
1 Casos de Uso para Requerimientos MUST

Caso de Uso 1: Iniciar sesión en el sistema

- **Descripción:** Permite a usuarios y administradores acceder a la plataforma con credenciales válidas.
- **Actores:** Usuario registrado, Administrador.
- **Precondiciones:**
 - Usuario registrado en Firebase Authentication
 - Servicio de autenticación operativo
- **Flujo principal:**
 1. Usuario ingresa a la página de login
 2. Sistema muestra formulario con campos: email y contraseña
 3. Usuario introduce credenciales y hace clic en *Iniciar sesión*
 4. Sistema verifica credenciales con Firebase Authentication
 5. Si la validación es exitosa:
 - (a) Sistema redirige al dashboard correspondiente al rol
 - (b) Genera token de sesión con vigencia de 24 horas
 6. Sistema registra el acceso en logs de auditoría
- **Flujos alternativos:**
 - **FA1 - Credenciales inválidas:**
 1. Sistema muestra mensaje de error: "Credenciales incorrectas"
 2. Ofrece opción *Recuperar contraseña*
 - **FA2 - Cuenta no verificada:**
 1. Sistema envía correo de verificación
 2. Bloquea acceso hasta confirmación del email
- **Postcondiciones:**
 - Sesión activa con privilegios según rol
 - Token almacenado en cookies seguras

Caso de Uso 2: Filtrar espacios disponibles

- **Descripción:** Búsqueda de espacios con filtros básicos (capacidad, horario, equipamiento)
- **Actores:** Usuario autenticado
- **Precondiciones:**
 - Base de datos de espacios cargada
 - Sesión activa
- **Flujo principal:**

1. Usuario navega a *Buscar espacios*
 2. Sistema muestra interfaz con filtros:
 - Tipo de espacio (dropdown)
 - Capacidad mínima (input numérico)
 - Rango de fechas (selector calendario)
 3. Usuario aplica filtros y hace clic en *Buscar*
 4. Sistema consulta PostgreSQL con los parámetros
 5. Backend devuelve resultados paginados (10 por página)
 6. Frontend muestra tarjetas interactivas con:
 - Imagen representativa
 - Ubicación y capacidad
 - Botón *Reservar*
- **Flujos alternativos:**
 - **FA1 - Sin resultados:**
 1. Muestra mensaje: "No se encontraron espacios con estos filtros"
 2. Ofrece botón para reiniciar búsqueda
 - **Postcondiciones:**
 - Resultados almacenados en caché por 15 minutos
 - Historial de búsqueda registrado en perfil de usuario

Caso de Uso 3: Realizar reserva de espacio

- **Descripción:** Reserva de espacios con validación de concurrencia
- **Actores:** Usuario autenticado
- **Precondiciones:**
 - Espacio disponible en el horario seleccionado
 - Límite de reservas activas no excedido
- **Flujo principal:**
 1. Usuario selecciona espacio de los resultados
 2. Sistema muestra modal de reserva con:
 - Calendario interactivo
 - Selector de hora inicial/final
 - Resumen de costo (si aplica)
 3. Usuario confirma detalles y hace clic en *Reservar*
 4. Sistema inicia transacción ACID en PostgreSQL:
 - (a) Bloquea el espacio para el horario seleccionado
 - (b) Registra reserva en tabla bookings
 - (c) Actualiza estado en vista tiempo real
 5. Sistema envía email de confirmación vía SendGrid
 6. Actualiza el historial de reservas del usuario
- **Flujos alternativos:**
 - **FA1 - Conflicto de reserva:**
 1. Sistema detecta espacio ocupado
 2. Notifica: "Este espacio ya fue reservado"
 3. Sugiere 3 alternativas similares
- **Postcondiciones:**
 - Espacio marcado como ocupado en el sistema
 - Comprobante PDF generado y almacenado en Firebase Storage

Caso de Uso 4: Visualizar ocupación en tiempo real

- **Descripción:** Mapa interactivo con estados actualizados de espacios
- **Actores:** Usuario autenticado, Administrador
- **Precondiciones:**
 - Servicio de WebSockets activo
 - Datos geoespaciales cargados
- **Flujo principal:**
 1. Usuario accede a *Mapa en vivo*
 2. Sistema carga plano arquitectónico con:
 - Iconos codificados por color (verde/rojo)
 - Tooltips con información al hacer hover
 3. Conexión WebSocket establecida con servidor Node.js
 4. Al ocurrir cambios de estado:
 - (a) Servidor emite evento `spaceUpdate`
 - (b) Frontend actualiza iconos correspondientes
 - (c) Muestra notificación toast con detalles
 5. Usuario puede interactuar con elementos para ver detalles
- **Flujos alternativos:**
 - **FA1 - Desconexión WebSocket:**
 1. Sistema cambia a polling cada 30s
 2. Muestra banner de advertencia
 3. Reintenta conexión automáticamente
- **Postcondiciones:**
 - Estados persistentes en base de datos
 - Histórico de cambios registrado para auditoría