

# Servidor HTTP en Express

“

Una de las tantas **funcionalidades** que ofrece Express es la posibilidad de **levantar** un **servidor web** de manera muy **sencilla** y en pocos pasos.



”

## {código}

```
const express = require('express');  
const app = express();
```

Requerimos el módulo de **Express** y almacenamos la función que nos devuelve en la constante **express**.

```
app.get('/', (req, res) => {  
  res.send('¡Hola mundo!');  
});
```

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

## {código}

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

```
const app = express();
```

```
app.get('/', (req, res) => {  
  res.send('¡Hola mundo!');  
});
```

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

Ejecutamos la función y almacenamos el objeto que devuelve en la constante **app**.

Ahora, a través de la misma, vamos a tener acceso a todas las propiedades y métodos que nos da Express.

## {código}

```
const express = require('express');  
const app = express();
```

```
app.get('/', (req, res) => {  
  res.send('¡Hola mundo!');  
});
```

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

Al objeto **app** le pedimos el método **listen**, que se encargará de levantar el servidor.

Recibe dos parámetros: el **puerto** y, opcionalmente, un **callback** que se ejecutará si el servidor termina de iniciar.

“

Con el servidor levantado, solo nos faltaría **definir** las **rut**as para empezar a manejar las **respuestas** de nuestra aplicación.



”

## {código}

```
const express = require('express');  
const app = express();
```

```
app.get('/', (req, res) => {  
  res.send('¡Hola mundo!');  
});
```

Al objeto **app** le pedimos el método **get**.

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

## {código}

```
const express = require('express');  
const app = express();
```

```
app.get('/', (req, res) => {  
  res.send('¡Hola mundo!');  
});
```

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

El método recibe dos parámetros:

El primero es un string que define la **URL** de la ruta.

El segundo es un **callback** con dos parámetros: request (req) y response (res).

**req** contendrá todas las propiedades y métodos de la petición que llegó.

**res** contendrá todas las propiedades y métodos de la respuesta que vamos a enviar.



## {código}

```
const express = require('express');  
const app = express();
```

```
app.get('/', (req, res) => {  
  res.send('¡Hola mundo!');  
});
```

El nombre de los parámetros del callback puede ser el que queramos.

Se estila usar **req** para request y **res** para response.

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

## {código}

```
const express = require('express');  
const app = express();
```

```
app.get('/', (req, res) => {  
  res.send('¡Hola mundo!');  
});
```

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

Dentro del callback definimos la respuesta que enviaremos.

Al objeto res (response), le pedimos el método **send**.

Como parámetro le pasamos lo que queremos mostrar en el navegador. En este caso, el texto: *¡Hola mundo!*

DigitalHouse>  
Coding School