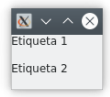
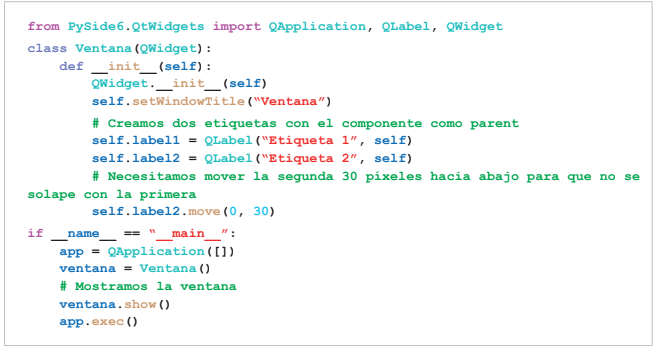
1. Contenedores de componentes. Diseño



3.1. Layouts

Hasta ahora hemos visto ventanas con un único componente y componentes contenidos dentro de otros componentes, como es el caso del ejemplo siguiente. En la imagen puedes ver el resultado.

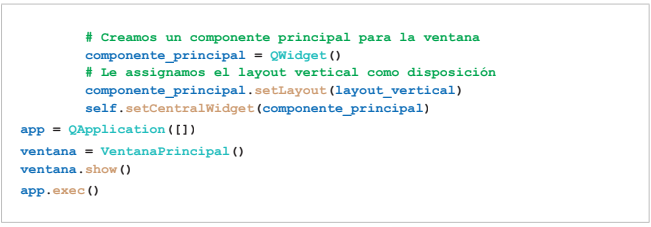




Pero ¿qué pasa si queremos añadir más componentes, tanto horizontalmente como verticalmente? ¿Qué pasa si redimensionamos la ventana? Tendríamos que ir calculando el número de píxeles a desplazar y el espacio que ocupan en la interfaz no quedaría modificado. Así pues, en este apartado vamos a estudiar una forma más eficiente de gestionar todo esto a través de layouts: diseños o disposiciones que podemos aplicar a una interfaz para ordenar sus componentes. Con la combinación de estos layouts es posible definir el diseño de cualquier interfaz gráfica de usuario.

# 3.2. QVBoxLayout

La primera disposición que estudiaremos será la disposición en vertical. Se irán añadiendo los componentes al final de una pila de componentes, uno encima de otro.



Hemos definido un componente principal de tipo QWIdget al que le asignamos un layout vertical. A este layout le añadimos los componentes que vamos a utilizar. Si ahora probamos a redimensionar la ventana, los componentes cambian automáticamente de tamaño para ajustarse al ancho de la ventana y repartirse de forma equitativa verticalmente.

## 3.3. QHBoxLayout

En este apartado nos centramos en la disposición horizontal de los componentes. Usamos un layout horizontal:



Si redimensionamos la ventana, los botones no crecen verticalmente, pero sí lo hacen horizontalmente de forma proporcional.

## 3.4. QGridLayout

Aunque con el uso de layouts verticales y horizontales podríamos conseguir casi cualquier disposición, esto puede no resultar cómodo de gestionar en algunas ocasiones. En estos casos, puede ser más útil utilizar e layout en forma de cuadrícula. Veamos un ejemplo:



El resultado es el siguiente: 

3.5. QFormLayout

En algunos casos puede resultar muy cómodo utilizar el QFormLayout, que es un diseño pensado para la elaboración de formularios. Normalmente se usa como entrada de información, pero también puede servir para visualizarla si se deshabilita la entrada de datos en los componentes. Veamos un ejemplo simple. El resultado es el que puedes ver.

Interfaz de usuario gráfica

Descripción generada automáticamente



3.6. QStackedLayout

Por último, vamos a ver un layout que permite apilar componentes, pero no verticalmente, de modo que todos son visibles, sino en profundidad, de forma que solo uno de los elementos será visible, como si lo trajésemos al frente. Para gestionar qué elemento es visible utilizamos setCurrentlndex o setCurrentWidget. Veamos un ejemplo:





El resultado es una interfaz parecida al uso de pestañas, pero con botones:

