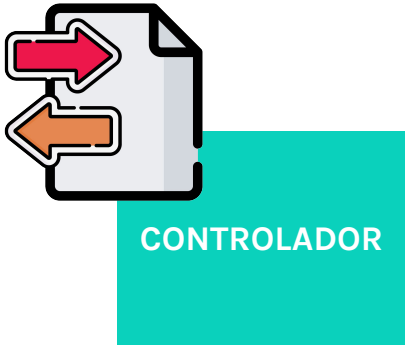
# Patrones de diseño MVC

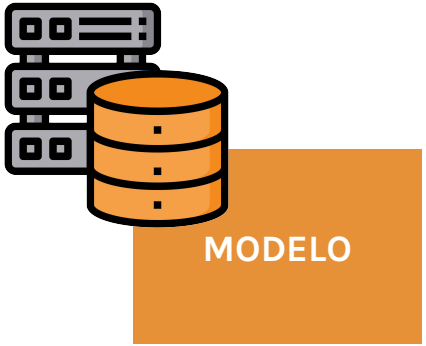
DigitalHouse >
Coding School

# Modelo Vista Controlador


Conforma la **interfaz gráfica** de la aplicación.




Conforma la capa intermedia entre las vistas y los modelos.




Conforma y contiene la **lógica** de la aplicación.

# Definiendo una ruta

A través de Express contamos con una **estructura básica** para definir cada una de las rutas y sus respuestas de nuestra aplicación:

```
{}
app.get('/', function (req , res) {
    res.send('¡Hola mundo!');
})
```

# Rutas parametrizadas

Usando la misma estructura básica para definir una ruta, aclaramos el parámetro haciendo uso de los dos puntos :, seguido del nombre que represente al dato que estará llegando en la URL.

```
app.get('/productos/:id', function (req,res) {
    // código
})
```

# Parámetros opcionales

Si queremos que un parámetro sea opcional, agregamos el signo de pregunta después del nombre del parámetro `?`.

En caso de que haya más de un parámetro, los opcionales deben ir siempre al final.

```
{}
app.get('/productos/:id?', function (req,res) {
    // código
})
```

# Sistema de ruteo

Definimos las rutas que consideremos necesarias para manejar distintos tipos de **request**.

```
// Ruta raíz de los productos / Inicio
router.get('/', (req, res) => {
    // código
});
// Ruta que muestra el detalle de un producto
router.get('/detalle/:id', (req, res) => {
    // código
});
```

# Creando un Controlador

Quitaremos el callback que habíamos definido en las rutas y lo escribiremos en el método index del controlador de productos.

```
{}
const controlador = {
    index: (req, res) => {
        res.send('¡Hola mundo!');
    },
};
```

Luego, exportaremos la variable en la última línea del archivo.

```
{}
module.exports = controlador;
```

# Implementando un Controlador

Para empezar a usar los métodos que definimos debemos requerir el módulo dentro del archivo de ruteo del recurso, en este caso `productos.js`, dentro de la carpeta `routes`.

```
{}  const prodController = require('../controllers/productosController');
```

Terminamos configurando aquella ruta a la que le quitamos el callback, reemplazando el callback anterior por el método del controlador que corresponda. Al ser un callback no le escribimos los paréntesis.

```
{}  router.get('/', productosController.index);
```


DigitalHouse>
Coding School