

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO



**Facultad
de Informática
y Electrónica**

FACULTAD: INFORMÁTICA Y ELECTRONICA

CARRERA:

SOFTWARE

MATERIA:

APLICACIONES INFORMÁTICAS II

ESTUDIANTE:

JEANPIERRE DAVID AGUILAR ENCALADA – 7144

DOCENTE:

ING. JULIO ROBERTO SANTILLAN CASTILLO

NIVEL: OCTAVO SEMESTRE

PARALELO: "A"

SEPTIEMBRE 2025 – FEBRERO 2026

Pila de Herramientas

DESARROLLO DE UN SISTEMA WEB ASISTIDO POR UN AGENTE VIRTUAL PARA LA OPTIMIZACIÓN DE SERVICIOS, GESTIÓN DE INVENTARIO Y LA INTERPRETACIÓN PRELIMINAR DE RESULTADOS EN EL LABORATORIO CLÍNICO FRANZ.

1. Backend: Node.js (NestJS) + TypeScript

Node.js

- Entorno de ejecución JavaScript orientado a eventos, eficiente para operaciones de entrada/salida concurrentes.
- Ecosistema npm maduro, con librerías ampliamente probadas para servicios web, seguridad, envío de notificaciones, manejo de archivos y más.
- Unifica el lenguaje de servidor y cliente (JavaScript/TypeScript), lo que facilita mantenimiento y curva de aprendizaje.

NestJS

- Framework backend modular que fomenta arquitectura por dominios, separación clara entre controladores, servicios y acceso a datos.
- Incluye patrones empresariales (inyección de dependencias, pipes de validación, guards, interceptores) que estandarizan cómo se organiza la lógica de negocio.
- Está orientado a escalabilidad y testeo, permitiendo crecer en complejidad sin perder estructura.

TypeScript

- Aporta tipado estático y contratos claros entre capas.
- Reduce errores en tiempo de desarrollo y facilita refactorizaciones seguras.
- Mejora la mantenibilidad del código a largo plazo y la legibilidad del modelo de datos.

Justificación:

Esta combinación (Node.js + NestJS + TypeScript) ofrece una base sólida para construir servicios robustos, con estructura modular clara, tipado fuerte y buenas prácticas integradas desde el inicio.

2. Base de Datos: PostgreSQL (modelo transaccional estructurado)

- Motor de base de datos relacional ACID, ampliamente adoptado en entornos productivos que requieren confiabilidad, integridad y consistencia.
- Soporta llaves foráneas, restricciones y validaciones de datos a nivel de esquema, lo que ayuda a garantizar calidad e integridad en la información almacenada.
- Incluye tipos avanzados como JSONB, que permiten guardar metadatos o trazas técnicas sin perder flexibilidad.

- Facilita organización lógica en múltiples esquemas dentro de la misma instancia (por ejemplo separar dominios), mejorando orden, gobierno de datos y control de acceso interno.
- Herramientas maduras para replicación, respaldo y monitoreo.

Justificación:

PostgreSQL ofrece estabilidad transaccional, control de integridad y capacidad de modelar entidades relacionadas de manera explícita, lo que es adecuado para información sensible y trazable.

3. Frontend: Next.js (App Router) + React + Tailwind CSS + shadcn/ui + Radix UI + react-hook-form + Zod

React

- Biblioteca para construir interfaces de usuario declarativas mediante componentes reutilizables.
- Favorece una organización clara del estado y de la vista, y facilita componer pantallas complejas a partir de piezas pequeñas.

Next.js (App Router)

- Framework sobre React que incorpora enrutamiento estructurado, renderizado en el servidor y segmentación clara entre componentes del lado del servidor y del lado del cliente.
- Optimiza tiempos de carga, ordena la jerarquía de páginas y facilita proteger secciones que requieren acceso autenticado.
- Aporta buenas prácticas de organización por ruta y por feature.

Tailwind CSS

- Framework de utilidades CSS que permite construir interfaces responsivas (mobile-first) con consistencia visual en tipografía, espaciado, colores de estado, bordes y layouts.
- Reduce la necesidad de hojas de estilo personalizadas extensas y promueve un diseño coherente en toda la aplicación.
- Hace más fácil establecer patrones visuales: tarjetas con borde, badges de estado, tablas limpias, etc.

shadcn/ui + Radix UI

- Colección de componentes accesibles (cards, menús, tablas, diálogos, badges) diseñados con foco en usabilidad, navegabilidad por teclado y cumplimiento de requisitos de accesibilidad.
- Evitan reinventar componentes base y promueven consistencia visual y de interacción entre pantallas.
- Incorporan roles ARIA y patrones accesibles ya resueltos.

react-hook-form + Zod

- Conjunto de herramientas para construir formularios con validación estructurada y mensajes de error

claros.

- Zod permite definir esquemas de validación tipados; react-hook-form maneja el ciclo de vida del formulario de manera eficiente.
- Ayuda a garantizar calidad de datos en el punto de entrada (por ejemplo formato de correo, requisitos mínimos de contraseña, campos obligatorios).

Justificación:

Este stack frontend prioriza accesibilidad, consistencia visual, tiempos de carga razonables, validación explícita de entradas y mantenibilidad del código de interfaz. Además, favorece un diseño responsive que funciona tanto en escritorio como en dispositivos móviles.

4. Contenerización: Docker

- Docker empaqueta aplicación, dependencias y configuración en contenedores reproducibles.
- Asegura entornos consistentes entre desarrollo, pruebas y despliegue, evitando diferencias de “en mi máquina sí funciona”.
- Facilita actualizaciones y rollbacks controlados, ya que cada versión puede distribuirse como una imagen versionada.
- Es compatible con proveedores de nube y orquestadores estándar.

Justificación:

La contenerización estandariza la ejecución del sistema y reduce el riesgo operativo al mover el software entre entornos.

5. Mensajería asíncrona y trabajos en segundo plano: Redis + BullMQ

Redis

- Almacén en memoria de baja latencia, comúnmente usado como cola ligera o caché temporal.
- Adecuado para manejar información efímera asociada a tareas que no requieren procesamiento inmediato por el usuario.

BullMQ (o colas sobre Redis)

- Librería para gestión de colas de trabajo y tareas en background.
- Permite programar y encadenar trabajos (por ejemplo notificaciones, recordatorios, verificaciones periódicas, alertas operativas).
- Mejora escalabilidad al separar tareas que pueden ejecutarse de forma diferida.

Justificación:

La combinación de Redis con una cola de trabajos desacopla tareas que no deben bloquear la interacción del usuario y habilita ejecución controlada, repetible y monitoreable de tareas automáticas.

6. Almacenamiento de Archivos: Amazon S3 (o servicio compatible S3)

- Servicio de almacenamiento de objetos altamente duradero y escalable.
- Permite versionamiento, control de acceso detallado y cifrado en reposo.
- Facilita exponer archivos bajo políticas seguras (por ejemplo URLs firmadas con expiración limitada).
- Integra modelos de costos predecibles y alta disponibilidad.

Justificación:

Un repositorio de objetos dedicado es adecuado para conservar documentos generados por el sistema (por ejemplo archivos PDF firmados, comprobantes o reportes), con controles claros de acceso y trazabilidad.

7. Seguridad y Autenticación: JWT y control de sesiones

JWT (JSON Web Tokens)

- Estándar para representar identidades y permisos de forma firmada.
- Permite emitir tokens con tiempo de expiración controlado.
- Facilita invalidar sesiones mediante listas de revocación o marcas de cierre de sesión.

Control de sesiones / refresh tokens

- Manejo explícito de sesiones activas y expiración.
- Posibilidad de invalidar todas las sesiones de un usuario.
- Registro de intentos de inicio de sesión y bloqueo tras repetidos fallos.

Gestión de roles y permisos

- Separación de perfiles (por ejemplo personal interno, administrador, usuario final), con restricciones diferenciadas.
- Control de acceso basado en rol y propiedad de la información.

Justificación:

El uso de tokens firmados, expiración controlada y autorización basada en rol establece una capa clara de seguridad y reduce el riesgo de exposición de información sensible.

8. Auditoría y Trazabilidad

- Registro persistente de eventos relevantes del sistema (accesos, acciones sensibles, errores críticos).
- Almacenamiento tipo “append-only”: las entradas se agregan y quedan disponibles para inspección

histórica.

- Capacidad de asociar cada evento con identidad del usuario, marca de tiempo y metadatos técnicos (por ejemplo agente, IP).
- Soporte para análisis posterior, cumplimiento interno y resolución de incidentes.

Justificación:

La auditoría formalizada aporta trazabilidad, respaldo en caso de disputas y evidencia técnica de qué ocurrió, cuándo y bajo qué condiciones.

9. Agente Virtual / Asistente Conversacional: Dialogflow CX + Vertex AI

Dialogflow CX

- Plataforma orientada a diseño de flujos conversacionales estructurados (intenciones, estados, rutas de diálogo).
- Incluye herramientas de orquestación y pruebas para asistentes virtuales orientados a soporte y autoservicio.
- Ofrece capacidades multicanal y gestión centralizada de conversaciones.

Vertex AI (modelos generativos administrados)

- Acceso a modelos de lenguaje avanzados bajo control de una plataforma gestionada.
- Soporta clasificación, generación de texto y recuperación asistida, con monitoreo y cuotas administradas.
- Proporciona mecanismos de seguridad y control de contenido en entornos regulados.

Justificación:

Esta combinación habilita un canal conversacional estructurado, con lenguaje natural y capacidad de ofrecer orientación inmediata al usuario, sin requerir construir infraestructura de IA propia desde cero.

10. Herramientas de Apoyo: GitHub, GitHub Actions, Postman

GitHub

- Plataforma de control de versiones distribuido (Git) que permite colaboración, historial de cambios y trazabilidad de decisiones técnicas.

GitHub Actions

- Automatización de tareas de integración continua y despliegue continuo (CI/CD).
- Posibilidad de ejecutar pruebas, validaciones de estilo, builds y empaquetado en contenedores de forma repetible.

Postman

- Entorno para diseñar, documentar y probar endpoints HTTP de manera interactiva.
- Facilita compartir colecciones de pruebas y verificar contratos de API.

Justificación:

Estas herramientas soportan el ciclo completo de desarrollo: versionado del código, pruebas automatizadas, despliegue controlado y documentación/verificación de interfaces de servicio.