

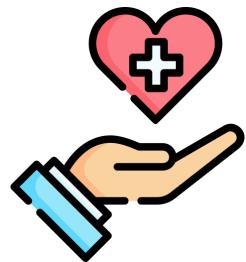
Condiciones de aprobación: Para aprobar debe sumar como mínimo 60 puntos y no menos del 50% en cada sección. Se puede recuperar por partes (teoría, objetos, arquitectura), pero no para la aprobación directa; en ese caso se deberá rendir el parcial completo nuevamente. Arquitectura se recuperará en conjunto primer y segundo parcial (Arq. parte I, Arq. parte II). Se podrá recuperar dos veces cada eje de la asignatura (objetos, datos, arquitectura).

SISALUD

Contexto general

Estamos diseñando y desarrollando SISALUD, una plataforma web y móvil impulsada por el Gobierno de la Ciudad de Buenos Aires. Su objetivo es anticiparse a eventos sanitarios relevantes para la salud pública (brotes de enfermedades respiratorias, alerta por contaminación del aire o agua, falta de vacunación, entre otros), y ofrecer a los ciudadanos información confiable, recomendaciones personalizadas y herramientas de seguimiento.

La solución está compuesta por múltiples servicios distribuidos. En este examen, nos enfocaremos en el diseño del servicio de “Alertas” y servicio de “Ciudadanos y Perfil Sanitario”. El servicio de **Encuestas Sanitarias y Feedback** ya fue desarrollado, y se encuentra documentado. El resto de los componentes son servicios preexistentes o que están en proceso de desarrollo por otros equipos. En ambos casos, debemos integrarnos a ellos cuando lo necesitemos.



Servicio de Ciudadanos y Perfil Sanitario (a desarrollar)

Este servicio debe:

- Permitir el registro y gestión de ciudadanos, incluyendo:
 - Datos personales: nombre, apellido, DNI, email
 - Ubicación geográfica: principal y secundaria (coordenadas, barrio, tipo de zona)
 - Perfil sanitario, compuesto por:
 - Grupo etario
 - Esquema de vacunación, indicando:
 - Tipo de vacuna (por ejemplo: antigripal, COVID, hepatitis)
 - Estado (completa, incompleta, vencida)
 - Fecha de última dosis
 - Factores de riesgo, los cuales deben:
 - Estar asociados a un tipo de evento sanitario (respiratorio, ambiental, alimentario)
 - Contemplar el grado de sensibilidad del ciudadano a ese tipo de evento (leve, moderado, alto)
- Permitir registrar y consultar cambios de ubicación de los ciudadanos (por ejemplo, mudanzas temporales o ubicación laboral).
- Permitir que el ciudadano actualice su información sanitaria, manteniendo trazabilidad histórica cuando sea necesario (opcional).
- Permitir realizar búsquedas por ubicación geográfica usando diferentes estrategias de cálculo de cercanía, como:
 - Por radio en kilómetros desde un punto
 - Por pertenencia a una zona geográfica (por ejemplo: barrios críticos predefinidos)
 - Por intersección con áreas afectadas (representadas como polígonos)

Estas búsquedas deben ser configurables para adaptarse a distintos servicios consumidores (por ejemplo, el servicio de Alertas o Estadísticas).

Servicio de Sanidad Proxy (ya desarrollado)

Este servicio consulta a múltiples fuentes externas como el Ministerio de Salud y la OMS (vía REST - pull based). Su responsabilidad es consolidar los datos y publicar eventos sanitarios unificados.

El servicio Sanidad Proxy expone un endpoint para que otros servicios se suscriban a novedades por tópico mediante webhooks. Es decir, otros servicios pueden registrar una URL que será invocada cuando ocurra un nuevo evento.

Ejemplo de tópicos disponibles:

- /eventos/brote-respiratorio
- /eventos/alerta-ambiental
- /eventos/toxina-alimentaria

Ejemplo de notificación enviada vía webhook:

```
{
  "tipoEvento": "brote-respiratorio",
  "fechaInicio": "2025-07-28T03:00:00Z",
  "zona": { "barrio": "Caballito" },
  "nivel": "critico"
}
```

Servicio de Alertas Sanitarias (a desarrollar)

Este servicio debe:

- Suscribirse a eventos sanitarios relevantes vía webhook (desde el servicio de Sanidad Proxy).
- Consultar al Servicio de Ciudadanos y Perfil Sanitario para identificar a los ciudadanos potencialmente afectados. El filtrado se basa en:
 - Ubicación (coordenadas o barrio)
 - Factores de riesgo asociados al tipo de evento sanitario (por ejemplo: enfermedades respiratorias, exposición alimentaria o ambiental)
- Generar recomendaciones personalizadas para los eventos que ocurrirán en las próximas 24 hs, siempre que la severidad sea “alta”, “crítica” o “extrema”. Las recomendaciones deben considerar:
 - El tipo de evento sanitario
 - El perfil sanitario del ciudadano (grupo etario, esquema de vacunación, factores de riesgo)
- Consultar a una API externa GPT4-Salud (mediante POST), enviando los datos del evento y del perfil ciudadano, para enriquecer el contenido de las recomendaciones.
- Enviar alertas por:
 - Notificación push (usando un servicio de mensajería interno)
 - Email (mediante servicio SMTP)
 - Cada alerta debería dar aviso de la situación actual a los usuarios afectados.
- Disparar webhooks de emergencia a organismos externos registrados (ej. SAME, Hospitales, Defensa Civil) cuando el nivel del evento sea “extremo”.
- Activar alarmas sanitarias físicas (por ejemplo, sirenas en centros barriales) a través del servicio de Estaciones de Monitoreo Sanitario, cuando el nivel sea “crítico” o “extremo”. Este umbral debe ser configurable.

Servicio de Estaciones Ambientales Inteligentes (ya desarrollado)

Existen actualmente unidades fijas distribuidas por la ciudad que monitorean la calidad del aire y la presencia de agentes ambientales perjudiciales. Estas unidades se denominan estaciones ambientales inteligentes. El servicio que las gestiona fue desarrollado por otro proveedor y se conecta mediante un broker de eventos, utilizando un modelo Pub/Sub.

- **estacion.datos_calidad_aire**: incluye indicadores como CO₂, PM2.5, PM10, ozono, dióxido de nitrógeno, etc.
- **estacion.presencia_alergenos**: reporta la presencia de elementos como polen, esporas o agentes irritantes en niveles críticos.
- **estacion.alerta_manual**: las estaciones pueden emitir alertas activadas manualmente en el lugar, por personal técnico, municipal o sanitario, cuando se detecta una situación de riesgo no captada automáticamente por sensores. Este evento puede ser consumido por otros servicios para auditoría, trazabilidad o complementación de alertas automáticas.

El Servicio de Alertas Sanitarias puede suscribirse a estos eventos si lo desea, por ejemplo, para reforzar o validar las recomendaciones a emitir en determinadas zonas.

Además, el servicio ofrece una API REST para activar alarmas físicas en una estación determinada (sirenas, luces, pantallas), en caso de que se necesite alertar presencialmente a los ciudadanos en el área afectada.

- POST /estaciones/{id}/activar-alarma
- Ejemplo de cuerpo JSON:

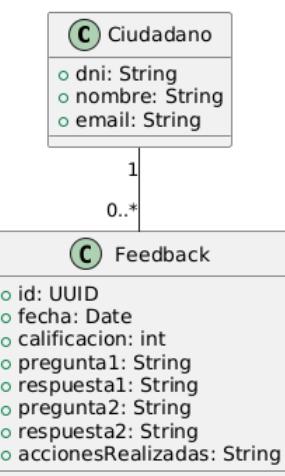

```
{
        "nivel": "critico",
        "duracion_segundos": 90,
        "componentes": ["sirena", "pantalla"]
      }
```
- Valores posibles para nivel: "preventivo", "advertencia", "critico"
- Componentes: "sirena", "pantalla", "luz"

Servicio de Encuestas Sanitarias y Feedback (ya desarrollado)

Este servicio permite que los ciudadanos brinden información posterior a una alerta sanitaria. Incluye:

- Calificación de 1 a 5 sobre el contenido recibido.
- Preguntas específicas sobre su experiencia durante el evento sanitario (por ejemplo: "¿Usó barbijo?", "¿Consultó a un médico?", "¿Evitó salir a la calle?").
- Registro de acciones preventivas tomadas.
- Capacidad para que administradores gestionen las preguntas disponibles.

El diseño de este servicio ya fue implementado por otros integrantes del equipo, quienes nos brindaron el diagrama de clases que representa su solución (ver al costado).



Servicio de Estadísticas (no desarrollado)

Este servicio no debe ser implementado, pero debe ser tenido en cuenta para exponer los datos recolectados por Ciudadanos y Feedback.

Alcance y requerimientos (de nuestros servicios)

El Sistema deberá:

1. Permitir el registro y gestión de ciudadanos (Servicio de Ciudadanos y Perfil Sanitario).
2. Consultar ciudadanos registrados a través del servicio de Ciudadanos y Perfil Sanitario, filtrando por ubicación y tipo de factor de riesgo. (Servicio de Ciudadanos y Perfil Sanitario).
3. Recibir eventos sanitarios relevantes vía webhook desde el servicio de Sanidad Proxy. (Servicio de Alertas Sanitarias).
4. Generar recomendaciones personalizadas para los eventos que ocurrirán en las próximas 24 hs (Servicio de Alertas Sanitarias).
5. Consultar a una API externa GPT4-Salud para enriquecer el contenido de las recomendaciones.
6. Enviar alertas sanitarias por notificación push y correo electrónico
7. Disparar webhooks de emergencia a organismos externos registrados cuando el nivel del evento sea "extremo". (Servicio de Alertas Sanitarias)
8. Activar alarmas sanitarias físicas (por ejemplo, sirenas en centros barriales) a través del servicio de Estaciones de Monitoreo Sanitario. (Servicio de Alertas Sanitarias)
9. Exponer información relevante a través de interfaces para su posterior uso por el servicio de Estadísticas. (Ambos Servicios)

Punto 1 - Modelo de Dominio (50 puntos)

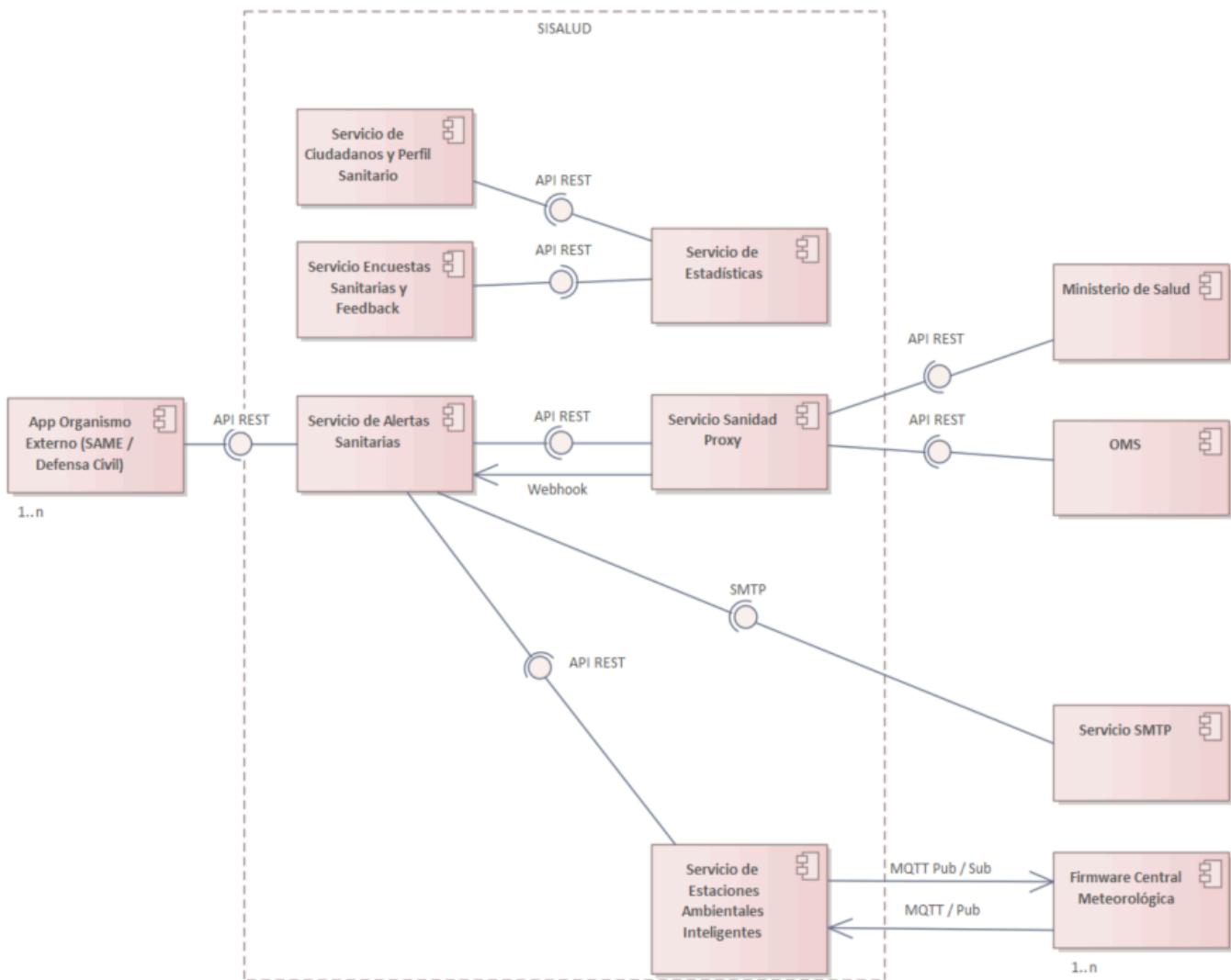
1. **(30 puntos) Servicio de Ciudadanos y Perfil Sanitario; Servicio de Alertas Sanitarias**
 - a. **(7.5 puntos cada servicio) Para cada uno de los servicios:** documentar la solución utilizando diagramas UML (diagrama de clases obligatorio).
 - b. **(7.5 puntos cada servicio) Para cada uno de los servicios:** detallar las decisiones de diseño tomadas en el punto anterior. Mencione y haga referencia explícita a los siguientes conceptos que haya tenido en cuenta para diseñar su solución: Cualidades de Diseño y Atributos de Calidad de Software; Principios SOLID; Code Smells; Utilización de patrones de diseño (en caso que haya utilizado alguno) y comparación contra otra propuesta.
2. **(10 puntos)** Muestre mediante código o pseudocódigo cómo identificar qué ciudadanos deben recibir una alerta sanitaria de tipo brote respiratorio en el barrio de Caballito. Muestre todas las capas involucradas. No es necesario detallar la implementación de los repositorios.
3. **(10 puntos)** Teniendo en cuenta la propuesta dada del **Servicio de Encuestas Sanitarias y Feedback**, indique si las siguientes afirmaciones son Verdaderas o Falsas. Justifique cada respuesta en no más de dos renglones.
 - Representar las preguntas como atributos fijos (pregunta1, pregunta2) dentro de Feedback reduce el acoplamiento con la lógica de encuestas y permite mayor control sobre las respuestas.

El modelo incumple el principio de responsabilidad única, ya que Feedback gestiona tanto la estructura de la encuesta como el contenido de las respuestas.

Para mejorar la extensibilidad del Sistema, sería conveniente que las preguntas fueran entidades independientes (implementadas mediante una herencia), relacionadas con la entidad Feedback, permitiendo mantener un historial y controlar versiones.

Punto 1 - Arquitectura parte I (50 puntos)

- 1) (20 puntos) Dado el siguiente Diagrama de Componentes del Sistema, analícelo y describa mejoras, faltantes y puntos de mejora a considerar.



- 2) (20 puntos) ¿Cómo sabría el servicio de Alertas Sanitarias cuáles son las estaciones ambientales cercanas? ¿Qué interfaz debería ofrecer el servicio de Estaciones para poder resolverlo? Detalle completamente su respuesta.

- 3) (10 puntos) Indique si las siguientes afirmaciones son verdaderas. Justifique muy brevemente:

Si el servicio de Alertas Sanitarias necesita recibir eventos desde múltiples tópicos de Sanidad Proxy, debe registrar un webhook distinto por cada uno de ellos.

Si el servicio de Estaciones Ambientales no expone una interfaz para resolver cuál es la estación más cercana, el cálculo de cercanía debería ser asumido por el servicio consumidor.

Dividir la API Web del servicio de Ciudadanos en interfaces separadas para lectura y escritura (por ejemplo, `/ciudadanos/actualizar` vs. `/ciudadanos/busqueda`) favorece los principios REST.
