



Repositorios npm en Nexus

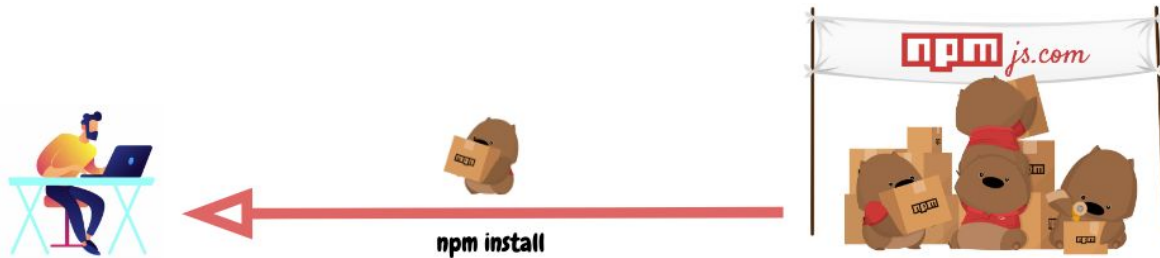
Victor Bayas

¿Qué es npm?

npm (**Node Package Manager**) es el administrador de paquetes predeterminado de NodeJS, un entorno de ejecución de JavaScript del lado del servidor.

Por lo general, los paquetes los vamos a obtener de internet desde el npm Registry (npmjs.com) que es propiedad de Microsoft, sin embargo, es posible utilizar repositorios propios o de terceros.

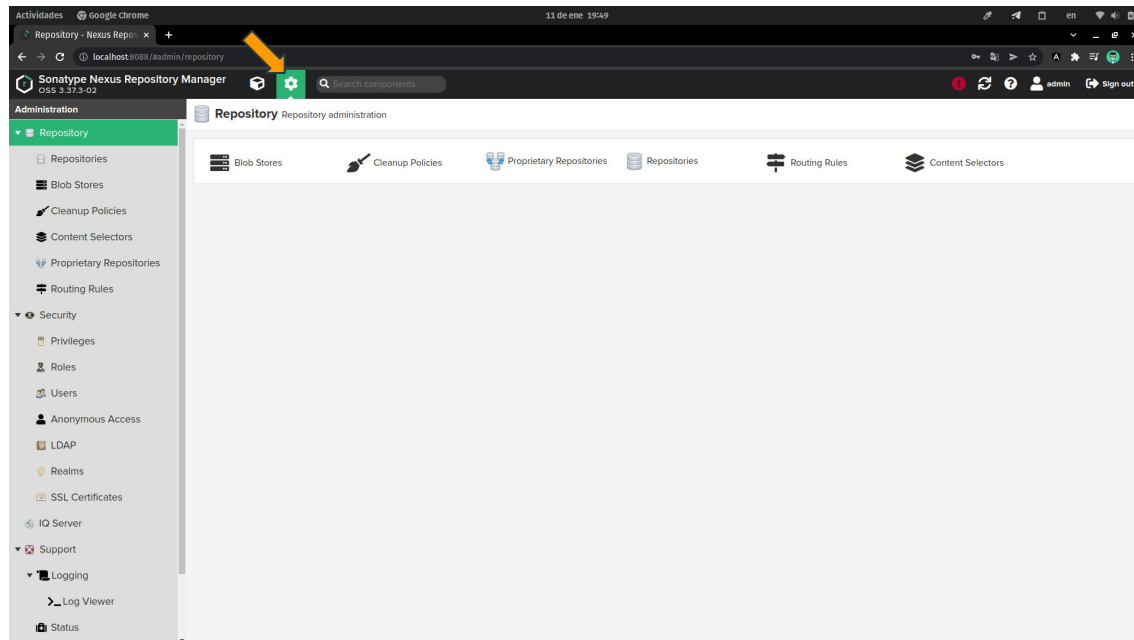
Por defecto los paquetes instalados localmente se guardan en una carpeta llamada **node_modules** dentro de la raíz de nuestro proyecto de NodeJS, también se puede instalar globalmente un paquete con el switch -g.



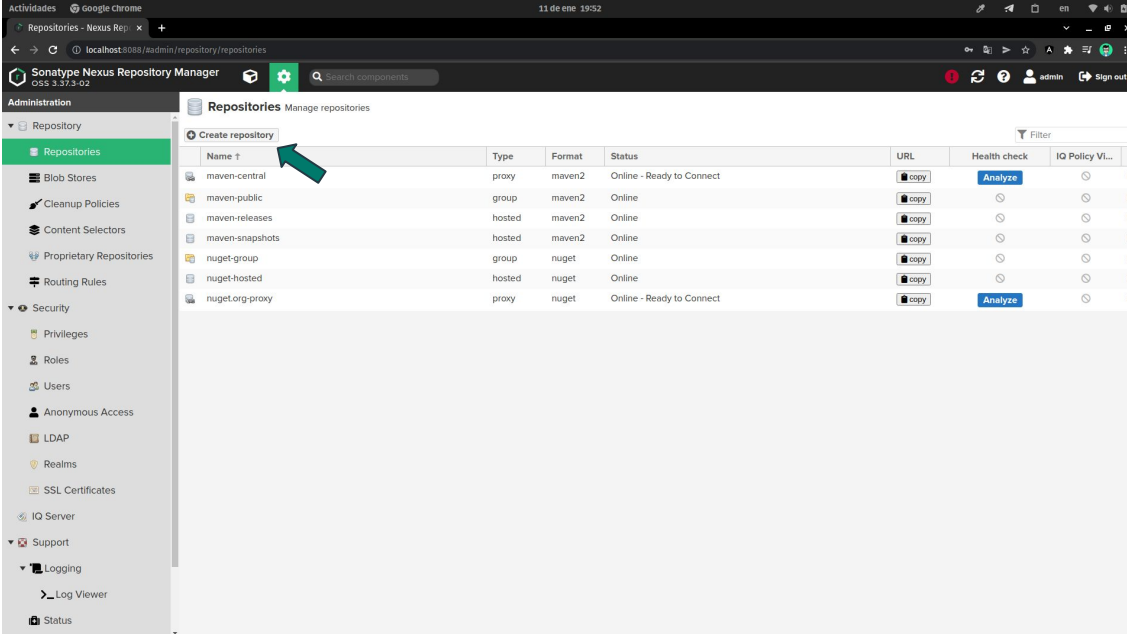


Creación de un repositorio npm

Nos dirigimos al dashboard de Nexus






Abrimos la sección de repositories



The screenshot shows the Sonatype Nexus Repository Manager web interface. The left sidebar contains the 'Administration' menu, with 'Repositories' highlighted by an orange arrow. The main content area displays the 'Repositories' page, which includes a 'Create repository' button and a table of existing repositories. A green arrow points to the 'Name' header of the table.

Name ↑	Type	Format	Status	URL	Health check	IO Policy VL...
maven-central	proxy	maven2	Online - Ready to Connect	copy	Analyze	
maven-public	group	maven2	Online	copy		
maven-releases	hosted	maven2	Online	copy		
maven-snapshots	hosted	maven2	Online	copy		
nuget-group	group	nuget	Online	copy		
nuget-hosted	hosted	nuget	Online	copy		
nuget.org-proxy	proxy	nuget	Online - Ready to Connect	copy	Analyze	

Creamos un nuevo repositorio de tipo npm (hosted)

-  npm (group)
-  npm (hosted)
-  npm (proxy)

Create repository



 **Repositories** /  **Select Recipe** /  **Create Repository: npm (hosted)**

Name: A unique identifier for this repository

npm-clases

Online: ☒ If checked, the repository accepts incoming requests

Storage

Blob store:

Blob store used to store repository contents

default

Strict Content Type Validation:

☒ Validate that all content uploaded to this repository is of a MIME type appropriate for the repository format

Hosted

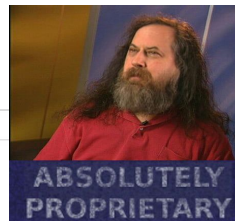
Deployment policy:



Controls if deployments of and updates to artifacts are allowed

Disable redeploy

Proprietary Components:

☐ Components in this repository count as proprietary for namespace conflict attacks (requires Sonatype Nexus Firewall)

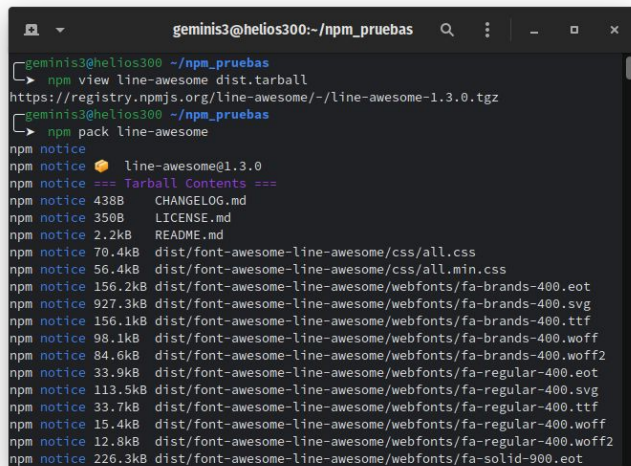




Subida de paquetes al
repositorio

Descarga de paquetes npm

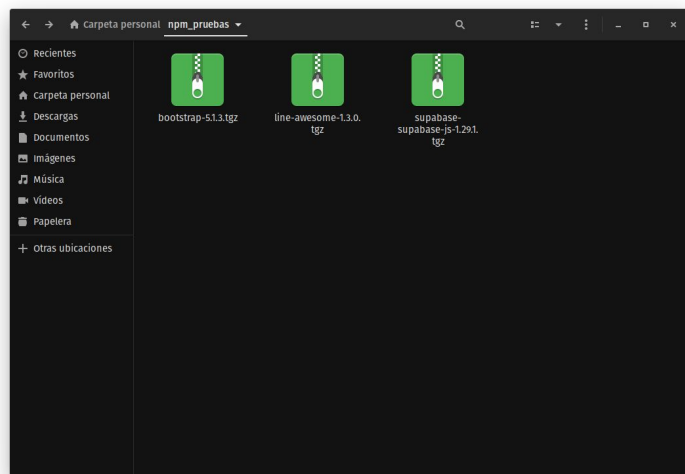
Utilizaremos el comando **npm view <nombre paquete> dist.tarball** seguido de **npm pack <nombre paquete>** para descargar un paquete en formato .tar.gz (tarball).



```
geminis3@helios300:~/npm_pruebas
└─> npm view line-awesome dist.tarball
https://registry.npmjs.org/line-awesome/-/line-awesome-1.3.0.tgz
└─> npm pack line-awesome
npm notice
npm notice 📦 line-awesome@1.3.0
npm notice === Tarball Contents ===
npm notice 438B  CHANGELOG.md
npm notice 350B  LICENSE.md
npm notice 2.2kB  README.md
npm notice 70.4kB dist/font-awesome-line-awesome/css/all.css
npm notice 56.4kB dist/font-awesome-line-awesome/css/all.min.css
npm notice 156.2kB dist/font-awesome-line-awesome/webfonts/fa-brands-400.eot
npm notice 927.3kB dist/font-awesome-line-awesome/webfonts/fa-brands-400.svg
npm notice 156.1kB dist/font-awesome-line-awesome/webfonts/fa-brands-400.ttf
npm notice 98.1kB dist/font-awesome-line-awesome/webfonts/fa-brands-400.woff
npm notice 84.6kB dist/font-awesome-line-awesome/webfonts/fa-brands-400.woff2
npm notice 33.9kB dist/font-awesome-line-awesome/webfonts/fa-regular-400.eot
npm notice 113.5kB dist/font-awesome-line-awesome/webfonts/fa-regular-400.svg
npm notice 33.7kB dist/font-awesome-line-awesome/webfonts/fa-regular-400.ttf
npm notice 15.4kB dist/font-awesome-line-awesome/webfonts/fa-regular-400.woff
npm notice 12.8kB dist/font-awesome-line-awesome/webfonts/fa-regular-400.woff2
npm notice 226.3kB dist/font-awesome-line-awesome/webfonts/fa-solid-900.eot
```

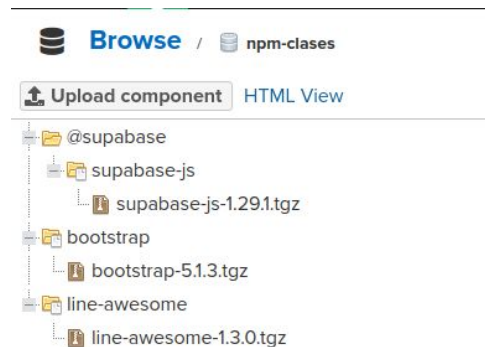
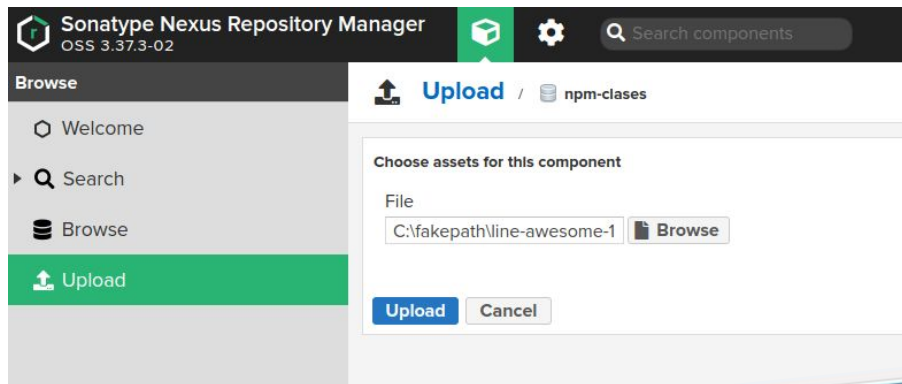

Descarga de paquetes npm

Nuestros tarballs se encontrarán en la carpeta desde la cual hemos ejecutado los comandos.





Subida de paquetes

Nos dirigimos a nuestro dashboard en la sección de **Upload**, elegimos nuestro repositorio **npm-clases** y subiremos los tarballs que hemos descargado previamente.



✓ Component uploaded to the repository, [view it now.](#)

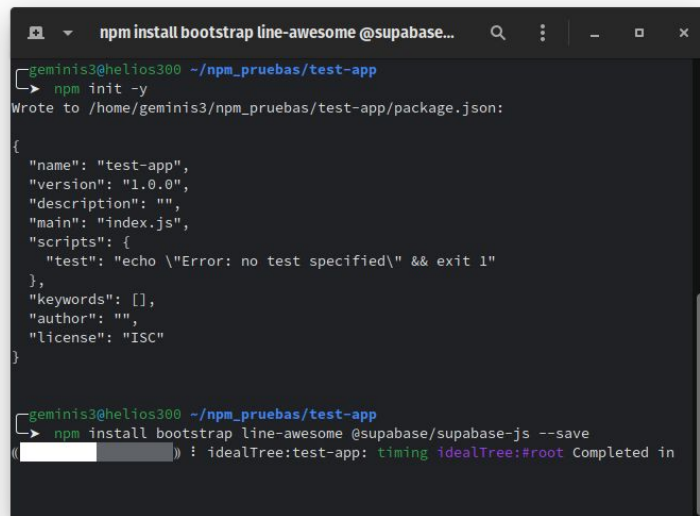


Configuración del lado del cliente

Uso del repositorio

Ahora que tenemos unos cuantos paquetes subidos en nuestro propio repositorio npm, es hora de hacer pruebas.

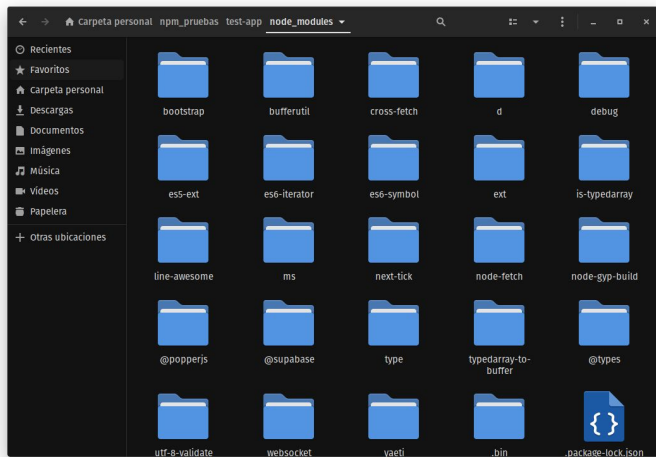
Crearemos un nuevo proyecto de NodeJS con **npm init -y** e instalaremos los mismos paquetes que tenemos en nuestro repositorio.



```
npm install bootstrap line-awesome @supabase...  
geminis3@helios300 ~/npm_pruebas/test-app  
└─> npm init -y  
Wrote to /home/geminis3/npm_pruebas/test-app/package.json:  
  
{  
  "name": "test-app",  
  "version": "1.0.0",  
  "description": "",  
  "main": "index.js",  
  "scripts": {  
    "test": "echo \"Error: no test specified\" && exit 1"  
  },  
  "keywords": [],  
  "author": "",  
  "license": "ISC"  
}  
  
geminis3@helios300 ~/npm_pruebas/test-app  
└─> npm install bootstrap line-awesome @supabase/supabase-js --save  
└─> ! idealTree:test-app: timing idealTree:#root Completed in
```

Uso del repositorio

Dentro de nuestro proyecto tenemos una carpeta llamada **node_modules** que contiene los paquetes instalados. Para demostrar el funcionamiento de nuestro repositorio **eliminaremos dicha carpeta** (esto tomará su tiempo en un proyecto real).



Apuntamos hacia nuestro repositorio

Para que npm sepa que deberá descargar los paquetes desde el repositorio que previamente levantamos en Nexus, deberemos crear un archivo llamado **.npmrc** con el siguiente contenido:

```
.npmrc x
.npmrc
1  # Aquí ponemos la URL de nuestro repositorio
2
3  # En el caso de tener deshabilitado el acceso anónimo,
4  # deberemos pasarle el usuario/contraseña como base64
5  # _auth=<hash base64>
6
7  registry=http://localhost:8088/repository/npm-clases/
```



Probamos el repositorio

Volvemos a descargar nuestras dependencias con **npm install** .

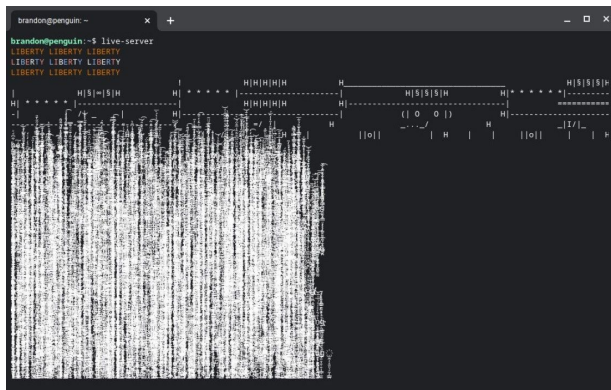
La url del paquete
@supabase/supabase-js
corresponde a nuestro repo

```
geminis3@helios300:~/npm_pruebas/test-app
└─$ npm i
added 29 packages in 2s

2 packages are looking for funding
  run 'npm fund' for details
└─$ npm view @supabase/supabase-js dist.tarball
http://localhost:8088/repository/npm-clases/@supabase/supabase-js/-/supabase-js-1.29.1.tgz
└─$
```

Conclusiones

Contar con nuestro propio repositorio npm nos permitirá almacenar versiones específicas de paquetes utilizados a lo largo de nuestros proyectos, garantizando su disponibilidad en caso de que el paquete desaparezca del npm Registry o como sucedió recientemente con la librería Colors.js donde su creador decidió publicar una actualización corrompida como forma de protesta social.



A terminal window titled 'brandon@penguin ~' showing the command 'live-server' being executed. The output displays a corrupted version of the Colors.js package, with the word 'LIBERTY' repeated multiple times and a large block of garbled, non-readable text below it, indicating a corrupted update.



Gracias :)