

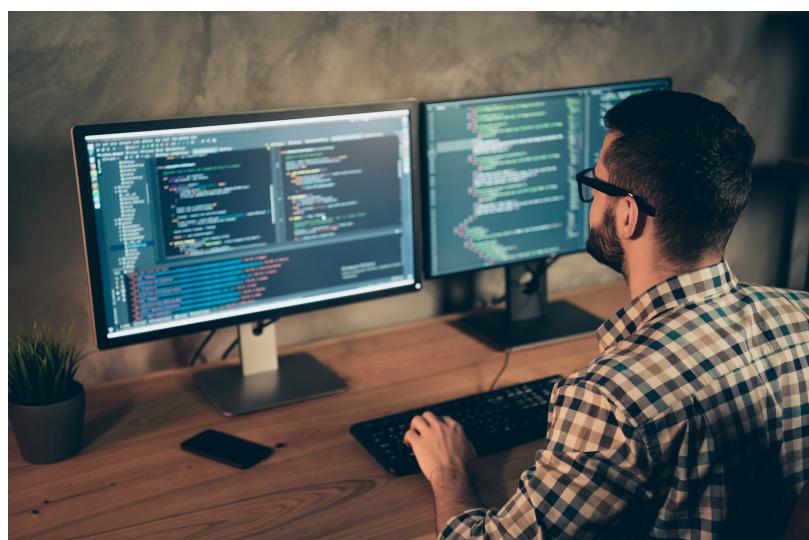


**Back end nível intermedio**

**Node.js**

Dentro del desarrollo de software es sumamente importante contar con dos componentes. El *front end* y el *back end*. Ambos logran la perfecta armonía de un sistema web, sin embargo, dentro del back end existen diferentes elementos que pueden hacer que el rendimiento y el funcionamiento del sistema sea mejor. Tal es el caso de Node.js, con el cual, serás capaz de diseñar aplicaciones web que sean escalables, es decir, que tu sistema tenga la capacidad de ir creciendo de acuerdo con el número de usuarios que lo utilicen.

En este tema, revisarás la importancia de una página web escalable mediante **Node.js**; además, aprenderás a instalarla y configurarla para realizar conexiones con bases de datos y, así, obtener la información que el front end necesita mostrar al usuario.



## Introducción y arquitectura Node.js

De acuerdo con Node.js (s.f.), surge en 2009 gracias a la necesidad que tenían los desarrolladores web de incrementar la capacidad de la página web a medida que más usuarios interactuaban con ella, es decir, imagina que te encuentras desarrollando el back end de una página web para vender libros. Tu página debe ser capaz de atender tanto a 1 usuario como a 2,000 a la vez, sin que la página sufra alguna ralentización o que algún usuario note que no funciona correctamente. A lo anterior se le conoce como: un sistema escalable, ya que cuenta con la capacidad de adaptarse de acuerdo con la demanda que tenga.

Ahora bien, ¿realmente es importante que el desarrollo web sea escalable?

Imagina lo siguiente: a medida que ingresan los usuarios a tu desarrollo web, los recursos de este aumentan su demanda. Si tu desarrollo no es capaz de adaptarse, crecer o disminuir el uso de recursos a medida que los usuarios entran y salen, el desarrollo colapsaría y eso traería diferentes problemas, como los siguientes:

- Disminución de ventas de producto; se debe a que, si el desarrollo no funciona correctamente, tendrás un número menor de usuarios, lo cual se traduce en menos ventas.
- Pérdida de fidelidad de clientes; si el desarrollo no opera correctamente, los clientes dejarán de confiar en el desarrollo y optarán por buscar una alternativa que realmente funcione.
- Eventualmente, el desarrollo no será sustentable y los gastos para mantenerlo sobrepasarán las ganancias, por lo cual, la inversión se perderá y será un proyecto que ha fracasado. Incluso puede llegar a ser la pérdida de la compañía, ya que el costo - beneficio no es el óptimo.

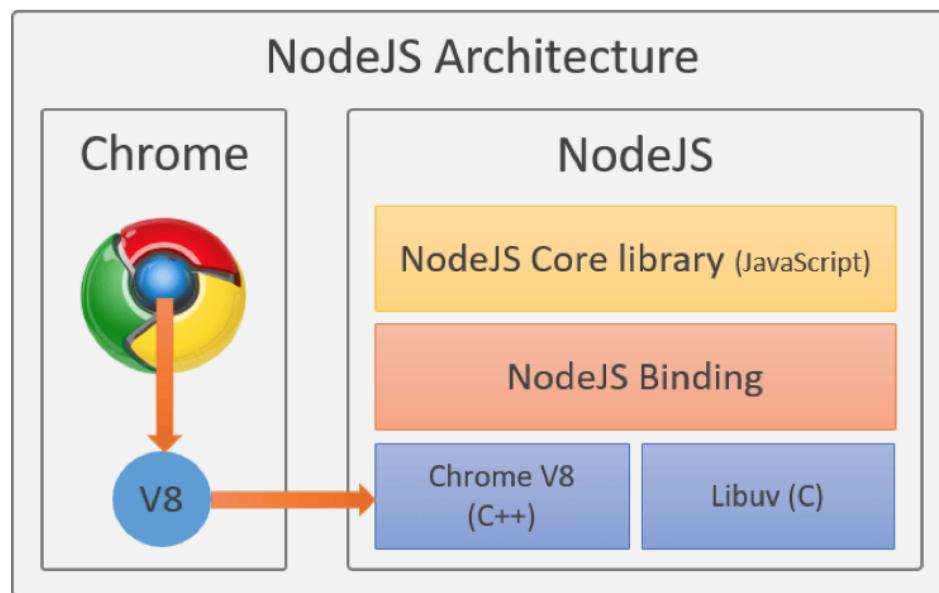
Algunas ventajas que tendrás al utilizar Node.js son las siguientes:

1. Optimiza el rendimiento y la escalabilidad de los desarrollos web.
2. El código usado es JavaScript, lo cual provoca ahorrar mucho tiempo en establecer la conexión entre el front end y el back end.
3. NPM (Node package manager), da acceso a muchos paquetes que puedes reutilizar en cualquier momento, y te permite automatizar las herramientas usadas para compilar tu desarrollo web.
4. Es compatible con diferentes sistemas operativos.

Una vez que has revisado la importancia de la escalabilidad de un sitio web continuarás con Node.js y su arquitectura.

Node.js es el primer intento para utilizar JavaScript del lado del servidor, está construido sobre el motor de Chrome V8 Js, por lo tanto, así como en el front end puedes programar en JavaScript, gracias a Node.js, también podrás programar el back end, lo cual permitirá escuchar las peticiones del front end y dar la información necesaria para que se muestre al usuario en el momento que se solicite.

De acuerdo con Node.js (s.f.), la arquitectura es la siguiente:



Fuente: oblancarte. (2017). *Introducción a NodeJS (JavaScript del lado del Servidor)*. Recuperado de <https://www.oscarblancarteblog.com/2017/05/29/introduccion-a-nodejs-2/>

En la arquitectura se puede ver de la siguiente manera: Node.js está dentro del navegador de Chrome, es un motor de JavaScript V8, el cual tiene diferentes capas:

- Chrome V8: motor de JavaScript.
- Libuv: librería utilizada para procesar entradas y salidas asíncronas.
- NodeJS Core Library: librería encargada de la comunicación entre la API de NodeJS y el Motor.

Retomando el ejemplo del desarrollo web para venta de libros: El usuario ingresa a la página web y busca el libro que desea. Al momento que se hace la búsqueda, el front end se conecta con Node.js quien tiene el código JavaScript para hacer la búsqueda en la base de datos y brindar la información necesaria al usuario en el momento que la solicite, sin importar cuantos usuarios se encuentren en ese momento utilizando el desarrollo.

## Instalación y configuración del entorno

Ya que conoces un poco acerca de Node.js y su arquitectura, es momento de revisar como se instala y como se configura, para esto deberás realizar los siguientes pasos:

1. Ingresá a la siguiente liga:



node. (s.f.). Descargas. Recuperado de  
<https://nodejs.org/es/download/>

**El uso y descarga del software deberá apegarse a los términos y condiciones del sitio oficial del fabricante y su uso será responsabilidad de quien lo descargue. Tecmilenio no tiene licencia ni posee los derechos sobre dicho software.**

2. Descarga el instalador de acuerdo con el sistema operativo que necesites. Se encuentra disponible para:

- Windows
- MacOS
- Linux

### Descargas

Versión actual: 16.17.0 (includes npm 8.15.0)

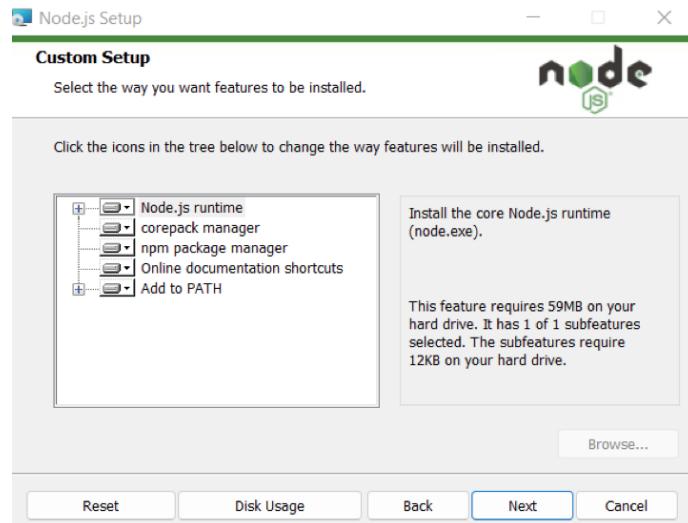
Descargue el código fuente de Node.js o un instalador pre-compilado para su plataforma, y comience a desarrollar hoy.

LTS	Actual
Recomendado para la mayoría	Últimas características
 Instalador Windows	 Instalador macOS
node-v16.17.0-x64.msi	node-v16.17.0.pkg
<a href="#">Instalador Windows (.msi)</a>	32-bit
<a href="#">Binario Windows (.zip)</a>	64-bit
<a href="#">Instalador macOS (.pkg)</a>	64-bit / ARM64
<a href="#">Binario macOS (.tar.gz)</a>	64-bit
<a href="#">Binario Linux (x64)</a>	ARM64
<a href="#">Binario Linux (ARM)</a>	64-bit
<a href="#">Código Fuente</a>	ARMv7
	ARMv8
	node-v16.17.0.tar.gz

### Plataformas adicionales

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

- Ejecuta el instalador descargado. Solo necesitas avanzar en el proceso y dejar que el instalador haga todo lo necesario.



Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

- Abre la línea de comandos para verificar que la instalación se hizo de forma correcta.
- En la línea de comandos escribe `node -v` y pulsa la tecla enter. Te aparecerá la versión instalada.

```
C:\Users\Clau>node -v  
v16.17.0
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

- En la línea de comandos escribe `npm -v` para comprobar si el Node Package Manager (NPM), se ha instalado correctamente y que versión se instaló.

```
C:\Users\Clau>npm -v  
8.15.0
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

¡Listo!, Node.js está listo para que empieces a utilizarlo.

Para comprobar que todo funciona correctamente, haz el siguiente ejercicio:

1. Crea un archivo que se llame servidor.js y que contenga el siguiente código:

```
const http = require('http');

const hostname = '127.0.0.1';
const port = 3000;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World');
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

2. Ejecuta el servidor web usando en la línea de comandos: `node servidor.js`.

3. Ingresa a la página `http://localhost:3000` y verás el mensaje que dirá “Hola mundo”.

Los siguientes enlaces son externos a la Universidad Tecmilenio, al acceder a ellos considera que debes apegarte a sus términos y condiciones.

## Schemas, mutaciones y Queries

Recuerda que Node.js se utiliza en el back end, por lo tanto, obtendrá la información de donde sea necesario. Normalmente la información es obtenida de una API o de una base de datos; para la segunda opción utilizarás los Schemas y los Queries.

Si retomamos el ejemplo de una página web de venta de libros, toda la información acerca de los libros se encuentra almacenada en diferentes bases de datos. Para que el front end obtenga la información que necesita, se pueden utilizar los Queries.

De acuerdo con MYSQLTUTORIAL (2022), mediante los Queries puedes realizar diferentes acciones como:

- Conectarte a la base de datos MySQL.
- Crear tablas.
- Insertar filas.
- Obtener la información de las bases de datos.
- Actualizar la información de las bases de datos.
- Eliminar información de las bases de datos.

Para saber más sobre cómo crear tablas, insertar filas, actualizar y eliminar información, te sugerimos revisar el siguiente enlace:



MySQLTUTORIAL. (s.f.). MySQL Node.js. Recuperado de  
<https://www.mysqltutorial.org/mysql-nodejs/>

**Los siguientes enlaces son externos a la Universidad Tecmilenio, al acceder a ellos considera que debes apegarte a sus términos y condiciones.**

A continuación, solo revisarás como conectarte y obtener la información de las bases de datos.

Observa el siguiente ejemplo:

Pedro ingresa a la página web con la intención de comprar un libro de programación. Aún no está seguro del libro que va a comprar, por lo cual decide buscar en esta página, las opciones disponibles, para así tomar la mejor decisión. Seguido de ello, Pedro escribe la palabra “programación” en la caja de texto y oprime el botón de buscar.

Se realiza la conexión a la base de datos de la siguiente forma:

1. Es necesario instalar el driver para MySQL en node.js, para esto necesitarás realizar lo siguiente:

a. Ejecuta el siguiente comando para crear el paquete de Json:

```
npm init
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

b. Instala el paquete de MySQL usando el siguiente comando:

```
npm install mysql
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

2. Crea la base de datos usando el siguiente comando:

```
CREATE DATABASE "Nombrebasededatos"
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

3. Importa el módulo mysql usando el siguiente comando:

```
let mysql = require('mysql');
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

4. Crea la conexión con la base de datos:

```
let connection = mysql.createConnection({  
    host: 'localhost',  
    user: 'root',  
    password: '',  
    database: 'Nombredelabasededatos'  
});
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Se ejecuta el siguiente Query desde Node.js para conectarse a la base de datos y obtener la información:

```
let mysql = require('mysql');
let config = require('./config.js');

let connection = mysql.createConnection(config);

let sql = `SELECT book_name* FROM libros WHERE Clasificacion=Programacion`;
connection.query(sql, (error, results, fields) => {
  if (error) {
    return console.error(error.message);
  }
  console.log(results);
});

connection.end();
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

De acuerdo con Data Pilot (2020), los Schemas se utilizan en bases de datos que no son SQL. Un ejemplo de esta base de datos es MongoDB.

Los Schemas definen la forma de los documentos que estarán dentro de una colección.

A continuación, verás los pasos para crear un Schema:

1. Instala Mongoose utilizando el siguiente comando:

```
npm install mongoose
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

2. Importa el paquete de Mongoose usando el siguiente comando:

```
const mongoose = require("mongoose");
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

3. Crea una instancia del Schema usando el siguiente comando:

```
const mongoose = require("mongoose");
const Schema = mongoose.Schema;
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

4. Define el Schema, retoma el ejemplo de los libros y usa el siguiente comando:

```
let book = new Schema({
  name: {
    type: String
  },
  price: {
    type: Number
  },
  instock: {
    type: Boolean
  }
});
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Dentro del desarrollo web es recomendable utilizar bases de datos que sean MySQL, ya que son las más utilizadas por las empresas, sin embargo; es importante conocer otras alternativas que se tienen como MongoDB.

## Express.js

De acuerdo con Expresar (s.f.), esta herramienta es un framework de Node.js que te puede ayudar a configurar diferentes características para las aplicaciones web y móviles. A su vez, permite crear el esqueleto de una aplicación web de tal forma que ya no tengas que realizarlo desde cero.

También, permite configurar el entorno de desarrollo y ayuda a realizar tareas comunes al momento de publicar la aplicación web.

Algunas de las tareas que puede realizar Express.js son las siguientes:

- Escribir las peticiones con diferentes verbos HTTP, GET, POST, DELETE.
- Integra tu desarrollo con motores de renderización.
- Configura ajustes del desarrollo web como el puerto desde el cual se realizará la conexión.
- Añade las peticiones de middleware.
- Brinda seguridad a las cabeceras.
- Administra inicios de sesión de los usuarios.

Para su instalación debes de seguir los siguientes pasos:

1. Crea un directorio donde se vaya a encontrar la aplicación y conviértelo en el directorio de trabajo mediante los siguientes comandos:

```
$ mkdir myapp  
$ cd myapp
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

2. Utiliza el comando `npm init` para crear un archivo `package.json`:

```
$ npm init
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

3. Instala Express usando el siguiente comando:

```
$ npm install express
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Una vez que ya lo has instalado puedes crear aplicaciones web que te permitirán enrutar las peticiones HTTP, configurar el Middleware, visualizar las vistas de HTML y cambiar o administrar las configuraciones de las aplicaciones que controlan la aplicación creada, entre otras.

Para el método de Get deberás utilizar el siguiente comando:

```
app.get(  
    '/',  
    (request, response) => {  
        response.send('pagina Principal')  
    }  
)
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Para el método POST deberás utilizar el siguiente comando:

```
app.post('/', function (req, res) {  
    var user_name = req.body.user;  
    var password = req.body.password;  
    console.log("User name = " + user_name + ", password is " + password);  
    res.end("yes");  
});
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Para el método LISTEN deberás utilizar el siguiente comando:

```
app.listen(  
    3000,  
    () => {  
        console.log('Servidor corriendo')  
    }  
)
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Tecmilenio no guarda relación alguna con las marcas mencionadas como ejemplo. Las marcas son propiedad de sus titulares conforme a la legislación aplicable, se utilizan con fines académicos y didácticos, por lo que no existen fines de lucro, relación publicitaria o de patrocinio.

Node.js te permitirá lograr que tu desarrollo web tenga un escalamiento a medida que tenga más o menos demanda, lo cual provocará que el desarrollo siempre esté disponible sin importar la cantidad de usuarios que lo estén utilizando ese momento, traduciendo esto en mayor venta, mayores clientes y un sentido de alta disponibilidad.

¿Consideras útil Node.js para tus desarrollos back end?

¿Crees que es necesaria la escalabilidad en tus desarrollos web?

# Referencias bibliográficas

- Data Pilot. (2020). *A NodeJS mongoose Schema Example*. Recuperado de <https://kb.objectrocket.com/mongo-db/a-nodejs-mongoose-schema-example-1204>
- Expresar. (s.f.). *Installing*. Recuperado de <https://expressjs.com/en/starter/installing.html>
- MySQLTUTORIAL. (2022). *Connecting to the MySQL database server from Node.js*. Recuperado de <https://www.mysqltutorial.org/mysql-nodejs/connect/>
- MySQLTUTORIAL. (s.f.). *MySQL Node.js*. Recuperado de: <https://www.mysqltutorial.org/mysql-nodejs/>
- Node.js. (s.f.). *How do I start with Node.js after I installed it?* Recuperado de <https://nodejs.org/en/docs/guides/getting-started-guide/>
- oblancarte. (2017). *Introducción a NodeJS (JavaScript del lado del Servidor)*. Recuperado de <https://www.oscarblancarteblog.com/2017/05/29/introduccion-a-nodejs-2/>

## Para saber más

**Los siguientes enlaces son externos a la Universidad Tecmilenio, al acceder a ellos considera que debes apegarte a sus términos y condiciones.**

### Lecturas

Para conocer más acerca de **Node.js**, te sugerimos leer lo siguiente:

- Mdn Web docs. (s.f.). *Introducción a Express/Node*. Recuperado de [https://developer.mozilla.org/es/docs/Learn/Server-side/Express\\_Nodejs/Introduction](https://developer.mozilla.org/es/docs/Learn/Server-side/Express_Nodejs/Introduction)
- Kiessling, M. (s.f.). *The Node Beginner Book*. Recuperado de <https://www.nodebeginner.org/>

### Videos

Para conocer más acerca de **Node.js**, te sugerimos revisar lo siguiente:

- Bitech Studio. (2018, 15 de enero). *¿Qué es Node.js? Breve explicación animada* [Archivo de video]. Recuperado de <https://www.youtube.com/watch?v=xJzzu7MVZXw>
- Programador X. (2021, 29 de enero). *¿Javascript en el Backend? ¿Qué es Node.js?* [Archivo de video]. Recuperado de <https://www.youtube.com/watch?v=EsKJlhWs1XY>

## Asegúrate de:

- Comprender la utilidad de Node.js en el back end para lograr la escalabilidad de un desarrollo web.
- Comprender como obtener la información de una base de datos usando Node.js para brindar la información necesaria al usuario.
- Comprender la importancia de la escalabilidad de un desarrollo web para mantener el mejor rendimiento de un desarrollo web.

## Requerimientos técnicos

- Computadora.
- Node.js
- Express.js

## Prework

- Deberás revisar el tema de Javascript.
- Deberás revisar el tema de bases de datos.
- Descarga Node.js:



node. (s.f.). *Descargas*. Recuperado de  
<https://nodejs.org/es/download/>

**El uso y descarga del software deberá apegarse a los términos y condiciones del sitio oficial del fabricante y su uso será responsabilidad de quien lo descargue. Tecmilenio no tiene licencia ni posee los derechos sobre dicho software.**

- Descarga Express.js:



Express. (s.f.). *Installing*. Recuperado de  
<https://expressjs.com/en/starter/installing>

**El uso y descarga del software deberá apegarse a los términos y condiciones del sitio oficial del fabricante y su uso será responsabilidad de quien lo descargue. Tecmilenio no tiene licencia ni posee los derechos sobre dicho software.**

- Descarga el txt de la actividad (Archivo adjunto)

La obra presentada es propiedad de ENSEÑANZA E INVESTIGACIÓN SUPERIOR A.C. (UNIVERSIDAD TECMILENIO), protegida por la Ley Federal de Derecho de Autor; la alteración o deformación de una obra, así como su reproducción, exhibición o ejecución pública sin el consentimiento de su autor y titular de los derechos correspondientes es constitutivo de un delito tipificado en la Ley Federal de Derechos de Autor, así como en las Leyes Internacionales de Derecho de Autor.

El uso de imágenes, fragmentos de videos, fragmentos de eventos culturales, programas y demás material que sea objeto de protección de los derechos de autor, es exclusivamente para fines educativos e informativos, y cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por UNIVERSIDAD TECMILENIO.

Queda prohibido copiar, reproducir, distribuir, publicar, transmitir, difundir, o en cualquier modo explotar cualquier parte de esta obra sin la autorización previa por escrito de UNIVERSIDAD TECMILENIO. Sin embargo, usted podrá bajar material a su computadora personal para uso exclusivamente personal o educacional y no comercial limitado a una copia por página. No se podrá remover o alterar de la copia ninguna leyenda de Derechos de Autor o la que manifieste la autoría del material.