

**CENTRO UNIVERSITÁRIO**

**UNIRUY – CAMPUS PARALELA**

**TÓPICOS EM DESENVOLVIMENTO E SISTEMAS**

**APLICAÇÃO PROTÓTIPO**

**PROF. HELENO CARDOSO**

**Aluno: Pedro Henrique da Silva Silveira**

**Matricula: 191051253**

**SALVADOR/BA 2020.1**

Pedro Henrique da Silva Silveira

APLICAÇÃO PROTÓTIPO

Aplicação protótipo, apresentado ao programa de graduação em Análise e Desenvolvimento de Sistemas da Faculdade Ruy Barbosa, como parte dos requisitos necessários à obtenção do título de bacharelado em Analista de Sistemas.

Prof: Heleno Augusto

Salvador – BA

2020.1

Sumário

[3.1 Requisitos funcionais 5](#_Toc38359940)

[3.2 Requisitos não funcionais 5](#_Toc38359941)

[3.2.1 Usabilidade 5](#_Toc38359942)

[3.2.2 Confiabilidade 6](#_Toc38359943)

[3.2.3 Desempenho 6](#_Toc38359944)

[3.2.4 Suporte 7](#_Toc38359945)

[3.2.5 Implementação 7](#_Toc38359946)

[3.2.6 Interface 7](#_Toc38359947)

[3.2.7 Jurídico 7](#_Toc38359948)

[4. Levantamento de Requisitos 7](#_Toc38359949)

[4.1 Modelos de sistema 7](#_Toc38359950)

[4.2 Diagrama de Casos de Uso 12](#_Toc38359951)

[Objetos de entidade 13](#_Toc38359952)

[Objetos de limites 13](#_Toc38359953)

[Objetos de controle 13](#_Toc38359954)

[4.2.3.1 Casos de uso – Verificar Limite 14](#_Toc38359955)

[Objetos de entidade 14](#_Toc38359956)

[Objetos de limite: 14](#_Toc38359957)

[Objeto de controle: 14](#_Toc38359958)

[4.2.3.2 Caso de Uso da Ordem de Processo 15](#_Toc38359959)

[Objetos de entidade: 15](#_Toc38359960)

[Objetos de limite: 16](#_Toc38359961)

[Objetos de controle: 16](#_Toc38359962)

[4.2.3.3 Caso de Uso Adicionar Receita 17](#_Toc38359963)

[Objetos de entidade: 17](#_Toc38359964)

[Objetos de limite: 17](#_Toc38359965)

[Objetos de controle: 18](#_Toc38359966)

[4.2.3.4 Caso de uso Atualização de receita 18](#_Toc38359967)

[Objetos de entidade: 19](#_Toc38359968)

[Objetos de limite: 19](#_Toc38359969)

[Objetos de controle: 19](#_Toc38359970)

[4.2.3.5 Caso de uso Remover receita 20](#_Toc38359971)

[Objetos de entidade: 20](#_Toc38359972)

[Objetos de limite: 20](#_Toc38359973)

[Objetos de controle: 20](#_Toc38359974)

[4.2.3.6 Caso de Uso Adicionar fornecedor 21](#_Toc38359975)

[Objetos de entidade: 21](#_Toc38359976)

[Objetos de limite: 21](#_Toc38359977)

[Objetos de controle: 22](#_Toc38359978)

[4.2.3.6 Caso de uso Remover fornecedor 22](#_Toc38359979)

[Objetos de entidade: 22](#_Toc38359980)

[Objetos de limite: 23](#_Toc38359981)

[Objetos de controle: 23](#_Toc38359982)

[4.3 Modelo de Objeto 23](#_Toc38359983)

[4.3.1 Generalização de classe e diagramas de hierarquia 23](#_Toc38359984)

[4.4 Diagrama de associação 24](#_Toc38359985)

[4.5 Interface do Usuário 24](#_Toc38359986)

1. Introdução

Um estudo de caso em uma empresa de restauração e gerenciamento de restaurantes corporativos no local citou questões relacionadas a uma lista de requisitos de recursos básicos que deve ser mantida manualmente pela equipe. Para acompanhar seus níveis de estoque, eles precisam calcular uma lista dos mantimentos utilizados durante um período de tempo, calcular e analisar os requisitos para o futuro e fazer o próximo pedido aos fornecedores, se necessário. Esse processo demanda muito tempo e esforço humano e também é propenso a erros humanos. Isso representa um problema de uma situação que uma equipe pode enfrentar, assim como muitos outros restaurantes, enfrenta. Leva muito tempo para acompanhar manualmente as vendas e fazer pedidos corretos aos fornecedores, desperdiçando trabalho útil em trabalhos triviais. Um produto que ajudaria a resolver os problemas mencionados acima seria proveitoso para clientes assim e empresas similares, pois ajudaria a converter o tempo improdutivo em algo mais útil, removendo as complicações e esforços desnecessários e propensos a erros.

2. Proposta do sistema

2.1 Visão geral

Propus desenvolver um software que monitore o estoque nos e o atualize de acordo com as vendas diárias. Cada item deve alimentar e está vinculado aos respectivos recursos (ou ingredientes) e, à medida que cada produto é vendido, os ingredientes utilizados na fabricação desse produto também são utilizados. Essas mudanças no inventário são acompanhadas através da utilização de um banco de dados local.

Propomos acompanhar todos os ingredientes, vinculando-os dinamicamente ao produto e, como resultado, criar uma relação dependente com esse produto. Em um período de tempo específico, se o inventário estiver abaixo do nível limite, serão gerados formulários de pedidos para fornecedores específicos para reabastecer os itens necessários para a próxima semana. O projeto também faz previsões inteligentes sobre o inventário necessário para a semana seguinte, com base na previsão em possíveis ocasiões ou eventos que podem influenciar vendas próximas futuras. No final da semana, o software leva em consideração todos os níveis de limite, previsões e outros fatores para gerar um formulário de pedido, que depois de verificado pelo gerente, é enviado aos fornecedores.

# 3.1 Requisitos funcionais

O Sistema visa fornecer uma interface eficiente ao usuário para gerenciar o inventário, além de fornecer ao usuário opções variadas de gerenciamento por meio de várias funções disponíveis. Os níveis de ingredientes são monitorados continuamente com base no uso ou definido pelo usuário, são verificados quanto aos níveis de limite no inventário e, portanto, o usuário é alertado sobre os baixos níveis de certos ingredientes. O design é tal que o usuário não precisa atualizar manualmente o inventário todas as vezes, o sistema faz isso pelo usuário.

O Sistema calcula e prevê a quantidade de uso para dias específicos definidos, predefinidos pelo usuário (admin), também alerta o usuário sobre uma ação iminente para solicitar ingredientes antes do dia específico definido pelo usuário. Portanto, o usuário nunca precisa se preocupar em calcular manualmente o uso estimado dos ingredientes, o sistema faz para o usuário.

A interface simples do sistema possui funções como adicionar uma quantidade, remover ou atualizar a quantidade. Também se estende a funções como adicionar um fornecedor para um ingrediente, remover o fornecedor, verificar níveis de limite, processar pedidos, alterar pedidos processados ​​etc.

# 3.2 Requisitos não funcionais

# 3.2.1 Usabilidade

O sistema deve ser fácil de usar, tanto pelos gerentes quanto pelos chefs, para que eles não precisem ler uma extensa quantidade de manuais.

O sistema deve ser rapidamente acessível por gerentes e usuários comuns.

O sistema deve ser intuitivo e simples na maneira como exibe todos os dados e relacionamentos relevantes.

Os menus do sistema devem ser facilmente navegáveis pelos usuários com botões fáceis de entender.

# 3.2.2 Confiabilidade

O sistema deve fornecer status de inventário preciso ao usuário continuamente. Quaisquer imprecisões são tratadas pela confirmação regular dos níveis reais com os níveis exibidos no sistema.

O sistema deve adicionar com êxito qualquer receita, ingredientes, fornecedores ou ocasiões especiais fornecidas pelo usuário e fornecer estimativas e status do inventário relevantes às entidades recém-atualizadas.

O sistema deve fornecer um login ativado por senha ao usuário para evitar que qualquer entidade estrangeira altere os dados no sistema.

O sistema deve fornecer atualizações ao usuário após a conclusão dos processos solicitados e, se os processos solicitados falharem, deve fornecer ao usuário o motivo da falha.

O sistema não deve atualizar os dados em nenhum banco de dados para processos com falha.

# 3.2.3 Desempenho

O sistema não deve ficar para trás, porque os funcionários que o usam não têm tempo de inatividade para aguardar a conclusão de uma ação.

O sistema deve concluir a atualização dos bancos de dados, adicionando receita, ingrediente, fornecedor e ocasiões com sucesso toda vez que o usuário solicitar esse processo.

Todas as funções do sistema devem estar disponíveis para o usuário toda vez que o sistema for ligado.

Os cálculos realizados pelo sistema devem cumprir as normas definidas pelo usuário e não devem variar, a menos que sejam explicitamente alterados pelo usuário.

# 3.2.4 Suporte

O software foi projetado para funcionar mesmo em sistemas com a configuração mínima.

O sistema é adaptável mesmo que plug-ins ou módulos adicionais sejam adicionados posteriormente.

Os dados podem ser exportados para o gerente para tornar o sistema mais portátil.

# 3.2.5 Implementação

A Interface do usuário do sistema foi criada em javascript com o framework React.

O banco de dados é implementado no NodeJS e Postgres.

# 3.2.6 Interface

O sistema deve oferecer uma maneira fácil e simples de visualizar o inventário atual.

O sistema deve poder exibir os relacionamentos entre fornecedores, ingredientes e receitas de maneira intuitiva.

# 3.2.7 Jurídico

O software deve ser licenciado individualmente para empresas menores, bem como através de um contrato de licença múltipla para empresas maiores.

O cliente deve concordar com o EULA antes de usar nosso software.

# 4. Levantamento de Requisitos

# 4.1 Modelos de sistema

Cenário 1 – Adicionando receita de uma pizza

|  |  |
| --- | --- |
| Nome do cenário | addReceita |
| Primeira ação | *manager*: Manager |
| Fluxo de eventos | 1. Há um novo item de menu 'Pizza' oferecido pelo restaurante e o item não está no banco de dados de receitas. Assim, o gerente inicia a adição de uma nova receita ao banco de dados pressionando o botão Adicionar receita.  2. O sistema exibe o formulário addReceita na frente do gerente.  3. O gerente preenche os detalhes da receita da pizza, como o pão usado, as coberturas que serão usadas e a quantidade de cada um dos recursos utilizados também é inserida pelo gerente. O gerente também adiciona qualquer novo ingrediente que ainda não esteja na lista de ingredientes, invocando a função de adicionar ingrediente no sistema.  4. O gerente envia as alterações e o sistema adiciona a nova receita de Pizza ao banco de dados de receitas e quaisquer novos ingredientes ao banco de dados de ingredientes.  5. O gerente recebe um reconhecimento da conclusão do processo. |

Cenário 2 – Deletando receita criada

|  |  |
| --- | --- |
| Nome do cenário | removerReceita |
| Primeira ação | *manager*: Manager |
| Fluxo de eventos | 1. O gerente está passando pelas vendas e percebe que as vendas de pizza estão em baixa e decide remover Pizza como um item de menu.  2. O gerente ativa a função remove receita no sistema.  3. O sistema apresenta a lista de receitas que estão atualmente no banco de dados. O gerente seleciona a receita da Pizza para remover e envia a alteração. O sistema confirma se o gerente tem certeza de remover a receita da pizza dos itens de menu.  4. O gerente confirma a alteração e o sistema reconhece a remoção da receita do banco de dados. |

Cenário 3 – Atualizar receita de pizza

|  |  |
| --- | --- |
| Nome do cenário | updateReceita |
| Primeira ação | *manager*: Manager |
| Fluxo de eventos | 1. O gerente ativa a função updateReceita no sistema.  2. O sistema agora apresenta uma lista de receitas a serem selecionadas pelo gerente.  3. O gerente agora seleciona Pizza na lista de receitas e escolhe atualizá-lo.  4. O gerente agora recebe um formulário que apresenta as propriedades atuais da receita. O gerente agora adiciona / remove ingredientes, seu uso, etc. e depois envia as alterações.  5. O sistema agora atualiza o banco de dados da receita com as novas propriedades e reconhece as alterações. |

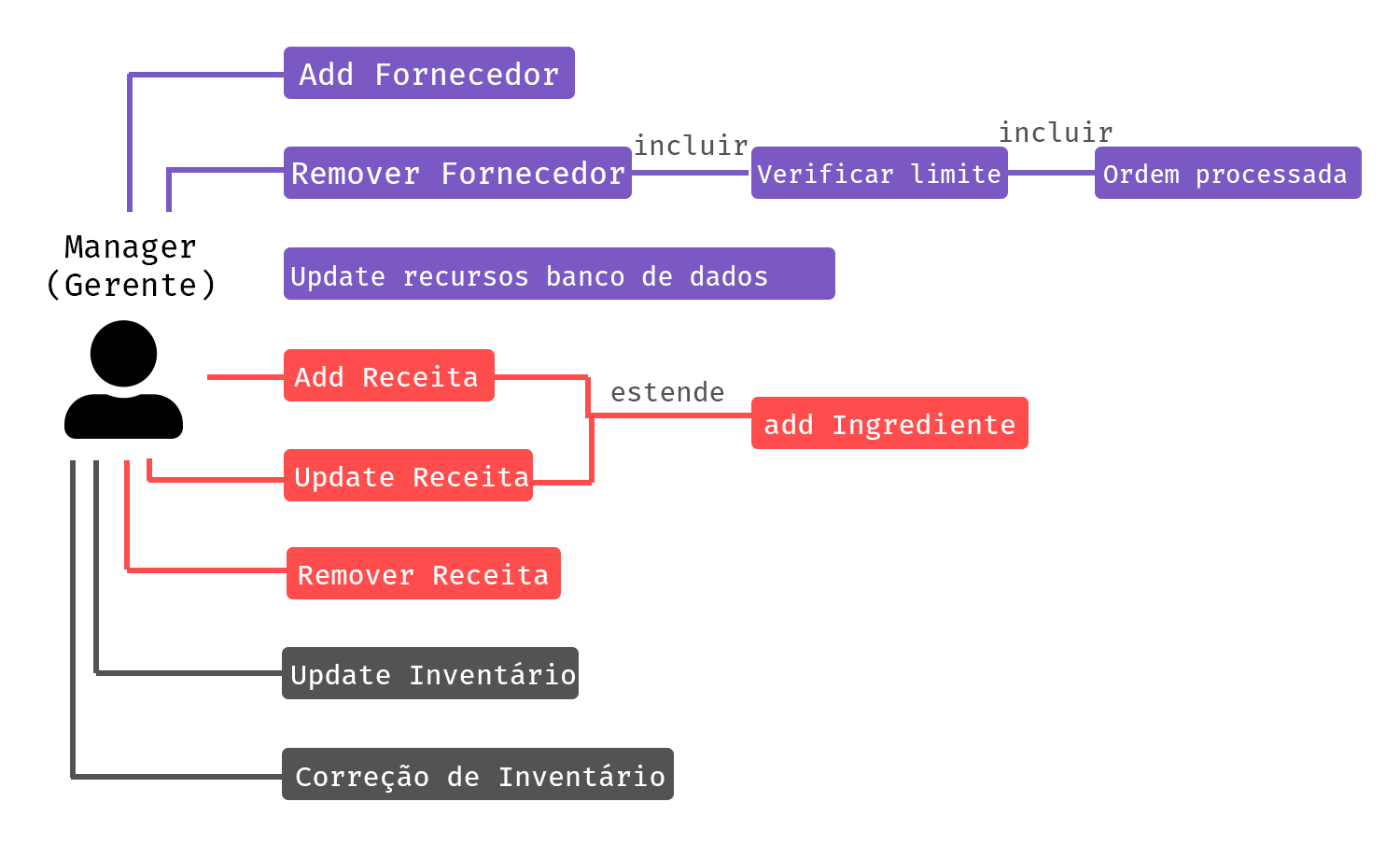
Cenário 4 – Atualizar banco de dados

|  |  |
| --- | --- |
| Nome do cenário | Atualize o banco de dados de recursos no final do dia. |
| Primeira ação | *manager*: Manager |
| Fluxo de eventos | 1. O gerente ativa a função de atualização do banco de dados de recursos.  2. O gerente recebe um formulário de comida vendido e o gerente entra nos itens de menu que foram vendidos e também nas quantidades correspondentes que foram vendidas no dia.  3.O gerente envia o formulário e o sistema deduz as quantidades de ingredientes usadas na confecção desses itens de menu. O sistema agora verifica os níveis de limite e, se os encontrar acima dos limites, informa ao gerente que todos os ingredientes estão acima do limite e que há recursos suficientes disponíveis para uso. |

Cenário 5 – Correção de inventário

|  |  |
| --- | --- |
| Nome do cenário | Correção de inventário |
| Primeira ação | *manager*: Manager |
| Fluxo de eventos | 1. O gerente considera que há alguma diferença entre os níveis de estoque mostrados pelo sistema e os níveis reais de estoque. O gerente inicia a função de verificação de estoque no sistema.  2. O gerente recebe um formulário com uma lista de todos os ingredientes e seus níveis correspondentes.  3. O gerente agora realiza uma pesquisa do estoque real e os combina com os níveis mostrados pelo sistema. Se forem encontradas discrepâncias, o gerente atualiza a nova quantidade próxima à caixa apresentada ao lado do nome do ingrediente no formulário.  4. Após atualizar quaisquer discrepâncias de inventário, o gerente envia as alterações e o inventário é atualizado de acordo.  5. Os níveis de limite também são atualizados tendo em mente os erros incorridos nos cálculos anteriores e o gerente recebe um aviso de conclusão do processo. |

# 4.2 Diagrama de Casos de Uso



# 4.2.3 Casos de uso

|  |  |
| --- | --- |
| Nome do cenário | Atualizar recursos de banco de dados |
| Primeira ação | *manager*: Manager |
| Fluxo de eventos | 1. O gerente ativa a função de atualização do banco de dados de recursos.  2. O sistema apresenta um formulário ao gerente. O formulário solicita detalhes dos itens vendidos durante o curso da semana e a quantidade correspondente dos alimentos vendidos.  3. O gerente insere os dados dos alimentos vendidos para a semana e a quantidade que foi vendida e pressiona o botão Salvar.  4. O Sistema lê os dados de alimentos vendidos e depois lê, no banco de dados de ingredientes, os ingredientes que foram usados na fabricação dos itens de alimentos vendidos.  5. O sistema agora calcula a quantidade de recursos utilizados e deduz a quantidade de ingredientes usados no banco de dados de recursos.  6. O sistema agora chama a caixa de usuário VerificarLimite. |

# Objetos de entidade

1. Gerente: O usuário que inicia e gerencia todo o caso de uso. O gerente insere os dados de alimentos vendidos no formulário que é apresentado a ele e também controla quais ações devem ser tomadas depois que todos os cálculos terminarem.

2. Recursos: Estes são os ingredientes disponíveis para uso na confecção de receitas.

# Objetos de limites

1. Botão Atualizar banco de dados de recursos: este é o botão que o gerente pressiona para iniciar os processos no caso de uso.

2. Formulário de alimento vendido: Este é o formulário que o gerente preenche para fornecer detalhes sobre os alimentos que foram vendidos no decorrer da semana.

3. Formulário de Pedido de Processo: Nesse formulário, o gerente insere as quantidades correspondentes a cada ingrediente que o gerente deseja atualizar.

# Objetos de controle

Este objeto é criado quando o gerente ativou a função AtualizarBancoDeDados. O Objeto é responsável por criar o formulário VendaAlimento e solicitar níveis de recursos do banco de dados, comparando novos valores com os valores limite e também enviando os novos valores de volta ao banco de dados. O objeto também envia agradecimentos ao gerente assim que os processos solicitados são concluídos.

# 4.2.3.1 Casos de uso – Verificar Limite

|  |  |
| --- | --- |
| Nome do cenário | Verificar Limite |
| Primeira ação | *manager*: Manager |
| Fluxo de eventos | 1. O sistema agora compara os níveis atuais dos recursos com os níveis limite dos recursos. Agora ele lista todos os ingredientes que estão abaixo do nível do limite, juntamente com o uso previsto dos ingredientes e o apresenta ao gerente.  2. O gerente agora pode enviar pedidos pressionando o botão Processar pedido. Esta ação chama o processo ProcessarPedido.  3. Se o Gerente pressionar cancelar, nenhum pedido será processado. |

# Objetos de entidade

1. Recursos: Estes são os ingredientes disponíveis para uso na confecção de receitas. 2. Níveis de limite: esses são os níveis de ingrediente abaixo dos quais um alerta é gerado para o gerente gerar novos pedidos.

# Objetos de limite:

ProcessarPedido: este botão chama o processamento de pedidos para os ingredientes que estão abaixo dos níveis limite.

# Objeto de controle:

Objeto de controle de atualização: este objeto é criado quando o gerente ativou a função AtualizarBancoDeDados. O Objeto é responsável por criar o formulário de AlimentosVendidos e solicitar níveis de recursos do banco de dados, comparar novos valores com os valores limite e também enviar os novos valores de volta ao banco de dados. O objeto também envia agradecimentos ao gerente assim que os processos solicitados são concluídos.

# 4.2.3.2 Caso de Uso da Ordem de Processo

|  |  |
| --- | --- |
| Nome do cenário | **Ordem de Processo** |
| Primeira ação | *manager*: Manager |
| Fluxo de eventos | 1. O sistema reúne uma lista de fornecedores dos quais os ingredientes correspondentes são pedidos. Combina os ingredientes correspondentes com os fornecedores dos quais os ingredientes estão disponíveis.  2. O sistema agora cria um pedido e, em seguida, apresenta o formulário de resumo do pedido ao gerente. O formulário possui três opções: uma para aprovar o pedido, a outra para revisar o pedido e outra para cancelar o pedido.  3. O gerente, neste momento, pode aprovar o pedido gerado. Pressionando o botão aprovar pedido no formulário de resumo do pedido. Isso enviará os pedidos gerados aos fornecedores. O gerente recebe a confirmação da recepção do pedido e o processo termina.  4. O gerente também pode optar por revisar o pedido e inserir as quantidades a serem solicitadas manualmente para cada ingrediente correspondente. O gerente pode fazer isso escolhendo o botão Revisar pedido. Um formulário atualizado de resumo do pedido é apresentado ao gerente e o fluxo retorna ao ponto 2.  5. O gerente também pode optar por cancelar o pedido pressionando o botão Cancelar ordem. Nesse caso, o pedido gerado é cancelado e nenhum pedido é enviado. |

# Objetos de entidade:

1. Gerente: O gerente ativa o caso de uso para adicionar uma nova receita ao banco de dados. Ele é responsável por adicionar ingredientes à receita.

2. Ingrediente: Os ingredientes que estão abaixo do nível limite são adquiridos como uma lista e os mesmos precisam ser aprovados pelo gerente.

3. Fornecedor: o pedido gerado é enviado ao fornecedor como uma solicitação de consignação.

# Objetos de limite:

Ordem de processo: Este botão é usado quando o formulário de ordem gerado é aprovado pelo gerente e o seguinte é enviado ao fornecedor.

Cancelar pedido: o botão Cancelar pedido é usado se os pedidos não forem gerados corretamente. Se o gerente não estiver satisfeito com o formulário de pedido, o cancelamento do pedido será usado.

Revisar pedido: o pedido de revisão permite que sejam feitas alterações no formulário de pedido gerado atualmente. O gerente edita as quantidades adequadamente.

# Objetos de controle:

Controle de pedido de processo: este objeto de controle gerencia todo o processo de produção de um pedido, revisão do pedido e cancelamento do pedido atual. Ele contém uma lista de ingredientes necessários do fornecedor. Somente quando o gerente aprovar a lista final, o fornecedor receberá o formulário solicitado.

# 4.2.3.3 Caso de Uso Adicionar Receita

|  |  |
| --- | --- |
| Nome do cenário | **Adicionar Receita** |
| Primeira ação | *manager*: Manager |
| Fluxo de eventos | 1. O gerente ativa a função "Criar nova receita" em seu terminal  2. O sistema responde apresentando um formulário ao gerente. O formulário solicita detalhes associados à receita.  3. O gerente preenche o formulário inserindo os ingredientes a serem usados na nova receita. Ele também adiciona qualquer novo ingrediente usado na receita executando o caso de uso Adicionar Ingrediente que estende esse caso. O gerente também insere a quantidade de ingrediente a ser usado em uma única ordem da receita. Após o preenchimento do formulário, o Gerente envia o formulário ao Sistema.  4. O sistema reconhece que a nova receita foi criada. Também o adiciona ao banco de dados de receitas e a qualquer novo ingrediente no banco de dados de ingredientes. |

# Objetos de entidade:

1. Gerente: O gerente ativa o caso de uso para adicionar uma nova receita ao banco de dados. Ele é responsável por adicionar ingredientes à receita.

2. Receita: Esta é uma nova receita que está sendo adicionada ao banco de dados pelo gerente. A receita é definida pelos itens que são usados ​​na fabricação da receita.

3. Ingredientes: Os ingredientes são os itens dos quais a receita é feita.

# Objetos de limite:

Botão Adicionar receita: este botão inicia o caso de uso para adicionar receita, o que ajuda a adicionar uma nova receita ao banco de dados.

Formulário Adicionar receita: Este formulário solicita ao gerente detalhes da receita em termos dos ingredientes necessários e da quantidade necessária por ingrediente.

Botão Adicionar ingrediente: Este botão inicia o caso de uso Adicionar Ingrediente. Isso é iniciado entre o caso de uso Adicionar Receita, se algum ingrediente não estiver disponível para o gerente ao adicionar a receita e o gerente desejar adicionar esse ingrediente específico à receita.

# Objetos de controle:

Adicionar controle de receita: esse objeto de controle gerencia todo o processo de adição da nova receita ao banco de dados. Ele gerencia o processo de adição da nova receita ao banco de dados, o novo ingrediente no banco de dados de ingredientes e também o reconhecimento da conclusão do processo ao gerente.

# 4.2.3.4 Caso de uso Atualização de receita

|  |  |
| --- | --- |
| Nome do cenário | **Atualização de receita** |
| Primeira ação | *manager*: Manager |
| Fluxo de eventos | 1. O gerente ativa “Atualizar receita” no sistema.  2. O sistema responde trazendo a lista de receitas.  3. O gerente seleciona uma receita para alterar.  4. O sistema agora mostra um formulário AtualizarReceita com a lista de ingredientes na receita e a quantidade correspondente.  5. O gerente altera a receita adicionando / removendo ingredientes ou atualizando a quantidade de ingredientes usados na receita. O gerente também pode adicionar novos ingredientes que não estão disponíveis no momento, invocando a base de usuários AdicionarIngrediente.  6. O gerente conclui a atualização pressionando o botão Concluir no sistema.  7. O sistema confirma que a alteração foi feita e atualiza os bancos de dados. |

# Objetos de entidade:

Gerente: O gerente ativa o caso de uso para atualizar uma receita. O gerente é responsável por adicionar / remover ingredientes de e para a receita.

Receita: Esta é uma entidade que está sendo atualizada pelo gerente. A receita é definida pelos itens que são usados ​​na fabricação da receita.

Ingredientes: Os ingredientes são os itens que a receita é feita.

# Objetos de limite:

Botão Atualizar receita: Este botão inicia o caso de uso para adicionar receita, o que ajuda a adicionar uma nova receita ao banco de dados.

Atualizar formulário de receita: Este formulário solicita ao gerente que insira as alterações que serão feitas na receita.

Adicionar Ingrediente: Este botão inicia o caso de uso AdicionarIngrediente. Isso é iniciado entre o caso de adição de receita, se algum ingrediente não estiver disponível para o gerente durante a atualização da receita e o gerente desejar adicionar esse ingrediente específico à receita.

# Objetos de controle:

Atualizar controle de receita: esse objeto de controle gerencia todo o processo de atualização da receita. Ele gerencia o processo de adicionar / remover os ingredientes da receita. Ele também reconhece o status do processo para o gerente.

# 4.2.3.5 Caso de uso Remover receita

|  |  |
| --- | --- |
| Nome do cenário | **Remover receita** |
| Primeira ação | *manager*: Manager |
| Fluxo de eventos | 1. O gerente ativa a função “RemoverReceita” em seu terminal.  2. O sistema responde mostrando a lista atual de receitas salvas no sistema.  3. O gerente escolhe quais receitas remover e as remove selecionando um botão de exclusão na janela do terminal.  4. O sistema confirma com cada exclusão com o gerente se ele deseja excluir a receita.  5. O gerente confirma sua decisão com um sim / não.  6. O Sistema reconhece a decisão removendo a receita se respondeu com "sim" ou cancelando a exclusão se respondeu com "não". Em seguida, ele exibe uma confirmação da decisão, exibindo uma exclusão bem-sucedida ou uma solicitação cancelada. |

# Objetos de entidade:

Gerente: O gerente ativa o caso de uso para remover uma receita.

Receita: Esta é uma entidade que está sendo removida pelo gerente. A receita é definida pelos itens que são usados na fabricação da receita.

# Objetos de limite:

RemoverReceita: Este botão inicia o caso de uso para adicionar receita, o que ajuda a adicionar uma nova receita ao banco de dados.

Formulário remoção de receita: este formulário apresenta uma lista de receitas atualmente no banco de dados e solicita ao gerente que selecione qual receita deve ser excluída.

# Objetos de controle:

Remover controle de receita: esse objeto de controle gerencia todo o processo de atualização da receita. Ele gerencia o processo de adicionar / remover os ingredientes da receita. Ele também reconhece o status do processo para o gerente.

# 4.2.3.6 Caso de Uso Adicionar fornecedor

|  |  |
| --- | --- |
| Nome do cenário | **Adicionar fornecedor** |
| Primeira ação | *manager*: Manager |
| Fluxo de eventos | 1. O gerente ativa a função "Adicionar fornecedor" em seu terminal.  2. O sistema responde exibindo um formulário a ser preenchido pelo gerente para o fornecedor a ser criado.  3. O gerente preenche o formulário preenchendo as informações do fornecedor a ser criado e também os ingredientes que serão solicitados a esse fornecedor. Após todas as informações terem sido preenchidas, o gerente envia o formulário.  4. O sistema pega as informações do formulário e adiciona ao fornecedor o banco de dados de fornecedores. Em seguida, ele exibe um aviso ao gerente de que o fornecedor foi adicionado. |

# Objetos de entidade:

Gerente: Esta entidade inicia o caso de uso e adiciona detalhes do fornecedor a ser adicionado.

Fornecedor: Esta é a entidade que fornece certos ingredientes ao restaurante.

# Objetos de limite:

Botão Adicionar fornecedor: esse é o botão que o gerente pressiona para iniciar a adição de fornecedor ao banco de dados.

Adicionar formulário do fornecedor: este é o formulário que o gerente preenche para adicionar os detalhes do fornecedor ao banco de dados. Detalhes como o nome do fornecedor, os ingredientes disponíveis no fornecedor e o tempo de remessa exigido pelo fornecedor para os ingredientes solicitados são mencionados nos detalhes.

# Objetos de controle:

Adicionar controle de fornecedor: esse objeto de controle gerencia o processo de apresentação do formulário de adição de fornecedor ao gerente, lendo os dados inseridos e atualizando o banco de dados com os novos dados.

# 4.2.3.6 Caso de uso Remover fornecedor

|  |  |
| --- | --- |
| Nome do cenário | **Remover fornecedor** |
| Primeira ação | *manager*: Manager |
| Fluxo de eventos | 1. O gerente ativa a função "Remover fornecedor" em seu terminal  2. O sistema responde mostrando a lista atual de fornecedores salvos no sistema.  3. O gerente escolhe quais fornecedores deseja remover e os remove selecionando um botão de exclusão na janela do terminal.  4. O sistema confirma a cada exclusão do gerente se ele deseja remover o fornecedor.  5. O gerente confirma sua decisão com um sim / não.  6. O Sistema reconhece a decisão removendo o fornecedor se respondeu com "sim" ou cancelando a exclusão se respondeu com "não". Em seguida, ele exibe uma confirmação da decisão, exibindo uma exclusão bem-sucedida ou uma solicitação cancelada. |

# Objetos de entidade:

Gerente: Esta entidade inicia o caso de uso para remover o fornecedor da lista de fornecedores.

Fornecedor: Esta é a entidade que fornece certos ingredientes ao restaurante.

# Objetos de limite:

Botão Remover fornecedor: esse é o botão que o gerente pressiona para iniciar o caso de uso.

Remover formulário do fornecedor: este é o formulário que apresenta a lista de fornecedores que estão atualmente no banco de dados e solicita ao gerente que selecione o fornecedor a ser removido do banco de dados.

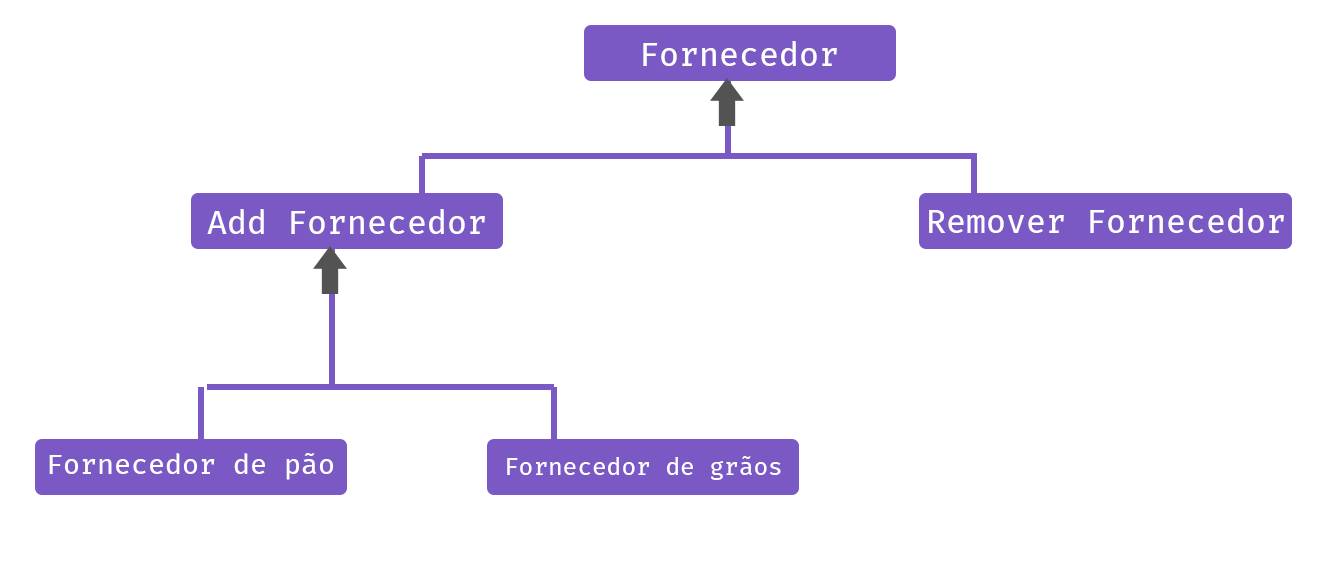
# Objetos de controle:

Remover controle de fornecedor: este é o objeto de controle que facilita o processo de remoção do fornecedor selecionado do banco de dados do fornecedor.

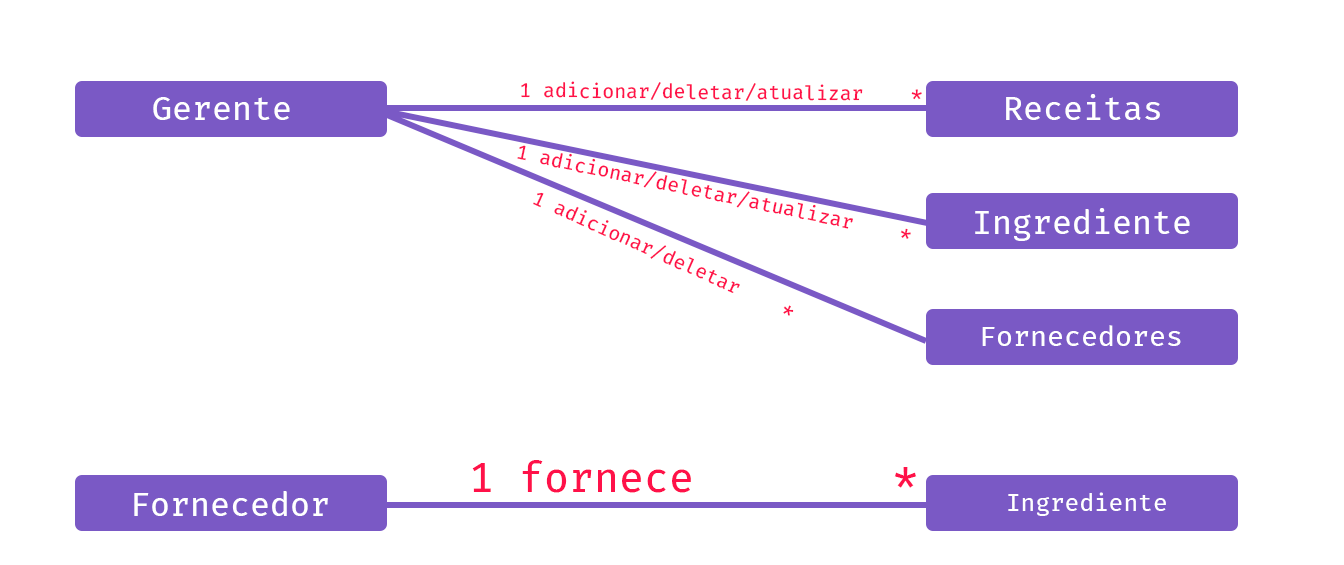
# 4.3 Modelo de Objeto

# 4.3.1 Generalização de classe e diagramas de hierarquia

Classe de fornecedor e diagrama de hierarquia.

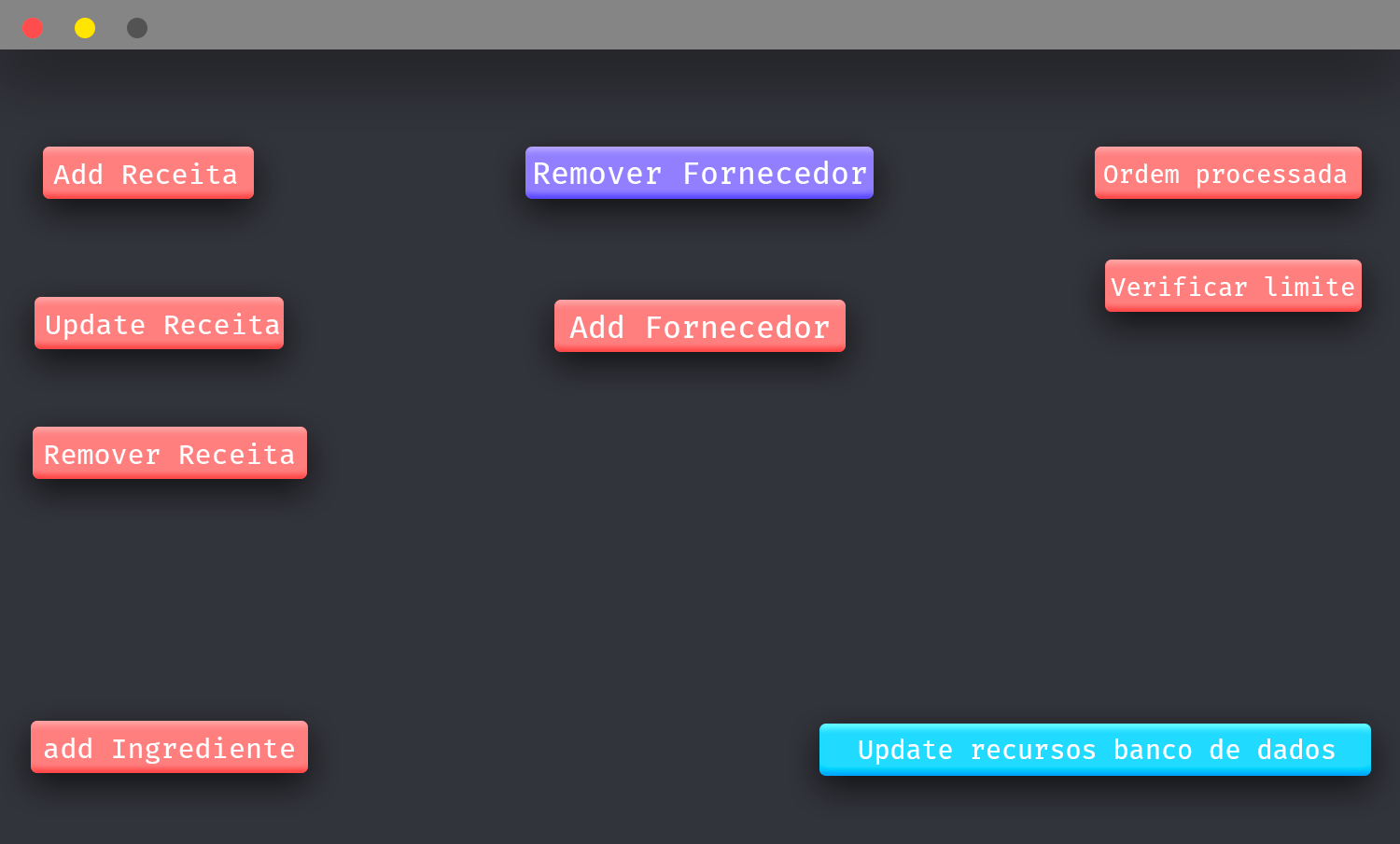


# 4.4 Diagrama de associação

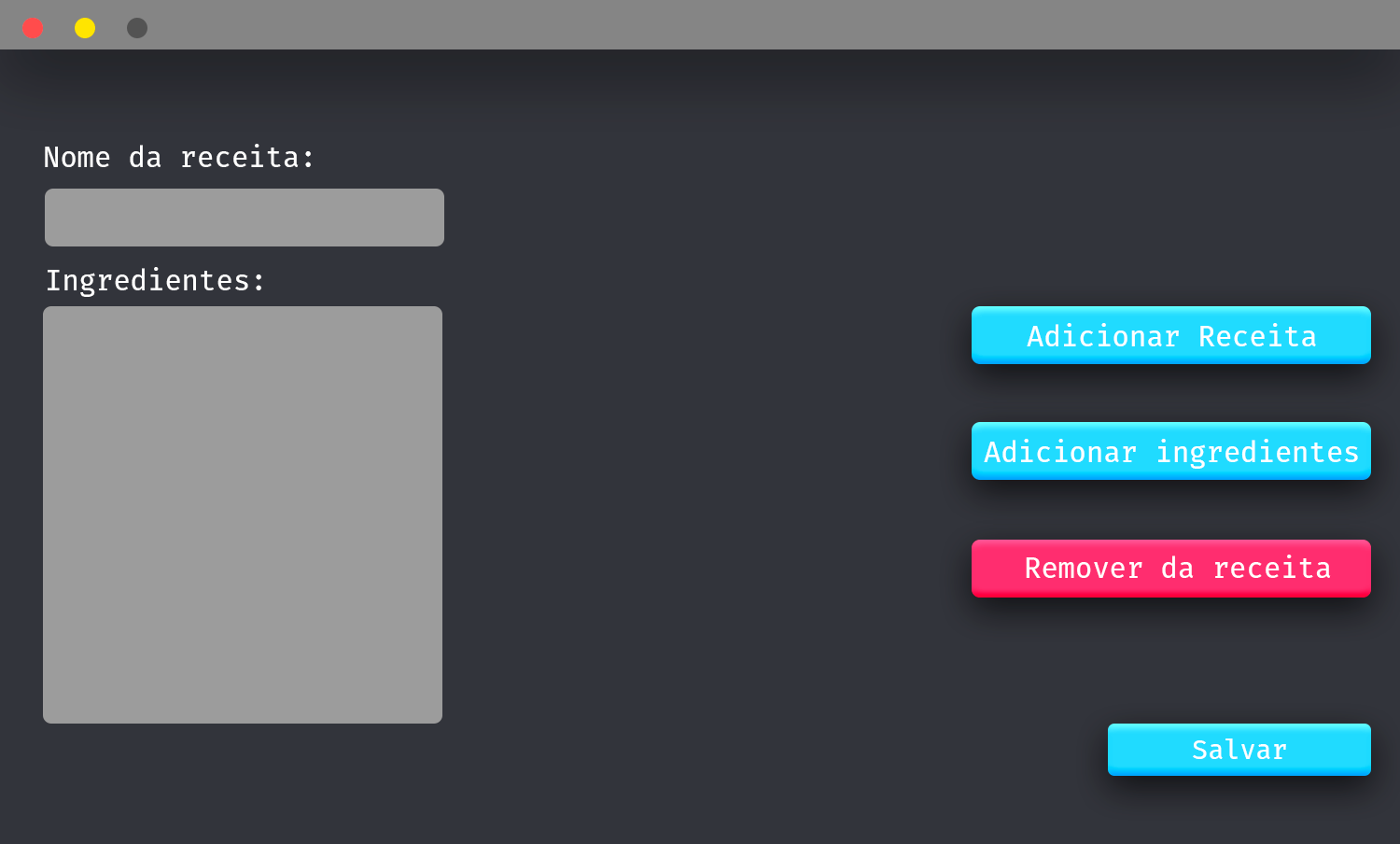


# 4.5 Interface do Usuário

Tela inicial



Tela de adicionar receita



Tela de adicionar fornecedor

