + - 1. **SPACERS – ESPACIADORES**

Como su nombre lo indica su utilidad es añadir un espacio por ejemplo entre un botón y otro. O entre un widget y otro.

|  |  |  |
| --- | --- | --- |
| WIDGET | MÉTODOS | VISUAL |
| Horizontal Spacer | QspacerItem | https://pythones.net/wp-content/uploads/2019/04/horizontal-space-min.png |
| Vertical Spacer | QspacerItem | https://pythones.net/wp-content/uploads/2019/04/vertical-spacer-min.png |

* + - 1. **BUTTONS – BOTONES**

Encontramos una variedad de botones dentro de esta categoría. Tanto el simple botón, como checkbox, o conjunto de botones como aquellos clásicos de los diálogos donde nos formulan aceptar o denegar una sentencia.

|  |  |  |
| --- | --- | --- |
| WIDGET | MÉTODOS | VISUAL |
| PushButton | QPushButton | PushButton |
| ToolButton | QToolButton | ToolButton |
| RadioButton | setChecked()  setText()  text()  isChecked() | RadioButton |
| CheckBox | QCheckBox | CheckBox |
| CommandLinkButton | QCommandLinkButton | CommandLinkButton |
| DialogButtonBox | QDialogButtonBox | DialogButtonBox |

* + - 1. **ITEMS VIEWS – VISOR DE ITEMS**

Los visores de items son útiles para “visualizar” Listas de elementos y seleccionar alguno de ellos. También se nos permite crear y modificar tablas, o crear arboles de elementos. (Lo olvide en el ejemplo, pero se conocen como arboles de datos aquellos que desprenden de un superior o se dividen en ramificaciones.)

|  |  |  |
| --- | --- | --- |
| WIDGET | MÉTODOS | VISUAL |
| List Views  (Visor de listas) | QListWidget | List View |
| Tree Views  (Visor en formato árbol) | QTreeView | Tree View |
| Table Views  (Visor de Tablas) | QTableWidget | Table View |
| Column Views  (Visor de columnas) | QColumnView | Column View |

* + - 1. **CONTAINERS – CONTENEDORES**

Los contenedores nos permiten organizar los widgets pyqt dentro de ellos para administrar mejor el espacio o jugar con el diseño de la aplicación; añadiendo barras, pestañas u organizando las sub ventanas que de ella se derivan.

|  |  |  |
| --- | --- | --- |
| WIDGET | MÉTODOS | VISUAL |
| Group Box  (Caja, Grupo de elementos) | QGroupBox | https://pythones.net/wp-content/uploads/2019/04/GroupBox-min.png |
| Scroll Area  (Área de scroll) | QScrollArea | ToolButton |
| Tool Box  (Caja de herramientas) | QToolBox | Tool Box |
| Tab Widget  (Pestañas) | QTabWidget | Tab Widget |
| Stacked Widget  (Widgets Apilables) | QStackedWidget | Stacked Widget |
| Frame | Qframe | frame |

|  |  |  |
| --- | --- | --- |
| MDI Area  Widget que sirve para organizar mejor las sub-ventanas que tenga nuestra aplicación. | QMdiArea | QMdi Widget |
| Dock Widget  (Barra lateral, Superior o inferior estilo Dock. ) | QDockWidget | https://pythones.net/wp-content/uploads/2019/04/Dock-Widget-min.png |

* + - 1. **INPUT WIDGETS PYQT**

Como hemos visto anteriormente el significado de input puede verse como “ingresar” en nuestro idioma o “entrada de datos”. Los widgets pertenecientes a esta categoría nos brindan formatos de entrada de información. Tanto mediante opciones como un Combo / Spin Box o el clásico Line Edit, Text Edit, etc.

|  |  |  |
| --- | --- | --- |
| WIDGET | MÉTODOS | VISUAL |
| Combo Box | QComboBox | Combo Box |
| Font Combo Box | QFontComboBox | Font Combo Box |
| Line Edit | QlineEdit | Line Edit |
| Text Edit | QtextEdit | Text Edit |
| Plain Text Edit | QPlainTextEdit | Plain Text Edit | |
| Spin Box | QSpinBox | Spin Box | |
| Double Spin Box | QDoubleSpinBox | Double Spin Box | |
| Time Edit | QtimeEdit | Time Edit | |
| Date Edit | QDateEdit | Date Edit | |
| Date / Time Edit | QdateTimeEdit | Date - Time Edit | |
| Dial | Qdial | Dial | |
| Horizontal Scroll | QAbstractHorizontalScroollArea | Horizontal Scroll | |

|  |  |  |
| --- | --- | --- |
| Vertical Scroll | QAbstractScrollArea | Vertical Scroll Bar |
| Horizontal Slider | Qslider | Horizontal Slider |
| Vertical Slider | Qslider | Vertical Slider |
| Key Sequence Edit | QkeySequenceEdit | Key Sequence Edit |

* + - 1. **DISPLAY WIDGETS**

Podemos definir esta categoría como aquella que contiene los widgets que nos brindan los datos de salida, tanto mediante labeles, gráficos, calendarios, barras de progreso, etc.

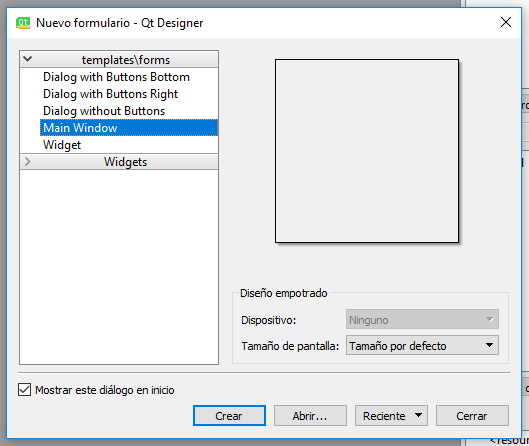
|  |  |  |
| --- | --- | --- |
| WIDGET | MÉTODOS | VISUAL |
| Label | Qlabel | Label |
| Text Browser | QtextBrowser | Text Browser |
| Graphic View | QGraphicsView | Graphic View |
| Calendar Widgets | QcalendarWidget | Calendar Widget |
| LCD Number | QLCDNumber | LCD Number |
| Progress Bar | QProgressBar | Progress Bar |
| Horizontal Line | QLine | Horizontal Line |

|  |  |  |
| --- | --- | --- |
| Vertical Line | QLine Métodos | Vertical Line |
| OpenGl Widget | QOpenGLWidget | OpenGl Widget |

**Actividad:**

**NUESTRO PRIMER PROYECTO**

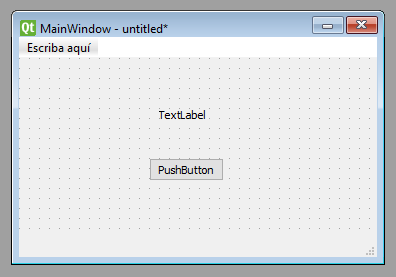
Cada vez que ponemos en marcha Qt Designer el programa nos pide qué queremos crear. Como sólo vamos a tantear el terreno vamos a indicarle crear una “Main Window” o ventana principal que normalmente representará la raíz de un programa gráfico:



Qt Designer, o mejor dicho Qt, funciona con un sistema de Widgets. Podríamos considerar un Widget como un componente de la interfaz. Prácticamente todo son Widgets, desde la propia ventana, los botones, las etiquetas de texto… y lo que en realidad representan en código son clases y objetos, pero eso lo veremos luego.

Por ahora recomiendo hacer un pequeño experimento para aprender un poco de qué va el asunto.

Podríamos por ejemplo arrastrar un par de widgets de la Caja de widgets a la ventana que nos aparece en medio. Uno de tipo Push Button y una Label, puedes incluso redimensionar la ventana para hacerla más pequeña, que nos quede algo así:



Con esto tenemos listo el diseño de nuestro programa, podríamos ver cómo queda presionando la combinación Ctrl + R.

