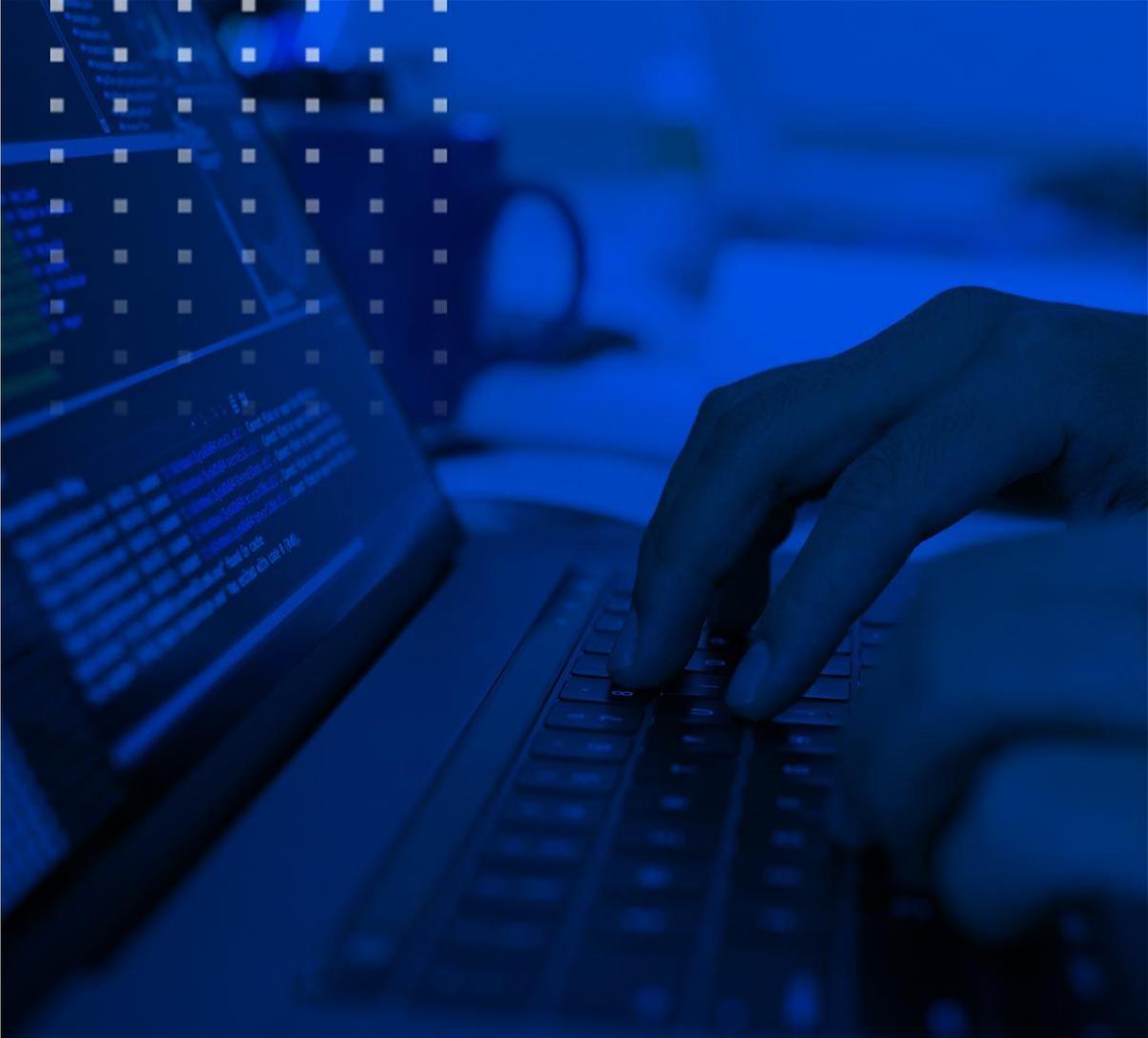


CURSO

FULL STACK DEVELOPER NIVEL INICIAL

UNIDAD 17
Node.js
Captura de datos



RUTAS

```
// GET method route
app.get('/', function (req, res) {
  res.send('GET > vamos a la página homepage');
});

// POST method route
app.post('/', function (req, res) {
  res.send('POST > vamos a la página homepage');
});
```

MÉTODOS

- **POST (create):** cuando mandamos información para insertar por ejemplo un registro en la base de datos. La información es enviado en el cuerpo de la petición, es decir que los datos no son visibles al usuario.
- **GET (read):** es usado para modo lectura, por ejemplo: cuando queremos listar a todos los usuarios de nuestra base de datos. Los parámetros son enviados por la URL.
- **PUT (update):** cuando queremos actualizar un registro. Actualizar la información de un usuario X.
- **DELETE (delete):** cuando queremos eliminar un registro. Borrar un usuario X de nuestra base de datos.

GET

```
app.get("/hello", (req, res) => {  
    res.send({  
        mensaje: 'Petición get'  
    });  
});
```

POST

```
this.app.post("/hello", (req, res) => {  
  res.json({  
    mensaje: 'Petición post'  
  });  
});
```

OBTENER DATOS DE UN POST

Enviar un body en la petición

Tenemos que configurar un middleware en express

```
//Lectura y parseo del body
```

```
this.app.use ( express.json() );
```

¡A Practicar!

Obtener datos POST

1. Escribí un json válido en postman.
2. Recibí la información en formato json y reflejarla en la respuesta.

Para ello crea un **middleware** y realizá las modificaciones necesarias en el **body**.

3. Desestructurá del body datos específicos.

PARÁMETROS DE SEGMENTOS

Por ejemplo, actualización de un usuario. En la ruta:

```
router.put("/:id", usuariosPut );
```

En el controlador:

```
const usuariosPut=("/", (req, res) => {  
  const id =req.params.id;  
  res.json({  
    mensaje: 'Petición put',  
    id  
  });  
});
```

Query params

Una de las formas que tenemos para enviar parámetros al API REST son los Query Params, los cuales son una serie de clave-valor que se agregan al final de la URL, justo después del signo de interrogación (?).

<http://localhost:8081/api/usuario?nombre=cosme&apellido=fulanito&age=35>

Query params con express

Express lo hace muy fácil al completar el Request.query objeto por nosotros:

```
app.get('/', (req, res) => {  
  console.log(req.query)  
})
```

¡A Practicar!

Query params

1. Escribí una url con Query params-
2. Obtené todos los datos, verificá la respuesta con postman.
3. Desestructurá el contenido recibido para obtener datos específicos.
4. Luego, realizá la modificaciones necesarias para obtener valores por defecto.



UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES