

Clase 21: Despliegue de aplicaciones web con Render

Sitio: [Centro de E-Learning - UTN.BA](#)

Curso: Curso de Backend Developer - Turno
Noche

Libro: Clase 21: Despliegue de aplicaciones web
con Render

Imprimido

por:

Día:

Nilo Crespi

Friday, 23 de January de 2026,
10:53

Descripción

Objetivos

- Comprender el proceso de despliegue de una aplicación web.
- Conocer los conceptos teóricos implicados en el despliegue de aplicaciones.
- Aprender a desplegar una aplicación utilizando la plataforma Render

Tabla de contenidos

- 1. Introducción
- 2. Despliegue de Aplicaciones Web
- 3. Render
- 4. Espacio de práctica

1. Introducción

El despliegue de aplicaciones web es un proceso fundamental para llevar un proyecto de desarrollo desde el entorno local hasta un servidor accesible públicamente. En este contexto, Render se presenta como una plataforma en la nube que facilita este proceso, ofreciendo una integración sencilla con GitHub, soporte para diversas tecnologías como Node.js y React, y características como escalabilidad automática y bases de datos administradas.

Este material abordará el proceso de despliegue utilizando Render, desde la preparación del código hasta la configuración del entorno de producción. Se explicarán conceptos clave como CI/CD, variables de entorno y seguridad en producción, proporcionando una guía práctica para desplegar aplicaciones full-stack con React, Node.js y MongoDB Atlas.

2. Despliegue de Aplicaciones Web

Despliegue de Aplicaciones Web

El despliegue de aplicaciones web es el proceso de hacer que una aplicación esté disponible en la web para que los usuarios puedan interactuar con ella. Esto incluye subir el código a un servidor, configurar bases de datos y asegurarse de que todo funcione correctamente en producción.

Pasos comunes en el despliegue de una aplicación web

Preparación del código:

Antes de hacer el despliegue, debemos asegurarnos de que el código de la aplicación esté optimizado para producción. Esto incluye:

- Minimizar archivos JavaScript y CSS.
- Eliminar código no utilizado.
- Configurar correctamente las variables de entorno (por ejemplo, claves de API, credenciales de base de datos).

Configuración del servidor:

Una vez que el código está listo, necesitamos configurarlo en un servidor. Esto puede hacerse utilizando plataformas como Render, AWS, Heroku, entre otras.

Subir el código al servidor:

Utilizamos herramientas como Git para subir el código desde nuestro repositorio a la plataforma de despliegue. Render, por ejemplo, se integra directamente con GitHub.

Configuración del entorno de producción:

En el entorno de producción, es necesario realizar configuraciones adicionales como la conexión a bases de datos, el manejo de variables de entorno y la optimización de la red.

Despliegue de la base de datos:

Si la aplicación utiliza una base de datos, debemos asegurarnos de que esté configurada y accesible desde la aplicación en producción.

Pruebas de producción:

Una vez desplegada, debemos realizar pruebas en el entorno de producción para asegurarnos de que todo funcione correctamente.

3. Render

Render: plataforma para el despliegue de aplicaciones web

Render es una plataforma en la nube que facilita el despliegue de aplicaciones web sin necesidad de gestionar servidores.

Características principales de Render:

- Despliegue automático desde GitHub.
- Soporte para Node.js, Python, React, Vue, entre otros.
- Escalabilidad automática según el tráfico.
- Alojamiento de Backend y Frontend.
- Bases de datos administradas (MongoDB, PostgreSQL).
- Certificados SSL gratuitos para HTTPS.

Conceptos teóricos implicados en el despliegue

CI/CD (Integración Continua y Entrega Continua)

- CI (Integración Continua) es un proceso donde todos los desarrolladores integran sus cambios al código principal con frecuencia. Las pruebas se ejecutan automáticamente para detectar errores tempranos.
- CD (Entrega Continua) es el proceso donde el código validado por CI se despliega automáticamente en producción. Esto reduce los tiempos de entrega.
- **Contenedores y Docker:** Docker es una tecnología que permite ejecutar aplicaciones de manera aislada. Aunque Render maneja la infraestructura, es útil entender cómo los contenedores funcionan para aplicaciones más avanzadas.
- **Variables de Entorno:** son configuraciones necesarias para la aplicación, como credenciales para bases de datos o claves API. Render permite configurarlas directamente en su plataforma.
- **Seguridad:** En producción, se utiliza SSL para asegurar la comunicación entre el cliente y el servidor. Render proporciona certificados SSL gratuitos.

Proceso de despliegue en Render

Aquí se detallan los pasos para desplegar una aplicación en Render, utilizando como ejemplo una

aplicación con un frontend en React y un backend en Node.js con MongoDB.

Paso 1: Preparar el código

Primero, debemos asegurarnos de que nuestro código esté listo para producción. Esto incluye optimizar los archivos y asegurarnos de que las variables de entorno estén configuradas correctamente.

Ejemplo: Supongamos que tienes un proyecto con un backend en Node.js y un frontend en React.

Frontend (front):

- Configurar puerto y el host en Vite, en el archivo vite.config.js
- Ejecuta npm run build para crear la versión optimizada de tu aplicación React.
- Esto generará una carpeta dist con todos los archivos necesarios para producción.

Backend (back):

- Asegúrate de que tu archivo server.js esté configurado para producción (como la configuración de puertos y bases de datos).

Paso 2: Subir el código a GitHub y configurar bases de datos

- Crear un repositorio en **GitHub** para que Render pueda acceder al código y hacer el despliegue.

Paso 3: Configuración Mongo Atlas

- Crear Cluster, (plan gratuito) y creamos la base de datos "Create Database".
- En MongoAtlas conectar a compass, exportar e importar datos.
- Habilitar IP de Render en MongoAtlas, IP 0.0.0.0/0, (Network Access) -

Paso 4: Crear una cuenta en Render

1. Regístrate en Render.
2. Vincula tu cuenta de GitHub con Render.

Paso 5: Crear un nuevo servicio en Render

1. Dentro de la plataforma de Render, selecciona New Web Service.
2. Conecta el repositorio de GitHub donde tienes tu aplicación.
3. Selecciona el branch de GitHub que quieres desplegar (generalmente main).
4. Define los comandos de construcción y ejecución para el backend y el frontend:

Para el backend (Node.js), el comando podría ser: `npm install`

Para el frontend (React), el comando podría ser: `npm install && npm run build`

Paso 6: Configurar variables de entorno

En Render, necesitas configurar las variables de entorno, como las credenciales de la base de datos o las claves de API.

1. En Render, ve a la sección Environment.
2. Añade las variables necesarias, por ejemplo:

DB_URI: La URI de conexión de tu base de datos MongoDB.

API_KEY: Una clave para alguna API externa que utilices.

Paso 7: Desplegar la aplicación

Render empezará a construir tu aplicación y, si todo va bien, desplegará el código automáticamente en el entorno de producción. Esto incluirá la creación de la infraestructura necesaria para el backend y el frontend.

Una vez que se complete el proceso, Render proporcionará una URL pública donde podrás acceder a tu aplicación.

Paso 8: Verificar el despliegue

Una vez desplegada la aplicación, visita la URL proporcionada por Render y verifica que todo funcione correctamente.

1. Si todo está bien, deberías poder ver tu aplicación en funcionamiento.
2. Si tenés algún problema, consulta los logs de Render para verificar si hay errores en la construcción o en la ejecución.

Paso 9: Realizar pruebas de producción

Después de desplegar la aplicación, realiza pruebas de funcionalidad para asegurarte de que todo funciona como se espera en el entorno de producción.

Front

Build Command Render runs this command to build your app before each deploy.	<input type="text" value="\$ cd front && npm install && npm run build"/> <div>Cancel Save Changes</div>
Pre-Deploy Command Optional Render runs this command before the start command. Useful for database migrations and static asset uploads.	<input type="text" value="\$"/> <div>Edit</div>
Start Command Render runs this command to start your app with each deploy.	<input type="text" value="\$ cd front && npm run preview"/> <div>Edit</div>
Auto-Deploy By default, Render automatically deploys your service whenever you update its code or	<div>Yes</div> <div>Edit</div>

Back - configuraciones

Build Command Render runs this command to build your app before each deploy.	<input type="text" value="\$ cd back && npm install"/> <div>Edit</div>
Pre-Deploy Command Optional Render runs this command before the start command. Useful for database migrations and static asset uploads.	<input type="text" value="\$"/> <div>Edit</div>
Start Command Render runs this command to start your app with each deploy.	<input type="text" value="\$ node back/server.js"/> <div>Edit</div>
Auto-Deploy	<input type="text" value=""/>

Back - Variables de entorno

Events

Logs

Disks

Environment

Shell

Previews

Jobs

Metrics

Scaling

Settings

Environment

+ Create environment group

Environment Variables

Set environment-specific config and secrets (such as API keys), then read those values from your code. [Learn more.](#)

Key

Value

MONGODB_URI

.....

👁

SESSION_SECRET

.....

👁

Edit

11 de 12

23/1/2026, 10:53

4. Espacio de práctica

Espacio de práctica

Desplegar una aplicación web full-stack utilizando React (o Vite) para el frontend, Node.js con Express para el backend, y MongoDB Atlas para la base de datos, haciendo que la aplicación esté accesible públicamente a través de Render.