



Universidad del Valle de Guatemala

Facultad de Ingeniería

Ingeniería en Software

PROYECTO SANITAS

Plan Maestro de Pruebas

Carnet/Autores:

22272, Bianca Renata Calderón Caravantes

22473, Madeline Nahomy Castro Morales

22716, Aroldo Xavier López Osoy

22233, Daniel Eduardo Dubon Ortiz

22386, Flavio André Galán Donis

18020, Ángel Martín Ortega Yung

Catedrático:

Cristián Muralles

Sección 20

Grupo 05

Fecha:

09/08//2024

I. Introducción

Este plan de pruebas está diseñado para asegurar que el nuevo programa para las enfermeras de la clínica de la universidad funcione correctamente y cumpla con todos los requisitos especificados. El alcance de este plan incluye pruebas de funcionalidades clave, pruebas de interfaz de usuario, pruebas de rendimiento y pruebas de seguridad.

Pruebas UX:

También se abordarán pruebas de UX desde la entrada inicial del usuario hasta la finalización de tareas específicas, con un enfoque especial en la funcionalidad y mensajes emergentes que guían a las doctoras y usuarios de la clínica universitaria.

II. Recursos

Tester	% de participación
Bianca Renata Calderón Caravantes	16.66%
Madeline Nahomy Castro Morales	16.66%
Aroldo Xavier López Osoy	16.66%
Daniel Eduardo Dubon Ortiz	16.66%
Flavio André Galán Donis	16.66%
Ángel Martín Ortega Yung	16.66%

III. Alcance

- El plan abarca:
 - Pruebas de todas las funcionalidades nuevas diseñadas específicamente para las necesidades de las enfermeras.
 - Pruebas de aceptación del usuario para asegurar que el flujo de trabajo de las enfermeras sea intuitivo y eficiente.

- Pruebas de rendimiento para garantizar que el programa pueda manejar el número esperado de usuarios sin degradar el rendimiento.
- Pruebas de seguridad para proteger los datos sensibles de los pacientes.
- **Se agregó al alcance las pruebas UX:**
 - Pruebas de interfaz de usuario para asegurar que la interfaz de usuario es clara, coherente y facilita una rápida navegación y obtención de datos.
 - Pruebas de interacción del usuario para observar y analizar cómo los usuarios interactúan con la plataforma, haciendo énfasis en los mensajes emergentes y alertas presentadas en el flujo de la aplicación.
 - Pruebas de usabilidad para validar la eficacia de las funcionalidades nuevas, con el fin de evaluar cómo afectan la experiencia general del usuario.

IV. Fuera del Alcance

- El mantenimiento de hardware de los dispositivos en los que se ejecutará el programa no está incluido en este plan de pruebas.
- Las integraciones con sistemas de terceros no se probarán en esta fase inicial.

V. Características a probar

- **Descripción de las pruebas a realizar**
 1. Se establecerá el rango de las fechas en las que el sistema será probado.
 2. Se describirán las pruebas que se realizarán al sistema utilizando las plantillas siguientes:
 3. Pruebas de Funcionalidad:

ID Caso	Escenari	Variable 1	Variable	Variable	Resultado Esperado
---------	----------	------------	----------	----------	--------------------

Prueba	o		2	3	
1	Check if Cui exists	CUI: "1234567890123"	Método: GET	-	Estado: 200, CUI existente
2	Check if Cui doesn't exist	CUI: "6234567842123"	Método: GET	-	Estado: 200, CUI no existente
3	Check error response	CUI: "" (vacío)	Método: GET	-	Estado: 403, Respuesta de error
4	Normal case: Create a new patient record	Función: createTestPatient()	-	-	Creación exitosa del paciente
5	Create a new patient record without CUI	CUI: None	Nombres : "Juan"	Apellidos : "Pérez"	Estado: 400, Error: "CUI is required."
6	Create a new patient record with	CUI: uniqueCUI (Duplicado)	Nombres : "Juan", "Carlos"	Apellidos : "Pérez", "González"	Estado: 409, Error: "CUI already exists."

	duplicate CUI				
7	Retrieve existing family medical history	Patient ID: patientId	-	-	Estado: 200, Datos de historial familiar definidos
8	Retrieve default family medical history for non-existent patient	Patient ID: 999999	-	-	Estado: 200, Historial familiar vacío
9	Invalid ID provided	Patient ID: "invalid123"	-	-	Estado: 400, Error: "No valid patientId supplied."
10	should return 403 if no ID is provided	ID: None	-	-	Estado: 403, Sin acceso por falta de ID
11	should return a collaborator	ID: 2	-	-	Estado: 200, Datos específicos del colaborador
12	should	ID: 9999 (fake)	-	-	Estado: 200, Datos por

	return default data				defecto (vacíos)
13	Get patient that exists	Patient ID: patientId	-	-	Estado: 200, Detalles específicos del paciente
14	Fail when retrieving patient that doesn't exist	Patient ID: "999123999"	-	-	Estado: 400, Error: "No patient with the given id found."
15	should return 403 if no CARNET is provided	CARNET: None	-	-	Estado: 403, Acceso denegado por falta de CARNET
16	should return a patient	CARNET: 1	-	-	Estado: 200, Detalles específicos del estudiante
17	should not found a patient	CARNET: 9999 (fake)	-	-	Estado: 404, Estudiante no encontrado
18	Retrieve	Patient ID: patientId	-	-	Estado: 200, Historial

	existing personal medical history				médico personal definido
19	Retrieve default family medical history for non-existe nt patient	Patient ID: 999999	-	-	Estado: 200, Historial médico personal vacío
20	Invalid ID provided	Patient ID: "invalid123"	-	-	Estado: 400, Error: "No valid patientId supplied."
21	Retrieve existing surgical history	Invalid ID provided	-	-	Estado: 200, Historial quirúrgico detallado
22	Retrieve default surgical history for non-existe nt patient	Patient ID: 999999	-	-	Estado: 200, Historial quirúrgico vacío
23	Invalid ID	Patient ID: "abckdj"	-	-	Estado: 400, Error: "No

	provided				valid patientId supplied."
24	Retrieve existing traumatologic history	Patient ID: patientId	-	-	Estado: 200, Historial traumatológico detallado
25	Retrieve default traumatologic history for non-existent patient	Patient ID: 999999	-	-	Estado: 200, Historial traumatológico vacío
26	Invalid ID provided	Patient ID: "invalid123"	-	-	Estado: 400, Error: "No valid patientId supplied."
27	should return DB status: UP if the database connection is successful	URL: LOCAL_API_URL	-	-	Estado: 200, Verificar estado de la conexión a DB

28	should return patients by carnet	Carnet	"A01234 567"		Status: 200, Data not empty
29	should return patients by employee code	NumeroColaborador	"C001"		Status: 200, Data not empty
30	should return patients by name with diacritic	Nombres	"Pérez"		Status: 200, Data not empty
31	should return patients by name no diacritic	Nombres	"Maria"		Status: 200, Data not empty
32	should return an error if search parameter	Carnet	None		Status: 400, "Search parameter must be provided"

	is not provided				
33	should return an error if search type is not provided	None	"A01234567"		Status: 400, "Search type not provided"
34	should return an error if invalid search type is received	InvalidType	"A01234567"		Status: 400, "Invalid search type received"
35	should return an empty array if no patients found, Carnet	Carnet	"NonExistentPatient"		Status: 200, Data empty
36	should return an empty array if no patients	NumeroColaborador	"NonExistentPatient"		Status: 200, Data empty

	found, NumeroC olaborado r				
	should return an empty array if no patients found, Nombres	Nombres	"NonExis tentPatie nt"		Status: 200, Data empty

4. Pruebas de Experiencia de Usuario (UX):

Perfil o Comportamiento de Clientes/Usuarios Potenciales

- **Doctoras**

- **Comportamiento:** Utilizan la plataforma intensivamente para gestionar registros médicos, revisar historiales y actualizar información de los pacientes. Buscan eficiencia y precisión para minimizar el tiempo de administración y maximizar el tiempo de atención al paciente.
- **Necesidades:** Interfaz clara y rápida, acceso seguro a información sensible, facilidad para ingresar y actualizar datos, y notificaciones eficaces que confirmen las acciones sin interrumpir el flujo de trabajo.

- **Pacientes (estudiantes/colaboradores/visitantes)**

- **Comportamiento:** Principalmente ingresan a la plataforma para llenar sus antecedentes médicos una sola vez y, ocasionalmente, para actualizar información menor como contactos de emergencia o direcciones.

- **Necesidades:** Proceso de registro e ingreso de datos simple y guiado, con instrucciones claras y validaciones que aseguren la completitud y exactitud de la información ingresada.

Casos de Uso que Aplican para el UX Dentro del Plan de Pruebas

- **Registro de Nuevos Usuarios (Pacientes y Doctoras)**

- **Objetivo:** Asegurar que el proceso de registro sea intuitivo y seguro.
- **Pruebas Relacionadas:** Completitud y validación del formulario de registro, y confirmación de la cuenta a través del correo electrónico.

- **Inicio de Sesión y Autenticación**

- **Objetivo:** Verificar que los usuarios puedan iniciar sesión de manera segura y recuperar su acceso fácilmente en caso de olvido de contraseña.
- **Pruebas Relacionadas:** Pruebas de ingreso con credenciales válidas e inválidas y la adecuada notificación de errores.

- **Llenado de Antecedentes Médicos por Parte de los Pacientes**

- **Objetivo:** Confirmar que los pacientes puedan llenar su información médica de manera completa y por única vez.
- **Pruebas Relacionadas:** Acceso al formulario, restricciones de llenado múltiple y claridad en la presentación de campos requeridos y opcionales.

- **Acceso y Visualización de Antecedentes Médicos por las Doctoras**

- **Objetivo:** Garantizar que las doctoras puedan acceder y visualizar los antecedentes médicos de los pacientes de forma segura y eficiente.
- **Pruebas Relacionadas:** Verificación de permisos de acceso, facilidad de navegación en la ficha del paciente, y correcta visualización de datos en la sidebar durante la consulta.

- **Actualización de Información Médica por las Doctoras**

- **Objetivo:** Asegurar que las doctoras puedan actualizar la información médica sin dificultades y que los cambios se reflejen correctamente.
- **Pruebas Relacionadas:** Funcionalidad de edición de registros, confirmación de cambios y revisión de historial de modificaciones.

- **Seguridad y Privacidad de la Información**

- **Objetivo:** Verificar que todas las interacciones con la plataforma sean seguras y que la información sensible esté protegida adecuadamente.
- **Pruebas Relacionadas:** Tests de seguridad de la interfaz, validación de permisos de usuario y pruebas de vulnerabilidad.

ID Caso Prueba	Escenario	Variable 1	Variable 2	Variable 3	Resultado Esperado
37	Probar el proceso de registro para nuevos usuarios.	Complejidad del formulario de registro	Validación de datos ingresados	Confirmación de correo electrónico	El usuario debe poder registrarse sin inconvenientes y recibir una confirmación por correo electrónico para verificar su cuenta.
38	Verificar el proceso de login con credenciales válidas e inválidas.	Ingreso de credenciales correctas	Ingreso de credenciales incorrectas	Mensajes de error apropiados	Los usuarios deben poder iniciar sesión con credenciales válidas y recibir mensajes de error claros cuando las credenciales son incorrectas.
39	Evaluar el proceso de llenado de antecedentes médicos por parte de los pacientes.	Acceso al formulario de antecedentes	Complejidad y validez de la información	Confirmación de que la información solo	Los pacientes pueden llenar sus antecedentes médicos una

			ión proporci onada	puede ser llenada una vez	sola vez y recibir una confirmación de que la información ha sido guardada correctamente.
40	Evaluar la capacidad de las doctoras para actualizar información médica de los pacientes.	Acceso al formulario de actualización	Cambios en los datos y su guardado	Confirma ción y registro de los cambios	Las doctoras deben poder actualizar la información médica de los pacientes y recibir una confirmación de que los cambios se han guardado correctamente.
41	Verificar que la sidebar muestre correctamente solo los antecedentes existentes del paciente durante la consulta.	Acceso a la ficha del paciente	Visualiza ción correcta de antecede ntes existentes en la sidebar	Actualiza ción en tiempo real de la sidebar al añadir nuevos antecede ntes	Las doctoras deben ver los antecedentes del paciente listados de manera clara en la sidebar y estos deben actualizarse correctamente al añadir nuevos datos.

5. Pruebas de cargas:

Identificador de la Prueba	Parte de la aplicación probada	Condición	Resultado Esperado	Método o herramienta a utilizar

6. Pruebas de seguridad:

Identificador de la Prueba	Condición	Elemento a probar	Resultado esperado

VI. Criterios de aceptación o fallo

ID criterio	Descripción	Aprobación	Fallo
1	Cobertura de pruebas	100% de los casos de prueba son exitosos	Analizar y resolver las fallas en los casos de prueba
2	Porcentaje de casos exitosos	100% de los casos de prueba son exitosos	Analizar y resolver las fallas en los casos de prueba
3	Cobertura de componentes	Todos los componentes son probados	Aumentar la cobertura de pruebas para componentes faltantes
4	Porcentaje de defectos corregidos	100% de los casos de prueba son exitosos	Priorizar la corrección de defectos pendientes
5	Validación de mensajes de error	Todos los errores deben estar acompañados de un mensaje de validación	Implementar o mejorar los mensajes de validación

VII. Criterios de suspensión y reanudación

Criterio de Suspensión	Criterio de Reanudación
Existencia de defectos críticos que impiden la ejecución de más pruebas	Corrección de defectos críticos y confirmación de funcionalidad
Alcanzar un porcentaje de fallos del 30% en las pruebas ejecutadas	Revisión y mejora del código para reducir la tasa de fallos
Falta de recursos críticos (por ejemplo, acceso a bases de datos)	Restauración del acceso a recursos críticos
Interrupciones en el entorno de prueba (por ejemplo, fallos de servidor)	Restauración del entorno de prueba y verificación de estabilidad

VIII. Infraestructura

- Para garantizar la ejecución adecuada de las pruebas, se utilizará la siguiente infraestructura y herramientas tecnológicas:
 - Sistemas Operativos: Sistemas operativos basados en Unix, proporcionando un entorno estable y reproducible para el desarrollo y las pruebas.
 - Gestores de Paquetes: Nix, npm y yarn, utilizados para manejar las dependencias de las aplicaciones de manera eficiente.
 - Lenguajes de Programación: JavaScript, el principal lenguaje de programación utilizado en el desarrollo del frontend y algunas partes del backend.
 - Frameworks y Bibliotecas: React, utilizado para construir la interfaz de usuario de la aplicación.
 - Servicios en la Nube: AWS, proporcionando infraestructura en la nube para alojar servicios, bases de datos y aplicaciones.
 - Herramientas de Desarrollo y Linting: biome, utilizado para garantizar la consistencia del código y adherencia a las mejores prácticas de codificación.

- Bases de Datos: PostgreSQL, usado como sistema de gestión de bases de datos relacional para almacenar y gestionar los datos de la aplicación.
- AWS SAM SDK: Utilizado para modelar y gestionar aplicaciones compuestas por funciones Lambda y recursos de AWS relacionados de manera local y en la nube.

IX. Suposiciones

- Para la ejecución de las pruebas, se asumirán los siguientes elementos y condiciones:
 - Pre-Prueba de Traducciones: Se asume que todas las traducciones y localizaciones han sido probadas y verificadas antes de la entrega del sistema para pruebas de integración y funcionales.
 - Disponibilidad de Recursos: Se asume que todos los recursos y servicios en AWS estarán disponibles sin interrupciones durante el periodo de pruebas.
 - Estabilidad del Entorno de Nix: Se asume que el entorno de Nix proporcionará una plataforma estable y sin cambios que puedan afectar la reproducibilidad de las pruebas. Esto incluye la base de datos.

X. Riesgos

No.	Riesgos	Probabilidad (1-5)	Impacto (1-5)	Severidad (Prob*Impct)	Plan de Mitigación
	Retrasos en la implementación de las funcionalidades.	2	5	10	Evaluar el avance del desarrollo de las funcionalidades y re-planificar acorde al avance de

					ser necesario.
	Los usuarios no están listos para las pruebas de aceptación (UAT)	1	5	5	Coordinar con las oficinas centrales la selección temprana de los usuarios.
	Fallas de infraestructura en AWS	3	4	12	Implementar estrategias de redundancia y recuperación ante desastres. Monitoreo continuo de la infraestructura.
	Inconsistencias en la configuración del ambiente de	2	3	6	Establecer y validar procedimientos de configuraci

	desarrollo.				ón antes de la fase de pruebas.
	Errores de integración con sistemas heredados	3	4	12	Realizar pruebas de integración tempranas y frecuentes con los sistemas heredados.
	Cambios en los requisitos durante las pruebas	4	4	16	Establecer un proceso claro de gestión de cambios con aprobaciones requeridas antes de su implementación.
	Problemas de rendimiento inesperados	2	5	10	Realizar pruebas de rendimiento progresivas

					a lo largo del desarrollo para detectar problemas temprano.
	Escasez de recursos técnicos para las pruebas	1	4	4	Planificar y asegurar recursos técnicos con anticipación. Asignar tareas según la disponibilidad y habilidades del equipo.
	Falta de entrenamiento adecuado para los usuarios de prueba	2	3	6	Proporcionar sesiones de capacitación antes de comenzar las pruebas UAT.

--	--	--	--	--	--