**Pré-Requisitos**

1. Campo Actor no BPMN não pode ser nulo
2. O nome dos parâmetros na modelagem do processo tem que ter “easybpms” + o nome do pacote
3. O parâmetro de entrada de todas as atividades é igual ao id da entidade de domínio da aplicação
4. O id do processo no bpmn tem que igual ao nome da classe da entidade de domínio que inicia o processo
5. O parâmetro que verifica se o processo foi iniciado no bpms é o id da entidade de domínio
6. As variáveis do processo precisam ter o nome do pacote + nome da classe + atributo
7. É necessário alterar o id do processo no bpmn a cada versão do processo
8. Os valores das variáveis do processo devem ser inicializados na aplicação antes de iniciar o processo para que sejam mapeados para os parâmetros de entrada das atividades durante a execução. Aqueles que não foram setados ainda, inserir null.
9. O observador de tarefa executada é identificado por meio do parâmetro de entrada de cada atividade do processo. Assim, cada tarefa terá apenas um observador. Obs: Todos os parâmetros de entrada de uma atividade têm que corresponder a mesma entidade de domínio na aplicação, pois somente um parâmetro é escolhido da lista para identificar a entidade que é observável
10. A primeira atividade do processo não pode ser uma atividade de usuário, pois é a atividade que inicia o processo. Ou seja, quando cadastra uma viagem e faz o primeiro notifyObservers, o bpms inicia o processo e já para na próxima atividade de usuário que é a atividade “Analisar Solicitação de Viagem” por exemplo
11. Para gerar o Context é necessário criar o pacote com.easybpms.codegen. A primeira caixa de diálogo consiste em buscar o caminho onde estão armazenados os processos bpmn e a segunda caixa de diálogo consiste em selecionar o caminho do pacote com.easybpms.codegen criado.
12. Service Task – Defina uma interface nas propriedades do processo, que corresponde à classe que implementa o serviço. Para essa interface insira como operação o método da classe que será chamado quando chegar nessa atividade. O parâmetro do método é definido como uma mensagem de entrada (In Message). O retorno do método é definido como uma mensagem de saída (Out Message). Nas propriedades da Service Task, defina a implementação como “Java” e na parte de operação, selecione a operação criada para a interface definida nas propriedades do processo.
13. ServiceTask - Limitação da ferramenta easybpms – O método da classe de serviço da aplicação pode te apenas um parâmetro.
14. ServiceTask – O tipo da mensagem de entrada na operação do método deve ser o mesmo tipo que o parâmetro do método em Java. O tipo da mensagem de saída na operação do método deve ser o mesmo tipo que o retorno do método em Java. Atualmente a api suporta um parâmetro de entrada e um parâmetro de saída na ServiceTask. O mapeamento de parâmetros de entrada e saída da ServiceTask devem ser para Parameter e Result respectivamente
15. SendTask e ManualTask – somente completa os workItens. Não tem mais suporte
16. A contagem das instâncias de processo a serem persistidas no banco de dados é iniciada toda vez que desligar o computador. Então é importante zerar os bancos (jBPM e easybpms) ao desligar o PC para que as instâncias fiquem coerentes nos dois bancos, ou seja, quando criar uma instância de processo no jBPM a mesma instância será criada no easybpms.
17. Tenancy (easybpms-core-txlocal) – Criar a property tenancy (nomedopacote.nomedaclasse.tenancy). Criar o parâmetro de entrada tenancy para cada atividade (easybpms.nomedopacote.nomedaclasse.tenancy). Criar o atributo tenancy para cada entidade de domínio que interage com o processo. Implementar a interface de usuário do easybpms e setar o tenancy a qual o usuário pertence.
18. Tenancy (easybpms-core-txlocal) - Limitação da ferramenta easybpms - É necessário ter pelo menos um usuário criado e o tenancy no qual ele pertence antes de iniciar o processo, para que a primeira instância atividade possa ser atribuída ao usuário cadastrado.
19. É necessário declarar todos os atributos de todas as entidades nas variáveis de processo para que possa fazer o mapeamento com os parâmetros de entrada e saída
20. No Start Process Observer, somente os valores dos atributos da entidade de domínio que inicia o processo serão capturados e mapeados para as variáveis de processo. Se as atividades de usuário não possuírem o parâmetro de entrada igual ao id da entidade que inicia o processo, quando o BPMS chegar nessa atividade, o mapeamento da variável de processo id para o parâmetro de entrada id não será realizado. Assim, quando o Task Executed Observer for buscar no BD a tarefa pendente, ele não vai encontrar a instância parâmetro de entrada correspondente ao id da entidade. E, portanto, não vai executar a atividade.
21. Não é possível ter um observável (entidade de domínio no parâmetro de entrada) diferente para cada tarefa. Explicação: Quando inicia o processo, apenas o observador Start Process é notificado. Ele, por sua vez, vai captar somente os valores dos atributos da sua respectiva entidade de domínio. Se a próxima tarefa tiver um parâmetro de entrada correspondente a outra entidade, quando for executar essa tarefa, no BD do easybpms vai ter armazenado somente o valor do atributo id da entidade que iniciou o processo. O id da entidade que executou a tarefa é recuperado no Task Executed. E, portanto, ele não está armazenado como parâmetro de entrada
22. Atualmente a API EasyBPMS só suporta a manipulação de apenas uma entidade de domínio durante toda a execução do processo