Documento de requisitos – Página web para gestión de reservas universitarias
Ingeniería de Software
Roy Sandoval – ID: 00051613

Requerimientos mínimos

Tipo de Requisito	Código	Descripción
Funcional	RF1	El usuario puede ver la lista de espacios disponibles.
Funcional	RF2	El usuario puede crear una reserva con nombre, fecha, hora y espacio.
Funcional	RF3	El sistema debe evitar reservas duplicadas (misma fecha y hora).
Funcional	RF4	El usuario puede cancelar su reserva.
Funcional	RF5	Se mostrará una imagen o plano del espacio seleccionado.
No Funcional	RNF1	El sistema debe responder en menos de 3 segundos.
No Funcional	RNF2	La información se guarda localmente (LocalStorage o archivo JSON).
No Funcional	RNF3	Los datos deben mostrarse en una interfaz clara y accesible.
No Funcional	RNF4	Se debe aplicar una estructura de carpetas ordenada.
No Funcional	RNF5	Código limpio, comentado y con commits claros.

Historias de usuario

1. Como estudiante, quiero ver una lista de espacios universitarios disponibles para poder elegir cuál reservar.

Independiente: No depende de otras HU.
Valiosa: Permite conocer opciones disponibles.
Estimable: Lista fija de espacios.
Testable: Se puede verificar que los espacios aparecen correctamente.

Criterios de aceptación:
- Debe mostrarse una lista con nombre, tipo y disponibilidad del espacio
- Cada espacio debe incluir una imagen o ícono representativo (RF5).
- El sistema debe cargar los datos en menos de 3 segundos (RNF1).
- La interfaz debe ser clara y legible (RNF3).

2. Como estudiante, quiero registrar una reserva indicando mi nombre, fecha, hora y espacio para asegurar mi horario.

Valiosa: Permite completar el objetivo principal del sistema.
Pequeña: Un solo formulario.
Testable: Se valida la creación del objeto de reserva.

Criterios de aceptación:
- El formulario debe incluir campos de nombre, fecha, hora y espacio (RF2).
- No se deben permitir campos vacíos.
- La reserva debe guardarse en LocalStorage o JSON (RNF2)
- Al confirmar, debe mostrarse un mensaje visual de éxito.

3. Como estudiante, quiero evitar crear reservas duplicadas en el mismo espacio, fecha y hora para evitar conflictos.

Independiente: Requiere HU2 pero lógica separada.
Testable: Puede probarse con datos duplicados.

Criterios de aceptación:

- Si el usuario intenta reservar el mismo espacio, fecha y hora, debe mostrarse un mensaje de error y no guardar la reserva (RF3).
- La validación debe ocurrir antes de guardar los datos.

4. Como estudiante, quiero cancelar una reserva existente para liberar el espacio si ya no lo necesito.

Valiosa: Aporta control al usuario.
Pequeña: Solo requiere eliminar un registro.
Testable: Se puede verificar que desaparece del almacenamiento.

Criterios de aceptacion

- Debe mostrarse una lista de reservas activas con botón “Cancelar” (RF4).
- Al cancelar, se debe eliminar del LocalStorage o JSON (RNF2)
- Debe mostrarse una notificación visual de confirmación.

5. Como estudiante, quiero visualizar una imagen o plano del espacio seleccionado para confirmar que se ajusta a mis necesidades.

Negociable: Tipo de imagen o formato puede variar.
Valiosa: Mejora la experiencia del usuario.
Testable: La interfaz debe mostrar correctamente la imagen.

Criterios de aceptacion:

- Al seleccionar un espacio en la lista, debe mostrarse una imagen representativa (RF5).
- La imagen debe cargarse en menos de 3 segundos (RNF1)
- Debe adaptarse a distintos tamaños de pantalla (RNF3).

Diagrama UML (Casos de uso)