

# Introducción

En el desarrollo de aplicaciones web con Node.js y Express, la gestión de configuraciones sensibles y personalizables es fundamental. Las **variables de entorno** permiten almacenar información como credenciales, claves API, configuraciones de base de datos y otros parámetros que pueden cambiar según el entorno de ejecución (desarrollo, pruebas o producción).

Al utilizar variables de entorno, se mejora la seguridad y la portabilidad de la aplicación, evitando la exposición de datos sensibles en el código fuente y facilitando la configuración en distintos entornos sin necesidad de modificar el código.

## ¿Qué son las Variables de Entorno?

Las **variables de entorno** son valores externos que pueden influir en el comportamiento de un proceso en ejecución. Se almacenan en el sistema operativo y pueden ser accedidas desde la aplicación para definir configuraciones sin necesidad de hardcodear valores.

En el contexto de **Express.js**, las variables de entorno se usan comúnmente para:

- Configurar el **puerto** del servidor.
- Definir las **credenciales** de la base de datos.
- Almacenar claves API para servicios externos.
- Habilitar o deshabilitar funciones según el entorno de ejecución (desarrollo, producción, pruebas).

## Buenas Prácticas al Usar Variables de Entorno

- **No compartir archivos `.env`:** Exclúyelo del control de versiones usando un `.gitignore`.
- **Definir valores predeterminados:** Usa valores por defecto en caso de que una variable no esté definida.

## Usar un archivo `.env` por cada entorno

En lugar de usar un solo archivo `.env`, crea distintos archivos para cada entorno:

- `.env.development` (para desarrollo)
- `.env.test` (para pruebas)
- `.env.production` (para producción)

# ¿Qué son los entornos en el desarrollo de software?

En el desarrollo de software, un **entorno** es el conjunto de configuraciones, herramientas y recursos en los que una aplicación se ejecuta y opera. Dependiendo del **ciclo de vida del software**, una aplicación puede ejecutarse en diferentes entornos, cada uno con configuraciones específicas para facilitar el desarrollo, las pruebas y la producción.

## Principales tipos de entornos

Los entornos más comunes en una aplicación son:

### Desarrollo (Development)

- Utilizado por los programadores para escribir y probar código.
- Se ejecuta en una máquina local o en un servidor de desarrollo.
- Suele tener **recursos abiertos** (logs detallados, hot-reload, sin restricciones de seguridad estrictas).
- Base de datos local o de prueba.

### Pruebas (Testing o Staging)

- Se usa para pruebas antes de pasar a producción.
- Puede incluir **pruebas unitarias, de integración y de aceptación**.
- Se ejecuta en un entorno similar a producción, pero aislado.
- **No afecta a usuarios reales.**

### Producción (Production)

- Es el entorno real donde la aplicación es utilizada por los usuarios.
- Optimizado para rendimiento y seguridad (sin logs innecesarios, con monitoreo, caché y backups).
- Usa una base de datos y servidores reales.
- Se ejecuta en servicios en la nube (AWS, Google Cloud, Vercel, etc.).

## ¿Por qué son importantes los entornos?

- **Separación de configuraciones:** Cada entorno tiene su propia base de datos y recursos.
- **Evitar errores en producción:** Se prueban los cambios en entornos controlados antes de publicarlos.
- **Optimización y seguridad:** Producción requiere configuraciones específicas (caché, logs mínimos, encriptación).