

# Parámetros compartidos



La gran **ventaja** de utilizar un motor de plantillas es que nos permite **compartir** información **desde** los **controladores** hacia las **vistas** y disponer de ella como queramos.



# ¿Cómo mostramos una vista?

Sobre el objeto **response** ejecutamos el método `render()`, pasándole como argumento el nombre de la vista que queremos renderizar.

Recordemos que no hace falta el directorio inicial ni la extensión.

```
const controller = {  
  mostrarPelículas: (req, res) => {  
    res.render('películas')  
  }  
}
```

# ¿Cómo envío datos a una vista?

El método `render()` puede recibir un **objeto literal** como segundo parámetro. En ese objeto podremos almacenar la información que queremos enviar junto con la vista que estemos renderizando.

Toda la información que enviemos estará disponible para usar en la vista.

```
{  
  const controller = {  
    mostrarPeliculas: (req, res) => {  
      res.render('peliculas', { } )  
    }  
  }  
}
```

# El objeto de datos

Al objeto podemos asignarle tantas propiedades y valores como necesitemos.

El nombre de cada propiedad será el que usaremos para disponer de esa información de la vista.

```
{  
  const nombrePelicula = 'Batman';  
  const controller = {  
    mostrarPelicula: (req, res) => {  
      res.render('peliculas', { pelicula: nombrePelicula })  
    }  
  }  
}
```

# ¿Cómo recibe los datos la vista?

Para poder renderizar la variable enviada desde el controlador, la encerramos en la etiqueta de impresión: `<%= %>`.

Como `pelicula` es una variable de JavaScript, podemos usar todos los métodos disponibles, sabiendo que es un string, podríamos escribir algo como `<%= pelicula.toUpperCase() %>` para que se muestre **'BATMAN'**.

html

```
<div>
  <h1>Película Título: </h1>
  <span><%= pelicula %></span>
</div>
```

“

Podemos asignarle la cantidad de propiedades que deseemos a ese objeto.

**La vista**, por su parte, no recibirá un objeto literal con todas esas propiedades, sino que **recibirá las propiedades sueltas como variables.**

”



# Enviando más de un dato a la vista

Enviar más de un dato a la vista es tan simple como agregar más propiedades al objeto literal que habíamos armado.

```
const peliculas = ['Deadpool', 'The Joker', 'Batman'];
const controller = {
  mostrarPeliculas: (req, res) => {
    res.render('peliculas', {
      lista: peliculas,
      enOferta: true,
      genero: "acción"
    })
  }
}
```



# ¿Cómo recibe los datos la vista?

De la misma manera que vimos antes, podemos acceder a cada propiedad del objeto que creamos en el controlador usando sus nombres directamente ya que Express se encarga de transformarlas en variables dentro de la vista.

html

```
<% for(let pelicula of lista) { %>
  <h2>
    El título de la película es <%= pelicula %> y el género es <%= genero %>
  </h2>
  <% if (enOferta) { %>
    <p>25% de descuento alquilando esta película</p>
  <% } %>
<% } %>
```

DigitalHouse>  
Coding School