



Organizzazione Cartelle Principale

```
BibliotecaMultimediale/
├── src/                                # Codice sorgente (.cpp)
│   ├── Model/                         # Domain Logic (Business Logic)
│   │   ├── MediaItem/                # Gerarchia classi media
│   │   ├── Repository/              # Data Access Layer
│   │   ├── Persistence/              # Serializzazione/Persistenza
│   │   └── Visitor/                  # Pattern Visitor per UI
│   ├── View/                         # Interfaccia Utente (Qt Widgets)
│   │   ├── MainWindow/               # Finestra principale
│   │   ├── CategoryView/            # Vista categorie
│   │   ├── DetailView/              # Vista dettaglio
│   │   └── EditView/                 # Vista modifica/creazione
│   ├── Controller/                  # Logica controllo MVC
│   └── main.cpp                      # Entry point applicazione
├── include/                          # Header files (.h)
│   ├── Model/
│   │   ├── MediaItem/
│   │   ├── Repository/
│   │   ├── Persistence/
│   │   └── Visitor/
│   ├── View/
│   │   ├── MainWindow/
│   │   ├── CategoryView/
│   │   ├── DetailView/
│   │   └── EditView/
│   └── Controller/
├── resources/                        # Assets (icone, immagini, QRC)
│   ├── icons/
│   ├── images/
│   └── resources.qrc
├── data/                             # File dati esempio
│   ├── sample_library.json
│   └── test_data.json
├── docs/                             # Documentazione progetto
│   └── relazione.pdf
├── BibliotecaMultimediale.pro        # File progetto Qt
└── README.md                         # Istruzioni compilazione
```



File Dettagliati per Categoria



MODEL - Domain Logic



MediaItem/ - Gerarchia Classi Core

```
include/Model/MediaItem/
├─ MediaItem.h           # Classe base astratta
├─ Book.h               # Libro
├─ Comic.h              # Fumetto
├─ Movie.h              # Film
├─ Magazine.h           # Rivista
└─ TVSeries.h           # Serie TV

src/Model/MediaItem/
├─ MediaItem.cpp
├─ Book.cpp
├─ Comic.cpp
├─ Movie.cpp
├─ Magazine.cpp
└─ TVSeries.cpp
```



Repository/ - Data Access

```
include/Model/Repository/
├─ MediaRepository.h     # Repository principale
└─ IRepository.h         # Interfaccia base (opzionale)

src/Model/Repository/
└─ MediaRepository.cpp
```



Persistence/ - Serializzazione

```
include/Model/Persistence/
├─ JSONPersistence.h     # Serializzazione JSON
└─ XMLPersistence.h      # Serializzazione XML (per progetti coppia)

src/Model/Persistence/
├─ JSONPersistence.cpp
└─ XMLPersistence.cpp
```



Visitor/ - Pattern per UI Specifica

```
include/Model/Visitor/
├── UIVisitor.h           # Interfaccia base Visitor
├── DetailFormVisitor.h   # Generazione form dettaglio
└── EditFormVisitor.h     # Generazione form modifica

src/Model/Visitor/
├── DetailFormVisitor.cpp
└── EditFormVisitor.cpp
```

VIEW - User Interface

MainWindow/ - Finestra Principale

```
include/View/MainWindow/
├── MainWindow.h          # Finestra principale
└── StatusBar.h           # Barra stato personalizzata

src/View/MainWindow/
├── MainWindow.cpp
└── StatusBar.cpp
```

CategoryView/ - Vista Categorie

```
include/View/CategoryView/
├── CategoryView.h        # Vista lista categoria
├── CategorySidebar.h     # Sidebar categorie
└── SearchWidget.h        # Widget ricerca

src/View/CategoryView/
├── CategoryView.cpp
├── CategorySidebar.cpp
└── SearchWidget.cpp
```

DetailView/ - Vista Dettaglio

```
include/View/DetailView/
├── DetailView.h          # Vista dettaglio readonly
├── BookDetailWidget.h     # Widget dettaglio libro
├── MovieDetailWidget.h    # Widget dettaglio film
├── ComicDetailWidget.h    # Widget dettaglio fumetto
└── MagazineDetailWidget.h # Widget dettaglio rivista
```

```
└─ TVSeriesDetailWidget.h      # Widget dettaglio serie
```

```
src/View/DetailView/
```

```
└─ DetailView.cpp
└─ BookDetailWidget.cpp
└─ MovieDetailWidget.cpp
└─ ComicDetailWidget.cpp
└─ MagazineDetailWidget.cpp
└─ TVSeriesDetailWidget.cpp
```

EditView/ - Vista Modifica

```
include/View/EditView/
```

```
└─ EditView.h                  # Vista modifica/creazione
└─ BookEditWidget.h           # Form modifica libro
└─ MovieEditWidget.h          # Form modifica film
└─ ComicEditWidget.h          # Form modifica fumetto
└─ MagazineEditWidget.h       # Form modifica rivista
└─ TVSeriesEditWidget.h       # Form modifica serie
```

```
src/View/EditView/
```

```
└─ EditView.cpp
└─ BookEditWidget.cpp
└─ MovieEditWidget.cpp
└─ ComicEditWidget.cpp
└─ MagazineEditWidget.cpp
└─ TVSeriesEditWidget.cpp
```

CONTROLLER - Business Logic

```
include/Controller/
```

```
└─ MainController.h           # Controller principale MVC
└─ CategoryController.h       # Gestione categorie
└─ MediaController.h          # Gestione media CRUD
└─ PersistenceController.h     # Gestione salvataggio/caricamento
```

```
src/Controller/
```

```
└─ MainController.cpp
└─ CategoryController.cpp
└─ MediaController.cpp
└─ PersistenceController.cpp
```

File di Configurazione

BibliotecaMultimediale.pro

```
QT += core widgets

CONFIG += c++17

TARGET = BibliotecaMultimediale
TEMPLATE = app

# Directories
INCLUDEPATH += include
VPATH += src

# Headers
HEADERS += \
    include/Model/MediaItem/MediaItem.h \
    include/Model/MediaItem/Book.h \
    include/Model/MediaItem/Comic.h \
    include/Model/MediaItem/Movie.h \
    include/Model/MediaItem/Magazine.h \
    include/Model/MediaItem/TVSeries.h \
    include/Model/Repository/MediaRepository.h \
    include/Model/Persistence/JSONPersistence.h \
    include/Model/Visitor/UIVisitor.h \
    include/Model/Visitor/DetailFormVisitor.h \
    include/Model/Visitor/EditFormVisitor.h \
    include/View/MainWindow/MainWindow.h \
    include/View/CategoryView/CategoryView.h \
    include/View/DetailView/DetailView.h \
    include/View/EditView/EditView.h \
    include/Controller/MainController.h

# Sources
SOURCES += \
    src/main.cpp \
    src/Model/MediaItem/MediaItem.cpp \
    src/Model/MediaItem/Book.cpp \
    src/Model/MediaItem/Comic.cpp \
    src/Model/MediaItem/Movie.cpp \
    src/Model/MediaItem/Magazine.cpp \
    src/Model/MediaItem/TVSeries.cpp \
    src/Model/Repository/MediaRepository.cpp \
    src/Model/Persistence/JSONPersistence.cpp \
    src/Model/Visitor/DetailFormVisitor.cpp \
    src/Model/Visitor/EditFormVisitor.cpp \
    src/View/MainWindow/MainWindow.cpp \
    src/View/CategoryView/CategoryView.cpp \
```

```
src/View/DetailView/DetailView.cpp \
src/View/EditView/EditView.cpp \
src/Controller/MainController.cpp
```

Resources

```
RESOURCES += resources/resources.qrc
```

resources.qrc

```
<!DOCTYPE RCC>
<RCC version="1.0">
  <qresource prefix="/icons">
    <file>icons/book.svg</file>
    <file>icons/movie.svg</file>
    <file>icons/comic.svg</file>
    <file>icons/magazine.svg</file>
    <file>icons/tvseries.svg</file>
    <file>icons/new.svg</file>
    <file>icons/open.svg</file>
    <file>icons/save.svg</file>
    <file>icons/edit.svg</file>
    <file>icons/delete.svg</file>
    <file>icons/view.svg</file>
  </qresource>
  <qresource prefix="/images">
    <file>images/logo.png</file>
    <file>images/placeholder.png</file>
  </qresource>
</RCC>
```

Comandi Setup Iniziale

1. Creare struttura cartelle

```
mkdir -p
```

```
BibliotecaMultimediale/{src/{Model/{MediaItem,Repository,Persistence,Visitor},
View/{MainWindow,CategoryView,DetailView,EditView},Controller},include/{Model/{MediaItem,Repository,Persistence,Visitor},View/{MainWindow,CategoryView,DetailView,EditView},Controller},resources/{icons,images},data,docs}
```

2. Posizionarsi nella directory progetto

```
cd BibliotecaMultimediale
```

3. Generare Makefile

```
qmake
```

4. Compilare

```
make
```

```
# 5. Eseguire
./BibliotecaMultimediale
```

Priorità Implementazione

Fase 1 - Core Model (ALTA PRIORITÀ)

1. `MediaItem.h/cpp` - Classe base
2. `Book.h/cpp` - Prima classe concreta
3. `MediaRepository.h/cpp` - Repository base
4. `JSONPersistence.h/cpp` - Persistenza minima

Fase 2 - UI Base (MEDIA PRIORITÀ)

1. `MainWindow.h/cpp` - Finestra principale
2. `CategoryView.h/cpp` - Vista categorie base
3. Collegamento Model-View tramite Controller

Fase 3 - Completamento (BASSA PRIORITÀ)

1. Altre classi media (Comic, Movie, Magazine, TVSeries)
2. Visitor pattern per UI specifica
3. `DetailView` e `EditView` complete
4. UI polish e testing



Note Importanti

- **Separazione netta:** Model completamente indipendente da Qt GUI
- **Visitor pattern:** Solo per generazione UI specifica per tipo
- **Repository:** Centralizza accesso dati, facilita testing
- **Controller:** Mediazione pulita tra Model e View
- **Modularità:** Ogni componente in file separato per manutenibilità