

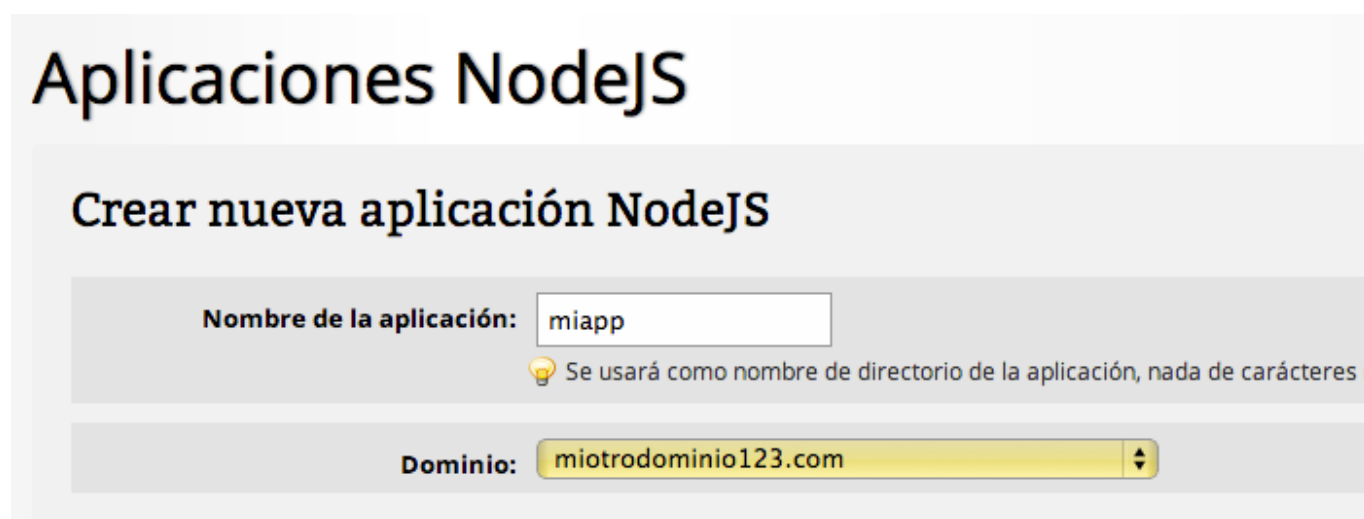
Cómo instalar Ghost 2.0 con nodejs en tu plan de alojamiento

Esta guía está pensada para instalar Ghost 2.0 o superior en un plan de alojamiento de [guebs.com](https://www.guebs.com). Aunque los principios generales son válidos para cualquier entorno, debes tener en consideración que los pasos específicos se adaptan a un plan de alojamiento creado en [guebs.com](https://www.guebs.com).

Si tu plan es anterior en su fecha de creación a noviembre de 2018, entonces es posible que todas las funcionalidades de Ghost no estén disponibles en tu plan. Es necesario que te pongas en contacto con nosotros para que migremos tu plan de plataforma, para que estas funcionalidades estén disponibles.

1 - Crear la aplicación desde el Panel de Hosting

[Accede al panel de Hosting](#) y luego pincha sobre el icono Aplicaciones NodeJs. Una vez allí rellena y envía el formulario, tal y como se ve en la siguiente imagen. Es recomendable elegir la última versión de NodeJs disponible:



Aplicaciones NodeJS

Crear nueva aplicación NodeJS

Nombre de la aplicación: Se usará como nombre de directorio de la aplicación, nada de caracteres r...

Dominio:

Ahora mismo Nodejs 10.x.x es la versión más moderna de la que podemos disponer.

Debes tener en cuenta que tendrás que hacer uso de ssh para poner en marcha esta aplicación:

[Como acceder vía SSH a tu cuenta](#)

2 - Ubicar los archivos de Ghost e instalar las dependencias.

Al completar el paso anterior sabremos cual es el directorio de nuestra aplicación NodeJS. Supongamos que es */home/tuusuario/nodejs/miapp*.

Ahora tenemos que acceder vía SSH al hosting e instalar ghost . Usaremos ghost-cli para hacer una instalación que luego usaremos. En primer lugar lo instalamos

```
</p>  
<p>npm install ghost-cli</p>  
<p>
```

Después hacemos la instalación

```
<br />  
cd nodejs/ ; mkdir test; cd test/</p>  
<p>~/node_modules/ghost-cli/bin/ghost install local<br />  
? Checking system Node.js version<br />  
? Checking current folder permissions<br />  
? Checking memory availability<br />  
? Checking for latest Ghost version<br />  
? Setting up install directory<br />  
? Downloading and installing Ghost v2.10.1<br />  
? Finishing install process<br />  
? Configuring Ghost<br />  
? Setting up instance<br />  
i Ensuring user is not logged in as ghost user [skipped]<br />  
i Checking if logged in user is directory owner [skipped]<br
```


ERROR DETAILS:

3.- Configurar la base de datos : mysql

Ghost viene configurado de serie para ser configurado con mysql. Para ello, sólo debemos crear el archivo de configuración y posteriormente actualizarlo con los datos de nuestra base de datos.

Para ello hacemos lo siguiente desde la raíz de la aplicación

```
cp -a core/server/config/env/config.production.json ./
```

Editamos el archivo con las credenciales de la base de datos mysql, de modo que quedará algo parecido a lo siguiente:

```
{
  "database": {
    "client": "mysql",
    "connection": {
      "host" : "localhost",
      "user" : "usuario",
      "password" : "password",
      "database" : "basedatos"
    }
  },
  "paths": {
    "contentPath": "content/"
  },
  "logging": {
    "level": "info",
    "rotation": {
      "enabled": true
    }
  },
  "transports": ["file", "stdout"]
}
```

4.- Editar la dirección web de tu sitio

El ghost instalado tiene por defecto localhost:xxxx como dirección por defecto de tu sitio web. Hay que cambiarlo para que refleje el dominio del sitio. Para ello tenemos que editar el archivo core/server/config/defaults.json. En la primera línea encontraremos el parámetro url, que debemos dejar como sigue:

```
"url": "http://tudominio.com",
```

5.- Creación del usuario inicial

Hecho esto, ya podemos acceder al administrador de ghost para crear el usuario inicial a través de <https://www.tudominio.com/ghost/>



4- Usar Sqlite con Ghost

También es posible usar Sqlite en lugar de Mysql. Si tu plan de alojamiento está aquí en guebs.com, te lo recomendamos. Para ello, sólo debes crea el archivo config.production.js como los has creado en el paso 5. Entonces debes dejarlos configurados con la cadena correspondiente a la dirección donde se archivará la base de datos sqlite:

config.production.js

```
{

"database": {
  "client": "sqlite3",
  "connection": {
    "filename": "content/data/miblog.db"
  }
},

"paths": {
  "contentPath": "content/"
},
"logging": {
  "level": "info",
  "rotation": {
    "enabled": true
  },
  "transports": ["file", "stdout"]
}
}
```

Hecho esto, tu sitio estará funcionando con Sqlite como motor de base de datos.