



Tecnológico de Monterrey

Desarrollo e implantación de sistemas de software

Grupo 106

Plan de Proyecto

Profesores:

Adan Octavio Ruiz Martinez

Jorge Alvarez Bujanos

Leonardo S. Gamez Peña

Lorena Guadalupe Gómez Martínez

Alberto Emmanuel Benavides Contreras

Equipo 6:

Eugenia Uresti Arriaga A01284839

Mónica Parra Franco A01251941

Gerardo Manzur Morales A01177622

Jorge Nuñez Gurrola A00833455

Javier Banegas A00827812

13 de junio de 2024

1. Introducción.....	3
1.1 Visión del proyecto.....	3
2. Descripción del proyecto.....	3
2.1 Contexto y Antecedentes.....	3
2.2 Objetivo.....	4
2.3 Análisis Costo Beneficio.....	4
2.3.1 Impacto Económico.....	4
2.3.2 Impacto Cuantitativo.....	5
3. Metodología de Gestión de proyectos.....	6
3.1 Métricas.....	8
4. Organización del proyecto.....	9
4.1 Roles del equipo y funciones.....	9
4.2 Identificación de stakeholders y su función en el proyecto.....	9
5. WBS/Backlog.....	9
5.2 Definición de Hecho.....	10
5.3 Inclusión de Actividades Relacionadas a CI/CD.....	10
6. Plan de comunicación.....	11
7. Plan de riesgos.....	12
7.1 Registro de riesgos.....	12
7.2 Matriz de riesgos.....	14
8. Manejo de Cronograma.....	14
8.1 Calendario de ceremonias SCRUM.....	14
8.2 Calendario de EPIC.....	15
9. Plan de Administración de Configuración.....	15
9.1 CMDB.....	15
11. Referencias.....	17

1. Introducción

En Oracle, la constante búsqueda de eficiencia y optimización en el desarrollo de software ha llevado a la identificación de áreas clave donde la automatización y la mejora de procesos pueden tener un impacto significativo. El desarrollo e implementación del Oracle Java DevOps ChatBot surge como respuesta a la necesidad crítica de mejorar la gestión de tareas y la comunicación entre los equipos de desarrollo y operaciones, asegurando que el tiempo y recursos de Oracle se utilicen de la manera más efectiva posible.

1.1 Visión del proyecto

La visión de este proyecto “Oracle Java DevOps ChatBot” se centra en optimizar el proceso de desarrollo de software en Oracle mediante la implementación de herramientas de DevOps. Con el objetivo de lograr un flujo rápido de trabajo, sin comprometer la confiabilidad, estabilidad, resiliencia y seguridad del entorno, se propone la automatización de tareas, a través de un ChatBot basado en Oracle Java para mejorar en un 20% la productividad y visibilidad del equipo de desarrollo.

Dentro de esta herramienta, se ofrecen diferentes funcionalidades clave, destacando la gestión eficiente de tareas. Tanto los desarrolladores como los managers tendrán la oportunidad de agregar o eliminar tareas según las necesidades del proyecto. Además podrán visualizar las tareas, asegurando una comprensión clara y detallada de las asignaciones y el progreso, tanto para los desarrolladores individualmente como para los managers que supervisan al equipo completo.

Para garantizar que este proyecto cumpla con los estándares establecidos por el Chief Transformation Officer (CTO) de Oracle, se integran tecnologías de última generación, tales como Oracle Cloud Infrastructure, Autonomous Database, Kubernetes, Docker, Java, Microservices, Spring Boot y API Gateway.

2. Descripción del proyecto

2.1 Contexto y Antecedentes

En el marco de los esfuerzos continuos de Oracle para optimizar las operaciones de desarrollo de software, se identificó una oportunidad significativa para mejorar la eficiencia mediante la automatización de la gestión de tareas y la comunicación. Los desarrolladores a menudo enfrentan interrupciones que reducen su tiempo efectivo de codificación, afectando

directamente la productividad y extendiendo los plazos de entrega de proyectos. El Oracle Java DevOps ChatBot se concibe como una solución a este problema estructural, destinado a facilitar una gestión ágil y automática de las tareas diarias.

2.2 Objetivo

El objetivo principal del Oracle Java DevOps ChatBot es incrementar la eficiencia operativa del desarrollo de software en Oracle, reduciendo las interrupciones y optimizando el uso del tiempo de los desarrolladores. Se espera que esta herramienta permita a los equipos concentrarse más en la codificación y menos en la gestión administrativa, mejorando así el tiempo de comercialización de nuevas soluciones y tecnologías.

2.3 Análisis Costo Beneficio

2.3.1 Impacto Económico

Un estudio realizado por Software.com investigó cuánto tiempo los desarrolladores de su comunidad lograban dedicar efectivamente a la programación diariamente. Tras encuestar a más de 250,000 desarrolladores, descubrieron que, en promedio, solo lograban programar una hora al día. Identificaron que interrupciones como roadblocks, reuniones constantes y otras distracciones limitaban severamente su tiempo productivo. Mason Mcleod, CTO de Software.com, destacó: “Code time is often undervalued, continually interrupted, and almost wholly unmeasured”. El Java Oracle Bot está diseñado para liberar a los desarrolladores de estas distracciones, permitiéndoles concentrarse en la programación.

El salario promedio anual de un desarrollador recién egresado en Oracle es de \$50,000 USD. Con 2,700 desarrolladores, esto suma un costo laboral total de \$135,000,000 USD anualmente. Actualmente, cada desarrollador resuelve 15 tickets de JIRA al mes. Cada desarrollador dedica aproximadamente 2 horas diarias a tareas de programación. Nuestro objetivo es aumentar esta productividad en un 20%, lo que efectivamente incrementa el tiempo de programación a 2.4 horas por día. Este aumento permitirá resolver efectivamente 0.69 tickets adicionales por mes debido a las 8 horas adicionales de programación generadas por un incremento del 20% en su eficiencia diaria durante su jornada laboral. Esto implica que el Java Oracle Bot podría ayudar a Oracle a resolver 1863 tickets adicionales cada mes.

Incremento propuesto del 20%

$$\text{Incremento} = 2 \text{ horas} \times \frac{20}{100} = 0.4 \text{ horas}$$

$$\text{Tiempo nuevo} = 2 \text{ horas} + 0.4 \text{ horas} = 2.4 \text{ horas}$$

Horas adicionales por mes por desarrollo

$$0.4/\text{día} \times 20 \text{ días laborales} = 8 \text{ horas adicionales al mes}$$

Incremento estimado de tickets resueltos por desarrollador por mes debido al tiempo adicional

$$11.55 \text{ horas por ticket} \div 8 \text{ horas adicionales al mes} = 0.69 \text{ tickets nuevos}$$

Incremento de tickets resueltos por mes a nivel empresa

$$2,700 \text{ desarrolladores} \times 0.69 \text{ tickets adicionales} = 1,863 \text{ tickets adicionales}$$

Impacto Financiero Estimado

Si consideramos el costo por ticket basado en el salario promedio, donde cada ticket adicional tiene un valor directo de \$277.66, el valor adicional generado por los 1863 tickets adicionales al mes es de aproximadamente \$517,280.58 mensuales o \$6,207,366.96 anualmente.

Incremento de tickets resueltos al año a nivel empresa

$$1863 \text{ tickets nuevos} \times 12 \text{ meses} = 24300 \text{ tickets}$$

Valor monetario generado por los nuevos tickets resueltos al año

$$22356 \text{ tickets nuevos} \times 277.66 \text{ USD} = \$6,207,366.96 \text{ USD}$$

2.3.2 Impacto Cuantitativo

Desarrolladores Activos: 2,700

Equipos Totales: Más de 250

Equipo Piloto: 50 miembros

Si cada uno de estos 50 miembros del equipo piloto mejora su eficiencia en un 20%, esto se traduce en un incremento neto de 400 horas de desarrollo adicionales al mes. Estas horas son el resultado de una mejor eficiencia durante el horario laboral regular y no implican horas extras.

Incremento neto de horas de desarrollo adicionales al mes con el equipo piloto

$$50 \text{ miembros} \times 8 \text{ horas adicionales} = 400 \text{ horas}$$

Equipo Piloto

El equipo piloto servirá para medir el impacto directo al Java Oracle bot sobre un grupo controlado, ofreciendo datos para decisiones de implementación. La eficiencia del bot se evaluará no solo por la cantidad de realizado sino también por la calidad y la reducción de tiempo de interrupción.

Costo Por los Softwares Utilizados

Para los softwares que se utilizaron en este desarrollo usamos OCI que fue proporcionado por Oracle y Azure DevOps. En el caso de Azure, nosotros como somos 5 integrantes, el costo es gratuito para los primeros 5 usuarios. En el caso que esta no sea la situación (que sean más de 5), Azure DevOps estará cobrando adicionalmente \$6 dólares mensuales por cada miembro.

Mejora en los tickets JIRA

Gracias al Chatbot se podrán hacer análisis continuos para así poder lograr un incremento en la realización de los tickets de JIRA. Esto lo podremos ver ya que el chatbot tiene la función de poner la fecha en la cual una tarea fue creada. Esto podrá ser monitoreado y al hacer estos análisis continuos, con reportes y métricas, se podrá ver la cantidad de tiempo que este se tomó y así hacer el esfuerzo para poder hacer ese incremento.

Este análisis muestra cómo la implementación del Java Oracle Bot no solo mejoraría la eficiencia y productividad de los desarrolladores, sino que también justifica económicamente la inversión en el desarrollo del chatbot debido al valor significativo que podría agregar a la empresa en términos de horas productivas y reducción de costos operativos.

3. Metodología de Gestión de proyectos

En este proyecto, se está utilizando la metodología Agile, específicamente SCRUM. Esta metodología se enfoca en la entrega incremental e iterativa de productos a través de sprints de una semana de duración. Al inicio de cada sprint, se lleva a cabo una reunión de planificación para definir los objetivos y tareas a completar durante el sprint. En nuestro calendario, se programan dos reuniones de planificación al inicio del sprint: una con el socio formador y otra con el equipo de desarrollo.

Durante el sprint se realizan reuniones diarias de 15 minutos donde el equipo revisa el progreso, identifica obstáculos y planifica el trabajo del día, asegurando que se avance

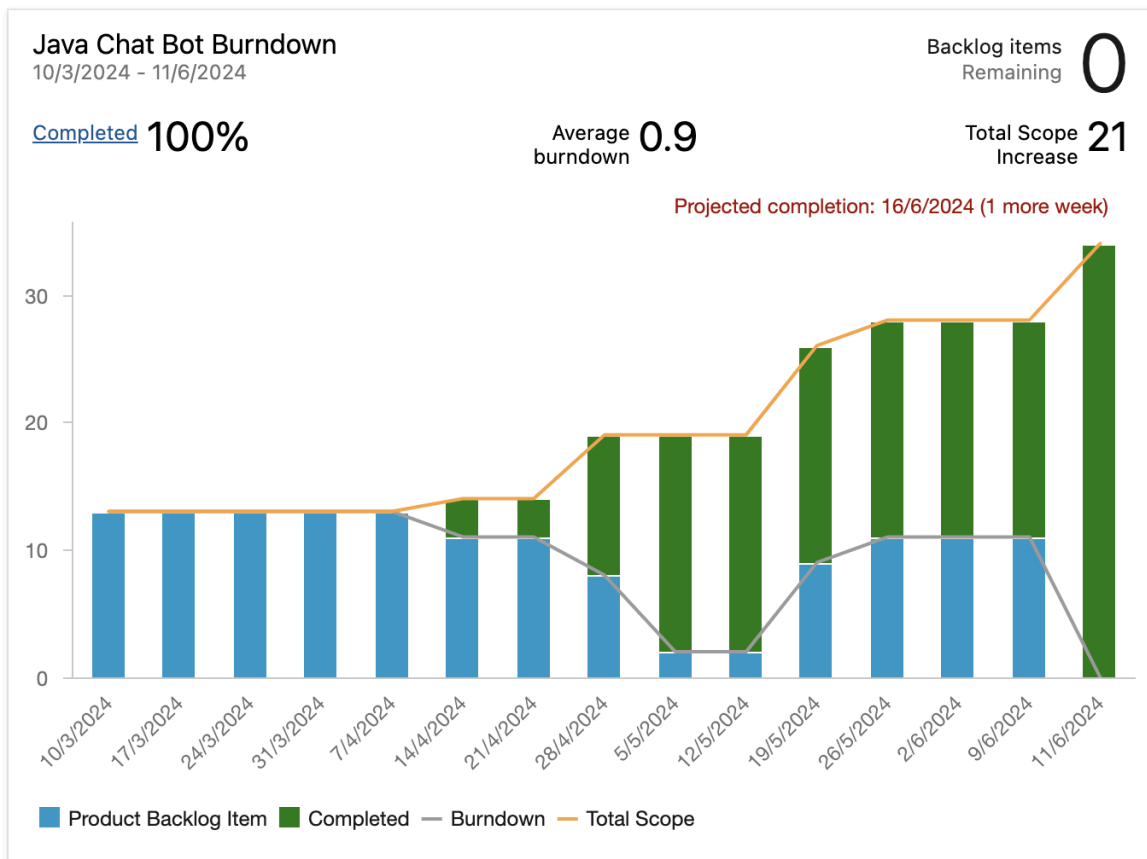
constantemente hacia los objetivos del sprint. Para esto se ha programado una reunión adicional junto con el socio formador para proporcionar una actualización sobre el progreso del proyecto.

Al final del sprint se lleva a cabo una revisión donde se evalúa el trabajo completado, se demuestra el progreso, se recibe retroalimentación y se discute lo que funcionó bien, lo que no y como se pueden hacer mejoras para el próximo sprint.

Dentro de esta metodología existen diferentes roles, estos incluyen al product owner, responsable de maximizar el valor del producto y gestionar el backlog del producto; el scrum master, que facilita el proceso scrum, ayuda a eliminar impedimentos y asegura que el trabajo siga las prácticas scrum; y el equipo de desarrollo, que trabaja en las tareas y entrega de los incrementos del producto final de cada sprint.

Se ha optado por utilizar esta metodología debido a los beneficios que ofrece, tales como la flexibilidad y adaptabilidad, mejora continua, transparencia y comunicación. Estos beneficios permiten realizar cambios rápidos en respuesta a nuevos requerimientos y fomentan un entorno de trabajo colaborativo y transparente.

3.1 Métricas



Para medir el rendimiento del equipo y asegurar el éxito del proyecto, se utilizará la métrica de un burndown chart, esta es una herramienta que muestra el progreso del equipo y traza la cantidad de trabajo restante en el sprint a lo largo del tiempo. En el eje vertical se muestra el trabajo restante, medido en tareas, mientras que el eje horizontal representa los días del sprint. Esta se utiliza diariamente durante el sprint para monitorear el progreso y poder ajustar el plan si es necesario.

Este gráfico muestra el progreso del proyecto Java Chat Bot desde el 10 de marzo de 2024 hasta el 11 de junio de 2024. Las barras azules representan la cantidad de trabajo pendiente, mientras que las barras verdes indican el trabajo completado en cada periodo. La línea naranja muestra el ritmo ideal de trabajo necesario para completar el proyecto según lo planeado. La línea gris representa la tendencia de trabajo real, indicando cómo el equipo ha avanzado en comparación con el plan inicial. A pesar de los aumentos en el alcance total del proyecto, que han ampliado la carga de trabajo prevista (reflejado en las elevaciones de la línea de tendencia ideal), el equipo ha logrado completar el 100% de las tareas planificadas. La fecha de finalización proyectada se ha ajustado ahora al 16 de junio de 2024, extendiendo el proyecto solo una semana más de lo planeado inicialmente debido a este aumento en el alcance.

4. Organización del proyecto

4.1 Roles del equipo y funciones

Product Owner/Backend - Monica Parra Franco

Scrum Master/Base de datos - Eugenia Uresti Arriaga

Desarrollador de Software/DevOps - Jorge Nuñez Gurrola

Desarrollador de Software/Tester - Gerardo Manzur Morales

Desarrollador de Software/Tester - Javier Banegas Moreno

4.2 Identificación de stakeholders y su función en el proyecto

Equipo de Desarrollo: Responsable del diseño, desarrollo, pruebas y despliegue del chatbot. Este grupo trabaja en las tareas técnicas diarias y asegura la calidad del software.

Project Manager: Gestiona el proyecto en su totalidad, asegurando la coordinación entre los miembros del equipo, el cumplimiento de los plazos y la comunicación efectiva con todos los stakeholders.

SCRUM Master: Apoya al equipo en la implementación de las metodologías Agile, facilitando los sprints, y ayudando a resolver los obstáculos que afectan el progreso del proyecto.

Managers de Producto de Oracle: Definen los requisitos del proyecto y aseguran que las entregas del ChatBot se alineen con las necesidades del negocio y las expectativas del mercado.

Usuarios Finales: Compuestos por los desarrolladores de Oracle que utilizarán el chatbot. La retroalimentación de este grupo es vital para la iteración del producto y la mejora continua de la experiencia del usuario.

Equipo de Soporte Técnico: Encargado del soporte post implementación, esencial para la resolución de problemas técnicos y la optimización continua del chatbot en el entorno de producción.

Equipo de Seguridad de TI: Evalúa y mitiga los riesgos asociados con la seguridad de los datos y el cumplimiento de las políticas de seguridad de la información de Oracle.

5. WBS/Backlog

El backlog está organizado en varias epics que abarcan los componentes esenciales del desarrollo del proyecto y actividades de clase:

- **Epic: Desarrollo de la Base de Datos**
 - Features:

- Diseño UML de la Base de Datos
 - Creación y revisión de la base de datos
 - Desarrollo de triggers y procedimientos almacenados
- **Epic: Configuración del Entorno de Desarrollo**
 - Features:
 - Configuración del entorno local y de desarrollo
 - Implementación de servicios en Docker
 - Configuración de Kubernetes para despliegue
- **Epic: Desarrollo de Funcionalidades del Chatbot**
 - Features:
 - Desarrollo del Frontend - Telegram Bot
 - Desarrollo del Backend - Java/Spring Boot
 - Integración de microservicios con bases de datos
- **Epic: Gestión de Proyecto y Metodología DevOps**
 - Features:
 - Curso y aplicación de metodologías DevOps
 - Integración y manejo de versiones con GitHub
 - Implementación de pruebas automáticas y manejo de configuración
- **Epic: Actividades Complementarias**
 - Features:
 - Preparación de la presentación final
 - Documentación técnica y manual de usuario
 - Revisión de seguridad y cumplimiento de estándares

5.2 Definición de Hecho

Para que una historia de usuario o tarea se considera completada, debe cumplir con los siguientes criterios:

- Revisión de código completo y aprobación en pull requests.
- Ejecución exitosa de todas las pruebas unitarias y de integración correspondientes.
- Actualización de la documentación técnica y de usuario en el repositorio.
- Verificación en el entorno de pruebas y validación por parte de los usuarios finales.

5.3 Inclusión de Actividades Relacionadas a CI/CD

- Configuración de pipelines en Azure DevOps para CI/CD.
- Automatización de pruebas y análisis de calidad de código.

- Implementación de monitorización de la cobertura de pruebas.
- Gestión de releases y mantenimiento del changelog para un control efectivo de versiones.

6. Plan de comunicación

La identificación de stakeholders en el proyecto “Oracle Java Chatbot” abarca a todas las personas interesadas que influyen en el desarrollo y la implementación del bot. Esto incluye el equipo de desarrolladores que participan en la creación del bot; los managers de proyecto y desarrollo, responsables de supervisar el progreso del proyecto, el CTO de Oracle y los usuarios finales.

Método de comunicación	Audencia	Mensajes clave	Propósito de la comunicación	Frecuencia
Zoom	Oracle	Compartir avances y dudas	Compartir los avances del proyecto para recibir retroalimentación y aclarar dudas acerca del proyecto	Dos veces por semana. (Lunes y Viernes)
Reunión en persona	Equipo de desarrollo	Compartir avances y repartir trabajo	Dar seguimiento al proyecto, cada miembro del equipo comparte sus avances y definimos nuevas tareas	Semanal. (Martes)
Reunión en persona	Profesor de reto	Compartir avances	Compartir los avances del proyecto y recibir retroalimentación	4 veces por semana. (Lunes, Martes, Jueves y Viernes)

7. Plan de riesgos

7.1 Registro de riesgos

ID	Riesgo	Probabilidad	Impacto	Nivel de riesgo	Estrategia	Propietario	Estado
R-01	Limitaciones en la disponibilidad de recursos	Alta	Alto	Muy alto	Establecer un plan de gestión de recursos.	Todo el equipo	Resuelto
R-02	Mal definición de requerimientos	Alta	Alto	Muy alto	Implementar una revisión con Oracle para validar los requerimientos del sistema.	Todo el equipo	Resuelto
R-03	Problemas de compatibilidad con las versiones utilizadas	Media	Medio	Moderado	Mantener actualizadas las versiones de las herramientas.	Todo el equipo	Controlado
R-04	Cambios en los requerimientos del sistema	Media	Medio	Moderado	Mantener una comunicación constante con Oracle para anticipar posibles cambios y ajustar el plan.	Todo el equipo	Rechazado
R-05	Ausencia de miembros del	Media	Medio	Moderado	Mantener documentación	Todo el equipo	Controlado

	equipo				n sobre las responsabilidades y tareas de cada uno, para facilitar la distribución de tareas en caso de ausencia de algún miembro.		
R-06	Mal cálculo de complejidad	Alta	Alto	Muy alto	Establecer revisiones con el equipo para analizar el progreso del proyecto y la complejidad percibida, y hacer cambios al plan si es necesario.	Todo el equipo	Controlado
R-07	Problemas de comunicación con Oracle	Media	Alto	Alto	Establecer canales de comunicación claros entre el equipo y Oracle.	Todo el equipo	Controlado
R-08	Cumplimiento de seguridad	Alta	Alto	Muy alto	Seguir prácticas de seguridad desde el inicio	Todo el equipo	Controlado

					y realizar revisiones de seguridad regularmente.		
R-09	Integración de tecnología	Media	Alto	Alto	Implementación de prototipos y pruebas piloto antes de la integración completa.	Todo el equipo	Controlado

7.2 Matriz de riesgos

			Impacto	
		Bajo	Medio	Alto
	Baja	Trivial	Tolerable	Moderado
Probabilidad	Media	Tolerable	Moderado	Alto
	Alta	Moderado	Alto	Muy alto

Enlace: [Matriz de riesgos](#)

8. Manejo de Cronograma

8.1 Calendario de ceremonias SCRUM

Inicio de Sprint - Sprint Planning	Sprint (1 semana)					Final de Sprint - Sprint Review
	Daily Scrum	Daily Scrum	Daily Scrum	Daily Scrum	Daily Scrum	
Reunion con Socio Formador (Zoom) Lunes 3:30-4:00pm						
			Trabajo			Reunion con equipo en persona Lunes de 1-2pm
Reunion con equipo en persona Lunes de 4-5pm						

Enlace: [Calendario](#)

Este calendario muestra la programación para un sprint de una semana de duración. Al inicio del sprint, se realizará la planificación del sprint con dos reuniones: una con el socio formador vía Zoom el lunes de 3:30 a 4:00 pm y otra en persona con el equipo de 4:00 a 5:00 pm. Durante la semana del Sprint, hay reuniones diarias de Daily Scrum, y el equipo trabaja en las tareas asignadas. El viernes de esa misma semana, se llevará a cabo otra reunión con el socio formador vía Zoom de 3:30 a 4:00 pm para proporcionar una

actualización sobre el progreso del proyecto. Al final del sprint, se lleva a cabo la revisión con una reunión en persona con el equipo para evaluar el progreso y los resultados del sprint, esta reunión se realizará el lunes siguiente de 1:00 a 2:00 pm.

8.2 Calendario de EPIC

El proyecto se ha dividido en varios EPICs, cada uno con una duración estimada basada en el backlog y las capacidades del equipo. Los EPICs se desarrollarán secuencialmente con algunas posibles superposiciones:

- **Epic de Configuración del Entorno:** Marzo 2024
- **Epic de Desarrollo de la Base de Datos:** Marzo - Abril 2024
- **Epic de Implementación del Frontend y Backend:** Abril 2024
- **Epic de Integración y Pruebas:** Mayo - Junio 2024
- **Epic de Preparación para el Lanzamiento:** Junio 2024

Estimamos que el cierre del proyecto, con todas las pruebas y ajustes finales completados, será para mediados de junio de 2024. Este cronograma está sujeto a ajustes basados en la evolución del proyecto y los feedbacks durante las iteraciones.

Esta estructura de manejo de cronograma está diseñada para mantener el proyecto en curso dentro de los marcos temporales planificados, asegurando que todas las fases del proyecto se completen de manera eficiente y efectiva.

9. Plan de Administración de Configuración

9.1 CMDB

ID	Nombre	Descripción	Tipo	Versión del Software	Versión del Release	Estado	Ubicación
01	OracleSrv	Servidor principal para aplicaciones Java.	Nube	N/A	N/A	Operativo	Oracle Cloud
02	JVM01	Máquina virtual para desarrollo en Java.	Virtual	Ubuntu 20.04	N/A	Operativo	Oracle Cloud

03	OracleDB	Base de datos autónoma para el ChatBot.	Oracle DB	19c	N/A	Operativo	Oracle Cloud Service
04	Oracle Cloud Services	Licencia para el uso de servicios de infraestructura y plataforma de la nube de Oracle	Licencia de servicio de la nube	12.2.1	N/A	Activa	N/A
05	JavaSDK	Java Development Kit	Software	22	N/A	Libre	N/A
06	SpringBoot	Framework para desarrollo en Java	Software	2.5	N/A	Libre	N/A
07	Maven	Herramienta de gestión de proyectos de software como Java.	Herramienta de construcción	Apache Maven 3.9.6	N/A	Libre	N/A
08	Git	Sistema de control de versiones para rastrear cambios en código fuente.	Control de versiones	2.44.0	N/A	Libre	N/A
09	Docker	Plataforma de contenerización para desarrollo y despliegue de aplicaciones en entornos portables y ligeros.	Plataforma de contenedores	4.27.2	N/A	Libre	N/A
010	Visual Studio Code (VSCode)	Editor de código fuente	IDE	1.75.1	N/A	Libre	N/A

011	Kubernetes	Automatiza la implementación y gestión de microservicios en contenedores	Plataforma de orquestación de contenedores	1.28	N/A	Libre	N/A
012	Storage	Almacenamiento persistente en nube	Cloud	N/A	N/A	Operativo	Oracle Cloud
013	Oracle Java ChatBot	Frontend para el ChatBot en Telegram.	Aplicación de Mensajería/Bot dentro de Telegram.	Versión K	0.2	Operativo	https://web.telegram.org/k/

11. Referencias

- Documentación API de Telegram. [Telegram](#)
- Documentación de Oracle Cloud Infrastructure (OCI) [OCI](#)
- Documentación de Docker [Docker](#)
- Documentación de Java [Java](#)
- Documentación de Spring Boot [Spring Boot](#)
- Documentación de Java SE Development Kit (JDK) 8 [JDK 8](#)
- Documentación de Apache Maven [Maven](#)
- Documentación de Terraform [Terraform](#)