



Akademy-es 2009

aacid@kde.org



# Cómo hacer una buena API

Albert Astals Cid

Julio, 2009



- 1 Créditos
- 2 Características de una buena API
- 3 El proceso de diseño
- 4 Guías de diseño



- 1 Créditos
- 2 Características de una buena API
- 3 El proceso de diseño
- 4 Guías de diseño



Esta presentación está basada en "The Little Manual of API Design" de Jasmin Blanchette, QtSoftware, Nokia



- 1 Créditos
- 2 Características de una buena API
- 3 El proceso de diseño
- 4 Guías de diseño



# Características de una buena API

Los 5 puntos



- Es fácil de aprender y memorizar
- Hace el código más leible
- Hace difícil que te equivoques
- Es fácil de extender
- Es completa



# Fácil de aprender y memorizar

API para vagos desmemoriados



- Nombres consistentes
- Patrones consistentes
- Predecible
- Semánticamente coherente
- Principio de la menor sorpresa
- No demasiados *convenience methods*



- El código se escribe una vez pero se lee muchas
- En APL una línea de 15 caracteres sirve para calcular una lista de todos los numeros primos hasta N
- !Perl
- Ser conciso no es una necesidad para la legibilidad





Qt3:

```
slider = new QSlider(8, 128, 1, 6, Qt::Vertical, 0,  
"volume");
```

Qt4:

```
slider = new QSlider(Qt::Vertical);  
slider->setRange(8, 128);  
slider->setValue(6);  
slider->setObjectName("volume");
```



*Qt Jambi:*

```
QGridLayout layout = new QGridLayout;  
layout.addWidget(slider, 0, 0);  
layout.addWidget(spinBox, 0, 1);  
layout.addWidget(resetButton, 2, 1);  
layout.setRowStretch(1, 1);  
setLayout(layout);
```



*Swing:*

```
GridBagLayout layout = new GridBagLayout();  
GridBagConstraints constraint = new  
GridBagConstraints();  
constraint.fill = GridBagConstraints.HORIZONTAL;  
constraint.insets = new Insets(10, 10, 10, 0);  
constraint.weightx = 1;  
layout.setConstraints(slider, constraint);  
constraint.gridwidth =  
GridBagConstraints.REMAINDER;  
constraint.insets = new Insets(10, 5, 10, 10);  
constraint.weightx = 0;  
layout.setConstraints(spinner, constraint);  
9 líneas más...
```



*Página 10 del manual*

*Qt3:*

```
button = new QPushButton(this, "Hello world");
```

*Qt4:*

```
button = new QPushButton(this);  
button->setObjectName("Hello world");
```

*addItem vs insertItem*



- Espacio de nombres
- ABI/API estable
- d-pointer
- no retornar referencias



- 1 Créditos
- 2 Características de una buena API
- 3 El proceso de diseño
- 4 Guías de diseño



- Conoce tus requisitos
- Escribe antes los casos de uso que el código
- Mira ejemplos de API similares
- Define la API antes de implementar nada
- Haz que te revisen la API
- Escribe ejemplos contra la API
- Preparate contra extensiones
- No publiques API internas
- Ante la duda, ...



- 1 Créditos
- 2 Características de una buena API
- 3 El proceso de diseño
- 4 Guías de diseño





- Nombres autoexplicativos (QPainterPath/QWorkspace)
- Evita los booleanos
- No uses sinónimos (widgets/controls)
- Orden de parámetros
- Similitud con clases existentes
- No uses abreviaciones
- Nombres específicos mejor que generales
- No seas un esclavo de la API padre



- Escoge buenos valores predeterminados
- No hagas tu API demasiado lista
- Atención a los casos extremos
- Cuidado con las API virtuales



- Busca API basada en propiedades
- La mejor API es la API que no existe



**Gracias.** ¿Preguntas?

<http://www.kde-espana.es>

Albert Astals Cid  
[aacid@kde.org](mailto:aacid@kde.org)