

Bitácoras de Trabajo del Proyecto HospiTec

Ignacio

15 de mayo: Hoy empecé a investigar cómo crear y ejecutar scripts en PostgreSQL. Instalé PostgreSQL y pgAdmin4 en mi máquina de desarrollo. Me aseguré de que ambos programas estuvieran correctamente configurados para comenzar a trabajar en los scripts de la base de datos.

16 de mayo: Continué la investigación sobre las mejores prácticas para la creación de scripts en PostgreSQL. Leí documentación y algunos tutoriales para entender cómo estructurar correctamente los scripts de creación y población de datos.

17 de mayo: Desarrollé los scripts de creación de tablas en PostgreSQL. Definí la estructura de las tablas necesarias para el sistema: camas, equipo, historial, horarios de cama, paciente, paciente teléfono, patologías, personal, personal teléfonos, procedimientos, reserva, rol, salón y tratamiento.

18 de mayo: Comencé a escribir los scripts para la población de datos iniciales en PostgreSQL. Me aseguré de que los datos estuvieran correctamente formateados y que las relaciones entre las tablas se mantuvieran intactas.

19 de mayo: Finalicé los scripts de población de datos y realicé varias pruebas para asegurarme de que los datos se insertaran correctamente en la base de datos. También verifiqué la integridad de los datos y las relaciones entre las tablas.

20 de mayo: Empecé a trabajar en los stored procedures y triggers necesarios para la base de datos. Implementé un trigger para incrementar la capacidad de camas cuando se inserta una nueva cama. También comencé a desarrollar un stored procedure para verificar que la fecha de salida no sea anterior a la fecha de entrada.

21 de mayo: Continué desarrollando los stored procedures y triggers. Implementé un stored procedure para asegurar que no se puedan registrar dos números telefónicos iguales en la tabla de teléfonos. Realicé pruebas para asegurarme de que los stored procedures y triggers funcionaran correctamente.

22 de mayo: Finalicé la implementación de todos los stored procedures y triggers necesarios para la base de datos. Realicé pruebas exhaustivas para asegurarme de que todos funcionaran correctamente y que la integridad de los datos se mantuviera.

23 de mayo: Empecé a desarrollar la API en ASP.NET Core 5. Definí la estructura básica de la API y los endpoints necesarios para interactuar con la base de datos. Implementé los métodos CRUD para la tabla de camas.

24 de mayo: Continué desarrollando la API, añadiendo endpoints para gestionar equipos médicos y personal. Implementé los métodos CRUD para estas tablas y realicé pruebas para asegurarme de que los datos se gestionaran correctamente.

25 de mayo: Añadí funcionalidad CRUD a la API para las tablas de historial y horarios de cama. Aseguré que todos los endpoints funcionaran correctamente y que los datos se insertaran, actualizaran y eliminaran según fuera necesario.

26 de mayo: Trabajé en los endpoints para gestionar pacientes y teléfonos de pacientes. Implementé los métodos CRUD necesarios y realicé pruebas exhaustivas para asegurarme de que todo funcionara correctamente.

27 de mayo: Continué con la implementación de la API, añadiendo funcionalidad para las tablas de patologías, personal teléfonos y procedimientos. Realicé pruebas para verificar que todos los endpoints funcionaran correctamente.

28 de mayo: Finalicé la implementación de la API, añadiendo los endpoints para las tablas de reserva, rol, salón y tratamiento. Verifiqué que todos los métodos CRUD funcionaran correctamente y que la API estuviera completamente operativa.

29 de mayo: Realicé pruebas integrales de la API para asegurarme de que todos los endpoints funcionaran correctamente y que no hubiera errores en las solicitudes y respuestas. También hice ajustes y correcciones según fuera necesario.

1 de junio: Realicé una revisión final de la API, asegurándome de que todos los endpoints estuvieran documentados y que la API estuviera lista para ser utilizada por el frontend. Comencé a preparar la API para el despliegue en Azure.

2 de junio: Configuré el entorno de despliegue en Azure y subí la API. Realicé pruebas para asegurarme de que la API funcionara correctamente en el entorno de producción y que todas las conexiones con la base de datos estuvieran establecidas.

5 de junio: Hice pruebas finales y ajustes necesarios para garantizar que la API estuviera completamente operativa y lista para ser utilizada por el sistema HospiTEC. Verifiqué que todas las interacciones con la base de datos fueran fluidas y sin errores.

Andrés Molina

15 de mayo: Hoy comencé a desarrollar la vista de Pacientes en Angular. Definí la estructura básica de la vista y las rutas necesarias para navegar entre las diferentes secciones.

16 de mayo: Implementé la funcionalidad básica de la vista de Pacientes, incluyendo la interfaz de usuario y las primeras conexiones con la API para obtener datos iniciales.

17 de mayo: Añadí la funcionalidad para que los pacientes puedan crear reservaciones para procedimientos médicos. Aseguré que los datos se enviaran correctamente a la API y que las reservaciones se guardaran en la base de datos.

18 de mayo: Trabajé en la interfaz de usuario para que los pacientes puedan ver su historial clínico. Conecté esta funcionalidad con la API para obtener los datos del historial desde la base de datos.

19 de mayo: Añadí la funcionalidad para que los pacientes puedan calificar su estadía en el hospital y al personal que los atendió. Implementé una interfaz de usuario intuitiva para este propósito.

20 de mayo: Comencé a trabajar en la vista de Personal Administrativo. Definí las funcionalidades básicas y las rutas necesarias para navegar entre las diferentes secciones de esta vista.

21 de mayo: Implementé la funcionalidad para que los administradores del hospital puedan ingresar información general del sistema, como salones, camas, equipo médico y procedimientos.

22 de mayo: Conecté la vista de Personal Administrativo con la API para gestionar los datos en la base de datos. Adapté los métodos de la API al contexto específico de esta vista.

23 de mayo: Aseguré que la sincronización de envío y recepción de datos entre el frontend y el backend fuera fluida y coordinada. Realicé pruebas para verificar que los datos se gestionaran correctamente.

25 de mayo: Realicé pruebas de usuario para asegurarme de que todas las funcionalidades implementadas funcionaran correctamente. Hice ajustes y correcciones según fuera necesario.

28 de mayo: Continué realizando pruebas y ajustes en la vista de Pacientes y Personal Administrativo. Verifiqué que todas las conexiones con la API fueran correctas y que los datos se gestionaran sin errores.

29 de mayo: Finalicé la vista de Pacientes, asegurándome de que todas las funcionalidades estuvieran correctamente implementadas y que los datos se gestionaran correctamente.

1 de junio: Realicé una revisión final de la vista de Personal Administrativo, asegurándome de que todas las funcionalidades estuvieran implementadas y que la conexión con la API fuera correcta.

2 de junio: Hice pruebas finales de la vista de Personal Administrativo, verificando que todas las funcionalidades estuvieran operativas y que los datos se gestionaran correctamente.

5 de junio: Finalicé la implementación de la vista de Personal Administrativo, asegurándome de que todo funcionara correctamente y sin errores. Realicé pruebas finales y ajustes según fuera necesario.

Luis González

15 de mayo: Hoy comencé a desarrollar las pantallas de la vista Administrador en Angular, comenzando con la gestión de salones. Definí la estructura básica y las rutas necesarias.

16 de mayo: Implementé la funcionalidad para crear, modificar y eliminar salones. Aseguré que se pudiera registrar el número de salón, nombre, capacidad en camas, y otros detalles necesarios.

17 de mayo: Continué trabajando en la gestión de salones, añadiendo validaciones y asegurándome de que todos los datos se guardaran correctamente en la base de datos.

18 de mayo: Comencé a trabajar en la gestión de equipo médico. Implementé la funcionalidad para crear y modificar equipo médico, incluyendo detalles como nombre, proveedor y cantidad disponible.

19 de mayo: Desarrollé la interfaz de usuario para la gestión de equipo médico y conecté esta funcionalidad con la API para gestionar los datos en la base de datos.

20 de mayo: Añadí más detalles a la funcionalidad de gestión de equipo médico, asegurándome de que todos los datos se guardaran y actualizaran correctamente.

21 de mayo: Comencé a trabajar en la gestión de camas, permitiendo crear y modificar camas con detalles como número de cama, equipo médico asociado y salón.

22 de mayo: Implementé la interfaz de usuario para la gestión de camas y conecté esta funcionalidad con la API para gestionar los datos en la base de datos.

23 de mayo: Realicé pruebas exhaustivas para asegurarme de que todas las funcionalidades de creación, modificación y eliminación en las vistas de gestión de salones, equipo médico y camas funcionaran correctamente.

25 de mayo: Investigé sobre MongoDB y cómo se utiliza para los registros. Definí junto con Mariana cómo debería funcionar el almacenamiento de información en MongoDB.

28 de mayo: Implementé la base de datos en MongoDB y desarrollé la API para obtener y devolver información. Aseguré la integración correcta con el sistema.

29 de mayo: Realicé pruebas para verificar que la base de datos de MongoDB y la API funcionaran correctamente. Hice ajustes y correcciones según fuera necesario.

1 de junio: Configuré la base de datos de MongoDB en Azure y aseguré que estuviera lista para su uso en producción. Verifiqué la integración con el resto del sistema.

2 de junio: Continué realizando pruebas de la API y la base de datos de MongoDB. Aseguré que todos los datos se gestionaran correctamente y que la comunicación con la base de datos fuera fluida.

5 de junio: Realicé pruebas finales de las funcionalidades y la integración con MongoDB. Hice ajustes necesarios para garantizar un rendimiento óptimo y un funcionamiento fluido del sistema.

Andrés Guzmán

15 de mayo: Hoy comencé a desarrollar la vista de Doctor en Angular. Definí las funcionalidades básicas y la estructura de las rutas necesarias para esta vista.

16 de mayo: Implementé la interfaz de usuario básica para la vista de Doctor, permitiendo a los médicos navegar entre diferentes secciones y realizar acciones básicas.

17 de mayo: Añadí la funcionalidad para que los médicos puedan agregar pacientes al sistema. Implementé la interfaz de usuario y conecté esta funcionalidad con la API.

18 de mayo: Desarrollé la funcionalidad para que los médicos puedan registrar información detallada de los pacientes, como nombre, apellidos, cédula, teléfono, dirección, fecha de nacimiento y patologías.

19 de mayo: Implementé la funcionalidad para agregar y modificar el historial clínico de los pacientes. Aseguré que los datos se guardaran correctamente en la base de datos a través de la API.

20 de mayo: Desarrollé la interfaz de usuario para la gestión del historial clínico y conecté esta funcionalidad con la API. Realicé pruebas para asegurarme de que todo funcionara correctamente.

21 de mayo: Aseguré que todos los métodos CRUD necesarios para la vista de Doctor estuvieran implementados y sincronizados con la API. Realicé pruebas exhaustivas para verificar su correcto funcionamiento.

22 de mayo: Continué trabajando en la vista de Doctor, añadiendo más funcionalidades y mejorando la interfaz de usuario. Aseguré que todas las interacciones con la API fueran fluidas.

23 de mayo: Investigé cómo subir y unificar las APIs desarrolladas por Ignacio y Luis. Probé varias soluciones hasta conseguir una integración correcta.

25 de mayo: Logré unificar las APIs y comencé el proceso de despliegue en Azure. Realicé pruebas para asegurarme de que las APIs funcionaran correctamente en el entorno de producción.

28 de mayo: Completé el despliegue de las APIs y la web en Azure. Verifiqué que la integración fuera correcta y funcional.

29 de mayo: Realicé pruebas de despliegue final y ajustes necesarios para asegurar que todo el sistema funcionara correctamente en Azure.

1 de junio: Hice pruebas de usuario y ajustes finales para garantizar que la vista de Doctor funcionara perfectamente y sin errores.

2 de junio: Continué realizando pruebas y ajustes en la vista de Doctor, asegurando que todas las funcionalidades estuvieran correctamente implementadas.

5 de junio: Finalicé la implementación de la vista de Doctor, asegurándome de que todo funcionara correctamente y sin errores. Realicé pruebas finales y ajustes según fuera necesario.

Mariana

15 de mayo: Hoy comencé a trabajar en la vista Administrador en Angular, enfocándome en la gestión de procedimientos médicos. Definí las funcionalidades básicas y la estructura de las rutas necesarias.

16 de mayo: Implementé la interfaz de usuario básica para la gestión de procedimientos médicos. Añadí la funcionalidad para crear y modificar procedimientos, asegurando que se pudiera registrar el nombre y los días mínimos de recuperación en salón.

17 de mayo: Trabajé en la interfaz de usuario para la gestión de procedimientos médicos, asegurando que fuera intuitiva y fácil de usar. Conecté esta funcionalidad con la API para gestionar los datos en la base de datos.

18 de mayo: Comencé a trabajar en la gestión de personal, permitiendo crear, modificar y eliminar personal administrativo y médico. Definí las funcionalidades básicas y las rutas necesarias.

19 de mayo: Implementé la interfaz de usuario para la gestión de personal. Aseguré que se pudiera registrar toda la información necesaria sobre el personal, incluyendo nombre, apellidos, cédula, teléfono, dirección, fecha de nacimiento y fecha de ingreso a la institución.

20 de mayo: Añadí más detalles a la funcionalidad de gestión de personal, permitiendo diferenciar entre personal administrativo, doctores y enfermeros. Realicé pruebas para asegurarme de que todo funcionara correctamente.

21 de mayo: Comencé a trabajar en la funcionalidad para la carga inicial de pacientes desde un archivo XLSX (Excel). Definí el procedimiento para importar datos a una tabla temporal y luego a las tablas finales del sistema.

22 de mayo: Implementé la funcionalidad de carga de pacientes y realicé pruebas para asegurarme de que los datos se importaran correctamente desde el archivo Excel. Aseguré que el procedimiento almacenado realizara todas las validaciones necesarias.

23 de mayo: Trabajé en el reporte de áreas de mejora, tomando las calificaciones de los pacientes y generando informes de los aspectos evaluados. Definí la estructura básica del reporte y la interfaz de usuario.

25 de mayo: Implementé la lógica inicial para la base de datos de MongoDB, asegurando que estuviera correctamente configurada para el almacenamiento de datos. Trabajé en la integración de MongoDB con el sistema.

28 de mayo: Desarrollé todas las funciones CRUD para las secciones de gestión de procedimientos, personal y carga de pacientes. Realicé pruebas para asegurarme de que todo funcionara correctamente.

29 de mayo: Configuré la eliminación de datos en la vista Administrador, asegurando la integridad de los datos. Realicé pruebas exhaustivas para verificar que todo funcionara correctamente.

1 de junio: Continué realizando pruebas y ajustes en la vista Administrador, asegurando que todas las funcionalidades estuvieran correctamente implementadas y sincronizadas.

2 de junio: Realicé pruebas finales de todas las funcionalidades en la vista Administrador, verificando que todo funcionara correctamente y sin errores.

5 de junio: Finalicé la implementación de la vista Administrador, asegurándome de que todo funcionara correctamente y sin errores. Realicé pruebas finales y ajustes según fuera necesario.