* **Modelo de negócio**

Good Trip é uma marca de roupas e acessórios que contempla artigos de diversos estilos. As peças são feitas de maneira artesanal e o proprietário não armazena dados de estoque e fornecedor. Atualmente não existe outra maneira de comercialização dos produtos senão a presencial. Os produtos são distribuídos para terceiros, estes, após a comercialização, repassam de volta o valor acordado pelas peças.

Há também a possibilidade de fazer encomendas de peças com pagamento adiantado. As roupas são categorizadas da seguinte forma: camisa, camiseta regata, blusão, casaco. Já os acessórios são: boné, colar, pulseira, relógio, bolsa. Não são armazenados informações sobre estoque, tampouco de fornecedores de insumos.

* **Solicitação de Software**

A Good Trip deseja uma aplicação web que seja capaz de gerenciar operações de venda diretamente para os clientes.

É necessário que haja um cadastro das peças de roupa e acessórios, onde as mesmas devem possuir como atribuições a categoria, nome da peça, custo, preço, tamanho, data de confecção. Os itens vendidos deveram ser registrados em uma nota fiscal cotendo as seguintes informações: comprador, data da venda, valor total da compra, forma de pagamento, além de informações dos itens vendidos.

As vendas devem ter seu tempo de entrega calculado de acordo com o CEP do cliente, ser registradas em notas fiscais.

Os pratos também possuem cadastro e são identificados por código do prato, nomenclatura do prato, tipo do prato, preço do prato e TMP (Tempo médio de produção).

O usuário com o perfil de garçom é responsável por abrir o pedido e vincula-lo a uma mesa. O garçom é responsável pela abertura, alteração e encerramento dos pedidos e das mesas, assim como é responsável por emitir a nota de pagamento e processar o pagamento com o cliente.

O perfil de usuário cozinheiro é responsável por receber os pedidos assim que confirmados pelo garçom e será possível trabalhar na produção e entrega dos pratos, salientando que os pedidos nesse momento assumirão o formato de comanda na tela do cozinheiro e receberá a coloração de acordo com o prazo de entrega que é relacionado ao TMP.

Devera existir também o usuário do cliente onde serão cadastrados: cpf, nome do cliente e telefone de contato e ranking (para identificar assiduidade) opcional. A partir desse cadastro o usuário terá acesso ao ambiente onde ele poderá fazer o seu pedido e acompanhar os seus bônus, resultante do cartão fidelidade.

O perfil administrador tem acesso total às funcionalidades do software, tais como: acompanhamento de produtividade da cozinha, acompanhamento de lucros, gerenciamento da base de dados de seus clientes, fácil visualização de produtos com maior e menor movimentação.

**Glossário**

Garçom: Responsável por registrar os pedidos do cliente no sistema.

Gestor: Usuário do sistema que tem acesso geral do mesmo.

Cliente: Usuário que faz o pedido.

Mesa: Lugar onde o cliente vai ficar e meio de marcação para o pedido.

Pedido: Especificações do prato.

Prato: Conjunto de alimentos.

Cozinheiro: Recebe na tela todos os pedidos que são enviados pelo garçom.

<NOME DA FÁBRICA DE SOFTWARE>

Good Trip

Documentação de Software

Versão: 1.0

Data: 25-FEV-19

Identificador do documento: LR

Versão do Template Utilizada na Confecção: 1.0

Localização: <caminho de acesso no CVS ou URL>

**Introdução**

* 1. Propósito

Este documento especifica os requisitos dos sistemas a serem desenvolvidos pela <Nome da fábrica de software>, fornecendo aos desenvolvedores as informações necessárias para o projeto e implementação, assim como para a realização dos testes de homologação do sistema.

* 1. Público Alvo

Este documento se destina aos arquitetos de software, engenheiros de software e testadores.

* 1. Escopo

Este documento realiza a elicitação de requisitos de um determinado sistema, Muraki App.

1.4 Definições e Abreviações

Garçom: Responsável por registrar os pedidos do cliente no sistema.

Gestor: Usuário do sistema que tem acesso geral do mesmo.

Cliente: Usuário que faz o pedido.

Mesa: Lugar onde o cliente vai ficar e meio de marcação para o pedido.

Pedido: Especificações do prato.

Prato: Conjunto de alimentos.

Cozinheiro: Recebe na tela todos os pedidos que são enviados pelo garçom.

1.5 Referências

<Esta seção é destinada à descrição das referências utilizadas pelo documento, como por exemplo. URLs e livros. Ver exemplos a seguir:

“Glossário da Usina”, <id\_doc glossário>, Versão: <versão>. Localização: <localização>”>

1.6 Visão geral do documento

**Na seção 2** apresenta uma visão geral do sistema, caracterizando qual é o seu escopo e descrevendo seus usuários.

**A seção 3** especifica as premissas e restrições dos requisitos leventados

**Na seção 4** são enumerados todos os requisitos funcionais, e

**Na seção 5** os não-funcionais do sistema.

**Visão Geral do Produto**

< Descreve o objetivo do sistema, suas respectivas funcionalidades, qual o público alvo do sistema, qual a necessidade de implementar o produto, o impacto do sistema e sucesso que a solução irá trazer.>

2.1 Descrição dos usuários

<Descreve quais os usuários finais do sistema.>

**Premissas e restrições**

Premissa 1: O produto deve ser entregue em no máximo 3 meses após o início de seu desenvolvimento.

Premissa 2: Qualquer alteração no planejamento inicial do projeto deve ser comunicada previamente ao cliente.

Premissa 3: O software deve atender a todos os requisitos funcionais e não funcionais previamente acordados entre as partes envolvidas.

Premissa 4: A equipe da fábrica de software é responsável pelo treinamento dos técnicos para o uso adequado da ferramenta.

Premissa 5: Deve ser entregue, constituindo o produto final, além do software, manual de usuário e manual de sistema.

Premissa 6: O custo limite do projeto deve ser de no máximo R$2.000,00.

Premissa 7: Qualquer alteração no custo final do projeto deve ser comunicada imediatamente ao setor financeiro.

**Requisitos Funcionais**

4.1 RF001 Manter Cliente

O sistema deverá permitir que os usuários realizem inclusão, alteração, consulta e exclusão de clientes.

4.2 RF002 Manter Garçom

O sistema deverá permitir que os usuários realizem inclusão, alteração, consulta e exclusão de garçons.

4.3 RF003 Manter Gestor

O sistema deverá permitir que os usuários realizem inclusão, alteração, consulta e exclusão de gestores.

4.4 RF004 Manter Pedido

O sistema deverá permitir que os usuários realizem inclusão, alteração, consulta e exclusão de pedidos.

4.5 RF005 Manter Comanda

O sistema deverá permitir que os usuários realizem inclusão, alteração, consulta e exclusão de comandas.

**Requisitos Não Funcionais**

1. Segurança: O sistema web deve ser somente acessado pelos usuários previamente cadastrados. Os dados usados no sistema não devem ser armezados em ambiente local. Os usuários devem usar credenciais de no mínimo 8 caracteres, contendo pelo menos um número e um símbolo.

2. Desempenho: O sistema deve suportar múltiplas requisições, os usuários devem poder fazer uso livremente do sistema sem atrasos maiores que 10s para gerar novos pedidos.

3. Usabilidade: O software deve manter o estado dos pedidos caso seja encerrado por engano.

4. Confiabilidade: O software deve manter o estado dos pedidos caso seja encerrado por engano. O valor total da nota fiscal deve ser exatamente o correspondente ao soma dos itens solicitados.

5. Padrões: A metodologia de desenvolvimento da equipe deve ser ágil. O sistema web deve ser programado usando as seguintes tecnologias: Angular 2, PHP, base dados da Microsoft.

6. Hardware e Software: O sistema web deve funcionar plenamente no seguintes sistemas: Windows, Linux, Mac OS.

5.1 RNF001 Acessibilidade

O sistema web deve ser acessado apenas por usuários previamente cadastrados no sistema.

5.2 RNF002 Persistência de dados

Os dados armazenados em ambiente local devem ser transferidos para base de dados online.

5.3 RNF003 Quesito de segurança

Os usuários devem possuir senhas com pelo menos 8 caracteres, sendo um símbolo e um número.

5.4 RNF004 Desempenho do sistema

Os pedidos deve ser processados em até 10 segundos.

5.5 RNF005 Armazenamento de dados

O último estado das informações do sistema devem ser carregados, caso a ordem de serviço ainda não tenha sido finalizada.

5.6 RNF006 Corretitude de valores

O valor total da ordem de serviço deve ser exatamente a soma dos pedidos.

5.7 RNF007 Metodologia de desenvolvimento

A equipe de desenvolvimento deve adotar o a padrão ágil.

5.8 RF008 Linguagem de programação

O software deve ser implementado em linguagem de programação C#.

5.9 RF009 Base de dados

O software deve fazer uso da base de dados da Microsoft.

**Diagramas**