**PRUEBAS DE CAJA NEGRA: PLATAFORMA COLABORATIVA**

**SPRINT 01 Funcionalidades Básicas + IA :**

| **Asignación de tareas** | **Código:**  **CP001** | **HU:**  **HU13** |
| --- | --- | --- |
| **Descripción:**  El sistema evaluará si el administrador,puede asignar tareas a los miembros del equipo para organizar el trabajo. | | |
| **Técnica de prueba de caja negra:**  Validación por historia de usuario | | |
| **Datos de entrada:**  Se agregan tareas al proyecto y se asignan a cualquiera de los usuarios que pertenecen al proyecto. | **Resultado esperado:**  Que los miembros del proyecto asignen las tareas a uno o más usuarios. | **Resultado obtenido:**   * Si se puede asignar a cualquier miembro del equipo la tarea. * Pueden ser uno o más. |
| **Captura:** | | |

| **Editar proyectos** | **Código:**  **CP002** | **HU:**  **HU14** |
| --- | --- | --- |
| **Descripción:**  El sistema permite al usuario editar los datos del proyecto para mantenerlo actualizado. | | |
| **Técnica de prueba de caja negra:**  Validación por Historia de usuario | | |
| **Datos de entrada:**   * Pertenecer a un proyecto. * Se le permite editar el nombre, descripción y fechas. * Se pueden guardar los cambios. | **Resultado esperado:**   * Que el usuario pueda cambiar el nombre del proyecto, la descripción y editar fechas. * Te permite guardar los cambios. * Te notifica si se guardó correctamente o hubo algún error. | **Resultado obtenido:**   * Si te permite editar el nombre del proyecto y agregarle una descripción. * Si te notifica que los datos se guardaron de manera correcta. |
| **Captura:**  **-----------------------------------------------------------------------------------** | | |

| **Visualización de tareas** | **Código:** CP003 | **HU:** HU15 |
| --- | --- | --- |
| **Descripción:**  Validar que el usuario pueda visualizar las tareas y su información (estado, prioridad, responsable, fechas) y cambiar su estado mediante arrastrar y soltar (drag & drop). | | |
| **Técnica de prueba de caja negra:**  Validación por historia de usuario | | |
| **Caso:** 3.1 | | |
| **Datos de entrada:**  El usuario accede al tablero desde su sesión, donde ya existen tareas asignadas en backlog. | **Resultado esperado:**  El sistema debe mostrar las tareas en la columna correspondiente, con un estado específico. | **Resultado obtenido:**  En el tablero del proyecto, las tareas fueron correctamente listadas bajo la columna, el sistema, facilitando la comprensión del estado actual del trabajo. |
| **Captura:** | | |
| **Caso:** 3.2 | | |
| **Datos de entrada:**  El usuario observa una tarea registrada previamente con prioridad, responsable asignado y fecha de vencimiento configurada. | **Resultado esperado:**  Cada tarjeta debe mostrar visualmente estos tres elementos un ícono o etiqueta para la prioridad, el nombre o iniciales del responsable y la fecha límite. | **Resultado obtenido:**  En la vista del tablero, al menos una de las tareas muestra etiquetas visibles como "bug", indicadores de prioridad mediante colores o etiquetas, así como identificadores del responsable así mismo la fecha de vencimiento no se visualiza explícitamente. |
| **Captura:** | | |
| **Caso:** 3.3 | | |
| **Datos de entrada:**  El usuario selecciona una tarea ubicada en “Backlog” y la arrastra hacia la columna “In Progress”. | **Resultado esperado:**  La tarea debe moverse a la columna de destino y reflejar su nuevo estado. El sistema debe mantener visible toda la información de la tarea y confirmar el cambio con una notificación o respuesta visual clara. | **Resultado obtenido:**  Al realizar el movimiento de arrastrar y soltar, la tarea fue ubicada correctamente en la columna correspondiente, sin perder sus datos ,el sistema respondió con una notificación emergente indicando "Task created successfully", lo que confirma que el cambio de estado fue exitoso y funcional. |
| **Captura:** | | |

| **Configuración del entorno MERN (SENN)** | **CP004** | **HT01** |
| --- | --- | --- |
| **Descripción:**  Como desarrollador, necesito un entorno MERN (MongoDB, Express, React, Node) configurado con Docker para asegurar consistencia en el desarrollo y despliegue, se hicieron algunas modificaciones en el entorno de desarrollo cambiando MongoDB por Supabase, React por Next.  Criterios de aceptación:   * El proyecto se puede levantar con docker-compose up sin errores. * Las variables de entorno están correctamente definidas y accesibles. * Existe un README con instrucciones básicas de instalación y ejecución. | | |
| **Técnica de prueba de caja negra:**  Operaciones CRUD en tareas. | | |
| **Datos de entrada:**  Iniciar Sesión  Seleccionar proyecto: TEST 1 y agregar nueva tarea:   * Agregar o editar descripción * Agregar o editar comentario. * Eliminar tarea. | **Resultado esperado:**  Las operaciones básicas de CRUD se ejecutan correctamente e inicio de sesión. | **Resultado obtenido:**  Las operaciones CRUD se realizan correctamente, integrando con el servidor de la base de datos de Supabase, mediante las interfaces gráficas y también de las consultas, obteniendo una correcta integración del entorno de desarrollo SENN(modificado del MERN) |
| **Captura:** | | |

| **Nombre:** Integración del modelo IA | **Código:** CP005 | **HT:** HT02 |
| --- | --- | --- |
| **Descripción:** Como desarrollador, necesito integrar un modelo de IA entrenado en Google Colab al backend para responder consultas automáticas de los usuarios.  **Criterios de aceptación:**   * El backend puede recibir preguntas y responder usando el modelo. * El modelo está cargado al iniciar el servidor. | | |
| **Técnica de prueba de caja negra:** Pruebas Funcionales | | |
| **Caso 5.1:** Tiempo de Respuesta | | |
| **Datos de entrada:**   * Endpoint de API * Token de Autenticación * Payload de Entrada (JSON) | **Resultado esperado:**  El tiempo de respuesta debe ser inferior a 2s en condiciones normales. | **Resultado obtenido:**  Se observa que la respuesta es de 1s. |
| **Captura:** | | |
| **Caso 5.2:** Resultado Esperado | | |
| **Datos de entrada:**   * Endpoint de API * Token de Autenticación * Interacciones (JSON) | **Resultado esperado:**  **Primera Solicitud:**  Enviar la primera solicitud con la pregunta inicial y capturar y registrar la respuesta del modelo. | **Resultado Obtenido:**  El modelo logra entender lo solicitado y entrega una respuesta satisfactoria. |
| **Captura:** | | |