

Documentación Intervalos.

El modelo empleado para esta App Web es MVC, aunque no es necesario el modelo ya que no trabajamos con ninguna base de datos y también se utilizó el framework materialize para el diseño web, el arbol de documentos es el siguiente:

root-

```
wwwroot-
  css-
    materialize.css
  fonts-
    roboto-
      fuentes
  js-
    materialize.js
    intervals.js
controller-
  HomeController.cs
Views-
  Home-
    Index.cshtml
  Shared
    _Layout.cshtml
    _Error.cshtml
```

Metodos

Archivo: rotate.js

```
/*Esta función se llama cuando la página está lista y cargada en el browser*/
function ini()

/*El primer form que tiene los datos de los intervalos no es necesario ir al servidor
para crear
el formulario de ingreso de cada elemento de los intervalos, por eso se escucha el
evento de submit*/
$('body').on('submit', '#form-create', function(e)

//Antes de enviar los datos de los intervalos al servidor, se transforman a JSON y se
envían por AJAX
$('body').on('submit', '#form-overlapping', function(e)

//Evento para verificar que se introduzcan solamente numeros
$('body').on('keydown', 'input', function(e)

//Función inicial para crear el formulario de intervalos (número de intervalos)
init_body();

*Función para crear el layout del formulario para ingresar los datos de cada elemento
del intervalo*/
function create_input_intervals()

/*Función para imprimir los intervalos*/
```

```
function print_intervals(intervals_input, intervals_output) {
```

Archivo: HomeController.cs

```
/*Este metodo es el que manda a llamar la pagina de los intervalos a traves de  
intervals.js con Ajax*/
```

```
public IActionResult overlapping(string Intervals)
```

```
/*Metodo que converite un array de string de 2d en una lista*/
```

```
private List<List<int>>> Convert_string_list_intervals(string[,] intervals)
```

```
/*Metodo que compara cada intervalo con los demás y los une en dado caso*/
```

```
private List<List<int>>> Compare_intervals(List<List<int>>> intervals)
```