*v*

***“Año de la recuperación y consolidación de la economía peruana”***





Presentación:

<https://www.canva.com/design/DAG5fT4mfjs/FVuNUNKjv_0pJk045HDp7Q/edit?utm_content=DAG5fT4mfjs&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton>

GItHub:

<https://github.com/LuisanaNapan/Sistema-de-Gesti-n-de-citas-medicas.git>



**Índice**

[**Unidad III – Diseño de Software (Semanas 9–12) 3**](#_or6rl3aclm67)

[**Capítulo 6. Diseño de arquitectura y patrones 3**](#_watgz08zyn2m)

[1. Estratégia de diseño de software: 3](#_taf52i2xtbhs)

[2. Tipo de Arquitectura del sistema: 3](#_l110d5e9dx2s)

[3. Patrones de diseño aplicados: 3](#_te6rwj55kf7l)

[4. Diseño Estructural: 5](#_5xyukd3aaheg)

[**Capítulo 7. Diseño detallado de la base de datos 6**](#_o36fhsfgns6)

[1. Modelo lógico y físico: 6](#_4nefp4fg18f4)

[2. Script SQL 7](#_rnpwmg1xuiej)

[3. Procedimientos almacenados, vistas y triggers: 8](#_yete0kb7ctsr)

[4. Seguridad y respaldo: 10](#_6si06pkq0s56)

[**Capítulo 8. Diseño detallado de sistemas en red y móviles 11**](#_uo67lgk3oub1)

[1. Modelo de comunicación 11](#_m56orhtbranp)

[2. Diseño de sistema web o móvil 11](#_a6sixwuk8s7m)

[3. Gestión de datos en red 11](#_kzvaw3amyljm)

[4. Seguridad en red y móviles: 13](#_kly3zibbv28w)

[5. Justificación técnica 14](#_cqrwqnqbryn6)

[**Unidad IV – Diseño de Interacción Humano-Computadora (Semanas 13–14) 14**](#_pn3lyou5vxys)

[**Capítulo 9. Diseño de interfaz y experiencia de usuario (UX/UI) 14**](#_d0vyekjgezc0)

[1. Perfil del usuario / usuario meta 14](#_bhxsia2gke80)

[2. Principio de diseño aplicados (HCI) explicar como se aplica en el prototipo 16](#_vasye2bz7fhs)

[3. Diseño del prototipo (baja y alta fidelidad) 17](#_w36ye21soxnt)

[4. Flujo de Navegación del sistema 24](#_funz2k6dgmr)

[5. Relación entre prototipo y requerimientos 26](#_52cxbl2937uv)

[**Anexos: 27**](#_q6pv3ujtmns)

# 

# **Unidad III – Diseño de Software (Semanas 9–12)**

## **Capítulo 6. Diseño de arquitectura y patrones**

### ***Estratégia de diseño de software:***

* **Modularidad:** Dividir el sistema en componentes independientes para un mejor orden.
* **Cohesión alta y acoplamiento bajo:** cada módulo hace una sola cosa bien y depende poco de otros.
* **Abstracción:** ocultar detalles internos y exponer sólo lo necesario.
* **Reutilización:** usar componentes, librerías y estructuras existentes cuando sea posible.
* **Separación de responsabilidades (SoC):** UI, lógica, y datos deben estar separados.
* **Escalabilidad:** el diseño debe soportar crecimiento futuro.
* **Trazabilidad:** que el diseño se pueda relacionar con los requerimientos.

### ***Tipo de Arquitectura del sistema:***

Arquitectura Cliente-Servidor: La arquitectura cliente–servidor permite que las funciones del sistema estén claramente separadas entre quienes solicitan servicios (clientes) y quienes los procesan (servidores). En el sistema de Gestión de Citas Médicas, esta arquitectura facilita que pacientes, médicos y administradores accedan al sistema desde navegadores web, mientras que un servidor central se encarga de procesar las solicitudes, validar información y gestionar la base de datos.

* **Se centraliza toda la información**, lo que asegura que las citas, historiales y agendas médicas estén siempre actualizadas.
* **Mejora la seguridad**, ya que los datos se almacenan y controlan desde un único servidor protegido.
* **Permite acceso remoto**, de modo que los usuarios pueden gestionar sus citas desde cualquier lugar con conexión a Internet.
* **Facilita el mantenimiento**, ya que cualquier actualización se realiza en el servidor, sin necesidad de modificar los dispositivos de los usuarios.
* **Optimiza el rendimiento**, separando la interfaz ligera del cliente de los procesos complejos que se ejecutan en el servidor.

Esta arquitectura hace que el sistema sea más escalable, seguro y eficiente, permitiendo que varios usuarios trabajen simultáneamente sin afectar el desempeño general.

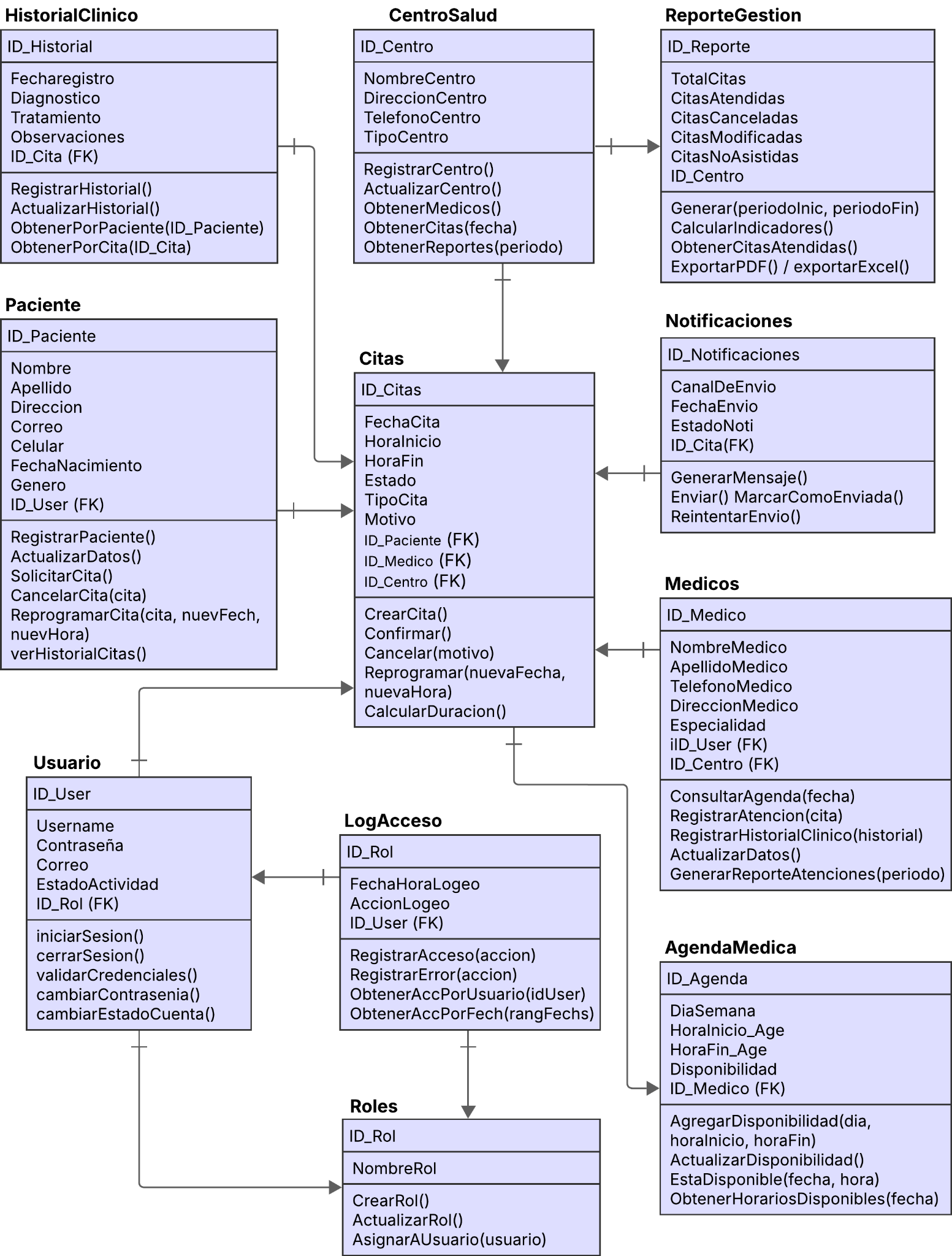
### ***Patrones de diseño aplicados:***

* 1. **Patrón Arquitectónico Cliente–Servidor:** El sistema utiliza la arquitectura cliente–servidor, donde los usuarios (paciente, médico y administrador) acceden al sistema desde una interfaz web o móvil, mientras que un servidor central procesa las solicitudes y gestiona la base de datos. Este patrón permite centralizar la información, mejorar la seguridad y facilitar el mantenimiento del sistema.
  2. **Patrón MVC (Modelo–Vista–Controlador):** El sistema separa su funcionamiento en tres capas:
* Modelo: base de datos y lógica de acceso a datos.
* Vista: interfaces de usuario (web o móvil).
* Controlador: lógica que recibe solicitudes, valida datos y coordina las operaciones.

Este patrón organiza el sistema, facilita la actualización de la interfaz y mejora el mantenimiento.

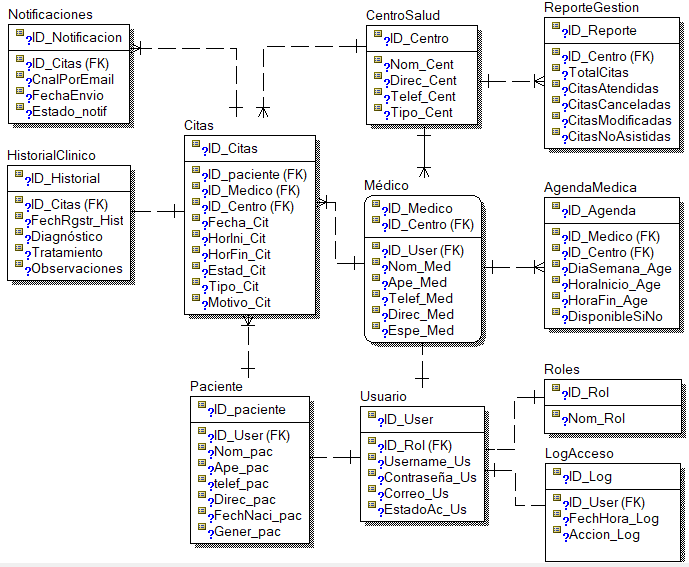
* 1. **Patrón DAO (Data Access Object):** El acceso a la base de datos se maneja a través de objetos especializados que encapsulan las consultas SQL, evitando que la lógica de negocio se mezcle con el manejo de datos. Esto mejora la reutilización del código y permite que los cambios en la base de datos no afecten al resto del sistema.
  2. **Patrón Singleton:** Se utiliza para garantizar que exista una única instancia del objeto de conexión a la base de datos, evitando duplicación de conexiones y optimizando los recursos del sistema. Este patrón asegura un manejo eficiente y controlado de la comunicación con la base de datos.
  3. **Patrón Observer (Observador):** Se aplica en el proceso de notificaciones automáticas a los pacientes. Cuando una cita es creada, reprogramada o cancelada, el sistema detecta el cambio de estado y genera una notificación automáticamente.Este patrón permite que los cambios en una entidad (cita) generen respuestas automáticas (notificaciones).

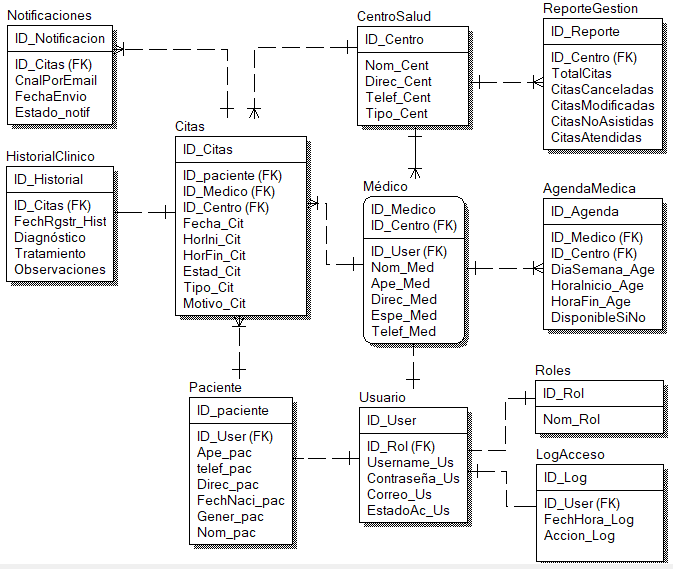
### ***Diseño Estructural:***



## **Capítulo 7. Diseño detallado de la base de datos**

### ***Modelo lógico y físico:***

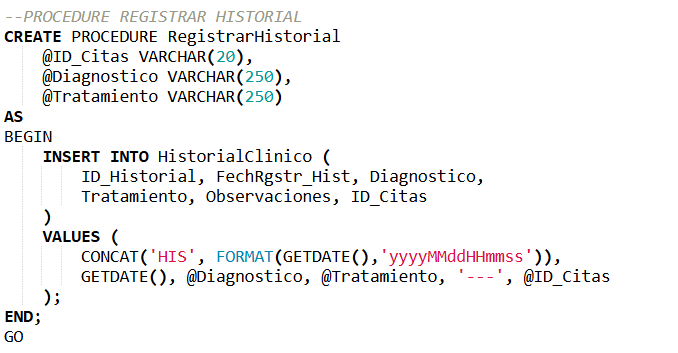


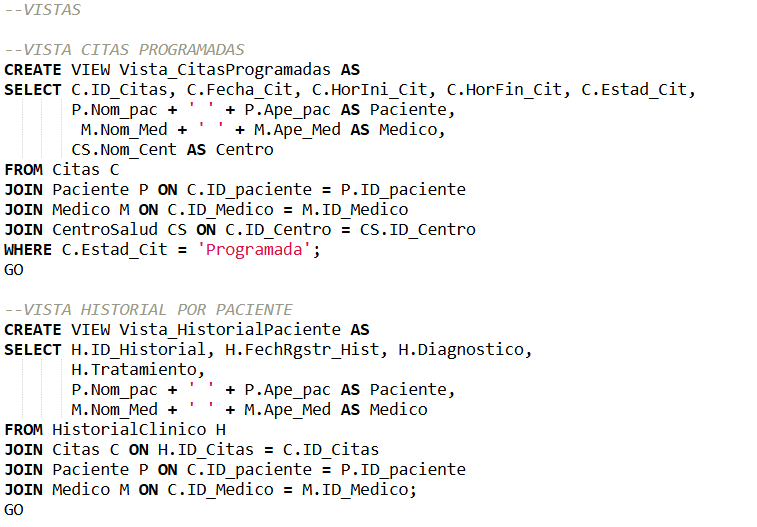


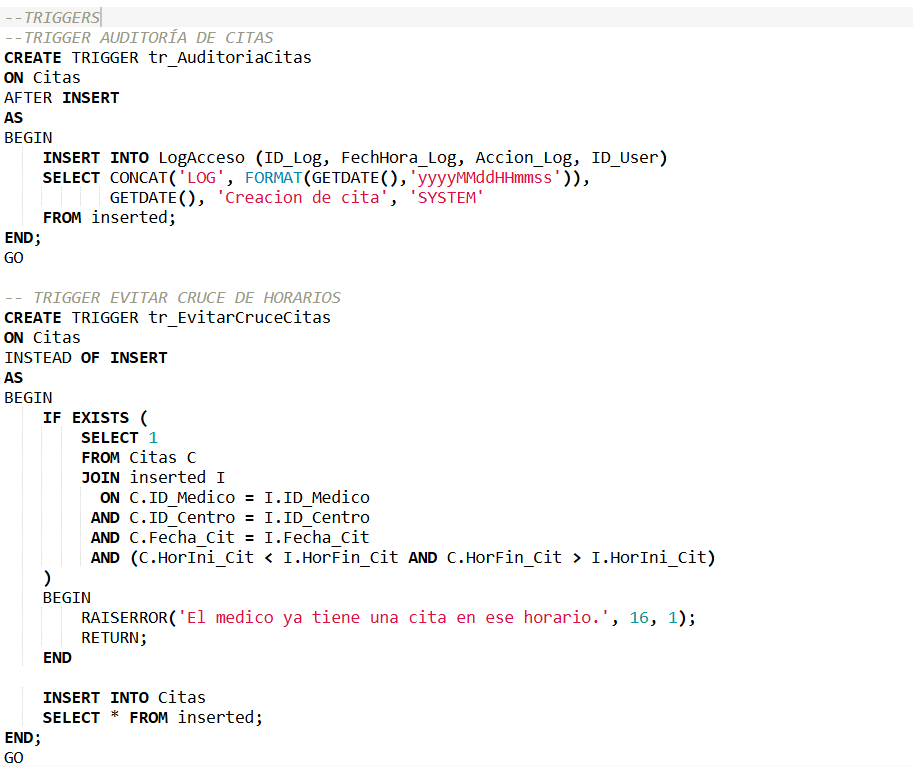
### ***Script SQL***

<https://drive.google.com/file/d/1dMLKi3o091eVE43b7MqvaXX3LII8hhme/view?usp=sharing>

### ***Procedimientos almacenados, vistas y triggers:***







### ***Seguridad y respaldo:***

Seguridad: La seguridad es fundamental en un sistema que maneja información sensible de pacientes, médicos y centros de salud. Para este proyecto se aplican las siguientes medidas:

* 1. Control de acceso por roles: Cada usuario del sistema cuenta con un rol específico que determina los permisos que puede ejercer:
     1. **Administrador:** acceso total a la gestión de usuarios, médicos, reportes y configuración del sistema.
     2. **Médico:** acceso a sus citas, historial clínico de pacientes y registro de evoluciones médicas.
     3. **Paciente:** acceso a sus citas programadas, historial clínico y actualización de su información.
     4. Esto asegura que cada actor solo vea y manipule la información correspondiente a su función.
  2. Autenticación y protección de contraseñas: El sistema cuenta con:
     1. Inicio de sesión seguro con usuario y contraseña.
     2. Almacenamiento de contraseñas utilizando **métodos de hash** (SHA-256, bcrypt o equivalente).
     3. Sesiones controladas y expiración automática por inactividad.
  3. Seguridad en la base de datos: A nivel del motor de datos se aplican:
     1. Restricción de permisos mediante **GRANT** y **REVOKE**.
     2. Las tablas solo permiten operaciones según el rol del usuario.
     3. Uso de *stored procedures* para evitar acceso directo a las tablas.
     4. Validación de datos para evitar ataques de inyección SQL.
  4. Seguridad en comunicación y red
     1. Uso de protocolo HTTPS para encriptar la información transmitida.
     2. Validación de formularios para evitar ataques como XSS o CSRF.
     3. Control de acceso mediante firewall del servidor.
     4. Auditoría del sistema mediante el registro de actividades importantes (log de accesos, creación de citas, cancelaciones, etc.).
  5. Protección de datos personales (Ley 29733): Se cumplen los lineamientos de la Ley de Protección de Datos Personales del Perú, garantizando:
     1. Confidencialidad
     2. Integridad
     3. Disponibilidad
     4. Consentimiento informado para el manejo de datos clínicos.
  6. Respaldo: El sistema implementa un plan de respaldo para evitar pérdida de información ante fallos, errores humanos o desastres.
     1. Copias de seguridad completas
        1. Se realizan una vez al día (backup completo).
        2. Se almacenan en un servidor alterno o disco seguro.
     2. Backups diferenciales
        1. Ejecutados cada 6 horas para registrar cambios desde el último backup completo.
     3. Backups de transacciones
        1. Ejecutados cada 30 minutos, permitiendo restaurar la base casi al minuto exacto en caso de falla.
     4. Replicación en un servidor secundario
        1. Copia espejo de la base de datos para garantizar continuidad del servicio.
        2. Útil en caso de caída del servidor principal.
     5. Plan de recuperación ante desastres (DRP), Incluye:
        1. Procedimiento para restaurar la base desde los backups.
        2. Tiempo estimado de recuperación (RTO menor a 30 minutos).
        3. Pruebas de restauración realizadas periódicamente.
        4. Copias externas (en la nube o discos externos) para mayor protección.

## **Capítulo 8. Diseño detallado de sistemas en red y móviles**

### ***Modelo de comunicación***

El sistema de Gestión de Citas Médicas utiliza un modelo de comunicación cliente–servidor basado en una arquitectura de tres capas: presentación, lógica de negocio y datos. Los usuarios acceden al sistema mediante un navegador web o dispositivo móvil, el cual se comunica con el servidor a través del protocolo HTTP/HTTPS.

### ***Diseño de sistema web o móvil***

**Diseño del Sistema Web:** El Sistema de Gestión de Citas Médicas se implementa como un sistema web, accesible desde navegadores modernos en computadoras, laptops, tablets o teléfonos móviles. Su diseño utiliza una arquitectura cliente–servidor, donde la interfaz web permite que los usuarios (pacientes, médicos y administradores) interactúen con el sistema sin necesidad de instalar software adicional.

El diseño web se adapta a distintos tamaños de pantalla mediante una interfaz responsiva, garantizando acceso seguro, rápido y cómodo desde cualquier dispositivo conectado a Internet.

### ***Gestión de datos en red***

La implementación del sistema de gestión de citas médicas requiere una infraestructura que permita mantener la disponibilidad, integridad y consistencia de los datos en todo momento. Debido a que será utilizado por diferentes actores (pacientes, médicos, administradores) desde distintos puntos del centro de salud o incluso de manera remota, es necesario trabajar mediante un modelo de gestión de datos en red, el cual garantiza un manejo eficiente y seguro de la información.

El sistema contará con una base de datos centralizada, alojada en un servidor interno o en la nube, a la cual se conectarán todos los módulos del sistema. Esto permite que: Cada cita registrada se sincroniza en tiempo real, Se elimina la duplicidad de datos del paciente y se eviten errores como sobreasignaciones y pérdida de información.

* 1. **Manejo de transacciones concurrentes:**

Dado que varios usuarios pueden realizar operaciones simultáneamente, es necesario contar con mecanismos que prevengan errores o modificaciones conflictivas. Para ello se aplican.

* + 1. **Bloqueos de registros (locking):** Evitan que dos usuarios editen la misma información al mismo tiempo reduciendo así los errores.
    2. **Control de concurrencia mediante transacciones ACID**,Garantiza que cada operación sea:
       1. Atómica
       2. Consistente
       3. Aislada
       4. Duradera
    3. **Niveles de aislamiento:** Impiden que transacciones paralelas produzcan datos inconsistentes.
    4. **Integración en red con otros módulos del centro de salud:** El sistema podrá comunicarse con:
       1. Historia clínica electrónica
       2. Módulo de disponibilidad médica
       3. Gestión de pagos
       4. Módulo de reportes

Para esto se emplean APIs REST, protocolos HTTPS, y estándares de comunicación que permiten:

* + - 1. Intercambio seguro de datos
      2. Compatibilidad con sistemas existentes
      3. Escalabilidad en redes corporativas

Esta interoperabilidad mejora el flujo de trabajo y reduce redundancias. Sincronización en tiempo real:

* + - 1. WebSockets
      2. Polling optimizado
      3. Notificaciones push

Estas tecnologías permiten que los cambios se reflejen instantáneamente, mejorando la experiencia del usuario y disminuyendo errores en la programación de citas.

### ***Seguridad en red y móviles:***

Dado que se trata de un sistema que maneja información sensible (datos personales y médicos), es obligatorio aplicar estándares de seguridad informática bajo normativas como las de protección de datos personales.

* 1. **Cifrado de datos en tránsito:**

Toda comunicación entre cliente y servidor debe realizarse con TLS 1.2/1.3, garantizando que:

* + 1. Las credenciales no serán interceptadas.
    2. La información de citas y datos personales viajan cifrados.
  1. **Cifrado de datos en reposo:**

La base de datos implementa cifrado a nivel de disco o tabla para proteger la información en caso de accesos no autorizados o robo del equipo servidor.

* 1. **Autenticación y control de acceso**

El sistema integra:

* + 1. Autenticación multifactor (MFA) para personal administrativo.
    2. Roles y permisos (RBAC) para limitar acciones según el tipo de usuario.
    3. Tokens JWT firmados para sesiones seguras en móviles y navegadores.
  1. **Seguridad en dispositivos móviles**

Para los usuarios que acceden desde teléfonos:

* + 1. Validación de integridad del dispositivo.
    2. Sesiones con expiración automática.
    3. Restricción de almacenamiento local de datos sensibles.
    4. Implementación de OAuth2 para inicio de sesión seguro.
  1. **Protección contra ataques comunes**

El sistema debe contar con:

* + 1. Filtros contra SQL Injection.
    2. Protección contra XSS y CSRF.
    3. Verificación de integridad del backend con firewalls de aplicaciones (WAF).
  1. **Auditoría y trazabilidad**

Se generan logs seguros de:

* + 1. Intentos de inicio de sesión
    2. Cambios de horarios
    3. Reservas, cancelaciones y reprogramación
    4. Lo que ayuda al cumplimiento normativo y facilita la detección de anomalías.

### ***Justificación técnica***

El diseño técnico del sistema se basa en una arquitectura web cliente–servidor porque permite que pacientes, médicos y administradores accedan al sistema desde cualquier dispositivo con Internet sin instalar aplicaciones. Además, usar una base de datos centralizada asegura que la información de las citas y los historiales esté siempre actualizada y no se duplique. También se aplican medidas de seguridad como HTTPS, roles y control de acceso para proteger los datos médicos. En resumen, este diseño es práctico, seguro y adecuado para un sistema que debe funcionar en red y con varios usuarios al mismo tiempo.

# **Unidad IV – Diseño de Interacción Humano-Computadora (Semanas 13–14)**

## **Capítulo 9. Diseño de interfaz y experiencia de usuario (UX/UI)**

### ***Perfil del usuario / usuario meta***

* 1. **Definición del usuario final según ODS y necesidades reales:**

El sistema se desarrolla alineado con la ODS 3: Salud y Bienestar, cuyo propósito es garantizar acceso oportuno y de calidad a los servicios de salud. Bajo este enfoque, los usuarios finales se definen según las necesidades reales que tienen dentro del proceso de atención médica.

* + 1. **Paciente (Usuario Final Principal):**
       1. **Relación con ODS 3:** Es la persona que recibe el servicio de salud y necesita acceso seguro y rápido a sus citas y datos médicos.
       2. **Necesidades reales:**
          1. Programar una cita sin tener que trasladarse o hacer colas.
          2. Ver su historial clínico de manera clara y protegida.
          3. Recibir recordatorios y notificaciones automáticas.
          4. Acceder al sistema desde el celular o computadora.
          5. Contar con una interfaz simple, entendible y accesible.
    2. **Médico (Usuario Intermedio):**
       1. **Relación con ODS 3:** El sistema debe permitir atención médica eficiente, organizada y segura.
       2. **Necesidades reales:**
          1. Consultar su agenda de manera ordenada.
          2. Evitar cruces de horarios gracias a validaciones del sistema.
          3. Registrar diagnósticos y tratamientos fácilmente.
          4. Acceder rápido al historial del paciente.
    3. **Administrador del Centro de Salud (Usuario Gestor)**
       1. **Relación con ODS 3:** La buena gestión del sistema mejora la calidad del servicio para pacientes y médicos.
       2. **Necesidades reales:**
          1. Registrar y administrar médicos, pacientes y citas.
          2. Monitorear estadísticas y reportes.
          3. Asegurar la protección de los datos almacenados
          4. Mantener el sistema en funcionamiento óptimo.
  1. **Contexto de uso del sistema.**

El sistema se utiliza dentro del entorno de servicios médicos y en espacios relacionados con la atención de salud.

* + 1. **Entorno donde se usa**
       1. Consultorios médicos.
       2. Policlínicos y centros de salud.
       3. Hospitales.
       4. Domicilios de pacientes que ingresan para reservar citas.
       5. Oficinas del personal administrativo.
    2. **Situaciones reales de uso**
       1. **Paciente**
          1. Ingresa a la plataforma para reservar, reprogramar o cancelar una cita.
          2. Revisa su historial clínico o consultas anteriores.
          3. Recibe recordatorios o notificaciones.
       2. **Médico**
          1. Consulta su agenda diaria antes de atender pacientes.
          2. Registra diagnósticos, tratamientos y observaciones de cada cita.
          3. Revisa historial clínico previo a la atención.
       3. **Administrador**
          1. Controla registros de doctores, pacientes y usuarios.
          2. Supervisa citas atendidas, canceladas o pendientes.
          3. Genera reportes para la toma de decisiones.
    3. **Características tecnológicas del contexto**
       1. Acceso mediante un sistema web o plataforma interna.
       2. Conexión a una base de datos SQL Server con roles y permisos.
       3. Funciona en computadoras, laptops o tabletas del centro de salud.
       4. Puede ser usado por pacientes desde teléfonos móviles.

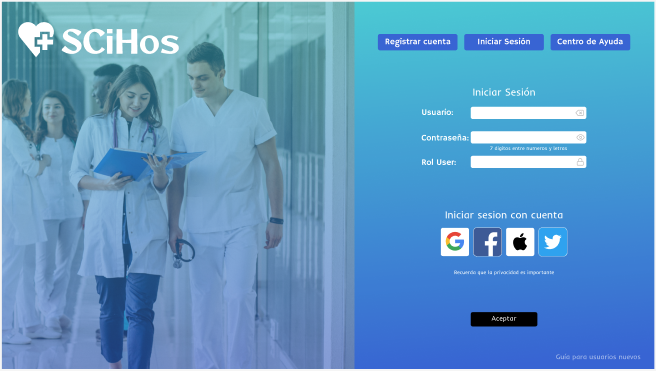
### ***Principio de diseño aplicados (HCI) explicar como se aplica en el prototipo***

* 1. **Consistencia:** En el prototipo se mantiene un estilo visual coherente:
     1. Colores predominantes: azul, blanco y turquesa, negro.
     2. Tipografía uniforme en todos los módulos (paciente, médico, administrador).
     3. Botones con la misma forma, color y estilo (“Nueva cita”, “Cancelar cita”, “Modificar cita”, etc.).
     4. El calendario mensual siempre aparece del lado izquierdo en la vista del médico y del administrador.
     5. Las tarjetas de citas (3 PM, 2 PM, 8 AM) siempre tienen el mismo formato y colores por estado.
  2. **Visibilidad:** 
     1. En el panel del médico, los botones críticos como Nueva cita, Cancelar cita, Reprogramar, están en negro con texto blanco y tamaño grande, haciéndolos fáciles de detectar.
     2. La agenda diaria está bien distribuida por horas, con bloques de color para representar cada cita (verde “Ocupado”, morado “Aún falta”).
     3. Los datos del paciente se muestran en un popup central con campos claramente etiquetados.
     4. En la pantalla de inicio, las opciones “Registrar cuenta”, “Iniciar sesión”, “Centro de ayuda” son visibles en la parte superior.
  3. **Accesibilidad:**
     1. Los textos son grandes, legibles y con buen contraste (blanco sobre azul).
     2. Los botones son amplios y fáciles de presionar, especialmente para usuarios en móviles.
     3. La interfaz es simple y puede ser utilizada con mouse, pantalla táctil o teclado.
     4. La estructura “dos columnas” permite que se adapte a pantallas más pequeñas sin perder funcionalidad.
     5. Los colores verde, amarillo y morado para citas ayudan a distinguir estados visualmente (incluso para usuarios con dificultades leves de visión).
  4. **Control de usuario:**
     1. El médico puede aceptar o rechazar una cita desde la tarjeta informativa.
     2. Los pacientes pueden modificar, cancelar o reprogramar una cita antes de que sea confirmada desde notificaciones.
     3. El administrador puede gestionar roles, ver citas en espera, enviar reportes y confirmar cambios.
     4. Popups como “Aceptar cita” o “Enviar boleta al correo” permiten control total sobre el flujo.
  5. **Simplicidad:**
     1. Las pantallas muestran solo lo necesario según el rol (Paciente, Médico, Administrador).
     2. No hay exceso de opciones: todo está agrupado en menús breves.
     3. Los popups tienen pocas entradas, bien organizadas y de tamaño uniforme.
     4. La pantalla de registro/inicio de sesión es clara y con pocos campos.
     5. El uso de colores sólidos y espacios amplios mejora la comprensión.

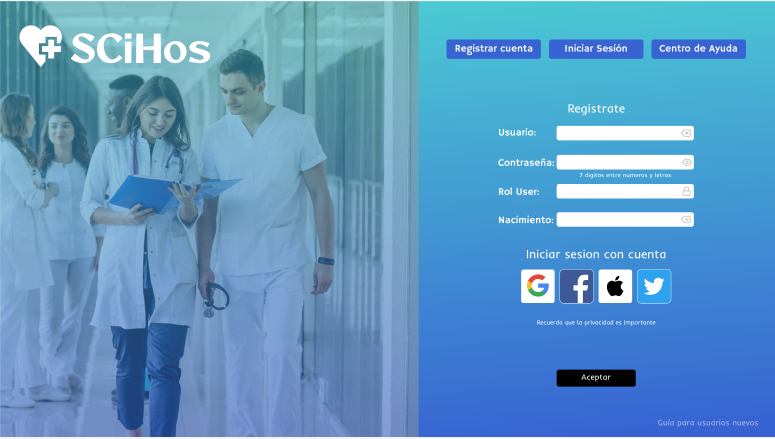
### ***Diseño del prototipo (baja y alta fidelidad)***

* 1. **Prototipos realizados en figma**

Inicio de Sesión: Cuando el usuario ya tiene una cuenta puede ingresar con su cuenta directamente.



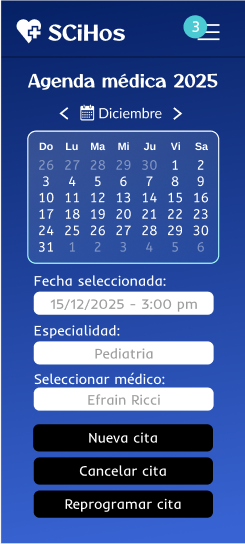
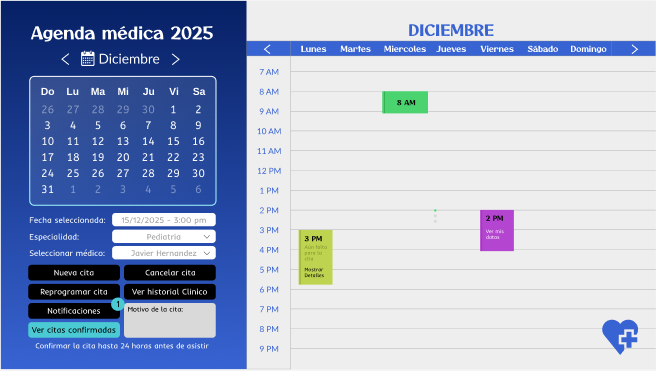
Registro de Usuario: Cuando el usuario no tiene cuenta aún debe de crear una nueva cuenta para poder ingresar.



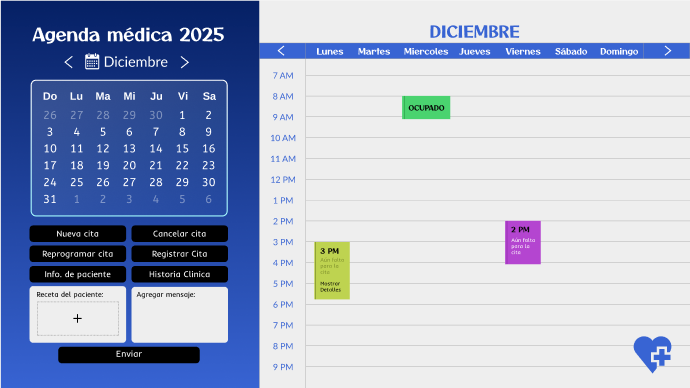
Menú principal:



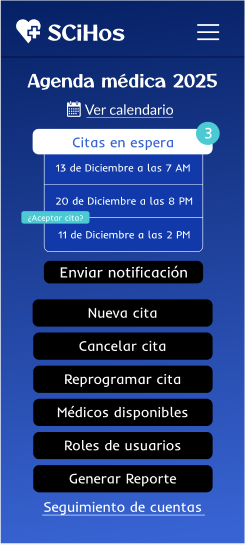
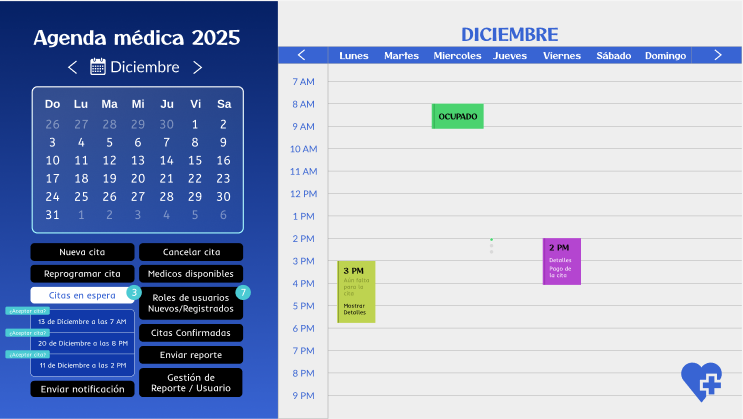
Agenda para registrar cita → paciente



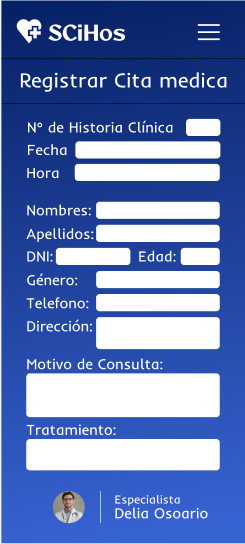
Ver calendario → Médico



Control/Confirmar/reporte → Administrador



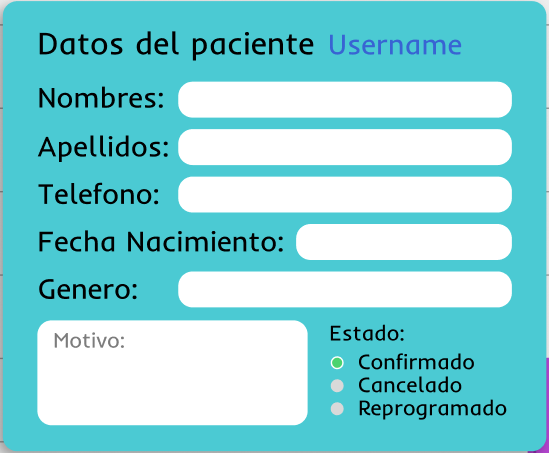
Registro de cita



Gestión de Reporte:



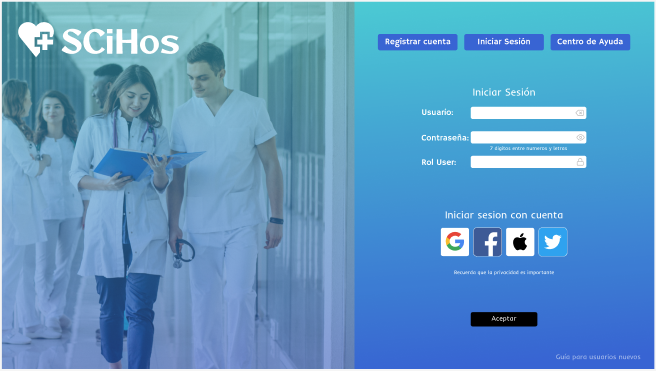
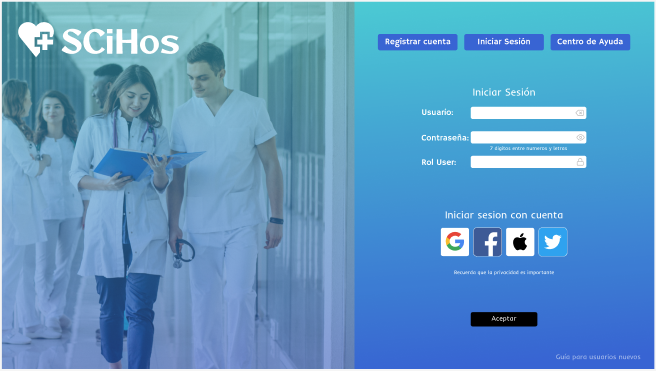
PopUp

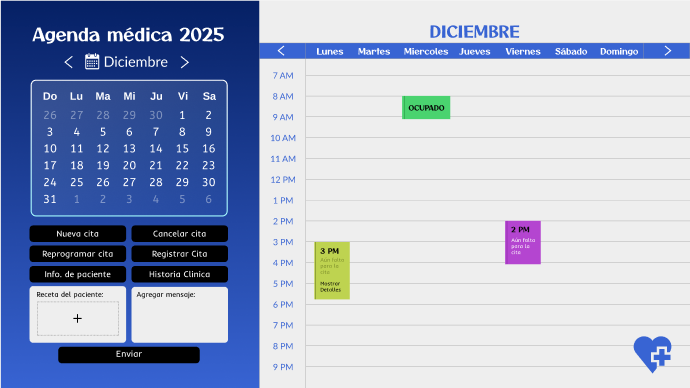
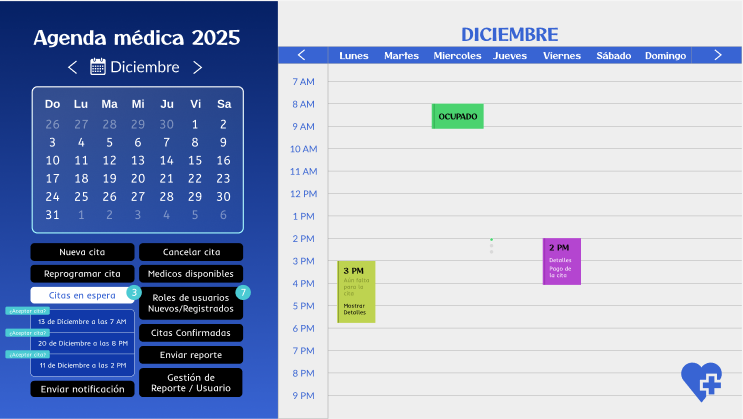
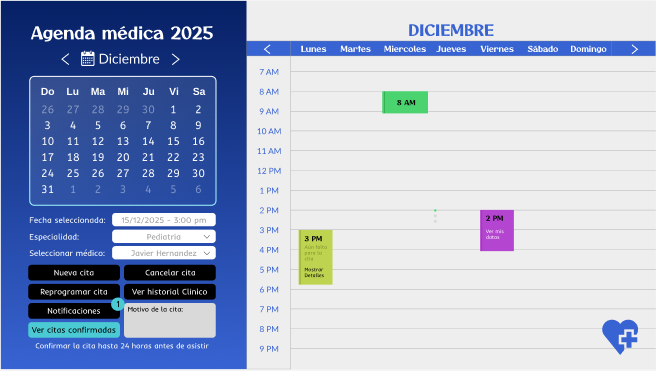


|  |  |
| --- | --- |

****

* 1. **Captura de las pantallas principales**





* 1. **Justificación del diseño**

El diseño de este sistema se justifica porque busca solucionar los principales problemas identificados en la gestión de citas médicas: desorden en horarios, pérdida de información y falta de organización entre pacientes, médicos y administradores. Por esta razón, las pantallas y funciones fueron creadas de manera simple y clara, para que cada usuario pueda encontrar fácilmente lo que necesita.

El sistema está dividido en módulos (usuarios, citas, agenda, historial, notificaciones, administración y reportes) para mantener un orden y facilitar futuros cambios. Además, se aplicaron principios básicos de diseño como consistencia, simplicidad y visibilidad, lo que permite que las interfaces sean fáciles de usar y entender. Cada pantalla está pensada según el rol que la utiliza: el paciente reserva citas, el médico consulta su agenda y registra la atención, y el administrador gestiona usuarios y reportes.

En resumen, el diseño se justifica porque es práctico, organizado y centrado en las necesidades reales de los usuarios del sistema, logrando una interfaz clara y funcional que mejora el proceso de atención médica.

**Encuesta realizada para mejorar el diseño:**

<https://docs.google.com/forms/d/e/1FAIpQLScinmZO1jX0PYYxC5gSDYs7TSaWrRuCYJrbbnpW82p1AIQl4g/viewform?usp=header>



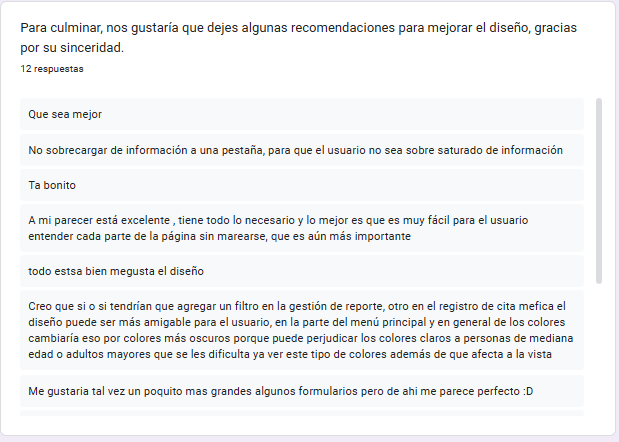


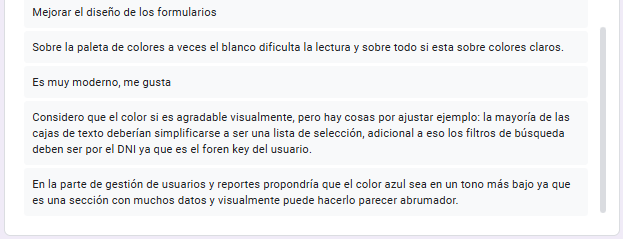






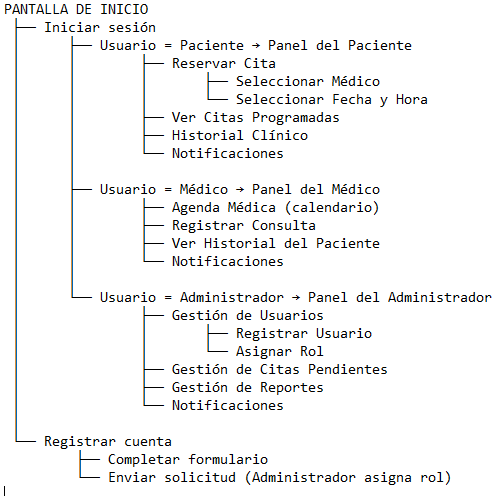
Opiniones de los usuarios:



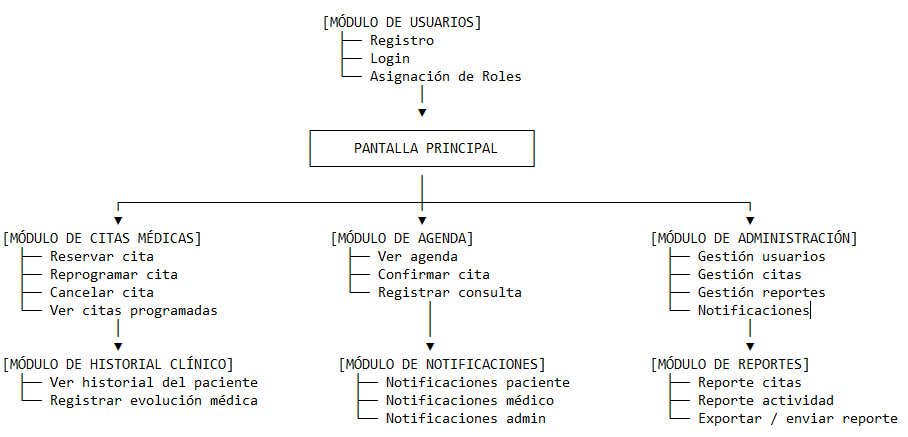


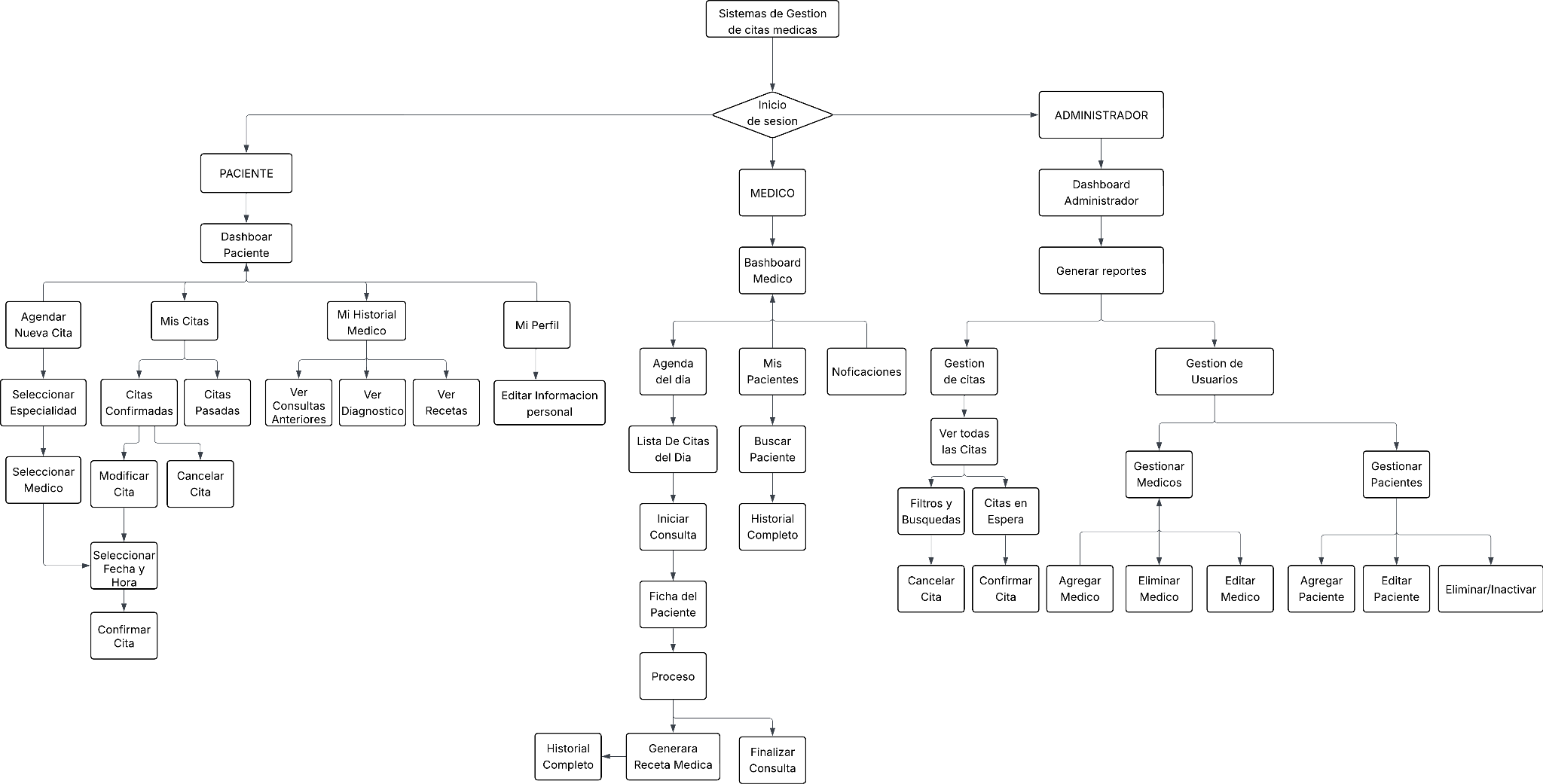
### ***Flujo de Navegación del sistema***

* 1. **Mapa de navegacion que representa el recorrido del usuario**

****

* 1. **Diagrama de navegación o módulo-navegación**

****



El diagrama módulo–navegación representa la estructura principal del sistema y la relación entre los diferentes módulos. Cada módulo se conecta con las funciones necesarias según el rol del usuario (paciente, médico, administrador). Esto permite visualizar cómo está organizado el sistema internamente y cómo fluye la interacción entre los módulos para completar los procesos principales, como registrar usuarios, gestionar citas, consultar historiales clínicos, administrar datos y generar reportes.

### ***Relación entre prototipo y requerimientos***

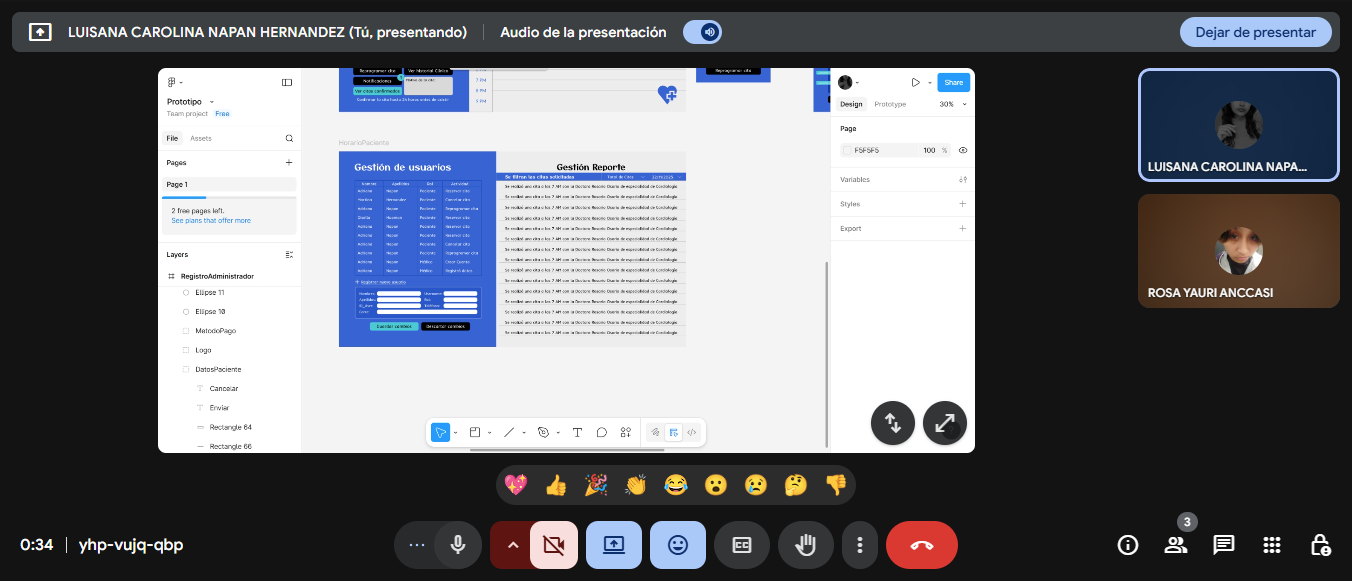
* 1. **Cada pantalla debe evidenciar que requerimiento satisface**

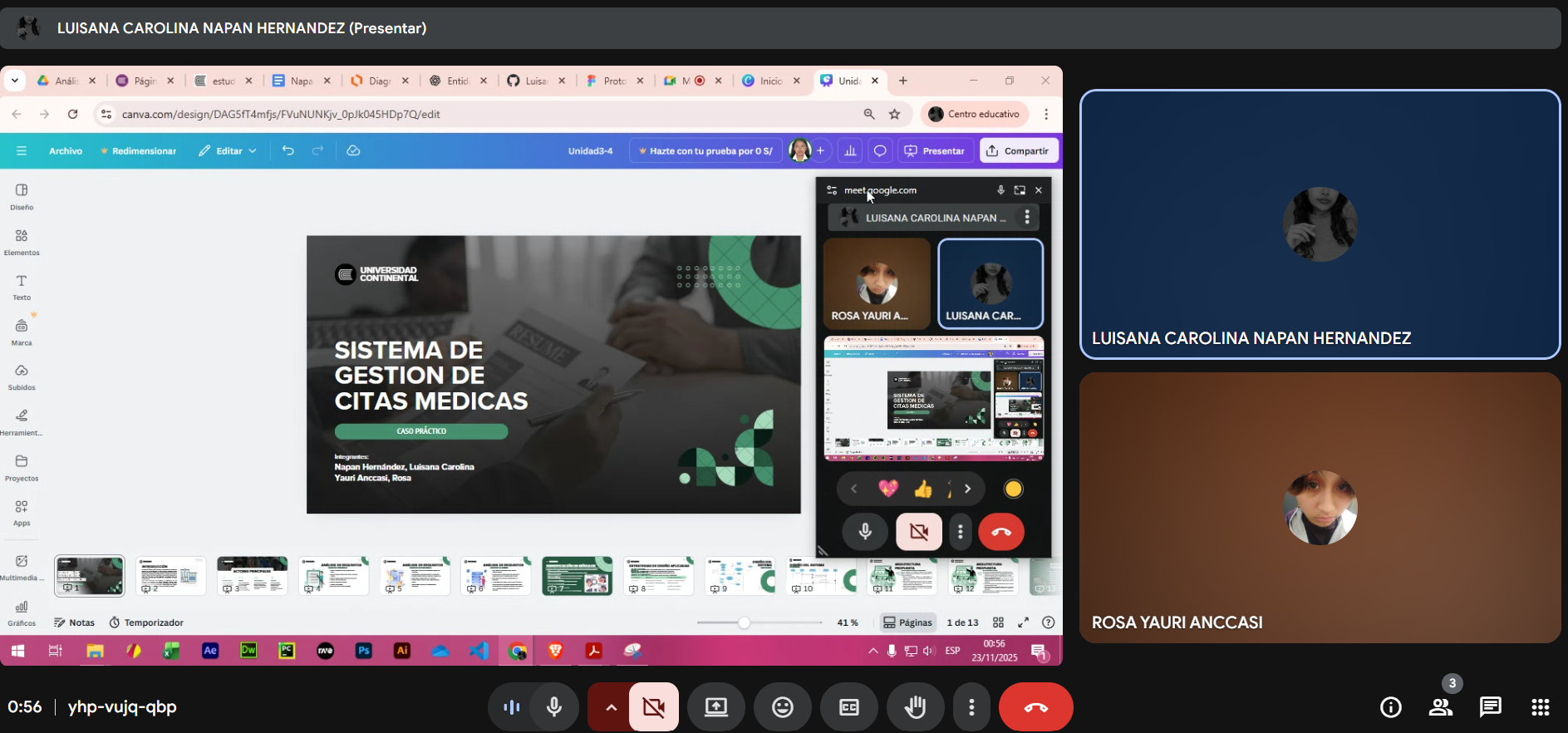
En el prototipo se puede observar cómo cada interfaz está relacionada directamente con los requerimientos funcionales y no funcionales.

* + 1. Pantalla de Inicio de Sesión
       1. RF02: Inicio de sesión seguro.
       2. RNF01: Seguridad con autenticación.
    2. Pantalla de Registro
       1. RF01: Registro único de pacientes.
       2. RNF03: Usabilidad y simplicidad.
    3. Menú principal del Paciente
       1. RF08: Consultar y reservar citas.
       2. RF09: Reprogramar o cancelar citas.
       3. RNF03: Interfaz intuitiva y clara.
    4. Agenda del Médico
       1. RF10: Visualizar agenda médica.
       2. RNF05: Tiempo de respuesta menor a 3 segundos.
    5. Pantalla de historial clínico
       1. RF11 y RF12: Gestión y visualización del historial.
    6. Panel del Administrador
       1. RF04 y RF05: Gestión de pacientes y médicos.
       2. RF13: Reportes administrativos.
       3. RNF04: Escalabilidad del sistema.

Cada sección del prototipo fue diseñada asegurando la trazabilidad entre los requerimientos definidos y las funcionalidades visuales, de manera que el diseño no solo represente la interfaz, sino también la estructura funcional del sistema final.

# **Anexos:**





**Commits:**

****

