



**ESCUELA POLITÉCNICA NACIONAL**  
**FACULTAD DE INGENIERÍA DE SISTEMAS**  
**Metodologías Ágiles**

---

**PERÍODO ACADÉMICO:** 2023\_B

**ASIGNATURA:** Metodologías Ágiles

**PROFESOR:** Patricio Michael Paccha

---

**TIPO DE INSTRUMENTO:** Informe

**INSTRUMENTO N°:** 1

**TÍTULO:** Sistema de Gestión de Contraseñas

**FECHA DE ENTREGA:** 01/03/2024

**GRUPO:** 1

**ESTUDIANTES:**

---

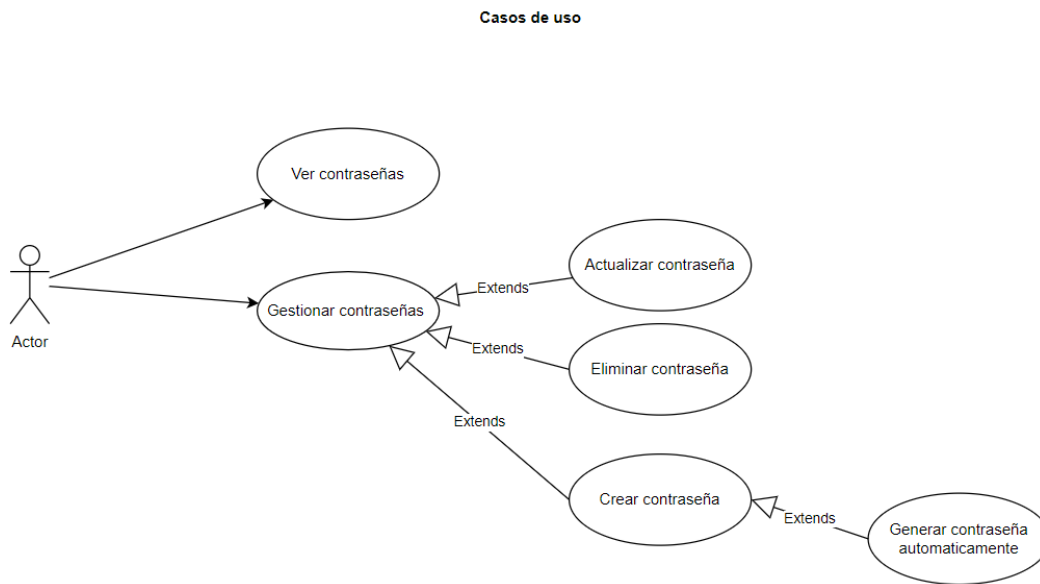
- DAVID ALEXANDER AVEROS OTAVALO
  - RICARDO ANDRES BECERRA VIVANCO
  - NARDY ESTEFANIA CACHIPUENDO CHIGUANO
  - GARY RICARDO CAMPAÑA RAMIREZ
  - KEVIN MATEO CANO CRUCERIRA
  - RAFAEL SEBASTIAN CASTRO PAREDES
- 



## DOCUMENTO DE DISEÑO

### 1. DIAGRAMA DE CASOS DE USO

Para el desarrollo de este proyecto, se elaboró el diagrama de casos de uso debido a que nos brinda una representación clara de la interacción entre el usuario y el sistema. Este diagrama, además, actúa como guía esencial para los desarrolladores, permitiéndoles comprender y aplicar de manera adecuada las necesidades identificadas por los usuarios para la implementación del sistema.



*Ilustración 1: Diagrama de casos de uso*

### 2. DIAGRAMA DE COMPONENTES

Después de identificar los requisitos del usuario para el sistema, se tomó la decisión de crear un diagrama de componentes, porque se optó por desarrollar el proyecto utilizando React. De esta manera, se llevó a cabo la implementación del siguiente diagrama de componentes, el cual se presenta en la Figura 2 para ofrecer una idea clara de esta estructura

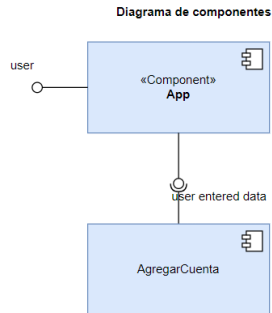


Ilustración 2: Diagrama de componentes

### 3. ARQUITECTURA

Se optó por implementar el proyecto bajo la arquitectura cliente-servidor con despliegue en Docker.

Docker funciona con una arquitectura de tipo cliente servidor. El cliente se comunica con Docker Daemon, la pieza responsable de construir, desplegar y distribuir los contenedores. Ambos componentes pueden ejecutar en el mismo sistema o de forma remota comunicándose a través de la API.

El cliente de Docker es la interfaz de línea de comandos que transmite las instrucciones de los usuarios. Después de haber transmitido una orden desde el cliente hasta el demonio, éste se descarga la versión que necesite del Registry, a continuación crea el contenedor a partir de la imagen y lo ejecuta en el host.

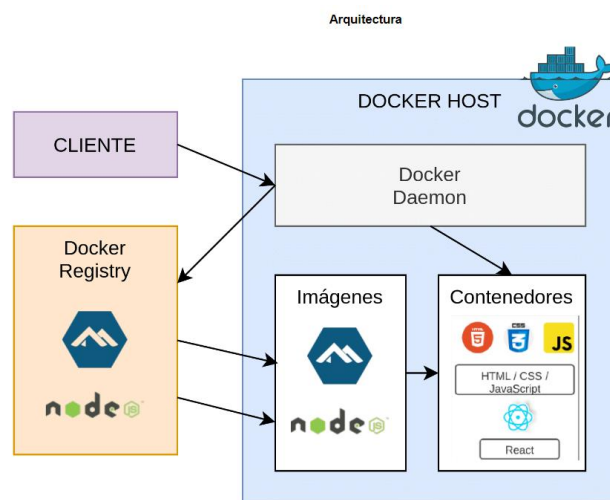


Ilustración 3: Arquitectura

**Componentes:**

- Cliente: Aplicación web desarrollada con React que se ejecuta en el navegador del usuario.
- Imagen: es un paquete ejecutable, ligero e independiente, que contiene todo lo necesario para ejecutar cierto software. Una imagen se compone de múltiples capas, una por cada dockerfile definido.
- Docker Registry: es el lugar para almacenar imágenes y distribuirlas fácilmente. Existen registros públicos como Docker Hub o Docker Cloud. También se pueden almacenar en el registro local privado.
- Contenedor: es una instancia en tiempo de ejecución de una imagen. Consiste en una aplicación aislada construida a partir de una o más imágenes. Los contenedores no empaquetan sistema operativo, sino que usan el del host.
- Dockerfile: es un documento de texto con los comandos necesarios para ejecutar la imagen.