

Prupuesta

Prupuesta - Trabajo Final Aplicaciones TCP/IP

Integrantes

- Aguilera Conti, Agustin
- Principe, Agustin Nicolas

Aplicación en Línea de Control de Inventario

Resumen

La propuesta es desarrollar una aplicación en línea de control de inventario que permita registrar nuevos productos, visualizar los existentes, cargar imágenes asociadas a dichos productos y la creación de usuarios con diferentes permisos de manera eficiente. La aplicación se desarrollará utilizando FastAPI como framework de backend, una base de datos para almacenar la información del usuarios e inventario y se implementará en un entorno Dockerizado y desplegado en Kubernetes. Además, se incluirá un sistema de monitoreo en el clúster y se realizarán pruebas de carga para asegurar el rendimiento y la escalabilidad del sistema.

Introducción

Surge la necesidad de crear esta aplicación de control de inventario debido a la creciente demanda de optimizar la gestión de productos en empresas y negocios, en este caso se busca garantizar la seguridad y privacidad de la información, dar una solución tecnológica robusta y escalable y por último comprometerse con el rendimiento óptimo del sistema.

Definiciones

- Sistema de Control de Inventario: Aplicación en línea que permite en control de inventario y la gestion de usuarios con diferentes permisos.
- FastAPI: Framework de desarrollo web de alto rendimiento y fácil uso basado en Python.

- Docker: Plataforma de contenedores que permite empaquetar la aplicación y todas sus dependencias en un entorno aislado.
- Kubernetes: Sistema de orquestación de contenedores que facilita el despliegue, la gestión y la escalabilidad de la aplicación en un entorno de producción.
- Monitoreo en el clúster: Sistema de supervisión en tiempo real que permite controlar el rendimiento, la disponibilidad y la salud del clúster de Kubernetes.
- Pruebas de carga: Evaluación del comportamiento del sistema bajo diferentes cargas de trabajo para garantizar su rendimiento y escalabilidad.

Descripción de la Propuesta

- Objetivos del proyecto: Desarrollar una aplicación en línea de gestión de inventario que sea fácil de usar, segura y escalable.
- Alcance del proyecto: La aplicación permitirá a los usuarios con diferentes niveles de permisos, consumir y/o crear productos en el sistema y agregar imágenes para dichos productos.
- Requerimientos del proyecto: El sistema deberá tener una interfaz de usuario intuitiva, autenticación segura, almacenamiento de productos en una base de datos, y capacidades de monitoreo y pruebas de carga.
- Metodología de desarrollo: Se utilizará una metodología ágil, como Scrum, para garantizar la entrega iterativa y la colaboración efectiva durante el desarrollo del proyecto.

Beneficios y Valor Agregado

- Optimización de Stock para reducir costos.
- Mejora en la experiencia del cliente a la hora de consultar precios.
- Escalabilidad para adaptarse al crecimiento de clientes y la carga de pedidos.

Plan de Trabajo Tentativo

- **Fase 1:** Análisis y diseño de la aplicación.
 - Definir los requisitos de la aplicación, identificar funcionalidades clave y necesidades de la base de datos.
 - Diseñar la arquitectura de la aplicación, determinando los componentes y la comunicación entre ellos.

- **Fase 2:** Desarrollo del backend utilizando FastAPI y configuración de la base de datos.
 - Configurar el entorno de desarrollo con las herramientas necesarias y establecer la base de datos.
 - Desarrollar el backend utilizando FastAPI, definiendo rutas, controladores y modelos de datos.
 - Aplicar pruebas unitarias y de integración para garantizar el correcto funcionamiento del backend.
- **Fase 3:** Desarrollo del frontend para la interfaz de usuario.
 - Desarrollar el frontend utilizando tecnologías como HTML, CSS y JavaScript.
 - Diseñar una interfaz de usuario atractiva y funcional que cumpla con los requisitos establecidos.
 - Realizar pruebas exhaustivas del frontend para asegurar su correcto funcionamiento.
- **Fase 4:** Dockerización de la aplicación y creación de los archivos de configuración de Kubernetes.
 - Configurar la aplicación para ser ejecutada en contenedores Docker.
 - Crear los archivos de configuración de Kubernetes para facilitar el despliegue y la gestión de la aplicación.
- **Fase 5:** Despliegue de la aplicación en un clúster de Kubernetes.
 - Configurar y desplegar la aplicación en un clúster de Kubernetes.
 - Asegurar que la aplicación esté correctamente funcionando en el entorno de producción.
- **Fase 6:** Configuración del sistema de monitoreo en el clúster.
 - Implementar un sistema de monitoreo utilizando herramientas como Prometheus y Grafana.
 - Configurar alertas y paneles de monitoreo para supervisar el rendimiento y la disponibilidad de la aplicación.
- **Fase 7:** Realización de pruebas de carga para evaluar el rendimiento y la escalabilidad del sistema.
 - Realizar pruebas de carga simulando una gran cantidad de usuarios y tráfico.
 - Evaluar el rendimiento y la capacidad de respuesta del sistema bajo condiciones de carga extremas.

- **Fase 8:** Ajustes finales, corrección de errores y entrega del proyecto.
 - Realizar ajustes y mejoras basados en los resultados de las pruebas y el monitoreo.
 - Corregir errores y asegurar que la aplicación cumpla con los requisitos establecidos.
 - Entregar el proyecto finalizado y listo para su uso.