

TEMA 0 (INTRO).

- MODELO MVC.
- MODELO MVC EN EL MUNDO WEB.
- SPAs(SINGLE PAGE APPLICATIONS).
- ¿QUE ES ANGULAR?

TEMA 1.

- PLANTILLAS.
- VISTAS.
- DIRECTIVAS (ngApp, ngModel, etc).
- EXPRESIONES ({{}}).
- SCOPE (\$scope).
- MODELO.
- DOBLE DATA BINDING.
- CREANDO UNA APP ANGULAR.

EJ1.1: Utilizando "primera-app-mio.html", crea una app Angular.

EJ1.2: Utiliza ngModel para comprobar cómo funciona el maravilloso doble data binding de angular.

EJ1.3: Añade alguna expresión que evalúe esos valores que hemos recogido gracias a ngModel.

TEMA 2.

- MODULOS.
- CONTROLADORES (Vistas con lógica).

EJ2.1: Utilizando "gna-uilogic-mio.html", crea una app Angular (usando module) y añádele un controlador que genere números aleatorios.

- AÑADIENDO PROPIEDADES AL SCOPE.
- AÑADIENDO METODOS AL SCOPE (ngClick).

EJ2.2: Añade a tu controlador un método que ejecute la función que calcula números aleatorios cuando este sea convocado. Crea un botón y utiliza ngClick para poder llamar a este

método desde la UI.

- SERVICIOS (\$interval, \$timeout, etc).

EJ2.3: Mejora tu generador de números aleatorios de tal manera que al hacer click, se genere un número aleatorio cada segundo. Puedes usar \$interval inyectando el servicio en tu app.

EJ2.4: Refactoriza tu aplicación de tal manera que tu lógica no este en el index de tu app, sino en su correspondiente fichero .js

TEMA 3.

- DIRECTIVAS PROPIAS (template, link y attrs)

EJ3.1: Utilizando "gna-como-directiva.html" crea una app Angular que esté compuesta de una directiva creada por ti, que agrupe el código html que tenias hasta ahora.

EJ3.2: Utiliza la función link, para poder asociar la lógica de generar números aleatorios a la directiva, y sin hacer más falta el controlador.

EJ3.3: Pásale a tu gna a través de un atributo, el intervalo de tiempo en el que quieres que se genera un nuevo número.

TEMA 4.

- SERVICIOS PROPIOS

EJ4.1: Utilizando "gna-servicio.html" crea una app Angular nueva que mejore la que teníamos, delegando la lógica de generar números aleatorios en un servicio.

- ATACANDO API REST EXTERNA.
- PETICIONES GET Y POST (\$http)

EJ4.2: Utilizando "gna-api.html" crea una app Angular nueva que en vez de obtener los números aleatorios de un servicio angular, los obtenga utilizando una API externa.