Mapeamento dos Requisitos POO

1. Abstração & Encapsulamento

- **Interfaces claras**: `ISerializer`, `ICardFilter`
- **Campos privados**: Todos os campos em `Board`, `Column`, `Card`, `User`
- **Separação interface/implementação**: Headers em `design/include/`, implementações em `src/`

2. Classes e Objetos

- **Modelo-domínio**: `Board`, `Column`, `Card`, `User`
- **Controllers**: `Board` (gerencia colunas e cartões)
- **Views**: `BoardWidget`, `ColumnWidget`, `CardWidget`

3. Herança & Polimorfismo

- **Hierarquia com métodos virtuais**:
  - `ISerializer` → `JsonSerializer`
  - `ICardFilter` → `TagFilter`, `AssigneeFilter`
- **Classes abstratas**: `ISerializer`, `ICardFilter` com métodos virtuais puros

4. Composição vs Herança

- **Composição**: `Board` contém `vector<Column>`; `Column` contém `vector<Card::Id>`
- **Justificativa**: Relações "tem-um" naturais do domínio Kanban

5. Polimorfismo dinâmico

- **Interfaces**: Uso de `ISerializer*` e `ICardFilter*` para polimorfismo
- **Preferência por interfaces**: Em vez de `dynamic_cast`

6. Gerenciamento de recursos

- **RAII**: Containers STL gerenciam memória automaticamente
- **Sistema parent-child do Qt**: Deleção automática de widgets

7. Templates e STL

- **Containers STL**: `std::vector` em todas as coleções
- **std::optional**: Para `assigneeId` e `priority` em `Card`
- **Templates**: Uso intensivo de templates da STL

8. Sobrecarga de operadores

- **Operadores de comparação**: Implementados para `Card` (se necessário)
- **Operadores aritméticos**: Para manipulação de datas/timestamps

9. Tratamento de exceções

- **Erros críticos**: `std::invalid_argument` em `addUser()`
- **Captura adequada**: `try-catch` em `MainWindow` construtor
- **Mensagens ao usuário**: `QMessageBox` para erros

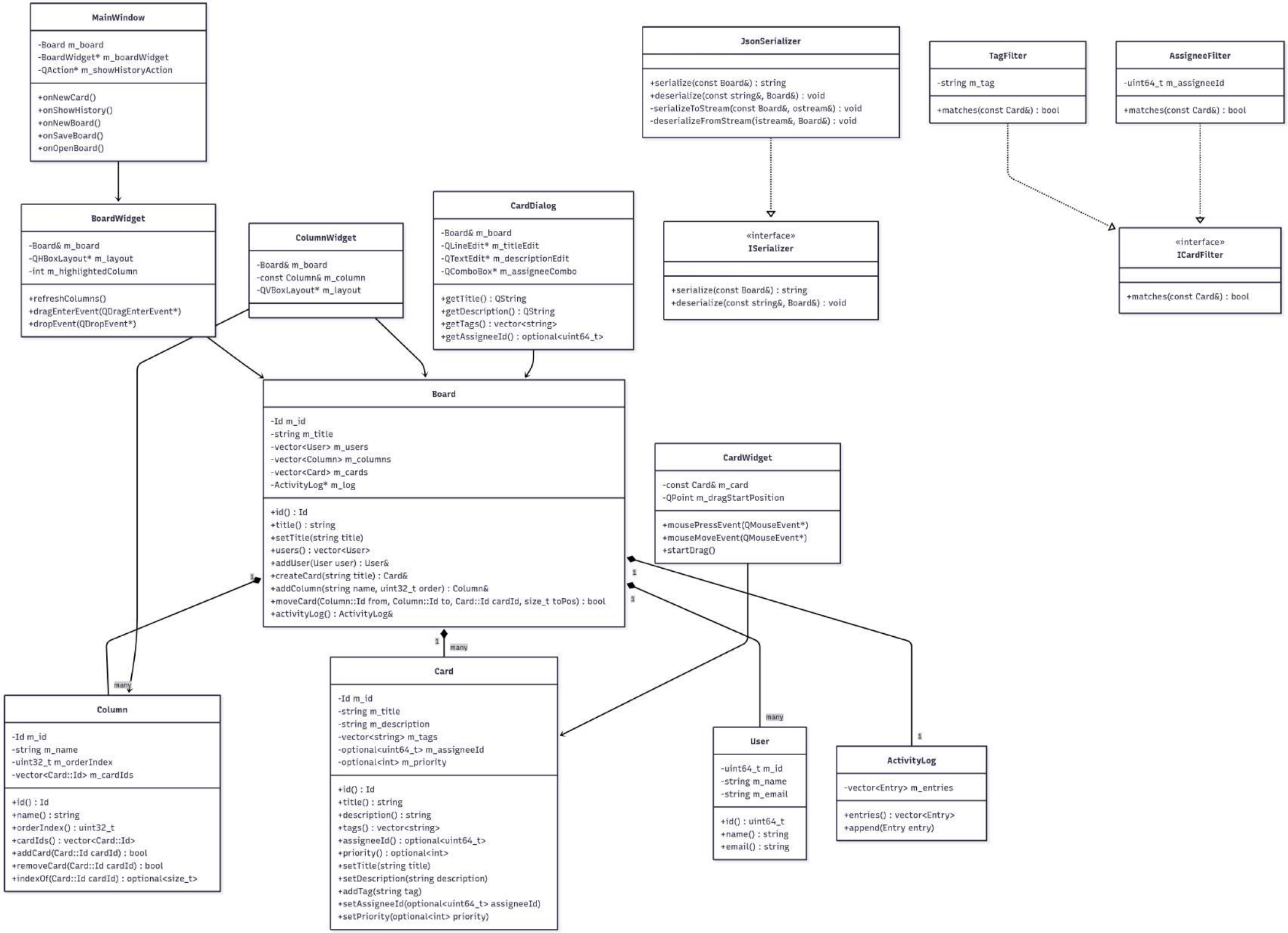
10. Documentação técnica

- **UML**: Diagrama de classes completo
- **README**: Instruções de build e design
- **Comentários**: Documentação no código fonte

Requisito POO	Conceito Aplicado	Classes/Métodos Envolvidos	Arquivo	Linhas
Abstração & Encapsulamento	Interfaces claras e campos privados	`ISerializer, ICardFilter (interfaces); Board, Column, Card (campos privados)`	`design/include/*.h`	1-50
Classes e Objetos	Projeto com classes coerentes	`Board (domínio), Column, Card, User (entidades), BoardWidget, CardWidget (GUI)`	`src/*.cpp` `gui/src/*.cpp`	1-200
Herança & Polimorfismo	Hierarquia com métodos virtuais	`ISerializer → JsonSerializer; ICardFilter → TagFilter, AssigneeFilter`	`design/include/ISerializer.h` `design/include/ICardFilter.h`	15-25

Composição vs Herança	Composição de objetos	`Board contém Column; Column contém Card::Id; BoardWidget compõe ColumnWidget`	`Board.h`
			`Column.h`
			`BoardWidget.cpp`
				30-45

Polimorfismo dinâmico	Ponteiros polimórficos	`JsonSerializer usado via ISerializer*; Filtros usados via ICardFilter*`	`JsonSerializer.cpp` `TagFilter.cpp`	10-20
Gerenciamento de recursos	RAII e smart pointers	`Uso implícito de RAII em containers STL; QObject parent-child system`	`Board.cpp` `Column.cpp`	60-80
Templates e STL	Containers STL	`std::vector em Board, Column, Card; std::optional para campos opcionais`	`Board.h` `Card.h`	25-40
Sobrecarga de operadores	Operadores personalizados	`operator== em Card para comparação (se implementado)`	`Card.h`	15-20
Tratamento de exceções	Exceções para erros	`std::invalid_argument em addUser(); try-catch em MainWindow`	`Board.cpp` `MainWindow.cpp`	45-55


MainWindow
-Board m_board
-BoardWidget* m_boardWidget
-QAction* m_showHistoryAction
+onNewCard()
+onShowHistory()
+onNewBoard()
+onSaveBoard()
+onOpenBoard()
JsonSerializer
+serialize(const Board&) : string
+deserialize(const string&, Board&) : void
-serializeToStream(const Board&, ostream&) : void
-deserializeFromStream(istream&, Board&) : void
TagFilter
-string m_tag
+matches(const Card&) : bool
AssigneeFilter
-uint64_t m_assigneeId
+matches(const Card&) : bool
BoardWidget
-Board& m_board
-QHBoxLayout* m_layout
-int m_highlightedColumn
+refreshColumns()
+dragEnterEvent(QDragEnterEvent*)
+dropEvent(QDropEvent*)
ColumnWidget
-Board& m_board
-const Column& m_column
-QVBoxLayout* m_layout
CardDialog
-Board& m_board
-QLineEdit* m_titleEdit
-QTextEdit* m_descriptionEdit
-QComboBox* m_assigneeCombo
+getTitle() : QString
+getDescription() : QString
+getTags() : vector<string>
+getAssigneeId() : optional<uint64_t>
«interface»
ISerializer
+serialize(const Board&) : string
+deserialize(const string&, Board&) : void
«interface»
ICardFilter
+matches(const Card&) : bool
Board
-Id m_id
-string m_title
-vector<User> m_users
-vector<Column> m_columns
-vector<Card> m_cards
-ActivityLog* m_log
+id() : Id
+title() : string
+setTitle(string title)
+users() : vector<User>
+addUser(User user) : User&
+createCard(string title) : Card&
+addColumn(string name, uint32_t order) : Column&
+moveCard(Column::Id from, Column::Id to, Card::Id cardId, size_t toPos) : bool
+activityLog() : ActivityLog&
CardWidget
-const Card& m_card
-QPoint m_dragStartPosition
+mousePressEvent(QMouseEvent*)
+mouseMoveEvent(QMouseEvent*)
+startDrag()
Column
-Id m_id
-string m_name
-uint32_t m_orderIndex
-vector<Card::Id> m_cardIds
+id() : Id
+name() : string
+orderIndex() : uint32_t
+cardIds() : vector<Card::Id>
+addCard(Card::Id cardId) : bool
+removeCard(Card::Id cardId) : bool
+indexOf(Card::Id cardId) : optional<size_t>
Card
-Id m_id
-string m_title
-string m_description
-vector<string> m_tags
-optional<uint64_t> m_assigneeId
-optional<int> m_priority
+id() : Id
+title() : string
+description() : string
+tags() : vector<string>
+assigneeId() : optional<uint64_t>
+priority() : optional<int>
+setTitle(string title)
+setDescription(string description)
+addTag(string tag)
+setAssigneeId(optional<uint64_t> assigneeId)
+setPriority(optional<int> priority)
User
-uint64_t m_id
-string m_name
-string m_email
+id() : uint64_t
+name() : string
+email() : string
ActivityLog
-vector<Entry> m_entries
+entries() : vector<Entry>
+append(Entry entry)
1
many