

**Instituto de Gestão e Tecnologia da Informação**

**Relatório do Projeto Aplicado**

Modelo arquitetural para refatoração da camada de backend do processo de faturamento de pedidos

Matheus Fenólio do Prado

Orientado: Professor Reinaldo Galvão

Junho de 2022



**Matheus Fenólio do Prado**

**INSTITUTO DE GESTÃO E TECNOLOGIA DA INFORMAÇÃO**RELATÓRIO DO PROJETO APLICADO

Modelo arquitetural para refatoração da camada de backend do processo de faturamento de pedidos

Relatório de Projeto Aplicado desenvolvido para fins de conclusão do curso MBA em Arquitetura de software e soluções.  
  
Orientador: Professor Reinaldo Galvão

**Espirito Santo do Pinhal  
Junho de 2022**

**Sumário**

[1. CANVAS do Projeto Aplicado 4](#_Toc101281510)

[1.1 Desafio 5](#_Toc101281511)

[1.1.1 Análise de Contexto 5](#_Toc101281512)

[1.1.2 Personas 6](#_Toc101281513)

[1.1.3 Benefícios e Justificativas 7](#_Toc101281514)

[1.1.4 Hipóteses 8](#_Toc101281515)

[1.2 Solução 9](#_Toc101281516)

[1.2.1 Objetivo SMART 9](#_Toc101281517)

[1.2.2 Premissas e Restrições 11](#_Toc101281518)

[1.2.3 Backlog de Produto 13](#_Toc101281519)

[2. Área de Experimentação 14](#_Toc101281520)

[2.1 Sprint 1 16](#_Toc101281521)

[2.1.1 Solução 16](#_Toc101281522)

[● Evidência do planejamento: 16](#_Toc101281523)

[● Evidência da execução de cada requisito: 16](#_Toc101281524)

[● Evidência dos resultados: 16](#_Toc101281525)

[2.1.2 Experiências vivenciadas 16](#_Toc101281526)

[2.2 Sprint 2 17](#_Toc101281527)

[2.2.1 Solução 17](#_Toc101281528)

[● Evidência do planejamento: 17](#_Toc101281529)

[● Evidência da execução de cada requisito: 17](#_Toc101281530)

[● Evidência dos resultados: 17](#_Toc101281531)

[2.2.2 Experiências vivenciadas 17](#_Toc101281532)

[2.3 Sprint 3 18](#_Toc101281533)

[2.3.1 Solução 18](#_Toc101281534)

[● Evidência do planejamento: 18](#_Toc101281535)

[● Evidência da execução de cada requisito: 18](#_Toc101281536)

[● Evidência dos resultados: 18](#_Toc101281537)

[2.3.2 Experiências vivenciadas 18](#_Toc101281538)

[3. Considerações Finais 19](#_Toc101281539)

[3.1 Resultados 19](#_Toc101281540)

[3.2 Contribuições 19](#_Toc101281541)

[3.3 Próximos passos 19](#_Toc101281542)

## 1. CANVAS do Projeto Aplicado

## Desafio

### 1.1.1 Análise de Contexto

Uma empresa possui um sistema para fazer ordens de compra. Ele foi primariamente baseado para atender poucas pessoas realizando vendas na rede local, porém com o aumento da demanda, novos vendedores, tanto internos quanto externos foram contratados. Isso gerou uma sobrecarga no processo que a realiza o faturamento de pedidos no ERP principal de controle do setor fiscal. Essa sobrecarga frequentemente gera erros, travamentos de ordens, além de não pode ser executado por vendedores externos.

**Matriz CSD**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Certezas** | **Suposições** | **Duvidas** |
| **Atores** | * O time de vendas sabe que o sistema não foi planejado para o uso atual. * O time de desenvolvimento tem vontade de refatorar esse processo pelo numero de chamados gerados para o suporte. |  |  |
| **Cenários** | * Ordens devem ser faturadas o mais rápido possível. * Sistema deve ser capaz de se recuperar caso um faturamento de errado * Deve ser possível faturar pedidos externamente |  |  |
| **Regras** | * É preciso criar um plano de refatoração * Desenho da nova arquitetura | * Precisa ser escalável |  |

**POEMS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Pessoas** | **Objetos** | **Ambiente** | **Mensagem** | **Serviços** |
| Quem está envolvido no contexto em análise? | Que objetos fazem parte do ambiente? | Quais são as características do ambiente? | Que mensagens são comunicadas? | Quais serviços são oferecidos? |
| **Desenvolvedores** | Notebook | Escritório | Desenvolvimento de demandas relacionadas aos sistemas próprios | Desenvolvimento das funcionalidades e correção de bugs |
| **Suporte de T.I** | Notebook | Escritório | Atuando no suporte a usuários dos sistemas internos | Apoio as operações realizadas nos sistemas |
| **Vendedores internos** | Notebook | Escritório | Vendendo produtos para clientes | Proporcionando novos negócios para a empresa. |
| **Vendedores externos** | Notebook | Residencia do cliente | Entrando em contato com possíveis novos clientes | Proporcionando novos negócios para a empresa. |

|  |  |
| --- | --- |
| **Registros** | **Insights** |
| As informações iniciais foram obtidas através de entrevista com vendedores, e com pessoal de suporte da empresa | * Envolver bastante os desenvolvedores e arquitetos no desenho da solução * Envolver o setor de vendas, suporte e processos. |

### 1.1.2 Personas



**Roberto Castro**

* Gerente de TI
* 45 anos
* 9 anos de empresa

**O que ele pensa e sente?**

* Fica feliz em ver a empresa crescendo com ajuda da tecnologia.
* Frustração por conta da quantidade de tickets por conta do sistema que não foi desenhado para situação atual.
* Tem medo que o processo de faturamento pare o setor de vendas.

**O que ele escuta?**

* Processo de faturamento é lento e com falhas constantes.
* Vendedores externos precisam ligar para vendedores internos pedindo para que uma ordem seja faturada.

**O que fala e faz?**

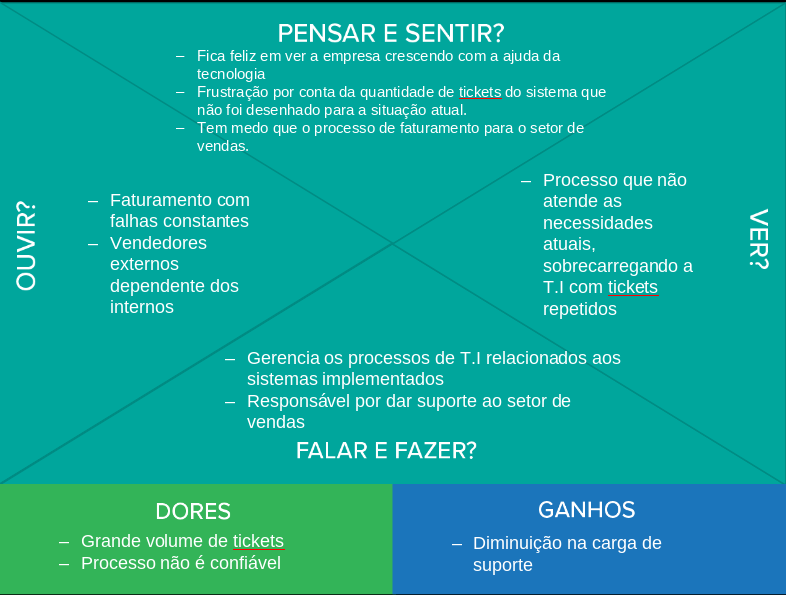
* Gerencia os processos de T.I relacionados aos sistemas implementados.
* Responsável por dar suporte ao setor de venda**.**

**O que ele vê?**

* Processo que não atende as necessidades atuais, sobrecarregando a T.I com tickets repetidos.

**Quais são suas necessidades?**

* Precisa de um processo mais confiável e que atenda as novas necessidades.



**Ana Camargo**

* Vendedora interna
* 45 anos
* 10 anos de empresa

**O que ele pensa e sente?**

* Vendas demoram muito para serem faturadas.
* Com frequência precisa abrir um ticket para uma ordem que travou por um erro no faturamento.
* Perde tempo faturando ordens para vendedores externos que não tem acesso ao processo.

**O que ele escuta?**

* Ordens não estão fluindo e ela precisa ser mais produtiva para conseguir atender mais clientes.

**O que fala e faz?**

* Realiza vendas por telefone a clientes fidelizados.

**O que ele vê?**

* Processo ineficiente que a atrapalha na performance e experiencia de seu atendimento.

**Quais são suas necessidades?**

* Processo mais rápido e tolerante a falha.



**Altamir Abrantes**

* Vendedor externo
* 25 anos
* 2 anos de empresa

**O que ele pensa e sente?**

* Não tem autonomia, pois não pode faturar ordens por conta própria.
* O sistema não é capaz de atender suas demandas.

**O que ele escuta?**

* Suas ordens demoram demais para serem efetivadas.

**O que fala e faz?**

* Realiza visita a possíveis novos clientes.

**O que ele vê?**

* Sistema que o impede de atender o cliente na hora de sua visita, pois o mesmo não o permite.

**Quais são suas necessidades?**

* Um processo que o permita atender o cliente no mesmo instante de sua visita, para garantir a satisfação do mesmo e garantir sua fidelização.



### 1.1.3 Benefícios e Justificativas

|  |  |
| --- | --- |
| **Itens** | **Faturamento de pedidos** |
| **Objetivos** | Realizar a inserção de dados necessários para criação de notas fiscais e boletos no sistema fiscal |
| **Atividades** | Consumir dados de uma filha e processa-los no sistema fiscal |
| **Questões** | - Podemos ter comportamentos diferentes para cada rodem?  - Como devemos lidar com falhas? |
| **Barreiras** | Consistência dos dados necessários |
| **Ações do cliente** | Checar os dados do pedido e do cliente.  Iniciar o processo de faturamento |
| **Funcionalidades** | Fatura um pedido |
| **Interação** | Reage a um evento produzido pelo sistema de compras |
| **Mensagem** | Faturamento em progresso  Faturamento com status {sucesso | falha} |
| **Onde ocorre** | Após o ERP de pedidos produzir um evento na fila |
| **Tarefas aparentes** | Restruturar o fluxo.  Criar uma maneira de notificar o ERP de pedidos.  Criar o producer e consumer |
| **Tarefas escondidas** | N/A |
| **Processos de suporte** | LOGS |
| **Saída desejável** | Pedido faturado |



### 1.1.4 Hipóteses

* Matriz de observações para hipóteses.

|  |  |
| --- | --- |
| **Observação** | **Hipótese** |
| Devemos tentar reprocessar uma ordem em caso de erro? | Podemos implementar um sistema de retry antes de retornar o erro |
| Devemos usar mensageiria? | Para facilitar o processo de expor essa funcionalidade. Um endpoint seria mais adequado |
| Serviço deve ficar o tempo todo online? | Podemos criar regras para escalar o microsserviço baseado na hora para evitar uso desnecessário de recursos. |
| Devemos fazer a migração completa ou aos poucos? | Podemos fazer migração aos poucos, criando um estratégia de migrar os vendedores que atendem cliente com menor criticidade |

* Priorização de Ideias.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Ideias** | **B** | **A** | **S** | **I** | **C** | **O** | **Somatório** | **Priorização** |
| Refatorar o serviço de faturamento para rodar em container | 5 | 5 | 1 | 4 | 5 | 5 | 25 | 1 |
| Fazer o novo microsserviço reagir a novas ordens enviadas para faturamento | 4 | 5 | 5 | 1 | 3 | 5 | 23 | 2 |
| Criar um sistema de retry | 4 | 5 | 3 | 3 | 5 | 5 | 25 | 4 |
| Adicionar um sistema de para informar o vendedor do status do faturamento | 5 | 3 | 5 | 2 | 5 | 5 | 25 | 3 |

## 1.2 Solução

### 1.2.1 Objetivo SMART

Usar o C4 model para refatorar um sistema de faturamento que não atende as necessidades atuais da empresa e que foi desenvolvido usando um estrutura que não facilita a manutenção, não tem recuperação de falhas e que constantemente necessita o suporte de T.I para destravar uma operação.

O objetivo é em dois meses obter um desenho de um sistema que se adeque as novas necessidades da empresa, implementando boas praticas de projeto para garantir que no futuro não seja necessário um esforço grande para manter ou atualizar seus processos.

### 1.2.2 Premissas e Restrições

**Premissas**:

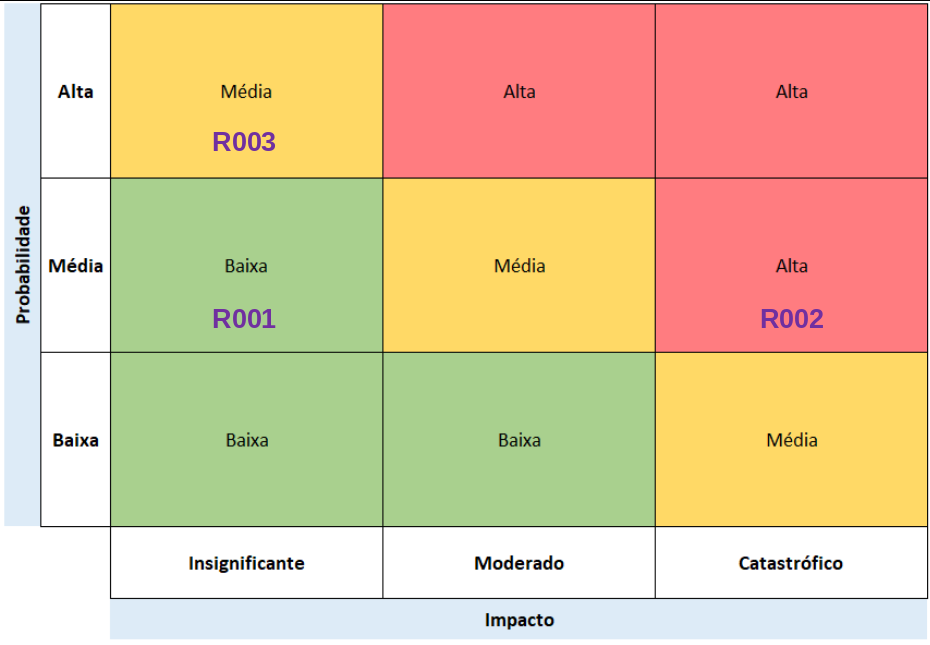
* A refatoração será em etapas para garantir uma transição tranquila
* Time de desenvolvimento terá essa refatoração como prioridade
* Projetos com acesso ao código fonte serão modificados conforme necessidade

**Restrições**

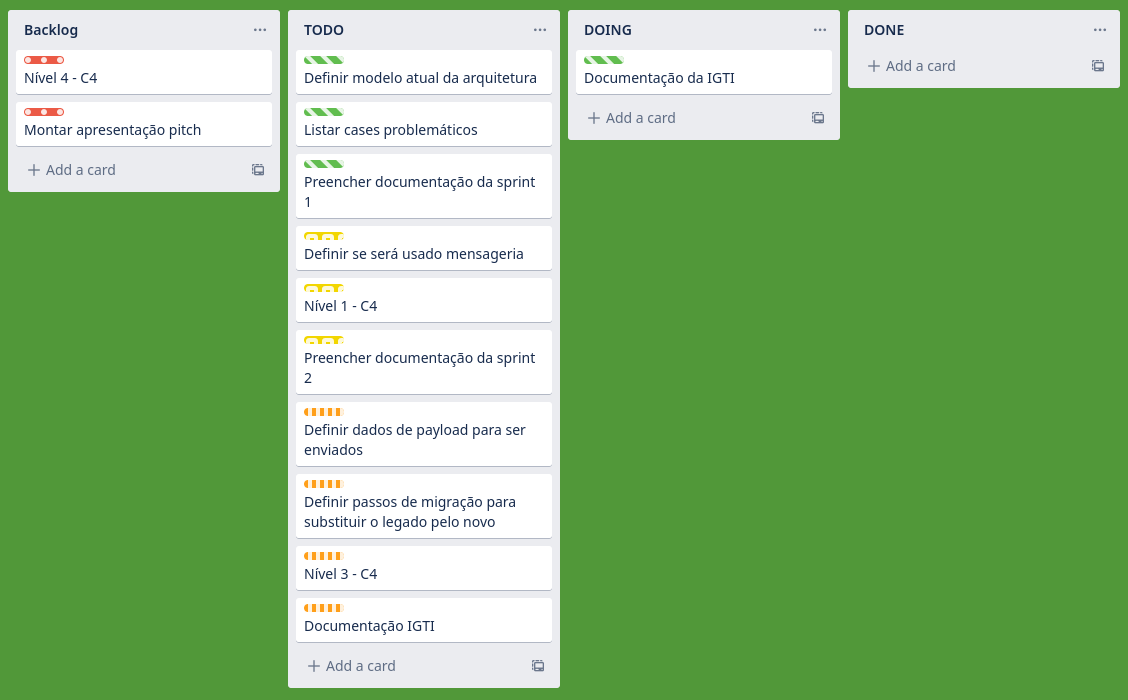
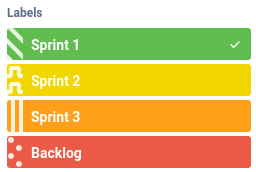
* Sistema de faturamento não pode parar em momento algum para evitar perda de vendas
* Projeto usará bibliotecas concedidas pelo programa fiscal para garantir que o processo seja executado corretamente
* Novo projeto deve ser pensado para ser executado com a infraestrutura já existentes da empresa

**Riscos do projeto:**

* **R001** – Risco de ataque ao expor o serviço fora do escritório
* **R002** – Inconsistência entre processo legado e o novo
* **R003** – Não ser possível implementar o status em 2 meses
* Matriz de Riscos.



### 1.2.3 Backlog de Produto



# 2. Área de Experimentação

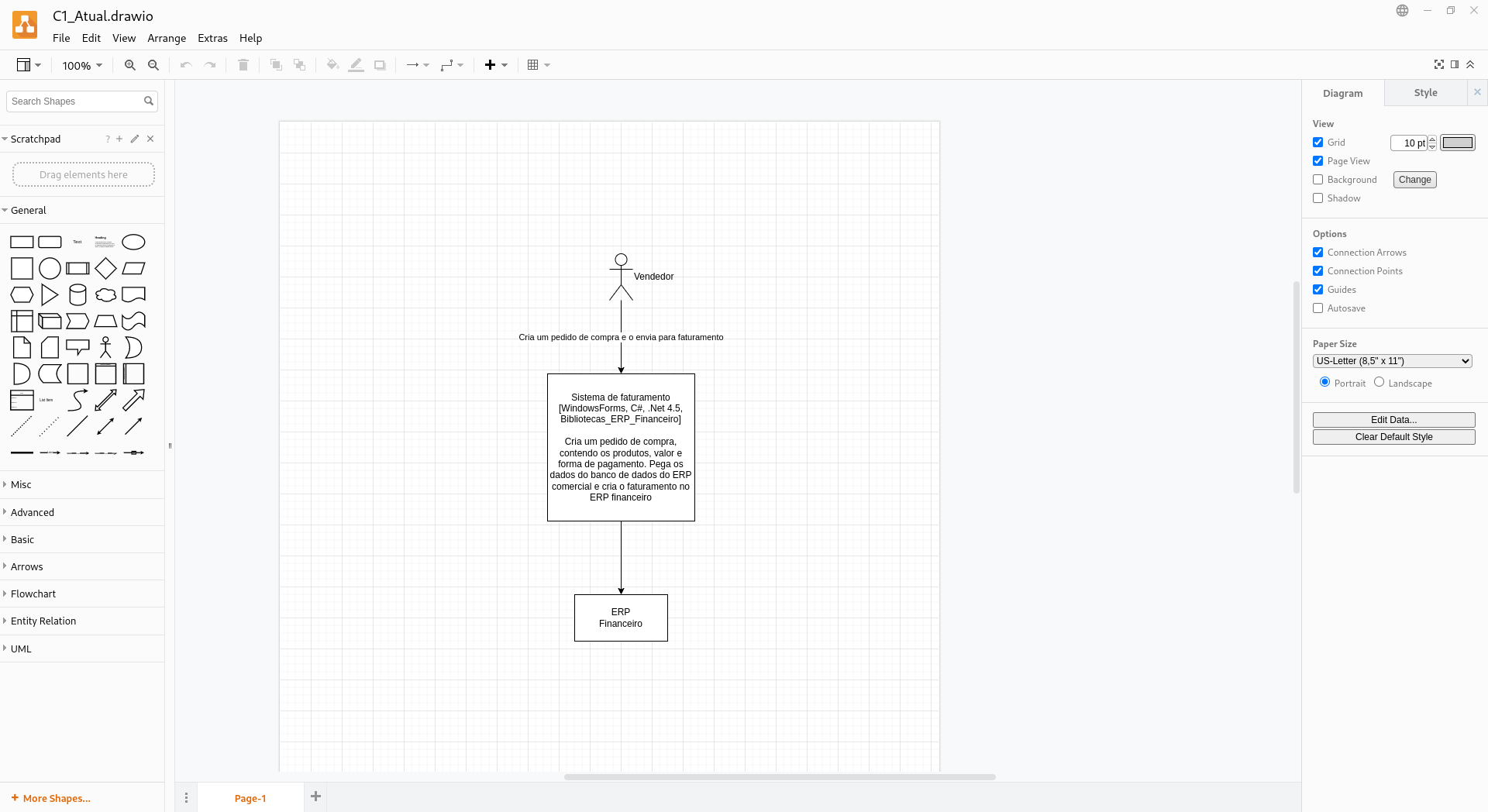
## 2.1 Sprint 1

### 2.1.1 Solução

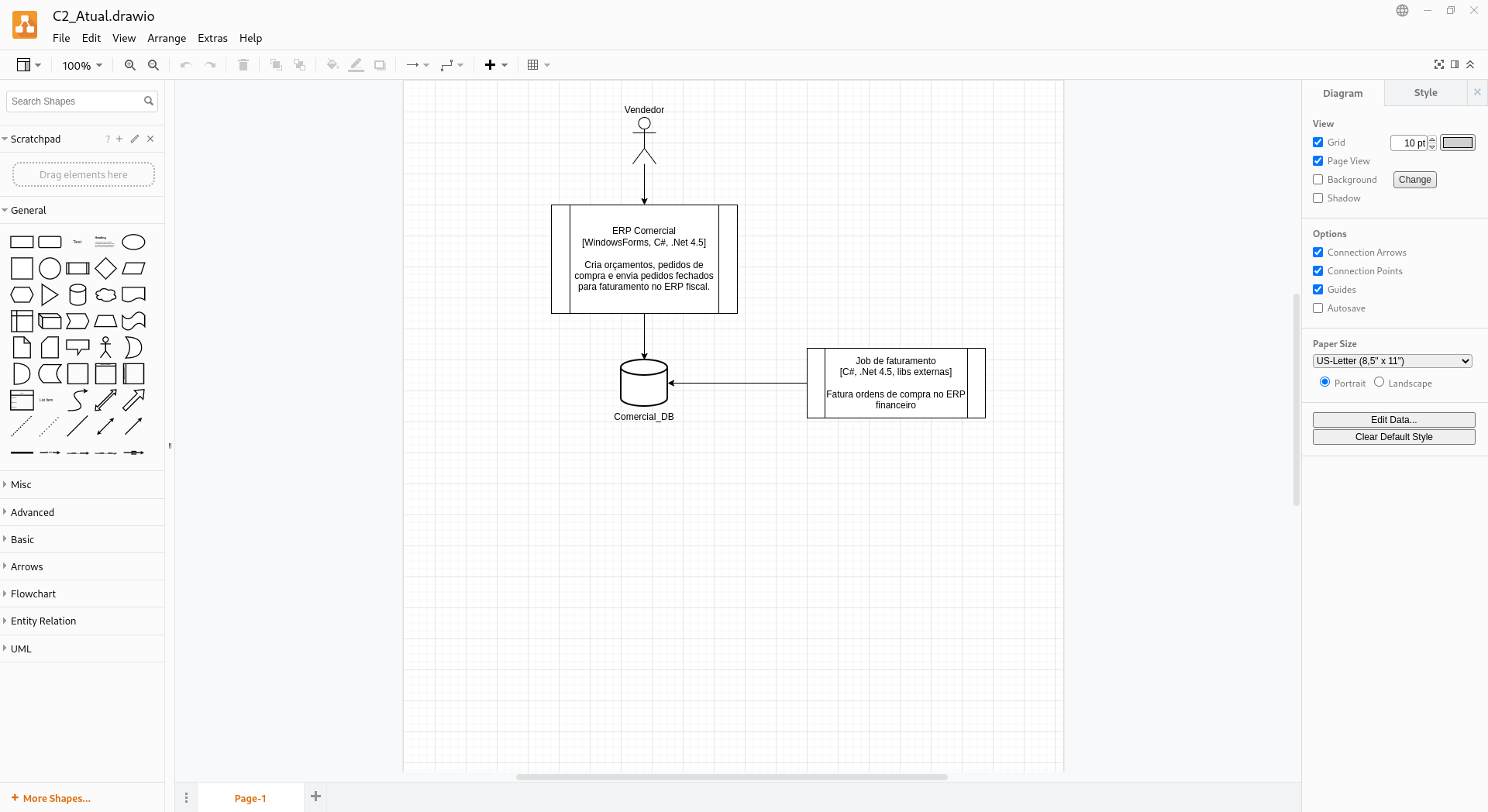
#### Evidência do planejamento:



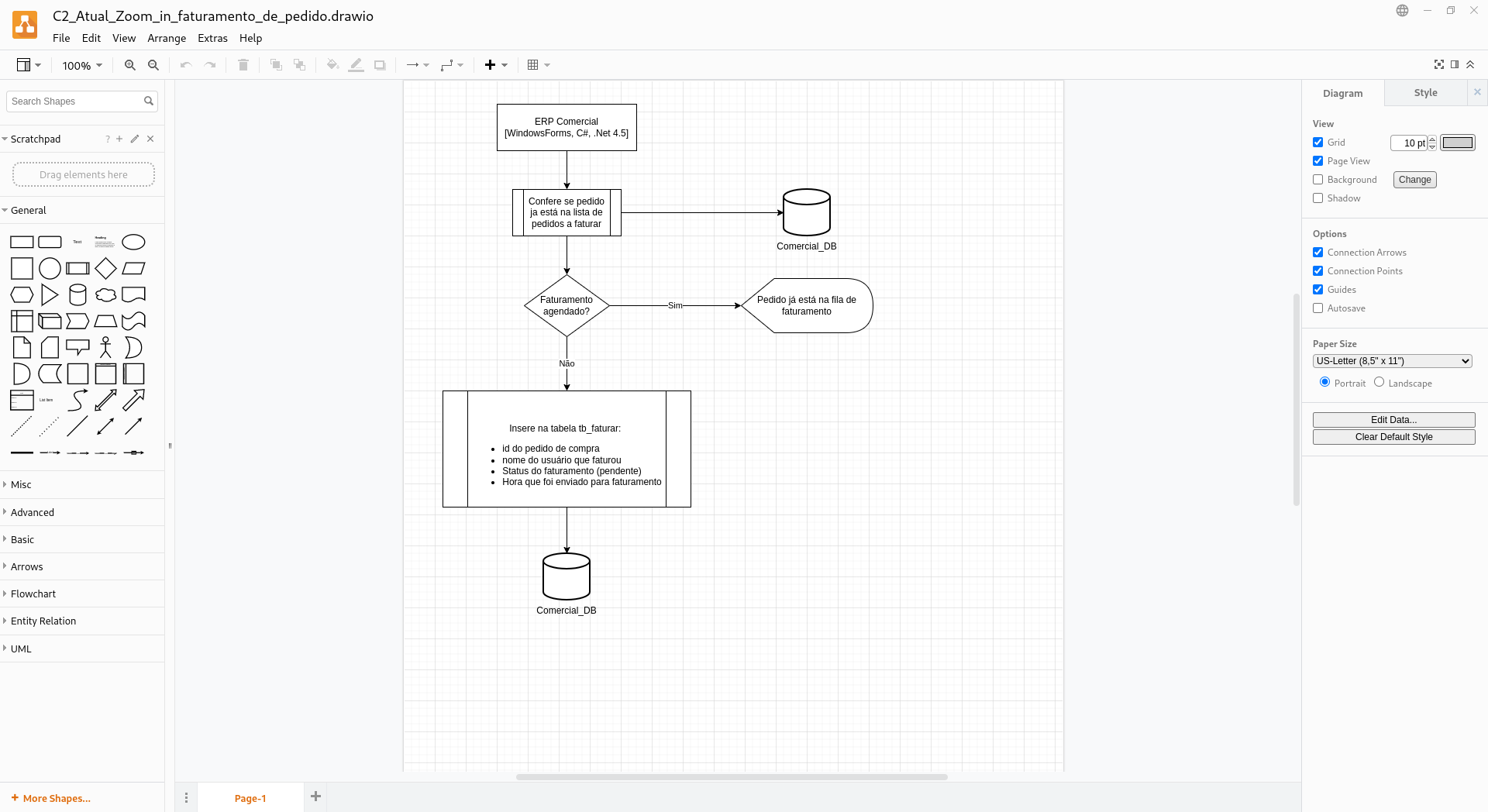
#### Evidência da execução de cada requisito:



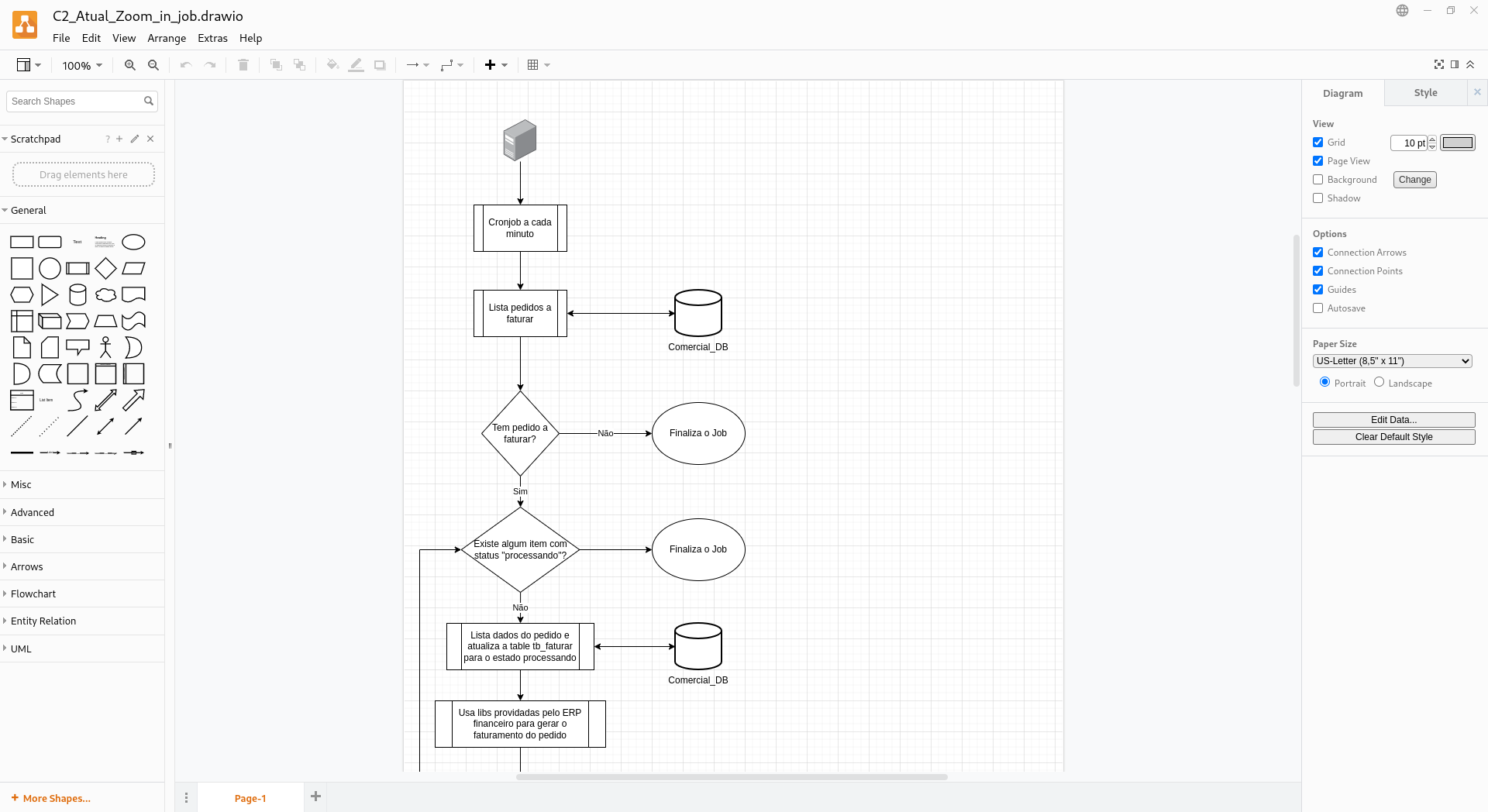
**Diagrama C4, nível 1 da arquitetura atual**. Disponível em <https://github.com/matheusfenolio/projeto-aplicado/blob/main/Sprint_1/C1_Atual.drawio>



**Diagrama C4, nível 2 da arquitetura atual**. Disponível em <https://github.com/matheusfenolio/projeto-aplicado/blob/main/Sprint_1/C2_Atual.drawio>



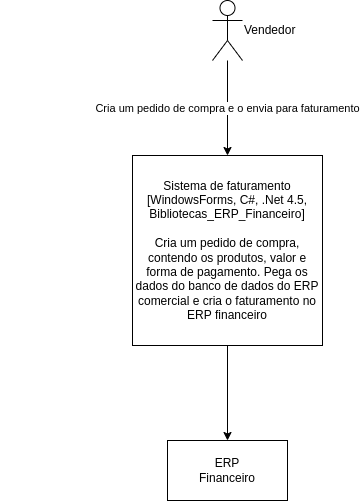
**Diagrama C4, nível 2 da arquitetura atual. Zoom in no faturamento de pedido**. Disponível em <https://github.com/matheusfenolio/projeto-aplicado/blob/main/Sprint_1/C2_Atual_Zoom_in_faturamento_de_pedido.drawio>



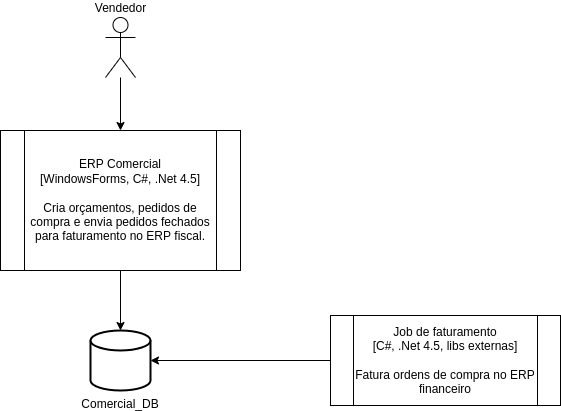
**Diagrama C4, nível 2 da arquitetura atual. Zoom in no job de faturamento**. Disponível em <https://github.com/matheusfenolio/projeto-aplicado/blob/main/Sprint_1/C2_Atual_Zoom_in_job.drawio>

#### Evidência dos resultados:

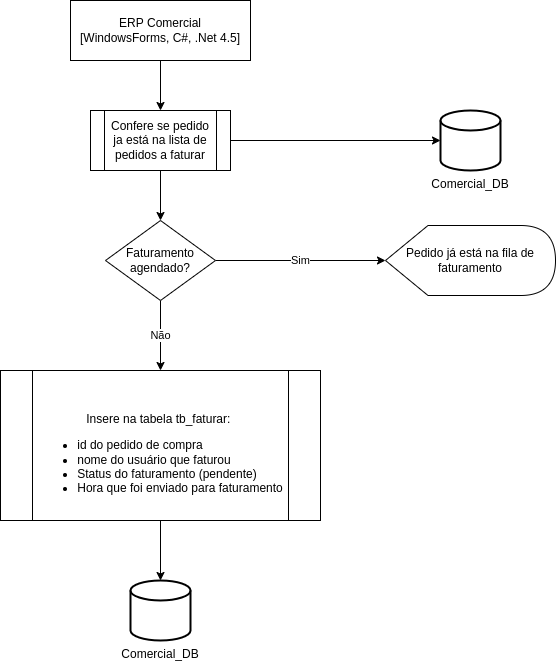
**Diagrama C1 da arquitetura atual:**



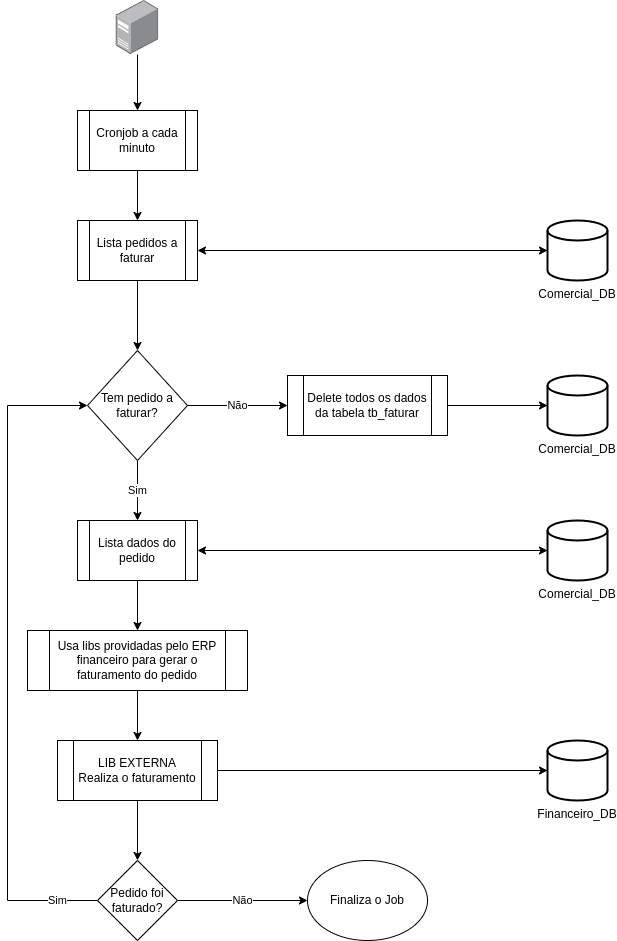
**Diagrama C2 da arquitetura atual:**



**Zoom in ERP Comercial:**



**Zoom in Job de faturamento:**



**Problemas apresentados:**

* **ERP Comercial**
* Vendedor não tem retorno se o faturamento deu certo
* Vendedores externos não podem realizar faturamento pois o processo funciona apenas localmente
* **Job de faturamento**
* Se um faturamento não tiver sucesso, ele bloqueia novas tentativas de faturamento do mesmo e de outros pedidos
* Processo de faturamento depende de um Cronjob para iniciar
* Não exite Logs para investigação de erros

### 2.1.2 Experiências vivenciadas

* Por nunca ter utilizado o C4 model para analisar um problema, me surpreendi como ele pode detalhar e isolar níveis para que possamos focar em problemas específicos, facilitando a visualização para refatoração em etapas, sem afetar o processo existente.

Com isso consegui pensar e formas mais eficientes do sistema, bem como encontrar falhas e cenários não premeditados.

* Definir quais módulos fazem parte de uma operação que logo após terá um "zoom in" foi um pouco difícil, pois sempre analisei esse problema de uma forma sequencial, desde seu inicio até o fim.

Com esse nível de detalhamento ficou mais fácil de atacar e explicar para os responsáveis na empresa qual era o problema e como isso estava afetando os processos atuais.

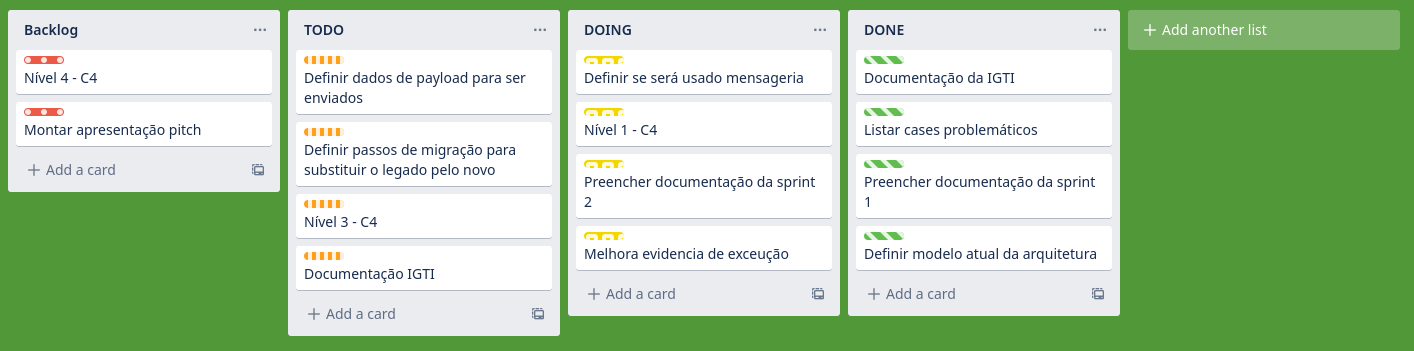
* Eu comecei a usar uma ferramente de script para criar os diagramas, chamada Mermaid porém ao final, ela gerou gráficos muito confusos, por isso optei por utilizar o Draw.io. Por ser uma ferramenta gratuita e de fácil utilização.

Eu pretendo fazer alguns testes com outras ferramentas, buscando uma melhor apresentação, assim seguindo a sugestão do C4 de usar um diagrama por códigos ao invés de desenha-lo. Creio que será uma boa experiencia para próxima fase desse projeto.

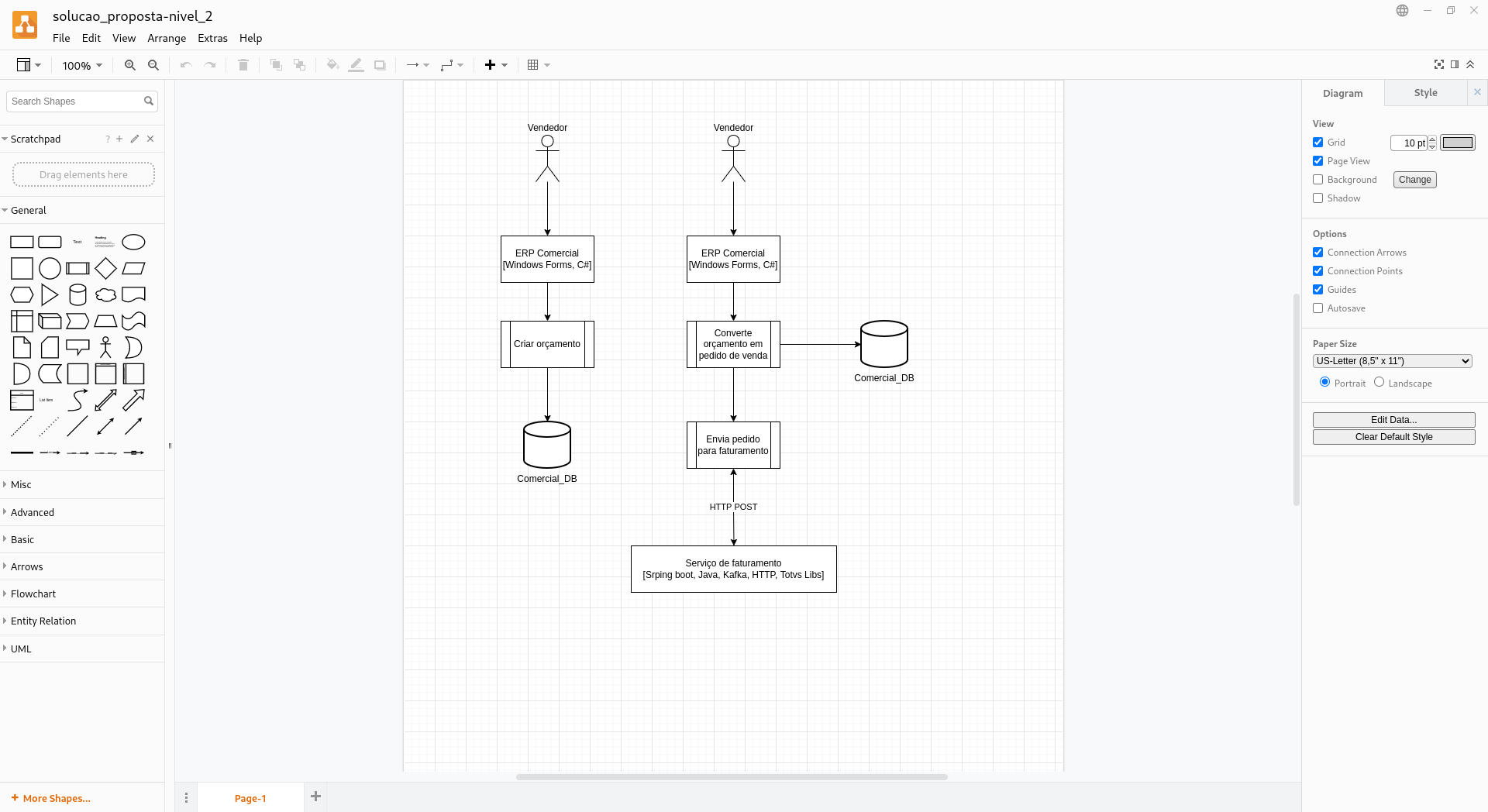
## 2.2 Sprint 2

### 2.2.1 Solução

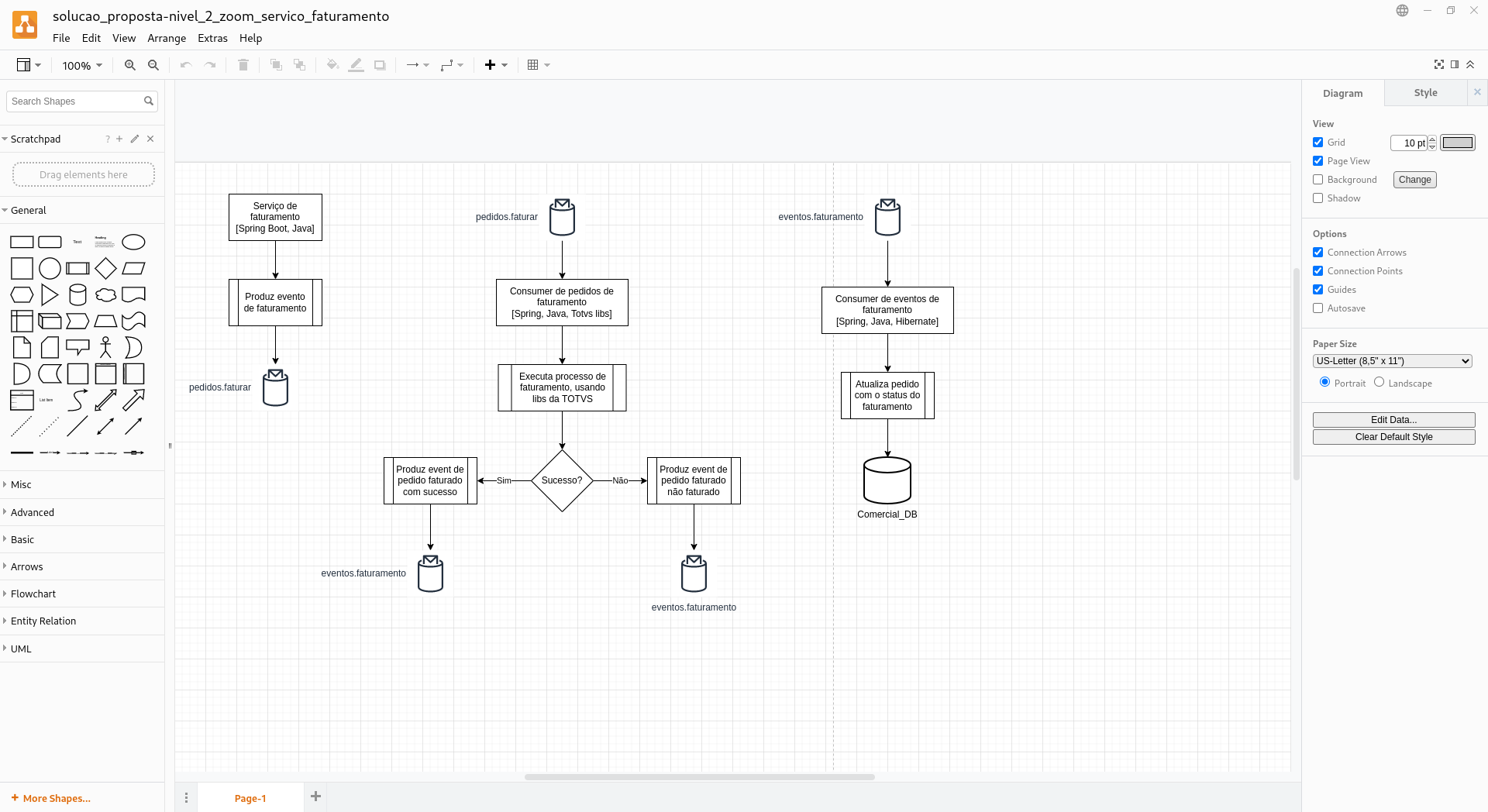
#### Evidência do planejamento:



#### Evidência da execução de cada requisito:



**Diagrama C4, nível 2 da arquitetura proposta**. Disponível em <https://github.com/matheusfenolio/projeto-aplicado/blob/main/sprint_2/diagramas/solucao_proposta-nivel_2>

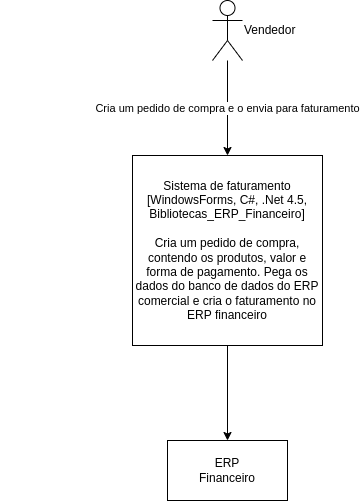


**Diagrama C4, nível 2 da arquitetura proposta. Zoom in serviço de faturamento**. Disponível em <https://github.com/matheusfenolio/projeto-aplicado/blob/main/sprint_2/diagramas/solucao_proposta-nivel_2_zoom_servico_faturamento>

#### Evidência dos resultados:

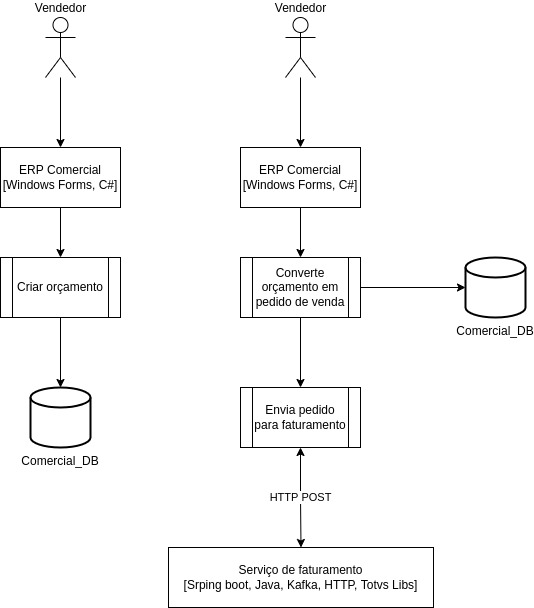
**Solução proposta**

**C4 – Nível 1**



Nesse nível podemos ver que a proposta do faturamento é fazer com que o pedido feito no ERP Comercial chegue até o ERP Financeiro para realizar a cobraça do cliente. Nessa visão não teremos alterações para a nova versão proposta.

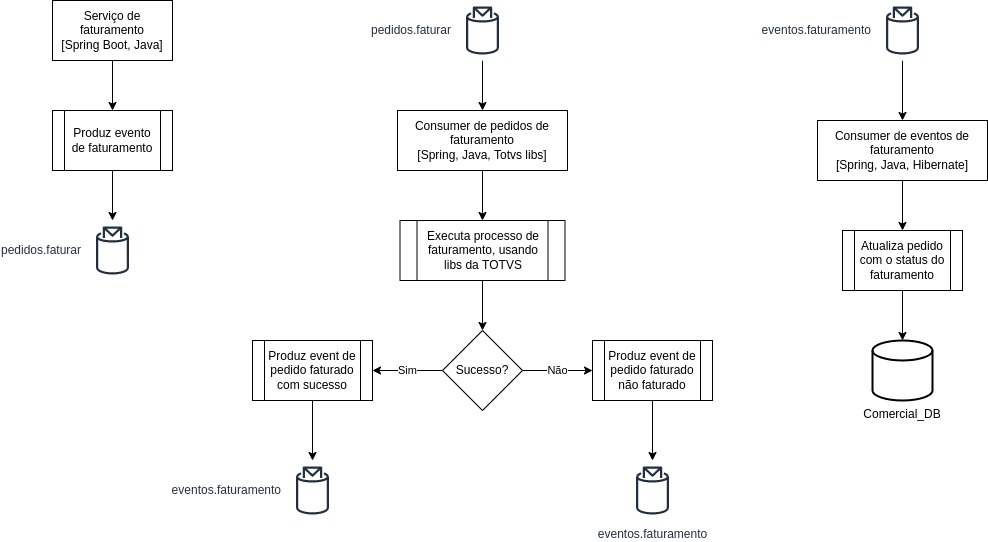
**C4 – Nível 2**



Dando um zoom no ERP comercial, nosso vendedor vai criar um orçamento para o cliente e vai salvar esse orçamento na banco de dados do setor de vendas.

Após o cliente aprovar o pedido, o vendedor vai converter esse orçamento para pedido e logo em seguida vai enviar o pedido para ser faturado. Esse processo de faturamento será mostrado nos próximo diagramas.

**Zoom serviço de faturamento**



Pela solução proposta, o pedido precisa passar por três serviços diferentes.

O primeiro serviço é responsável por receber uma requisição HTTP do ERP Comercial e produzir essa mensagem em um tópico de mensageiria.

O Segundo serviço vai ouvir as mensagens produzidas pelo serviço anterior e sempre que uma nova mensagem chegar, ele vai reagir e realizar o processo de faturamento. O processo de faturamento é executado através de bibliotecas disponibilizadas pelo ERP Financeiro. Com as bibliotecas o serviço dois vai processar os dados do ERP Comercial e gerar tanto a nota fiscal quando os métodos de pagamento no ERP Financeiro. Ao final ele produz uma mensagem de status em outro tópico do Kafka.

Por final o terceiro serviço vai receber as mensagens de status do serviço dois e realizar uma atualização no banco de dados do comercial para informar o status de faturamento do pedido.

**Solução proposta**

Essa solução tem como objetivo reduzir os problemas de ordens travadas, demora na execução dos processos e possibilitar o uso da funcionalidade para usuários externos através do uso do protocolo HTTP. Além da separação das responsabilidades em microsserviços, possibilitando o uso de contêineres.

Apesar do aumento da complexidade e custo da infraestrutura, esse novo processo vai possibilitar um serviço mais dinâmico e fluido, garantindo que os pedidos sejam processados na hora em que forem requisitados.

Outras vantagens com realação a versão antiga é que usando mensageiria, não corremos o risco de perder informações. Caso um dos serviços pare de funcionar, ele vai ler os tópicos novamente assim que o serviço for reestabelecido.

### 2.2.2 Experiências vivenciadas

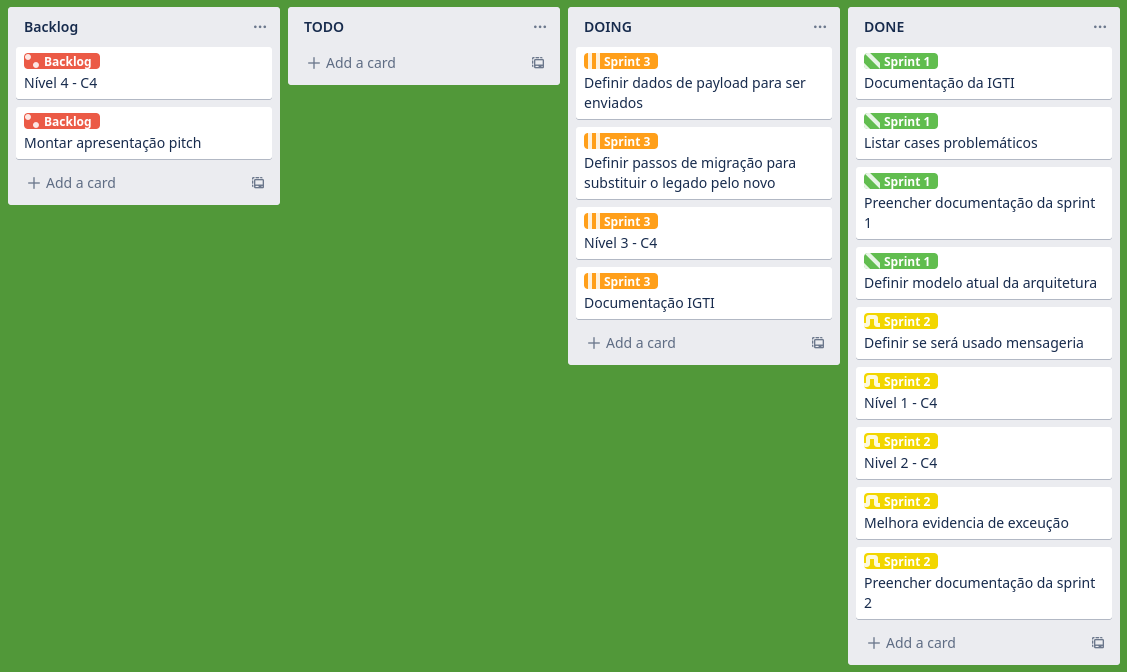
Gostei muito da facilidade que o diagrama C4 nós da para a visualização do problema. Ele nos faz ver de pedaço em pedação, evitando uma grande confusão quando se tenta ver o processo como um todo.

Como a ideia era refatorar uma das funções do processo, um dos diagramas foi reaproveitado, já que não era necessário mudar nenhum outro já existente, fora do processo de de faturamento.

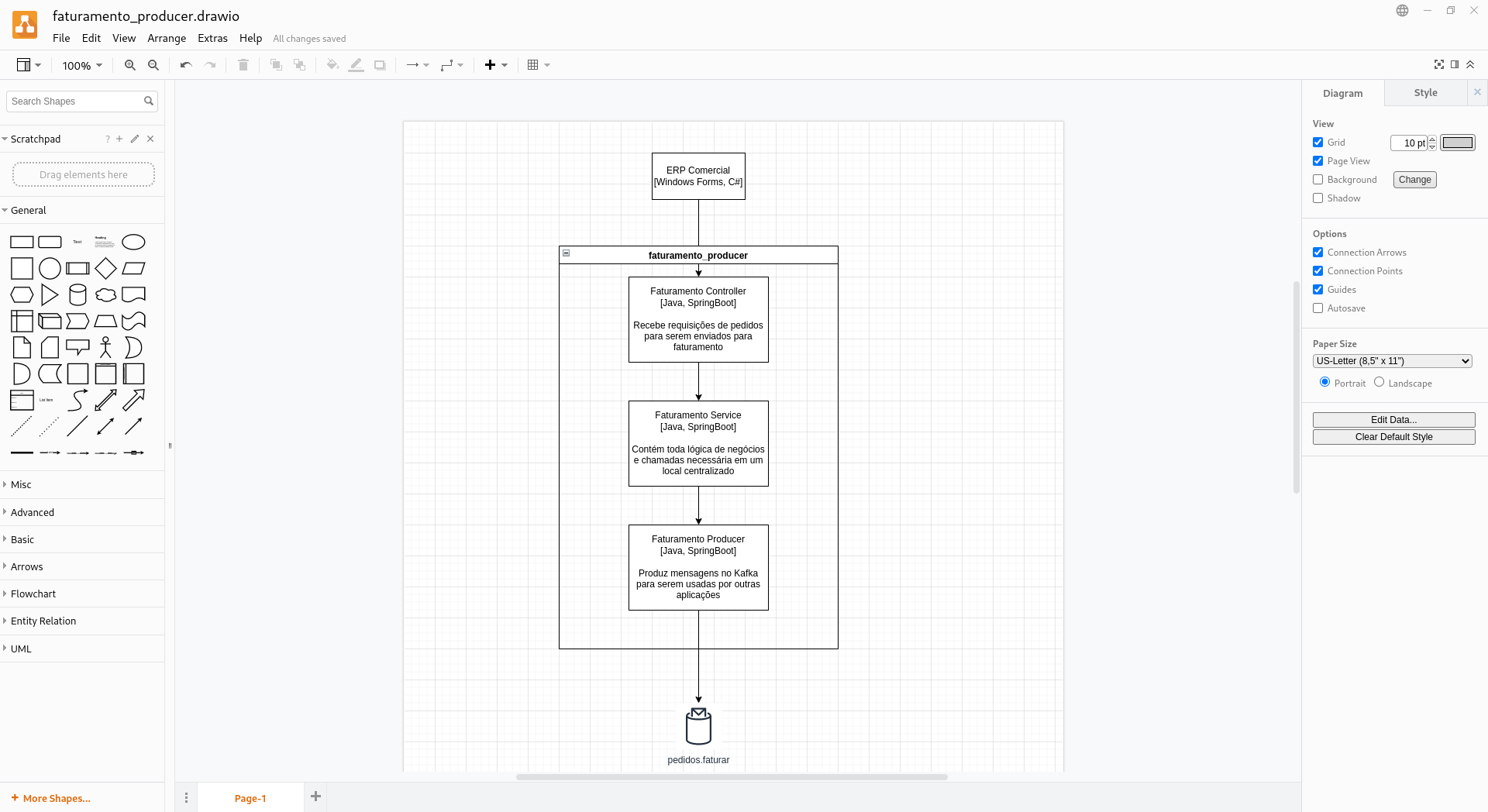
## 2.3 Sprint 3

### 2.3.1 Solução

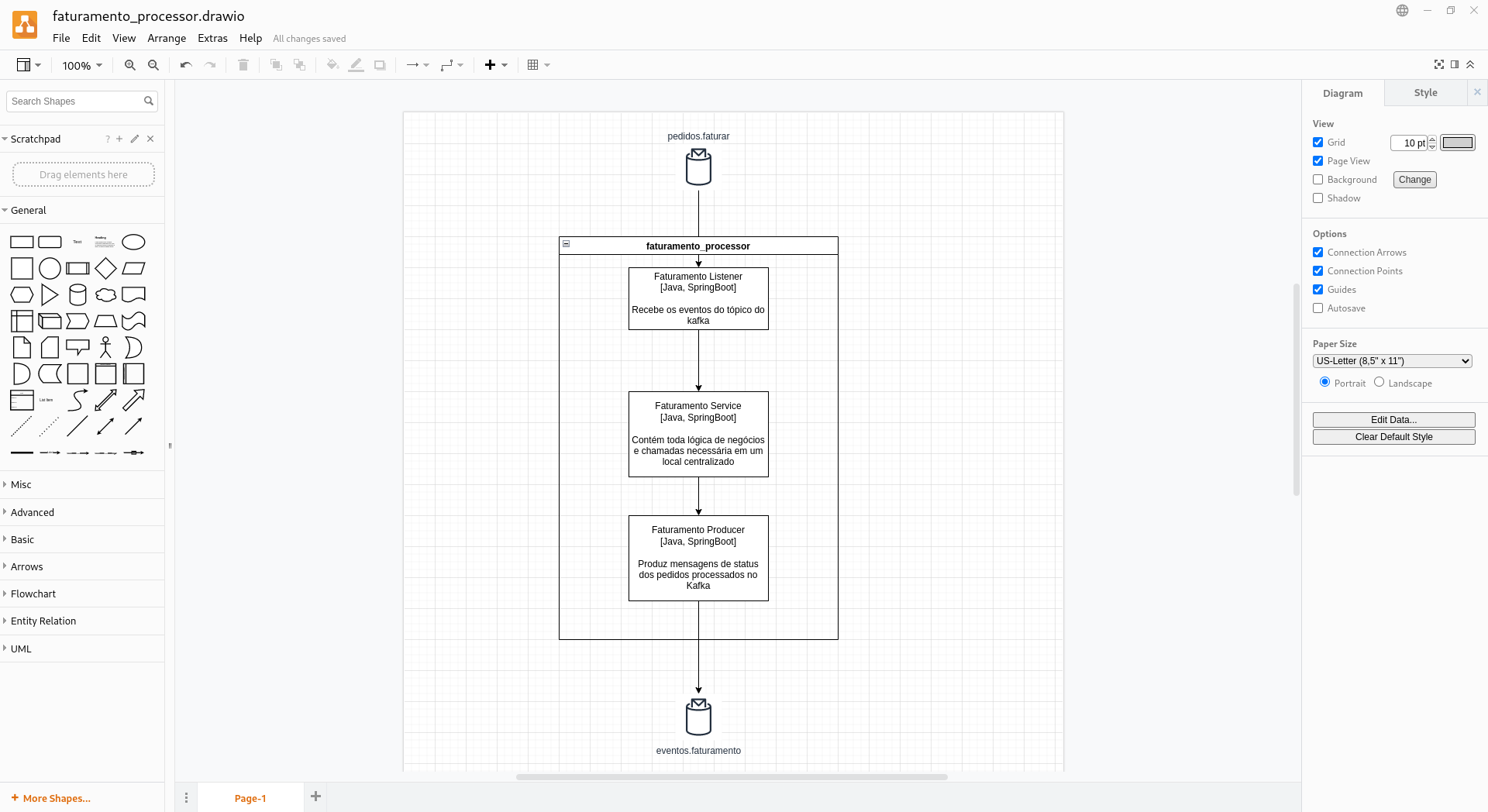
#### Evidência do planejamento:



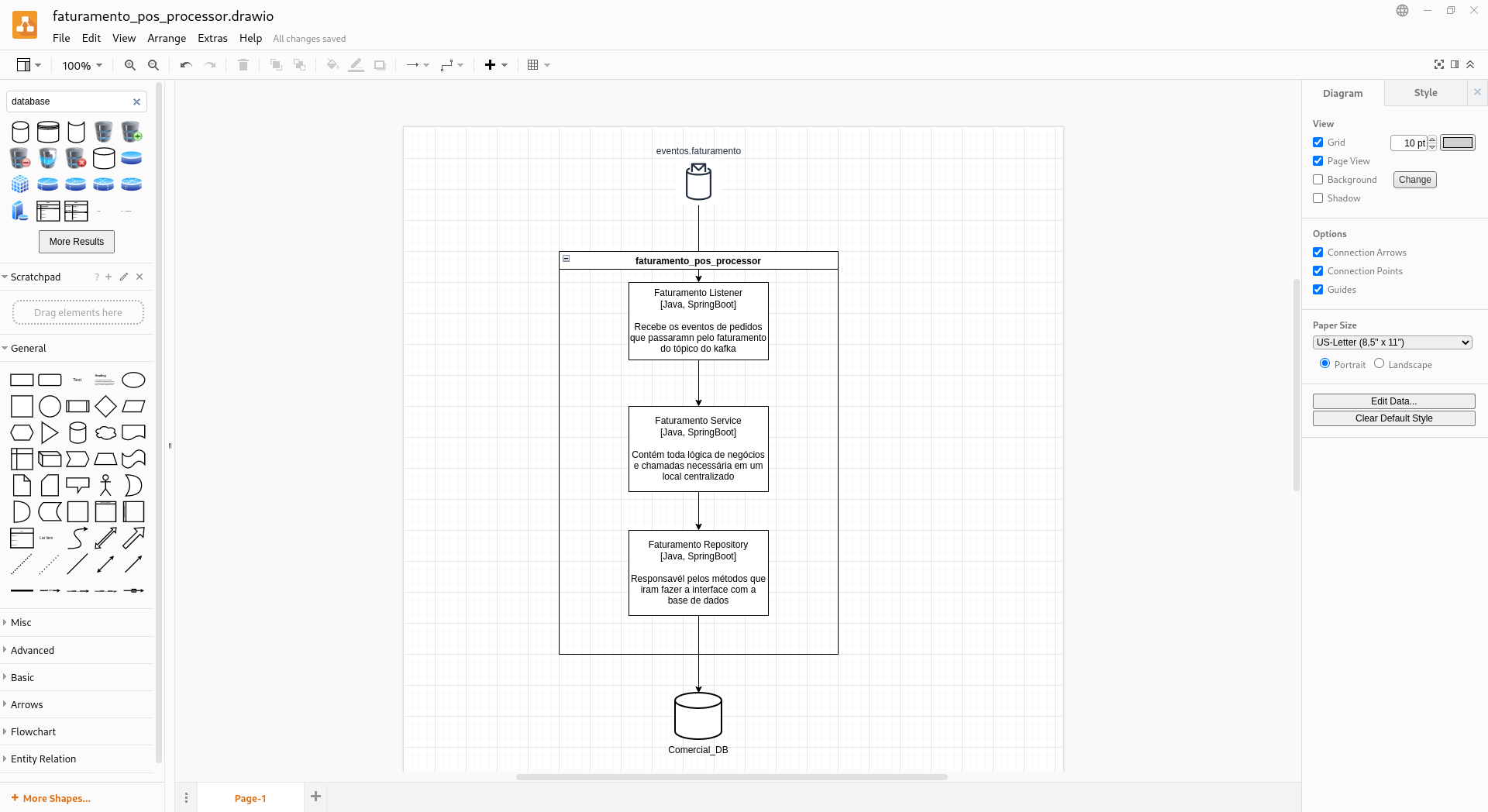
#### Evidência da execução de cada requisito:



**Diagrama C4, nível 3 da arquitetura proposta. Faturamento Producer**. Disponível em [https://github.com/matheusfenolio/projeto-aplicado/blob/main/sprint\_3/diagramas](https://github.com/matheusfenolio/projeto-aplicado/blob/main/sprint_2/diagramas/solucao_proposta-nivel_2_zoom_servico_faturamento)



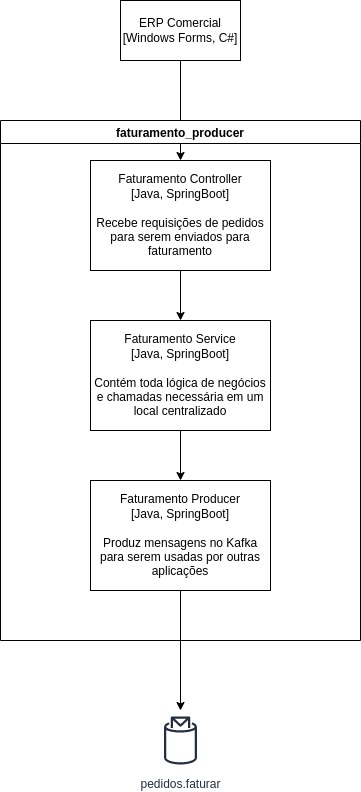
**Diagrama C4, nível 3 da arquitetura proposta. Faturamento Processor**. Disponível em [https://github.com/matheusfenolio/projeto-aplicado/blob/main/sprint\_3/diagramas](https://github.com/matheusfenolio/projeto-aplicado/blob/main/sprint_2/diagramas/solucao_proposta-nivel_2_zoom_servico_faturamento)



**Diagrama C4, nível 3 da arquitetura proposta. Faturamento pos Processor**. Disponível em [https://github.com/matheusfenolio/projeto-aplicado/blob/main/sprint\_3/diagramas](https://github.com/matheusfenolio/projeto-aplicado/blob/main/sprint_2/diagramas/solucao_proposta-nivel_2_zoom_servico_faturamento)

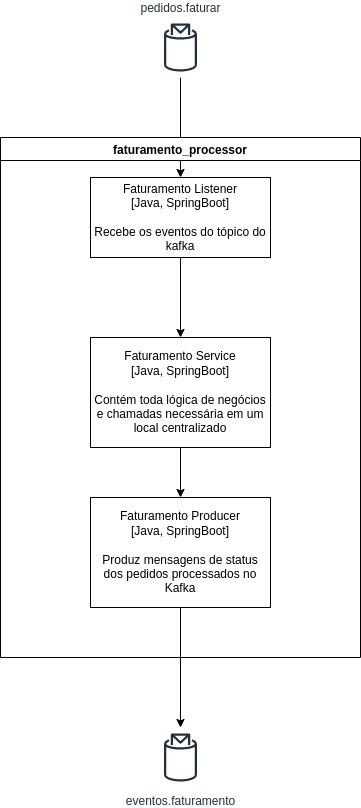
#### Evidência dos resultados:

**C4 Level 3 Faturamento Producer - Component**



Seguindo o nível três do diagrama C4 para o serviço faturamento\_producer, ele é responsável por receber todas as requisições HTTP do ERP Comercial, validar os dados, executar alguma lógica caso necessário e por fim produzir a mensagem no tópico do Kafka para ser processada.

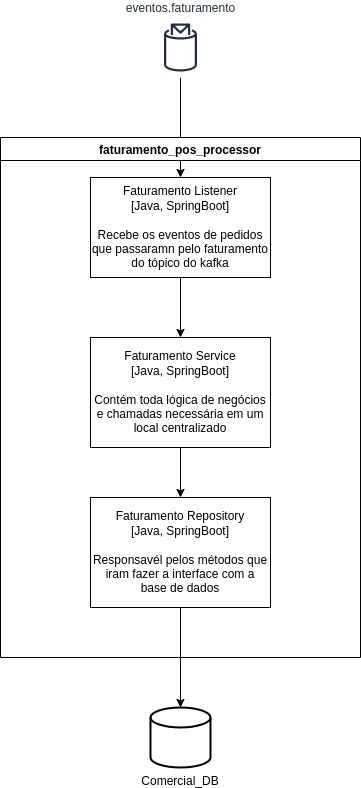
**C4 Level 3 Faturamento Processor - Component**



O serviço de faturamento\_processor é onde temos a lógica do faturamento em si. Ele reage a mensagens produzidas pelo faturamento\_producer e usando as bibliotecas de integração disponibilizadas pelo ERP Financeiro, o serviço cria os dados necessários para realizar emissão de nota fiscal entre outros.

Mais tarde, ele produz uma nova mensagem no tópico de status do faturamento, para que o pedido possa ser atualizado com o resultado desse processamento.

**C4 Level 3 Faturamento pos Processor - Component**



Ao final, o serviço faturamento\_pos\_processor vai captar as mensagens produzidas pelo faturamento\_processor e atualizar o banco de dados do comercial com o status do faturamento do pedido.

Nesse processo novas funções podem ser implementadas, como um gatilho para uma notificação na tela do usuário que o informaria da finalização do processo de faturamento. Porém para essa etapa vamos focar unicamente em dar um retorno no banco de dados.

### 

**Processo de refatoração**

A refatoração do processo atual vai ser rápida, porém ela é muito interdependente, não sendo possível trocar os módulos a parte.

Com o processo de faturamento é basicamente um processo linear, todos os serviços precisam estar em funcionamento para garantir que o processo antigo seja desativado sem afetar o comercial.

Abaixo, seguem os passos a serem seguidos para garantir a validação e ao mesmo tempo uma troca dos processos sem afetar negativamente as operações da empresa.

1. Criar o microsserviço de producer e conecta-lo ao Kafka.

2. Fazer o ERP atual, chamar o producer e garantir que as mensagens estão chegando ao tópico. Porém continuar com o processo antigo.

3. Criar o microsserviço que processa as ordens e conecta-lo para ouvir eventos do Kafka

4. Desativar o processo antigo, tirando a tabela onde os pedidos eram inseridos e desligando o serviço que era usado via Cronjob.

5. Criar o microsserviço que vai escutar os eventos de pedidos processados.

6. Plugar o microsserviço de pós processamento ao Kafka para escutar os eventos vindos do microsserviço de processamento.

### 2.3.2 Experiências vivenciadas

Com a estratégia do C4 Model me ajudou a ver de uma maneira mais simplificada a implementação, me permitindo isolar as responsabilidades e definir estratégias de substituição do processo antigo sem que haja impacto nas operações.

No final eu pude facilmente entender que por mais que a complexidade seja maior, essa solução pode priorizar serviços mais críticos e escalar apenas o módulos necessários.

Os diagramas também renderam boas ideias, fazendo que, com mais algumas etapas possam ser implementadas com melhorias que podem ser realizadas a partir da nova estrutura.

Um ponto interessante que não foi pensado é a segurança para requisição HTTP do primeiro serviço. Na primeira fase seria usado um já existente "basic auth" porém uma nova solução deve ser pensada para esse sistema.

Por final um ponto muito positivo é a quantidade de documentação que isso gera. Quando se trata do sistema antigo, nenhuma informação foi documentada, o que fazia a empresa dependente do programador que realizou o serviço desde o começo. Além disso isso, temos a possibilidade do rastreio de possíveis gargalos e problemas de segurança em caso de necessidade.

# 3. Considerações Finais

## 3.1 Resultados

Por meio desse projeto eu pude fazer a uma definição melhor do meu problema, podendo isolar as responsabilidades e definir uma estratégia de solução.

Com a primeira Sprint eu pude mostrar e detalhar o fluxo de como o processo antigo era problemático e onde eram seus gargalos. Com isso eu tive uma forma de analisar o processo como um todo e inclusive encontrar mais pontos de melhoria.

Já a segunda Sprint foi a mais produtiva, pois foi onde consegui propor uma solução para os problemas identificados anteriormente. Nessa fase eu também pude ter uma ideia melhor de qual seria o esforço para criar e migrar a plataforma, bem como o custo operacional dela.

Por fim na terceira Sprint eu pude me aprofundar mais ao que diz respeito a arquitetura dos serviços, isso me deu uma ajuda para separar as responsabilidades, bem como fazer eles terem um pacote menor de desenvolvimento, focando assim em um estratégia de microsserviços.

Tudo isso contribui e muito para minha carreira, esse projeto me ensinou a ver os cenários de maneira mais critica e dividida, bem como documentar isso de uma forma que qualquer outra pessoa possa entender e ver os ganhos. Posso destacar também o quanto fico feliz de finalmente ter uma solução para um problema que parece tão simples, porém nunca teve uma proposta de melhoria.

Para terminar, posso declarar que vejo um projeto concluído com sucesso, que ao mesmo tempo em que me ajudou a aplicar o conhecimento adquirido durante o curso, também pude resolver um problema da vida real, mostrando que a teoria bem aplicada torna a pratica mais fácil e objetiva.

## 3.2 Contribuições

A proposta do projeto sempre foi a completa substituição do serviço de faturamento de pedidos, porém sem que houvesse uma parada devida a migração, garantindo que fosse possível restaurar o processo antigo em caso de pane.

O uso de microsserviços trouxe uma enorme vantagem, pois agora é possível dividir os recursos para cada responsabilidade, sem que tivéssemos desperdício em processos menos críticos.

Com o Kafaka tivemos uma outra revolução, usando uma estrutura orientada a eventos, garantíamos que o pedido sempre teria seguimento em qualquer etapa da processo, mesmo que um dos serviços estivesse fora do ar. Além de que na arquitetura antiga, os pedidos só eram faturados de um em um minuto, o que acarretava não apenas em pedidos que demoravam para serem processados, mas bem como desperdício de recursos, fazendo a chamada do processo em momentos desnecessários.

Outro ponto a se destacar é a atualização das tecnologias. Com isso podemos implementar uma vasta gama de opções para melhorar o processo já existente, bem como a implementação de novos. Um que foi possível graças a nova arquitetura, foi a de faturamento por vendedores que estejam fora da empresa. Por focarmos em requisições HTTP, fica muito mais fácil e seguro expor esses serviços para a internet.

Por final, a arquitetura passa a ser mais confiável e performática, abrindo toda uma nova geração de processos que podem ser baseados nela para auxiliar as operações da empresa.

## 3.3 Próximos passos

Um ponto que seria ótimo, é o nível quatro do diagrama, garantindo assim a definição da estrutura do código para os microsserviços. Com esse diagrama poderíamos ter efeitos iguais aos da arquitetura, podendo encontrar e ou prever problemas futuros. Outro ganho seria o de criar uma arquitetura mais flexível para futuras implementações nos processos.

Após essas definições, usar o Scrum para definir as tarefas necessárias para implementação. A priorização e detalhamento das tarefas se farão necessárias para que possamos priorizá-las e dividi-las.

Outro destaque vai para o sistema de autenticação. Nesse projeto eu não abordei esse tópico, pois ele vai se basear em uma forma de autenticação já existente, porém ultrapassada. Um novo projeto apenas para refatorar esse processo de autenticação seria necessário, assim poderíamos garantir não apenas o aumento da segurança, como a padronização para outros serviços que se beneficiariam de tal recurso.