Desafío: Proceso de Solicitud de Consulta Médica con Camunda 8 y Spring Boot

Contexto General

Una **obra social** recibe solicitudes de turnos médicos de sus afiliados. Cada solicitud incluye los siguientes datos:

- num_socio (Número de socio)
- especialidad (Especialidad médica solicitada)
- motivo (Motivo de la consulta)

El **proceso de negocio** debe cumplir los siguientes pasos:

- Verificar si el paciente está habilitado para recibir el servicio.
- Consultar la disponibilidad de turnos para la especialidad.
- Confirmar y registrar el turno si es posible.
- Notificar al paciente en caso de rechazo o no disponibilidad.
- Manejar errores de negocio y técnicos de forma diferenciada.

Este desafío debe usar como base el Ejercicio 3.1: Validación de Datos y Cobertura:

Link a los archivos de ejemplo

@ Requisitos del Desafío

🍯 Modelado BPMN

- Modelar el proceso de solicitud de consulta médica en Camunda Modeler.
- Usar correctamente Service Tasks con type asignado
- Incluir pruebas curl de los distintos caminos y revisión en operate
- Incluir manejo de errores diferenciando:
 - X Errores de negocio (sin reintentos, fallo del proceso).
 - o A Errores técnicos (con reintentos configurados en Retries).

Tareas, Job Workers y Errores

Worker	% Tipo de Servicio	Variables Esperadas / Resultados	Errores de Negocio (sin reintentos)	Errores Técnicos (reintentos posibles)
verificarCober tura	Validar cobertura	num_socio → apto: true/false	Socio no registrado, Socio con deuda	API caída, Timeout, Error DB
revisarAgenda	Consultar turnos	especialidad, motivo → turnoDisponible, fecha_turno	Sin turnos disponibles	API caída, Timeout, Error DB
confirmarTurno	Registrar turno	num_socio, fecha_turno → Confirmación	Turno duplicado, Fecha inválida	Error DB, Timeout
notificarSinTu rno	Notificar sin turno	num_socio → Notificación enviada	Email inválido	SMTP caído, Error API notificación
notificarRecha zo	Notificar rechazo	num_socio → Notificación enviada	Email inválido	SMTP caído, Error API notificación

- Errores de Negocio: No apto, sin turnos, turno duplicado → X Fallar el Job (failJob), sin reintentos.
- **Simulación de errores**: Se espera que se simulen errores en los Workers para probar los distintos caminos.

Wariables del Proceso (Ejemplo JSON)

```
{
  "num_socio": "123456",
  "especialidad": "Cardiología",
  "motivo": "Chequeo general",
  "apto": true,
  "turnoDisponible": true,
  "fecha_turno": "2025-06-01T10:00:00"
}
```

📆 Entregables del Desafío

- Diagrama BPMN funcional con todos los elementos modelados.
- Implementación en **Spring Boot** de los Job Workers con @JobWorker y manejo de errores.
- Archivo test. http con las solicitudes Rest que contemplen los distintos caminos Simulación de errores de negocio y técnicos usando **Operate**.
- Configuración de reintentos (Retries = 3) para errores técnicos.
- Opcional: Uso de **Boundary Events** para manejo avanzado de errores y timeouts.

Criterios de Evaluación

Aspecto a Evaluar	Excelente (4 pts)	Bueno (3 pts)	Aceptable (2 pts)	Insuficiente (1 pt)
Modelado BPMN	Diagrama claro, completo, usa correctamente Service Tasks, errores diferenciados, eventos y flujos.	Diagrama funcional pero con errores menores o faltantes.	Diagrama incompleto o con errores de estructura.	Diagrama ausente o incorrecto.
Job Workers (Java + Spring Boot)	Implementa todos los Workers con @JobWorker, suscripciones correctas, manejo de errores, código claro y comentado.	Workers funcionales, pero con fallas menores o poca documentación.	Workers incompletos o con errores en suscriptores/proc esamiento.	Workers ausentes o con errores críticos.

Variables del Proceso	Variables bien gestionadas, mapeadas y utilizadas en los Workers. JSON de ejemplo válido.	Uso parcial de variables, faltan algunas.	Uso limitado, errores o variables clave no utilizadas.	Variables ausentes o mal implementada s.
Errores de Negocio	Correcto manejo, sin reintentos. Workers fallan correctamente en estos casos.	Parcialmente implementados o con errores menores.	Lógica incompleta o con fallas.	Sin manejo diferenciado o incorrecto.
Errores Técnicos y Reintentos	Reintentos configurados, errores simulados correctamente. Uso de failJob.	Reintentos parciales o errores en la lógica.	Reintentos no implementados correctamente.	No implementa reintentos ni lógica para errores técnicos.
Pruebas en Operate	Pruebas completas: caminos felices, errores de negocio y errores técnicos. Evidencias claras.	Pruebas parciales, algunos casos omitidos.	Pruebas mínimas o incompletas.	No realiza pruebas o no documenta resultados.

📊 Escala de Puntuación Final

Puntuación	Nivel de Logro	Descripción
24 - 28 pts	★ Excelente	Domina conceptos clave, código funcional, pruebas completas.
19 - 23 pts	≜ Bueno	Cumple con lo requerido, pequeños errores a corregir.
14 - 18 pts	Aceptable	Proyecto funcional, pero con limitaciones claras.
<14 pts	insuficiente	Faltan entregables clave, errores críticos, o falta de comprensión del flujo.



Camunda. (s.f.). *Gateways in BPMN*. Recuperado el **14 de mayo de 2025**, de https://docs.camunda.io/docs/components/concepts/job-workers/

Camunda Academy. (s.f.). *BPMN Course Material and Certification*. Recuperado el **14 de mayo de 2025**, de https://academy.camunda.com/c8-develop-workers-spring

Rücker, B. (2021). Writing Good Workers for Camunda Cloud. Recuperado de https://blog.bernd-ruecker.com/writing-good-workers-for-camunda-cloud-61d322cad862

Camunda Academy. (s.f.). Recuperado el 14 de mayo de 2025, de https://academy.camunda.com/