

Séptimo Sprint: Tareas de Implementación e Integración de Proyecto

Facultad de Ingeniería

Iris Ayala, 23965
Gabriel Bran, 23590
David Domínguez, 23712
Luis Padilla, 23663
Anggie Quezada, 23643

Guatemala

2025

Video Sprint 7: https://youtu.be/Uno9Y_WwdBw

Product Backlog

Historias de usuario

1. Como gerente, quiero ver un dashboard con indicadores generales (movimiento de materiales, KPI's, detalle de proyectos y materiales, alertas, etc.) para tener una vista rápida del estado del sistema.
2. Como gerente, quiero ver mi inventario en tiempo real, para saber qué materiales están disponibles.
3. Como gerente, quiero registrar la entrada y salida de materiales, para mantener actualizado el inventario.
4. Como ingeniero, quiero asignar materiales a proyectos específicos, para evitar confusión en la distribución de recursos.
5. Como ingeniero, quiero recibir una alerta cuando un material tenga bajo stock, para reabastecerlo con tiempo.
6. Como secretaria, quiero recibir una alerta cuando un material tenga bajo stock, para cotizar materiales con tiempo con los proveedores.
7. Como ingeniero, quiero actualizar el estado de los proyectos en la plataforma.
8. Como gerente, quiero ver los avances de los proyectos en un solo reporte.
9. Como gerente, quiero tener un sitio web con información de la empresa, para atraer más clientes.
10. Como gerente, quiero registrar un nuevo proyecto.
11. Como usuario, quiero poder iniciar sesión en el sistema, para acceder a mis funcionalidades según mi rol.
12. Como gerente, quiero registrar nuevos usuarios en el sistema, para que puedan acceder según su rol.

Lista de tareas para construir el sistema

Infraestructura

Tarea	Descripción	Puntos	Prioridad
Configurar React	Configurar el entorno de desarrollo, instalar librerías, módulos	1	Alta
Configurar Express	Configurar el entorno de desarrollo, instalar librerías, módulos	1	Alta
Configurar la BD	Crear la base de datos	2	Alta
Configurar la conexión del backend con la BD	Configurar conexión desde backend a PostgreSQL	3	Alta
Configurar archivo .env	Guardar claves secretas, rutas, configuración JWT y variables sensibles .env.	1	Alta
Crear dockerfile para backend	Contenerizar backend con Node y Express para el servicio	2	Alta
Crear dockerfile para frontend	Contenerizar frontend con Node para el servicio	2	Alta
Crear docker-compose	Configurar docker-compose con servicio backend y PostgreSQL conectado	2	Alta

Como usuario, quiero poder iniciar sesión en el sistema, para acceder a mis funcionalidades según mi rol.

Tarea	Descripción	Puntos	Prioridad
Diseñar las pantallas	Realizar los diseños para maquetar la pantalla de login	1	Media
Crear formulario de login	Maquetar pantalla con campos de email y contraseña, y botón de enviar.	2	Alta
Validar campos	Validar que los campos no estén vacíos y tengan formato correcto.	3	Alta
Integrar API de login	Hacer petición al backend con email y contraseña.	2	Alta
Guardar token y datos en localStorage	Guardar token de sesión y usuario al iniciar sesión.	2	Alta
Crear endpoint /login	Crear endpoint POST para recibir email y contraseña.	2	Alta
Validar usuario en base de datos	Verificar existencia de usuario y comparar contraseña hasheada.	2	Alta
Generar token JWT	Crear y devolver un token JWT con la información del usuario.	2	Alta
Probar el endpoint	Probar el endpoint utilizando Postman	2	Alta
Crear tabla usuarios	Crear tabla con los campos necesarios	2	Alta

Como gerente, quiero registrar nuevos usuarios en el sistema, para que puedan acceder según su rol.

Tarea	Descripción	Puntos	Prioridad
Diseñar las pantallas	Realizar los diseños para maquetar la pantalla de login	1	Media
Crear formulario de register	Maquetar pantalla con campos de nombre, email y contraseña, y botón de enviar.	2	Alta
Validar campos	Validar que los campos no estén vacíos y tengan formato correcto.	3	Alta
Integrar API de register	Hacer petición al backend con los datos.	2	Alta
Crear endpoint /register	Crear endpoint POST para recibir nombre, email y contraseña.	2	Alta
Probar el endpoint	Probar el endpoint utilizando Postman	2	Alta
Insertar usuario en la tabla	Insertar al usuario en la bd de datos para tener su informacion.	2	Alta

Como gerente, quiero ver mi inventario en tiempo real, para saber qué materiales están disponibles.

Tarea	Descripción	Puntos	Prioridad
Diseñar las pantallas	Realizar los diseños para maquetar la pantalla de inventario	1	Media
Crear vista de inventario	Maquetar pantalla con tabla para listar materiales.	3	Alta
Hacer tabla interactiva	Mostrar materiales con botones para editar, borrar, asignar, estados, etc	5	Alta
Crear endpoint GET /materiales	Obtener todos los materiales desde la base de datos.	2	Alta
Crear tabla de materiales	Tabla con todos los campos necesarios como el MER	1	Alta

Como gerente, quiero registrar la entrada de materiales, para mantener actualizado el inventario.

Tarea	Descripción	Puntos	Prioridad
-------	-------------	--------	-----------

Crear formulario de entrada de material	Formulario para registrar nueva entrada de material	3	Alta
Validar datos de entrada	Verificar que campos no estén vacíos ni datos incorrectos.	3	Alta
Crear endpoint POST	Registrar entrada de material, actualizar stock y registrar movimiento.	3	Alta
Insertar material en la BD	Insertar la información del material en la BD	2	Alta

Como ingeniero/gerente, quiero asignar materiales a proyectos específicos, para evitar confusión en la distribución de recursos.

Tarea	Descripción	Puntos	Prioridad
Crear formulario de asignación de materiales	Seleccionar proyecto y materiales con cantidad a asignar.	5	Alta
Validar stock disponible antes de asignar	Mostrar alerta si el stock es insuficiente.	5	Alta
Crear endpoint POST	Registrar asignación y disminuir stock correspondiente.	5	Alta
Validar stock en backend	No permitir asignar más de lo disponible.	2	Alta
Insertar la asignación en la BD	Registrar proyecto, material, cantidad asignada y fecha.	2	Alta

Como ingeniero/secretaria/gerente, quiero recibir una alerta cuando un material tenga bajo stock, para reabastecerlo con tiempo.

Tarea	Descripción	Puntos	Prioridad
Mostrar notificación visual	Mostrar alerta en pantalla de inventario o dashboard.	5	Alta
Crear lógica de verificación de stock mínimo	Al modificar stock, verificar si se alcanza stock mínimo.	5	Alta
Enviar alerta al frontend	Devolver notificación al usuario correspondiente	3	Alta

Como gerente, quiero registrar un nuevo proyecto.

Tarea	Descripción	Puntos	Prioridad
Crear formulario de registro de proyecto	Formulario con los campos necesarios	3	Alta
Validar campos	Evitar campos vacíos y datos inválidos.	3	Alta
Crear endpoint POST /proyectos	Insertar nuevo proyecto en la base de datos.	3	Alta
Crear tabla proyectos	Tabla con los campos necesarios	2	Alta

Como ingeniero/gerente, quiero un dashboard para ver los proyectos, quiero actualizar el estado de los proyectos en la plataforma, para monitorear el progreso.

Tarea	Descripción	Puntos	Prioridad
Crear vista de dashboard de proyectos	Visualización general de todos los proyectos con filtros por estado.	5	Alta
Mostrar reportes disponibles por proyecto	Listar reportes adjuntos	5	Alta
Crear elementos interactivos para editar el progreso de los proyectos	Mostrar opciones de estado, borrar, ver reportes, etc.	5	Alta

Crear endpoint PATCH /proyectos/:id/estado	Actualizar campo estado en un proyecto específico.	3	Alta
Crear endpoint GET /proyectos	Devolver resumen de proyectos agrupados por estado.	2	Alta
Crear endpoint GET /proyectos/:id/reportes	Devolver lista de reportes adjuntos a un proyecto.	2	Alta

Como gerente, quiero tener un sitio web con información de la empresa, para atraer más clientes.

Tarea	Descripción	Puntos	Prioridad
Crear página de inicio pública	Maquetar vista con información general de la empresa, bienvenida, logo, diseño	2	Media
Crear sección de servicios ofrecidos	Mostrar los servicios que brinda Pool Center con descripciones e imágenes.	2	Media
Crear sección de proyectos realizados (portafolio)	Mostrar imágenes o ejemplos de proyectos anteriores.	2	Media
Crear página de contacto	Información de contacto para que clientes se contacten	2	Media

Sprint Backlog

Tarea	Descripción	Horas de desarrollo	Puntos	Responsable	Fecha fin
Modificar endpoint POST materiales	Ajustar el endpoint para registrar materiales	3	3	Luis	
Control de acceso basado en roles	Implementar control de acceso con roles	2	8	Iris	
Test 1 back	Prueba unitaria backend	4	3	David	
Test 2 back	Prueba unitaria backend	2	3	Bran	
Test 3 back	Prueba unitaria backend	2	3	Luis	
Test 1 front	Prueba unitaria frontend	2	3	Anggie	
Test 2 front	Prueba unitaria frontend	2	3	Anggie	
Test 3 front	Prueba unitaria frontend	1	3	Anggie	
Cambiar el estilo del topbar	Ajustar estilos visuales del topbar	1	2	Anggie	
Verificar usuario con roles en cookie	Validar información del usuario desde cookie con roles	1	3	David	
Endpoint GET material by id	API para obtener material por ID	1	5	David	
Modificar endpoint POST bodega material	Ajuste de endpoint POST para bodega (en lista)	1	5	Luis	
Endpoint get total de productos	API para obtener el total de productos	1	3	Luis	
Endpoint get proyectos en ejecución	API para obtener proyectos en ejecución	1	3	David	
Endpoint get total de proyectos finalizados	API para obtener proyectos finalizados	1	3	Bran	

Endpoint get total de clientes	API para obtener total de clientes	1	3	Bran	
Endpoint get proyectos por tipo servicio	API para obtener proyectos según tipo de servicio	1	3	David	
Componentes KPI front	Componente visual de indicadores (KPI) en frontend	1	3	Anggie	
Componente/graficas servicios front	Componente gráfico para servicios en frontend	1	5	Iris	
Componente popup Agregar material	Componente popup para agregar material	3	8	Iris	
Componente popup Movimiento material	Componente popup para registrar movimiento de material	3	8	Iris	
Cerrar sesión y redirigir a login	Implementar cierre de sesión y redirección a login	1	3	Anggie	
Probar integración de endpoints en front	Validar integración de endpoints en frontend	5	3	Iris	

Incremento

Repositorio Fronted: https://github.com/AleWWH1104/Fronted_ISoftware1.git

Repositorio Backend: https://github.com/DavidDominguez-11/Backend_ISoftware1

Mock Ups de las pantallas a desarrollar

Enlace a las pantallas: <https://www.figma.com/design/MhHCqPy5YCJfuHK4S1FneK/Pool-Center?node-id=0-1&t=1iPqqrcTjnKkUHIr-1>

Testing

Fronted: <https://youtu.be/T8EIuPAXskM>

Backend: <https://youtu.be/egeAMSnWgC4>

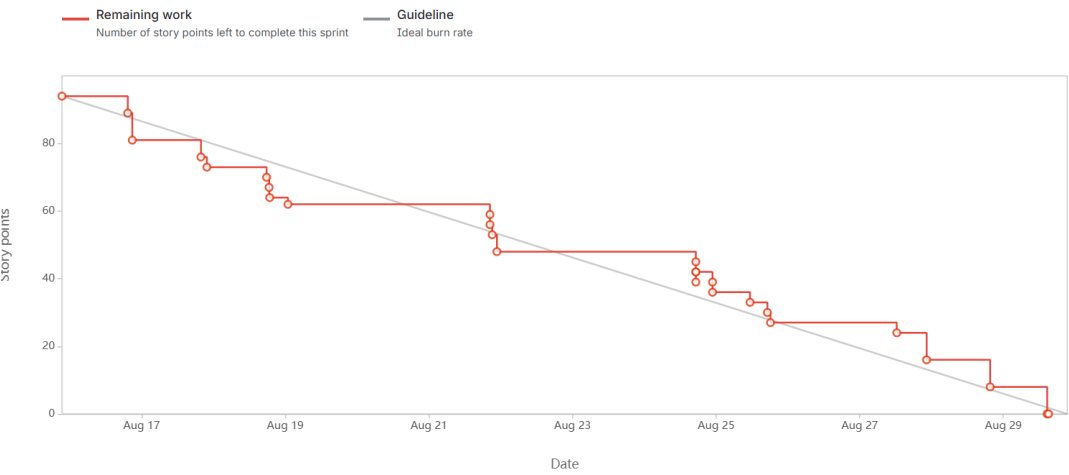
Resultados del Sprint

Video sprint: https://youtu.be/Uno9Y_WwdBw

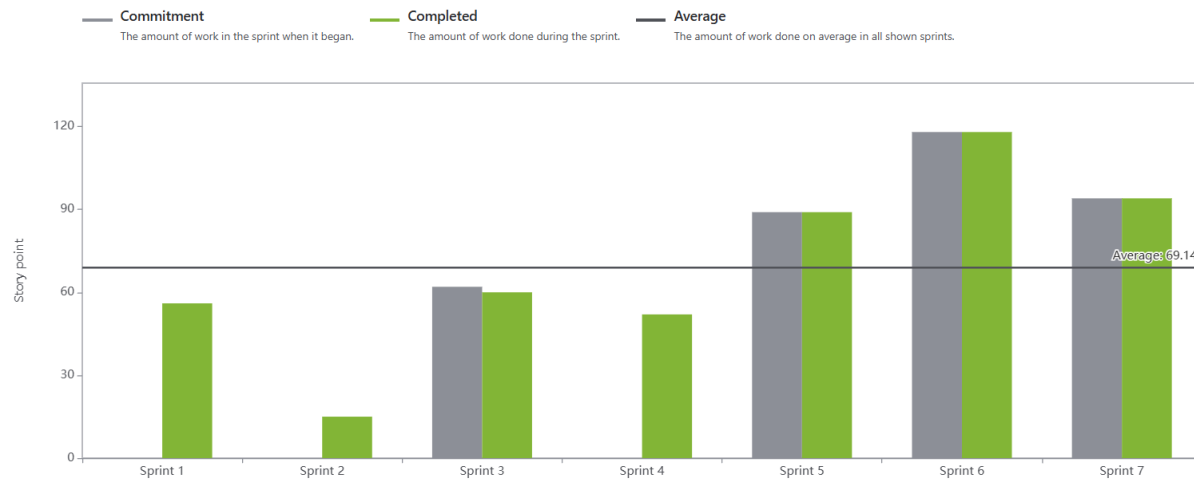
Cumplimiento de tareas

Type	Key	Summary	Status	Comments	Sprint	Assignee
<input checked="" type="checkbox"/>	SCRUM-186	Probar integracion de endpoints en front	DONE	Add comment	🔗 Sprint 7	irisale.ayala
<input checked="" type="checkbox"/>	SCRUM-185	Cerrar sesion y redirigir a login	DONE	Add comment	🔗 Sprint 7	Anggie Daniella Mol..
<input checked="" type="checkbox"/>	SCRUM-184	Componente popup Movimiento material	DONE	Add comment	🔗 Sprint 7	irisale.ayala
<input checked="" type="checkbox"/>	SCRUM-183	Componente popup Agregar material	DONE	Add comment	🔗 Sprint 7	irisale.ayala
<input checked="" type="checkbox"/>	SCRUM-182	Componente/graficas servicios front	DONE	Add comment	🔗 Sprint 7	irisale.ayala
<input checked="" type="checkbox"/>	SCRUM-181	Componentes KPI front	DONE	Add comment	🔗 Sprint 7	Anggie Daniella Mol..
<input checked="" type="checkbox"/>	SCRUM-180	Endpoint get cuantos proyectos segun el tipo de servicio	DONE	Add comment	🔗 Sprint 7	dominguezdavid396
<input checked="" type="checkbox"/>	SCRUM-179	Endpoint get total de clientes	DONE	Add comment	🔗 Sprint 7	gabrielbranb
<input checked="" type="checkbox"/>	SCRUM-178	Endpoint get total de proyectos finalizados	DONE	Add comment	🔗 Sprint 7	gabrielbranb
<input checked="" type="checkbox"/>	SCRUM-177	Endpoint get proyectos en ejecucion	DONE	Add comment	🔗 Sprint 7	dominguezdavid396
<input checked="" type="checkbox"/>	SCRUM-176	Endpoint get total de productos	DONE	Add comment	🔗 Sprint 7	luisfran1164
<input checked="" type="checkbox"/>	SCRUM-175	Modificar endpoint POST bodega material (en lista)	DONE	Add comment	🔗 Sprint 7	luisfran1164
<input checked="" type="checkbox"/>	SCRUM-174	Endpoint GET material by id	DONE	Add comment	🔗 Sprint 7	dominguezdavid396
<input checked="" type="checkbox"/>	SCRUM-173	Verificar informacion del usuario con roles en la cookie	DONE	Add comment	🔗 Sprint 7	dominguezdavid396
<input checked="" type="checkbox"/>	SCRUM-172	Cambiar el estilo del topbar	DONE	Add comment	🔗 Sprint 7	Anggie Daniella Mol..
<input checked="" type="checkbox"/>	SCRUM-171	Test 3 front	DONE	Add comment	🔗 Sprint 7	Anggie Daniella Mol..
<input checked="" type="checkbox"/>	SCRUM-170	Test 2 front	DONE	Add comment	🔗 Sprint 7	Anggie Daniella Mol..
<input checked="" type="checkbox"/>	SCRUM-169	Test 1 front	DONE	Add comment	🔗 Sprint 7	Anggie Daniella Mol..
<input checked="" type="checkbox"/>	SCRUM-168	Test 3 back	DONE	Add comment	🔗 Sprint 7	luisfran1164
<input checked="" type="checkbox"/>	SCRUM-167	Test 2 back	DONE	Add comment	🔗 Sprint 7	gabrielbranb
<input checked="" type="checkbox"/>	SCRUM-166	Test 1 back	DONE	Add comment	🔗 Sprint 7	dominguezdavid396
<input checked="" type="checkbox"/>	SCRUM-165	Control de acceso basado en roles	DONE	Add comment	🔗 Sprint 7	irisale.ayala
<input checked="" type="checkbox"/>	SCRUM-164	Modificar endpoint POST materiales	DONE	Add comment	🔗 Sprint 7	luisfran1164

Gráfico burndown.



Métrica de velocidad



Indicador numérico del éxito del sprint junto con la justificación

El indicador de éxito de este sprint es de 9/10, ya que se cumplió con la totalidad de los objetivos planteados, manteniendo una planificación clara y un ritmo constante de trabajo. Todas las tareas comprometidas fueron completadas en tiempo y forma, hubo retrasos pero el equipo mostró un alto nivel de coordinación y compromiso. Además, creo que se subisternaron ciertas tareas porque terminamos antes de tiempo, por lo que se pudo trabajar mas, pero en si, hubo una buena organizacion y avance del proyecto.

Discusión del éxito del sprint basado en las métricas.

Según las gráficas, el sprint fue exitoso: el burndown muestra una reducción constante del trabajo pendiente hasta llegar a cero en la fecha final, lo que indica que se completaron todas las historias comprometidas. Además, en la métrica de velocidad se refleja que la cantidad de trabajo completado estuvo alineada con el compromiso asumido, manteniéndose dentro del promedio de los últimos sprints. En conjunto, logramos cumplir el plan y cerrar el sprint con todos los objetivos alcanzados.

Evidencia con Product Owner

<https://youtu.be/AmoXi003ywQ>

Ya teniendo el prototipo funcionando, lo único que quiere ver el gerente es ver avances, y está feliz sabiendo que el trabajo va avanzando con el tiempo planeado.

Si le gusto lo que hemos logrado con este sprint, ahora, menciono, que usaramos algún inventario viejo que tuviera para poder tener datos reales de él así pueda probarlo en la siguiente vez que lo veamos.

Ejecución del presupuesto

Estimación de Puntos de Casos de Uso

- FPA (actores): 3 actores = (1 simple + 1 medio + 1 complejo) = $1 \times 1 + 1 \times 2 + 1 \times 3 = 6$
- FPCU (casos de uso): 10 CU = 3 simples, 4 medios, 3 complejos = $3 \times 5 + 4 \times 10 + 3 \times 15 = 125$
- PCU = FPA + FPCU = $6 + 125 = 131$

Ajustes de complejidad

- FCT (promedio de factores técnicos): 0.77
- FA (promedio de factores ambientales): 0.85
- PCUA = $131 \times 0.77 \times 0.85 \approx 85.73$

Esfuerzo Estimado

- FC = 20 HH / punto (por estar en zona de bajo riesgo)
- Esfuerzo inicial E = $85.73 \times 20 = 1,714.6$ HH
- Horas adicionales (2 semanas de 5 días \times 4h/día) = 40 HH
- Esfuerzo total actualizado = $1,714.6 + 40 = 1,754.6$ HH
- En medio tiempo (4 h/día), eso equivale al doble de días calendario, pero el porcentaje se mide sobre HH.
- Este sprint consumió 40 HH, que representan el 2.28% del tiempo total actualizado.

Estimación de costos

- K (costos indirectos): 1.7
- Costo total sin ganancia = $E \times K \times THP = 1,754.6 \times 1.7 \times 8.5 \approx \$25,363.36$
- Margen de ganancia (30%): $\approx \$7,609$

Precio final del proyecto: $\approx \$32,972$ USD

Retrospectiva del sprint

El sprint fue exitoso dado que empezamos a hatar todos los end-points del back-end con featrures del front-end permitiendo ver las funcionalidades implementadas y que el product owner empiece a familiarizarse con el software. Al mismo tiempo se completaron todas las tareas incluso con un parcial de por medio en el sprint lo que consideramos como un logro para el equipo, ya que eso significa que hemos aprendido a llevar un ritmo de trabajo constante sin importar factores externos al proyecto. Consideramos que se podría haber mejorado la estetica de la gráfica sin embargo por periodos de tiempo estuvimos por debajo de la linea deseada lo cual es un indicador de buen progreso.

Gestión de tiempos

Link del excel: [Gestion de tiempos.xlsx](#)

Link del historial de docs: [Sprint 7.docx](#)

Pruebas de experiencia del usuario

1. Alcance de las pruebas

El plan de pruebas contempla los siguientes niveles:

Backend: validación de consistencia de la base de datos, consistencia de estados y rendimiento de consultas.

Frontend: pruebas de integración (humo y top-down), validación de navegación global, búsqueda, filtros y consumo correcto de endpoints.

Experiencia de Usuario (UX): evaluación de la usabilidad, satisfacción y percepción de facilidad de uso en tareas críticas. Incluye pruebas con usuarios reales o representativos, además de encuestas para medir NPS.

2. Herramientas a utilizar

Backend:

Jest → ejecución de pruebas unitarias y de integración.

Supertest → validación de endpoints y respuestas de la API.

Frontend:

Playwright → automatización de pruebas de interfaz y journeys de usuario.

UX:

Herramientas de encuestas (Google Forms, Microsoft Forms o Typeform) → recolección de feedback y cálculo de NPS.

(Opcional) Hotjar o Microsoft Clarity → mapas de calor y grabación de sesiones para detectar patrones de navegación.

3. Tipos de pruebas a elaborar

3.1 Backend

Consistencia de base de datos: validar que las operaciones CRUD mantienen integridad.

Consistencia de estados: verificar que el sistema conserva estados esperados en distintos flujos.

Rendimiento de consultas: medición de tiempos de respuesta y optimización de queries críticas.

3.2 Frontend

Pruebas de humo: verificación inicial de que la aplicación levanta y permite navegación básica.

Integración top-down: validación de componentes de alto nivel antes de detallar subcomponentes.

Pruebas de navegación global: consistencia y fluidez al moverse por secciones.

Pruebas de búsqueda y filtros: validación de resultados relevantes y exactos.

Consumo de endpoints: validar que los datos de backend se presentan de forma correcta.

3.3 UX

Pruebas de usabilidad: usuarios representativos ejecutan tareas clave (registro/login, navegación, búsqueda/filtros, acción crítica).

Encuesta post-uso (NPS + feedback abierto): medir satisfacción y percepción de facilidad de uso.

Pruebas de métricas UX:

Tiempo para completar una tarea.

Número de clics.

Errores cometidos.

Tasa de éxito de las tareas.

4. Encuesta para medir NPS

La encuesta tendría las siguientes preguntas:

NPS:

“En una escala de 0 a 10, ¿qué tan probable es que recomiende nuestra plataforma a un amigo?”

Usabilidad percibida:

“¿Qué tan fácil le resultó completar la tarea principal?”

Muy difícil – Difícil – Neutral – Fácil – Muy fácil.

Satisfacción con la navegación:

“¿Pudo encontrar lo que buscaba sin dificultad?”

Sí – Parcialmente – No.

Feedback abierto:

“¿Qué fue lo más frustrante en su experiencia?”

“¿Qué fue lo más útil o satisfactorio en su experiencia?”

Plan de Pruebas Master

Tests de Front

Test 1: Prueba de integración, verificar que, tras una autenticación exitosa se navega a /home.

Escenario	Variables de Entrada	Resultado Esperado
Autenticación exitosa	email = test@example.com, password = password123, respuesta POST /auth/login = 200 con token válido	Bueno – Se redirige a /home
Credenciales inválidas	email = test@example.com, password = wrongpass, respuesta POST /auth/login = 401	Malo – Muestra error y permanece en /login
Servidor sin respuesta	email = test@example.com, password = password123, POST /auth/login sin respuesta (timeout)	Malo – Muestra error de conexión y permanece en /login
Token inválido o vacío	email = test@example.com, password = password123, respuesta POST /auth/login = 200 con token vacío	Malo – Muestra error de autenticación y permanece en /login
Datos de usuario faltantes	email = test@example.com, password = password123, respuesta POST /auth/login = 200 sin user	Malo – Muestra error y permanece en /login

Test 2: Prueba de regresión visual, Captura de errores visibles con credenciales incorrectas

Escenario	Variables de Entrada	Resultado Esperado
Credenciales incorrectas	email = fail@example.com, password = wrongpass, respuesta POST /auth/login = 401 con mensaje error	Bueno – Se muestra el mensaje 'Credenciales incorrectas'
Servidor sin respuesta	email = fail@example.com, password = wrongpass, POST /auth/login sin respuesta (timeout)	Malo – Se muestra mensaje de error genérico de conexión
Error distinto desde backend	email = fail@example.com, password = wrongpass, respuesta POST /auth/login = 500	Malo – Se muestra mensaje de error interno
Mensaje de error no coincide con esperado	email = fail@example.com, password = wrongpass, respuesta POST /auth/login = 401 con texto distinto	Malo – Mensaje mostrado no es el esperado para credenciales inválidas

Test 3: Prueba funcional: Mostrar mensajes de error si los campos están vacíos

Escenario	Variables de Entrada	Resultado Esperado
Campos vacíos	email = "", password = ""	Bueno – Muestra 'Email es requerido' y 'Contraseña es requerida'
Email vacío y password lleno	email = "", password = password123	Malo – Muestra solo 'Email es requerido'
Email lleno y password vacío	email = test@example.com, password = ""	Malo – Muestra solo 'Contraseña es requerida'
Campos con espacios	email = ' ', password = ' '	Malo – Detecta como vacíos y muestra mensajes de requerido
Campos llenos correctamente	email = test@example.com, password = password123	Bueno – No muestra mensajes de error y continúa flujo normal

Tests de Back

Test 1: AUTH1 - Flujo Completo de Sesión de Usuario

Registro		
Escenario	Variables de Entrada	Resultado Esperado
Registro exitoso	Fullname=Usuario de Prueba, email=auth_test@validation.com, password=password123	Bueno – Usuario creado en BD, contraseña hasheada, status=201
Email duplicado	Fullname=Usuario de Prueba, email=auth_test@validation.com (ya registrado), password=password123	Malo – status=409, mensaje 'El correo ya está registrado'
Datos faltantes	Fullname="", email=auth_test@validation.com, password=password123	Malo – status=400, mensaje de validación de campos
Login		
Escenario	Variables de Entrada	Resultado Esperado
Login exitoso	email=auth_test@validation.com, password=password123	Bueno – status=200, token/cookie devuelto
Credenciales inválidas	email=auth_test@validation.com, password=incorrecta	Malo – status=401, mensaje 'Contraseña incorrecta'
Usuario inexistente	email=noexiste@validation.com, password=password123	Malo – status=404, mensaje 'Usuario no encontrado'
Verificar Token		
Escenario	Variables de Entrada	Resultado Esperado
Token válido	Cookie=token válido obtenido de login	Bueno – status=200, datos de usuario correctos
Token inválido	Cookie=token corrupto o alterado	Malo – status=401, mensaje 'Token inválido'
Sin token	No enviar cookie	Malo – status=401, mensaje 'No autorizado'
Logout		
Escenario	Variables de Entrada	Resultado Esperado
Logout exitoso	Sesión activa con token válido	Bueno – status=200, mensaje 'Sesión cerrada correctamente'
Logout sin sesión activa	Sin token o token inválido	Malo – status=401 o 400, mensaje de error correspondiente

Test 2: Gestión de Proyectos según Rol (PROJ1)

Escenario	Variables de Entrada	Resultado Esperado
Acceso de Admin a proyectos	Login como admin@ejemplo.com, GET /services/projects	Bueno – Respuesta 200, lista de proyectos correcta, tiempo <500ms
Acceso de Supervisor a proyectos	Login como supervisor, GET /services/projects	Bueno – Respuesta 200, lista de proyectos correcta
Acceso de Obrero a proyectos	Login como obrero, GET /services/projects	Bueno – Respuesta 200, lista de proyectos correcta
Estructura de datos incorrecta	Respuesta con campos faltantes o tipo de datos erróneo	Malo – Respuesta inválida, no cumple con formato esperado
Tiempo de respuesta excedido	Tiempo >500ms en GET /services/projects	Malo – Incumple requisito de rendimiento

Test 3: Asignación de Roles (ROLE1)

Escenario	Variables de Entrada	Resultado Esperado
Asignación de rol exitosa	POST /services/register-user-rol con usuario_id=3, roles=[2], cookie de admin válida	Bueno – Respuesta 200, mensaje de confirmación, relación creada en BD
Asignar rol inexistente	POST /services/register-user-rol con rol_id que no existe	Malo – Respuesta de error clara, sin cambios en BD
Asignar rol sin permisos	POST /services/register-user-rol con rol que no tiene permisos	Malo – Respuesta de error clara, sin cambios en BD
Sin autenticación	POST /services/register-user-rol sin cookie/token válido	Malo – Respuesta 401, acceso denegado
Usuario inexistente	POST /services/register-user-rol con usuario_id inexistente	Malo – Respuesta de error clara, sin cambios en BD

Tests de Front

Porque es una de las herramientas más modernas, completas y robustas para pruebas automatizadas de interfaces web. Ha ganado popularidad en la comunidad de desarrollo debido a su enfoque confiable, su rendimiento superior y su amplio soporte multiplataforma. A diferencia de otras herramientas, Playwright permite ejecutar pruebas en todos los navegadores principales, lo que garantiza una mayor cobertura de compatibilidad real con distintos entornos de usuario (Playwright, s. f).

Una gran ventaja de Playwright es su documentación clara y actualizada. Además, soporta funciones avanzadas como capturas de pantalla, grabación de videos, testing de componentes, pruebas móviles emuladas y pruebas paralelas. Playwright cubre un amplio espectro de pruebas: desde pruebas funcionales (interacción de usuarios con formularios, navegación, flujos de autenticación), pruebas de integración (validación de la comunicación entre frontend y backend) hasta pruebas de regresión visual (comparación de capturas para detectar cambios inesperados en la UI) (Playwright, s. f).

Tests de Back

Porque es uno de los frameworks de pruebas más populares en el ecosistema JavaScript gracias a su velocidad, simplicidad y robustez. Ofrece características integradas como mocks, aserciones, cobertura de código y pruebas en paralelo. Por su parte, Supertest es una librería especializada en pruebas de APIs HTTP, que permite simular peticiones directamente al servidor sin necesidad de levantarlo en un puerto real. Esto mejora el rendimiento de las pruebas y permite realizar pruebas de integración completas (Asmare, 2023).

Con Jest + Supertest se pueden cubrir pruebas unitarias (validación de funciones internas y controladores), pruebas de integración (comprobación de interacciones entre rutas, base de datos y lógica del servidor) y pruebas funcionales (verificar que la API responde como se espera desde la perspectiva del cliente). Esta combinación es eficiente, confiable y adecuada tanto para proyectos pequeños como escalables (Asmare, 2023).