propuesta

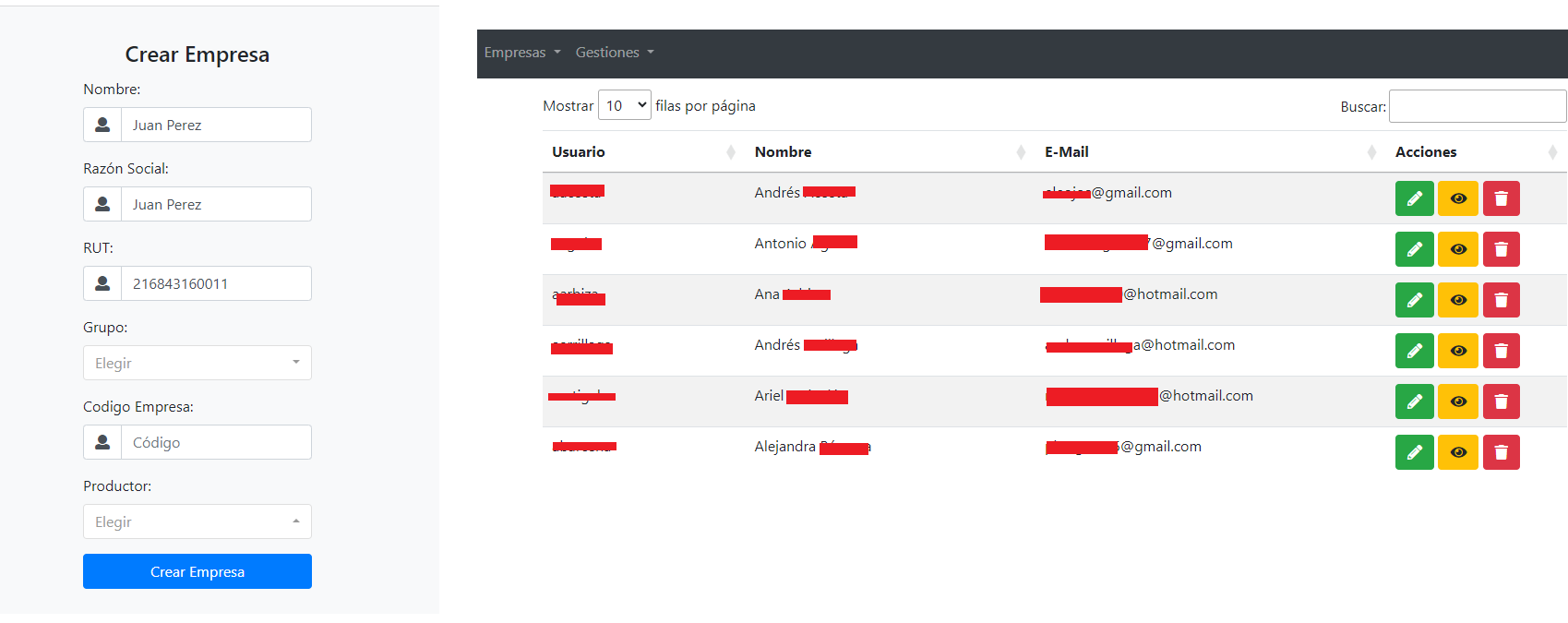
2024-08-20

FUCREA es una organización que nuclea a más de 500 productores rurales de Uruguay. Estos productores se organizan en grupos de 10 o 20 empresas que tienen un asesor especializado. Estos asesores relevan información de resultados productivos y económicos de las empresas. El objetivo de este proyecto es facilitar el uso de estos datos para la toma de decisiones, generando análisis y visualizaciones de estos datos. Los principales desafíos del proyecto son diseñar un mecanismo de autorización que refleje la lógica de negocios de FUCREA y garantice que los usuarios solo acceden a la información para la que tienen permisos, y desarrollar las visualizaciones interactivas de los datos de manera que permitan a los usuarios sacar conclusiones valiosas a partir de la información disponible.

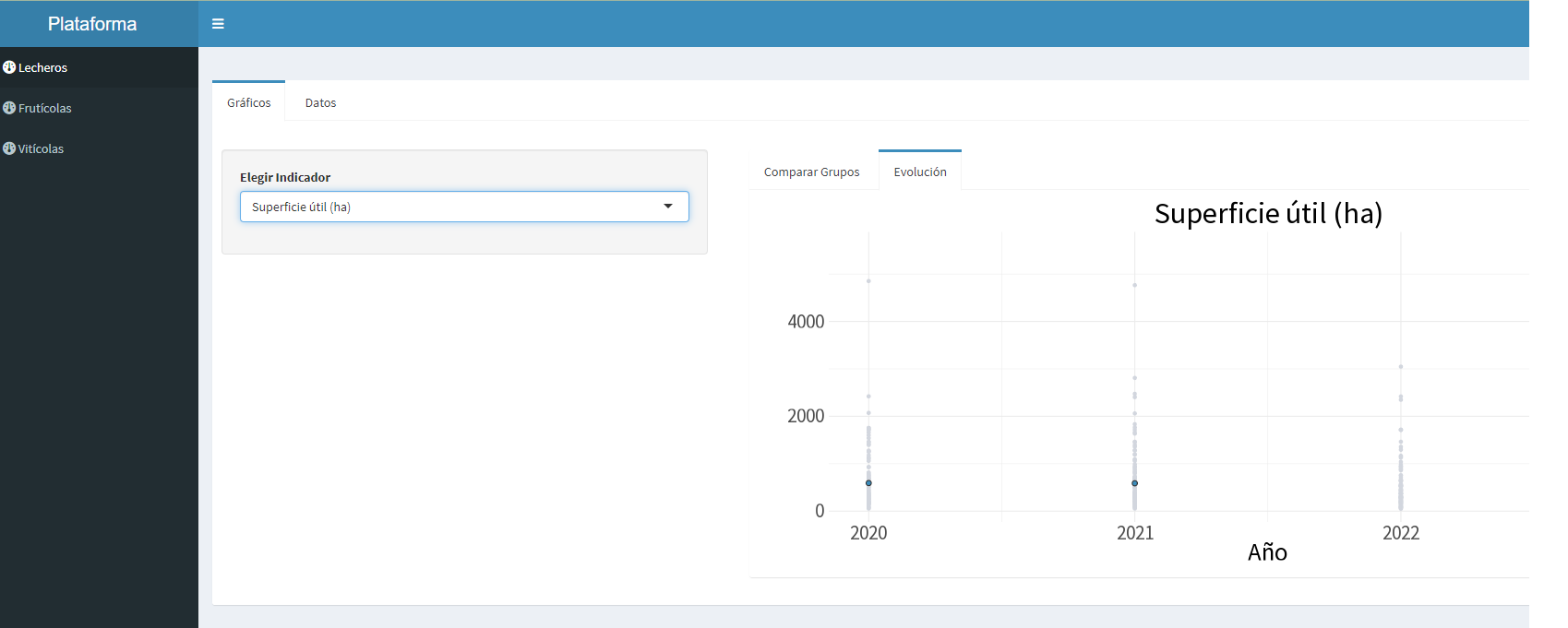
Esta presentación explora cómo combinar R y Shiny con Node usando Docker para lograr este objetivo. La solución final garantiza la seguridad y privacidad de los datos, implementando un modelo de autorización que refleja la lógica de negocio, y permite visualizarlos en una aplicación interactiva.

Los grupos CREA estan organizados por sectoriales que agrupan empresas de rubros similares (ganaderas, lecheras, agrícolas). A su vez, las sectoriales se dividen en grupos de 10 o 20 empresas, que reciben asistencia técnica de un asesor. Esto hace necesario que el sistema tenga mútiples roles con permisos diferenciados: los coordinadores generales pueden acceder a todos los datos, los coordinadores sectoriales solo a los de las empresas de su sector, los asesores solo pueden acceder a empresas de sus grupos y los productores solo pueden acceder a los datos de sus empresas.

A su vez, este esquema jerárquico debe permitir que hay asesores que tienen varios grupos, productores con más de una empresa, etc. Estas relaciones dinámicas son administradas por el staff de FUCREA en una aplicación Web y una base de datos relacional. Los coordinadores, asesores y productores acceden a un dashboard desarrollado en Shiny.



El dashboard consume información de la base de datos a través de una API, por lo que los cambios que los asesores realizan a través de la aplicación Web se reflejan en tiempo real en el dashboard.



Configurar y mantener la infraestructura para que todos estos componentes funcionen correctamente en ambientes de desarrollo, testing y producción puede ser un costo considerable. Docker permite automatizar este proceso y reducir considerablemente estos costos. Todos los componentes del sistema corren en una instancia AWS con Docker como único requisito de software.

Una imagen Docker contiene todo el software necesario para correr una aplicación. Todas las herramientas utilizadas en este proyecto tienen imagenes (semi) oficiales de Docker disponibles. Si tenemos Docker instalado, podemos descargar estas imágenes y ejecutar un contenedor que corre estos procesos con un solo comando. La imagen incluye todos las dependencias y configuraciones necesarias para ejecutar R, Shiny, Postgres o Node.js. Podemos customizadar las imagenes oficiales, agregando otras librerías y nuestras propias aplicaciones.

docker image build .

Este comando toma la imágen oficial, instala las dependencias adicionales y copia el código de la aplicación Shiny. La imágen encapsula el código de la aplicación y sus dependencias. Una vez creada la imágen, podemos crear un contenedor, un proceso que ejecuta el código de la aplicación.

Docker Compose es una herramienta complementaria a Docker, que permite orquestar varios contenedores. Docker Compose centraliza en un solo archivo (docker-compose.yml) en un lenguaje declarativo todos los componentes del sistema. Esto incluye los contenedores Docker para ejecutar Postgres, la aplicación Node y el dashboard Shiny. Red, puertos, volumenes.

docker-compose up -d

Este comando crea todos los contenedores especificados, los volúmenes y la red. También configura los volumenes para comunicarse por los puertos correctos. Esto hace que toda la infraestructura esté centralizada en archivos yml y nos evita debugear variables de entorno y parámetros de configuración.

Las herramientas presentadas demuestran cómo R no solo es una herramienta de análisis, sino que puede ser parte de un sistema más amplio que incluye gestión de datos, seguridad, y despliegue automatizado. Docker facilita la interoperabilidad de los sistema explotando las capacidades del equipo en forma óptima.