

1. Diagrama de Classes

Baseado no código, o diagrama de classes seria assim:

Classes envolvidas:

1. CadastroTela

- Atributos:
 - usuarioField: JTextField
 - senhaField: JPasswordField
 - funcaoField: JTextField
 - setorField: JTextField
 - cadastrarButton: JButton
- Métodos:
 - CadastroTela()
 - cadastrarUsuario()
 - showUserError()
 - showUserInfo()

2. DatabaseConnection

- Atributos:
 - URL: String
 - USER: String
 - PASSWORD: String
- Métodos:
 - getConnection(): Connection

3. LoginTela

- Atributos:
 - usuarioField: JTextField
 - senhaField: JPasswordField
 - loginButton: JButton
 - cadastroButton: JButton
- Métodos:
 - LoginTela()
 - fazerLogin()

- abrirCadastroTela()
- openTodoListScreen()
- showUserError()
- showUserInfo()

4. Main

- Métodos:
 - main(String[] args)

5. Task

- Atributos:
 - id: int
 - description: String
 - done: boolean
- Métodos:
 - Task(int id, String description, boolean done)
 - Task(String description)
 - getId()
 - setId(int id)
 - getDescription()
 - setDescription(String description)
 - isDone()
 - setDone(boolean done)

6. TodoList

- Atributos:
 - mainPanel: JPanel
 - taskInputField: JTextField
 - addButton: JButton
 - taskList: JList<String>
 - listModel: DefaultListModel<String>
 - deleteButton: JButton
 - markDoneButton: JButton
 - filterComboBox: JComboBox<String>

- clearCompletedButton: JButton
- clearAll: JButton
- tasks: List<Task>
- Métodos:
 - TodoList()
 - addTask()
 - addTaskToDatabase(String description)
 - loadTasksFromDatabase()
 - deleteTask()
 - deleteTaskFromDatabase(int taskId)
 - markTaskDone()
 - updateTaskStatusInDatabase(int taskId, boolean done)
 - filterTasks()
 - clearCompletedTasks()
 - clearAll()
 - updateTaskList()
 - showUserError()

Relações:

- **CadastroTela** tem um método que interage com **DatabaseConnection** para inserir dados no banco.
- **LoginTela** tem um método que interage com **DatabaseConnection** para validar credenciais.
- **TodoList** tem um relacionamento de "usa" com **Task**, pois manipula tarefas.
- **Task** contém informações de tarefas individuais, e o **TodoList** gerencia essas tarefas.
- **TodoList** também utiliza **DatabaseConnection** para adicionar, excluir e carregar tarefas do banco.

```

+-----+
|  CadastroTela  |
+-----+
| - usuarioField |
| - senhaField   |
| - funcaoField  |
| - setorField   |
| - cadastrarButton |
+-----+
| + CadastroTela() |
| + cadastrarUsuario() |
| + showUserError() |
| + showUserInfo() |
+-----+

```

```

|
| uses
v

```

```

+-----+
|  LoginTela  |
+-----+
| - usuarioField |
| - senhaField   |
| - loginButton  |
| - cadastroButton |
+-----+
| + LoginTela() |
| + fazerLogin() |
| + abrirCadastroTela() |
| + openTodoListScreen() |
| + showUserError() |
| + showUserInfo() |
+-----+

```

```

+-----+
| DatabaseConnection |
+-----+
| - URL               |
| - USER              |
| - PASSWORD          |
+-----+
| + getConnection()  |
+-----+

```

```

+-----+
|  Task  |
+-----+
| - id    |
| - description |
| - done  |
+-----+
| + Task(int id, String description, boolean done) |
| + Task(String description) |
| + getId() |
| + setId() |
| + getDescription() |
| + setDescription() |
| + isDone() |
| + setDone() |
+-----+

```

```

|
| uses
v

```

```

+-----+
|  TodoList  |
+-----+
| - mainPanel |
| - taskInputField |
| - addButton |
| - taskList  |
| - listModel |
| - deleteButton |
| - markDoneButton |
| - filterComboBox |
| - clearCompletedButton |
| - clearAll |
| - tasks |
+-----+
| + TodoList() |
| + addTask() |
| + addTaskToDatabase() |
| + loadTasksFromDatabase() |
| + deleteTask() |
| + deleteTaskFromDatabase() |
| + markTaskDone() |
| + updateTaskStatusInDatabase() |
| + filterTasks() |
| + clearCompletedTasks() |
| + clearAll() |
| + updateTaskList() |
| + showUserError() |
+-----+

```

2. Diagrama de Fluxo (Flowchart)

O **fluxo de operações principais** seria algo assim:

Fluxo de Cadastro de Usuário (CadastroTela)

1. **Início**
2. Usuário preenche os campos (usuário, senha, função, setor)
3. **Validar campos:**
 - Se algum campo estiver vazio, exibe erro e volta para o preenchimento.
 - Se os campos estão completos, **salva no banco** usando DatabaseConnection.
4. Exibe mensagem de sucesso.
5. **Fecha a tela de cadastro e abre o login.**
6. **Fim.**

Fluxo de Login (LoginTela)

1. **Início**
2. Usuário preenche os campos (usuário, senha)
3. **Validar credenciais** no banco usando DatabaseConnection.
 - Se válido, exibe "Login bem-sucedido" e abre a tela de tarefas.
 - Se inválido, exibe erro.
4. **Fim.**

Fluxo de Gerenciamento de Tarefas (ToDoList)

1. **Início**
2. Usuário digita uma tarefa e clica em "Adicionar".
 - A tarefa é adicionada ao banco e à lista.
3. **Excluir Tarefa:**
 - Usuário seleciona a tarefa e clica em "Excluir".
 - A tarefa é removida do banco e da lista.
4. **Marcar como Concluída:**
 - Usuário seleciona a tarefa e clica em "Concluir".
 - A tarefa é atualizada no banco e a lista é atualizada.
5. **Filtrar Tarefas:**
 - Usuário escolhe um filtro (todas, ativas, concluídas).
 - A lista é atualizada conforme o filtro.

6. Limpar Tarefas Concluídas:

- Usuário clica em "Limpar Concluídas".
- Exclui as tarefas concluídas.

7. Limpar Todas as Tarefas:

- Usuário clica em "Limpar Tarefas".
- Exclui todas as tarefas.

8. Fim.

Esses fluxos representam o comportamento básico da aplicação.