**PLANTILLA DE REPORTE DE ERROR**

|  |  |
| --- | --- |
| Número de Bug | ***ST-001*** |
| Nombre del Tester | *Romina Sáez* |
| Titulo | *Candidato no puede acceder a prueba técnica - Error 404 y redirección* |
| Día del Reporte | *26/08/2025* |
| Sistema Operativo | *Windows 10 /* |
| Browser | *Chrome* |
| Prioridad | *Alta* |
| Asignado a: | *Romina Sáez* |

**Descripción**

Cuando un candidato hace clic en el link del email de invitación, la aplicación muestra error 404 "Sesión no encontrada" y se reinicia automáticamente, redirigiendo al dashboard administrativo (/admin/dashboard) en lugar de mostrar la interfaz de la prueba técnica.

**Pasos para la reproducción**

* Administrador crea candidato desde panel administrativo*

* Administrador invita candidato a prueba técnica específica*

* Candidato recibe email con link de invitación: /test/[token]*

* Candidato hace clic en el link del email*

* Observar redirección automática al dashboard de administración*

* Verificar logs del servidor y consola del navegador*

**Resultado Esperado**

* Candidato debe ver la interfaz de prueba técnica*

* Debe cargar las preguntas del test asignado*

* Timer debe iniciarse correctamente*

* Candidato debe poder comenzar a responder*

**Resultado Actual**

** Error HTTP 404: "Failed to load resource"**

** Backend: "Sesión no encontrada"**

** Frontend se reinicia automáticamente**

** Redirección forzosa al dashboard administrativo (/admin/dashboard)**

** Candidato no puede acceder a su prueba técnica**

***Causa Raíz Identificada***

***Problema Principal:*** *Desincronización crítica entre el esquema de la base de datos y las consultas SQL en el código.*

***Análisis detallado de la causa raíz:***

1. ***Discrepancia de esquema de base de datos:***
   * *Tabla answers: Código usa created\_at que NO existe en la estructura real*
   * *Tabla test\_sessions: Código usa completed\_at pero la tabla tiene finished\_at*
   * *Falta de validación de columnas antes de ejecutar consultas*
2. ***Impacto en cascada:***
   * *Errores SQL impiden persistencia de respuestas*
   * *Estados de prueba no se actualizan correctamente*
   * *Frontend funcional pero backend no puede guardar datos*
   * *Experiencia de usuario degradada sin feedback claro*
3. ***Fallas en el proceso de desarrollo:***
   * *Migraciones de base de datos desincronizadas*
   * *Ausencia de validación de esquema en tiempo de ejecución*
   * *Manejo de errores insuficiente para problemas de esquema*
   * *Testing insuficiente del flujo completo end-to-end*

***Solución Implementada***

***1. Verificación del esquema real de la base de datos:***

*sql*

*-- Verificar estructura real de tablas*

*PRAGMA table\_info(answers);*

*PRAGMA table\_info(test\_sessions);*

***2. Corrección en sessions.js - saveAnswer (Línea ~250):***

***ANTES (incorrecto):***

*javascript*

*const insertQuery = `*

*INSERT INTO answers (session\_id, question\_id, answer\_text, time\_spent\_seconds, max\_score, attempts\_count, created\_at, last\_modified\_at)*

*VALUES (?, ?, ?, ?, ?, ?, ?, ?)*

*`;*

*const values = [..., timestamp, timestamp]; // created\_at + last\_modified\_at*

***DESPUÉS (corregido):***

*javascript*

*const insertQuery = `*

*INSERT INTO answers (session\_id, question\_id, answer\_text, time\_spent\_seconds, max\_score, attempts\_count, last\_modified\_at)*

*VALUES (?, ?, ?, ?, ?, ?, ?)*

*`;*

*const values = [..., timestamp]; // solo last\_modified\_at*

***3. Corrección en sessions.js - finishTest (Línea ~290):***

***ANTES (incorrecto):***

*javascript*

*const updateQuery = `*

*UPDATE test\_sessions*

*SET status = 'completed', completed\_at = ?, time\_spent\_seconds = ?*

*WHERE token = ?*

*`;*

***DESPUÉS (corregido):***

*Javascript*

*const updateQuery = `*

*UPDATE test\_sessions*

*SET status = 'completed', finished\_at = ?, time\_spent\_seconds = ?*

*WHERE token = ?*

*`;*

1. ***Mejora en manejo de errores:***

*javascript*

*// Agregar validación de esquema y mejor error handling*

*try {*

*const result = db.prepare(insertQuery).run(...values);*

*console.log('✅ Respuesta guardada exitosamente');*

*res.json({ message: 'Respuesta guardada exitosamente' });*

*} catch (error) {*

*console.error('❌ Error SQL detallado:', error.message);*

*console.error('Query que falló:', insertQuery);*

*console.error('Valores:', values);*

*res.status(500).json({*

*error: 'Error al guardar respuesta',*

*details: process.env.NODE\_ENV === 'development' ? error.message : undefined*

*});*

*}*

***Prevención Futura***

1. ***Implementar validación de esquema automática:***

*javascript*

*// Función para validar columnas antes de queries*

*function validateTableSchema(tableName, requiredColumns) {*

*const tableInfo = db.prepare(`PRAGMA table\_info(${tableName})`).all();*

*const existingColumns = tableInfo.map(col => col.name);*

*const missingColumns = requiredColumns.filter(col => !existingColumns.includes(col));*

*if (missingColumns.length > 0) {*

*throw new Error(`Missing columns in ${tableName}: ${missingColumns.join(', ')}`);*

*}*

*}*

1. ***Testing de integración end-to-end:***
   * *Probar flujo completo desde invitación hasta finalización*
   * *Validar persistencia de datos en cada paso*
   * *Verificar compatibilidad entre frontend y backend*
2. ***Migraciones controladas:***
   * *Documentar cambios de esquema de base de datos*
   * *Mantener sincronización entre código y estructura de BD*
   * *Implementar rollback automático para cambios problemáticos*

***Impacto***

* ***Severidad:*** *Crítica - Funcionalidad principal no operativa*
* ***Usuarios afectados:*** *Todos los candidatos invitados*
* ***Tiempo sin servicio:*** *Desde separación frontend/backend*
* ***Pérdida de datos:*** *Ninguna (candidatos creados correctamente)*

***Prevención Futura***

1. ***Validación de esquema:*** *Implementar verificación automática de estructura de BD*
2. ***Testing end-to-end:*** *Cubrir flujo completo candidato desde invitación hasta finalización*
3. ***Logs estructurados:*** *Mejorar logging para identificar errores silenciosos*
4. ***Transacciones atómicas:*** *Asegurar que creación de candidato + sesión sea una operación atómica*

***Evidencia***

***Log Backend (Error 404):***

*=== DEBUG GET TOKEN ===*

*Token recibido: 8c054479-5e4a-4730-9f45-4227e6a8b9e0*

*Sesión encontrada: undefined*

*No se encontró sesión con token: 8c054479-5e4a-4730-9f45-4227e6a8b9e0*

***Log Frontend (Error de carga):***

*Failed to load resource: the server responded with a status of 404 (Not Found)*

*:4000/api/sessions/token/8c054479-5e4a-4730-9f45-4227e6a8b9e0:1*

*Recurso no encontrado*

***Verificación en Base de Datos:***

*sql*

*-- Candidato existe:*

*SELECT \* FROM candidates WHERE session\_token = '8c054479-5e4a-4730-9f45-4227e6a8b9e0';*

*-- Resultado: 1 registro encontrado*

*-- Sesión NO existe:*

*SELECT \* FROM test\_sessions WHERE token = '8c054479-5e4a-4730-9f45-4227e6a8b9e0';*

*-- Resultado: 0 registros - AQUÍ ESTÁ EL PROBLEMA*

***Comportamiento visual:***

* *Página carga por milisegundos*
* *Se reinicia inmediatamente*
* *Usuario es redirigido a /admin/dashboard*
* *No hay mensaje de error visible para el candidato*

***Estado***

* ***Identificado*** *- Causa raíz encontrada*
* ***En desarrollo*** *- Solución implementada*
* ***Testing*** *- Pendiente verificación completa*
* ***Resuelto*** *- Pendiente deploy y validación*