

HISTORIAS DE USUARIO

Nro: HU-001	Título: Registro de instructores	Prioridad: Alta Estimación: 3 horas
Historia de usuario: Como instructor de yoga terapéutico quiero registrarme en el sistema, para acceder a las funcionalidades del sistema.		
Sabré que he terminado cuando se cumplan todos los criterios de aceptación: <ol style="list-style-type: none">1. Registro exitoso: Dado que soy un nuevo instructor, cuando completo el formulario con nombre, correo y contraseña válidos, entonces el sistema debe crear mi cuenta y confirmar el registro.2. Correo ya registrado: Dado que intento registrarme, cuando uso un correo que ya está guardado en el sistema, entonces el sistema debe impedir el registro y mostrar el mensaje: "El instructor ya está registrado".3. Datos faltantes: Dado que intento registrarme, cuando omito algún campo obligatorio, entonces el sistema debe señalar el campo que deje vacío.		
Tareas de implementación: <ol style="list-style-type: none">1. Definir modelo "Instructor". (0.5 horas)2. Crear servicio para registrar instructores: validar correo único, hash de contraseña. (1 hora)3. Crear dentro del controlador "registrarInstructorHandler" con manejo de errores. (0.5 horas)4. Crear endpoint "POST/api/instructores" en el router. (0.5 horas)5. Probar casos normales y de errores en Postman. (0.5horas)		

Nro: HU-002	Título: Inicio de sesión de instructores	Prioridad: Alta Estimación: 2 horas
Historia de usuario: Como instructor registrado, quiero iniciar sesión con mi correo y contraseña, para acceder de forma segura al sistema.		
Sabré que he terminado cuando se cumplan todos los criterios de aceptación: <ol style="list-style-type: none"> 1. Login exitoso: Dado que tengo una cuenta registrada, cuando ingreso mi correo y contraseña correcta, entonces el sistema debe autenticarme y mostrar la ventana principal. 2. Contraseña incorrecta: Dado que ingreso mi correo, cuando escribo la contraseña incorrectamente, entonces el sistema debe mostrar el mensaje: “Contraseña incorrecta”. 3. Correo no registrado: Dado que intento iniciar sesión, cuando ingreso un correo que no existe, entonces el sistema debe mostrar el mensaje: “Instructor no registrado”. 		
Tareas de implementación: <ol style="list-style-type: none"> 1. Crear un servicio para autenticar: buscar instructor por correo y verificar contraseña. (0.5 horas) 2. Crear función para generar JWT con el “id” del instructor. (0.5 horas) 3. Crear dentro del controlador “loginInstructorHandler”. (0.5 horas) 4. Crear endpoint “POST/api/auth/login” en el router. (0.5 horas) 		

Nro: HU-003	Título: Registrar pacientes	Prioridad: Alta Estimación: 4 horas
<p>Historia de usuario:</p> <p>Como instructor de yoga terapéutico, quiero registrar pacientes, para asignarles series terapéuticas y hacer seguimiento de su progreso.</p>		
<p>Sabré que he terminado cuando se cumplan todos los criterios de aceptación:</p> <ol style="list-style-type: none"> 1. Registro exitoso: Dado que el instructor está autenticado en el sistema, cuando complete el formulario con los datos del nuevo paciente y confirme, entonces el sistema debe registrar al paciente y mostrar sus datos básicos. 2. Correo ya registrado: Dado que el instructor intenta registrar un paciente, cuando el correo electrónico ingresado ya existe en el sistema, entonces debe mostrarse un mensaje de error: "Paciente ya registrado". 3. Datos faltantes: Dado que el instructor intenta registrar un paciente, cuando omite un dato obligatorio (nombre, correo o fecha de nacimiento), entonces el sistema debe señalar el campo que dejó vacío. 		
<p>Tareas de implementación:</p> <ol style="list-style-type: none"> 1. Definir modelo "Paciente" en models (0.5 horas) 2. Implementar lógica en el servicio para crear pacientes con validación de correo único. (1 hora) 3. Crear un endpoint "POST/api/pacientes" en el router y enlazarlo al controlador. (0.5 horas) 4. Crear dentro del controlador "crarPacienteHandler" que use el servicio y maneje errores. (0.5 horas) 5. Crear middleware de autenticación JWT y aplicarlo al endpoint. (0.5 horas) 6. Probar con Postman datos válidos y errores comunes (correo duplicado, faltantes) (1 hora) 		

Nro: HU-004	Título: Inicio de sesión de pacientes	Prioridad: Alta Estimación: 3 horas
<p>Historia de usuario:</p> <p>Como paciente registrado, quiero iniciar sesión con mi correo y contraseña, para acceder a mi serie terapéutica y registrar mi progreso.</p>		
<p>Sabré que he terminado cuando se cumplan todos los criterios de aceptación:</p> <ol style="list-style-type: none"> 4. Login Exitoso de Paciente: Dado que soy un paciente con una cuenta activa con una contraseña establecida, cuando ingreso mi correo y contraseña correcta en el formulario de login y presiono "Entrar como Paciente", entonces el sistema debe validar mis credenciales y mostrarme la pantalla de inicio para comenzar mi sesión terapéutica. 5. Credenciales Incorrectas: Dado que soy un paciente registrado, cuando ingreso una contraseña incorrecta, entonces el sistema debe rechazar el acceso y mostrar un mensaje de error sobre credenciales inválidas. 6. Cuenta No Activada: Dado que mi cuenta de paciente fue creada por un instructor, pero aún no he establecido mi contraseña, cuando intento iniciar sesión, entonces el sistema debe rechazar el acceso y mostrar un mensaje indicando que la cuenta no está activa o las credenciales son incorrectas. 		
<p>Tareas de implementación:</p> <ol style="list-style-type: none"> 7. Crear servicio loginPaciente() en authService que valide credenciales contra la base de datos de pacientes, verificando que la cuenta esté activa. (1 hora) 8. Refactorizar el controlador authController para que el manejador de login acepte un rol y llame al servicio correspondiente (loginInstructor o loginPaciente). (0.5 horas) 9. Modificar la lógica del frontend en inicioSesion.ts para enviar el rol seleccionado en el cuerpo de la petición a la API. (0.5 horas) 10. Probar el flujo de login completo para ambos roles (instructor y paciente) y los casos de error. (1 hora) 		

Nro: HU-005	Título: Actualizar información de pacientes	Prioridad: Baja Estimación: 3.5 horas
Historia de usuario: Como instructor de yoga terapéutico, quiero actualizar la información de un paciente, para mantener sus datos correctamente.		
Sabré que he terminado cuando se cumplan todos los criterios de aceptación: <ol style="list-style-type: none"> 1. Actualización exitosa de datos: Dado que seleccione un paciente registrado, cuando modifique sus datos y los guarde, entonces el sistema debería actualizar la información correctamente y notificarme del éxito. 2. Error por datos inválidos: Dado que he ingresado información inválida como un correo sin @, cuando intente guardar los cambios, entonces el sistema debería mostrar un mensaje de error y no guardar los cambios. 		
Tareas de implementación: <ol style="list-style-type: none"> 1. Diseñar modelo de actualización (por ID) y estructura del request. (0.5 horas) 2. Codificar función “actualizarPaciente()” en servicio. (0.5 horas) 3. Crear dentro del controlador “actualizarPacienteHandler” (0.5 horas) 4. Añadir validaciones (correo válido, campos requeridos). (1hora) 5. Crear ruta PUT /api/pacientes/:id protegida con JWT. (0.5 horas) 6. Probar con casos válidos e inválidos. (0.5 horas) 		

Nro: HU-006	Título: Crear serie terapéutica	Prioridad: Alta Estimación: 4.5 horas
Historia de usuario: Como instructor de yoga terapéutico, quiero diseñar series de posturas para mis pacientes, para ofrecer tratamientos personalizados.		
Sabré que he terminado cuando se cumplan todos los criterios de aceptación: <ol style="list-style-type: none"> Guardado exitoso de la serie: Dado que estoy creando una serie, cuando completo el nombre, tipo de terapia y selecciono al menos una postura, entonces el sistema debe guardar la nueva serie y vincularla a mi perfil de instructor. Error por datos incompletos: Dado que he dejado campos vacíos, cuando intente guardar la serie, entonces el sistema debería mostrar un mensaje de advertencia y evitar el guardado. Cancelación: Dado que he comenzado a crear una serie, cuando seleccione la opción de cancelar, entonces el sistema debería descartar los datos ingresados y regresar a la pantalla anterior. 		
Tareas de implementación: <ol style="list-style-type: none"> Crear modelo "Serie". (0.5 horas) Crear servicio "crearSerie()". (1 hora) Crear dentro del controlador "crearSerieHandler()" (0.5 horas) Validaciones (campos obligatorios, formato, etc.). (1 hora) Crear ruta "POST /api/series". (0.5 horas) Probar flujos correctos y fallos. (1 hora) 		

Nro: HU-007	Título: Asignar serie a paciente	Prioridad: Alta Estimación: 6 horas
<p>Historia de usuario:</p> <p>Como instructor de yoga terapéutico, quiero asignar una serie a un paciente, para que pueda iniciar su tratamiento.</p>		
<p>Sabré que he terminado cuando se cumplan todos los criterios de aceptación:</p> <ol style="list-style-type: none"> 1. Asignación correcta: Dado que he creado una serie terapéutica, cuando seleccione un paciente y confirme la asignación, entonces el sistema debería vincular correctamente la serie al paciente. 2. Reemplazo de serie: Dado que el paciente ya ha tenido una serie asignada previamente, cuando asigne una nueva serie, entonces el sistema debería reemplazar la anterior serie. 3. Sin series disponibles: Dado que no he creado ninguna serie, cuando intente asignar una a un paciente, entonces el sistema debería mostrar un mensaje indicando que no existen series disponibles. 		
<p>Tareas de implementación:</p> <ol style="list-style-type: none"> 1. Crear modelo para relacionar paciente y serie. (0.5 horas) 2. Crear endpoint para asignar serie al paciente. (1 hora) 3. Validar la existencia de paciente y serie. (1 hora) 4. Implementar lógica para reemplazar serie anterior. (1 hora) 5. Devolver confirmación y feedback adecuado. (0.5 horas) 6. Pruebas funcionales de la asignación y reemplazo. (1 hora) 7. Mensaje si no hay series creadas. (1 hora) 		

Nro: HU-008	Título: Consultar estado de los pacientes	Prioridad: Alta Estimación: 6.5 horas
<p>Historia de usuario:</p> <p>Como instructor de yoga, quiero consultar el estado de cada uno de mis pacientes después de sus sesiones, para tener un seguimiento de su progreso terapéutico.</p>		
<p>Sabré que he terminado cuando se cumplan todos los criterios de aceptación:</p> <ol style="list-style-type: none"> 1. Visualización del estado de una sesión: Dado que un paciente ha completado una sesión, cuando consulte su estado, entonces el sistema debería mostrar el nivel de dolor inicial y final, junto con su comentario. 2. Historial de sesiones: Dado que el paciente ha realizado varias sesiones, cuando acceda a su historial, entonces el sistema debería mostrar una lista con fechas, duración y datos registrados por sesión. 3. Sin sesiones registradas: Dado que el paciente no ha iniciado su serie, cuando intente consultar su estado, entonces el sistema debería informar que no existen sesiones registradas. 		
<p>Tareas de implementación:</p> <ol style="list-style-type: none"> 1. Crear modelo de sesiones con campos: dolor, comentario, fechas. (1 hora) 2. Crear servicio para listar sesiones por paciente. (1 hora) 3. Crear un endpoint para consultar historial de un paciente. (1 hora) 4. Validar existencia del paciente y datos asociados. (0.5 horas) 5. Devolver mensaje claro si no hay sesiones registradas. (0.5 horas) 6. Crear vista de detalle con datos de cada sesión. (1 hora) 7. Pruebas de visualización de historial y estado individual. (1.5 horas) 		

Nro: HU-009	Título: Ejecutar serie terapéutica	Prioridad: Alta Estimación: 7 horas
<p>Historia de usuario:</p> <p>Como paciente de yoga quiero realizar la serie que me asignaron, para seguir con mi proceso terapéutico.</p>		
<p>Sabré que he terminado cuando se cumplan todos los criterios de aceptación:</p> <ol style="list-style-type: none"> 1. Registro del dolor inicial: Dado que he recibido una serie asignada, cuando inicie una sesión, entonces el sistema debería solicitar que indique la intensidad del dolor o molestia inicial. 2. Avance automático entre posturas: Dado que he comenzado la sesión, cuando finalice el tiempo de una postura, entonces el sistema debería mostrar automáticamente la siguiente en el orden establecido. 3. Registro final: Dado que he terminado todas las posturas, cuando finalice la sesión, entonces el sistema debería solicitar que indique nuevamente la intensidad del dolor y escriba un comentario obligatorio. 4. Consulta de apoyo visual: Dado que he necesitado orientación sobre una postura, cuando seleccione la opción de información adicional, entonces el sistema debería mostrar imágenes, video, instrucciones, beneficios y precauciones. 		
<p>Tareas de implementación:</p> <ol style="list-style-type: none"> 1. Crear flujo de ejecución de la serie con temporizador por postura. (1.5 horas) 2. Solicitar dolor inicial antes de iniciar la sesión. (0.5 horas) 3. Implementar transición automática entre posturas con temporizador. (1 hora) 4. Solicitar dolor final y comentario obligatorio al terminar. (0.5 horas) 5. Integrar consulta de información adicional (fotos, video, texto). (1.5 horas) 6. Validar que el paciente tenga una serie asignada activa. (0.5 horas) 7. Registrar todos los datos en el historial del paciente. (1.5 horas) 		