
Documento de Diseño - SSD

PROYECTO

Instituto Tecnológico de Monterrey

Oracle Cloud

INTEGRANTES

Bella Elisabet Perales Meléndez y Alcocer - A00833423

Mariel Gisela Pérez Ferrusquía - A00832811

Daniela Nuño Martinez - A01177702

Andrés Fernando Garza Garcia - A01138704

Marcelo García Pablos Vélez - A00815371

12/6/2024

Monterrey, Nuevo León

Índice

1. Introducción
 - 1.1 Propósito
 - 1.2 Alcance
 - 1.3 Resumen
 - 1.4 Material de Referencia
 - 1.5 Definiciones y Acrónimos
2. Visión General de Sistema
 - 2.1 Perspectiva del producto
 - 2.2 Funciones del producto
 - 2.3 Características de los usuarios
 - 2.4 Restricciones generales
3. Arquitectura del Sistema
 - 3.1 Diseño Arquitectónico
 - 3.2 Descripción de Descomposición
 - 3.3 Justificación del Diseño
4. Diseño de Datos
 - 4.1 Descripción de Datos
 - 4.2 Diccionario de Datos
5. Diseño de Componentes
6. Diseño de Interfaz Humana
 - 6.1 Visión General de la Interfaz de Usuario
 - 6.2 Imágenes de Pantalla
 - 6.3 Objetos de Pantalla y Acciones
7. Matriz de Requerimientos
8. Apéndices
9. Referencias

1. Introducción

1.1 Propósito

Este documento describe el diseño del Chatbot de Telegram que permite el seguimiento de tareas de los equipos de desarrollo, utilizando la infraestructura de Oracle Cloud Infrastructure (OCI). El Bot de Java de Oracle tiene como objetivo automatizar tareas de desarrolladores y proporcionar visibilidad al gerente del equipo. 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.

1.2 Alcance

Desarrollar e implementar un ChatBot que permita a los desarrolladores revisar, agregar, eliminar y marcar tareas, así como proporcionar al manager una visión general de las tareas del equipo. La aplicación web proporcionará una interfaz adicional para la gestión de tareas y proyectos. Se busca implementar el proyecto en los equipos de trabajo de Oracle.

1.3 Resumen

Este documento proporciona una descripción detallada de la arquitectura del sistema, el diseño de datos, el diseño de componentes y el diseño de la interfaz humana. También incluye una matriz de requerimientos que mapea los componentes del sistema con los requerimientos funcionales.

1.4 Material de Referencia

- Documento de Especificación de Requerimientos de Software (SRS) del Chat Bot
- Documentación de la API de Telegram Bot
- Documentación de Spring Boot
- Documentación de React
- Documentación de Oracle Cloud Infrastructure

1.5 Definiciones y Acrónimos

- OCI: Oracle Cloud Infrastructure
- API: Interfaz de Programación de Aplicaciones
- DB: Base de Datos
- SRS: Especificación de Requerimientos de Software
- UI: Interfaz de Usuario

2. Visión general del sistema

2.1 Perspectiva del producto

El sistema consiste de un chatbot de Telegram y una aplicación web, ambos interactuando con una base de datos alojada en OCI para almacenar y recuperar información sobre tareas y proyectos.

2.2 Funciones del producto

- Gestión de tareas (ver, crear, borrar, actualizar) para desarrolladores
- Gestión de tareas (ver) para gerentes
- Gestión de proyectos (ver) para gerentes

- Interfaz de chatbot de Telegram para interacciones
- Interfaz web para gestión adicional de tareas para desarrolladores
- Interfaz web para gestión adicional visualización de tareas para managers

2.3 Características de los usuarios

Los usuarios principales del sistema serán empleados del equipo de desarrollo de software de Oracle, específicamente:

- Desarrolladores:
 - Podrán trabajar en modalidades híbridas y remotas.
 - Podrán gestionar sus tareas personales (crear, ver, actualizar y eliminar)
 - Podrán utilizar el Chat Bot de Telegram y la aplicación web.
- Managers:
 - Podrán trabajar en modalidades híbridas y remotas.
 - Podrán observar las tareas de su equipo de desarrollo.
 - Podrán observar la información de su proyecto.
 - Podrán utilizar el Chat Bot de Telegram y la aplicación web.

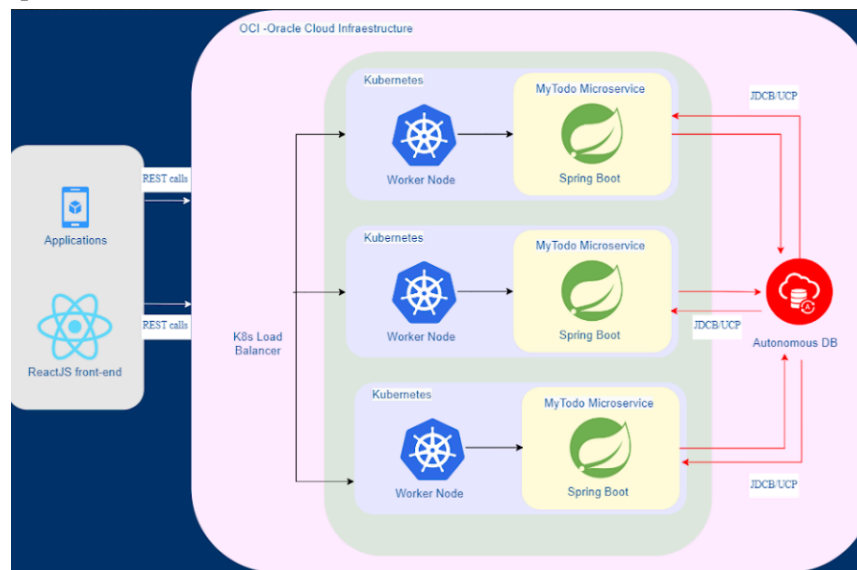
2.4 Restricciones generales

- El sistema debe integrarse con la plataforma de Telegram de manera segura.
- Debe garantizar la seguridad y privacidad de los datos de tareas y proyectos.
- Utilizar los servicios y herramientas aprobados por Oracle.
- Implementar prácticas de DevOps durante su ciclo de vida.

3. Arquitectura del sistema

3.1 Diseño Arquitectónico

El sistema del Bot de Java de Oracle sigue un enfoque de arquitectura de microservicios. Consta de varios componentes desplegados de forma independiente que se comunican a través de interfaces. La aplicación web y el chatbot de Telegram son servicios separados que interactúan con una API backend construida usando Spring Boot. La base de datos se aloja en Oracle Cloud Infrastructure (OCI). A continuación se presenta el diagrama de arquitectura utilizado para el desarrollo del sistema:



3.2 Descripción de Descomposición

El sistema se descompone en los siguientes componentes principales:

- Chatbot de Telegram: Desarrollado en Java, este componente maneja las interacciones y comandos de los usuarios a través de la plataforma de mensajería Telegram.
- Aplicación Web (Frontend): Proporciona una interfaz de usuario web enriquecida para la gestión de tareas y está construida con React.
- API Backend: Implementa la lógica de negocio y la capa de acceso a datos, desarrollada usando Spring Boot.
- Base de Datos: Componente responsable del almacenamiento persistente de datos de tareas, proyectos, empleados y otra información relevante, alojada en Oracle Cloud Infrastructure (OCI).

3.3 Justificación del Diseño

La arquitectura de microservicios permite el desarrollo, despliegue y escalamiento independiente de los componentes del chatbot y la aplicación web. Proporciona los siguientes beneficios clave: escalabilidad, resiliencia y agilidad. La elección de tecnologías como Spring Boot se basa por su robustez y facilidad de desarrollo para la API backend, mientras que React proporciona un enfoque modular y eficiente para construir la UI frontend. Por último, el uso de Oracle Cloud Infrastructure (OCI) para alojar la base de datos fue debido a su confiabilidad, escalabilidad y rendimiento, además de cumplir con los estándares y políticas de Oracle.

4. Diseño de datos

4.1 Descripción de Datos

El sistema almacena los datos de tareas y proyectos en una base de datos relacional alojada en Oracle Cloud Infrastructure. Las entidades utilizadas en el sistema incluyen:

- Task: Representa una tarea con atributos como título, descripción, fecha límite y estado.
- Project: Representa un proyecto con atributos como nombre, descripción y tareas asociadas.
- Employee: Representa a un usuario (empleado o gerente) con atributos como nombre, rol y tareas/proyectos asociados.
- Department: Representa el departamento al que pertenece un usuario con atributos como nombre.

4.2 Diccionario de Datos

- Task:
 - id (int): Identificador único de la tarea.
 - name (String): Título de la tarea.
 - description (String): Descripción de la tarea.
 - datecreated (Timestamp): Fecha en la que la tarea fue creada.
 - status (int): Estado actual de la tarea.
 - datelimit (Timestamp): Fecha límite de entrega de la tarea.
 - type (String): tipo de tarea
 - employeeid (int): Identificador del empleado al que le pertenece la tarea.

- proyectid (int): Identificador del proyecto al que pertenece la tarea.
- **Project:**
 - id (int): Identificador único del proyecto.
 - name(String): Nombre del proyecto.
 - datestart (Timestamp): Fecha de inicio del proyecto.
 - dateend (Timestamp): Fecha de finalización del proyecto.
 - status (int): Estado actual del proyecto.
 - departamentid (int): Identificador del departamento al que pertenece el proyecto.
- **Employee:**
 - id (int): Identificador único del usuario.
 - name (String): Nombre del usuario.
 - lastname (String): Apellido del usuario.
 - mail (String): Correo del usuario.
 - cellphone (String): Número de contacto del usuario.
 - address (String): Dirección del usuario.
 - status (int): Estado actual del empleado.
 - manager (boolean): Indicador de rol en la empresa.
 - mynumber (int): Número de asociado del empleado.
 - departamentid (int): Identificador del departamento al que pertenece el empleado.
- **Department:**
 - id (int): Identificador único del departamento.
 - name (String): Nombre del departamento.

5. Diseño de componentes

- **Chatbot de Telegram:**
 - Maneja la autenticación y autorización de usuarios.
 - Procesa los comandos y consultas de los usuarios relacionados con tareas.
 - Se comunica con la API backend para recuperar y actualizar datos.
 - Implementa la lógica para enviar y recibir mensajes a los usuarios a través de Telegram.
- **Aplicación Web (Frontend):**
 - Proporciona interfaces de usuario para la gestión de tareas.
 - Se comunica con la API backend para recuperar y actualizar datos.
 - Implementa la autenticación y autorización de usuarios.
- **Base de Datos:**
 - Almacena los datos de tareas, proyectos, departamentos y empleados.
 - Proporciona mecanismos de recuperación y persistencia de datos.

6. Diseño de interfaz humana

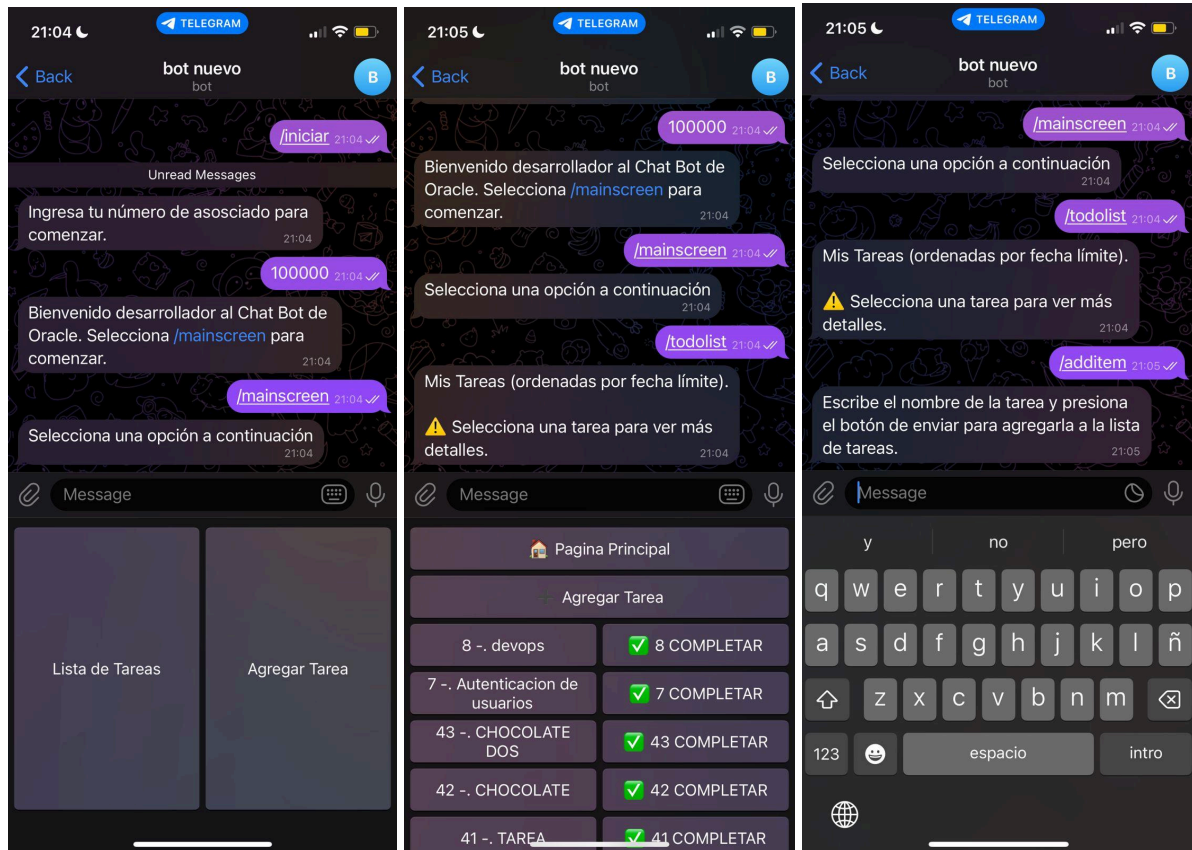
6.1 Visión General de la Interfaz de Usuario

El sistema proporciona dos interfaces principales para interactuar con los usuarios:

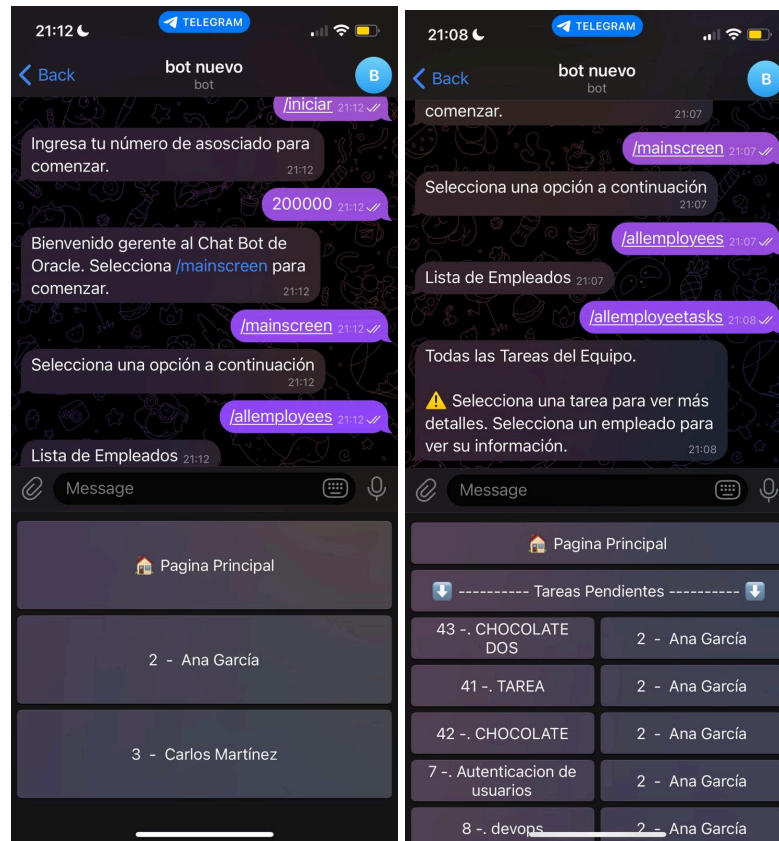
- Chat Bot de Telegram: Los usuarios pueden interactuar con el chatbot usando comandos de lenguaje natural para ver, crear y gestionar tareas.
- Aplicación Web: La aplicación web permite a los desarrolladores ver y actualizar sus tareas, y permite a los managers ver las tareas de su equipo.

6.2 Imágenes de Pantalla

- Vista de Desarrollador:



- Vista de Manager:



6.3 Objetos de Pantalla y Acciones

- Chatbot de Telegram:
 - /iniciar: Inicia la conversación con el chatbot, solicitando el número de asociado del usuario.
 - /todolist: Muestra al desarrollador la lista de tareas que tiene asignadas.
 - /additem: Permite a los desarrolladores crear una nueva tarea.
 - /allemployees: Muestra todos los empleados en un proyecto al manager.
 - /allemployeetasks: Muestra todas las tareas del equipo de desarrolladores al manager.
 - /addproject: Agrega un proyecto al manager solo si este no tiene asignado uno antes.
 - /mainscreen: Muestra la página principal.
- Aplicación Web:
 - Pantalla de Inicio de Sesión: Permite a los usuarios autenticarse y acceder al sistema.
 - Panel de Tareas (Desarrollador): Muestra una lista de tareas asignadas con opciones para ver, crear y eliminar tareas.
 - Panel de Tareas (Manager): Muestra una lista de tareas asignadas a todo su equipo de desarrollo.

7. Matriz de requerimientos

7.1 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

7.2 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

8. Apéndices

Este documento de diseño de software proporciona una visión completa de la arquitectura del sistema, el diseño de datos, el diseño de componentes y el diseño de la interfaz humana para el sistema de gestión de tareas mediante chatbot de Telegram y aplicación web. Sirve como un plano para el desarrollo e implementación del sistema, asegurando que cumpla con los requerimientos especificados y proporcione una experiencia de usuario sin problemas. A continuación se presentan los documentos de referencia que se utilizaron para su investigación y desarrollo.

- Documento de Especificación de Requerimientos de Software (SRS) del Chat Bot
 - <https://docs.google.com/document/d/1d8fMSqpMRXOqMrOBpCMLmw6vZy2pL2ww03RD9IrUNgg/edit?usp=sharing>
- Documentación de la API de Telegram BotS
 - <https://core.telegram.org/bots/api>

- Documentación de Spring Boot
 - <https://docs.spring.io/spring-boot/documentation.html>
- Documentación de React
 - <https://es.react.dev/learn>
 - <https://es.legacy.reactjs.org/docs/getting-started.html>
 - <https://react.dev/reference/react>
- Documentación de Oracle Cloud Infrastructure
 - <https://docs.oracle.com/es-ww/iaas/Content/home.htm>
 - <https://docs.oracle.com/es-ww/iaas/Content/home.htm>
 - <https://docs.oracle.com/es-ww/iaas/Content/General/Reference/aqswhitepapers.htm>

9. Referencias

- Telegram (2024) Telegram Bot API. Recuperado de: <https://core.telegram.org/bots/api>
- More M. (2023) ¿Qué es la UX y la UI? Recuperado de: <https://www.iebschool.com/blog/que-es-la-ux-y-la-ui-analitica-usabilidad/>
- Oracle (2024) Documentación de Oracle Cloud Infrastructure. Recuperado de: <https://docs.oracle.com/es-ww/iaas/Content/home.htm>
- Heut P. (2022) Arquitectura de software: Qué es y qué tipos existen. Recuperado de: <https://openwebinars.net/blog/arquitectura-de-software-que-es-y-que-tipos-existen/>
- SpringBoot (2024) Spring Boot. Recuperado de: <https://docs.spring.io/spring-boot/index.html>