UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA

Implementação De Workflow Em Modelo Misto

Simão Diogo Silva Ferreira

Mestrado em Engenharia Informática

Especialização em Engenharia de Software

Versão publica

Trabalho de Projeto orientado por:

Prof. Doutor João Pedro Guerreiro Neto

2020

Agradecimentos

Este percurso académico não se deve só a mim, mas a todos os que me rodeiam. Num percurso deste nível nada seria mais justo que agradecer às pessoas que mais me apoiaram.

À minha família que esteve sempre em todos os momentos que precisei, em especial o meu irmão que me ajudou e ensinou muito.

Aos meus amigos de infância, em especial 3 deles, que souberam tornar mais leve estes meses de trabalho desafiando-me a saber dosear trabalho e laser. O primeiro teria de ser o meu melhor amigo que contribuiu com conversas duradoras pelas noites fora. O segundo aquele que permitiu jogatinas durantes todos os dias. E a terceira que esteve cá o início do percurso de mestrado, incentivou e deu-me força para o fazer.

Aos meus professores da faculdade, e sim, agora um pouco de *graxa*, agradeço apesar das *secas*, muito contribuíram para a minha formação. Um obrigado especial ao meu orientador, pelo seu apoio, incentivo e disponibilidade que sempre revelou, principalmente em tempos de covid.

E por fim, um especial agradecimento a empresa que me acolheu neste projeto. A ajuda, a motivação, o entusiasmo e até a pressão contribuíram a conclusão deste trabalho. Permito-me ainda, deixar uma menção honrosa a um colega de trabalho e da faculdade por tudo o que partilhamos que deu proveito a um melhor trabalho.

Resumo

O workflow de um processo é a sequência de ações necessárias para que esse processo seja realizado com sucesso, e é definido como uma regra ou um conjunto de regras que levam à sua correta execução, do início ao fim. As ações podem ser de origem automática ou manual, no caso de ser automática a documentação, informação e/ou tarefas são passadas de pessoa para pessoa segundo as regras do processo.

O workflow representa uma relação entre tecnologia e processos de negócio. Também pode atuar como canal para transmissão de informação, uma vez que lida diretamente com pessoas e com a sua interação nas organizações.

Existe já software responsável por fazer a gestão deste workflow que garante que os passos que automatizam os processos ocorrem na sequência correta.

A Gestão de Processos de Negócio (ou Business Process Management, BPM) une gestão de negócios e tecnologia da informação com o objetivo de otimizar os resultados das organizações através da melhoria dos processos de negócio.

A utilização do BPM ao longo dos últimos anos tem vindo a crescer de forma bastante significativa, tendo já sido implementada em várias empresas que notam a sua rapidez e utilidade na melhoria dos processos.

Existem ferramentas denominadas por “sistemas de gestão de processos do negócio” (sistemas BPM) que monitorizam o funcionamento dos processos de uma forma rápida e barata. Dessa forma, os gestores podem analisar e alterar processos baseados em dados reais e em tempo real, e não apenas por intuição. Também possibilitam a análise de bottlenecks, consequências e fatores cruciais com facilidade e rapidez, visando a melhoria da empresa.

Neste projeto foi desenvolvido um sistema de gestão de workflows misto, onde uma máquina de estados é responsável pelos processos humanizados e um motor de BPM é responsável pelos processos automáticos.

Palavras-chave: Workflow, BPM, BPMN, Máquina de estados, Implementação mista.

Abstract

The workflow of a process is the sequence of actions necessary for that process to be correctly completed and is defined as a rule or group of rules which lead this sequence to its end. The actions can be manual automatic origin; in case they are automatic, the documentation, information and/or tasks pass from person to person according to the rules of the procedure.

The workflow represents a relationship between the technology and the business processes. It can also act as a channel for information transmission since it deals directly with people and their interaction within an organization.

There already is software for workflow management that ensures the steps that automate a process are performed in the correct sequence.

The Business Process Management (BPM) merges business management and information technology with the goal of optimizing the results of organizations through the improvement of the business processes.

The use of BPM over the years has been growing significantly, having been already implemented by various organizations that notice its speed and utility in the improvement of process management.

There are tools named “systems of business process management” - BPM systems - that monitor the BPM in a fast and cheap way. They allow managers to analyse and edit process workflow based on real-world and timely data, not only on intuition. They also allow managers to easily and quickly analyse bottlenecks, consequences, and crucial factors, which improves the performance of the organization.

This project developed a workflow management system that implements a state machine responsible for the manual processes and a BPM engine responsible for the automatic processes.

Keywords: Workflow, BPM, BPMN, State Machine, Mixed Implementation.

Índice

Lista de Figuras

Lista de Tabelas

Tabela de Siglas

Abreviatura	Significado
API	Application Programming Interface
AWS	Amazon Web Services
BA	BackOffice Administrativo
BE	BackEnd
BPD	Business Process Diagram
BPM	Business Process Management
BPMN	Business Process Model and Notation
CMMN	Case Management Model and Notation
COC	Cadeia de Custodia
FE	FrontEnd
GF	Gestão Florestal
JSON	JavaScript Object Notation
REST	Representational State Transfer
YML/YAML	YAML Ain't Markup Language

1 Introdução

Workflow [1] é a sequência de ações de um processo necessárias de modo que seja realizado com sucesso, e é definido como uma regra ou um conjunto de regras que levam à sua correta execução, do início ao fim. As ações podem ser de origem automática ou manual, onde as ações manuais consistem normalmente em adicionar informações ou documentação ao sistema e as automáticas em passar o processo para outra pessoa juntamente com as informações e documentação associada.

O workflow representa uma relação entre tecnologia e processos de negócio. Também pode atuar como canal de informação, uma vez que lida diretamente com pessoas e com a sua interação nas organizações.

Existe software responsável por fazer a gestão dos workflows que garantem que os passos que automatizam os processos ocorrem na sequência e tempo corretos.

A Gestão de Processos de Negócio (ou *Business Process Management*, BPM) é um conceito que une gestão de negócios e tecnologia da informação com o objetivo de otimizar os resultados das organizações através da melhoria dos processos de negócio.

A utilização do BPM ao longo dos últimos anos tem vindo a crescer de forma bastante significativa [2] dado a já ter sido implementado em várias empresas onde notam a sua rapidez e utilidade na melhoria dos processos, como por exemplo empresas na Indonésia [3].

Existem ferramentas denominadas por sistemas de gestão de processos do negócio (sistemas BPM) que monitorizam o funcionamento dos processos de uma forma rápida e barata. Dessa forma, os gestores podem analisar e alterar processos baseados em dados reais e não apenas por intuição. Também possibilita a direção da empresa a analisar bottlenecks, consequências e fatores cruciais dos processos que gerem com facilidade e rapidez visando a melhoria do desempenho.

Até os próprios desenvolvedores são beneficiados com a BPM uma vez podem desenvolver o workflow sem se preocuparem com certas lógicas tais como de como, onde e o porque vem estas informações e para onde o que irão executar certas operações visto que já foram desenhadas todas as possíveis situações de seguimento.

Por fim, a referência [4] define o Modelo e Notação de Processos de Negócio (ou *Business Process Model and Notation*, BPMN) como a notação da metodologia de BPM, consistindo numa série de ícones padrão para o desenho de processos, o que facilita o entendimento do utilizador. Usando elementos como os objetos de fluxo, objetos de conexão, *swim lanes* e artefactos é dada a oportunidade de fazer um diagrama de processos de negócio simples (*Business Process Diagram,* BPD). Também é permitido no BPD, construir o tipo de um Objeto de Fluxo ou um Artefacto para tornar o diagrama mais compreensível. Foram então desenvolvidos vários motores que implementam a BPMN, oferecendo facilidade na criação de BPD que por sua vez ajudam os stakeholders. Juntamente com as BPMN, existem diversos motores que permitem a implementação de uma gestão de processos rápida onde este motor irá executar o workflow.

O *Flowable*, cuja documentação está disponível em [5] , é um motor BPMN dinâmico com tabelas de decisão e mecanismos de gestão de caso CMMN (Case Management Model and Notation), todos escritos em java, open source e com uma perfeita integração com Spring.

Este relatório consistiu na implementação mista de um workflow, incluindo o canal de informação entre as pessoas da organização, baseado no *Flowable*.

Como implementação mista quer-se dizer a implementação da funcionalidade de lidar com processos humanizados e automatizados simultaneamente.

1.1 Contexto do projeto

A implementação acima mencionada foi realizada no projeto chamado FSIM. Este projeto visa promover a certificação das florestas portuguesas segundo as normas internacionais FSC e PEFC. O objetivo é manter a sustentabilidade e a produtividade das florestas. Este projeto irá ajudar múltiplas entidades gestores, proprietários florestais e clientes destes (controlo de Cadeia de Custódia) normalizando os processos seguidos por estes.

O interesse dos proprietários é ter o preço da madeira superior, sendo que a certificação oferece isso. Além disso, a tendência é para que as grandes marcas (como por exemplo o IKEA) deixem de adquirir madeira não certificada.

Este projeto suporta auditorias, gestão de não conformidades, gestão documental, gestão de processo com workflow simples e mais complexos e alarmística por mail ou por SMS para os responsáveis das várias tarefas.

Para isto tudo funcionar o FSIM está dividido em dois grandes ambientes, um destinado aos proprietários de terrenos (GF Gestão florestal) e outro para as empresas que trabalham com madeira e derivados (COC Cadeia de Custódia); por fim há um ambiente auxiliar, que se resume a uma administração para as outras aplicações (BA BackOffice Administrativo). É possível aderir ao módulo dos proprietários ou aos dois módulos.

O projeto foi desenvolvido em Angular 7 para o FrontEnd e uma API para o BackEnd desenvolvida em SpringBoot.

1.2 Motivação

Foi descoberta a necessidade de criar um sistema de tarefas onde seja possível os utilizadores especificarem alguns tipos. Tanto o workflow como as pessoas responsáveis por este seriam diferentes dependendo do tipo de tarefa. Com isto foi possível perceber a necessidade da existência de um sistema altamente configurável onde teríamos um maior controlo sobre o workflow.

A possibilidade da utilização de um motor de BPMN, visto que este é muito configurável e automático, pareceu uma solução possível. O Flowable foi o escolhido de entre vários motores por alguns motivos entre os quais a integração perfeita ao Spring e o facto de ser open source. Entretanto foram encontradas algumas limitações do mesmo, como por exemplo:

- falta de controlo sobre da tarefa em si
- os motores de BPMN utilizam as suas próprias tabelas e linguagem para saber que processo avançar a seguir
- fazer *queries* externas a estas tabelas é um pouco custoso em relação ao tempo pois estas tabelas estão estruturadas de maneira complexa e de maneira para o facilitar o funcionamento do motor

Assim sendo a solução proposta foi a criação de uma máquina de estados para a gestão das tarefas, sendo que nesta solução as transações entre estados poderiam disparar outros workflows, controlados pelo Flowable. Nesta solução há também estados que só avançam depois de um retorno de sucesso ou insucesso do Flowable. Assim certos processos das tarefas seriam totalmente automáticos, como por exemplo o envio de email.

A necessidade de fazer *queries* às tabelas do *Flowable* deve-se à necessidade de descobrir as tarefas que o utilizador ou grupo de utilizadores têm.

Outra grande motivação para o desenvolvimento desta solução baseada numa máquina de estados seria a persistência, para evitar a perda de informação no caso de haver um problema no servidor.

1.3 Objetivos

Como objetivo desta implementação foi o desenvolvido 2 módulos capazes de cobrir diversos casos de usos tais como os seguintes:

- Um utilizador regista-se na plataforma
- faz um pedido à entidade gestora
- é criado uma tarefa na entidade gestora
- a entidade gestora atribui um técnico para contactar o membro novo
- a entidade gestora altera o estado da tarefa e envia para o técnico
- é disparado um *workflow* para enviar o email para o técnico a avisá-lo que tem uma nova tarefa para cumprir
- independentemente do resultado, o estado da tarefa é alterado e enviado para o técnico
- o técnico decide se faz um pré-preenchimento ou se faz um preenchimento local.

Cada modulo terá a responsabilidade de executar processos distintos, um para os manuais/humanizados e outro para os automáticos, ambos os módulos terão de ser capazes de interagirem um com o outro para a execução de processos mistos.

1.4 Instituição de Acolhimento

A instituição escolhida de acolhimento para a elaboração dos dois projetos acima descritos foi a AKTM, uma consultoria de informática na área de tecnologias de informação, pertencente ao grupo Inoweiser, juntamente com mais seis empresas distintas, cujo objetivo principal é usar a tecnologia e a inovação para fornecer soluções aos seus clientes. Este grupo tem mais de 15 anos de experiência no mercado nacional e internacional.

1.5 Equipa

A equipa de desenvolvimento é constituída por um líder técnico, três programadores juniores de BackEnd, em que um deles sou eu, um programador sénior de BE, um programador júnior de FrontEnd, um programador sénior de FrontEnd, dois testers, um analista, um programador de jasper-reports, e um gestor de projeto.

A minha responsabilidade nesta equipa foi desenvolver um módulo sobre o Workflow das tarefas dos utilizadores.

1.6 Organização do documento

O documento está organizado nos seguintes capítulos:

1 – Introdução: Este capítulo descreve a motivação, os objetivos, contribuições deste projeto, assim como a estrutura do documento.

2 – Conceitos e Ferramentas Utilizadas: Neste capítulo é apresentado o ecossistema do projeto FSIM tais como alguma pesquisa bibliográfica sobre a gestão de tarefas.

3 – O Projeto e Arquitetura: Este capítulo relata em maior pormenor o FSIM e a sua arquitetura como também a estrutura do BackEnd.

4 – **Tarefas**: Neste capítulo é tratada a evolução das tarefas, desde a recolhe de requisitos até a sua implementação

5 – Flowable no Projeto: Neste capítulo é falado de decisões de implementação e da utilização deste em projetos

6 – Projeto posterior ao FSIM: É referente a um projeto que aconteceu posteriormente ao FSIM onde certas melhorias do TSK tiveram presentes.

7 – **Conclusão e Trabalho Futuro**: Este capítulo contém conclusões das implementações e possíveis melhorias as mesmas

2 Conceitos e Ferramentas Utilizadas

Este capítulo apresenta alguns conceitos sobre Workflow, gestão de tarefas, e sobre o ecossistema onde foi feito o desenvolvimento.

2.1 O que é BPM?

Segundo [2], a Gestão de Processos de Negócio (BPM, do inglês *Business Process Management*) é um conceito que une gestão de negócios e tecnologia da informação com o objetivo de otimizar os resultados das organizações através da melhoria dos processos de negócio. A utilização de BPM ao longo dos últimos anos tem vindo a crescer de forma bastante significativa e é possível notar a sua rapidez e utilidade na melhoria dos processos nas empresas que já a usam. Além disso, a perspetiva é aumentar a sua utilização.

O termo "processos operacionais" refere-se aos processos de rotina (repetitivos) desempenhados pelas organizações no seu dia a dia, ao contrário de "processos de decisão estratégica", os quais são desempenhados pela direção. O BPM diverge da remodelação de processos de negócio, uma abordagem sobre gestão popular na década de 90, cujo foco não eram as alterações revolucionárias nos processos de negócio, mas a sua melhoria contínua.

Existem ferramentas denominadas por sistemas de gestão de processos do negócio (sistemas BPM) que monitorizam o funcionamento dos processos de uma forma rápida e barata, contribuindo para que os gestores possam analisar e alterar processos baseados em dados reais e não apenas por intuição. Oferecem também à direção a possibilidade de analisar *bottleneck*, consequências, fatores cruciais com facilidade e rapidez visando a melhoria do desempenho.

Até os próprios desenvolvedores são beneficiados com a BPM uma vez podem desenvolver o workflow sem se preocuparem com certas logicas tais como de como, onde e o porque vem estas informações e para onde o que irão executar certas operações visto que já foi desenhado todas as possíveis situações de seguimento.

Nos anos oitenta e noventa, a Gestão de qualidade estava no topo da lista de prioridades das empresas em todo o mundo, desde então várias pessoas começaram a pesquisar e a desenvolver sobre este assunto. Foram escritos artigos como "Don't automate, obliterate" pela Harvard Business Review, livros como Business Process Management: The Third Wave tornando-se um dos assuntos mais importante nas empresas.

Alguns pontos importantes para os gestores interessados em implementar BPM para promover os resultados das suas organizações:

- O BPM tem como perspetiva a otimização e transformação de processos que evoluiu a partir das experiências passadas
- O BPM não consiste exclusivamente nas ferramentas de software que a ele estão associadas. Na verdade, o foco do BPM deve ser a melhoria e transformação de processos de negócios para que as organizações possam alcançar os resultados esperados do negócio: aumento de

produtividade, redução de burocracia, melhoria na rentabilidade, redução de defeitos e desperdícios, satisfação e fidelização de clientes.

- O BPM não deve ser confundido com uma metodologia ou com uma ferramenta, é um conjunto de boas práticas que recomenda formas de aplicar essas metodologias e ferramentas na gestão de processos.

Por fim é importante saber que o BPM não é rápido nem simples de ser implementada numa empresa. Envolve muitas mudanças tanto a nível dos recursos humanos como a nível de administração.

Para tal, esta gestão é feita através de práticas de gestão já conhecidas, como o mapeamento de processos, modelagem, documentação, planos de comunicação, automatização, entre outras, sempre com o objetivo de melhorar e transformar os processos de modo a conseguir os resultados esperados.

Uma das perspetivas do BPM é o foco nas pessoas, sendo estas o centro dos processos de negócio. Alguns sistemas BPM vêm seguindo esta corrente em busca de oferecer aos stakeholders mais facilidade e flexibilidade no uso com ferramentas simples e intuitivas.

- **Modelação:** A modelação de processos é feita nos próprios sistemas de BPM. Alguns seguem a notação mais usada atualmente, o BPMN, que usa ícones padrão para o desenho de processos. Esta etapa é importante para a automação pois é nela que os processos são descobertos e desenhados e também pode ser feita alguma alteração no percurso do processo, procurando a otimização.
- **Simulação:** A simulação acontece depois do desenho e da definição dos atores de processos, sendo o passo responsável por testar se as regras pré-estabelecidas se encontram de acordo com os objetivos da empresa e se são atribuídas às pessoas corretas.
- **Execução:** Depois da modelação e da simulação entramos na execução do processo O sistema BPM utilizado faz com que as tarefas sejam atribuídas aos seus devidos responsáveis, controlando o seu tempo de execução por pessoa e pelo processo em geral. Também podem ser utilizadas regras de negócio pré-estabelecidas.
- **Controlo:** O controlo ideal de BPM está presente durante todas as etapas do processo: antes, durante e depois. Desde a modelação até as análises pós-conclusão da execução, deve se realizar o controlo.
- **Otimização:** A otimização é crucialmente importante quando se trata de BPM. É essencial para que existam melhorias nos processos de modo a alcançar resultados positivos mais rapidamente, melhorar o serviço aos clientes e, possivelmente, com custos inferiores. Depende principalmente do controlo.

Pode ser encontrado um guia sobre a gestão de processos de negócios no livro [6] onde é possível ver várias informações mais detalhadas sobre este o BPM e a sua utilização.

2.1.1 Notação, elementos e Ícones

A notação é composta por uma quantidade significativa de ícones sendo que estes estão divididos em quatro categorias:

- Os **objetos de fluxos** são os principais elementos dentro da BPMN e consistem em três tipos de elementos: eventos, atividades e *gateways*.
- Os **objetos de conexão** têm o propósito de conectar os objetos de fluxos são de três tipos: Fluxo de Sequência, Fluxo de Mensagem, Associação.
- As ***swim lanes*** são *Pools* ou *Lanes.*
- Os **artefactos** são objetos de dados, grupos ou anotação.

As próximas secções definem com maior detalhe estes elementos.

2.1.1.1 Eventos (objeto de fluxo)

Os eventos são representados por um círculo:

Opcionalmente, pode ser adicionado um ícone dentro do círculo que indica o tipo de evento, como um envelope representando uma mensagem ou um relógio para a hora. Além disso, os eventos também funcionam como sinais, cada evento pode ser *Catching* ou *Throwing*. Quando um evento é *Catching* só inicia ou continua o processo quando recebe um sinal por outro lado; os *Throwing* enviam um sinal. Assim os eventos de início são sempre *Catching* e o de fim é sempre *Throwing*; os intermédios podem ser dos dois tipos. Um exemplo é um evento de início com um relógio, que espera por uma determinada hora para começar o processo.

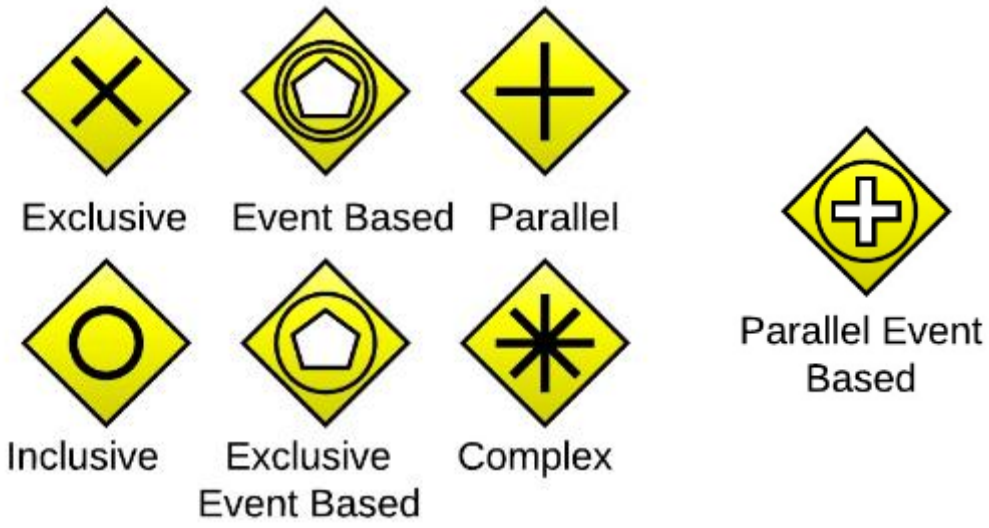
2.1.1.2 Atividades (objeto de fluxo)

As atividades são representadas por um retângulo em que os cantos são arredondados e descrevem o tipo de trabalho (ou atividade) que deve ser realizado. Esta atividade pode ser atómica ou composta.

Estas também contêm um ícone que representa que atividade a ser realizada.

2.1.1.3 Gateways (objeto de fluxo)

Os *Gateways* são pontos do workflow onde há uma bifurcação ou uma fusão de fluxos. Os ícones são representados num diamante.



Há vários Gateways diferentes:

- **exclusivos**: usados para representar fluxos alternativos onde apenas um dos caminhos de saída pode ser seguido; neste tipo de gateways a próxima tarefa a executar dependerá de uma condição a ser avaliada na tarefa atual.
- **baseado em eventos**: tal como no anterior, define a bifurcação exclusiva entre duas tarefas a ser executadas, mas em vez de a tarefa seguinte ser determinada com base numa condição atual, é com base na ocorrência ou não de um determinado evento.
- **paralelo**: usado para representar que o workflow se bifurca, mas que ambas as tarefas seguintes são executadas concorrentemente.
- **inclusivo**: são um misto entre paralelo e exclusivo, pois são usados para representar vários caminhos paralelos, onde cada um irá ou não ocorrer com base numa condição individual.
- **paralelo baseado em eventos**: dois processos paralelos são iniciados com base em um evento.
- **exclusivo baseado em eventos**: exclusivos utiliza um evento que está sendo avaliado para determinar quais caminhos mutuamente exclusivos serão seguidos.
- **complexo** usado para modelar comportamentos mais complexos de sincronização que não podem ser facilmente representados com os outros ícones. Geralmente são associados a texto que explica a lógica do Gateway.

2.1.1.4 Fluxo de sequência (objeto de conexão)

Fluxo de sequência é representado por uma linha cheia e uma seta pintada de preto numa das ponta, mostrando a ordem das atividades. Também pode ter um símbolo na outra ponta de diamante, que indica que é um fluxo condicional.

2.1.1.5 Fluxo de mensagem (objeto de conexão)

É representado com uma linha tracejada, um círculo aberto no início e uma ponta de seta aberta no final.

Este símbolo é usado para representar mensagens que fluem através dos limites organizacionais (ou seja, entre *pools – ver secção 2.2.1.7*). Um fluxo de mensagens nunca pode ser usado para conectar atividades ou eventos no mesmo conjunto.

2.1.1.6 Associação (objeto de conexão)

Este símbolo consiste numa linha composta por pontos e associa um objeto de fluxo a um texto ou artefacto; não representa qualquer relação direta com o workflow, sendo apenas documentação.

2.1.1.7 Swim lanes

As *pool* representam os principais participantes de um processo, geralmente separando diferentes organizações. Contém uma ou mais *lanes*. Cada *lane* representa um grupo de pessoas ou uma função na empresa, sendo possível criar *workflow* que mostram as interações previstas entre os vários intervenientes, como por exemplo tarefas executadas entre administração e novos funcionários.

Onde piscina é a *pool* e raia é *lane*.

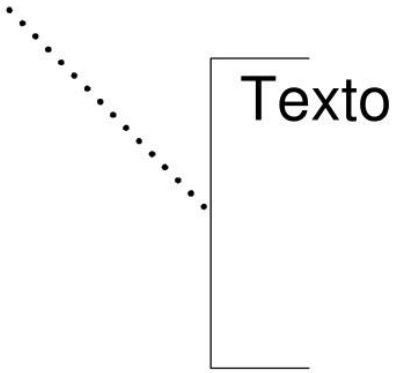
2.1.1.8 Artefactos

Os artefactos não afetam o workflow, mas indicam os objetos usados nas tarefas a que estão associados. Há vários tipos:

- **Dados**: mostram que dados são necessários ou produzidos na atividade.

- **Grupo**: representado com um retângulo de canto arredondado e linhas tracejadas. É usado para agrupar atividades diferentes, salientando a sua semelhança ou relação de proximidade.

- **Anotação**: usada como forma de aumentar a documentação do diagrama, explicitando passos ou oferecendo comentários aos diferentes aspetos do workflow, no sentido de aumentar a legibilidade do mesmo.



Podemos então obter um exemplo através do [7]. Neste exemplo podemos notar a existência de camadas que representam quem irá realizar que ação, tais como Gateways atividades e fluxos de sequência.

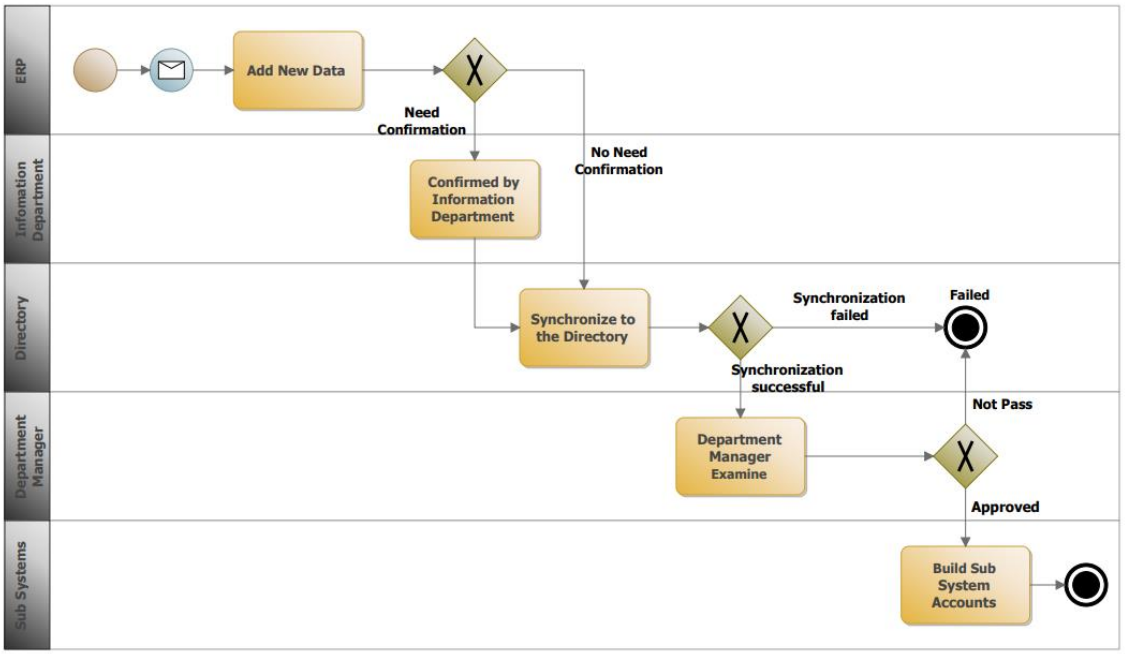


Figura 2-I BPMN exemplo

2.2 O que é BPMN?

De acordo com [3] [6], o Modelo e Notação de Processos de Negócio (do inglês Business Process Model and Notation, BPMN) tem como principal objetivo fornecer uma notação da metodologia de gestão de processos de negócios. Trata-se de uma grande quantidade de ícones-padrão para o desenho dos processos (ver secção 2.2.1). A notação também pode ser utilizada para modelagem da arquitetura de processos.

A BPMN foi desenvolvida pela *Business Process Management Initiative* (BPMI) de forma a unificar como é que as empresas fazem a modelação dos processos. Atualmente é mantida pelo *Object Management Group* e a versão atual do BPMN é a 2.0.

Com a BPMN é possível desenhar diagramas que representam os processos de negócio, baseados também nos fluxogramas de UML. Como o objetivo dos BPMN é apoiar a gestão de processos de negócios para os técnicos, utilizadores, empresas entre outros, é a notação fornecida tende a ser intuitiva.

A notação também pode ser útil para definir melhorias em processos, documentar processos existentes, e identificar e automatizar processos, pois é uma linguagem comum que pode servir de comunicação entre os designs de processos de negócio e a implementação.

Os softwares que adotam a utilização de BPMN podem ou não exigir o cumprimento da conformidade do BPMN 2.0, mas apenas se o software for completamente compatível com a especificação. Software apenas parcialmente compatíveis só podem exigir conformidade nas partes que foram baseadas na especificação.

2.3 O que é o Flowable?

Segundo a documentação oficial do Flowable [5], este fornece um conjunto principal de motores open source de processos de negócio. Fornece workflow e gestão de processos de negócio (BPM) para os desenvolvedores administradores e utilizadores de negócio.

No core está um motor de processo BPMN dinâmico experimentado e testado, extremamente rápido, com tabelas de decisão DMN e motores de gestão de caso CMMN, todos em Java.

Tudo isto pode ser executado e incorporado numa aplicação Java ou um serviço num servidor, *cluster* ou *cloud*. Podem ser executados de forma independente. Têm integração com Spring. Podem ser utilizados com APIS Java ou pedidos REST.

2.3.1 Porque esta escolha?

Após uma análise de vários motores BPMN foi possível perceber que o Flowable poderia ser o que melhor se encaixa às necessidades do projeto FSIM. Os requisitos mais importantes para esta análise foram:

- Se é *Open Source*
- Se é atual (o último lançamento foi recente e não foi descontinuado)
- Se funciona com Java
- Se funciona com Spring
- Se suporta o BPMN 2.0 Core

Na tabela abaixo podem ser encontrados alguns motores de BPMN e a sua comparação

Nome	Versão	Lançamento da última versão	Suporte BPMN V2 Core	Framework	Open Source
ActiveVOS	9.2.2	2013	NE	Java EE	NE
Activiti	7.1.108	2019/11/06	NE	Java	SIM
Bizagi BPM Suite	11.0	2016/08	NE	Java EE, .NET	NE
Bonita BPM	7.1.5	2016/01/08	NE	Java	SIM
Camunda BPM	7.11	2019/05/22	SIM	Java	SIM
Flowable	6.4.2	2019/07/11	SIM	Java, Spring	SIM
Imixs-Workflow	5.1.0	2015/10/21	SIM	Java EE	SIM
jBPM	7.29.0	2019/11/05	SIM	Java, Java EE, Spring	SIM
Sydle SEED	10.04	2014/07	NE	Java on Cloud	NE

Tabela 1 - Tabela de comparação de alguns Motores de BPMN

. NE significa não encontrado

Através de uma comparação rápida podemos verificar que o Flowable cumpre todos os requisitos pretendidos.

2.4 O que é uma máquina de estados?

Uma máquina de estados finita (ou autómato finito) é um modelo matemático usado para representar softwares. O conceito é concebido como uma máquina abstrata que deve estar em um estado. A máquina está em apenas um estado de cada vez, sendo estado atual. A mudança de estado (transição) é descrita por uma condição que precisa ser realizada para que a transição ocorra.

As tarefas no projeto irão ter uma semelhança *Issues* do Jira, que é um software comercial desenvolvido pela *Atlassian* uma ferramenta que permite a monitorização de tarefas e acompanhamento de projetos garantindo uma gestão de todas as atividades necessárias de serem realizadas em único lugar. O workflow padrão para as *issues* (tarefas) esta representado na seguinte imagem.

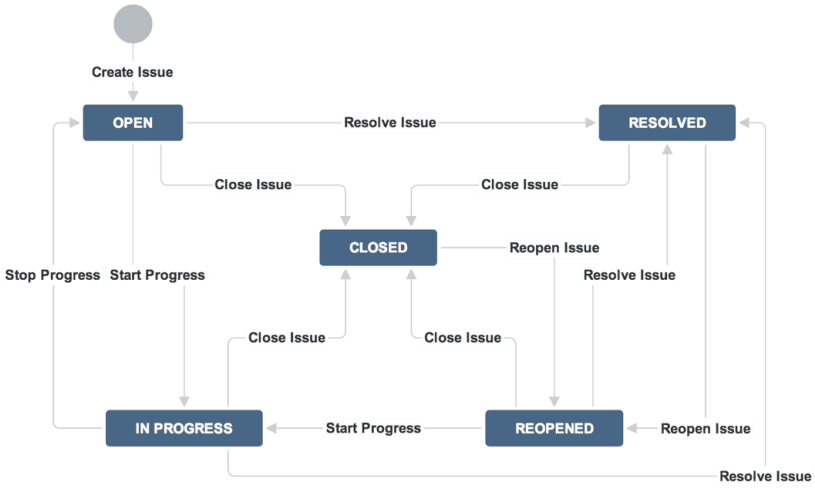


Figura 2-II - WorkFlow das issues do Jira

Onde podemos perceber a existência de cinco estados (*open, in progress, resolved, closed, reopened*) e algumas transições entre os estados como resolver uma *issue*, começar o progresso, entre outras.

2.5 Ecossistema de trabalho

Neste capítulo é descrito o ecossistema onde o FSIM está a ser desenvolvido usando diversas ferramentas, linguagens e *frameworks*.

O FrontEnd, interface com utilizador, é desenvolvido através de Angular [8] enquanto para a parte de BackEnd é usado Java Spring e para a base de dados *Postgres*.

Como este trabalho é direcionado ao desenvolvimento de um módulo de gestão de tarefas irá ser descartado o FrontEnd e como este é produzido.

Para além de Java Spring também é utilizado YML e JSON.

O *json* é usado tanto para validar os atributos de certas entidades tanto no sentido de BackEnd para FrontEnd como o inverso para objeto usado para enviar informações e guardar informações.

O YML é usado como ficheiros de configuração.

Por fim é usado também *Swagger* para contratualizar o FrontEnd com o BackEnd.

2.5.1 Java Spring Framework

O *Spring* é uma *framework* open source para Java contendo recursos que podem ser utilizados para criar qualquer aplicação Java de forma rápida, fácil e eficaz. Cada vez mais esta *framework* é usada tanto em adição como mesmo substituição do modelo *Enterprise JavaBeans*. Através da documentação encontrada em [9] podemos perceber a existência de vários módulos que o Spring oferece. Dependendo do tipo de aplicação e da necessidade é possível só utilizar alguns destes.

Os módulos são os seguintes:

- Spring Core Container;
- Programação orientada a aspetos;
- Autenticação e autorização;
- Convenção sobre configuração;
- Acesso a dados;
- Inversão do container de controle;
- Sistema de mensagens;
- Controlador de modelo de vistas;
- Estrutura de acesso remoto;
- Gestão de transações;
- Gestão remoto;
- Testes;

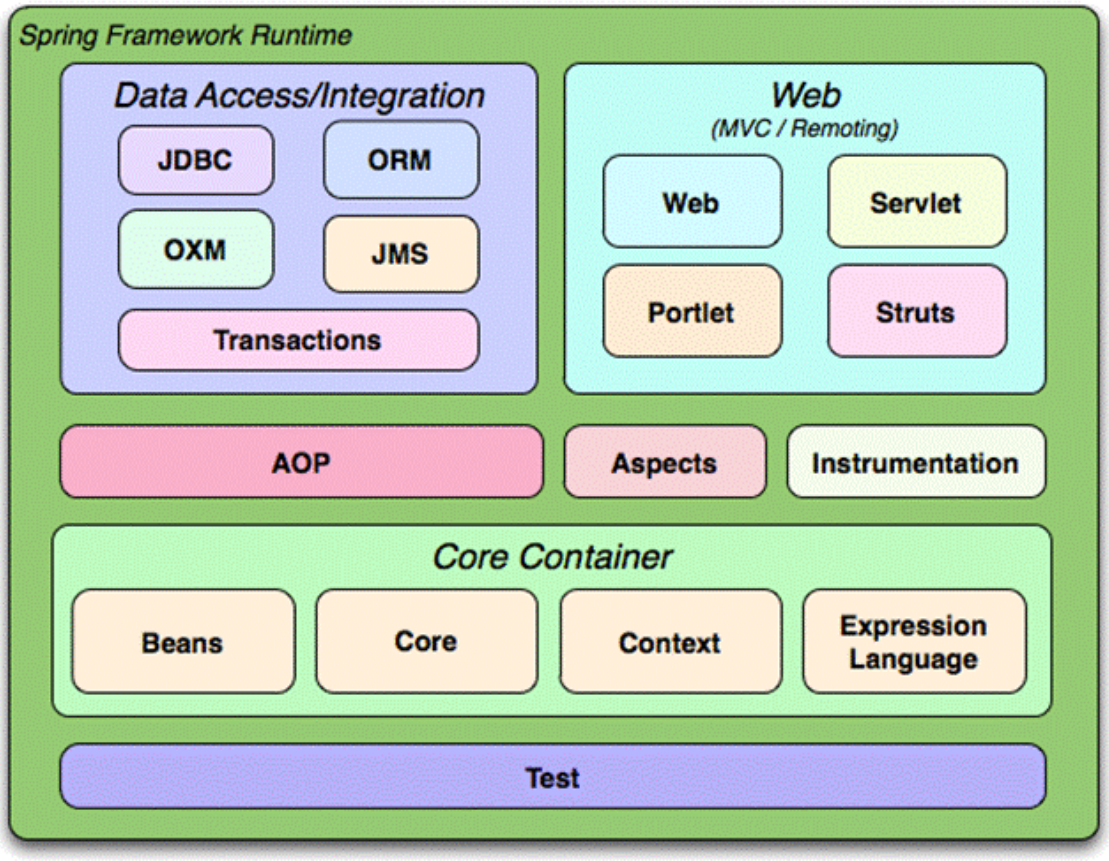


Figura 2-III - Modelos do Spring

Os módulos mais usados para o desenvolvimento do FSIM foram:

Spring Core Container: É o contém os módulos cores do Spring, isto é, necessário para todo o funcionamento e é configurável consoante a necessidade. Os módulos Core e Beans fornecem os fundamentais da estrutura, incluindo inversão de controlo (IoC) e recursos de injeção de dependência.

Inversão do container de controle: É fundamental para o Spring o container de inversão de controlo (IoC) que fornece uma forma consistente de configurar e gerir os objetos Java usando reflexão. Este é responsável pelo ciclo de vida de objetos específicos, criá-los através de métodos de inicialização e configurá-los através de conectá-los.

A maneira de definir que objetos, também denominados por *beans*, são estes pode ser através de dois formas, um ficheiro de configuração XML ou através de anotação nas classes. Estas formas devem conter as definições necessários para poder criar os beans.

Estes objetos podem ser obtidos através de pesquisa de dependências ou por injeções destas. A pesquisa de dependências é um padrão que um *caller* pede ao container um objeto com um nome específico ou de um tipo específico. Por outro lado, a injeção de dependências é um padrão em que o container passa os objetos pelo nome para os outros objetos por meio de construtores, propriedades ou métodos de fábrica. No FSIM é usado a injeção de dependências através de construtores.

Em muitos casos usar este módulo não é necessário, mas a sua utilização permite uma facilidade de configuração e personalização.

Acesso a dados: Este módulo oferece suporte a todas as *frameworks* populares de acesso a base de dados usados em java, tais como: JDBC, iBatis / MyBatis, Hibernate, Java Data Objects (JDO), Java Persistence API (JPA), Oracle TopLink, Apache OJB e Apache Cayenne, entre outros.

O *Spring* oferece para todas as *frameworks* suportadas alguns recursos como gestão de recursos, tratamento de exceções, participação em transações, desempacotamento de recursos e abstração para manipulação.

Para além do mais os desenvolvedores permitiriam o uso das API do Hibernate e JPA diretamente, mas por sua vez isto requer uma gestão de transações pois o Spring deixa de assumir a responsabilidade de obter e fechar os recursos a base de dados e também deixa de suportar a conversão de exceções.

Auxiliado com o módulo de transações esta *framework* de acesso a dados oferece uma abstração para o trabalho com *frameworks* acesso a dados. O Spring é o único *framework* que disponibiliza em java ambientes de gestão de acesso de dados fora de um servidor ou de um container de aplicações, para além disso não oferece uma API de acesso a dados comum, mas sim oferece todo o poder das *AP* suportadas mantendo-as intactas.

Gestão de transações: Este módulo traz é responsável por oferecer um mecanismo de abstração ao Java. Esta é capaz de trabalhar com transações locais e globais (as locais não requerem um servidor de aplicação), *nested transations, save points* e trabalhar em quase todos os ambientes de Java. Para a realização deste mecanismo temos duas maneiras possíveis de gerir as transações, programando o *template* de transições do Spring ou usando XML ou Java *anotations* para o mesmo

Auxiliado com o módulo de acesso a dados do Spring é possível configurar um sistema transacional sem precisar de confiar no JTA ou no EJB. A *framework* transacional também se integra aos mecanismos de mensagens e cache.

Testes: Este módulo permite realizar testes a logica do backend, fazendo *mocks* (simulado) certas partes apenas para testar uma certa funcionalidade ou função. Estes testes são normalmente desenvolvidos usando “junit” que é um *framework* focada em executar testes unitários.

2.5.2 Postgres

O Postgres é um diminutivo de PostgreSQL é um sistema de gestão de base de dados relacionada que é open source, que segue os padrões técnicos e destaca a extensibilidade. Através da sua documentação, [10], é possível saber que foi projetado para ligar com uma variedade de carga de trabalho e é o padrão para macOS Server, mas também esta disponível para outros sistemas operativos como o Windows e o Linux.

Apresenta transações com as propriedades ACID (*Atomicity, Consistency, Isolation, Durability*), automaticamente atualiza vistas, vistas materializadas (objetos obtidos em *queries*), *triggers*, chaves estrangeiras e procedimentos de armazenamento.

Para além desta base de dados também é usado algumas extensões a mesma, como a plpgsql, postgis, postgis_topology, fuzzystrmatch e postgis_tiger_geocoder. As “postgis” são é uma extensão que permite armazenar geometria, esta que pode representar um terreno. A “plpgsql” permite auxiliar nas queries feitas. A “fuzzystrmatch” oferece funções para detetar similaridades entre 2 *Strings*.

2.5.3 JSON

JSON significa *JavaScript Object Notation* é um formato de ficheiro padrão que tem como objetivo transmitir para humanos através de texto legível por eles objetos de dados que consistem em pares de atributos-valor, um formato de dados serializável. É o principal substituto de XML e possui uma gama diversificada de aplicações.

Este formato é independente do idioma e mesmo sendo derivado de JavaScript é utilizado em muitas linguagens de programação para gerar e analisar dados. O tipo de formato padrão usado na internet é JSON, para enviar a receber objetos.

A sintaxe do desta linguagem pode ser encontrada na sua documentação [11].

Cada objeto tem uma *String* que representa o nome do atributo e um valor. Pode ter este par o número qualquer desde que sejam separados por “,”.

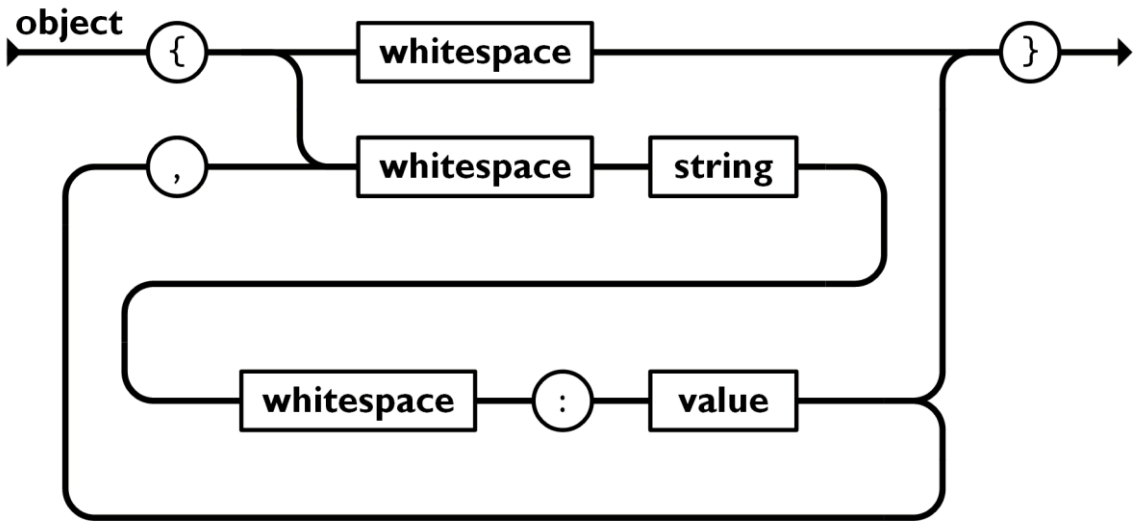


Figura 2-IV - JSON Object notation

O valor pode ser uma *string*, um número, um objeto, um *array*, um *boolean* (*true* ou *false*) ou vazio (*null*).

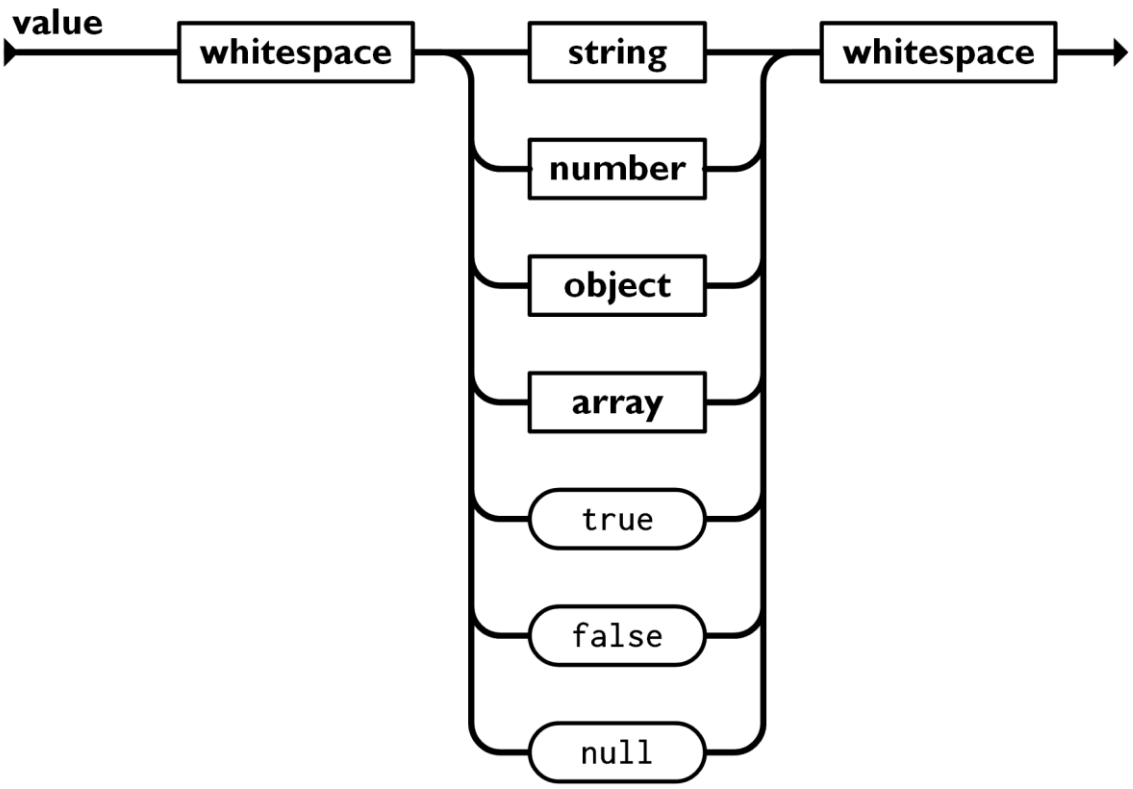


Figura 2-V - JSON value notation

Por fim um *array*, ou lista, é um conjunto de valores também separados por "," e começam e acabam com [].

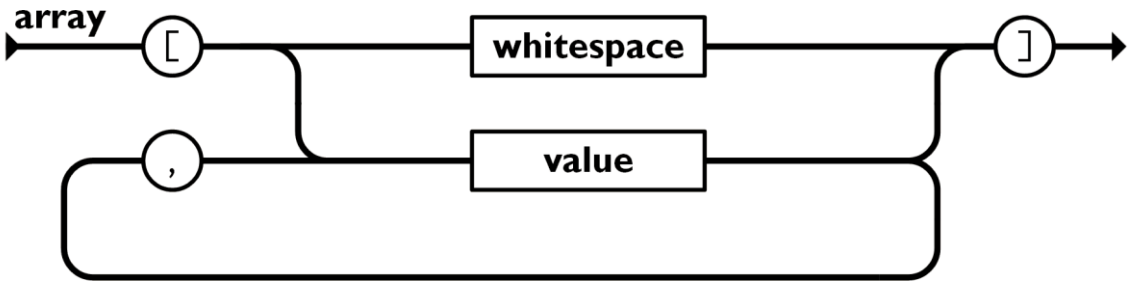


Figura 2-VI JSON array notation

2.5.4 YML

YML, ou YAML, é um acrónimo recursivo para "YAML Ain't Markup Language" tal como JSON é uma linguagem de serialização legível por humanos. Habitualmente é usado como ficheiro de configuração e em aplicações em que os dados são armazenados ou enviados. A sua documentação pode ser lida em [12]. A sintaxe desta linguagem é muito semelhante à de JSON onde usa "[]" para representar listas, "{}" para representar mapas, entretanto usa também a indentação como *Python* para definir atributos dentro de atributos.

Normalmente o resultado de um processamento de YML é um objeto tipo mapa, dicionário ou *hashes,* porém é possível ser outro.

Outra vantagem que o YML contém é a possibilidade de definir um objeto através de outro, usando referencias sendo assim possível definir um objeto e utilizá-lo dentro de outros.

2.5.5 Swagger

Por fim, o Swagger este tem como objetivo de contratualizar o BackEnd com o FrontEnd. A sua documentação pode ser lida no seu site oficial [13]. Existem várias formas de utilizar este desde a criação de anotações em Java ou a criar um ficheiro de configuração em JSON ou YML. No FSIM é criado um ficheiro YML e usado um script chamada *Swagger generator* que converte o YML em objetos java e numa API REST java no BackEnd que terá de ser implementada futuramente, por outro lado cria serviços em *TypeScript* no lado do FrontEnd, *TypeScript* visto que é linguagem usada pelo FrontEnd, no entanto o *Swagger generator* esta preparado para gerar na maior parte de linguagem existentes. Sendo que no lado no BackEnd terá sempre de ser implementado e no FrontEnd apenas são chamadas de serviços.

Através da documentação e do exemplo encontrado em [14] podemos entender que um ficheiro Swagger é divido em 3 partes, a primeira refere configuração padrão e descrições da API. De seguida encontra-se as *uris/paths*, estas são parte de um URL, onde a parte inicial está definida na primeira parte, que será um pedido REST. Nestes pedidos é definido todo o necessário, desde o que ele precisa como o que irá retornar. Por fim a última parte onde é definido objetos que serão usados nos pedidos.

3 O Projeto e Arquitetura

Este capítulo encontra-se omisso por motivos de confidencialidade.

4 Tarefas

Este capítulo encontra-se omisso por motivos de confidencialidade.

5 Flowable no Projeto

Este capítulo encontra-se omisso por motivos de confidencialidade.

6 Projeto posterior ao FSIM

Este capítulo encontra-se omisso por motivos de confidencialidade.

7 Conclusão e Trabalho Futuro

Conforme o planeado o módulo de TSK e de WKF no projeto FSIM foram finalizados e cumpriram os requisitos.

Para além disto foi possível notar estes 2 módulos num outro projeto de raiz e como eles reagiram. Começando pelo TSK notou-se um tempo excessivo na criação das configurações das tarefas e mesmo respeitando os requisitos no FSIM a intenção que o FrontEnd tinha no projeto 2 não eram compatíveis originando uma mudança que tinha logica, as novas chamadas HTTPS de mudança de estado das tarefas. O *bug* encontrado neste módulo e a sua resolução também deu a entender a complexidade desnecessária que o TSK possuía.

Posto isto para trabalho futuro do TSK podemos referir 3 possíveis soluções futuras ao módulo. Primeiro seria uma possível restruturação do código do módulo TSK devido a uma má definição de como este estaria, isto é, certos serviços possuem demasiada responsabilidade e outros quase nenhuma e são apenas utilizados uma vez, o maior problema nisto é que sempre que encontrado um bug ou a necessidade de fazer uma melhoria é perdido demasiado tempo a entender o que é necessário modificar. Isto foi notado quando foi necessário fazer as alterações para o segundo projeto.

Outra melhoria seria à API do TSK pois esta limita muito o conhecimento que o FrontEnd tem realizando as dificuldades que foram representadas acima.

Por fim a maior solução seria a criação de um micro serviço que convertia XML para YML, mais apropriadamente, uma maneira de através dos desenhos dos *flows* das tarefas em XML ser possível criar o YML das configurações das tarefas, isto porque foi notado um tempo excessivo na criação das configurações das tarefas no segundo projeto, mesmo tendo sido feito por um programador e como nestes 2 projetos houve a necessidade de desenhar estas para definir os requisitos da aplicação seria de aproveitar para uma redução de tempo de implementação em novos projetos.

Por outro lado, o WKF não apresentou problemas na incorporação com o novo projeto, além de a sua utilização aparentou ser bastante simples, tendo desconhecimento de como o código se encontra não posso afirmar se possuí um problema semelhante ao TSK no entanto todos os requisitos foram cumpridos, mesmo em projetos diferentes.

8 Bibliografia

[1] “Workflow - Wikipedia,” 24 Novembro 2019. [Online]. Available: en.wikipedia.org/wiki/Workflow. [Acedido em 11 Dezembro 2019].

[2] “Business process management - Wikipedia,” 11 Novembro 2019. [Online]. Available: https://en.wikipedia.org/wiki/Business_process_management. [Acedido em 11 Dezembro 2019].

[3] M. H. B. a. A. H. E.R., “Model for BPM implementation assessment: evidence from companies in Indonesia,” em *Business Process Management Journal*, Emerald Publishing Limited, 2019, pp. Vol. 25 No. 5, pp 825-859.

[4] “Business Process Model and Notation - Wikipedia,” 02 Dezembro 2019. [Online]. Available: https://en.wikipedia.org/wiki/Business_Process_Model_and_Notation. [Acedido em 11 Dezembro 2019].

[5] Flowable, “Java Business Process Engines | Flowable,” 2019. [Online]. Available: https://www.flowable.org/. [Acedido em 11 Dezembro 2019].

[6] DE NEGÓCIO, Gerenciamento de Processos, Brasil: BPM CBOK, 2013.

[7] E. B. Wondershare, “ERP Management BPMN | Free ERP Management BPMN Templates,” Wondershare, 2020. [Online]. Available: https://www.edrawsoft.com/template-erp-management-bpmn.html.

[8] Angular, “Angular - Introduction to the Angular Docs,” Google, 2010. [Online]. Available: https://angular.io/start. [Acedido em 2019 Dezembro 05].

[9] R. Johnson, The Spring Framework - Reference Documentation, USA: Spring Framework, 2004, p. 502.

[10] P. G. D. Group, “PostgreSQL: PostgreSQL 12 Released!,” PostGreSQL, 03 Outubro 2019. [Online]. Available: https://www.postgresql.org/about/news/1976/. [Acedido em 06 Dezembro 2019].

[11] JSON, “Introducing JSON,” [Online]. Available: https://www.json.org/json-en.html. [Acedido em 06 Dezembro 2019].

[12] “The Official YAML Web Site,” YAML, [Online]. Available: https://yaml.org/. [Acedido em 06 Dezembro 2019].

[13] S. Software, “The Best APIs are Built with Swagger Tools | Swagger,” SmartBear Software, [Online]. Available: https://swagger.io/. [Acedido em 06 Dezembro 2019].

[14] “Swagger Editor,” Swagger, [Online]. Available: https://editor.swagger.io/.

[15] OMG, Business Process Model And Notation, v2.0, 2011.

[16] Spring Data Core, “CrudRepository (Spring Data Core 2.3.4.RELEASE API),” Pivotal Software, Inc., 2011–2020. [Online]. Available: https://docs.spring.io/spring-data/commons/docs/current/api/org/springframework/data/repository/CrudRepository.html.

[17] LOMBOK - v1.18.12, “Overview (Lombok),” The Project Lombok Authors, 2009-2020. [Online]. Available: https://projectlombok.org/api/.

[18] A. G. S. D. F. H. Gunnar Morling, “MapStruct 1.4.1.Final Reference Guide,” MapStruct community, 11 10 2020. [Online].

[19] S. Softwear, “API Code & Client Generator | Swagger Codegen,” SmartBear Softwear, 2020. [Online]. Available: https://swagger.io/tools/swagger-codegen/.