

Examen Práctico – Corte III

Esta evaluación es una extensión al proceso de negocio Fallas técnicas de EMSA. Este fue el caso que utilizamos para introducir el uso de la máquina de Camunda embebida en Spring Boot.

Junto con este enunciado usted encuentra dos folders:

- **tecFailures**: Contiene el proyecto de fallas técnicas en EMSA.
- **FSM-Service**: Es el proyecto del sistema para gestión de tickets. Este es el sistema externo que se desea integrar al proceso de fallas técnicas.

Actividad a realizar: Para este examen, usted debe completar las extensiones realizadas a este caso de estudio, las cuales se describen a continuación.

Entrega: Una vez realizadas las modificaciones, deberá crear un repositorio público en GitHub con el proyecto y en el aula virtual colocar el enlace a este repositorio.

A. Generación de ticket para una falla técnica.

En el sistema BPM de gestión de fallas técnicas se integrará ahora la gestión de un ticket para cada falla técnica recibida. La generación de estos tickets se hace con un sistema externo (**FSM-Service**), el cual encuentra en la carpeta del mismo nombre.

En la carpeta resources del proyecto **tecFailures** encuentra el modelo BPMN modificado y con algunos pendientes que usted debe completar.

Lea detenidamente y complete lo que a continuación se le indica:

1. (10) En la Service task “Generar Ticket de Visita en FSM”, que integra al sistema externo “FSM-Service” (a través de una implementación Connector) completar el envío al sistema externo de la información recolectada en el formulario que implementa la User task “Agendar Visita Técnica”.
2. (15) Dentro del call activity “Realizar visita técnica” ajuste los Connector Inputs del Service task “Actualizar estado del ticket de Visita en FSM” (que tiene implementación Connector) de modo que realice correctamente la petición a la url “/actualizar” del sistema externo FSM-Service y capture la respuesta de la petición en una nueva variable llamada RespuestaFSM.

Tenga en cuenta el alcance de variables en un modelo BPMN, ya que:

- La variable TicketIdVisita es externa al subprocesso “realizar visita técnica” (debe usar “In mappings”)
- La variable RespuestaFSM es interna en este subprocesso (debe usar Out mappings)

3. (5) Muestre la variable de respuesta del sistema externo capturada en el punto anterior en un nuevo campo del formulario que implementa la user task “Generar Factura de Pago”. Coloque a este nuevo input el label “Estado del Ticket”.
4. (2) Despliegue el proyecto y verifique su funcionamiento. Si es del caso realice los ajustes necesarios.

B. Información sobre el pago

Es necesario mejorar el manejo del pago recibido vs. el pago total cobrado por la atención de una falla técnica, así que, para visualizar estos valores, se ha decidido que usted desarrolle los siguientes ajustes:

5. (10) Agregue una nueva actividad de tipo “Service task” con implementación “Java Class” llamado “CalculateBalance” que calcule el saldo del cliente después de la tarea “Recibir Pago”. Esta clase, que implementa un *JavaDelegate*, debe determinar si existe un saldo a favor del cliente o un saldo faltante de pago (comparando ValorTotal con ValorPagado), y devolver las siguientes variables al proceso en Camunda:
 - **estadoPago** → String que contiene el texto "Saldo a favor", "Saldo faltante" o "Completo", según sea el caso.
 - **saldoCliente** → Valor numérico que representa el saldo faltante, el saldo a favor o cero en caso de que el pago haya sido completo.
6. (5) Cree una nueva actividad de tipo “User task” que muestre un resumen del pago utilizando “Generated Task Forms”. En el formulario deben mostrarse:
 - **estadoPago** → el obtenido en la Service Task anterior (CalculateBalance)
 - **saldoCliente** → el obtenido en la Service Task anterior (CalculateBalance)
7. (3) Despliegue el proyecto y verifique su funcionamiento. Si es del caso realice los ajustes necesarios.