



El puzzle de

QtQuick

# Charla:

- Evolución de Qt hasta llegar a QtQuick
- QML práctico

# Graphics View System

Provee un canvas en el cual pintar objetos 2D hechos por uno mismo.

# Graphics View System

- Gran eficiencia en el manejo de objetos
- Control de eventos
- Api de bajo nivel
- Propenso a crear código espagueti

# Graphics View System (Ejemplos)

¡Mirad vuestro escritorio! ¡Plasma!

# QGraphicsAnchorLayout

Layout que permite “enganchar” los objetos entre ellos o dentro del mismo layout.

# QGraphicsAnchorLayout

- Basado en políticas
- Simple
- Reposicionamiento gratis
- No te hace pensar

# Kinetic

Framework para la creación de animaciones,  
fácil y sencillo



# Kinetic

- Ahorra mucho trabajo
- Animaciones predefinidas
- Escalable

# Kinetic (Ejemplo)

-Miremos nuestro escritorio otra vez!

# QML

Lenguaje declarativo “Meta Markup Modeling”

# QML

- Lenguaje declarativo basado en javascript
- Soporta Javascript
- Intuitivo y divertido de usar
- Basado en web
- -Elementos ya creados
- -Layouts basados en estados
- -¡Animaciones gratis!
-

# QML (Código)

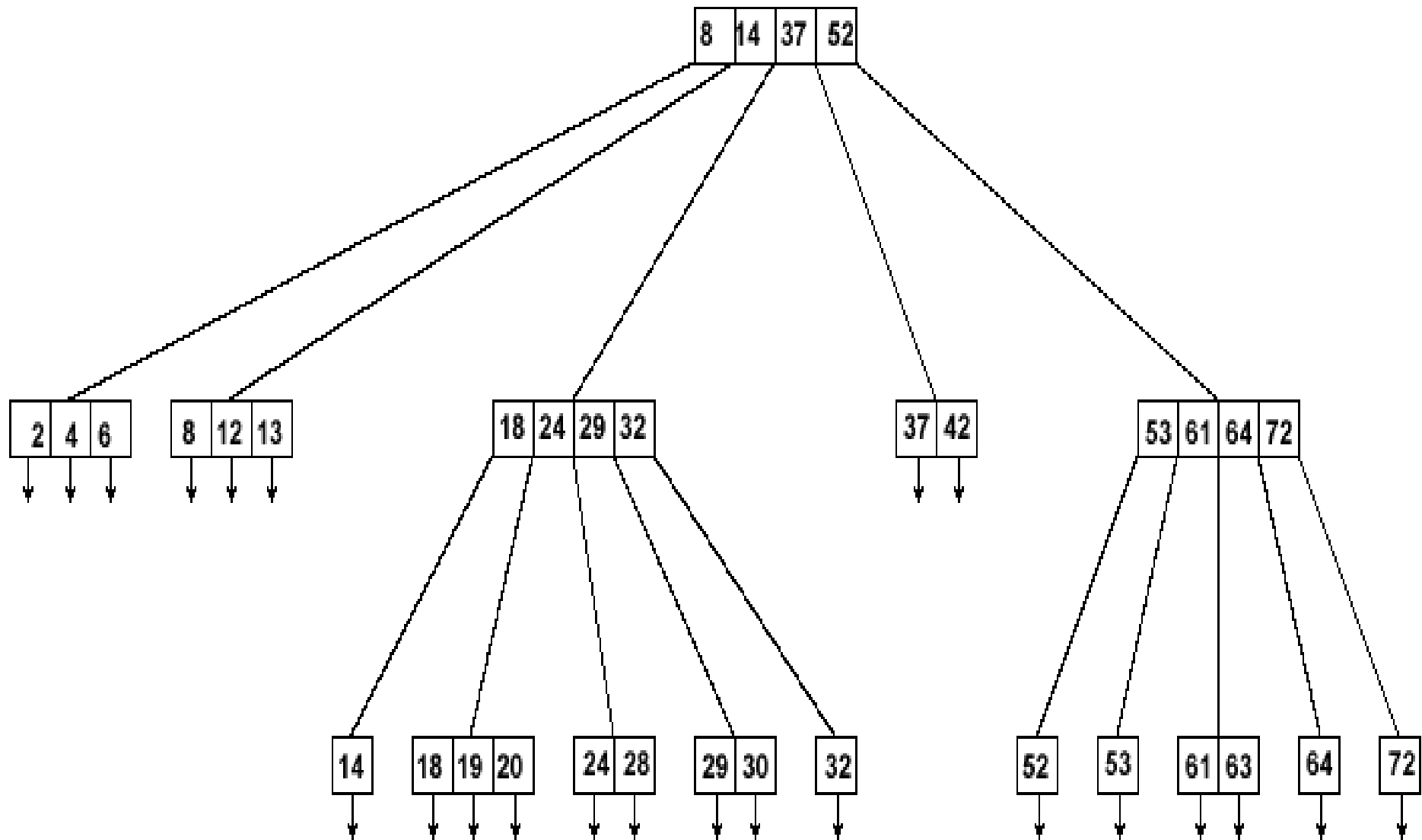
Usuarios cierren sus ojos

# QObject

“Super” objeto en el cual se basa todo Qt

# QOject

- Comunicación entre objetos
- Árboles de QObjets
- Bosques de QObjets!
- Propiedades dinámicas





# QML a programar!

- Se empieza a desarrollar desde abajo
- La gui se declara en los archivos .qml

# Alguna ayuda? QtCreator!

- Wizard para la creación de aplicaciones
- "Syntax Highlighting"
- Comprobación de errores
- Completación de código
- Designer!

# QML y C++, amigos :)

- Posibilidad de incrustar QML en nuestra aplicación
- Comunicación entre QML y C++
- Vistas (MVC) hechas con QML