



Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

Introducción koa

IIC2513 - Tecnologías y Aplicaciones Web

Sebastián Vicencio R.
2do Semestre 2020

async/await

(Más) async moderno

async/await

¿Qué es?

Funcionalidad introducida en ES2017

Syntactic sugar sobre promesas para escribir código en una **forma que parezca síncrona**

async functions

Marcar función como asíncrona

```
async function myNewAsyncFunction() {  
    return 'some value';  
}
```

Función **retorna automáticamente una promesa**

await keyword

Esperar por una promesa

```
const value = await asyncFunction();
```

Se **antepone** a una promesa

Sólo **funciona dentro** de una async function

Promesas

¿Recuerdan esta función?

```
function asyncFunction() {  
    return Promise.resolve('You see? I promised');  
}
```

```
asyncFunction().then(function(promisedValue) {  
    console.log(promisedValue);  
})
```

async/await

Re-escribiendo promesas

```
async function myNewAsyncFunction() {  
    const value = await asyncFunction();  
    console.log('Inside async function:', value);  
  
    return 'some value';  
}
```

```
myNewAsyncFunction().then(value => console.log(value));
```

Como retorna una promesa, se puede **encadenar con then**

koa

Introducción al framework

koa

Características principales

1. Creado por el equipo que estuvo **detrás de Express**
2. **Liviano**
3. Arquitectura basada en **middlewares**
4. **Promesas (y async functions)** por sobre callbacks

Aplicación koa

Es un objeto que recibe un
request y produce un response

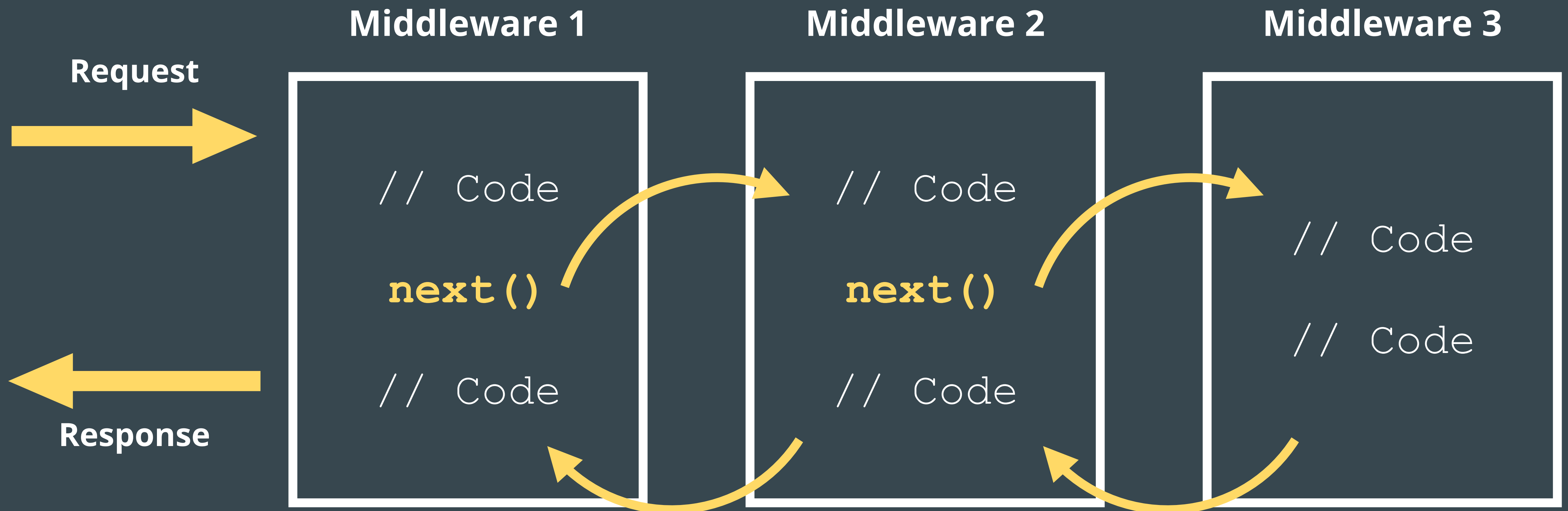
Middleware en koa

¿Qué es?

- Función que se ejecuta "entre medio" de proceso request/response
- Middleware "aporta" de alguna manera a la construcción del response
- Una aplicación koa puede tener muchos middlewares
- Middlewares siempre se ejecutan en un determinado orden

Middleware en koa

Cascada o funcionamiento tipo stack



Middleware en koa

¿Cómo se ve?

```
async function middleware(ctx, next) {  
  console.log(ctx.request);  
  console.log(ctx.response);  
  
  await next(); // call next middleware  
  // and await for it to finish if it returns a promise  
  
  console.log('Do more things');  
  console.log('after next middleware finishes');  
}
```

Ejemplo koa

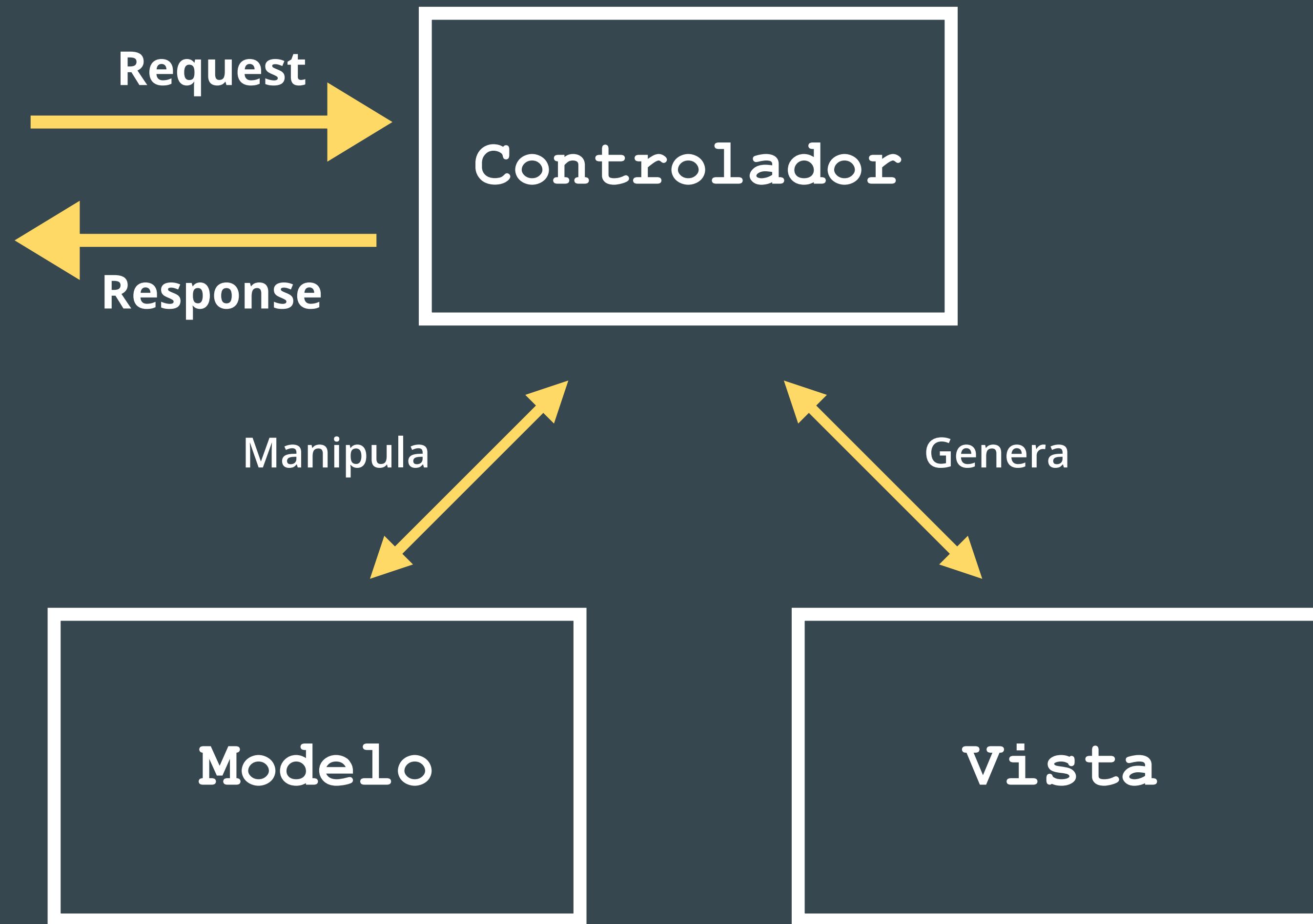
Una aplicación básica

MVC

Modelo - Vista - Controlador

MVC

¿Cómo funciona?



Template del curso

Basado en MVC

Modelo

Sequelize ORM

- Realiza mapping entre modelos (objetos JS) y tablas de BD
- Sistema de migraciones
- Validaciones de modelo
- Asociaciones entre modelos
- Queries "sin escribir" SQL

Vista

koa-ejs

- HTML con placeholders donde es posible incluir código JS
- Control de flujo
- Expresiones
- Inclusión de otros templates

Controlador

koa-router

- Asociar paths a una función tipo middleware
- Dentro de la función se suele generar el response
- Esquema jerárquico con sub-routers

Cómo utilizar template

Un ejercicio en clases