

UNIVERSIDADE DO VALE DO ITAJAÍ
ESCOLA DO MAR, CIÊNCIA E TECNOLOGIA
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

Disciplina: Hand on Work 5

Professor: Lucas Debatin, MSc

Aluno(s): 7770324 - Alexandre Francisco de Souza

Sistema Locação de Veículos

1. [INTRODUCAO](#)
2. [REQUISITOS DO SISTEMA](#)
 - 2.1. [ANÁLISE](#)
3. [DATABASE](#)
 - 3.1. [DER](#)
 - 3.2. [MER](#)
 - 3.3. [CÓDIGO SQL](#)
 - 3.3.1. [PROJETO LÓGICO](#)
 - 3.3.2. [PROJETO FÍSICO](#)
 - 3.3.3. [QUERY CRIAÇÃO DA BASE DE DADOS E TABELAS](#)
 - 3.4. [SIMULAÇÃO SQL](#)
 - 3.4.1. [INSERT \(10\)](#)
 - 3.4.2. [UPDATE \(5\)](#)
 - 3.4.3. [SELECT \(10\)](#)
 - 3.4.4. [DELETE \(5\)](#)
 - 3.4.5. [PROCEDIMENTO E GATILHO \(PROCEDURE | TRIGGER\)](#)
 - 3.4.5.1. [PROCEDURE](#)
 - 3.4.5.2. [TRIGGER](#)
4. [ENGENHARIA DE SOFTWARE](#)
 - 4.1. ANÁLISE DE RISCOS
 - 4.2. [CICLO DE VIDA DO DESENVOLVIMENTO DO APLICATIVO](#)
5. [RELATÓRIO](#)
 - 5.1. RECURSOS DO PROJETO
 - 5.1.1. EQUIPE
 - 5.1.2. INFRAESTRUTURA
 - 5.1.3. HOMOLOGAÇÃO
 - 5.2. DIAGRAMA DE CASOS DE USO
 - 5.2.1. DIAGRAMA
 - 5.2.2. UCP
 - 5.3. CRONOGRAMA
 - 5.4. METODOLOGIA APLICADA
 - 5.5. VIDEO DO PROJETO
 - 5.5.1. LINK GITHUB

PLANO DE PROJETO
Sistema Locação de Veículos

1. INTRODUÇÃO

- Nome: **MKR Rent a Car**
 - Descrição: **Sistema Mobile - Locações de veículos.**
 - Objetivo: **Desenvolver um sistema para controlar as locações de veículos. Destinado a pequenas empresas do ramo de locação de veículos.**
 - Prazo total: **20 Dias (16/04)**
-

O projeto de Locação de veículo (**MKR Rent a Car**) tem como base o mercado de locação, entre cliente e locadora com seus principais atributos e regras de negócios, com base comercial, busquei retratar o fator Cliente (locador) e a empresa (locadora). Entendo que a locação é de maior complexidade e esse Projeto Integrador não foca em todo o processo e sim a introdução ao mesmo.

Em vista as melhores práticas de projeto e o conhecimento adquirido nos cursos de Database Design for Apps e Engenharia de software, busquei ser fiel aos ensinamentos.

Quanto a locação de veículos destacamos o processo, em que, o cliente tem a disposição veículos de marcas e modelos variados, através da locadora (**MKR Rent a Car**), onde por meio de reserva e datas pré-definidas, o cliente terá à disposição veículos e saberá quando devolver o bem locado.

Essa é uma breve introdução do projeto.

2. REQUISITOS DO SISTEMA

2.1. ANÁLISE

- I. Controle de Clientes:
 - a. Campos: Id, c_nome, c_email, c_senha;
 - b. Funções: cadastrar, editar, excluir e listar.

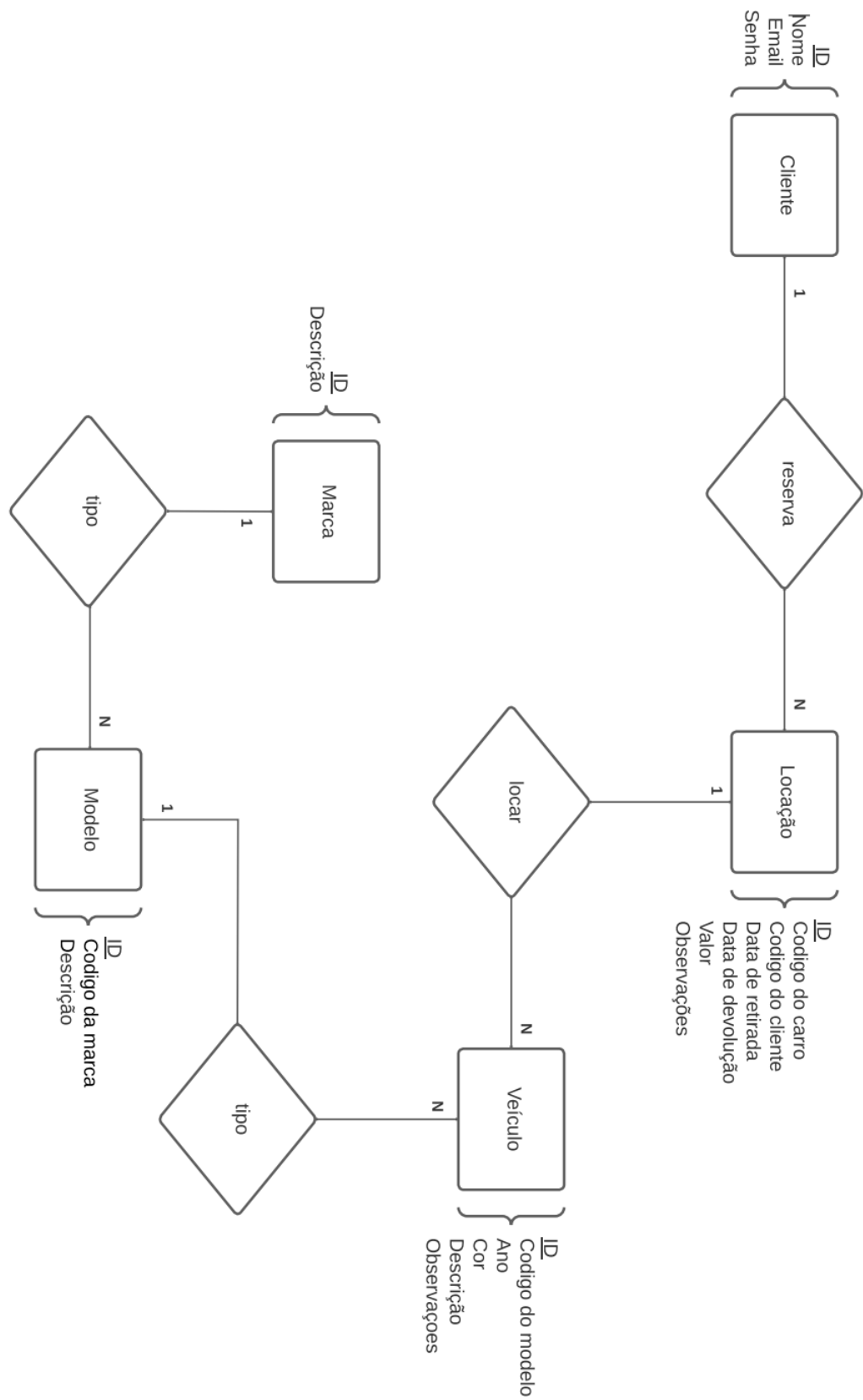
- II. Controle de Locação:
 - a. Campos: Id_loc, l_codveic, l_dta_retirada, l_dta_devolve, l_valor, l_obs;
 - b. Funções: cadastrar, editar, excluir e listar.

- III. Controle de Veículo:
 - a. Campos: Id_car; c_ano, c_cor, c_desc, c_obs;
 - b. Funções: cadastrar, editar, excluir e listar.

- IV. Controle de Modelos:
 - a. Campos: Id_model, md_desc;
 - b. Funções: cadastrar, editar, excluir e listar.

- V. Controle de Marcas:
 - a. Campos: Id_mar, mc_desc, mc_ativo;
 - b. Funções: cadastrar, editar, excluir e listar.

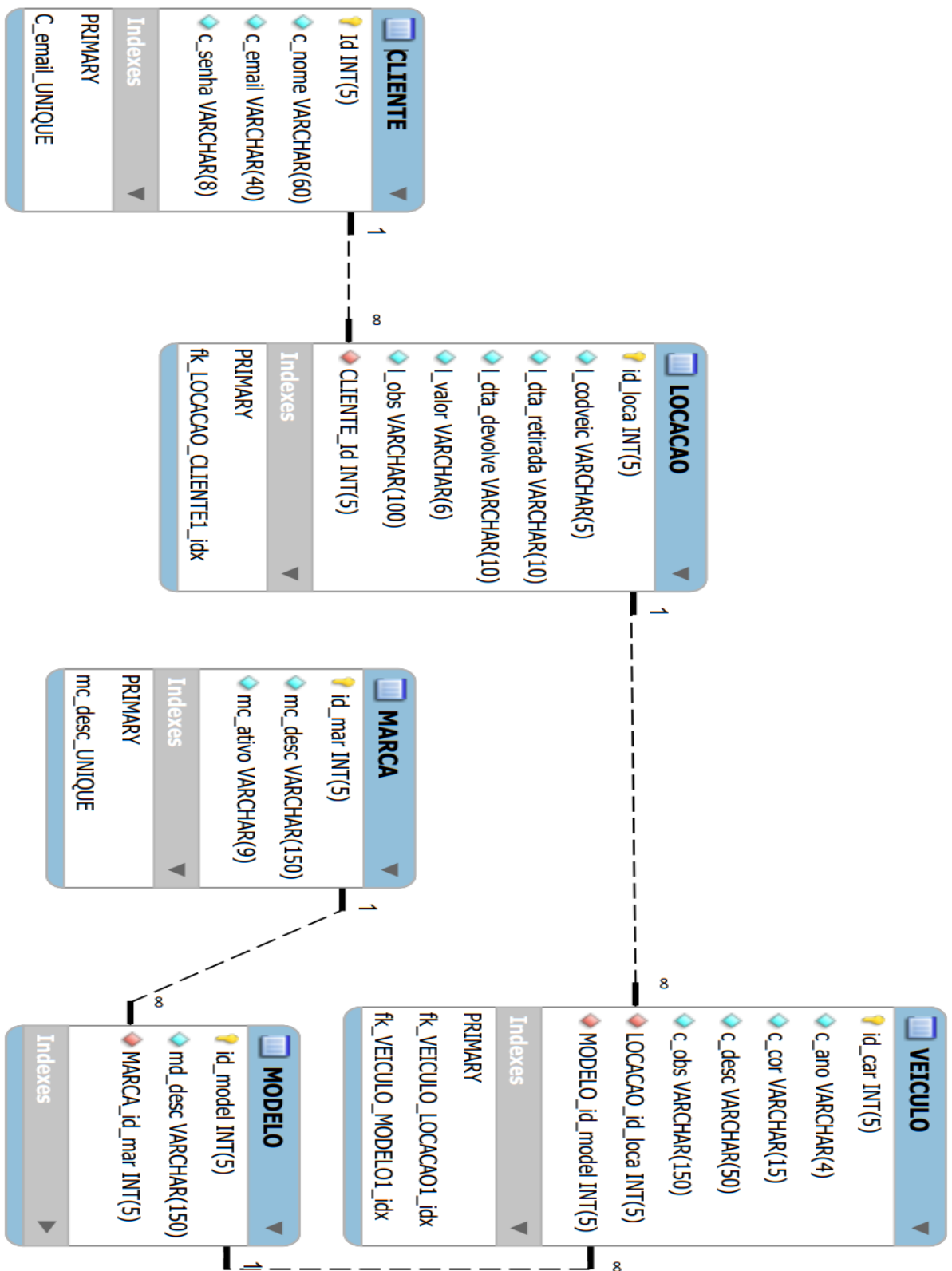
Projeto Conceitual (DER)



3. DATABASE

3.1.DER

3.2. MER



3.3. CÓDIGO SQL

3.3.1. PROJETO LÓGICO

CLIENTE Id, c_nome, c_email, c_senha

LOCACAO Id_loc, l_codveic, l_dta_retirada, l_dta_devolve, l_valor, l_obs;

VEICULO Id_car, c_ano, c_cor, c_desc, c_obs;

MODELO Id_model, md_desc;

MARCA Id_mar, mc_desc, mc_ativo;

3.3.2. PROJETO FÍSICO

CLIENTE

Campo	Tipo	Tamanho	Descrição	Observação
Id	INT	5	Código Cliente	PK, NN
c_nome	VARCHAR	60	Nome do Cliente	NN
c_email	VARCHAR	40	Email do Cliente	NN, UQ
c_senha	VARCHAR	8	Senha do Cliente	NN

LOCACAO

Campo	Tipo	Tamanho	Descrição	Observação
Id_loca	INT	5	Código locacao	PK, NN
l_codveic	VARCHAR	5	Código do veículo	NN
l_dta_retirada	VARCHAR	10	Data de retirada	NN
l_dta_devolve	VARCHAR	10	Data de devolução	NN
l_valor	VARCHAR	6	Valor cobrado	NN
l_obs	VARCHAR	100	Outras informações	NN

VEICULO

Campo	Tipo	Tamanho	Descrição	Observação
Id_car	INT	5	Código veículo	PK, NN
c_ano	VARCHAR	4	Ano de fabricação	NN
c_cor	VARCHAR	15	Cor do veículo	NN
c_desc	VARCHAR	50	Descrição do veículo	NN
C_obs	VARCHAR	150	Outras observações	NN

MODELO

Campo	Tipo	Tamanho	Descrição	Observação
Id_model	INT	5	Código Modelo	PK, NN, AI
md_desc	VARCHAR	150	Descrição do modelo	NN

MARCA

Campo	Tipo	Tamanho	Descrição	Observação
Id_mar	INT	5	Código Marca	PK, NN, AI
mc_desc	VARCHAR	150	Descrição da marca	NN
mc_ativo	VARCHAR	9	Verificação	NN

3.3.3. QUERY CRIAÇÃO DA BASE DE DADOS E TABELAS

```
-- MySQL Script generated by MySQL Workbench
-- Sun May  7 16:23:47 2023
-- Model: MKR Rent a Car    Version: 1.0

-- Projeto HOW 5 UNIVALI - 2023
-- Curso ADS
-- Prof.Lucas Debatin, MSc
-- Academico: Alexandre Francisco de Souza

-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL,ALLOW_INVALID_DATES';

-- -----
-- Schema mkr_rent_a_car
-- -----

-- -----
-- Schema mkr_rent_a_car
-- -----

CREATE SCHEMA IF NOT EXISTS `mkr_rent_a_car` DEFAULT CHARACTER SET utf8
COLLATE utf8_unicode_ci ;
USE `mkr_rent_a_car` ;

-- -----
-- Table `mkr_rent_a_car`.`CLIENTE`
-- -----

CREATE TABLE IF NOT EXISTS `mkr_rent_a_car`.`CLIENTE` (
  `Id` INT(5) NOT NULL,
  `c_nome` VARCHAR(60) NOT NULL,
  `c_email` VARCHAR(40) NOT NULL,
  `c_senha` VARCHAR(8) NOT NULL,
  PRIMARY KEY (`Id`),
  UNIQUE INDEX `C_email_UNIQUE` (`c_email` ASC))
ENGINE = InnoDB;

-- -----
-- Table `mkr_rent_a_car`.`LOCACAO`
-- -----

CREATE TABLE IF NOT EXISTS `mkr_rent_a_car`.`LOCACAO` (
  `id_loca` INT(5) NOT NULL,
  `l_codveic` VARCHAR(5) NOT NULL,
  `l_dta_retirada` VARCHAR(10) NOT NULL,
```

```

`l_dta_devolve` VARCHAR(10) NOT NULL,
`l_valor` VARCHAR(6) NOT NULL,
`l_obs` VARCHAR(100) NOT NULL,
`CLIENTE_Id` INT(5) NOT NULL,
PRIMARY KEY (`id_loca`),
INDEX `fk_LOCACAO_CLIENTE1_idx` (`CLIENTE_Id` ASC),
CONSTRAINT `fk_LOCACAO_CLIENTE1`
  FOREIGN KEY (`CLIENTE_Id`)
  REFERENCES `mkr_rent_a_car`.`CLIENTE` (`Id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----
-- Table `mkr_rent_a_car`.`MARCA`
-----

CREATE TABLE IF NOT EXISTS `mkr_rent_a_car`.`MARCA` (
  `id_mar` INT(5) NOT NULL AUTO_INCREMENT,
  `mc_desc` VARCHAR(150) NOT NULL,
  `mc_ativo` VARCHAR(9) NOT NULL,
  PRIMARY KEY (`id_mar`))
ENGINE = InnoDB;

-----
-- Table `mkr_rent_a_car`.`MODELO`
-----

CREATE TABLE IF NOT EXISTS `mkr_rent_a_car`.`MODELO` (
  `id_model` INT(5) NOT NULL AUTO_INCREMENT,
  `md_desc` VARCHAR(150) NOT NULL,
  `MARCA_id_mar` INT(5) NOT NULL,
  PRIMARY KEY (`id_model`),
  INDEX `fk_MODELO_MARCA1_idx` (`MARCA_id_mar` ASC),
  CONSTRAINT `fk_MODELO_MARCA1`
    FOREIGN KEY (`MARCA_id_mar`)
    REFERENCES `mkr_rent_a_car`.`MARCA` (`id_mar`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----
-- Table `mkr_rent_a_car`.`VEICULO`
-----

CREATE TABLE IF NOT EXISTS `mkr_rent_a_car`.`VEICULO` (
  `id_car` INT(5) NOT NULL,
  `c_ano` VARCHAR(4) NOT NULL,

```

```

`c_cor` VARCHAR(15) NOT NULL,
`c_desc` VARCHAR(50) NOT NULL,
`c_obs` VARCHAR(150) NOT NULL,
`LOCACAO_id_loca` INT(5) NOT NULL,
`MODELO_id_model` INT(5) NOT NULL,
PRIMARY KEY (`id_car`),
INDEX `fk_VEICULO_LOCACAO1_idx` (`LOCACAO_id_loca` ASC),
INDEX `fk_VEICULO_MODELO1_idx` (`MODELO_id_model` ASC),
CONSTRAINT `fk_VEICULO_LOCACAO1`
  FOREIGN KEY (`LOCACAO_id_loca`)
    REFERENCES `mkr_rent_a_car`.`LOCACAO` (`id_loca`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
CONSTRAINT `fk_VEICULO_MODELO1`
  FOREIGN KEY (`MODELO_id_model`)
    REFERENCES `mkr_rent_a_car`.`MODELO` (`id_model`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

```

3.4. SIMULAÇÃO SQL

3.4.1. INSERT

```
INSERT INTO CLIENTE (Id, c_nome, c_email, c_senha) VALUES
('1', 'Alexandre Francisco de Souza', 'alex.souza@edu.univali.br',
'55555555'),
('2', 'João Carlos Vieira', 'joao@edu.univali.br', '66666666'),
('3', 'Claudete Baltazar', 'Claudete@edu.univali.br', '77777'),
('4', 'Leandro Santa', 'lelesa@edu.univali.br', '77777'),
('10', 'Felipe Camargo', 'felipao@edu.univali.br', '77777'),
('6', 'Eduardo Silva', 'dudusilva@edu.univali.br', '77777'),
('7', 'Maria de Jesus', 'mjesus@edu.univali.br', '77777'),
('8', 'Cleiton machado', 'cleitonma@edu.univali.br', '77777'),
('9', 'Alessandro tomate', 'aletomate@edu.univali.br', '77777'),
('5', 'Molusco Desonesto', 'AmorVenceu2@edu.univali.br', '666');

INSERT INTO MARCA (id_mar, mc_desc, mc_ativo) VALUES
('1', 'Hyundai', 'ZeroKM'),
('2', 'Fiat', 'ZeroKM'),
('3', 'Kia', 'ZeroKM');

INSERT INTO MODELO (md_desc, MARCA_id_mar) VALUES
('HB20', 1),
('Uno', 2),
('Soul', 3);

INSERT INTO VEICULO (id_car, c_ano, c_cor, c_desc, c_obs, LOCACAO_id_loca,
MODELO_id_model) VALUES
(2, '2022', 'preto', 'Fiat Palio', 'sem detalhes', 1, 2),
(3, '2022', 'preto', 'Fiat UNO', 'sem detalhes', 1, 2),
(4, '2022', 'preto', 'KIA SERATO', 'sem detalhes', 1, 3);

INSERT INTO LOCACAO (id_loca, l_codveic, l_dta_retirada, l_dta_devolve,
l_valor, l_obs, CLIENTE_Id) VALUES
(1, 'V001', '2023-05-10', '2023-05-13', '300.00', 'Locação para viagem de
fim de semana', 1),
(2, 'V002', '2023-05-10', '2023-05-13', '300.00', 'Locação para viagem de
fim de semana', 2),
(3, 'V003', '2023-05-10', '2023-05-13', '300.00', 'Locação para viagem de
fim de semana', 3);
```

3.4.2. UPDATE

1. Atualizar o nome de um cliente com o ID 1:

sql

Copy code

```
UPDATE CLIENTE
SET c_nome = 'João Silva'
WHERE Id = 1;
```

2. Atualizar o valor de uma locação com o ID 10:

sql

Copy code

```
UPDATE LOCACAO
SET l_valor = '500,00'
WHERE id_loca = 10;
```

3. Atualizar a descrição de um modelo de carro com o ID 3:

sql

Copy code

```
UPDATE MODELO
SET md_desc = 'SUV compacto'
WHERE id_model = 3;
```

4. Atualizar a cor de um veículo com o ID 25:

sql

Copy code

```
UPDATE VEICULO
SET c_cor = 'vermelho'
WHERE id_car = 25;
```

5. Atualizar o **status** de uma marca de carro com o ID 6:

sql

Copy code

```
UPDATE MARCA
SET mc_ativo = 'inativo'
WHERE id_mar = 6;
```

3.4.3. SELECT

SELECT SIMPLES

1. Selecionar todos os dados da tabela CLIENTE:

```
SELECT * FROM CLIENTE;
```

2. Selecionar todos os veículos disponíveis para locação:

```
SELECT * FROM VEICULO WHERE LOCACAO_id_loca IS NULL;
```

3. Selecionar a descrição de todos os modelos de veículos:

```
SELECT md_desc FROM MODELO;
```

4. Selecionar todos os clientes que já alugaram um veículo:

```
SELECT DISTINCT c_nome FROM CLIENTE JOIN LOCACAO ON CLIENTE.Id =  
LOCACAO.CLIENTE_Id;
```

5. Selecionar a marca e modelo de todos os veículos alugados, juntamente com o nome do cliente que alugou o veículo:

```
SELECT MARCA.mc_desc, MODELO.md_desc, CLIENTE.c_nome  
FROM VEICULO  
JOIN LOCACAO ON VEICULO.LOCACAO_id_loca = LOCACAO.id_loca  
JOIN MODELO ON VEICULO.MODELO_id_model = MODELO.id_model  
JOIN MARCA ON MODELO.MARCA_id_mar = MARCA.id_mar  
JOIN CLIENTE ON LOCACAO.CLIENTE_Id = CLIENTE.Id;
```

JOIN

1. Simulação de locações por cliente:

```
SELECT c.c_nome, l.l_dta_retirada, l.l_dta_devolve, l.l_valor  
FROM CLIENTE c  
JOIN LOCACAO l ON c.Id = l.CLIENTE_Id;
```

2. Simulação de locações por modelo de veículo:

```
SELECT m.md_desc, l.l_dta_retirada, l.l_dta_devolve, l.l_valor  
FROM MODELO m  
JOIN VEICULO v ON m.id_model = v.MODELO_id_model  
JOIN LOCACAO l ON v.LOCACAO_id_loca = l.id_loca;
```

3. Simulação de locações por marca de veículo:

```
SELECT ma.mc_desc, l.l_dta_retirada, l.l_dta_devolve, l.l_valor  
FROM MARCA ma  
JOIN MODELO m ON ma.id_mar = m.MARCA_id_mar  
JOIN VEICULO v ON m.id_model = v.MODELO_id_model  
JOIN LOCACAO l ON v.LOCACAO_id_loca = l.id_loca;
```

4. Simulação de locações por **data** de devolução:

```
SELECT c.c_nome, l.l_dta_retirada, l.l_dta_devolve, l.l_valor  
FROM CLIENTE c  
JOIN LOCACAO l ON c.Id = l.CLIENTE_Id  
WHERE l.l_dta_devolve BETWEEN '2023-05-01' AND '2023-05-25';
```

5. Simulação de locações com veículos ainda não devolvidos:

```
SELECT c.c_nome, l.l_dta_retirada, l.l_dta_devolve, l.l_valor  
FROM CLIENTE c  
JOIN LOCACAO l ON c.Id = l.CLIENTE_Id  
JOIN VEICULO v ON l.id_loca = v.LOCACAO_id_loca  
WHERE l.l_dta_devolve > NOW();
```

3.4.4. DELETE

1. Excluindo um registro específico da tabela CLIENTE:

```
DELETE FROM CLIENTE WHERE Id = 1;
```

2. Excluindo todas as locações que possuem um determinado valor na coluna l_valor da tabela LOCACAO:

```
DELETE FROM LOCACAO WHERE l_valor = '100';
```

3. Excluindo todos os registros da tabela VEICULO que possuem um determinado valor na coluna c_ano e cuja locação associada ainda não foi concluída:

```
DELETE FROM VEICULO  
WHERE c_ano = '2010'  
AND LOCACAO_id_loca IN (  
    SELECT id_loca FROM LOCACAO WHERE l_dta_devolve IS NULL  
);
```

4. Excluindo todas as locações associadas a um determinado cliente na tabela LOCACAO:

```
DELETE FROM LOCACAO WHERE CLIENTE_Id = 2;
```

5. Excluindo todos os registros da tabela MODELO que não estão associados a nenhum veículo na tabela VEICULO:

```
DELETE FROM MODELO WHERE id_model NOT IN (  
    SELECT MODELO_id_model FROM VEICULO  
);
```


3.4.5. PROCEDIMENTO E GATILHO (PROCEDURE | TRIGGER)

3.4.5..1. PROCEDURE (PROCEDIMENTOS ARMAZENADO)

Procedure recebe o nome de um cliente e retorne o número total de locações

```
DELIMITER //
```

```
CREATE PROCEDURE total_locacoes_cliente (IN nome_cliente VARCHAR(100), OUT  
total_locacoes INT)  
BEGIN  
    SELECT COUNT(*) INTO total_locacoes FROM locacao  
    WHERE id_cliente = (SELECT id_cliente FROM cliente WHERE nome =  
nome_cliente);  
END //
```

```
DELIMITER ;
```

STATUS DA PROCEDURE: **SHOW PROCEDURE STATUS**
EXECUTAR A PROCEDURE

```
CALL total_locacoes('João');
```

3.4.5..2. TRIGGER (REGISTRAR LOG DATA DE LOCAÇÃO)

```
DELIMITER $$  
CREATE TRIGGER registrar_data_locacao  
BEFORE INSERT ON locacao  
FOR EACH ROW  
BEGIN  
    SET NEW.data_locacao = NOW();  
END$$  
DELIMITER ;
```

Criei o campo data_locacao para registrar log data de locação;

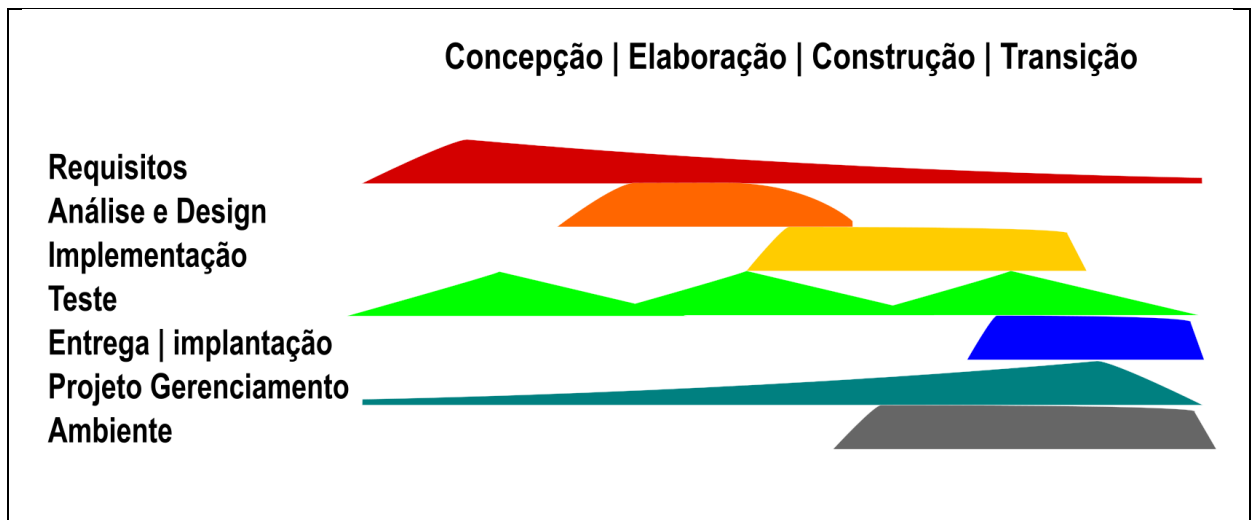
```
ALTER TABLE locacao  
ADD COLUMN data_locacao DATETIME;
```

4. ENGENHARIA DE SOFTWARE

4.1. ANÁLISE DE RISCOS

Detalhes	Impacto	Probabilidade	Gatilho	Plano Contingência
Pequenas mudanças no projeto original	Médio	Média	Adicionar novas funcionalidades as já existentes	Modificação no plano de projeto de acordo com as novas exigências. Adicionar as modificações na próxima atualização do sistema em conformidade ao aos novos cronogramas

4.2. CICLO DE VIDA DO DESENVOLVIMENTO DO APLICATIVO



5. RELATÓRIO

5.1. RECURSOS DO PROJETO

5.1.1. EQUIPE (PESSOAS)

- Alexandre F. Souza

Gerencia Do Projeto | Analista De Sistema |

Desenvolvedor Pleno | Analista QA

5.1.2. INFRAESTRUTURA (HARDWARE)

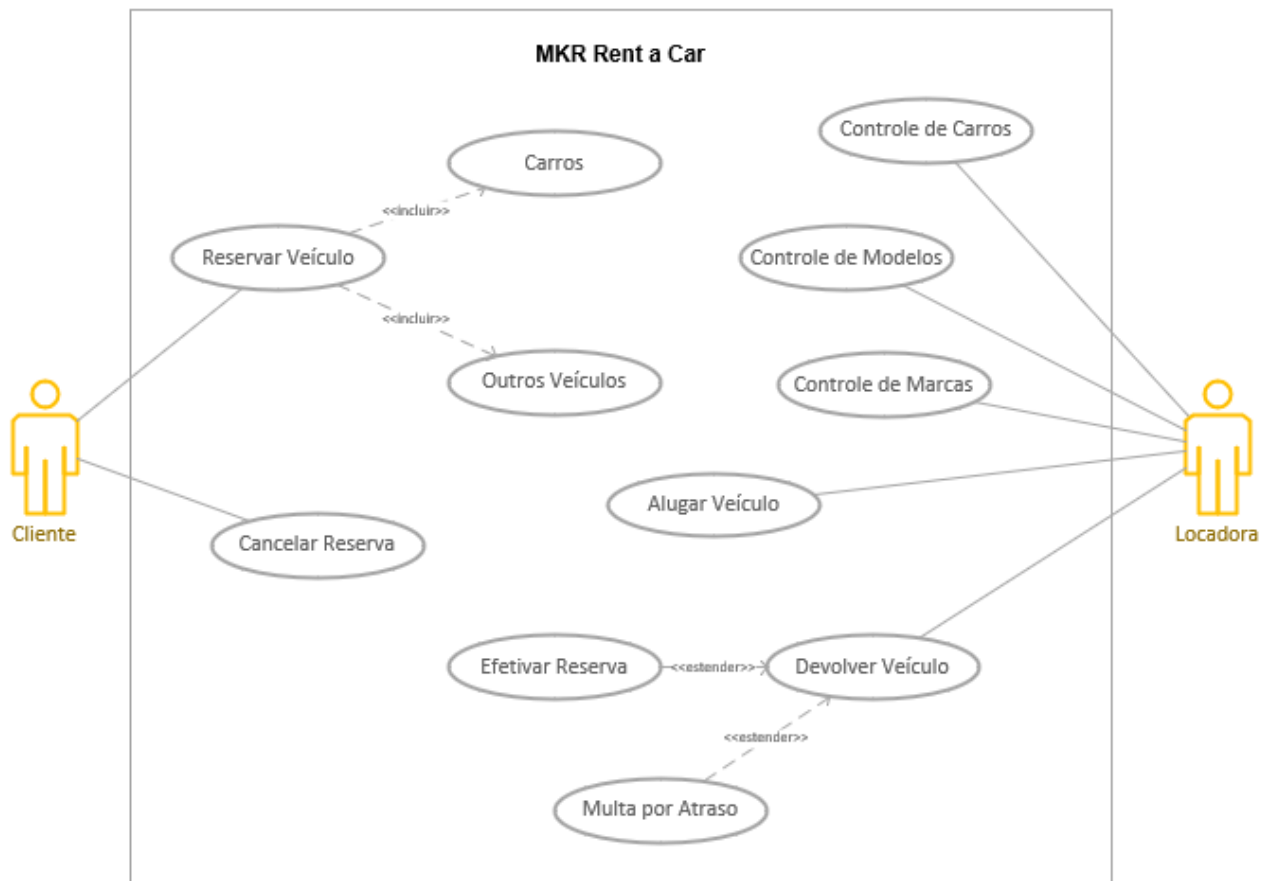
- Servidor com MySQL (BD Banco de dados)
 - Produção: **Cloud**
 - Homologação: **LOCALHOST**
- Ambiente De Desenvolvimentos E Testes:
 - **Notebook** i5, Vídeo dedicada 2mb, Mem12gb, SSD M.2 240gb
 - **PC** AMD Athlon II X2 270 3.40 GHz, Mem 8gb, SSD 1tb

5.1.3. HOMOLOGAÇÃO (SOFTWARE)

- **POO**
programação orientada a objetos, estruturas de controle de fluxo, tipos de dados, entre outros conceitos fundamentais
- **JDK (Java Development Kit)**
JDK inclui a JVM (Java Virtual Machine), o compilador Java e outras ferramentas.
- **Eclipse, NetBeans e IntelliJ IDEA**
A escolha pela IDE ideal ao projeto, será realizada na próxima etapa da documentação.
- **Bibliotecas**
Spring Framework, Hibernate, JUnit

5.2. DIAGRAMA DE CASOS DE USO

5.2.1. DIAGRAMA



1. Manter Cliente: criar, alterar, excluir e listar (médio);
2. Manter Locação: criar, alterar, excluir e listar (médio);
3. Locar Veículo: Realizar a locação e confirmação por e-mail (simples);
4. Devolução Veículo: Devolver, confirmação por e-mail, verificar multa por atraso e verificar reservas (médio);
5. Multa: Gerar e enviar notificação por e-mail para Cliente (simples);
6. Confirmar Reserva: Reservar e notificar por e-mail (simples);
7. Locação: via sistema, acesso, pesquisa e confirmação da reserva (simples);
8. Cancelar Reserva: via sistema, acessar e cancelamento da locação (simples);

5.2.2. UCP

Ator	Complexidade
Cliente	c
Locadora	c

Atores			
Complexidade	Quantidade	Peso	Total
Simple (s)	0	1	0
Médio (m)	0	2	0
Complexo (c)	2	3	6

Caso de Uso	Transações	Complexidade
Manter Cliente	criar, alterar, excluir e listar	m
Manter Locação	criar, alterar, excluir e listar	m
Locar Veículo	Realizar a locação e confirmação	s
Devolução Veículo	Devolver, confirmação por e-mail	m
Multa	Gerar e enviar notificação por e-mail	s
Confirmar Reserva	Reservar e notificar por e-mail	s
Locação	via sistema, acesso, pesquisa e c	s
Cancelar Reserva	via sistema, acessar e cancelam	s

Casos de Uso			
Complexidade	Quantidade	Peso	Total
Simple (s)	5	5	25
Médio (m)	3	10	30
Complexo (c)	0	15	0

EF				
Código	Descrição	Peso	Valor	Resultado
F1	Familiaridade com o processo de desenvolvimento orientado a objetos adotado	1,5	4	6
F2	Colaboradores de meio período	-1	0	0
F3	Capacidade de análise	0,5	4	2
F4	Experiência em desenvolvimento de aplicações deste gênero	0,5	0	0
F5	Experiência em orientação a objetos	1	3	3
F6	Motivação	1	2	2
F7	Dificuldade na linguagem de programação	-1	0	0
F8	Requisitos estáveis	2	5	10

TCF				
Código	Descrição	Peso	Valor	Resultado
F1	Sistema distribuído	2	0	0
F2	Performance	1	2	2
F3	Eficiência para o usuário final (on-line)	1	4	4
F4	Processamento interno complexo	1	1	1
F5	Código deve ser reusável	1	4	4
F6	Fácil para instalar	0,5	0	0
F7	Fácil para usar	0,5	5	2,5
F8	Portável	2	2	4
F9	Fácil para mudar	1	4	4
F10	Concorrente	1	1	1
F11	Seguro	1	4	4
F12	Fornecer acesso direto para third parties (sistemas/componentes externos)	1	0	0
F13	É requerido treinamento do usuário para usar o software	1	2	2

UUCP	61
TCF	0,885
EF	0,71
UCP	38,32935
Horas por UCP	10
Estimativa Horas	383,2935
Dias	47,9116875
Valor hora	R\$ 25,00
Valor total	R\$ 9.582,34

5.3. CRONOGRAMA

CRIAÇÃO	Atividades	Duração	Início	Fim
	Documentação	08:00:00	20/abr	11/mai
	Models	04:00:00	23/abr	26/abr
	Controllers	10:00:00	26/abr	29/abr
	Views	14:00:00	2/mai	5/mai
	Testes	04:00:00	5/mai	8/mai
	implantação (Servidor)	04:00:00	8/mai	11/mai
	Total	44:00:00		

5.4. METODOLOGIA APLICADA

O principais desafios ao começar um projeto, durante a criação do software é conseguir uma execução bem-feita e uma entrega final coerente com o que fora proposto inicialmente. A gestão de projetos, análise, desenvolvimento, testes, entre outros, exige muita atenção e mudanças durante o projeto torna o projeto bastante custoso e desprende refazer cronogramas, por isso a escolha pelo METODOLOGIA AGIL, vem de encontro a essas flexibilidade. As práticas Ágeis tornam o projeto mais eficiente e assertivo.

Nesse projeto iremos utilizar algumas práticas como, SCRUM, LEAN (Construir, medir, aprender), KANBAN (TO DO, DOING, DONE), entre outros.

Abaixo exemplo do ciclo de um projeto, utilizando de metodologia Ágil.

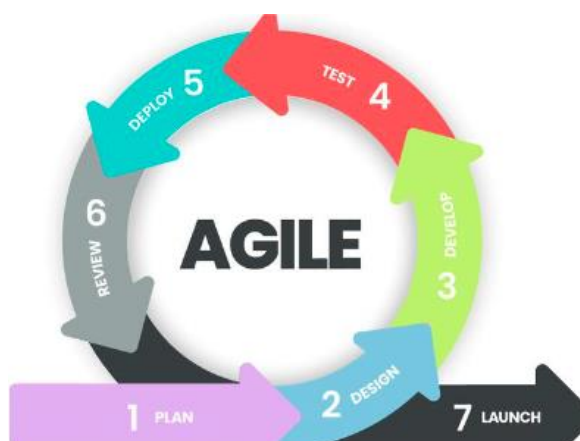


Imagem da internet: fonte: [link](#)

5.5. VIDEO DO PROJETO

5.5.1. LINK GITHUB: [CLIQUE AQUI!](#)