[Creando APIs con Node JS, Express y MongoDB](https://legacy.gitbook.com/book/gmoralesc/creando-apis-con-node-js-express-y-mongodb)

Portada

[Introducción](https://gmoralesc.gitbooks.io/creando-apis-con-node-js-express-y-mongodb/content/)

Configurando el proyecto con express

[Crear un simple Web Server](https://gmoralesc.gitbooks.io/creando-apis-con-node-js-express-y-mongodb/content/configurando-el-proyecto-con-express/crear-un-simple-web-server.html)

[Utilizando Express](https://gmoralesc.gitbooks.io/creando-apis-con-node-js-express-y-mongodb/content/configurando-el-proyecto-con-express/utilizando-express.html)

[Herramientas de desarrollo](https://gmoralesc.gitbooks.io/creando-apis-con-node-js-express-y-mongodb/content/configurando-el-proyecto-con-express/configurando-el-ambiente-de-desarrollo.html)

[Configuración y variables de entorno](https://gmoralesc.gitbooks.io/creando-apis-con-node-js-express-y-mongodb/content/configurando-el-proyecto-con-express/configuracion-y-variables-de-entorno.html)

[Middleware, manejo de errores y logs](https://gmoralesc.gitbooks.io/creando-apis-con-node-js-express-y-mongodb/content/configurando-el-proyecto-con-express/middleware-manejo-de-errores-y-logs.html)

Router y Routes en Express

[Utilizando el Router y Routes de Express](https://gmoralesc.gitbooks.io/creando-apis-con-node-js-express-y-mongodb/content/router-y-routes-en-express/utilizando-el-router-y-routes-de-express.html)

[Creando el layout del API](https://gmoralesc.gitbooks.io/creando-apis-con-node-js-express-y-mongodb/content/router-y-routes-en-express/creando-el-layout-del-api.html)

[Capturando y procesando parámetros de las peticiones](https://gmoralesc.gitbooks.io/creando-apis-con-node-js-express-y-mongodb/content/router-y-routes-en-express/capturando-y-procesando-parametros-de-las-peticiones.html)

Persistencia de datos con MongoDB

[Instalando y configurando MongoDB](https://gmoralesc.gitbooks.io/creando-apis-con-node-js-express-y-mongodb/content/persistencia-de-datos-con-mongodb/instalando-mongo-db.html)

[Conectando con MongoDB](https://gmoralesc.gitbooks.io/creando-apis-con-node-js-express-y-mongodb/content/persistencia-de-datos-con-mongodb/conectando-con-mongodb.html)

[Mongoose Models](https://gmoralesc.gitbooks.io/creando-apis-con-node-js-express-y-mongodb/content/persistencia-de-datos-con-mongodb/mongoose-models.html)

[Procesando parámetros comunes con middleware](https://gmoralesc.gitbooks.io/creando-apis-con-node-js-express-y-mongodb/content/persistencia-de-datos-con-mongodb/procesando-parametros-comunes-con-middleware.html)

[Estandarización de la respuesta](https://gmoralesc.gitbooks.io/creando-apis-con-node-js-express-y-mongodb/content/persistencia-de-datos-con-mongodb/estandarizacion-de-la-respuesta.html)

[Paginación](https://gmoralesc.gitbooks.io/creando-apis-con-node-js-express-y-mongodb/content/persistencia-de-datos-con-mongodb/paginacion.html)

[Mongoose Schemas](https://gmoralesc.gitbooks.io/creando-apis-con-node-js-express-y-mongodb/content/persistencia-de-datos-con-mongodb/mongoose-schemas.html)

[Ordenamiento](https://gmoralesc.gitbooks.io/creando-apis-con-node-js-express-y-mongodb/content/persistencia-de-datos-con-mongodb/ordenamiento.html)

[Creando recursos](https://gmoralesc.gitbooks.io/creando-apis-con-node-js-express-y-mongodb/content/persistencia-de-datos-con-mongodb/creando-recursos.html)

[Relaciones entre recursos](https://gmoralesc.gitbooks.io/creando-apis-con-node-js-express-y-mongodb/content/persistencia-de-datos-con-mongodb/relaciones-entre-colecciones.html)

[Consultar recursos anidados](https://gmoralesc.gitbooks.io/creando-apis-con-node-js-express-y-mongodb/content/persistencia-de-datos-con-mongodb/consultar-recursos-anidados.html)

[Añadir recursos anidados](https://gmoralesc.gitbooks.io/creando-apis-con-node-js-express-y-mongodb/content/persistencia-de-datos-con-mongodb/anadir-recursos-anidados.html)

Asegurando el API

[Añadir y remover campos de un documento](https://gmoralesc.gitbooks.io/creando-apis-con-node-js-express-y-mongodb/content/asegurando-el-api/anadir-y-remover-campos-de-un-documento.html)

[Encriptando información sensible](https://gmoralesc.gitbooks.io/creando-apis-con-node-js-express-y-mongodb/content/asegurando-el-api/encriptando-informacion-sensible.html)

[Publicado con GitBook](https://www.gitbook.com/)

[Configuración y variables de entorno](https://gmoralesc.gitbooks.io/creando-apis-con-node-js-express-y-mongodb/content/)

**Herramientas de desarrollo**

Antes de continuar vamos a mezclar el *feature* anterior y crear una rama para trabajar:

git checkout master
git merge feature/03-add-development-tools
git checkout -b feature/04-environment-and-configuration

**Cross-env**

Nuestra aplicación no siempre se ejecutará en ambiente desarrollo, por lo cual debemos establecer cual es el ambiente en que se esta ejecutando nuestra aplicación y dependiendo de ello podemos utilizar diferentes tipos de configuración e incluso conectarnos a diferentes fuentes de datos, para ello vamos a utilizar la variable de entorno process.NODE\_ENV, que puede ser accedido desde cualquier parte de nuestra aplicación por el objeto global process .

Otro aspecto muy importante es garantizar la compatibilidad entre los diferentes sistemas operativos ya que cada sistema establece las variables de manera diferente, para asegurar que funcione en diferentes sistemas operativos utilizaremos una librería de \_npm \_llamada cross-env , la instalamos como una dependencia de nuestra proyecto y asi aseguramos que siempre esté disponible en nuestra aplicación:

npm i -S cross-env

Entonces finalmente para utilizarla modificaremos nuestros *npm scripts* en el archivo package.json de la siguiente manera:

"scripts" : { "dev" : "cross-env NODE\_ENV=development nodemon" , "start" : "cross-env NODE\_ENV=production node index" , "test" : "cross-env NODE\_ENV=test echo \"Error: no test specified\"&& exit 1" },

Antes de finalizar hacemos commit de nuestro trabajo:

git add .
git commit -m "Add cross-env and setting the environments"

Más información acerca de:

[process](https://nodejs.org/api/process.html)

[cross-env](https://www.npmjs.com/package/cross-env)

**Dotenv**

En este momento nuestras variables de configuración están almacenadas en un archivo JavaScript y sus valores están marcados en el código fuente de la aplicación, lo cual no es una buena práctica pues se estaría exponiendo datos sensibles, la idea es extraer esta información y tenerla en archivos separados por entorno pues no es la misma base de datos de desarrollo que la de producción.

Ahora debemos distinguir que es configuración de sistema y del usuario, la primera son valores que necesita la aplicación para funcionar como: numero del puerto, la dirección IP y demás, las segundas son mas relacionadas con la personalización de la aplicación como: numero de objetos por pagina, personalización de la respuesta al usuario, etc.

Utilizaremos una librería que nos permite hacer esto muy fácil llamada *dotenv* , que nos carga todas las variables especificadas en un archivo a el entorno de Node JS, procedemos a crear el archivo de configuración en la raíz del proyecto llamado .env y le colocamos el siguiente contenido:

SERVER\_IP=127.0.0.1
SERVER\_PORT=3000

Procedemos a modificar nuestro archivo de configuración en /server/config/index.js :

require ( 'dotenv' ).config( '' ); const config = {
server: {
hostname: process.env.SERVER\_HOSTNAME,
port: process.env.SERVER\_PORT,
},
}; module .exports = config;

Como podemos ver por defecto la librería *dotenv* buscará el archivo llamado .env , cargará las variables y su valores en el entorno del proceso donde se está ejecutando Node JS. Así nuestra configuración del sistema no estará mas en el código fuente de la aplicación y adicional podemos reemplazar el archivo de configuración en cada entorno donde se publique la aplicación.

Antes de terminar un aspecto muy importante es NO incluir el archivo de configuración .env en el repositorio de git, pues esto es justo lo que queremos evitar: que nuestra configuración y valores no quede en el código fuente de la aplicación, para ello modificamos el archivo .gitignore de la siguiente forma:

node\_modules/
.DS\_Store
.Thumbs.db
.env

Antes de finalizar hacemos commit de nuestro trabajo:

git add .
git commit -m "Add dotenv and set config file"

Más información acerca de:

[dotenv](https://github.com/motdotla/dotenv)