

Tabelas banco de dados PI

```
create database estacionamento;
```

```
use estacionamento;
```

```
create table Funcionario(  
id INT PRIMARY KEY AUTO_INCREMENT,  
nome VARCHAR (50) NOT NULL,  
usuario VARCHAR(25) NOT NULL,  
senha varchar(12) not null  
);  
  
create table Reservas(  
id INT PRIMARY KEY AUTO_INCREMENT,  
id_funcionario INT,  
data_reserva DATETIME NOT NULL,  
vaga int not null,  
foreign key (id_funcionario) references Funcionario(id)  
);  
  
create table Vagas(  
    id_vaga int primary key,  
    disponível boolean not null
```

);

```
create table Clientes(  
id int primary key auto_increment,  
nome varchar(80) not null,  
documento varchar(11) not null,  
endereco varchar(150) not null,  
placa varchar(20) not null  
);
```

```
create table Transacoes (  
id int primary key auto_increment,  
id_cliente int ,  
data_transacao datetime not null,  
valor decimal(10,2) not null,  
descricao varchar(250),  
foreign key (id_cliente) references Clientes(id)  
  
);
```

Metodo Verificação BD

```
private void verificarDadosBD()
{
    using (MySqlCommand cmd = new MySqlCommand())
    {
        cmd.Connection = mConn;

        cmd.CommandText = "SHOW DATABASES LIKE
'estacionamento";
        if(cmd.ExecuteScalar() == null)
        {
            cmd.CommandText = "CREATE DATABASE
estacionamento;";
            cmd.ExecuteNonQuery();

            cmd.CommandText = @"CREATE TABLE Funcionario(
                id INT PRIMARY KEY AUTO_INCREMENT,
                nome VARCHAR (50) NOT NULL,
                usuario VARCHAR(25) NOT NULL,
                senha varchar(12) not null
            );";
            cmd.ExecuteNonQuery();
        }
    }
}
```

```
cmd.CommandText = @"INSERT INTO Funcionario
(nome,usuario,senha) VALUES (@nome,@usuario,@senha)";
cmd.Parameters.AddWithValue("@nome", "admin");
cmd.Parameters.AddWithValue("@usuario", "admin");
cmd.Parameters.AddWithValue("senha", "admin");
cmd.ExecuteNonQuery();
```

```
cmd.CommandText = @"create table Reservas(
    id INT PRIMARY KEY AUTO_INCREMENT,
    id_funcionario INT,
    data_reserva DATETIME NOT NULL,
    vaga int not null,
    foreign key (id_funcionario) references
Funcionario(id)
);";
```

```
cmd.ExecuteNonQuery();
```

```
cmd.CommandText = @"create table Vagas(
    id_vaga int primary key,
    disponível boolean not null
);";
```

```
cmd.ExecuteNonQuery();
```

```
cmd.CommandText = @"create table Clientes(
```

```
id int primary key auto_increment,  
nome varchar(80) not null,  
documento varchar(11) not null,  
endereco varchar(150) not null,  
placa varchar(20) not null  
);";
```

```
cmd.ExecuteNonQuery();
```

```
cmd.CommandText = @"create table Transacoes (  
    id int primary key auto_increment,  
    id_cliente int ,  
    data_transacao datetime not null,  
    valor decimal(10,2) not null,  
    descricao varchar(250),  
    foreign key (id_cliente) references
```

```
Clientes(id)
```

```
);";
```

```
cmd.ExecuteNonQuery();
```

```
} else
```

```
{
```

```
    MessageBox.Show("Conexão bem Sucedida!");
```

}

}

}