Diseño de Sistemas. Examen Final.
Fecha: 08/02/2017

Apellido y Nombre:..... Legajo:.....

Docente con quien cursó la materia: ……

Contexto: Un Banco de Datos Meteorológicos (BDM) almacena observaciones y mediciones atmosféricas de diferentes fuentes y formatos. Esas fuentes (las estaciones meteorológicas) pueden emitir mensajes METAR, SYNOP, TAF, entre otros. El formato de los mensajes está definido por estándares internacionales y son cadenas de texto fácilmente interpretables a través de expresiones regulares. Un ejemplo de mensaje METAR[1]:

```
METAR KORD 041656Z 19020G26KT
6SM BKN070 12/08 A3016
```

Donde el primer campo especifica el tipo de mensaje, el segundo el es ID de la estación de origen, el tercero corresponde a la fecha-hora de la medición, el cuarto las características del viento, el quinto sobre visibilidad, el sexto sobre las condiciones del cielo, el séptimo temperatura/punto de rocío y el octavo presión.
Los mensajes pueden ser transmitidos automáticamente por estaciones meteorológicas o manualmente por un operador a través de un email.

Proyecto-BDM: Usted en el rol de arquitecto debe diseñar un sistema que, además de dar soporte a la recepción masiva de mensajes, pueda almacenarlos en un medio relacional y a su vez permita hacer consultas a través de una interfaz Web. A continuación una breve descripción de los requerimientos funcionales y no funcionales solicitados por el referente técnico del cliente y los usuarios:

- **Interfaces de entrada:** El sistema expondrá un componente REST a través del cual recibirá las cadenas de texto que conforman el mensaje. También recibirá esos mensajes meteorológicos a través de una casilla de email.
- **Interfaz de consulta:** El sistema permitirá hacer consultas por tipo de mensaje, origen, rango temporal, mediciones y otros criterios. Los resultados de las consultas podrán ser exportables en formato xml, xlsx, pdf y csv.
- **Integración:** Debido a la gran diversidad de tecnologías de conectividad en las estaciones meteorológicas, el sistema debe garantizar respuesta a toda solicitud, sin contemplar la posibilidad de retransmisión por parte del emisor. Por lo tanto, ante la posible masividad en el tráfico de mensajes, el sistema debe asegurar alta disponibilidad y aceptable desempeño de forma permanente.

[1] Fuente: https://www.wunderground.com/metarFAQ.asp

- **Seguridad:** El acceso a través de la interfaz Web estará limitado a usuarios autorizados a realizar consultas, a su vez, el sistema permitirá gestión remota a usuarios con el rol de administrador.
- **Auditoría:** En caso que el proceso vigía (watchdog[2]) detecte un problema general o parcial, éste deberá notificar por SMS al número de celular asociado a los usuarios administradores que estén suscriptos a esa lista de distribución.

Se pide:

1. Arquitectura

Proponer una arquitectura del sistema a alto nivel y comunicarla preferentemente con diagramas. Si considera apropiado tomar como referencia patrones y estilos arquitectónicos y/o de integración, indicarlos y justificar su uso. Debe quedar claro:

- Componentes de alto nivel.
- Sistemas externos.
- Comunicación entre componentes.

2. Modelo de Dominio

Modelar el dominio con el paradigma de orientación a objetos y comunicarlo con un diagrama de clases. El modelo debe contemplar todos los requerimientos del caso. Si considera apropiado utilizar patrones de diseño, indicarlos y justificar su uso.

3. Persistencia relacional

Utilizando un DER, código anotado y/o prosa, se pide explicar cómo persistir el modelo de objetos del punto anterior en una base de datos relacional, indicando claramente:

- Qué elementos del modelo es necesario persistir.
- Las claves primarias, las foráneas y las restricciones (constraints) según corresponda.
- Estrategias de mapeo de herencia utilizadas, si fueran necesarias.
- Las estructuras de datos que deban ser desnormalizadas, si corresponde.
- Justificaciones sobre las decisiones de diseño tomadas anteriormente.

Explicar supuestos y justificar decisiones de diseño.

[2] Es un componente que vigila el procesamiento de otro componente. Los watchdog habitualmente se emplean en sistemas de tiempo real para asegurarse que el procesamiento dependiente del tiempo esté activo. *Douglass, Real-Time Design Patterns.*