

Programacion II [PRACTICA]

1
0
0
1
0
1
0
1
0
1
0
1
0
1
0
0
0

1
0
1
0
0
1
0
1
0
1
0
0
1
0
1
0



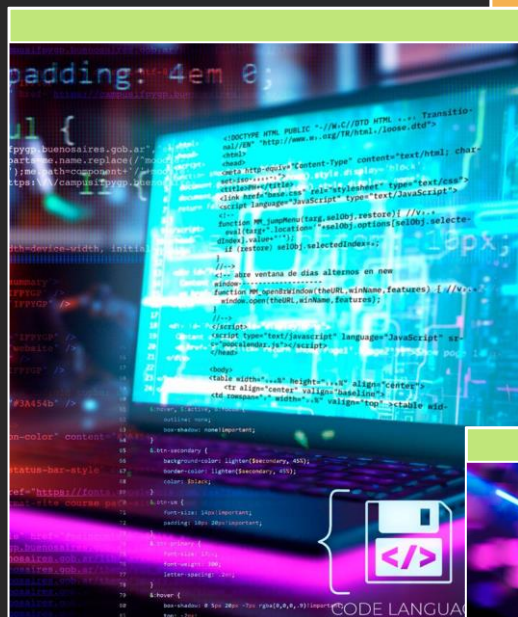
CLASE

06



INTERFAZ GRAFICA DE USUARIOS (G.U.I)

1
1
0
0
1
0
1
0
1
0
1
0
1
1
1
0

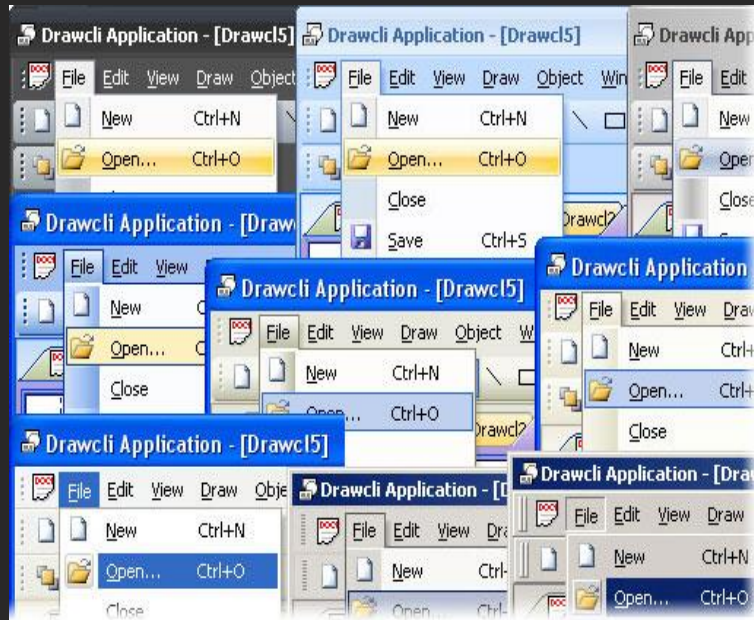


1
0
1
0
1
0
1
1
1
0
0
0
1
1
1
0

Interfaz Gráfica de Usuario o GUI



En informática llamamos Interfaz Gráfica de Usuario o GUI (por sus siglas en inglés Graphical User Interface) al software que permite la interacción con la máquina de manera gráfica, esto es con elementos como botones, ventanas, iconos o enlaces.



Es término interfaz gráfica de usuario nació a finales de la década de los 70, cuando las interfaces de usuario eran interfaces de línea de comandos. De esta manera, las primeras interfaces que contaban con iconos y gráficos bonitos empezaron a destacar y rápidamente los desarrolladores y marcas apostaron por desarrollar e implementar esta tecnología, puesto que acercaba el idioma complejo de la informática a cualquier usuario.

Estructura de una Aplicación QT



Todo proyecto con interfaz visual (GUI) en Qt, tiene los mismos elementos que son

- Ficheros de formularios (forms) - Con la extensión *.ui
- Ficheros Cabecera (headers) - Con la extensión *.h
- Ficheros Fuente (sources) - Con la extensión *.cpp
- Fichero de proyecto (Project) - Con la extensión *.pro

Estructura de una Aplicación QT- forms



Ficheros de formularios (forms).- Con la extensión *.ui,

En lenguaje XML es la interfaz gráfico de formularios o ventana de la aplicación.

- No todas las ventanas tienen que estar en un fichero *.ui, pueden estar implementadas en C++.
- Estos ficheros son diseñadas gráficamente en el Qt Designer, el cual genera automáticamente el código XML de dichos ficheros.

Estructura de una Aplicación QT - headers ...

Ficheros Cabecera (headers).- Con la extensión *.h.

- Lo más recomendable es que la declaración de cada clase vaya en un fichero cabecera por separado,.
- Y el nombre del fichero coincida con el nombre de la clase que se declara.

Ejemplo, si vamos a desarrollar una clase llamada calculator, el fichero cabecera que la declara por completo la llamaríamos "calculator.h".

Estructura de una Aplicación QT - Sources...

Ficheros Fuente (sources).- Con la extensión *.cpp.

- Lo más recomendable es que la implementación o desarrollo de cada miembro de una clase, esté contenido por completo en un fichero con el mismo nombre que la clase que implementa.
- Ejemplo, la implementación de la clase calculator, estaría en el fichero de nombre "calculator.cpp". Dentro de los ficheros fuentes, siempre hay uno que contiene la función principal (main) y cuyo nombre será "main.cpp".

Estructura de una Aplicación QT - Project ...

Fichero de proyecto (project).- Con la extensión *.pro.

- Es un fichero que puede generarse y rellenarse automáticamente con el Qt Creator,
- o que puede no existir, y deba antes ser creador con el comando "qmake -project" antes de la compilación definitiva.
- El nombre del mismo coincidirá con el de la carpeta de proyecto creada.

¿Que es QObject?



Es la clase principal. Todos los widgets usados para crear la GUI derivan de esta clase

Característica

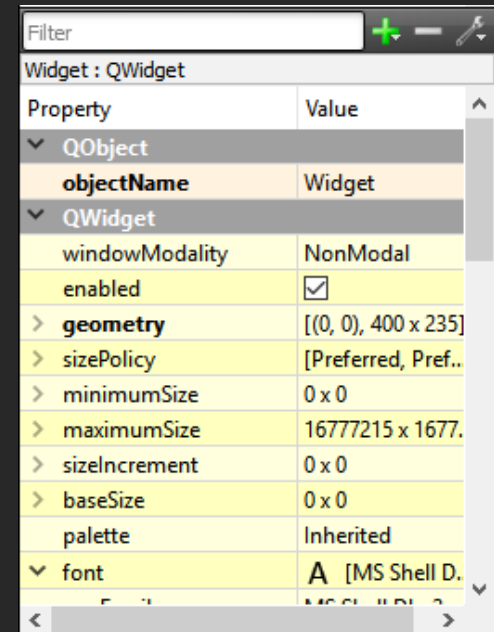
- No es copiable, toda instancia de la misma es única
- Todo objeto QObject es único e individual
- Tienen un nombre (propiedad *objectName*)
- Cada objeto, esta situado en algún lugar específico de la jerarquía de clases QObject
- Finalmente cada QObject, tiene conexión con otros con otros objetos mediante conexiones señales-slot

QObject - Propiedades de un objeto



Todo objeto, necesita guardar su estado en algún sitio, donde se defina por completo el estado del mismo y todas sus particularidades, esto es lo que llamamos propiedades de un objeto.

En el QtDesigner, a abrir un fichero form (*.ui), podemos pulsar sobre cualquiera de los elementos que hay en el form, y a la derecha en la parte inferior, ver todas las propiedades que dicho objeto tiene, y los valores que esas propiedades toma para ese objeto en particular



Property	Value
Widget : QWidget	
QObject	
objectName	Widget
QWidget	
windowModality	NonModal
enabled	<input checked="" type="checkbox"/>
geometry	[(0, 0), 400 x 235]
sizePolicy	[Preferred, Pref..
minimumSize	0 x 0
maximumSize	16777215 x 1677.
sizeIncrement	0 x 0
baseSize	0 x 0
palette	Inherited
font	A [MS Shell D.

<< A PROGRAMAR >>



mainwindow.cpp @ QtCreatorHolaMundo - Qt Creator

File Edit View Build Debug Analyze Tools Window Help

Projects

- QtCreatorHolaMundo
 - QtCreatorHolaMundo.pro
 - Headers
 - mainwindow.h
 - Sources
 - main.cpp
 - mainwindow.cpp
 - Forms
 - mainwindow.ui

Welcome Edit Design Debug Projects Help

Open Documents

- main.cpp
- mainwindow.cpp*
- mainwindow.h
- mainwindow.ui

```
1  #include "mainwindow.h"
2  #include "ui_mainwindow.h"
3
4  MainWindow::MainWindow(QWidget *parent)
5      : QMainWindow(parent)
6      , ui(new Ui::MainWindow)
7  {
8      ui->setupUi(this);
9  }
10
11 MainWindow::~MainWindow()
12 {
13     delete ui;
14 }
15
16
17
18
```

1 Issues 2 Search Results 3 Application Output 4 Compile Output 5 Terminal 6 QML Debugger Console 7 General Messages 9 Test Results

Retos de Programacion

```
<div className="menu">
  {categoryList.map((val) => {
    return (
      <DropDownItem
        leftIcon={val.icon} gotoMenu={val.id} key={val.id}>
        <DropDownItem
          leftIcon={val.icon}
          gotoMenu={val.id}
          key={val.id}
          rightIcon={<RightArrowIcon />}
        </DropDownItem>
      </DropDownItem>
    )
  })}
</div>
```

```
test react-scripts test
react-scripts build
"scripts": {
  "start": "react-scripts start"
}
```

1
0
0
1
0
1
0
1
0
0
1
0
0
1
0
1
0
1
0
0

1
0
1
0
0
1
1
0
0
1
0
1
1
0
1
1
1
1
1

1. Crear un nuevo proyecto en Qt Creator y agregar un botón desde el archivo .ui. Al hacer clic, mostrar un mensaje en una etiqueta.
2. Diseñar una ventana con un formulario de entrada de texto (QLineEdit) y un botón. Al presionar el botón, mostrar el texto ingresado en un QLabel.
3. Agregar tres botones a un archivo .ui que controlen el color de fondo de la ventana (rojo, verde, azul).
4. Crear un layout en el archivo .ui con varios widgets (botones, etiquetas) y probar su comportamiento al cambiar el tamaño de la ventana.

02 Calculadora (Front)



Desarrollar una calculadora (solo el front) tal como esta en la imagen. Usando:

- Grid Layout
- QPushButton
- QLabel

1
0
1
0
0
1
0
0
1
0
1
1
0
1
1
1
1

