

SERVIDOR HTTP EN EXPRESS

“

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



```
{ código }
```

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

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

```
{ código }
```

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

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

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.listen();
```

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

```
{ código }
```

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

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

El método recibe dos parámetros:

El primero, el número de puerto en el que queremos que se ejecute la aplicación.

El segundo (*opcional*), un callback que retorne un `console.log` para saber si el servidor se levantó correctamente.

Con el servidor levantado sólo nos faltaría **definir** las **rut**as para empezar a manejar los **response** de nuestra aplicación.



```
{ código }
```

```
const express = require('express');  
const app = express();  
  
app.listen(3000, () => console.log('Servidor corriendo'));  
  
app.get('/', (req, res) => {  
  res.send('¡Hola mundo!');  
})
```

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


```
{ código }
```

```
const express = require('express');  
const app = express();  
  
app.listen(3000, () => console.log('Servidor corriendo'))  
  
app.get('/', (req, res) => {  
  res.send('¡Hola mundo!');  
})
```

El método recibe dos parámetros:

El primero, un string que define la url de la ruta.

El segundo, un callback con dos parámetros: objetos request y response que nos pone a disposición Express cada vez que trabajamos con algún método de petición HTTP.

```
{ código }
```

```
const express = require('express');  
const app = express();  
  
app.listen(3000, () => console.log('Servidor corriendo'));  
  
app.get('/', (req, res) => {  
    res.send('¡Hola mundo!');  
})
```

El nombre de esos parámetros puede ser el que queramos. Se estila usar **req** para **request** y **res** para **response**.

```
{ código }
```

```
const express = require('express');  
const app = express();  
  
app.listen(3000, () => console.log('Servidor corriendo'))  
  
app.get('/', (req, res) => {  
  res.send('¡Hola mundo!');  
})
```

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 browser. En este caso, el texto ¡Hola mundo!.