

UNIVERSIDADE DO VALE DO ITAJAÍ (UNIVALI)  
CURSO DE ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

GARAGEM S.A. - HANDS ON WORK III

Andrieli Luana Massarolo, Célio Parischi Junior, Guilherme Amaral Funk,  
Mateus João Feldhaus, Miquelane Pereira Melo, Pedro Barros Marzano.

FLORIANÓPOLIS  
2020

## **RESUMO**

A Garagem S.A. é uma empresa que atua no ramo de estacionamento de veículos. Para melhorar a qualidade do atendimento, agilidade e controle do funcionamento, um sistema de software foi imaginado e criado. Para isso, houve a necessidade da identificação das Regras de Negócio, além da elicitação e documentação dos Requisitos Funcionais e Requisitos Não Funcionais. Com a validação dos requisitos, a equipe de desenvolvimento pôde iniciar a implementação e desenvolvimento do protótipo, o qual encontra-se em fase de desenvolvimento.

Palavras-chave: Software, requisitos, regras, validação, documentação, desenvolvimento.

## LISTA DE ILUSTRAÇÕES

Figura 01 Código em desenvolvimento.....	26
Figura 02 Código em desenvolvimento.....	26
Figura 03 Protótipo de interface. ....	27
Figura 04 Banco de Dados .....	27

## LISTA DE TABELAS

Tabela 01 Regras de Negócio.....	12
Tabela 02 Requisitos Funcionais.....	16
Tabela 03 Requisitos Não Funcionais .....	23

## SUMÁRIO

<b>1. INTRODUÇÃO .....</b>	<b>6</b>
<b>2. DESENVOLVIMENTO .....</b>	<b>6</b>
2.1. REGRAS DE NEGÓCIO DA GARAGEM S.A. ....	7
2.2. REQUISITOS FUNCIONAIS (RF) .....	10
2.3. REQUISITOS NÃO FUNCIONAIS (RNF) .....	11
<b>3. DOCUMENTAÇÃO DAS REGRAS DE NEGÓCIO, REQUISITOS FUNCIONAIS E REQUISITOS NÃO FUNCIONAIS .....</b>	<b>12</b>
3.1 . REGRAS DE NEGÓCIO .....	12
3.2. REQUISITOS FUNCIONAIS.....	16
3.3. REQUISITOS NÃO FUNCIONAIS (RNF) .....	23
<b>4. IMAGENS.....</b>	<b>26</b>
<b>5. CRONOGRAMA .....</b>	<b>28</b>
<b>6. REFERÊNCIAS .....</b>	<b>29</b>

## 1. INTRODUÇÃO

O documento em questão traz o desenvolvimento de uma aplicação em C#, utilizando Windows Forms e persistência de dados com SQL. Bem como as regras de negócio, requisitos funcionais e não funcionais empregando o conhecimento adquirido na matéria de Engenharia de Requisitos e Programming and Data Persistence.

## 2. DESENVOLVIMENTO

O cliente Proprietário da Garagem S.A. procurou a equipe de desenvolvimento para a criação de uma aplicação para seu negócio onde pudesse facilitar o seu dia a dia. A Garagem S.A. trata-se de um estacionamento localizado no centro da cidade e o objetivo do cliente é ter uma aplicação que lhe permita ter o controle de entrada e saída de veículos, bem como a emissão de ticket de estacionamento e o cálculo dos valores em relação às horas que o cliente utilizou o estacionamento.

Para a realização dessa aplicação são necessárias várias etapas, incluindo entrevistas e reuniões. A partir de algumas reuniões e entrevistas realizadas com os clientes, os analistas responsáveis coletaram várias informações, como a descrição de como o cliente gostaria que o sistema funcionasse e a descrição de seu negócio, com a listagem de regras e restrições.



A definição das Regras de Negócio é uma etapa fundamental. Com ela é possível conhecer como o negócio funciona e identificar vários requisitos que devem ser implementados. A partir disso, surge a necessidade de desenvolver, elicitar, analisar e especificar os Requisitos de Software. A especificação de Requisitos de Software, conhecida pela sigla SRS (Software Requirements

Specification) é um elemento desenvolvido para se ter a descrição detalhada de todos os aspectos do software a ser desenvolvido (PRESSMAN; MAXIM, 2016).

Antes da implementação do software, há ainda a etapa de validação dos requisitos. Na qual são feitas reuniões com todos os envolvidos no projeto. Essa é uma etapa muito importante, pois é essencial que a tática de validação seja efetiva. Devem participar dessa etapa pessoas estratégicas para garantir que o software que será desenvolvido contém os requisitos certos.

A identificação das Regras de Negócio e elicitação dos Requisitos Funcionais e Não Funcionais fazem parte de todo esse processo, e com isso, é fundamental que haja a documentação desses elementos. Para tanto, a seguir, os mesmos estão detalhados e documentados.

## 2.1. REGRAS DE NEGÓCIO DA GARAGEM S.A.

**Regra de negócio 0001:** A garagem funciona de Segunda a Sábado entre as 06:00 e 22:00 horas.

**Regra de negócio 0002:** Um funcionário pode adicionar e modificar as informações de um veículo que entra. O preenchimento de todas as caixas é obrigatório.

**Regra de negócio 0003:** A hora de saída deve estar entre o horário de funcionamento.

**Regra de negócio 0004:** A garagem aceita o estacionamento de motos e carros. A entrada de veículos de transporte coletivo (mini-vans) é proibida.

**Regra de negócio 0005:** A quantidade de horas a serem cobradas é sempre de hora cheia, e o valor de cada hora é de R\$ 4,00 para motos e R\$ 5,00 para carros.

**Regra de negócio 0006:** Ao adicionar um veículo ao sistema, um comprovante com todas as informações referentes ao mesmo deve ser impresso e entregue ao motorista do veículo.

**Regra de negócio 0007:** Um veículo só pode ser retirado mediante apresentação do recibo de entrada.

**Regra de negócio 0008:** Quando a garagem estiver com pelo menos 80% de capacidade, o motorista deve deixar a chave com o funcionário da garagem.

**Regra de negócio 0009:** Quando a garagem atingir 100% de sua capacidade, mais nenhum veículo pode ser aceito até que a capacidade seja menor que 100%.

**Regra de negócio 0010:** O cliente pode pagar o estacionamento em dinheiro (espécie) ou com cartões de crédito e débito diretamente no guichê do estacionamento.

**Regra de negócio 0011:** Quando o funcionário do estacionamento adiciona a hora de saída de um veículo no sistema, um ticket com os dados do veículo e o valor do serviço é impresso, e este deve ser entregue ao motorista do veículo.



Após a entrevista com o cliente e entendidas as regras do negócio do mesmo, os analistas responsáveis começaram a desenvolver a documentação com os requisitos funcionais e não funcionais na qual a aplicação deve ter, que são elas:

## 2.2. REQUISITOS FUNCIONAIS (RF)

**Requisito Funcional 001:** O sistema deve permitir que o funcionário grave o nome do proprietário do veículo, placa do veículo e horário de entrada (são campos obrigatórios (not null)). Automaticamente um comprovante de entrada deve ser impresso.

**Requisito Funcional 002:** O sistema deve permitir que o funcionário grave a situação do veículo.

**Requisito Funcional 003:** O sistema não pode gravar entradas após as 21h00min.

**Requisito Funcional 004:** O sistema deve emitir um alerta de que o estacionamento chegou a 80% de ocupação (16 veículos em situação de entrada e sem registro de saída).

**Requisito Funcional 005:** O sistema deve gravar se o proprietário do veículo deixou as chaves

**Requisito Funcional 006:** O sistema deve emitir um alerta quando houver atingido 100% da capacidade e impedir que novas entradas sejam registradas (20 veículos em situação de entrada e sem registro de saída).

**Requisito Funcional 007:** O sistema deve permitir que o horário de saída seja gravado e automaticamente deve calcular o preço (em Reais), baseando-se no horário de entrada. Um recibo com os dados e valor total a ser pago deve ser impresso.

**Requisito Funcional 008:** O sistema deve possuir um campo para cadastrar o tipo de veículo que deu entrada e lançar o preço automaticamente de acordo com o campo selecionado. Ao selecionar o tipo “carro”, deve lançar o preço de R\$ 5,00/hora, e ao lançar o tipo “moto”, deve lançar o preço de R\$ 4,00/hora.

### 2.3. REQUISITOS NÃO FUNCIONAIS (RNF)

**Requisito Não Funcional 001:** O sistema deverá prover recursos para processamento de dados de entrada de no máximo 5 segundos.

**Requisito Não Funcional 002:** O sistema deverá prover recursos para processamento de dados de saída de no máximo 5 segundos.

**Requisito Não Funcional 003:** Todos os registros de clientes cadastrados no sistema poderão ser mantidos (alterados/consultados/excluídos) com o sistema off-line e novos registros de clientes (inclusão) poderão ser incluídos também com o sistema off-line

**Requisito Não Funcional 004:** O Sistema Deve Priorizar o Ambiente de Execução em 64 Bits mas também deve ser compatível com 32 Bits

**Requisito Não Funcional 005:** O Sistema deverá ser executado em Sistema Operacional Windows em versões a partir do XP (XP, Vista 7 e 8), considerando sempre o Service Pack mais recente. O sistema, deverá rodar nos sistemas operacionais elencados neste requisito. O comportamento deve ser o mesmo, tanto no que se refere às funcionalidades quanto à instalação.

### 3. DOCUMENTAÇÃO DAS REGRAS DE NEGÓCIO, REQUISITOS FUNCIONAIS E REQUISITOS NÃO FUNCIONAIS

#### 3.1 REGRAS DE NEGÓCIO

Tabela 01 Regras de Negócio

Identificação RN0001	
Nome:	Horário de funcionamento do estabelecimento.
Módulo:	
Data de criação:	04.09.2020
Autor:	Feldhaus, Mateus J.
Versão:	1.0
Dependência:	RF003.
Descrição:	A garagem funciona de Segunda a Sábado entre 06:00 e 22:00.
Tipo:	Fatos.

Identificação RN0002	
Nome:	Adicionar veículo ao Sistema.
Módulo:	
Data de criação:	04.09.2020
Autor:	Feldhaus, Mateus J.
Versão:	1.0
Dependência:	RF001, RF002.
Descrição:	Um funcionário pode adicionar e modificar as informações de um veículo que entra. O preenchimento de todas as caixas é obrigatório.
Tipo:	Termos.

Identificação RN0003	
Nome:	Horário de entrada e de saída.
Módulo:	
Data de criação:	04.09.2020
Autor:	Feldhaus, Mateus J.
Versão:	1.0
Dependência:	RF003.
Descrição:	A hora de saída deve estar entre o horário de funcionamento.
Tipo:	Restrição.

Identificação RN0004	
Nome:	Veículos aceitos na garagem.
Módulo:	
Data de criação:	04.09.2020
Autor:	Feldhaus, Mateus J.
Versão:	1.0
Dependência:	RF008.
Descrição:	A garagem aceita o estacionamento de motos e carros. A entrada de veículos de transporte coletivo (mini-vans) é proibida.
Tipo:	Restrição

Identificação RN0005	
Nome:	Quantidade e valor de horas a serem cobradas.
Módulo:	
Data de criação:	04.09.2020
Autor:	Feldhaus, Mateus J.
Versão:	1.0
Dependência:	RF008.
Descrição:	O valor a ser cobrado é sempre de hora cheia, e o valor de cada hora é de R\$ 4,00 para motos e R\$ 5,00 para carros.
Tipo:	Cálculos.

Identificação RN0006	
Nome:	Impressão do comprovante de entrada.
Módulo:	
Data de criação:	04.09.2020
Autor:	Feldhaus, Mateus J.
Versão:	1.0
Dependência:	RF007.
Descrição:	Ao adicionar um veículo ao sistema, um comprovante com todas as informações referentes ao mesmo deve ser impresso e entregue ao motorista do veículo.
Tipo:	Termos.

Identificação RN0007	
Nome:	Retirada de veículo.
Módulo:	
Data de criação:	04.09.2020
Autor:	Feldhaus, Mateus J.
Versão:	1.0
Dependência:	RF007.
Descrição:	Um veículo só pode ser retirado mediante apresentação do recibo de entrada.
Tipo:	Restrição.

Identificação RN0008	
Nome:	Entrega da chave do veículo.
Módulo:	
Data de criação:	04.09.2020
Autor:	Feldhaus, Mateus J.
Versão:	1.0
Dependência:	RF004, RF005.
Descrição:	Quando a garagem estiver com pelo menos 80% da capacidade, o motorista deve deixar a chave com o funcionário da garagem.
Tipo:	Fato.

Identificação RN0009	
Nome:	Garagem lotada.
Módulo:	
Data de criação:	04.09.2020
Autor:	Feldhaus, Mateus J.
Versão:	1.0
Dependência:	RF006.
Descrição:	Quando a garagem atingir 100% de sua capacidade, mais nenhum veículo pode ser aceito até que a capacidade seja menor que 100%.
Tipo:	Restrição.

Identificação RN0010	
Nome:	Formas de pagamento.
Módulo:	
Data de criação:	04.09.2020
Autor:	Feldhaus, Mateus J.
Versão:	1.0
Dependência:	
Descrição:	O cliente pode pagar o estacionamento em dinheiro (espécie) ou com cartões de crédito e débito diretamente no guichê do estacionamento.
Tipo:	Fato.

Identificação RN0011	
Nome:	Ticket de saída.
Módulo:	
Data de criação:	04.09.2020
Autor:	Feldhaus, Mateus J.
Versão:	1.0
Dependência:	RF007.
Descrição:	Quando o funcionário do estacionamento adiciona a hora de saída de um veículo no sistema, um ticket com os dados do veículo e o valor do serviço é impresso, e este deve ser entregue ao motorista do veículo
Tipo:	Fato.

### 3.2. REQUISITOS FUNCIONAIS

Tabela 02 Requisitos Funcionais

<b>[RF001] Gravar a entrada</b>	
Ator:	Funcionário.
Prioridade:	Essencial.
Entradas e pré-condições:	Nome do proprietário do veículo, placa e registro automático do horário de entrada. Esses campos devem ser de preenchimento obrigatório (not null).
Saídas e pós-condições:	Gravar no banco de dados o nome do proprietário, placa, veículo e horário de entrada. Automaticamente um comprovante de entrada será impresso.
Fluxos de eventos principal:	1 → Solicitar nome completo do proprietário do veículo; 2 → Anotar a placa do veículo; 3 → Anotar modelo do veículo; 4 → Registrar automaticamente o horário; 5 → Imprimir um comprovante de entrada.



<b>[RF002] Gravar a situação do veículo</b>	
Ator:	Funcionário.
Prioridade:	Essencial.
Entradas e pré-condições:	Checagem do veículo.
Saídas e pós-condições:	Gravar no banco de dados as avarias constatadas ou a informação de que o veículo não possui avarias
Fluxos de eventos principal:	<p>1 → Checar o veículo externamente;</p> <p>2 → Gravar que o veículo contém avarias e especificá-las ou anotar que o veículo não possui avarias.</p>

<b>[RF003] Impedir entradas após as 21:00 horas.</b>	
Ator:	Funcionário.
Prioridade:	Essencial.
Entradas e pré-condições:	Veículos que chegarem após as 21:00 horas devem ser impedidos de dar entrada.
Saídas e pós-condições:	Sistema impede que novas entradas sejam registradas após as 21:00 horas.
Fluxos de eventos principal:	<p>1 → Tentar registrar entrada após as 21:00;</p> <p>2 → Sistema impede o registro.</p>

<b>[RF004] Emitir alerta quando atingir 80% de ocupação</b>	
Ator:	Funcionário.
Prioridade:	Essencial.
Entradas e pré-condições:	16 veículos (80%) devem estar registrados com entrada e sem registro de saída.
Saídas e pós-condições:	Gravar no banco de dados 24 entradas de veículos ativas (sem registro de saída) e emitir um alerta de que 80% da capacidade foi atingida.
Fluxos de eventos principal:	<p>1 → 24 entradas de veículos registradas pelo funcionário;</p> <p>2 → As entradas devem estar ativas e sem registro de saída;</p> <p>3 → Emissão de um alerta de que 80% da capacidade foi atingida.</p>

<b>[RF005] Gravar se o proprietário do veículo deixou as chaves</b>	
Ator:	Proprietário do veículo.
Prioridade:	Desejável.
Entradas e pré-condições:	As chaves do veículo devem ser deixadas se o estacionamento tiver alcançado 80% da capacidade de ocupação.
Saídas e pós-condições:	Gravar no banco de dados a informação de que as chaves do veículo foram deixadas.
Fluxos de eventos principal:	1 → Proprietário dá entrada no estacionamento; 2 → Se 80% da capacidade tiver sido alcançada, ele é obrigado a deixar as chaves; 3 → Funcionário registra a informação de que o proprietário do veículo deixou as chaves.

<b>[RF006] Emitir alerta quando atingir 100% de ocupação</b>	
Ator:	Funcionário.
Prioridade:	Essencial.
Entradas e pré-condições:	20 veículos (100%) devem estar registrados com entrada e sem registro de saída.
Saídas e pós-condições:	Gravar no banco de dados 30 entradas de veículos ativas (sem registro de saída) e emitir um alerta de que 100% da capacidade foi atingida. O sistema deve impedir que novas entradas sejam registradas até que não haja o registro de ao menos uma saída.
Fluxos de eventos principal:	<p>1 → 30 entradas de veículos registradas pelo funcionário;</p> <p>2 → As entradas devem estar ativas e sem registro de saída;</p> <p>3 → Emissão de um alerta de que 100% da capacidade foi atingida;</p> <p>4 → Impedir que novas entradas sejam registradas até que não haja o registro de ao menos uma saída.</p>

<b>[RF007] Gravar horário de saída e calcular o preço</b>	
Ator:	Funcionário.
Prioridade:	Essencial.
Entradas e pré-condições:	Registrar a saída.
Saídas e pós-condições:	Gravar no banco de dados o registro da saída. O sistema deve calcular automaticamente o preço a ser pago, com base no horário de entrada/saída. Um recibo deve ser impresso automaticamente.
Fluxos de eventos principal:	<p>1 → Proprietário do veículo chega para retirar seu veículo;</p> <p>2 → Funcionário grava o horário de saída;</p> <p>3 → Preço a ser pago é calculado com base no horário de entrada;</p> <p>4 → Recibo é impresso automaticamente;</p> <p>5 → Fazer a cobrança;</p> <p>6 → Após o pagamento, a saída é permitida;</p> <p>7 → Gravar no banco de dados o registro de uma saída.</p>

<b>[RF008] Preço deve variar conforme o tipo de veículo</b>	
Ator:	Funcionário.
Prioridade:	Essencial.
Entradas e pré-condições:	O sistema deve gravar o tipo de veículo que deu entrada e selecionar o preço automaticamente.
Saídas e pós-condições:	Gravar no banco de dados o tipo de veículo que deu entrada. Ao selecionar o campo veículo “moto” deve gravar o preço de R\$ 4,00 / hora. Ao selecionar o campo veículo “carro” deve gravar o preço R\$ 5,00 / hora.
Fluxos de eventos principal:	<p>1 → Funcionário seleciona o campo “veículo” de acordo com o tipo de veículo que deu entrada;</p> <p>2 → O sistema deve puxar o preço automaticamente de acordo com o tipo de veículo;</p> <p>3 → O tipo de veículo e o preço por hora são gravados no banco de dados.</p>

### 3.3.REQUISITOS NÃO FUNCIONAIS (RNF)

Tabela 03 Requisitos Não Funcionais

Identificador: RNF 001	Categoria: Desempenho
Nome: Tempo limite para processamento de dados de entrada	
Data de criação: 05/09/2020	Autor: Miquelane Melo
Data da última alteração: N/A	Autor: N/A
Versão: 1	Prioridade: Essencial
Descrição: O sistema deverá prover recursos para processamento de dados de entrada de no máximo 5 segundos.	

Identificador: RNF 002	Categoria: Desempenho
Nome: Tempo limite para processamento de dados de Saída	
Data de criação: 05/09/2020	Autor: Miquelane Melo
Data da última alteração: N/A	Autor: N/A
Versão: 1	Prioridade: Essencial
Descrição: O sistema deverá prover recursos para processamento de dados de saída de no máximo 5 segundos.	

Identificador: RNF 003	Categoria: Disponibilidade
Nome: Utilização do módulo de Informações Cadastrais em modo off-line	
Data de criação: 05/09/2020	Autor: Miquelane Melo
Data da última alteração: N/Av	Autor: N/A
Versão: 1	Prioridade: Essencial
<p>Descrição: Todos os registros de clientes cadastrados no sistema poderão ser mantidos (alterados/consultados/excluídos) com o sistema off-line e novos registros de clientes (inclusão) poderão ser incluídos também com o sistema off-line. Todos os relatórios do módulo de informações cadastrais também precisarão rodar off-line.</p>	

Identificador: RNF 004	Categoria: Ambiente
Nome: Compatibilidade de Modelo Arquitetural	
Data de criação: 05/09/2020	Autor: Miquelane Melo
Data da última alteração: N/A	Autor: N/A
Versão: 1	Prioridade: Essencial
<p>Descrição: O Sistema Deve Priorizar o Ambiente de Execução em 64 Bits mas também deve ser compatível com 32 Bits</p>	



Identificador: RNF 005	Categoria: Compatibilidade
Nome: Compatibilidade com sistemas operacionais Windows	
Data de criação: 05/09/2020	Autor: Miquelane Melo
Data da última alteração: N/A	Autor: N/A
Versão: 1	Prioridade: Essencial
<p>Descrição:</p> <p>O Sistema deverá ser executado em Sistema Operacional Windows em versões a partir do XP (XP, Vista 7 e 8), considerando sempre o Service Pack mais recente. O sistema, deverá rodar nos sistemas operacionais elencados neste requisito. O comportamento deve ser o mesmo, tanto no que se refere às funcionalidades quanto à instalação.</p>	

## 4. IMAGENS

Telas do código e do Banco de Dados.

```
1 referência
private void btn_salvar_Click(object sender, EventArgs e)
{
    MySqlConnectionStringBuilder conexaoBD = new MySqlConnectionStringBuilder();
    conexaoBD.Server = "localhost";
    conexaoBD.Database = "cadastro";
    conexaoBD.UserID = "root";
    conexaoBD.Password = "jazmin*2018";
    MySqlConnection realizaConexaoDB = new MySqlConnection(conexaoBD.ToString());

    try
    {
        realizaConexaoDB.Open();
        MySqlCommand comandoMySQL = realizaConexaoDB.CreateCommand();
        comandoMySQL.CommandText =
            "INSERT INTO garagem (nome_veiculo, placa_veiculo, "
            + "proprietario_veiculo, situacao_veiculo, hora_entrada, hora_saida) "
            + "VALUES("
            + textBoxVeiculo.Text + ", "
            + textBoxPlaca.Text + ", "
            + textBoxProprietario.Text + ", "
            + textBoxSituacao.Text + ", "
            + textBoxHoraEntrada.Text + ", "
            + textBoxHoraSaida.Text + ")";
        comandoMySQL.ExecuteNonQuery();

        realizaConexaoDB.Close();
        MessageBox.Show("Inserido com sucesso!");
        atualizarGrid();

        textBoxVeiculo.Text = "";
        textBoxPlaca.Text = "";
        textBoxProprietario.Text = "";
        textBoxSituacao.Text = "";
        textBoxHoraEntrada.Text = "";
        textBoxHoraSaida.Text = "";
    }
}
```

Figura 01: Código em desenvolvimento.

```
private void atualizarGrid()
{
    //crio a estrutura da conexao e passa os parametros.
    MySqlConnectionStringBuilder conexaoBD = new MySqlConnectionStringBuilder();
    conexaoBD.Server = "localhost";
    conexaoBD.Database = "cadastro";
    conexaoBD.UserID = "root";
    conexaoBD.Password = "jazmin*2018";
    // realizo a conexao com o banco de dados.
    MySqlConnection realizaConexaoDB = new MySqlConnection(conexaoBD.ToString());

    try
    {
        realizaConexaoDB.Open();

        MySqlCommand comandoMySQL = realizaConexaoDB.CreateCommand();
        comandoMySQL.CommandText = "SELECT * from garagem";
        MySqlDataReader reader = comandoMySQL.ExecuteReader();

        dataGridView1.Rows.Clear();

        while (reader.Read())
        {
            DataGridViewRow row = (DataGridViewRow)dataGridView1.Rows[0].Clone();
            row.Cells[0].Value = reader.GetInt32(0); // ID
            row.Cells[1].Value = reader.GetString(1); // Veiculo
            row.Cells[2].Value = reader.GetString(2); // Placa
            row.Cells[3].Value = reader.GetString(3); // Proprietario
            row.Cells[4].Value = reader.GetString(4); // Situação do veiculo
            row.Cells[5].Value = reader.GetString(5); // Hora entrada
            row.Cells[6].Value = reader.GetString(6); // Hora saída
            dataGridView1.Rows.Add(row);
        }
    }
}
```

Figura 02: Código em desenvolvimento.

**Cadastro veículo**

ID veículo:  Nome Veículo:  Placa:  Proprietário:

Situação Veículo:    Valor (R\$):

	ID	Veículo	Placa	Proprietário	Situação	Hora Entrada	Hora Saída
▶	1	Gol	jft4033	Pedro Marzano	Amassado no parachoque traseiro	16	18
	2	Palio	thh5789	Manuela Berrutti	ok	13	17
	3	gol	13r232	pedro	ok	12	14
	5	Volvo	rtt5678	Luiz	Lanterna esquerda traseira quebrada	8	14
	6	Kombi	jkf4033	Maria do carmo	ok	9	13
	7	Masseratti	kj4987	Jorge	Lataria amarrada na porta esquerda	12	13
*							

Valor hora/fração: R\$5,00 Capacidade máxima: 15 veículos

@garagemSA

UNIVALI - Hands On Work III

Figura 03: Protótipo de interface.

Query 1

```

1 • create table garagem (
2   id_veiculo int not null auto_increment,
3   nome_veiculo varchar(20) not null,
4   placa_veiculo varchar(10) not null,
5   proprietario_veiculo varchar(30) not null,
6   situacao_veiculo text,
7   hora_entrada varchar(6),
8   hora_saida varchar(6),
9   PRIMARY KEY(id_veiculo)
10 );
11
12 • select * from garagem;

```

Result Grid

	id_veiculo	nome_veiculo	placa_veiculo	proprietario_veiculo	situacao_veiculo	hora_entrada	hora_saida
▶	1	Gol	jft4033	Pedro Marzano	Amassado no parachoque traseiro	16	18
	2	Palio	thh5789	Manuela Berrutti	ok	13	17
	3	gol	13r232	pedro	ok	12	14
	5	Volvo	rtt5678	Luiz	Lanterna esquerda traseira quebrada	8	14
	6	Kombi	jkf4033	Maria do carmo	ok	9	13

garagem 1

Output

Action Output

#	Time	Action	Message
1	17:22:40	select * from garagem LIMIT 0, 100	6 row(s) returned

Figura 04: Banco de dados.

## 5. CRONOGRAMA

05.09 - 11.09 - 1a Semana	12.09 - 18.09 - 2a Semana	19.09 - 26.09 - 3a Semana
Brainstorm das funcionalidades do sistema	Desenvolvimento do banco de dados do sistema	Implementação dos códigos e tratativas com os dados MySql
Definição de funções de cada integrante	Desenvolvimento do formulário e conexão com banco de dados	
Definição dos requisitos e regras de negócio		Teste final do sistema integrado ao banco de dados
Protótipo do layout do sistema		Envio do relatório do projeto final
Envio do relatório inicial do projeto para análise		

## REFERÊNCIAS

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software: uma abordagem profissional**. 8. ed. Porto Alegre: AMGH, 2016. 683 p.