

StudyParche

Documento de pruebas y validación

Andrés Olivar & Danna Prada
Universidad Industrial de Santander
2026

PRUEBAS Y VALIDACIÓN

Documento complementario

Capítulo 1

Pruebas de caja negra

1.1. Introducción

Las pruebas de caja negra validan el comportamiento del sistema desde la perspectiva del usuario.

1.1.1. Casos de prueba

Actualización de Grupo de Estudio

■ Precondiciones:

- Usuario registrado y autenticado en la aplicación.
- Usuario pertenece al grupo con el rol de líder.

■ Entrada:

- Nombre del grupo.
- Descripción del grupo.
- Imagen de portada.
- Materia.
- Horario semanal.
- Horario de reunión fija.

■ Pasos de ejecución:

1. Ingresar a la aplicación.
2. Seleccionar la sección “Mis grupos”.
3. En la tarjeta del grupo, pulsar el botón “Editar grupo”.
4. Revisar los campos precargados.
5. Modificar los campos deseados.

6. Seleccionar el botón “Guardar cambios” para confirmar.

- **Resultado esperado:** El sistema muestra un mensaje de confirmación: “Grupo actualizado correctamente” y refleja las modificaciones realizadas.
- **Resultado obtenido:** El sistema muestra un mensaje de confirmación: “Grupo actualizado correctamente” y aparece en la sección “Mis grupos” reflejando las modificaciones realizadas.
- **Estado:** Aprobada.

Salir de Grupo de estudio

- **Precondiciones:**
 - Usuario registrado y autenticado en la aplicación.
 - Usuario pertenece al grupo de estudio del que desee salir.
- **Entrada:** No se requieren datos adicionales por parte del usuario.
- **Pasos de ejecución:**
 1. Ingresar a la aplicación.
 2. Seleccionar la sección “Mis grupos”.
 3. Desplazarse a la sección de grupos en los que el usuario es líder.
 4. En la tarjeta del grupo, dar clic sobre el ícono de salida.
 5. Pulsar el botón “Salir de grupo”.
- **Resultado esperado:** El sistema muestra la pregunta de control, seguido el mensaje de confirmación. El grupo ahora no está disponible en la sección denominada “Mis Grupos”.
- **Resultado obtenido:** El sistema mostró el mensaje: “Vas a salir de “Nombre de grupo”. Esta acción se puede revertir uniéndote de nuevo”, al seleccionar el botón “Salir” posteriormente mostró el mensaje de confirmación: “Saliste del grupo correctamente”. El grupo ahora no está presente en la sección denominada “Mis Grupos”.
- **Estado:** Aprobada.

Consultar Información del Grupo

- **Precondiciones:**
 - Usuario registrado y autenticado en la aplicación.
 - Usuario pertenece a un grupo de estudio.
- **Datos de Entrada:** No se requieren datos adicionales, la información se muestra automáticamente.

■ **Pasos de ejecución:**

1. Ingresar a la aplicación.
2. Seleccionar la sección “Mis grupos”.
3. Seleccionar la tarjeta del grupo para acceder al entorno.
4. Seleccionar la pestaña “Sobre el grupo”.
5. Visualizar:
 - Lista de miembros del grupo.
 - Horario semanal (lunes a sábado por horas).

■ **Resultado esperado:** El usuario puede acceder correctamente a la información del grupo y se visualizan los miembros, el horario semanal y las reuniones del grupo de estudio.

■ **Resultado obtenido:** La pestaña “Sobre el grupo” se abrió correctamente mostrando la lista de miembros, el horario semanal y el calendario con las reuniones resaltadas. La información fue accesible sin errores.

■ **Estado:** Aprobada.

Cancelar solicitud de inscripción a grupo privado

■ **Precondiciones:**

- Usuario registrado y autenticado en la aplicación.
- Usuario ha solicitado unirse a un grupo privado.
- La solicitud de inscripción se encuentra en estado Pendiente.

■ **Datos de Entrada:** No se requieren datos de entrada adicionales.

■ **Pasos de ejecución:**

1. Ingresar a la aplicación.
2. Seleccionar la sección “Mis inscripciones”.
3. Localizar la solicitud pendiente en la lista de inscripciones.
4. Pulsar la opción “Cancelar solicitud”.
5. Confirmar la acción pulsando nuevamente el botón “Cancelar solicitud”.

■ **Resultado esperado:** El sistema muestra una pregunta de control: “Esta acción cambiará el estado de la solicitud a cancelada” después de la confirmación se elimina la solicitud pendiente seleccionada y se muestra un mensaje de confirmación: “Solicitud eliminada con éxito”.

■ **Resultado obtenido:**

El sistema mostró la pregunta de control: “Esta acción cambiará el estado de la solicitud a cancelada”, tras confirmar, se mostró el mensaje de confirmación: “Solicitud cancelada correctamente”. La solicitud cancelada dejó de visualizarse en el historial de inscripciones.

- **Estado:** Aprobada.

Creación de grupo de estudio

- **Precondiciones:** Usuario registrado y autenticado en la aplicación. Usuario vinculado a una escuela y programa académico.
- **Entrada:**
 - Nombre del grupo.
 - Descripción del grupo (Opcional).
 - Imagen de portada (Opcional).
 - Materia.
 - Horario semanal (Opcional).
 - Horario de reunión fija (Opcional).
 - Visibilidad (Selección entre público o privado).
- **Pasos de ejecución:**
 1. Ingresar a la aplicación.
 2. Seleccionar la opción “Crear grupo”.
 3. Completar los campos requeridos y opcionales según corresponda.
 4. Confirmar la creación del grupo.
- **Resultado esperado:** El sistema muestra un mensaje de confirmación: “Grupo creado correctamente”. El nuevo grupo aparece en la lista de grupos del usuario con los datos ingresados.
- **Resultado obtenido:** El sistema mostró el mensaje de confirmación y el grupo se creó correctamente con los datos proporcionados por el usuario.
- **Estado:** Aprobada.

Consultar Inscripciones a Grupos Privados

- **Precondiciones:**
 - Usuario registrado y autenticado en la aplicación.
 - Usuario ha solicitado unirse a uno o más grupos privados.
- **Datos de Entrada:** No se requieren datos de entrada adicionales.
- **Pasos de ejecución:**
 1. Ingresar a la aplicación.

2. Seleccionar la sección “Mis inscripciones”.

- **Resultado esperado:** El sistema muestra el historial de solicitudes de inscripción del usuario a grupos privados. Para cada solicitud se visualiza:
 - Nombre del grupo.
 - Estado de la solicitud (Aceptada, Rechazada o Pendiente).
 - Número de la solicitud.
 - Fecha de creación de la solicitud.
 - Fecha de respuesta (si aplica).
 - En caso de solicitudes aceptadas o rechazadas, se muestra claramente el estado correspondiente junto con los datos asociados.
- **Resultado obtenido:**
 - El historial de inscripciones se visualizó correctamente en la sección “Mis inscripciones”.
 - Se mostraron los datos esperados: nombre del grupo, estado de la solicitud, número de solicitud, fecha de creación y fecha de respuesta.
 - No se presentaron errores en la visualización ni en la carga de datos.
- **Estado:** Aprobada.

1.1.2. Unión e Inscripción a Grupos

Unirse a un Grupo de Estudio Público

- **Precondiciones:**
 - Usuario registrado y autenticado en la aplicación.
 - Usuario ha identificado un grupo al cual desea pertenecer.
- **Entrada:** No se requieren datos adicionales por parte del usuario.
- **Pasos de ejecución:**
 1. Ingresar a la aplicación.
 2. Seleccionar la sección “Buscar grupos”.
 3. Escribir el nombre del grupo o aplicar filtros por materia y visibilidad.
 4. Pulsar el botón “Unirse” en la tarjeta del grupo.
 5. Pulsar el botón “Unirse” en la pregunta de confirmación.
- **Resultado esperado:** El sistema muestra la pregunta de control, seguido el mensaje de confirmación. El grupo ahora aparece en la sección denominada “Mis grupos”.

- **Resultado obtenido:** El sistema mostró el mensaje: “¿Quieres unirte al grupo “Nombre del grupo”?”, posteriormente mostró el mensaje de confirmación: “Ya haces parte de “Nombre del grupo””. El grupo ahora apareció en la sección correspondiente “Mis grupos”.
- **Estado:** Aprobada.

Solicitud de Unión a Grupo Privado

- **Precondiciones:**
 - Usuario registrado y autenticado en la aplicación.
 - Usuario ha identificado un grupo privado al cual desea pertenecer.
- **Entrada:** No se requieren datos adicionales por parte del usuario.
- **Pasos de ejecución:**
 1. Ingresar a la aplicación.
 2. Seleccionar la sección “Buscar grupos”.
 3. Localizar el grupo privado por nombre o filtros.
 4. Pulsar el botón “Solicitar” en la tarjeta del grupo.
 5. Pulsar el botón “Enviar solicitud”.
- **Resultado esperado:** El sistema muestra la pregunta de control, seguido el mensaje de confirmación. El grupo ahora aparece en la sección denominada “Mis inscripciones”.
- **Resultado obtenido:** El sistema mostró el mensaje: “¿Quieres solicitar ingreso al grupo privado “Nombre del grupo”?”, posteriormente mostró el mensaje de confirmación: “Solicitud enviada para “Nombre del grupo””. El grupo ahora apareció en la sección correspondiente “Mis inscripciones”.
- **Estado:** Aprobada.

Notificación de Solicitud de Unión a Grupo Privado

- **Precondiciones:**
 - Usuario registrado y autenticado.
 - Usuario es líder de un grupo de estudio.
- **Datos de Entrada:** No se requieren datos adicionales.
- **Pasos de ejecución:**
 1. Otro usuario solicita unirse al grupo privado.
- **Resultado esperado:** El líder recibe una notificación en la bandeja de notificaciones, mostrando el nombre del grupo y el nombre del solicitante.

- **Resultado obtenido:** El líder recibió la notificación en la bandeja de notificaciones. La notificación mostró correctamente el nombre del grupo y el nombre del solicitante. No se presentaron errores en la entrega ni en la visualización de la notificación.
- **Estado:** Aprobada.

1.1.3. Publicaciones y Recursos

Publicar en el Mural del Grupo

- **Precondiciones:**
 - Usuario registrado y autenticado en la aplicación.
 - Usuario pertenece al grupo de estudio en el que desea agregar una publicación.
- **Entrada:**
 - Título de la publicación.
 - Color del texto (opcional).
 - Fondo de la publicación (opcional).
 - Contenido de la publicación.
- **Pasos de ejecución:**
 1. Ingresar a la aplicación.
 2. Seleccionar la sección “Mis grupos”.
 3. Seleccionar la tarjeta del grupo.
 4. En la pestaña Mural, seleccionar el botón “Nueva publicación”.
 5. Completar el título, color de texto (opcional), fondo de la publicación (opcional) y contenido.
 6. Seleccionar la opción “Publicar”.
- **Resultado esperado:** El sistema muestra el mensaje de confirmación y la publicación es agregada al mural del grupo correspondiente.
- **Resultado obtenido:** El sistema mostró el mensaje: “Publicación guardada. La publicación ya está disponible en el mural”. La publicación ahora está disponible en la sección “Publicadas” en el mural del grupo.
- **Estado:** Aprobada.

Guardar publicación como borrador

■ Precondiciones:

- Usuario registrado y autenticado en la aplicación.
- Usuario pertenece al grupo de estudio en el que desea guardar borrador de una publicación.

■ Entrada:

- Título de la publicación.
- Color del texto (opcional).
- Fondo de la publicación (opcional).
- Contenido de la publicación.

■ Pasos de ejecución:

1. Ingresar a la aplicación.
2. Seleccionar la sección “Mis grupos”.
3. Seleccionar la tarjeta del grupo.
4. En la pestaña Mural, seleccionar el botón “Nueva publicación”.
5. Completar el título, color de texto (opcional), fondo de la publicación (opcional) y contenido.
6. Seleccionar la opción “Guardar borrador”.

- **Resultado esperado:** El sistema muestra el mensaje de confirmación y la publicación es agregada a la sección “Mis borradores” en la sección de mural del grupo correspondiente.

- **Resultado obtenido:** El sistema mostró el mensaje: “Borrador guardado. Puedes seguir editándolo más tarde”. La publicación ahora está disponible en la sección “Mis borradores” en la sección de mural del grupo correspondiente.

- **Estado:** Aprobada.

Publicar Recurso en el grupo

■ Precondiciones:

- Usuario registrado y autenticado en la aplicación.
- Usuario pertenece a un grupo de estudio.

■ Datos de Entrada:

- Título del recurso.
- Descripción del recurso.

- Archivo adjunto (opcional).

■ **Pasos de ejecución:**

1. Ingresar a la aplicación.
2. Seleccionar la sección “Mis grupos”.
3. Seleccionar la tarjeta del grupo para acceder al entorno.
4. Seleccionar la pestaña “Recursos”.
5. Completar el título, descripción y adjuntar archivo si se desea.
6. Pulsar el botón “Guardar recurso”.

■ **Resultado esperado:** El sistema envía el aviso: “Recurso compartido, el recurso ya aparece en el panel del grupo”. El recurso se almacena en la sección “Recursos” del entorno del grupo. El recurso es visible para todos los miembros del grupo.

■ **Resultado obtenido:** El sistema mostró el aviso: “Recurso compartido, el recurso ya aparece en el panel del grupo”. El recurso se almacenó correctamente en la sección “Recursos” del entorno del grupo. El recurso fue visible para todos los miembros del grupo sin errores en la visualización.

■ **Estado:** Aprobada.

Agregar una Nota Adhesiva en un Grupo de Estudio

■ **Precondiciones:**

- Usuario registrado y autenticado en la aplicación.
- Usuario pertenece a un grupo de estudio.

■ **Datos de Entrada:**

- Mensaje que el usuario desea dejar en la nota adhesiva.
- Color seleccionado para la nota adhesiva.
- Configuración de visibilidad: pública (visible para todos los miembros del grupo), privada (visible solo para el usuario y un destinatario), o híbrida (visible para el usuario y varios destinatarios seleccionados).

■ **Pasos de Ejecución:**

1. Ingresar a la aplicación.
2. Seleccionar la opción “Mis grupos”.
3. Seleccionar el grupo de estudio en el que se desea agregar la nota.
4. Ir a la sección de “Notas” (indicada con un símbolo de estrellita).
5. Hacer clic en el botón “Nueva nota”.

6. Escribir el mensaje de la nota adhesiva.
7. Seleccionar el color de la nota.
8. Elegir la configuración de visibilidad: pública, privada o híbrida.
9. Crear la nota.

- **Resultado Esperado:** La nueva nota adhesiva aparece en la sección de notas del grupo y se muestra el color seleccionado así como el contenido escrito en la nota y la fecha de publicación.

- **Resultado Obtenido:**

La nota adhesiva se visualiza correctamente en la sección de notas del grupo, muestra el color elegido, el texto ingresado y la fecha de publicación. No se presentaron errores y no se mostró un mensaje de confirmación, pero la nota es visible de inmediato.

- **Estado:** Aprobado.

Comentar en una publicación del mural de un grupo de estudio

- **Precondiciones:**

- Usuario registrado y autenticado en la aplicación.
- Usuario pertenece a un grupo de estudio con publicaciones en su mural.

- **Datos de Entrada:** Contenido del comentario escrito por el usuario.

- **Pasos de ejecución:**

1. Ingresar a la plataforma/aplicación.
2. Seleccionar la sección “Mis grupos”.
3. Seleccionar el grupo que contiene el mural y la publicación en la que se desea comentar.
4. Ingresar a la sección “Mural” del grupo.
5. Dentro de la sección de mural, pulsar en “Ver detalle” en la publicación donde se quiere comentar.
6. Ingresar el contenido en el campo “Escribir comentario”.
7. Pulsar el botón “Enviar comentario”.

- **Resultado esperado:**

El comentario se añade correctamente a la sección de comentarios de la publicación seleccionada. El comentario recién creado se visualiza junto a los demás comentarios existentes en la publicación.

- **Resultado obtenido:**

El comentario se publicó exitosamente y se visualizó inmediatamente en la sección de comentarios de la publicación. No se presentaron errores en la publicación ni en la visualización.

- **Estado:** Aprobada.

Reaccionar a un comentario en una publicación del mural de un grupo de estudio

- **Precondiciones:**

- Usuario registrado y autenticado en la aplicación.
- Usuario pertenece a un grupo de estudio.

- **Datos de Entrada:** No se requieren datos adicionales.

- **Pasos de ejecución:**

1. Ingresar a la aplicación.
2. Seleccionar la sección “Mis grupos”.
3. Ubicar y seleccionar el grupo donde se encuentra la publicación.
4. Ir a la sección “Mural” del grupo.
5. Ubicar la publicación donde está el comentario al que se quiere reaccionar.
6. Pulsar “Ver detalle” en la publicación para visualizar los comentarios.
7. Localizar el comentario específico.
8. Hacer clic en el ícono de corazón para reaccionar al comentario.

- **Resultado Esperado:**

El contador de reacciones del comentario se incrementa en uno y se visualiza inmediatamente el cambio en el número de reacciones del comentario.

- **Resultado Obtenido:**

La reacción fue aplicada exitosamente al comentario y el contador de reacciones se incrementó en uno y se reflejó visualmente. No se presentaron errores en la reacción ni en la visualización.

- **Estado:** Aprobada.

Responder a un comentario en una publicación del mural de un grupo de estudio

- **Precondiciones:**

- Usuario registrado y autenticado en la aplicación.
- Usuario pertenece a un grupo de estudio.

- **Datos de Entrada:** No se requieren datos adicionales.

- **Pasos de ejecución:**

1. Ingresar a la plataforma/aplicación.

2. Seleccionar la sección “Mis grupos”.
3. Ubicar y seleccionar el grupo donde se encuentra la publicación.
4. Ir a la sección “Mural” del grupo.
5. Seleccionar la publicación donde está el comentario al que se quiere responder.
6. Pulsar “Ver detalle” en la publicación para visualizar los comentarios.
7. Ubicar el comentario específico al que se desea responder.
8. Seleccionar la opción “Responder” debajo de ese comentario.
9. Escribir la respuesta en el campo de texto.
10. Hacer clic en “Responder” para publicar la respuesta.

■ **Resultado Esperado:**

La respuesta se muestra inmediatamente como un comentario anidado (en hilo) debajo del comentario original, desplazado ligeramente a la derecha para indicar visualmente que es una respuesta.

■ **Resultado Obtenido:**

La respuesta fue publicada con éxito y se observa la nueva respuesta como un hilo anidado debajo del comentario original en el mural del grupo. No se presentaron errores y la estructura en hilo se mostró correctamente.

■ **Estado:** Aprobada.

Notificación de Nueva Publicación

■ **Precondiciones:**

- Usuario registrado y autenticado.
- Usuario pertenece a un grupo de estudio.

■ **Datos de Entrada:** No se requieren datos adicionales.

■ **Pasos de ejecución:**

1. Un miembro del grupo realiza una publicación.

■ **Resultado esperado:**

- Todos los miembros del grupo reciben una notificación en la bandeja indicando el nuevo recurso o publicación.
- La notificación incluye el nombre del grupo y el nombre del miembro que la realizó.

■ **Resultado obtenido:**

- Todos los miembros del grupo recibieron la notificación en la bandeja indicando la nueva publicación.

- La notificación mostró correctamente el nombre del grupo y el nombre del miembro que la realizó.
- No se presentaron errores en la entrega ni en la visualización de la notificación.

■ **Estado:** Aprobada.

Notificación de Reacción a Publicación o Comentario

■ **Precondiciones:**

- Usuario registrado y autenticado en la aplicación.
- Usuario pertenece a un grupo de estudio.
- Usuario ha realizado una publicación o un comentario en el grupo.
- Otro usuario ha reaccionado a esa publicación o comentario.

■ **Datos de Entrada:** No se requieren datos adicionales.

■ **Pasos de ejecución:**

1. Ingresar a la aplicación.
2. Dirigirse al apartado de “Notificaciones”.
3. Verificar la notificación de la nueva reacción.

■ **Resultado Esperado:**

- Se muestra una notificación con el mensaje “El usuario [nombre] ha reaccionado a tu publicación” o “El usuario [nombre] ha reaccionado a tu comentario”, según corresponda.

■ **Resultado Obtenido:**

- La notificación aparece correctamente en el apartado de notificaciones.
- El usuario puede visualizar que ha recibido una reacción a su publicación o comentario.
- No se presentaron errores durante la visualización de la notificación.

■ **Estado:** Aprobada.

1.1.4. Chat de Grupo

Envío de Mensajes de Texto en Chat de Grupo

■ **Precondiciones:**

- Usuario registrado y autenticado en la aplicación.
- Usuario pertenece a un grupo de estudio y un chat relacionado a ese grupo.

- **Datos de Entrada:** Texto escrito por el usuario (ejemplo: “Hola equipo”).
- **Pasos de ejecución:**
 1. Ingresar a la aplicación.
 2. Seleccionar la burbuja de chat o entrar al entorno del grupo.
 3. Escribir un mensaje en el campo de texto.
 4. Pulsar “Enviar” o dar Enter.
- **Resultado esperado:**
 - El mensaje aparece inmediatamente en el chat.
 - Todos los miembros del grupo reciben el mensaje de forma automática.
 - El historial del chat se actualiza mostrando el nuevo mensaje junto con los anteriores.
- **Resultado obtenido:**
 - El mensaje escrito por el usuario se mostró inmediatamente en el chat.
 - Todos los miembros del grupo recibieron el mensaje de forma automática.
 - El historial del chat se actualizó correctamente mostrando el nuevo mensaje junto con los anteriores.
- **Estado:** Aprobada.

Envío de Mensajes de Audio en Chat de Grupo

- **Precondiciones:**
 - Usuario registrado y autenticado en la aplicación.
 - Usuario pertenece a un grupo de estudio.
- **Datos de Entrada:** Grabación de audio realizada por el usuario.
- **Pasos de ejecución:**
 1. Ingresar al chat del grupo.
 2. Pulsar el ícono de micrófono.
 3. Grabar el mensaje de audio.
 4. Soltar para Enviar.
- **Resultado esperado:**
 - El mensaje de audio aparece en el chat con un reproductor.
 - Todos los miembros del grupo pueden reproducirlo.

- El historial del chat se actualiza mostrando el nuevo mensaje de audio junto con los anteriores.

■ **Resultado obtenido:**

- El mensaje de audio se grabó y envió exitosamente.
- El audio apareció en el chat con su reproductor integrado.
- Todos los miembros del grupo pudieron reproducirlo sin errores.
- El historial del chat se actualizó correctamente mostrando el nuevo mensaje de audio junto con los anteriores.

■ **Estado:** Aprobada.

Editar chat del grupo por el líder

■ **Precondiciones:**

- Usuario registrado y autenticado en la aplicación.
- Usuario es líder del grupo de estudio.

■ **Datos de Entrada:**

- Título del chat.
- Descripción del chat.
- Imagen de fondo del chat.

■ **Pasos de ejecución:**

1. Ingresar al entorno del grupo.
2. Seleccionar la burbuja de chat.
3. Seleccionar el chat a editar.
4. Pulsar el botón “Editar”.
5. Modificar los campos deseados.
6. Pulsar el botón “Guardar cambios”.

- **Resultado esperado:** El sistema muestra un mensaje de confirmación: “Chat actualizado correctamente” y automáticamente se observa el cambio del fondo del chat y el nombre del chat si se editó . Al pulsar botón “Info” se observa la sección de información del chat específico se pueden ver los cambios realizados en la descripción.

- **Resultado obtenido:** El líder editó el chat y al pulsar “Guardar cambios” el sistema mostró el mensaje de confirmación: “Chat actualizado correctamente”, fue instantánea la actualización del fondo del chat o del nuevo nombre del chat. En la sección de información dentro del chat específico se visualizaron los cambios realizados.

■ **Estado:** Aprobada.

1.1.5. Notificaciones

Marcar como Leída una Notificación

■ **Precondiciones:**

- Usuario registrado y autenticado en la aplicación.
- Usuario pertenece a un grupo de estudio.
- El usuario tiene al menos una notificación en su bandeja.

■ **Pasos de Ejecución:**

1. Ingresar a la aplicación.
2. En el sidebar fijo de la izquierda, seleccionar “Notificaciones”.
3. Dentro del apartado de notificaciones, hacer clic en “Marcar como leída”.

■ **Resultado Esperado:**

- La notificación se ve de un color más opaco, indicando que ha sido marcada como leída.

■ **Resultado Obtenido:**

- La notificación aparece visualmente atenuada, cumpliendo con el cambio de color esperado.

■ **Estado:** Aprobado.

Notificación de Solicitud Personalizada

■ **Precondiciones:**

- Usuario registrado y autenticado en la aplicación.
- Usuario pertenece al grupo de estudio.

■ **Entrada:**

- No se requieren datos.

■ **Pasos de ejecución:**

1. Ingresar a la aplicación.
2. En el sidebar fijo de la izquierda, seleccionar la opción “Notificaciones”.
3. En la sección “Notificaciones”, visualizar la notificación correspondiente a la Solicitud personalizada.

■ **Resultado esperado:** El sistema muestra en la sección de notificaciones la solicitud personalizada enviada por otro usuario, incluyendo el mensaje: “holiii dame un folder de calculoo” y el remitente “Leida”.

- **Resultado obtenido:** El sistema muestra en la sección de notificaciones la solicitud personalizada con el mensaje “holiii dame un folder de calculoo” enviada por el usuario “Leida”, confirmando que la notificación fue recibida correctamente.
- **Estado:** Aprobada.

Eliminar una Notificación de la Bandeja de Notificaciones

- **Precondiciones:**
 - Usuario registrado y autenticado en la aplicación.
 - El usuario tiene al menos una notificación guardada.
- **Pasos de Ejecución:**
 1. Ingresar a la aplicación.
 2. En el sidebar fijo de la izquierda, seleccionar “Notificaciones”.
 3. Ubicar la notificación que se desea eliminar.
 4. Hacer clic en el ícono de la basurita (el cual muestra el indicador “Eliminar notificación”).
- **Resultado Esperado:**
 - Aparece un aviso indicando “Notificación eliminada”.
 - La notificación deja de aparecer en la lista de notificaciones.
- **Resultado Obtenido:**
 - El aviso “Notificación eliminada” se muestra correctamente y la notificación ya no está en la lista.
- **Estado:** Aprobado.

1.1.6. Perfil y Reseñas

Editar Perfil De Usuario

- **Precondiciones:**
 - Usuario registrado y autenticado en la aplicación.
- **Datos de Entrada:**
 - Foto de perfil.
 - Foto de fondo para la sidebar.
 - Biografía corta del usuario.

■ **Pasos de ejecución:**

1. Ingresar a la aplicación.
2. En la sidebar fija, seleccionar la opción “Editar perfil”.
3. Cargar una nueva foto de perfil.
4. Cargar una nueva foto de fondo para la sidebar.
5. Escribir una biografía corta en el campo de información personal.
6. Pulsar el botón “Guardar”.

■ **Resultado esperado:** Los cambios realizados (foto de perfil, foto de fondo y biografía) se reflejan inmediatamente en la sidebar del perfil de usuario.

■ **Resultado obtenido:** Los cambios se guardaron exitosamente. La nueva foto de perfil y la foto de fondo se visualizaron correctamente en la sidebar. La biografía corta se mostró en la sección de perfil sin errores. La actualización fue inmediata y consistente en la interfaz.

■ **Estado:** Aprobada.

Reaccionar a un Grupo de Estudio

■ **Precondiciones:**

- Usuario registrado y autenticado en la aplicación.
- Usuario pertenece a un grupo de estudio.

■ **Datos de Entrada:** No se requieren datos adicionales.

■ **Pasos de ejecución:**

1. Ingresar a la plataforma/aplicación.
2. Seleccionar la sección “Mis grupos”.
3. Seleccionar el grupo en el que se encuentra la publicación a la cual se desea reaccionar.
4. Ingresar a la sección “Mural” del grupo.
5. Ubicar la publicación donde se quiere reaccionar.
6. Pulsar el botón en forma de corazón para dar “like”.

■ **Resultado esperado:**

El contador de “likes” de la publicación se incrementa en uno. Se muestra visualmente el cambio del número de reacciones en la publicación.

■ **Resultado obtenido:**

El contador de “likes” se incrementó en uno y se visualizó inmediatamente. No se presentaron errores en la reacción ni en la visualización del conteo.

■ **Estado:** Aprobada.

Asignar Reseña al Grupo de Estudio

■ Precondiciones:

- Usuario registrado y autenticado en la aplicación.
- Usuario pertenece a un grupo de estudio.

■ Datos de Entrada: Texto de la reseña escrita por el usuario.

■ Pasos de ejecución:

1. Ingresar a la aplicación.
2. Autenticarse con credenciales válidas.
3. Seleccionar la sección “Mis grupos”.
4. En la tarjeta del grupo, pulsar el botón “Reseñas”.
5. En el campo “Escribe tu reseña”, redactar el texto deseado.
6. Pulsar el botón “Publicar reseña”.

■ Resultado esperado:

- La reseña se añade a la lista de reseñas del grupo de estudio.
- El sistema muestra un mensaje de confirmación: “Reseña publicada correctamente”.
- La reseña recién creada aparece junto a las demás reseñas visibles en la sección.

■ Resultado obtenido:

- La reseña se publicó exitosamente en la sección de reseñas del grupo.
- El mensaje de confirmación “Reseña publicada correctamente” se mostró en pantalla.
- La reseña se visualizó inmediatamente junto a las demás reseñas existentes.
- No se presentaron errores en la publicación ni en la visualización.

■ Estado: Aprobada.

Editar una Reseña de un Grupo de Estudio

■ Precondiciones:

- Usuario registrado y autenticado en la aplicación.
- Usuario pertenece a un grupo de estudio.
- El usuario ya ha publicado una reseña en ese grupo de estudio.

■ Datos de Entrada: Contenido de la nueva reseña.

■ Pasos de ejecución:

1. Ingresar a la aplicación.
2. Seleccionar la sección “Mis grupos”.
3. Dentro de “Mis grupos”, seleccionar el botón “Reseñas”.
4. Localizar la reseña que se desea editar.
5. Pulsar el ícono del lápiz que indica “Editar reseña”.
6. Editar el contenido de la reseña como se desee.
7. Pulsar el botón “Guardar”.

■ **Resultado esperado:**

La reseña se actualiza correctamente en la sección de reseñas. Se muestra un aviso con un check que dice “Reseña actualizada”.

■ **Resultado obtenido:**

La reseña fue editada de forma exitosa, el sistema reflejó el aviso de confirmación con un check: “Reseña actualizada” y la reseña modificada se visualizó inmediatamente en la sección de reseñas del grupo. No se presentaron errores en la edición ni en la visualización.

■ **Estado:** Aprobada.

Asignar Reseña al miembro de un grupo de Estudio

■ **Precondiciones:**

- Usuario registrado y autenticado en la aplicación.
- Usuario pertenece a un grupo de estudio existente.

■ **Datos de Entrada:**

- Selección del grupo de estudio.
- Texto de la reseña escrita por el usuario.

■ **Pasos de ejecución:**

1. Ingresar a la aplicación.
2. Seleccionar la sección “Mis grupos”.
3. En la tarjeta del grupo, pulsar la foto de perfil del líder o en la pestaña “Sobre el grupo”, pulsar la foto de perfil de un miembro del grupo de estudio.
4. En el campo “Grupo en común”, seleccionar el grupo correspondiente.
5. En el campo “Escribe tu reseña”, redactar el texto deseado.
6. Pulsar el botón “Enviar reseña”.

■ **Resultado esperado:**

- La reseña se añade a la lista de reseñas del grupo de estudio.
- El sistema muestra un mensaje de confirmación: “Reseña publicada correctamente”.
- La reseña recién creada aparece junto a las demás reseñas visibles en la sección.

■ **Resultado obtenido:**

- La reseña se publicó exitosamente en la sección de reseñas del grupo.
- El mensaje de confirmación “Reseña publicada correctamente” se mostró en pantalla.
- La reseña se visualizó inmediatamente junto a las demás reseñas existentes.
- No se presentaron errores en la publicación ni en la visualización.

■ **Estado:** Aprobada.

Eliminar una reseña de un Grupo de Estudio

■ **Precondiciones:**

- Usuario registrado y autenticado en la aplicación.
- Usuario pertenece a un grupo de estudio.
- El usuario ha realizado previamente una reseña en el grupo de estudio.

■ **Datos de Entrada:** No se requieren datos adicionales.

■ **Pasos de ejecución:**

1. Ingresar a la aplicación.
2. Seleccionar la sección “Mis grupos”.
3. Localizar el grupo en el que se quiere eliminar la reseña.
4. Ingresar a la sección “Reseñas” del grupo.
5. Localizar la reseña que se desea eliminar.
6. Seleccionar el botón “Eliminar reseña” (ícono de basurita).
7. Aparecerá un mensaje de confirmación indicando: “Esta acción no se puede deshacer. Esta acción cambiará el estado de la reseña a eliminada”.
8. Confirmar haciendo clic en “Eliminar”.

■ **Resultado esperado:** La reseña se elimina correctamente de la lista de reseñas del grupo y se muestra una advertencia o mensaje tipo warning que dice “Reseña eliminada”. El sistema refleja que la reseña ya no está disponible en el historial de reseñas.

■ **Resultado obtenido:**

La reseña dejó de visualizarse en la lista de reseñas del grupo, inmediatamente el sistema mostró la advertencia de confirmación: “Reseña eliminada”. No se presentaron errores en la eliminación ni en la visualización posterior.

■ **Estado:** Aprobada.

1.1.7. Funciones Especiales

Activar el Botón de Pánico

- **Precondiciones:**

- Usuario registrado y autenticado en la aplicación.
- Usuario pertenece a un grupo de estudio.

- **Datos de Entrada:** No se requieren datos adicionales.

- **Pasos de ejecución:**

1. Ingresar a la aplicación.
2. Seleccionar la sección “Mis grupos”.
3. Seleccionar la tarjeta del grupo para acceder al entorno.
4. Seleccionar la pestaña “Botón de Pánico”.
5. Pulsar el botón “Pánico”.

- **Resultado esperado:** El sistema envía una notificación automática a todos los miembros del grupo, incluyendo al líder, además envía un mensaje automático por parte del usuario al chat grupal. El usuario puede ver que se indica en cuánto tiempo podrá volver a activarlo en ese grupo.

- **Resultado obtenido:** Al activar el botón de pánico, el sistema envió la notificación correctamente a todos los miembros del grupo y al líder, así mismo en el chat de ese grupo se envió el mensaje automático “Activé el Boton del Panico. Necesito ayuda urgente ahora” por parte de ese usuario. Se observa correctamente el mensaje de confirmación de “Alerta enviada”. Se informó que podrá volver a usarlo en x minutos, se mostró en pantalla sin errores la disminución en minutos.

- **Estado:** Aprobada.

Buscador de Parches BDP Exitoso

- **Precondiciones:**

- Usuario registrado en la aplicación.
- Usuario ha ingresado a la aplicación (no es necesario pertenecer a un grupo de estudio).

- **Datos de Entrada:** Materia seleccionada por el usuario para estudiar (ejemplo: “Álgebra Lineal”).

- **Pasos de ejecución:**

1. Ingresar a la aplicación.
2. Seleccionar el botón “Unirse al BDP”.

3. Escribir la materia que desea estudiar.
4. Confirmar la solicitud de búsqueda de compañeros pulsando “Buscar aliados en la UIS”.
5. Se inicia un cronómetro de espera denominado “Tiempo en cola”.
6. El sistema busca automáticamente otros dos usuarios que hayan solicitado LFR con la misma materia.
7. Se indica que ya se encontró parche y se observa un lugar de reunión recomendado.

■ **Resultado esperado:**

- Mientras se realiza la búsqueda, el estado mostrado en el cuadro es “Encontrando”.
- Si se encuentran dos compañeros adicionales:
 - El estado cambia a “Encontrado”.
 - Se asigna automáticamente un lugar de reunión al azar dentro de la universidad.
- Si no se encuentran compañeros dentro del tiempo de espera:
 - El sistema finaliza la búsqueda al cumplirse el cooldown de 5 minutos.
 - Se muestra un mensaje de cierre de búsqueda.

■ **Resultado obtenido:**

- El sistema mostró el estado “Encontrando” durante la búsqueda.
- Luego cambió a “Encontrado” al localizar dos compañeros adicionales.
- Se asignó un lugar de reunión en la universidad.
- La sesión BDP se inició correctamente.

■ **Estado:** Aprobada.

Solicitud Personalizada

■ **Precondiciones:**

- Usuario registrado y autenticado en la aplicación.
- Usuario pertenece al grupo de estudio.

■ **Entrada:**

- Mensaje de solicitud personalizada.

■ **Pasos de ejecución:**

1. Ingresar a la aplicación.
2. Seleccionar la sección “Mis grupos”.
3. Seleccionar el grupo de estudio correspondiente.

4. Entrar a la pestaña “Sobre el grupo”.
5. Pulsar sobre el círculo del perfil de un miembro.
6. Seleccionar la opción “S. Personalizada”.
7. Ingresar el mensaje de solicitud.
8. Pulsar el botón “Enviar solicitud”.

- **Resultado esperado:** El sistema muestra un mensaje de confirmación indicando que la solicitud personalizada fue enviada correctamente y se registra en el historial del grupo.
- **Resultado obtenido:** El sistema muestra el mensaje de confirmación “Solicitud personalizada enviada correctamente” y la solicitud aparece reflejada en la sección de notificaciones de el usuario receptor del mensaje.
- **Estado:** Aprobada.

1.1.8. Añadir escuelas, carreras y facultades

- **Entrada:** Rellenar los campos en el panel de administrador.
- **Salida:** El sistema guarda la nueva facultad, carrera o escuela en la base de datos.

Capítulo 2

Pruebas de caja blanca

2.1. Introducción

Las pruebas de caja blanca validan la lógica interna del sistema.

2.1.1. Casos de prueba

2.1.2. Pruebas de Grupo de Estudio

Validación de campos obligatorios en creación de grupo de estudio

ID	Campo omitido	Acción realizada	Resultado esperado	Resultado obtenido	Estado
CP-01	NombreGrupoEstudio	Crear grupo sin nombre	Error indicando campo faltante	El nombre del grupo es obligatorio.	Aprobado
CP-02	Descripcion	Crear grupo sin descripción	Error indicando campo faltante	La descripción del grupo es obligatoria.	Aprobado
CP-03	IdMat	Crear grupo sin materia	Error indicando campo faltante	La materia es obligatoria.	Aprobado

Cuadro 2.1: Casos de prueba con campos omitidos según definición en la base de datos

Nota: Se marca como *Aprobado* porque el sistema identifica el campo específico omitido.

Listing 2.1: Validación de campo NombreGrupoEstudio en entidad JPA

```
1 @Column(name = "NombreGrupoEstudio", nullable = false, length = 200)
2 private String nombreGrupoEstudio;
```

Listing 2.2: Validacion frontend para nombre de grupo

```

1 if (!nombreGrupo.trim()) {
2     setError('El_nombre_del_grupo_es_obligatorio.');
```

Validación de campos obligatorios en actualización de grupo de estudio

ID	Campo Omitido	Acción realizada	Resultado esperado	Resultado obtenido	Estado
CP-04	NombreGrupoE.	Intentar actualizar grupo quitando el nombre	El sistema debe impedir la actualización y mostrar advertencia	El nombre del grupo es obligatorio	Aprobado
CP-05	Descripcion	Intentar actualizar grupo quitando la descripción	El sistema debe impedir la actualización y mostrar advertencia	La descripción del grupo es obligatoria	Aprobado
CP-06	IdMat	Intentar actualizar grupo quitando la materia	El sistema debe impedir la actualización o mantener la materia anterior	El sistema conserva la materia previa y muestra mensaje genérico de validación	Aprobado
CP-07	Rol de líder	Intentar actualizar grupo sin rol de líder	El sistema debe impedir la acción y no mostrar la opción de edición	La opción de actualizar no aparece para miembros sin rol de líder	Aprobado

Cuadro 2.2: Casos de prueba para validación de campos obligatorios y permisos en la actualización de grupos de estudio

Listing 2.3: Validación frontend para nombre de grupo en actualización (CP-04)

```

1         if (!nombreGrupo.trim()) {
2             notifyError('El_nombre_del_grupo_es_obligatorio.');
```

Listing 2.4: Lógica de actualización condicional en backend

```

1 // En ActualizarDominioDesdeDTO.java, línea 17-19
2     if (dto.getNombreGrupoEstudio() != null) {
3         grupo.setNombreGrupoEstudio(dto.getNombreGrupoEstudio());
4     }

```

Salir de Grupo de estudio

ID	Condición	Acción realizada	Resultado esperado	Resultado obtenido	Estado
CP-08	Rol de miembro	Ejecutar salida del grupo siendo miembro	El sistema elimina al usuario de la lista de miembros, el grupo sigue activo	El miembro cambia su estado a DESERTOR y el grupo permanece	Aprobado
CP-9	Rol de líder	Ejecutar salida del grupo siendo líder	El sistema quita privilegios de líder y cambia el estado del usuario a desertor.	El miembro se elimina correctamente y el grupo permanece.	Aprobado

Cuadro 2.3: Casos de prueba para salida de grupos de estudio con rol de líder y miembro

Listing 2.5: Lógica backend para salir de grupo (CP-08)

```

1 // En MiembroGrupoRepositorioSQL.java, líneas 346-348
2 orm.setEstadoGrupo(MiembroGrupoEntidad.EstadoGrupo.Desertor);
3 orm.setFechaSalida(LocalDateDateTime.now());
4     entityManager.flush();

```

Listing 2.6: Confirmación frontend para salir de grupo

```

1 const confirmarSalida = await confirm({
2     title: 'Salir_del_grupo',
3     description: 'Vas a salir de "${grupo.nombreGrupoEstudio}".
4     Esta acción se puede revertir uniendote de nuevo.',
5     confirmText: 'Salir',
6     cancelText: 'Cancelar',
7 });

```

Consultar Información del Grupo

ID	Campo	Acción realizada	Resultado esperado	Resultado obtenido	Estado
CP-10	Información Grupo	Consultar información del grupo con horario y miembros	El sistema debe mostrar datos completos del grupo	Se muestran horario y lista de miembros	Aprobado
CP-11	Información Grupo	Consultar información del grupo sin horario	El sistema debe mostrar datos disponibles y omitir horario	Se muestran miembros, sin horario	Aprobado

Cuadro 2.4: Casos de prueba para consulta de información de grupos de estudio

2.1.3. Unión e Inscripción a Grupos

Unión e Inscripción a Grupos

ID	Condición	Acción realizada	Resultado esperado	Resultado obtenido	Estado
CP-12	Grupo público disponible	Ejecutar unión a grupo público desde buscador	El sistema agrega al usuario como miembro y el grupo ya no aparece en buscador	Usuario agregado correctamente y grupo oculto en buscador	Aprobado
CP-13	Grupo privado disponible	Ejecutar solicitud de unión a grupo privado	El sistema registra solicitud y la muestra en “Mis inscripciones”.	Solicitud registrada, visible en pestaña de inscripciones y el grupo oculto en buscador	Aprobado

Cuadro 2.5: Casos de prueba para unión a grupos públicos y privados

Casos validación excepciones Unión Grupos

ID	Condición	Acción realizada	Resultado esperado	Resultado obtenido	Estado
CP-14	Grupo inexistente	Intentar unirse a grupo no registrado	El sistema no muestra el grupo en buscador ni permite unión	Grupo no aparece en opciones de unión	Aprobado
CP-15	Usuario ya miembro	Intentar unirse a grupo público donde ya es miembro	El sistema filtra y no muestra el grupo en buscador	Grupo no aparece en buscador para miembros existentes	Aprobado

Cuadro 2.6: Casos de prueba excepciones de grupo inexistente y si ya es miembro

Listing 2.7: Lógica interna del proceso de unión a grupo (puntos de decisión)

```

1
2 if (!existentes.isEmpty()) {
3     MiembroGrupoEntidad orm = existentes.get(0);
4     // Rama 1: Usuario expulsado
5     if (orm.getEstadoGrupo() == EstadoGrupo.Expulsado) {
6         throw new OperacionNoPermitidaException(
7             "El_usuario_ha_sido_expulsado_del_grupo_y_no_puede_
8             volver_a_unirse"
9         );
10    }
11
12    // Rama 2: Usuario desertor (puede reingresar)
13    if (orm.getEstadoGrupo() == EstadoGrupo.Desertor) {
14        orm.setEstadoGrupo(EstadoGrupo.Activo);
15        orm.setFechaIngreso(LocalDateTime.now());
16        orm.setFechaSalida(null);
17        orm.setTipoMiembro(TipoMiembro.Miembro);
18        return MiembroGrupoMapper.toDomain(orm);
19    }
20
21    // Rama 3: Ya es miembro activo
22    throw new ConflictoException("El_usuario_ya_es_miembro_del_
23        grupo");
24
25    // Rama 4: Usuario nuevo (no existe registro)
26    MiembroGrupoEntidad orm = new MiembroGrupoEntidad();
27    orm.setUsuarioID(usuarioId);
28    orm.setUsuarioEscID(usuarioEscId);
29    orm.setGrupoEstudioID(grupoId);

```

```

29     orm.setEstadoGrupo(EstadoGrupo.Activo);
30
31     return MiembroGrupoMapper.toDomain(orm);

```

2.1.4. Publicaciones y recursos

Casos validación Publicar en el mural

ID	Condición de entrada	Acción realizada	Resultado esperado	Resultado obtenido	Estado
CP-19	Publicación válida	Ejecutar publicación con título y contenido	El sistema guarda y muestra la publicación en el mural	Publicación visible en mural	Aprobado
CP-20	Publicación sin título	Intentar publicar sin título	El sistema impide la acción y muestra advertencia	Título vacío añade un título antes de guardar la publicación	Aprobado
CP-21	Publicación sin contenido	Intentar publicar sin contenido	El sistema impide la acción y muestra advertencia	Contenido vacío añade al menos un bloque antes de guardar la publicación	Aprobado
CP-22	Publicación vacía	Intentar publicar sin título ni contenido	El sistema impide la acción y muestra advertencia	Campos incompletos Debes completar el título y el contenido antes de guardar la publicación.	Aprobado

Cuadro 2.7: Casos de prueba para publicar en mural y guardar borradores

Guardar Publicación como Borrador

ID	Campo Omitido/Condición	Acción realizada	Resultado esperado	Resultado obtenido	Estado
CP-23	Borrador válido	Ejecutar guardar borrador con título y contenido	El sistema guarda la publicación en estado borrador	Publicación visible en sección de borradores	Aprobado
CP-24	Borrador sin título	Intentar guardar borrador sin título	El sistema impide la acción y muestra advertencia	Título vacío añade un título antes de guardar la publicación	Aprobado
CP-25	Borrador sin contenido	Intentar guardar borrador sin contenido	El sistema impide la acción y muestra advertencia	Contenido vacío añade al menos un bloque antes de guardar la publicación	Aprobado
CP-26	Borrador vacío	Intentar guardar borrador sin título ni contenido	El sistema impide la acción y muestra advertencia	Campos incompletos Debes completar el título y el contenido antes de guardar la publicación.	Aprobado

Cuadro 2.8: Casos de prueba para publicar en mural y guardar borradores

Listing 2.8: Lógica interna de creación de publicaciones y procesamiento de imágenes

```

1
2 // ASIGNACION DE ESTADO POR DEFECTO
3 if (dto.getEstado() != null) {
4     publicacion.setEstado(EstadoPublicacion.valueOf(dto.getEstado()));
5     } else {
6         publicacion.setEstado(EstadoPublicacion.PUBLICADO);
7     }
8 // PROCESAMIENTO DE BLOQUES
9 if (dto.getBloques() != null && !dto.getBloques().isEmpty()) {
10     for (BloquePublicacionDTO bloqueDTO : dto.getBloques()) {
11         // DECISION POR TIPO DE BLOQUE
12         if ("Imagen".equals(bloqueDTO.getTipoBloque())) {
13             String fileKey = bloqueDTO.getFileKey();
14             // VALIDACION DE ARCHIVO EXISTENTE
15             if (fileKey != null && archivos != null &&
16                 archivos.containsKey(fileKey)) {
17                 MultipartFile archivo = archivos.get(fileKey);

```

```

17         if (archivo != null && !archivo.isEmpty()) {
18             // VALIDACION DE FORMATO DE IMAGEN
19             if (!storageServicio.esImagenValida(archivo)) {
20                 throw new ValidacionException("Imagen_no_valida");
21             }
22
23             String urlImagen = storageServicio.guardarImagen(archivo,
24                 publicacionID);
25
26             bloque.setUrlImagen(urlImagen);
27         } else if (bloqueDTO.getUrlImagen() != null) {
28             // USO DE URL EXISTENTE
29             bloque.setUrlImagen(bloqueDTO.getUrlImagen());
30         } else {
31             // BLOQUE DE TEXTO
32             bloque.setContenidoTexto(bloqueDTO.getContenidoTexto());
33         }
34     }
35 }
36
37 // PROCESAMIENTO DE FONDO
38 if (archivos != null && archivos.containsKey("fondoPublicacion")) {
39     MultipartFile archivoFondo = archivos.get("fondoPublicacion");
40     if (archivoFondo != null && !archivoFondo.isEmpty()) {
41         if (!storageServicio.esImagenValida(archivoFondo)) {
42             throw new ValidacionException("Fondo_no_valido");
43         }
44         String urlFondo = storageServicio.guardarFondo(archivoFondo,
45             publicacionID);
46     }
47 }
48
49 // VALIDACION EN SERVICIO DE ALMACENAMIENTO
50 if (archivo == null || archivo.isEmpty()) {
51     throw new ValidacionException("Archivo_vacio");
52 }

```

Publicar Recurso

ID	Campo ó Condición	Acción realizada	Resultado esperado	Resultado obtenido	Estado
CP-24	Recurso con archivo	Ejecutar publicación con título y archivo adjunto	El sistema guarda y comparte el recurso en el grupo	Recurso publicado correctamente	Aprobado
CP-25	Recurso sin descripción ni archivo	Ejecutar publicación con título y descripción sin archivo	El sistema guarda y muestra advertencia de descripción obligatoria	Mensaje “Debes agregar una descripción” mostrado	Aprobado
CP-26	Recurso sin título	Ejecutar publicación sin título (con archivo o descripción)	El sistema impide la acción y muestra advertencia de título obligatorio	Mensaje “Debes ingresar un título” mostrado	Aprobado
CP-27	Recurso vacío	Intentar publicar sin título, archivo ni descripción	El sistema impide la acción y muestra advertencia de título obligatorio	Mensaje “Debes ingresar un título” mostrado	Aprobado

Cuadro 2.9: Casos de prueba para publicación de recursos en grupos de estudio

Listing 2.9: Lógica interna para la publicación de recursos y manejo de archivos

```

1
2 // DECISION PRINCIPAL: EXISTE ARCHIVO O ES SOLO ENLACE
3 if (archivo != null) {
4     // CREACION DE DIRECTORIO SI NO EXISTE
5     if (!Files.exists(uploadPath)) {
6         Files.createDirectories(uploadPath);
7     }
8     // OBTENCION DE EXTENSION
9     if (originalFilename != null) {
10        int lastDot = originalFilename.lastIndexOf('.');
11        if (lastDot > -1 && lastDot < originalFilename.length() - 1) {
12            extension = originalFilename.substring(lastDot + 1);
13        }
14    }
15    // GUARDADO DEL ARCHIVO
16    Files.copy(archivo, filePath,
17        StandardCopyOption.REPLACE_EXISTING);
18    archivoRecurso.setTipoAdjunto("Archivo");
19    // OBTENCION DE TAMANO DEL ARCHIVO
20    try {
21        long tamaño = Files.size(filePath);
22        archivoRecurso.setPesoBytes(tamaño);
23    } catch (IOException e) {
24        archivoRecurso.setPesoBytes(0L);
25    }

```

```

25         // ASIGNACION DE EXTENSION
26         if (extension != null) {
27             archivoRecurso.setExtension(extension);
28         }
29
30     } else {
31         // CASO ENLACE (SIN ARCHIVO)
32         archivoRecurso.setTipoAdjunto("Enlace");
33
34         if (recurso.getDescripcion() != null &&
35             !recurso.getDescripcion().isBlank()) {
36             archivoRecurso.setUrlArchivo(recurso.getDescripcion());
37         } else {
38             archivoRecurso.setUrlArchivo(null);
39         }
40
41     // MANEJO DE ERROR GENERAL
42     try {
43         archivoServicio.crearArchivo(archivoRecurso);
44     } catch (IOException e) {
45         throw new ErrorProcesamientoException("Error_al_guardar_el_
46             archivo");

```

Validación interna para creación de nota adhesiva

ID	Condición evaluada	Acción realizada	Resultado esperado	Resultado obtenido	Estado
CP-28	IdMural o IdMiembroCreador nulo	Intentar crear nota sin IDs requeridos	El sistema lanza excepción de validación	Mensaje “IdMural e IdMiembroCreador son requeridos”	Aprobado
CP-29	Mensaje nulo o vacío	Intentar crear nota sin contenido	El sistema lanza excepción de validación	Mensaje “Mensaje no puede estar vacío”	Aprobado
CP-30	TipoVisibilidad nulo	Intentar crear nota sin tipo de visibilidad	El sistema lanza excepción de validación	Mensaje “TipoVisibilidad es requerido”	Aprobado
CP-31	Datos válidos	Crear nota con todos los campos correctos	El sistema permite continuar el flujo sin errores	Nota creada correctamente	Aprobado

Cuadro 2.10: Casos de prueba de caja blanca para validaciones internas en la creación de notas adhesivas

Listing 2.10: Validaciones internas para la creación de notas adhesivas

```

1
2 if (dto.getIdMural() == null || dto.getIdMiembroCreador() == null) {
3     throw new ValidationException("IdMural_e_IdMiembroCreador_son_
4         requeridos");
5 }
6
7 if (dto.getMensaje() == null || dto.getMensaje().trim().isEmpty()) {
8     throw new ValidationException("Mensaje_no_puede_estar_vacio");
9 }
10
11 if (dto.getTipoVisibilidad() == null) {
12     throw new ValidationException("TipoVisibilidad_es_requerido");
13 }

```

Comentar en Publicación del Mural

ID	Campo/Condición	Acción realizada	Resultado esperado	Resultado obtenido	Estado
CP-32	Comentario válido	Ejecutar comentar publicación con texto ingresado	El sistema guarda y muestra el comentario en el mural	Comentario publicado correctamente	Aprobado
CP-33	Comentario vacío	Intentar comentar publicación sin texto	El sistema desactiva el botón y muestra advertencia de campo obligatorio	Botón gris, solo se activa al ingresar al menos un carácter	Aprobado

Cuadro 2.11: Casos de prueba para comentar en publicaciones del mural

Responder a Comentario

ID	Campo ó Condición	Acción realizada	Resultado esperado	Resultado obtenido	Estado
CP-34	Respuesta válida	Ejecutar responder comentario con texto ingresado	El sistema guarda y muestra la respuesta debajo del comentario	Respuesta publicada correctamente	Aprobado
CP-35	Respuesta vacía	Intentar responder comentario sin texto	El sistema desactiva el botón y muestra advertencia de campo obligatorio	Botón gris, solo se activa al ingresar al menos un carácter	Aprobado
CP-36	Longitud máxima	Ejecutar responder comentario con texto que excede límite permitido	El sistema impide la acción y muestra advertencia de longitud máxima	Mensaje de validación mostrado y respuesta no publicada	Aprobado

Cuadro 2.12: Casos de prueba para responder a comentarios en publicaciones del mural

2.1.5. Chat de Grupo

Validación interna para envío de mensajes en chat

ID	Condición evaluada	Acción realizada	Resultado esperado	Resultado obtenido	Estado
CP-37	Mensaje con texto válido	Enviar mensaje solo con texto	El sistema permite el envío del mensaje	Mensaje enviado correctamente	Aprobado
CP-38	Mensaje de texto vacío	Intentar enviar mensaje sin contenido	El sistema lanza excepción de validación	Mensaje “El mensaje de texto debe contener texto no vacío”	Aprobado
CP-39	Archivo + texto simultáneo	Enviar mensaje con archivo y texto	El sistema lanza excepción	Mensaje “No se permite enviar ambos”	Aprobado
CP-40	Archivo nulo o vacío	Enviar mensaje tipo archivo sin archivo válido	El sistema lanza excepción	Mensaje “Debe contener un archivo válido”	Aprobado
CP-41	Tipo de adjunto inválido	Enviar archivo con tipo no permitido	El sistema lanza excepción	Mensaje “Tipo de adjunto no soportado”	Aprobado
CP-42	Metadatos incompletos	Enviar archivo sin mimeType o tipoAdjunto	El sistema lanza excepción	Mensaje de validación mostrado	Aprobado

Cuadro 2.13: Casos de prueba de caja blanca para validaciones internas en envío de mensajes

Listing 2.11: Lógica interna para envío de mensajes de texto y archivo

```
1 // DECISION PRINCIPAL: MENSAJE TIPO ARCHIVO O TEXTO
2 if (dto instanceof EnviarMensajeArchivoDTO) {
3     boolean tieneArchivo = archivoDTO.getArchivo() != null &&
4         archivoDTO.getArchivo().length > 0;
5     boolean tieneTexto = archivoDTO.getContenidoTexto() != null &&
6         !archivoDTO.getContenidoTexto().isBlank();
7     // NO PERMITIR TEXTO Y ARCHIVO AL MISMO TIEMPO
8     if (tieneArchivo && tieneTexto) {
9         throw new ValidacionException("No se permite enviar mensaje_
10             con_texto_y_archivo_al_mismo_tiempo");
11     }
12     // ARCHIVO OBLIGATORIO
13     if (!tieneArchivo) {
14         throw new ValidacionException("El mensaje_tipo_archivo_debe_
15             contener_un_archivo_valido");
16     }
17     // VALIDACION DE METADATOS DEL ARCHIVO
```

```

15     if (adjDTO == null || adjDTO.getMimeType() == null ||
16         adjDTO.getTipoAdjunto() == null) {
17         throw new ValidacionException("Debe_especificar_tipoAdjunto_y_
18             mimeType");
19     }
20     // VALIDACION DE TIPOS PERMITIDOS
21     String tipo = adjDTO.getTipoAdjunto().toLowerCase();
22     if (!(tipo.equals("audio") || tipo.equals("imagen") ||
23         tipo.equals("pdf") || tipo.equals("archivo"))) {
24         throw new ValidacionException("Tipo_de_adjunto_no_
25             soportado");
26     }
27     } else {
28     // VALIDACION MENSAJE DE TEXTO
29     boolean tieneTexto = dto.getContenidoTexto() != null &&
30         !dto.getContenidoTexto().isBlank();
31     if (!tieneTexto) {
32         throw new ValidacionException("El_mensaje_de_texto_debe_
33             contener_texto_no_vacio");
34     }
35 }

```

Validación interna para actualización de chat grupal

ID	Condición evaluada	Acción realizada	Resultado esperado	Resultado obtenido	Estado
CP-41	Chat inexistente	Intentar actualizar chat con ID inválido	El sistema lanza excepción	Mensaje “No existe el chat a actualizar”	Aprobado
CP-42	Nombre válido	Actualizar chat con nuevo nombre	El sistema actualiza el nombre	Nombre actualizado correctamente	Aprobado
CP-43	Nombre vacío o nulo	Intentar actualizar con nombre inválido	El sistema no modifica el nombre	Nombre se mantiene sin cambios	Aprobado
CP-44	Descripción válida	Actualizar descripción del chat	El sistema actualiza la descripción	Descripción actualizada correctamente	Aprobado
CP-45	Fondo válido	Actualizar fondo con imagen válida	El sistema reemplaza la imagen	Fondo actualizado correctamente	Aprobado
CP-46	Fondo incompleto	Enviar fondo sin nombreImagen o vacío	El sistema no actualiza el fondo	Fondo se mantiene sin cambios	Aprobado

Cuadro 2.14: Casos de prueba de caja blanca para actualización de chat grupal

Listing 2.12: Lógica interna para actualización de chat grupal

```

1 // VALIDACION DE EXISTENCIA
2 if (chatExistente == null) {
3     throw new EntidadNoEncontradaException("No_existe_el_chat_a_
4         actualizar");
5 }
6 // ACTUALIZACION DE NOMBRE
7 if (dto.getNombreChat() != null && !dto.getNombreChat().isBlank()) {
8     chatExistente.setNombreChat(dto.getNombreChat());
9 }
10 // ACTUALIZACION DE DESCRIPCION
11 if (dto.getDescripcion() != null) {
12     chatExistente.setDescripcion(dto.getDescripcion());
13 }
14 // ACTUALIZACION DE FONDO
15 if (dto.getFondoChat() != null && dto.getFondoChat().length > 0 &&
16     dto.getNombreImagen() != null) {
17     eliminarImagenAnterior(chatExistente.getFondoChat());
18     String nombreFinal = generarNombreImagen(dto.getNombreImagen());
19     String ruta = guardarImagen(dto.getFondoChat(), nombreFinal);
20     chatExistente.setFondoChat(ruta);
21 }

```

2.1.6. Perfil y Reseñas

Validación interna para creación de reseñas de grupo

ID	Condición evaluada	Acción realizada	Resultado esperado	Resultado obtenido	Estado
CP-53	Usuario no miembro	Intentar reseñar grupo sin pertenecer	El sistema lanza excepción	“Debes ser miembro del grupo”	Aprobado
CP-54	Reseña duplicada	Intentar reseñar grupo nuevamente	El sistema lanza excepción	“Ya has reseñado este grupo”	Aprobado
CP-55	Reseña válida	Crear reseña correctamente	El sistema guarda la reseña	Reseña registrada correctamente	Aprobado

Cuadro 2.15: Casos de prueba de caja blanca para reseñas de grupo

Listing 2.13: Lógica interna para creación de reseñas de grupo

```

1 // VALIDACION: USUARIO DEBE SER MIEMBRO
2 Integer emisorMiembroGrupoId = resenaRepositorio
3 .obtenerMiembroGrupoId(...)
4 .orElseThrow(() -> new EntidadNoEncontradaException());
5

```

```

6 // VALIDAR RESENA DUPLICADA
7 if (resenaRepositorio.existeResenaDeGrupo(...)) {
8     throw new ConflictoException("Ya_has_resenado_este_grupo_
9         anteriormente");
10 }

```

Validación interna para creación de reseñas entre miembros

ID	Condición evaluada	Acción realizada	Resultado esperado	Resultado obtenido	Estado
CP-47	Emisor no pertenece al grupo	Intentar crear reseña sin ser miembro	El sistema lanza excepción	“El emisor no es miembro del grupo”	Aprobado
CP-48	Auto-reseña	Intentar reseñarse a sí mismo	El sistema lanza excepción	“No puedes reseñarte a ti mismo”	Aprobado
CP-49	Reseña duplicada	Crear reseña existente	El sistema lanza excepción	“Ya existe una reseña”	Aprobado
CP-50	Líder → Miembro	Crear reseña con emisor líder	Tipo reseña correcto	Lider_a_Miembro asignado	Aprobado
CP-51	Miembro → Líder	Crear reseña a líder	Tipo reseña correcto	Miembro_a_Lider asignado	Aprobado
CP-52	Miembro → Miembro	Crear reseña entre miembros	Tipo reseña correcto	Miembro_a_Miembro asignado	Aprobado

Cuadro 2.16: Casos de prueba de caja blanca para reseñas entre miembros

Listing 2.14: Lógica interna para creación de reseñas entre miembros

```

1
2 // VALIDACION: EL EMISOR DEBE SER MIEMBRO DEL GRUPO
3 Integer emisorMiembroGrupoId = resenaRepositorio
4 .obtenerMiembroGrupoId(...)
5 .orElseThrow(() -> new EntidadNoEncontradaException());
6
7 // NO PERMITIR AUTO-RESENA
8 if (emisorMiembroGrupoId.equals(receptorMiembroGrupoId)) {
9     throw new ValidacionException("No_puedes_reseñarte_a_tu_mismo");
10 }
11 // VALIDAR RESENA DUPLICADA
12 if (resenaRepositorio.existeResenaEntreMiembros(...)) {
13     throw new ConflictoException("Ya_existe_una_resena");
14 }
15 // OBTENER ROLES
16 String rolEmisor = ...
17 String rolReceptor = ...
18 // DETERMINAR TIPO DE RESENA

```

```

19 if (emisorEsLider && !receptorEsLider) {
20     return TipoResena.Lider_a_Miembro;
21 } else if (!emisorEsLider && receptorEsLider) {
22     return TipoResena.Miembro_a_Lider;
23 } else {
24     return TipoResena.Miembro_a_Miembro;
25 }

```

2.1.7. Funciones especiales

Validación interna para activación de alerta de pánico

ID	Condición evaluada	Acción realizada	Resultado esperado	Resultado obtenido	Estado
CP-56	Miembro inexistente	Activar alerta con miembro inválido	El sistema lanza excepción	“Miembro no encontrado”	Aprobado
CP-57	Miembro no pertenece al grupo	Activar alerta con grupo incorrecto	El sistema lanza excepción	“No pertenece al grupo”	Aprobado
CP-58	Miembro sin permisos (inactivo)	Activar alerta sin estado válido	El sistema lanza excepción	“No puede usar el botón”	Aprobado
CP-59	Alerta activa existente	Activar alerta duplicada	El sistema lanza excepción	“Ya tienes una alerta activa”	Aprobado
CP-60	Activación válida	Activar alerta correctamente	El sistema crea alerta y notifica	Alerta creada y notificaciones enviadas	Aprobado
CP-61	Notificación a receptores	Activar alerta con múltiples miembros	El sistema notifica a todos excepto emisor	Notificaciones enviadas correctamente	Aprobado
CP-62	Chat existente	Activar alerta con chat disponible	El sistema envía mensaje al chat	Mensaje visible en chat	Aprobado
CP-63	Chat inexistente	Activar alerta sin chat	El sistema no envía mensaje pero continúa	Alerta creada sin error	Aprobado

Cuadro 2.17: Casos de prueba de caja blanca para la activación del botón de pánico

Listing 2.15: Lógica interna para activación de alerta de pánico

```

1
2 // VALIDACION: MIEMBRO EXISTE
3 MiembroGrupo generador = miembroGrupoRepositorio.buscarPorId(miembroId)
4 .orElseThrow(() -> new EntidadNoEncontradaException());
5
6 // VALIDAR QUE PERTENECE AL GRUPO
7 if (!generador.getGrupoEstudioId().equals(grupoId)) {
8     throw new OperacionNoPermitidaException();

```

```

9  }
10 // VALIDAR ESTADO ACTIVO
11 if (!generador.puedeEnviarMensajes()) {
12     throw new OperacionNoPermitidaException();
13 }
14 // VALIDAR ALERTA ACTIVA EXISTENTE
15 if (repositorio.tieneAlertaActiva(grupoId, miembroId)) {
16     throw new ConflictoException();
17 }
18 // CREAR ALERTA Y NOTIFICAR
19 for (MiembroGrupo receptor : receptores) {
20     if (!receptor.getMembrosGrupoId().equals(miembroId)) {
21         repositorio.guardarDestinatario(...);
22         notificacionServicio.crearNotificacion(...);
23     }
24 }
25 // VALIDAR EXISTENCIA DE CHAT
26 if (chatGrupalId != null) {
27     chatGrupalServicio.enviarMensaje(...);
28 }

```

Expiración automática de sesiones en el BDP

ID	Condición	Acción	Resultado esperado	Resultado obtenido	Estado
CP-80	Sesión expirada	Ejecutar tarea programada	Estado cambia a EXPIRADO	Sesión expirada correctamente	Aprobado
CP-81	Participantes sin confirmar	Expirar sesión	Estado cambia a NO_LLEGO	Participantes actualizados	Aprobado

Listing 2.16: Lógica interna de matchmaking para formación de sesión

```

1  private void intentarMatchmaking(String idMat, LocalDateTime ahora) {
2
3      List<BdpCola> candidatos = colaRepo.buscarTresParaMatch(idMat, ahora);
4
5      // VALIDACION: MINIMO DE PARTICIPANTES
6      if (candidatos.size() < PARTICIPANTES_POR_SESION) {
7          return;
8      }
9      // CREACION DE SESION
10     BdpSesion sesion = new BdpSesion();
11     sesion.setEstado(EstadoSesion.FORMANDO);
12     for (BdpCola candidato : candidatos) {
13         // CAMBIO DE ESTADO
14         candidato.setEstado(EstadoCola.MATCH_ENCONTRADO);
15         // NOTIFICACION
16         wsPublisher.publicar(
17             candidato.getIdUsr(),

```

```

17         candidato.getIdEsc(),
18         new BdpWebSocketDTO("MATCH_ENCONTRADO",
19                               sesion.getIdSesion(), null, null, null)
20     );
21 }

```

Lógica de emparejamiento (matchmaking)

ID	Condición	Acción	Resultado esperado	Resultado obtenido	Estado
CP-73	Menos de participantes requeridos	Ejecutar matchmaking	No se crea sesión	No ocurre match	Aprobado
CP-74	Participantes suficientes	Ejecutar matchmaking	Se crea sesión	Sesión creada correctamente	Aprobado
CP-75	Participantes válidos	Asignar sesión	Estado cambia a MATCH_ENCONTRADO	Usuarios notificados	Aprobado

Listing 2.17: Validaciones y transición de estados al confirmar llegada

```

1
2 public BdpSessionResult confirmarLlegada(Integer idSesion, String idUsr,
3     String idEsc) { // VALIDAR SESION ACTIVA
4     if (sesion.getEstado() == EstadoSesion.COMPLETADO ||
5         sesion.getEstado() == EstadoSesion.EXPIRADO) {
6         throw new ConflictoException("La_sesion_ya_no_esta_activa");
7     }
8
9     // VALIDAR PARTICIPANTE
10    BdpSessionParticipante participante = participanteRepo
11        .obtenerPorSesionYUsuario(idSesion, idUsr, idEsc)
12        .orElseThrow(() -> new OperacionNoPermitidaException());
13    // VALIDAR ESTADO PREVIO
14    if (participante.getEstado() == EstadoParticipante.CONFIRMADO) {
15        throw new ConflictoException("Ya_confirmaste_tu_llegada");
16    }
17    participante.setEstado(EstadoParticipante.CONFIRMADO);
18    return obtenerSesion(idSesion);
19 }

```

Listing 2.18: Proceso automático de expiración de sesiones

```

1
2 @Scheduled(cron = "0_*_*_*_*_*")
3 public void verificarSesionesExpiradas() {
4     List<BdpSession> sesionesExpiradas =
5         sesionRepo.buscarSesionesActivasExpiradas(ahora);
6     for (BdpSession sesion : sesionesExpiradas) {
7         // CAMBIO DE ESTADO
8     }
9 }

```

```
7      sesion.setEstado(EstadoSesion.EXPIRADO);
8      // ACTUALIZAR PARTICIPANTES
9      for (BdpSesionParticipante participante : participantes) {
10         if (participante.getEstado() !=
11             EstadoParticipante.CONFIRMADO) {
12             participante.setEstado(EstadoParticipante.NO_LLEGO);
13         }
14     }
15 }
```

Capítulo 3

Validaciones de algunos requerimientos No funcionales

MOGE_NF01: Tiempo de redirección del sistema

- **Descripción:** El sistema debe garantizar tiempos de redirección eficientes entre módulos internos, con el fin de proporcionar una experiencia de usuario fluida.
- **Criterios de aceptación:**
 - El tiempo total de redirección no debe superar los **2 segundos** en condiciones normales de operación.
 - El tiempo de respuesta debe mantenerse estable en al menos el **95 % de las solicitudes**.
 - El sistema debe soportar hasta **100 usuarios concurrentes** sin degradación significativa del rendimiento.

MOGE_NF02: Desempeño de comunicaciones en tiempo real (WebSocket)

- **Descripción:** El sistema debe garantizar el desempeño adecuado de las comunicaciones en tiempo real mediante el uso de WebSocket, asegurando baja latencia en el envío y recepción de mensajes.
- **Criterios de aceptación:**
 - El sistema debe soportar al menos **100 conexiones simultáneas activas**.
 - El tiempo de entrega de mensajes no debe superar los **300 ms** en condiciones normales.
 - El sistema debe mantener la estabilidad de las conexiones sin pérdidas de sesión en al menos el **95 % de los casos**.

MOGE_NF03: Optimización de carga de recursos multimedia

- **Descripción:** El sistema debe optimizar la carga y almacenamiento de recursos multimedia (imágenes, archivos y otros contenidos), con el fin de reducir tiempos de respuesta y consumo de ancho de banda.

■ **Criterios de aceptación:**

- Los archivos multimedia deben almacenarse de forma estructurada por categorías.
- El tiempo de carga de recursos multimedia no debe superar los **2 segundos** en condiciones normales.
- El sistema debe soportar carga concurrente de archivos sin afectar significativamente el rendimiento general.

MOGE_NF04: Optimización de consultas a base de datos

- **Descripción:** El sistema debe optimizar las consultas a la base de datos para garantizar eficiencia en el acceso a la información y evitar sobrecarga en el servidor.

■ **Criterios de aceptación:**

- El sistema debe evitar el problema de consultas tipo N+1 mediante el uso de estrategias como *JOIN FETCH* en JPQL.
- Las consultas deben aprovechar índices definidos en la base de datos.
- El tiempo de respuesta de las consultas críticas no debe superar los **500 ms**.
- Se debe garantizar la integridad referencial conforme al esquema de la base de datos institucional.

3.1. Pruebas de rendimiento con JMeter

Con el objetivo de validar los requerimientos no funcionales definidos para el sistema, se realizaron pruebas de carga utilizando la herramienta *Apache JMeter*. Estas pruebas permiten medir el comportamiento del sistema bajo condiciones de concurrencia, evaluando métricas como tiempo de respuesta, throughput y porcentaje de errores.

3.1.1. Configuración de la prueba

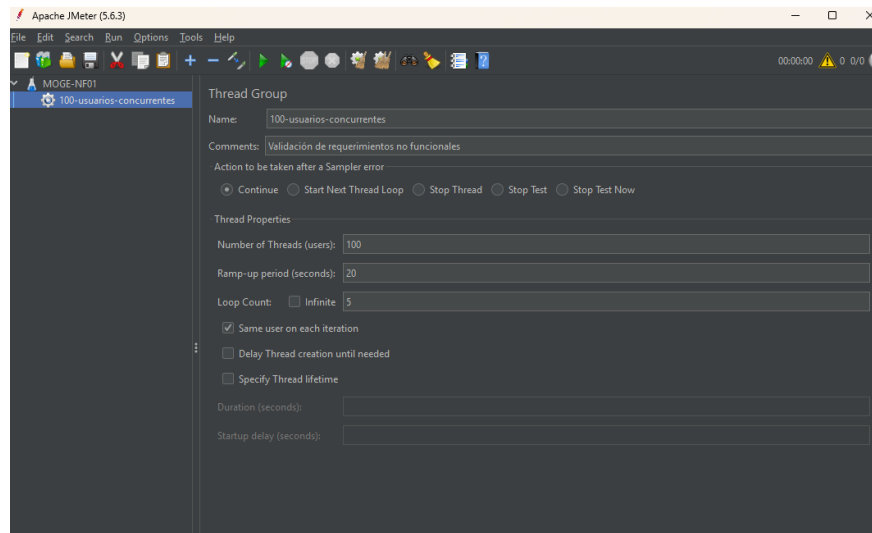


Figura 3.1: Configuración del Thread Group en JMeter

Los parámetros utilizados fueron los siguientes:

- Número de usuarios concurrentes: **100**
- Ramp-up period: **20 segundos**
- Número de iteraciones por usuario: **5**
- Total de solicitudes generadas: **500**

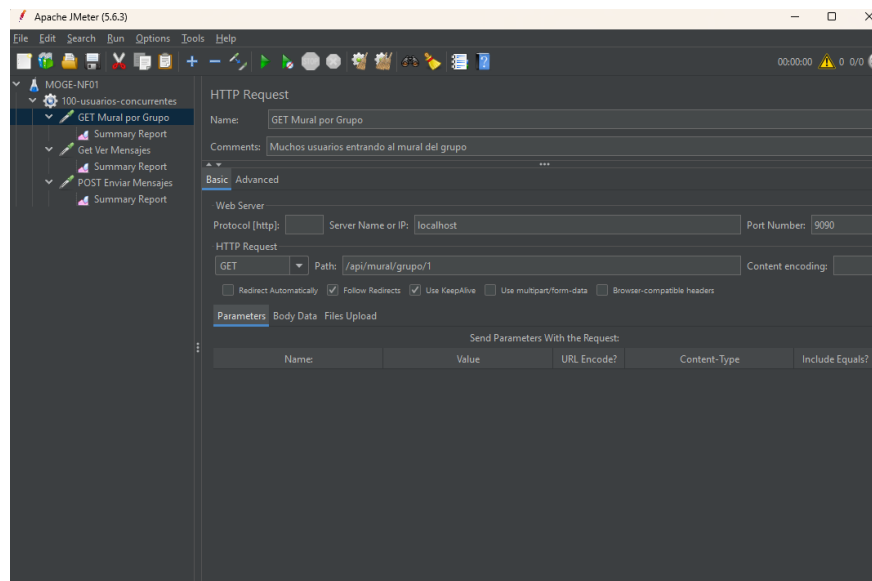


Figura 3.2: Reporte de resultados - Summary Report

3.1.2. Resultados obtenidos

Métrica	Valor
Número de solicitudes	500
Tiempo promedio	14 ms
Tiempo mínimo	3 ms
Tiempo máximo	993 ms
Desviación estándar	77.30 ms
Porcentaje de error	0.0 %
Throughput	25.45 req/s

Cuadro 3.1: Resultados de la prueba de carga inicial

3.1.3. Análisis del requerimiento MOGE_NF01

El requerimiento establece que el tiempo de respuesta no debe superar los **2 segundos** bajo condiciones normales.

- Tiempo promedio obtenido: **14 ms**
- Tiempo máximo registrado: **993 ms**
- Porcentaje de error: **0.0 %**

Resultado: El sistema cumple satisfactoriamente con el requerimiento MOGE_NF01, ya que los tiempos de respuesta se mantienen muy por debajo del límite establecido.

Observación: El valor máximo cercano a 1 segundo puede atribuirse a condiciones de carga inicial (*cold start*) o concurrencia simultánea en el arranque.

—

3.1.4. Prueba de carga concurrente multi-módulo

Con el fin de evaluar el comportamiento del sistema en un escenario más realista, se ejecutó una prueba simultánea sobre múltiples endpoints, incluyendo operaciones de lectura y escritura.

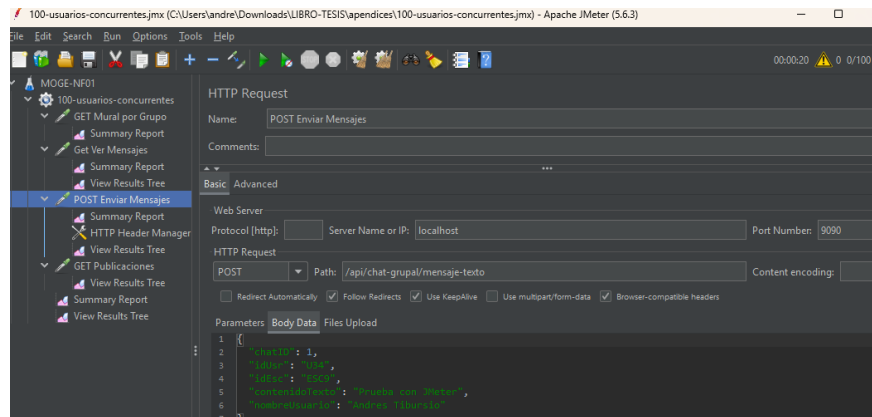


Figura 3.3: Configuración de prueba concurrente multi-endpoint

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
POST Enviar Mensajes	2000	10	7	26	2.20	0.00%	25.1/sec	10.45	21.69	426.2
GET Mural por Grupo	500	11	8	61	3.38	0.00%	25.1/sec	10.45	21.69	426.2
GET Ver Mensajes	500	6	5	25	2.01	0.00%	25.1/sec	10.45	21.69	426.2
GET Publicaciones	500	2	1	48	2.60	0.00%	25.1/sec	10.45	21.69	426.2
TOTAL	2000	7	1	61	4.27	0.00%	100.2/sec	32.60	85.16	1314.4

Figura 3.4: Resultados de la prueba concurrente

3.1.5. Parámetros de la prueba

- Usuarios concurrentes: **100**
- Total de solicitudes: **2000**
- Endpoints evaluados: 4
- Duración aproximada: **80 segundos**

3.1.6. Resultados por requerimiento no funcional

Requerimiento	Endpoint	Tiempo Promedio	Criterio	Cumple
MOGE_NF01	GET /api/mural/grupo/1	2 ms	¡2000 ms	Sí
MOGE_NF03	GET /api/mural/publicaciones	6 ms	¡2000 ms	Sí
MOGE_NF04	GET /api/chat/mensajes	11 ms	¡500 ms	Sí
Chat	POST /api/chat-grupal/mensaje-texto	10 ms	–	Sí

Cuadro 3.2: Validación de requerimientos no funcionales bajo carga

3.1.7. Métricas globales

- Tiempo promedio total: **7 ms**
- Tiempo mínimo: **1 ms**
- Tiempo máximo: **61 ms**
- Porcentaje de error: **0.0 %**
- Throughput total: **100.23 req/s**

3.1.8. Análisis general

Los resultados obtenidos permiten concluir que:

- El sistema soporta correctamente **100 usuarios concurrentes**.
- No se presentaron errores durante la ejecución de las pruebas.
- Los tiempos de respuesta se mantienen significativamente por debajo de los límites definidos.
- El throughput se mantiene estable, indicando buena capacidad de procesamiento.

3.1.9. Conclusión

Las pruebas realizadas con JMeter demuestran que el sistema cumple con los requerimientos no funcionales establecidos, garantizando un comportamiento eficiente, estable y escalable bajo condiciones de carga concurrente.