

# Lección 4 - Uso de plantillas y carga de contenido estático

## Tabla de contenido

- [Introducción](#)
- [Objetivos](#)
- [Desarrollo](#)
  - [Crear y Utilizar Plantillas:](#)
  - [Actualizar el Bootstrap y JQuery](#)
  - [Descargar e Instalar JQuery](#)
  - [Invocar los recursos estáticos de Bootstrap](#)
- [Conclusión](#)
- [Conoce más del autor](#)

## Introducción

En esta lección, nos sumergiremos en el uso de plantillas HTML y la carga de contenido estático, como archivos CSS y JavaScript, en una aplicación web desarrollada con Revel. La combinación de plantillas dinámicas y contenido estático es esencial para crear interfaces de usuario atractivas y funcionales en nuestras aplicaciones web.

---

## Objetivos

### 1. Integración de Bootstrap y JQuery:

- Aprenderemos cómo incorporar las bibliotecas Bootstrap y JQuery en un proyecto Revel para mejorar la apariencia y la funcionalidad de nuestra aplicación web.

### 2. Creación y Uso de Plantillas:

- Exploraremos cómo crear y utilizar plantillas HTML en Revel para generar contenido dinámico y personalizado en nuestras páginas web.

### 3. Renderización de Datos en Plantillas:

- Comprenderemos cómo pasar datos desde los controladores de Revel a las plantillas HTML y cómo renderizar estos datos dinámicamente en la interfaz de usuario de nuestra aplicación web.
-

# Desarrollo

## Crear y Utilizar Plantillas:

- En el controlador correspondiente, pase los datos a la plantilla y renderícela (`app/controllers/App.go`):

```
package controllers

import (
    "github.com/revel/revel"
)

type App struct {
    *revel.Controller
}

// Index es el controlador para la página de inicio.
// Renderiza una plantilla con los datos proporcionados.
func (c App) Index() revel.Result {
    datos := map[string]interface{}{
        "Titulo":  "Página de Inicio",
        "Mensaje": "¡Hola desde Revel!",
    }
    return c.Render(datos)
}
```

- En el directorio `app/views` de su proyecto Revel, cree archivos de plantilla con la extensión `.html`. Por ejemplo, `app/views/App/Index.html` podría ser una plantilla para la acción `Index` en el controlador `App`.

```
{{set . "title" "Home"}}
{{template "header.html" .}}

<header class="jumbotron" style="background-color:#4fdaaec7">
  <div class="container">
    <div class="row">
      <h1>{{.datos.Titulo}}</h1>
      <p>{{.datos.Mensaje}}</p>
    </div>
  </div>
</header>
```

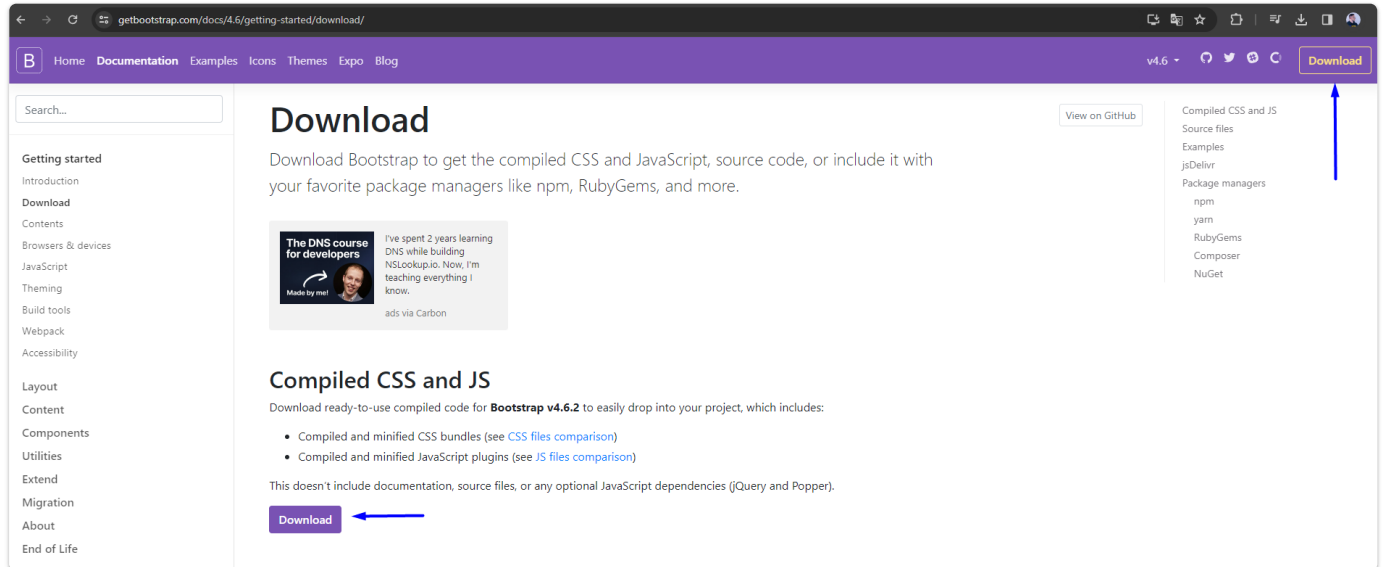
```
<div class="container">
  <div class="row">
    <div class="span6">
      {{template "flash.html" .}}
    </div>
  </div>
</div>

{{template "footer.html" .}}
```

## Actualizar el Bootstrap y JQuery

Buscar: Bootstrap en Google y selecciona en enlace de 4.6

Hacer click en el botón "Download", Luego en "Download Compiled CSS and JS".



Descomprimir los ficheros y ubicarlos en: `./public/css` y `./public/js`.

## Descargar e Instalar JQuery

Buscar: JQuery

Hacer click en el botón "Download JQuery v3.7.1" Descarga la primera opción: [Download the compressed, production jQuery 3.7.1](#) Guardas el fichero y lo agregas a `./public/js`

## Invocar los recursos estáticos de Bootstrap

Modifica el fichero `./app/views/header.html`:

```
<link rel="stylesheet" type="text/css"
href="/public/css/bootstrap.min.css">
```

```
<script src="/public/js/jquery-3.7.1.min.js"></script>
```

```
<script src="/public/js/bootstrap.min.js"></script>
```

Probar que aparece la web sin estilo, para ello, mejorar el HTML para que tenga coherencia con lo estilos de Bootstrap 4.6:

```
{{set . "title" "Home"}}
{{template "header.html" .}}

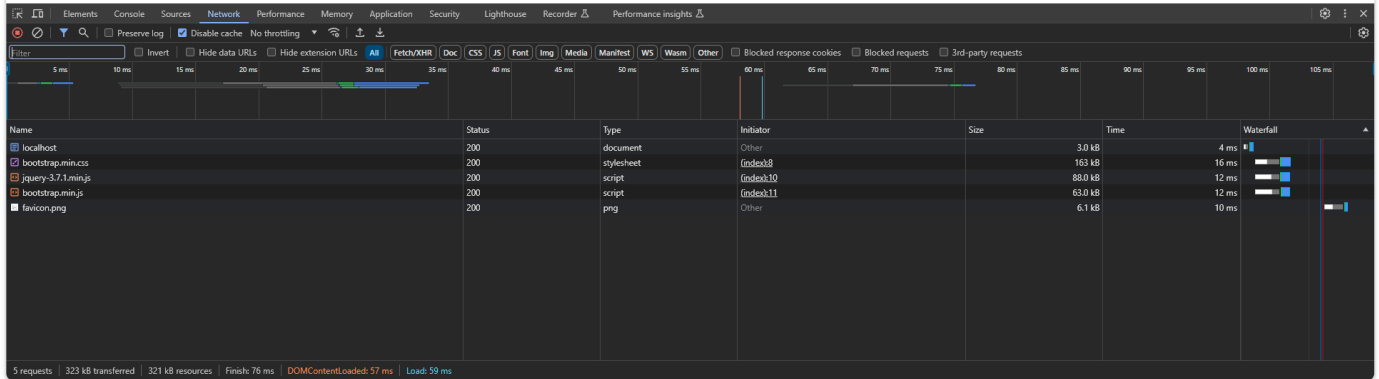
<header class="jumbotron bg-info text-white">
  <div class="container">
    <div class="row">
      <h1 class="display-1">{{.datos.Titulo}}</h1>
    </div>
    <div class="row">
      <p class="display-4">{{.datos.Mensaje}}</p>
    </div>
  </div>
</header>

<div class="container">
  <div class="row">
    <div class="span6">
      {{template "flash.html" .}}
    </div>
  </div>
</div>

{{template "footer.html" .}}
```

# Página de Inicio

¡Hola desde Revel!



Observamos que las solicitudes a los nuevos ficheros son correctas.

## Conclusión

Al dominar el uso de plantillas y la carga de contenido estático en Revel, podremos construir aplicaciones web más dinámicas y atractivas para nuestros usuarios. Estas habilidades son fundamentales para crear interfaces de usuario atractivas y funcionales en nuestras aplicaciones web, lo que mejora significativamente la experiencia del usuario final.

## Conoce más del autor

¡Encuéntrame en las siguientes redes sociales para estar al tanto de mis proyectos y actividades!

 <b>Red Social</b>	 <b>Enlace</b>
 <b>Página web</b>	<a href="https://jersonmartinez.com">jersonmartinez.com</a>
 <b>LinkedIn</b>	<a href="#">Jerson Martínez - DevOps Engineer</a>
 <b>Canales de YouTube</b>	<a href="#">DevOpsea</a>   <a href="#">Side Master</a>
 <b>GitHub</b>	<a href="#">Perfil en GitHub</a>
 <b>Twitter (X)</b>	<a href="#">@antoniomorenosm</a>