

UC - Informática Industrial

“Automação de uma Linha de Produção”

Mestrado em Engenharia Eletrotécnica e de Computadores
Ramo de Automação

Beatriz Barbosa Guichard Lucena Coutinho up201906333@edu.fe.up.pt

Duarte Ribeiro Afonso Branco up201905327@edu.fe.up.pt

Manuel de Magalhães Carvalho Cerqueira da Silva up201806391@edu.fe.up.pt

Maria Inês Agostinho Simões up201904665@edu.fe.up.pt

Miguel Ramos Martins Rua Almeida up201907088@edu.fe.up.pt

Turma 1MEEC_A02 - Grupo A02_B

Índice

Introdução	2
Arquitetura de alto nível	3
Implementação	5
Comparação com design patterns e arquiteturas comuns	6
Agendamento	8
Testes e Resultados	10
Conclusões	12
Distribuição das notas	13
Referências Bibliográficas	14
Anexos	15

Introdução

No âmbito da Unidade Curricular de Informática Industrial do 1º ano do Mestrado em Engenharia Eletrotécnica e de Computadores, foi realizado um projeto com o objetivo de automatizar uma linha de produção.

Para o controlo direto do chão de fábrica utilizou-se um SoftPLC, recorrendo ao Codesys V3.5. Já para a implementação do ERP e MES, foi utilizada a linguagem de programação Java.

Foi implementado o protocolo de comunicação UDP/IP para que o ERP recebesse pedidos dos clientes. A comunicação entre o MES e o ERP é realizada através do protocolo TCP/IP, e a comunicação entre o MES e o PLC é realizada através do protocolo OPC-UA.

No presente relatório apresenta-se uma descrição de alto nível do sistema desenvolvido, assim como uma explicação sucinta da implementação e funcionamento dos seus principais componentes. Finalmente, são evidenciados os testes realizados e é feita uma análise dos resultados obtidos.

Arquitetura de alto nível

O projeto desenvolvido encontra-se estruturado num modelo de hierarquia funcional, dividido em três níveis: ERP, MES e PLC.

O ERP é usado para gerir e integrar as partes essenciais de uma empresa, permitindo que os diferentes departamentos facilmente partilhem informações, levando a uma gestão mais produtiva e eficaz. Desta forma, o ERP é responsável pela receção dos pedidos dos clientes, a sincronização de relógio com os outros sistemas e, principalmente, por planear a produção dos pedidos dos clientes, procurando maximizar o lucro. Os pedidos dos clientes são codificados em ficheiros XML e recebidos pelo ERP através do protocolo UDP/IP. Consequentemente, é gerado um MPS (*Master Production Schedule*) onde é definido, para cada dia, a quantidade e tipo de peças esperadas no fim do dia e a capacidade do armazém, com o objetivo de satisfazer os pedidos do cliente, evitando penalizações e minimizando os custos de produção. Note-se que o MPS decompõe a peça final desejada nas transformações necessárias desde a peça inicial (matéria-prima). Tendo em conta o MPS, são gerados os planos de produção de peças e de compra de matéria-prima. No início de cada dia, o plano de produção gerado é enviado para o MES. O plano de compra procura minimizar o custo de aquisição e indica o fornecedor escolhido, a data de chegada da matéria prima, a sua quantidade e o seu custo. Para além disso, o ERP calcula o custo total associado a cada pedido, tendo em conta os custos associados à produção de cada peça individual e possíveis penalizações, ademais de estar responsável pela sincronização com os outros sistemas, particularmente, com o MES.

O MES é um sistema de informação que liga, monitoriza e controla sistemas de produção e o fluxo de informações no chão de fábrica. O seu principal objetivo é garantir uma execução eficiente das operações, otimizando a produção. De facto, ao receber dados em tempo real sobre o produto e as atividades em execução no chão de fábrica, o MES permite uma monitorização eficaz da qualidade e uma verificação da precisão da execução das etapas do processo. Consequentemente, neste projeto, o MES tem como principal funcionalidade a otimização da utilização do equipamento disponível no chão de fábrica, garantindo que, para cada peça a ser produzida, é selecionado o melhor caminho a seguir para que esta sofra todas as transformações necessárias. O MES recebe do ERP o plano de produção e associa-o a dois tipos de receitas. O primeiro tipo de receita determina a ferramenta adequada a usar com base no estado atual do chão de fábrica, enquanto o segundo define o trajeto que uma peça deve seguir. Adicionalmente, o MES utiliza os dados que recebe para calcular estatísticas sobre o estado e desempenho da produção.

No nível inferior, o PLC tem como função controlar o chão de fábrica, com base nas receitas (especificações) enviadas pelo MES. O controlo das máquinas, ferramentas e tapetes é

abstraído em *Function Blocks* com entradas e saídas uniformizadas com o objetivo de facilitar a sua adaptabilidade. Além disso, o PLC permite rastrear as peças no chão da fábrica.

No que diz respeito às interações entre os sistemas, o ERP envia para o MES, através de uma ligação TCP, o plano de produção, onde são especificadas as atividades a realizar para o dia atual, permitindo que o MES escolha o equipamento adequado a usar e o caminho correto que a peça deve seguir. Como mencionado previamente, este plano de produção é diário e enviado no início de cada dia.

Além disso, para as comunicações entre o ERP e o MES, foi adotado o protocolo *publisher-subscriber* usando MQTT com TCP *WebSockets*. Este protocolo permite a troca de mensagens assíncronas entre diferentes componentes do sistema, estando alocado para cada servidor TCP uma *thread* que permite este mesmo assincronismo. Igualmente, para a comunicação entre os pedidos do cliente e o ERP, foi utilizado um servidor com a mesma morfologia, apenas com a ligeira diferença de utilizar o protocolo UDP em oposição ao servidor TCP. No que concerne à comunicação entre o PLC e o MES, adotou-se o protocolo OPC-UA (*OPC Unified Architecture*) para a transferência de dados, sendo escritas as variáveis que identificam as receitas e lidas as necessárias para o cálculo de estatísticas.

Adicionalmente, foram desenvolvidas duas interfaces gráficas: uma para o ERP e outra para o MES. A interface do ERP permite a visualização em tempo real dos planos de produção e compra gerados, bem como o custo associado a cada pedido. Por sua vez, a interface do MES permite visualizar em tempo real a lista de pedidos recebidos do ERP, o estado de cada pedido e as estatísticas de produção das máquinas e das docas de descarga. As imagens relativas às interfaces desenvolvidas encontram-se em [anexo](#).

No [Anexo](#) encontram-se os diagramas de classes e sequência do ERP e do MES, que facilitam a compreensão da estrutura do código e das interações entre as diferentes funções.

Implementação

Nesta secção, serão discutidos os detalhes de implementação do ERP, MES e PLC. Serão abordados os componentes e funcionalidades de cada um desses sistemas, fornecendo uma visão mais detalhada sobre sua estrutura e funcionamento.

No que diz respeito ao ERP, este possui duas *threads*, uma responsável pela receção dos pedidos dos clientes e outra responsável pelas tarefas de agendamento, mais concretamente o agendamento e envio das atividades a serem realizadas no MES para o dia atual da execução.

A *thread* responsável pela receção de pedidos do cliente implementa um servidor UDP que recebe o ficheiro XML com as informações do pedido do cliente que, posteriormente, converte para um objeto da classe *OrderInfo*. Seguidamente, as peças associadas ao pedido, na classe *OrderInfo*, são mapeadas através da classe *Piece*. Na classe *Piece* são identificados o *OrderID*, *PieceID*, *DueDate* e a data de chegada de matéria-prima, assim como todas variáveis necessárias para o cálculo do custo e penalizações.

No que diz respeito ao processamento dos pedidos, é utilizada a classe *MasterProductionSchedule* que implementa o MPS descrito na [secção anterior](#). Esta classe, é composta por um vetor de objetos da classe *Day* e implementa os métodos de escalonamento descritos no tópico [Agendamento](#). Considerando que um dia dura 60 segundos, como descrito no guião, são inseridas na classe *Day* todas as atividades a serem realizadas no espaço de um dia. A classe *Activity* implementa as atividades necessárias para satisfazer um pedido. Identifica o período de tempo no dia para a realização de uma atividade, assim como o número de identificação do pedido. Esta classe é estendida em cinco subclasses que pretendem representar as atividades consideradas na resolução dos problema: *ActivitySupplies*, *ActivitySwitchTools*, *ActivityTransportToMachines*, *ActivityProduction* e *ActivityUnloading*.

Já o MES possui três *threads*, duas para a recepção do objeto *Day* e o tempo atual da fábrica e uma terceira como o ciclo principal deste sistema.

Em relação a este ciclo, primeiramente, verifica-se o conteúdo na entrada do servidor TCP encarregado de receber o objeto da classe *Day*, portador das informações de produção do dia em questão. Desse modo, caso o objeto da classe *Day* seja recebido com sucesso, o programa orquestra a sua inserção de acordo com o algoritmo de agendamento descrito no [Agendamento](#), que, por sua vez, o adiciona à lista escalonada de processos a serem executados. Após a inserção do dia, o programa executa o método *execute()* da classe *Scheduler*, desencadeando assim a execução das ações escalonadas, considerando tanto o agendamento, quanto a disponibilidade do PLC.

Para concluir, o PLC, ao receber as ações provenientes do MES, determina o tipo de peça a ser retirada do armazém, quais máquinas devem ser utilizadas, a ordem em que estas devem ser preenchidas e a ferramenta que deve estar presente.

Comparação com *design patterns* e arquiteturas comuns

A utilização de *design patterns* durante o desenvolvimento trouxe diversos benefícios. Na verdade, *design patterns* facilitam a manutenção e escalabilidade, permitindo a reutilização de código e promovendo a interoperabilidade. Além disso, aumentam a robustez e segurança do sistema desenvolvido.

O *composite design pattern* é um padrão estrutural que permite estruturar objetos semelhantes numa estrutura de árvore, de tal modo a que possam ser manipulados como apenas um objeto, através do uso de polimorfismo. Para além disso, novos elementos podem ser introduzidos no sistema sem afetar o código já desenvolvido. No projeto, este padrão foi utilizado para manipular as várias subclasses da classe *Activity*.

O *command design pattern* permite criar instâncias de diferentes tipos de ações sem expor a lógica de criação, tornando o código mais modular e fácil de estender. Este padrão foi utilizado pelo MES na implementação do tipo ações a realizar.

Foi ainda aplicado o *recipe design pattern*, proposto pela norma ISA-88, que oferece um método sistemático de gerir a execução de um processo através da definição de uma receita que refere os passos necessários, separando-a do equipamento necessário. O uso deste padrão no MES promoveu a reutilização de código, contribuindo para a flexibilidade do sistema.

De facto, o projeto desenvolvido segue a arquitetura definida pela norma ISA-95. Esta arquitetura consiste em cinco níveis com o objetivo de facilitar a integração de sistemas de gestão empresarial e de controlo, cada nível representando um aspecto específico do processo de produção.

Foram implementados os níveis um (controlo - PLC), três (planeamento - MES) e quatro (gestão - ERP). A implementação em níveis contribuiu para uma melhor interoperabilidade e flexibilidade, permitindo a adição e modificação de sistemas em diferentes níveis sem interromper todo o sistema, e uma integração simplificada de novas funcionalidades.

A ISA-95 define ainda os fluxos de informação entre os diferentes componentes do sistema. Estes fluxos podem ser divididos em quatro classes: capacidade de produção, definição da produção, agendamento da produção e desempenho da produção. Entre o ERP e MES, há troca de informações contínua que abrange o agendamento de produção. Este fluxo de dados permite a coordenação de horários, quantidades e recursos necessários para as atividades de produção, garantindo uma colaboração eficiente entre os sistemas. Já entre o MES e o PLC, realiza-se uma troca de informações sobre as ordens/definição de produção. Estes fluxos ajudam a estabelecer uma abordagem consistente para a interoperabilidade entre os vários níveis dos sistemas.

Por sua vez, o modelo RAMI 4.0 propõe uma arquitetura abrangente que incorpora tanto a integração vertical quanto a integração horizontal. A integração vertical refere-se à conexão fluida

entre sistemas e fluxos de informação, desde o chão de fábrica até à gestão empresarial, permitindo uma visão holística e coordenada das operações. A integração horizontal facilita a comunicação e a partilha de informações entre todos os elementos do sistema.

O RAMI 4.0 incorpora também o ciclo de vida do produto. É reconhecida a importância de considerar todo o ciclo de vida de um produto, desde o seu projeto e desenvolvimento até à sua produção e manutenção. Ao abranger o ciclo de vida do produto, o RAMI 4.0 tem como objetivo otimizar a interoperabilidade ao longo de todas as fases, garantindo uma gestão integrada e contínua. Isto permite uma visão geral do produto, o que facilita a tomada de decisões em cada etapa. Assim sendo, no projeto, é possível identificar todas as transições e estatísticas associadas ao ciclo de produção através da classe *Piece*, contribuindo para decisões informadas.

Por fim, e com base no artigo "*Service Granularity in Industrial Automation and Control System*", pode-se considerar que o projeto desenvolvido segue uma abordagem de granularidade fina, de acordo com o tipo *Functionality Granularity*. Isto deve-se ao facto de o código desenvolvido ser altamente modular e reutilizável, tornando-o adaptável às necessidades específicas e facilitando a integração com novos componentes.

Agendamento

Nesta seção, será abordado o agendamento, abrangendo o *Master Production Schedule* (MPS) e o agendamento diário detalhado realizado pelo MES.

No contexto do MPS, descrito na seção [Arquitetura de Alto Nível](#), criou-se um algoritmo com o objetivo de inserir nos vários dias, representados na classe *Day*, as diversas atividades, representadas nas classes do tipo *Activity*, de forma genérica.

Com propósito de planejar todos os pedidos possíveis evitando penalizações, o algoritmo adotou a estratégia de dedicar a totalidade da fábrica a cada pedido do cliente, permitindo a paralelização de algumas atividades.

Assim, ciclicamente, são agendadas as várias atividades da seguinte forma: começa-se por agendar as atividades de descarga a partir do dia de entrega definido pelo cliente, em seguida, as de produção, incluindo a troca de ferramentas e, por fim, as de pedido de matérias-primas aos fornecedores. No caso da produção e troca de ferramentas, são tidas em consideração as várias transições necessárias para se produzir a peça pedida. Note-se que, no processo de agendamento, se começa da última transição para a primeira. Adicionalmente, as atividades de fornecimento são inseridas tendo em consideração a disponibilidade de matéria-prima no armazém.

Relativamente ao processo de inserção, são calculados os *slots* de tempo disponíveis para a atividade no dia em que se está a tentar inserir considerando os tempos estimados. É importante realçar que os *slots* disponíveis têm em consideração se a atividade pode ser realizada em paralelo com as atividades existentes.

Caso alguma atividade associada ao pedido do cliente não consiga ser inserida até ao momento atual, esse pedido é rejeitado. O agendamento de um pedido rejeitado é tentado novamente nos três dias depois da data de entrega original. Se não for possível agendar o pedido nesse período, este é abandonado, uma vez que é considerado inviável devido às penalizações por atraso, o que resultaria em prejuízo.

No MES, são percorridas as atividades do dia e interpretadas de acordo com o seu tipo. Cada atividade é transformada em ações bem definidas, representadas pelas receitas de trocar máquinas e pelas receitas de escolher caminhos. Clarificando, a receita de trocar máquinas representa a ação de troca de ferramentas, indicando quais ferramentas devem ser colocadas nas máquinas. Por outro lado, a receita de escolher caminhos descreve as ações relacionadas ao trajeto de peças dentro da fábrica. Especifica os caminhos a serem seguidos, as máquinas a utilizar e, no caso de ser necessário, o número de peças envolvidas.

Por fim, após obter uma lista de ações para as atividades recebidas, o algoritmo do MES ordena-as com base na prioridade, determinada pela data de execução das atividades. Em seguida, as ações são avaliadas de acordo com o tipo de receita e agendadas nas linhas de

tempo correspondente para cada tipo de ação. Desta forma, a inserção paralelizada nas linhas do tempo permite a otimização da produção.

Testes e Resultados

Para validar o sistema desenvolvido foram realizados testes que permitiram avaliar situações distintas. Apresentados a seguir encontram-se os três testes mais relevantes, em conjunto com uma imagem da interface gráfica do ERP que mostra as atividades planejadas para o pedido, organizadas por dias:

- **Teste 1:** produção de três peças P6 (P1 - P6)

Neste teste, é avaliado o tempo que o sistema demora a realizar um pedido simples. O teste demorou 206 segundos (3 minutos e 26 segundos), considerando que 20 segundos foram dedicados para as transformações de cada peça nas máquinas.

Day 0	Day 1	Day 2	Day 3
	Supplies, SwitchTools	TransportToMachines, Production	Unloading

Figura 1: *Master Production Plan* do teste 1

- **Teste 2:** produção de duas peças P5 (P2 - P4 - P7 - P9 - P5)

Este teste tem como objetivo analisar a resposta da fábrica perante um pedido que envolve múltiplas transformações. O tempo total foi 440 segundos (7 minutos e 20 segundos), considerando que 45 segundos foram dedicados para as transformações de cada peça nas máquinas.

Day 0	Day 1	Day 2	Day 3
	Supplies	SwitchTools, TransportToMachines, Production	SwitchTools, TransportToMachines, Production

Figura 2: Primeira parte do *Master Production Plan* do teste 2

Day 4	Day 5	Day 6	Day 7
SwitchTools, TransportToMachines, Production	SwitchTools	TransportToMachines, Production	Unloading

Figura 3: Segunda parte do *Master Production Plan* do teste 2

- **Teste 3:** produção de seis peças P4 (P2 - P4)

Finalmente, este teste permite observar o comportamento da fábrica quando o número de peças solicitadas excede a capacidade das máquinas para realizar essa transformação simultaneamente. O tempo total foi de 218 segundos (3 minutos e 38 segundos), considerando que 20 segundos foram dedicados às transformações de cada peça nas máquinas.

Day 0	Day 1	Day 2	Day 3
	Supplies, SwitchTools	TransportToMachines, Production	Unloading

Figura 4: *Master Production plan* do teste 3

Analisando os resultados dos testes é possível verificar que o algoritmo de agendamento apresenta algumas limitações. Na verdade, o algoritmo desenvolvido permite apenas produzir um pedido de cada vez, o que significa que as máquinas que não possuem a ferramenta necessária para realizar uma transformação não são utilizadas para completar outras transformações.

Além disso, não é realizada uma verificação prévia para determinar se a máquina já possui a ferramenta correta, resultando na alocação de tempo para uma atividade de troca de ferramentas que pode não ser necessária. Assim sendo, pode-se concluir que o algoritmo de agendamento constitui um *bottleneck*, já que o MES e o PLC permitem uma maior eficiência. Resumidamente, o potencial do chão de fábrica não foi totalmente explorado, resultando numa eficiência abaixo da possível.

Conclusões

Ao longo do presente relatório, foi apresentada, de forma simplificada, a estrutura e implementação do sistema desenvolvido. Pode-se afirmar que o projeto concluiu-se com sucesso, conseguindo automatizar a linha de produção, integrando os diferentes níveis: ERP, MES e PLC. A arquitetura adotada, baseada nas normas lecionadas ao longo da unidade curricular, juntamente com a aplicação de *design patterns*, possibilitaram um desenvolvimento estável e eficiente do sistema, apesar de lacunas resultantes da inexperiência do grupo.

A abordagem mais conservadora, embora tenha sido útil para alcançar um *Minimum Viable Product* e obter resultados satisfatórios, acabou por prejudicar a eficiência da fábrica, uma vez que não é possível realizar pedidos em paralelo. Especificamente, os algoritmos de agendamento, que deveriam contribuir para a otimização da produção e evitar penalizações financeiras, foram afetados. No entanto, foi possível garantir a execução das atividades propostas pelos clientes no chão de fábrica.

Devido às dificuldades no desenvolvimento do algoritmo de agendamento, não foi possível explorar todo o potencial do chão de fábrica, apesar do MES e o PLC permitirem maior eficiência. No entanto, é importante evidenciar que, devido à elevada modularidade do código implementado, melhorias no algoritmo de desenvolvimento seriam facilmente integradas na estrutura existente.

Em conclusão, pode-se afirmar que a execução deste projeto contribuiu para o aprofundamento do conhecimento em linguagens de programação orientadas a objetos, controladores programáveis e conceitos relacionados a algoritmos de agendamento. Além disso, foi evidenciada a importância da comunicação entre equipa para um desenvolvimento harmonioso do projeto.

Distribuição das notas

Tendo em conta o trabalho desenvolvido e empenho de cada elemento do grupo, decidiu-se, consensualmente, que a distribuição das notas deve ser a seguinte:

- Beatriz Barbosa Guichard Lucena Coutinho (201906333): **20%**
- Duarte Ribeiro Afonso Branco (201905327): **20%**
- Manuel de Magalhães Carvalho Cerqueira da Silva (201806391): **20%**
- Maria Inês Agostinho Simões (201904665): **20%**
- Miguel Ramos Martins Rua Almeida (201907088): **20%**

Referências Bibliográficas

“Builder.” n.d. Refactoring.Guru. <https://refactoring.guru/design-patterns/builder>.

“Composite.” n.d. Refactoring.Guru. <https://refactoring.guru/design-patterns/composite>.

Hanna, Terrell. n.d. “What is manufacturing execution system (MES)? | Definition from TechTarget.” TechTarget. <https://www.techtarget.com/searcherp/definition/manufacturing-execution-system-MES>

Tribolet, Perry. 2018. “The Recipe Design Pattern. Many of today’s software development... | by Perry Tribolet.”

Medium.<https://medium.com/@perrytribolet/the-recipe-design-pattern-e924ce8ab1c9>.

1588B-2022 - IEEE standard for a precision clock synchronization Protocol for Networked Measurement and Control Systems, n.d. <https://ieeexplore.ieee.org/document/9895348/>.

Sousa, Mário de, Aydin Homa, Alois Zoitl, and Martin Wollschlaeger. Service granularity in Industrial Automation and Control Systems, n.d. <https://ieeexplore.ieee.org/document/9212048/>.

Anexos

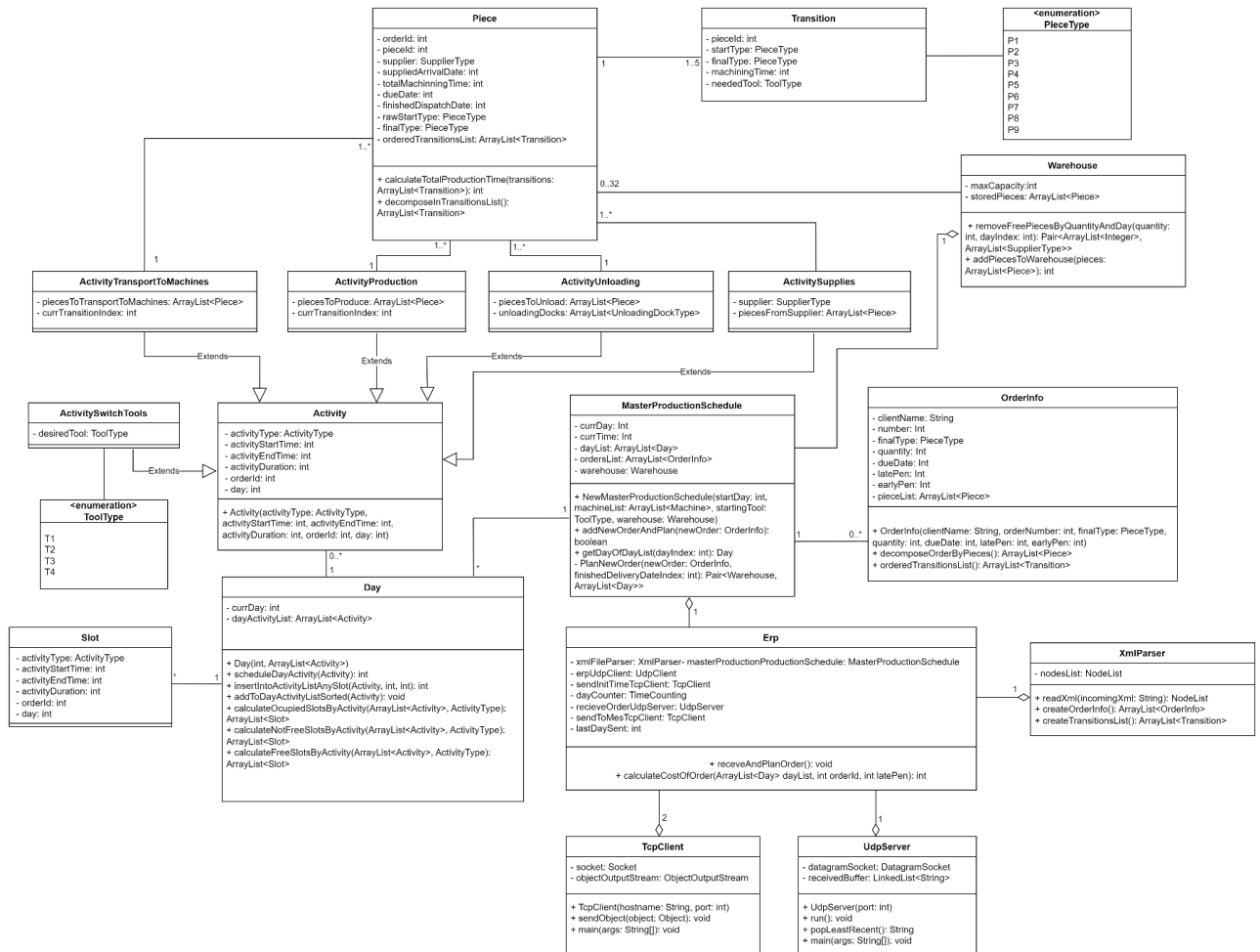


Figura 5: Diagrama de classes UML do ERP

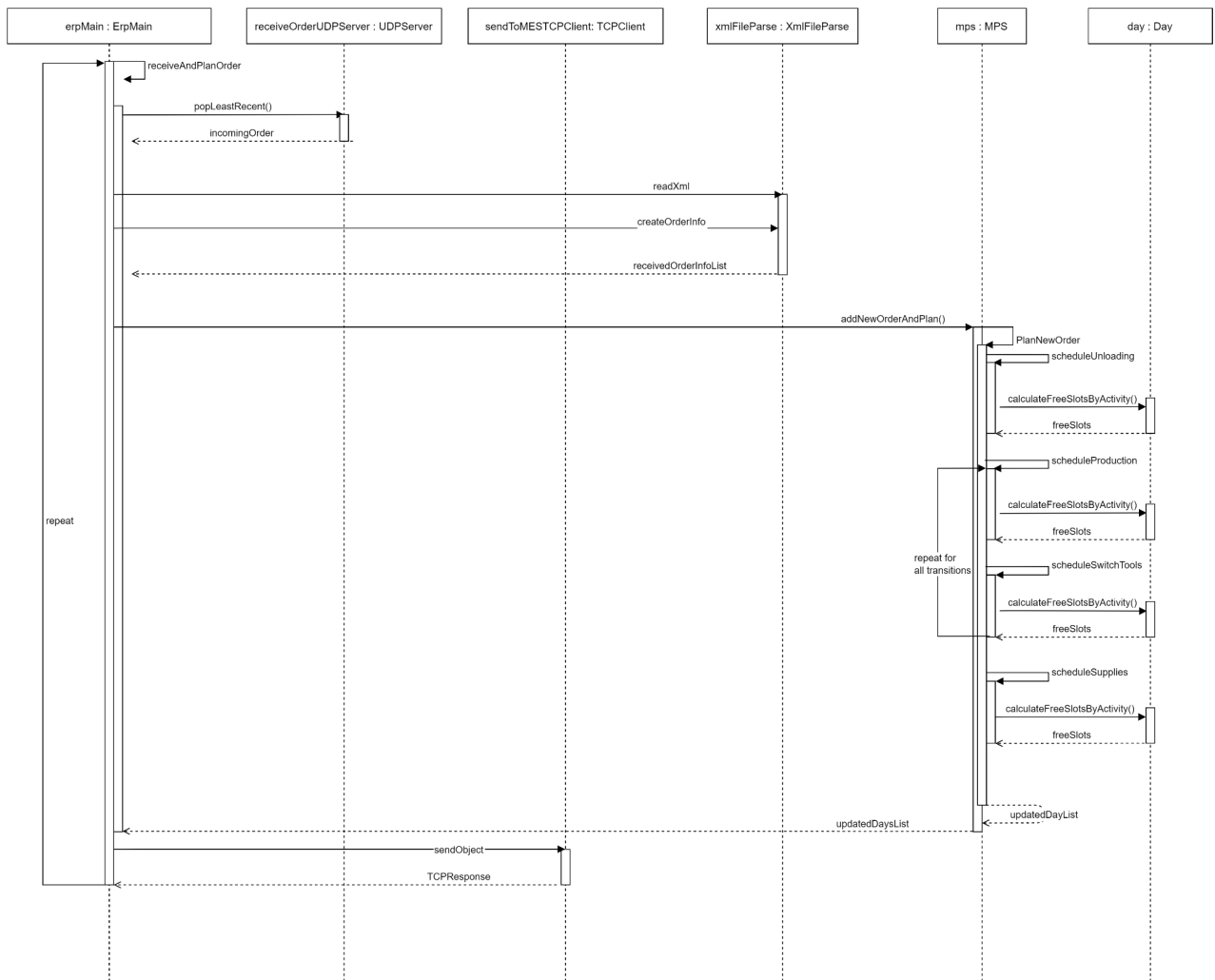


Figura 6: Diagrama de sequência UML do ERP

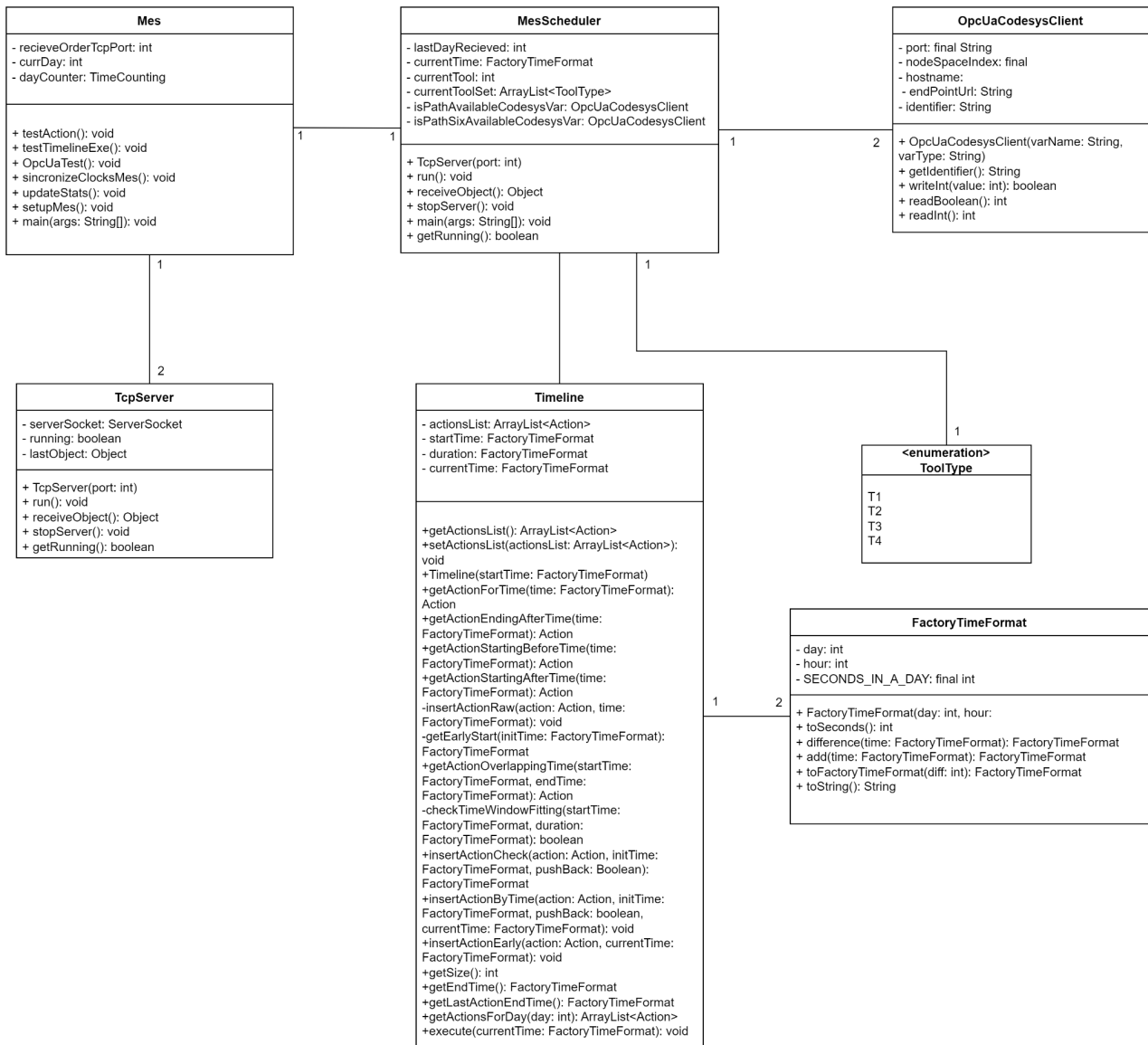


Figura 7: Diagrama de classes UML do MES

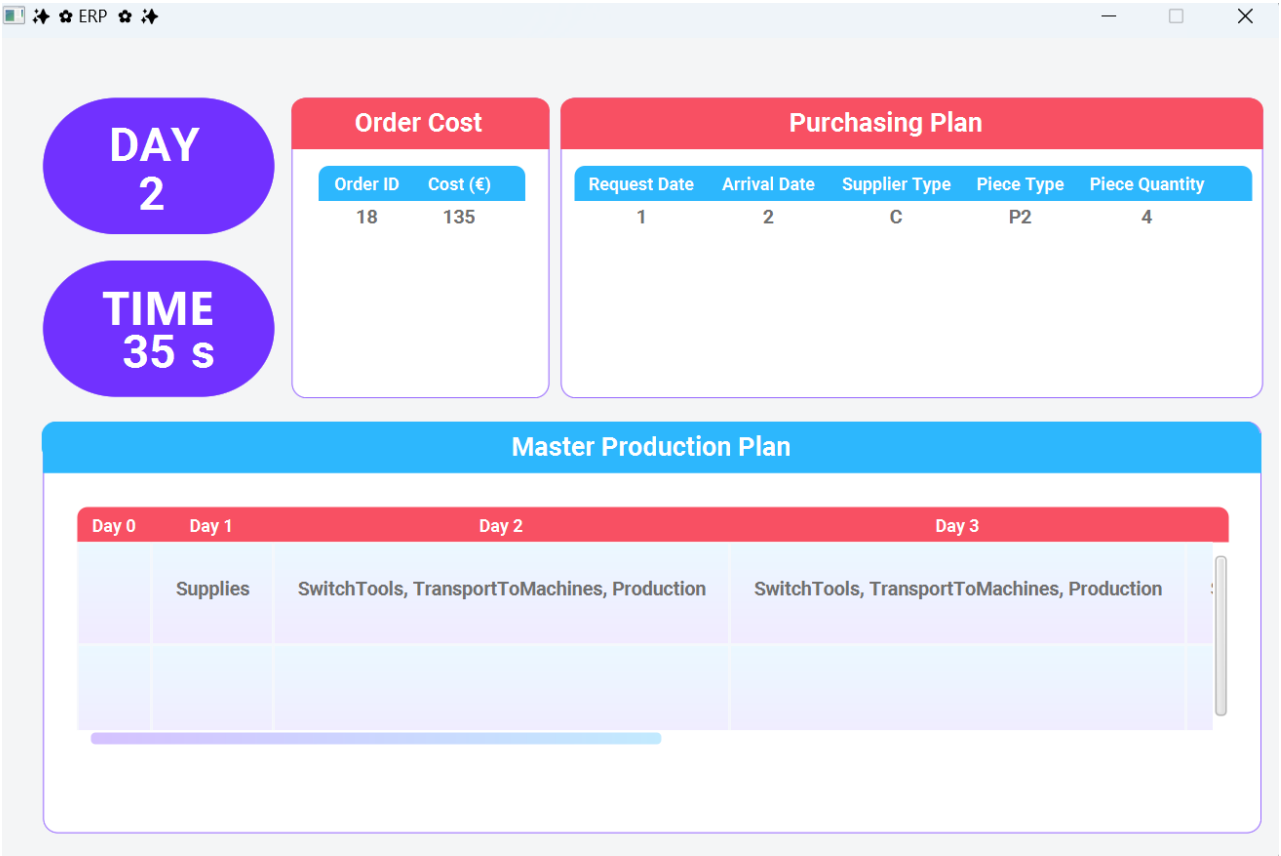


Figura 8: Interface gráfico do ERP



Figura 9: Interface gráfico do MES