

Pruebas y Casos de Prueba

En esta sección se describen las pruebas realizadas y la planificación de casos de prueba para las principales entidades del sistema. El objetivo es asegurar la calidad, funcionalidad y correcto comportamiento del sistema bajo diferentes escenarios.

1. Entidad User (Usuario)

Pruebas realizadas:

- **Creación de usuario:** Validación de que se puedan crear usuarios con datos válidos y que los campos obligatorios no puedan estar vacíos.
- **Validación de roles:** Verificar que los roles asignados (usuario o administrador) se respeten y que se asignen correctamente al crear usuarios.
- **Autenticación:** Comprobación de que el sistema permita iniciar sesión solo con credenciales válidas.
- **Manejo de errores:** Intento de registro con datos inválidos o duplicados (email ya registrado).

Casos de prueba planeados:

Caso	Descripción	Entrada	Resultado esperado
1	Crear usuario con todos los campos válidos	Datos válidos	Usuario creado exitosamente
2	Crear usuario con email duplicado	Email ya existente	Error de duplicado
3	Autenticación con credenciales válidas	Email y contraseña correctos	Inicio de sesión exitoso
4	Autenticación con contraseña incorrecta	Email válido, contraseña incorrecta	Error de autenticación
5	Crear usuario sin rol	Sin rol asignado	Error o asignación de rol por defecto

2. Entidad Cabin (Cabaña)

Pruebas realizadas:

- **Creación y edición:** Validación de creación y modificación con campos obligatorios completos.
- **Asociación con categorías:** Verificar que la cabaña pueda asociarse correctamente a una o varias categorías.
- **Visualización:** Comprobación de la correcta presentación de datos en listados y detalle.
- **Eliminación:** Confirmar que al eliminar una cabaña, ésta ya no esté disponible para reserva ni en los listados.

Casos de prueba planeados:

Caso	Descripción	Entrada	Resultado esperado
1	Crear cabaña con datos completos	Datos válidos	Cabaña creada correctamente
2	Crear cabaña sin nombre	Nombre vacío	Error de validación
3	Editar cabaña existente	Cambios en campos permitidos	Cabaña actualizada
4	Asociar cabaña a categoría existente	ID de categoría válida	Asociación exitosa
5	Eliminar cabaña	ID válida	Cabaña eliminada y no visible

3. Entidad Category (Categoría)

Pruebas realizadas:

- **CRUD básico:** Crear, editar, listar y eliminar categorías.
- **Validación de nombre único:** Evitar duplicados en nombres de categorías.

Casos de prueba planeados:

Caso	Descripción	Entrada	Resultado esperado
1	Crear categoría con nombre válido	Nombre único	Categoría creada
2	Crear categoría con nombre duplicado	Nombre ya existente	Error por duplicidad
3	Editar nombre de categoría	Nuevo nombre válido	Categoría actualizada
4	Eliminar categoría usada en cabañas	ID de categoría en uso	Error o restricción
5	Listar todas las categorías	Ninguna	Lista completa

4. Entidad Booking (Reserva)

Pruebas realizadas:

- **Creación de reservas:** Validar que no existan reservas duplicadas para la misma cabaña y fechas.
- **Consulta de historial:** Comprobar que un usuario pueda ver su historial de reservas correctamente.
- **Cancelación y modificación:** Verificar que se puedan cancelar o modificar reservas en los casos permitidos.
- **Calificación de reservas:** Asegurar que solo se puedan calificar reservas finalizadas y que el rating se guarde correctamente.

Casos de prueba planeados:

Caso	Descripción	Entrada	Resultado esperado
1	Crear reserva con fechas válidas y sin solapamiento	Datos válidos	Reserva creada
2	Crear reserva duplicada para misma cabaña y fechas	Fechas que coinciden	Error por duplicado
3	Consultar historial de usuario	Usuario con reservas	Listado de reservas

4	Cancelar reserva activa	Reserva válida	Reserva cancelada
5	Calificar reserva finalizada	Rating entre 1 y 5	Calificación guardada

5. Entidad AuthenticationService (Autenticación y Registro)

Pruebas realizadas:

- **Registro de usuario:** Validar que un usuario pueda registrarse correctamente con datos válidos y que se genere un token JWT.
- **Inicio de sesión exitoso:** Comprobar que un usuario pueda autenticarse con email y contraseña válidos, obteniendo un token JWT.
- **Error en inicio de sesión por contraseña inválida:** Verificar que se lance un error o excepción cuando la contraseña sea incorrecta.
- **Error en inicio de sesión por email inexistente:** Comprobar que se lance un error cuando el email no exista en la base de datos.
- **Generación de token JWT:** Asegurar que tras registro o login exitoso se genere y retorne correctamente el token JWT.

Casos de prueba planeados:

Caso	Descripción	Entrada	Resultado esperado
1	Registro con datos válidos	Datos de registro completos	Usuario creado y token JWT generado
2	Registro con email ya existente	Email duplicado	Error de duplicado
3	Inicio de sesión con credenciales válidas	Email y contraseña correctos	Login exitoso y token JWT generado
4	Inicio de sesión con contraseña incorrecta	Email válido, contraseña errónea	Error de autenticación

5	Inicio de sesión con email inexistente	Email no registrado	Error de usuario no encontrado
---	---	---------------------	-----------------------------------

Consideraciones Generales

- Se realizaron pruebas unitarias para validar la lógica de negocio en los servicios.
- Se llevaron a cabo pruebas de integración para validar el correcto funcionamiento entre capas (controlador, servicio, repositorio).
- Se implementaron pruebas manuales de la interfaz de usuario para verificar usabilidad y validación visual.
- Se recomienda la creación de pruebas automatizadas de extremo a extremo (E2E) para garantizar la estabilidad completa del sistema.