**DOCUMENTO DE REQUISITOS**

**EMPRESA Quimera HO**

**Sistema SoftCooks**

**Lajeado, Novembro de 2015**

**Histórico de Alterações**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| 24/11/15 | 5.0 | Finalização do documento. | Lucas/Jonas |
| 15/11/15 | 4.1 | Alteração nas especificações dos requisitos funcionais. | Lucas/Jonas |
| 27/10/15 | 4.0 | Alteração nas especificações dos requisitos funcionais. | Lucas/Jonas |
| 12/10/15 | 3.0 | Alterações nos requisitos. | Lucas/Jonas |
| 26/08/15 | 3.0 | Alteração na descrição do negócio, requisitos, objetivo. | Jonas |
| 12/08/15 | 2.0 | Alterações nos requisitos, negócio a ser atendido e objetivo do projeto. | Lucas/Jonas |
| 05/08/15 | 1.0 | Criação do documento de requisitos. | Lucas/Jonas |
|  |  |  |  |

Equipe de desenvolvimento do projeto:

|  |  |
| --- | --- |
| Nome | Função |
| Lucas | Análise/Desenvolvimento |
| Jonas | Análise/Desenvolvimento |

**SUMÁRIO DO DOCUMENTO**

1. INTRODUÇÃO

1.1 Convenções e abreviaturas

Um requisito pode receber em sua descrição a notação “Inválido”. Isso significa que o mesmo foi invalidado. O objetivo é manter um histórico do documento, já que este não encontra-se versionado.

**1.1.1 Nomenclatura dos requisitos**

Identificação dos requisitos: Os requisitos funcionais e não funcionais serão referenciados no documento sob a nomenclatura 'RF0001' e 'RNF0001', respectivamente. As letras significam Requisito Funcional e Requisito Não-Funcional, e os números representam uma sequência crescente a partir de um.

**1.1.2 Prioridade dos requisitos**

|  |  |
| --- | --- |
| Prioridade | Descrição |
| Alta | O sistema deve atender a este requisito. |
| Média | O sistema deveria implementar este requisito. |
| Baixa | O sistema poderia atender a este requisito. |

**1.1.3 Demais convenções ou padrões do documento**

**2. DESCRIÇÃO DO NEGÓCIO**

|  |
| --- |
| Alimentar-se é uma das necessidades básicas do ser humano, e para pessoas com um tino empreendedor, a comercialização de lanches, pratos prontos e bebidas, mostra-se uma ótima oportunidade de renda. Estabelecimentos como lanchonetes restaurantes encaixam-se neste perfil de negócio. Onde o rápido atendimento dos clientes, ambiente climatizado e agradável, além da qualidade dos produtos oferecidos, é um pré-requisito para o bom desempenho financeiro da empresa e do sucesso perante o público.  Em uma sociedade onde a comodidade e a tecnologia formam uma dupla de sucesso, possibilitar ao cliente, uma forma de realizar pedidos de seus pratos favoritos, sem sair de casa, é um diferencial de peso. O atendimento delivery, adapta-se às necessidades de uma sociedade, que cada vez mais, prioriza a flexibilidade no atendimento, além da boa comida.  A possibilidade da realização de pedidos, sem a necessidade de sair de casa, realizando o pagamento através de variadas formas de pagamento, com certeza é um grande diferencial. Outro ponto cativante encontra-se na agilidade na entrega, somado a uma grande variedade de pratos, que após seu pedido pode ter seu andamento acompanhado via app mobile ou web. |

**3. OBJETIVOS DO SISTEMA**

1. Controlar entrada de pedidos on-line.
2. Disponibilizar variadas formas de pagamento.
3. Informar ao cliente o status do seu pedido.
4. Gerenciar cardápio digital.
5. Gerenciar situação dos pratos.

**4. ATORES**

|  |  |
| --- | --- |
| 1. Cliente | 6. |
| 2. Funcionário | 7. |
| 3. Sistema | 8. |
| 4. | 9. |
| 5. | 10. |

**5. REQUISITOS FUNCIONAIS**

|  |  |  |
| --- | --- | --- |
| N° | Descrição do requisito | Prioridade |
| RF0001 | Manter cadastro de clientes. | Alta |
| RF0002 | Manter cadastro de funcionários. | Alta |
| RF0003 | Manter cadastro de cardápios. | Alta |
| RF0004 | Manter cadastro de pratos. | Alta |
| RF0005 | Manter cadastro de ingredientes. | Alta |
| RF0006 | Manter cadastro de condimentos. | Alta |
| RF0007 | Manter cadastro de temperos. | Alta |
| RF0008 | Receber pagamentos on-line | Alta |
| RF0009 | Possibilitar o cancelamento de pedidos. | Baixa |
| RF0010 | Informar ao usuário a situação do pedido | Média |
| RF0011 | Manter cadastro de interfaces/telas/processos do sistema. | Média |
| RF0012 | Oferecer pratos conforme estoque de ingredientes condimentos e temperos.  Inválido: Necessidade questionada. | Média |
| RF0013 | Criar e oferecer promoções. | Média |
| RF0014 | Receber avaliação do cliente. | Média |
| RF0015 | Manter cadastro de bebidas. | Alta |
| RF0016 | O sistema deve possuir sistema de gerenciamento de permissões por grupo de usuário. | Alta |
| RF0017 | O sistema deve disponibilizar interface para controlar entrada e saída de pratos e cardápios do sistema. | Alta |
| RF0018 | O sistema deve permitir o cadastro de tempo de cancelamento dos pedidos. | Média |
| RF0019 | O sistema deve permitir o cadastro de grupo de usuários. | Alta |
| RF0020 | O sistema deve permitir atribuir permissões para grupos de usuário.  Inválido: É um requisito RNF | Alta |
| RF0021 | O sistema deve permitir atribuir um grupo a um usuário para o controle de permissões.  Inválido: É um requisito RNF | Alta |
| RF0022 | O sistema deve permitir que um funcionário altere o status de um pedido. | Alta |
| RF0023 | O sistema deve interagir com dispositivos móveis.  Inválido: É um requisito RNF | Alta |
| RF0024 | O sistema deve disponibilizar uma interface para interação com os pedidos, por parte do funcionário. | Alta |
| RF0025 | O sistema deve possibilitar a realização de pedidos, por parte do cliente. | Alta |

**5.1 Especificação**

|  |  |  |  |
| --- | --- | --- | --- |
| RF0001 – Manter cadastro de clientes. | | | |
| O sistema deve permitir inserir, apagar, editar e listar clientes no sistema, sendo que, a ação de listar clientes esteja restrita a funcionários cadastrados e com as devidas permissões no sistema.  Informações como nome, idade, data de nascimento, CPF, RG, nome da mãe, sexo, telefone, celular, e-mail, endereço, cidade, bairro e senha de acesso estão previstas para este cadastro. | | | |
| Prioridade | Complexidade | Situação | Versão |
| Alta | Baixa | Aprovado | 2.0 |

|  |  |  |  |
| --- | --- | --- | --- |
| RF0002 – Manter cadastro de funcionários. | | | |
| O sistema deve permitir inserir, apagar, editar e listar funcionários no sistema, esta ação esta restrita a funcionários cadastrados com as devidas permissões no sistema, ou seja, clientes não poderão manipular dados de funcionários. Informações como nome, idade, data de nascimento, CPF, RG, nome da mãe, sexo, telefone, celular, e-mail, endereço, cidade, bairro, salário, data de admissão, data de desligamento e senha de acesso, estão previstas para este cadastro. | | | |
| Prioridade | Complexidade | Situação | Versão |
| Alta | Média | Aprovado | 4.0 |

|  |  |  |  |
| --- | --- | --- | --- |
| RF0003 – Manter cadastro de cardápios. | | | |
| O sistema deve permitir inserir, apagar, editar e listar cardápios no sistema, esta ação esta restrita a funcionários cadastrados com as devidas permissões no sistema. Informações como descrição do cardápio e uma lista de pratos estão previstas para este cadastro. | | | |
| Prioridade | Complexidade | Situação | Versão |
| Alta | Média | Aprovado | 4.0 |

|  |  |  |  |
| --- | --- | --- | --- |
| RF0004 – Manter cadastro de pratos. | | | |
| O sistema deve permitir inserir, apagar, editar e listar pratos no sistema, esta ação está restrita a funcionários cadastrados com as devidas permissões no sistema. Informações como descrição do prato, preço, status, tipo, tamanho, lista de temperos, lista de bebidas, lista de ingredientes e lista de condimentos, estão previstas para este cadastro. | | | |
| Prioridade | Complexidade | Situação | Versão |
| Alta | Baixa | Aprovado | 4.0 |

|  |  |  |  |
| --- | --- | --- | --- |
| RF0005 – Manter cadastro de ingredientes. | | | |
| O sistema deve permitir inserir, apagar, editar e listar ingredientes no sistema, esta ação está restrita a funcionários cadastrados com as devidas permissões no sistema. Apenas a descrição do ingrediente esta prevista para este cadastro. | | | |
| Prioridade | Complexidade | Situação | Versão |
| Alta | Baixa | Aprovado | 4.0 |

|  |  |  |  |
| --- | --- | --- | --- |
| RF0006 – Manter cadastro de condimentos. | | | |
| O sistema deve permitir inserir, apagar, editar e listar condimentos no sistema, esta ação esta restrita a funcionários cadastrados com as devidas permissões no sistema. Apenas a descrição do condimento esta prevista para este cadastro. | | | |
| Prioridade | Complexidade | Situação | Versão |
| Alta | Baixa | Aprovado | 4.0 |

|  |  |  |  |
| --- | --- | --- | --- |
| RF0007 – Manter cadastro de temperos. | | | |
| O sistema deve permitir inserir, apagar, editar e listar temperos no sistema, esta ação esta restrita a funcionários cadastrados com as devidas permissões no sistema. Apenas a descrição do tempero esta prevista para este cadastro. | | | |
| Prioridade | Complexidade | Situação | Versão |
| Alta | Baixa | Aprovado | 4.0 |

|  |  |  |  |
| --- | --- | --- | --- |
| RF0008 – Receber pagamento on-line | | | |
| O sistema deve estar preparado para receber pagamentos on-line. Em tais pagamentos, devem ser aceitos os cartões de crédito Visa, MasterCard, Diners e American Express. Além de pagamentos vi PayPal.  O gateway de pagamento utilizado será o TrySoft. O sistema deve permitir a configuração do parcelamento via cartão de crédito, atribuindo o número máximo e mínimo de parcelas.  O sistema também deve implementar de forma transparente a “retentativa” de pagamentos. | | | |
| Prioridade | Complexidade | Situação | Versão |
| Alta | Alta | Aprovado | 3.0 |

|  |  |  |  |
| --- | --- | --- | --- |
| RF0009 – Possibilitar o cancelamento de pedidos. | | | |
| O sistema deve possibilitar ao usuário o cancelamento de determinado pedido.  Deve existir um botão Cancelar, na tela de pedidos do usuário. Tal botão deve aparecer apenas para pedidos com status Em andamento, e que o tempo de solicitação não tenha ultrapassado um tempo configurado no sistema. Quando um cancelamento é efetivado, um funcionário com devidas permissões deve ser notificado. A notificação será feita através da tela de confirmação de pedidos. | | | |
| Prioridade | Complexidade | Situação | Versão |
| Média | Média | Aprovado | 3.0 |

|  |  |  |  |
| --- | --- | --- | --- |
| RF0010 – Informar ao cliente a situação do pedido | | | |
| O status do pedido será alterado quando:  Um pedido é efetuado, um pedido é efetivado por um funcionário, um pedido é dado como pronto pelo funcionário e quando um pedido é dado como entregue, também através de um funcionário. Estas mudanças de status serão mostradas para o cliente através da interface de pedidos e também do aplicativo para celular. | | | |
| Prioridade | Complexidade | Situação | Versão |
| Média | Média | Aprovado | 3.0 |

|  |  |  |  |
| --- | --- | --- | --- |
| RF0011 – Manter cadastro de interfaces/telas/processos do sistema. | | | |
| O sistema deve permitir inserir, apagar, editar e listar interfaces/telas/processos no sistema, esta ação esta restrita a funcionários cadastrados com as devidas permissões no sistema. Apenas a descrição esta prevista para este cadastro. | | | |
| Prioridade | Complexidade | Situação | Versão |
| Média | Média | Aprovado | 3.0 |

|  |  |  |  |
| --- | --- | --- | --- |
| RF0012 – Oferecer pratos conforme estoque de ingredientes condimentos e temperos. - Inválido | | | |
|  | | | |
| Prioridade | Complexidade | Situação | Versão |
| Média | Alta | Aprovado | 3.0 |

|  |  |  |  |
| --- | --- | --- | --- |
| RF0013 – Criar e oferecer promoções. | | | |
| O sistema deve possibilitar criar promoções para pratos informados.  A promoção deve receber um prato seu novo preço e opcionalmente um acompanhamento assim como uma data de fim.  As promoções devem receber destaque na tela inicial do site. De ser exibida também sua validade.  O sistema deve retirar a promoção do ar, caso a data atual seja maior ou igual à data cadastrada. | | | |
| Prioridade | Complexidade | Situação | Versão |
| Média | Baixa | Aprovado | 3.0 |

|  |  |  |  |
| --- | --- | --- | --- |
| RF0014 – Receber avaliação do cliente. | | | |
| O sistema deve disponibilizar para o cliente a possibilidade de avaliar cada pedido finalizado.  Tal avaliação será feita através de carinhas felizes, tristes e sérias. Além de um formulário para observações que o usuário queira exteriorizar. | | | |
| Prioridade | Complexidade | Situação | Versão |
| Média | Baixa | Aprovado | 3.0 |

|  |  |  |  |
| --- | --- | --- | --- |
| RF0015 – Manter cadastro de bebidas. | | | |
| O sistema deve permitir inserir, apagar, editar e listar bebidas no sistema, esta ação esta restrita a funcionários cadastrados com as devidas permissões no sistema. Apenas a descrição da bebida esta prevista para este cadastro | | | |
| Prioridade | Complexidade | Situação | Versão |
| Alta | Baixa | Aprovado | 4.0 |

|  |  |  |  |
| --- | --- | --- | --- |
| RF0016 – O sistema deve possuir sistema de gerenciamento de permissões por grupo de usuário. | | | |
| O sistema deve disponibilizar para funcionários com devidas permissões, uma interface para que sejam cadastradas permissões de inserir, atualizar, apagar, listar, executar e administrar determinadas telas/interfaces/processos do sistema.  Estas configurações devem estar vinculadas a um grupo de usuários. | | | |
| Prioridade | Complexidade | Situação | Versão |
| Alta | Alta | Aprovado | 4.0 |

|  |  |  |  |
| --- | --- | --- | --- |
| RF0017 – O sistema deve disponibilizar interface para controlar entrada e saída de pratos e cardápios do sistema. | | | |
| O sistema deve disponibilizar para funcionários com devidas permissões, uma interface para controlar entrada e saída de pratos e cardápios do sistema. Nesta tela deve ser possível escolher quais pratos e ou cardápios serão mostrados para o usuário no site. | | | |
| Prioridade | Complexidade | Situação | Versão |
| Média | Média | Aprovado | 4.0 |

|  |  |  |  |
| --- | --- | --- | --- |
| RF0018 – O sistema deve permitir o cadastro de tempo de cancelamento dos pedidos. | | | |
| O sistema deve permitir inserir, editar e listar o tempo de cancelamento, esta ação esta restrita a funcionários cadastrados com as devidas permissões no sistema. Apenas o tempo de cancelamento esta previsto para este cadastro | | | |
| Prioridade | Complexidade | Situação | Versão |
| Média | Baixa | Aprovado | 4.0 |

|  |  |  |  |
| --- | --- | --- | --- |
| RF0019 – O sistema deve permitir o cadastro de grupo de usuários. | | | |
| O sistema deve permitir inserir, editar e listar grupos de usuário, esta ação esta restrita a funcionários cadastrados com as devidas permissões no sistema. Apenas a descrição do grupo de usuário esta previsto para este cadastro. | | | |
| Prioridade | Complexidade | Situação | Versão |
| Média | Baixa | Aprovado | 4.0 |

|  |  |  |  |
| --- | --- | --- | --- |
| RF0020 – O sistema deve permitir atribuir permissões para grupos de usuário. | | | |
| O sistema deve permitir a atribuição de permissões a grupos de usuário, esta ação esta restrita a funcionários cadastrados com as devidas permissões no sistema. Para utilizar esta funcionalidade, o funcionário deve selecionar um grupo, para depois, escolher quais permissões estarão atribuídas a determinadas telas/interfaces/processos. | | | |
| Prioridade | Complexidade | Situação | Versão |
| Alta | Alta | Aprovado | 4.0 |

|  |  |  |  |
| --- | --- | --- | --- |
| RF0021 – O sistema deve permitir atribuir um grupo a um usuário para o controle de permissões. | | | |
| O sistema deve permitir a atribuição de usuários a um ou mais grupos, esta ação esta restrita a funcionários cadastrados com as devidas permissões no sistema. Para utilizar esta funcionalidade, o funcionário deve selecionar um usuário, para depois, escolher com quais grupos este estará vinculado. | | | |
| Prioridade | Complexidade | Situação | Versão |
| Média | Alta | Aprovado | 4.0 |

|  |  |  |  |
| --- | --- | --- | --- |
| RF0022 – O sistema deve permitir que um funcionário altere o status de um pedido. | | | |
| O sistema deve permitir através da interface de interação do funcionário, alterar o status de determinado pedido. Tal alteração deve seguir as seguintes regras.  Caso o status esteja Em Andamento, ele só poderá passar para o status Na Cozinha ou Fechado.  Caso o status esteja Na Cozinha, ele só poderá passar para o status A Caminho ou Fechado.  Caso o status esteja A Caminho, ele só poderá passar para o status Fechado.  O status Cancelado será atribuído apenas pelo cliente. | | | |
| Prioridade | Complexidade | Situação | Versão |
| Média | Alta | Aprovado | 4.0 |

|  |  |  |  |
| --- | --- | --- | --- |
| RF0023 – O sistema deve interagir com dispositivos móveis. | | | |
| O sistema deve interagir com dispositivos móveis. A cada troca de status, o sistema deve interagir com um app. Avisando o usuário vinculado ao pedido, sobre a troca ocorrida. | | | |
| Prioridade | Complexidade | Situação | Versão |
| Média | Alta | Em aprovação | 4.0 |

|  |  |  |  |
| --- | --- | --- | --- |
| RF0024 – O sistema deve disponibilizar uma interface para interação com os pedidos, por parte do funcionário. | | | |
| O sistema deve disponibilizar para funcionários com as devidas permissões, uma interface de interação e extração de relatórios sobre pedidos e clientes.  Esta interface inicialmente deve possibilitar:  1 - Alteração de status de pedidos.  2 - Relatórios sobre pratos mais vendidos.  3 - Informações sobre cancelamento de pedidos.  3.1 - Quem cancelou.  3.2 - Motivo do cancelamento.  3.3 - Prato cancelado.  4 - Mediação de mensagens e avaliações de clientes.  5 - Acesso rápido para inserir/remover/alterar promoções. | | | |
| Prioridade | Complexidade | Situação | Versão |
| Alta | Alta | Aprovado | 4.0 |

|  |  |  |  |
| --- | --- | --- | --- |
| RF0025 – O sistema deve possibilitar a realização de pedidos, por parte do cliente. | | | |
| O sistema deve disponibilizar para clientes devidamente cadastrados e logados no sistema, uma interface realização de pedidos.  Para tal o cliente deve realizar sua escolha em um ambiente acessível ao público em geral.  Tal escolha poderá ser de um ou mais partos disponibilizados para tal.  Depois de efetivada a escolha dos pratos o cliente deve efetuar um pedido, clicando em um botão bem destacado na aplicação, com valor FAZER PEDIDO.  Caso o cliente não esteja logado no sistema, ele deve ser direcionado para a tela de login de clientes.  Caso o sistema não encontre seus dados cadastrais, o sistema deve direcionar o futuro cliente para uma tela de cadastro.  Depois de realizado o cadastro com sucesso, o cliente deve ser redirecionado para a tela de pagamento de pedidos. Nesta tela devem ser listados todos os pratos escolhidos anteriormente pelo cliente, assim como, seus respectivos preços.  O cliente deve informar uma forma de pagamento. E efetivar o pedido clicando em FINALIZAR PEDIDO.  O cliente deve ser informado sobre a situação do pedido logo após o término do processo de pagamento.  Caso o pagamento tenha sido efetuado com sucesso, o sistema redireciona o cliente para a tela Meus Pedidos. Nesta tela ele verifica o status da cada pedido efetuado, além de fazer sua avaliação e deixar seu comentário.  Caso o pagamento não tenha sido autorizado, o sistema deve oferecer outras sugestões de pagamento para o cliente. | | | |
| Prioridade | Complexidade | Situação | Versão |
| Alta | Alta | Aprovado | 4.0 |

**6. REQUISITOS NÃO FUNCIONAIS**

|  |  |  |
| --- | --- | --- |
| N° | Descrição do requisito | Prioridade |
| RNF001 | O sistema deve estar acessível via dispositivos móveis. | Alta |
| RNF002 | O sistema deve ter um tempo de resposta inferior a 5 segundos. | Alta |
| RNF003 | O sistema deve obedecer às regras do e-commerce brasileiro. | Alta |
| RNF004 | O sistema deve manter alta disponibilidade - Inválida | Alta |
| RNF005 | O sistema deve respeitar a privacidade dos dados contidos em sua base. | Alta |
| RNF006 | O sistema deve ser desenvolvido sobre o SGDB MySql. | Média |
| RNF007 | O sistema deve ser desenvolvido utilizando a linguagem PHP. | Média |
| RNF008 | Dever ser utilizado o padrão de arquitetura MVC. | Alta |
| RNF009 | Deve ser utilizado o framework de ORM Doctrine | Média |
| RNF010 | O sistema deve ser desenvolvido utilizando Orientação a objetos. | Alta |

**6.1 Especificação**

|  |  |  |  |
| --- | --- | --- | --- |
| RNF0001 – O sistema deve estar acessível via dispositivos móveis. | | | |
| O sistema deve possuir um layout responsivo, tornando a experiência do usuário agradável e dinâmica também em dispositivos móveis. O layout não deve ser adaptativo e sim responsivo. As funções de pedido e cadastro devem obrigatoriamente devem estar acessíveis. | | | |
| Prioridade | Complexidade | Situação | Versão |
| Alta | Alta | Aprovado | 3.0 |
| RNF0002 – O sistema deve ter um tempo de resposta inferior a 5 segundos. | | | |
| Cada requisição feita pole usuário ou funcionário, não devem ultrapassar 5 segundos.  Salvo processos que dependam de terceiros. Ex.: Pagamento on line. | | | |
| Prioridade | Complexidade | Situação | Versão |
| Alta | Alta | Aprovado | 3.0 |

|  |  |  |  |
| --- | --- | --- | --- |
| RNF0003 – O sistema deve obedecer às regras do e-commerce brasileiro. | | | |
| Informações em:  <http://www.planalto.gov.br/ccivil_03/_ato2011-2014/2013/decreto/d7962.htm>  http://www.sebrae.com.br/sites/PortalSebrae/bis/Aspectos-legais-do-e%E2%80%93commerce | | | |
| Prioridade | Complexidade | Situação | Versão |
| Alta | Alta | Aprovado | 3.0 |

|  |  |  |  |
| --- | --- | --- | --- |
| RNF0004 – O sistema deve manter alta disponibilidade. -Inválido | | | |
| Este requisito não pode ser garantido pelo sistema. | | | |
| Prioridade | Complexidade | Situação | Versão |
| Alta | Alta | Aprovado | 3.0 |

|  |  |  |  |
| --- | --- | --- | --- |
| RNF0005 – O sistema deve respeitar a privacidade dos dados contidos em sua base. | | | |
| O sistema deve implementar métodos de criptografia e contenção de vazamento de informação.  O sistema deve utilizar um sistema de criptografia assimétrico denominado Blowfish. | | | |
| Prioridade | Complexidade | Situação | Versão |
| Alta | Alta | Aprovado | 3.0 |

|  |  |  |  |
| --- | --- | --- | --- |
| RNF0006 – O sistema deve ser desenvolvido sobre o SGDB MySql. | | | |
| O banco de dados MySQL é um sistema de gerenciamento (SGBD) que usa a Linguagem de Consulta Estruturada (SQL – Structured Query Language) como interface. A flexibilidade, versatilidade e a facilidade encontrada em seu uso fizeram do MySQL um dos bancos de dados preferidos. Usado por algo em torno de dez milhões de instalações no mundo todo. O MySQL é usado pela NASA; pela Friendster; pelo Bradesco; pela Nokia; pela HP; pela Sony; pela Lufthansa e por diversas outras corporações das mais variadas pátrias e pessoas físicas de várias cidadanias. | | | |
| Prioridade | Complexidade | Situação | Versão |
| Média | Baixa | Aprovado | 3.0 |

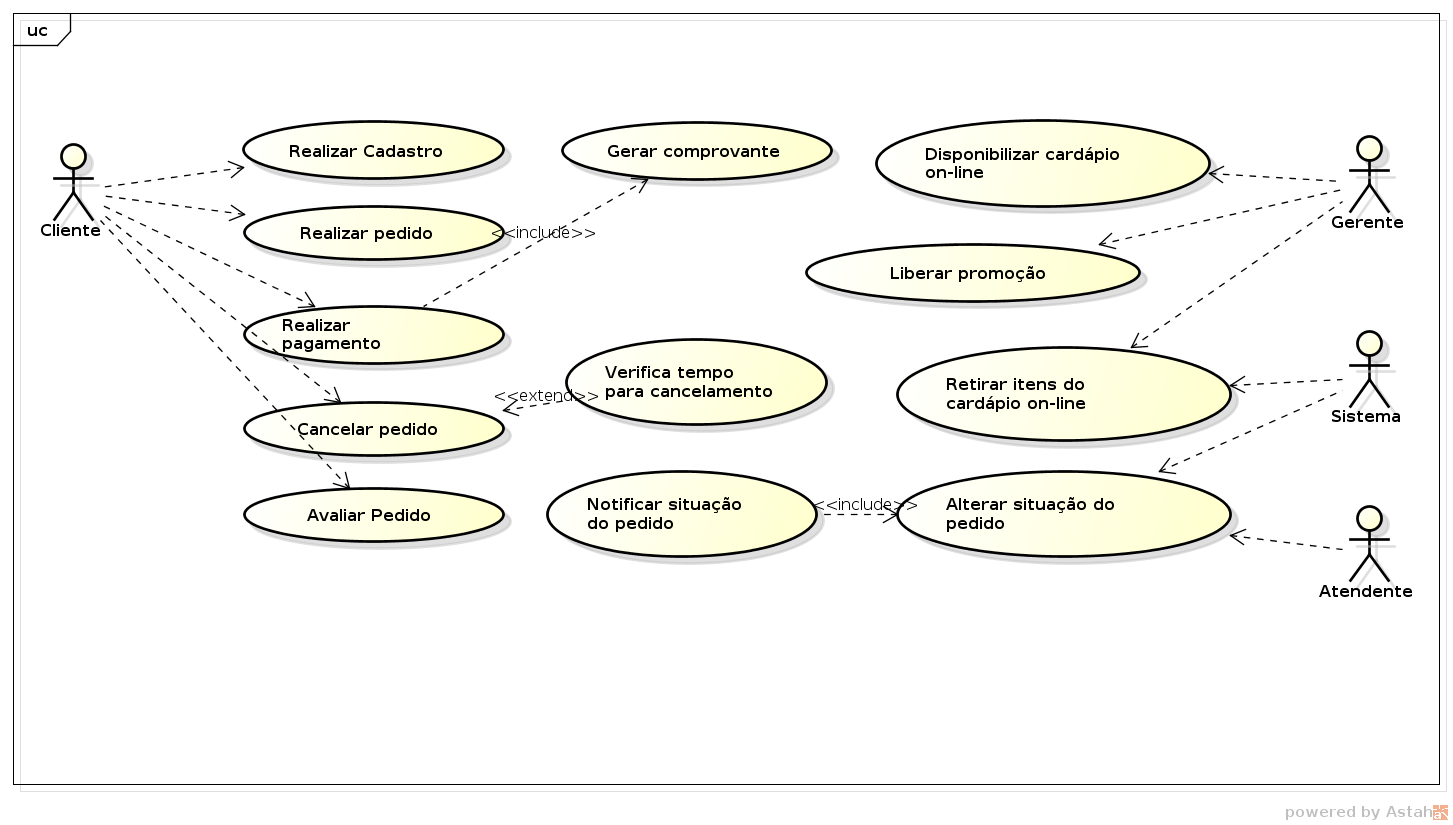
|  |  |  |  |
| --- | --- | --- | --- |
| RNF0007 – O sistema deve ser desenvolvido utilizando a linguagem PHP. | | | |
| **PHP** (um [acrônimo](https://pt.wikipedia.org/wiki/Acrônimo_recursivo) recursivo para *"****P****HP:* ***H****ypertext* ***P****reprocessor"*, originalmente ***P****ersonal* ***H****ome* ***P****age*) é uma interpretada livre, usada originalmente apenas para o desenvolvimento de aplicações presentes e atuantes no [lado do servidor](https://pt.wikipedia.org/wiki/Server-side), capazes de gerar conteúdo dinâmico na [World Wide Web](https://pt.wikipedia.org/wiki/World_Wide_Web). Figura entre as primeiras linguagens passíveis de inserção em documentos [HTML](https://pt.wikipedia.org/wiki/HTML), dispensando em muitos casos o uso de arquivos externos para eventuais processamentos de dados. O [código](https://pt.wikipedia.org/wiki/Código_fonte) é interpretado no lado do servidor pelo módulo PHP, que também gera a página web a ser visualizada no [lado do cliente](https://pt.wikipedia.org/wiki/Client_Side). | | | |
| Prioridade | Complexidade | Situação | Versão |
| Média | Baixa | Aprovado | 3.0 |

|  |  |  |  |
| --- | --- | --- | --- |
| RNF0008 – Dever ser utilizado o padrão de arquitetura MVC. | | | |
| **Model-view-controller** (**MVC**), em português **modelo-visão-controlador**, é um padrão de arquitetura de software ([*design pattern*](https://pt.wikipedia.org/wiki/Design_pattern)) que separa a representação da informação da interação do usuário com ele. O *modelo* (*model*) consiste nos dados da aplicação, regras de negócios, lógica e funções. Uma *visão* (*view*) pode ser qualquer saída de representação dos dados, como uma tabela ou um diagrama. É possível ter várias visões do mesmo dado, como um gráfico de barras para gerenciamento e uma visão tabular para contadores. O *controlador* (*controller*) faz a mediação da entrada, convertendo-a em comandos para o modelo ou visão. As ideias centrais por trás do MVC são a [reusabilidade de código](https://pt.wikipedia.org/wiki/Reutilização_de_código) e [separação de conceitos](https://pt.wikipedia.org/wiki/Separação_de_conceitos). | | | |
| Prioridade | Complexidade | Situação | Versão |
| Média | Baixa | Aprovado | 3.0 |

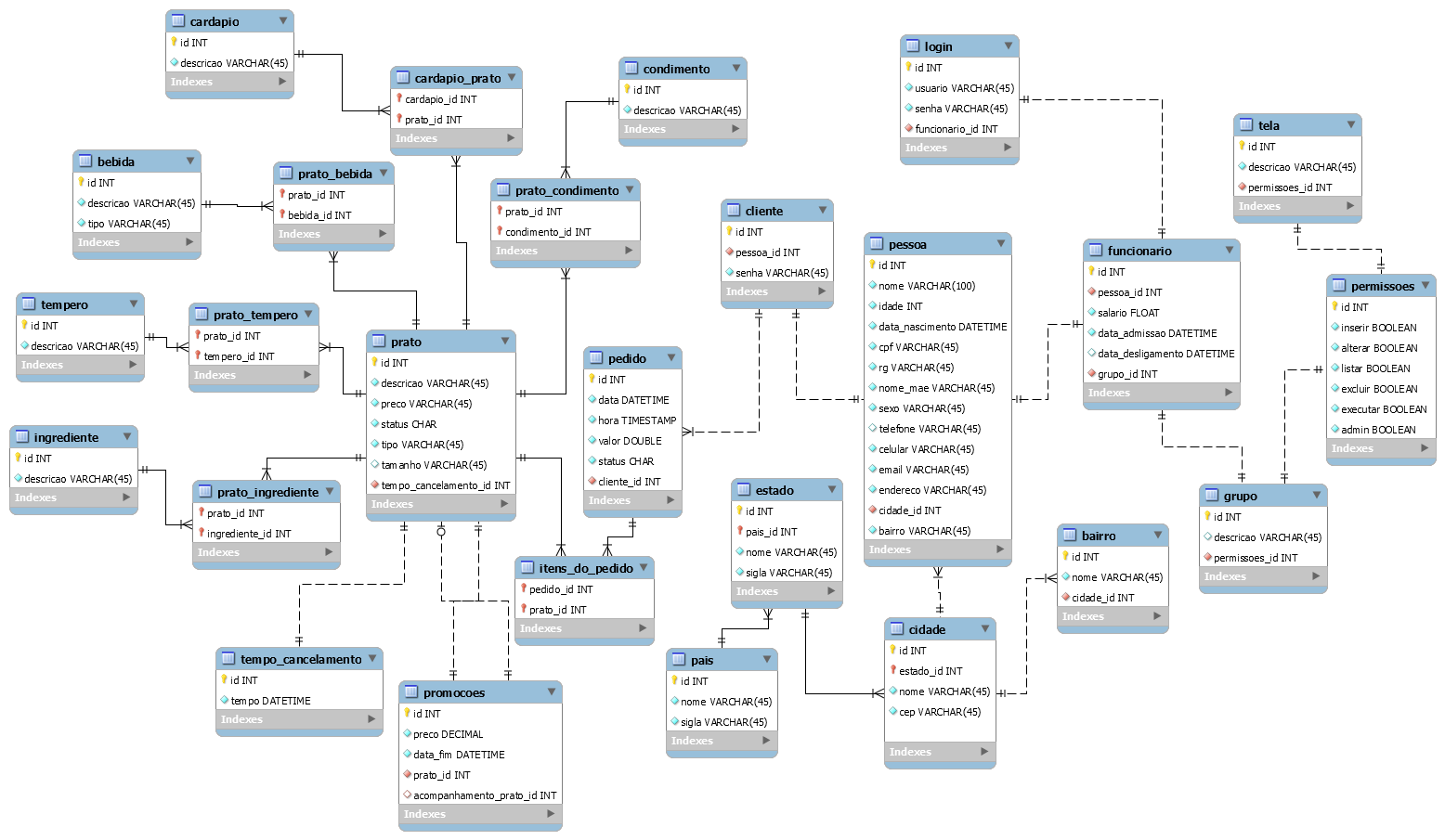
|  |  |  |  |
| --- | --- | --- | --- |
| RNF0009 – Deve ser utilizado o framework de ORM Doctrine | | | |
| O Doctrine é um mapeador objeto-relacional (ORM) que provê um meio transparente de persistir objetos PHP em bases de dados relacionais. A implementação de Doctrine está construída sobre uma camada de abstração de banco de dados.  ORM (Object-relational mapping) é uma camada intermediária entre um banco de dados relacional e objetos. De um lado, temos as tabelas e seus relacionamentos, de outro os objetos de uma linguagem OO. | | | |
| Prioridade | Complexidade | Situação | Versão |
| Média | Baixa | Aprovado | 3.0 |

|  |  |  |  |
| --- | --- | --- | --- |
| RNF0010 – O sistema deve ser desenvolvido utilizando Orientação a objetos. | | | |
| A **orientação a objetos** é um modelo de [análise](https://pt.wikipedia.org/wiki/Análise_de_sistemas), [projeto](https://pt.wikipedia.org/wiki/Padrão_de_projeto_de_software) e [programação](https://pt.wikipedia.org/wiki/Linguagem_de_programação) de sistemas de [*software*](https://pt.wikipedia.org/wiki/Software) baseado na composição e interação entre diversas unidades de software chamadas de objetos. | | | |
| Prioridade | Complexidade | Situação | Versão |
| Alta | Média | Aprovado | 3.0 |

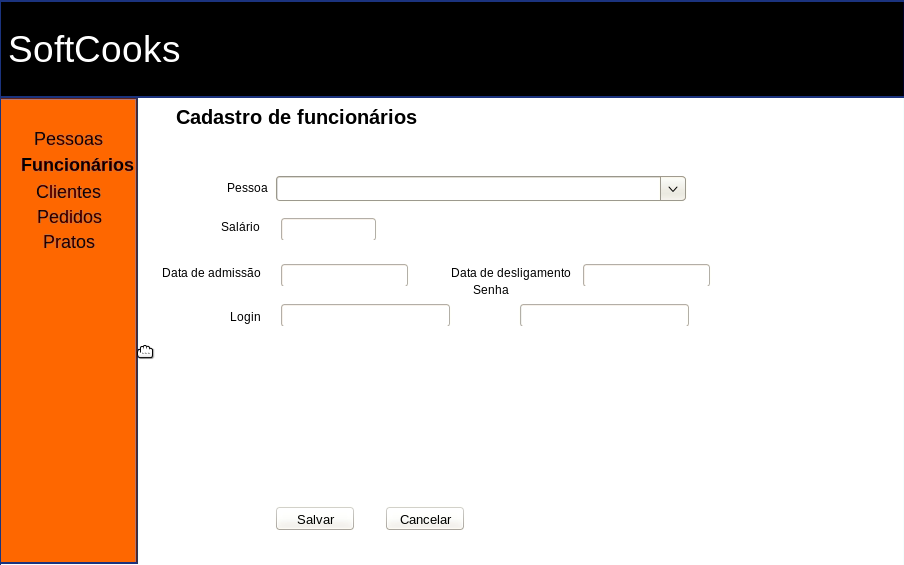
**7. DIAGRAMA DE CASOS DE USO**



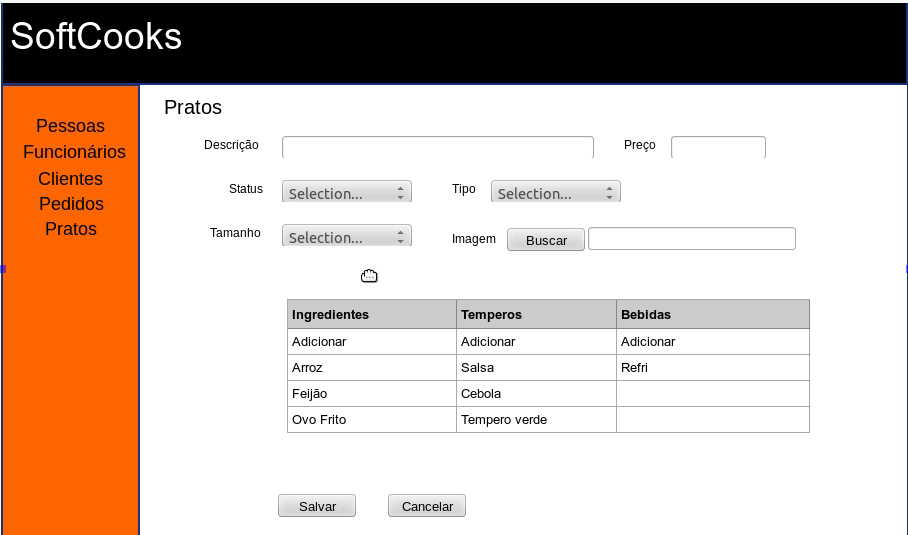
1. **MODELO DO BANCO DE DADOS**



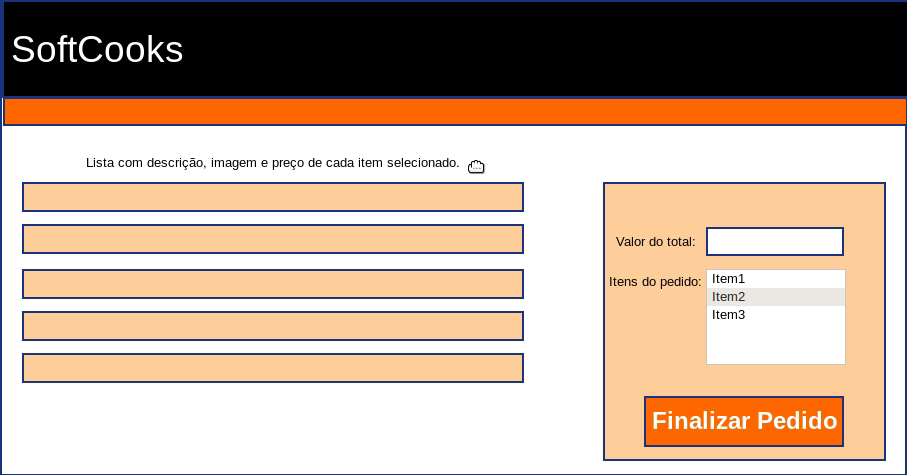
**9. TELAS DO SISTEMA**



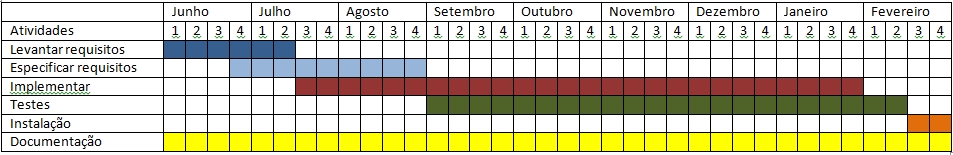






****

**10. CRONOGRAMA**



**11. CUSTO E ORÇAMENTO**

|  |  |  |
| --- | --- | --- |
| **Valores por atribuição** | | |
| **Atribuição** | **Valores por hora** | **Semanas trabalhadas** |
| Programador | 12,00 | 26 |
| Analista | 17,00 | 15 |
| Testes/Instalação/Documentação | 10,00 | 60 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Custo semanal de cada atribuição** | | | | |
| **Atribuição** | **Horas por semana** | **Semanas trabalhadas** | **Custo** | |
| Programador | 44 | 26 | 13.728 | |
| Analista | 44 | 15 | 11.220 | |
| Testes/Instalação/Documentação | 44 | 60 | 26.400 | |
| Custo total do projeto | | | | 51.348 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Orçamento** | | | |
| **Custo do Projeto** | **Percentual de lucro** | **Lucro** | **Valor a receber** |
| 51.348 | 30% | 15.404 | 66.752 |

O projeto será desenvolvido inicialmente, por três pessoas, cumprindo os papéis testador, analista e desenvolvedor. Tais pessoas devem possuir o mínimo das atribuições citadas acima, pois no decorrer do projeto, alternâncias entre funções irão ocorrer conforme necessidades.

A estratégia para venda do produto, será a de vender por um preço muito acessível, mas atrelada a uma mensalidade. O cliente receberá atualizações com melhorias no software, e caso encontre problemas poderá relatá-los para posterior ajuste. O cliente também é instigado a solicitar modificações no software, estas serão avaliadas e realizado um orçamento para a implementação.