

# RUTAS PARAMETRIZADAS

“

Este tipo de rutas nos permite  
terminar de **configurar** nuestra  
**estructura** HTML para que sea  
100% **dinámica**.



# COMPORTAMIENTO DINÁMICO

Para lograr un comportamiento dinámico en las rutas que definimos en nuestro proyecto, es necesario entender que, al momento de hacer una petición al servidor *-sin importar qué método sea-*, podemos enviar **parámetros** en la **url**.

## ¿Qué quiere decir?

Que podremos definir qué valor de la url irá **variando** en cada petición.

# PARÁMETROS DINÁMICOS

Para definir un parámetro dinámico en la ruta escribimos el caracter `:` seguido de un **nombre representativo** del valor que estará viajando por url.

Suponiendo que estamos en el archivo de rutas del recurso *películas*, podríamos definir una ruta parametrizada así:

```
{ }
```

```
router.get("/:genero", peliculasController... )
```

De esta forma creamos una ruta que, según el género que reciba, devolverá una respuesta distinta en cada caso:

- `https://localhost/peliculas/drama` devuelve sólo las de drama
- `https://localhost/peliculas/accion` devuelve sólo las de acción

Podemos definir la **cantidad** de **parámetros** dinámicos que queramos dentro de una ruta, *sin importar el orden en que lo hagamos*:

```
router.get("/:genero/region/:anio", (req, res) ...)
```



Cuando a la ruta **no le llegan** todos los parámetros que tiene definidos, la misma **no funcionará**.

# PARÁMETROS OPTATIVOS

Para definir un parámetro **optativo** en la ruta escribimos el caracter `:` seguido de un **nombre representativo** del valor que estará viajando por url, seguido del caracter `?` al final.

Suponiendo que estamos en el archivo de rutas del recurso *películas*, podríamos definir un parámetro optativo así:

```
{ }
```

```
router.get("/:duracion?", peliculasController... )
```

Si la ruta recibe ese parámetro, filtrará la respuesta según ese valor. Si no llegara a recibir el parámetro, *seguirá funcionando*:

- `https://localhost/peliculas/120` devuelve sólo las de esa duración
- `https://localhost/peliculas` devuelve todas las películas

El **tipo** de parámetros y la **cantidad** que definamos en nuestras rutas va a depender **directamente** de la **lógica** de nuestra aplicación y de lo que necesitemos **mostrar** en cada caso.



# ATRAPANDO VALORES

A través de la propiedad `params` del objeto `request` podremos acceder a los valores que viajan en la url.

`params` es un objeto literal con *clave:valor*:

- la **clave** será el **nombre del parámetro** que definimos al crear la ruta
- el valor será el dato que viaje en la url

Teniendo una ruta parametrizada que usa el método `genero` del controlador de peliculas...

```
{ }
```

```
router.get("/:genero", peliculasController.genero);
```

... veamos cómo atrapar ese género que llegará por url



# ATRAPANDO VALORES

Dentro del método **genero** usaremos el objeto **request** para acceder al objeto **params** y mostrar por consola el resultado:

```
{}
```

```
const controller = {  
  genero: (req, res) => {  
    console.log(req.params.genero)  
  }  
}
```

Si la url fuera: `https://localhost/peliculas/terror`, veríamos por consola:

```
>_
```

```
terror
```

Recordá que **params** es un **objeto literal** al cual podrás acceder con la notación de punto a todas sus **propiedades** y **valores**.

*Dentro de los **métodos** de tus rutas irás desarrollando la **lógica** necesaria en cada caso para trabajar de manera **dinámica** con cada parámetro que te llegue.*

