

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

Disciplina: ENGENHARIA DE SOFTWARE

Professor: Lucas Debatin, MSc

Aluno(s): 7770324 - Alexandre Francisco de Souza
7800746 - João Carlos Vieira

PLANO DE PROJETO
Sistema Locação de Veículos

1. INTRODUCAO
2. CICLO DE VIDA DO PROJETO
3. RECURSOS DO PROJETO
 - 3.1. EQUIPE
 - 3.2. INFRAESTRUTURA
 - 3.3. HOMOLOGAÇÃO
4. DIAGRAMA DE CASOS
 - 4.1. DIAGRAMA DE CASOS DE USO
 - 4.2. UCP
5. SIMULAÇÃO
 - 5.1. ESTIMATIVA CUSTO MÃO DE OBRA
 - 5.2. CRONOGRAMA DO PROJETO
6. METODOLOGIA DO PROJETO

PLANO DE PROJETO

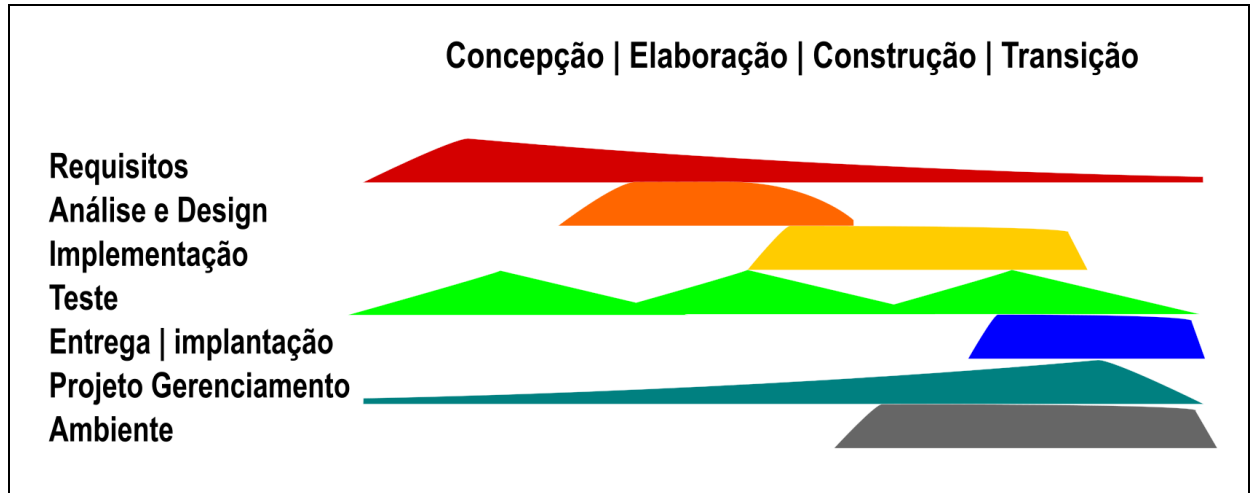
Sistema Locação de Veículos

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: **21 dias.**

Após análise do projeto, foram levantados os requisitos abaixo citado;

2. CICLO DE VIDA DO DESENVOLVIMENTO DO APLICATIVO



3. RECURSOS DO PROJETO

3.1. Equipe (Pessoal):

- Alexandre F. Souza: **Gerencia do Projeto | Analista de Sistema | Desenvolvedor Pleno**
- João Carlos Vieira: **Analista responsável pelo projeto, Desenvolvedor, Analista QA**

3.2. Infraestrutura (Hardware):

01 – Servidor com MySQL (BD Banco de dados)

- Produção: **Cloud**
- Homologação: **LOCALHOST**

02 – Ambiente de Desenvolvimentos e Testes:

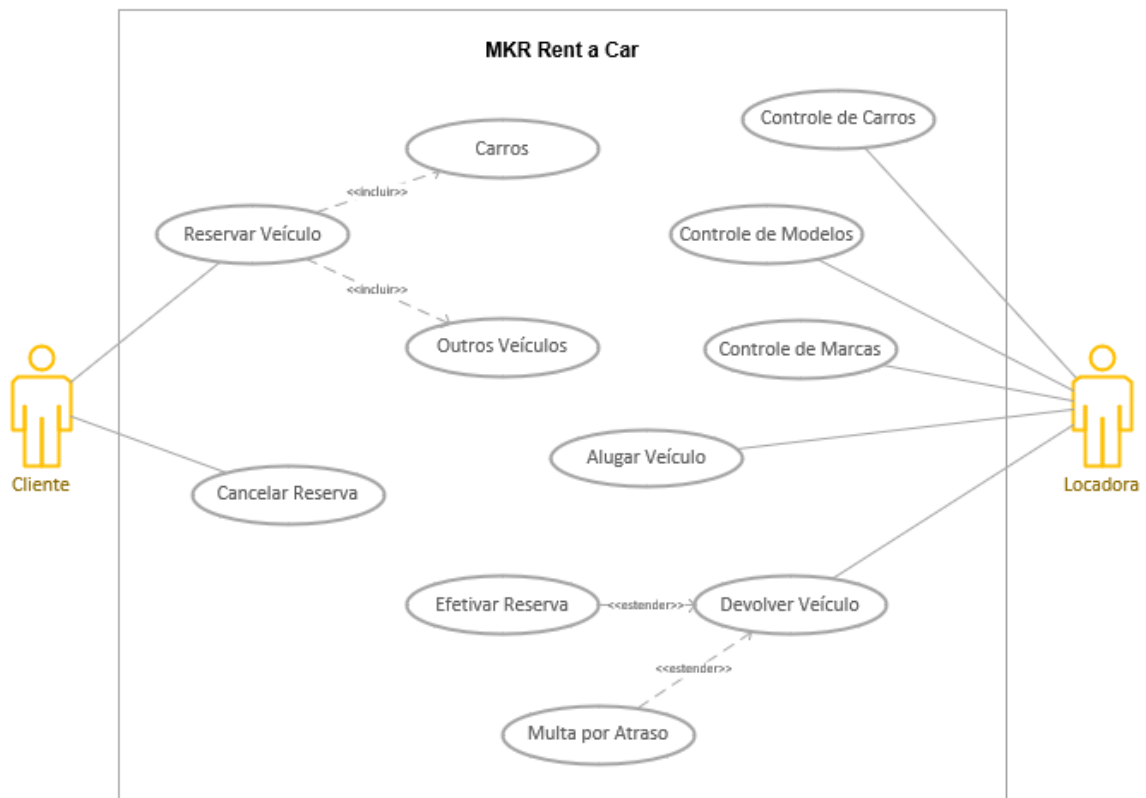
- **Notebook** i5, Vídeo dedicada 2mb, Mem12gb, SSD M.2 240gb
- **PC** AMD Athlon II X2 270 3.40 GHz, Mem 8gb, SSD 1tb

3.3. 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

4. DIAGRAMA DE CASOS DE USO

4.1. Diagrama de casos de uso



Arquivo no GitHub: [Clique Aqui!](#)

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);

4.2. 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 cadastro	s
Cancelar Reserva	via sistema, acessar e cancelar	s

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

Arquivo no GitHub: [Clique Aqui!](#)

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

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

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

Arquivo no GitHub: [Clique Aqui!](#)

5. SIMULAÇÃO

5.1. Estimativa custo mão de obra

Salário Programador	R\$ 4.200,00
Valor por LOC	R\$ 0,60

Esforço (pessoa)	R\$ 1,63
Custo	R\$ 1.547,50

Arquivo no GitHub: [Clique Aqui!](#)

5.2. 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		

Arquivo no GitHub: [Clique Aqui!](#)

6. METODOLOGIA DO PROJETO

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.

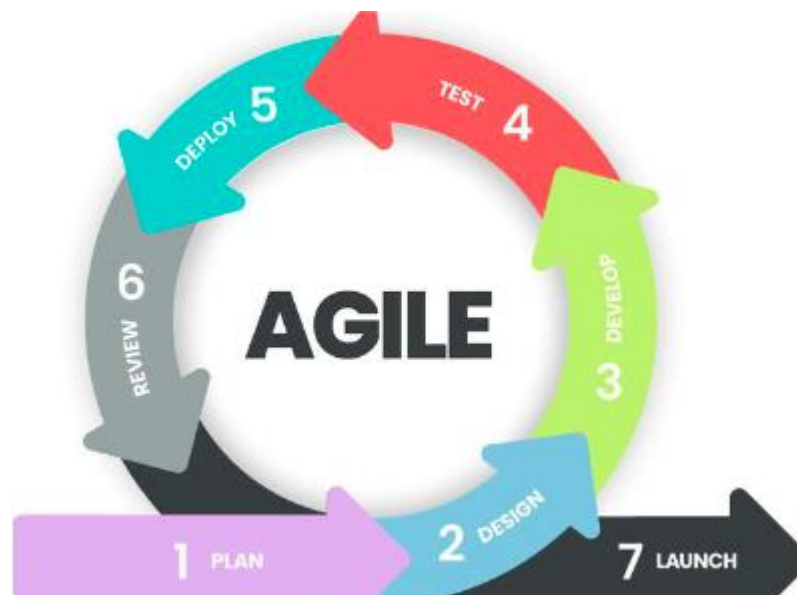


Imagem da internet: fonte: [link](#)