



Documentación de Pruebas - API NexCognitive

1. Información General del Proyecto

- **Nombre de la Aplicación:** NexCognitive
- **Descripción:** Aplicación web desktop dedicada al tratamiento de patologías psicológicas y cognitivas utilizando juegos adaptados a las necesidades del paciente.
- **Fecha de Inicio de Pruebas:** 2 de marzo de 2025
- **Fecha de Finalización de Pruebas:** 12 de marzo de 2025
- **QA Responsable:** Lina Parra

2. Introducción

Este documento detalla las pruebas realizadas para la API de NexCognitive, incluyendo pruebas de autenticación, gestión de profesionales, gestión de pacientes, manejo de sesiones de juego, estadísticas del dashboard y seguridad.

Se diseñó una matriz de casos de prueba iniciales tanto para las funcionalidades de API como para UI y automatización, sin embargo debido al tiempo disponible, solo se ejecutaron las pruebas de API.

3. Alcance de las Pruebas

- Validación de endpoints de autenticación.
- Gestión de pacientes (creación, edición, desactivación, consulta y detalles).
- Gestión de profesionales (registro, edición, eliminación, consulta).
- Manejo de sesiones de juego (creación, actualización de estados y envío de resultados). Incluyendo la obtención de listas filtradas y la recuperación de categorías y tipos disponibles.
- Verificación de estadísticas y últimas sesiones de juego en el dashboard.
- Pruebas de seguridad y manejo de errores.

4. Metodología de Pruebas

Las pruebas se realizaron en Postman Versión 11.36.1 utilizando solicitudes HTTP y validaciones de respuesta. Se ejecutaron casos positivos y negativos para garantizar la robustez de la API. Asimismo, se puede observar la documentación en la versión web de Postman a través del siguiente enlace: [Functional testing NexCognitive](#)

5. Matriz de casos de prueba iniciales

A continuación, se detallan algunos casos generales para cada funcionalidad crítica del MVP. Para ver la lista de comprobación y resultados de las pruebas completas para la API de NexCognitive, consulta el siguiente enlace: [Lista de comprobación API de Nexcognitive](#)

I. Autenticación y gestión de profesionales

Backend (API - Seguridad y Autenticación)

ID	Funcionalidad	Descripción	Tipo de Prueba	Automatizable
CP-BE-001	Registro de Profesionales	Validar que un usuario puede registrarse correctamente.	API	Postman
CP-BE-002	Inicio de Sesión	Verificar autenticación con credenciales correctas e incorrectas.	API	Postman
CP-BE-003	Seguridad en Login	Intentar acceder con credenciales incorrectas múltiples veces.	Seguridad	Manual

ID	Nombre del Caso	Precondiciones	Pasos	Resultado Esperado	Estado
AUTH-001	Iniciar sesión con credenciales válidas	Usuario registrado	1. Enviar POST /api/login con email y password válidos.	200 OK con token JWT.	Aprobado
AUTH-002	Intentar iniciar sesión con credenciales incorrectas	Usuario registrado	1. Enviar POST /api/login con email válido y password incorrecto.	401 Unauthorized.	Aprobado
AUTH-003	Validar estructura del token recibido	Usuario autenticado correctamente	1. Iniciar sesión. 2. Verificar que el token tiene el formato JWT válido.	200 OK con token en formato header.payload.signature	Aprobado
AUTH-004	Intentar iniciar sesión con credenciales incorrectas	Usuario registrado	1. Enviar POST /api/login con email válido y password incorrecto. 2. Verificar la respuesta.	401 Unauthorized con mensaje de error.	Aprobado

Frontend (UI - Formulario de Login y Registro)

ID	Funcionalidad	Descripción	Tipo de Prueba	Automatizable
CP-FE-001	Diseño del Login	Verificar responsividad en desktop.	UI	Manual
CP-FE-002	Validaciones en Formulario	Comprobar errores en campos vacíos y email inválido.	UI	Selenium
CP-FE-003	Flujo Completo de Login	Ingresa usuario y contraseña correctos y validar redirección al dashboard.	UI	Selenium

II. Gestión de pacientes

Backend (API - CRUD de Pacientes)

ID	Funcionalidad	Descripción	Tipo de Prueba	Automatizable
CP-BE-005	Creación de Pacientes	Verificar que un profesional puede registrar un paciente.	API	Postman
CP-BE-006	Edición de Paciente	Validar que los datos del paciente pueden modificarse.	API	Postman
CP-BE-007	Desactivación Paciente	Verificar que un paciente puede eliminarse correctamente.	API	Postman

ID	Nombre del Caso	Precondiciones	Pasos	Resultado Esperado	Estado
PAT-001	Registrar un nuevo paciente	Usuario autenticado	1. Enviar POST /api/patients con datos válidos. 2. Verificar la respuesta.	200 OK con ID del paciente.	Aprobado
PAT-002	Editar datos de un paciente existente	Usuario autenticado con permisos	1. Enviar PUT /api/patients/{id} con datos actualizados. 2. Verificar la respuesta.	200 OK con datos modificados.	Aprobado

Frontend (UI - Formulario de Pacientes y Listado)

ID	Funcionalidad	Descripción	Tipo de Prueba	Automatizable
CP-FE-004	Diseño del Módulo	Verificar que la lista de pacientes se carga correctamente.	UI	Manual
CP-FE-005	Creación de Pacientes	Comprobar que el formulario de pacientes permite ingresar datos y guardarlos.	UI	Selenium
CP-FE-006	Edición de Pacientes	Verificar que los datos se actualizan correctamente en la interfaz.	UI	Selenium

III. Gestión de juegos y partidas

Backend (API - Juegos y Asignaciones)

ID	Funcionalidad	Descripción	Tipo de Prueba	Automatizable
CP-BE-008	Creación de Juegos	Validar que los juegos pueden registrarse en la base de datos.	API	Postman
CP-BE-009	Asignación de Juegos	Verificar que un juego se puede asignar a un paciente con distintos niveles.	API	Postman

ID	Nombre del Caso	Precondiciones	Pasos	Resultado Esperado	Estado
GAME-001	Crear una sesión de juego correctamente	Usuario autenticado	1. Enviar POST /api/game-sessions/create con datos válidos.	200 OK Created con sessionId.	Aprobado
GAME-002	Intentar iniciar una sesión ya iniciada	Usuario autenticado con sesión en curso	1. Enviar PATCH /api/game-sessions/patient/join/{sessionId}/start nuevamente.	409 Conflict.	Aprobado

Frontend (UI - Selección de Juegos y Configuración)

ID	Funcionalidad	Descripción	Tipo de Prueba	Automatizable
CP-FE-007	Listado de Juegos	Validar que el catálogo de juegos se muestra correctamente.	UI	Manual
CP-FE-008	Asignación de Juegos	Verificar que un profesional puede asignar un juego desde la UI.	UI	Selenium

IV. Visualización y Análisis de Resultados

Backend (API - Registro y Consulta de Resultados)

ID	Funcionalidad	Descripción	Tipo de Prueba	Automatizable
CP-BE-011	Registro de Resultados	Validar que los puntajes de juegos se almacenan correctamente.	API	Postman
CP-BE-012	Consulta de Historial	Verificar que los resultados pueden visualizarse por paciente.	API	Postman

ID	Nombre del Caso	Precondiciones	Pasos	Resultado Esperado	Estado
DASH-001	Obtener estadísticas correctamente	Usuario autenticado	1. Enviar GET /api/dashboard/statistics.	200 OK con datos estadísticos.	Aprobado
DASH-002	Obtener las últimas 10 sesiones de juego	Usuario autenticado con sesiones previas	1. Enviar GET /api/dashboard/latestgames.	200 OK con máximo 10 registros.	Aprobado
DASH-003	Intentar acceder a estadísticas sin autenticación	Usuario no autenticado	1. Enviar GET /api/dashboard/statistics sin token.	401 Unauthorized	Aprobado

Frontend (UI - Dashboard de Resultados)

ID	Funcionalidad	Descripción	Tipo de Prueba	Automatizable
CP-FE-009	Historial de Juegos	Validar que los datos de juegos completados se muestran correctamente.	UI	Selenium

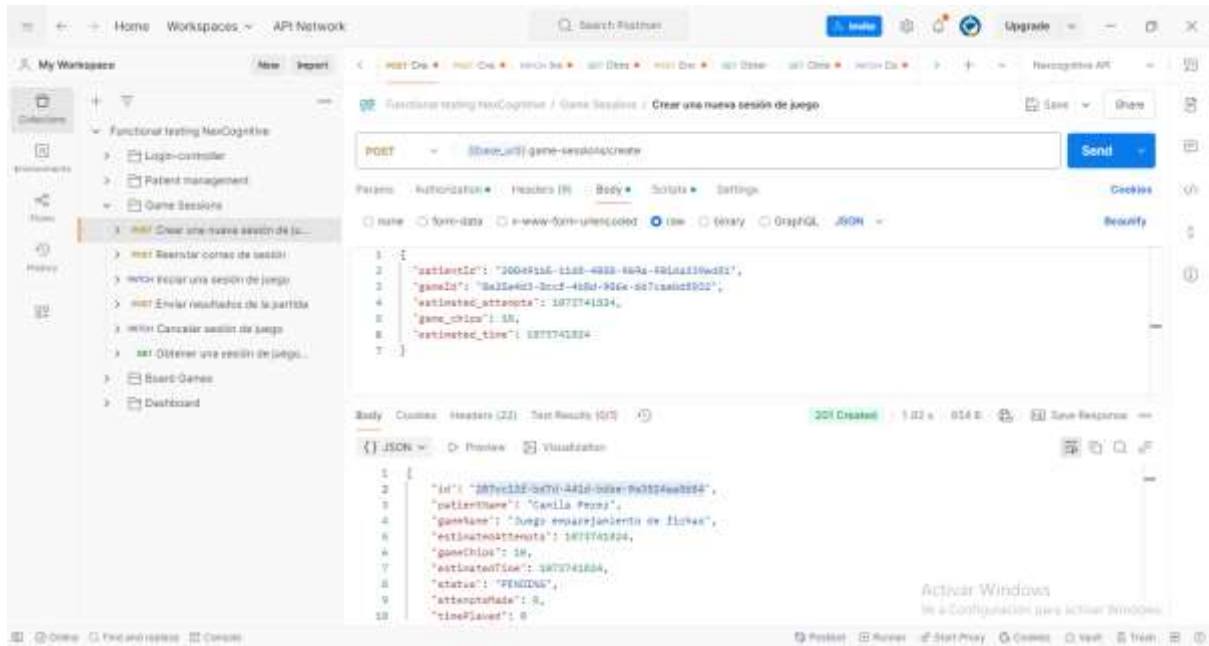
A continuación, se presentan algunas capturas de pantalla de las pruebas más básicas realizadas en Postman, mostrando ejemplos de solicitudes y respuestas obtenidas:

- [illegible]

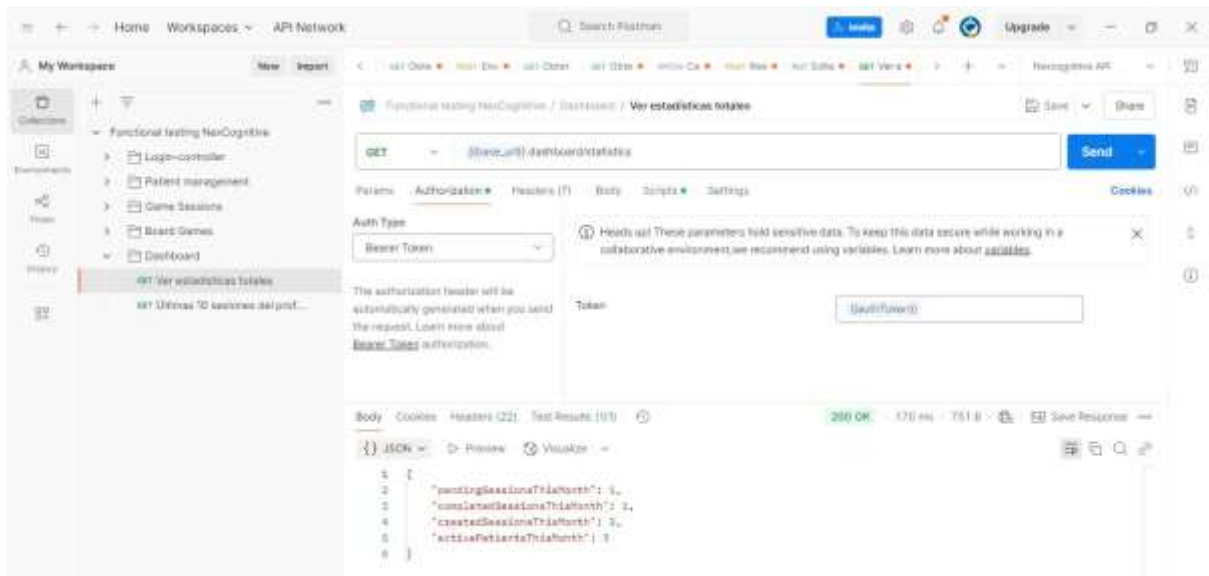
-
- The screenshot displays the Postman application interface. On the left, a sidebar shows a collection of API endpoints under 'My Workspace'. The main workspace is set to 'Functional testing for Cognitive' and shows a REST client request for the endpoint '/patients'. The request is a POST with a JSON body. The bottom section shows the response, which is a 200 OK status with a JSON body containing patient details.
- Request:**
- ```
POST /patients
{
 "id": 123456789,
 "name": "Carla Perez",
 "age": 33,
 "email": "carla_33_perez@example.com",
 "diagnosis": "ataxia"
}
```
- Response:**
- ```
200 OK
{
  "idPatient": "20240125-1120-4020-949a-001a139a081",
  "id": 123456789,
  "name": "Carla Perez",
  "email": "carla_33_perez@example.com",
  "professionalId": "2ba34240-907e-4040-b040-0401923044a3"
}
```

NEXCOGNITIVE

- Ejemplo de creación de sesión de juego (POST /api/game-sessions/create).



- Ejemplo de consulta de estadísticas (GET /api/dashboard/statistics).



7. Conclusión

Las pruebas han validado la correcta funcionalidad de los endpoints críticos del MVP. Se recomienda automatizar pruebas clave para garantizar estabilidad a largo plazo.