**Modelo de Arquitectura de Software**

**Sistema de Recordatorio de Cumpleaños**

1. **Requisitos del Sistema** (y Características Principales):

Gestión de Almacenamiento: El Sistema debe permitir al Usuario guardar y administrar una lista de cumpleaños, incluyendo información como Nombres y Fechas.

Notificaciones Programadas: El Sistema debe enviar notificaciones/recordatorios al Usuario el día del cumpleaños que fue almacenado. La notificación debe ser en forma de Mensaje Emergente (Notificaciones Push) (OneSignal).

Eliminar Fecha: El Sistema debe permitir al Usuario eliminar una Fecha de Cumpleaños almacenada.

Modificar Fecha: El Sistema debe permitir al Usuario modificar una Fecha de Cumpleaños, en caso de necesitar corregir una fecha o actualizarla con nueva información.

Integración con Calendarios: Sería beneficioso que el sistema pueda integrarse con calendarios (como el de Google) para sincronizar automáticamente los recordatorios de cumpleaños con el calendario del usuario.

Interfaz de usuario intuitiva: El sistema debe tener una interfaz de usuario fácil de usar, que permita al usuario agregar, modificar y eliminar recordatorios de manera sencilla.

1. **Estilo de Arquitectura**:

**Cliente/Servidor**

**Cliente**: El cliente es la interfaz de usuario a través de la cual los usuarios interactúan con el sistema. El cliente permite a los usuarios realizar acciones como agregar fecha de cumpleaños, modificar recordatorios, ver notificaciones de cumpleaños y eliminar recordatorios.

**Servidor**: El servidor es la parte central del sistema que maneja la lógica del negocio y almacena la información de los contactos y las fechas de cumpleaños. El servidor se encarga de procesar las solicitudes del cliente, como agregar, editar o eliminar recordatorios, programar recordatorios, enviar notificaciones y realizar operaciones de búsqueda en la base de datos de contactos.

Comunicación: El cliente y el servidor se comunican entre sí a través de una interfaz de programación de aplicaciones (API). El cliente envía solicitudes al servidor para realizar acciones o recuperar información, y el servidor responde con los resultados correspondientes.

**REST**:

\*Utiliza los métodos estándar de HTTP, como GET, POST, PUT y DELETE, para realizar operaciones en los recursos.

\*Se basa en la representación de los recursos utilizando formatos comunes, como JSON o XML.

\*Permite una arquitectura más ligera y escalable en comparación con SOAP, ya que no tiene una sobrecarga adicional de procesamiento y validación de mensajes XML.

El Sistema implica operaciones, como agregar, editar y eliminar contactos, programar recordatorios y enviar notificaciones. Estas operaciones son bien adecuadas para un enfoque basado en solicitudes HTTP y REST.

En este Sistema, la mayoría de las interacciones son unidireccionales, es decir, el cliente solicita información o realiza acciones en el servidor y espera respuestas. Las notificaciones de cumpleaños también son enviadas desde el servidor al cliente. Por lo tanto, un enfoque basado en solicitudes HTTP y REST es suficiente para manejar estas comunicaciones.

Base de datos: El servidor puede utilizar una base de datos para almacenar y gestionar la información de los contactos, las fechas de cumpleaños y cualquier otra información relevante.

1. **Diagrama de Arquitectura**:

