

Universidade Federal de Ouro Preto



Engenharia de Software II

Sistema de *Pedidos de Restaurante*

Grupo: *As Chinchilas do Deserto*

Alunos:	Ana Luiza Almeida Soares Bernardo Brandão Freguglia Bruno José Baêta Barbosa Everton de Souza Kenedy Hugo Eduardo Ziviani Suleimane Ducure Ulisses Viana Taveira Wendel Francisco Furtado e Lima
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
5	Especificação de Requisitos	3
5.1	Requisitos Funcionais	3
5.2	Requisitos Não Funcionais	5
6	Plano de VVT	5
6.1	Requisitos a serem testados	6
6.2	Estratégias e ferramentas de teste	6
6.3	Equipe e infra-estrutura	6
6.4	Execução do Plano de Teste	6
7	Medição e Qualidade de Software	6
8	Observações	6
9	Referências	7

1 Histórico de Revisões

Data	Versão	Descrição	Autor
08/03/2021	0.0	Início da escrita do documento	Ana Luiza
08/03/2021	0.0	Justificativa do Processo Utilizado	Ulisses
08/03/2021	0.0	Escrita do Cronograma	Ana Luiza
14/03/2021	0.0	Diagrama de Casos de Uso	Ana Luiza e Suleimane
/03/2021	0.0	Atividade	Autor
/03/2021	0.0	Atividade	Autor

Tabela 1: Revisões do Documento

2 Processo e Software

O grupo as Chinchilas do Deserto escolheu como processo de software para o projeto de desenvolvimento do software de pedidos de restaurante o Modelo Incremental.

Sendo uma melhoria do Modelo Cascata, o Modelo Incremental trabalha com incrementos, ou seja, pequenos pedaços de software entregues de cada vez. Este modelo combina elementos do Modelo em Cascata aplicados de maneira iterativa, ou seja, de forma que o progresso aconteça através de sucessivos refinamentos, melhorados a cada iteração.

Cada incremento é desenvolvido de forma linear e exposto aos comentários dos clientes, sendo necessária alguma alteração na implementação é feito um novo incremento e mais uma vez ele é apresentado aos clientes. A cada aprimoramento é lançada uma nova versão, e todas as atividades de especificação, projeto, implementação e validação são intercalados, acontecendo em cada nova versão.

Esse feedback rápido e a implementação rápida de um software útil foram as principais características que fizeram o grupo optar por esse modelo. Além disso, a natureza da ferramenta desenvolvida, um Software de Pedidos, exige uma troca rápida de versões e um feedback rápido das versões em desenvolvimento pode ser feito com facilidade.

3 Cronograma

Segue na tabela a baixo um cronograma inicial das atividades que serão realizadas no projeto e seus respectivos responsáveis.

Nome	Tarefa	Prazo
Ana Luiza e Bruno	Diagrama de Casos de Uso	08/03 a 13/03
Bernardo	Descrição dos Cenários	08/03 a 13/03
Suleimane	Justificativa do Processo	08/03 a 10/03
Suleimane	Levantamento dos Requisitos	15/03 a 20/03
Everton e Ulisses	Descrição da Técnicas de Medição	22/03 a 01/04
Wendel	Implementação parcial do Software	15/03 a 05/04
Hugo	Primeira apresentação	01/04 a 05/04
Todos os participantes	Segunda apresentação	05/04 a 08/04
Todos os participantes	Última apresentação	08/04 a 12/04

Tabela 2: Cronograma

4 Levantamento de Requisitos

Independente da metodologia de desenvolvimento utilizada, o levantamento de requisitos é o ponto de partida de qualquer projeto de software, pois é a partir dos resultados obtidos durante esta etapa que será possível definir como as próximas etapas do desenvolvimento serão executadas.

Para o levantamento de requisitos do seguinte projeto, foi utilizado o levantamento orientado a pontos de vista. O método VORD (viewpoint-oriented requirements definition – definição de requisitos orientada a ponto de vista) consiste em definir os diferentes usuários do sistema como pontos de vista e descrever as ações que esses usuários irão realizar. A técnica é mais adequada devida às condições em que o trabalho está inserido e ao formato em que vai ser desenvolvido, pois é um processo que pode ser feito sem o contato com o cliente, já que não existe um cliente real. Além disso, a técnica permite a visualização completa das funções do sistema e das relações entre usuários, e pode prever possíveis conflitos no sistema. Etapas:

1. Identificação do Pontos de vista:
 - Responsável pelo restaurante
 - Cliente do restaurante
2. Estruturação e documentação dos Pontos de Vista: O ponto de vista do cliente é superior hierarquicamente ao ponto de vista do restaurante. O cliente executa as ações principais, enquanto o restaurante executa ações de manutenção e validação. Ações executadas pelos pontos de vista:

- Responsável pelo restaurante: autorizar ou negar cadastro, autorizar ou negar pedido, editar o cardápio;
- Cliente do restaurante: fazer cadastro e/ou login, pesquisar e selecionar pratos, editar informações do cadastro.

Para demonstrar as ações dos nossos Pontos de Vista, utilizamos o Diagrama dos Casos de Uso colocando-os como autores.

O Diagrama dos casos de uso é uma importante ferramenta no processo de desenvolvimento de software pois esquematiza os papéis de cada usuário e as funções do software. Como o software a ser desenvolvido é bem simples, apresentamos abaixo o diagrama referente a ele. Tendo como atores o cliente do restaurante (Ator Primário) e o responsável pelo restaurante (Ator Secundário) e ações:

- Visualizar cardápio: O cliente pode visualizar os pratos disponíveis;
- Selecionar pratos: O cliente pode selecionar os pratos escolhidos por ele;
- Fazer Cadastro / Login: O cliente pode fazer um cadastro inicial ou se tiver cadastro pode logar na conta do mesmo;
- Fazer pedido: O cliente pode, após selecionados os pratos, finalizar o pedido;
- Editar Cardápio: O restaurante pode adicionar e retirar pratos do cardápio, assim como alterar os preços;
- Validar Cadastro / Login: O restaurante deve validar os cadastro do cliente;
- Aprovar/ Negar pedido: O restaurante pode aceitar, ou não, o pedido do cliente;

5 Especificação de Requisitos

5.1 Requisitos Funcionais

Requisitos Funcionais são assuntos de importância fundamental ou essencial ao produto. Eles descrevem o que o produto tem de fazer ou que ações processuais deve tomar.

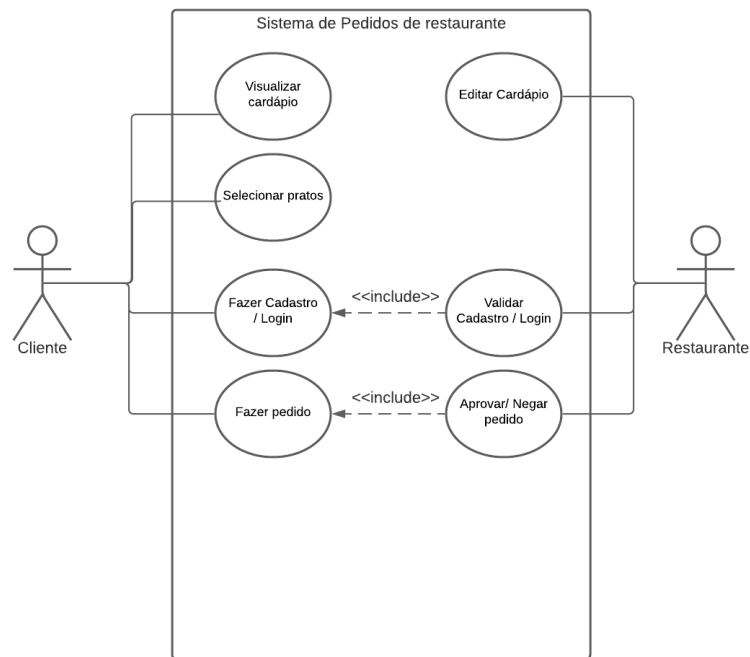


Figura 1: Diagrama de Casos

RF01– O software deve mostrar os pratos disponíveis para venda no dia da consula, assim como os preços e a descrição do prato para que o usuário escolha o(s) prato(s) desejado(s). **Informações:** Nome do Refeição, Imagem, Descrição dos itens contidos no prato, Quantidade de porções e Preço. **Regras:** O sistema deve permitir que o visitante ou o cliente do restaurante pesquise os pratos por nome ou simplesmente percorra a lista disponível para aquele dia. Deve permitir também que o usuário selecione o prato desejado e informe a quantidade desejada, sendo limitada até 5 unidades. Caso não haja disponibilidade do prato procurado, deve ser informado ao cliente que não houve resultado para a consulta. A obtenção dos dados de pratos disponíveis deve ser disponibilizada e atualizada pelo restaurante.

- A quantidade de itens selecionados deve ser de 0 a 5 unidades
- O usuário pode escolher entre todas as opções disponíveis
- O usuário pode olhar o cardápio e selecionar os itens sem se cadastrar no sistema.

RF02– O software deve, após a escolha dos pratos pelo usuário, listar os pratos escolhidos e respectivas quantidades. Também deve mostrar as

informações do usuário para a realização da entrega. **Informações:** Nomes dos Pratos, Quantidades, Preço de cada refeição, Preço Total, Endereço do cliente (rua, número, bairro, complemento, referência), Forma de pagamento (com opção para troco), Nome do cliente.

- O usuário pode modificar seu dados quantas vezes desejado, sendo esses dados o Endereço e o Nome
- O usuário pode muda a forma de pagamento quantas vezes quiser
- O usuário pode editar a lista de refeições: excluir ou adicionar um prato, aumentar ou diminuir a quantidade de porções
- Caso o usuário não seja cadastrado, é informado ao usuário a necessidade do cadastro e aparece a opção para o feito

RF03—
(Etc...)

5.2 Requisitos Não Funcionais

Requisitos Não Funcionais são as propriedades que as funções devem ter, tais como desempenho e usabilidade. Não se detenha ao seu nome pouco apropriado (nós o usamos porque é a maneira mais comum de se referir a estes tipos de requisitos)—estes requisitos são tão importantes quanto as exigências funcionais, para o sucesso do produto.

RNF01. A efetivação da reserva do pacote, só deve ser liberada após o cliente estar logado no sistema. **Informações:** usuário e senha. **Regras:** o cliente terá acesso para comprar, consultar e alterar.

ETC

6 Plano de VVT

Asseguram que o software cumpra com suas especificações e atenda às necessidades dos usuários. Você deve apresentar um plano de testes, ferramentas que serão utilizadas e coisas do tipo.

Veja um exemplo no link: https://www.cin.ufpe.br/~gta/rup-vc/extend.formal_resources/guidances/examples/resources/test_plan_v1.htm

6.1 Requisitos a serem testados

Esta seção descreve em linhas gerais o conjunto de requisitos a serem testados no projeto a ser desenvolvido, comunicando o que deve ser verificado. Exemplos de requisitos a serem testados são: desempenho, segurança, interface de usuário, controle de acesso, funcionalidades.

6.2 Estratégias e ferramentas de teste

Apresenta um conjunto de tipos de testes a serem realizados, respectivas técnicas empregadas e critério de finalização de teste. Além disso, é listado o conjunto de ferramentas utilizadas.

6.3 Equipe e infra-estrutura

Contém descrição da equipe e da infra-estrutura utilizada para o desenvolvimento das atividades de testes, incluindo: pessoal, equipamentos, software de apoio, materiais, dentre outros. Isto visa garantir uma estrutura adequada para a execução das atividades de testes previstas no plano.

6.4 Execução do Plano de Teste

7 Medição e Qualidade de Software

Apresente aqui o formato da Medição e qualidade de software. Você deve mostrar os meios que irá avaliar a qualidade do seu software. Apresente o plano e os resultados a partir da prática de ferramentas de detecção de code smells, por exemplo. Em Java, temos uma ferramenta chamada JDEODORANT. Você pode avaliar as métricas de qualidade também, por exemplo, em Java, temos CKMetrics¹

8 Observações

Apresente aqui as dificuldades na disciplina, trabalho prático e coisas do tipo.

¹<https://github.com/mauricioaniche/ck>

Funcionalidade: Envelope		Tempo Despendido(h): 1h
Contador: 02		Criticidade: Baixa

Objeto de Teste:		
<ul style="list-style-type: none"> ➤ Validar uso do assistente para criar documento no modelo Envelope. ➤ Verificar impressão usando um envelope. 		
Descrição do Caso de Teste:		
<ul style="list-style-type: none"> ➤ O aplicativo deverá atender corretamente a configuração fornecida pelo usuário (nas 3 abas: Envelope, Formato e Impressora). ➤ A impressão no envelope deverá ser realizada com sucesso. 		
Pré-Condição:		
<ul style="list-style-type: none"> ➤ O usuário deverá informar dados do Destinatário e Remetente. ➤ Existir uma impressora compatível para impressão no envelope. 		
Dados de Entrada:		
ID	Passo	Procedimento
1	P1	Executar o aplicativo LibreOffice 4.2 (Opção texto)
	P2	Selecionar a opção Texto
	P3	Acessar o menu a opção Envelope (Inserir>>Envelope)
	V1	O aplicativo deverá exibir a tela do assistente para Envelope
	P3	Preencher os dados do campo Destinatário e Remetente
	P4	Clicar no botão Inserir
	P5	Se necessário ajustar a largura da caixa de texto do Destinatário e Remetente
2	P7	Clicar no ícone da Impressora
	V2	Verificar se o envelope foi impresso corretamente.
Resultado Esperado: As operações deverão funcionar corretamente cumprindo todas as regras acima citadas.		

Figura 2: Exemplo

9 Referências

- [1] Chapman, S.J. – Electric Machinery Fundamentals, 4th Edition;
- [2] Fitzgerald, A. E. – Máquinas Elétricas, 2da Edição;
- [3] <https://www.devmedia.com.br/tecnicas-para-levantamento-de-requisitos/9151>