

Universidad Mariano Galv ez De Guatemala

Guatemala 24 de Agosto 2025

Arquitectura de computadoras II



Bernon Isaac Romero Leonardo

Ingenier a en Sistemas

Jornada Vespertina

Secci n A

996020160

Introducción

Diseño Arquitectónico por Componentes – Sistema IGSS Influenza

1. Diseño Arquitectónico – Sistema IGSS Influenza

El sistema propuesto es online y busca llevar el registro y control de las vacunas contra la influenza de los afiliados al IGSS.

Dado que se espera una gran cantidad de accesos simultáneos, se propone una arquitectura en capas cliente-servidor con enfoque de microservicios.

- Capa de Presentación: interfaz web y móvil para que los afiliados se registren, consulten su información y gestionen citas.
- Capa de Lógica de Negocio: microservicios separados que manejan las funciones principales (registro de afiliados, gestión de vacunas, citas, constancias, validaciones).
- Capa de Persistencia: repositorio central (base de datos relacional) para almacenar la información de afiliados, vacunas, citas y centros de vacunación.
- Capa de Seguridad: servicios de autenticación y autorización para garantizar que el acceso y los datos estén protegidos.

Este diseño modular facilita la escalabilidad y la confiabilidad, además de permitir que cada componente se pueda mantener y mejorar de manera independiente.

2. Consideración de alta demanda de acceso

El sistema debe estar preparado para atender a miles de usuarios en línea al mismo tiempo. Para ello:

- Se usará un balanceador de carga para distribuir las solicitudes entre varios servidores web.
- Los microservicios podrán escalar de manera independiente según la demanda.
- Se incorporará caché para optimizar las consultas más frecuentes (por ejemplo, centros de vacunación disponibles).
- La base de datos se diseñará con posibilidad de replicación y respaldo para mantener la disponibilidad.

3. Funcionalidades básicas

El sistema incluirá las siguientes funciones principales:

- Registrar afiliados interesados en vacunación online.
- Identificar si el afiliado requiere primera o segunda dosis.
- Validar registro de afiliados activos vs interesados (DPI, número de afiliación, nombres, apellidos).
- Mostrar centros de vacunación disponibles.
- Agendar cita de vacunación.
- Procesar pre-requisitos para la vacunación online.
- Completar formularios digitales.
- Apoyar el proceso de vacunación presencial (validación de datos en sitio).
- Registrar la vacuna aplicada (nombre, fecha, centro).
- Emitir y permitir descargar constancia de vacunación en línea.

4. Atributos de calidad (FURPS) y atributos especiales

El diseño arquitectónico se guía por los atributos de calidad:

- Funcionalidad: incluye todas las operaciones necesarias (registro, validaciones, citas, constancias).
- Usabilidad: interfaz sencilla, accesible desde web y móvil, con formularios claros.
- Confiabilidad: el sistema valida datos de DPI y afiliación, tiene mecanismos de recuperación y evita errores en la gestión de vacunas.
- Rendimiento: uso de balanceadores de carga, cacheo de consultas y servidores escalables para tiempos de respuesta rápidos.
- Mantenibilidad: al estar dividido en microservicios, cada módulo se puede actualizar o ampliar sin afectar el resto.

Atributos especiales:

- Seguridad en conexiones web: uso de HTTPS/TLS, autenticación con usuario/contraseña o token, y validación de sesiones.
- Escalabilidad en servidor web: capacidad de crecer horizontalmente con más instancias de servidor, ajustándose a la alta demanda de acceso.

1. Identificación de clases de diseño del dominio del problema

Con base en los requerimientos del sistema, se identifican las clases principales que representan las entidades y procesos del dominio:

- Afiliado: representa a la persona registrada en el IGSS, con sus datos de identificación como DPI, número de afiliación, nombres, apellidos, etc.
- Vacuna: representa el tipo de vacuna aplicada (en este caso influenza), con su nombre, fecha, centro de vacunación, dosis (primera o segunda).
- Cita: representa la cita programada por el afiliado, incluyendo fecha, hora y centro de vacunación.
- CentroVacunacion: almacena los datos de los centros habilitados para aplicar la vacuna (dirección, cupos, disponibilidad).
- Constancia: documento digital generado para el afiliado después de recibir su vacuna, con firma de validación.

Estas clases son la base del sistema porque reflejan directamente las funciones de registrar, validar, agendar y emitir constancias.

2. Identificación de clases de diseño de la infraestructura

Además de las clases del dominio, se necesitan clases para manejar la parte técnica del sistema:

- **InterfazUsuario:** componente que representa la capa de presentación, donde el afiliado se registra y agenda su cita (puede ser web o móvil).
- **ServicioAutenticacion:** maneja la seguridad de accesos, validación de usuarios y protección de datos.
- **APIControlador:** clase que recibe las peticiones de la interfaz de usuario y las redirige al componente correspondiente.
- **GestorBaseDatos:** se encarga de la comunicación con la base de datos para registrar afiliados, citas y vacunas.
- **Notificador:** servicio que envía recordatorios y confirmaciones al afiliado (correo electrónico o SMS).

Estas clases de infraestructura aseguran que el sistema sea seguro, escalable y fácil de usar.

3. Elaboración de clases no reutilizables (atributos y operaciones)

Clase	Atributos principales	Operaciones principales
Afiliado	dpi, noAfiliacion, nombres, apellidos, fechaNacimiento	registrarAfiliado(), validarAfiliado()
Vacuna	nombre, dosis, fecha, centro	registrarVacuna(), consultarHistorial()
Cita	fecha, hora, centro, estado	agendarCita(), cancelarCita(), validarCita()
CentroVacunacion	nombre, direccion, capacidad, cuposDisponibles	consultarDisponibilidad(), asignarCupo()
Constancia	codigo, fechaEmision, firmaDigital	generarConstancia(), descargarConstancia()
InterfazUsuario	tipo, diseño, formularios	mostrarFormulario(), recibirDatos()
ServicioAuth	usuario, clave, token	autenticar(), generarToken(), validarSesion()
APIControlador	endpoint, protocolo	procesarSolicitud(), enviarRespuesta()
GestorBaseDatos	conexion, consultas	guardarDatos(), leerDatos(), actualizarDatos()
Notificador	tipoNotificacion, mensaje, destinatario	enviarCorreo(), enviarSMS()

4. Fuentes persistentes de datos

El sistema requiere una base de datos centralizada para almacenar de forma persistente la información. Se plantea el uso de un modelo relacional:

- Tabla Afiliados: dpi, noAfiliacion, nombres, apellidos, fechaNacimiento, estado.
- Tabla Vacunas: idVacuna, nombre, dosis, fecha, centro, dpiAfiliado.
- Tabla Citas: idCita, fecha, hora, centro, estado, dpiAfiliado.
- Tabla CentrosVacunacion: idCentro, nombre, direccion, capacidad, cuposDisponibles.
- Tabla Constancias: idConstancia, codigo, fechaEmision, dpiAfiliado, firmaDigital.
- Tabla UsuariosSistema: idUsuario, usuario, clave, rol, token.

La base de datos funcionará como repositorio central, lo que asegura consistencia de la información.

5. Representaciones del diseño por componentes (especificación UML previa al diagrama)

Los componentes principales del sistema son:

- **Componente de Registro de Afiliados**
 - Responsabilidad: Alta de afiliados interesados en vacunación.
 - Interfaces: Formulario web/móvil, validación de DPI y número de afiliación.
- **Componente de Gestión de Vacunas**
 - Responsabilidad: Control de dosis (primera/segunda), nombre de vacuna aplicada, fecha y centro.
 - Interfaces: Operaciones de registro de dosis y consultas por historial.
- **Componente de Agenda de Citas**
 - Responsabilidad: Mostrar centros disponibles y permitir agendar cita.
 - Interfaces: Operaciones de creación, modificación y cancelación de cita.
- **Componente de Validación de Afiliados Activos**
 - Responsabilidad: Conectar con la base de datos del IGSS para confirmar estado activo.
 - Interfaces: API de validación, consulta de datos personales.
- **Componente de Constancias**
 - Responsabilidad: Generar constancia digital descargable de vacunación.
 - Interfaces: Servicio web que devuelve PDF o archivo digital.
- **Componente de Seguridad**
 - Responsabilidad: Autenticación y autorización.
 - Interfaces: Login, validación de token, gestión de roles (afiliado, médico, administrador).
- **Componente de Persistencia**
 - Responsabilidad: Acceso a la base de datos central.
 - Interfaces: CRUD para afiliados, vacunas, citas y centros.

6. Diagramas de despliegue UML (especificación previa)

Se identifican los nodos principales para el despliegue:

- **Cliente (afiliado)**
 - Dispositivo web/móvil (navegador o app híbrida).
- **Servidor de Aplicaciones (backend con microservicios)**
 - Alojamiento de los componentes: registro, validación, agenda, vacunas, constancias y seguridad.
- **Balanceador de Carga**
 - Distribuye peticiones entre múltiples instancias del servidor.
- **Servidor de Base de Datos**
 - Contiene la información de afiliados, citas, centros y registros de vacunación.
- **Servidor de Constancias (opcional, desacoplado)**
 - Genera los documentos PDF solicitados.

Paso 7. Rediseño iterativo considerando alternativas

Durante la evaluación del diseño se consideraron dos enfoques:

- **Arquitectura monolítica:** todos los módulos integrados en una sola aplicación.
 - Ventaja: simplicidad inicial.
 - Desventaja: difícil de escalar con alta demanda.
- **Arquitectura en microservicios (seleccionada):** cada módulo funciona como un servicio independiente.
 - Ventaja: escalabilidad, mantenibilidad y confiabilidad.
 - Desventaja: mayor complejidad inicial en la implementación.
 -
- La decisión final es arquitectura de microservicios desplegados en la nube, con balanceo de carga y seguridad reforzada.

Diagrama de Componentes

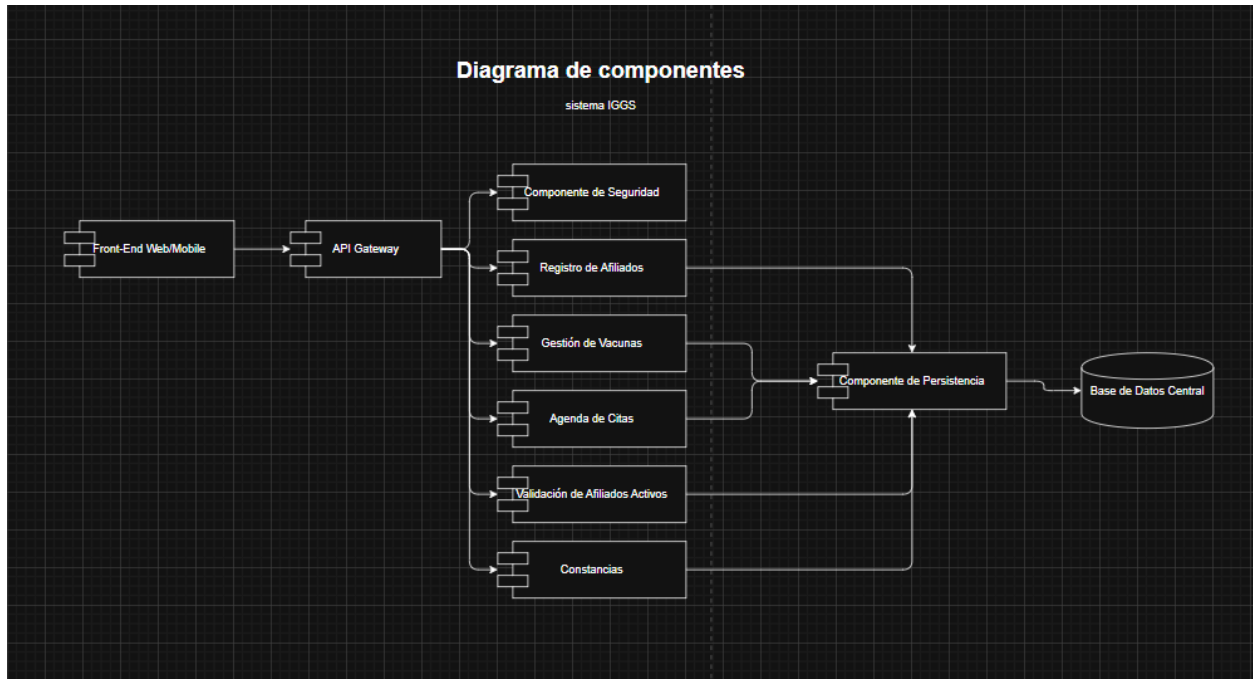


Diagrama de Estados

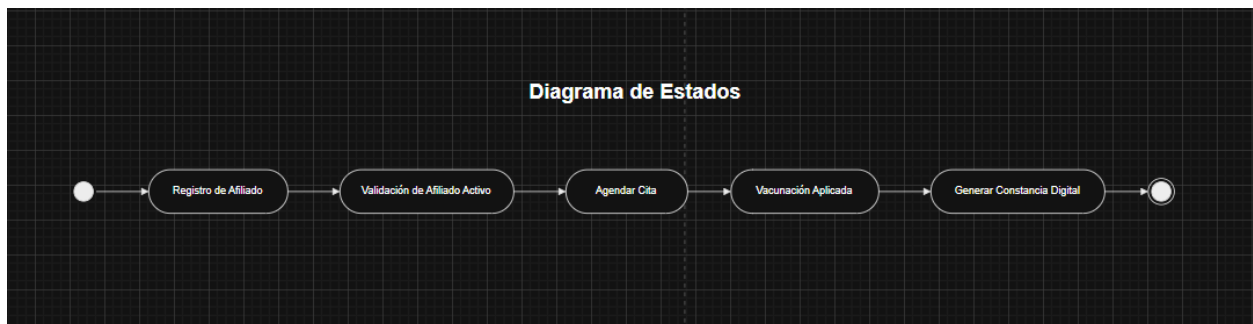


Diagrama de Estados

