 **Classe ServerLider**

* **Descrição**: Representa o servidor que atua como o "Líder" do grupo. Ele é responsável por receber comandos SQL do cliente e distribuir esses comandos para todos os membros.
* **Atributos**:
  + List<Member> members - Lista de membros conectados ao grupo.
  + int leaderId - ID do líder do grupo.
* **Métodos**:
  + void addMember(Member member) - Adiciona um novo membro ao grupo.
  + void removeMember(Member member) - Remove um membro do grupo.
  + void receiveCommand(String sqlCommand) - Recebe um comando SQL do cliente.
  + void distributeCommand(String sqlCommand) - Distribui o comando SQL para todos os membros do grupo.
  + void detectFaults() - Verifica se há falhas nos membros do grupo e remove membros defeituosos.

 **Classe Member**

* **Descrição**: Representa um membro do grupo que está conectado ao "Líder". Cada membro tem uma base de dados associada onde executa comandos SQL.
* **Atributos**:
  + int memberId - ID do membro.
  + DatabaseConnection dbConnection - Conexão com a base de dados associada ao membro.
* **Métodos**:
  + void receiveCommand(String sqlCommand) - Recebe um comando SQL do Líder.
  + void executeCommand(String sqlCommand) - Executa o comando SQL na base de dados.
  + boolean isAlive() - Verifica se o membro está ativo.

 **Classe ClienteSQL**

* **Descrição**: Representa a interface de cliente SQL usada para enviar comandos SQL ao "Líder".
* **Atributos**:
  + ServerLider leader - Referência ao servidor "Líder".
* **Métodos**:
  + void sendCommand(String sqlCommand) - Envia um comando SQL ao "Líder".
  + void displayResult(String result) - Exibe o resultado do comando executado.

 **Interface LiderInterface**

* **Descrição**: Interface RMI para a comunicação entre o cliente SQL e o "Líder".
* **Métodos**:
  + void receiveCommand(String sqlCommand) - Método remoto para o cliente enviar comandos ao "Líder".
  + void addMember(MemberInterface member) - Método remoto para adicionar membros ao grupo.

 **Interface MemberInterface**

* **Descrição**: Interface RMI para a comunicação entre o "Líder" e os membros.
* **Métodos**:
  + void receiveCommand(String sqlCommand) - Método remoto para o "Líder" enviar comandos para os membros.
  + boolean isAlive() - Método remoto para o "Líder" verificar se o membro está ativo.

 **Classe DatabaseConnection**

* **Descrição**: Representa a conexão de um membro com seu banco de dados.
* **Atributos**:
  + Connection connection - Objeto de conexão com o banco de dados.
* **Métodos**:
  + void executeSQL(String sqlCommand) - Executa o comando SQL na base de dados.
  + void connect() - Conecta-se ao banco de dados.
  + void disconnect() - Desconecta-se do banco de dados.