



Recuerda marcar tu asistencia



Asistencia en LMS



Semana 2

Express JS

Ing Edwin Garcia



El futuro digital
es de todos

MinTIC

Hechos

QUE

CONECTAN

Misión
TIC 2022

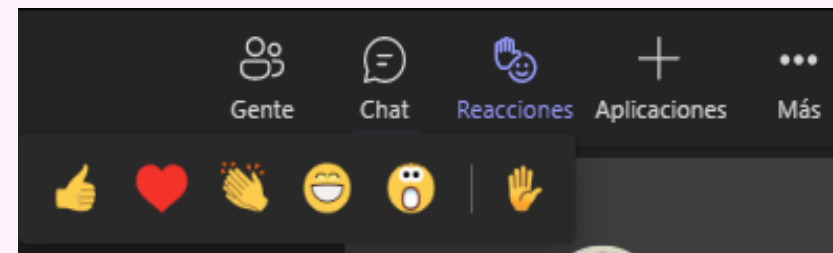
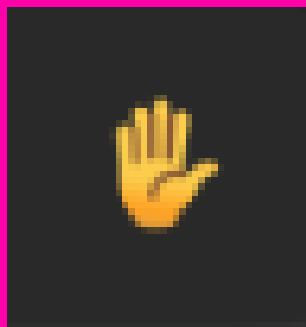
unab
VIGILADA MINEDUCACIÓN



Recuerden marcar su asistencia

Asistencia en LMS

**Pedir la
palabra para
realizar una
pregunta o
una
intervención**





- Express es el framework web más popular de *Node*, y es la librería subyacente para un gran número de otros frameworks web de Node populares. Proporciona mecanismos para:
- Escritura de manejadores de peticiones con diferentes verbos HTTP en diferentes caminos URL (rutas).



- Integración con motores de renderización de "vistas" para generar respuestas mediante la introducción de datos en plantillas.
- Establecer ajustes de aplicaciones web como qué puerto usar para conectar, y la localización de las plantillas que se utilizan para renderizar la respuesta.
- Añadir procesamiento de peticiones "middleware" adicional en cualquier punto dentro de la tubería de manejo de la petición.



```
> npm i express
```



```
JS index.js > ...
1  var express = require('express');
2  var app = express();
3
4  app.get('/', function(req, res) {
5    res.send('Hola Mundo!');
6  });
7
8
9  app.listen(3000, function() {
10    console.log('Aplicación ejemplo, escuchando el puerto 3000!');
11  });
12
```



localhost:3000

Hola Mundo!



```
index.js > ...
1  var express = require('express');
2  var app = express();
3
4  app.get('/', function(req, res) {
5    res.send('Hola Mundo!');
6  });
7
8  // About page route
9  app.get('/about', function(req, res) {
10    res.send('Acerca de esta wiki');
11  });
12
13 app.listen(3000, function() {
14   console.log('Aplicación ejemplo, escuchando el puerto 3000!');
15 });
16
```



localhost:3000/about

Acerca de esta wiki



Rutas

Las rutas le permiten igualar patrones particulares de caracteres en la URL, y extraer algunos valores de ella y pasarlos como parámetros al manejador de rutas (como atributo del objeto petición pasado como parámetro).

Además estas se puede organizar dentro de un directorio, manejarse por entidades y aún así ser exportadas para ser usadas por otro modulo.



Desarrollo de Aplicaciones Web



El futuro digital
es de todos

MinTIC

```

v OPEN EDITORS
  JS index.js
  X JS paginas.js routes
  {} package.json
v EXPRESSTEST
  > node_modules
  v routes
  JS paginas.js
  JS index.js
  {} package-lock.json
  {} package.json

routes > JS paginas.js > ...
1  const express = require('express')
2  const router = express.Router()
3
4  router.get('/', function(req, res) {
5    res.send('Acerca de esta wiki');
6  });
7
8
9  module.exports = router

```

```

EXPLORER
v OPEN EDITORS
  X JS index.js
  JS paginas.js routes
  {} package.json
v EXPRESSTEST
  > node_modules
  v routes
  JS paginas.js
  JS index.js
  {} package-lock.json
  {} package.json

JS index.js > ...
1  var express = require('express');
2  var app = express();
3
4
5  app.get('/', function(req, res) {
6    res.send('Hola Mundo!');
7  });
8
9  //importar rutas
10 app.use("/api/acerca", require("../routes/paginas"));
11
12

```



Desarrollo de Aplicaciones Web



El futuro digital
es de todos

MinTIC



localhost:3000/api/acerca

Acerca de esta wiki

Mision
TIC 2022



Hechos
QUE CONECTAN ✓



Controllers

Se puede construir módulos que nos permiten procesar la lógica de nuestros proyecto y también segmentarla por entidades, esta funcionalidad de la el nombre de controladores.



▼ controllers

JS equiposCtrl.js

```
2
3  const equipoListar = ()=>{
4      listado = {
5          '1' : 'nacional',
6          '2' : 'Millonarios',
7          '3' : 'Cali',
8          '4' : 'Junior'
9      }
10     return listado
11 }
12
```

```
module.exports = {
  ...
  equipoListar
}
```



Probamos nuestro código en index.js

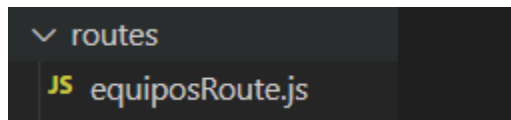
```
console.log("Listado equipos".bgBlue)  
console.log(equipos.equipoListar())
```

```
PS C:\Users\LENOVO\Desktop\node\expressTest> node index  
Listado equipos  
{ '1': 'nacional', '2': 'Millonarios', '3': 'Cali', '4': 'Junior' }  
Aplicación ejemplo, escuchando el puerto 3000!  
█
```



Agregando el controlador a las rutas

Escribimos el código para la ruta



```
routes > JS equiposRoute.js > router.get('/') callback
1  const express = require('express')
2  const router = express.Router()
3  const equiposCtrl = require('../controllers/equiposCtrl')
4
5  router.get('/', function(req, res) {
6    res.send(equiposCtrl.equipoListar());
7  });
8
9
10 module.exports = router
```



En index procedemos a agregar la ruta programada previamente

```
app.use("/api/equipo/",require('./routes/equiposRoute'))
```



Códigos de respuesta de servidor

Los códigos de estado de respuesta HTTP indican si se ha completado satisfactoriamente una solicitud HTTP específica. Las respuestas se agrupan en cinco clases:



- 1.Respuestas informativas (100–199),
- 2.Respuestas satisfactorias (200–299),
- 3.Redirecciones (300–399),
- 4.Errores de los clientes (400–499),
- 5.y errores de los servidores (500–599).



Respuestas informativas

100 Continue

Esta respuesta provisional indica que todo hasta ahora está bien y que el cliente debe continuar con la solicitud o ignorarla si ya está terminada.



Respuestas satisfactorias

- GET: El recurso se ha obtenido y se transmite en el cuerpo del mensaje.
- HEAD: Los encabezados de entidad están en el cuerpo del mensaje.
- PUT o POST: El recurso que describe el resultado de la acción se transmite en el cuerpo del mensaje.

200 OK

La solicitud ha tenido éxito. El significado de un éxito varía dependiendo del método HTTP:



Redirecciones

301 Moved Permanently (en-US)

Este código de respuesta significa que la URI del recurso solicitado ha sido cambiado. Probablemente una nueva URI sea devuelta en la respuesta.



Errores de cliente

400 Bad Request

Esta respuesta significa que el servidor no pudo interpretar la solicitud dada una sintaxis inválida.

401 Unauthorized

Es necesario autenticar para obtener la respuesta solicitada. Esta es similar a 403, pero en este caso, la autenticación es posible.



403 Forbidden

El cliente no posee los permisos necesarios para cierto contenido, por lo que el servidor está rechazando otorgar una respuesta apropiada.

404 Not Found

El servidor no pudo encontrar el contenido solicitado. Este código de respuesta es uno de los más famosos dada su alta ocurrencia en la web.



Errores de servidor

500 Internal Server Error

El servidor ha encontrado una situación que no sabe cómo manejarla.



Métodos de petición HTTP

HTTP define un conjunto de **métodos de petición** para indicar la acción que se desea realizar para un recurso determinado. Aunque estos también pueden ser sustantivos, estos métodos de solicitud a veces son llamados *HTTP verbs*. Cada uno de ellos implementan una semántica diferente, pero algunas características similares son compartidas por un grupo de ellos



GET

El método GET solicita una representación de un recurso específico. Las peticiones que usan el método GET sólo deben recuperar datos.

POST

El método POST se utiliza para enviar una entidad a un recurso en específico, causando a menudo un cambio en el estado o efectos secundarios en el servidor.



PUT

El modo PUT reemplaza todas las representaciones actuales del recurso de destino con la carga útil de la petición.

DELETE

El método DELETE borra un recurso en específico.



POST

El **método HTTP POST** envía datos al servidor. El tipo del cuerpo de la solicitud es indicada por la cabecera Content-Type.

La diferencia entre PUT y POST es que PUT es idempotente: llamarlo una o varias veces sucesivamente tiene el mismo efecto (no tiene efecto secundario // colateral), mientras que varios POST idénticos pueden tener efectos adicionales, como pasar una orden muchas veces.

Una solicitud POST es típicamente enviada por un formulario HTML y resulta en un cambio en el servidor.



<https://developer.mozilla.org/es/docs/Web/HTTP/Methods>

<https://developer.mozilla.org/es/docs/Web/HTTP/Status>

Bibliografía





El futuro digital
es de todos

MinTIC

**Practicar los
temas tratados**

Mision
TIC2022



Hechos
QUE CONECTAN ✓

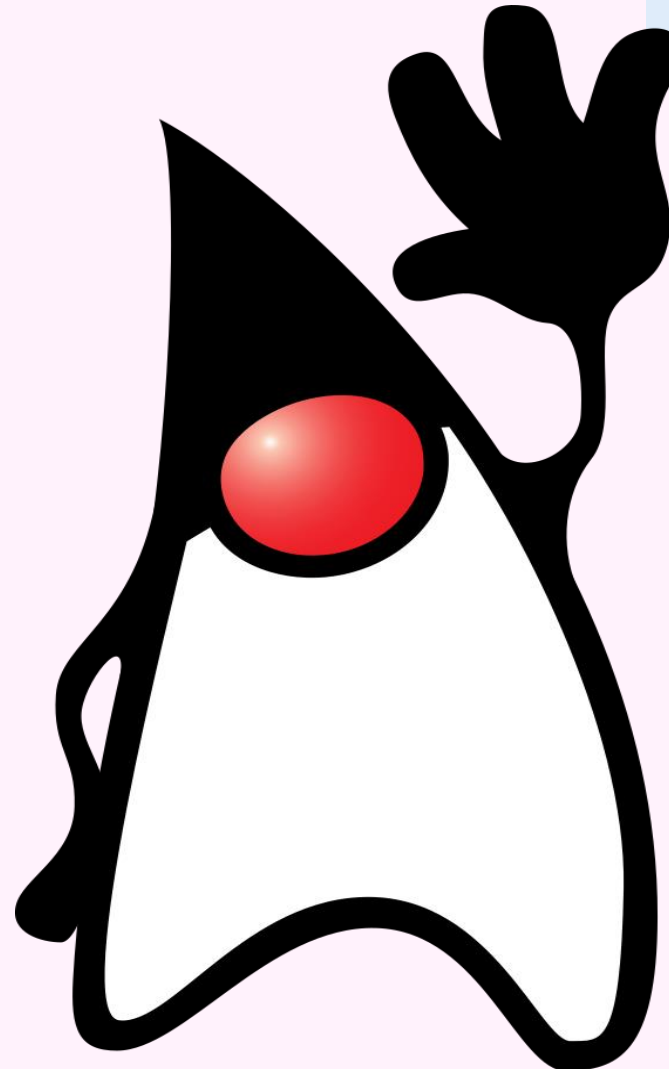
Nos vemos en la siguiente sesión



**El futuro digital
es de todos**

MinTIC

**Muchas
Gracias**



www.mintic.gov.co

**Misión
TIC2022**

unab
VIGILADA MINEDUCACIÓN



El futuro digital
es de todos

MinTIC



Misión
TIC 2022

Hechos
QUE CONECTAN ✓