

Pruebas de Contenido

1. Identificador	PC-01
2. Descripción	Verifica contenido en la página de Inicio
3. Precondiciones	El servidor debe estar en ejecución.
4. Pasos	1. Visitar '/'. 2. Verificar presencia de enlaces, imagen y texto.
5. Resultado Esperado	Los elementos deben estar presentes en la página.
6. Resultado Obtenido	Todos los elementos fueron encontrados correctamente.
7. Resultado	Éxito
8. Observaciones	Contenido correctamente estructurado.
9. Responsable	Steven Restrepo Torres

1. Identificador	PC-02
2. Descripción	Verifica contenido en el formulario de evaluación
3. Precondiciones	Servidor activo y formulario accesible.
4. Pasos	1. Ir a '/Cuestionario_Riesgos'. 2. Revisar campos del formulario.
5. Resultado Esperado	Campos de nombre, episodio y preguntas deben mostrarse.
6. Resultado Obtenido	Formulario cargado exitosamente.
7. Resultado	Éxito
8. Observaciones	La vista muestra todos los elementos requeridos.
9. Responsable	Steven Restrepo Torres

Pruebas de Navegación

1. Identificador	PN-01
2. Descripción	Simula una navegación completa usando Playwright
3. Precondiciones	Servidor local en ejecución.
4. Pasos	1. Ir a Inicio. 2. Realizar evaluación. 3. Exportar PDF. 4. Volver al historial. 5. Eliminar evaluación.
5. Resultado Esperado	Todas las acciones deben completarse sin errores.
6. Resultado Obtenido	Navegación fluida sin fallos.
7. Resultado	Éxito
8. Observaciones	Cobertura completa de flujo funcional.
9. Responsable	Steven Restrepo Torres

Pruebas de Bases de Datos

1. Identificador	BD-01
2. Descripción	Insertión de evaluación válida en MySQL
3. Precondiciones	Base de datos corriendo y Sequelize configurado.
4. Pasos	1. Crear evaluación con campos válidos.
5. Resultado Esperado	Evaluación insertada y ID generado.
6. Resultado Obtenido	Evaluación creada exitosamente.
7. Resultado	Éxito
8. Observaciones	Validación correcta en entorno real.
9. Responsable	Steven Restrepo Torres

1. Identificador	BD-02
2. Descripción	Buscar evaluación por nombre
3. Precondiciones	Debe existir una evaluación previamente insertada.
4. Pasos	1. Ejecutar query usando Sequelize.
5. Resultado Esperado	Se encuentra una evaluación con nombre esperado.
6. Resultado Obtenido	Evaluación localizada correctamente.
7. Resultado	Éxito
8. Observaciones	Consulta exitosa.
9. Responsable	Steven Restrepo Torres

Pruebas Unitarias

1. Identificador	PU-01
2. Descripción	Crear evaluación válida (SQLite en memoria)
3. Precondiciones	Modelo temporal en SQLite definido.
4. Pasos	1. Crear evaluación. 2. Verificar nombre e ID.
5. Resultado Esperado	Evaluación válida con ID asignado.
6. Resultado Obtenido	Evaluación creada sin errores.
7. Resultado	Éxito
8. Observaciones	Correcta validación de modelo.
9. Responsable	Steven Restrepo Torres

1. Identificador	PU-02
2. Descripción	Falla al crear evaluación con nombre inválido
3. Precondiciones	Validación activa en modelo Sequelize.
4. Pasos	1. Insertar nombre con números.
5. Resultado Esperado	Error lanzado por validación.
6. Resultado Obtenido	Validación arrojó excepción.
7. Resultado	Éxito
8. Observaciones	Reglas de validación funcionan correctamente.
9. Responsable	Steven Restrepo Torres

Pruebas de Integración

1. Identificador	PI-01
2. Descripción	Carga masiva y respuesta del endpoint
3. Precondiciones	K6 configurado y servidor activo.
4. Pasos	1. Ejecutar script k6. 2. Medir tiempo y respuesta.
5. Resultado Esperado	Respuestas 200 y sin errores.
6. Resultado Obtenido	Todas las respuestas dentro de parámetros esperados.
7. Resultado	Éxito
8. Observaciones	Endpoint soporta alta concurrencia.
9. Responsable	Steven Restrepo Torres

Pruebas de Seguridad

1. Identificador	PS-01
2. Descripción	Rechazo de intento SQL Injection
3. Precondiciones	Middleware y validaciones activas.
4. Pasos	1. Enviar payload con intento de inyección.
5. Resultado Esperado	Solicitud rechazada (400) con errores.
6. Resultado Obtenido	Petición rechazada correctamente.
7. Resultado	Éxito
8. Observaciones	Validación activa contra inyecciones.
9. Responsable	Steven Restrepo Torres

1. Identificador	PS-02
2. Descripción	Rechazo de intento XSS
3. Precondiciones	Middleware activo y validaciones regex.
4. Pasos	1. Enviar string con <script>. 2. Analizar respuesta.
5. Resultado Esperado	Error de validación (400).
6. Resultado Obtenido	Respuesta rechazada con errores definidos.
7. Resultado	Éxito
8. Observaciones	Regex protege correctamente de scripts.
9. Responsable	Steven Restrepo Torres

1. Identificador	PS-03
2. Descripción	Detectar ausencia de fallback en política CSP.
3. Precondiciones	Política CSP configurada en el servidor.
4. Pasos	Escanear con OWASP ZAP y revisar encabezados de respuesta
5. Resultado Esperado	CSP definida con default-src.
6. Resultado Obtenido	No se encontraba definida originalmente.
7. Resultado	Éxito (ya solucionado)
8. Observaciones	Se agregó default-src: 'self' como fallback.
9. Responsable	Steven Restrepo Torres

1. Identificador	PS-04
2. Descripción	Identificar vulnerabilidad por scripts embebidos permitidos.
3. Precondiciones	Política CSP activa.
4. Pasos	Ejecutar análisis ZAP y observar encabezado script-src.
5. Resultado Esperado	No permitir 'unsafe-inline'.
6. Resultado Obtenido	ZAP lo marcó como presente (falso positivo).
7. Resultado	Éxito (falso positivo mitigado)
8. Observaciones	Se usan nonces dinámicos generados por Helmet; ZAP no los interpreta.
9. Responsable	Steven Restrepo Torres

1. Identificador	PS-05
2. Descripción	Identificar riesgo de inyección de CSS.
3. Precondiciones	Política CSP activa.
4. Pasos	Revisar encabezado style-src con ZAP.
5. Resultado Esperado	No permitir inline styles.
6. Resultado Obtenido	Falso positivo (se usan nonces).
7. Resultado	Éxito (falso positivo mitigado)
8. Observaciones	Riesgo mitigado mediante configuración dinámica.
9. Responsable	Steven Restrepo Torres

1. Identificador	PS-06
2. Descripción	Detectar si el servidor permite Access-Control-Allow-Origin: *.
3. Precondiciones	Middleware CORS configurado.
4. Pasos	Ver respuesta HTTP al escanear con ZAP.
5. Resultado Esperado	Orígenes restringidos.
6. Resultado Obtenido	Originalmente abierto, luego restringido.
7. Resultado	Éxito (falso positivo mitigado)
8. Observaciones	Se configuraron orígenes permitidos mediante variables de entorno.
9. Responsable	Steven Restrepo Torres

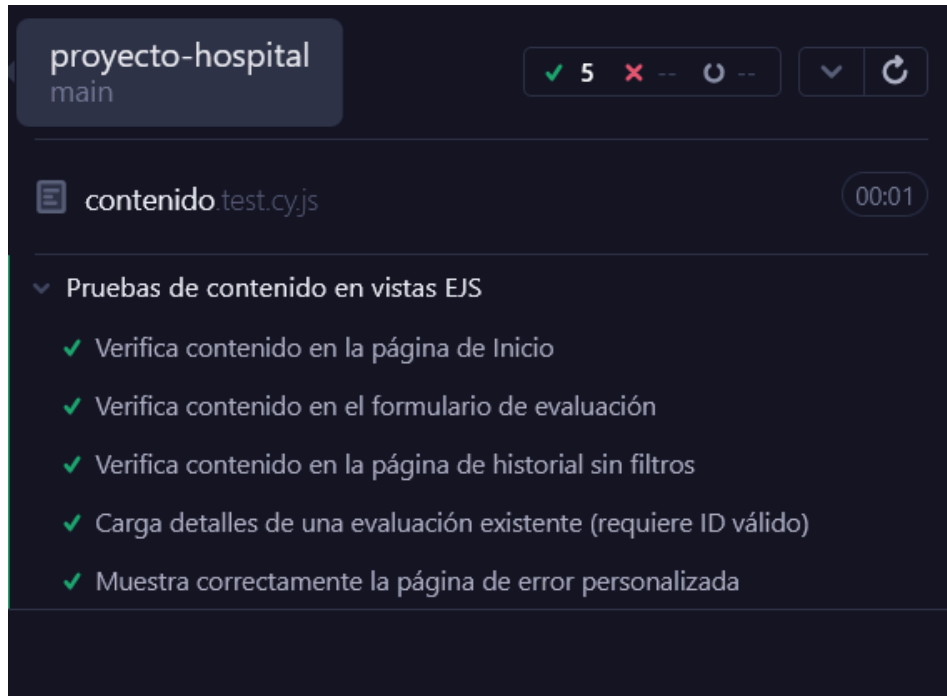
1. Identificador	PS-07
2. Descripción	Verificar que las peticiones sensibles tengan protección contra CSRF.
3. Precondiciones	Formularios y endpoints POST habilitados.
4. Pasos	ZAP detecta falta de tokens CSRF.
5. Resultado Esperado	Protección activa por token.
6. Resultado Obtenido	No detectado al inicio.
7. Resultado	Éxito (falso positivo mitigado)
8. Observaciones	Se implementó @fastify/csrf-protection con cookies seguras.
9. Responsable	Steven Restrepo Torres

1. Identificador	PS-08
2. Descripción	Detectar inclusión de archivos JS desde CDN externos.
3. Precondiciones	Recursos externos cargados desde HTML.
4. Pasos	Analizar recursos mediante ZAP.
5. Resultado Esperado	Solo cargar desde fuentes confiables.
6. Resultado Obtenido	Controlado; se usa jsDelivr y FontAwesome.
7. Resultado	Éxito (falso positivo mitigado)
8. Observaciones	Se evalúa incluir SRI en el futuro para mayor seguridad.
9. Responsable	Steven Restrepo Torres

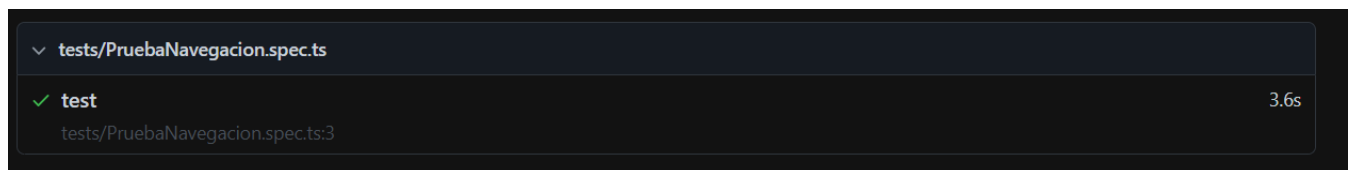
1. Identificador	PS-09
2. Descripción	Detectar política CSP definida en dos lugares.
3. Precondiciones	App corriendo en entorno de pruebas.
4. Pasos	Analizar cabeceras y contenido HTML.
5. Resultado Esperado	CSP definida solo en encabezado HTTP.
6. Resultado Obtenido	Doble definición detectada, luego corregida.
7. Resultado	Éxito (falso positivo mitigado)
8. Observaciones	Se eliminó meta CSP; se mantiene solo en headers con Helmet.
9. Responsable	Steven Restrepo Torres

Pruebas Visuales de los test realizados:

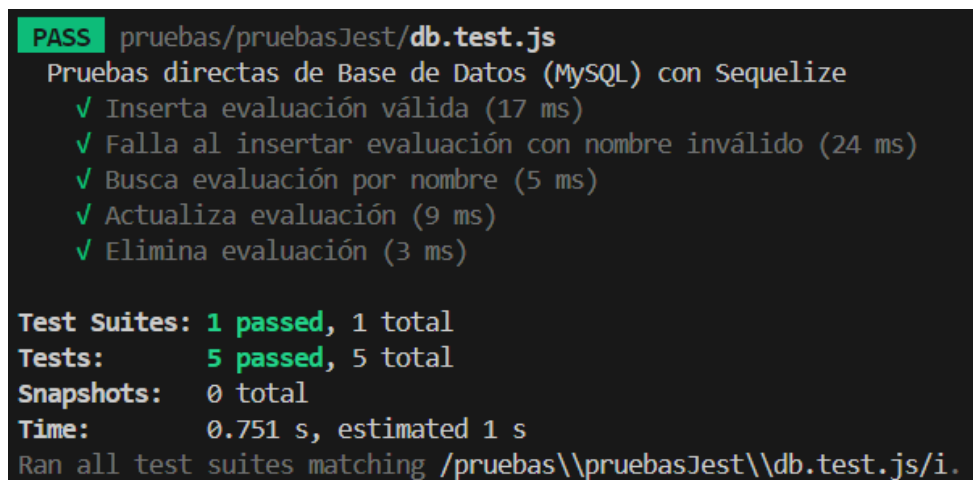
Prueba de contenido:



Prueba de Navegación:



Prueba de base de datos:



Pruebas unitarias:

```
PASS pruebas/pruebasJest/evaluacion.orm.test.js
  CRUD con Sequelize y SQLite (modo local)
    ✓ Crear evaluación válida (8 ms)
    ✓ Falla al crear evaluación con nombre inválido (números) (3 ms)
    ✓ Leer todas las evaluaciones (3 ms)
    ✓ Leer una evaluación por ID (5 ms)
    ✓ Actualizar evaluación existente (4 ms)
    ✓ Eliminar evaluación (4 ms)
    ✓ Filtro por nombre (2 ms)

Test Suites: 1 passed, 1 total
Tests:       7 passed, 7 total
Snapshots:   0 total
Time:        0.781 s, estimated 1 s
Ran all test suites matching /pruebas\\pruebasJest\\evaluacion.orm.test.js/i.
```

Pruebas de integración:

test		
succeeded now in 2m 4s		
Q Search logs		
> ✓ Set up job		0s
> ✓ Checkout del código		2s
> ✓ Instalar Node.js		0s
> ✓ Instalar dependencias del proyecto		7s
> ✓ Instalar k6		37s
> ✓ Ejecutar pruebas de rendimiento con k6		33s
> ✓ Ejecutar pruebas de concurrencia con k6		40s
> ✓ Post Instalar Node.js		0s
> ✓ Post Checkout del código		1s
> ✓ Complete job		0s

Pruebas de seguridad:

```
PASS pruebas/pruebasJest/seguridad.test.js
  Pruebas de Seguridad con CSRF real
    ✓ Rechaza SQL Injection (nombre = '' OR 1=1 --") (43 ms)
    ✓ Rechaza XSS (nombre con <script>) (9 ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:   0 total
Time:        0.885 s, estimated 1 s
Ran all test suites matching /pruebas\\pruebasJest\\seguridad.test.js/i.
```