

MTSDS

Francisco Oliveira - 8210093

January 2026

Conteúdo

1	Contexto	3
2	Requisitos do Sistema	4
2.1	Requisitos Funcionais	4
2.2	Requisitos Não-Funcionais	4
2.3	Modelação de Domínios e Bounded Contexts	5
2.4	Classificação dos Subdomínios	5
2.5	Definição dos Bounded Contexts	5
3	Microserviços e Endpoints	6
3.1	Customers	7
3.2	Products	8
3.3	Orders	9
3.4	Inventory	11
3.5	Planning	12
3.6	Production	13
3.7	Machines	15
3.8	Notifications	17
4	Conclusão	19

Lista de Figuras

1	Diagrama da Arquitetura	7
2	Microserviço Customers	7
3	Microserviço Products	8
4	Diagrama de Sequência para a Criação de um Produto	9
5	Microserviço Orders	9
6	Criação de uma Encomenda	10
7	Cancelamento de uma Encomenda	10
8	Diagrama de Classes Orders	11
9	Microserviço Inventory	11

10	Microserviço Planning	12
11	Criação de um Plano	12
12	Diagrama de Classes Planning	13
13	Microserviço Production	14
14	Diagrama de Sequência de Produção	15
15	Diagrama de Classes do Production Service	15
16	Machines Service	16
17	Diagrama de Sequência Machines	16
18	Diagrama Classes Machines	17
19	Notifications Service	18
20	Diagrama Sequência Notifications	19
21	Diagrama Classes Notifications	19

1 Contexto

A empresa *Fábrica 4.0* atua no setor da fabricação industrial e da gestão de informação associada aos processos produtivos há mais de uma década. Numa fase inicial, toda a sua operação encontrava-se centralizada num único sistema monolítico, desenvolvido para abranger as funcionalidades essenciais do negócio industrial, nomeadamente a gestão de clientes, produtos, encomendas, inventário, planeamento da produção, execução no chão de fábrica e geração de relatórios operacionais.

Inicialmente, esta abordagem revelou-se adequada, oferecendo uma solução integrada e relativamente simples de manter. Contudo, à medida que a empresa cresceu e os processos de fabricação se tornaram mais complexos, o sistema começou a evidenciar limitações significativas. O aumento do volume de encomendas, a necessidade de um planeamento mais dinâmico e a integração com máquinas industriais tornaram o sistema monolítico lento, rígido e difícil de evoluir.

Com o tempo, pequenas alterações nos processos produtivos passaram a exigir ciclos longos de desenvolvimento e testes, atrasando a introdução de melhorias. Adicionalmente, falhas num único módulo podiam comprometer todo o sistema, afetando diretamente a produção. Esta situação revelou-se particularmente crítica em ambientes industriais, onde a indisponibilidade de informação em tempo real pode provocar atrasos, desperdício de recursos e perdas financeiras.

Perante este cenário, a direção da *Fábrica 4.0* decidiu adotar uma nova estratégia tecnológica, baseada na migração para uma arquitetura de microserviços. O principal objetivo foi decompor o sistema existente em serviços menores, independentes e alinhados com domínios específicos do negócio industrial, tais como encomendas, inventário, planeamento, execução, máquinas e controlo de qualidade.

O processo de migração iniciou-se com o mapeamento dos principais fluxos produtivos e operacionais, identificando os domínios com maior carga e criticidade. A separação entre o planeamento e a execução da produção, bem como a gestão autónoma de máquinas e inventário, tornou-se essencial para garantir que operações intensivas não afetassem outros componentes do sistema. Cada microserviço passou a possuir a sua própria base de dados e a comunicar com os restantes de forma controlada, recorrendo tanto a chamadas diretas como a eventos assíncronos.

Esta transição para uma arquitetura de microserviços representou igualmente uma mudança cultural dentro da organização. As equipas passaram a trabalhar de forma mais autónoma, assumindo maior responsabilidade sobre os serviços que desenvolvem e mantêm. Simultaneamente, a nova arquitetura criou as bases para uma fábrica mais digital, flexível e alinhada com os princípios da Indústria 4.0, permitindo à *Fábrica 4.0* crescer de forma sustentável e responder de forma eficaz às exigências futuras do setor industrial.

2 Requisitos do Sistema

O projeto “Industry 4.0” surge da necessidade de modernizar a infraestrutura tecnológica da fábrica inteligente. O sistema monolítico anterior apresentava problemas críticos de escalabilidade e resiliência, com elevada latência na execução de encomendas de produção e dificuldade em integrar dados de máquinas em tempo real.

A nova solução foi desenhada sobre uma arquitetura de **Microserviços**, visando a descentralização de dados, tolerância a falhas e escalabilidade independente.

2.1 Requisitos Funcionais

O sistema foi decomposto em domínios funcionais autónomos, garantindo que cada microserviço resolve um problema de negócio específico:

- **Gestão de encomendas (Orders):**
 - Permitir a criação, consulta e cancelamento de encomendas de produção.
 - Validação de stock e clientes antes da execução da encomenda.
 - Publicar eventos de encomenda criada ou cancelada para os serviços dependentes.
- **Planeamento de Produção (Planning):**
 - Receber encomendas e gerar planos de produção.
 - Atribuir máquinas e horários às encomendas.
 - Publicar planos prontos para execução.
- **Execução de Produção (Production):**
 - Transformar planos de produção em tarefas executáveis nas máquinas.
 - Suportar cancelamento e monitorização de jobs.
 - Reportar status para o serviço de notificações e sincronização de encomendas.
- **Notificações (Notifications):**
 - Enviar alertas a operadores e gestores sobre encomendas, cancelamentos e falhas.

2.2 Requisitos Não-Funcionais

Para garantir robustez num ambiente industrial, o sistema deve obedecer aos seguintes critérios:

- **Escalabilidade Independente (Kubernetes):** Microserviços críticos, como **Notifications** e **Planning**, devem poder ser escalados horizontalmente sem afetar outros serviços.
- **Desacoplamento e Assincronismo (RabbitMQ):** Eventos demorados, como envio de notificações ou atualização de histórico, devem ser processados de forma assíncrona..
- **Isolamento de Dados:** Cada microserviço possui a sua própria base de dados (PostgreSQL), evitando dependências ocultas.
- **Auditoria e Segurança:** Operações críticas, como cancelamento de encomendas, devem ser protegidas e auditadas.

2.3 Modelação de Domínios e Bounded Contexts

A decomposição do sistema segue a metodologia **Domain-Driven Design (DDD)**, garantindo que cada microserviço possui responsabilidade única e modelo de dados isolado, evitando acoplamento excessivo.

2.4 Classificação dos Subdomínios

Core Domain (Domínio Principal):

- **Gestão de Encomendas (Orders):** Domínio central, inclui criação, validação, cancelamento e publicação de eventos de encomendas.
- **Planeamento de Produção (Planning):** Garante a atribuição correta de máquinas e horários, evitando conflitos.

Supporting Subdomains (Domínios de Suporte):

- **Execução de Produção (Production):** Converte planos em jobs nas máquinas, suporta cancelamento e monitorização.
- **Notificações (Notifications):** Envia alertas, gere preferências de contacto e garante rastreabilidade.

Generic Subdomains (Domínios Genéricos):

- **Clientes e Inventário (CheckCustomer, CheckInventory):** Validação de existência de clientes e stock disponível.

2.5 Definição dos Bounded Contexts

Orders Context:

- Microserviço: `orders-service`
- Responsabilidade: Gestão do ciclo de vida das encomendas, publicação de eventos para os outros serviços.

- Linguagem Ubíqua: “Created”, “Cancelled”, “Validated”.

Planning Context:

- Microserviço: `planning-service`
- Responsabilidade: Planeamento automático da produção, atribuição de máquinas e horários.
- Publica planos prontos para execução.

Production Context:

- Microserviço: `production-service`
- Responsabilidade: Executar tarefas de produção, monitorizar jobs e suportar cancelamentos.

Notifications Context:

- Microserviço: `notifications-service`
- Responsabilidade: Envio de alertas, gestão de preferências, auditoria de mensagens.

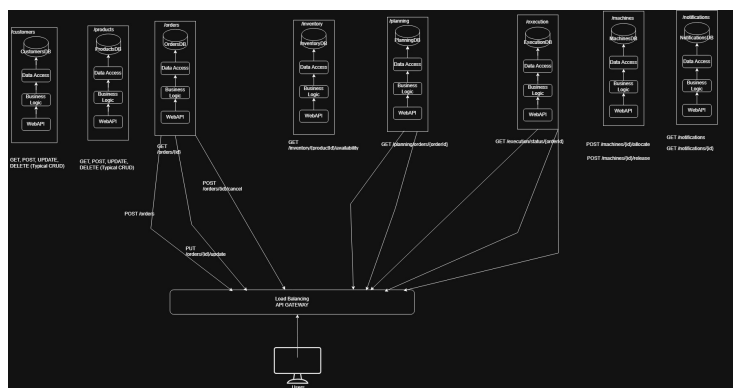
Validation Context:

- Microserviços: `check-customer`, `check-inventory`
- Responsabilidade: Verificação síncrona de clientes e stock antes da criação de encomendas.

3 Microserviços e Endpoints

A adoção de uma arquitetura baseada em microserviços implicou a decomposição do sistema monolítico da *Fábrica 4.0* em serviços independentes, cada um responsável por um domínio específico do negócio. Esta abordagem permitiu uma melhor separação de responsabilidades, maior escalabilidade e facilidade de manutenção, bem como a possibilidade de evolução independente de cada componente.

Cada microserviço disponibiliza um conjunto de *endpoints* bem definidos, normalmente expostos através de interfaces REST, permitindo a comunicação controlada entre serviços e com aplicações externas. Nas subsecções seguintes são apresentados os principais microserviços identificados no sistema, bem como as suas responsabilidades gerais.



3.1 Customers

O microserviço *Customers* é responsável pela gestão da informação relativa aos clientes da empresa. Este serviço centraliza dados como identificação, contactos, histórico de interações e condições comerciais, garantindo que a informação do cliente é consistente e acessível aos restantes serviços do sistema.

Os seus *endpoints* permitem operações como criação, atualização, consulta e desativação de clientes, sendo utilizado principalmente pelo microserviço de encomendas para validar e associar clientes às respetivas encomendas.

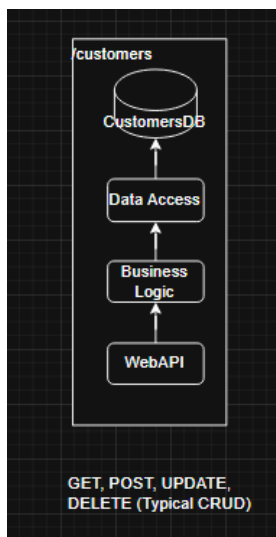


Figura 2: Microserviço Customers

3.2 Products

O microserviço *Products* gere o catálogo de produtos fabricados e comercializados pela empresa.

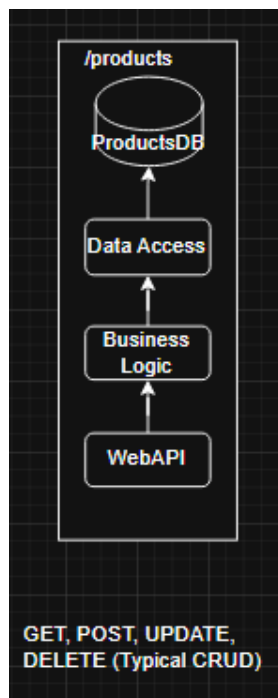


Figura 3: Microserviço Products

Este serviço fornece *endpoints* para consulta e manutenção do catálogo de produtos, sendo essencial para o correto funcionamento dos processos de encomendas, planeamento e produção.

Este microserviço comunica com o serviço Inventory, enviando uma mensagem assíncrona com o ID do Produto e a quantidade de Stock no momento de criação, o Inventory no futuro tratará da gestão de Stock interna da empresa, se por acaso um produto for atualizado uma mensagem é enviada de novo para o inventory com a nova quantidade de Stock, o serviço Products usa uma outbox para sincronizar o inventário com o serviço Inventory, de maneira a evitar dados inválidos.

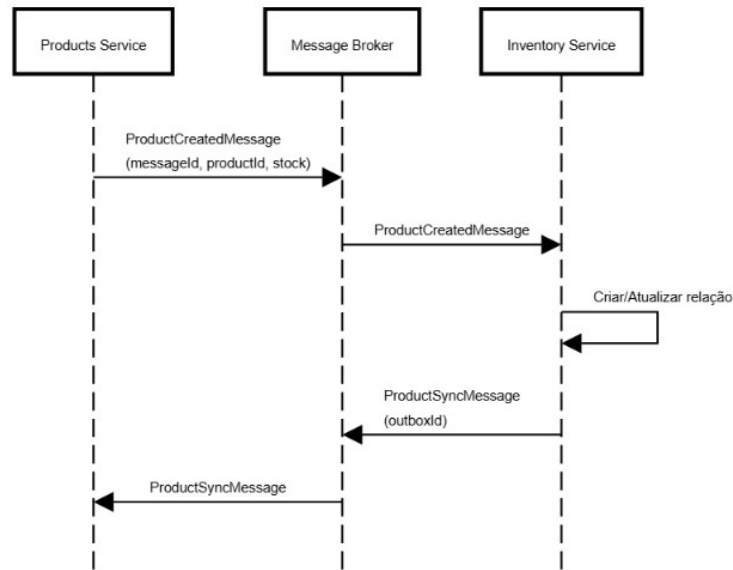


Figura 4: Diagrama de Sequência para a Criação de um Produto

3.3 Orders

O microserviço *Orders* é responsável pela gestão das encomendas dos clientes, desde o seu registo até ao seu acompanhamento ao longo do ciclo produtivo. Este serviço orquestra a interação entre clientes, produtos, inventário e planeamento da produção.

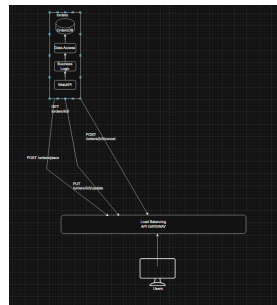


Figura 5: Microserviço Orders

Os seus *endpoints* permitem criar novas encomendas, consultar estados, atualizar quantidades e acompanhar o progresso das mesmas, sendo um dos serviços centrais do sistema. O serviço *orders* faz um número de verificações antes de fazer a encomenda, depois de validada é enviada a mensagem com os detalhes para o microserviço *Planning*, onde será gerado um plano para a Produção da

encomenda.

Quando uma encomenda estiver completa o serviço orders recebe uma mensagem para marcar a encomenda como completa.

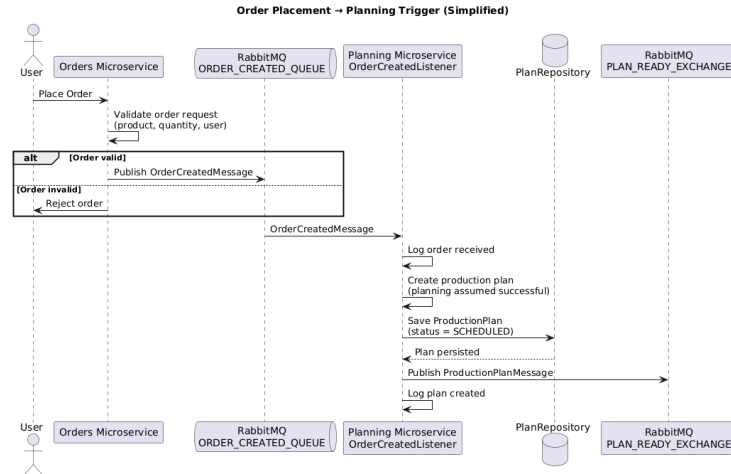


Figura 6: Criação de uma Encomenda

Para cancelar uma encomenda é enviada uma mensagem para o serviço Production, onde será feita uma verificação se a encomenda se encontra em condições para ser cancelada (não estar em produção).

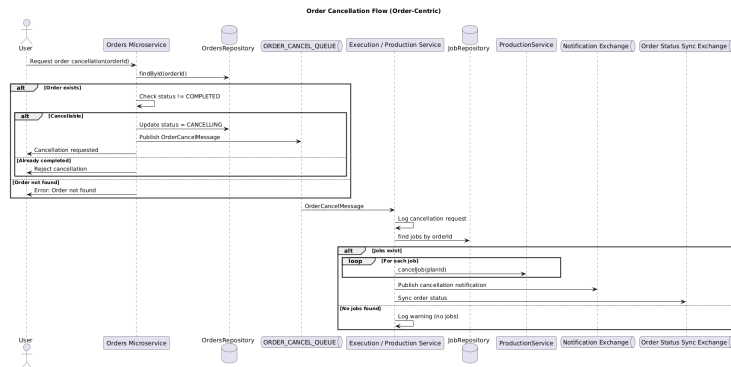


Figura 7: Cancelamento de uma Encomenda

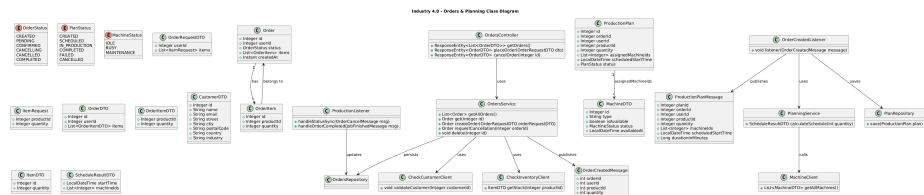


Figura 8: Diagrama de Classes Orders

3.4 Inventory

O microserviço *Inventory* gere o stock de matérias primas em stock. Este serviço mantém a informação atualizada sobre quantidades disponíveis, reservas de stock e movimentos de entrada e saída.

Através dos seus *endpoints*, é possível consultar níveis de inventário, efetuar reservas para encomendas e atualizar automaticamente o stock em função da produção e expedição.

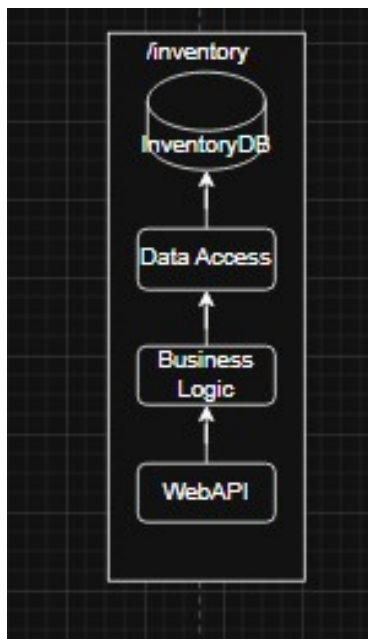


Figura 9: Microserviço Inventory

Este serviço age como uma simulação para o stock de matérias primas para criar um produto, ou seja, quando um produto é criado no catálogo com stock 500, na verdade quer dizer que existe capacidade para criar 500 produtos, na fabricação de produtos este valor é subtraído ao inventory.

3.5 Planning

O microserviço *Planning* é responsável pelo planeamento da produção, tendo em consideração as encomendas ativas, a disponibilidade de recursos e as máquinas disponíveis. Este serviço permite gerar planos de produção dinâmicos e adaptáveis a alterações de prioridades ou restrições.

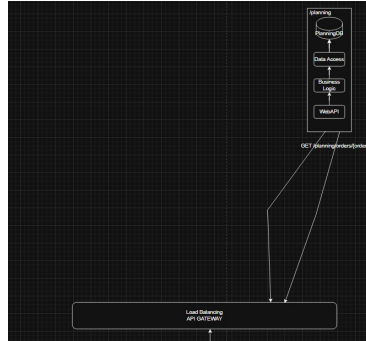


Figura 10: Microserviço Planning

Os *endpoints* disponibilizados permitem criar e atualizar planos de produção, nesta implementação o serviço planning verifica primeiro a disponibilidade das máquinas, se não estiverem disponíveis, ele verifica quais estão mais próximas de acabar o seu trabalho, e organiza um plano de produção logo após essas máquinas estarem disponíveis e prontas a serem utilizadas.

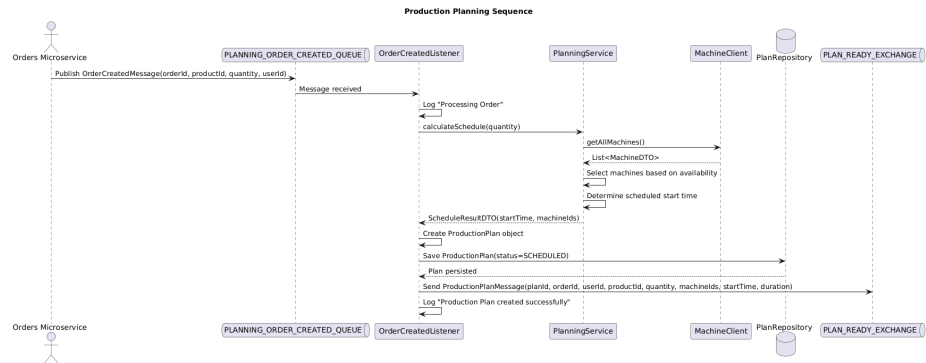


Figura 11: Criação de um Plano

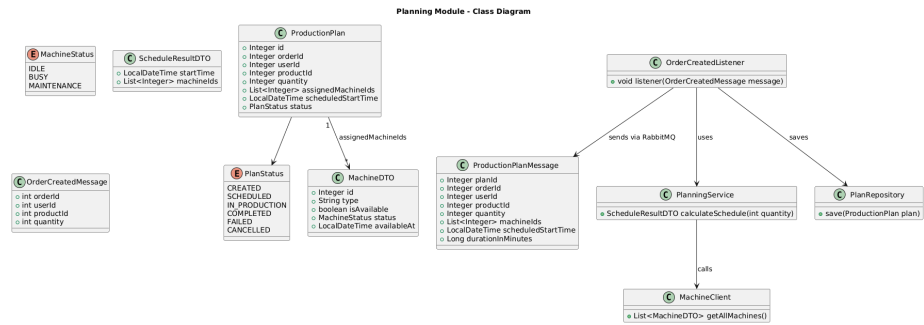


Figura 12: Diagrama de Classes Planning

3.6 Production

O microserviço *Production* gere a execução da produção no chão de fábrica. Este serviço acompanha a execução das encomendas de produção, recolhendo informação em tempo real sobre o progresso, tempos de execução e máquinas a serem utilizadas.

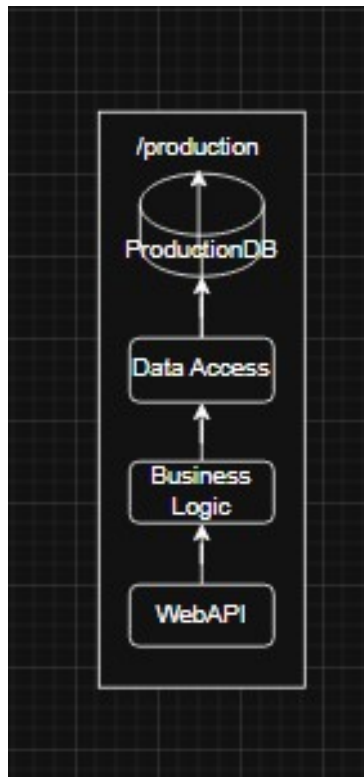


Figura 13: Microserviço Production

O serviço permite começar e concluir encomendas de produção, bem como reportar estados para outros serviços do sistema.

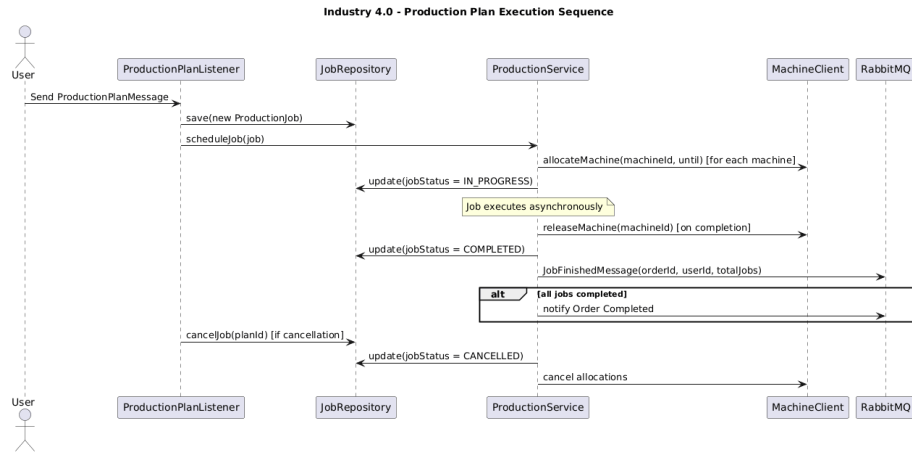


Figura 14: Diagrama de Sequência de Produção

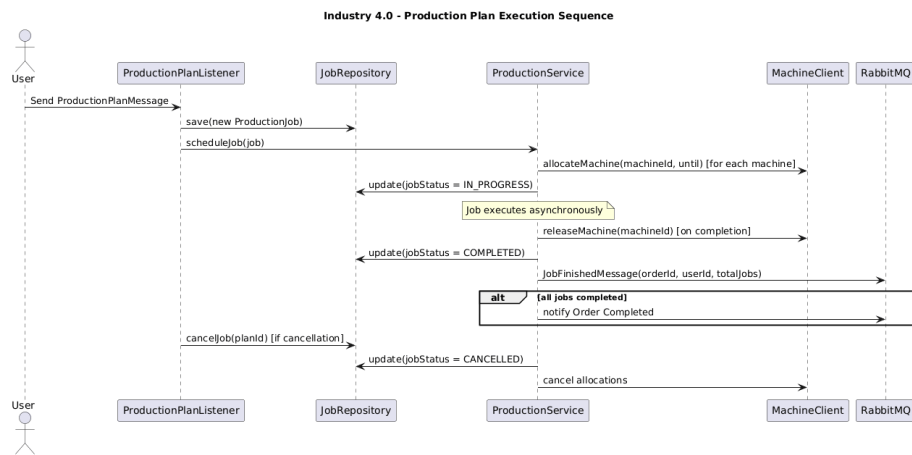


Figura 15: Diagrama de Classes do Production Service

3.7 Machines

O microserviço *Machines* é responsável pela gestão das máquinas industriais e pela integração com os equipamentos de produção. Este serviço monitoriza o estado das máquinas e a sua disponibilidade.

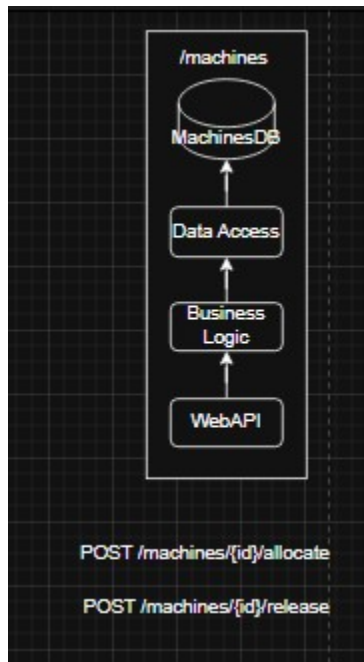


Figura 16: Machines Service

Através deste serviço podemos agendar produções com máquinas dependendo da sua disponibilidade.

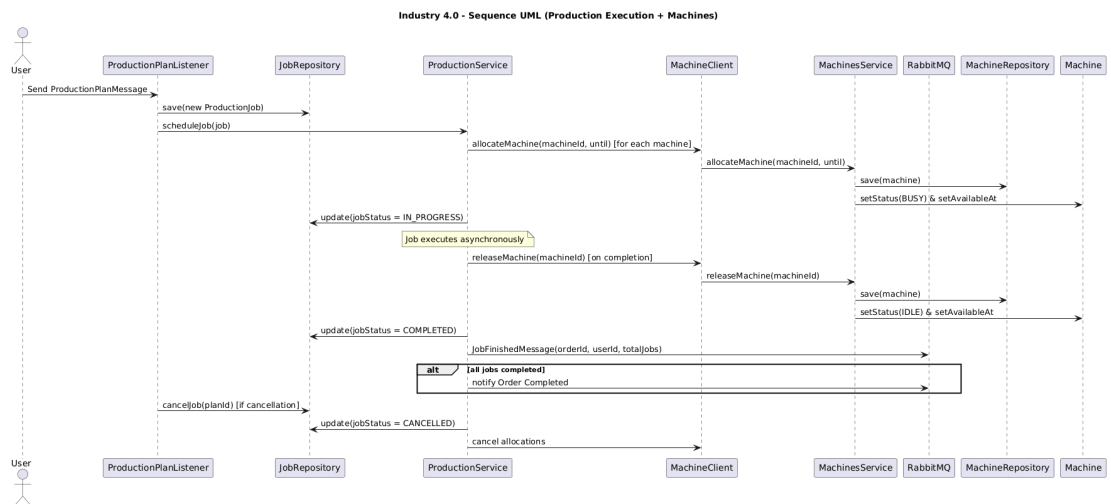


Figura 17: Diagrama de Sequência Machines

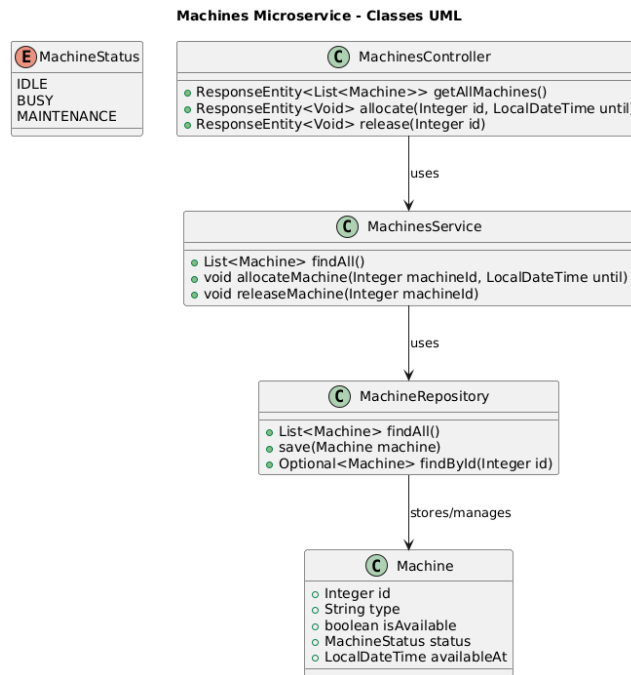


Figura 18: Diagrama Classes Machines

3.8 Notifications

O microserviço *Notifications* trata da comunicação e envio de notificações aos utilizadores e sistemas externos. Este serviço é utilizado para alertar sobre eventos relevantes, como alterações no estado das encomendas.

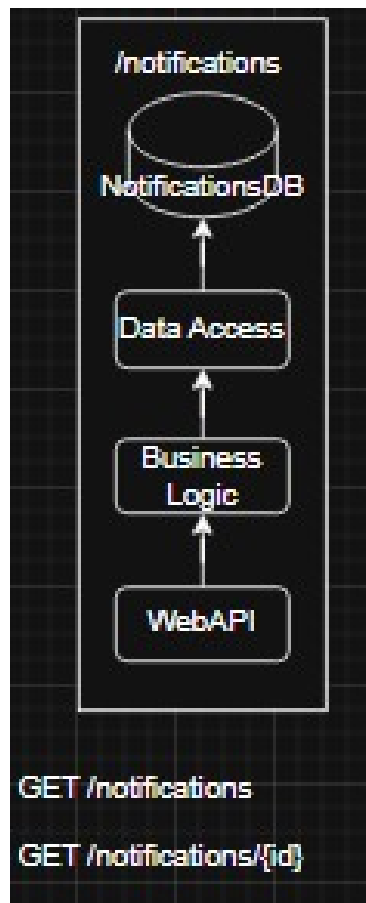


Figura 19: Notifications Service

Os microserviço permite ver as notificações e enviar notificações no caso de encomendas completas ou cancelamentos.

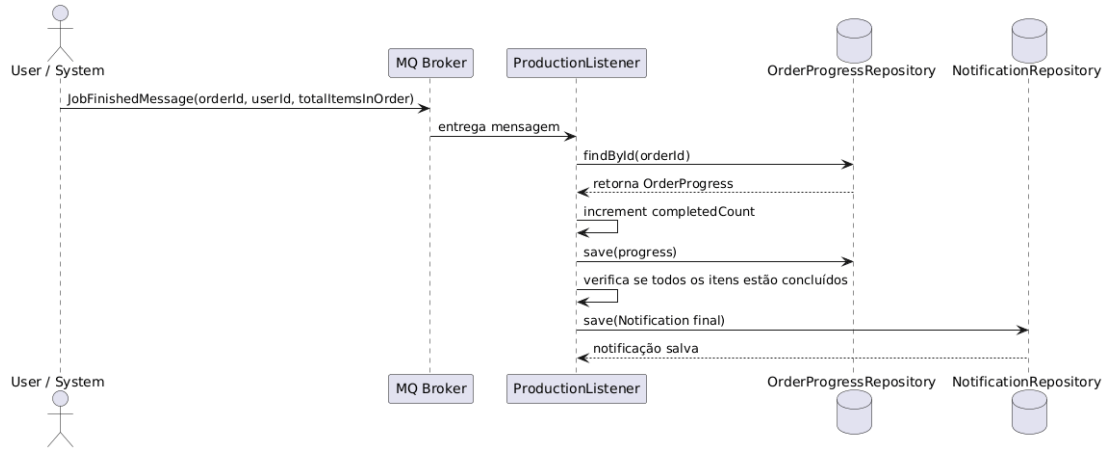


Figura 20: Diagrama Sequência Notifications

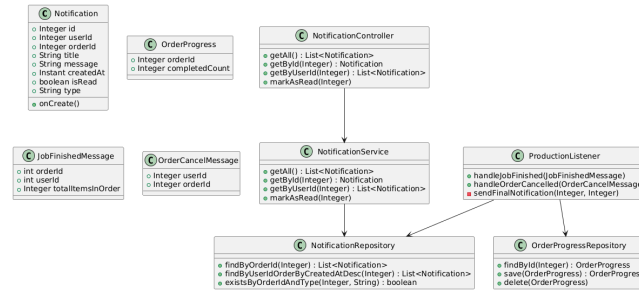


Figura 21: Diagrama Classes Notifications

4 Conclusão

A migração do sistema monolítico da *Fábrica 4.0* para uma arquitetura baseada em microserviços representou uma mudança tecnológica e cultural significativa para a organização. A decomposição do sistema em serviços independentes permitiu uma melhor separação de responsabilidades, maior escalabilidade, resiliência e flexibilidade, garantindo que cada domínio do negócio desde clientes, produtos e encomendas até à produção e gestão de máquinas pudesse evoluir de forma autónoma e eficiente.

A adoção de microserviços trouxe benefícios imediatos, como a possibilidade de implementar alterações e melhorias em serviços específicos sem afetar o restante sistema, maior controlo sobre os fluxos de informação e a capacidade de integração em tempo real com máquinas e processos de produção. Adicionalmente, a utilização de *endpoints* bem definidos e comunicação assíncrona entre

serviços contribuiu para a consistência dos dados e para a redução de erros operacionais.

Do ponto de vista organizacional, esta transição incentivou maior autonomia e responsabilidade das equipas, fortalecendo a cultura de colaboração e melhoria contínua. Ao mesmo tempo, a arquitetura baseada em microserviços estabeleceu as bases para uma fábrica digital alinhada com os princípios da Indústria 4.0, preparada para responder às exigências futuras do setor industrial e para crescer de forma sustentável.

Em resumo, a adoção desta abordagem tecnológica permitiu à *Fábrica 4.0* transformar o seu sistema de informação, tornando-o mais robusto, eficiente e adaptável, consolidando a capacidade da empresa de inovar e manter competitividade num mercado industrial em constante evolução.