Estrategia de seguridad variables de entorno

La estrategia de variables de entorno es una forma de proteger la información secreta que se utiliza en el código de una aplicación, como contraseñas o claves de acceso. En lugar de guardar esta información directamente en el código, se guarda en un lugar separado llamado "variables de entorno".

Estas variables de entorno son como contenedores de información que se pueden acceder desde el código de la aplicación, pero no se almacenan directamente en él. Esto hace que la información sea más segura porque si alguien accede al código de la aplicación, no podrá ver la información confidencial.

Además, la estrategia es flexible porque se puede cambiar la información secreta sin tener que modificar el código de la aplicación. Por ejemplo, si se cambia una contraseña, solo se tiene que cambiar en la variable de entorno y no en el código de la aplicación.

En este proyecto para el desarrollo de la aplicación almacenamos estas variables de entorno en un archivo .env.

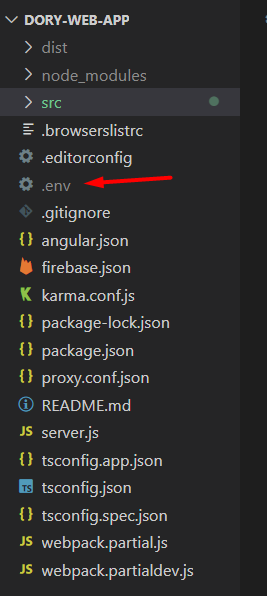


Figura 1 archivo .env para ambiente de desarrollo

Esto le permitirá a la aplicación acceder a los valores de estas variables (en el archivo .env) desde el código fuente cuando estamos probándola en un servidor de desarrollo en nuestro computador.

Cuando la aplicación se ejecuta en un servidor que está en línea, como un servidor de pruebas o de producción, se necesita configurar estas variables en el servidor.

Dependiendo del servidor que estemos utilizando, la configuración de estas variables puede ser diferente. Si el servidor es de nuestra propiedad, podemos configurar las variables en una ubicación específica. Pero si el servidor es un servicio de terceros, como en nuestro caso Heroku, las variables pueden configurarse de una manera diferente.

En este documento, se explicará por qué es importante utilizar esta estrategia y cómo se implementa.

¿Por qué utilizar la estrategia de seguridad de variables de entorno en un archivo .env?

1. Protección de información sensible

Uno de los principales motivos para utilizar la estrategia de seguridad de variables de entorno es proteger la información sensible de los usuarios y de la aplicación. En muchos casos, las aplicaciones web o móviles manejan información sensible como contraseñas, claves de API, tokens de acceso, credenciales de bases de datos, entre otros datos que deben mantenerse seguros. Si esta información se incluye directamente en el código fuente de la aplicación, cualquier persona que tenga acceso a él podría obtenerla y utilizarla de manera malintencionada.

2. Flexibilidad en ambientes de desarrollo

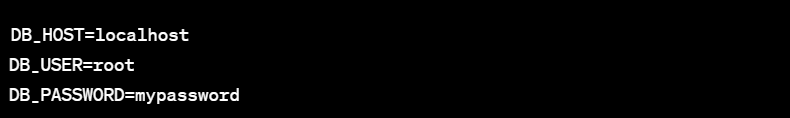
Al utilizar la estrategia de seguridad de variables de entorno en un archivo .env, se pueden crear diferentes archivos para diferentes ambientes de desarrollo, lo que brinda flexibilidad al desarrollador y al equipo. Por ejemplo, se puede tener un archivo .env para el ambiente de desarrollo, otro para el ambiente de pruebas y otro para el ambiente de producción. De esta manera, se pueden establecer diferentes valores para las variables de entorno según el ambiente de la aplicación.

3. Seguridad y privacidad en colaboración

La colaboración en equipos de desarrollo es fundamental para el éxito de un proyecto, sin embargo, en algunos casos no todos los miembros del equipo necesitan conocer ciertas claves o contraseñas. Al utilizar la estrategia de seguridad de variables de entorno, se pueden establecer permisos específicos para acceder a los archivos .env, lo que permite mantener la privacidad y la seguridad de la información.

Implementación de la estrategia de seguridad de variables de entorno en un archivo .env para el ambiente de desarrollo

La implementación de esta estrategia es relativamente sencilla. Primero, se debe crear un archivo. env en la raíz del proyecto como se muestra en la figura 1. En el se incluirán las variables de entorno con sus respectivos valores. Por ejemplo:



Luego, en el código fuente de la aplicación, se deben utilizar las variables de entorno en lugar de los valores directos. Por ejemplo, en lugar de escribir localhost para la dirección de la base de datos, se utilizará process.env.DB\_HOST.

Es importante tener en cuenta que el archivo .env debe ser incluido en el archivo .gitignore para evitar que se incluya en el repositorio y se exponga la información sensible que contiene.

En la receta “Receta 4 - Configurar variables de entorno en el ambiente de desarrollo” encontrara mas detalles acerca de este tema.

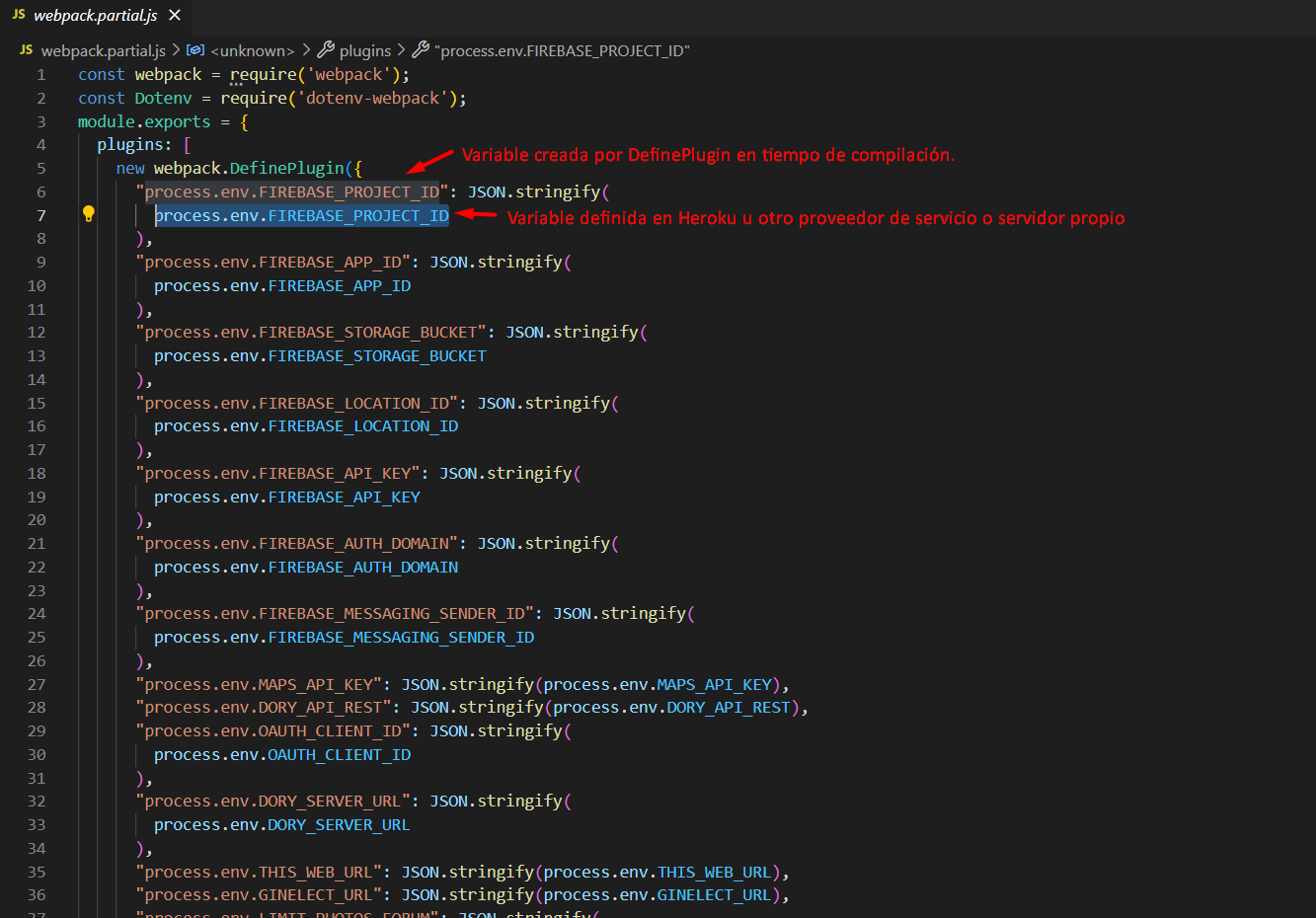
Implementación de la estrategia de seguridad de variables de entorno en un servidor en línea de pruebas o de producción

En un servidor de pruebas o producción, se debe configurar las variables de entorno en el sistema operativo, en el servidor web, o donde lo indique el proveedor de servicio que se esté utilizando. De esta manera, la aplicación podrá acceder a las variables de entorno y utilizarlas en el código fuente.

En nuestro caso hemos desarrollado una aplicación Angular desplegada en Heroku, y no es posible acceder a las variables de entorno directamente desde el sistema operativo ya que nuestro código se ejecuta en el navegador. En estos casos, es posible utilizar una herramienta como DefinePlugin de Webpack para definir estas variables durante la compilación del código en Heroku. De modo que podamos tenerlas disponibles en el navegador durante la ejecución de la aplicación.

Podemos utilizar esta herramienta para definir variables de entorno que se hayan definido previamente en Heroku o cualquier otro proveedor de servicios en la nube. De esta manera, podemos ocultar la información sensible en variables de entorno definidas en el proveedor de servicios en lugar de dejarla en el código fuente de la aplicación.

En nuestro caso le hemos pedido a DefinePlugin que tome todas las variables definidas en Heroku y nos defina esas variables en tiempo de compilación:



Luego en nuestro código podremos acceder a ellas a través de procces.env.NOMBRE\_VARIABLE

En la Receta 6 “Como configurar las variables de entorno en Heroku” se detalla como configurar las variables de entorno en un servicio como Heroku.