

# Integración de Shiny con Aplicaciones WebNode.js para visualización y análisis de datos de empresas agropecuarias

**Keywords:** “NodeJS”, “Docker”, “Postgres”, “Shiny”, “AWS”

Rafael La Buonora

2024-08-20

Esta presentación muestra la integración Node JS, Posgres y Shiny para inteligencia de negocios en el sector agropecuario. 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 la organización 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 clientes sacar conclusiones valiosas a partir de la información disponible.

Esta presentación explora cómo combinar Postgres, Node 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.

El sistema soporta la creación y administración de múltiples roles con permisos de acceso diferenciados a los datos: 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.

La presentación se organiza de la siguiente manera: en la primera sección se presentan las bases de datos relacionales y Postgresql como una forma segura y escalable de almacenar grandes volúmenes de datos en tablas interrelacionadas.

También se presenta Sequelize, un Object Relational Mapping basado en NodeJS que permite convertir las filas de la base de datos relacional a objetos que modelan las reglas de negocio de la organización. Esto permite implementar el modelo de autorización para que los usuarios solo accedan a la información para la que tienen permisos. Además, permite que los administradores configuren estos permisos.

También se presenta brevemente Shiny, un framework implementado en R para realizar aplicaciones web interactivas que permitan visualizar datos usando las herramientas disponibles en el ecosistema R como ggplot.

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.

La última parte de la presentación muestra como se utiliza Docker para integrar la base de datos relacional implementada en Postgres, la aplicación Web y el dashboard Shiny en un ambiente de producción hospedados en AWS.

Una imagen Docker contiene todo el software necesario para correr una aplicación. Todas las herramientas utilizadas en este proyecto tienen imágenes (semi) oficiales de Docker disponibles. Con 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 customizar las imágenes oficiales, agregando otras librerías y nuestras propias aplicaciones.

Para correr nuestra aplicación Shiny en un contenedor, partimos de una imagen disponible en [Docker Hub](#), instalamos las dependencias adicionales y copiamos el código de la aplicación Shiny. La imagen encapsula el código de la aplicación y sus dependencias. Una vez creada la imagen, podemos crear un contenedor a partir de esta imagen. Este contenedor es 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 en un lenguaje declarativo todos los componentes del sistema: Postgres, la aplicación Node y el dashboard Shiny. Este comando crea todos los contenedores especificados, los volúmenes donde se almacenan datos persistentes y una red que permite que estén interconectados. Docker Compose permite que 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 sistemas explotando las capacidades del equipo en forma óptima.