



U N I V E R S I D A D E
LUSÓFONA
D O P O R T O

GESTÃO DE ESTOQUE HIPERMERCADO

Gabriel Faria de Souza Lima- 21800381

Docente: Pedro Stretch

Engenharia de Software

SUMÁRIO

Capítulo 1:

Introdução.....	6
------------------------	----------

Descrição dos objetivos do documento.....	6
---	---

Público a que se destina.....	6
----------------------------------	---

O propósito do projeto a desenvolver.....	6
---	---

Capítulo 2:

Âmbito do projeto.....	6
-----------------------------------	----------

apresentação do projeto.....	6
---------------------------------	---

Decomposição funcional do sistema.....	7
--	---

Constrangimentos do processo.....	7
-----------------------------------	---

Capítulo 3:

Calendarização.....	7
----------------------------	----------

Enumeração dos marcos a atingir(milestones).....	7
--	---

Identificação das tarefas a realizar.....	8
---	---

Identificação dos artefactos a desenvolver.....	9
---	---

Calendarização do projeto.....	9
-----------------------------------	---

Capítulo 4:

Organização da Equipa.....	10
---------------------------------------	-----------

Equipa e tipo de organização.....	10
--------------------------------------	----

Membros da equipa.....	10
---------------------------	----

Lista das funções/competências necessárias.....	10
---	----

tarefas/funções de cada membro da equipe.....	10
---	----

Capítulo 5:

Descrição técnica dos sistemas propostos.....	11
Requisitos	
Funcionais.....	11
Requisitos	
não-funcionais.....	14
Descrição da estrutura física do hardware.....	15
Descrição da interface	
gráfica.....	17

Capítulo 6:

Standards, técnicas e ferramentas.....	20
Organização do	
código.....	20
Testes.....	20
Documentação.....	21

Capítulo 7:

Plano de qualidade	21
Introdução.....	21
O que é	
qualidade?.....	21
Importância da qualidade em projetos de software.....	21
Responsabilidades do Gestor de Qualidade.....	22
Padrões de	
referência.....	22

Capítulo 8:

Plano de gestão de configurações	23
Enumeração de parâmetros e variáveis.....	24
Descrição dos vários “modos” do sistema.....	24

Capítulo 9:

Plano de documentação

.....	25
Documentos a produzir.....	25
Descrição de cada documento.....	25

Capítulo 10:

Plano de gestão de dados

.....	27
Descrição da forma como os dados são recolhidos.....	27
Descrição da forma como os dados são armazenados.....	27
Descrição da forma como os dados são manipulados.....	28
Apresentação das bases de dados.....	28
Estado inicial da base de dados.....	31
Permissão de acesso de dados.....	33
Conformidade com o RGPD.....	34

Capítulo 11:

Plano de gestão de recursos

.....	34
Introdução.....	35
Identificação dos recursos afetados ao projeto.....	35
Orçamento do projeto.....	36

Capítulo 12:

Plano de testes

.....	36
-------	----

Objetivos do plano de teste.....	36
----------------------------------	----

Descrição dos Tipos de integridade.....	36
---	----

Descrição dos testes a realizar e dos níveis de testes.....	37
---	----

Descrição de como são gerados os dados para os testes.....	38
--	----

Descrição dos testes de integração dos módulos.....	38
---	----

Capítulo 13:

Plano de treino.....	38
----------------------	----

Métodos de treino.....	38
------------------------	----

Recursos para cada método.....	39
--------------------------------	----

Base de dados do treino.....	40
------------------------------	----

Duração do treino.....	40
------------------------	----

Número de participantes e responsabilidades.....	40
--	----

Avaliação.....	40
----------------	----

Capítulo 14:

Plano de segurança.....	40
-------------------------	----

Segurança no desenvolvimento.....	40
-----------------------------------	----

Testes de segurança.....	41
--------------------------	----

Segurança no alojamento da base de dados.....	41
Segurança contra acessos não autorizados.....	41
Integridade dos Dados.....	42
Realização de backups na base de dados e a aplicação.....	42

Capítulo 15:

Plano de gestão de risco	42
Introdução.....	42
Impacto dos riscos.....	43
Identificação e análise dos riscos.....	43
Plano de Resposta aos riscos identificados.....	45

Capítulo 16:

Plano de manutenção	46
Identificação do modelo a adoptar.....	46
Apoio ao utilizador após a entrega do produto.....	46
Periodicidade da manutenção.....	46
Ações a realizar na manutenção.....	46
Responsáveis pela manutenção.....	46

Capítulo 17:

Conclusões.....	47
Reflexões finais.....	47
Áreas a melhorar.....	47
Expectativas de trabalho futuro.....	47
Referências.....	47

CAPÍTULO 1: INTRODUÇÃO

- Descrição dos objetivos do documento

Este documento é um plano de projeto que tem como objetivo detalhar, demonstrar e especificar na sua parte teórica o sistema de gestão de um hipermercado, separando o projeto em planos e especificando os demais.

● Público a que se destina

O público alvo deste documento é aos interessados ou seja todas as pessoas que vão se beneficiar na produção deste software, como por exemplo os funcionários do hipermercado.

● O propósito do projeto a desenvolver

O projeto tem como objetivo permitir a gestão do stock, a inclusão de funcionalidades destinadas para a recepção das encomendas e a facilitação no departamento de venda e devoluções dos produtos dentro de um hipermercado.

CAPÍTULO 2: ÂMBITO DO PROJETO

● apresentação do projeto

Como já foi abordado, o projeto tem como objetivo criar um software que irá permitir a gestão do stock de um hipermercado, catalogando cada produto com seu respectivo código, descrição, preço e código de barras, facilitando consideravelmente toda a parte de organização e distribuição de todos os produtos do mercado que são a parte mais crucial para o mesmo. toda a movimentação dos produtos no hipermercado será efectuada recorrendo ao uso de leitores de códigos de barras, modernizando as funções e deixando-as mais eficazes. pretende-se incluir funcionalidades que consigam permitir a recepção das encomendas do fornecedor tal como a deslocação do armazém para as secções e respectiva prateleira, abrindo portas para um nível de encomendas controladas e permitindo a melhor organização dos produtos. Além da implementação notável da gestão de todos os produtos, o sistema também possui funcionalidades para a venda dos produtos, permitindo que o cliente, caso não fique satisfeito, possa devolver um produto num prazo máximo a se definir.

● Decomposição funcional do sistema

A previsão é de 3 módulos principais para o funcionamento e melhor organização desse sistema, sendo eles PRODUTOS || ENCOMENDAS || CAIXA

Pretende-se que existam leitores de códigos de barras, também será necessário um servidor de base de dados para poder comportar toda a catalogação entre todos os módulos

apresentados, também se espera máquinas capazes de rodar o software e sua base de dados para melhor administrar todos os processos e por último pretende-se um caixa capaz de efetuar as transações assim como guardar a data da compra para fazer a devolução possível.

● Constrangimentos do processo

O sistema se introduzido corretamente com todas as estruturas hardware e softwares alojadas, deverá ter um desempenho muito superior a um sistema sem a implementação do mesmo. Podendo efetivamente ajudar o negócio como um todo.

O processo tem de ter uma atenção dobrada na hora de catalogar todos os instrumentos principais do negócio, pois uma falha nesse quesito pode limitar o encaminhamento do sistema inteiro.

CAPÍTULO 3: CALENDARIZAÇÃO

● Enumeração dos marcos a atingir(milestones)

O projeto tem 6 metas como principais marcos a se atingir.

●Descobrir os requisitos

Como de costume todo projeto se inicia com a descoberta dos requisitos, através das fontes de requisitos como os interessados no projeto por exemplo.

●Especificação dos requisitos

Após a descoberta dos requisitos e após a fase de análise e negociação movemos para especificação dos requisitos onde cada requisito passará por uma rigorosa descrição para se entender o que se espera de cada requisito.

●bases de dados

Criação da base de dados com todas as tabelas (fornecedor, produto, armazém, secções, caixa, compra, fatura/recibo) que já foram analisadas para o melhor cumprimento das funcionalidades e seus atributos referentes.

●aplicação (código)

Utilização do conhecimento em programação para o uso efetivo dos dados da base de dados com uma interface apropriada para o mesmo, todo o trabalho de interação com o utilizador, design gráfico e funcional será atribuída a essa milestone (utilização de php,html,mysql,javascript,css e qualquer linguagem cabível ao melhor funcionamento.

●testes

Etapa de testar o sistema como um todo e verificar se é satisfatório e cumpre com todos os requisitos apresentados e suas respectivas funcionalidades.

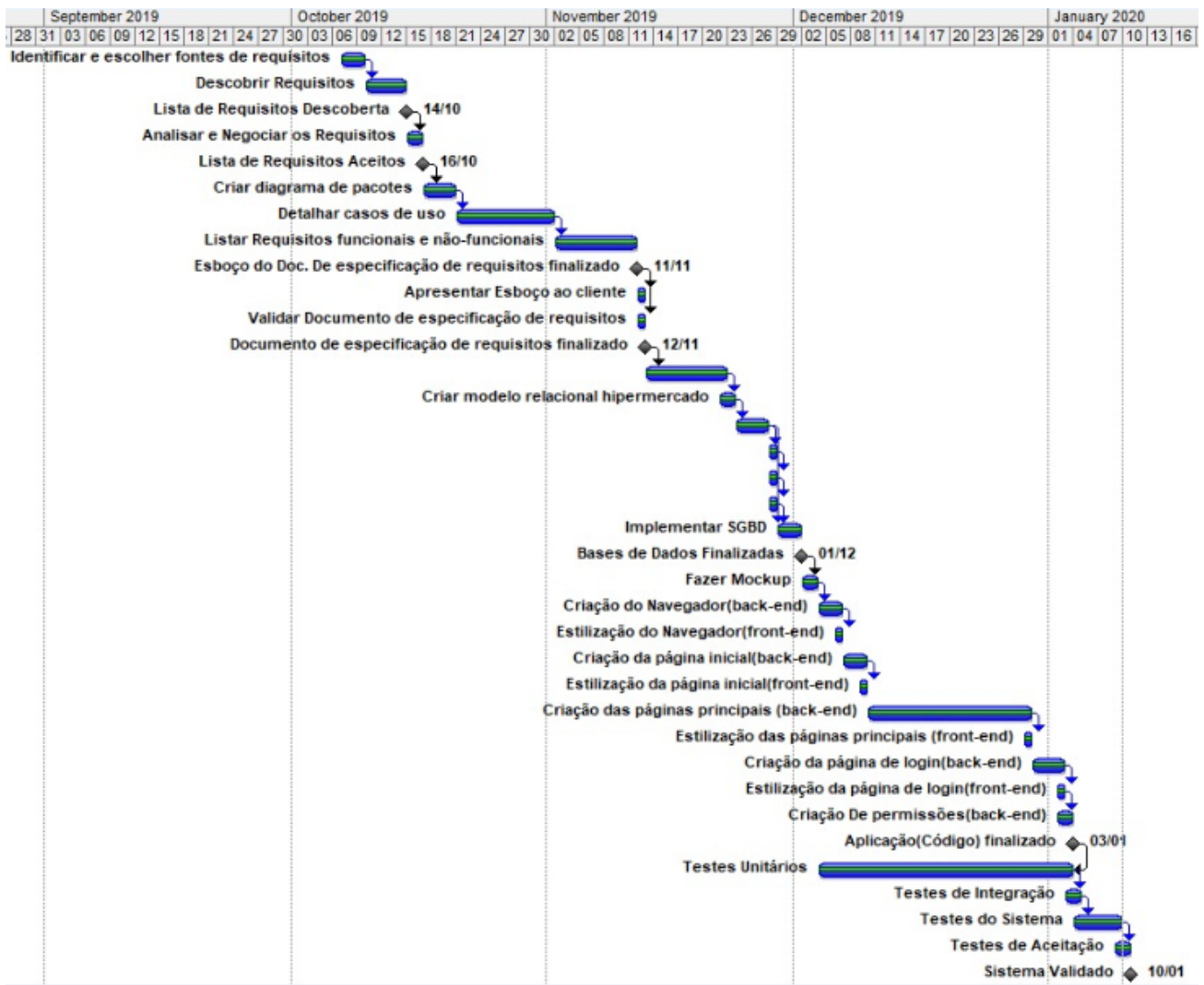
● Identificação das tarefas a realizar

- Identificação de todas as fontes de requisitos possíveis para a criação deste sistema e a escolha dos mesmos. auxiliando os processos posteriores
- Descoberta dos requisitos a partir das fontes de requisitos, comunicação com a parte interessada e com os desenvolvedores para a eficiência completa na descoberta, os mesmos serão negociados e identificados apenas os concordados em ambas as partes como útil para o sistema
- Especificação de todos os requisitos, explicando seu funcionamento fornecendo documentação ao projeto, estabelecendo uma ordem que o sistema terá que se dirigir e terá que cumprir com os mesmos.
- Com os requisitos já especificados podemos partir para a criação das bases de dados que agirão em todo o sistema, armazenando todas as informações pertinentes para os interessados, explicados na especificação.
- Criação do modelo entidade-associação e modelo relacional da base de dados.
- Implementação direta da base de dados no SQL(Structured Query Language) que será utilizado para a implementação do código, servindo como base de todo o sistema.
- Com a base de dados em sql, o projeto parte para a aplicação em código, fazendo toda a listagem,edição,inserção,exclusão dos dados no próprio sistema, os requisitos têm de ser todos atendidos e posteriormente podem ser adicionados quantos requisitos necessários(concordados entre ambas as partes) para a melhoria do sistema. O código sempre passa por análise rigorosa e descritiva.
- Cada uma das implementações em código no sistema deverá passar por um teste, validando sua eficiência e funcionalidade ao sistema.

● Identificação dos artefactos a desenvolver

O cliente vai receber todo sistema web com seus requisitos e funcionalidades, bem como a sua base de dados a ser desenvolvida.

- Calendarização do projeto



CAPÍTULO 4: ORGANIZAÇÃO DA EQUIPA

- **Equipa e tipo de organização**

Nome equipe: Equipe de Desenvolvedores.

organização: Por apresentar poucos integrantes, toda a etapa relacionada aos requisitos será feita em conjunto(desde sua identificação, até sua especificação e validação), com a finalidade de se chegar a um uma única idéia entre os integrantes do que se espera do sistema.

Todos os desenvolvedores irão participar em conjunto efetivamente em todas as etapas citadas anteriormente(Capítulo 3).

- **Membros da equipe**

Gabriel de Souza Faria - Estudante de engenharia da informática

- **Lista das funções/competências necessárias**

Espera-se familiaridade com linguagens de programação estruturada, linguagens de programação orientada a objetos, conhecimentos específicos da linguagem PHP e MYSQL para a implementação de todos os requisitos no serviço WEB.

- **tarefas/funções de cada membro da equipe**

Gabriel de Souza Faria Lima - Participação no desenvolvimento prático e lógico da base de dados e estrutura de código do sistema market management,com todos os sistemas de modelagem sugeridos.

CAPÍTULO 5: DESCRIÇÃO TÉCNICA DOS SISTEMAS PROPOSTOS

- **Descrição do software que será necessário desenvolver**

5.1 Requisitos Funcionais

5.1.1 registrar utilizador no sistema

Um utilizador só pode aceder ao sistema caso esteja autenticado ao mesmo, caso contrário deverá ser direcionado para a página inicial de cadastro/login. Para se registrar o utilizador precisa criar um nome de usuário e uma senha, e inserir um email válido. após o registro o utilizador recebe uma permissão de “convidado” e ainda não pode aceder ao sistema até que seu superior hierárquico lhe dê uma permissão condizente com sua função na empresa.

5.1.2 Definir Permissões

O sistema deverá permitir que um administrador defina as permissões dos seus funcionários, delimitando assim, os níveis de acesso que um utilizador pode ter, bem como as funcionalidades que este pode aceder.

5.1.3 registrar compras

Quando um cliente efetuar uma compra em um dos caixas, o sistema deverá registrar essa compra antes de emitir a fatura. Guardando dessa forma todas informações da compra e o NIF do cliente(caso este deseje adicioná-lo). Esse registro é fundamental para saber se um cliente pode ou não devolver um produto e para o controle de vendas.

5.1.4 emitir fatura

Após o registro de uma compra, o caixa deverá emitir uma fatura com as informações da compra em questão, como por exemplo: data da compra, produtos comprados e suas quantidades, forma de pagamento, NIF do cliente(caso o mesmo tenha optado por preencher esse campo).

A fatura é essencial para o controle de vendas e para saber se um produto pode ou não ser devolvido.

5.1.5 devolver produto

O sistema deverá permitir que um cliente devolva um produto, se a compra do mesmo foi realizada no máximo 15 dias antes do ato da devolução, após a devolução, o produto devolvido, a data e o motivo devem ficar registrados. E o produto devolvido volta para a listagem do armazém.

5.1.6 criar promoção

O sistema deverá permitir que um utilizador possa criar uma promoção e vinculá-la a um produto, estabelecendo a data limite e os parâmetros da promoção.

5.1.7 cancelar promoção

O Sistema deverá permitir que um utilizador possa cancelar uma promoção, mesmo que esta ainda esteja em tempo válido.

5.1.8 deslocar produto

Uma das principais atividades da organização que irá adquirir o sistema é a deslocação dos produtos entre secções e armazéns. O sistema irá dar suporte e automatizar esse processo.

5.1.9 deslocar encomenda

O utilizador poderá tornar uma encomenda como entregue, definindo sua data de chegada(caso não seja a mesma que esteja efetuando a deslocação), fazendo com que os produtos dessa encomenda sejam deslocados para o armazém ou para uma secção.

5.1.10 registrar produto

O sistema deverá permitir que o utilizador insira novos produtos ao catálogo de produtos disponíveis, bem como suas informações de preço, peso, código de barras.....

5.11 registrar encomenda

O sistema deverá permitir que o utilizador adicione novas encomendas, definindo quais e quantos produtos quer e seu respectivo fornecedor.

5.1.12 registrar fornecedor

O sistema deverá permitir que o utilizador insira novos fornecedores à lista de fornecedores, bem como suas respectivas informações de contato, email e endereço.

5.1.13 registrar secção

O sistema deverá permitir que o utilizador crie novas secções, para que sejam alocados novos tipos de produto de acordo com a respectiva secção.

5.1.14 listar compras

O sistema deverá permitir que o utilizador exiba em seu ecrã uma listagem de todas as compras efetuadas, com suas informações de data, custo, tipo de pagamento, produtos e suas quantidades e a fatura referente à compra.

5.1.15 listar promoção

O sistema deverá permitir que o utilizador exiba em seu ecrã uma listagem de todas as promoções ativas, com suas informações de data de validade, tipo de promoção e à qual produto se destina essa promoção. O utilizador também poderá acessar as promoções do passado (que já não se encontram ativas), e poderá reativar essas promoções estabelecendo uma nova data de validade.

5.1.16 listar armazém

O sistema deverá permitir que o utilizador exiba em seu ecrã uma listagem de todos os produtos que se encontram dentro do armazém, com suas informações de quantidade e a data de chegada.

5.1.17 listar secção

O sistema deverá permitir que o utilizador exiba em seu ecrã uma listagem de todos os produtos que se encontram dentro de cada secção, com suas informações de quantidade (total e a que se encontra na respectiva secção) e a data de chegada.

5.1.18 listar produto

O sistema deverá permitir que o utilizador exiba em seu ecrã uma listagem do catálogo do mercado, mostrando assim, todos os produtos que o hipermercado tem disponível à venda (estando eles em estoque ou não), exibindo informações como preço por unidade, peso do produto e qual unidade de medida do produto em questão.

5.1.19 listar encomenda

O sistema deverá permitir que o utilizador exiba em seu ecrã uma listagem de todas as encomendas, mostrando informações como: o status da venda (entregue ou não entregue), quem as forneceu, os produtos que compõem essa encomenda e suas quantidades e os custos dessa encomenda.

5.1.20 listar fornecedor

O sistema deverá permitir que o utilizador exiba em seu ecrã uma listagem de todos os fornecedores, exibindo informações como: endereço, e-mail e contacto.

5.1.X Aviso de Produtos Esgotados

O sistema deverá notificar o utilizador quando um produto acabar (sua quantidade total for 0).

5.1.21 Aviso de Produtos Acabando

O sistema deverá notificar o utilizador quando um produto estiver próximo de acabar, ou seja, sua quantidade total esteja em número baixo, número esse que será definido pelo utilizador.

5.2 Requisitos Não-Funcionais

5.2.1 Validação de ociosidade

Após x tempo autenticado(ainda a definir quanto) ao sistema, sem a utilização do mesmo, o sistema deverá encerrar automaticamente a sessão por questões de segurança.

5.2.2 controlar número de tentativas erradas(bloquear acesso)

Por questões de segurança, após x tentativas inválidas de login(nº ainda a definir), o utilizador que está a tentar aceder ao sistema ficará bloqueado por tempo (ainda a definir este tempo).

5.2.3 Iniciar Sessão no Sistema

Para aceder ao sistema é necessário introduzir um login e uma senha, estes serão perguntados na tela inicial, caso o utilizador já não esteja autenticado. Para possuir um login e senha o utilizador tem de estar registrado previamente. Caso contrário, o utilizador não poderá acessar nenhuma funcionalidade do sistema.

5.2.4 Encerrar Sessão

O utilizador poderá encerrar uma sessão sempre que precisar se ausentar do sistema ou por qualquer outro motivo, sem perder as modificações que foram feitas.

5.2.5 Usabilidade

O sistema deverá apresentar interface com maior intuitividade possível e simplicidade. Aumentando a compreensão e aprendizagem do sistema por parte dos usuários.

5.2.6 Base de Dados

Deverá ser Desenhada e implementada uma base de dados que possa armazenar todas as informações de forma eficiente e segura de um hipermercado abordadas até então.

5.2.7 Multiplataforma

O sistema poderá ser acessado por telemóveis e outros sistemas operativos, com intuito de criar maior portabilidade e dinamismo por parte dos gerentes e

gestores da organização que irá adquirir o sistema, já que poderão acessar o sistema mesmo não estando em seu local de trabalho.

5.2.8 Linguagens

Este sistema será uma aplicação web, então as linguagens back-end adotadas foram o PHP, JavaScript. E as front-end foram HTML, CSS.

5.2.9 Critérios de pesquisa em páginas do sistema

O sistema permitirá o utilizador pesquisar por alguma palavra chave, essa podendo ser qualquer trecho de informação armazenada na base de dados da respectiva página, ou seja o utilizador pesquisar ou código de barras na área de listagem das secções ou pesquisar por uma fatura ou uma data de alguma compra realizada no hipermercado todas as pesquisas têm de serem feitas de acordo com as colunas da base de dados, respeitando valores que tem de ser números inteiro e valores que tem de ser alguma referência textual a um valor na base de dados cada página informará o critério específico.

5.2.10 Critérios de cadastro em páginas do sistema

O sistema permitirá o utilizador cadastrar novos elementos desejados como novas secções, encomendas, compras, produtos, fornecedores, devoluções introduzindo todo tipo de informação necessária para o armazenamento na base de dados, seguindo sempre a estrutura já preenchida na mesma, respeitando valores inteiros que precisam ser valores inteiros e textos que precisam ser textos para validar a inserção de dados.

●Descrição da estrutura física do hardware

O sistema interage com o caixa automático que irá armazenar o processo de venda de seus produtos, passando todas as informações impostas por todos os critérios discutidos nos requisitos, para a base de dados do hipermercado, onde o servidor web terá uma ligação com a mesma, o sistema market management necessita de um sistema de login para o acesso do software. O sistema de login por sua vez precisa de sua própria base de dados para armazenamento dos utilizadores e das permissões dos mesmos no sistema, é encarregado de todos os processos previstos e disponibiliza a utilização do sistema. Todas as informações sobre gestão de estoque e gestão de compras são fornecidas através da base de dados do hipermercado e essenciais para o desempenho total dos requisitos abordados.

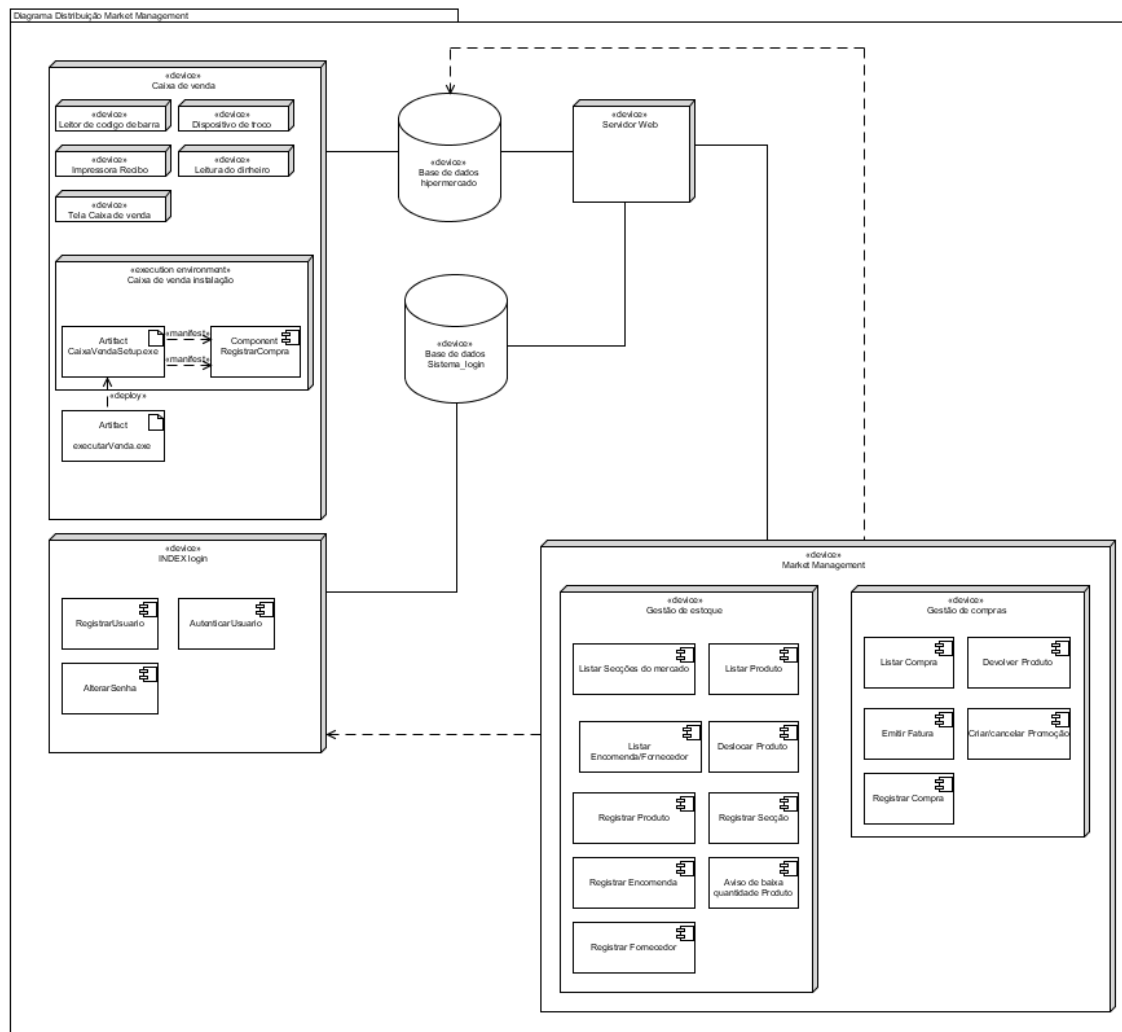


Figura 2 – Diagrama de distribuição Market Management

- O diagrama de distribuição tem como objetivo mostrar quais componentes hardware (representado pelas caixas “nodes”) existem, que softwares executam-se em cada um desses “nodes” e como as peças se conectam entre si. Os diagramas de componentes são usados para descrever:
 - Os arquivos de código fonte desenvolvidos em um ambiente de desenvolvimento integrado
 - Os arquivos executáveis necessários para fornecer um sistema em execução
 - Bancos de dados físicos que armazenam informações nas tabelas de um banco de dados relacional ou nas páginas de um banco de dados orientado a objetos

● Descrição da interface gráfica:

Todos os mock ups feitos abaixo podem e serão modificados com inserção de todas as funcionalidades pretendidas e um polimento do produto final extremamente maior em questão visual e funcional, o mockUp serve como uma forma de apenas dar uma ideia vaga de como seria alguma das interfaces gráficas do sistema, não possuindo todas as interfaces que podem ser adicionadas no resultado final. A interface gráfica do sistema tem como principal objetivo combater a frustração dos utilizadores, tentando promover uma interface de fácil aprendizagem, eficiente e com alta memorabilidade ou seja a proficiência após período sem utilização, o programa deve ser intuitivo e auto explicativo para os utilizadores.

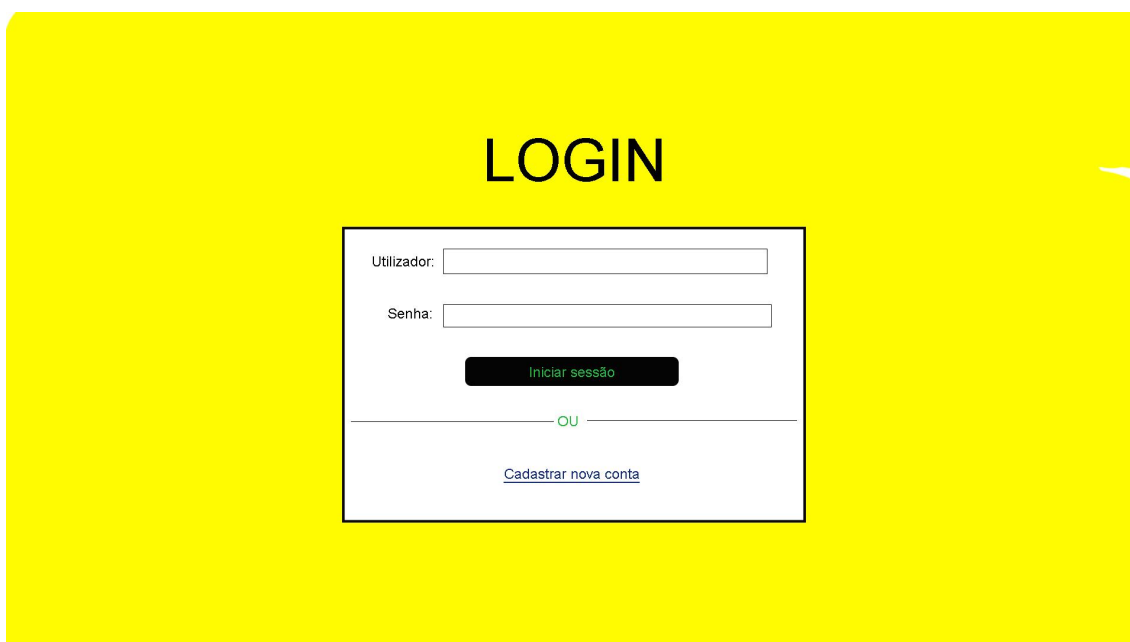


Figura 3 – MockUp index login Market Management

● Contextualização:

Página index do programa com campos de autenticação do utilizador ao programa, um nome de utilizador ou email do utilizador feito no cadastro e sua senha respectiva, a página conta com um link que direciona para a área de registo de um novo utilizador (utilizador este que terá permissão de um convidado e não poderá acessar o programa a primeira instância).

REGISTRAR

Nome de utilizador:

Email:

Senha:

Confirme a senha:

Registrar

OU

[Já possuo uma conta, voltar ao login.](#)

Figura 4 – MockUp Área de registro de utilizadores Market Management

- Contextualização:

Página de registro de novos utilizadores no sistema, com campos de preenchimento que serão armazenados pela base de dados e utilizados na autenticação do utilizador, a página conta com um link que direciona para a área de login caso o utilizador já tenha uma conta no sistema.



Figura 5 – Página principal Market Management

- Contextualização:

Página principal do sistema que será a primeira tela a ser vista após o login, possui diversos links direcionais para a movimentação entre a listagem dos elementos principais do sistema e possui o botão responsável pelo término da sessão do utilizador quando o mesmo desejar.



código barra	descrição	Quantidade armazém	Quantidade Total
1	Banana	50 kg	200 kg
2	lorgute	100 ml	250 ml
3	Soro em pó	20 gr	200gr

Figura 6– MockUp área de listagem dos elementos principais Market Management

- Contextualização:

Página de listagem dos elementos principais a partir de uma tabela, mostrando todo tipo de informação acerca do elemento, a página conta funcionalidade de retorno à página principal, cadastro de novos elementos(Em relação a página atual) e na barra de navegação o utilizador pode guiar-se entre cada um dos elementos que serão listados no site (Produto, Secção,Compra,Encomenda cada um desses elementos possuem seus sub elementos como fornecedores, devoluções de compras)

The image shows a web interface for Market Management. At the top, there is a dark red header bar containing a home icon and a back arrow. The main background is bright yellow. In the center, there is an orange rectangular form titled 'Cadastrar novo elemento' with a red underline. Inside the form, there are two input fields: 'Nome elemento:' and 'Quantidade:'. At the bottom of the form is a dark red button with the text 'Cadastrar novo elemento' in white.

Figura 7 – Área de cadastro de novo elemento Market Management

- **Contextualização:**

Demonstração de como a interface gráfica funcionária, Recolhendo todo tipo de informação relevante para a criação do novo elemento, sendo armazenado e adicionado na base de dados a página de cadastrar novo elemento também está diretamente relacionada a em que página o utilizador se encontra no momento, a página também conta com links de redirecionamento para a página anterior ou a página principal do market management.

CAPÍTULO 6: STANDARDS, TÉCNICAS E FERRAMENTAS

- **Standards**

- A organização do código deverá ser feita por funções, facilitando o trabalho em equipe, a manutenção do site e o fácil entendimento de toda a estrutura do mesmo
- Os testes serão implementados sobre critérios específicos da função a ser testada, sempre visando acima de tudo a eficiência da mesma, ou seja, produzir mais por menos custo.

- Toda a documentação deve adotar fonte calibre com tamanho 12, sendo os títulos tamanho 28, e os subtítulos 20. Os documentos devem estar em formato .PDF, e todos eles devem ser entregues a todos os interessados.

CAPÍTULO 7: PLANO DE QUALIDADE

● Introdução

Quando uma empresa decide procurar outra para criação de um software, aquela que, de alguma forma, melhorar seus serviços e/ou produtos; ou expandir sua clientela, suas vendas, com intuito quase sempre de aumentar receitas e lucros; quer melhorar o controle interno de seus procedimentos, seja para reduzir custos, ou apenas para otimizar seu tempo de execução de tarefas, ou agregar valor à sua marca, dando à seus clientes, um algo a mais, fidelizando assim sua clientela e por aí em diante.

Desta forma, ao aceitar criar um software, seja para uso próprio ou para vendê-lo a um cliente, é necessário que exista um planejamento muito bem gerenciado, com todas as etapas necessárias. Seguir normas e padrões de qualidade, entre outras atribuições já conhecidas para planejamento e gerenciamento de projetos. Neste caso, projeto de criação de software.

● O que é qualidade?

A relevância de cada produto pode variar com o contexto e ao longo do tempo, pois as pessoas podem mudar seus posicionamentos e atualizar suas preferências, com relação a um objeto de uma questão. Portanto, a qualidade não é absoluta, mas depende da perspectiva do avaliador. Por isso, qualidade é um conceito complexo, porque possui significados diversos para diferentes pessoas.

Portanto qualidade de software é um conjunto de propriedades a serem satisfeitas, em determinado grau, de modo que o software satisfaça as necessidades do usuário, e para isso acontecer é necessário que o gerente de projeto tenha, de forma clara e definida, o objetivo daquele projeto de desenvolvimento.

● Importância da qualidade em Projetos de Software

Conforme pudemos perceber, ter um plano de qualidade de software se faz necessário, justamente para poder atender de maneira eficaz e eficiente os requisitos por parte do cliente. Sem um bom planejamento no controle de qualidade de software, o produto a ser entregue pode não atender às normas de mercado, ou aos requisitos do cliente. É na gestão da qualidade que se determina a política, os objetivos e responsabilidades. Lembrando que os processos de gerência, de planejamento e

controle de qualidade devem ser flexíveis, adaptáveis à realidade de cada produto (software) a ser preparado, testado.

São elaborados por meios de normas e padrões, mas devem ser planejados de maneira que atenda aos requisitos dos clientes. Para atender da melhor maneira possível, dentro das melhores e apropriadas práticas, é necessário ter uma estratégia de testes, com controle e planejamento.

● Responsabilidades do Gestor de Qualidade

Gerência de Projeto de software é a aplicação do conhecimento, habilidade, ferramentas e técnicas no desenvolvimento de software com o objetivo de atender aos requisitos do projeto.

As principais dificuldades que um gerente de projeto enfrenta para assegurar a qualidade do seu projeto e do seu produto são conseguir atingir ou exceder as expectativas e balancear demandas e conflitos.

Algumas responsabilidades de um Gestor de Qualidade:

1. Foco no Cliente – Este é o principal e requer uma combinação de conformidade aos requisitos (o projeto deve produzir o que se comprometeu a produzir) e adequação ao uso (o produto ou serviço deve satisfazer as necessidades reais). A satisfação do cliente é considerada como parte integrante das especificações do projeto;

2. Liderança – Saber conduzir tudo e todos para um objetivo comum definido no início do projeto;

3. Envolvimento da Equipe – Todos devem se sentir à vontade para falar, e o Gerente do projeto deve estar apto a ouvir;

4. Prevenção ao invés de inspeção – O custo de prevenir erros é sempre menos que o custo de corrigi-los quando descobertos em inspeções;

● Padrões de Referência

Para Sommerville (2007, p.426), para se garantir a qualidade do software, é necessário definir padrões a serem seguidos ao se implantar um processo de desenvolvimento de software. Existem dois tipos de padrões que podem ser implantados como parte do processo na garantia de qualidade:

I. Padrões de produto. Onde são aplicados padrões ao produto de software em desenvolvimento. Incluindo padrões de documentos, como a estrutura de documentos de requisitos; e, padrões de codificação, estes padrões definem como uma linguagem de programação deverá ser usada.

II. Padrões de processo. Definem o processo em si. Devem ser seguidos durante o desenvolvimento do software. Podem definir, por exemplo, processos de

especificação, projeto e validação, e uma descrição dos documentos que devem ser escritos ao longo dos processos.

Assim define, Sommerville (2007, p.432), itens fundamentais em projetos:

1. Planejamento da qualidade: É o desenvolvimento de um plano de qualidade de um projeto. Este plano de qualidade deve estabelecer um padrão de qualidade de software desejado, baseado em requisitos, e descrever como ele deve ser avaliado, não deixando margem para que o engenheiro possa fazer suposições diferentes ou conflitantes aos atributos que deverão ser otimizados.

É no plano da qualidade que deve ser definido os atributos de qualidade mais importantes para o software, pois existe uma grande variedade de atributos de qualidade de softwares, e geralmente não é possível para nenhum sistema ser otimizado em todos os seus atributos. O plano também deve definir qual processo de avaliação de qualidade. Deve ser padronizada e que esteja presente no produto a ser entregue.

2. Controle de qualidade: É aqui que há a monitoração do processo de desenvolvimento de software para assegurar que os procedimentos e os padrões de garantia de qualidade estão sendo seguidos, conforme estabelecido no planejamento de qualidade.

A abordagem básica usada no gerenciamento da qualidade descrito por Sommerville deve ser compatível com a usada pela International Organization for Standardization (ISO).

A ISO 9000 é um padrão internacional de qualidade que aplica o gerenciamento da qualidade de processo para gerar produtos que atendam às expectativas de seus usuários. Esses padrões foram criados sob a premissa de que, se o desenvolvimento e o gerenciamento do sistema são de boa qualidade, então o produto ou serviço resultante também será de boa qualidade. Um sistema de qualidade em conformidade com a ISO 9000 assegurará que seu processo de desenvolvimento possui um nível de controle, disciplina e repetibilidade, garantindo a qualidade de seus produtos.

CAPÍTULO 8: PLANO DE GESTÃO DE CONFIGURAÇÕES

O utilizador quando registrado fica à espera da confirmação do seu superior hierárquico. esse por sua vez, confirma o registro e delega uma permissão ao utilizador, permissão que definirá o que o utilizador pode e não pode fazer.

Quando o utilizador registrado acede ao sistema, ele inicia uma sessão com a permissão atribuída.

● Gestor do Sistema

- Equipe de suporte que tem acesso à todas funcionalidades do sistema (gabriel)

● Administrador

- Gerentes, supervisores e outros grupos restritos dentro dos utilizadores. Podem utilizar funcionalidades que os utilizadores não podem.

● Utilizador

- Utilizadores em back office, como por exemplo: analista de estoque, gerente de logística. Podem deslocar produtos, aprovar encomendas e outras tarefas internas da empresa.

● Utilizador registrado

- utilizadores em front-office autenticado. Têm acesso à página inicial do sistema e a algumas funcionalidades do sistema, de acordo com as respectivas competências do seu cargo.

● Utilizador não-registrado

- utilizadores em front-office não autenticado. Têm acesso à página inicial do sistema e podem se registrar, após o registro devem esperar a confirmação do seu superior hierárquico.

● Público

- O sistema não é destinado ao público, nem tão pouco aos clientes do hipermercado, logo, o sistema não contará com funcionalidades para os mesmos.

CAPÍTULO 9: PLANO DE DOCUMENTAÇÃO

● Documentos à produzir

#	Documento	Responsável	
		Redação	Aprovação
1	Especificação de Requisitos	Gabriel Faria	Pedro Stretch
2	Plano de Qualidade	Gabriel Faria	Pedro Stretch
3	Plano de Gestão de Configurações	Gabriel Faria	Pedro Stretch
4	Plano de Documentação	Gabriel Faria	Pedro Stretch
5	Plano de Gestão de Dados	Gabriel Faria	Pedro Stretch
6	Plano de Gestão de Recursos	Gabriel Faria	Pedro Stretch
7	Plano de Testes	Gabriel Faria	Pedro Stretch
8	Plano de Treino	Gabriel Faria	Pedro Stretch
9	Plano de Segurança	Gabriel Faria	Pedro Stretch
10	Plano de Gestão de Riscos	Gabriel Faria	Pedro Stretch
11	Plano de Manutenção	Gabriel Faria	Pedro Stretch

Tabela 1 - Documentos necessários e seus responsáveis

● Descrição de cada documento

#1 Documento de Especificação de Requisitos

Documento destinado a todos os interessados do projeto, utilizado como base contratual entre desenvolvedores de software e seus clientes.

Este documento delimita o escopo do conjunto de funcionalidades que um sistema deve prover, bem como descreve os atributos de qualidade que devem ser suportados.

Também serve de apoio aos desenvolvedores de software, já que esse documento aborda sobre todos os requisitos acordados até então, de forma detalhada.

#2 Plano de Qualidade

Este plano de qualidade deve estabelecer um padrão de qualidade de software desejado, baseado em requisitos, e descrever como ele deve ser avaliado, não deixando margem para que o engenheiro possa fazer suposições diferentes ou conflitantes aos atributos que deverão ser otimizados.

#3 Plano de Gestão de Configurações

Este documento tem como objetivo definir parâmetros e variáveis que podem alterar a forma como o sistema é usado, ou seja, o sistema pode se comportar e ser utilizado de diferentes formas.

Essa diferença de comportamento, bem como a descrição dos parâmetros que as alteram, são definidos no Plano de Gestão de Configurações.

#4 Plano de Documentação

Documento que enumera e descreve todos os outros documentos que compõem o desenvolvimento do projeto e que devem ser entregues, além de registrar seus respectivos responsáveis.

#5 Plano de Gestão de Dados

Este documento aborda quais dados serão coletados e gerados pelo projeto, e como serão preservados e disponibilizados, considerando questões éticas, legais e de confidencialidade.

Além de descrever quais as metodologias e padrões que serão utilizados nesses processos, como e sob que condições esses dados serão compartilhados e/ou tornados abertos para a comunidade de pesquisa.

#6 Plano de Gestão de Recursos

Este documento identifica quais recursos são afetados pelo projeto, separando-os ainda por tipos(humanos, tecnológicos, infraestrutura) e como eles serão utilizados. Este documento também aborda os custos e o orçamento do projeto.

#7 Plano de Testes

O plano de teste é um documento com uma abordagem sistemática para o teste de sistemas como hardware ou software. Ele geralmente consiste numa modelagem detalhada do fluxo de trabalho durante o processo. Os testes são divididos em diferentes níveis de testes, cada um dos testes é detalhado e ainda o resultado esperado de um teste e o obtido.

#8 Plano de Treino

Este documento visa planejar o treinamento dos usuários que irão adquirir o sistema, a fim de promover um maior entendimento do sistema por parte dos interessados. Esses treinos podem ser presenciais ou a distância. Além disso, este documento também descreve os recursos utilizados nesses treinos.

#9 Plano de Segurança

Este documento planeja toda a segurança no desenvolvimento do projeto em questão. Sobre tudo a integridade e segurança dos dados, medidas contra acessos não autorizados e ainda testes de segurança para verificar se o sistema cumpre com as normas estabelecidas e os padrões de segurança desejados.

#10 Plano de Gestão de Riscos

Este documento tem o objetivo de identificar e analisar os riscos, além de criar medidas preventivas e corretivas para esses riscos. Define também o quão ruim seria se um determinado risco acontecesse.

#11 Plano de Manutenção

Este documento visa planejar o suporte ao cliente após o sistema pronto e entregue ao mesmo, contém informações como: como esse suporte será dado ao cliente, a periodicidade dessas manutenções, quais ações serão realizadas e seus respectivos responsáveis.

CAPÍTULO 10: PLANO DE GESTÃO DE DADOS

● Descrição da forma como os dados são recolhidos

Todos os dados são recolhidos através de formulários especializados para cada função pretendida ao sistema como por exemplo a criação de um novo produto, uma nova seção, todos esses processos necessitam de informações e essas serão providenciadas pelo utilizador do sistema no preenchimento do formulário citado. Esses formulários estão diretamente ligados à base de dados do nosso sistema que providenciará suporte integral com armazenamento e manipulação desses dados.

● Descrição da forma como os dados são armazenados

Assim que preenchido os formulários, uma função será responsável por recolher os valores preenchidos, e inserir os mesmos na base de dados do sistema, armazenando variáveis sobre a sessão do utilizador e sobre todas as informações de cadastro, edição, exclusão de produtos, seções, encomendas, fornecedores, compras.

- Descrição da forma como os dados são manipulados

O sistema utiliza-se dos dados armazenados na base de dados quase em todas as páginas do aplicativo, os dados armazenados na base de dados são por sua vez exibidos ao utilizador em páginas de listagem(Requisitos funcionais). Algumas colunas da base de dados são o resultado de algum cálculo como o preço de um produto que é relacionado ao seu peso, então os dados que fazem parte do seu cálculo também são manipulados em ordem para inserir o valor correto da operação sempre atualizada na base de dados do sistema.

- Apresentação das bases de dados

1-Base de dados hipermercado:

1.1-Modelo Entidade-Associação:

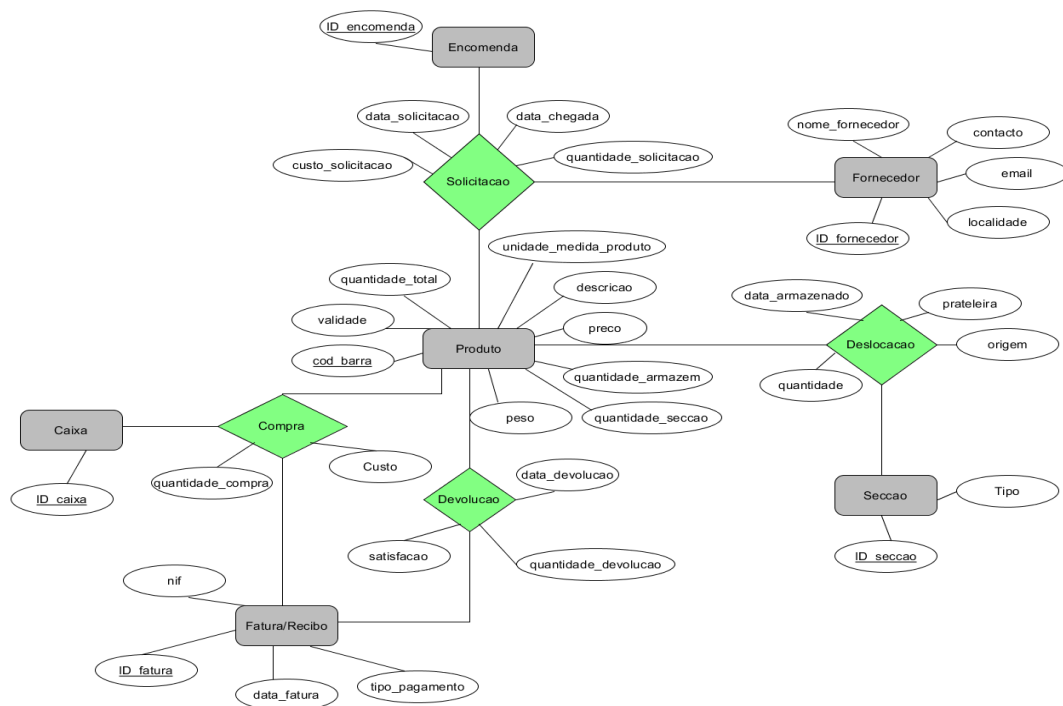


Figura 8 –Modelo entidade-associação da base de dados hipermercado

- Contextualização:

Legenda: <Nome de uma associação>,[Nome de uma entidade].

A figura mostra que a entidade principal desse modelo será o produto.

Entidade esta que participa efetivamente com todas as outras através de associações. O produto tem uma associação <Solicitação> diretamente relacionada com encomendas e fornecedor que por possuir atributos próprios dá origem a uma tabela com uma chave sendo composta pelas as chaves das entidades associadas (produto,encomenda e fornecedor), todas as entidades associadas também geram uma tabela e possuem seus atributos que são as características que permitem descrever as entidades e associações.

As associações foram escolhidas após algumas abordagens de qual seria a melhor forma de registrar o processo de movimentação do produto e a decisão mais pertinente foi a de cada uma dessas associações terem seus atributos próprios, nesse modelo podemos ver <Compra>, <Devolução>, <Deslocação>, <Solicitação>.

<Compra> tem seus atributos próprios armazenando a quantidade da compra, o custo desta e é diretamente relacionada com [Produto] ,[Caixa] e [Fatura/recibo] tendo uma chave composta que referencia qual produto foi comprado, em que caixa foi comprado e a qual fatura essa compra está associada.

Em caso de insatisfação do cliente, <Devolução> tem seus atributos próprios armazenando a quantidade da devolução, qual foi o motivo da devolução do produto, ou seja a satisfação do cliente e a data dessa devolução, é diretamente relacionada com [Produto] e [Fatura/recibo] tendo uma chave composta que referencia qual produto foi devolvido e a qual fatura essa devolução está associada.

<Deslocação> tem seus atributos próprios que giram em torno de produto armazenando a quantidade armazenada, a prateleira que está armazenado e a origem desse produto ou seja em que secção ele estava antes de estar na secção atualmente armazenado, é diretamente relacionada com [Produto] ,[Secção] tendo uma chave composta que referencia qual produto está sendo deslocado, e em qual seção este produto está sendo armazenado.

<Solicitação> tem seus atributos próprios que giram em torno de produto armazenando a data da solicitação e a data de chegada da encomenda do produto, a quantidade solicitada e o custo desta solicitação, é diretamente relacionada com [Produto] ,[Encomenda] e [Fornecedor] tendo uma chave composta que referencia qual produto está sendo encomendado, de qual fornecedor pertence esse produto e em qual encomenda essa solicitação está associada.

1.2-Modelo Relacional:

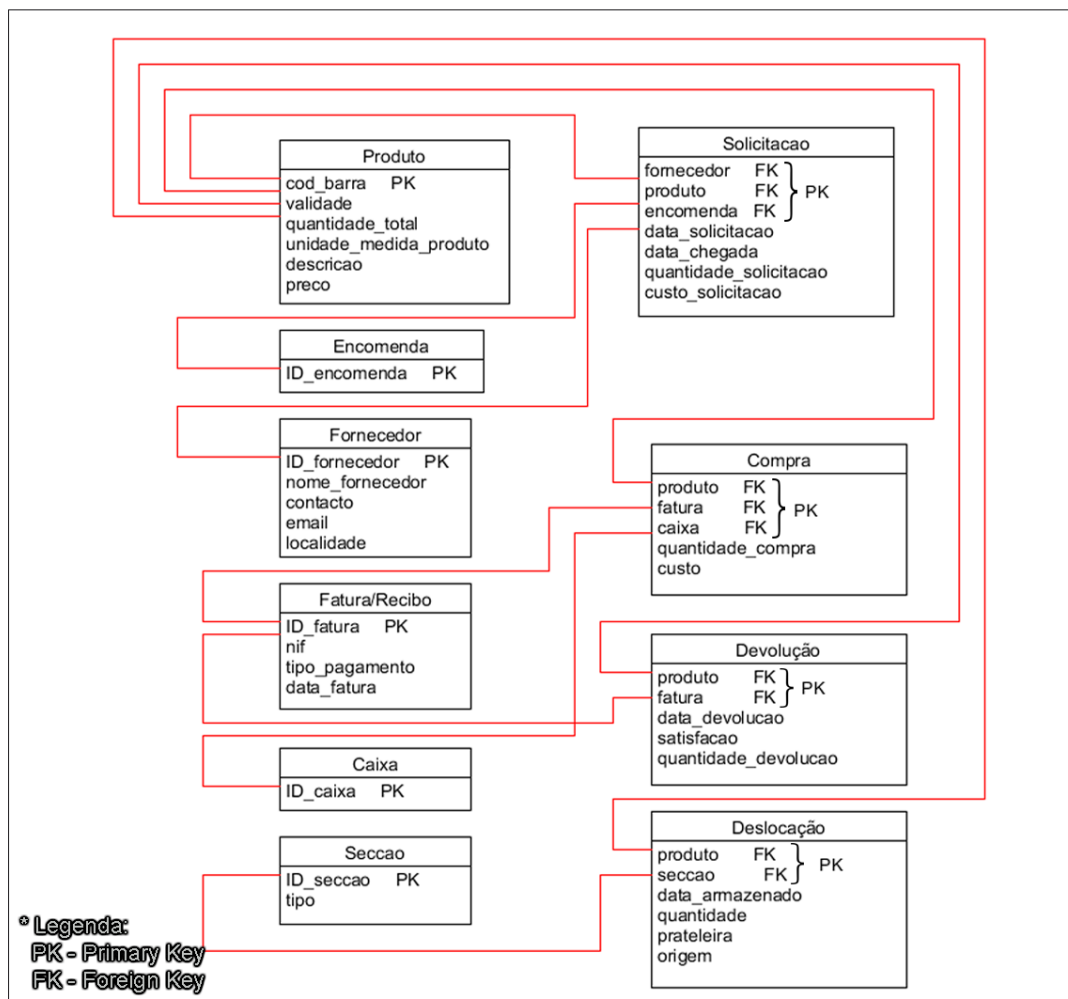


Figura 9 –Modelo relacional da base de dados hipermercado

- Contextualização:

O modelo relacional tem como objetivo demonstrar de uma forma mais natural todas as tabelas geradas pelo modelo entidade associação (pagina 3), demonstrando as associações entre cada tabela e demonstrando como as chaves (valores da coluna PK) são referenciadas (FK)

• Estado inicial da base de dados

```

INSERT INTO produto (cod_barra, quantidade_total, quantidade_armazen, quantidade_seccao, descricao, preco_unitario, anno, unidade_medida, produto_validade) VALUES
(1, NULL, NULL, NULL, 'Banana', 0.79, '100', 'g', '2019-11-27'),
(2, NULL, NULL, NULL, 'Maça', 0.75, '100', 'g', '2019-11-27'),
(3, NULL, NULL, NULL, 'Laranja', 1.05, '500', 'ml', '2019-12-29'),
(4, NULL, NULL, NULL, 'Leite ester. gordo', 0.55, '1', 'l', '2019-12-27'),
(5, NULL, NULL, NULL, 'Iogur', 0.5, '50', 'g', '2019-11-27'),
(7, NULL, NULL, NULL, 'Frango', 5.5, '1', 'kg', '2019-12-20'),
(8, NULL, NULL, NULL, 'Carne de porco', 4.2, '1', 'kg', '2019-12-24'),
(9, NULL, NULL, NULL, 'Queijo', 5.5, '1', 'kg', '2019-12-27'),
(10, NULL, NULL, NULL, 'Pacote', 1.2, '1', 'kg', '2019-12-24'),
(11, NULL, NULL, NULL, 'Queijo', 2, '500', 'g', '2019-12-29'),
(12, NULL, NULL, NULL, 'Margarina', 8, '500', 'g', '2019-12-24'),
(13, NULL, NULL, NULL, 'Linguiça', 3, '1', 'kg', '2019-12-29'),
(14, NULL, NULL, NULL, 'Lampião', 0.99, '1000', 'g', NULL);

INSERT INTO fatura (ID_fatura, ref, data_fatura, tipo_pagamento) VALUES
(1, 12345678, '2019-12-31 00:00:00', 'Cartão'),
(2, NULL, '2019-12-31 00:00:00', 'Dinheiro');

INSERT INTO caixa (ID_caixa) VALUES
(1),
(2),
(3),
(4);

INSERT INTO secao (ID_secao, tipo) VALUES
(1, 'Frutas'),
(2, 'Frutas'),
(3, 'Vegetais'),
(4, 'Vegetais'),
(5, 'Cereais'),
(6, 'Verduras'),
(7, 'Bebidas');

INSERT INTO fornecedor (ID_fornecedor, nome_fornecedor, contacto, email, localidade) VALUES
(1, 'Yara', '123-456-789', 'fornecedor1.com', 'Porto'),
(2, 'Zoe', '123-456-789', 'fornecedor2.com', 'Lisboa'),
(3, 'Sol Marçoso', '123-456-789', 'fornecedor3.com', 'Lisboa'),
(4, 'Da Terra', '123-456-789', 'fornecedor4.com', 'Porto'),
(5, 'Tudo Portugal', '123-456-789', 'fornecedor5.com', 'Aveiro'),
(6, 'Textotrega', '789-456-123', 'fornecedor6.com', 'Aveiro');

INSERT INTO encomenda (ID_encomenda) VALUES
(1),
(2),
(3),
(4);

INSERT INTO compra (caixa_fk, fatura_fk, produto_fk, quantidade_compra, custo) VALUES
(1, 1, 1, 500, NULL),
(1, 1, 1000, NULL),
(1, 1, 7, 30, NULL),
(1, 1, 5, 4500, NULL),
(4, 2, 14, 3000, NULL);

INSERT INTO devolucao (produto_fk, fatura_fk, data_devolucao, quantidade_devolucao, satisfacao) VALUES
(5, 2, '2019-12-31', 500, 'Muito');
(5, 2, '2019-12-31', 500, 'Muito');

INSERT INTO deslocação (produto_fk, seccao_fk, gratificacao, origem, data_deslocação, quantidade_deslocação) VALUES
(1, 1, NULL, 'Encomenda', '2019-12-31 17:29:14', 9040),
(2, 2, NULL, 'Encomenda', '2019-12-31 23:17:13', 607),
(2, 2, NULL, 'Encomenda', '2019-12-28 22:46:24', 2369),
(3, 3, NULL, 'Encomenda', '2019-12-28 18:37:17', 6200),
(3, 3, NULL, 'Encomenda', '2019-12-28 22:46:24', 550),
(4, 3, NULL, 'Encomenda', '2019-12-31 23:17:13', 40),
(4, 3, NULL, 'Encomenda', '2019-12-28 22:46:24', 50),
(5, 3, NULL, 'Encomenda', '2019-12-28 22:46:24', 1500),
(5, 3, NULL, 'Encomenda', '2019-12-28 22:46:24', 3005),
(7, 4, NULL, 'Encomenda', '2019-12-31 17:32:44', 130),
(8, 4, NULL, 'Encomenda', '2019-12-28 22:46:24', 50),
(9, 4, NULL, 'Encomenda', '2019-12-28 22:46:24', 50),
(10, 1, NULL, 'Encomenda', '2019-12-28 22:46:24', 2000),
(10, 2, NULL, 'Encomenda', '2019-12-28 22:46:24', 6000),
(11, 3, NULL, 'Encomenda', '2019-12-28 22:46:24', 500),
(12, 3, NULL, 'Encomenda', '2019-12-28 22:46:24', 300),
(13, 4, NULL, 'Arroz', '2019-12-31 23:12:50', 10),
(14, 1, NULL, 'encomenda', '2019-12-31 17:29:14', 4500);

INSERT INTO solicitacao (encomenda_fk, produto_fk, fornecedor_fk, data_encomenda, data_chegada, quantidade_solicitacao, custo_solicitacao) VALUES
(1, 1, 1, '2019-12-22 00:00:00', '2019-12-20 00:00:00', 50, 50),
(1, 1, 1, '2019-12-22 00:00:00', '2019-12-20 00:00:00', 50, 50),
(1, 7, 1, '2019-12-22 00:00:00', '2019-12-20 00:00:00', 40, 120),
(1, 1, 1, '2019-12-22 00:00:00', '2019-12-20 00:00:00', 50, 40),
(2, 2, 4, '2019-12-22 00:00:00', '2019-12-27 12:38:27', 300, 30),
(2, 2, 4, '2019-12-22 00:00:00', '2019-12-27 12:38:27', 300, 300),
(2, 10, 4, '2019-12-22 00:00:00', '2019-12-27 12:38:27', 3000, 5),
(3, 7, 6, '2019-12-29 19:01:37', '2019-12-31 17:32:44', 50, 35),
(3, 12, 6, '2019-12-29 19:01:37', '2019-12-31 17:32:44', 250, 2,7),
(4, 3, 5, '2019-12-31 23:13:57', '2019-12-31 17:29:14', 4500, 35),
(4, 14, 3, '2019-12-31 23:13:57', '2019-12-31 17:29:14', 4500, 35),
(5, 3, 5, '2019-12-31 23:13:57', '2019-12-31 23:17:13', 407, 7),
(5, 4, 5, '2019-12-31 23:13:57', '2019-12-31 23:17:13', 40, 34);

```

Figura 10 –Estado Inicial da base de dados hipermercado

Alguns valores são inseridos na base de dados para melhor entendimento de quais colunas serão listadas ou armazenadas.

Os comandos acima representam o “LMD” (Linguagem de Manipulação de Dados) e tem como objetivo demonstrar as funções utilizadas para a recuperação, inclusão, remoção e modificação de informações em bancos de dados. Uma base de dados se encontra em estado inicial após o povoamento do BD com dados iniciais.

2-Base de dados sistema login.

2.1-Modelo Entidade-Associação:

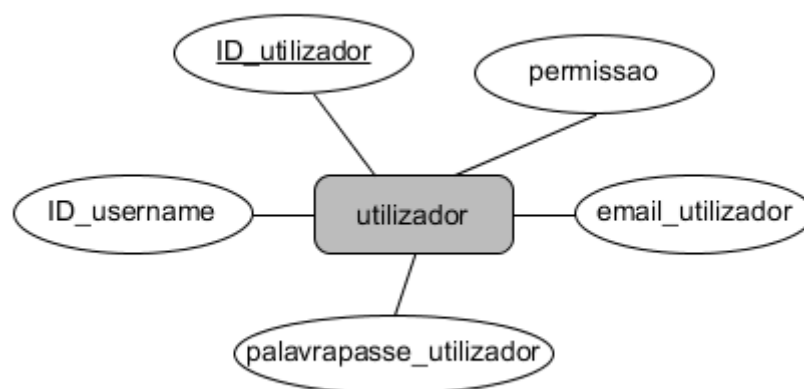


Figura 11 –Modelo entidade-associação base de dados sistema login

- Contextualização:

A figura mostra a única entidade participante dessa base de dados.

Entidade esta que armazenará todas as informações esperadas por um utilizador do sistema. seu username ou seu email poderão ser utilizados posteriormente para a autenticação nos servidores da base de dados juntamente com a palavra passe associada. A coluna permissão será abordada a fundo no próximo ponto deste capítulo e é responsável pelas permissões de acesso aos dados que um utilizador poderá exercer no sistema.

2.2-Modelo Relacional:

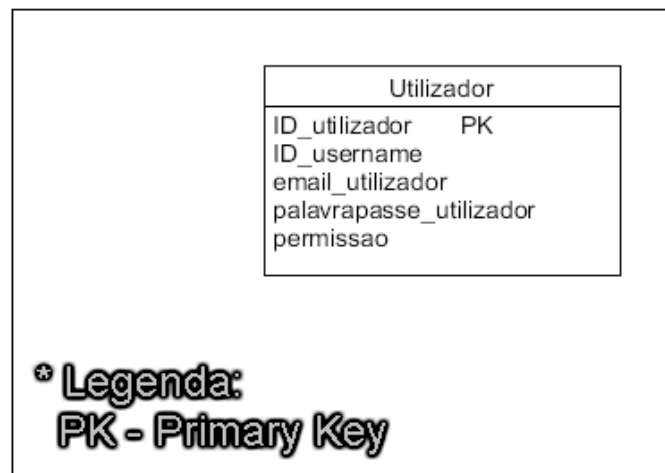


Figura 12 –Modelo relacional base de dados sistema login

- Contextualização:

O modelo relacional tem como objetivo demonstrar de uma forma mais natural todas as tabelas geradas pelo modelo entidade associação, demonstrando os atributos e os seus índices como atributo chave primário neste caso.

● Normalização das bases de dados:

- Normalização:

O método de normalização surge como forma de transformar a base de dados em uma relação arbitrária, com suas respectivas restrições e características.

1FN:

A primeira forma normal implica que nenhum atributo pode admitir um conjunto de valores, como por exemplo um produto ter duas descrições.

Todas as bases de dados desse sistema estão normalizadas na primeira forma normal.

2FN:

A segunda forma normal exige que todas as tabelas não possuam dependências parciais. Ou seja, em uma tabela que contenha duas chaves ou mais (chave composta) possuírem atributos que necessitam apenas de um elemento da chave e não a chave composta, como por exemplo se nossa tabela “solicitação” possuísse uma coluna “endereço fornecedor” que apenas dependeria da chave do fornecedor e não a chave composta entre fornecedor, produto e encomenda.

Todas as bases de dados desse sistema estão normalizadas na segunda forma normal.

3FN:

A terceira forma normal exige que todas as tabelas não possuam dependências parciais visto na 2FN e nem dependências transitivas, ou seja os atributos têm de ser dependentes apenas do atributo chave primária (PK), não podendo ser dependentes de outro atributo dentro da mesma tabela, como por exemplo se dentro de fatura tivéssemos o código do produto juntamente com sua descrição, que seria dependente apenas do código do produto e não da chave primária ID_fatura.

Todas as bases de dados desse sistema estão normalizadas na terceira forma normal.

● Permissão de acesso de dados

Apenas os administradores ou gerentes do sistema terão acesso total aos dados. Os dados da base de dados durante todo o aplicativo nunca são mostrados em totalidade ao utilizador, apenas os valores associados a uma página de listagem poderão ser vistos por todos os funcionários.

como dito anteriormente o sistema possui um plano de gestão de configurações e a permissão atribuída a coluna “permissão” da BD sistema login, essa coluna tem a função de influenciar diretamente quais dados poderão ser listados ao funcionário, ou seja, é o que permite um gerente ou um administrador, ter funcionalidades diferentes no mesmo site, como por exemplo a página de listagem de compras que poderá ser acessada apenas pelo gerente.

● Conformidade com o RGPD

O RGPD tem como objetivo proteger a privacidade dos dados pessoais dos cidadãos da União Europeia e controlar a forma como as organizações processam a utilização e o armazenamento dos dados.

Sendo obrigatório garantir os seguintes requisitos:

– Registos de tratamento de dados pessoais :

Os dados pessoais desses sistemas são todos registrados pela base de dados do sistema de login.

– Segurança de dados pessoais:

Os dados sensíveis são criptografados dentro da base de dados, fazendo com que o tratamento desses dados seja exclusivo apenas para o servidor validar e ninguém possa ter acesso a dados pessoais do utilizador.

– Notificação de incidentes de violação de dados pessoais:

Em caso de alguma violação aos dados o utilizador será notificado e as devidas atitudes tomadas em relação à violação.

– Avaliações de Impacto sobre a proteção de dados:

Constante avaliação durante o funcionamento da nossa proteção de dados como meio de auxiliar e rastrear seus impactos no sistema.

– Nomeação de um encarregado da proteção de dados:

Como dito anteriormente, a permissão de dados será designada apenas para um pequeno grupo de pessoas e essas estarão encarregadas diretamente com a proteção dos dados.

CAPÍTULO 11: PLANO DE GESTÃO DE RECURSOS

● Introdução

Para que o projeto seja bem sucedido e seus prazos sejam atingidos, os recursos têm de serem previamente identificados e devem estar presentes no início do desenvolvimento do software. Pois se não houverem recursos suficientes o projeto não avançará.

Além de quais recursos serão necessários é importante saber o quanto esse recurso será utilizado e quando será utilizado. Para que se possa medir os custos do projeto e obter boas estimativas para os prazos.

Este plano abordará quais recursos o projeto dispõe e necessitará para sua confecção final. Desde os recursos básicos, como o de apoio e infraestrutura, até os recursos mais complexos, como os tecnológicos e ferramentas utilizadas.

● Identificação dos recursos afetados ao projeto

● Recursos humanos

1 - O projeto dispõe de dois desenvolvedores/gestores de projetos.

● Recursos tecnológicos

Linguagens de Programação : PHP, JavaScript.

Linguagens de Marcação: HTML.

Servidor HTTP & Banco de Dados: XAMPP(Apache e MySQL)

Computadores

● Recursos de apoio e infraestrutura

Infraestrutura: espaço de trabalho 45m².

Apoio: templates, Curso de Engenharia de Software.

Encargos: Eletricidade, água, internet.

- Orçamento do projeto

Tipo	Identificação	Preço	Custo
Infraestrutura	Espaço de trabalho 45m ²	600€	2.5€/h
Apoio	templates, livros, curso de engenharia de software	700€	2.9€/h
Encargos	Eletricidade, água, telefone	200€	0.8€/h
TOTAL		1500€	6.25€/h

Tabela 2 - Orçamento do Projeto

CAPÍTULO 12: PLANO DE TESTES

- Objetivos do Plano de Teste

O objetivo de testes surge como uma forma de colocar a prova cada um dos requisitos feitos, se cada um deles passa pelos parâmetros decididos. sempre visando aumentar o nível de qualidade do sistema

- Descrição dos Tipos de integridade

- Integridade Física:

A integridade física está associado a garantir a forma correta de armazenar e buscar valores dos dados, para a verificação de erros na integridade dos dados, a integridade física faz uso de algoritmos de detecção de erros que são usados para manter a integridade dos dados após a transcrição manual de um sistema de computador para outro por um intermediário humano.

- Integridade Lógica:

A integridade lógica procura garantir a correção e a racionalidade de um dado garantido a coesão, ou seja o “sentido” dos dados em seu devido ambiente, a

integridade lógica utiliza de implementações de restrições de checagem de chaves estrangeiras, programa de asserções, e outras verificações de sanidade em tempo de execução.

● Descrição dos testes a realizar e dos níveis de testes

● Níveis de teste:

-Unitário

-integração

-Sistema

-Aceitação

● Testes unitários:

Testes unitários estão ligados às menores unidades do software, esse tipo de teste é feito em sub-rotinas, métodos, classes ou mesmo pequenos trechos de código, o teste tem como objetivo encontrar algum mau funcionamento do sistema, uma falha dentro desta unidade do sistema e serão realizados por desenvolvedores.

● Testes de integração :

O teste de integração surge com o objetivo de encontrar falhas provenientes da integração interna dos componentes de um sistema. Geralmente os tipos de falhas encontradas são de transmissão de dados, ou seja uma componente está retornando ou guardando um valor não desejado, uma falha no sistema. O teste de integração conduz à descoberta de possíveis falhas associadas à interface do sistema. Não faz parte do escopo dessa fase de teste o tratamento de interfaces com outros sistemas (integração entre sistemas).

● Testes de sistema:

Na fase de teste de sistema, o objetivo é testar o sistema sob ponto de vista de seu usuário final, em busca de alguma falha ou desvio dos objetivos do sistema. Os testes são executados em condições que um usuário utilizaria o sistema em um dia comum.

● Testes de aceitação:

Os testes de aceitação serão realizados por um grupo restrito de usuários finais do sistema, que executam o sistema em condições habituais como meio de verificar se

seu comportamento está de acordo com o solicitado e os critérios de aceitação pelo utilizador que irá decidir se aceita ou não o sistema. Pode incluir testes funcionais, de configuração, de recuperação de falhas, de segurança e de desempenho.

● Descrição de como são gerados os dados para os testes

O sistema permite a inicialização do teste com dados vazios, pois o próprio site possibilita o utilizador a inserir valores de forma dinâmica. Como forma de garantir a máxima eficiência do sistema, os dados gerados para o teste serão os mesmos utilizados para inicialização de base de dados (Capítulo 10).

● Descrição dos testes de integração dos módulos

A integração pode seguir abordagem incremental ou não. Enquanto na abordagem não-incremental o sistema é agrupado por completo, na abordagem incremental o sistema é agrupado em etapas, facilitando assim o isolamento do erro. Abordagens incrementais diferem entre si na forma em que se constrói as etapas de agrupamento de módulos. Por exemplo, na abordagem descendente os módulos de alto nível são testados e integrados primeiro, permitindo encontrar primeiro os erros de lógica e fluxo de dados de alto nível. Por outro lado, a abordagem ascendente requer o teste e integração dos módulos de baixo nível primeiro.

CAPÍTULO 13: PLANO DE TREINO

● Métodos de treino:

Os métodos de treinos desse sistema procuram incentivar a aproximação do cliente com o sistema, o primeiro deles é por meios presenciais, diferente de um texto, um método desses cria uma conexão muito maior entre o cliente e o sistema, com a deslocação de um profissional que entenda o sistema e comprometa-se as horas de treinamento. O utilizador deve usar desse tratamento presencial para comunicar com o profissional sobre todas as funcionalidades existentes no sistema assim como qualquer dúvida sobre algum procedimento específico, juntamente com a explicação e o treino do cliente ao software, espera-se um registro ou digital que possa sempre estar em constante atualização do que foi lecionado no dia, acerca de funcionamento geral e dúvidas não permitindo algum tipo de perda de informação sobre o treinamento e possibilitando o utilizador revisar sempre que necessário posteriormente.

Além do método presencial, espera-se um método de treino por conferências e transmissões ao vivo, seguindo o mesmo exemplo de um meio presencial, o profissional deve estar preparado para sanar qualquer tipo de dúvida do sistema com

uma linguagem de fácil entendimento ao utilizador, qualquer treinamento também passa por um registro que servirá como apoio constante aos utilizadores do sistema o método por conferência possui um custo menor que o presencial devido a praticidade de não ter de deslocar para um sessão de treinamento do sistema.

● Recursos para cada método:

● Método presencial:

- Recursos humanos

O método deve contar com um profissional do sistema que possa treinar o utilizador.

- Recursos tecnológicos

Um computador para a execução do sistema e para o armazenamento dos registros de cada um dos dias lecionados.

- Recursos de apoio e infraestrutura

Infraestrutura: Uma sala para a realização do treinamento

Encargos: Eletricidade, água, internet.

● Método Conferências e transmissões ao vivo:

- Recursos humanos

O método deve contar com um profissional do sistema que possa treinar o utilizador.

- Recursos tecnológicos

Um computador para a execução do sistema e para o armazenamento dos registros de cada um dos dias lecionados.

Serviço de compartilhamento de tela e bate papo para comunicação e demonstração ativa do treinamento.

- Recursos de apoio e infraestrutura

Encargos: Eletricidade, água, internet.

● Bases de dados do treino

Com o intuito de maior eficiência possível no treinamento dos utilizadores, as bases de dados utilizadas no treino serão uma cópia das mesmas utilizadas no sistema principal.

● Duração do treino

O treino terá a duração por sessão em um mínimo de 1 hora e 30 minutos de treinamento e um máximo de qualquer horário combinado e aceito entre profissional e utilizador. A duração total do treino é relacionada com o aprendizado do utilizador, os treinos não podem acabar sem que o utilizador saiba efetivamente utilizar todas as funcionalidades desejadas e consiga entender o sistema como um todo.

● Número de participantes e responsabilidades

O número de participantes tem de sempre estar em níveis aceitáveis e não exagerados para evitar um mal treinamento pela incapacidade do contacto direto com o utilizador, o que apenas irá atrasar o plano de treino de modo geral. Em cada método de treino o número de participantes pode variar, com o método presencial admitindo um número menor de pessoas e o método por conferências podendo aceitar mais participantes, mas sempre tendo em mente o número de participantes viáveis para um bom treinamento.

● Avaliação

A avaliação do treino consiste em analisar a performance do utilizador no sistema após os treinos lecionados, espera-se que o mesmo consiga utilizar do sistema sem algum problema ou dúvida, isso é considerado um plano de testes com sucesso.

CAPÍTULO 14: PLANO DE SEGURANÇA

● Segurança no desenvolvimento

O sistema será desenvolvido com a segurança de dados sensíveis sendo a primeira prioridade em áreas que necessitam do mesmo, como o sistema de autenticação do sistema, que trata todos os dados de uma forma a impedir algum ataque ao site.

● Testes de segurança

Os testes de segurança existem como forma de avaliar as vulnerabilidades do sistema em todos os módulos que possam apresentar uma abertura a diferentes tipos de ataques. O teste de segurança permite encontrarmos essas vulnerabilidades antes que elas possam ser exploradas.

● Segurança no alojamento da base de dados

A base de dados é instrumento primordial para o funcionamento do sistema e devido a essa importância é prioridade garantir a segurança da mesma.

As tarefas de administração serão divididas entre a equipe de desenvolvedores e usuários com permissões posteriores , essa divisão de tarefas diminui o risco de usuários poderem abusar de seus privilégios e também reduz a área da superfície de ataque no caso de contas invadidas.

Cada usuário de uma organização deve ter uma conta individual nomeada, cada conta nomeada é então vinculada a privilégios personalizados, selecionados de acordo com a função dessa pessoa dentro da organização, contas não devem ser compartilhadas em hipótese alguma. Um log sobre cada uma das contas é esperado, trazendo melhorias significativas na segurança.

Os logs são necessários para o rastreamento do que cada conta usa e deixa de usar na base de dados. Fazendo um registro irrefutável das ações realizadas por contas nomeadas, com as informações da hora do acontecimento. Esses registros ajudam na organização e gestão da segurança base de dados

Algumas soluções podem ser implementadas como a utilização de uma DBF database firewall que permite a análise de todo o tráfego ao banco de dados, protegendo em tempo real contra ataques e possuindo a funcionalidade de patching virtual, função essa que protege a base de dados contra vulnerabilidades sem a atualização da base de dados.

● Segurança contra acessos não autorizados

A base de dados não será acessível a qualquer pessoa, com sua localização sendo acessível apenas ao grupo de desenvolvedores e funcionando com sistemas de autenticação linkados a contas nomeadas e específicas para cada integrante, o acesso não autorizado é barrado pelo sistema de autenticação da própria base de dados.

● Integridade dos dados

A integridade dos dados surge como forma de garantir a precisão e consistência (validade) durante todo o seu ciclo de vida, este é um termo para estender o bom funcionamento dos dados. Todos os dados passam por backups constantes para garantir a integridade e a segurança contra qualquer inconveniência, os dados poderão ser repostos e a base de dados poderá voltar ao seu ciclo de vida normal, A base de dados do sistema passa por verificações na integridade de todas as colunas e tabelas verificando a melhor eficiência no armazenamento, garantindo toda e qualquer condição para o mesmo, como por exemplo um valor que não poderá ter 3 valores decimais ou um campo de nome que não poderá ultrapassar 50 caracteres essas ações em prol da integridade dos dados é essencial para precisão e eficiência dos processos do negócio.

● Realização de backups a base de dados e a aplicação

A perda de informação é um dos riscos que mais pode prejudicar o sistema e tratando o problema da forma que se deve, os backups da base de dados são realizados diariamente e após toda e qualquer tipo de modificação um backup será criado, garantindo que nenhum dado possa ser perdido e possam ser repostos caso aconteça algum erro inesperado.

CAPÍTULO 15: PLANO DE GESTÃO DE RISCO

● Introdução

Risco é o efeito (positivo ou negativo) de um evento ou de uma série de eventos que se manifesta em um ou em vários locais. Ele é calculado a partir da probabilidade deste evento se manifestar e do impacto que ele poderia causar. Alguns elementos devem ser identificados para se analisar riscos, incluindo:

- **Probabilidade:** com que frequência ele poderia acontecer?
- **Impacto:** o quão ruim será se ele acontecer?
- **Mitigação:** Como você pode reduzir a sua probabilidade (e quanto poderia reduzir)?
- **Contingência:** Como você poderia reduzir seu impacto (e quanto poderia reduzir)?

● Impactos dos riscos

Legenda de Impactos:

Impacto 1 - Negligenciável Impacto 2 - marginal

Impacto 3 - crítico Impacto 4 - Catastrófico

#	Risco	Prob.	Impacto
1	Modificação dos requisitos	80%	2
2	Pessoal sem Experiência	30%	2
3	Requisitos mal detalhados	50%	3
4	Imprevistos com Recursos	50%	3
5	Atraso na completude dos marcos do projeto	60%	3
6	idéias e conceitos novos	80%	2

Tabela 3 - Riscos e suas propriedades

● Identificação e análise dos riscos

1- Modificação dos requisitos por parte do cliente:

O cliente pode vir a querer modificar, remover ou adicionar requisitos durante as fases de desenvolvimento do projeto, já que o contrato pode vir a permitir isso. Por esse motivo e por outros que os requisitos podem sofrer modificações no “meio” do projeto e a probabilidade disso acontecer é bem alta como apresentada na tabela 3.

Seu impacto não é tão grande, mas como sua probabilidade é alta, se não tivermos cuidado com o controle de alterações e a gestão de requisitos, rapidamente a situação foge do controle e vários requisitos podem ser modificados sem nenhuma rastreabilidade.

2- Pessoal sem Experiência:

Integrantes da equipe que acabara de ser contratados e ainda não estão familiarizados com o projeto, ou até mesmo integrantes antigos que estão utilizando/testando uma nova tecnologia no projeto, podem apresentar dificuldades pela falta de experiência e isso afeta diretamente nos prazos do projeto, já que esses

indivíduos precisam de um tempo para se adaptar e adquirir mais conhecimento sobre o projeto e ferramentas para a confecção do mesmo.

3 - Requisitos mal detalhados:

Os requisitos quando mal detalhados geram muitos problemas com os prazos e custos finais e insatisfação por parte do cliente, podendo levar o projeto ao seu declínio. Sua probabilidade é mediana, porém possui drásticas consequências como citadas acima, então não devem restar dúvidas e ambiguidades quando o assunto for: “escrever uma especificação de requisitos”.

Para isso, devem ser organizadas algumas reuniões com os todos interessados, para se chegar a uma única idéia sobre o que o sistema deve suportar e quais funcionalidades devem ter. Após isso, deve ser apresentado para o cliente um esboço desse documento para análise dos requisitos, esboço esse, que quando validado, fica acordado entre cliente e desenvolvedor o que deve se fazer até esse momento.

4 - Imprevisto com os recursos:

O esgotamento de um recurso, pode levar um projeto à falência. Sua chance de ocorrência é mediana, porém suas consequências podem ser críticas. Mas esse impacto pode variar de recurso a recurso, onde alguns são mais dispensáveis, não afetando tanto o desenvolvimento do projeto. Enquanto outros são indispensáveis, devendo estar presente durante toda a etapa de desenvolvimento do projeto.

5 - Atraso na completude dos marcos do projeto:

Um dos grandes riscos é a gestão do tempo e da entrega dos artefatos ao cliente, já que inúmeros imprevistos podem acontecer e todos eles podem gerar um aumento nas datas de entrega. Seu impacto pode variar e a sua probabilidade é relativamente alta.

6 - idéias e conceitos novos:

Ideias e conceitos novos sempre surgem, tanto por parte do cliente quanto dos desenvolvedores. Pois passam a ter uma maior percepção do que o sistema deve ter após o início do desenvolvimento do mesmo. Seu impacto não é tão expressivo, mas como sua probabilidade é muito alta, ou seja, isso irá acontecer algumas vezes(mesmo que poucas) deve-se dar uma atenção especial.

● Plano de Resposta aos riscos identificados

1- Modificação dos requisitos por parte do cliente:

A gestão de requisitos surge para minimizar esse transtorno com atividades de controle e documentação dessas alterações. Deve ser adotado um modelo de gestão de requisitos, como por exemplo: o de Kotonya Sommerville. Onde as alterações são controladas e rastreadas, e o documento de especificação e requisitos passa a ter versões identificando essas alterações, os responsáveis e as datas.

2- Pessoal sem Experiência:

Treinar o mais rápido possível e acompanhar o dia-a-dia de trabalho desses indivíduos, tentar estender alguns prazos(quando houver flexibilidade por parte do cliente). Ter uma equipe mais experiente, disposta a treinar em horários especiais esse pessoal novo ou sem experiência.

3 - Requisitos mal detalhados:

A equipe de desenvolvedores deve marcar imediatamente uma reunião para esclarecimento das dúvidas e resolução dos problemas encontrados com os requisitos. Essas mudanças, bem como os motivos, devem ser todas documentadas.

4 - Imprevisto com os recursos:

Todos os recursos devem ser previamente planejados com uma boa margem de erro, caso se utilize mais do que se achava necessário no início do projeto. Utilizar sempre o mínimo necessário dos recursos a fim de economizá-los e obter maior eficiência nos resultados. Outro plano de resposta, é ter uma reserva extra de recursos, para ser utilizada em casos extremos ou até mesmo, em último caso, recorrer a um empréstimo bancário para a finalização do projeto.

5 - Atraso na completude dos marcos do projeto:

O projeto deve ser calendarizado com margens realísticas de erro, e as datas definidas têm que ser seguidas à risca. A adoção de softwares de gestão de recursos com diagramas de gantt pode ser fundamental, além de um bom líder de equipe que incentive e ajude a equipe a cumprir as metas em seus devidos prazos.

6 - idéias e conceitos novos:

O plano de resposta desse risco segue os parâmetros do risco “1” da tabela x, podendo ser mais flexível caso essas idéias e conceitos sejam descobertos antes do início do desenvolvimento do projeto.

CAPÍTULO 16: PLANO DE MANUTENÇÃO

- **Identificação do modelo a adoptar para a manutenção do produto**

A manutenção do produto será feita pessoalmente e diretamente na máquina do utilizador, um desenvolvedor irá instalar todos os processos necessários, deixando o produto pronto sem qualquer margem de erro possível em outros métodos como a instalação por web, toda a instalação poderá ser acompanhada pelo utilizador, o que permitirá uma maior noção do que acontece de manutenção para manutenção, um ficheiro de registos sobre a manutenção também oferecerá essa informação.

- **Apoio ao utilizador após a entrega do produto**

Após a entrega do produto o utilizador poderá e deve contactar os desenvolvedores acerca de dúvidas ou algum erro no sistema, os desenvolvedores por sua vez estarão preparados para aplicar qualquer tipo de manutenção necessária, tal como prover qualquer tipo de apoio que o utilizador final precisa acerca do sistema.

- **Periodicidade da manutenção**

A manutenção do sistema acontecerá acompanhada de qualquer tipo de modificação ao sistema, seja ela novas versões do sistema ou alguma correção que precisa ser feita ao sistema atual.

- **Ações a realizar na manutenção**

O desenvolvedor deve atualizar todos os módulos que se espera a manutenção, promovendo as novas atualizações ou correções de erros, é dever do mesmo entregar uma nova versão com todos os problemas solucionados sejam eles em usabilidade, execução ou até mesmo alguma melhoria na interface do produto, toda área que verificada com alguma carência em sua funcionalidade deverá passar por ações de solução ao problema.

- **Responsáveis pela manutenção**

A equipa de desenvolvedores são responsáveis diretos por toda a manutenção do sistema.

CAPÍTULO 17: CONCLUSÕES

● Reflexões Finais

Planejar o desenvolvimento de um software é uma tarefa difícil, complexa e que exige pessoas com experiência. Os vários diferentes documentos que necessitam ser produzidos para uma boa gestão de software, demanda da contratação de equipes qualificadas para cada área de cada plano (qualidade, testes, gestão de dados, manutenção, etc.).

Antes desta contratação é necessário saber quais planos serão feitos. Quanto maior for o nível de planejamento de uma empresa, maior será a necessidade de contratar pessoal qualificado nas respectivas áreas. E isso aumentará os custos com os recursos. Embora os custos aumentem, as empresas com grande maturidade adotam altos níveis de planejamento, podendo segregar cada área para uma respectiva equipe, pois os resultados obtidos na maioria dos casos é muito superior, o que aumenta a sua imagem no mercado, além de produzir um produto (softwares) de melhor qualidade.

● Áreas a melhorar

Melhor calendarização e planejamento dos prazos, melhor gestão do tempo de trabalho.

● Expectativas de trabalho futuro

maior eficiência, proporcionada pelo contato com os diferentes documentos ao longo de todo semestre, aprimorando assim expressivamente os conhecimentos de planejamento e gestão de software como um todo.

REFERÊNCIAS

O livro A Guide to the Project Management Body of Knowledge (PMBOK Guide).

http://www.techoje.com.br/site/techoje/categoria/detalhe_artigo/242

[https://edisciplinas.usp.br/pluginfile.php/33094/mod_resource/content/1/Gerenciamento de Qualidade.pdf](https://edisciplinas.usp.br/pluginfile.php/33094/mod_resource/content/1/Gerenciamento_de_Qualidade.pdf)

<http://www.fapesp.br/gestaodedados/>

<http://www.sibi.usp.br/apoio-pesquisador/dados-pesquisa/plano-gestao-dados-2/>

<https://www.siteware.com.br/projetos/planejamento-de-recursos/>

<https://www.devmedia.com.br/plano-de-teste-um-mapa-essencial-para-teste-de-software/13824>

<https://medium.com/cwi-software/planejamento-de-testes-de-software-d2c1a1dd0951>

https://edisciplinas.usp.br/pluginfile.php/3385127/mod_resource/content/1/Aula10-GerenciaProjeto-Riscos.pdf

<https://blog.softexpert.com/plano-de-riscos-completo-12-etapas/>