

PRO402-9527-225081-ONL-TALLER DE TESTING Y CALIDAD DE SOFTWARE



MÓDULO: PRO402-9527-225081-ONL-TALLER DE TESTING Y CALIDAD DE SOFTWARE

SEMANA: Semana 3

Docente: Cynthia Torres.

Estudiantes: Karla Pesce R., Sergio Molina V., Jaime Codoceo C.

Introducción.....	3
Objetivo de las pruebas	3
Alcance.....	4
Consideraciones de la prueba.....	4
Ciclos (back end).....	5
Juego de datos (back end)	6
Ejecución de pruebas (front end).....	7
Caso específico: Manejo de duplicados con sobreescritura.....	9
Condiciones y Resultados de Pruebas Unitarias	9
Ciclos (caso back end)	10
Juego de datos (caso back end)	11
Respaldo prueba	12
Prueba Acceso (caso front end).....	13
Condiciones de la prueba (caso front end)	13
Ejecución de la prueba	14
Respaldo resultados	15
Observaciones.....	15
Conclusión	16
Reflexiones finales	16
Links de acceso.....	17

Introducción

Este informe documenta la verificación y validación del formulario de “Pacientes”, desarrollado como Web App con Google Apps Script (back end) y HTML/JS (front end), usando Google Sheets como repositorio de datos. El objetivo fue comprobar el cumplimiento de la Actividad Sumativa: presencia y validación de 10 campos obligatorios, control de duplicados por RUT con confirmación de sobrescritura, búsqueda por apellido con coincidencia parcial y funcionamiento correcto de los botones Guardar, Limpiar y Cerrar. Las pruebas se ejecutaron con enfoque de caja negra (es decir, que no revisamos el código interno ni la implementación, sino que probamos el comportamiento externo a partir de los requisitos), validaciones en clientes (en este caso se pidió a compañeros o amigos que completaran el formulario) y revalidaciones en servidor, registrando evidencia con capturas y resultados en la hoja “Pacientes” (trazabilidad mediante timestamp con “*createdAt*” / “*updatedAt*”)

Objetivo de las pruebas

Verificar y Validar (V&V) que el formulario cumple con los requerimientos funcionales y no funcionales básicos, con evidencia trazable (Capturas de pantalla), conforme a lo requerido.

Alcance

- Front-End (FE): validaciones de campos, funcionamiento de botones, búsqueda por apellido, flujos de éxito y error.
- Back-End (BE): creación/actualización en Google Sheets, detección de duplicados por rut, re-validaciones mínimas (email, fecha).
- Datos: hoja Pacientes con encabezados estándar y timestamp.

Consideraciones de la prueba

Las pruebas se realizaron en el navegador Google Chrome, en un equipo con Windows 11, accediendo a la aplicación desplegada públicamente.

Se aplicaron técnicas de caja negra, evaluando el comportamiento del sistema frente a distintos tipos de entradas, sin revisar el código interno.

Los resultados fueron comparados con los valores esperados y se respaldaron mediante capturas de pantalla asociadas a cada caso.

Ciclos (back end)

Se realizaron pruebas generales en el back end enfocadas en la creación y actualización de registros en la hoja de cálculo.

La siguiente tabla resume los principales ciclos de prueba:

Ciclo	Objetivo	Resultado Esperado	Resultado Obtenido
BE-1	Insertar nuevo paciente.	Fila creada con <i>createdAt/updatedAt</i> (timestamp)	
BE-2	Duplicado por rut, sin sobrescribir (overwrite)	Respuesta "EXISTS" y no cambia la fila	
BE-3	Duplicado con overwrite = true	Fila actualizada, <i>updatedAt</i> cambia	

Juego de datos (back end)



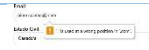




El siguiente conjunto de datos fue utilizado para validar reglas básicas de ingreso de información, incluyendo duplicados de RUT, correos electrónicos válidos/ inválidos y fechas en diferentes formatos.







Clave	Descripción	Resultado Obtenido
Rut único	12.345.678-5	
Rut duplicado	12.345.678-5	
emails válidos o inválidos	Válidos o inválidos	
Fechas	Válidas (ejemplo: 27/08/2025), inválidas (ie: 31/02/2025), fuera de rango (01/01/1800)	

Ejecución de pruebas (front end)

En el front end se verificaron validaciones de campos, funcionamiento de botones y flujos principales de uso, incluyendo la creación de registros, la búsqueda de pacientes y la gestión de duplicados.

La siguiente tabla muestra el detalle de las pruebas ejecutadas:

#	Petición / Origen	Ciclo / Subciclo	Descripción prueba	Objetivo	Resultado Esperado	Resultado Obtenido
1	Formulario	FE-valid	Campos vacíos - > guardar	Validar (requerido)	Bloquea envío, muestra aviso	
2	RUT	FE-RUT-1	Rut inválido (12345678-5)	Validar dígito verificador	Muestra “rut inválido” y no guarda.	
3	email	FE-mail-1	Email invalido	Validar formato	Bloquea por html5 / servidor	
4	Teléfono	FE-tel-1	<7 chars / con letras	Validar patrón	Rechaza por pattern (patrón)	
5	Fecha	FE-fecha-1	31/02/2025	Validar día/mes	Muestra “Fecha inválida dd/mm/yyyy”	
6	Fecha	FE-fecha-2	01/01/1800	Rango año	Rechaza por rango (>=1900)	
7	Éxito	FE-ok-1	Guardar con datos válidos	Crear registro	Muestra “Ficha creada correctamente”	

8	Duplicado	FE-dup-1	Guardar mismo rut	Confirmación	Muestra dialogo “¿Sobrescribir?”	
9	Overwrite	FE-dup-2	Confirmar sobrescribir	Actualizar	Muestra “Ficha actualizada correctamente”	
10	Buscar	FE-search-1	Buscar “Gat”	Búsqueda parcial	Tabla con coincidencias	
11	Cargar fila	FE-search-2	Click en resultado	UX edición (UX: user experience)	Carga datos al formulario.	
12	Botones	FE-UI-1 (UI: user interface)	limpiar	UX	Formulario vacío	
13	Botones	FE-UI-2	Cerrar	UX	Cierra	

Caso específico: Manejo de duplicados con sobreescritura

Dado que el manejo de duplicados por RUT es un requisito crítico de la actividad, se documentó este flujo de manera más exhaustiva, tanto en back end como en front end.

Condiciones y Resultados de Pruebas Unitarias

Aplicación: Pacientes

País: Chile

Módulo: Formulario Web (ingreso/búsqueda)

- Función Probada (back end): *guardarFicha* (payload, overwrite) con *overwrite = true*
- Objeto probado (front end): Flujo “duplicado por rut” -> diálogo de confirmación -> actualización de fila -> toast (mensaje) “Ficha actualizada correctamente”
- Responsable de la Prueba: Karla Pesce R.

Condiciones de la prueba

Manejo de duplicados por rut; si el registro existe, el sistema preguntará si desea sobrescribir, al confirmar debe actualizar y reflejar el cambio en *updatedAt* (y conservar *createdAt*).

Ciclos (caso back end)

La siguiente tabla muestra el ciclo de actualización de registros con overwrite:

Ciclo (id)	Ejecución de la prueba	Objetivo del ciclo	Resultado esperado	Resultado obtenido
BE-DUP-OK	<ul style="list-style-type: none">- Insertar paciente A.- Reintentar con mismo rut y datos distintos, llamando <code>guardarFicha(..., overwrite=true)</code>	Actualizar registro existente por RUT manteniendo <code>createdAt</code>	<code>Ok:true, action:'Updated'</code> ; fila mantiene <code>createdAt</code> original y cambia <code>updatedAt</code> ; columnas actualizadas con nuevos datos	<i>Respaldo de pruebas (Imagen 1, 2 y 3). Pag 12.</i>

Juego de datos (caso back end)

El juego de datos específico para esta prueba incluyó todos los campos de un paciente, simulando modificaciones en dirección, comentarios y otros datos.

Clave	Descripción
Rut	15054052-6
Nombres	"Jaime Andrés" -> cambio de prueba: "Jaime A."
Apellidos	"Codoceo Camus"
Dirección	" -> cambio "Av. Prueba 456"
Ciudad	"Santiago"
Teléfono	+56992235343
Email	"jaime.ccamus@gmail.com"
Fecha de Nacimiento	"18/01/1982"
Estado civil	"Soltero/a"
Comentarios	"Paciente prueba" -> cambio prueba: "Paciente prueba actualizado"

Criterio de aceptación: Con el mismo rut, la segunda ejecución debe actualizar los campos modificados y refrescar *updatedAt*, manteniendo *createdAt*.

Respaldo prueba

1.- Cambio de info para pruebas

← ↻ 🔒 https://script.google.com/macros/s/AKfycbzqXmOEaiR0wAN8rsuc9Z63A1eiy0xqzRYg1ZpQqQcqPQjelxjRZGdeynopDW-PsVWrlJA/exec

Un usuario de Google Apps Script creó esta aplicación [Denuncia](#)

Ficha de Pacientes

Formulario con validación de **RUT**, **email** y **fecha (dd/mm/yyyy)**. Incluye búsqueda por apellido y control de sobrescritura.

RUT	Nombres
15054052-6	Jaime A.
Apellidos	Dirección
Codoceo Camus	Av. Prueba 456
Ciudad	Teléfono
Santiago	+56992235343
Email	Fecha de nacimiento (dd/mm/yyyy)
jaime.ccamus@gmail.com	18/01/1982
Estado Civil	Comentarios
Soltero/a	Paciente prueba actualizado

[Guardar](#) [Limpiar](#) [Cerrar](#)

2.- Prueba ok:

Ficha de Pacientes

Formulario con validación de **RUT**, **email** y **fecha (dd/mm/yyyy)**. Incluye búsqueda por apellido y control de sobrescritura.

Ficha actualizada correctamente.

RUT	Nombres
15054052-6	Jaime A.
Apellidos	Dirección
Codoceo Camus	Av. Prueba 456
Ciudad	Teléfono
Santiago	+56992235343
Email	Fecha de nacimiento (dd/mm/yyyy)
jaime.ccamus@gmail.com	18/01/1982
Estado Civil	Comentarios
Soltero/a	Paciente prueba actualizado

3.- Revisión *updatedAt*

telefono	email	fechaNacimiento	estadoCivil	comentarios	createdAt	updatedAt
+56992235343	jaime.ccamus@	18/01/1982	Soltero/a	Paciente prueba actualizado	27/8/2025 21:25:09	2/9/2025 10:59:59

Prueba Acceso (caso front end)

Abrir el enlace público /exec de la Web App en el navegador (ideal: ventana de incógnito)

Condiciones de la prueba (caso front end)

- Ingresar a la url de la app (web app)
- Completar todos los campos y guardar (primer ok)
- Sin cambiar el rut, modificar campos (dirección, comentarios, etc) y Guardar.
- En el diálogo, confirmar sobrescritura

Ejecución de la prueba

En el front end se verificó el diálogo de confirmación de sobrescritura y la actualización de los datos.

#	Petición / Origen	Ciclo / Subciclo	Descripción prueba	Objetivo	Resultado esperado
1	Formulario (FE)	FE-DUP-2	Guardar con rut existente y datos distintos, confirmación de sobrescritura	Validar flujo de actualización	Aparece dialogo: “Ya existe este rut.¿Desea sobrescribirlo?, Luego indicar con un toast “Ficha actualizada correctamente”, luego actualizar dato en <i>updatedAt</i> .

Respaldo resultados

1.- confirmación sobreescritura:

Un usuario de Google Apps Script creó esta aplicación

Ficha de Pacientes

Formulario con validación de RUT, email y fecha (dd/mm/yy)

RUT

Nombres

Una página incrustada en n-u3pbfp62y4lf2mrac24pdry2tidl3e2t66bohi-0lu-script.googleusercontent.com dice

Ya existe este RUT. ¿Deseas sobrescribirlo?

Aceptar Cancelar

2.- Mensaje de actualización:

Ficha de Pacientes

Formulario con validación de RUT, email y fecha (dd/mm/yyyy). Incluye búsqueda por apellido y control de sobreescritura.

Ficha actualizada correctamente.

RUT

15054052-6

Nombres

Jaime A.

Apellidos

Codoceo Camus

Dirección

Av. Prueba 456

Ciudad

Santiago

Teléfono

+56992235343

Email

jaime.ccamus@gmail.com

Fecha de nacimiento (dd/mm/yyyy)

18/01/1982

Estado Civil

Soltero/a

Comentarios

Paciente prueba actualizado

3.- actualización en *updatedAt*:

telefono	email	fechaNacimiento	estadoCivil	comentarios	createdAt	updatedAt
+56992235343	jaime.ccamus@gmail.com	18/01/1982	Soltero/a	Paciente prueba actualizado	27/8/2025 21:25:09	2/9/2025 10:59:59

Observaciones

- El patrón del email en servidor es básico (contiene @); HTML5 ya refuerza en cliente.
- Teléfono tiene un patrón simple; se podría robustecer en caso de ser necesario.
- Despliegue del formulario: nos debemos asegurar de que el URL /macros/s/...exec sea visible para “Cualquiera” en la implementación.

Conclusión

El sistema cumple con los requisitos solicitados: valida los 10 campos, respeta dd/mm/yyyy, gestiona duplicados por RUT con confirmación, busca por apellido (ignorando mayúsculas y tildes) y permite cargar registros. Los botones operan conforme a la UX esperada. El despliegue público /exec está operativo. Como mejoras a futuro: se puede robustecer el filtro para email y teléfono, bitácora / auditoria y pruebas automatizadas básicas. Con la evidencia que recopilamos, el desarrollo está aprobado y listo para entrega al potencial cliente.

Reflexiones finales

El ejercicio permitió aplicar verificación y validación en todas las fases de desarrollo. Se utilizó la técnica de caja negra, evaluando el comportamiento externo del sistema frente a distintos datos de entrada. La evidencia recopilada mostró que los errores más frecuentes en proyectos suelen relacionarse con validaciones insuficientes, mensajes de error poco claros o falta de control sobre duplicados.

Incorporar V&V desde etapas tempranas evita estos problemas, reduce costos de corrección y entrega un producto confiable. En este caso, el formulario demostró ser funcional y usable, diferenciándose de un software no testeado que podría fallar en producción.

Links de acceso

Planilla Pacientes (Google Sheets): [acá](#)

Códigos Apps Script: (habilitar extensión Apps Script): [acá](#)

Drive Imágenes y Respaldo: [acá](#)

App Web: [acá](#)

GitHub: [acá](#)