



PROJETO T.A.L.G.S

Tia Ana Lanches Gerencial System

Grupo 3 – Projeto Integrador

Eduardo Rodrigues Estrela
Luis Fernando Soares de Oliveira
Renan Eduardo da Silva Souza

Índice

Capítulos e Subseções

1. Sobre o Sistema	4
1.1. Descrição do Sistema T.A.L.G.S	4
1.2. Entrevista	5
2. Requisitos Funcionais	6
2.1. Termos Recorrentes	7
2.2. Requisitos Funcionais do Sistema	8
3. Requisitos Não Funcionais	11
3.1. Termos Recorrentes	11
3.2. Requisitos Não Funcionais do Sistema	12
3.3. Links Relacionados	15
4. Protótipos	16
4.1. Back-End	16
4.1.1. Diagrama de Classes	16
4.1.2. Diagrama de Atividades UML	16
4.2. Banco de Dados	28
4.2.1. Diagrama Relacional	28

Capítulo 1

1.1 – Sobre o Sistema

- O que é?

O Sistema T.A.L.G.S (Tia Ana Lanches Gerencial System) será usado para o gerenciamento de vendas, estoque e pedidos de uma lanchonete, propriedade de Ana Kátia, localizada em Planaltina - DF, e a ideia principal é criar um sistema de fácil utilização, ultraleve, responsivo e intuitivo.

- O que fará?

O sistema será responsável pela organização e indexação dos estoques, datas de validade e pedidos de abastecimento de estoque, bem como criar e visualizar registros de compras e vendas de produtos e adicionar, remover e editar dados dos produtos que constam no sistema.

- Onde e quando será usado?

O sistema será implementado por meio de uma interface para web, podendo ser acessado de qualquer dispositivo remotamente.

Ele deverá ser usado quando a cliente, por necessidade de organização, realizar operações de compra e venda de produtos, atualização do estoque, adição ou subtração de produtos do catálogo. Em resumo, sua utilização ocorre em momentos em que a cliente deveria guardar, de forma manual e lenta, informações sobre seus produtos, vendas e operações do gênero, para a automação de alguns desses processos, e para a facilitação de outros.

- Existem sistemas que fazem o mesmo?

É notável a existência de sistemas que realizam todas as operações acima descritas, porém, nenhum deles executa todas as funções acima listadas, apenas uma parte delas. Outra característica deles é a implementação por meio de aplicativos locais escritos em linguagens específicas à um sistema operacional ou aparelho eletrônico específicos.

- O diferencial do nosso sistema

O sistema T.A.L.G.S. possui alguns diferenciais dos sistemas já existentes no campo da automação de vendas. Alguns desses diferenciais podem ser listados:

- Acesso multiplataforma por meio da Web
- Atualização de produtos em tempo real
- Gerenciamento de pedidos integrado
- Rápido, otimizado e intuitivo
- Gratuito

1.2 – Entrevista

A cliente do sistema foi entrevistada o dia 11/09/2023, para o levantamento de funcionalidades requeridas do sistema. As seguintes perguntas foram feitas durante a entrevista:

- Quais as funcionalidades o sistema deverá possuir?
- Como seria o design final do produto?

Algumas necessidades também foram estipuladas diretamente pela cliente:

- Fácil utilização
- Necessidade de autenticação para gerenciamento de operações
- Segurança dos dados do sistema

Após a entrevista, o que se pode concluir das operações básicas e essenciais do sistema foi:

- O sistema deve ser capaz de receber dados de produtos por parte do usuário, armazená-los em um banco de dados, e ser capaz de recuperar / acessar esses dados.
- O sistema também deve ser capaz de gerir o estoque desses produtos, avisando com uma antecedência customizada pelo próprio usuário atendente quando o mesmo de um determinado produto estiver em falta, bem como se um pedido já foi feito.
- É também necessária a refatoração automática da quantidade de produtos no banco de dados, uma vez que uma venda for efetuada, um estoque for atualizado ou um produto for perdido, sendo necessária uma advertência para o sistema e parâmetros para a realização do mesmo.
- O sistema requer uma funcionalidade de autenticação de usuários, para garantir a segurança dos dados armazenados.

Uma lista completa com os requisitos funcionais e não funcionais foi compilada e se encontra nos capítulos 2 e 3 respectivamente.

Capítulo 2

2.1 – Termos recorrentes

Esta subseção trata de termos que o leitor irá encontrar (não só, mas também) por todo este capítulo que possuem significados implícitos ou desconhecidos ao leitor comum, tratando da seção de Requisitos Funcionais.

1) - Dados: Define-se como dados de produtos neste sistema os dados referentes ao:

- a) Nome
- b) Número de identificação
- c) Preço (sendo preço os valores de compra, revenda e lucro obtidos por venda de produto)
- d) Quantidade de vendas em X espaço de tempo (sendo X um valor definido em dias, meses ou semanas)
- e) Quantidade em estoque do produto

2) Banco de dados: Define-se como banco de dados um sistema de banco de dados, de preferência relacional, gerido por um SGBD que guarda as diversas informações do sistema, pode ser acessado e atualizado de acordo com as regras definidas pelos desenvolvedores e pelas necessidades requeridas pelo cliente final.

2.2 – Requisitos Funcionais

Leia-se RF [XX] como Requisito Funcional [Numerador].

RF 01 - Capacidade de armazenar e recuperar dados dos produtos:

- Capacidade de receber entrada de dados e armazená-los dentro de um banco de dados integrado com o sistema.
- Capacidade de editar um ou mais produtos simultaneamente para alterar dados específicos.
- Capacidade de alterar, em tempo real ou próximo ao tempo real, a demonstração de dados graficamente nas páginas do sistema mediante alteração deles no sistema.

RF 02 - Recuperar os dados dos produtos:

- Capacidade de acessar entradas no banco de dados integrado do sistema.
- Capacidade de representar graficamente, por meio do documento HTML, dados que se encontram no banco de dados do sistema
- Necessidade de uma ou mais áreas reservadas no documento HTML para saída de informações dos produtos no banco de dados.

RF 03 - Capacidade de pesquisar produtos e pedidos no banco de dados:

- Capacidade de receber parâmetros de busca e recuperar os dados de itens com base na informação desejada dentro do banco de dados integrados ao sistema.
- Necessidade de um elemento de formulário para campos de pesquisa.
- Necessidade de um escopo mínimo de parâmetros para a pesquisa de itens no banco de dados.
- Necessidade de parâmetros de pesquisa mais abertos para facilitar a pesquisa de itens no banco de dados, permitindo uma melhor usabilidade de um único elemento de input.

RF 04 - Armazenar e acessar dados da entrada e saída de produtos do estoque:

- Necessidade de um banco de dados separado que armazena informações do estoque dos produtos, bem como informações detalhadas sobre tanto entrada quanto saída de estoque.
- Necessidade de um subsistema dentro do sistema capaz de refatorar o estoque dentro do banco de dados de acordo com entrada ou saída do estoque.
- Necessidade de rastreamento de quantidades plurais de um mesmo produto com mesmas porções, e datas de validade com valores próximos.

RF 05 - Capacidade de criar, gerenciar, editar e excluir produtos do sistema:

- Necessidade de um banco de dados separado que armazena os dados de identificação dos produtos, sem suas informações de estoque.
- Necessidade de um subsistema dentro do sistema maior capaz de gerenciar e editar os dados de identificação dos produtos.

RF 06 - Capacidade de gerar, gerenciar, editar e excluir pedidos de produtos remotamente pelos clientes:

- Necessidade de um banco de dados à parte que armazena os dados de pedidos dos clientes consumidores do estabelecimento, bem como seus status (em andamento, entregue, cancelado).
- Necessidade de um subsistema dentro do sistema maior capaz de gerenciar e editar os dados de pedidos de produtos, e possui a capacidade de gerenciar automaticamente os status de pedidos de acordo com ações dos dois usuários do sistema (usuário atendente e usuário consumidor).
- Capacidade de requisitar pedidos personalizados pelo sistema.
- Capacidade para permitir a edição e cancelamento de pedidos pelo próprio usuário consumidor.

RF 07- Capacidade de configurar avisos por produto em caso de baixa de estoque:

- Necessidade de um subsistema, configurado manualmente e a cada caso pelo usuário capaz de identificar quando um estoque está abaixo do ideal indicado por este, e que o avise sobre o estoque em baixa de determinado(s) produto(s).
- Necessidade de configuração manual do usuário atendente.

RF 08 - Capacidade de configurar avisos por produto em caso de prazo de validade próxima:

- Necessidade de um subsistema capaz de identificar quando uma quantia plural de um mesmo produto está próxima de perder sua validade, e que avise ao usuário atendente sobre tal.

RF 09 - Necessidade de sistemas de autenticação:

- Necessidade de um subsistema capaz de identificar usuários mediante apresentação de um dado único (como CPF ou nome de usuário) e uma senha a ser definida pelo próprio usuário.
- Necessidade de um banco de dados à parte capaz de armazenar os dados dos usuários do sistema, como nome, senhas (de preferência criptografadas), identificadores e compras associadas.

RF 10 - Necessidade de hierarquia de acesso às informações do sistema:

- Necessidade de hierarquização dos diferentes tipos de usuários do sistema em si, como usuários atendentes e usuários consumidores, para evitar acesso e/ou modificação indevida de dados no sistema.
- Necessidade de implementação de medidas de segurança para acesso e modificação de dados por diferentes tipos hierárquicos de usuários.

Capítulo 3

3.1 – Termos recorrentes

Esta subseção trata de termos que o leitor irá encontrar (não só, mas também) por todo este capítulo que possuem significados implícitos ou desconhecidos ao leitor comum, tratando da seção de Requisitos Não Funcionais.

1) - Framework: Segundo a Enciclopédia do Wikipédia, a definição de framework se encontra a seguir:

“Um framework em desenvolvimento de software, é uma abstração que une códigos comuns entre vários projetos de software provendo uma funcionalidade genérica.”

Fonte: <https://pt.wikipedia.org/wiki/Framework>

- 2) - Autenticação: Defina-se como autenticação o processo de recolher uma identificação única e uma senha do usuário, compará-las com as que se encontram registradas no sistema, e autorizar, caso a comparação seja sucedida e positiva, o usuário para a aplicação.
- 3) - Requisições: Defina-se como requisições os processos de um cliente HTTP de enviar requisições dos tipos GET e POST e receber respostas do servidor HTTP.
- 4) - Requisições HTMX: Defina-se as requisições do Framework HMTX como processos em que o cliente, por meio do corpo do documento HTML faz requisições de outras páginas HTML automaticamente por meio de gatilhos, e o servidor retorna informação de acordo com o método selecionado em formato de HTML.

3.2 – Requisitos Não-Funcionais

Leia-se RF [xx] como Requisito Funcional [Numerador]

RNF 01 - O sistema precisa ser acessível por dispositivos móveis e computadores.

- Essa necessidade se dá pela imprevisibilidade das circunstâncias de trabalho do cliente final.
- Essa necessidade será suprida utilizando uma página dinâmica para Web como forma de visualização e manipulação do sistema.

RNF 02 - O sistema deverá processar várias requisições simultâneas com o escopo AJAX (Asynchronous Javascript And XML).

- Isso se deve à necessidade de acesso assíncrono e simultâneo de múltiplos usuários do sistema, caso necessário.

RNF 03 - O sistema não apresentará a um cliente externo quaisquer dados de cunho considerado privativo.

- Para propósitos de segurança dos dados, e implementando camadas de acesso dos dados do cliente, os dados só poderão ser acessados uma vez que o usuário com credenciais no sistema se autentique e requirite.
- A autenticação necessária para acesso (implementada a partir do Framework Django) possui métodos nativos e confiáveis de criptografia para armazenamento de senhas.

RNF 04 - O sistema deverá estar funcionalmente disponível para uso em no máximo um minuto a partir de sua inicialização.

- Devido à alta imprevisibilidade da linha de trabalho do cliente, o sistema deve possuir a capacidade de estar em estado funcional em questão de segundos, e manter esse estado até segunda ordem do cliente.

RNF 05 - O sistema deverá implementar camadas de acesso de dados a partir do sistema web.

- A página web, deve ser capaz de não apresentar certos tipos de dados considerados privativos (tais como lucro sobre produtos, pedidos para estoque, entre outros) caso o requisitante não esteja autenticado no sistema como um funcionário.
- Porém, nem todas as informações devem ser privadas do acesso externo, tais como contato da loja, informações de localização física, informações sobre produtos, etc.

RNF 06 - O sistema será desenvolvido com as linguagens Python 3, HTML, CSS, Javascript, JQuery (Framework de Animações de Front-End), Jinja 2 e utilizando os frameworks Django (Python), HTMX (Javascript com injeção em HTML) e Bulma CSS (Framework para o desenvolvimento de CSS).

- A linguagem Python, versão 3, será predominante no sistema, será usada para a utilização e configuração do framework Django em sua totalidade, bem como para a definição de ambientes virtuais para Python que guardam localmente as dependências do sistema, para que não haja a necessidade de pré-instalação no momento de utilização do sistema.
- HTML será a linguagem utilizada para estruturar visualmente o sistema web.
- CSS será utilizado para estilizar uma parcela do conteúdo da página.
- Javascript será utilizado em suma para animações da página e lidar com poucos eventos dentro do sistema web.
- JQuery será utilizado para animar as páginas.
- Jinja, versão 2 é uma linguagem dependente de Python, que será utilizada para dinamizar as páginas HTML com a injeção de conteúdo

presente no banco de dados, utilizando de pontes (ou Middleware) do Framework Django, diretamente no documento HTML.

- O framework Django será utilizado, devido à sua completude de ferramentas e métodos, para o gerenciamento e conexões entre estes mesmos.
- O framework HTMX será utilizado para ampliar a dinamização do sistema web, tornando-o essencialmente uma aplicação de página única (single page application). O HTMX é responsável por permitir que qualquer tag HTML (apenas tags, não meta-tags como “html”, “meta”, “link”, etc) responsável por um elemento possa fazer requisições AJAX, como GET e POST, apenas antes disponível para hyperlinks e formulários, mas ampliando também a quantidade de requisições utilizáveis.
- O framework Bulma será utilizado na estilização de alguns componentes que já possuem um layout pronto, e para melhorar a responsividade dos elementos em diferentes telas de dispositivos.

RNF 07 - Realizar treinamento com os usuários, sobre as funcionalidades do sistema.

- Para a eliminação/mitigação de problemas quanto ao uso do sistema, um treinamento deverá ser realizado, para explicação do sistema e seu funcionamento.
- Treinamento pode significar apenas uma única sessão, ou uma bateria de aulas, a depender da necessidade de tal.

RNF 08 - Disponibilizar manuais de uso do aplicativo junto com a aplicação, acessível de modo offline pelos usuários do sistema.

- Toda distribuição do sistema deverá possuir um manual, que informa aos usuários sobre o uso do sistema, do mesmo modo que o treinamento faria, disponível remotamente junto com o sistema.

RNF 09 - O sistema deve ter a possibilidade de ser iniciado a qualquer momento do dia, sem restrição quanto à conexão com a internet.

- O framework Django pode ser iniciado, tanto como host local quanto como um site para web, sem a necessidade de conexão com a internet.

- O framework Django pode ser configurado para usar uma URL ou endereço IP específico, bem como a porta TCP que será utilizada, o que o torna bem prático na hora de utilizar o sistema, mesmo que ainda necessite de um servidor DNS externo.

3.3. Links Relacionados

- HTMX: <https://htmx.org/>
- Python: <https://www.python.org/>
- Django: <https://www.djangoproject.com/>
- JQuery: <https://jquery.com/>
- Bulma CSS: <https://bulma.io/>

Capítulo 4

Esta seção do documento irá apresentar diversos diagramas visuais, com o intuito de demonstrar visualmente e de modo mais simples o funcionamento do sistema e seus componentes. Os diagramas poderão ser visualizados por meios externos à visualização deste documento.

4.1. Back-End

4.1.1. Diagrama de Classes UML:

- [Clique aqui para visualizar](#)

4.1.2. Diagrama de Atividades:

- [Clique aqui para visualizar](#)

4.2. Banco de Dados

4.2.1. Diagrama Entidade - Relacionamento

- [Clique aqui para visualizar](#)