

---

# Kandau: Persistència d'objectes per aplicacions KDE

*[kandau.berlios.de](http://kandau.berlios.de)*

Albert Cervera i Areny

# Índex

---

- Introducció
- Característiques
- Exemple

---

# Introducció

# Introducció

---

- Sistema de persistència d'objectes
- Utilitza Qt i llibreries KDE
- Proveeix introspecció
- Generació automàtica i semi-automàtica d'IUs

---

# Característiques

# Persistència

---

- Orientat a transaccions: `commit()` i `rollback()` sense `begin()`
- Gestionat per la classe `Manager`. Gestió automàtica de la càrrega i alliberament dels objectes.

```
Book *book = new Book(); delete book;
```



```
Book *book = Book::create();
```

- Possibilitat de desar en un fitxer XML, BBDD SQL, fitxer `.desktop`.
- Possibilitat de crear nous *backends*

# Introspecció

---

- Introspecció de classes. Conèixer quines classes hi ha disponibles.
- Introspecció de propietats d'una classe.
- Introspecció de relacions entre classes.
- Metainformació relacionada amb la classe.

Tot sense instanciar cap objecte.

# Interfície Gràfica d'Usuari

- Generació automàtica de diàlegs.
- Ofereix nous widgets que exploten la introspecció.



# Interfície Gràfica d'Usuari

- Càrrega de fitxers .ui del Designer i la seva gestió automàtica.
- Utilitza gestors de widgets.

The image shows two screenshots related to Qt Designer. The left screenshot displays the Qt Designer interface with a form titled 'Form2'. The form contains a 'Number' field with the value '0', a 'Date' field with the value '00/00', a 'Customer' dropdown menu, and an 'Articles' table with columns 'code', 'label', and 'description'. The 'Articles' table has one row with '1' in the 'code' column, '2 Article Two' in the 'label' column, and 'Description of article number tw' in the 'description' column. The 'Properties' panel on the right shows various widget properties such as 'maximumSize', 'paletteForegroundColor', 'font', and 'whatsThis'. The right screenshot shows a preview window titled 'CustomerOrder(1) - Test' with the same form rendered. The 'Number' field now contains '99', the 'Date' field contains '10/06/2006', the 'Customer' dropdown shows 'bar ()', and the 'Articles' table has one row with '1' in the 'code' column, '2 Article Two' in the 'label' column, and 'Description of article number tw' in the 'description' column. The preview window has 'OK' and 'Cancel' buttons at the bottom.

# Altres utilitats

---

- Sistema de notificacions d'esdeveniments
  - Emetre un signal quan es modifica una propietat concreta, un objecte, una classe...
- Objectes dinàmics
  - Crear noves classes, afegir i treure propietats i relacions en temps d'execució.
  - Podem tornar a crear l'esquema de la base de dades.
  - Podem establir relacions entre objectes dinàmics i de C++.

---

# Exemple: L'aplicació Prestatge

# Declaració de la classe Book

---

```
class Book : public Object
{
    Q_OBJECT
    Q_PROPERTY( QString title READ title WRITE setTitle )
    Q_PROPERTY( QString isbn READ isbn WRITE setIsbn )
    Q_PROPERTY( uint year READ year WRITE setYear )
public:
    DCLASS( Book );
    void setTitle( const QString& title );
    const QString& title() const;
    ...
    void setAuthor( Author *author );
    Author* author() const;
private:
    QString m_title;
    QString m_isbn;
    uint m_year;
};
```

# Implementació de la classe Book

---

```
ICLASS( Book );

void Book::createRelations()
{
    OBJECT( Author );
}

void Book::setTitle( const QString & title )
{
    MODIFIED;
    m_title = title;
}

const QString & Book::title( ) const
{
    return m_title;
}
```

# Implementació de la classe Book

---

```
void Book::setAuthor( Author *author )  
{  
    SETOBJECT( Author, author );  
}
```

```
Author* Book::author() const  
{  
    return GETOBJECT( Author );  
}
```

# Declaració de la classe Author

---

```
class Author : public Object
{
    Q_OBJECT
    Q_PROPERTY( QString firstName READ firstName WRITE setFirstName );
    Q_PROPERTY( QString lastName READ lastName WRITE setLastName );
    Q_PROPERTY( QString fullName READ fullName );
    Q_PROPERTY( QString biography READ biography WRITE setBiography );
    Q_PROPERTY( uint birthYear READ birthYear WRITE setBirthYear );
public:
    DCLASS( Author );
    ...
    Collection* bibliography();
private:
    QString m_firstName;
    QString m_lastName;
    QString m_biography;
    uint m_birthYear;
};
```

# Implementació de la classe Author

---

```
ICLASS(Author);

static const LabelDescription authorLabels[] = {
    { "firstName", I18N_NOOP("First Name")},
    { "lastName", I18N_NOOP("Last Name")},
    { "fullName", I18N_NOOP("Full Name") },
    { "biography", I18N_NOOP("Biography") },
    { "birthYear", I18N_NOOP("Birth Year") },
    { "bibliography", I18N_NOOP("Bibliography") },
    LabelDescriptionLast
};

void Author::createRelations()
{
    COLLECTION( Book );
    ADDMETAINFO( "labels", new LabelsMetaInfo( authorLabels ) );
}
```

# Implementació de la classe Author

---

```
QString Author::fullName( ) const
{
    QString sfirst = m_firstName.stripWhiteSpace();
    QString slast = m_lastName.stripWhiteSpace();
    if ( sfirst != "" && slast != "" )
        return slast + ", " + sfirst;
    else
        return sfirst + slast;
}

Collection* Author::bibliography()
{
    return GETCOLLECTION( Book );
}
```

# Main

---

```
Classes::setup();
DbBackendIface *backend = new XmlDbBackend( "books.xml", true );
Manager *manager = new Manager( backend );

ObjectRef<Author> monzo = Author::create();
monzo->setFirstName( "Quim" );
monzo->setLastName( "Monzó" );
monzo->setBirthYear( 1952 );
monzo->setBiography( "Born in Barcelona in 1952, has worked as "
    "a graphic designer, translator, journalist and has "
    "contributed to radio and TV programs." );
ObjectRef<Book> benzina = Book::create();
benzina->setTitle( "Benzina" );
benzina->setIsbn( "84-7727-434-7" );
benzina->setYear( 1983 );
monzo->bibliography()->add( benzina );
Manager::self()->commit();
kdDebug() << "The author of the book 'Benzina' is " <<
    benzina->author()->fullName() << endl;
```

# Generació d'una IGU

---

```
#include <classmainwindow.h>
```

```
...
```

```
ClassMainWindow *window = new ClassMainWindow();  
window->show();  
app.setMainWidget( window );  
return app.exec();
```

# Iterar a través de les classes

---

```
ClassInfoIterator it( Classes::begin() );
ClassInfoIterator end( Classes::end() );
ClassInfo *info;
for ( ; it != end; ++it ) {
    info = *it;
    kdDebug() << "Class: " << info->name() << endl;
    kdDebug() << "\tProperties:" << endl;
    PropertiesInfoConstIterator pit( info->propertiesBegin() );
    PropertiesInfoConstIterator pend( info->propertiesEnd() );
    PropertyInfo *pinfo;
    for ( ; pit != pend; ++pit ) {
        pinfo = *pit;
        kdDebug() << "\t\t" << pinfo->name() << " ("
            << QVariant::typeName( pinfo->type() )
            << ")" << endl;
    }
}
```

# Crear una classe dinàmica

---

```
Classes::addClass( "BookExtraInfo", DynamicObject::createInstance, 0 );
ClassInfo *ci = Classes::classInfo( "BookExtraInfo" );
PropertyInfo *p;

p = new PropertyInfo();
p->setName( "notes" );
p->setType( QVariant::String );
ci->addProperty( p );

p = new PropertyInfo();
p->setName( "numberOfCopies" );
p->setType( QVariant::ULongLong );
ci->addProperty( p );

p = new PropertyInfo();
p->setName( "dateOfAcquisition" );
p->setType( QVariant::Date );
ci->addProperty( p );
ci->addObject( "Book", "book", &Book::createInstance );
ci->addCollection( "Book", "similarBooks" );
```



# Futur?

---

- Sistema de consultes
- Millora de la infraestructura per a IGU
- Migració a Qt/KDE 4
- Optimització
- Crear espais de noms/millorar la nomenclatura
- Adaptar bases de dades al llarg del temps

---

Fi

# Més informació

---

- [kandau.berlios.de](http://kandau.berlios.de)
- [albert@wickywicky.net](mailto:albert@wickywicky.net)