Taller Práctico: Implementación Ágil en el Desarrollo de la Web del Hospital

Mauricio Barrios B. - Fecha: 10-03-2025

El motivo del presente informe es aplicar Metodologías Ágiles y Scrum en el desarrollo del sitio web del hospital.

**1.Visión del Producto**

Objetivo: Desarrollar una plataforma web para gestión hospitalaria que permita:

* Registro y perfil de usuarios (médicos, pacientes, administradores).
* Mantenedores CRUD para doctores, pacientes, usuarios y citas.
* Sistema de citas médicas con selección de horarios por pacientes.
* Control de acceso diferenciado por roles.

**2.Roles SCRUM**

Product Owner: Representante del hospital (prioriza el backlog).

Scrum Master: Líder técnico/experto en metodologías ágiles.

Equipo Dev:

* 2 Backend developers
* 2 Frontend developers
* 1 QA

**3.Priorización y Creación de Historias de Usuario**

1. HU1: Sistema de Autenticación de Usuarios

Descripción: Como usuario, quiero registrarme e iniciar sesión con correo y contraseña para acceder al sistema según mi rol (administrador, médico, paciente).

Criterios de Aceptación (AC):

* Formulario de registro con email, contraseña y selección de rol.
* Validación de formato de email y contraseña (mínimo 8 caracteres, alfanumérica).
* Almacenamiento seguro en mediante API REST(PHP + MySQL).
* Mensajes de error claros (ej: "contraseña incorrecta").
* Redirección a home según rol.

INVEST:

* Independiente: Sí (no depende de otras HU).
* Negociable: Sí (detalles como roles adicionales pueden ajustarse).
* Valiosa: Esencial para la seguridad y acceso al sistema.
* Estimable: 3-5 días (frontend + backend).
* Pequeña: Sí (alcance definido).
* Testeable: Sí (pruebas de registro, login y roles).

SMART:

* Específico: Implementar autenticación con roles.
* Medible: 100% de funcionalidad operativa.
* Alcanzable: Con React y MySQL.
* Relevante: Requisito base para otras funcionalidades.
* Tiempo: 1 sprint.

Prioridad: Alta (1/5).

2. HU2: Mantenedor de Doctores (CRUD)

Descripción: Como administrador, quiero gestionar los datos de los médicos (crear, leer, actualizar, eliminar) para mantener la información actualizada.

Criterios de Aceptación (AC):

* Formulario para agregar médicos (nombre, especialidad, horario, contacto).
* Tabla con lista de médicos registrados (ordenable y filtrable).
* Opciones de editar y eliminar.
* Validación de campos obligatorios (ej: especialidad no vacía).

INVEST:

* Independiente: Sí (depende solo de HU1 para el rol de administrador).
* Negociable: Sí (campos adicionales como foto pueden agregarse luego).
* Valiosa: Central para gestionar recursos médicos.
* Estimable: 4-6 días (incluye validaciones y UI).
* Pequeña: Sí (CRUD básico).
* Testeable: Sí (pruebas de creación y edición).

SMART:

* Específico: CRUD completo para médicos.
* Medible: 4 operaciones funcionales.
* Alcanzable: Con React y MySQL.
* Relevante: Requisito para agendar citas.
* Tiempo: 1 sprint.

Prioridad: Alta (2/5).

3. HU3: Mantenedor de Pacientes (CRUD)

Descripción: Como administrador, quiero gestionar los datos de los pacientes (crear, leer, actualizar, eliminar) para mantener historiales médicos.

Criterios de Aceptación (AC):

* Formulario para agregar pacientes (nombre, rut, historial médico, contacto).
* Tabla con lista de pacientes y búsqueda por nombre/RUT.
* Validación de RUT único y formato correcto.

INVEST:

* Independiente: Sí (similar a HU2).
* Valiosa: Esencial para seguimiento de pacientes.
* Estimable: 4-6 días.
* Testeable: Sí (pruebas de RUT y campos).

SMART:

* Específico: CRUD de pacientes con validaciones.
* Relevante: Base para citas médicas.

Prioridad: Alta (3/5).

4. HU4: Agendar Cita Médica

Descripción: Como paciente, quiero agendar una cita con un médico disponible para recibir atención.

Criterios de Aceptación (AC):

* Seleccionar médico, fecha/hora y motivo desde un calendario interactivo.
* Validar disponibilidad del médico.
* Mostrar confirmación con detalles de la cita.

INVEST:

* Independiente: No (depende de HU2 y HU3).
* Valiosa: Funcionalidad central del sistema.
* Estimable: 3-5 días (utilizar librería ReactJS para tablas y filtros).

SMART:

* Específico: Agendamiento con validación de disponibilidad.
* Medible: Citas creadas en base de datos.
* Testeable: Sí (pruebas con usuarios simulados)

Prioridad: Media (4/5).

5. HU5: Visualización de Citas por Rol

Descripción: Como médico o paciente, quiero ver mis citas asignadas en un calendario para gestionar mi tiempo.

Criterios de Aceptación (AC):

* Listado con citas filtradas por rol (médico ve sus citas; paciente ve las propias).
* Opción de filtrar por fecha o estado (pendiente, completada).

INVEST:

* Valiosa: Mejora la experiencia de usuario.
* Estimable: 2-3 días (uso de librerías de calendario, tablas y filtros ReactJS).

SMART:

* Alcanzable: Con ReactJS y APIs de conexión a BBDD.

Prioridad: Media (5/5).

Priorización Product Backlog:

|  |  |  |
| --- | --- | --- |
| **ID** | **Título** | **Prioridad** |
| HU1 | Autenticación de Usuarios | 1 |
| HU2 | Mantenedor de Doctores | 2 |
| HU3 | Mantenedor de Pacientes | 3 |
| HU4 | Agendar Cita Médica | 4 |
| HU5 | Visualización de Citas | 5 |

Priorización MoSCoW:

|  |  |  |
| --- | --- | --- |
| **ID** | **Título** | **Prioridad** |
| HU1 | Autenticación de Usuarios | Must Have |
| HU2 | Mantenedor de Doctores | Must Have |
| HU3 | Mantenedor de Pacientes | Must Have |
| HU4 | Agendar Cita Médica | Must Have |
| HU5 | Visualización de Citas | Should Have |

Estimación esfuerzo historias de usuario:

|  |  |  |
| --- | --- | --- |
| **Historia de usuario** | **Puntos de historia** | **Talla** |
| Autenticación de Usuarios | 5 | M |
| Mantenedor de Doctores | 8 | L |
| Mantenedor de Pacientes | 8 | L |
| Agendar Cita Médica | 5 | M |
| Visualización de Citas | 3 | S |

**4.Planificación de Sprint**

Clasificación de Sprint:

|  |  |  |
| --- | --- | --- |
| **Sprint** | **Historia** | **Tiempo** |
| 1 | Autenticación de Usuarios | 2 semanas |
| Mantenedor de Doctores |
| 2 | Mantenedor de Pacientes | 1 semana |
| 3 | Agendar Cita Médica | 1 semana |
| 4 | Visualización de Citas | 1 semana |

Progreso del Sprint:

|  |  |  |
| --- | --- | --- |
| **Sprint** | **Responsable** | **Estado** |
| 1 | Backend developer | Finalizado |
| 2 | Backend developer | En progreso |
| 3 | Frontend developer | En progreso |
| 4 | Frontend developer | Por hacer |