

Proyecto 1 (%)

Fecha de Entrega:

Sistema de Encuestas

Descripción General

El objetivo del proyecto es diseñar e implementar un sistema de encuestas backend que utilice Docker, Docker Compose, MongoDB, PostgreSQL, Redis y RestAPI. Este sistema permitirá a los usuarios crear, publicar y gestionar encuestas con diferentes tipos de preguntas, así como registrar y administrar listas de encuestados.

Tecnologías a Utilizar

- Docker y Docker Compose para la contenerización y orquestación de los servicios.
- MongoDB para almacenar datos no estructurados.
- PostgreSQL (u otra base de datos relacional) para almacenar datos estructurados.
- Redis para la gestión de caché.
- RestAPI para la comunicación entre el servidor y los clientes.

Requerimientos Funcionales

Autenticación y Autorización

- Registro y autenticación de usuarios con diferentes roles (administrador, creador de encuestas, encuestado).

Gestión de Encuestas

- Creación y configuración de encuestas con diversos tipos de preguntas.
- Publicación y gestión de accesos a las encuestas.
- Recolección de resultados.

Gestión de Encuestados

- Registro y administración de listas de encuestados.
- Asignación y seguimiento de encuestados a encuestas.

Lista de Endpoints

Esta lista es una referencia y no es exhaustiva, siéntanse libres de agregar nuevos endpoints. Además, todos aquellos endpoints que devuelvan una colección de registros deben permitir paginación.

Autenticación y Autorización

- **POST /auth/register** - Registra un nuevo usuario.
- **POST /auth/login** - Autentica a un usuario y retorna un token.
- **GET /auth/logout** - Cierra la sesión del usuario (requiere token).

Usuarios

- **GET /users** - Lista todos los usuarios (administrador).
- **GET /users/{id}** - Obtiene los detalles de un usuario específico.
- **PUT /users/{id}** - Actualiza la información de un usuario (solo su perfil o administrador).
- **DELETE /users/{id}** - Elimina un usuario (administrador).

Encuestas

- **POST /surveys** - Crea una nueva encuesta (requiere token de creador de encuestas o administrador).
- **GET /surveys** - Lista todas las encuestas disponibles públicamente.
- **GET /surveys/{id}** - Muestra los detalles de una encuesta específica.
- **PUT /surveys/{id}** - Actualiza los detalles de una encuesta (requiere token de creador de la encuesta o administrador).
- **DELETE /surveys/{id}** - Elimina una encuesta (requiere token de creador de la encuesta o administrador).
- **POST /surveys/{id}/publish** - Publica una encuesta para hacerla accesible (requiere token de creador de la encuesta o administrador).

Preguntas de Encuestas

- **POST /surveys/{id}/questions** - Añade preguntas a una encuesta específica (requiere token).
- **GET /surveys/{id}/questions** - Lista las preguntas de una encuesta específica.
- **PUT /surveys/{id}/questions/{questionId}** - Actualiza una pregunta específica de una encuesta (requiere token).
- **DELETE /surveys/{id}/questions/{questionId}** - Elimina una pregunta de una encuesta (requiere token).

Se debe incorporar soporte a los siguientes tipos de preguntas y campos:

- Preguntas abiertas (texto)
- Preguntas de elección simple
- Preguntas de elección múltiple
- Escala de calificación
- Preguntas Sí/No
- Preguntas numéricas

Respuestas de Encuestas

- **POST /surveys/{id}/responses** - Envía respuestas a una encuesta por parte de un encuestado.
- **GET /surveys/{id}/responses** - Lista todas las respuestas de una encuesta específica (requiere token de creador de la encuesta o administrador).

Encuestados

- **POST /respondents** - Registra un nuevo encuestado.
- **GET /respondents** - Lista todos los encuestados (requiere token).
- **GET /respondents/{id}** - Obtiene detalles de un encuestado específico (requiere token).
- **PUT /respondents/{id}** - Actualiza la información de un encuestado (requiere token).
- **DELETE /respondents/{id}** - Elimina un encuestado de la base de datos (requiere token).

Reportes y Análisis

- **GET /surveys/{id}/analysis** - Genera un análisis de las respuestas de una encuesta (requiere token de creador de la encuesta o administrador).

Integración Continua y Despliegue Continuo (CI/CD)

Para asegurar la calidad del software y facilitar despliegues consistentes y automáticos, se solicita a los estudiantes implementar un flujo de trabajo de Integración Continua y Despliegue Continuo (CI/CD) utilizando GitHub Actions o GitLab CI para la construcción automática de la imagen Docker de la aplicación. Además, deberán almacenar la imagen resultante en un repositorio como Docker Hub o GitHub Container Registry.

Construcción Automática con GitHub Actions o GitLab CI

1. **Crear el archivo de configuración:** Dependiendo de la plataforma elegida (GitHub o GitLab), crea un archivo `.yaml` en el directorio raíz del proyecto (`.github/workflows/` para GitHub Actions o `.gitlab-ci.yml` para GitLab CI).
2. **Definir el flujo de trabajo:** Configura el archivo para que ejecute la construcción de la imagen Docker cada vez que se haga push al repositorio o se realice un merge en la rama principal.
3. **Especificar los pasos de construcción:** Incluye instrucciones para construir la imagen Docker de la aplicación, etiquetarla adecuadamente y empujarla a un repositorio de imágenes (Docker Hub, GitHub Container Registry, etc.).

Almacenamiento de la Imagen en un Repositorio

1. **Configurar credenciales de acceso:** Asegúrate de configurar de manera segura las credenciales de acceso al repositorio de imágenes en tu flujo de trabajo de CI/CD.
2. **Almacenamiento de la imagen construida:** Utiliza los comandos adecuados dentro de tu flujo de trabajo CI/CD para empujar la imagen construida al repositorio elegido.
3. **Etiquetado de la imagen:** Adopta una convención de etiquetado coherente para las imágenes (por ejemplo, utilizando el número de versión o el hash del commit) para facilitar el seguimiento de las versiones.

Requerimientos No Funcionales

- Implementación de seguridad para la protección de datos.
- Diseño escalable y de alta disponibilidad.

Etapas del Proyecto

1. Diseño del sistema incluyendo arquitectura, modelado de base de datos y diseño de API Rest.
2. Implementación de la base de datos, lógica de negocio y API Rest.
3. Realización de pruebas unitarias, de integración y de carga.
4. Despliegue del sistema en un entorno de cloud computing.

Documentación

Proveer un archivo README.md con instrucciones claras sobre cómo construir y desplegar la aplicación usando Docker y Docker Compose, incluyendo cómo realizar pruebas básicas.

1 Entregables

1. Código Fuente de la aplicación.
2. Archivo Dockerfile.
3. Archivo docker-compose.yml.
4. Archivo con colección de consultas en Postman
5. Instrucciones de Uso (README).

Debe presentar un archivo comprimido en el TecDigital antes de las 10:00pm del día de entrega. La actividad se puede realizar en parejas.

Si la entrega se realiza después de la hora de entrega, se le penalizará con 5 puntos porcentuales que se acumulan cada 24 horas. Por ejemplo si entrega a las 10:05pm su evaluación tendrá una nota base de 95%, si entrega después de las 10:05 p.m. del siguiente día, su nota base será 90%, y así sucesivamente.

2 Evaluación

3 Recursos Adicionales

Se recomienda consultar la documentación oficial de Docker, Docker Compose, el lenguaje de programación elegido para la API REST, y PostgreSQL.