

Universidade Federal de Ouro Preto



Engenharia de Software II

Sistema de *Locadora*

Grupo: *Tilápia*

Alunos:	Avner Allen Figueiredo Fabricio César Tofani Lucas Natali Magalhães Silva Lucca Arantes Martins Pedro Alves Euzebio Rodrigo Gomes Chaves Vivyann Fernandes Cedraz
Professor:	Msc prof. Johnatan Oliveira
Horário:	Seg & Qua - 08:20 -10:00

Ouro Preto, 06 de Janeiro de 2021

Conteúdo

1	Histórico de Revisões	1
2	Processo e Software	1
3	Cronograma	1
4	Levantamento de Requisitos	2
4.1	Diagramas de casos de uso	3
5	Especificação de Requisitos	4
5.1	Requisitos Funcionais	4
5.2	Requisitos Não Funcionais	7
6	Plano de VVT	7
6.1	Requisitos a serem testados	7
6.1.1	Controle de acesso dos administradores	7
6.1.2	Validação do cadastro de itens	8
6.1.3	Validação cadastro de funcionários	8
6.1.4	Validação cadastro de clientes	8
6.2	Estratégias e ferramentas de teste	8
6.3	Equipe e infra-estrutura	8
6.4	Execução do Plano de Teste	9
6.4.1	Configuração das ferramentas de teste	9
6.4.2	Implementar os testes	9
6.4.3	Fazer correções	9
7	Medição e Qualidade de Software	9
7.1	Code Smells	9
7.2	Pylint	10
8	Observações	10
9	Referências	11

1 Histórico de Revisões

Data	Versão	Descrição	Autor
04/03/2021	1.0	Início da escrita do documento	Lucas
04/03/2021	1.0	Justificativa do processo de software	Lucca
12/03/2021	1.0	Levantamento dos requisitos do software	Lucca, Rodrigo, Vivyann
18/03/2021	1.0	Especificação dos requisitos	Fabício, Pedro, Rodrigo
19/03/2021	1.0	Diagramas de caso de uso	Fabício, Pedro, Rodrigo
25/03/2021	1.0	Plano de VVT	Fabício, Pedro
30/03/2021	1.0	Medição e qualidade de software	Lucas, Pedro

Tabela 1: Revisões do Documento

2 Processo e Software

Decidimos utilizar o framework Scrum, pelo fato de ele estar sendo cada vez mais utilizado no mercado, e seguir uma mentalidade inovadora. O Scrum favorece o trabalho em equipe, e oferece uma forma de organização de trabalho que se adapta muito bem às nossas necessidades. Pelo fato de termos um tempo relativamente pequeno para a entrega final (21/03 - 10/04), precisamos fazer com que o projeto esteja sempre convergindo para nosso real objetivo da forma mais concisa possível. E para que consigamos acompanhar essa conversão coerentemente, dividimos o tempo de desenvolvimento em intervalos iguais. Portanto, cada Sprint terá duração de uma semana.

3 Cronograma

Aqui apresentamos o cronograma de atividades do grupo

Nome	Tarefa	Prazo
Todos	Definição da linguagem para implementação, do modelo de processo e cronograma	01/03 a 08/03
Lucca	Justificativa sobre o processo de desenvolvimento de software escolhido.	01/03 a 13/03
Lucca, Rodrigo, e Vivyann	Levantamento dos requisitos do software.	09/03 a 13/03
Fabício, Pedro, e Rodrigo	Definição dos diagramas de casos de uso e descrição dos cenários de casos de uso.	13/03 a 20/03
Fabício, e Pedro	Plano de verificação, validação e teste de software.	21/03 a 27/03
Avner, Lucas, Pedro	Procurar técnicas e ferramentas para medição e qualidade de software	28/03 a 01/04
Lucas, Lucca, e Pedro	Implementação das classes e métodos.	21/03 a 10/04
Fabício, e Lucas	Implementação dos testes.	20/03 a 10/04
Rodrigo, e Vivyann	Gravação da primeira apresentação.	31/03 a 05/04
Rodrigo, e Vivyann	Primeira apresentação.	05/04
Avner, e Lucca	Gravação da segunda apresentação.	05/04 a 08/04
Avner, e Lucca	Segunda apresentação.	08/04
Fabício, Lucas e Pedro	Terceira apresentação	12/04

Tabela 2: Cronograma

4 Levantamento de Requisitos

Técnica utilizada para o levantamento dos requisitos: Entrevistas.

Por se tratar de um sistema com o qual o cliente estará constantemente em contato, vimos a entrevista como a melhor alternativa para o levantamento

dos requisitos do software, pois através dela é possível descobrir quais são as necessidades e dores que o cliente tem ao solicitar o projeto. Dessa forma, estamos seguindo um dos pilares do Modelo Ágil, que é prezar pela boa comunicação e relação entre as pessoas. Além disto, fica mais fácil extrair o que o sistema deve conter e quais funcionalidades ele deve apresentar.

Requisitos levantados durante a entrevista:

- O sistema deve permitir o CRUD de itens;
- O sistema deve permitir o CRUD de clientes;
- O sistema deve permitir o CRUD de funcionários;
- O sistema deve permitir a locação de um item;
- O sistema deve calcular automaticamente o valor do aluguel de um determinado item;
- O sistema deve permitir a administração dos itens locados;
- O sistema deve permitir buscar itens;
- O sistema deve permitir listar todos os itens existentes no estoque;
- O sistema deve gerar o relatório mensal até o momento dos itens alugados.

4.1 Diagramas de casos de uso

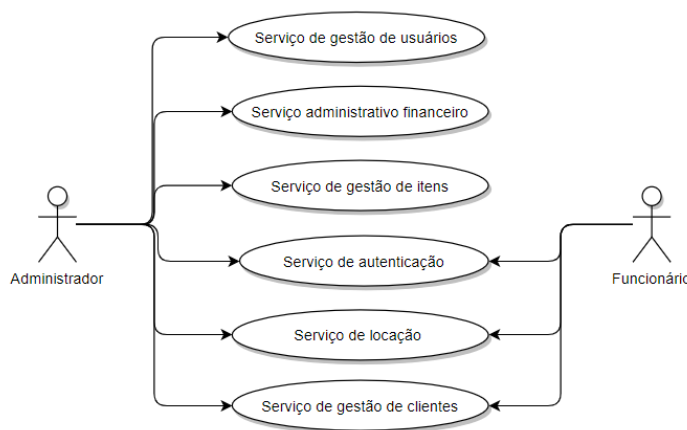


Figura 1: Diagrama de contexto



Figura 2: Hierarquia de atores

5 Especificação de Requisitos

5.1 Requisitos Funcionais

RF01– Cadastro de item no banco de dados **Informações:** Código do item, Nome do item, Valor, Descrição e status (alugado ou livre). **Regras:** O Administrador pode cadastrar novos itens inserindo cada um dos seus dados manualmente e ao enviar os itens são cadastrados na base de dados.

- Os itens possuem um código único.
- Descrição do deve ser opcional.

RF02– Visualização de item **Informações:** Os usuários podem ver as informações básicas dos itens e se eles estão alocados ou não. **Regras:** Pelo código do item o usuário consegue ver a informação detalhada de um determinado item e o seu status.

RF03– Remoção de um item no banco de dados. **Informações:** Código do item **Regras:** O sistema deve permitir que o administrador busque um item fornecendo seu código e, caso o item seja encontrado, ele é removido. Caso contrário, o sistema informa ao usuário que o item não existe e portanto, não pode ser removido.

- Os itens possuem código único.
- O sistema deve informar o usuário caso o item a ser removido não exista.
- O sistema deve deixar o cliente digitar novamente o código do item, caso a tentativa anterior de remoção tenha sido mal-sucedida.

RF04– Atualização de um item do banco de dados **Informações:** Código do item, Nome do item, Valor, Descrição e status (alugado ou livre). **Regras:** O sistema deve permitir que o administrador e o funcionário consulte um item a partir de um código. Caso o item seja encontrado, será possível editar os campos: Nome do item, valor, descrição e status.

- Os itens possuem código único.
- O sistema deve informar o usuário caso o item a ser atualizado não exista.

RF05– Criação de novos clientes **Informações:** Primeiro nome, Ultimo nome, cpf, email e telefone e itens alocados **Regras:** O sistema deve permitir o cadastro de novos clientes via inserção dos dados manual e depois disso os dados são inseridos na base de dados.

- O CPF do cliente tem que ser único
- Um usuário pode ter vários itens locados pra ele

RF06– Remoção de clientes **Informações:** cpf **Regras:** O sistema deve permitir que o administrador e o funcionário consultem um cliente a partir do cpf. Caso o cliente seja encontrado, ele é removido. Caso contrario, o usuário é informado que o cliente não existe no banco de dados

- O CPF do cliente tem que ser único

RF07– Atualização de Clientes **Informações:** Primeiro nome, Ultimo nome, cpf, email e telefone **Regras:** O sistema deve permitir que o usuário busque a partir do cpf um cliente. Caso o cliente seja encontrado, será possível alterar: Primeiro nome, Ultimo nome, cpf, email e telefone. Caso contrario, o sistema deverá informar que o cliente não se encontra

- O CPF do cliente tem que ser único

RF08– Visualização de um cliente **Informações:** Primeiro nome, Ultimo nome, cpf, email e telefone **Regras:** O sistema deve permitir ao administrador e o funcionário visualizar os dados do cliente, que são: Primeiro nome, Ultimo nome, cpf, email e telefone.

- Os dados devem estar dispostos na ordem especificada anteriormente

RF09– Locação de item para clientes. **Informações:** itens que estão alocados para um cliente. **Regras:** O sistema deve permitir que um funcionário aloque itens para um cliente acrescentando-os à lista de itens daquele cliente.

- um cliente pode ter vários itens alocados.

RF10– O sistema deve calcular automaticamente o valor da locação. **Informações:** Itens a ser alugado, valor dos itens a ser alugado, cliente que irá alugar, data de entrega do item. **Regras:** O sistema deve permitir que, ao fazer a locação de um item, possa ser calculado o valor a ser pago para o mesmo levando em conta o valor base do aluguel com adição do tempo (dias) que ficará alugado.

RF11– **Informações:** **Regras:**

RF12– O sistema deve permitir listar todos os itens existentes no estoque **Informações:** Itens a ser alugado, valor dos itens a ser alugado, cliente que irá alugar, data de entrega do item. **Regras:** O sistema deve permitir que o administrador e o funcionário obtenham uma lista de todos os itens que estão em estoque, ou seja, não estão alugados no momento da consulta

RF13– Busca de itens. **Informações:** Código do item. **Regras:** O sistema deve permitir que o usuário busque um determinado item pelo seu código e com isso os dados daquele determinado item apareçam na tela

RF14– O sistema deve gerar um relatório mensal até o momento dos itens alugados.

Informações: Itens a ser alugado, valor dos itens a ser alugado, cliente que irá alugar, data de entrega do item. **Regras:** O sistema deve fornecer ao administrador e o funcionário um relatório mensal de itens a ser alugados, contendo todos os campos relativos a cada item .

RF15– Criação de novos funcionários **Informações:** Primeiro nome, Ultimo nome, matrícula, login, senha **Regras:** O sistema deve permitir que o administrador crie um novo funcionário inserindo todos os seus dados

RF16– Remoção de funcionários **Informações:** matrícula **Regras:** O sistema deve permitir que o administrador remova o funcionário através de uma busca pela matrícula. Caso a matrícula seja encontrado, o funcionário é removido. Caso contrario, o sistema avisa ao usuário que o funcionário procurado não existe.

RF17– atualização dos dados do funcionário **Informações:** Primeiro nome, Último **Regras:** O sistema deve permitir que o administrador busque através da matrícula um funcionário. Caso o funcionário seja encontrado no banco de dados, o usuário poderá atualizar todas as informações do funcionário buscado.

RF18– Visualização de funcionários **Informações:** Nome, último nome, matrícula, login e senha. **Regras:** O sistema deve permitir que o administrador consiga visualizar os dados referentes aos funcionários, são eles: Nome, último nome, matrícula

5.2 Requisitos Não Funcionais

RNF01. Autenticação do usuário por meio de login e senha. **Informações:** usuário e senha. **Regras:** O usuário terá acesso ao sistema.

RNF02. Os funcionários terão acesso a um conjunto mínimo de funcionalidades, que compreendem funções operacionais relacionadas a itens e clientes. **Informações:** **Regras:** O funcionário terá acesso à locação, e CRUD de usuários.

RNF03. O administrador tem acesso a todas as funcionalidades do sistema. **Informações:** **Regras:** Acesso a todos os CRUDs e funcionalidades.

RNF04. O tempo de consulta de um item não pode extrapolar mais de 5 segundos. **Informações:** Código do item. **Regras:** Leitura dos dados de um item.

RNF05. O tempo de cálculo do aluguel não pode extrapolar mais de 10 segundos. **Informações:** Valor do aluguel do item. **Regras:** Leitura do valor da locação

RNF06. O sistema deve garantir a confiabilidade do banco na adição, atualização e remoção dos dados. **Informações:** Dados dos itens, dados dos clientes, dados dos funcionários. **Regras:** As informações permaneceram consistentes após as operações.

RNF07. O sistema deve fazer tratamentos de erro que sejam simples de entender. **Informações:** Capturar exceções cometidas pelo sistema **Regras:** O usuário será capaz de identificar qual erro foi encontrado facilmente.

6 Plano de VVT

6.1 Requisitos a serem testados

6.1.1 Controle de acesso dos administradores

- Verificar se o administrador tem acesso as funcionalidades

6.1.2 Validação do cadastro de itens

- Verificar o cadastro de itens
- Validar o funcionamento da busca de itens
- Verificar se a remoção de itens funciona
- verificar se todos os itens armazenados no mês constam no relatório

6.1.3 Validação cadastro de funcionários

- Verificar o cadastro de funcionários
- Validar a busca e listagem de funcionários
- Validar a remoção de funcionários

6.1.4 Validação cadastro de clientes

- Verificar o cadastro de clientes
- Validar o funcionamento da busca de cliente
- Verificar se a remoção de clientes funciona

6.2 Estratégias e ferramentas de teste

Serão realizados testes unitários, que tem como abjetivo verificar se as funções específica dos método para cada serviço do sistema. Para implementar os testes, a biblioteca PyUnit da linguagem de programação Python, será utilizada. Essa biblioteca faz parte da biblioteca padrão da linguagem, portanto não é necessário fazer a instalação de nenhum modulo. Além disso, o PyUnit é rápido e acessível em questão de implementação dos testes.

6.3 Equipe e infra-estrutura

Tarefa	Responsáveis
Descrever os casos de uso	Fabrício, Pedro e Rodrigo
Configurar as ferramentas de teste	Fabrício e Lucas
Implementar os testes	Fabrício e Lucas
Fazer correções	Fabrício e Lucas

6.4 Execução do Plano de Teste

6.4.1 Configuração das ferramentas de teste

- Instalar as ferramentas de teste
- Configurar o ambiente de teste
- Definir as funcionalidades a ser testadas

6.4.2 Implementar os testes

- Avaliar os métodos da biblioteca mais adequados
- escrever os testes baseado nos casos de uso
- executar os testes implementados
- avaliar os resultados dos testes

6.4.3 Fazer correções

- identificar qual caso de teste ocorreu o erro
- executar as correções necessárias
- volta a etapa de executar os testes

7 Medição e Qualidade de Software

7.1 Code Smells

Para fazer a detecção de Code Smells será utilizada a biblioteca Good Smell da linguagem de programação Python. Essa biblioteca tenta detectar trechos de códigos em que é possível deixar o código mais elegante, respeitando as boas praticas de programação. Os recursos utilizados para efetuar esse processo são da biblioteca padrão do Python, porém para fazer uso da Good Smell, é necessário fazer a instalação do modulo.

7.2 Pylint

O Pylint é uma ferramenta que verifica se há erros no código, tenta impor um padrão de codificação, busca por Code Smells, realiza sugestões sobre blocos que podem ser refatorados e oferecer detalhes sobre a complexidade do código. Indo mais a fundo sobre suas características, o Pylint realiza: verificação de comprimento do código de linha, se o nome das variáveis estão bem formados de acordo com seu padrão de codificação, se os módulos importados são usados, avalia se as interfaces declaradas são realmente implementadas, detecta código duplicado, possui ferramenta para criação de diagramas UML para código python.

8 Observações

9 Referências

- [1] Chapman, S.J. – Electric Machinery Fundamentals, 4th Edition;
- [2] Fitzgerald, A. E. – Máquinas Elétricas, 2da Edição;