**Casos Más Comunes en las Pruebas de Integración**

Las pruebas de integración evalúan la comunicación entre módulos del software. Los casos más comunes se enfocan en verificar la **interacción entre componentes**, la **transferencia de datos** y la **respuesta ante fallos**.

**1 Casos de Integración Común entre Frontend y Backend**

**Caso 1: Autenticación de Usuario**

**Objetivo:** Verificar que el usuario puede autenticarse correctamente.

**Datos de Entrada:**

* Usuario: carlos
* Contraseña: 12345

**Pasos:**

1. El usuario ingresa las credenciales en el frontend.
2. El frontend envía la solicitud al backend.
3. El backend valida los datos y responde con un token de sesión.

**Resultado Esperado:**

* Credenciales correctas → Se recibe un **token válido**.
* Credenciales incorrectas → Se muestra un **mensaje de error**.

**Caso 2: Recuperación de Datos desde una API**

**Objetivo:** Verificar que el frontend obtiene correctamente la información desde la API.

**Ejemplo:**

* API /productos devuelve una lista de productos.

**Pasos:**

1. El frontend realiza una solicitud GET a la API.
2. La API responde con un JSON que contiene los productos.
3. El frontend muestra los datos correctamente en la pantalla.

**Resultado Esperado:**

* Si la API responde correctamente → Se muestran los productos.
* Si la API está caída → Se muestra un mensaje de error.

**3 Casos de Integración con Bases de Datos**

**Caso 3: Registro de Usuario en la Base de Datos**

**Objetivo:** Verificar que los datos del usuario se almacenan correctamente en la base de datos.

**Datos de Entrada:**

* Nombre: Juan Pérez
* Correo: juan@example.com

**Pasos:**

1. Enviar la solicitud POST con los datos del usuario.
2. El backend procesa y almacena la información en la base de datos.
3. Realizar una consulta SQL para verificar que los datos se guardaron.

**Resultado Esperado:**

* Se debe crear un **nuevo registro** en la base de datos.

**Caso 4: Consulta y Modificación de Datos**

**Objetivo:** Verificar que se pueden recuperar y actualizar datos correctamente.

**Pasos:**

1. Consultar un usuario en la base de datos.
2. Modificar su información y guardar los cambios.
3. Volver a consultar los datos para confirmar la actualización.

**Resultado Esperado:**

* La base de datos debe reflejar los cambios realizados.

**5 Casos de Integración con Microservicios**

**Caso 5: Procesamiento de un Pago**

**Objetivo:** Verificar que el sistema procesa correctamente un pago con tarjeta de crédito.

**Datos de Entrada:**

* Tarjeta: 4111 1111 1111 1111
* Monto: $100 USD

**Pasos:**

1. El usuario ingresa los datos de pago.
2. El frontend envía los datos al microservicio de pagos.
3. El microservicio de pagos contacta al banco.
4. El banco responde con una confirmación.

**Resultado Esperado:**

* Pago aprobado → Se muestra un mensaje de éxito.
* Pago rechazado → Se muestra un mensaje de error.

**Caso 6: Envío de Notificaciones**

**Objetivo:** Verificar que el usuario recibe una notificación tras completar una acción.

**Pasos:**

1. Un usuario realiza una compra.
2. El sistema envía la confirmación al microservicio de notificaciones.
3. El microservicio genera un correo y lo envía al usuario.

**Resultado Esperado:**

* El usuario recibe un **correo de confirmación**.

**7 Casos de Integración con APIs Externas**

**Caso 7: Validación de un Usuario con un Servicio ExternoObjetivo:** Verificar que el sistema puede validar datos con un servicio externo (Ej: Google, Facebook).

**Pasos:**

1. El usuario inicia sesión con Google.
2. El sistema redirige la autenticación a Google.
3. Google responde con los datos del usuario.
4. El sistema registra al usuario si es nuevo.

**Resultado Esperado:**

* Si Google autoriza → El usuario accede a la aplicación.
* Si Google rechaza → Se muestra un error de autenticación.

**8 Casos de Integración con Sistemas de Almacenamiento**

**Caso 8: Carga y Descarga de Archivos**

**Objetivo:** Verificar que los archivos se suben y descargan correctamente desde un servidor.

**Pasos:**

1. El usuario selecciona un archivo para subir.
2. El sistema envía el archivo al servidor.
3. Se genera un enlace para descargarlo posteriormente.

**Resultado Esperado:**

* El archivo se almacena correctamente en el servidor.
* Se puede descargar sin errores.

**Conclusión**

Los casos de prueba de integración **aseguran que los módulos del sistema trabajan bien juntos**. Es crucial probar:  
✔ **Comunicación entre frontend y backend**  
✔ **Interacción con bases de datos**  
✔ **Interoperabilidad con microservicios**  
✔ **Uso de APIs externas**  
✔ **Almacenamiento y recuperación de datos**