



FACULTAD
DE INGENIERIA

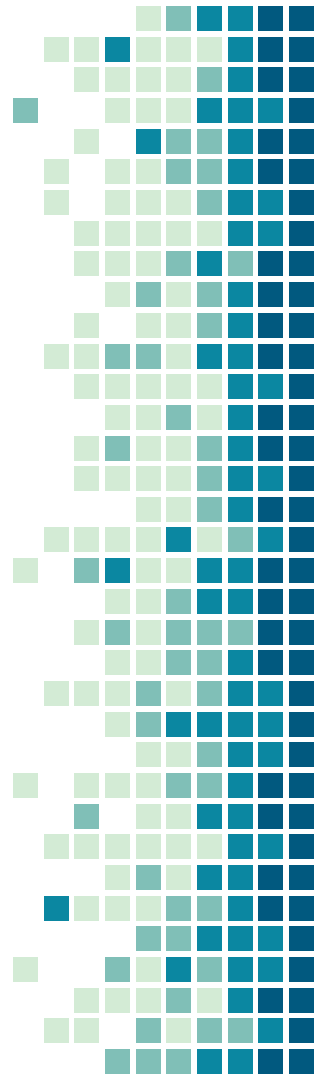
Universidad de Buenos Aires

Desarrollo de Aplicaciones Multiplataforma

Brian Ducca

Índice

- Web Api vs Web Service
- REST vs SOAP
- Métodos HTTP soportados por Express
- Middleware
- CORS

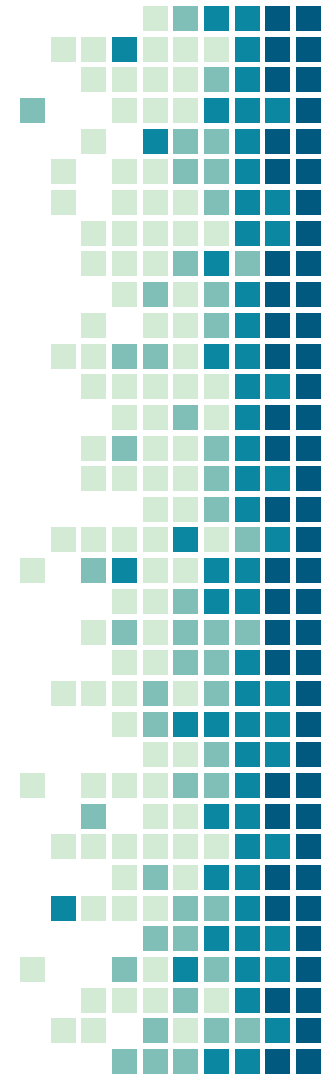


Web API

- Interfaz de programación de aplicaciones
- Acceso a características de bajo nivel o propietarias
- Detalla solamente la forma en que cada rutina debe ser llevada a cabo y la funcionalidad que brinda

Web Service

- Forma estandarizada de integrar aplicaciones web mediante el uso de XML, SOAP, WSDL y UDDI



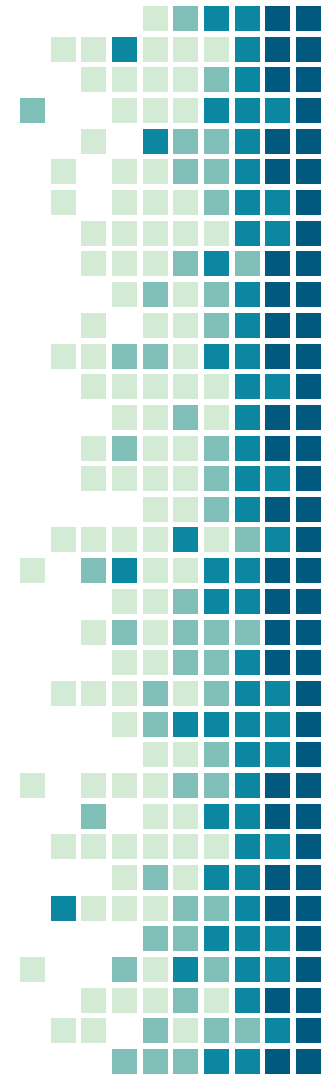
Rest

vs

Soap

- Estilo de arquitectura de software
- Liviano
- JSON
- No tan seguro como Soap

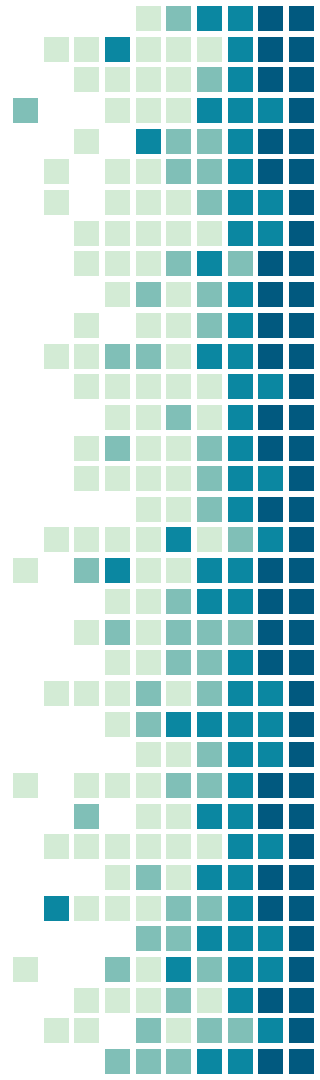
- Protocolo de mensajería
- XML
- Pueden ser transportados por protocolos: SMTP, MIME, HTTP
- Se utiliza mayormente en bancos por temas de seguridad



Manejadores de Rutas

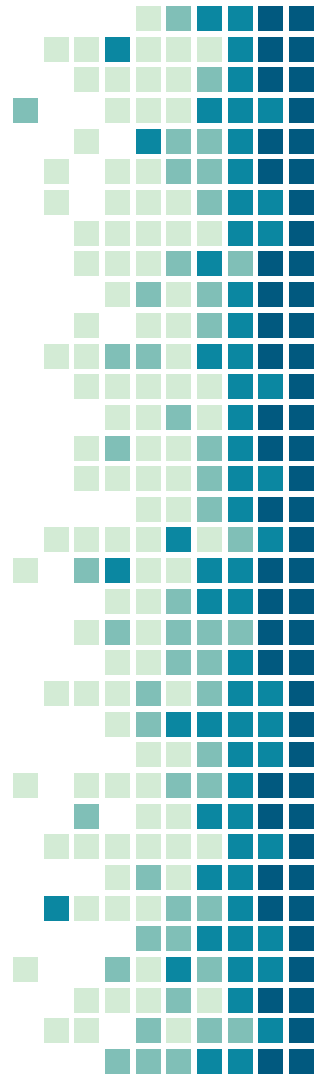
Existe un método de direccionamiento especial que no se deriva de ningún método HTTP:

```
app.all('/todo', function (req, res) {  
  res.send('Todo...');  
});
```



- Una matriz de funciones de devolución de llamada puede manejar una ruta

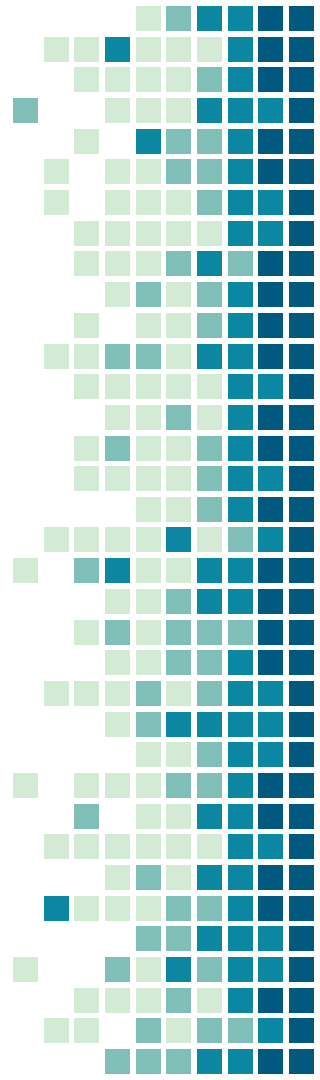
```
var cb0 = function (req, res, next) {  
  console.log('CB0');  
  next();  
}  
var cb1 = function (req, res, next) {  
  console.log('CB1');  
  next();  
}  
var cb2 = function (req, res) {  
  res.send('Hello from C!');  
}  
  
app.get('/example/c', [cb0, cb1, cb2]);
```



Middleware

Las funciones middleware son funciones que tienen acceso a los objetos de solicitud y respuesta "req" y "res" y a la siguiente función en el ciclo solicitud/respuesta de la aplicación.

Si la función no finaliza el ciclo de solicitud/respuesta, se debe invocar `next()` para pasar el control a la siguiente función.



- Para cargar la función middleware, se llama a `app.use()`, especificando la función.

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

```
var app = express ();
```

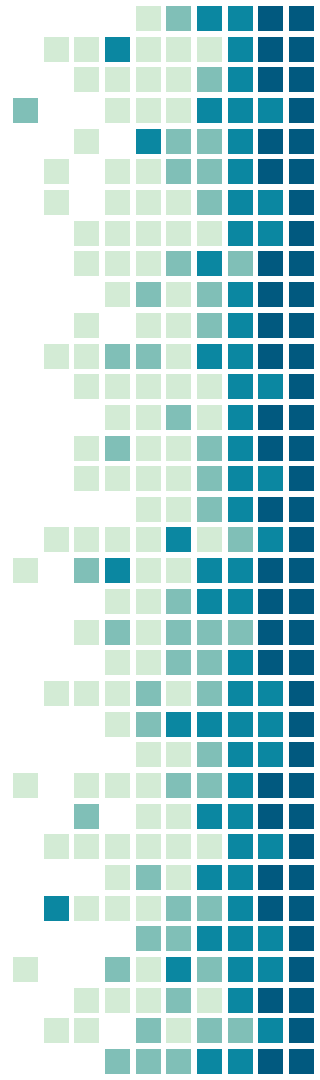
```
var myLogger = function (req, res, next) {
```

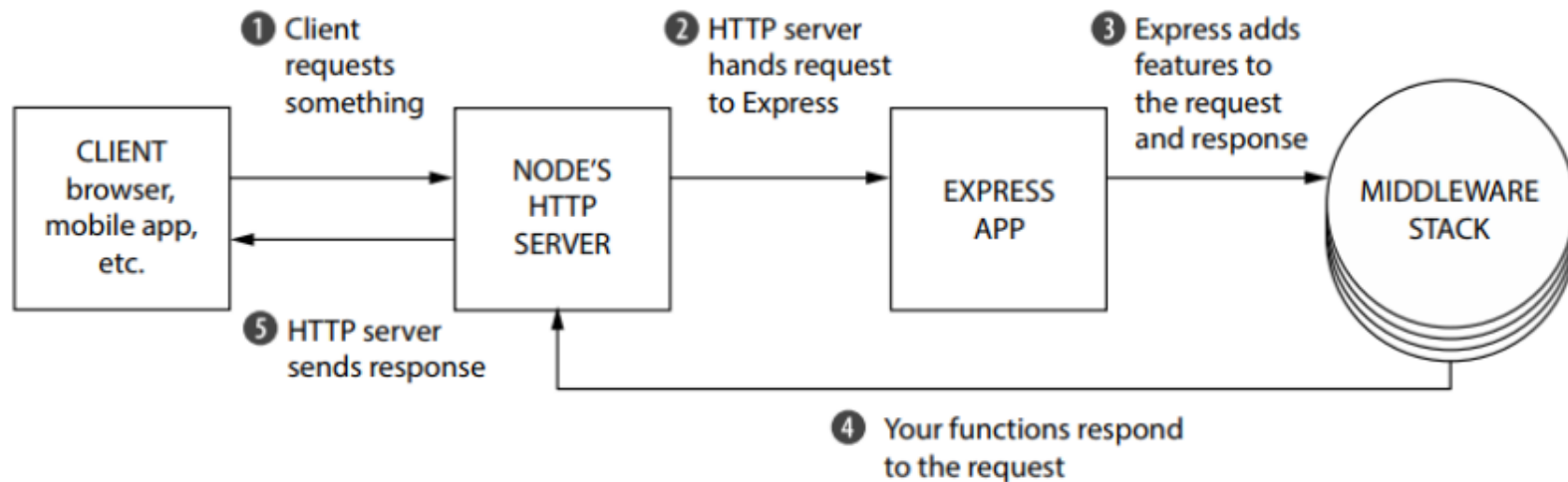
```
  console.log('LOGGED');
```

```
  next();
```

```
};
```

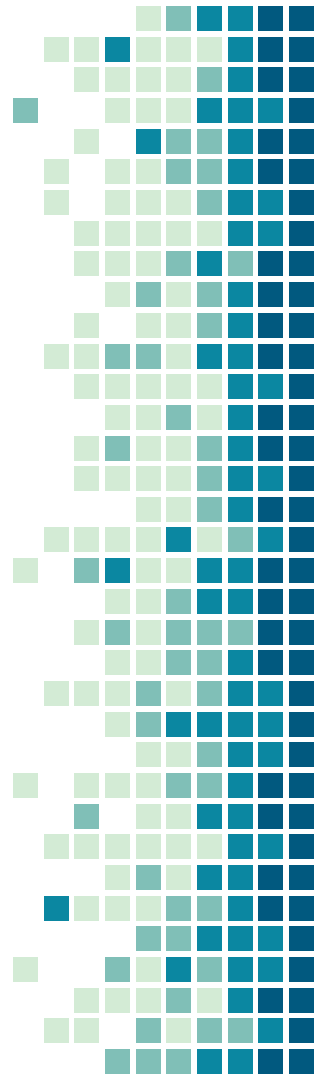
```
app.use(myLogger);
```





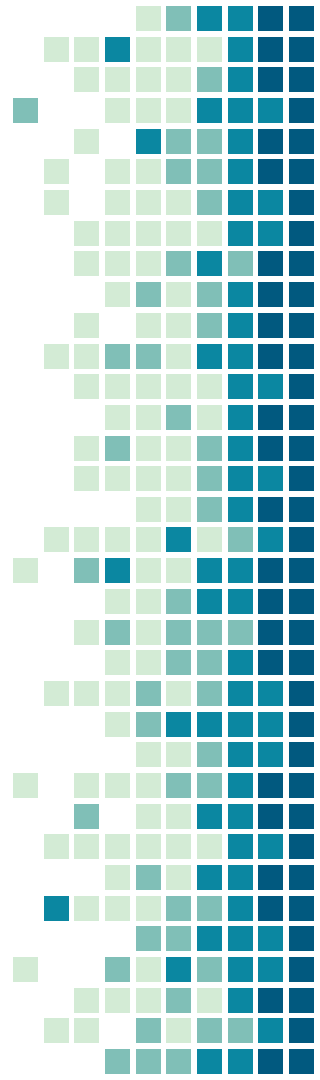
Tipos de Middleware

- Middleware a nivel de aplicación
- Middleware a nivel de direccionador
- Middleware de manejo de errores
- Middleware incorporado
- Middleware de terceros



Express Router

- Utilizado para crear manejadores de rutas modulares.
- Middleware
- Se crea un archivo de direccionador en el directorio de la aplicación , que luego se carga como un módulo dentro del index.js de la aplicación



Cors

- Mecanismo que utiliza cabeceras HTTP adicionales para permitir que un user-agent obtenga permiso para acceder a recursos desde un servidor en un origen distinto al que pertenece

