**Universidad de Sonora**

División de Ingeniería

Logotipo

Descripción generada automáticamente

**PROYECTO FINAL – DOCKER Y KUBERNETES**

**APLICACIÓN REACT CON NODEJS POSTGRES Y NGINX**

Bermúdez Rodríguez Lucía Carolina

Reynoso Aguirre Noe Benjamín

Guzmán Silvas Erica Jazmín

Félix Valenzuela Víctor Hugo

**Maestro: Álvaro Alfonso Valenzuela Figueroa**

Hermosillo, Sonora. 24 de noviembre de 2022.

Contenido

[Desarrollo entorno Docker 5](#_Toc144732617)

[Introducción a la arquitectura con Docker 5](#_Toc144732618)

[Creación del repositorio en GitHub 5](#_Toc144732619)

[Creación de la estructura del proyecto 7](#_Toc144732620)

[Carpeta k8s 7](#_Toc144732621)

[client-cluster-ip-service.yml 7](#_Toc144732622)

[client-deployment.yml 7](#_Toc144732623)

[database-persistent-volume-claim.yml 8](#_Toc144732624)

[ingress-service.yml 8](#_Toc144732625)

[postgres-cluster-ip-service.yml 9](#_Toc144732626)

[postgres-deployment.yml 9](#_Toc144732627)

[sc.yaml 10](#_Toc144732628)

[server-cluster-ip-service.yml 10](#_Toc144732629)

[server-deployment.yml 11](#_Toc144732630)

[Carpeta nginx 12](#_Toc144732631)

[Default.conf 12](#_Toc144732632)

[Dockerfile.dev 12](#_Toc144732633)

[Carpeta client 12](#_Toc144732634)

[Dockerfile 13](#_Toc144732635)

[Dockerfile.dev 13](#_Toc144732636)

[Package.json 13](#_Toc144732637)

[Carpeta src 14](#_Toc144732638)

[App.css 14](#_Toc144732639)

[App.js 15](#_Toc144732640)

[Maincomponent.css 16](#_Toc144732641)

[OtherPage.js 16](#_Toc144732642)

[Maincomponent.js 17](#_Toc144732643)

[Carpeta nginx 18](#_Toc144732644)

[Default.conf 18](#_Toc144732645)

[Carpeta server 18](#_Toc144732646)

[package.json 19](#_Toc144732647)

[index.js 20](#_Toc144732648)

[keys.js 21](#_Toc144732649)

[Dockerfile 21](#_Toc144732650)

[Dockerfile.dev 21](#_Toc144732651)

[docker-compose.yml 22](#_Toc144732652)

[Comandos para probar con Docker 23](#_Toc144732653)

[Desarrollo entorno Kubernetes 24](#_Toc144732654)

# Desarrollo entorno Docker

## Introducción a la arquitectura con Docker

Lo que se muestra en los siguientes apartados es el proceso de creación de nuestro proyecto final que consta de una aplicación multi-container que se ejecuta dentro de contenedores Docker, para este proyecto también creamos una versión de producción y una versión de desarrollo.

Hicimos también uso de Docker Compose para ejecutar nuestro entorno completo localmente (frontend, backend, nginx y postgres) ya que con Docker Compose es posible ejecutar todos estos recursos con un solo comando.

Como se puede ver en el siguiente diagrama, para el frontend usamos react, para el backend express node.js server, para la base de datos se usó postgres y nginx se utilizó para dos propósitos, uno para enrutamiento y otro como un servidor http.

Interfaz de usuario gráfica

Descripción generada automáticamente

Para seguir los siguientes pasos se requiere la instalación de Docker Desktop, Kubernetes y nodeJS

## Creación del repositorio en GitHub

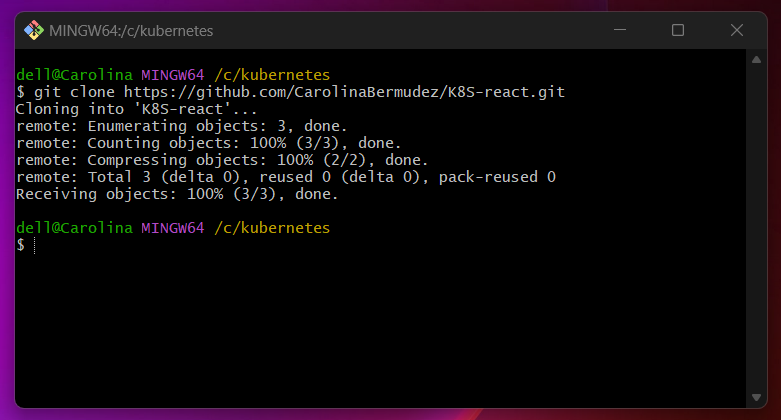
Se creó el repositorio con nombre K8S-react

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Para la clonación del repositorio la ubicación utilizada es en *C:\kubernetes*

* Click derecho Git Bash Here



* Dentro de *C:\kubernetes\K8S-react* hay que crear el archivo .gitignore

Captura de pantalla de un celular

Descripción generada automáticamente

## Creación de la estructura del proyecto

### Carpeta k8s

#### client-cluster-ip-service.yml

Texto

Descripción generada automáticamente

#### client-deployment.yml

Texto

Descripción generada automáticamente

#### database-persistent-volume-claim.yml

Texto

Descripción generada automáticamente

#### ingress-service.yml

Texto

Descripción generada automáticamente

#### postgres-cluster-ip-service.yml

Texto

Descripción generada automáticamente

#### postgres-deployment.yml

Texto

Descripción generada automáticamente

#### sc.yaml

Texto

Descripción generada automáticamente

#### server-cluster-ip-service.yml

Texto

Descripción generada automáticamente

#### server-deployment.yml

Texto

Descripción generada automáticamente

### Carpeta nginx

#### Default.conf

Texto

Descripción generada automáticamente

#### Dockerfile.dev

Texto

Descripción generada automáticamente

### Carpeta client

* Para crear esta carpeta hay que crear la aplicación react llamada client con el siguiente comando en el directorio /kubernetes/K8S-react

npx create-react-app client

Texto

Descripción generada automáticamente

* Una vez creada la carpeta client se crean los siguientes archivos

#### Dockerfile

Texto

Descripción generada automáticamente

#### Dockerfile.dev

Texto

Descripción generada automáticamente

* Se modifican los siguientes archivos

#### Package.json

Solo se agregan estas dependencias

Texto

Descripción generada automáticamente

• Dentro de la carpeta client se corre el siguiente comando

npm i

#### Carpeta src

* Se modifican los archivos siguientes

##### App.css

Texto

Descripción generada automáticamente

##### App.js

Texto

Descripción generada automáticamente

* Se agregan tres archivos más

##### Maincomponent.css

Texto

Descripción generada automáticamente

##### OtherPage.js

Texto

Descripción generada automáticamente

##### Maincomponent.js

Texto

Descripción generada automáticamente

#### Carpeta nginx

Esta carpeta se crea manualmente y se agrega el siguiente archivo

##### Default.conf

Texto

Descripción generada automáticamente

* Dentro de la carpeta client se corre el siguiente comando

npm i

### Carpeta server

* Esta carpeta se crea manualmente y se ejecuta el siguiente comando

npm init

Texto

Descripción generada automáticamente

#### package.json

Imagen de la pantalla de un computador

Descripción generada automáticamente con confianza baja

#### index.js

Texto

Descripción generada automáticamente

#### keys.js

Texto, Chat o mensaje de texto

Descripción generada automáticamente

#### Dockerfile

Texto

Descripción generada automáticamente

#### Dockerfile.dev

Texto

Descripción generada automáticamente

* Dentro de la carpeta server se corre el siguiente comando

npm i

### docker-compose.yml

Texto

Descripción generada automáticamente

## Comandos para probar con Docker

Texto

Descripción generada automáticamente

# Desarrollo entorno Kubernetes

Implementamos kubernetes en un entorno de desarrollo local, en el desarrollo de Docker se crearon imágenes del cliente y del server que se utilizaron para crear los recursos para kubernetes.

Se creo un archivo de configuración para cada objecto, en total serian ocho archivos y también se creó un pod por cada deployment, por último, para la base de datos se usó persistent volumen claim para postgres.

Se habilitará Ingres Service para poder acceder a la aplicación una vez ejecutados todos los manifiestos

Diagrama

Descripción generada automáticamente

Creacion de password

Texto

Descripción generada automáticamente

Instalación ingress nginx, también se puede instalar mediante el comando

minikube addons enable ingress

Texto

Descripción generada automáticamente Texto

Descripción generada automáticamente

Manifiestos k8s

Texto

Descripción generada automáticamente

Para acceder a la aplicación a través de localhost

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

Ahora al ingresar al localhost todo funciona correctamente y se muestra una página como esta

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

