



# Plan de Pruebas

TC3004B

Ana Carolina Ramírez González  
A00833324

Juan Pablo Madrigal Magaña  
A01571170

Gabriel Eduardo Diaz Roa  
A00833574

Diego Velazquez Saldaña  
A01177877

Milton Vera De La Torra  
A00833471

10 de marzo de 2024

## 1. Objetivo de las pruebas

El objetivo principal de las pruebas del Oracle Java DevOps ChatBot es asegurar que la aplicación cumpla con todos los requisitos funcionales y no funcionales especificados, proporcionando una experiencia de usuario fluida y eficiente. Esto incluye verificar la funcionalidad completa del ChatBot en la plataforma Telegram, su integración con la infraestructura backend, y garantizar que el ChatBot maneje adecuadamente las tareas de los desarrolladores y los requerimientos de visibilidad del manager. Además, se busca identificar y corregir cualquier defecto antes del lanzamiento, asegurando la estabilidad, seguridad y escalabilidad de la aplicación.

## 2. Alcance de las pruebas

A continuación, un listado de las diferentes pruebas a realizar, con sus respectivos propósitos:

Pruebas Funcionales: Para validar todas las funciones del ChatBot como son la creación, edición y visualización de tareas (to-do 's) por parte de los desarrolladores y managers, y la navegación entre interfaces.

Pruebas de Usabilidad: Para asegurar que la interacción con el ChatBot a través de Telegram es intuitiva y cumple con las expectativas de los usuarios finales.

Pruebas de Integración: Centradas en la correcta comunicación y funcionamiento conjunto del ChatBot con la infraestructura de Oracle Cloud, la base de datos y otros servicios internos, utilizando Telegram como plataforma de interacción.

Pruebas de Seguridad: Para identificar posibles vulnerabilidades en la aplicación y en la comunicación con Telegram, garantizando la protección de los datos de los usuarios y del sistema.

Pruebas de Rendimiento: Para evaluar la capacidad de la aplicación de manejar un volumen significativo de solicitudes simultáneas, manteniendo un rendimiento óptimo y tiempos de respuesta rápidos.

Pruebas de Compatibilidad: Para verificar que el ChatBot funciona correctamente en diferentes dispositivos (computadoras y teléfonos móviles) y sistemas operativos, asegurando una experiencia de usuario consistente.

## 3. Ambiente de pruebas

Como ambiente de pruebas, es necesario tener configurado un entorno que satisfaga las necesidades de hardware, software, y redes.

### **3.1. Requerimientos de Hardware**

Para realizar este proceso tenemos planeado que el proyecto se esté probando de manera local con su propia base de datos. La intención de esta base de datos sería que datos “dummy” para poder visualizar el proyecto de

### **3.2. Requerimientos de Software**

Las pruebas de este proyecto se harán en contenedores de desarrollo utilizando la herramienta de Docker para crear, administrar y ejecutar el chatbot. Ahora bien, para esto es necesario previamente tener instaladas dependencias como JDK y IDE para la compilación y ejecución de código de Java, donde se estará desarrollando el proyecto.

Ahora bien, se desean incluir pruebas automatizadas, por lo que se usaría JUnit para la elaboración de pruebas unitarias. Además, es necesario incluir una herramienta de automatización de pruebas que sea de integración continua para mantener un alto estándar de calidad mientras se van haciendo las iteraciones y se pueda verificar la integridad, problemas y en términos generales, la calidad del software.

En cuanto a la parte de Java, es necesario incluir también la conexión de la API de Telegram en Java (TelegramBots), así como un gestor para administrar las dependencias del proyecto (Maven).

Finalmente, es necesario contar con un sistema de gestión de versiones del proyecto por lo que se estaría utilizando Git para controlar el código fuente y las modificaciones que puedan ir surgiendo con el tiempo en el desarrollo en cuestión.

### **3.3. Requerimientos de Redes**

En cuanto a la parte de red, es indispensable la conexión a internet, de forma que se pueda interactuar con Telegram y recursos externos en caso de ser necesario. Además de esto, hay que asegurar que no hay ningún dilema en la comunicación entre el servidor y los clientes.

## **1. Especificaciones de pruebas**

- Caso de prueba 1: Dar de alta de cuenta de registrada
- Caso de prueba 2: Intentar dar de alta de cuenta no registrada
- Caso de prueba 3: Iniciar sesión con credenciales válidas.
- Caso de prueba 4: Intentar iniciar sesión con credenciales incorrectas.
- Caso de prueba 5: Editar información de la cuenta con datos válidos.
- Caso de prueba 6: Intentar editar la cuenta con información no válida.
- Caso de prueba 7: Navegar entre interfaces.
- Caso de prueba 8: Solicitar la lista de tareas como desarrollador.
- Caso de prueba 9: Agregar una nueva tarea con detalles válidos.
- Caso de prueba 10: Intentar agregar una tarea sin completar los campos con datos correctos.
- Caso de prueba 11: Cambiar el estado de una tarea

- Caso de prueba 12: Ver la lista de proyectos del equipo como manager.
- Caso de prueba 13: Seleccionar un proyecto y ver las tareas asociadas como manager.
- Caso de prueba 14: Obtener la lista de desarrolladores bajo supervisión como manager.
- Caso de prueba 15: Seleccionar un desarrollador y ver sus tareas asociadas como manager.
- Caso de prueba 16: Ver los detalles de una tarea específica como manager.

### **Caso de prueba 1: Dar de alta cuenta registrada en la base de datos**

#### **Precondiciones:**

1. El usuario se encuentra registrado en la base de datos de usuarios de la aplicación.

#### **Datos de entrada:**

1. Cuenta de Desarrollador:
  - a. Correo: a01571170@tec.mx
2. Cuenta de Manager:
  - a. Correo: j.p569@hotmail.com

#### **Flujo de prueba:**

1. Cuenta de Desarrollador:
  - a. El usuario inicia sesión en el chat de Telegram.
  - b. Usa el comando "/alta\_cuenta"
  - c. Proporciona las credenciales para el desarrollador (correo: [a01571170@tec.mx](mailto:a01571170@tec.mx)) en la entrada de texto.
  - d. Recibe un correo con un código para confirmar su identidad.
  - e. Proporciona el código recibido en la entrada de texto del chatbot.
2. Cuenta de Manager:
  - a. El usuario inicia sesión en el chat de Telegram.
  - b. Usa el comando "/alta\_cuenta"
  - c. Proporciona las credenciales para el manager (correo: j.p569@hotmail.com) en la entrada de texto.
  - d. Recibe un correo con un código para confirmar su identidad.
  - e. Proporciona el código recibido en la entrada de texto del chatbot.

Resultados esperados:

1. Cuenta de Desarrollador:
  - a. El sistema verifica la autenticidad del desarrollador con el código recibido y se procesa la solicitud de alta de cuenta.
  - b. El desarrollador es dirigido a la interfaz principal de desarrollador en el chatbot.
2. Cuenta de Manager:
  - a. El sistema verifica la autenticidad del manager con el código recibido y se procesa la solicitud de alta de cuenta.
  - b. El manager es dirigido a la interfaz principal de manager en el chatbot.

Resultado de prueba:

## Caso de prueba 2: Intentar dar de alta de cuenta no registrada

Precondiciones:

1. El usuario no está registrado en la base de datos

Datos de entrada:

1. Correo: prueba@ejemplo.com (correo no registrado).

Flujo de prueba:

1. El usuario inicia sesión en el chat de Telegram.
2. Usa el comando "/alta\_cuenta"
3. Proporciona las credenciales de la cuenta no registrada (correo: [prueba@ejemplo.com](mailto:prueba@ejemplo.com))

Resultados esperados:

1. Se muestra un mensaje en el chat indicando que el usuario no está registrado en la base de datos de usuarios.
2. El chatbot sigue mostrando la interfaz de bienvenida.

Resultados de prueba:

### Caso de prueba 3: Iniciar sesión con credenciales válidas

Precondiciones:

1. El usuario ya ha dado de alta su cuenta
2. El token de autenticación ya expiró en el dispositivo del usuario

Datos de entrada:

1. Cuenta de Desarrollador:
  - a. Correo: [a01571170@tec.mx](mailto:a01571170@tec.mx)
2. Cuenta de Manager:
  - a. Correo: [j.p569@hotmail.com](mailto:j.p569@hotmail.com)

Flujo de prueba:

1. Cuenta de Desarrollador:
  - a. El usuario inicia sesión en el chat de Telegram.
  - b. Usa el comando `/iniciar_sesion`
  - c. Proporciona las credenciales para el desarrollador (correo: [a01571170@tec.mx](mailto:a01571170@tec.mx)) en la entrada de texto.
  - d. Recibe un correo con un código para confirmar su identidad.
  - e. Proporciona el código recibido en la entrada de texto del chatbot.
2. Cuenta de Manager:
  - f. El usuario inicia sesión en el chat de Telegram.
  - g. Usa el comando `/iniciar_sesion`
  - h. Proporciona las credenciales para el manager (correo: [j.p569@hotmail.com](mailto:j.p569@hotmail.com)) en la entrada de texto.
  - i. Recibe un correo con un código para confirmar su identidad.
  - j. Proporciona el código recibido en la entrada de texto del chatbot.

Resultados esperados:

1. Cuenta de Desarrollador:
  - a. El sistema verifica la autenticidad de las credenciales del desarrollador.

<ul style="list-style-type: none"> <li>b. El desarrollador es dirigido a la interfaz principal de desarrollador en el chatbot.</li> <li>c. Se genera un token en el dispositivo del usuario, por lo que intentar iniciar sesión en la cuenta de nuevo no requerirá ingresar de nuevo tu correo.</li> </ul> <p>2. Cuenta de Manager:</p> <ul style="list-style-type: none"> <li>a. El sistema verifica la autenticidad de las credenciales del manager.</li> <li>b. El manager es dirigido a la interfaz principal de manager en el chatbot.</li> <li>c. Se genera un token en el dispositivo del usuario, por lo que intentar iniciar sesión en la cuenta de nuevo no requerirá ingresar de nuevo tu correo.</li> </ul>
<p>Resultado de prueba:</p>

#### Caso de prueba 4: Intentar iniciar sesión con credenciales incorrectas

<p>Precondiciones:</p> <ul style="list-style-type: none"> <li>1. El usuario no se ha dado de alta o no está registrado en la base de datos</li> </ul>
<p>Datos de entrada:</p> <ul style="list-style-type: none"> <li>1. Correo: <a href="mailto:prueba@ejemplo.com">prueba@ejemplo.com</a></li> <li>2. Correo: (vacío)</li> <li>3. Correo: a01571170</li> </ul>
<p>Flujo de prueba:</p> <ul style="list-style-type: none"> <li>1. Correo no dado de alta: <ul style="list-style-type: none"> <li>a. El usuario inicia sesión en el chat de Telegram.</li> <li>b. Usa el comando "/iniciar_sesion"</li> <li>c. Proporciona un correo no dado de alta (correo: <a href="mailto:prueba@ejemplo.com">prueba@ejemplo.com</a>)</li> </ul> </li> <li>2. Entrada vacía: <ul style="list-style-type: none"> <li>a. El usuario inicia sesión en el chat de Telegram.</li> <li>b. Usa el comando "/iniciar_sesion"</li> <li>c. No proporciona ninguna credencial en la entrada de texto (correo: (vacío) )</li> </ul> </li> <li>3. Entrada inválida: <ul style="list-style-type: none"> <li>a. El usuario inicia sesión en el chat de Telegram.</li> </ul> </li> </ul>

<ul style="list-style-type: none"> <li>b. Usa el comando "/iniciar_sesion"</li> <li>c. Proporciona una credencial inválida (correo: a01571170)</li> </ul>
<p>Resultados esperados:</p> <ul style="list-style-type: none"> <li>1. Correo no dado de alta: <ul style="list-style-type: none"> <li>a. El sistema detecta que el correo no se ha dado de alta.</li> <li>b. Se muestra un mensaje en el chat indicando que las credenciales son incorrectas o que no existe ningún usuario dado de alta con esa cuenta y se solicita al usuario que vuelva a intentarlo.</li> </ul> </li> <li>2. Entrada vacía: <ul style="list-style-type: none"> <li>a. El sistema detecta las credenciales incorrectas.</li> <li>b. Se muestra un mensaje en el chat indicando que las credenciales son incorrectas o que no existe ningún usuario dado de alta con esa cuenta y se solicita al usuario que vuelva a intentarlo.</li> </ul> </li> <li>3. Entrada inválida: <ul style="list-style-type: none"> <li>a. El sistema detecta las credenciales incorrectas.</li> <li>b. Se muestra un mensaje en el chat indicando que las credenciales son incorrectas o que no existe ningún usuario dado de alta con esa cuenta y se solicita al usuario que vuelva a intentarlo.</li> </ul> </li> </ul>
<p>Resultado de prueba:</p>

#### **Caso de prueba 5: Editar información de la cuenta con datos válidos**

<p>Precondiciones:</p> <ul style="list-style-type: none"> <li>1. El usuario a iniciado sesión correctamente</li> </ul>
<p>Datos de entrada:</p> <ul style="list-style-type: none"> <li>1. Nuevo nombre: NuevoNombre NuevaApellido</li> </ul>
<p>Flujo de prueba:</p>



1. El usuario usa el comando `"/editar_cuenta"`
2. El usuario usa el comando `"/editar_nombre"`
3. El usuario introduce la nueva información (nombre: NuevoNombre NuevoApellido)
4. El usuario usa el comando `"/confirmar"`

Resultados esperados:

1. El sistema actualiza la información de la cuenta con los nuevos datos proporcionados.
2. El usuario recibe una confirmación en el chat indicando que la cuenta ha sido editada exitosamente.

Resultados de prueba:

#### **Caso de prueba 6: Intentar editar información de la cuenta con datos inválidos**

Precondiciones:

1. El usuario ha iniciado sesión correctamente

Datos de entrada:

1. Nuevo nombre: 1111
2. Nuevo nombre: #
3. Nuevo nombre: (vacío)

Flujo de prueba:

1. El usuario usa el comando `"/editar_cuenta"`
2. El usuario usa el comando `"/editar_nombre"`
3. El usuario introduce la nueva inválida

Resultados esperados:

1. El sistema no actualiza la información de la cuenta con los nuevos datos

proporcionados ni pide confirmación, y envía un mensaje indicando que los datos ingresados son inválidos.

Resultados de prueba:

### Caso de prueba 7: Navegar entre interfaces

Precondiciones:

1. El usuario ha iniciado sesión correctamente

Datos de entrada:

1. Cuenta de Desarrollador:

Cómo comandos (en orden):

1. `"/editar_cuenta"`
2. `"/editar_nombre"`
3. `"/regresar"`
4. `"/inicio"`
5. `"/ver_tareas"`
6. `"Tarea1"`
7. `"/inicio"`

2. Cuenta de Manager:

Cómo comandos (en orden):

8. `"/editar_cuenta"`
9. `"/editar_nombre"`
10. `"/regresar"`
11. `"/inicio"`
12. `"/ver_proyectos"`
13. `"Proyecto1"`
14. `"/inicio"`

Flujo de prueba:

1. Cuenta de Desarrollador:

- a. El usuario usa el comando `"/editar_cuenta"`
- b. El usuario usa el comando `"/editar_nombre"`
- c. El usuario usa el comando `"/regresar"`
- d. El usuario usa el comando `"/inicio"`

<ul style="list-style-type: none"> <li>e. El usuario usa el comando <code>"/ver_tareas"</code></li> <li>f. El usuario ingresa una tarea para ver sus detalles</li> <li>g. El usuario usa el comando <code>"/inicio"</code></li> </ul> <p>2. Cuenta de Manager:</p> <ul style="list-style-type: none"> <li>a. El usuario usa el comando <code>"/editar_cuenta"</code></li> <li>b. El usuario usa el comando <code>"/editar_nombre"</code></li> <li>c. El usuario usa el comando <code>"/regresar"</code></li> <li>d. El usuario usa el comando <code>"/inicio"</code></li> <li>e. El usuario usa el comando <code>"/ver_proyectos"</code></li> <li>f. El usuario ingresa un proyecto para ver sus detalles</li> <li>g. El usuario usa el comando <code>"/inicio"</code></li> </ul>
<p>Resultados esperados:</p> <ul style="list-style-type: none"> <li>1. El sistema sigue con la navegación esperada y es capaz de regresar a la acción anterior o directamente a la interfaz de inicio.</li> <li>2. El sistema termina en la interfaz de inicio.</li> </ul>
<p>Resultados de prueba:</p>

#### **Caso de prueba 8: Solicitar la lista de tareas como desarrollador**

<p>Precondiciones:</p> <ul style="list-style-type: none"> <li>2. El usuario ha iniciado sesión correctamente</li> </ul>
<p>Datos de entrada:</p> <p>Cómo comando:</p> <ul style="list-style-type: none"> <li>1. <code>"/ver_tareas"</code></li> </ul>
<p>Flujo de prueba:</p> <ul style="list-style-type: none"> <li>1. El desarrollador usa el comando <code>"/ver_tareas"</code></li> </ul>
<p>Resultados esperados:</p> <ul style="list-style-type: none"> <li>1. El sistema responde mostrando la lista de tareas del desarrollador correspondiente.</li> </ul>

2. Se muestra una lista de tareas pendientes, en progreso y completadas.
Resultados de prueba:

### Caso de prueba 9: Agregar una nueva tarea con detalles válidos

<p>Precondiciones:</p> <ol style="list-style-type: none"> <li>1. El usuario ha iniciado sesión correctamente como desarrollador</li> </ol>
<p>Datos de entrada:</p> <ol style="list-style-type: none"> <li>1. Entrada de texto: Nombre de ejemplo, Prioridad: Alta, Fecha límite: 2024-05-26.</li> </ol>
<p>Flujo de prueba:</p> <ol style="list-style-type: none"> <li>1. El desarrollador usa el comando <code>/ver_tareas</code></li> <li>2. El desarrollador usa el comando <code>/agregar_tarea</code></li> <li>3. El desarrollador escribe los datos de la nueva tarea en la entrada de texto "Nombre de ejemplo, Prioridad: Alta, Fecha límite: 2024-05-26".</li> <li>4. El desarrollador usa el comando <code>/confirmar</code></li> </ol>
<p>Resultados esperados:</p> <ol style="list-style-type: none"> <li>1. El sistema registra la nueva tarea en la lista del desarrollador con el estatus "por hacer".</li> <li>2. Se muestra un mensaje en el chat indicando que la tarea ha sido creada exitosamente y navega al desarrollador a la interfaz de vista de tareas.</li> </ol>
Resultados de prueba:

### Caso de prueba 10: Intentar agregar una tarea sin completar los campos con datos correctos

Precondiciones:

1. El usuario ha iniciado sesión correctamente como desarrollador

Datos de entrada:

1. Entrada de texto: Nombre de ejemplo 2, Prioridad: Alta (sin fecha límite).
2. Entrada de texto: Nombre de ejemplo 2, Fecha límite: 2024-05-26 (sin prioridad).
3. Entrada de texto: Prioridad: Alta, Fecha límite: 2024-05-26 (sin nombre).

Flujo de prueba:

1. El desarrollador usa el comando `"/ver_tareas"`
2. El desarrollador usa el comando `"/agregar_tarea"`
3. El desarrollador escribe los datos de la nueva tarea de manera errónea de cualquiera de los modos mencionados.

Resultados esperados:

1. El sistema no registra la nueva tarea en la lista del desarrollador ni pide confirmación.
2. Se muestra un mensaje en el chat indicando que los datos ingresados no están completos, y que los datos deben de usar el formato `"nombre_de_proyecto, Prioridad: prioridad, Fecha límite: fecha"`.

Resultados de prueba:

### Caso de prueba 11: Cambiar el estado de una tarea

Precondiciones:

1. El usuario ha iniciado sesión correctamente como desarrollador
2. El usuario tiene una tarea ya existente llamada "Tarea1" y está en el estatus "En progreso"

Datos de entrada:

1. Tarea: Tarea1
2. Entrada de texto: Completada

Flujo de prueba:

<ol style="list-style-type: none"> <li>1. El desarrollador usa el comando <code>"/ver_tareas"</code></li> <li>2. El desarrollador usa escribe el nombre de la tarea "Tarea1" en la entrada de texto para ver sus detalles</li> <li>3. El desarrollador usa el comando <code>"/cambiar_estado"</code></li> <li>4. El desarrollador escribe en la entrada de texto "Completada"</li> </ol>
<p>Resultados esperados:</p> <ol style="list-style-type: none"> <li>1. El sistema actualiza el estado de la tarea en la lista del desarrollador a "Completada".</li> <li>2. Se muestra un mensaje en el chat indicando que el estado de la tarea ha sido modificado exitosamente.</li> </ol>
<p>Resultados de prueba:</p>

**Caso de prueba 12: Ver la lista de proyectos de tu equipo como manager.**

<p>Precondiciones:</p> <ol style="list-style-type: none"> <li>1. El usuario está registrado y ha iniciado sesión como manager</li> <li>2. El usuario tiene al menos un proyecto asociado a su cuenta</li> </ol>
<p>Datos de entrada:</p> <p>No hay datos de entrada</p>
<p>Flujo de prueba:</p> <ol style="list-style-type: none"> <li>a. El manager usa el comando <code>"/ver_proyectos"</code></li> </ol>
<p>Resultados esperados:</p> <p>Hay al menos un proyecto</p> <ol style="list-style-type: none"> <li>a. Se muestra en el chat un mensaje con la lista de los nombres de los proyectos en los que su equipo está trabajando. Estos nombres son como un link y pueden ser clickeables para el caso de prueba 13.</li> </ol> <p>No hay proyecto</p> <ol style="list-style-type: none"> <li>a. Se muestra un mensaje diciendo que no hay ningún proyecto a su nombre</li> </ol>
<p>Resultado de prueba:</p>

**Caso de prueba 13: Seleccionar un proyecto y ver las tareas asociadas como manager.**

Precondiciones:

1. El usuario está registrado y ha iniciado sesión como manager
2. El usuario tiene al menos un proyecto asociado a su cuenta
3. Dicho proyecto tiene al menos una tarea asociada

Datos de entrada:

Por comando:

- a. Nombre del proyecto

Por lista de proyectos:

No hay datos de entrada

Flujo de prueba:

Por comando:

- a. El manager usa el comando “/proyecto”
- b. El manager introduce el nombre del proyecto

Por lista de proyectos:

- a. El manager usa el comando “/ver\_proyectos”
- b. El manager da click en el proyecto del que desea ver las tareas asociadas

Resultados esperados:

Hay al menos una tarea asociada al proyecto

- a. Se muestra un nuevo mensaje con el título del proyecto en la parte superior, así como todas las tareas a las que el proyecto se asocia. Estas tareas son como un link y pueden ser clickeables para el caso de prueba 16.

No hay tareas asociadas al proyecto

- a. Se muestra un mensaje diciendo que no hay mensajes asociados a la tarea

Resultado de prueba:

**Caso de prueba 14: Obtener la lista de desarrolladores bajo su supervisión como manager.**

<p>Precondiciones:</p> <ol style="list-style-type: none"><li>1. El usuario está registrado y ha iniciado sesión como manager</li><li>2. Tener al menos un desarrollador asociado a su cuenta de manager</li></ol>
<p>Datos de entrada:</p> <p>No hay datos de entrada</p>
<p>Flujo de prueba:</p> <ol style="list-style-type: none"><li>a. El manager usa el comando “/desarrolladores”</li></ol>
<p>Resultados esperados:</p> <p>Hay al menos un desarrollador a nombre del manager</p> <ol style="list-style-type: none"><li>a. Se muestra un nuevo mensaje con todos los desarrolladores que están bajo supervisión del manager. Estas tareas son como un link y pueden ser clickeables para el caso de prueba 15.</li></ol> <p>No hay desarrolladores a nombre del manager</p> <ol style="list-style-type: none"><li>a. Se muestra un mensaje diciendo que no hay desarrolladores a nombre del manager</li></ol>
<p>Resultado de prueba:</p>

**Caso de prueba 15: Seleccionar un desarrollador y ver sus tareas asociadas como manager.**

<p>Precondiciones:</p> <ol style="list-style-type: none"><li>1. El usuario está registrado y ha iniciado sesión como manager</li><li>2. Tener al menos un desarrollador asociado a su cuenta de manager</li></ol>
<p>Datos de entrada:</p> <p>Por comando:</p> <ol style="list-style-type: none"><li>a. Nombre del proyecto</li><li>b. Nombre del desarrollador</li></ol>



Por lista de desarrolladores:  
No hay datos de entrada

Flujo de prueba:

Por comando:

- a. El manager usa el comando “/desarrollador”
- b. El manager introduce el nombre del proyecto
- c. El manager introduce el nombre del desarrollador

Por lista de desarrolladores:

- c. El manager usa el comando “/desarrolladores”
- d. El manager da click en el desarrollador que desea ver

Resultados esperados:

- a. Se muestra un nuevo mensaje con el nombre del proyecto y el nombre del desarrollador. Posteriormente, aparece la lista de tareas a las que el desarrollador está asignado para dicho proyecto. A estas tareas se les puede dar click para el caso de prueba 16.

Resultado de prueba:

### **Caso de prueba 16: Ver los detalles de una tarea específica como manager.**

Precondiciones:

1. El usuario está registrado y ha iniciado sesión como manager
2. El usuario tiene al menos un proyecto asociado a su cuenta
3. Dicho proyecto tiene al menos una tarea asociada

Datos de entrada:

Por comando:

- a. Nombre del proyecto
- b. Nombre de la tarea

Por lista de tareas:

No hay datos de entrada

Flujo de prueba:

Por comando:

- a. El manager usa el comando “/tarea”
- b. El manager introduce el nombre del proyecto

<p>c. El manager introduce el nombre de la tarea</p> <p>Por lista de tareas:</p> <ul style="list-style-type: none"> <li>a. El manager usa el comando “/ver_proyectos”</li> <li>b. El manager da click al nombre del proyecto que desea revisar</li> <li>c. El manager da click en el nombre de la tarea que desea revisar</li> </ul>
<p>Resultados esperados:</p> <ul style="list-style-type: none"> <li>a. Se muestra un nuevo mensaje con el título del proyecto y posteriormente el título de la tarea. Aquí muestra los responsables de la tarea, el valor/peso de la tarea, y el estatus de la tarea.</li> </ul>
<p>Resultado de prueba:</p>

### 3. Plan de pruebas

#### Recursos Humanos

- **Analista de Pruebas (Tester):** Diseño y ejecución de los casos de pruebas funcionales y no funcionales.
- **Desarrollador:** Resolución de fallas y errores detectados durante la prueba.
- **Manager:** Supervisión y validación de casos específicos donde se involucre este rol.

#### Herramientas

- Sistema de Gestión de Pruebas
- Herramienta de seguimiento de errores

#### Duración Estimada de las Pruebas

1. **Dada de alta de cuenta registrada/no registrada (Casos 1 y 2):**  
Tiempo estimado: 10 minutos por caso.  
Personal requerido: Analista de pruebas.
2. **Inicio de sesión con/sin credenciales válidas (Casos 3 y 4):**  
Tiempo estimado: 10 minutos por caso.  
Personal requerido: Analista de pruebas.
3. **Editar información de la cuenta con/sin datos válidos (Casos 5 y 6):**  
Tiempo estimado: 30 minutos por caso.  
Personal requerido: Analista de pruebas.
4. **Navegar entre interfaces (Caso 7):**  
Tiempo estimado: 15 minutos.  
Personal requerido: Analista de pruebas.
5. **Funcionalidades específicas del desarrollador (Casos 8 a 11):**  
Tiempo estimado: 30 minutos por caso.  
Personal requerido: Analista de pruebas, Desarrollador para validación.
6. **Funcionalidades específicas del manager (Casos 12 a 16):**  
Tiempo estimado: 20 minutos por caso.  
Personal requerido: Analista de pruebas, Manager para validación.

#### **4. Criterios de Aceptación / Rechazo**

Primeramente, es necesario establecer criterios de aceptación generales que tendrá el sistema.

- Cumplimiento de Requisitos de Usuario: Cantidad de requisitos de usuario cumplidos: al menos el 90% de los requisitos identificados deben ser cumplidos por el chatbot.
- Porcentaje de tiempo de actividad (uptime): El chatbot debe estar disponible y operativo al menos el 99% del tiempo en un período de un mes.
- Cumplimiento de estándares de seguridad: el chatbot debe cumplir con las normativas de seguridad específicas de Oracle y no debe haber ninguna violación de datos durante las pruebas de penetración.
- Cobertura de documentación: se debe proporcionar documentación completa para las características y funcionalidades del chatbot.
- Tiempo de respuesta promedio: el chatbot debe responder a las solicitudes de los usuarios en menos de 2 segundos, siendo el caso promedio por debajo del segundo.
- Capacidad de carga: el chatbot debe ser capaz de manejar al menos 100 solicitudes simultáneas sin experimentar una degradación significativa del rendimiento.

Ahora, es necesario establecer los criterios de aceptación del sistema que sean más específicos a lo desarrollado. Es importante recalcar que se deben de cumplir con todos los puntos para poder dar el siguiente paso, ya que de lo contrario habría una falla considerable que primeramente debe de ser resuelta. Aquí los criterios basados en BDD (Behaviour Driven Development) y gherkin.

#### **Criterios de Aceptación**

Tema	Punto	Criterio
Funcionalidad del Chatbot	Respuesta rápida y precisa a comandos	<b>Dado</b> que un usuario envía un mensaje de inicio o navegación al chatbot <b>Cuando</b> el chatbot recibe el mensaje <b>Entonces</b> el chatbot debería responder en menos de 1 segundo con la información solicitada.
	Acceso según roles y permisos	<b>Dado</b> que un usuario se autentica en el chatbot como administrador o desarrollador <b>Cuando</b> el usuario accede a las funciones correspondientes a su rol <b>Entonces</b> el chatbot debería permitir el acceso sin problemas y mostrar solo las opciones autorizadas para ese rol.
	Mensajes de error con comandos inválidos	<b>Dado</b> que un usuario envía un comando inválido al chatbot <b>Cuando</b> el chatbot recibe el comando inválido <b>Entonces</b> el chatbot debería responder con un mensaje de error claro y descriptivo indicando que el comando no es reconocido.
Gestión de proyectos y tareas	Gestión de proyectos para administradores	<b>Dado</b> que un administrador está autenticado en el chatbot <b>Cuando</b> el administrador crea, edita o elimina un proyecto <b>Entonces</b> el chatbot debería procesar la solicitud y realizar la acción correspondiente, mostrando un mensaje de confirmación.
	Gestión de tareas para desarrolladores	<b>Dado</b> que un desarrollador está autenticado en el chatbot <b>Cuando</b> el desarrollador visualiza sus tareas asignadas y las marca como completadas <b>Entonces</b> el chatbot debería mostrar la lista de tareas asignadas y permitir al desarrollador marcarlas como completadas, mostrando un mensaje de confirmación.
	Asignación de tareas a desarrolladores por administradores	<b>Dado</b> que un administrador está autenticado en el chatbot <b>Cuando</b> el administrador asigna o reasigna una tarea a un desarrollador <b>Entonces</b> el chatbot debería procesar la solicitud y actualizar la asignación de la tarea, mostrando un mensaje de confirmación.
Integración con Telegram	Conexión estable a los mensajes del usuario	<b>Dado</b> que el chatbot está conectado a Telegram <b>Cuando</b> los usuarios envían mensajes al chatbot <b>Entonces</b> el chatbot debería responder de manera consistente y estable, manteniendo una comunicación sin interrupciones con los usuarios.
	Cumplimiento de políticas de Telegram	<b>Dado</b> que el chatbot está integrado con Telegram <b>Cuando</b> se establece la conexión con Telegram y se procesan los mensajes de los usuarios <b>Entonces</b> el chatbot debería cumplir con las directrices y políticas de seguridad de Telegram, garantizando la seguridad y privacidad de los usuarios.
Seguridad	Protección de datos	<b>Dado</b> que la aplicación está en funcionamiento <b>Cuando</b> un usuario intenta acceder a datos sensibles, como información de usuario o proyectos <b>Entonces</b> la aplicación debería requerir autenticación y autorización adecuadas antes de permitir el acceso a los datos, garantizando así la protección de la información sensible.
Escalabilidad	Número creciente de proyectos, tareas y usuarios	<b>Dado</b> que la aplicación está en funcionamiento <b>Cuando</b> el número de proyectos, tareas y usuarios aumenta significativamente <b>Entonces</b> la aplicación debería mantener un rendimiento aceptable sin degradación significativa, garantizando una experiencia fluida para todos los usuarios.

