Documento de Diseño de la Base de Datos (Cloud SQL)

Este documento detalla el diseño de la base de datos relacional para el "Ecosistema Digital Inteligente para Cirugía Especial", la cual se alojará en Google Cloud Platform (GCP) utilizando Cloud SQL (MySQL/PostgreSQL). El objetivo es centralizar toda la información crítica del consultorio de manera segura, garantizando la integridad de los datos sensibles mediante cifrado en reposo.

1. Modelo Entidad-Relación (ERD)

El modelo Entidad-Relación (ERD) describe las principales entidades y sus interconexiones dentro del ecosistema digital, sirviendo como la columna vertebral para la gestión de pacientes y la operación diaria del consultorio. Este modelo asegura que la información estructurada, como perfiles de pacientes, historiales clínicos, citas y usuarios, se almacene de forma coherente y segura.

Las entidades principales y sus relaciones son las siguientes:

• PACIENTES y HISTORIAL_CLINICO: La entidad PACIENTES "tiene" una relación de uno a muchos con HISTORIAL_CLINICO, lo que significa que un paciente puede tener múltiples entradas en su historial clínico a lo largo del tiempo.

• HISTORIAL_CLINICO y DOCUMENTOS: La entidad HISTORIAL_CLINICO "contiene" DOCUMENTOS. Esto permite adjuntar archivos digitales como PDFs, imágenes o videos cortos a entradas específicas del historial clínico de un paciente.

• PACIENTES y CITAS: La entidad PACIENTES "agenda" CITAS. Un paciente puede tener múltiples citas programadas con el consultorio.

• CITAS y PROCEDIMIENTOS_CITAS: La entidad CITAS "se relaciona con" PROCEDIMIENTOS_CITAS. Esta tabla intermedia permite asociar una o más citas con procedimientos específicos.

• PROCEDIMIENTOS_CITAS y PROCEDIMIENTOS: La entidad PROCEDIMIENTOS_CITAS "detalla" PROCEDIMIENTOS. Esto vincula los procedimientos realizados con las citas correspondientes.

• USUARIOS y ROLES: La entidad ROLES "asigna" USUARIOS. Cada usuario del sistema tiene un rol definido que determina sus permisos de acceso a la información y funcionalidades.

• USUARIOS y LOGS_AUDITORIA: La entidad USUARIOS "realiza" operaciones que son registradas en LOGS_AUDITORIA. Este componente registra cronológicamente todas las actividades y cambios realizados en el sistema, siendo vital para la seguridad y el seguimiento de operaciones.

2. Especificación Completa de Tablas (Parte 1)

A continuación, se detalla la estructura y las características de las tablas PACIENTES e HISTORIAL_CLINICO, incluyendo sus columnas, tipos de datos, restricciones y relaciones.

Tabla: PACIENTES

Esta tabla centraliza la información demográfica y de contacto de todos los pacientes del consultorio.

Columna
Tipo de Dato
Restricciones
Descripción
id_paciente
INT
PK, NOT NULL, AUTO_INCREMENT
Identificador único y autoincremental para cada paciente.
nombre
VARCHAR(255)
NOT NULL
Nombre(s) del paciente.
apellido
VARCHAR(255)
NOT NULL
Apellido(s) del paciente.
fecha_nac
DATE
NULLABLE
Fecha de nacimiento del paciente.
datos_contacto
JSON
NULLABLE
Almacena información de contacto adicional (ej., teléfono, email, dirección) en formato JSON.

activo
BOOLEAN
NOT NULL, DEFAULT TRUE
Indica si el registro del paciente está activo (TRUE) o ha sido eliminado suavemente/archivado (FALSE).
Tabla: HISTORIAL_CLINICO
Esta tabla almacena el registro cronológico y detallado de cada interacción clínica con un paciente.
Columna
Tipo de Dato
Restricciones
Descripción
id_historial
INT
PK, NOT NULL, AUTO_INCREMENT
Identificador único y autoincremental para cada entrada del historial clínico.
id_paciente
INT
FK a PACIENTES.id_paciente, NOT NULL
Clave foránea que vincula la entrada del historial a un paciente específico.
fecha_consulta
DATE
NOT NULL
Fecha en la que se realizó la consulta o se registró la entrada.
diagnostico
TEXT
NULLABLE
Diagnóstico asociado a la consulta.
notas_medico
TEXT
NULLABLE
Notas detalladas del médico sobre la evolución, procedimientos o tratamientos.
activo
BOOLEAN
NOT NULL, DEFAULT TRUE
Indica si la entrada del historial está activa (TRUE) o ha sido marcada como inactiva/archivada (FALSE).

2. Especificación Completa de Tablas (Parte 2)
Se presenta la estructura detallada de las tablas USUARIOS, ROLES y LOGS_AUDITORIA, cruciales para la seguridad del sistema y el control de acceso basado en roles.
Tabla: USUARIOS
Esta tabla almacena la información de los usuarios que acceden al sistema, incluyendo sus credenciales seguras y el rol asignado, lo cual es fundamental para el control de acceso.
Columna
Tipo de Dato
Restricciones
Descripción
id_usuario
INT
PK
Identificador único para cada usuario del sistema.
nombre
VARCHAR
Nombre de usuario, utilizado para el inicio de sesión.
hash_password
VARCHAR
Contraseña del usuario almacenada de forma segura mediante una función de hash.
id_role
INT
FK a ROLES.id_role
Clave foránea que vincula al usuario con un rol específico, definiendo sus permisos.
activo

BOOLEAN
Indica si la cuenta del usuario está activa (TRUE) o bloqueada/inactiva (FALSE).
Tabla: ROLES
Esta tabla define los diferentes tipos de roles dentro del sistema y los permisos asociados a cada uno, implementando el control de acceso basado en roles.
Columna
Tipo de Dato
Restricciones
Descripción
id_role
INT
PK
Identificador único para cada rol.
nombre_rol
VARCHAR
Nombre descriptivo del rol (ej., "Admin Principal", "Asistente", "Invitado (Solo Lectura)").
permisos
JSON
Estructura JSON que detalla las capacidades y permisos asociados a este rol.
Tabla: LOGS_AUDITORIA
Esta tabla es esencial para la seguridad y el monitoreo, registrando cronológicamente todas las actividades y cambios realizados en el sistema. Los logs son inmutables (Write Once, Read Many - WORM).
Columna
Tipo de Dato
Restricciones
Descripción
id_log
INT
PK
Identificador único y autoincremental para cada entrada del log de auditoría.
tabla_afectada
VARCHAR
Nombre de la tabla de la base de datos que fue afectada por la operación.
id_registro_afectado
INT
ID del registro específico dentro de la tabla_afectada que fue modificado o accedido.
tipo_operacion
VARCHAR
Tipo de operación realizada (ej., 'login_exitoso', 'login_fallido', 'crear', 'actualizar', 'eliminar').
datos_anteriores
JSON
Almacena el estado de los datos del registro afectado antes de la operación (útil para actualizaciones).
datos_nuevos
JSON
Almacena el estado de los datos del registro afectado después de la operación.
fecha_hora
DATETIME
Marca de tiempo de cuándo ocurrió la operación.
id_usuario_autor
INT
FK a USUARIOS.id_usuario
Clave foránea que identifica al usuario que realizó la operación registrada.
Nota sobre restricciones adicionales: La información proporcionada para estas tablas en las fuentes no especifica explícitamente si las columnas deben ser NOT NULL, si son AUTO_INCREMENT (además de la clave primaria), o si tienen valores por defecto. Si esta granularidad es requerida, sería necesario definirla en una etapa posterior del diseño de la base de datos.

3. Estrategia de Indexación

La indexación es un componente crítico para optimizar el rendimiento de la base de datos, especialmente en operaciones de búsqueda, filtrado y ordenamiento de datos. Los índices permiten a la base de datos localizar rápidamente las filas de una tabla sin tener que escanear todas las filas, lo que reduce significativamente los tiempos de respuesta de las consultas y mejora la eficiencia operativa general del sistema.
Basado en los requisitos funcionales del "Ecosistema Digital Inteligente para Cirugía Especial", las siguientes columnas son candidatas clave para la creación de índices, dada su frecuente participación en operaciones de consulta:

- Tabla USUARIOS:
  - nombre: Fundamental para la autenticación de usuarios y búsquedas de personal, ya que el usuario ingresa su "Nombre de Usuario" para iniciar sesión.
  - id_role: Como clave foránea, esta columna será utilizada frecuentemente en uniones (JOINs) con la tabla ROLES para determinar los permisos de acceso del usuario, lo cual es vital para el control de acceso basado en roles.
  - activo: Para filtrar rápidamente los usuarios activos e inactivos del sistema.
- Tabla ROLES:
  - nombre_rol: Si se realizan búsquedas o se necesita filtrar roles por su nombre descriptivo (ej., "Admin Principal", "Asistente").
- Tabla LOGS_AUDITORIA:
  - fecha_hora: Esencial para el monitoreo de seguridad y el seguimiento cronológico de eventos, permitiendo consultas rápidas basadas en rangos de tiempo.
  - tabla_afectada y id_registro_afectado: Combinadas, estas columnas son cruciales para auditar cambios específicos en registros de tablas determinadas. Un índice compuesto o índices individuales en estas columnas facilitarían la recuperación de logs relacionados con una entidad específica.
  - tipo_operacion: Para filtrar registros de auditoría por tipo de evento (ej., 'login_exitoso', 'crear', 'actualizar', 'eliminar').
  - id_usuario_autor: Como clave foránea, se usará para identificar rápidamente todas las acciones realizadas por un usuario específico.
- Candidatos de Indexación en Tablas de Datos del Paciente (Inferido de Requisitos Funcionales): Aunque la especificación detallada de estas tablas no fue solicitada en esta sección, los requisitos funcionales del Dashboard implican la necesidad de indexación en ellas:
  - Tablas de Pacientes: Las columnas utilizadas para la "Búsqueda Avanzada" de pacientes por "nombre, ID, palabra clave en notas" son candidatas principales para índices, incluyendo nombre, apellido, y posiblemente campos relevantes en HISTORIAL_CLINICO.
  - Tablas de Citas: fecha_hora es crucial para la visualización de "Citas del Día" y la gestión de la agenda.
  - Tablas de Documentos: id_registro_afectado (o el FK al paciente/historial) y nombre_archivo para la "Gestión Documental Avanzada".

La implementación de estos índices se realizará para optimizar las consultas más frecuentes y críticas del sistema, garantizando la velocidad y la reactividad necesarias para una experiencia de usuario fluida.

4. Plan de Migración de Datos Existentes

La migración de datos existentes es una etapa crucial en la implementación del "Ecosistema Digital Inteligente para Cirugía Especial", particularmente la carga de los archivos digitales iniciales que el consultorio del Dr. Sánchez García ya posee.
El proceso general de migración de datos se realizará a un alto nivel de la siguiente manera:

1. Identificación y Recopilación de Archivos Digitales: Se identificarán y recopilarán todos los archivos digitales relevantes existentes en el consultorio (PDFs, imágenes como RX, videos cortos, etc.) que deben ser integrados en el nuevo sistema.
2. Extracción de Metadatos: Para cada archivo, se extraerá la información relevante (metadatos) que permitirá vincularlo correctamente en la base de datos relacional. Esto incluirá, pero no se limitará a, el nombre del archivo, tipo de archivo, fecha de subida (o creación original si es posible determinarla), y lo más importante, el identificador del paciente o del registro del historial clínico al que pertenece.
3. Carga de Archivos a Cloud Storage: Los archivos digitales en sí se subirán de forma segura a Google Cloud Storage (equivalente a S3/Blob). Este servicio asegura el almacenamiento centralizado y seguro, con los datos cifrados en reposo.
4. Ingesta de Metadatos en Cloud SQL: Los metadatos extraídos de los archivos, junto con cualquier otra información estructurada existente (ej., datos demográficos de pacientes

que ya tienen historial digital), se insertarán en la Base de Datos Relacional (Cloud SQL - MySQL/PostgreSQL). En la tabla DOCUMENTOS, se almacenará la url_almacenamiento que apunta a la ubicación del archivo en Cloud Storage, asegurando el vínculo entre el registro de la base de datos y el archivo físico.
5. Verificación y Validación: Una vez completada la migración, se realizarán procesos de verificación y validación para asegurar la integridad de los datos, la correcta asociación de archivos con los pacientes/historiales, y que todos los archivos sean accesibles desde el sistema.
La migración se enfocará en asegurar la integridad y seguridad de la información desde el primer día.
5. Especificaciones de Seguridad de la Base de Datos
La seguridad de la base de datos es un pilar fundamental del "Ecosistema Digital Inteligente para Cirugía Especial", dada la sensibilidad de la información de los pacientes. Las siguientes especificaciones de seguridad son primordiales para la protección de los datos:
1. Configuración de Usuarios y Permisos de Acceso (Control de Acceso Basado en Roles - RBAC):
    ◦ La base de datos, específicamente la tabla USUARIOS, integra un id_role que se vincula a la tabla ROLES, la cual define los permisos de cada tipo de usuario mediante una estructura JSON. Esto implementa un estricto "Control de Acceso Basado en Roles".
    ◦ Los roles definidos incluyen "Admin Principal" (control total, gestión de usuarios y logs), "Admin Secundario" (todo excepto gestión de usuarios y logs), "Asistente" (gestión de Pacientes, Citas, Documentos, Ver Reportes) e "Invitado (Solo Lectura)" (ver Pacientes, Citas, Reportes).
    ◦ A nivel de la base de datos (Cloud SQL), se configurarán usuarios con el "principio de privilegio mínimo", de modo que el backend (Servidor de Aplicación) solo tenga los permisos necesarios para realizar las operaciones de lectura, escritura y modificación de datos que requiera la lógica de negocio, sin otorgar acceso directo innecesario a los usuarios finales.
2. Cifrado en Reposo para Datos Sensibles:
    ◦ Todos los datos sensibles almacenados en la "Base de Datos Relacional (Cloud SQL)" y en el "Almacenamiento de Archivos (Cloud Storage)" estarán cifrados en reposo.
    ◦ Para las contraseñas, se utilizarán funciones de hash seguras (indicado por el campo hash_password en la tabla USUARIOS) en lugar de almacenar las contraseñas en texto plano. Esto protege contra la exposición de credenciales en caso de una brecha de datos.
3. Plan de Copias de Seguridad y Restauración:
    ◦ Aunque los detalles específicos no se proporcionan, un componente esencial de la seguridad y la continuidad operativa de una base de datos en la nube como Cloud SQL es la implementación de un robusto plan de copias de seguridad (backups) automáticas y la capacidad de restauración (recovery).
    ◦ Esto asegurará la disponibilidad de los datos en caso de fallos, corrupción de datos o desastres, permitiendo la recuperación rápida a un estado anterior y minimizando el tiempo de inactividad del sistema.
    ◦ Las copias de seguridad incluirán tanto la base de datos estructurada como los archivos almacenados en Cloud Storage.
Nota: La información proporcionada en las fuentes no especifica detalles técnicos granulares sobre las configuraciones exactas de indexación (ej., tipos de índices como B-tree, hash), el proceso de migración a nivel de script o herramientas, ni las configuraciones específicas de usuarios y roles dentro de Cloud SQL o la frecuencia y retención del plan de copias de seguridad. Estos detalles se definirán durante la fase de implementación y configuración de la base de datos para alinearse con las mejores prácticas de GCP y los requisitos de rendimiento específicos.

Continuando con el "Documento de Diseño de la Base de Datos (Cloud SQL)", a continuación se presentan las secciones adicionales solicitadas:
8. Estrategia de Indexación
La estrategia de indexación es fundamental para optimizar el rendimiento de la base de datos, especialmente en operaciones de búsqueda y filtrado de datos. Los índices permiten que el motor de la base de datos localice rápidamente filas sin tener que escanear una tabla entera, mejorando drásticamente la velocidad de las consultas. Dada la necesidad de "búsqueda instantánea de pacientes" y el acceso eficiente a historiales clínicos, una correcta indexación es crucial para la "eficiencia operativa" del ecosistema.
Candidatos para la Creación de Índices:

Basándose en los requisitos funcionales del "Ecosistema Digital Inteligente para Cirugía Especial", las siguientes columnas son candidatas primarias para la creación de índices, incluyendo índices únicos para claves primarias y foráneas, y índices no únicos para columnas frecuentemente utilizadas en búsquedas o filtros:

- PACIENTES Tabla:
  - id_paciente: Clave primaria, se indexará automáticamente. Es crucial para la búsqueda de pacientes por ID y la vinculación a otras tablas como HISTORIAL_CLINICO y CITAS.
  - nombre, apellido: Estas columnas serán utilizadas frecuentemente para la "búsqueda instantánea de pacientes" por nombre o apellido. Un índice combinado o índices separados en estas columnas acelerarán las búsquedas.
  - activo: Utilizada para filtrar pacientes activos en el sistema.
- HISTORIAL_CLINICO Tabla:
  - id_historial: Clave primaria, se indexará automáticamente.
  - id_paciente: Clave foránea, esencial para buscar el historial clínico de un paciente específico. La indexación de esta FK optimizará las uniones (JOINs) y las búsquedas por paciente.
  - fecha_consulta: Es probable que se realicen búsquedas y filtros por fecha de consulta para el "Registro Historial Clínico Detallado".
- CITAS Tabla:
  - id_cita: Clave primaria, se indexará automáticamente.
  - id_paciente: Clave foránea, utilizada para vincular citas a pacientes y para filtrar citas por paciente.
  - fecha_hora: Fundamental para el "Módulo de Agenda y Citas", permitiendo búsquedas rápidas por fecha y hora para visualizar las "Citas del Día".
  - estado: Utilizada para filtrar citas por su estado (ej., "Confirmada", "Cancelada", "Atendida").
- USUARIOS Tabla:
  - id_usuario: Clave primaria, se indexará automáticamente.
  - nombre: Utilizada para el login y la identificación de usuarios en los logs de auditoría.
  - id_role: Clave foránea, esencial para determinar los "permisos del usuario" y para la funcionalidad de "Seguridad, Roles y Accesos Rápidos".
- LOGS_AUDITORIA Tabla:
  - id_log: Clave primaria, se indexará automáticamente.
  - fecha_hora: Crucial para la recuperación cronológica de eventos en el "Servicio de Logs de Auditoría".
  - id_usuario_autor: Permite filtrar los logs por el usuario que realizó la acción.
  - tabla_afectada y id_registro_afectado: Permiten rastrear acciones sobre registros específicos. Un índice compuesto o índices individuales pueden ser beneficiosos aquí.

La implementación de estos índices se realizará durante la fase de modelado y diseño de la base de datos para asegurar el rendimiento óptimo del sistema desde el inicio.

9. Plan de Migración de Datos Existentes

La migración de datos existentes es una parte integral de la Fase 1 del proyecto, asegurando la "carga inicial de los documentos que usted y su equipo ya poseen en formato digital" para integrarlos al nuevo sistema. Este proceso busca "eliminar la dispersión de archivos" y centralizar la información.

El plan de migración de archivos digitales se abordará a alto nivel de la siguiente manera:

1. Identificación y Recopilación de Archivos: Se colaborará con el consultorio para identificar y recopilar todos los archivos digitales existentes (PDFs, imágenes de estudios como RX, videos cortos de procedimientos, etc.) que necesiten ser integrados al sistema. Es importante destacar que el proyecto NO incluye la digitalización (escaneo) de archivos físicos existentes; solo se migrarán los archivos que ya estén en formato digital.
2. Extracción y Estructuración de Metadatos:
   - Para cada archivo digital, se extraerán o se solicitarán los metadatos relevantes (ej., nombre del paciente asociado, fecha del documento, tipo de archivo, descripción).
   - Estos metadatos se estructurarán de acuerdo con el esquema de la "Base de Datos Relacional (Cloud SQL)". Por ejemplo, para los documentos de pacientes, se vincularán al id_historial o id_paciente en las tablas DOCUMENTOS o HISTORIAL_CLINICO.
3. Carga Segura a Cloud Storage:
   - Los archivos digitales grandes y pesados se subirán al servicio de "Almacenamiento

de Archivos (Cloud Storage - S3/Blob)" de Google Cloud Platform (GCP).
    ◦ Es crucial que, al igual que la base de datos, estos archivos estén cifrados en reposo en Cloud Storage, garantizando la seguridad de la información sensible del paciente.
4. Vinculación en la Base de Datos (Cloud SQL):
    ◦ Una vez que los archivos estén cargados en Cloud Storage, la "Base de Datos Relacional (Cloud SQL)" no almacenará los archivos directamente. En su lugar, la tabla DOCUMENTOS almacenará la url_almacenamiento (URL de acceso) que apunta a la ubicación segura del archivo en Cloud Storage, junto con otros metadatos como nombre_archivo y tipo_archivo.
    ◦ Esto permite que la aplicación de escritorio, a través del backend, acceda y visualice los documentos asociados al expediente del paciente de forma eficiente.
5. Verificación y Control de Calidad: Se realizarán verificaciones post-migración para asegurar que todos los archivos se hayan cargado correctamente, que los metadatos sean precisos y que las vinculaciones entre la base de datos y Cloud Storage funcionen según lo esperado.
La migración de datos se realizará en la Fase 1 del proyecto, aprovechando herramientas de Google Cloud Storage para este propósito.
10. Especificaciones de Seguridad de la Base de Datos
La seguridad de la base de datos es una "preocupación de seguridad de datos" fundamental que el ecosistema busca abordar. Se integra en múltiples capas de la arquitectura en la nube.
1. Configuración de Usuarios y Permisos de Acceso a la Base de Datos:
    ◦ El sistema implementa un estricto "Control de Acceso Basado en Roles (RBAC)". Al iniciar sesión, cada usuario es asociado a un rol específico ("Administrador", "Admin Principal", "Admin Secundario", "Asistente", "Invitado") que define sus permisos específicos sobre la información y funcionalidades.
    ◦ El backend verifica las credenciales contra la "Base de Datos Relacional (Cloud SQL)" y consulta el Sistema de Roles (ROLES en el modelo de datos) para determinar los permisos del usuario.
    ◦ Para la conectividad de la aplicación al backend y del backend a la base de datos, se aplicará el "principio de privilegio mínimo". Esto significa que el usuario de la base de datos que utiliza el backend para conectarse solo tendrá los permisos necesarios para realizar las operaciones requeridas (lectura, escritura, modificación en tablas específicas), sin privilegios administrativos excesivos.
    ◦ La gestión de usuarios y la asignación de roles se realiza a través del Dashboard, en la sección de "Administración", accesible solo para roles con permisos adecuados como el "Admin Principal".
2. Requisitos de Cifrado en Reposo para los Datos Sensibles:
    ◦ Una medida de seguridad crítica para proteger la información sensible del paciente es el cifrado de datos en reposo.
    ◦ Todos los datos sensibles estarán cifrados en reposo tanto en la "Base de Datos Relacional (Cloud SQL)" como en el "Almacenamiento de Archivos (Cloud Storage)". Esto asegura que, en caso de un acceso no autorizado al almacenamiento subyacente, los datos permanezcan ilegibles.
    ◦ Para las contraseñas de usuario, la base de datos incluye un campo hash_password VARCHAR en la tabla USUARIOS, lo que sugiere el uso de funciones de hash seguras en lugar de almacenar las contraseñas en texto plano. Esto es una práctica estándar esencial contra fugas de datos.
3. Plan de Copias de Seguridad y Restauración:
    ◦ Aunque no se detalla un plan específico de copias de seguridad en todos los documentos, la "Guía de Replicación Técnica" y "Arquitectura Nube" mencionan explícitamente la necesidad de "Implementar copias de seguridad automáticas y recuperación ante desastres" como una recomendación para la base de datos Cloud SQL.
    ◦ Esto es una buena práctica estándar para cualquier base de datos en producción, y la plataforma Google Cloud SQL ofrece capacidades de backup automático y point-in-time recovery, lo que garantiza la "alta disponibilidad" y minimiza el tiempo de inactividad. Un plan de restauración claro será crucial para mitigar el riesgo de pérdida de datos en caso de fallos del sistema o corrupción de datos.

6. Elección y Configuración Inicial de Cloud SQL
La base de datos relacional para el "Ecosistema Digital Inteligente para Cirugía Especial" se hospedará en Google Cloud SQL. Cloud SQL es un servicio de base de datos completamente

gestionado por Google Cloud Platform (GCP) que facilita la configuración, el mantenimiento, la gestión y el escalado de bases de datos relacionales en la nube.
Elección del Motor de Base de Datos (MySQL o PostgreSQL): Las fuentes del proyecto indican que se pueden utilizar sistemas de gestión de bases de datos como MySQL o PostgreSQL. La elección final entre MySQL y PostgreSQL se determinará en una fase posterior, basándose en consideraciones más detalladas, como:
• Requisitos Específicos de Funcionalidad: Si alguna de las bases de datos ofrece características o extensiones únicas que sean particularmente ventajosas para las funcionalidades avanzadas del ecosistema.
• Experiencia del Equipo de Desarrollo: La familiaridad y experiencia del equipo de desarrollo con un motor de base de datos en particular pueden influir en la elección para optimizar la eficiencia y reducir la curva de aprendizaje.
• Patrones de Uso y Rendimiento Esperado: Aunque ambos son robustos, pueden tener diferencias sutiles en el rendimiento para ciertos tipos de cargas de trabajo (ej., operaciones de lectura intensivas vs. escritura intensivas, complejidad de consultas).
• Consideraciones de Licenciamiento y Comunidad: MySQL es de código abierto con diversas opciones de licenciamiento, mientras que PostgreSQL es completamente de código abierto y conocido por su fuerte adherencia a los estándares SQL y su comunidad activa.
Independientemente del motor elegido, ambos son compatibles con Cloud SQL y ofrecen características de alta disponibilidad y recuperación de desastres.
Configuración Inicial de la Instancia en GCP (Alto Nivel): El proceso de configuración inicial de la instancia de Cloud SQL en GCP implica los siguientes pasos clave:
1. Creación de la Instancia:
    ◦ Acceder a la consola de Google Cloud Platform y navegar a la sección de Cloud SQL.
    ◦ Seleccionar la opción para crear una nueva instancia de base de datos.
    ◦ Elegir el motor de base de datos deseado (MySQL o PostgreSQL).
    ◦ Especificar la región donde se desplegará la instancia para asegurar baja latencia con el Servidor de Aplicación (Backend).
    ◦ Definir la configuración de la máquina (tipo de CPU, memoria) y el tipo de almacenamiento (ej., SSD) y su tamaño inicial.
2. Configuración de Red y Conectividad:
    ◦ Configurar la instancia para que opere dentro de la Virtual Private Cloud (VPC) del proyecto. Esto asegura que la base de datos esté en una red aislada y segura, fundamental para la seguridad de los datos.
    ◦ Establecer la conectividad de IP privada para el backend, lo que mejora la seguridad y el rendimiento al mantener el tráfico de la base de datos dentro de la red privada de Google.
    ◦ Configurar las redes autorizadas y reglas de firewall para permitir solo el tráfico desde el Servidor de Aplicación (Backend) y otras IPs de administración necesarias.
3. Creación de Usuarios Iniciales:
    ◦ Crear usuarios de base de datos con contraseñas seguras y el "principio de privilegio mínimo". Es decir, el usuario que utilizará el backend para conectarse a la base de datos solo tendrá los permisos necesarios para las operaciones que realice (lectura, escritura, modificación de tablas específicas), sin privilegios administrativos excesivos.
    ◦ Para la gestión de contraseñas de usuarios en la aplicación, se utilizarán funciones de hash seguras para almacenar hash_password en la tabla USUARIOS.
7. Consideraciones Adicionales
Además de las secciones anteriores, existen otras consideraciones importantes para el diseño y la implementación de la base de datos:
1. Proyección y Gestión del Almacenamiento:
    ◦ El servicio de infraestructura en la nube de Google Cloud incluye una capacidad de hasta 1 Terabyte (1000 GB) mensual como cortesía durante los primeros seis meses.
    ◦ Si bien las fuentes no mencionan explícitamente una proyección de 2TB, cualquier aumento en la capacidad de almacenamiento más allá del 1 Terabyte incluido tendrá un costo adicional.
    ◦ Es fundamental aclarar que este almacenamiento (tanto el inicial de 1TB como cualquier aumento futuro) incluye tanto el espacio utilizado por la Base de Datos Relacional (Cloud SQL) para los datos estructurados como el espacio para los archivos digitales grandes almacenados en Cloud Storage. Se deberá monitorear de cerca el consumo de almacenamiento para gestionar los costos de manera eficiente.
2. Distinción Clara entre Datos Estructurados y Archivos Grandes:
    ◦ Se reafirma que la Base de Datos Relacional (Cloud SQL) se utilizará exclusivamente

para almacenar información estructurada y crítica como perfiles de pacientes, historiales clínicos, citas, usuarios y roles.
    ◦ El Almacenamiento de Archivos (Cloud Storage - S3/Blob) se designará para el almacenamiento centralizado y seguro de archivos digitales pesados, como PDFs, imágenes (ej., RX) y videos cortos asociados a los expedientes de los pacientes. Esta separación optimiza el rendimiento y el costo de almacenamiento, y la base de datos Cloud SQL almacenará únicamente la url_almacenamiento que apunta a la ubicación del archivo en Cloud Storage.
3. Seguridad de los Datos Sensibles (Cifrado):
    ◦ Una consideración primordial es que todos los datos sensibles estarán cifrados en reposo, tanto en la Base de Datos Relacional (Cloud SQL) como en el Almacenamiento de Archivos (Cloud Storage). Esto es una medida de seguridad crítica para proteger la información del paciente contra accesos no autorizados.
    ◦ Las contraseñas de usuario se almacenarán utilizando funciones de hash seguras (campo hash_password) en lugar de texto plano.
4. Escalabilidad y Alta Disponibilidad:
    ◦ La elección de GCP y sus servicios (Cloud SQL, Cloud Storage) proporciona inherentemente escalabilidad (la capacidad de expandirse para manejar un mayor volumen de pacientes y datos sin interrupciones) y alta disponibilidad (minimizando el tiempo de inactividad del sistema). Estas características son fundamentales para la continuidad operativa del consultorio.
5. Requisito de Conectividad:
    ◦ Para un rendimiento óptimo del ecosistema, se requiere una conexión a internet estable de al menos 250 Mbps. La dependencia de la infraestructura en la nube hace que la calidad de la conexión sea crucial.
6. Vinculación del Costo de la Nube al Plan de Mantenimiento:
    ◦ La cortesía de Google Cloud por los primeros seis meses está directamente vinculada a la vigencia del Plan de Mantenimiento y Soporte mensual. Si el cliente decide no continuar con el plan de mantenimiento durante este periodo, la cortesía del servicio en la nube finalizará en ese mismo momento. Esta es una condición importante que la alta dirección debe considerar.
Con estas secciones, el "Documento de Diseño de la Base de Datos (Cloud SQL)" está completo.

Documento de Diseño de la Base de Datos (Cloud SQL)

Este documento detalla el diseño de la base de datos relacional para el "Ecosistema Digital Inteligente para Cirugía Especial", la cual se alojará en Google Cloud Platform (GCP) utilizando Cloud SQL (MySQL/PostgreSQL). El objetivo es centralizar toda la información crítica del consultorio de manera segura, garantizando la integridad de los datos sensibles mediante cifrado en reposo.

1. Modelo Entidad-Relación (ERD)

El modelo Entidad-Relación (ERD) describe las principales entidades y sus interconexiones dentro del ecosistema digital, sirviendo como la columna vertebral para la gestión de pacientes y la operación diaria del consultorio. Este modelo asegura que la información estructurada, como perfiles de pacientes, historiales clínicos, citas y usuarios, se almacene de forma coherente y segura.

Las entidades principales y sus relaciones son las siguientes:

• PACIENTES y HISTORIAL_CLINICO: La entidad PACIENTES "tiene" una relación de uno a muchos con HISTORIAL_CLINICO, lo que significa que un paciente puede tener múltiples entradas en su historial clínico a lo largo del tiempo.

• HISTORIAL_CLINICO y DOCUMENTOS: La entidad HISTORIAL_CLINICO "contiene" DOCUMENTOS. Esto permite adjuntar archivos digitales como PDFs, imágenes o videos cortos a entradas específicas del historial clínico de un paciente.

• PACIENTES y CITAS: La entidad PACIENTES "agenda" CITAS. Un paciente puede tener múltiples citas programadas con el consultorio.

• CITAS y PROCEDIMIENTOS_CITAS: La entidad CITAS "se relaciona con" PROCEDIMIENTOS_CITAS. Esta tabla intermedia permite asociar una o más citas con procedimientos específicos.

• PROCEDIMIENTOS_CITAS y PROCEDIMIENTOS: La entidad PROCEDIMIENTOS_CITAS "detalla" PROCEDIMIENTOS. Esto vincula los procedimientos realizados con las citas correspondientes.

• USUARIOS y ROLES: La entidad ROLES "asigna" USUARIOS. Cada usuario del sistema tiene un rol definido que determina sus permisos de acceso a la información y funcionalidades.

• USUARIOS y LOGS_AUDITORIA: La entidad USUARIOS "realiza" operaciones que son registradas en LOGS_AUDITORIA. Este componente registra cronológicamente todas las actividades y cambios realizados en el sistema, siendo vital para la seguridad y el seguimiento de operaciones.

2. Especificación Completa de Tablas (Parte 1)

A continuación, se detalla la estructura y las características de las tablas PACIENTES e HISTORIAL_CLINICO, incluyendo sus columnas, tipos de datos, restricciones y relaciones.

Tabla: PACIENTES

Esta tabla centraliza la información demográfica y de contacto de todos los pacientes del consultorio.

Columna
Tipo de Dato
Restricciones
Descripción
id_paciente
INT
PK, NOT NULL, AUTO_INCREMENT
Identificador único y autoincremental para cada paciente.
nombre
VARCHAR(255)
NOT NULL
Nombre(s) del paciente.
apellido
VARCHAR(255)
NOT NULL
Apellido(s) del paciente.
fecha_nac
DATE
NULLABLE
Fecha de nacimiento del paciente.
datos_contacto
JSON
NULLABLE
Almacena información de contacto adicional (ej., teléfono, email, dirección) en formato JSON.

activo
BOOLEAN
NOT NULL, DEFAULT TRUE
Indica si el registro del paciente está activo (TRUE) o ha sido eliminado suavemente/archivado (FALSE).
Tabla: HISTORIAL_CLINICO
Esta tabla almacena el registro cronológico y detallado de cada interacción clínica con un paciente.
Columna
Tipo de Dato
Restricciones
Descripción
id_historial
INT
PK, NOT NULL, AUTO_INCREMENT
Identificador único y autoincremental para cada entrada del historial clínico.
id_paciente
INT
FK a PACIENTES.id_paciente, NOT NULL
Clave foránea que vincula la entrada del historial a un paciente específico.
fecha_consulta
DATE
NOT NULL
Fecha en la que se realizó la consulta o se registró la entrada.
diagnostico
TEXT
NULLABLE
Diagnóstico asociado a la consulta.
notas_medico
TEXT
NULLABLE
Notas detalladas del médico sobre la evolución, procedimientos o tratamientos.
activo
BOOLEAN
NOT NULL, DEFAULT TRUE
Indica si la entrada del historial está activa (TRUE) o ha sido marcada como inactiva/archivada (FALSE).

2. Especificación Completa de Tablas (Parte 2)
Se presenta la estructura detallada de las tablas USUARIOS, ROLES y LOGS_AUDITORIA, cruciales para la seguridad del sistema y el control de acceso basado en roles.
Tabla: USUARIOS
Esta tabla almacena la información de los usuarios que acceden al sistema, incluyendo sus credenciales seguras y el rol asignado, lo cual es fundamental para el control de acceso.
Columna
Tipo de Dato
Restricciones
Descripción
id_usuario
INT
PK
Identificador único para cada usuario del sistema.
nombre
VARCHAR
Nombre de usuario, utilizado para el inicio de sesión.
hash_password
VARCHAR
Contraseña del usuario almacenada de forma segura mediante una función de hash.
id_role
INT
FK a ROLES.id_role
Clave foránea que vincula al usuario con un rol específico, definiendo sus permisos.
activo
BOOLEAN
Indica si la cuenta del usuario está activa (TRUE) o bloqueada/inactiva (FALSE).
Tabla: ROLES
Esta tabla define los diferentes tipos de roles dentro del sistema y los permisos asociados a cada uno, implementando el control de acceso basado en roles.
Columna
Tipo de Dato
Restricciones
Descripción
id_role
INT
PK
Identificador único para cada rol.
nombre_rol
VARCHAR
Nombre descriptivo del rol (ej., "Admin Principal", "Asistente", "Invitado (Solo Lectura)").
permisos
JSON
Estructura JSON que detalla las capacidades y permisos asociados a este rol.
Tabla: LOGS_AUDITORIA
Esta tabla es esencial para la seguridad y el monitoreo, registrando cronológicamente todas las actividades y cambios realizados en el sistema. Los logs son inmutables (Write Once, Read Many - WORM).
Columna
Tipo de Dato
Restricciones
Descripción
id_log
INT
PK
Identificador único y autoincremental para cada entrada del log de auditoría.
tabla_afectada
VARCHAR
Nombre de la tabla de la base de datos que fue afectada por la operación.
id_registro_afectado
INT

ID del registro específico dentro de la tabla_afectada que fue modificado o accedido.
tipo_operacion
VARCHAR
Tipo de operación realizada (ej., 'login_exitoso', 'login_fallido', 'crear', 'actualizar', 'eliminar').
datos_anteriores
JSON
Almacena el estado de los datos del registro afectado antes de la operación (útil para actualizaciones).
datos_nuevos
JSON
Almacena el estado de los datos del registro afectado después de la operación.
fecha_hora
DATETIME
Marca de tiempo de cuándo ocurrió la operación.
id_usuario_autor
INT
FK a USUARIOS.id_usuario
Clave foránea que identifica al usuario que realizó la operación registrada.
Nota sobre restricciones adicionales: La información proporcionada para estas tablas en las fuentes no especifica explícitamente si las columnas deben ser NOT NULL, si son AUTO_INCREMENT (además de la clave primaria), o si tienen valores por defecto. Si esta granularidad es requerida, sería necesario definirla en una etapa posterior del diseño de la base de datos.

3. Estrategia de Indexación

La indexación es un componente crítico para optimizar el rendimiento de la base de datos, especialmente en operaciones de búsqueda, filtrado y ordenamiento de datos. Los índices permiten a la base de datos localizar rápidamente las filas de una tabla sin tener que escanear todas las filas, lo que reduce significativamente los tiempos de respuesta de las consultas y mejora la eficiencia operativa general del sistema.

Basado en los requisitos funcionales del "Ecosistema Digital Inteligente para Cirugía Especial", las siguientes columnas son candidatas clave para la creación de índices, dada su frecuente participación en operaciones de consulta:

- Tabla USUARIOS:
  - nombre: Fundamental para la autenticación de usuarios y búsquedas de personal, ya que el usuario ingresa su "Nombre de Usuario" para iniciar sesión.
  - id_role: Como clave foránea, esta columna será utilizada frecuentemente en uniones (JOINs) con la tabla ROLES para determinar los permisos de acceso del usuario, lo cual es vital para el control de acceso basado en roles.
  - activo: Para filtrar rápidamente los usuarios activos e inactivos del sistema.
- Tabla ROLES:
  - nombre_rol: Si se realizan búsquedas o se necesita filtrar roles por su nombre descriptivo (ej., "Admin Principal", "Asistente").
- Tabla LOGS_AUDITORIA:
  - fecha_hora: Esencial para el monitoreo de seguridad y el seguimiento cronológico de eventos, permitiendo consultas rápidas basadas en rangos de tiempo.
  - tabla_afectada y id_registro_afectado: Combinadas, estas columnas son cruciales para auditar cambios específicos en registros de tablas determinadas. Un índice compuesto o índices individuales en estas columnas facilitarían la recuperación de logs relacionados con una entidad específica.
  - tipo_operacion: Para filtrar registros de auditoría por tipo de evento (ej., 'login_exitoso', 'crear', 'actualizar', 'eliminar').
  - id_usuario_autor: Como clave foránea, se usará para identificar rápidamente todas las acciones realizadas por un usuario específico.
- Candidatos de Indexación en Tablas de Datos del Paciente (Inferido de Requisitos Funcionales): Aunque la especificación detallada de estas tablas no fue solicitada en esta sección, los requisitos funcionales del Dashboard implican la necesidad de indexación en ellas:
  - Tablas de Pacientes: Las columnas utilizadas para la "Búsqueda Avanzada" de pacientes por "nombre, ID, palabra clave en notas" son candidatas principales para índices, incluyendo nombre, apellido, y posiblemente campos relevantes en HISTORIAL_CLINICO.
  - Tablas de Citas: fecha_hora es crucial para la visualización de "Citas del Día" y la gestión de la agenda.
  - Tablas de Documentos: id_registro_afectado (o el FK al paciente/historial) y nombre_archivo para la "Gestión Documental Avanzada".

La implementación de estos índices se realizará para optimizar las consultas más frecuentes y críticas del sistema, garantizando la velocidad y la reactividad necesarias para una experiencia de usuario fluida.

4. Plan de Migración de Datos Existentes

La migración de datos existentes es una etapa crucial en la implementación del "Ecosistema Digital Inteligente para Cirugía Especial", particularmente la carga de los archivos digitales iniciales que el consultorio del Dr. Sánchez García ya posee.

El proceso general de migración de datos se realizará a un alto nivel de la siguiente manera:

1. Identificación y Recopilación de Archivos Digitales: Se identificarán y recopilarán todos los archivos digitales relevantes existentes en el consultorio (PDFs, imágenes como RX, videos cortos, etc.) que deben ser integrados en el nuevo sistema.
2. Extracción de Metadatos: Para cada archivo, se extraerá la información relevante (metadatos) que permitirá vincularlo correctamente en la base de datos relacional. Esto incluirá, pero no se limitará a, el nombre del archivo, tipo de archivo, fecha de subida (o creación original si es posible determinarla), y lo más importante, el identificador del paciente o del registro del historial clínico al que pertenece.
3. Carga de Archivos a Cloud Storage: Los archivos digitales en sí se subirán de forma segura a Google Cloud Storage (equivalente a S3/Blob). Este servicio asegura el almacenamiento centralizado y seguro, con los datos cifrados en reposo.

4. Ingesta de Metadatos en Cloud SQL: Los metadatos extraídos de los archivos, junto con cualquier otra información estructurada existente (ej., datos demográficos de pacientes que ya tienen historial digital), se insertarán en la Base de Datos Relacional (Cloud SQL - MySQL/PostgreSQL). En la tabla DOCUMENTOS, se almacenará la url_almacenamiento que apunta a la ubicación del archivo en Cloud Storage, asegurando el vínculo entre el registro de la base de datos y el archivo físico.
5. Verificación y Validación: Una vez completada la migración, se realizarán procesos de verificación y validación para asegurar la integridad de los datos, la correcta asociación de archivos con los pacientes/historiales, y que todos los archivos sean accesibles desde el sistema.
La migración se enfocará en asegurar la integridad y seguridad de la información desde el primer día.
5. Especificaciones de Seguridad de la Base de Datos
La seguridad de la base de datos es un pilar fundamental del "Ecosistema Digital Inteligente para Cirugía Especial", dada la sensibilidad de la información de los pacientes. Las siguientes especificaciones de seguridad son primordiales para la protección de los datos:
1. Configuración de Usuarios y Permisos de Acceso (Control de Acceso Basado en Roles - RBAC):
    ◦ La base de datos, específicamente la tabla USUARIOS, integra un id_role que se vincula a la tabla ROLES, la cual define los permisos de cada tipo de usuario mediante una estructura JSON. Esto implementa un estricto "Control de Acceso Basado en Roles".
    ◦ Los roles definidos incluyen "Admin Principal" (control total, gestión de usuarios y logs), "Admin Secundario" (todo excepto gestión de usuarios y logs), "Asistente" (gestión de Pacientes, Citas, Documentos, Ver Reportes) e "Invitado (Solo Lectura)" (ver Pacientes, Citas, Reportes).
    ◦ A nivel de la base de datos (Cloud SQL), se configurarán usuarios con el "principio de privilegio mínimo", de modo que el backend (Servidor de Aplicación) solo tenga los permisos necesarios para realizar las operaciones de lectura, escritura y modificación de datos que requiera la lógica de negocio, sin otorgar acceso directo innecesario a los usuarios finales.
2. Cifrado en Reposo para Datos Sensibles:
    ◦ Todos los datos sensibles almacenados en la "Base de Datos Relacional (Cloud SQL)" y en el "Almacenamiento de Archivos (Cloud Storage)" estarán cifrados en reposo.
    ◦ Para las contraseñas, se utilizarán funciones de hash seguras (indicado por el campo hash_password en la tabla USUARIOS) en lugar de almacenar las contraseñas en texto plano. Esto protege contra la exposición de credenciales en caso de una brecha de datos.
3. Plan de Copias de Seguridad y Restauración:
    ◦ Aunque los detalles específicos no se proporcionan, un componente esencial de la seguridad y la continuidad operativa de una base de datos en la nube como Cloud SQL es la implementación de un robusto plan de copias de seguridad (backups) automáticas y la capacidad de restauración (recovery).
    ◦ Esto asegurará la disponibilidad de los datos en caso de fallos, corrupción de datos o desastres, permitiendo la recuperación rápida a un estado anterior y minimizando el tiempo de inactividad del sistema.
    ◦ Las copias de seguridad incluirán tanto la base de datos estructurada como los archivos almacenados en Cloud Storage.
Nota: La información proporcionada en las fuentes no especifica detalles técnicos granulares sobre las configuraciones exactas de indexación (ej., tipos de índices como B-tree, hash), el proceso de migración a nivel de script o herramientas, ni las configuraciones específicas de usuarios y roles dentro de Cloud SQL o la frecuencia y retención del plan de copias de seguridad. Estos detalles se definirán durante la fase de implementación y configuración de la base de datos para alinearse con las mejores prácticas de GCP y los requisitos de rendimiento específicos.

Continuando con el "Documento de Diseño de la Base de Datos (Cloud SQL)", a continuación se presentan las secciones adicionales solicitadas:

8. Estrategia de Indexación

La estrategia de indexación es fundamental para optimizar el rendimiento de la base de datos, especialmente en operaciones de búsqueda y filtrado de datos. Los índices permiten que el motor de la base de datos localice rápidamente filas sin tener que escanear una tabla entera, mejorando drásticamente la velocidad de las consultas. Dada la necesidad de "búsqueda instantánea de pacientes" y el acceso eficiente a historiales clínicos, una correcta indexación es crucial para la "eficiencia operativa" del ecosistema.

Candidatos para la Creación de Índices:

Basándose en los requisitos funcionales del "Ecosistema Digital Inteligente para Cirugía Especial", las siguientes columnas son candidatas primarias para la creación de índices, incluyendo índices únicos para claves primarias y foráneas, y índices no únicos para columnas frecuentemente utilizadas en búsquedas o filtros:

- PACIENTES Tabla:
  - id_paciente: Clave primaria, se indexará automáticamente. Es crucial para la búsqueda de pacientes por ID y la vinculación a otras tablas como HISTORIAL_CLINICO y CITAS.
  - nombre, apellido: Estas columnas serán utilizadas frecuentemente para la "búsqueda instantánea de pacientes" por nombre o apellido. Un índice combinado o índices separados en estas columnas acelerarán las búsquedas.
  - activo: Utilizada para filtrar pacientes activos en el sistema.
- HISTORIAL_CLINICO Tabla:
  - id_historial: Clave primaria, se indexará automáticamente.
  - id_paciente: Clave foránea, esencial para buscar el historial clínico de un paciente específico. La indexación de esta FK optimizará las uniones (JOINs) y las búsquedas por paciente.
  - fecha_consulta: Es probable que se realicen búsquedas y filtros por fecha de consulta para el "Registro Historial Clínico Detallado".
- CITAS Tabla:
  - id_cita: Clave primaria, se indexará automáticamente.
  - id_paciente: Clave foránea, utilizada para vincular citas a pacientes y para filtrar citas por paciente.
  - fecha_hora: Fundamental para el "Módulo de Agenda y Citas", permitiendo búsquedas rápidas por fecha y hora para visualizar las "Citas del Día".
  - estado: Utilizada para filtrar citas por su estado (ej., "Confirmada", "Cancelada", "Atendida").
- USUARIOS Tabla:
  - id_usuario: Clave primaria, se indexará automáticamente.
  - nombre: Utilizada para el login y la identificación de usuarios en los logs de auditoría.
  - id_role: Clave foránea, esencial para determinar los "permisos del usuario" y para la funcionalidad de "Seguridad, Roles y Accesos Rápidos".
- LOGS_AUDITORIA Tabla:
  - id_log: Clave primaria, se indexará automáticamente.
  - fecha_hora: Crucial para la recuperación cronológica de eventos en el "Servicio de Logs de Auditoría".
  - id_usuario_autor: Permite filtrar los logs por el usuario que realizó la acción.
  - tabla_afectada y id_registro_afectado: Permiten rastrear acciones sobre registros específicos. Un índice compuesto o índices individuales pueden ser beneficiosos aquí.

La implementación de estos índices se realizará durante la fase de modelado y diseño de la base de datos para asegurar el rendimiento óptimo del sistema desde el inicio.

9. Plan de Migración de Datos Existentes

La migración de datos existentes es una parte integral de la Fase 1 del proyecto, asegurando la "carga inicial de los documentos que usted y su equipo ya poseen en formato digital" para integrarlos al nuevo sistema. Este proceso busca "eliminar la dispersión de archivos" y centralizar la información.

El plan de migración de archivos digitales se abordará a alto nivel de la siguiente manera:

1. Identificación y Recopilación de Archivos: Se colaborará con el consultorio para identificar y recopilar todos los archivos digitales existentes (PDFs, imágenes de estudios como RX, videos cortos de procedimientos, etc.) que necesiten ser integrados al

sistema. Es importante destacar que el proyecto NO incluye la digitalización (escaneo) de archivos físicos existentes; solo se migrarán los archivos que ya estén en formato digital.
2. Extracción y Estructuración de Metadatos:
    ◦ Para cada archivo digital, se extraerán o se solicitarán los metadatos relevantes (ej., nombre del paciente asociado, fecha del documento, tipo de archivo, descripción).
    ◦ Estos metadatos se estructurarán de acuerdo con el esquema de la "Base de Datos Relacional (Cloud SQL)". Por ejemplo, para los documentos de pacientes, se vincularán al id_historial o id_paciente en las tablas DOCUMENTOS o HISTORIAL_CLINICO.
3. Carga Segura a Cloud Storage:
    ◦ Los archivos digitales grandes y pesados se subirán al servicio de "Almacenamiento de Archivos (Cloud Storage - S3/Blob)" de Google Cloud Platform (GCP).
    ◦ Es crucial que, al igual que la base de datos, estos archivos estén cifrados en reposo en Cloud Storage, garantizando la seguridad de la información sensible del paciente.
4. Vinculación en la Base de Datos (Cloud SQL):
    ◦ Una vez que los archivos estén cargados en Cloud Storage, la "Base de Datos Relacional (Cloud SQL)" no almacenará los archivos directamente. En su lugar, la tabla DOCUMENTOS almacenará la url_almacenamiento (URL de acceso) que apunta a la ubicación segura del archivo en Cloud Storage, junto con otros metadatos como nombre_archivo y tipo_archivo.
    ◦ Esto permite que la aplicación de escritorio, a través del backend, acceda y visualice los documentos asociados al expediente del paciente de forma eficiente.
5. Verificación y Control de Calidad: Se realizarán verificaciones post-migración para asegurar que todos los archivos se hayan cargado correctamente, que los metadatos sean precisos y que las vinculaciones entre la base de datos y Cloud Storage funcionen según lo esperado.
La migración de datos se realizará en la Fase 1 del proyecto, aprovechando herramientas de Google Cloud Storage para este propósito.
10. Especificaciones de Seguridad de la Base de Datos
La seguridad de la base de datos es una "preocupación de seguridad de datos" fundamental que el ecosistema busca abordar. Se integra en múltiples capas de la arquitectura en la nube.
1. Configuración de Usuarios y Permisos de Acceso a la Base de Datos:
    ◦ El sistema implementa un estricto "Control de Acceso Basado en Roles (RBAC)". Al iniciar sesión, cada usuario es asociado a un rol específico ("Administrador", "Admin Principal", "Admin Secundario", "Asistente", "Invitado") que define sus permisos específicos sobre la información y funcionalidades.
    ◦ El backend verifica las credenciales contra la "Base de Datos Relacional (Cloud SQL)" y consulta el Sistema de Roles (ROLES en el modelo de datos) para determinar los permisos del usuario.
    ◦ Para la conectividad de la aplicación al backend y del backend a la base de datos, se aplicará el "principio de privilegio mínimo". Esto significa que el usuario de la base de datos que utiliza el backend para conectarse solo tendrá los permisos necesarios para realizar las operaciones requeridas (lectura, escritura, modificación en tablas específicas), sin privilegios administrativos excesivos.
    ◦ La gestión de usuarios y la asignación de roles se realiza a través del Dashboard, en la sección de "Administración", accesible solo para roles con permisos adecuados como el "Admin Principal".
2. Requisitos de Cifrado en Reposo para los Datos Sensibles:
    ◦ Una medida de seguridad crítica para proteger la información sensible del paciente es el cifrado de datos en reposo.
    ◦ Todos los datos sensibles estarán cifrados en reposo tanto en la "Base de Datos Relacional (Cloud SQL)" como en el "Almacenamiento de Archivos (Cloud Storage)". Esto asegura que, en caso de un acceso no autorizado al almacenamiento subyacente, los datos permanezcan ilegibles.
    ◦ Para las contraseñas de usuario, la base de datos incluye un campo hash_password VARCHAR en la tabla USUARIOS, lo que sugiere el uso de funciones de hash seguras en lugar de almacenar las contraseñas en texto plano. Esto es una práctica estándar esencial contra fugas de datos.
3. Plan de Copias de Seguridad y Restauración:
    ◦ Aunque no se detalla un plan específico de copias de seguridad en todos los documentos, la "Guía de Replicación Técnica" y "Arquitectura Nube" mencionan

explícitamente la necesidad de "Implementar copias de seguridad automáticas y recuperación ante desastres" como una recomendación para la base de datos Cloud SQL.

- Esto es una buena práctica estándar para cualquier base de datos en producción, y la plataforma Google Cloud SQL ofrece capacidades de backup automático y point-in-time recovery, lo que garantiza la "alta disponibilidad" y minimiza el tiempo de inactividad. Un plan de restauración claro será crucial para mitigar el riesgo de pérdida de datos en caso de fallos del sistema o corrupción de datos.

6. Elección y Configuración Inicial de Cloud SQL

La base de datos relacional para el "Ecosistema Digital Inteligente para Cirugía Especial" se hospedará en Google Cloud SQL. Cloud SQL es un servicio de base de datos completamente gestionado por Google Cloud Platform (GCP) que facilita la configuración, el mantenimiento, la gestión y el escalado de bases de datos relacionales en la nube.

Elección del Motor de Base de Datos (MySQL o PostgreSQL): Las fuentes del proyecto indican que se pueden utilizar sistemas de gestión de bases de datos como MySQL o PostgreSQL. La elección final entre MySQL y PostgreSQL se determinará en una fase posterior, basándose en consideraciones más detalladas, como:

- Requisitos Específicos de Funcionalidad: Si alguna de las bases de datos ofrece características o extensiones únicas que sean particularmente ventajosas para las funcionalidades avanzadas del ecosistema.
- Experiencia del Equipo de Desarrollo: La familiaridad y experiencia del equipo de desarrollo con un motor de base de datos en particular pueden influir en la elección para optimizar la eficiencia y reducir la curva de aprendizaje.
- Patrones de Uso y Rendimiento Esperado: Aunque ambos son robustos, pueden tener diferencias sutiles en el rendimiento para ciertos tipos de cargas de trabajo (ej., operaciones de lectura intensivas vs. escritura intensivas, complejidad de consultas).
- Consideraciones de Licenciamiento y Comunidad: MySQL es de código abierto con diversas opciones de licenciamiento, mientras que PostgreSQL es completamente de código abierto y conocido por su fuerte adherencia a los estándares SQL y su comunidad activa.

Independientemente del motor elegido, ambos son compatibles con Cloud SQL y ofrecen características de alta disponibilidad y recuperación de desastres.

Configuración Inicial de la Instancia en GCP (Alto Nivel): El proceso de configuración inicial de la instancia de Cloud SQL en GCP implica los siguientes pasos clave:

1. Creación de la Instancia:
    - Acceder a la consola de Google Cloud Platform y navegar a la sección de Cloud SQL.
    - Seleccionar la opción para crear una nueva instancia de base de datos.
    - Elegir el motor de base de datos deseado (MySQL o PostgreSQL).
    - Especificar la región donde se desplegará la instancia para asegurar baja latencia con el Servidor de Aplicación (Backend).
    - Definir la configuración de la máquina (tipo de CPU, memoria) y el tipo de almacenamiento (ej., SSD) y su tamaño inicial.
2. Configuración de Red y Conectividad:
    - Configurar la instancia para que opere dentro de la Virtual Private Cloud (VPC) del proyecto. Esto asegura que la base de datos esté en una red aislada y segura, fundamental para la seguridad de los datos.
    - Establecer la conectividad de IP privada para el backend, lo que mejora la seguridad y el rendimiento al mantener el tráfico de la base de datos dentro de la red privada de Google.
    - Configurar las redes autorizadas y reglas de firewall para permitir solo el tráfico desde el Servidor de Aplicación (Backend) y otras IPs de administración necesarias.
3. Creación de Usuarios Iniciales:
    - Crear usuarios de base de datos con contraseñas seguras y el "principio de privilegio mínimo". Es decir, el usuario que utilizará el backend para conectarse a la base de datos solo tendrá los permisos necesarios para las operaciones que realice (lectura, escritura, modificación de tablas específicas), sin privilegios administrativos excesivos.
    - Para la gestión de contraseñas de usuarios en la aplicación, se utilizarán funciones de hash seguras para almacenar hash_password en la tabla USUARIOS.

7. Consideraciones Adicionales

Además de las secciones anteriores, existen otras consideraciones importantes para el diseño y la implementación de la base de datos:

1. Proyección y Gestión del Almacenamiento:
    - El servicio de infraestructura en la nube de Google Cloud incluye una capacidad de hasta 1 Terabyte (1000 GB) mensual como cortesía durante los primeros seis meses.
    - Si bien las fuentes no mencionan explícitamente una proyección de 2TB, cualquier aumento en la capacidad de almacenamiento más allá del 1 Terabyte incluido tendrá un costo adicional.
    - Es fundamental aclarar que este almacenamiento (tanto el inicial de 1TB como cualquier aumento futuro) incluye tanto el espacio utilizado por la Base de Datos Relacional (Cloud SQL) para los datos estructurados como el espacio para los archivos

digitales grandes almacenados en Cloud Storage. Se deberá monitorear de cerca el consumo de almacenamiento para gestionar los costos de manera eficiente.

2. Distinción Clara entre Datos Estructurados y Archivos Grandes:
    - Se reafirma que la Base de Datos Relacional (Cloud SQL) se utilizará exclusivamente para almacenar información estructurada y crítica como perfiles de pacientes, historiales clínicos, citas, usuarios y roles.
    - El Almacenamiento de Archivos (Cloud Storage - S3/Blob) se designará para el almacenamiento centralizado y seguro de archivos digitales pesados, como PDFs, imágenes (ej., RX) y videos cortos asociados a los expedientes de los pacientes. Esta separación optimiza el rendimiento y el costo de almacenamiento, y la base de datos Cloud SQL almacenará únicamente la url_almacenamiento que apunta a la ubicación del archivo en Cloud Storage.

3. Seguridad de los Datos Sensibles (Cifrado):
    - Una consideración primordial es que todos los datos sensibles estarán cifrados en reposo, tanto en la Base de Datos Relacional (Cloud SQL) como en el Almacenamiento de Archivos (Cloud Storage). Esto es una medida de seguridad crítica para proteger la información del paciente contra accesos no autorizados.
    - Las contraseñas de usuario se almacenarán utilizando funciones de hash seguras (campo hash_password) en lugar de texto plano.

4. Escalabilidad y Alta Disponibilidad:
    - La elección de GCP y sus servicios (Cloud SQL, Cloud Storage) proporciona inherentemente escalabilidad (la capacidad de expandirse para manejar un mayor volumen de pacientes y datos sin interrupciones) y alta disponibilidad (minimizando el tiempo de inactividad del sistema). Estas características son fundamentales para la continuidad operativa del consultorio.

5. Requisito de Conectividad:
    - Para un rendimiento óptimo del ecosistema, se requiere una conexión a internet estable de al menos 250 Mbps. La dependencia de la infraestructura en la nube hace que la calidad de la conexión sea crucial.

6. Vinculación del Costo de la Nube al Plan de Mantenimiento:
    - La cortesía de Google Cloud por los primeros seis meses está directamente vinculada a la vigencia del Plan de Mantenimiento y Soporte mensual. Si el cliente decide no continuar con el plan de mantenimiento durante este periodo, la cortesía del servicio en la nube finalizará en ese mismo momento. Esta es una condición importante que la alta dirección debe considerar.

Con estas secciones, el "Documento de Diseño de la Base de Datos (Cloud SQL)" está completo.