

Instituto Tecnológico y de Estudios Superiores de Monterrey
Campus Monterrey



TC3005B: Desarrollo e Implementación de Sistemas de Software

Documento de Diseño de pruebas

Equipo 4:

Eduardo Zentella Castillo	A00835387
Cesar Ivan Hernandez Melendez	A00829868
Jose David de la Garza Salas	A00834760
Pablo Andrés Martínez Sánchez	A01252489
Javier Eduardo Corrales Cardoza	A01742328

Lugar y Fecha:

Monterrey, Nuevo
León. 13 de Junio de
2024.

1. Índice

1.1 Tabla de contenidos

1. Índice	2
1.1 Tabla de contenidos	2
2. Introducción	4
2.1 Objetivo del Documento	4
2.2 Alcance	4
3. Alcance del Diseño de Pruebas	4
3.1 Componentes a Probar	4
3.2 Tipos de Pruebas	4
4. Estrategia de Pruebas	5
4.1 Enfoque de Pruebas	5
4.1.1 Pruebas Unitarias	5
4.1.2 Pruebas de Integración	5
4.1.3 Pruebas de Sistema	5
4.1.4 Pruebas de Aceptación	5
5. Criterios de Aceptación	6
5.1 Criterios de Entrada	6
5.2 Criterios de Salida	6
6. Diseño de Pruebas	7
6.1 Casos de Prueba	7
Caso de Prueba 1: Iniciar Sesión	7
Caso de Prueba 2: Acceso a Datos del Servidor	7
6.2 Matriz de Pruebas	7
6.3 Herramientas y Tecnologías Utilizadas	9
7. Entorno de Pruebas	10
7.1 Hardware	10
7.2 Software	10
7.3 Datos de Prueba	10
7.4 Configuración del Entorno	11
8. Gestión de Pruebas	12
8.1 Responsabilidades	12
8.2 Planificación	12
9. Procedimientos de Prueba	14
9.1 Ejecución de Pruebas	14
9.2 Registro de Resultados	14
9.3 Gestión de Defectos	15
10. Análisis de Riesgos	16
10.1 Identificación de Riesgos	16
10.2 Evaluación de Riesgos	16
10.3 Mitigación de Riesgos	17

11. Aprobaciones	18
12. Anexos	19
12.1 Anexo A	19
12.2 Anexo B	50
12.3 Anexo C	50
Referencias	50

2. Introducción

2.1 Objetivo del Documento

El objetivo de este documento es definir el diseño de pruebas para asegurar la calidad y funcionalidad del sistema del proyecto ChatBot Oracle. El documento detalla los diferentes tipos de pruebas que se llevarán a cabo, los criterios de aceptación, el entorno de pruebas, y las responsabilidades del equipo de pruebas.

2.2 Alcance

Este documento cubre las pruebas de unidades, integración, sistema y aceptación para el proyecto ChatBot Oracle. Se incluyen pruebas para el servidor web, base de datos, interfaz de usuario y la API del chatbot.

3. Alcance del Diseño de Pruebas

3.1 Componentes a Probar

- **Servidor Web:** Asegurar que el servidor web responde correctamente a las solicitudes y se comunica eficazmente con la base de datos y el chatbot.
- **Base de Datos:** Verificar la correcta gestión de datos, incluyendo la creación, lectura, actualización y eliminación (CRUD) de registros.
- **Interfaz de Usuario:** Validar que la interfaz de usuario es intuitiva, funcional y responde correctamente a las interacciones del usuario.
- **API del ChatBot:** Asegurar que la API del chatbot responde adecuadamente a los comandos y se integra correctamente con el servicio de mensajería de Telegram.

3.2 Tipos de Pruebas

- **Pruebas Unitarias:** Validar cada módulo individualmente para asegurar que funcionan correctamente por sí solos.
- **Pruebas de Integración:** Verificar la interacción entre módulos para asegurar que trabajan juntos como se espera.
- **Pruebas de Sistema:** Validar el sistema completo en un entorno que simule la operación real, asegurando que todos los componentes funcionan correctamente en conjunto.
- **Pruebas de Aceptación:** Asegurar que el sistema cumpla con los requisitos del cliente y que está listo para ser entregado.

4. Estrategia de Pruebas

4.1 Enfoque de Pruebas

4.1.1 Pruebas Unitarias

Las pruebas unitarias se centrarán en validar cada módulo de forma individual, utilizando casos de prueba diseñados específicamente para cada función y método dentro del código. Se utilizará JUnit como framework de pruebas unitarias para Java, asegurando que cada unidad de código cumpla con sus especificaciones y funcione correctamente de manera aislada.

4.1.2 Pruebas de Integración

Las pruebas de integración verificarán la correcta interacción entre los diferentes módulos del sistema. Se utilizarán casos de prueba que cubran las interfaces entre los componentes del sistema, asegurando que los datos se transmitan y procesen adecuadamente entre ellos. Mockito será utilizado para simular componentes y asegurar la integración sin problemas.

4.1.3 Pruebas de Sistema

Las pruebas de sistema se enfocarán en validar el sistema completo en un entorno que simule las condiciones reales de operación. Se ejecutarán casos de prueba que cubran todos los flujos principales del sistema, asegurando que todas las funcionalidades cumplan con los requisitos especificados. Selenium será utilizado para automatizar las pruebas de sistema a través de la interfaz de usuario, simulando las interacciones del usuario final.

4.1.4 Pruebas de Aceptación

Las pruebas de aceptación se realizarán para garantizar que el sistema cumpla con los requisitos funcionales y no funcionales definidos por el cliente. Se utilizarán casos de prueba que validen los escenarios de uso real del sistema, asegurando que todas las características sean correctas y cumplen con las expectativas del usuario final.

5. Criterios de Aceptación

5.1 Criterios de Entrada

Los criterios de entrada para las pruebas del proyecto ChatBot Oracle son fundamentales para garantizar un proceso de pruebas efectivo y bien estructurado. Aquí se detallan los criterios que deben cumplirse antes de iniciar las pruebas:

- **Finalización del desarrollo de todos los módulos:** Se espera que todos los módulos del sistema estén completamente desarrollados y listos para ser probados. Esto incluye tanto el desarrollo del servidor web, la base de datos, la interfaz de usuario y la API del ChatBot.
- **Disponibilidad del entorno de pruebas:** El entorno de pruebas debe estar completamente configurado y disponible para ejecutar las pruebas de manera efectiva. Esto incluye el servidor de pruebas Oracle Cloud con la dirección IP 159.54.145.9, asegurando que todos los componentes necesarios estén correctamente desplegados y configurados.
- **Definición completa de los casos de prueba:** Todos los casos de prueba deben estar diseñados, documentados y disponibles para su ejecución. Cada caso de prueba debe cubrir escenarios específicos para validar las funcionalidades del sistema de manera exhaustiva.

5.2 Criterios de Salida

Los criterios de salida establecen las condiciones que deben cumplirse al finalizar las pruebas del sistema para considerarlas exitosas:

- **Ejecución completa de los casos de prueba:** Todos los casos de prueba diseñados deben ejecutarse completamente sin errores graves o fallos críticos que afecten la funcionalidad del sistema.
- **Ausencia de defectos críticos o de alta prioridad:** No debe haber defectos identificados que impidan el funcionamiento básico del sistema o que afecten negativamente la experiencia del usuario.
- **Tasa de éxito de los casos de prueba:** La tasa de éxito de los casos de prueba debe ser al menos del 95%, asegurando que la mayoría de las funcionalidades del sistema estén validadas y funcionen según lo esperado.

6. Diseño de Pruebas

En este apartado se detallará el diseño de los casos de prueba específicos que se utilizarán para validar las diferentes funcionalidades del proyecto ChatBot Oracle. Cada caso de prueba se diseñará con el objetivo de verificar el correcto funcionamiento del sistema bajo diferentes condiciones y escenarios de uso.

6.1 Casos de Prueba

A continuación se describen algunos casos de prueba clave que serán implementados para validar el sistema (véase Anexo A para lista completa):

Caso de Prueba 1: Iniciar Sesión

- **Descripción:** Validar que el usuario pueda iniciar sesión correctamente utilizando credenciales válidas.
- **Datos de Entrada:** Nombre de usuario y contraseña válidos.
- **Resultados Esperados:** El sistema debe redirigir al usuario a la pantalla principal del ChatBot Oracle después de iniciar sesión correctamente.

Caso de Prueba 2: Acceso a Datos del Servidor

- **Descripción:** Verificar que los desarrolladores puedan acceder y recuperar datos almacenados en los servidores de manera correcta.
- **Datos de Entrada:** Solicitud de datos específicos almacenados en los servidores.
- **Resultados Esperados:** Los datos solicitados deben ser recuperados y mostrados correctamente en la interfaz de usuario del ChatBot Oracle.

Caso de Prueba 3: Comando para Ver Tareas Pendientes

- **Descripción:** Validar que el chatbot responde correctamente al comando para ver tareas pendientes.
- **Datos de Entrada:** Comando "/tasklist".
- **Resultados Esperados:** El chatbot muestra una lista de todas las tareas pendientes.

6.2 Matriz de Pruebas

La matriz de pruebas detallará la relación entre los diferentes módulos del sistema y los casos de prueba correspondientes. Esto permitirá una visión clara de qué funcionalidades y escenarios están cubiertos por cada prueba, asegurando una cobertura exhaustiva del sistema.

Matriz de Pruebas								
	Módulo	Sub-Módulo	Requerimiento Funcional	Caso de Uso	Descripción del Reporte	Caso de Prueba	Tipo	Prioridad
1	Gestión de tareas	Consulta	Revisar Tareas	Consultar Tareas para Desarrollador	Verificar que el ChatBot permita a los desarrolladores consultar una tarea existente de manera precisa y sin errores.	Verificar que el ChatBot responda correctamente a una solicitud de consulta de tareas pendientes de un desarrollador	DV	A
2			Revisar Todas las Tareas	Consultar Tareas para Manager	Verificar que el ChatBot permita al manager consultar todas las tareas del equipo de desarrollo de manera precisa y sin errores.	El ChatBot responde correctamente a una solicitud de consulta de tareas pendientes de todo el proyecto a una cuenta Manager	DV	A
3		Creación	Añadir Tareas al Chatbot	Añadir Tareas para Desarrollador	Verificar que el ChatBot permita a los desarrolladores añadir una nueva tarea de manera precisa y sin errores.	El ChatBot añade correctamente tareas que le da el desarrollador	DV	A
4		Eliminación	Eliminar Tareas del Chatbot	Eliminar Tareas para Desarrollador	Verificar que el ChatBot permita a los desarrolladores eliminar una tarea existente de manera	El ChatBot elimina correctamente tareas que le indique el desarrollador	DV	A

					precisa y sin errores.			
5		Estado	Completar Tareas del Chatbot	Completar Tarea para Desarrollador	Verificar que el ChatBot permita al desarrollador marcar una tarea como completada y actualice su estado correctamente en la base de datos.	El chatbot marca como completada a las tareas que le indique el desarrollador	DV	A
6	Seguridad	Autenticación	Autenticar Usuario en Inicio de Sesión	Autenticar Usuario en Inicio de Sesión según sea Desarrollador o Manager	Verificar que el ChatBot permita a los usuarios iniciar sesión de manera segura y confiable, además de diferenciar a un desarrollador de un manager.	La aplicación web responde correctamente a una solicitud de inicio de sesión mostrando la pestaña adecuada a la sesión solicitada con sus respectivas funcionalidades	DV	A

6.3 Herramientas y Tecnologías Utilizadas

- **Herramientas de Pruebas:** Se utilizará Selenium para la automatización de pruebas funcionales y de interfaz de usuario. Esta herramienta permitirá ejecutar los casos de prueba de manera eficiente y repetible, garantizando la consistencia en los resultados obtenidos.
- **Datos de Prueba:** Se utilizarán conjuntos de datos específicos diseñados para validar las diferentes funcionalidades del ChatBot Oracle. Estos datos incluirán escenarios típicos de uso y casos extremos para asegurar la robustez del sistema.

7. Entorno de Pruebas

El entorno de pruebas define los recursos necesarios, tanto de hardware como de software, que se utilizarán para llevar a cabo las pruebas del proyecto ChatBot Oracle. Este entorno debe simular lo más fielmente posible las condiciones reales en las que operará el sistema, para asegurar que las pruebas sean representativas y válidas.

7.1 Hardware

Para el entorno de pruebas, se utilizará el siguiente hardware:

- **Servidor Oracle Cloud:**
 - **Dirección IP:** 159.54.145.9
 - **Configuración:** Servidor de alta disponibilidad con capacidad de procesamiento y almacenamiento suficiente para manejar las cargas de trabajo previstas durante las pruebas.

7.2 Software

El entorno de pruebas incluirá el siguiente software:

- **Sistema Operativo:**
 - Oracle Linux Extended Support, que proporcionará un entorno estable y seguro para la ejecución del servidor y las aplicaciones del ChatBot Oracle.
- **Herramientas de Pruebas:**
 - **Selenium:** Utilizado para la automatización de pruebas de interfaz de usuario y pruebas funcionales. Selenium permite simular interacciones del usuario con la interfaz del ChatBot Oracle y verificar los resultados esperados.
- **Entorno de Desarrollo:**
 - **Spring Boot:** El backend del ChatBot Oracle está desarrollado utilizando Spring Boot, un framework que facilita la creación de aplicaciones web robustas y escalables.

7.3 Datos de Prueba

Para asegurar que las pruebas sean exhaustivas y representativas, se utilizarán conjuntos de datos específicos diseñados para cubrir una variedad de escenarios de uso del ChatBot Oracle. Estos conjuntos de datos incluirán:

- **Credenciales de Usuario:** Conjuntos de datos con nombres de usuario y contraseñas válidas e inválidas para probar los mecanismos de autenticación.
- **Solicitudes de Datos:** Datos representativos de las solicitudes típicas y atípicas que el ChatBot Oracle podría recibir, para validar la correcta recuperación y presentación de información desde los servidores.
- **Respuestas de la API:** Datos de prueba que simulan respuestas de la API de Telegram, para verificar que el ChatBot Oracle maneja correctamente la comunicación con el servicio de mensajería.

7.4 Configuración del Entorno

La configuración del entorno de pruebas incluirá la instalación y configuración del sistema operativo, las herramientas de pruebas y el entorno de desarrollo. Esto asegurará que el entorno de pruebas sea consistente con el entorno de producción, permitiendo que los resultados de las pruebas sean aplicables y fiables.

8. Gestión de Pruebas

La gestión de pruebas abarca la planificación, organización y control de todas las actividades relacionadas con las pruebas del ChatBot Oracle. Esto incluye la definición de roles, la planificación del cronograma y la asignación de recursos necesarios para llevar a cabo las pruebas de manera efectiva.

8.1 Responsabilidades

Las responsabilidades para la gestión y ejecución de las pruebas están distribuidas entre los miembros del equipo de pruebas, como se detalla a continuación:

Gerente de Pruebas:

- **Nombre:** Jose David de la Garza Salas
- **Responsabilidades:** Supervisar todas las actividades de prueba, coordinar el equipo de pruebas, asegurar la correcta ejecución de los planes de prueba y gestionar los recursos necesarios.

Equipo de Pruebas:

- **Eduardo Zentella Castillo**
 - **Responsabilidades:** Diseñar y ejecutar casos de prueba, reportar defectos, y colaborar en la resolución de problemas.
- **Cesar Ivan Hernandez Melendez**
 - **Responsabilidades:** Implementar pruebas automatizadas, mantener los scripts de prueba y colaborar en la ejecución de pruebas manuales.
- **Pablo Andrés Martínez Sánchez**
 - **Responsabilidades:** Desarrollar y mantener la infraestructura de pruebas, asistir en la ejecución de pruebas y gestionar los datos de prueba.
- **Javier Eduardo Corrales Cardoza**
 - **Responsabilidades:** Realizar pruebas de regresión, asegurar la cobertura completa de pruebas, y documentar los resultados de las pruebas.

8.2 Planificación

La planificación de las pruebas se detalla en un cronograma que establece las fases de prueba con sus respectivas fechas de inicio y fin. A continuación se muestra un resumen de las fases principales:

- **Fase de Preparación:**
 - **Fecha de Inicio:** 24/05/2024
 - **Fecha de Finalización:** 31/05/2024
 - **Actividades:** Configuración del entorno de pruebas, instalación de herramientas y software, definición de casos de prueba.
- **Fase de Ejecución de Pruebas:**
 - **Fecha de Inicio:** 03/06/2024
 - **Fecha de Finalización:** 13/06/2024
 - **Actividades:** Ejecución de pruebas unitarias, de integración, de sistema y de aceptación.
- **Fase de Evaluación y Reporte:**
 - **Fecha de Inicio:** 03/06/2024
 - **Fecha de Finalización:** 13/06/2024
 - **Actividades:** Análisis de resultados, reporte de defectos, evaluación de la calidad del sistema.

9. Procedimientos de Prueba

Los procedimientos de prueba detallan los pasos específicos para ejecutar los casos de prueba, registrar los resultados y gestionar los defectos identificados durante las pruebas. Este apartado asegura que todas las pruebas se realicen de manera consistente y que los resultados sean documentados de manera adecuada.

9.1 Ejecución de Pruebas

Cada caso de prueba tendrá un conjunto definido de pasos que deben seguirse para asegurar que las pruebas se ejecuten correctamente. A continuación se detallan los pasos generales para la ejecución de pruebas:

1. **Preparación del Entorno de Pruebas:**

- Verificar que todos los componentes necesarios (servidor, base de datos, API del ChatBot) estén configurados y operativos.
- Cargar los datos de prueba necesarios en el sistema.

2. **Ejecución de Casos de Prueba:**

- Seleccionar el caso de prueba a ejecutar.
- Ingresar los datos de entrada especificados en el caso de prueba.

- Ejecutar el caso de prueba siguiendo los pasos detallados.
 - Comparar los resultados obtenidos con los resultados esperados.
- 3. Registro de Resultados:**
- Documentar los resultados de cada caso de prueba, indicando si pasó o falló.
 - En caso de fallar, capturar detalles sobre el comportamiento inesperado y cualquier mensaje de error.

9.2 Registro de Resultados

Para asegurar un registro preciso y detallado de los resultados de las pruebas, se seguirá el siguiente procedimiento:

- 1. Documentación de Resultados:**
 - Utilizar una hoja de cálculo o una herramienta de gestión de pruebas para registrar los resultados.
 - Incluir detalles como el identificador del caso de prueba, descripción, datos de entrada, resultados esperados, resultados obtenidos y estado (pasó/falló).
- 2. Generación de Reportes:**
 - Al finalizar cada ciclo de pruebas, generar un reporte que resuma los resultados.
 - El reporte debe incluir métricas clave como el número de casos de prueba ejecutados, porcentaje de éxito, y lista de defectos identificados.

9.3 Gestión de Defectos

La gestión de defectos asegura que todos los problemas encontrados durante las pruebas sean reportados, rastreados y resueltos de manera eficiente. Los siguientes pasos describen el proceso de gestión de defectos:

- 1. Reporte de Defectos:**
 - Utilizar una herramienta de seguimiento de defectos para reportar cada defecto encontrado.
 - Incluir detalles como el identificador del caso de prueba, descripción del defecto, pasos para reproducir, gravedad, y cualquier evidencia adicional (capturas de pantalla, logs).
- 2. Rastreo de Defectos:**
 - Asignar cada defecto a un desarrollador responsable para su resolución.
 - Monitorear el estado del defecto (nuevo, en progreso, resuelto, cerrado) y actualizarlo según sea necesario.
- 3. Resolución de Defectos:**

- Los desarrolladores investigarán y corregirán los defectos reportados.
 - Una vez corregido, el defecto será reprobado para confirmar que la solución es efectiva y que no introdujo nuevos problemas.
4. **Revisión y Cierre:**
- Revisar todos los defectos resueltos para asegurar que han sido correctamente abordados.
 - Cerrar los defectos una vez confirmada su resolución.

10. Análisis de Riesgos

El análisis de riesgos identifica posibles problemas que podrían afectar las pruebas del proyecto ChatBot Oracle, evalúa su impacto y probabilidad, y desarrolla estrategias para mitigarlos. Este proceso es crucial para minimizar la interrupción y asegurar la calidad del sistema.

10.1 Identificación de Riesgos

A continuación se enumeran algunos posibles riesgos que podrían surgir durante las pruebas del ChatBot Oracle:

1. **Incompatibilidad de Sistemas:**
 - Los componentes del ChatBot (servidor, base de datos, API) pueden no funcionar bien juntos en el entorno de pruebas.
2. **Datos de Prueba Insuficientes o Incorrectos:**
 - Falta de datos de prueba representativos que cubran todos los escenarios posibles.
3. **Fallas en la Integración:**
 - Problemas al integrar diferentes módulos del sistema.
4. **Tiempo Insuficiente para Pruebas:**
 - Plazos ajustados que limitan la ejecución completa de todas las pruebas planificadas.

5. Falta de Recursos:

- Insuficiencia de personal o herramientas para ejecutar todas las pruebas necesarias.

6. Defectos Críticos No Identificados:

- Defectos importantes que no se detectan durante las pruebas iniciales.

7. Problemas de Comunicación con la API de Telegram:

- Fallos en la comunicación o actualización de la API de Telegram que afecten el funcionamiento del ChatBot.

10.2 Evaluación de Riesgos

Cada riesgo identificado se evalúa en términos de su impacto y probabilidad de ocurrencia. Los riesgos se clasifican en una escala de bajo, medio y alto.

1. Incompatibilidad de Sistemas:

- Impacto: Alto
- Probabilidad: Media

2. Datos de Prueba Insuficientes o Incorrectos:

- Impacto: Medio
- Probabilidad: Alta

3. Fallas en la Integración:

- Impacto: Alto
- Probabilidad: Media

4. Tiempo Insuficiente para Pruebas:

- Impacto: Alto
- Probabilidad: Alta

5. Falta de Recursos:

- Impacto: Medio
- Probabilidad: Media

6. Defectos Críticos No Identificados:

- Impacto: Alto
- Probabilidad: Media

7. Problemas de Comunicación con la API de Telegram:

- Impacto: Alto
- Probabilidad: Media

10.3 Mitigación de Riesgos

Para cada riesgo identificado y evaluado, se desarrollan estrategias de mitigación para reducir su impacto o probabilidad de ocurrencia.

1. Incompatibilidad de Sistemas:

- Realizar pruebas de integración tempranas y continuas para identificar y resolver problemas de compatibilidad.

- Utilizar entornos de pruebas similares al entorno de producción.
- 2. **Datos de Prueba Insuficientes o Incorrectos:**
 - Crear un conjunto completo y representativo de datos de prueba.
 - Revisar y validar los datos de prueba antes de su uso.
- 3. **Fallas en la Integración:**
 - Planificar y ejecutar pruebas de integración en etapas.
 - Implementar un control de versiones adecuado para manejar cambios en los módulos.
- 4. **Tiempo Insuficiente para Pruebas:**
 - Establecer prioridades claras para las pruebas más críticas.
 - Optimizar los procesos de prueba para maximizar la eficiencia.
- 5. **Falta de Recursos:**
 - Asignar recursos adicionales en función de las necesidades del proyecto.
 - Utilizar herramientas de automatización para reducir la carga de trabajo manual.
- 6. **Defectos Críticos No Identificados:**
 - Aumentar la cobertura de pruebas con casos de prueba exhaustivos.
 - Realizar revisiones de código y pruebas de pares.
- 7. **Problemas de Comunicación con la API de Telegram:**
 - Mantener actualizada la documentación de la API y estar al tanto de los cambios.
 - Implementar mecanismos de fallback para manejar errores en la comunicación con la API.

11. Aprobaciones

- **Firma del Gerente de Pruebas:**

- **Firma del Gerente del Proyecto:**

- **Firma del Cliente:**

12. Anexos

12.1 Anexo A

Lista completa de casos de prueba.

Casos de Prueba								
Nombre Caso de Prueba <modulo_numero_descripc ion>	Descripción Caso de Prueba	Precondici ón <NA / precondici ón>	Postcondición	Prior idad <Alta, Mediana, Baja >	Tipo de Caso de Prue ba <DV /EH/ GUI/ BND >	No. Paso	Descripción del Paso	Resultado Esperado

1	<p>Verificar que el ChatBot responda correctamente a una solicitud de consulta de tareas pendientes de un desarrollador</p>	<p>Este caso de prueba evalúa la funcionalidad del ChatBot para responder adecuadamente a la solicitud de un desarrollador de consultar sus tareas pendientes. Se verifica que el ChatBot presente una lista precisa y completa de las tareas pendientes del desarrollador, incluyendo detalles como la descripción de la tarea, la fecha de vencimiento y el estado actual. La respuesta del ChatBot debe ser clara y fácil de entender para el usuario, y debe manejar correctamente cualquier escenario donde no haya tareas pendientes para el usuario.</p>	<p>El ChatBot está activo y funcionando correctamente.</p> <p>El desarrollador ha iniciado una conversación con el ChatBot a través de Telegram.</p> <p>El desarrollador tiene tareas pendientes registradas en el sistema.</p>	<p>Después de ejecutar con éxito el caso de prueba, se espera que el ChatBot presente al usuario una lista precisa y completa de sus tareas pendientes. La información mostrada debe reflejar fielmente las tareas almacenadas en la base de datos del sistema. Además, la respuesta del ChatBot debe ser clara y fácil de entender para el usuario, proporcionando detalles como la descripción de la tarea, la fecha de vencimiento y el estado actual de cada tarea. En caso de no haber tareas pendientes para el usuario, el ChatBot debe informar claramente que no hay tareas pendientes disponibles.</p>	A	DV	1	<p>Verificar que el sistema y el ChatBot estén activos y funcionando correctamente.</p>	<p>Se confirma que el sistema y el ChatBot están activos y respondiendo a las solicitudes de los usuarios sin errores aparentes.</p>
---	---	---	---	--	---	----	---	---	--

2							El desarrollador abre Telegram y se autentica si es necesario.	El desarrollador abre la aplicación Telegram y se autentica si se le solicita.
3							El desarrollador busca y accede al chat del ChatBot.	El desarrollador busca el chat del ChatBot y accede a él correctamente.
4							Inicio de la conversación con un saludo o comando inicial al ChatBot.	El desarrollador inicia la conversación enviando un saludo o comando inicial al ChatBot.
5							El ChatBot responde al saludo, indicando su disponibilidad para recibir comandos.	El ChatBot responde al saludo del desarrollador indicando su disponibilidad para recibir comandos y

								solicitudes.
6							El desarrollador envía el comando "/tareas pendientes" al ChatBot para solicitar la lista de tareas pendientes.	
							El desarrollador envía el comando "/tareas pendientes" al ChatBot.	
							El ChatBot confirma la recepción del comando y comienza a procesar la solicitud.	
							El ChatBot procesa la solicitud para comprender la consulta del desarrollador sobre las tareas pendientes.	

						9	El ChatBot accede a la base de datos para buscar tareas pendientes asignadas al desarrollador.	El ChatBot accede a la base de datos para buscar las tareas pendientes asignadas al perfil del desarrollador.
						10	El ChatBot verifica la existencia de tareas pendientes asociadas al perfil del desarrollador.	El ChatBot verifica si existen tareas pendientes asociadas al perfil del desarrollador en la base de datos.
						11	Si existen tareas pendientes, el ChatBot genera una lista detallada de estas.	Si existen tareas pendientes, el ChatBot genera una lista detallada de las tareas pendientes y sus respectivos detalles.

						12	Si no hay tareas pendientes, el ChatBot prepara un mensaje indicativo de que no hay tareas.	Si no hay tareas pendientes asociadas al perfil del desarrollador, el ChatBot prepara un mensaje indicativo de esta situación.
						13	El ChatBot envía la respuesta adecuada al desarrollador a través de Telegram.	El ChatBot envía la respuesta adecuada al desarrollador a través de Telegram, ya sea la lista de tareas pendientes o un mensaje indicativo de que no hay tareas.
						14	El desarrollador recibe y revisa la información enviada por el ChatBot.	El desarrollador recibe la información enviada por el ChatBot y la revisa para asegurarse de que

								sea correcta y completa.
							El desarrollador cierra la consulta con un comando de cierre o agradecimiento, y el ChatBot responde de manera apropiada confirmando la finalización de la interacción.	
						15	El desarrollador cierra la consulta con un comando de cierre o agradecimiento, y el ChatBot responde de manera apropiada.	
7	El ChatBot añade correctamente tareas que le da el desarrollador	Este caso de prueba evalúa la funcionalidad del ChatBot para responder adecuadamente a la solicitud de un desarrollador de añadir tareas. Se verifica que el ChatBot guarde completa la tarea del desarrollador, incluyendo detalles como la descripción de la tarea, la fecha de vencimiento y el estado actual.	"El ChatBot está activo y funcionando correctamente. El desarrollador ha iniciado una conversación con el ChatBot a través de Telegram. El desarrollador tiene	Después de ejecutar con éxito el caso de prueba, se espera que el ChatBot guarde correctamente la tarea. La información mostrada debe reflejar fielmente las tareas almacenadas en la base de datos del sistema. Además, la respuesta del ChatBot debe ser clara y fácil	A	DV	1	El ChatBot conecta de manera exitosa a Telegram para su uso

		La respuesta del ChatBot debe dar un mensaje de guardado al usuario.	tareas que quiere registrar."	de entender para el usuario, proporcionando detalles como la descripción de la tarea, la fecha de vencimiento y el estado actual de cada tarea.					
8							El desarrollador envía el comando "/crear tarea" al ChatBot.	Se confirma que el sistema y el ChatBot están activos y respondiendo a las solicitudes de los usuarios sin errores aparentes.	
9							El ChatBot recibe el comando y procesa la solicitud.	El ChatBot conecta correctamente la solicitud por Telegram	
10							El ChatBot pide al usuario ingresar el nombre de la tarea	El ChatBot despliega correctamente el espacio para que el usuario escriba el	

								nombre de la tarea
1 1						5	El ChatBot pide al usuario ingresar una descripción de la tarea	El ChatBot despliega correctamente el espacio para que el usuario escriba la descripción de la tarea
1 2						6	El ChatBot espera que el usuario ingrese los datos	El ChatBot espera los datos del usuario
1 3						7	El ChatBot procesa la solicitud para entender la consulta de tareas pendientes.	El ChatBot procesa la información del usuario
1 4						8	El ChatBot envía la solicitud al servidor	El ChatBot procesa la información al servidor correctamente
						9	El servidor procesa la información	El servidor procesa la información correctamente,

								recibida del ChatBot
						10	El servidor guarda la información ingresada y procesa	El servidor guarda correctamente la información recibida
						11	El ChatBot regresa al menu de creación de tareas	El ChatBot vuelve a pedir información al usuario para ingresar una tarea
						12	El usuario presiona la solicitud de volver al menu principal	El usuario al presionar la opción de menu es capaz de regresar al menu de manera exitosa
						13	El ChatBot lo redirige al menú principal y vuelve a pedir comando de acción	El ChatBot redirige al usuario a la pantalla principal donde aguarda el siguiente comando

								del usuario
						14	El usuario presiona la solicitud para visualizar las tareas creadas	El ChatBot recibe el comando y redirige al usuario a la pantalla de tareas creadas
						15	La tarea creada aparece en la pantalla	El ChatBot y el servidor son capaces de mostrar las tareas que han sido creadas recientemente por el usuario
15	El ChatBot responde correctamente a una solicitud de consulta de tareas pendientes de todo el proyecto a una cuenta Manager	Este caso de prueba evalúa la funcionalidad del ChatBot para responder adecuadamente a la solicitud de un manager de consultar todas las tareas pendientes del proyecto. Se verifica que el ChatBot presente una lista precisa y completa de las tareas pendientes de todos los	El ChatBot está activo y funcionando correctamente. El desarrollador ha iniciado una conversación con el ChatBot a través de Telegram.	Se espera que el ChatBot presente al usuario una lista precisa y completa de todas las tareas pendientes. La información mostrada debe reflejar fielmente las tareas almacenadas en la base de datos del sistema. Además, la respuesta del	A	DV	1	Confirmar que el ChatBot esté activo y funcionando correctamente. El sistema muestra que el ChatBot está operativo.

		desarrolladores, incluyendo detalles como la descripción de la tarea, la fecha de vencimiento y el estado actual. La respuesta del ChatBot debe ser clara y fácil de entender para el usuario, y debe manejar correctamente cualquier escenario donde no haya tareas pendientes para el usuario.	El manager puede consultar tareas pendientes de los desarrolladores registrados en el sistema.	ChatBot debe ser clara y fácil de entender para el usuario, proporcionando detalles como la descripción de la tarea, la fecha de vencimiento y el estado actual de cada tarea. En caso de no haber tareas pendientes para el usuario, el ChatBot debe informar claramente que no hay tareas pendientes disponibles.					
16							El manager abre Telegram y accede a su cuenta si es necesario.	El manager accede exitosamente a su cuenta de Telegram.	
17							El manager busca y accede al chat del ChatBot.	El manager localiza y abre el chat con el ChatBot sin problemas.	
18							Inicio de la conversación con un	El ChatBot responde	

19							saludo o comando inicial al ChatBot.	al saludo, demostrando su disponibilidad para recibir comandos.
							El ChatBot responde al saludo, indicando su disponibilidad para recibir 5 comandos.	El ChatBot muestra un mensaje estándar de bienvenida o ayuda, indicando que está listo para asistir al manager.
							El manager envía el comando "/consultar tareas pendientes del proyecto" al ChatBot.	El ChatBot reconoce y confirma la recepción del comando específico para la consulta de tareas pendientes del proyecto.
							El ChatBot confirma la recepción del comando y solicita confirmación para proceder 7	El ChatBot solicita confirmación para asegurar que la solicitud del

							con la consulta.	manager es proceder con la consulta de tareas pendientes.
						8	El manager confirma la solicitud de consulta de tareas pendientes.	El ChatBot recibe la confirmación y procede con la consulta en la base de datos.
						9	El ChatBot procesa el comando para entender la consulta de tareas pendientes del proyecto completo.	El ChatBot procesa correctamente el comando sin errores, entendiendo la solicitud completa.
						10	El ChatBot accede a la base de datos para buscar todas las tareas pendientes asignadas a los desarrolladores del proyecto.	El ChatBot accede a la base de datos y comienza la búsqueda de todas las tareas pendientes eficazmente.

						11	El ChatBot verifica y recopila la información de tareas pendientes de todos los desarrolladores.	El ChatBot recopila y verifica la información, asegurando que todas las tareas pendientes son incluidas.
						12	El ChatBot genera una lista detallada de las tareas pendientes, incluyendo descripción, fechas de vencimiento y estado actual.	El ChatBot crea una lista detallada y organizada que refleja fielmente el estado actual de las tareas en la base de datos.
						13	El ChatBot envía la lista de tareas pendientes al manager a través de Telegram.	El manager recibe la lista completa de tareas pendientes a través de un mensaje claro y bien formateado en Telegram.

						14	El manager revisa la lista de tareas pendientes recibida.	El manager comprende y verifica la información recibida, encontrando que coincide con las expectativas y requerimientos del proyecto.
						15	El manager cierra la conversación con un comando de cierre o realiza una nueva solicitud.	El ChatBot responde adecuadamente al comando de cierre o procesa la nueva solicitud del manager sin errores.
La aplicación web responde correctamente a una solicitud de inicio de sesión mostrando la pestaña adecuada a la sesión solicitada con sus	El caso de prueba evalúa la capacidad de la aplicación web de verificar las credenciales proporcionadas por el usuario e iniciar la sesión adecuada junto sus implicaciones y datos almacenados respectivamente a cada cuenta o usuario	La aplicación web está activa y accesible al usuario en cuestión, el usuario tiene cuenta con una cuenta en la base de datos del proyecto junto a su información	Se espera que la aplicación web de acceso a la pestaña de la cuenta adecuada al usuario que intenta acceder, verificando las credenciales proporcionadas por el usuario y validando su autenticidad	A	DV	1	El usuario ingresa a la aplicación web a través de un navegador con conexión a internet.	El usuario accede exitosamente a la aplicación web y se carga la página principal.

	respectivas funcionalidades		n base, y si es el caso informacion respecto a sus tareas, el usuario tiene capacidad de utilizar sus credenciales para ser verificadas por el sistema y dando posible su posterior ingreso					
20							El usuario introduce su nombre de usuario.	El nombre de usuario del usuario se registra correctamente en el campo correspondiente.
21							El usuario introduce su contraseña.	La contraseña del usuario se ingresa correctamente en el campo correspondiente y permanece encriptada.

2 2							4	Se intenta acceder a la cuenta del usuario en cuestión validando los datos enviados.	Se verifica la autenticidad de las credenciales del usuario y se detecta si son válidas o no.
2 3							5	Se evalúan y aceptan las credenciales del usuario.	Las credenciales del usuario son aceptadas correctamente por el sistema y se procede al siguiente paso.
2 4							6	Se dirige a la página de usuario.	La página de usuario se carga correctamente, mostrando la información y las opciones pertinentes para el usuario.
2 5							7	El sistema manda una solicitud para visualizar las tareas del usuario	La solicitud para visualizar las tareas del usuario se

							de su respectiva cuenta.	envía correctamente al servidor.
26						8	El servidor responde a la solicitud enviada por parte de la aplicación y del usuario para mostrar en pantalla las tareas pendientes del desarrollador enlazado a la cuenta activa.	El servidor responde a la solicitud enviada por la aplicación y muestra las tareas pendientes del usuario.
27						9	Se visualiza el dashboard de tareas pendientes, si es que las hay, respectivas al usuario.	El dashboard de tareas pendientes se carga correctamente, mostrando todas las tareas relevantes del usuario.
28						10	El usuario puede revisar y gestionar sus tareas pendientes.	El usuario puede ver todas sus tareas pendientes de manera clara y organizada.

30						11	El usuario decide finalizar su sesión.	El usuario activa el proceso de cierre de sesión.
31						12	Se solicita al servidor que cierre la sesión del usuario.	Se solicita al servidor que cierre la sesión del usuario de manera segura.
32						13	El servidor procesa la solicitud de cierre de sesión.	El servidor procesa la solicitud de cierre de sesión y finaliza la sesión del usuario de manera adecuada.
33						14	Se muestra un mensaje de confirmación de cierre de sesión al usuario.	Se muestra un mensaje de confirmación al usuario, indicando que la sesión se ha cerrado exitosamente.

34							El usuario es redirigido a la página de inicio de sesión, asegurando que la sesión se haya cerrado correctamente y esté listo para iniciar sesión nuevamente si es necesario.	
35	El ChatBot elimina correctamente tareas que le indique el desarrollador	Este caso de prueba se centra en verificar la funcionalidad de eliminación de tareas existentes en el ChatBot. El objetivo principal es garantizar que el ChatBot permita a los usuarios eliminar una tarea seleccionada de manera precisa y sin errores. Para lograr esto, el usuario iniciará una conversación con el ChatBot a través de Telegram y seleccionará la	Antes de ejecutar este caso de prueba, se debe garantizar que el ChatBot esté activo y disponible para su uso. Además, se requiere que el usuario haya iniciado una conversac ión con el ChatBot a través de la plataforma	Al finalizar la ejecución de este caso de prueba, se espera que la tarea seleccionada haya sido eliminada con éxito de la base de datos del sistema. Además, se verificará que el ChatBot haya confirmado la eliminación exitosa al usuario y que la tarea eliminada ya no esté presente en la lista de tareas mostrada por el ChatBot. La postcondición	A	DV	15	El usuario es redirigido a la página de inicio de sesión para iniciar sesión nuevamente si es necesario.
							El usuario inicia una conversaci ón con el ChatBot a través de Telegram.	El ChatBot saluda al usuario y muestra un menú de opciones disponibles.

	<p>opción "Gestionar Tareas" del menú proporcionado. Luego, el ChatBot mostrará una lista de tareas existentes y solicitará al usuario que elija la tarea que desea eliminar. Una vez seleccionada la tarea, se solicitará al usuario una confirmación antes de proceder con la eliminación. Después de confirmar la eliminación, el ChatBot procederá a eliminar la tarea de la base de datos y confirmará la acción al usuario. Se verificará que la tarea eliminada ya no esté presente en la lista de tareas mostrada por el ChatBot y que tampoco esté presente en la base de datos del sistema. Este caso de prueba asegura que el proceso de eliminación de tareas sea transparente y efectivo,</p>	<p>de Telegram. Es fundamental al que exista al menos una tarea previamente creada en el sistema, ya que el objetivo de este caso de prueba es verificar la capacidad del ChatBot para eliminar tareas existentes de manera precisa y sin errores. Estas condiciones previas son necesarias para llevar a cabo la prueba de forma efectiva y garantizar resultados precisos.</p>	<p>también incluye la validación de que la tarea eliminada ya no esté presente en la base de datos del sistema, confirmando así que la operación de eliminación se ha completado satisfactoriamente y que el sistema se encuentra en un estado coherente y consistente.</p>					
--	--	--	---	--	--	--	--	--

		brindando una experiencia sin errores para el usuario.						
3 6							El ChatBot saluda al usuario y muestra un menú de opciones disponibles para la gestión de tareas. 2	El ChatBot muestra un menú de opciones claras y legibles para la gestión de tareas.
3 7							El usuario selecciona la opción "Gestionar Tareas" del menú. 3	El ChatBot muestra una lista de tareas existentes y solicita al usuario que seleccione una para eliminar.
3 8							El ChatBot muestra una lista de tareas existentes y solicita al usuario que seleccione la tarea que desea eliminar. 4	El ChatBot muestra una lista de tareas existentes con la opción de eliminar una tarea seleccionada.
3 9							El usuario revisa la lista de tareas y selecciona 5	El ChatBot confirma la selección

							la tarea específica que desea eliminar.	de la tarea y solicita una confirmación de eliminación.
40							El ChatBot muestra una confirmación de eliminación de la tarea seleccionada y solicita al usuario que confirme la acción.	El ChatBot muestra un mensaje claro solicitando al usuario que confirme la eliminación de la tarea seleccionada.
41							El usuario confirma la eliminación de la tarea.	El ChatBot procede a eliminar la tarea de la base de datos y muestra una confirmación de eliminación exitosa.
42							El ChatBot elimina la tarea seleccionada de la base de datos y confirma la eliminación.	El ChatBot confirma la eliminación exitosa de la tarea y proporciona

							exitosa al usuario.	na un mensaje claro al usuario.
43						9	El usuario recibe la confirmación de que la tarea ha sido eliminada correctamente.	El usuario recibe un mensaje de confirmación indicando que la tarea ha sido eliminada correctamente.
44						10	El usuario verifica que la tarea eliminada ya no esté presente en la lista de tareas mostrada por el ChatBot.	El usuario verifica que la tarea eliminada ya no esté presente en la lista de tareas mostrada por el ChatBot.
45						11	El usuario decide revisar nuevamente la lista de tareas para confirmar que la tarea eliminada no esté presente.	El ChatBot muestra la lista actualizada de tareas sin incluir la tarea eliminada.
46						12	El ChatBot muestra la lista actualizada de tareas,	El ChatBot muestra la lista actualizada

							sin incluir la tarea previamente eliminada.	a de tareas sin la tarea eliminada, confirmando que la eliminación fue exitosa.
47						13	El usuario verifica que la tarea eliminada no esté presente en la lista y confirma que la eliminación fue exitosa.	El usuario verifica que la tarea eliminada no esté presente en la lista de tareas y confirma que la eliminación fue exitosa.
48						14	El usuario cierra la gestión de tareas y finaliza la interacción con el ChatBot.	El usuario finaliza la gestión de tareas y la interacción con el ChatBot de manera satisfactoria.
49						15	El usuario verifica que la tarea eliminada ya no esté presente en la base de datos del sistema.	El usuario verifica que la tarea eliminada ya no esté presente en la base de datos del sistema, validando así que la

									eliminación se haya completado correctamente.
50	El chatbot marca como completa las tareas que le indique el desarrollador	Este caso de prueba se centra en verificar la funcionalidad del ChatBot para marcar como completadas las tareas indicadas por el desarrollador. El objetivo principal es garantizar que el ChatBot pueda gestionar correctamente el estado de las tareas, marcándolas como completadas según la solicitud del usuario.	El ChatBot está activo y disponible para su uso. El usuario ha iniciado una conversación con el ChatBot a través de la plataforma de Telegram. Existen tareas previamente creadas en el sistema.	Las tareas seleccionadas han sido marcadas como completadas en la base de datos del sistema. El usuario verifica que las tareas seleccionadas estén marcadas como completadas en la lista de tareas mostrada por el ChatBot. El estado de las tareas en la base de datos se ha actualizado correctamente para reflejar los cambios realizados por el usuario.	A	DV	1	Iniciar una conversación con el ChatBot a través de Telegram.	El ChatBot muestra un menú de opciones claras y legibles para la gestión de tareas.
51							2	Poner e comando "/tareas pendientes" del ChatBot.	El ChatBot muestra una lista de tareas existentes con la opción de marcar como completada a una

								tarea seleccionada.
52						3	Revisar la lista de tareas existentes mostrada por el ChatBot.	El ChatBot marca la tarea seleccionada como completada y proporciona un mensaje claro al usuario.
53						4	Seleccionar una tarea específica para marcar como completada.	La tarea seleccionada está marcada como completada en la lista de tareas mostrada por el ChatBot.
54						5	Verificar que la tarea seleccionada esté marcada como completada.	La tarea marcada como completada no aparece en la lista de tareas con opción de marcar como completada.
55						6	Verificar que la tarea marcada	La nueva tarea seleccion

							como completada no esté disponible para marcar nuevamente .	ada está marcada como completada en la lista de tareas mostrada por el ChatBot.
56							7 Seleccionar otra tarea de la lista para marcar como completada.	La nueva tarea seleccionada está marcada como completada en la lista de tareas mostrada por el ChatBot.
57							8 Verificar que la nueva tarea seleccionada esté marcada como completada.	La nueva tarea marcada como completada no aparece en la lista de tareas con opción de marcar como completada.
58							9 Verificar que la nueva tarea marcada como completada no esté disponible para marcar	El ChatBot informa al usuario que la tarea ya está completada y no

							nuevamente .	puede ser marcada nuevamente.
59						10	Intentar marcar como completada una tarea que ya está marcada como completada.	La tarea sigue estando marcada como completada en la lista de tareas mostrada por el ChatBot.
60						11	Verificar que la tarea que se intentó marcar como completada siga estando marcada como completada.	El ChatBot informa al usuario que la tarea seleccionada no existe.
61						12	Intentar marcar como completada una tarea que no existe en la lista de tareas.	La lista de tareas mostrada por el ChatBot sigue incluyendo o las tareas disponibles para marcar como completadas.

6 2						13	<p>Verificar que la lista de tareas mostrada por el ChatBot incluya las tareas disponibles para marcar como completadas.</p>	<p>Todas las tareas restantes en la lista son marcadas como completadas por el ChatBot.</p>
6 3						14	<p>Marcar como completadas todas las tareas restantes en la lista.</p>	<p>Todas las tareas restantes en la lista están marcadas como completadas en la lista de tareas mostrada por el ChatBot.</p>
6 4						15	<p>Verificar que todas las tareas restantes en la lista estén marcadas como completadas y luego cerrar la gestión de tareas y finalizar la interacción con el ChatBot.</p>	<p>El usuario finaliza la gestión de tareas y la interacción con el ChatBot de manera satisfactoria.</p>

12.2 Anexo B

Detalles del entorno de pruebas

Hardware:

- **Servidor:**
 - **Proveedor:** Oracle Cloud
 - **Dirección IP:** 159.54.145.9
- **Estaciones de Trabajo:**
 - **Modelo:** Macbook Air M2
 - **CPU:** Apple M2 chip
 - **RAM:** 16 GB
 - **Almacenamiento:** 512 GB SSD
 - **Sistema Operativo:** MacOS

Software:

- **Servidor:**
 - **Sistema Operativo:** Oracle Linux Extended Support
 - **Servidor de Aplicaciones:** Apache Tomcat 9.0
 - **Base de Datos:** Oracle Database 19c
 - **Framework:** Spring Boot 2.5
- **Estaciones de Trabajo:**
 - **IDE:** VS Code
 - **Lenguaje de Programación:** Java 11
 - **Herramientas de Pruebas:** Selenium, JUnit, Postman
 - **Navegadores:** Google Chrome, Arc

Datos de Prueba:

- **Usuarios de Prueba:**
 - **Usuario 1:**
 - **Nombre de usuario:** user_test1
 - **Contraseña:** password1
 - **Usuario 2:**
 - **Nombre de usuario:** user_test2
 - **Contraseña:** password2
- **Tareas de Prueba:**
 - **Tarea 1:**
 - **Descripción:** "Revisar documentación del proyecto"
 - **Estado:** "Pendiente"

- **Tarea 2:**
 - **Descripción:** "Desarrollar módulo de autenticación"
 - **Estado:** "Hecho"

Configuraciones de Red:

- **Firewall:**
 - **Puertos Abiertos:** 80 (HTTP), 443 (HTTPS), 22 (SSH)
 - **Reglas de Seguridad:** Solo accesible desde la red interna y direcciones IP autorizadas

Procedimientos de Respaldo:

- **Frecuencia:** Diaria
 - **Hora:** 00:00 AM
- **Almacenamiento de Respaldo:**
 - **Ubicación:** Servidor de Respaldo en Oracle Cloud
 - **Retención:** 30 días

Consideraciones de Seguridad:

- **Autenticación:**
 - **Método:** OAuth2
 - **Proveedores de Autenticación:** Google, Microsoft
- **Cifrado:**
 - **En tránsito:** TLS 1.2
 - **En reposo:** AES-256

13. Referencias

1. Equipo de Desarrollo de OracleBot. (2024). *Plan de Proyecto ChatBot Oracle*.
2. Equipo de Desarrollo de OracleBot. (2024). *Plan de Administración de Configuración de OracleBot*.
3. Telegram. (n.d.). *Telegram Bot API Documentation*. Recuperado de <https://core.telegram.org/bots/api>
4. Spring Team. (n.d.). *Spring Boot Reference Documentation*. Recuperado de <https://docs.spring.io/spring-boot/docs/current/reference/html/>
5. SeleniumHQ. (n.d.). *Selenium WebDriver Documentation*. Recuperado de <https://www.selenium.dev/documentation/>
6. Oracle. (n.d.). *Oracle Database 19c Documentation*. Recuperado de <https://docs.oracle.com/en/database/oracle/oracle-database/19/index.html>
7. JUnit Team. (n.d.). *JUnit 5 User Guide*. Recuperado de <https://junit.org/junit5/docs/current/user-guide/>