**Factory Production Scheduler for a Cutlery Polishing Scenario**

Um projeto final de

**Data Processing Automation (Python)**

Curso da NOVA Doctoral School

Autor: **Fábio A. Seixas-Lopes** | [fl@uninova.pt](mailto:fl@uninova.pt) | estudante n. 56395

**Introdução**

No âmbito do curso de Data Processing Automation, da NOVA Doctoral School, foi proposta a realização de um projeto final com o intuito da aplicação dos conhecimentos lecionados durante as aulas. Sendo o curso no contexto do programa doutoral, foi pensado um projeto que pudesse relacionar-se diretamente com o trabalho desenvolvido para a minha dissertação. Esta baseia-se na implementação de uma arquitetura IoT que visa potenciar mecanismos de Inteligência Artificial de modo a otimizar, de uma forma genérica, processos de indústria e manufatura. Estas otimizações surgirão aos seus utilizadores, como “sugestões” aos processos, consoante a análise aos dados adquiridos durante a produção, através de sensores e módulos de monitorização. Faria sentido usar já os conceitos lecionados para automatizar dados iniciais de sensores e de monitorização, para estabelecer padrões e partir à descoberta das fases de processos de produção que beneficiariam com otimizações, assim como a análise dessas eventuais otimizações e até a modulação prévia dos processos, para efeitos de planeamento. No entanto, para efeitos de consolidação do conhecimento, e dado o facto de que o desenvolvimento desta dissertação ainda está no início, pensou-se que seria interessante desenvolver uma destas otimizações para articular os conceitos de processamento e automação da análise de dados.

Assim, surge a oportunidade de criar um agendador, que irei designar como *“scheduler”*, de ordens de produção. Neste, visa-se conseguir receber como input vários documentos com a referência do produto, o tipo, a urgência, a quantidade, etc. e comparar com tabelas de referência que explicitam que tipos de produtos existem, quais as suas caraterísticas relevantes para o processo de produção, o tempo que demoram a produzir, que preparações são necessárias para o produzir, etc. Desta relação pretende-se que o *scheduler* articule os dados de modo a otimizar o processo de manufatura, de um ponto de vista de tempo consumido. Ou seja, o objetivo do *scheduler* é providenciar, tendo em conta o que é necessário produzir, um calendário que prioritize o menor tempo de produção possível e atenda a questões como a prioridade de algumas encomendas. No fim, espera-se que o *scheduler* consiga providenciar as ordens de produção ordenadas (em algum tipo de estrutura em texto, para que o resultado possa ser facilmente reutilizado) e um cronograma (com alguma representação gráfica). Para que o programa corra com alguma autonomia, planeou-se também a implementação de uma interface ou API local, que permita fazer *upload* de ficheiros, através de uma página HTML, com ordens de produção e permita ver o resultado deste programa.

Para o cenário deste projeto serão usadas tabelas de referência e descrição de produtos de uma empresa de produção de cutelaria portuguesa, com a qual trabalhei em projetos de investigação no passado e que partilhou esta informação para estes fins. O cenário específico será o de otimização do calendário de produção de uma única máquina de polimento de colheres.

**Implementação**

Neste capítulo apresenta-se os elementos deste projeto que são as tabelas de referência, com informação dos produtos, as ordens de produção, que explicitam o que é necessário produzir, o *scheduler*, que são as funções principais que relacionam os elementos anteriores e o algoritmo de otimização, e a API, que providencia uma interface para a utilização deste programa.

Tabelas de Referência

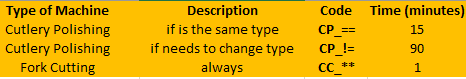
O processo de produção industrial contém sempre aspetos técnicos e características de produtos que condicionam a maneira como pode ser desenvolvido e para tal, qualquer ferramenta de otimização deve ter em conta o maior número possível de características que permitam compreender o processo e sobre estas, elaborar a sua otimização. No caso particular deste projeto, e para a realização de um calendário de produção, é necessária informação básica sobre os produtos que explicite o tempo de produção, o tipo de produto, as configurações e manutenções necessárias para que este seja produzido, etc.



*Fig. 1 –* Tabela de Produtos

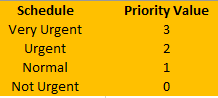
 Na Figura 1 está representada as primeiras linhas da folha correspondente aos produtos, no ficheiro “*product\_info.xlsx”*. Aqui, informação sobre a referência, nome, tipo, qualidade, rapidez de polimento e tempo de produção (para 3 horas, a título de exemplo), pode ser encontrada. No caso das caraterísticas iniciais, estas são intrínsecas ao próprio produto e as restantes (as últimas duas) são calculadas com recurso a outros valores apresentados nesta e noutras tabelas em outras folhas do documento (Figura 2), como é o caso da “*Colher Mesa Chicago”* (primeira entrada da tabela) em que é calculada a rapidez de polimento ao multiplicar a capacidade da bandeja de polimento para o tipo “*Table Spoon*” (12 por ciclo) por uma hora (60 minutos x 60 segundos) e dividir pela qualidade (*High*, que demora 48 segundos por ciclo). Ao dividir um lote de 3000 talheres por esta taxa de 900 talheres/hora, obtemos um tempo de produção de 3.33 horas (3 horas e 20 minutos).

(a)



(b)



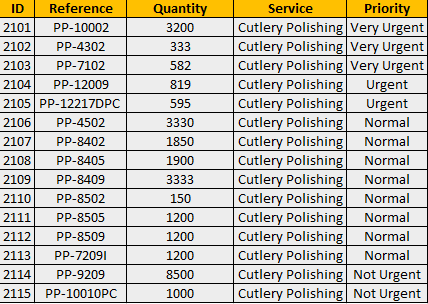
(c)

(d)

*Fig. 2 –* Outras tabelas do ficheiro *product\_info*. (a) – Ciclos por qualidade. (b) – Tempos de configuração da máquina. (c) – Tipos de manutenção. (d) – Tipos e valores de prioridade das ordens de produção.

Ordens de Produção

As ordens de produção (Figura 3) são documentos que definem que produtos serão necessários produzir, a sua quantidade e prioridade. Com esta informação, o programa fará as relações necessárias para encontrar o tempo de produção de cada um, assim como as associações que podem permitir diminuir o tempo geral de produção. Vários ficheiros com ordens de produção podem ser conjugados num mesmo calendário e cada ficheiro está numerado, sendo que os produtos terão um *ID* associado (que é o número do ficheiro seguido do número da ordem dentro desse ficheiro (e.g. ficheiro 21 e 1ª ordem dentro desse ficheiro, ID = 2101), sendo que os IDs inferiores a 9 estão reservados a tarefas de configuração, manutenção e outras).



*Fig. 3 –* Ordem de produção número 21.

*Scheduler*

O programa principal baseia-se nos ficheiros “*main.py*” e “*functions.py”*. Neste segundo estão as funções para leitura das tabelas, estruturação dos dados, cálculos, algoritmo de otimização e criação do gráfico correspondente ao calendário final. Algumas funções tornaram-se inutilizadas ao longo do desenvolvimento por necessidade de ir mudando o tipo ou formato de dados e variáveis, mas foram mantidas no projeto para o desenvolvimento futuro deste programa. Todas as funções têm uma descrição sucinta que refere o seu objetivo e o que retorna, e as variáveis têm nomes um pouco mais extensos para melhor leitura.

No “*main.py”* são chamadas as funções principais do programa, de uma forma sucinta, seguindo a seguinte ordem:

* Verifica se existem ficheiros de ordem de produção (se não existem, o programa termina);
* Lê todas as ordens de produção para uma lista de listas chamada *production\_buffer;*
* Aplica um algoritmo simples de otimização, ordenando todas as ordens carregadas de todos os ficheiros existentes por prioridade e tipo de produto (evitando assim a configuração mais demorada da máquina de polimento, que é maior quando se muda de tipo de produto);
* Adiciona ao *production\_buffer* os tempos de configuração da máquina necessários e as manutenções necessárias (que têm de ocorrer de algumas horas em algumas horas de funcionamento da máquina, ignorando o tempo que a máquina não está em funcionamento ou está em configuração);
* A partir do *production\_buffer* é criado um dicionário chamado *production\_schedule*, em que as chaves são o dia e os valores as ordens nesse mesmo dia;

e.g. {day1:[[order 100],[order 123], etc.],

day2:[[order 301],[order 643], etc.]}

* Com o *production\_schedule* é feito o *plot* da imagem do calendário com recurso à biblioteca *matplotlib* e grava-se a mesma numa pasta local.

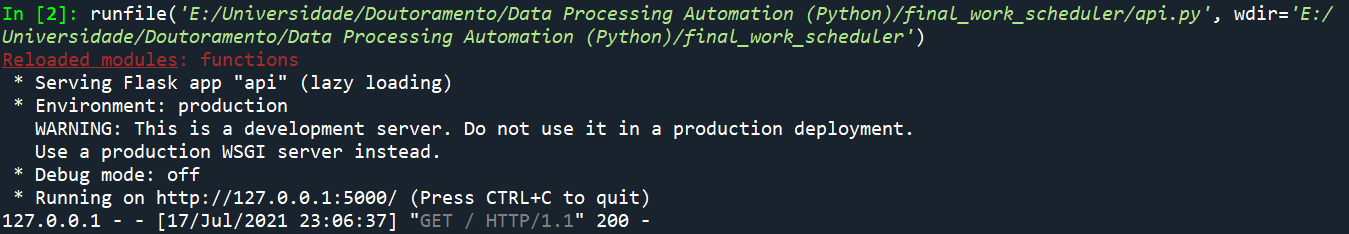
API

O ficheiro *“api.py”* aproveita as funções do programa principal e, usando a framework *Flask* para correr um servidor local, contém funções que permitem a navegação de uma página *.html* com o intuito de fazer *upload* de ficheiros, visualizar o calendário resultante e a ordens de produção organizadas em texto. Assim, esta interface tem um *index* onde o utilizador pode navegar as diferentes funções, que permite dois métodos *GET* (um para a imagem e outro para as ordens) e um *POST* para carregar os ficheiros. De um modo geral, é sempre verificado se existem ordens de produção antes de correr qualquer função e o utilizador é avisado caso isso aconteça, assim como se o ficheiro carregado tem a extensão correta.

É ainda importante referir que o programa usa pastas específicas para cada finalidade, sendo que guarda as ordens de produção carregadas na pasta “*production\_orders”*, a imagem do *schedule* na pasta “*schedule”* e a página .*html* na pasta “*template”.*

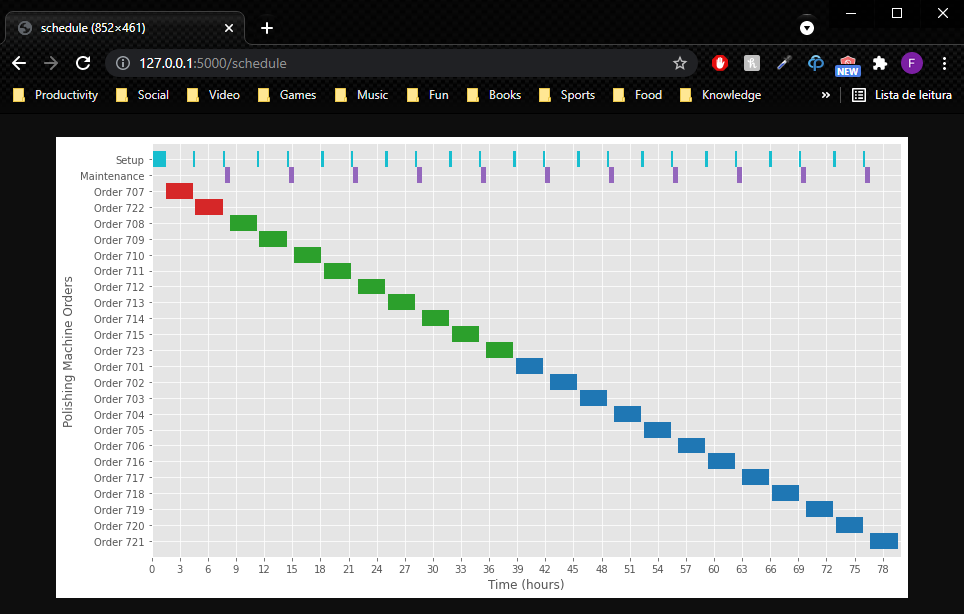
**Resultados**

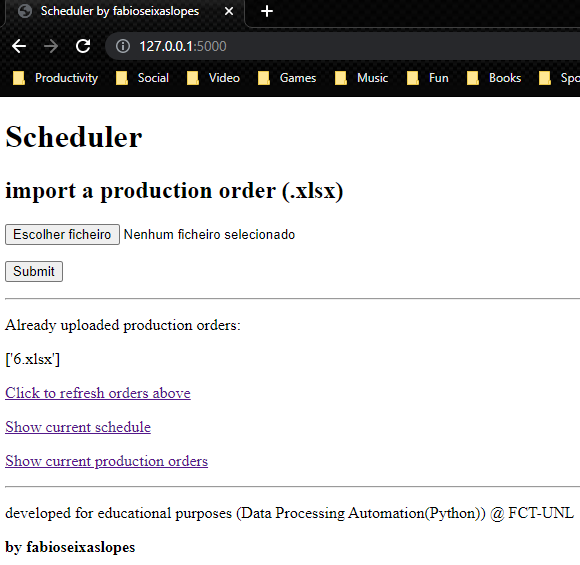
Após a breve explicação anterior, neste capítulo demonstra-se os passos de uma breve utilização do programa através da interface:

* Dentro do IDE (*Spyder*), corre-se o ficheiro *api.py*, que inicializa o servidor *Flask*. Este corre tipicamente no endereço local **http://127.0.0.1:5000/**e foi testado com o *Chrome;*

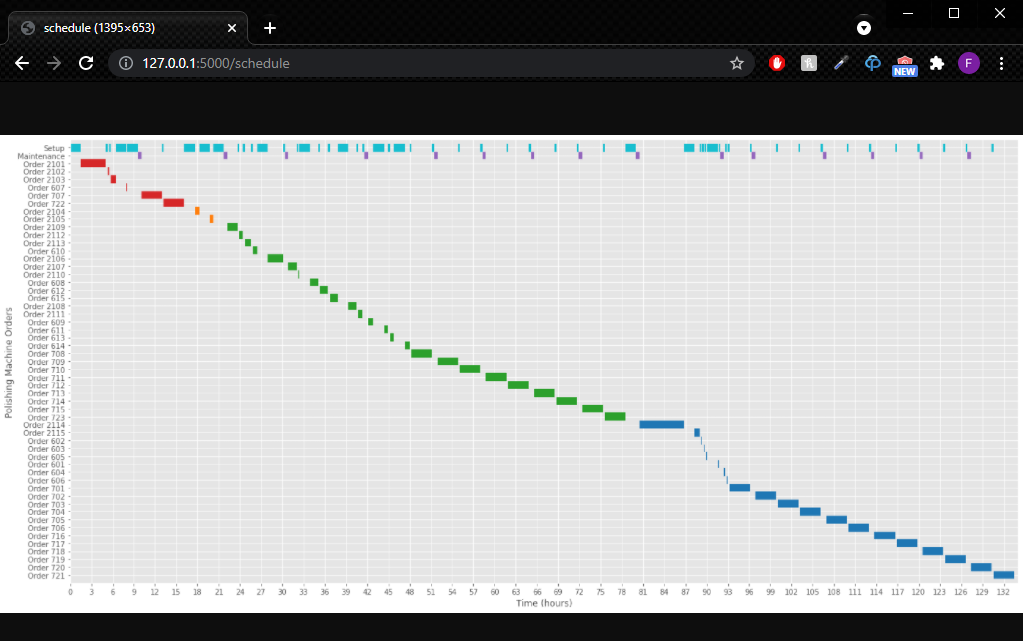
*Fig. 4 –* Início do programa e do Flask.

* No *index* é possível carregar os ficheiros, verificar quais estão carregados de momento e navegar para a visualização do calendário e da lista de ordens;
* Ao navegar para <http://127.0.0.1:5000/schedule> ou através do link da página inicial da interface, o programa calcula e cria o gráfico correspondente ao *schedule* das *production orders* carregadas;

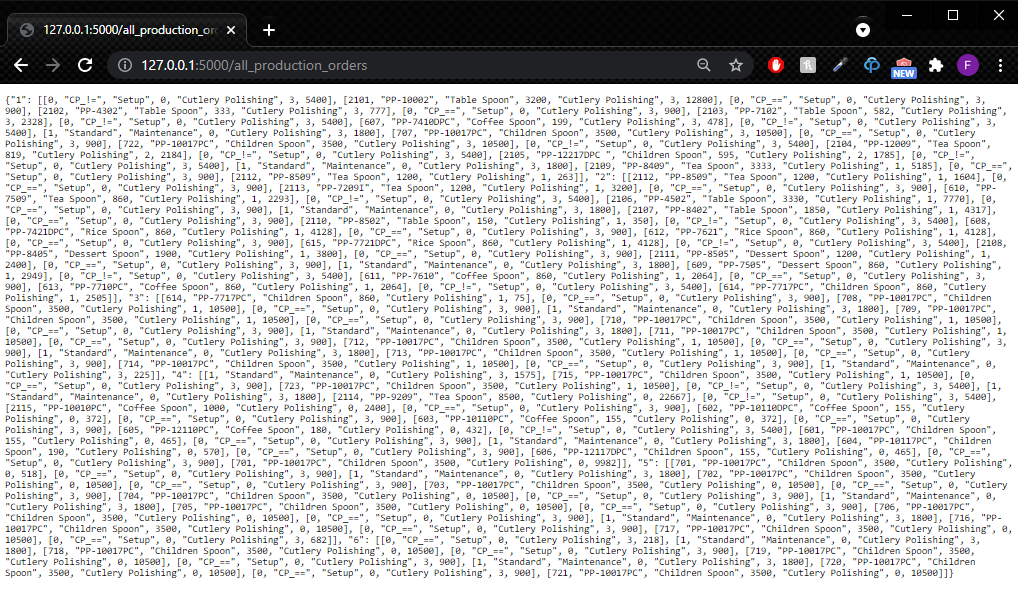




*Fig. 5 –* Página inicial da interface. *Fig. 6 – Schedule* do ficheiro 7.

* Ao carregar mais ficheiros com ordens de produção (existem 3 ficheiros de teste na pasta *upload\_these\_production\_orders*) obtemos um calendário mais completo e em http://127.0.0.1:5000/all\_production\_orders, ou através do *index*, navegamos até à representação do mesmo em *plain text.* No *schedule* a coloração das ordens é relacionada com a prioridade das mesmas e no topo estão sempre os tempos de configuração (*Setup*) e os tempos de manutenção (*Maintenance)*.

*Fig. 7 – Schedule* com ordens dos ficheiros 6, 7 e 21.



*Fig. 8 –* Ordens de produção referentes ao *schedule* apresentado na Figura 7.