

2 DAW - Despliegue de aplicaciones web

Área personal ▶ G.Sup - Desarr. de aplic. web ▶ 2° DAW ▶ daw2-despliegue ▶ Tema 4: Despliegue de aplicaciones ▶ Práctica Tema 4

NAVEGACIÓN





Área personal

Inicio del sitio

Páginas del sitio

Curso actual

daw2-despliegue

Participantes

Insignias

Información general

Tema 1: Introducción a Node.js y Express.js

Tema 2: Acceso a datos

con Node.js

Tema 3: Opciones avanzadas de

Express.js

Tema 4: Despliegue de aplicaciones

4.1. Despliegue de aplicaciones en servidores remotos

4.2. Uso básico de Docker

Práctica Tema 4

En esta última práctica haremos un despliegue dividido en etapas de la práctica del tema anterior. A continuación se irán desglosando esas etapas.

Etapa 0

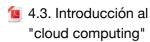
Como pasos previos antes de comenzar el despliegue, debemos tener un **VPS** contratado en algún proveedor de hosting, como por ejemplo OVH, y una cuenta registrada en algún sistema de control de versiones distribuido, como **GitHub**, BitBucket o similar.

Etapa 1: subida de la web a través del repositorio

Los primeros pasos que haremos en el despliegue de la aplicación serán:

- 1. Subir la web a un repositorio público en nuestra cuenta de GitHub o similar, sin incluir la carpeta node modules (MUY importante no incluirla, porque ocupará mucho)
- 2. Acceder por SSH al VPS y clonar con el comando "git clone" la web en alguna carpeta bien localizada. Por ejemplo, en /home/usuario/webs/recetas, siendo "usuario" el nombre de algún usuario de tu sistema operativo en el VPS. Recuerda que al hacer "git clone" se crea una carpeta con el nombre del proyecto en el sistema operativo.
- 3. Instalar MongoDB, Node y Apache en el VPS si no están aún instalados, siguiendo los pasos explicados en la sección 4.1
- 4. Asociar Node y Apache a través de Passenger, como se explica también en la sección 4.1, y asociar la web de recetas con un nombre de dominio que tengáis adscrito al VPS, por el puerto que queráis.

Tras estos pasos, suponiendo que vuestro nombre de dominio sea *vps112233.ovh.net* y la hayáis instalado en el puerto 8080, La web debería estar operativa con la URL *http://vps112233.ovh.net:8080*.





Mis cursos

ADMINISTRACIÓN







- Editar ajustes
- Roles asignados localmente
- Permisos
- Compruebe los permisos
- Filtros
- Registros
- Copia de seguridad
- Restaurar
- Calificación avanzada
- Ver libro de calificaciones
- Ver/Calificar todas las entregas
- Descargar todas las entregas

Administración del curso

Cambiar rol a...

AGREGAR UN BLOQUE

Agregar ...



Etapa 2.

En esta segunda etapa de la práctica, instalaremos Docker en el VPS, descargaremos una imagen de *mongo* y dejaremos en ejecución un contenedor con las siguientes características:

- Se deberá ejecutar en segundo plano
- Debe reiniciarse siempre que no se detenga manualmente
- Debe mapear el puerto 27020 del *host* al puerto por defecto de Mongo en el contenedor (27017). En el caso de que os dé problemas el puerto 27020 por cualquier motivo, deberéis indicarlo expresamente en el documento a entregar, explicando también en este caso qué otro puerto habéis escogido.
- Debe mapear la carpeta *datos-mongo-docker* de vuestra carpeta de usuario en el *host* (deberéis crear esta carpeta) con la carpeta *data/db* del contenedor.
- La aplicación de recetas deberá conectar ahora a esta otra base de datos, en lugar de a la inicial ubicada en el propio VPS.

Etapa 3.

En esta tercera etapa de la práctica, se pide que desarrolléis una aplicación con el framework **Nest.js** llamada **recetas-nest,** que también deberéis subir a un repositorio público en GitHub.

La aplicación tendrá un módulo llamado *receta*, con un controlador y servicio del mismo nombre (*receta*), y definiremos en el controlador dos rutas:

- Una que responderá a la raíz del controlador (/receta), y devolverá en formato JSON todas las recetas de la base de datos.
- Otra que responderá a la ruta /receta/:id y devolverá los datos de la receta cuyo id se especifica.

Desde el controlador, se utilizará el servicio asociado para realizar las correspondientes búsquedas. La aplicación debe conectar a la base de datos contenerizada en Docker en el paso anterior.

Una vez finalizada la aplicación se deberá desplegar en **Heroku**. Sigue los pasos indicados en el apartado 4.3 para configurar la aplicación para poderse desplegar de forma exitosa en Heroku.

Entrega

Se deberá entregar un documento PDF con la siguiente información

Etapa 1

- URL del repositorio GitHub donde esté subido el proyecto de Recetas_V3. Recuerda que NO debe contener la carpeta "node_modules", y que debe ser un repositorio público.
- URL a la que conectar en el VPS para poder probar la web, indicando el puerto, en el caso de que no sea el puerto por defecto (80).

Etapa 2

- Comando a teclear en el terminal del VPS para poner en marcha el contenedor con las características indicadas.
- Comando para conectar a esta otra base de datos desde nuestra web de recetas con Mongoose, en lugar de a la base de datos original.

Etapa 3

- URL del repositorio GitHub con el proyecto Nest.js. Recuerda que tampoco debe contener la carpeta "node_modules", pero sí la carpeta "dist" para su posterior despliegue en Heroku.
- URL de acceso a la aplicación en Heroku.

Sumario de calificaciones

Participantes	13
Enviados	0
Pendientes por calificar	0
Fecha de entrega	domingo, 16 de febrero de 2020, 23:55
Tiempo restante	24 días 10 horas
	Ver/Calificar todas las entregas Calificación

i Moodle Docs para esta página

I.E.S. San Vicente, 2019

Usted se ha identificado como Nacho Iborra Baeza (Salir) daw2-despliegue