

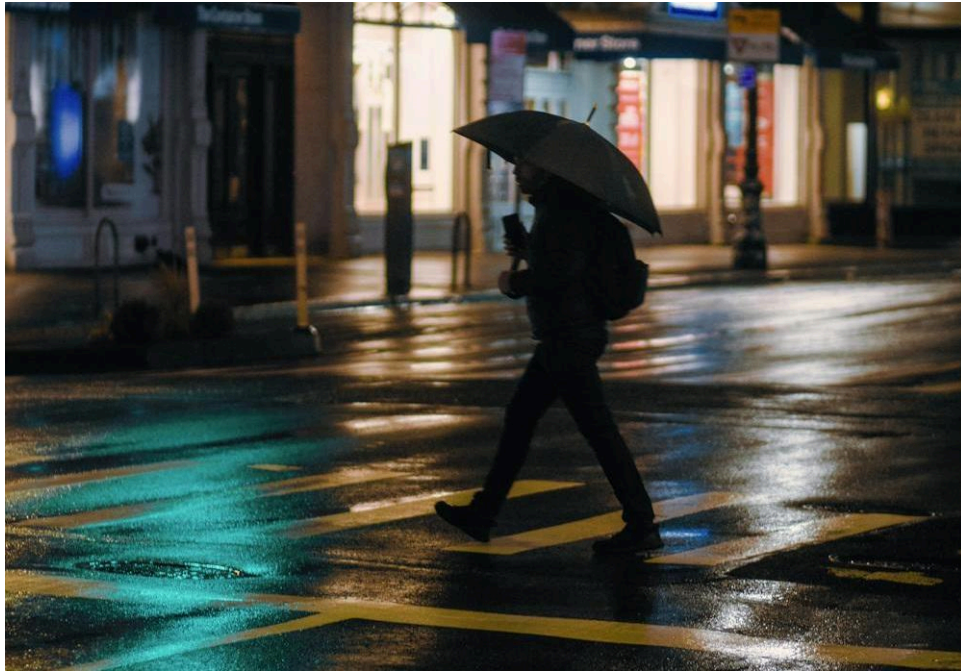
# Jornada AppSec

Criando API de postagem de status  
do pedido para delivery

Autor: Felipe Pinheiro

GitHub: <https://github.com/pinheiro-felipe/jornada-appsec-criando-api-de-postagem-de-status-do-pedido-para-delivery>

1º Dia: Domingo, 26 de maio de 2024  
Noite chuvosa e fazendo um pouco de frio.



Boa noite pessoal, meu nome é Felipe Pinheiro e hoje vou começar a nossa jornada appsec, criando api de postagem de status do pedido para delivery.

Eu sempre gostei de compartilhar conhecimento e trocar ideia sobre desenvolvimento de software seguro, mas isso sempre ocorreu no um a um. Há anos desejo trocar ideia com mais pessoas na internet, mas sempre deixei de lado ao priorizar outras tarefas do dia a dia.

E nesse momento às 23:43h eu decidi só começar. A proposta é ser um diário de appsec ao criar uma API de postagem de status do pedido para delivery e ajudar o máximo de pessoas possíveis. E quem sabe empilhar algum tijolinho de conhecimento que agregue valor de verdade na vida de você que lê.

Então, bora lá?!

Nessa jornada, eu vou passar a minha visão de como criar uma API segura, por isso posso incluir coisas que você não pensou ou deixar de fora coisas que você incluiria. E tudo bem isso acontecer.

Apesar de ser uma espécie de diário, não é engessado. Sempre que tivermos uma nova visão ou aprendermos algo novo, podemos voltar e alterar o que acharmos que deveríamos ter feito num determinado dia.

Vale uma observação: sempre escrevo no topo da página a data em que comecei o conteúdo, para dar a ideia de um diário. Em algum momento, você poderá notar intervalos maiores entre as datas, o que pode ter acontecido porque levei mais tempo para criar o conteúdo ou porque fiz uma pausa longa, seja por motivos pessoais ou profissionais

## **Começando Nossa Sessão de Brainstorming**

Vamos partir do ponto que já foi realizada uma pesquisa da nossa parte sobre o mercado de delivery, comportamento do consumidor, suas dores e aplicativos mais usados. Também levantamos diversas suposições e validamos.

Nosso objetivo neste momento não é a ideia em si, mas o processo de como criar uma API segura. Se você cria API segura de uma forma diferente, vamos trocar uma ideia e aprender mais ainda juntos.

Então partindo desse ponto estabelecido acima, vamos fazer o seguinte questionamento:

### **1- Qual problema ou oportunidade de negócio nós identificamos?**

No nosso caso a oportunidade fictícia identificada é: Clientes de delivery de alimentação desejam acompanhar o status do seu pedido, não só com mensagens de texto, mas com postagens que se assemelham a posts de mídias sociais e etapa por etapa, desde a preparação do pedido até o envio do mesmo.

### **2- Como é a jornada desse cliente hoje para acompanhar o status do seu pedido no delivery, desde a preparação até o envio?**

Assim que o cliente faz o pedido, o seu status muda para "Aguardando aprovação".

Se o pedido for cancelado, o seu status muda para "Pedido cancelado" e termina no mesmo momento a interação. Contudo, se o pedido for aceito, o seu status muda para "Preparando".

Assim que o pedido estiver pronto e sair para entrega, o seu status muda para "Saiu para entrega". Por fim, ao chegar ao destino, o seu status muda para "Entregue" ou "Não entregue".

### **3- Como deve ser a nova jornada do cliente, para acompanhar o status do seu pedido na nossa plataforma de delivery, desde a preparação até o envio?**

Na nova jornada, o cliente receberá notificações sobre o status do seu pedido em formato semelhante a posts de mídias sociais, com foto ou vídeo acompanhados de um texto informativo.

Alguns status serão padronizados para etapas comuns a todas as empresas cadastradas. No entanto, veremos adiante que, na fase de preparo, esses status poderão ser customizados por cada empresa de delivery.

A nova jornada funciona da seguinte forma: após o pedido ser realizado pelo cliente o seu status é de “**aguardando aprovação**”, ao ser aceito o pedido entra numa fila de espera chamada “**aguardando preparo**”. Isso é útil em situações de grande demanda no estabelecimento, pois informa ao cliente que o pedido foi aceito, mas ainda não está sendo preparado. Nesse período, o cliente pode cancelá-lo a qualquer momento, até que o preparo seja iniciado.

Vamos imaginar a seguinte situação: um estabelecimento vende hambúrgueres, pizzas e sucos. Cada produto possui um modo de preparo diferente e, por isso, deve ter status específicos no processo de preparo. Esses status podem ser agrupados de forma mais genérica ou mais detalhada. Por exemplo, diferentes tipos de hambúrgueres, como x-tudo e x-bacon, podem ser reunidos em um processo único chamado 'hambúrguer', representando o preparo de todos os hambúrgueres. Alternativamente, é possível adotar processos mais específicos, como 'hambúrguer x-tudo', para indicar apenas o preparo desse item em particular.

A vantagem de um processo mais genérico é exigir menos trabalho na criação dos arquivos de mídia que serão postados aos clientes em cada fase do preparo do pedido. Já um processo mais específico tem como benefício a possibilidade de usar arquivos de mídia que não sejam apenas ilustrativos, mas que representem de forma mais fiel o produto que está sendo preparado.

Um produto como o 'hambúrguer x-tudo' normalmente segue o mesmo processo de preparo de outro hambúrguer, como o 'hambúrguer x-salada'. No entanto, ao separá-los em processos distintos, as fases continuam as mesmas, mas é possível associar mídias diferentes a cada uma delas. Assim, no 'hambúrguer x-salada' serão postadas fotos do x-salada e, no 'hambúrguer x-tudo', fotos do x-tudo. No modelo anterior, havia apenas uma única mídia para os dois tipos de hambúrguer.



É importante ressaltar que cada postagem de status do pedido contará com uma foto ou vídeo acompanhado de uma descrição. Isso tem dois objetivos: entreter o cliente durante a espera e ao mesmo tempo ensinar sobre o próprio processo de preparo.

Para exemplificar, escolhemos a fase de preparo de uma pizza. Assim que o pedido de pizza (ou de qualquer outro alimento) é aceito, ele deve ser encaminhado para fila de espera chamada “**aguardando preparo**”. Posteriormente para ter seu preparo iniciado, basta adicionar o primeiro substatus correspondente à fase de preparo.

Criamos os seguintes substatus para o preparo de uma pizza. No entanto, cada estabelecimento deve cadastrar seus próprios substatus de acordo com suas particularidades.

Pizzaria e seus substatus na fase de preparo:

**Ingredientes preparados**

**Pizza recheada**

**Pizza assada**

**Pizza cortada**

**Pizza embalada**

Após o término de todos os substatus na fase de preparo, o responsável pela preparação do pedido pode enviá-lo para entrega e então seu status muda para “**Pronto e aguardando a saída**”.

Assim que o responsável pela entrega chega, ele escaneia o QR code do pedido e o retira. Seu status muda para “**Saiu para entrega**” dizendo que o pedido está a caminho, além de agradecer a preferência e desejar bom apetite.

Por fim, assim que chegar ao destino e for recebido, o entregador muda o status para “**Entregue**” ao informar o código que só o cliente sabe. Mas se ninguém receber o pedido ou por algum motivo a entrega se tornar impossível, o entregador muda o status para “**Não entregue**” adicionando um áudio informando o motivo da não entrega.

2º Dia: Segunda, 27 de maio de 2024

O dia amanheceu nublado e ainda fazendo um pouco de frio.



Bom dia pessoal, hoje nós vamos começar a destrinchar o problema ou oportunidade que identificamos. Vamos definir nosso objetivo na visão do negócio e o porquê da criação da API, além de quais funcionalidades serão necessárias para alcançá-lo.

Não se preocupe se faltar algo, pois quanto mais formos avançando, mais lacunas irão surgir e uma a uma elas serão preenchidas.

#### **4- Na visão de negócio: qual é o objetivo a ser alcançado com a criação desta API?**

Objetivo: Permitir que os clientes de delivery de alimentação, possam acompanhar o status do seu pedido, de forma semelhante a posts de mídias sociais, etapa por etapa até o seu envio.

#### **5- Qual é a funcionalidade start identificada no nosso objetivo?**

A funcionalidade que nós chamamos de **start**, é a funcionalidade que dá origem a esta API, é a razão dela existir. É a funcionalidade desejada por determinado usuário ou ator. Ela demonstra um único objetivo.

Para identificá-la, fazemos as seguintes perguntas:

1º- Quem é o usuário ou ator?

Dica: **ator**, é aquele que possui necessidades a serem atendidas em nosso contexto.

2º- O que ele "deseja" fazer?

Dica: o que ele deseja é a funcionalidade identificada em nosso objetivo.

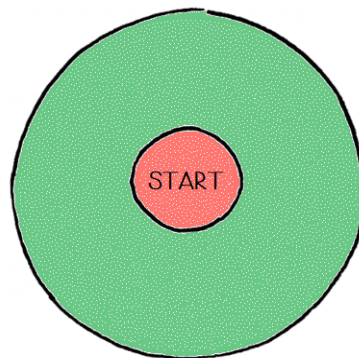
3º- Qual o formato de visualização desejado pelo ator?

4º- Em qual canal de comunicação o ator deseja visualizar as informações?

**Funcionalidade Start:** Representa "Quem" + "O que deseja" + "Forma de visualização" + "Onde", ou seja, de forma geral, quem deseja fazer o que?

**Cliente de delivery de alimentação "deseja" acompanhar o status do seu pedido em forma de posts de mídias sociais.**

Após a identificação da funcionalidade start acima, decidimos modelar não apenas ela de forma isolada. Decidimos supor que os clientes de delivery precisam de um novo aplicativo no mercado que provê uma API completa partindo dessa perspectiva central. Por isso vamos destrinchar tudo à sua volta, colocando a funcionalidade start como diferencial competitivo.



Observação: Não sei se tal ideia que usamos para começar nossa API, já existe em algum aplicativo, mas o foco nesse momento é o processo de criação da API de forma segura.

O que está na área do círculo em verde, dá suporte para que a funcionalidade start possa existir. Pois na API, existe a funcionalidade start que se deseja e as outras que precisam existir. É como se a funcionalidade start fosse o sol e as outras apenas orbitassem à sua volta.

Se a nossa API para delivery de alimentação não tiver essa funcionalidade start, não há razão para existir. Pois todas as outras funcionalidades que ficam na área verde, servem apenas para dar apoio a essa existência.

Então porque não criamos apenas essa funcionalidade start? Nesse momento estamos modelando nossa API, então optamos por exercitar a modelagem da aplicação de delivery o mais completa possível. Agora na implementação, seremos mais diretos e focaremos na funcionalidade que queremos.

## 6- Como destrinchamos a API a partir da funcionalidade start identificada?

Em nossa funcionalidade start, falta o "Como":

Para definirmos o nosso "como", fazemos a seguinte pergunta:

Como o [ator] vai fazer para + [funcionalidade start]?

Como o [cliente de delivery de alimentação] vai fazer para [acompanhar o status do seu pedido em forma de posts de mídias sociais]?

O como representa a jornada que o ator precisa percorrer para alcançar o objetivo desejado. Para responder a essa pergunta, devemos criar uma sequência de passos a serem realizados. Tudo com base no recorte de contexto estabelecido pela funcionalidade e ator.

Devemos transformar o "como" em comandos e eventos que devem ser executados em ordem cronológica.

**Comando:** São ações que representam as operações da API e são descritas usando o verbo no infinitivo. Assim que invocadas podem desencadear um ou mais eventos.

**Evento:** É construído da seguinte forma [sujeito] + [verbo] e sempre no passado. Considera-se na sua descrição que uma ação importante para o negócio já aconteceu. A descrição deve ser de fácil compreensão por pessoas não técnicas.

**Política:** Define que quando determinado evento acontecer qual comando será invocado. Para isso descreve-se uma regra de



negócio referente a decisão a ser tomada quando um ou mais eventos acontecerem. O evento ocorrido invocará um comando de forma automática ou manual. O ator responsável pela execução do evento manual, pode ser incluído. O que nos ajuda a entender quem tem a responsabilidade de tomar a decisão de invocar o comando.

Observação: Daqui para frente, desenho tudo o que for levantado de atores, comandos, eventos e políticas primeiro lá no miro ou em qualquer ferramenta semelhante. Lá consigo criar o fluxo de eventos etapa por etapa e ter uma visão do todo muito melhor. No miro conseguimos ter uma visão que abre a nossa mente, pois ele nos permite visualizar lacunas e refatorar o que for preciso de forma fácil.

Abaixo buscamos representar o "como do ator". Como ele vai fazer para alcançar o objetivo da funcionalidade start? Qual jornada ele precisa percorrer para acompanhar o status do seu pedido em forma de posts de mídias sociais? Quais são os passos?

Premissa: Vamos começar supondo que o pedido que o cliente deseja acompanhar o status, já foi realizado por ele.

Então quais comandos ele deve executar dentro do seu contexto em ordem cronológica? Quais eventos ocorrem?

O ator precisa:

Cliente

Comando [01]: Pesquisar pedidos

Evento: Pedidos pesquisados

Cliente

Comando [02]: Escolher pedido

Evento: Pedido escolhido

Cliente

Comando [03]: Pesquisar postagens de status do pedido

Evento: Postagens de status do pedido pesquisadas

Confira o desenho completo no miro:

[https://miro.com/app/board/uXjVK-93bzg=/?share\\_link\\_id=272758138846](https://miro.com/app/board/uXjVK-93bzg=/?share_link_id=272758138846)

Repare que antes de pesquisarmos por postagens de status do pedido, precisamos realizar uma pequena sequência de comandos, Pesquisar

**pedidos** e **Escolher pedido** Eles servem de pré-requisitos para obtermos as postagens de status do pedido que desejamos.

Vamos imaginar a seguinte situação:

Seu irmão, Lucas, encomenda uma pizza por um aplicativo de delivery do bairro chamado "Tô Com Fome". Após uns 15 minutos, do banho ele grita para você dar uma olhada no celular e ver como anda a pizza. Você que nunca viu esse aplicativo, naturalmente vai em busca de padrões. Então ao abrir o aplicativo você busca por pedidos que seu irmão tenha realizado e consequentemente encontra uma lista deles. Ao encontrar essa lista, você navega até o pedido mais recente. Ao encontrá-lo, você o seleciona para visualizar o status da pizza.

Com essa pequena história, procurei demonstrar que existem passos que realizamos todos os dias até chegar no objetivo desejado. E conosco é a mesma coisa, devemos nos colocar no lugar do nosso ator e fazer o caminho inverso.

Começa-se de trás para frente e busca-se chegar até a primeira ação que ele deveria ter realizado no contexto estabelecido.

Antes de pesquisarmos pelas postagens de status do pedido que é o objetivo da funcionalidade start, chegamos aos comandos e eventos, pensando da seguinte forma:

O que deveríamos ter feito antes de pesquisarmos as postagens de status do pedido, a nível de funcionalidade?

Organizando nosso pensamento de trás para frente, ficou da seguinte forma:

Queremos **pesquisar postagens de status do pedido**

**Cliente**

**Comando [03]:** Pesquisar postagens de status do pedido

**Evento:** Postagens de status do pedido pesquisadas

Mas, antes precisamos **escolher pedido**

Cliente

Comando [02]: Escolher pedido

Evento: Pedido escolhido

E antes ainda, precisamos pesquisar pedidos

Cliente

Comando [01]: Pesquisar pedidos

Evento: Pedidos pesquisados

Aqui chegamos ao limite da nossa jornada para **acompanhar postagem de status do pedido**. O que acontece antes do comando de **pesquisar pedidos**, corresponde a outra jornada que serve de pré-requisito para a nossa jornada acima.

Confira o desenho completo no miro:

[https://miro.com/app/board/uXjVK-93bzg=/?share\\_link\\_id=272758138846](https://miro.com/app/board/uXjVK-93bzg=/?share_link_id=272758138846)

O que fizemos acima, nos ajudou a identificar a jornada a ser percorrida pelo ator e qual o seu limite. Também nos ajudou a saber onde ele ou alguém deveria ter feito algo.

**Jornada:** É uma sequência de passos rumo a um objetivo que tenha um fim mensurável.

## 7- Como preenchemos as lacunas existentes em nossa funcionalidade start?

Existem lacunas para que os comandos e eventos identificados na funcionalidade start possam ser executados. Não sabemos nada do que acontece antes, até chegar o momento do primeiro comando de **pesquisar pedidos**, ser executado.

A partir de agora vamos pensar: O que acontece antes como pré-requisito para que exista um pedido que possa ser pesquisado e ter sua postagem de status acompanhada?

Vamos levantar todos os **comandos** ou **funcionalidades** que servem de pré-requisito para a funcionalidade start ser viabilizada. Para isso vamos replicar os passos que foram identificados dentro da nossa jornada de **acompanhar postagem de status do pedido**, dividir em etapas e preencher as lacunas com os pré-requisitos necessários.

Preenchendo as lacunas com pré-requisitos necessários:

Etapa 1: Cliente "precisa" pesquisar pedidos

Cliente

Comando [01]: Pesquisar pedidos

Evento: Pedidos pesquisados

---

Identificando pré-requisitos:

O que precisa acontecer antes para que os pedidos possam ser pesquisados? O cliente ou alguém precisa fazer o quê?

Cliente "precisa" realizar pedido

Como é a jornada de realização do pedido, por parte do cliente?

Quais comandos ou funcionalidades devem ser executados dentro do seu contexto em ordem cronológica para realização do pedido?

1º- Plataforma de delivery precisa carregar catálogo de produtos sugeridos com base num ranking de popularidade.

Plataforma de delivery (Execução automática)  
"precisa" carregar catálogo de produtos sugeridos

2º- Cliente precisa adicionar produto ao carrinho e poder continuar navegando para escolher outros

Cliente "precisa" adicionar produto ao carrinho

3º- O cliente precisa visualizar o carrinho com seus produtos adicionados, taxas e valores que serão cobrados.

Cliente "precisa" visualizar carrinho

Ao visualizar carrinho, como dependência:

Cliente "precisa poder" gerenciar produtos do carrinho

4º- O cliente precisa fechar o carrinho após estar certo dos produtos que adicionou ao carrinho.

Cliente "precisa" fechar carrinho

5º- O cliente precisa informar os dados que possibilitarão a entrega e cobrança.

Cliente "precisa" realizar checkout

O que precisa acontecer antes para que o cliente possa realizar o pedido? O cliente ou alguém precisa fazer o quê?

Possibilitar o funcionamento da plataforma de delivery, permitindo a execução de funcionalidades base.

---

Quais comandos ou funcionalidades devem ser executados dentro do seu contexto?

1º- Precisa existir empresa cadastrada na plataforma de delivery para oferecer serviço.

Empresa (Admin) "precisa" cadastrar empresa

Ao cadastrar empresa, como dependência:

Empresa (Admin) "precisa poder" gerenciar empresa

2º- Empresa precisa poder cadastrar funcionário para gerenciar perfil de acesso, atribuindo responsáveis as fases do pedido na plataforma de delivery.

Empresa (Admin) "precisa" cadastrar funcionário

Ao cadastrar funcionário, como dependência:

Empresa (Admin) "precisa poder" gerenciar funcionário



3º- A empresa precisa cadastrar produtos na plataforma de delivery para venda.

Empresa (Admin) "precisa" cadastrar produto

Ao cadastrar produto, como dependência:

Empresa (Admin) "precisa poder" gerenciar produto

4º- Empresa precisa poder gerenciar qualquer cliente que já tenha feito algum pedido nela. Isso para poder suspender ou cancelar a conta deste por algum motivo. Seja por fraude ou descumprimento de alguma regra da empresa. E ainda visualizar aqueles que mantêm a conexão com a empresa ativa para receber postagens do status do seu pedido.

Empresa (Admin) "precisa poder" gerenciar carteira de clientes

5º- O cliente precisa se cadastrar na plataforma de delivery para realizar o pedido.

Cliente "precisa" cadastrar cliente

Ao cadastrar cliente, como dependência:

Cliente "precisa poder" gerenciar cliente

6º- Cliente precisa poder gerenciar qualquer empresa que ele tenha realizado algum pedido. Isso para habilitar se deseja receber ou não as postagens de status do pedido. Postagens essas que começam assim que o pedido é realizado e prosseguem se o cliente não informar nenhum impedimento através do gerenciamento de conexão na sua carteira de empresas.

Cliente "precisa poder" gerenciar carteira de empresas

Etapa 2: Cliente "precisa" escolher pedido

Cliente

Comando [02]: Escolher pedido

Evento: Pedido escolhido

---

Identificando pré-requisitos:

O que precisa acontecer antes para que o pedido possa ser escolhido para visualização das postagens de status? O cliente ou alguém precisa fazer o quê?

Quais comandos ou funcionalidades devem ser executados dentro do seu contexto em ordem cronológica?

1º- A empresa precisa escolher um pedido que esteja “aguardando aprovação”, verificar se está de acordo com a regras da empresa e dizer se aceita ou cancela. Essas regras podem ser diversas como: - Hoje estou sem entregador, então retirada somente na loja. Todas ficam a cargo da empresa de delivery.

Empresa (Atendente) "precisa" aceitar pedido

Etapa 3: **Cliente** "precisa" **pesquisar postagens de status do pedido**

**Cliente**

**Comando [03]:** Pesquisar postagens de status do pedido

**Evento:** Postagens de status do pedido pesquisadas

---

Identificando pré-requisitos:

O que precisa acontecer antes para que as postagens de status do pedido possam ser pesquisadas? O cliente ou alguém precisa fazer o quê?

**Empresa (Responsável pela fase em que o pedido se encontra)**  
"precisa" **adicionar status ao pedido**

Quais comandos ou funcionalidades devem ser executados dentro do seu contexto em ordem cronológica para adição de status ao pedido?

1º- Empresa precisa na fase de preparo que normalmente acontece na cozinha, adicionar o novo status do pedido a cada etapa da preparação que ele avança.

**Empresa (Responsável pela preparação do pedido)**  
"precisa" **adicionar status ao pedido**, na fase de preparo

O que precisa acontecer antes para que o **cliente** possa adicionar status ao pedido, na fase de preparo? O cliente ou alguém precisa fazer o quê?

Possibilitar a execução das funcionalidades abaixo.

---

Quais comandos ou funcionalidades devem ser executados?

1º- Empresa precisa cadastrar processo de preparo. Isso é particular de cada empresa

e produto. São esses processos de preparo que serão usados para orientar o envio de status do pedido que está sendo produzido.

Empresa (Admin) "precisa" cadastrar processo de preparo

Ao cadastrar processo de preparo, como dependência:

Empresa (Admin) "precisa poder" gerenciar processo de preparo

2º- A empresa precisa cadastrar status de preparo. Isso é particular de cada empresa e produto. São esses status que serão usados para notificar ao cliente o passo a passo do seu pedido.

Empresa (Admin) "precisa" cadastrar status de preparo

Ao cadastrar status de preparo, como dependência:

Empresa (Admin) "precisa poder" gerenciar status de preparo

2º- Empresa precisa na fase de retirada para entrega, adicionar o novo status do pedido que está pronto e foi retirado para entrega.

Empresa (Responsável pela entrega do pedido) "precisa" adicionar status ao pedido, na fase de retirada para entrega

3º- Empresa precisa na fase de rota de entrega, adicionar status do pedido que foi entregue ou não entregue.

Empresa (Responsável pela entrega do pedido) "precisa" adicionar status ao pedido, na fase de rota de entrega

4º- Empresa precisa poder habilitar nova tentativa de entrega ao pedido, que tenha no máximo uma tentativa mal sucedida. Ao habilitar a nova tentativa, o pedido tem o status "saiu para entrega" adicionado novamente.

Empresa (Responsável pela entrega do pedido)  
"precisa poder" Habilitar nova tentativa de entrega, na  
fase de rota de entrega



## Jornadas background

---

As jornadas background, são ações que não dependem mais da interação humana, pois algum evento que ocorreu já a invocou ou invocará quando acontecer.

1º- Plataforma de delivery precisa acompanhar o processamento do pagamento e adicionar pedido.

Plataforma de delivery (Execução automática) "precisa" adicionar pedido

2º- Plataforma de delivery precisa notificar de forma automática todos os envolvidos no pedido, ou seja, a empresa e o cliente.

Plataforma de delivery (Execução automática) "precisa" notificar partes interessadas

3º- Plataforma de delivery após qualquer mudança de status do pedido, precisa adicionar alguma postagem de mídia que está relacionada ao status atual do pedido e que foi cadastrada de antemão pela empresa na plataforma.

Plataforma de delivery (Execução automática) "precisa" adicionar postagem de mídia referente ao status do pedido

Ao adicionar postagem de mídia referente ao status do pedido, como dependência:

Empresa (Responsável pela fase em que o pedido se encontra) "precisa poder" gerenciar postagem de mídia referente ao status do pedido

3º Dia: Quinta, 30 de maio de 2024

O sol finalmente brilhou, mas o dia passou e o pôr do sol chegou



Bom fim de tarde pessoal, hoje nós vamos continuar destrinchando o que fazer para alcançar o objetivo da nossa funcionalidade start. Continue seguindo tranquilo e como eu disse anteriormente, não se preocupe se faltar algo, pois quanto mais formos avançando mais lacunas irão surgir e uma a uma elas serão preenchidas.

## **8- Como detalhamos os pré-requisitos que preencheram as lacunas da funcionalidade start?**

Abaixo vamos copiar tudo que foi definido anteriormente como pré-requisito para preencher as lacunas da funcionalidade start e detalhá-los melhor. Antes procuramos facilitar o entendimento colocando menos informações, que agora serão acrescentadas.

Etapa 1: **Cliente** "precisa" **pesquisar pedidos**

**Cliente**

**Comando [01]:** Pesquisar pedidos

**Evento:** Pedidos pesquisados

---

Identificando pré-requisitos:

O que precisa acontecer antes para que os pedidos possam ser pesquisados? O cliente ou alguém precisa fazer o quê?

Cliente "precisa" realizar pedido

Como é a jornada de realização do pedido, por parte do cliente?

Quais comandos ou funcionalidades devem ser executados dentro do seu contexto em ordem cronológica?

1º- Plataforma de delivery precisa carregar catálogo de produtos sugeridos com base num ranking de popularidade.

Plataforma de delivery (Execução automática)  
"precisa" carregar catálogo de produtos sugeridos

Plataforma de delivery (Execução automática)  
Comando: Carregar catálogo de produtos sugeridos

Evento: Catálogo de produtos sugeridos carregado

Confira o desenho completo no miro:

[https://miro.com/app/board/uXjVK-93bzg=?share\\_link\\_id=272758138846](https://miro.com/app/board/uXjVK-93bzg=?share_link_id=272758138846)

2º- Cliente precisa adicionar produto ao carrinho e poder continuar navegando para escolher outros

Cliente "precisa" adicionar produto ao carrinho

Adicionar produto ao carrinho faz parte da jornada de adição do produto escolhido para compra ao carrinho.

Então, como é a jornada de adição do produto escolhido para compra ao carrinho, por parte do cliente?

Quais comandos ele deve executar dentro do seu contexto em ordem cronológica? Quais eventos ocorrem?

## Jornada de adição do produto escolhido para compra ao carrinho:

Início do fluxo principal:

O fluxo já começa com uma decisão a ser tomada.

Interrupção do fluxo para escolha do cenário:

Cenário 1: Adicionar produto pelo catálogo de produtos sugeridos

Início do cenário 1:

**Cliente**

**Política:** Quando produtos do catálogo de produtos sugeridos forem carregados e existir algum produto desejado, então escolher produto

**Cliente**

**Comando:** Escolher produto

**Evento:** Produto escolhido

**Cliente**

**Comando:** Adicionar produto ao carrinho

**Evento:** Produto ao carrinho adicionado

Fim do cenário 1

Cenário 2: Adicionar produto pelo catálogo de produtos da empresa

Início do cenário 2:

**Cliente**

**Política:** Quando produtos do catálogo de produtos sugeridos forem carregados e ainda assim cliente desejar acessar catálogo

de produtos de uma empresa específica, então escolher empresa

Cliente

Comando: Escolher empresa

Evento: Empresa escolhida

Cliente

Comando: Carregar catálogo de produtos da empresa

Evento: Catálogo de produtos da empresa carregado

Cliente

Política: Quando produtos do catálogo de produtos de uma empresa específica forem carregados e existir algum produto desejado, então escolher produto

Cliente

Comando: Escolher produto

Evento: Produto escolhido

Cliente

Comando: Adicionar produto ao carrinho

Evento: Produto ao carrinho adicionado

Fim do cenário 2

Cenário 3: Adicionar produto pelo perfil do produto

Início do cenário 3:

Cliente

Política: Quando produtos do catálogo de produtos sugeridos ou catálogo de produtos da empresa forem carregados e ainda assim cliente desejar



acessar um produto específico,  
então escolher produto

Cliente

Comando: Escolher produto

Evento: Produto escolhido

Cliente

Comando: Carregar produto

Evento: Produto carregado

Cliente

Comando: Adicionar produto ao  
carrinho

Evento: Produto ao carrinho  
adicionado

Fim do cenário 3

Fim do fluxo principal

Confira o desenho completo no miro:

[https://miro.com/app/board/uXjVK-93bzg=/?share\\_link\\_id=272758138846](https://miro.com/app/board/uXjVK-93bzg=/?share_link_id=272758138846)

3º- O cliente precisa visualizar o carrinho com seus produtos adicionados, taxas e valores que serão cobrados.

Cliente "precisa" visualizar carrinho

Visualizar carrinho faz parte da jornada de visualização do carrinho.

Então, como é a jornada de visualização do carrinho, por parte do cliente?

Quais comandos ele deve executar dentro do seu contexto em ordem cronológica? Quais eventos ocorrem?

## Jornada de visualização do carrinho:

Início do fluxo principal:

Cliente

Comando: Visualizar carrinho

Evento: Produtos do carrinho carregados

Evento: Subtotal calculado

Evento: Taxa de entrega adicionada

Evento: Total calculado

Fim do fluxo principal

Confira o desenho completo no miro:

[https://miro.com/app/board/uXjVK-93bzg=?share\\_link\\_id=272758138846](https://miro.com/app/board/uXjVK-93bzg=?share_link_id=272758138846)

Ao visualizar carrinho, como dependência:

Cliente "precisa poder" gerenciar produtos do carrinho

O cliente precisa conferir o carrinho e poder escolher se preciso, remover ou alterar a quantidade de algum produto adicionado.

Gerenciar produtos do carrinho faz parte da jornada de gerenciamento de produtos do carrinho. O cliente precisa ser capaz de olhar o carrinho, conferir os produtos e poder modificar o que achar necessário.

Então, como é a jornada de gerenciamento de produtos do carrinho, por parte do cliente?

Quais comandos ele deve executar dentro do seu contexto em ordem cronológica? Quais eventos ocorrem?

## Jornada de gerenciamento de produtos do carrinho:

Início do fluxo principal:

Cliente

Comando: Conferir produtos do carrinho

Evento: Produtos do carrinho conferidos

Interrupção do fluxo para escolha do cenário:

Cenário 1: Carrinho possui produto indesejado

Início do cenário 1:

Cliente

Política: Quando produtos do carrinho forem conferidos e existir produto indesejado, então remover produto do carrinho

Cliente

Comando: Remover produto do carrinho

Evento: Produto do carrinho removido

Fim do cenário 1

Cenário 2: Carrinho possui produto com quantidade diferente da desejada

Início do cenário 2:

Cliente

Política: Quando produtos do carrinho forem conferidos e existir quantidade diferente da desejada, então alterar quantidade do produto no carrinho

Cliente

**Comando:** Alterar quantidade do produto no carrinho

**Evento:** Quantidade do produto no carrinho alterada

Fim do cenário 2

Fim do fluxo principal

Confira o desenho completo no miro:

[https://miro.com/app/board/uXjVK-93bzg=/?share\\_link\\_id=272758138846](https://miro.com/app/board/uXjVK-93bzg=/?share_link_id=272758138846)

4º- O cliente precisa fechar o carrinho após estar certo dos produtos que adicionou ao carrinho.

**Cliente** "precisa" **fechar carrinho**

**Cliente**

**Comando:** Fechar carrinho

**Evento:** Carrinho fechado

Confira o desenho completo no miro:

[https://miro.com/app/board/uXjVK-93bzg=/?share\\_link\\_id=272758138846](https://miro.com/app/board/uXjVK-93bzg=/?share_link_id=272758138846)

5º- O cliente precisa informar os dados que possibilitam a entrega e cobrança.

**Cliente** "precisa" **realizar checkout**

Realizar checkout, faz parte da jornada de realização de checkout.

Então, como é a jornada de realização de checkout, por parte do cliente?

Quais comandos ele deve executar dentro do seu contexto em ordem cronológica? Quais eventos ocorrem?

## Jornada de realização do checkout:

Início do fluxo principal:

Cliente

Comando: Informar endereço de entrega

Evento: Endereço de entrega informado

Cliente

Comando: Informar dados de cobrança

Evento: Dados de cobrança informado

Cliente

Comando: Realizar compra

Evento: Dados de cobrança enviados para processamento

Fim do fluxo principal

Confira o desenho completo no miro:

[https://miro.com/app/board/uXjVK-93bzg=?share\\_link\\_id=272758138846](https://miro.com/app/board/uXjVK-93bzg=?share_link_id=272758138846)

O que precisa acontecer antes para que o cliente possa realizar o pedido? O cliente ou alguém precisa fazer o quê?

Possibilitar o funcionamento da plataforma de delivery, permitindo a execução de funcionalidades base.

---

Quais comandos ou funcionalidades devem ser executados?

1º- Precisa existir empresa cadastrada na plataforma de delivery para oferecer serviço.

Empresa (Admin) "precisa" cadastrar empresa

Ao cadastrar empresa, como dependência a empresa precisa poder: Gerenciar empresa



Gerenciar empresa é composto de comandos básicos, que podem ser executados individualmente a qualquer momento.

Para a empresa poder oferecer seu serviço de delivery, então a empresa precisa poder:

Empresa (Admin)

Comando: Cadastrar empresa

Evento: Empresa cadastrada

Empresa (Admin)

Comando: Pesquisar empresa

Evento: Empresa pesquisada

Empresa (Admin)

Comando: Alterar empresa

Evento: Empresa alterada

Empresa (Admin)

Comando: Suspende empresa

Evento: Empresa suspensa

Empresa (Admin)

Comando: Remover suspensão da empresa

Evento: Suspensão da empresa removida

Empresa (Admin)

Comando: Cancelar empresa

Evento: Empresa cancelada

Confira o desenho completo no miro:

[https://miro.com/app/board/uXjVK-93bzg=/?share\\_link\\_id=272758138846](https://miro.com/app/board/uXjVK-93bzg=/?share_link_id=272758138846)

2º- Empresa precisa poder cadastrar funcionário para gerenciar perfil de acesso, atribuindo aos responsáveis as fases do pedido na plataforma de delivery.

Empresa (Admin) "precisa" cadastrar funcionário

Ao cadastrar empresa, como dependência a empresa precisa poder: Gerenciar funcionário

Gerenciar funcionário é composto de comandos básicos, que podem ser executados individualmente a qualquer momento.

Para a empresa poder gerenciar funcionário, então a empresa precisa poder:

Empresa (Admin)

Comando: Cadastrar funcionário

Evento: Funcionário cadastrado

Empresa (Admin)

Comando: Pesquisar funcionários

Evento: Funcionários pesquisados

Empresa (Admin)

Comando: Pesquisar funcionário

Evento: Funcionário pesquisado

Empresa (Admin)

Comando: Alterar funcionário

Evento: Funcionário alterado

Empresa (Admin)

Comando: Suspender funcionário

Evento: Funcionário suspenso

Empresa (Admin)

Comando: Remover suspensão do funcionário

Evento: Suspensão do funcionário removida

Empresa (Admin)

Comando: Cancelar funcionário

Evento: Funcionário cancelado

Confira o desenho completo no miro:

[https://miro.com/app/board/uXjVK-93bzg=/?share\\_link\\_id=272758138846](https://miro.com/app/board/uXjVK-93bzg=/?share_link_id=272758138846)

3º- A empresa precisa cadastrar produtos na plataforma de delivery para venda.

Empresa (Admin) "precisa" cadastrar produto

Ao cadastrar produto, como dependência a empresa precisa poder: Gerenciar produto

Gerenciar produto é composto de comandos básicos, que podem ser executados individualmente a qualquer momento.

Para a empresa poder vender seu produto no sistema de delivery, então a empresa precisa poder:

Empresa (Admin)

Comando: Cadastrar produto

Evento: Produto cadastrado

Empresa (Admin)

Comando: Pesquisar produtos

Evento: Produtos pesquisados

Empresa (Admin)

Comando: Pesquisar produto

Evento: Produto pesquisado

Empresa (Admin)

Comando: Alterar produto

Evento: Produto alterado

Empresa (Admin)

Comando: Suspende produto

Evento: Produto suspenso

Empresa (Admin)

Comando: Remover suspensão do produto

Evento: Suspensão do produto removida

Empresa (Admin)

Comando: Remover produto

Evento: Produto removido

Confira o desenho completo no miro:

[https://miro.com/app/board/uXjVK-93bzg=?share\\_link\\_id=272758138846](https://miro.com/app/board/uXjVK-93bzg=?share_link_id=272758138846)

4º- Empresa precisa poder gerenciar qualquer cliente que já tenha feito algum pedido nela. Isso para poder suspender ou cancelar a conta deste por algum motivo. Seja por fraude ou descumprimento de alguma regra da empresa. E ainda visualizar aqueles que mantêm a conexão com a empresa ativa para receber postagens do status do seu pedido.

Empresa (Admin) "precisa poder" gerenciar carteira de clientes

Gerenciar carteira de clientes é composta de comandos básicos, que podem ser executados individualmente a qualquer momento.

Para a empresa gerenciar carteira de clientes na plataforma de delivery, então ela precisa poder:

Empresa (Admin)

Comando: Pesquisar clientes com conexão ativa

Evento: Clientes com conexão ativa pesquisados

Empresa (Admin)

Comando: Pesquisar cliente

Evento: Cliente pesquisado

Empresa (Admin)

Comando: Suspender cliente

Evento: Cliente suspenso

Empresa (Admin)

Comando: Remover suspensão do cliente

Evento: Suspensão do cliente removida

Confira o desenho completo no miro:

[https://miro.com/app/board/uXjVK-93bzg=/?share\\_link\\_id=272758138846](https://miro.com/app/board/uXjVK-93bzg=/?share_link_id=272758138846)

5º- O cliente precisa se cadastrar na plataforma de delivery para realizar o pedido.

Cliente "precisa" cadastrar cliente

Ao cadastrar cliente, como dependência cliente precisa poder: Gerenciar cliente

Gerenciar cliente é composto de comandos básicos, que podem ser executados individualmente a qualquer momento.

Para o cliente realizar pedido no sistema de delivery, então ele precisa poder:

Cliente

Comando: Cadastrar cliente

Evento: Cliente cadastrado

Cliente

Comando: Pesquisar cliente

Evento: Cliente pesquisado

Cliente

Comando: Alterar cliente

Evento: Cliente alterado

Cliente

Comando: Suspender cliente

Evento: Cliente suspenso

Cliente

Comando: Remover suspensão do cliente

Evento: Suspensão do cliente removida

Cliente

Comando: Cancelar cliente

Evento: Cliente cancelado

Confira o desenho completo no miro:

[https://miro.com/app/board/uXjVK-93bzg=/?share\\_link\\_id=272758138846](https://miro.com/app/board/uXjVK-93bzg=/?share_link_id=272758138846)

6º- O cliente precisa poder gerenciar qualquer empresa que ele tenha realizado algum pedido. Isso para habilitar se deseja receber ou não as postagens de status do pedido. Postagens essas que começam assim que o pedido é realizado e prosseguem se o cliente não informar nenhum impedimento através do gerenciamento da conexão na sua carteira de empresas.

**Cliente** "precisa poder" gerenciar carteira de empresas

Gerenciar carteira de empresas é composto de comandos básicos, que podem ser executados individualmente a qualquer momento.

Para o cliente gerenciar carteira de empresas e permitir ou não receber postagem de status do pedido, então ele precisa poder:

**Cliente**

**Comando:** Pesquisar empresas com conexão ativa

**Evento:** Empresas com conexão ativa pesquisadas

**Cliente**

**Comando:** Pesquisar empresa

**Evento:** Empresa pesquisada

**Cliente**

**Comando:** Suspende empresa

**Evento:** Empresa suspensa

**Cliente**

**Comando:** Remover suspensão da empresa

**Evento:** Suspensão da empresa removida

**Cliente**

**Comando:** Suspende conexão

**Evento:** Conexão suspensa

**Cliente**

**Comando:** Remover suspensão da conexão

**Evento:** Suspensão da conexão  
removida

Confira o desenho completo no miro:

[https://miro.com/app/board/uXjVK-93bzg=/?share\\_link\\_id=272758138846](https://miro.com/app/board/uXjVK-93bzg=/?share_link_id=272758138846)

## Etapa 2: Cliente "precisa" escolher pedido

Cliente

Comando [02]: Escolher pedido

Evento: Pedido escolhido

---

Identificando pré-requisitos:

O que precisa acontecer antes para que o pedido possa ser escolhido para visualização das postagens de status? O cliente ou alguém precisa fazer o quê?

Quais comandos ou funcionalidades devem ser executados dentro do seu contexto em ordem cronológica?

1º- A empresa precisa escolher um pedido que esteja aguardando aprovação, verificar se está de acordo com as regras da empresa e dizer se aceita ou cancela. Essas regras podem ser diversas como:  
- Hoje estou sem entregador, então retirada somente na loja. Todas ficam a cargo da empresa de delivery.

Empresa (Atendente) "precisa" aceitar pedido

Aceitar pedido faz parte da jornada de aceitação do pedido.

Então, como é a jornada de aceitação do pedido por parte da empresa?

Quais comandos ela deve executar dentro do seu contexto em ordem cronológica? Quais eventos ocorrem?

Jornada de aceitação do pedido:

Início do fluxo principal:

Empresa (Atendente)

Comando: Pesquisar pedidos com status "pagamento confirmado"

Evento: Pedidos com status "pagamento confirmado" pesquisados



**Empresa (Atendente)**

**Comando:** Escolher pedido

**Evento:** Pedido escolhido

Interrupção do fluxo para escolha do cenário:

Cenário 1: Pedido aceito

Início do cenário 1:

**Empresa (Atendente)**

**Política:** Quando pedido escolhido estiver dentro das regras da empresa, então aceitar pedido

**Empresa (Atendente)**

**Comando:** Aceitar pedido

**Evento:** Status aguardando preparo adicionado

Fim do cenário 1

Cenário 2: Pedido cancelado

Início do cenário 2:

**Empresa (Atendente)**

**Política:** Quando o pedido escolhido estiver fora das regras da empresa, então cancelar o pedido

**Empresa (Atendente)**

**Comando:** Cancelar pedido

**Evento:** Status cancelado adicionado

Fim do cenário 2

Após o cenário ser escolhido acima, voltamos ao fluxo principal e terminamos logo em seguida.

Fim do fluxo principal

Confira o desenho completo no miro:

[https://miro.com/app/board/uXjVK-93bzg=/?share\\_link\\_id=272758138846](https://miro.com/app/board/uXjVK-93bzg=/?share_link_id=272758138846)

Etapa 3: Cliente "precisa" pesquisar postagens de status do pedido

Cliente

Comando [03]: Pesquisar postagens de status do pedido

Evento: Postagens de status do pedido pesquisadas

---

Identificando pré-requisitos:

O que precisa acontecer antes para que as postagens de status do pedido possam ser pesquisadas? O cliente ou alguém precisa fazer o quê?

Empresa (Responsável pela fase em que o pedido se encontra) "precisa" adicionar status ao pedido

Quais comandos ou funcionalidades devem ser executados dentro do seu contexto em ordem cronológica para adição de status ao pedido?

1º- A empresa precisa na fase de preparo que normalmente acontece na cozinha, adicionar o novo status do pedido a cada etapa da preparação que ele avança.

Empresa (Responsável pela preparação do pedido) "precisa" adicionar status ao pedido, na fase de preparo

Percebemos que adicionar status ao pedido possui algumas jornadas que se complementam de forma sequencial, até alcançar o objetivo de adicionar todos os status ao pedido.

Adicionar status ao pedido, faz parte da jornada de adição de status ao pedido, na fase de preparo.

Então, como é a jornada de adição de status ao pedido, na fase de preparo?

Quais comandos ela deve executar dentro do seu contexto em ordem cronológica? Quais eventos ocorrem?

Jornada de adição de status ao pedido, na fase de preparo:

Início do fluxo principal:

Empresa (Responsável pela preparação do pedido)

Comando: Pesquisar pedidos com status aguardando preparo

Evento: Pedidos com status aguardando preparo pesquisados

Empresa (Responsável pela preparação do pedido)

Comando: Escolher pedido

Evento: Pedido escolhido

Empresa (Responsável pela preparação do pedido)

Comando: Escolher status

Evento: Status escolhido

Empresa (Responsável pela preparação do pedido)

Comando: Adicionar status

Evento: Status adicionado

Interrupção do fluxo para escolha do cenário:

Cenário 1: Pedido enviado para entrega

Início do cenário 1:

Empresa (Responsável pela preparação do pedido)

Política: Quando último status do pedido for adicionado, então enviar pedido para entrega

Empresa (Responsável pela preparação do pedido)

Comando: Enviar pedido para entrega

Evento: Pedido para entrega enviado

Fim do cenário 1

Após o cenário ser escolhido acima, voltamos ao fluxo principal e terminamos logo em seguida.

Fim do fluxo principal

Confira o desenho completo no miro:

[https://miro.com/app/board/uXjVK-93bzg=/?share\\_link\\_id=272758138846](https://miro.com/app/board/uXjVK-93bzg=/?share_link_id=272758138846)

O que precisa acontecer antes para que o cliente possa adicionar status ao pedido, na fase de preparo? O cliente ou alguém precisa fazer o quê?

Possibilitar a execução das funcionalidades abaixo.

---

Quais comandos ou funcionalidades devem ser executados?

1º- A empresa precisa cadastrar processo de preparo. Isso é particular de cada empresa e produto. São esses processos de preparo que serão usados para orientar o envio de status do pedido que está sendo produzido.

Empresa (Admin) "precisa" cadastrar processo de preparo

Ao cadastrar processo de preparo, como dependência empresa precisa poder: Gerenciar processo de preparo

Gerenciar processo de preparo é composto de comandos básicos, que podem ser executados individualmente a qualquer momento.

Para a empresa gerenciar o processo de preparo, então a ela precisa poder:

Empresa (Admin)

Comando: Cadastrar processo de preparo

Evento: Processo de preparo cadastrado

Empresa (Admin)

Comando: Pesquisar processos de preparo

Evento: Processos de preparo pesquisados

Empresa (Admin)

Comando: Pesquisar processo de preparo

Evento: Processo de preparo pesquisado

Empresa (Admin)

Comando: Alterar processo de preparo

Evento: Processo de preparo alterado

Empresa (Admin)

Comando: Suspender processo de preparo

Evento: Processo de preparo suspenso

Empresa (Admin)

Comando: Remove  
suspensão do processo de  
preparo

Evento: Suspensão do  
processo de preparo  
removida

Empresa (Admin)

Comando: Remove  
processo de preparo

Evento: Processo de  
preparo removido

Confira o desenho completo no  
miro:

[https://miro.com/app/board/uXjVK-93bzg=/?share\\_link\\_id=272758138846](https://miro.com/app/board/uXjVK-93bzg=/?share_link_id=272758138846)

2º- A empresa precisa cadastrar status de  
preparo. Isso é particular de cada empresa  
e produto. São esses status que serão  
usados para notificar ao cliente o passo a  
passo do seu pedido.

Empresa (Admin) "precisa" cadastrar  
status de preparo

Ao cadastrar status de preparo,  
como dependência empresa  
precisa poder: Gerenciar status  
de preparo

Gerenciar status de preparo é  
composto de comandos básicos,  
que podem ser executados  
individualmente a qualquer  
momento.

Para a empresa gerenciar o  
status referente a cada fase da

preparação do pedido, então a ela precisa poder:

Empresa (Admin)

Comando: Cadastrar status de preparo

Evento: Status de preparo cadastrado

Empresa (Admin)

Comando: Pesquisar todos status de preparo

Evento: Todos status de preparo pesquisados

Empresa (Admin)

Comando: Pesquisar status de preparo

Evento: Status de preparo pesquisado

Empresa (Admin)

Comando: Alterar status de preparo

Evento: Status de preparo alterado

Empresa (Admin)

Comando: Suspende status de preparo

Evento: Status de preparo suspenso

Empresa (Admin)

Comando: Remove suspensão do status de preparo

Evento: Suspensão do status de preparo removida

Empresa (Admin)

Comando: Remover status de preparo

Evento: Status de preparo removido

Confira o desenho completo no miro:

[https://miro.com/app/board/uXjVK-93bzg=?share\\_link\\_id=272758138846](https://miro.com/app/board/uXjVK-93bzg=?share_link_id=272758138846)

2º- Empresa precisa na fase de retirada para entrega, adicionar o novo status do pedido que está pronto e foi retirado para entrega.

Empresa (Responsável por entregar o pedido)

"precisa" adicionar status ao pedido, na fase de retirada para entrega

Como dito anteriormente, percebemos que adicionar status ao pedido possui algumas jornadas que se complementam de forma sequencial, até alcançar o objetivo de adicionar todos os status ao pedido.

Adicionar status ao pedido, faz parte da jornada de adição de status ao pedido, na fase de retirada para entrega

Então, como é a jornada de adição de status ao pedido, na fase de retirada para entrega?

Quais comandos ela deve executar dentro do seu contexto em ordem cronológica? Quais eventos ocorrem?

Jornada de adição de status ao pedido, na fase de retirada para entrega:

Início do fluxo principal:

Empresa (Responsável pela entrega do pedido)



**Comando:** Pegar pedidos prontos e aguardando saída

**Evento:** Pedidos prontos e aguardando saída pegos

Empresa (Responsável pela entrega do pedido)

**Comando:** Escanear QR code do pedido

**Evento:** QR code do pedido escaneado

Interrupção do fluxo para escolha do cenário:

Cenário 1: Pedido aceito

Início do cenário 1:

Empresa (Responsável pela entrega do pedido)

**Política:** Quando o QR code do pedido for escaneado, então adicionar status "saiu para entrega"

Empresa (Responsável pela entrega do pedido)

**Comando:** Adicionar status saiu para entrega

**Evento:** Status saiu para entrega adicionado

Fim do cenário 1

Após o cenário ser executado acima, voltamos ao fluxo principal e terminamos logo em seguida.

Fim do fluxo principal

Confira o desenho completo no miro:

[https://miro.com/app/board/uXjVK-93bzg=/?share\\_link\\_id=272758138846](https://miro.com/app/board/uXjVK-93bzg=/?share_link_id=272758138846)

3º- Empresa precisa na fase de rota de entrega, adicionar status do pedido que foi entregue ou não entregue.

Empresa (Responsável pela entrega do pedido)  
"precisa" adicionar status ao pedido, na fase de rota de entrega

A última jornada para alcançar o objetivo de adicionar todos os status ao pedido é na fase de rota de entrega.

Adicionar status ao pedido, faz parte da jornada de adição de status ao pedido, na fase de rota de entrega.

Então, como é a jornada de adição de status ao pedido, na fase de rota de entrega?

Quais comandos ela deve executar dentro do seu contexto em ordem cronológica? Quais eventos ocorrem?

Jornada de adição de status ao pedido, na fase de rota de entrega:

Início do fluxo principal:

Empresa (Responsável pela entrega do pedido)

Comando: Pesquisar pedidos com status "saiu para entrega" por entregador

Evento: Pedidos com status "saiu para entrega" por entregador pesquisados

Empresa (Responsável pela entrega do pedido)

Comando: Escolher pedido

Evento: Pedido escolhido

Empresa (Responsável pela entrega do pedido)

Comando: Escolher status

Evento: Status escolhido

Interrupção do fluxo para escolha do cenário desejado:

#### Cenário 1: Pedido entregue

Início do cenário 1

Empresa (Responsável pela entrega do pedido)

**Política:** Quando o status escolhido for "entregue", então informar o código do cliente

Empresa (Responsável pela entrega do pedido)

**Comando:** Informar código do cliente

**Evento:** Código do cliente informado

Empresa (Responsável pela entrega do pedido)

**Comando:** Adicionar status entregue

**Evento:** Status entregue adicionado

Fim do cenário 1

#### Cenário 2: Pedido não entregue

Início do cenário 2

Empresa (Responsável pela entrega do pedido)

**Política:** Quando o status escolhido for "não entregue", então informar o motivo da não entrega

Empresa (Responsável pela entrega do pedido)

**Comando:** Informar o motivo da não entrega

**Evento:** Áudio adicionado

Empresa (Responsável pela entrega do pedido)

Comando: Adicionar status não entregue

Evento: Status não entregue adicionado

Fim do cenário 2

Após o cenário ser escolhido acima, voltamos ao fluxo principal e terminamos logo em seguida.

Fim do fluxo principal

Confira o desenho completo no miro:

[https://miro.com/app/board/uXjVK-93bzg=?share\\_link\\_id=272758138846](https://miro.com/app/board/uXjVK-93bzg=?share_link_id=272758138846)

4º- Empresa precisa poder habilitar nova tentativa de entrega ao pedido, que tenha no máximo uma tentativa mal sucedida. Ao habilitar a nova tentativa, o pedido tem o status "saiu para entrega" adicionado novamente.

Empresa (Responsável pela entrega do pedido) "precisa poder" Habilitar nova tentativa de entrega, na fase de rota de entrega

Habilitar nova tentativa de entrega, faz parte da jornada de habilitação de nova tentativa de entrega, na fase de rota de entrega.

Então, como é a jornada de habilitação de nova tentativa de entrega, na fase de rota de entrega?

Quais comandos ela deve executar dentro do seu contexto em ordem cronológica? Quais eventos ocorrem?

Jornada de habilitação de nova tentativa de entrega, na fase de rota de entrega:

Início do fluxo principal:

Empresa (Responsável pela entrega do pedido)

**Comando:** Pesquisar pedidos com status "não entregue" por entregador

**Evento:** Pedidos com status "não entregue" por entregador pesquisados

Empresa (Responsável pela entrega do pedido)

**Comando:** Escolher pedido

**Evento:** Pedido escolhido

Empresa (Responsável pela entrega do pedido)

**Comando:** Habilitar nova tentativa de entrega

**Evento:** Status "saiu para entrega" adicionado

Fim do fluxo principal

Confira o desenho completo no miro:

[https://miro.com/app/board/uXjVK-93bzg=/?share\\_link\\_id=272758138846](https://miro.com/app/board/uXjVK-93bzg=/?share_link_id=272758138846)

## Jornadas background

---

As jornadas background, são ações que não dependem mais da interação humana, pois algum evento que ocorreu já a invocou ou invocará quando acontecer.

1º- Plataforma de delivery precisa acompanhar o processamento do pagamento e adicionar pedido.

Plataforma de delivery (Execução automática) "precisa" adicionar pedido

Adicionar pedido, faz parte da jornada background de adição do pedido.

Então, como é a jornada de adição do pedido?

Quais comandos devem ser executados dentro do seu contexto em ordem cronológica? Quais eventos ocorrem?

Jornada background de adição do pedido:

Início do fluxo principal:

Plataforma de delivery (Execução automática)

Comando: Acompanhar processamento do pagamento

Evento: Processamento do pagamento acompanhado

Interrupção do fluxo para escolha do cenário:

Cenário 1: Pagamento processado

Início do cenário 1:

Plataforma de delivery (Execução automática)

Política: Quando pagamento for processado, então adicionar pedido com status de pagamento confirmado

Comando: Adicionar pedido

Evento: Pedido adicionado

Fim do cenário 1

Cenário 2: Pagamento não processado

Início do cenário 2:

Plataforma de delivery (Execução automática)

**Política:** Quando pagamento não for processado, então adicionar pedido com status de pagamento recusado

**Comando:** Adicionar status de pagamento

**Evento:** Status de pagamento recusado adicionado

Fim do cenário 2

Após o cenário ser escolhido acima, voltamos ao fluxo principal e terminamos logo em seguida.

Fim do fluxo principal

Confira o desenho completo no miro:

[https://miro.com/app/board/uXjVK-93bzg=?share\\_link\\_id=272758138846](https://miro.com/app/board/uXjVK-93bzg=?share_link_id=272758138846)

2º- Plataforma de delivery precisa notificar de forma automática todos os envolvidos no pedido, ou seja, a empresa e o cliente.

Plataforma de delivery (Execução automática) "precisa" notificar partes interessadas

Notificar partes interessadas pode ser executado conforme a necessidade e critério de notificação estabelecido.

Plataforma de delivery (Execução automática)

**Política:** Quando qualquer pedido for adicionado, então notificar

**Comando:** Notificar partes interessadas

**Evento:** Partes interessadas notificadas

Confira o desenho completo no miro:

[https://miro.com/app/board/uXjVK-93bzg=?share\\_link\\_id=272758138846](https://miro.com/app/board/uXjVK-93bzg=?share_link_id=272758138846)

3º- Plataforma de delivery após qualquer mudança de status do pedido, precisa adicionar alguma postagem de mídia que está relacionada ao status atual do pedido e que foi cadastrada de antemão pela empresa na plataforma.

Plataforma de delivery (Execução automática) "precisa" adicionar postagem de mídia referente ao status do pedido

Adicionar postagem de mídia referente ao status do pedido, pertence a jornada background de adição de postagem de mídia ao status do pedido, que é invocada sempre que um novo status é adicionado ao pedido.

Então, como é a jornada de adição de postagem de mídia referente ao status do pedido?

Quais comandos devem ser executados dentro do seu contexto em ordem cronológica? Quais eventos ocorrem?

Jornada background de adição de postagem de mídia referente ao status do pedido:

Início do fluxo principal:

O fluxo já começa com uma decisão a ser tomada.

Interrupção do fluxo para escolha do cenário:

Cenário 1: Postagem adicionada

Início do cenário 1:

Plataforma de delivery (Execução automática)

Política: Quando qualquer status for adicionado ao pedido, então verificar se existe suspensão de conexão entre cliente e empresa

Plataforma de delivery (Execução automática)

Comando: Verificar suspensão de conexão



**Evento:** Suspensão de conexão verificada

Plataforma de delivery (Execução automática)

**Política:** Quando conexão entre cliente e empresa existir, então adicionar postagem de mídia ao status do pedido

Plataforma de delivery (Execução automática)

**Comando:** Adicionar postagem de mídia referente ao status do pedido

**Evento:** Mídia adicionada

**Evento:** Descrição da mídia adicionada

**Evento:** Postagem de mídia referente ao status do pedido adicionada

Fim do cenário 1

Após o cenário ser executado acima, voltamos ao fluxo principal e terminamos logo em seguida.

Fim do fluxo principal

Confira o desenho completo no miro:

[https://miro.com/app/board/uXjVK-93bzg=/?share\\_link\\_id=272758138846](https://miro.com/app/board/uXjVK-93bzg=/?share_link_id=272758138846)

Ao adicionar postagem de mídia referente ao status do pedido pela **execução automática** da plataforma, como dependência **empresa** precisa poder: **Gerenciar postagem de mídia referente ao status do pedido**

Gerenciar postagem de mídia referente ao status do pedido é composta por alguns comandos básicos, que podem ser executados sempre que uma postagem for adicionada.

Para a empresa gerenciar postagem de status de mídia ao pedido, então ela precisa poder:

**Empresa** (Responsável pela fase em que o pedido se encontra)

**Comando:** Pesquisar postagens de mídia referente ao status do pedido

**Evento:** Postagens de mídia referente ao status do pedido pesquisadas

Empresa (Responsável pela fase em que o pedido se encontra)

**Comando:** Remover postagem de mídia referente ao status do pedido

**Evento:** Postagem de mídia referente ao status do pedido removida

Confira o desenho completo no miro:

[https://miro.com/app/board/uXjVK-93bzg=?share\\_link\\_id=272758138846](https://miro.com/app/board/uXjVK-93bzg=?share_link_id=272758138846)