

## Ejercicios de Refuerzo

---

Para cada refuerzo de módulo se debe crear una carpeta (ej.: *refuerzo\_modulo\_3/*, *refuerzo\_modulo\_4/*, etc...) y para cada ejercicio dentro de los refuerzos se debe crear una carpeta con el consecutivo (*refuerzo\_modulo\_3/lab1/*, *refuerzo\_modulo\_3/lab2/*, *refuerzo\_modulo\_4/lab1/*, etc...).

### 1. Refuerzo Módulo 3

#### 1.1. AngularJS: Expresiones, Filtros, data-binding

- 1.1.1. Crear un archivo con estructura HTML5 llamado ***"index.html"***
- 1.1.2. Descargar y almacenar las librerías necesarias de *AngularJS* en una carpeta llamada ***"js"***
- 1.1.3. Crear un script de JavaScript llamado ***"lab1a.js"***
- 1.1.4. En ***"lab1a.js"*** crear un modulo general para la aplicación con nombre ***"lab1a"***, utilizar el mismo nombre para el *ng-app* en el ***"index.html"***.
- 1.1.5. Crear un controlador llamado ***"lab1aController"***.
- 1.1.6. Crear 6 modelos para visualizar la siguiente información:
  - 1.1.6.1. ***\$scope.name*** => Nombre completo (con filtro Mayúscula sostenida)
  - 1.1.6.2. ***\$scope.id*** => Cédula de ciudadanía
  - 1.1.6.3. ***\$scope.profession*** => Profesión de la persona
  - 1.1.6.4. ***\$scope.salary*** => Sueldo (con filtro Currency en pesos)
  - 1.1.6.5. ***\$scope.phone*** => teléfono
  - 1.1.6.6. ***\$scope.address*** => Dirección
- 1.1.7. Utilizar *data-binding* de *AngularJS* para cambiar el modelo ***Nombre completo*** a través de un campo del tipo *input*
- 1.1.8. Probar en WAMP Server el ejercicio creado.

## 1.2. AngularJS: Expresiones, Iteraciones

- 1.2.1. Crear un archivo con estructura HTML5 llamado ***"index.html"***
- 1.2.2. Descargar y almacenar las librerías necesarias de *AngularJS* en una carpeta llamada ***"js"***
- 1.2.3. Crear un script de JavaScript llamado ***"lab1b.js"***
- 1.2.4. En ***"lab1b.js"*** crear un modulo general para la aplicación con nombre ***"lab1b"***, utilizar el mismo nombre para el *ng-app* en el ***"index.html"***.
- 1.2.5. Crear un controlador llamado ***"lab1bController"***.
- 1.2.6. Crear 2 modelos para visualizar la siguiente información:
  - 1.2.6.1. ***\$scope.angular\_filters*** => Array de los 9 filtros básicos de Angular (<https://docs.angularjs.org/api/ng#filter>)
  - 1.2.6.2. ***\$scope.angular\_input*** => Array de los 11 tipos de inputs básicos de Angular (<https://docs.angularjs.org/api/ng#input>)
- 1.2.7. Iterar y mostrar los modelos del punto 1.2.6.2.
- 1.2.8. Probar en WAMP Server el ejercicio creado.

## 1.3. AngularJS: Eventos, modelos

- 1.3.1. Crear un archivo con estructura HTML5 llamado ***"index.html"***
- 1.3.2. Descargar y almacenar las librerías necesarias de *AngularJS* en una carpeta llamada ***"js"***
- 1.3.3. Crear un script de JavaScript llamado ***"lab1c.js"***
- 1.3.4. En ***"lab1c.js"*** crear un modulo general para la aplicación con nombre ***"lab1c"***, utilizar el mismo nombre para el *ng-app* en el ***"index.html"***.
- 1.3.5. Crear un controlador llamado ***"lab1cController"***.
- 1.3.6. Crear 2 modelos para visualizar la siguiente información:
  - 1.3.6.1. ***\$scope.link\_image*** => modelo con la url del enlace a una imagen de prueba como: <http://dummyimage.com/200x200/000/34ffddaa>
  - 1.3.6.2. ***\$scope.changeLinkImage*** => función que cambia el valor del enlace de la url en el modelo ***\$scope.link\_image***.
- 1.3.7. Debe utilizar la directiva *ng-click* para el cambio de la imagen.
- 1.3.8. Probar en WAMP Server el ejercicio creado.

## 1.4. AngularJS: Modulos, Controllers, Routes

- 1.4.1. Crear un archivo con estructura HTML5 llamado ***"index.html"***
- 1.4.2. Descargar y almacenar las librerías necesarias de *AngularJS* en una carpeta llamada ***"js"***
- 1.4.3. Crear un script de JavaScript llamado ***"app.js"***
- 1.4.4. En ***"app.js"*** crear un modulo general para la aplicación con nombre ***"lab1d"***, utilizar el mismo nombre para el *ng-app* en el ***"index.html"***.
- 1.4.5. Implementar la siguiente tabla (tabla 1.) según los contenidos de los anteriores 3 ejercicios, por favor tenga en cuenta que las rutas y los controladores los va a definir en el *\$routeProvider* del módulo externo de *AngularJS* -> *ngRoute*, todo esto sobre el ***"app.js"*** del ejercicio en proceso:

Script	Nombre Módulo	Nombre Controlador	Controlador en base al ejercicio	Nombre de la vista en nueva carpeta views	Contenido de la vista según el body del index.html por ejercicio	ruta que debe interpretar AJAX
lab1a.js	lab1a	lab1aController	1.1.	lab1a.html	contenido body del index.html ejercicio 1.1.	/lab1a
lab1b.js	lab1b	lab1bController	1.2.	lab1b.html	contenido body del index.html ejercicio 1.2.	/lab1b
lab1c.js	lab1c	lab1cController	1.3.	lab1c.html	contenido body del index.html ejercicio 1.3.	/lab1c

- 1.4.6. En el ***"index.html"*** debe incluir los enlaces a las rutas de la tabla anterior en un sencillo menú.
- 1.4.7. Probar en WAMP Server el ejercicio creado.

## 2. Refuerzo Módulo 5

### 2.1. Workflows, NPM, package.json, GulpJS, Gulp plugins

- 2.1.1. Generar el archivo ***“package.json”***
- 2.1.2. Traer los archivos del ejercicio ***“1.4. AngularJS: Modulos, Controllers, Routes”***
- 2.1.3. Instalar las siguientes dependencias en desarrollo para la gestión de módulos en Node.JS para GulpJS y el workflow:
  - 2.1.3.1. gulp
  - 2.1.3.2. Minificación de los documento en HTML tanto el index.html
  - 2.1.3.3. Minificación de los scripts escritos en JavaScript excepto *angular.min.js*
- 2.1.4. Crear manualmente el archivo ***“gulpfile.js”***
- 2.1.5. Crear y configurar las tareas:
  - 2.1.5.1. ***html-min*** => aplicación del plugin instalado en el numeral 2.1.3.2.
  - 2.1.5.2. ***js-min*** => aplicación del plugin instalado en el numeral 2.1.3.3.
  - 2.1.5.3. ***default*** => Ejecución de las tareas 2.1.5.1. y 2.1.5.2.
- 2.1.6. Probar en WAMP Server el ejercicio.

### 2.2. Workflows, NPM, package.json, GulpJS, Gulp plugins, bower, bower.json

- 2.2.1. Generar el archivo ***“package.json”*** y ***“bower.json”***
- 2.2.2. Traer los archivos del ejercicio ***“1.4. AngularJS: Modulos, Controllers, Routes”*** y ponerlos en la raíz de este ejercicio.
- 2.2.3. Instalar las siguientes dependencias en desarrollo para la gestión de módulos en Node.JS para GulpJS y el workflow:
  - 2.2.3.1. gulp
  - 2.2.3.2. Minificación del documento en HTML y las vistas.
  - 2.2.3.3. Minificación de los scripts escritos en JavaScript
- 2.2.4. Instalar y gestionar las dependencias a través de bower para AngularJS y el módulo externo ngRoute guardarlos en bower.json como dependencias.

2.2.5. Crear manualmente el archivo ***“gulpfile.js”***

2.2.6. Crear y configurar las tareas:

2.2.6.1. ***html-min*** => aplicación del plugin instalado en el numeral 2.2.3.2.

2.2.6.2. ***js-min*** => aplicación del plugin instalado en el numeral 2.2.3.3.

2.2.6.3. angular-concat => Concatenar los archivos minificados de angular.min.js y angular-route.min.js instalados a través de bower.

2.2.6.4.

2.2.6.5. ***default*** => Ejecución de las tareas 2.2.5.1. y 2.2.5.2.

2.2.7. Probar en WAMP Server el ejercicio.

## 3. Refuerzo Módulo 6

### 3.1. Cordova, platforms, AngularJS

3.1.1. Crear proyecto en Cordova

3.1.2. utilizar el ejercicio ***“1.1. AngularJS: Expresiones, Filtros, data-binding”*** guardándolo en la carpeta ***“www”*** del proyecto cordova creado, por favor ***NO*** sobre escribir el ***“index.html”***, más bien copiar lo necesario del contenido de la etiqueta body del ejercicio utilizado.

3.1.3. Adicionar las plataformas:

3.1.3.1. Browser

3.1.3.2. FirefoxOS

3.1.3.3. Android

3.1.4. Probar el ejercicio con la plataforma ***browser*** y tomar un *print screen* de la pantalla

3.1.5. Probar el ejercicio con la plataforma ***FirefoxOS*** y tomar un *print screen* de la pantalla

3.1.6. Probar el ejercicio con la plataforma ***Android*** y tomar un *print screen* de la pantalla

### **3.2. Cordova, platforms, AngularJS, Routes**

- 3.2.1. Crear proyecto en Cordova
- 3.2.2. utilizar el ejercicio **“1.4. AngularJS: Modulos, Controllers, Routes”** guardándolo en la carpeta **“www”** del proyecto cordova creado, por favor **NO** sobre escribir el **“index.html”**, más bien copiar lo necesario del contenido de la etiqueta body del ejercicio utilizado.
- 3.2.3. Adicionar las plataformas:
  - 3.2.3.1. Browser
  - 3.2.3.2. FirefoxOS
  - 3.2.3.3. Android
- 3.2.4. Probar el ejercicio con la plataforma **browser** y tomar un *print screen* de la pantalla
- 3.2.5. Probar el ejercicio con la plataforma **FirefoxOS** y tomar un *print screen* de la pantalla
- 3.2.6. Probar el ejercicio con la plataforma **Android** y tomar un *print screen* de la pantalla

### **3.3. Cordova, platforms, AngularJS, Routes, Workflows, NPM, package.json, GulpJS, Gulp plugins, bower, bower.json (Opcional y Avanzado)**

- 3.3.1. Crear proyecto en Cordova
- 3.3.2. utilizar el ejercicio **“2.2 Workflows, NPM, package.json, GulpJS, Gulp plugins, bower, bower.json”** guardándolo en la carpeta **“www”** del proyecto cordova creado, por favor **NO** sobre escribir el **“index.html”**, más bien copiar lo necesario del contenido de la etiqueta body del ejercicio utilizado.
- 3.3.3. Adicionar las plataformas:
  - 3.3.3.1. Browser
  - 3.3.3.2. FirefoxOS
  - 3.3.3.3. Android
- 3.3.4. Probar el ejercicio con la plataforma **browser** y tomar un *print screen* de la pantalla

- 3.3.5. Probar el ejercicio con la plataforma **FirefoxOS** y tomar un *print screen* de la pantalla
- 3.3.6. Probar el ejercicio con la plataforma **Android** y tomar un *print screen* de la pantalla