**Arquitectura de Proyecto**

**Componentes Principales:**

**Front-End (React):**

El front-end estará construido utilizando React y el diseño fue creado en Figma.

Los usuarios interactúan con la interfaz de usuario, que ofrece funcionalidades para la lista de clientes, consulta de clientes y registro/actualización de clientes.

**Back-End (Python y Flask):**

Flask maneja las solicitudes HTTP y proporciona una API REST para las operaciones de gestión de clientes.

Se implementa un sistema de autenticación basado en **JWT**.

Las operaciones en la base de datos se gestionan a través del back-end.

**Base de Datos (MongoDB):**

Se almacenan los detalles de los clientes, incluida la información de aceptación de promociones.

Se tiene una colección que contiene los registros de los clientes manipulable.

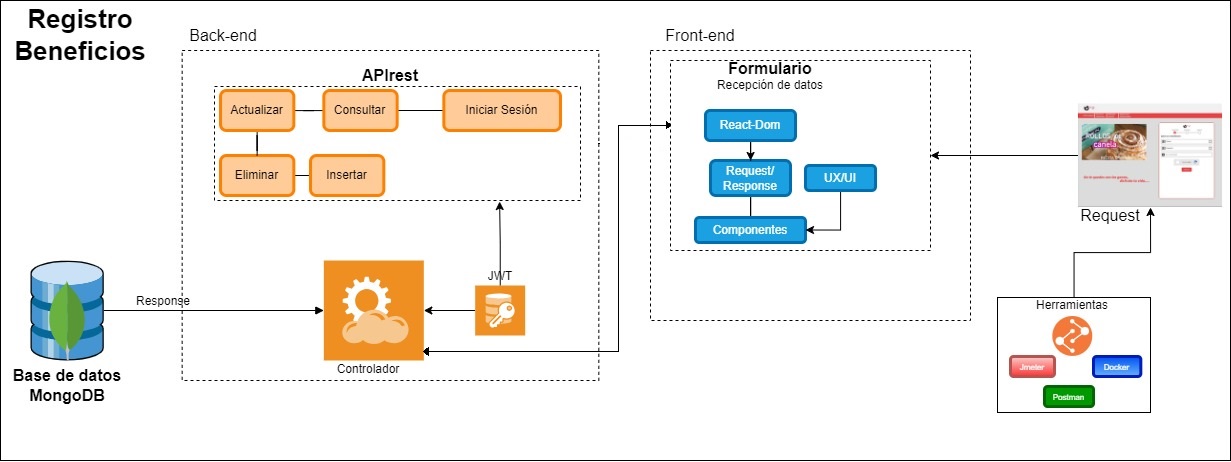
**Docker:**

Se utilizan contenedores Docker para empaquetar y desplegar la aplicación y sus componentes.

**Pruebas de Estrés (JMeter):**

JMeter se utiliza para pruebas de estrés en las operaciones del back-end.

**Diagrama de Arquitectura:**

****

**Flujo de Datos y Funcionalidades:**

1. **Lista de Clientes:**

El front-end consume la API para obtener la lista de clientes desde el back-end.

Los datos se recuperan de la base de datos MongoDB y se presentan en la interfaz de usuario.

1. **Consulta de Clientes:**

Los usuarios pueden buscar y consultar información de clientes.

El front-end envía una solicitud de búsqueda al back-end a través de la API, que recupera los datos de la base de datos y los devuelve al front-end.

1. **Registro/Actualización de Clientes:**

Los usuarios pueden registrar nuevos clientes o actualizar los datos existentes utilizando un formulario en el front-end.

El front-end consume la API para enviar los datos al back-end.

El back-end valida y procesa la solicitud, actualiza los datos en la base de datos MongoDB y devuelve una respuesta al front-end.

1. **Aceptación de Promociones:**

El formulario de registro incluye un campo "Acepta Promociones".

Esta preferencia se almacena en la base de datos MongoDB junto con los demás datos del cliente.

1. **Token:**

Envía un token creado desde la API al front que lo almacena en “localStorage” para verificar y permitir consumir la API al logearse.

1. **Diseño en Figma:**

El diseño del front-end se creó utilizando la herramienta de diseño Figma. El diseño guía para la creación de la interfaz del cliente en React.

**Resumen Ejecutivo**

|  |  |
| --- | --- |
| Título: | Título: Registro de Beneficios por Aceptación de Uso de Datos Personales |
| Fecha: | 21-08-2023 |
| Autor: | Alex Yugcha |
| Organización: | Equipo de Trabajo "Proyectos" |

**Propósito**

El objetivo de este proyecto es registrar a los clientes y su consentimiento para el uso de sus datos personales, a cambio de recibir beneficios, como productos. Se busca recopilar información de los clientes mediante esta interacción.

**Resultado**

Se ha logrado implementar un formulario de registro ágil, comprensible y claro, que agiliza el proceso de aceptación de uso de datos y entrega de beneficios.

**Resumen**

El proyecto se desarrolló mediante la metodología Scrum, con un plazo de 7 semanas y una semana adicional para la presentación final. Se dividieron en total tres sprints, cada uno con entregables específicos. En el **primer sprint**, se realizó una capacitación exhaustiva en las tecnologías a utilizar. En el **segundo sprint**, se llevó a cabo la creación de un API funcional con rutas protegidas para la gestión de datos. El **tercer sprint** se enfocó en la creación del frontend utilizando React, culminando en la entrega de una Land Page completa. Además de la implementación técnica, se elaboró la documentación correspondiente y se realizó una presentación integral del proyecto. Las etapas estructuradas y los entregables concretos aseguraron un desarrollo fluido y exitoso del proyecto.

**Conclusiones y Recomendaciones**

La colaboración y guía del equipo de trabajo resultaron en un desarrollo exitoso, enriqueciendo la perspectiva y conocimientos del autor. Se recomienda el aprendizaje continuo. En cuanto al sistema creado, se sugiere que su uso sea limitado a empresas locales de menor escala.

**Beneficios**

Este proyecto ha permitido al autor adquirir un valioso aprendizaje en la gestión de proyectos y en el uso de diversas tecnologías. Profesionalmente, se encuentra mejor preparado para abordar desafíos y soluciones variados.

**Palabras Clave:**

Registro de datos, beneficios a cambio, apirest, backend, react y frontend, aprendizaje continuo.

**Tecnologías usadas**

|  |  |  |
| --- | --- | --- |
| *Figma* | *Python* | *React* |
| *Notion* | *JMeter* | *Docker* |
| *Postman* | *Docusaurus* | *MongoDB* |