

SERVIDOR CON BALANCE DE CARGA

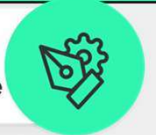
Retomemos nuestro trabajo para poder ejecutar el servidor en modo fork o cluster, ajustando el balance de carga a través de Nginx.

EJECUTAR SERVIDORES NODE

Formato: link a un repositorio en Github con el proyecto cargado.

Sugerencia: no incluir los node_modules

Desafío
entregable



>> Consigna:

Tomando con base el proyecto que vamos realizando, agregar un parámetro más en la ruta de comando que permita ejecutar al servidor en modo fork o cluster. Dicho parámetro será 'FORK' en el primer caso y 'CLUSTER' en el segundo, y de no pasarlo, el servidor iniciará en modo fork.

- Agregar en la vista info, el número de procesadores presentes en el servidor.
- Ejecutar el servidor (modos FORK y CLUSTER) con nodemon verificando el número de procesos tomados por node.
- Ejecutar el servidor (con los parámetros adecuados) utilizando Forever, verificando su correcta operación. Listar los procesos por Forever y por sistema operativo.

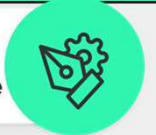
CODER HOUSE

EJECUTAR SERVIDORES NODE

Formato: link a un repositorio en Github con el proyecto cargado.

Sugerencia: no incluir los node_modules

Desafío
entregable



- Ejecutar el servidor (con los parámetros adecuados: modo FORK) utilizando PM2 en sus modos modo fork y cluster. Listar los procesos por PM2 y por sistema operativo.
- Tanto en Forever como en PM2 permitir el modo escucha, para que la actualización del código del servidor se vea reflejado inmediatamente en todos los procesos.
- Hacer pruebas de finalización de procesos fork y cluster en los casos que corresponda.

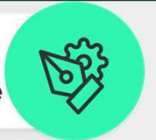
CODER HOUSE

SERVIDOR NGINX

Formato: link a un repositorio en Github con el proyecto cargado.

Sugerencia: no incluir los node_modules

Desafío
entregable



>> Consigna:

Configurar Nginx para balancear cargas de nuestro servidor de la siguiente manera:

Redirigir todas las consultas a `/api/randoms` a un cluster de servidores escuchando en el puerto 8081. El cluster será creado desde node utilizando el módulo nativo `cluster`.

El resto de las consultas, redirigirlas a un servidor individual escuchando en el puerto 8080.

Verificar que todo funcione correctamente.

Luego, modificar la configuración para que todas las consultas a `/api/randoms` sean redirigidas a un cluster de servidores gestionado desde nginx, repartiéndolas equitativamente entre 4 instancias escuchando en los puertos 8082, 8083, 8084 y 8085 respectivamente.

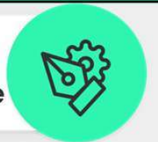
CODER HOUSE

SERVIDOR NGINX

Formato: link a un repositorio en Github con el proyecto cargado.

Sugerencia: no incluir los node_modules

Desafío
entregable



>> Aspectos a incluir en el entregable:

Incluir el archivo de configuración de nginx junto con el proyecto.

Incluir también un pequeño documento en donde se detallen los comandos que deben ejecutarse por línea de comandos y los argumentos que deben enviarse para levantar todas las instancias de servidores de modo que soporten la configuración detallada en los puntos anteriores.

Ejemplo:

- `pm2 start ./miservidor.js -- --port=8080 --modo=fork`
- `pm2 start ./miservidor.js -- --port=8081 --modo=cluster`
- `pm2 start ./miservidor.js -- --port=8082 --modo=fork`
- ...

CODER HOUSE