

IEEE Std 830-1998

(Revision of
IEEE Std 830-1993)

IEEE Std 830-1998

IEEE Recommended Practice for Software Requirements Specifications

IEEE Computer Society

Sponsored by the
Software Engineering Standards Committee

20 Febrero 2024 SH94654

Índice

1. Introducción
 - 1.1 Propósito
 - 1.2 Alcance
 - 1.3 Definiciones, acrónimos y abreviaciones
 - 1.4 Referencias
 - 1.5 Overview
2. Descripciones Generales
 - 2.1 Perspectiva de producto
 - 2.2 Perspectiva Funcionales
 - 2.3 Características del usuario
 - 2.4 Limitaciones
 - 2.5 Asunciones y defunciones
3. Requerimientos específicos
4. Tabla de requerimientos

1. Introducción

El Bot de Java de Oracle tiene como objetivo automatizar tareas de desarrolladores y proporcionar visibilidad al gerente del equipo. Este SRS describe los requerimientos y funcionalidades del proyecto. El proyecto del Bot de Java de Oracle representa una iniciativa significativa dentro del ecosistema de desarrollo de software de Oracle, con el objetivo de adoptar los principios de la filosofía DevOps. Proporcionando orientación y estableciendo estándares que los servicios de Oracle deben cumplir, esta Especificación de Requisitos de Software (SRS) sirve como un plan integral, detallando las funcionalidades, requisitos y parámetros del Bot de Java de Oracle.

1.1 Propósito

El propósito de este SRS es delinear los requisitos y funcionalidades esenciales del Bot de Java de Oracle así como proporcionar una visión general completa para las partes interesadas involucradas en su desarrollo e implementación. Con la orientación de este documento, los desarrolladores, gerentes de proyectos y otras partes interesadas obtendrán una comprensión clara de los objetivos y el alcance del proyecto Oracle Java Bot.

1.2 Alcance

El alcance del proyecto Oracle Java Bot abarca un amplio espectro de consideraciones, incluyendo estándares arquitectónicos, protocolos de seguridad y marcos tecnológicos. Los aspectos clave del alcance del proyecto incluyen:

a. Producto(s) de Software:

El enfoque principal está en el desarrollo del Bot de Java de Oracle, que personifica los principios de la filosofía DevOps. El Bot de Java de Oracle aprovechará los principios de Aplicaciones Nativas en la Nube, utilizando Oracle Cloud Infrastructure, Oracle Autonomous Database, Kubernetes, Docker y REST API para sus necesidades de infraestructura.

b. Enfoque de Desarrollo:

El desarrollo se adherirá a la arquitectura de Microservicios, implementada usando Java y el framework Spring Boot. Se utilizarán API Gateway y Container Registry para optimizar los procesos de desarrollo e implementación.

c. Consideraciones Operacionales:

La eficiencia operativa se asegurará mediante la automatización de los ciclos de desarrollo y la implementación de prácticas de integración y despliegue continuos (CI/CD).

El Bot de Java de Oracle facilitará la colaboración y comunicación sin problemas entre los miembros del equipo, mejorando la productividad y la visibilidad dentro del ciclo de vida del desarrollo.

d. Estándares de Seguridad:

La seguridad será una preocupación primordial, con todos los componentes de la arquitectura de software diseñados e implementados con rigurosas medidas de seguridad en su lugar.

2. Descripción general

2.1 Perspectiva del producto

El producto estará completamente integrado y funcional con Oracle Cloud Infrastructure, que es un servicio de infraestructura y plataforma en la nube que proporciona y facilita la creación de aplicaciones de datos a gran escala. El ChatBot utiliza Telegram como el canal de comunicación oficial entre los empleados y desarrolladores de Oracle para garantizar la seguridad y privacidad de la información y los datos clasificados.

En las siguientes subsecciones se proporcionará una descripción más detallada del producto y sus funciones.

2.1.1 Interfaces del sistema

El ChatBot requiere dos interfaces de sistemas principales para funcionar completamente, OCI y la API de Telegram. OCI, Oracle Cloud Infrastructure es responsable de alojar el ChatBot y todos sus servidores asociados, administrando los recursos en la nube, el despliegue y la escalabilidad de la aplicación. Telegram facilitó la comunicación entre el ChatBot y los empleados a través del propio servicio de mensajería de Telegram, más específicamente mediante el manejo de la interacción del usuario al igual que el envío y recepción de mensajes.

2.1.2 Interfaces de usuario

Características lógicas: Los usuarios podrán utilizar el ChatBot con el menú proporcionado. Esto tendrá comandos predefinidos y guías de sintaxis de comandos para que los usuarios puedan navegar fácilmente por el servicio.

Aspectos de optimización: La interfaz admitirá mensajes de error cortos y detallados. La interfaz dará prioridad a las guías de instrucciones mínimas y a la carga cognitiva de los usuarios.

2.1.3 Interfaces de hardware

Debido a que el ChatBot se proporciona como una aplicación basada en la nube que interactúa con los usuarios a través de Telegram, los requisitos directos de la interfaz de hardware son mínimos así que están más relacionados con la infraestructura y los componentes del servidor en general.

2.1.4 Interfaces de software

El ChatBot requerirá los siguientes productos de software:

- Oracle Cloud Infrastructure: utilizado para el despliegue, gestión y escalabilidad de la aplicación ChatBot.
- Mnemotécnico: OCI

- Número de versión: Última versión disponible
- Fuente: Oracle Corporation
- API de Telegram: utilizada para la comunicación entre el ChatBot y el usuario.
- Mnemotécnico: TBA
- Número de versión: Última versión disponible
- Fuente: Telegram

2.1.5 Interfaces de comunicaciones

El ChatBot utilizará protocolos HTTPS para garantizar una comunicación segura al utilizar la API de Telegram y los servicios de OCI. Varios protocolos de red locales feders serán determinados completamente por las políticas de seguridad de red de OCI.

2.1.6 Restricciones de memoria

Las operaciones de ChatBots se optimizarán completamente para la eficiencia, buscando y asegurando siempre un bajo uso de memoria para acomodar múltiples instancias en el servicio en la nube. El almacenamiento de registros, datos de usuarios y datos operativos utilizará las soluciones de almacenamiento en la nube de OCI, Oracle Cloud Interface.

2.1.7 Operaciones

- a. Operaciones iniciadas por el usuario: Directamente a través del ChatBot vía Telegram para la gestión de tareas y consultas.
- b. Períodos de operaciones interactivas: La interacción de actualizaciones de tareas será una interacción en tiempo real con el ChatBot.
- c. Períodos de operaciones desatendidas: Recordatorios de tareas automatizadas y actualizaciones de estado de sus tareas enviadas a los usuarios.

2.1.8 Requisitos de adaptación del sitio

- a. Secuencias de inicialización: Personalización y asignación de tareas individuales y prioridades basadas en el proyecto del equipo.
- b. Características relacionadas con el sitio o la misión: La configuración de las frecuencias de notificación, la configuración de seguridad específica y el uso de otras herramientas de Oracle se basan en las políticas del sitio.

2.2 Funciones del producto

El objetivo principal del ChatBot proporcionado a Oracle Corporation es mejorar la productividad y la visión de las actividades del equipo asignadas a individuos en varios períodos de tiempo. Hay varias acciones que nuestro ChatBot puede proporcionar, dependiendo del rol del usuario, para facilitar esta visión de actividades pendientes.

Los gerentes pueden obtener una visión general de las actividades de cada miembro. Esto incluye el seguimiento de su proceso individual, la verificación de sus actividades pendientes y la asignación, eliminación o cambio de actividades. Los gerentes también pueden verificar las

actividades marcadas como completadas por los desarrolladores y reasignarse si es necesario o marcarlas como completadas.

Los desarrolladores pueden visualizar cada una de sus propias actividades personales, previamente asignadas por el Gerente. Se les permite marcar sus actividades como completadas una vez que hayan terminado, pero el Gerente puede reasignarlas.

2.3 Características del usuario

Las características generales de los usuarios pueden describirse de las siguientes maneras: nivel educativo, experiencia y experiencia técnica. Las siguientes subsecciones tienen descripciones relacionadas con cada categoría de características:

- Nivel educativo: Los usuarios del Bot de Java de Oracle tendrán un mínimo de una licenciatura en informática o un campo relacionado. La formación educativa es esencial para garantizar que los usuarios tengan los conocimientos conceptuales necesarios para usar el Bot de Java de manera eficiente y para sus tareas específicas. Otros conocimientos requeridos incluyen conceptos de programación, procesos de desarrollo de software y la dinámica de un equipo completo de desarrollo de software.
- Experiencia: Los usuarios del Bot de Java de Oracle deben tener experiencia en desarrollo de software, específicamente desarrollando con el lenguaje de programación Java. Los usuarios también deben estar familiarizados con las tecnologías de contenedores Docker o Kubernetes, así como con las tecnologías de infraestructura y base de datos de Oracle Cloud.
- Experiencia técnica: Las áreas de experiencia más importantes que los usuarios y el gerente deben tener son las herramientas de desarrollo y las metodologías ágiles. Esto incluye escribir, probar e implementar código. El gerente específicamente debe conocer las técnicas de gestión de proyectos para monitorear el progreso del equipo.

2.4 Limitaciones

Las limitaciones que podrían limitar el desarrollo son las siguientes:

- Políticas regulatorias: El desarrollo debe incluir medidas para la seguridad y privacidad de los datos de acuerdo con el país de implementación.
- Limitaciones de hardware: Prácticamente no hay ninguna, ya que la aplicación se basa en una red en la nube.
- Interfaces con otras aplicaciones: La principal aplicación externa es Telegram, que debe cumplir con los protocolos de interfaz de acuerdo con la implementación del Bot.
- Operación paralela: El soporte de acceso paralelo y actualizaciones por parte de una variedad de usuarios sin sacrificar la integridad de los datos es crucial.
- Funciones de auditoría: Registro y auditoría para rastrear las acciones de los usuarios y los cambios en las tareas.
- Funciones de control: Dependiendo del rol del usuario, el control y los permisos deben estar restringidos.
- Lenguaje de alto nivel: Sólo Java con el framework Spring Boot.

- Requisitos de confiabilidad: Se requiere una alta confiabilidad porque los usuarios deben poder utilizar el Bot siempre que sea necesario.
- Criticidad: Crucial para mejorar la productividad y brindar visibilidad de tareas al equipo.
- Seguridad: Las evaluaciones de seguridad y el cifrado de datos son necesarios para proteger la información del usuario.

2.5 Suposiciones y dependencias

Las suposiciones que se pueden hacer sobre el software son que todos los servicios del lado del servidor siempre estarán disponibles (OCI, Kubernetes, Bases de datos) y serán flexibles según los requisitos finales del proyecto. La API para interfaces externas, principalmente Telegram, también será estable y accesible en todo momento. Las dependencias del software están relacionadas con las suposiciones. Los servicios de Oracle que incluyen servicios en la nube son cruciales para las funcionalidades del Bot. Otra dependencia es el conocimiento del equipo en sí y su familiaridad con las prácticas de desarrollo ágil.

2.6 Distribución de requisitos

En futuras versiones del software, se podrán considerar los siguientes requisitos:

- Integración de IA más avanzada.
- Expansión de interfaces externas (otras plataformas de mensajería).
- Más funcionalidades.
- Integración de análisis en el propio Bot para que pueda aprender patrones de usuario y adaptar los resultados.

3. Requisitos específicos

Esta sección del SRS debe contener todos los requisitos de software con un nivel de detalle suficiente para permitir a los diseñadores diseñar un sistema que satisfaga dichas necesidades, y a los probadores probar que el sistema satisface esos requisitos. A lo largo de esta sección, cada requisito establecido debe ser externamente perceptible por los usuarios, operadores u otros sistemas externos. Estos requisitos deben incluir, como mínimo, una descripción de cada entrada (estímulo) en el sistema, cada salida (respuesta) del sistema y todas las funciones realizadas por el sistema en respuesta a una entrada o en apoyo de una salida. Como esta es a menudo la parte más grande e importante del SRS, se aplican los siguientes principios:

- a. Los requisitos específicos deben establecerse de conformidad con todas las características mencionadas en la sección siguiente.
- b. Todos los requisitos deben ser identificables de forma única.
- c. Se debe prestar especial atención a la organización de los requisitos para maximizar la legibilidad.

3.1 Interfaces externas

3.1.1 Interfaces de usuario:

- La aplicación contará con un Bot de Telegram como interfaz de usuario principal para la interacción con los empleados.

- Una interfaz de chat fácil de usar mostrará opciones para agregar, ver y eliminar tareas.
- Se proporcionarán solicitudes de autenticación para el acceso seguro a las funciones de gestión de tareas.

3.1.2 Interfaz de hardware:

No se requieren interfaces de hardware específicas para esta aplicación, ya que es una aplicación nativa de la nube implementada en Oracle Cloud Infrastructure (OCI).

3.1.3 Interfaz de software:

- Oracle Autonomous Database: Para la persistencia de datos de tareas y credenciales de usuario.
- OCI Container Registry (OCR) y Oracle Container Engine for Kubernetes (OKE): Para gestionar los contenedores y la orquestación de la aplicación.
- SpringBoot: Framework para crear las API REST.
- API Gateway: Para gestionar el acceso a las API REST.

3.1.4 Interfaz de comunicación:

- Comunicación con el Chatbot de Telegram a través de su API.
- Comunicaciones entre los microservicios SpringBoot y Oracle Autonomous Database utilizando JDBC o un cliente de base de datos nativo.

3.2 Funciones

- Rol del empleado:
 - Ver una lista de sus tareas actuales en tiempo real.
 - Agregar nuevas tareas a su lista.
 - Eliminar tareas que hayan creado previamente.
- Seguridad y autenticación
Puede ser apropiado dividir los requisitos funcionales en subfunciones o subprocesos. Esto no implica que el diseño del software también se divida de esa manera.

3.3 Requisitos de rendimiento

Los requisitos numéricos estáticos incluye lo siguiente:

- a. El número de terminales a soportar;
- b. El número de usuarios simultáneos a soportar;
- c. Cantidad y tipo de información a manejar.

3.4 Requisitos lógicos de la base de datos

- a. Tipos de información utilizada por varias funciones;
- Información del usuario que se relaciona con cada miembro del equipo y gerentes, roles, detalles y, lo que es más importante, información de autenticación.

- Detalles que se relacionan con las tareas de cada miembro del equipo que pueden incluir descripciones, estado, fechas límite, consejos basados en plazos, etc.
- b. Frecuencia de uso;
 - Cada vez que un usuario inicia sesión en el chatbot.
 - En una sola sesión, cuando un usuario necesita consultar sus tareas, se debe acceder a la base de datos y, si es necesario, actualizar la información de las tareas.
- c. Capacidades de acceso;
 - El equipo debe tener acceso a sus propios perfiles (no a otros), sus tareas y sus informes de progreso.
 - Los gerentes deben tener acceso a todos los perfiles de usuario (excepto otros gerentes), todas las tareas y todos los informes de progreso.
- d. Entidades de datos y sus relaciones;
 - Usuarios -> tareas: uno a muchos
 - Tareas -> usuarios: muchos a uno
 - Gerentes -> usuarios, tareas: uno a muchos
- e. Restricciones de integridad;
 - Restricciones proporcionadas para ID de usuario, proyecto y tarea para prevención de duplicación.
- f. Requisitos de retención de datos.
 - Los detalles y el historial de las tareas se conservarán durante un período prolongado de tiempo para una consulta eficiente y la prevención de la pérdida de datos del usuario.

3.5.1 Cumplimiento de estándares

- a. Formato de informe
 - Una estructura estandarizada proporciona la información mínima necesaria que se incluirá en los informes, así como formatos de fecha, fuente del texto, tamaño del texto y otros elementos gráficos.
- b. Nomenclatura de datos
 - Los datos se nombran de manera consistente según la convención de nomenclatura de Oracle.
- c. Procedimientos contables
 - No está claro si serán necesarios procedimientos contables, pero si lo son, deben cumplir con los esquemas financieros de Oracle y realizar un seguimiento de los costos relacionados con el uso de datos.
- d. Seguimiento de auditoría

- Incluir registro de auditoría para capturar cambios de tareas y proyectos con estados antes y después, lo cual es importante para el control interno.
- Los cambios deben registrarse junto con sus respectivas marcas de tiempo, ID y toda la información relacionada con los cambios.

3.6 Atributos del sistema de software

Hay varios atributos del software que pueden servir como requisitos. Es importante que se especifiquen los atributos requeridos para que su logro pueda verificarse objetivamente.

3.6.1 Fiabilidad

- Manejo de errores: El sistema debe manejar los errores y registrar o alertar a los administradores del sistema cuando ocurran problemas para priorizar la corrección de errores.
- Tolerancia: Deben existir mecanismos para mantener el sistema en funcionamiento incluso si se producen fallos.
- Pruebas: Se deben realizar pruebas periódicas para garantizar que el sistema falle lo menos posible, mejorando la integridad de los datos y la experiencia del usuario.

3.6.2 Disponibilidad

- Punto de control y recuperación: Implementación de mecanismos para guardar periódicamente los estados y registros del sistema (funciones de guardado automático), para la recuperación del último estado "bueno" conocido en caso de falla.
- Alta disponibilidad: Uso de OCI para implementar el chatbot en varios dominios y no solo en uno, esto asegura el funcionamiento sin pausas.
- Monitoreo y alertas: El monitoreo debe ser continuo para detectar problemas de rendimiento y del sistema con alertas disponibles para cualquier situación de falla.

3.6.3 Seguridad

- a. Utilizar ciertas técnicas criptográficas;

Cifrado de datos cuando el sistema está en reposo y cuando los datos están en tránsito.
Las técnicas específicas están por determinar.

- b. Mantener conjuntos de datos de registro o historial específicos;

Mantener registros de acceso al sistema y cambios en los datos (inmutables y seguros) con los detalles necesarios.

- c. Asignar ciertas funciones a diferentes módulos;

Diseñar con arquitectura modular para restringir la comunicación entre módulos de acuerdo con los privilegios.

d. Restringir las comunicaciones entre algunas áreas del programa;
Separar las capacidades de los usuarios con control de acceso basado en roles para que los usuarios solo puedan acceder y modificar datos relacionados con ellos mismos y no con otros.

e. Verificar la integridad de los datos para variables críticas.
Comprobaciones periódicas de integridad de datos para detectar y actuar sobre modificaciones no autorizadas, si ocurren.

3.6.4 Mantenibilidad

Se debe tener en cuenta la modularidad para permitir la facilidad con las actualizaciones, la depuración y las modificaciones de mejora que afectan solo a las partes necesarias del sistema. Seguir las prácticas y estándares de codificación de Oracle al desarrollar con Java y Springboot para una mejor legibilidad y mantenimiento. La documentación debe incluir la arquitectura general del sistema, las API, los procedimientos y las bases de código.

3.6.5 Portabilidad

- Lenguaje y marcos: Java y Spring Boot exclusivamente debido a los estándares de Oracle, y para garantizar la portabilidad del sistema.
- Uso de contenedores: El despliegue de aplicaciones mediante contenedores Docker es necesario para garantizar que se pueda desplegar en cualquier sistema (siempre que admita estas tecnologías).

3.7 Organización de los requisitos específicos

- Empleados: Capaces de agregar, ver y eliminar tareas personales; autenticarse para acceder a las tareas.
- Gerentes: Ver tareas en todo su equipo.

3.7.1 Modo del sistema

- Tareas: Atributos (título, descripción, fecha límite, estado), Funciones (crear, ver, eliminar, marcar como completa).
- Usuarios: Atributos (nombre, rol, credenciales), Funciones (autenticarse, ver tareas).

3.7.2 Clase de usuario

- Empleados: Pueden interactuar con el bot para administrar tareas personales, que requieren funciones para agregar, ver, completar y eliminar tareas, con un proceso de autenticación para el acceso.
- Gerentes: Necesitan funcionalidades para supervisar las tareas del equipo, incluida la visualización de todas las tareas y la autenticación.

3.7.3 Objetos

- Tareas: Con propiedades como ID, descripción, fecha límite y estado de finalización, y funcionalidades para operaciones CRUD.
- Usuarios (Empleados/Gerentes): Identificados por atributos como ID, nombre, rol y credenciales, con funcionalidades que incluyen inicio de sesión y resumen de tareas.

3.7.4 Característica

- Administración de tareas: Adición, visualización, edición, finalización de tareas a través de comandos de Telegram.
- Autenticación de usuario: Mecanismo seguro para verificar la identidad del usuario antes de permitir el acceso a las funcionalidades relacionadas con las tareas.

3.7.5 Estímulo

Entradas del usuario a través de Telegram, como comandos para la gestión de tareas o consultas.

3.7.6 Respuesta

- Actualizaciones de tareas
- Respuestas a las consultas de los usuarios.

3.7.7 Jerarquía funcional

- Funcionalidad principal: Central para la aplicación, incluida la gestión de tareas y la autenticación de usuarios, asegurando que las operaciones fundamentales estén claramente definidas.
- Integración y comunicación: Detallar las interacciones con los servicios de OCI, la base de datos autónoma de Oracle para la persistencia de datos y Telegram para la comunicación con el usuario.
- Seguridad y cumplimiento: Abarca autenticación, autorización, cifrado de datos y cumplimiento de las regulaciones de privacidad.

3.8 Comentarios adicionales

Para el proyecto ChatBot, emplear una estrategia de organización híbrida, fusionando la clase de usuario, el objeto, la característica, el estímulo-respuesta y la jerarquía funcional, proporciona una visión multidimensional de los requisitos del sistema. Este enfoque integral facilita un marco detallado y organizado para los desarrolladores, alineándose con los objetivos del proyecto y las expectativas de las partes interesadas.

Para el proyecto ChatBot, diseñado para servir a los empleados de Oracle para la gestión de tareas a través de una aplicación Cloud Native que utiliza SpringBoot, Oracle Autonomous Database y los servicios de OCI, se adaptará la organización de los requisitos específicos en la Sección 3 del documento SRS. Este enfoque se alinea con las características únicas del proyecto, incorporando

clases de usuarios, objetos del sistema, características, estímulos y respuestas, y una jerarquía funcional para cubrir de manera integral todos los aspectos del sistema.

4. Requerimientos

4.1 Requerimientos Funcionales

Categoría	ID	Descripción	Justificación	Prioridad
Usuario	REQ-001	Los usuarios pueden agregar, ver y eliminar tareas.	Esencial para la gestión de tareas individuales y productividad.	Alta
Modo Sistema	REQ-002	Las tareas tienen atributos (título, descripción).	Fundamental para el seguimiento y gestión detallados de tareas.	Alta
Modo Sistema	REQ-003	Las tareas admiten funciones (crear, ver, eliminar, marcar como completada).	Permite una gestión integral de las tareas dentro del sistema.	Alta
Modo Sistema	REQ-004	La gestión de tareas incluye agregar, ver, completar y eliminar tareas.	Proporciona a los usuarios control sobre su flujo de trabajo y prioridades.	Alta

4.2 Requerimientos No Funcionales

Categoría	ID	Descripción	Justificación	Prioridad
Interfaz de Usuario	RNF-001	La pantalla principal tiene un (1) campo de entrada y cuatro (4) botones.	Facilita la interacción del usuario y la navegación dentro de la aplicación.	Alta
Interfaz de Usuario	RNF-002	Se muestran mensajes de retroalimentación con atajos/sugerencias sobre lo que se puede hacer a continuación después de ejecutar un comando.	Mejora la experiencia del usuario al proporcionar orientación y facilitar la navegación.	Media