

# Relatório de Explicação

**Alunos:** Levy Santiago, Tulio Campos, Adson Cardoso

## 1. Uso da Aplicação

A primeira janela da aplicação é a janela de controle. O único botão habilitado é o botão para ligar a antena do leitor RFID. Se não for possível realizar a conexão com o banco de dados, o leitor RFID não é ligado. Ao realizar a conexão com o banco, a antena RFID é ligada, e todas as vezes que o leitor lê uma tag, uma consulta é feita no BD. Caso o dispositivo não esteja conectado ao computador, a conexão com o banco é finalizada e um aviso é dado. Considerando que este esteja conectado, se o código lido estiver cadastrado, os campos de nome, sobrenome são preenchidos e se o cliente estiver com o pagamento em dias, sua entrada é permitida.

Caso a tag não esteja cadastrada, o botão “Cadastrar” é habilitado para que um novo cliente possa ser cadastrado. Ao clicar no botão, a janela de Cadastro é aberta. Nela, o campo da Tag já é preenchido automaticamente com o código que foi lido, basta informar o resto dos dados do cliente. Os serviços e formas de pagamentos que aparecem para escolher, são todos obtidos do BD. Para saber o valor de cada ou o valor total de mais de um, basta selecionar uma certa quantidade (apertando CTRL) e clicar no botão “Valor”. Ao finalizar o cadastro, basta clicar em “Cadastrar”.

Caso a tag esteja cadastrada, mas o cliente que tem essa tag está com o pagamento vencido, então o status mostra que sua entrada não é permitida. Com isso, o botão “Pagamento” é habilitado para que o cliente realize o pagamento do mês atual. Ao clicar no botão, a janela para pagamento é aberta, já com o nome, sobrenome, código (RFID) do cliente, o valor total do contrato e as formas de pagamento a serem escolhidas. Ao finalizar a escolha da forma de pagamento e clicar em pago, o sistema atualiza a data do próximo pagamento (próximo mês) do cliente.

## 2. Identificando utilizações de funcionalidades vistas no semestre

No evento de leitura da tag RFID (tagGained), na interface gráfica GUI\_Principal, são requisitados (SELECT) alguns dados para o banco de dados, dentre estes, é solicitada a data do servidor para verificar se o contrato do cliente está vencido, se sim, é feito um UPDATE na tabela cliente setando o cliente como inativo.

Na interface gráfica Cadastro, no evento do botão cadastrar (btn\_cadastrarActionPerformed) é feito um INSERT do cliente no banco, nesse método, os códigos sql estão todos dentro do método “inserirCliente” da classe Conector. No evento do botão valor (btn\_valorActionPerformed), ainda em Cadastro, é requisitado ao banco de dados que faça uma soma dos valores dos serviços selecionados a partir de um SELECT. Também existe uma função ‘inicializarCampos’ que realiza alguns SELECTS para inicializar os campos “serviços” e “forma de pagamento”.

Na interface gráfica Pagamento, no evento “btn\_pagoActionPerformed” é feito um SELECT para atualizar o campo forma de pagamento e duas atualizações para setar o cliente como ativo e atualizar a data do próximo pagamento. Em “inicializarCampos” é usado um método “getValorContrato” que faz um SELECT usando JOIN retornando a soma

(soma feita no banco) dos valores dos serviços cadastrados no contrato. Neste mesmo método é realizada uma consulta das formas de pagamento cadastradas no banco.

A classe Conector é onde estão todos métodos usados para realizar consultas e atualizações no banco.

### **3. Modo de conexão com o Banco de Dados**

A conexão com o BD foi feita utilizando o JDBC. Como o Banco de Dados utilizado foi o Postgres, então o driver usado foi o "org.postgresql.Driver". Usando o DriverManager.getConnection é possível estabelecer a conexão com o banco, passando como parâmetro a URL do banco, que no caso foi "jdbc:postgresql://localhost:5432/academiaDB", o usuário e a senha para acesso ao banco. Esta conexão é iniciada a partir do momento em que o leitor RFID é ligado, quando este é desligado, a conexão é finalizada.

No momento em que é feito um UPDATE ou INSERT, por exemplo, é usada uma classe chamada PreparedStatement, ela deixa o código preparado para ser executado, e depois é chamado o método stmt.executeUpdate para executar o código e atualizar o banco. Quando é feito um SELECT, o PreparedStatement também é usado, mas é necessário usar uma classe chamada ResultSet, ela é quem controla o cursor para manipular os dados da tabela resultante. Para acessar as linhas da tabela resultante da consulta, é usado o método "next" dessa classe. No momento em que a tabela é gerada, o cursor começa apontando para o início dela, ou seja, antes da primeira linha. Por isso, para acessar a primeira linha, primeiro é preciso executar o método "next" uma vez, fazer o que tem que ser feito com a primeira linha, depois executar novamente para que o cursor aponte para a próxima linha.

Para obter o valor de uma tupla da tabela recebida, é usado o método getString da classe ResultSet. Este método retorna o valor da coluna na linha que o cursor está apontando, ela pode receber como parâmetro o nome da coluna ou o seu índice (sendo que o índice da primeira coluna é 1).