

Sistema Lavacar

<G.R. Systems>



Cliente: <Mike (LavaCarT)>
Responsáveis no cliente: Gabriel
Eugenio Brito, Rodrigo Alves Guerra

Versão: 1.0

Curitiba
2016

Sistema Lavacar

Responsáveis no cliente: **Gabriel Eugenio Brito, Rodrigo Alves Guerra.**

Responsáveis pelo projeto e desenvolvimento: **Gabriel Eugenio Brito, Rodrigo Alves Guerra.**

Curitiba
2016

Histórico de Modificações

Data	Versão	Descrição	Autor
20/09/2016	0.2	Levantamento de requisitos	Gabriel E. e Rodrigo G.
27/09/2016	0.3	Diagrama de Casos de Uso	Gabriel E. e Rodrigo G.
04/09/2016	0.4	Início da interface	Gabriel E.
25/10/2016	0.5	Menu principal e interface do funcionário prontas, a parte do gerente roda no console	Gabriel E.
25/10/2016	0.6	Adicionado alguns diagramas de atividade	Rodrigo G.
26/10/2016	0.7	Diagramas restantes de atividade e diagrama de classes	Gabriel E. e Rodrigo G.
01/11/2016	0.8	Adicionado alguns diagramas de sequência	Gabriel E.
06/11/2016	0.9	Mais diagramas de sequência	Rodrigo G.
14/11/2016	1.0	Conclusão do relatório	Gabriel E. e Rodrigo G.

SUMÁRIO

1 INTRODUÇÃO

1.1 Objetivo Geral

1.2 Objetivos Específicos

1.3 Conteúdo do Projeto

2 LEVANTAMENTO DE REQUISITOS

2.1 Questões Organizacionais

2.2 Questões Econômicas

2.3 Questões Técnicas

2.4 Questões Operacionais

2.5 Requisitos Funcionais e Não Funcionais

3 ESTUDO DE VIABILIDADE

3.1 Viabilidade Organizacional

3.2 Viabilidade Econômica

3.3 Viabilidade Técnica

3.4 Viabilidade Operacional

3.5 Recursos a serem utilizados

4 RESULTADOS

4.1 Conteúdo dos Resultados

4.2 Modelagem

5 CONCLUSÕES

5.1 Trabalhos Futuros

6 REFERÊNCIAS BIBLIOGRÁFICAS

LISTA DE FIGURAS

[Figura 1. Diagrama de casos de uso](#)

[Figura 2. Diagrama parcial de classes](#)

LISTA DE ABREVIATURAS E SIGLAS

IDE: Integrated Development Environment

GUI: Common Gateway Interface - Interface de Porta Comum.

API: Application Programming Interface

UML: Unified Modeling Language

1 INTRODUÇÃO

O sistema proposto é uma simplificação de um sistema para o gerenciamento de um lavacar. O projeto será desenvolvido em um período de 3 meses exclusivamente para a empresa LavaCarT.

1.1 Objetivo Geral

O programa tem por objetivo auxiliar no controle de um sistema de lavacar.

1.2 Objetivos Específicos

O sistema tem como objetivos possibilitar os funcionários à cadastrar clientes, manter o controle dos veículos na fila, informar preços, estimar o tempo de espera aos clientes e gerar relatórios.

1.3 Conteúdo do Projeto

Este documento está dividido em cinco partes:

Capítulo 2: Levantamento de Requisitos.

Capítulo 3: Estudo de Viabilidade.

Capítulo 4: Resultados.

Capítulo 5: Conclusões.

Capítulo 6: Referências Bibliográficas.

2 LEVANTAMENTO DE REQUISITOS

2.1 Questões Organizacionais

1.O sistema trará benefícios para o controle de finanças da empresa?

Sim, será possível gerar e analisar relatórios de finanças diários, mensais e anuais com o número de serviços realizados, lucro obtido e gastos.

2.Os usuários irão colaborar com o novo sistema?

Todos os usuários, sejam eles clientes do lavacar ou funcionários, poderão contribuir com ideias para futuras melhorias no sistema.

3.Quão bem o sistema será aceito pelos usuários?

Após o treinamento dos funcionários para a utilização, o sistema deve ser bem aceito, pois torna os processos mais simples, práticos e rápidos.

4.Como o sistema será incorporado ao processo de negócio?

No processo antigo, os cadastros eram feitos no papel e a parte de finanças em tabelas no Excel. Agora cadastros, controle de serviços, relatórios de finanças, controle de estoque, tudo será feito no sistema, tornando o processo muito mais ágil.

2.2 Questões Econômicas

1.Que custo estima-se para o desenvolvimento do sistema?

Além do pagamento para os desenvolvedores, será necessário treinar os funcionários para a utilização do novo sistema. Estima-se que o custo total seja em torno de R\$ 9000,00.

2.Será necessário adquirir novos hardwares?

Não, o sistema será desenvolvido para o hardware atual da empresa.

3.Que benefícios visíveis o sistema proporciona?

Melhorias no atendimento ao cliente, verificação de desempenho da empresa, possibilidade de fidelizar clientes e redução de gastos.

4. Como espera-se que o sistema traga um aumento nos lucros da empresa?

Como a velocidade de cadastro e controle dos serviços é maior, é possível atender mais clientes em menos tempo.

2.3 Questões Técnicas

1. Os desenvolvedores têm familiaridade com o negócio?

Não, mas é algo que pode ser trabalhado em entrevistas feitas com o gerente e os funcionários da empresa.

2. Os desenvolvedores têm familiaridade com as tecnologias utilizadas (sistema orientado a objetos, interface gráfica, etc)?

Com programação orientada a objetos sim, não tanto com interfaces gráficas. No entanto, os membros da equipe estão dispostos a aprender o que for necessário para o desenvolvimento do sistema.

3. Futuramente será possível modificar o sistema para incluir outras funcionalidades?

Sim, o sistema será desenvolvido de maneira desacoplada, tornando mais fácil qualquer manutenção de código no futuro, desde alterações nos cálculos financeiros até a troca da fonte de informações do sistema (arquivos de texto, bancos de dados, entre outros). Também é possível criar uma forma de cadastro de serviços onde o próprio gerente realize o processo, sem o apoio de um técnico.

4. A equipe é capaz de realizar um projeto deste tamanho no tempo determinado?

Embora a equipe seja pequena, não será necessário integrar o novo sistema à outros antigos da empresa, o que torna o projeto mais simples.

2.4 Questões Operacionais

1. Quem poderá operar o sistema?

Clientes do lavacar, funcionários e o próprio gerente.

2.Com o novo sistema, em quanto tempo espera-se que um funcionário cadastre um novo cliente e esteja pronto para iniciar o serviço? De 2 a 4 minutos, dependendo da velocidade de digitação do funcionário.

3.Caso o gerente decida criar um novo tipo de serviço, ele pode fazer isso por conta própria ou é necessário chamar um técnico?

Inicialmente, para adicionar um novo serviço, é necessário conhecimentos em programação, mas isto é algo que possa ser pensado para futuras versões.

4.É necessário algum software ou hardware adicional para rodar o sistema?

Não, basta executar o arquivo .jar resultante na máquina.

2.5 Requisitos Funcionais, Não Funcionais e Restrições de Projeto

Requisitos Funcionais

- Requisitos essenciais

RF-1 : Cadastro de clientes e veículos

RF-2 : Controlar a execução de um serviço (lavagem, polimento, etc), ou seja, descontar automaticamente os materiais necessários do estoque e salvar o valor recebido para futuros usos

RF-3 : Adicionar veículos em uma fila, para que a lavagem seja feita em ordem de chegada

RF-4 : Cadastro de funcionários

RF-5 : Listar todos os clientes e funcionários

RF-6 : Gerar relatórios diário, mensal e anual, contendo o número de serviços realizados, lucro obtido e gastos como salário dos funcionários e contas a pagar

RF-7 : Controle de estoque (quantidade de sabão, esponjas, panos, etc)

RF-8 : Acrescentar mais produtos ao estoque (a compra será feita externamente e o gerente deverá informar ao sistema quantos produtos de cada tipo comprou)

RF-9 : Editar informações dos clientes (telefone, endereço) e veículos (cor, descrição) cadastrados no sistema

- Requisitos importantes

RF-10 : Listar os veículos de um determinado cliente ou todos os veículos cadastrados

RF-11 : Alterar o preço dos serviços

RF-12 : Alterar o salário dos funcionários

- Requisitos desejáveis

RF-13 : Estimar o tempo de espera até a execução do próximo serviço

RF-14 : Realizar um diagnóstico de um veículo que está aguardando na fila, identificando possíveis problemas no veículo e sugerindo outros serviços

RF-15 : Permitir que os clientes visualizem os preços

RF-16 : Permitir que os clientes visualizem as informações da empresa, como telefone e site

Requisitos Não Funcionais

- Requisitos essenciais

RNF-1 : Um funcionário deve ser capaz de cadastrar um veículo e iniciar o serviço em menos de 3 minutos

RNF-2 : Salvar as informações em algum lugar para recuperar em uma outra execução do programa (arquivos de texto, banco de dados ou outro)

- Requisitos importantes

RNF-3 : Garantir que apenas o gerente/administrador possa contratar funcionários, alterar preços e gerar relatórios

RNF-4 : Garantir que apenas funcionários possam adicionar veículos à fila e cadastrar clientes e veículos.

RNF-5 : O valor e o tempo de serviço variam de acordo com o tamanho do veículo

RNF-6 : Permitir o cadastro de mais de um veículo para cada cliente

- Requisitos desejáveis

RNF-7 : A implementação de um novo serviço deve levar menos de 2h

Restrições de Projeto

RP-1 : Para cadastrar um novo serviço é necessário editar o código-fonte

3 ESTUDO DE VIABILIDADE

3.1 Viabilidade Organizacional

Será um auxílio na gerência dos serviços executados pelos funcionários. Também reduz o tempo para que o cliente seja atendido, melhorando o processo.

3.2 Viabilidade Econômica

O sistema resultará em uma economia de custos e aumento da receita, pois o controle de serviços é muito mais simples e eficiente. Além disso espera-se um aumento dos lucros, pois é possível atender mais clientes em menos tempo.

3.3 Viabilidade Técnica

O hardware é capaz de executar o programa sem a ajuda de softwares adicionais. Para a manutenção do software, no entanto, é necessário um compilador para linguagem orientada a objetos. Será preciso uma interface gráfica e possivelmente integração com bancos de dados. Todos os membros da equipe possuem experiência com sistemas simples, mas apenas um (Rodrigo) já trabalhou com interfaces gráficas e bancos de dados.

3.4 Viabilidade Operacional

Após o treinamento dos funcionários para a utilização, o sistema deve ser bem aceito, pois torna os processos mais simples e práticos. Entretanto, será necessário o comprometimento dos funcionários e da organização para a atualização do sistema, reportando eventuais erros e possibilitando melhorias no projeto.

3.5 Recursos a serem utilizados

Os recursos utilizados para o desenvolvimento do sistema serão uma IDE com suporte à linguagem orientada a objeto e a utilização de pelo menos um computador. Estima-se que serão necessárias em torno de 90h de trabalho para cada membro da equipe, totalizando cerca de 180h. À R\$30,00 por hora de trabalho, estipula-se que o custo do projeto seja em torno de R\$ 5.400,00 reais.

4 RESULTADOS

4.1 Conteúdo dos Resultados

Com o controle de estoque e os relatórios gerados é possível analisar a situação atual da empresa e com base nisso decidir o futuro da mesma. Também pudemos ver uma melhora na velocidade do atendimento, o que se converte em um lucro maior no final do mês.

Os clientes agora podem visualizar as informações à qualquer momento, sendo desnecessário interromper o trabalho de um funcionário, e eles adoraram a possibilidade de estimar o tempo até o término do serviço.

Os funcionários se adaptaram bem ao novo sistema, afinal, o cadastro anteriormente feito no papel, agora é totalmente digital. O gerente também se mostrou satisfeito com o sistema.

4.2 Modelagem

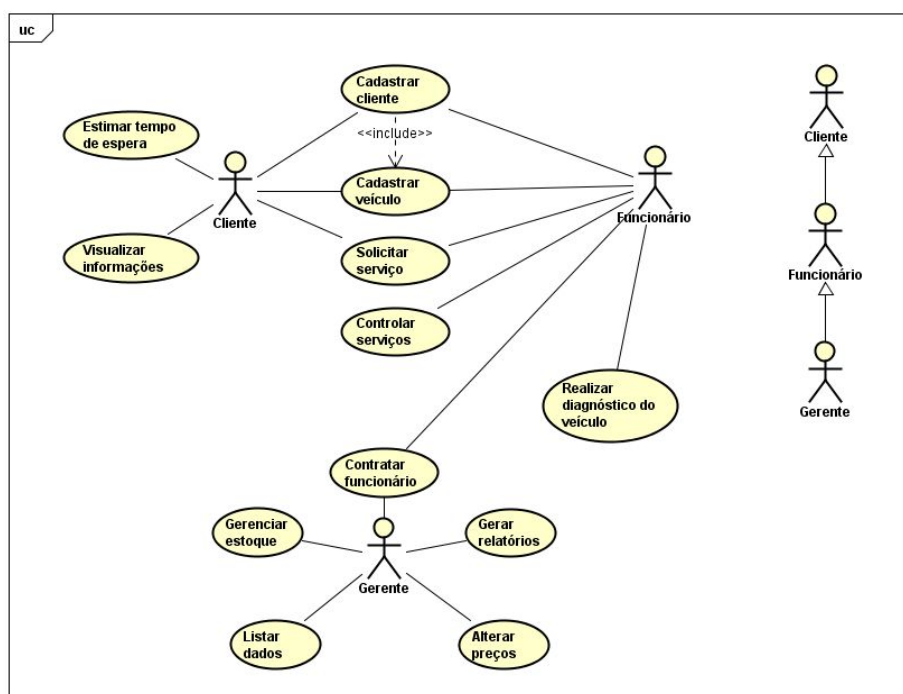


Figura 1. Diagrama de casos de uso

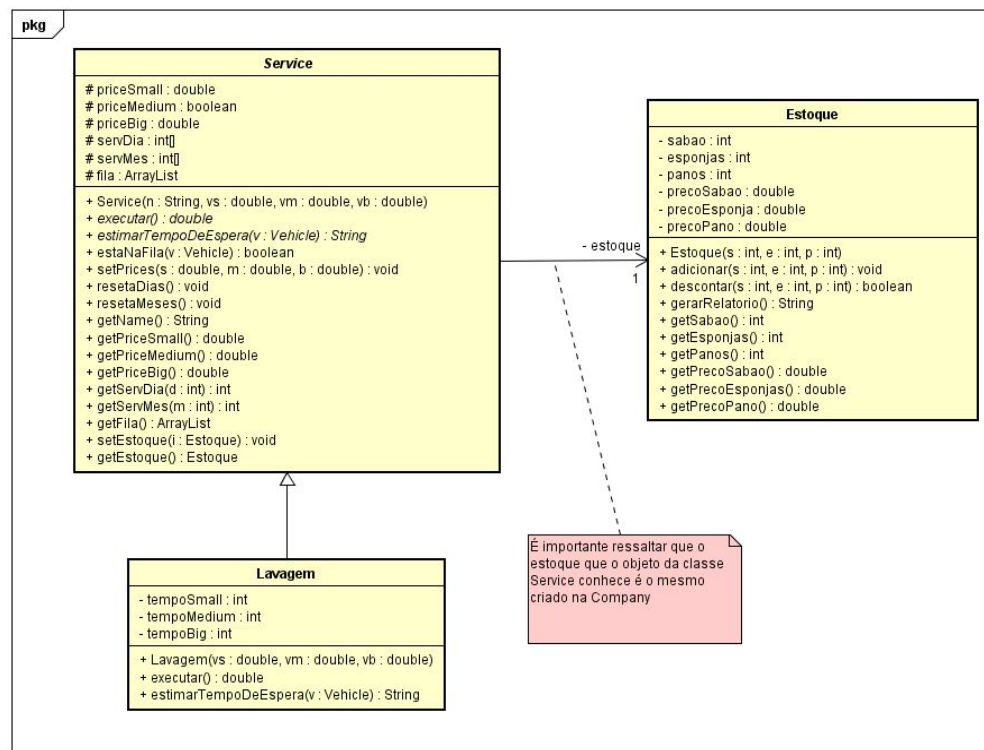


Figura 2. Diagrama parcial de classes

Por questão de formatação, somente alguns diagramas foram incluídos no relatório. Mais detalhes sobre estes e vários outros diagramas UML podem ser encontrados no arquivo .asta incluído.

5 CONCLUSÕES

Nas fases iniciais tivemos algumas dificuldades referentes à como organizar o sistema de forma que resolvesse todos os problemas da empresa. No entanto, conforme o projeto avançou e os diagramas foram feitos, a implementação se tornou algo trivial.

Procuramos uma forma de salvar dados do sistema e por (ainda) não termos muito conhecimento em bancos de dados, optamos por utilizar arquivos de texto. Como a busca nesses arquivos é algo no mínimo trabalhoso, tivemos a ideia de utilizar dois arquivos para cada tipo de informação, um que o compilador lê e recupera as informações (feito com o recurso serializable do java) e outro para leitura dos usuários (strings concatenadas e formatadas de uma maneira mais agradável ao olhar).

Em relação à interface gráfica, no início não sabíamos muito bem como iríamos desenvolvê-la, mas assim que descobrimos o NetBeans IDE GUI Builder, o desafio se tornou muito mais simples.

Por fim, quanto a parte de projeto, percebemos o quanto os diagramas UML auxiliam no desenvolvimento de um software, proporcionando um entendimento maior do sistema antes de sua criação, de forma que a implementação se torna muito mais simples.

5.1 Trabalhos Futuros

Futuramente, é possível elaborar um modo em que o próprio gerente possa cadastrar novos serviços, sem a ajuda de um técnico para modificar o código. Também podemos fazer melhorias na interface, integrar o sistema com algum banco de dados e quem sabe até fazer com que vários computadores se conectem ao mesmo banco de dados, compartilhando informações entre si.

6 REFERÊNCIAS BIBLIOGRÁFICAS

Java™ Platform, Standard Edition 7 API Specification. Disponível em <<https://docs.oracle.com/javase/7/docs/api/>>.

GUERRA, GLAUCIO. Introdução a Serialização De Objetos.

Disponível em

<<http://www.devmedia.com.br/introducao-a-serializacao-de-objetos/3050>>

Designing a Swing GUI in NetBeans IDE. Disponível em <<https://netbeans.org/kb/docs/java/quickstart-gui.html>>.

ROBSON, RIBEIRO LINHARES. Aulas de Técnicas De Programação.

Página pessoal Disponível em <<http://www.dainf.ct.utfpr.edu.br/~robson/>>.

RITA, BERARDI. Aulas de Análise e Projeto De Sistemas.