

Desarrollo de aplicaciones para móviles

Modulo 1: Programación y plataformas

IONIC Instalación y “Hola Mundo”



Presentación:

En la presente unidad nos instalaremos IONIC 2 framework.

Para poder proceder con su instalación deberemos instalar Node JS.

Una vez que tengamos todo el entorno instalado procederemos a realizar nuestra primer aplicación con IONIC 2.



Objetivos:

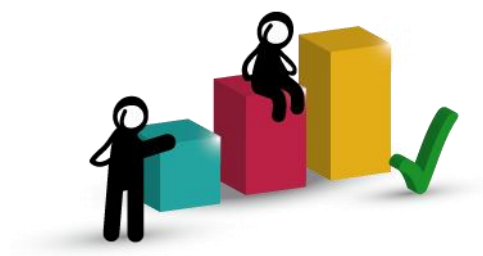
Que los participantes*:

- Instalen IONIC 2
- Instalen node js
- Conozcan que es Angular 2
- Conozcan que es typescript
- Creen su primer aplicación utilizando IONIC 2



Bloques temáticos*:

- Ionic
- Angular 2
- Soluciones implementadas en Angular 2
- Instalar editor de typescript
- ¿Qué es typescript?
- Instalar IONIC
- Crear una aplicación con IONIC
- Ejecutar la aplicación en el navegador



Consignas para el aprendizaje colaborativo

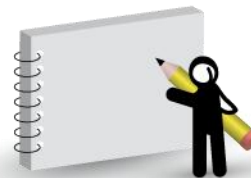
En esta Unidad los participantes se encontrarán con diferentes tipos de actividades que, en el marco de los fundamentos del MEC*, los referenciarán a tres comunidades de aprendizaje, que pondremos en funcionamiento en esta instancia de formación, a los efectos de aprovecharlas pedagógicamente:

- Los foros proactivos asociados a cada una de las unidades.
- La Web 2.0.
- Los contextos de desempeño de los participantes.

Es importante que todos los participantes realicen algunas de las actividades sugeridas y compartan en los foros los resultados obtenidos.

Además, también se propondrán reflexiones, notas especiales y vinculaciones a bibliografía y sitios web.

El carácter constructivista y colaborativo del MEC nos exige que todas las actividades realizadas por los participantes sean compartidas en los foros.



Tomen nota*

Las actividades son opcionales y pueden realizarse en forma individual, pero siempre es deseable que se las realice en equipo, con la finalidad de estimular y favorecer el trabajo colaborativo y el aprendizaje entre pares. Tenga en cuenta que, si bien las actividades son opcionales, su realización es de vital importancia para el logro de los objetivos de aprendizaje de esta instancia de formación. Si su tiempo no le permite realizar todas las actividades, por lo menos realice alguna, es fundamental que lo haga. Si cada uno de los participantes realiza alguna, el foro, que es una instancia clave en este tipo de cursos, tendrá una actividad muy enriquecedora.

Asimismo, también tengan en cuenta cuando trabajen en la Web, que en ella hay de todo, cosas excelentes, muy buenas, buenas, regulares, malas y muy malas. Por eso, es necesario aplicar filtros críticos para que las investigaciones y búsquedas se encaminen a la excelencia. Si tienen dudas con alguno de los datos recolectados, no dejen de consultar al profesor-tutor. También aprovechen en el foro proactivo las opiniones de sus compañeros de curso y colegas.

Ionic



Ionic es un framework para el desarrollo de aplicaciones mobile híbridas.

Esta basado en Angular 2 y Apache Cordova, y nos provee herramientas y servicios para desarrollar aplicaciones

híbridas con tecnologías web como CSS, HTML 5.

Para instalar IONIC debemos primer instalar Java JDK, Node JS y apache Cordova.

Podes seguir el siguiente tutorial para realizarlo, igualmente en este documento vas a encontrar todos los pasos:

<http://ionic.io/2>

Angular 2

Angular 2 es otro framework, no simplemente una nueva versión. A los que no conocían Angular 1 ésto les será indiferente, pero los que ya dominaban este framework sí deben entender que el conocimiento que necesitan adquirir es poco menos que si comenzasen desde cero. Obviamente, cuanta más experiencia en el desarrollo se tenga, más sencillo será lanzarse a usar Angular 2 porque muchas cosas sonarán de antes.

De ser una plataforma para la creación de Web Apps, ha evolucionado como motor de una enorme cantidad de proyectos del ámbito empresarial y de ahí para aplicaciones en la Web Mobile Híbrida, llevando la tecnología al límite de sus posibilidades.

Es el motivo por el que comenzaron a detectarse problemas en Angular 1, o necesidades donde no se alcanzaba una solución a la altura de lo deseable. Son las siguientes.

Javascript:

Para comenzar encontramos problemas en la creación de aplicaciones debido al propio Javascript. Es un lenguaje con carácter dinámico, asíncrono y de complicada depuración. Al ser tan particular resulta difícil adaptarse a él, sobre todo para personas que están acostumbradas a manejar lenguajes más tradicionales como Java o C#, porque muchas cosas que serían básicas en esos lenguajes no funcionan igualmente en Javascript.

Desarrollo del lado del cliente:

Ya sabemos que con Angular te llevas al navegador mucha programación que antes estaba del lado del servidor, comenzando por el renderizado de las vistas. Esto hace que surjan nuevos problemas y desafíos. Uno de ellos es la sobrecarga en el navegador, haciendo que algunas aplicaciones sean lentas usando Angular 1 como motor.

Soluciones implementadas en Angular 2

TypeScript / Javascript:

La sugerencia de usar TypeScript para desarrollar en Angular es casi una imposición porque la documentación y los generadores de código están pensados en TypeScript. Se supone que en futuro también estarán disponibles para Javascript, pero de momento no es así. De todos modos, para la tranquilidad de muchos, TypeScript no agrega más necesidad de procesamiento a las aplicaciones con Angular 2, ya que este lenguaje solamente lo utilizas en la etapa de desarrollo y todo el código que se ejecuta en el navegador es al final Javascript, ya que existe una transpilación previa.

Lazy SPA:

Ahora el inyector de dependencias de Angular no necesita que estén en memoria todas las clases o código de todos los elementos que conforman una aplicación. En resumen, ahora con Lazy SPA el framework puede funcionar sin conocer todo el código de la aplicación, ofreciendo la posibilidad de cargar más adelante aquellas piezas que no necesitan todavía.

Renderizado Universal:

Angular nació para hacer web y renderizar en HTML en el navegador, pero ahora el renderizado universal nos permite que no solo se pueda renderizar una vista a HTML. Gracias a esto, alguien podría programar una aplicación y que el renderizado se haga, por ejemplo, en otro lenguaje nativo para un dispositivo dado.

Otra cosa que permite el renderizado universal es que se use el motor de renderizado de Angular del lado del servidor. Es una de las novedades más interesantes, ya que ahora podrás usar el framework para renderizar vistas del lado del servidor, permitiendo un

mejor potencial de posicionamiento en buscadores de los contenidos de una aplicación. Esta misma novedad también permite reducir el impacto de la primera visita, ya que podrás tener vistas "precocinadas" en el servidor, que puedes enviar directamente al cliente.

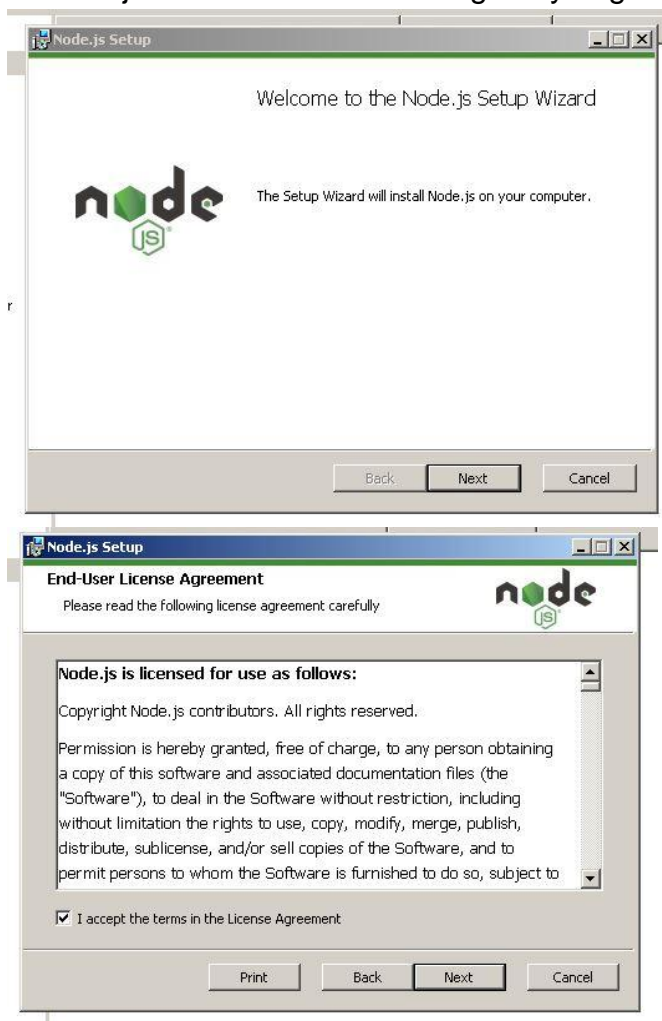
Data Binding Flow:

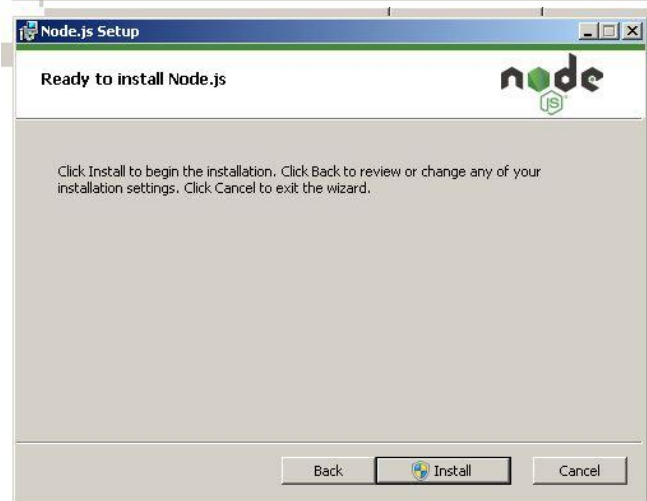
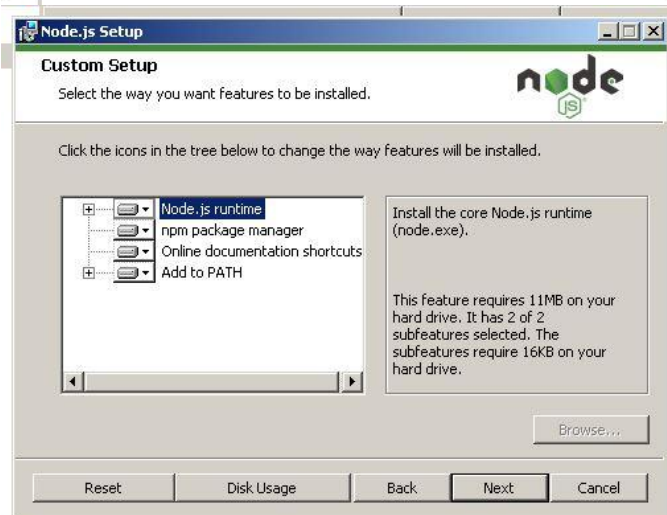
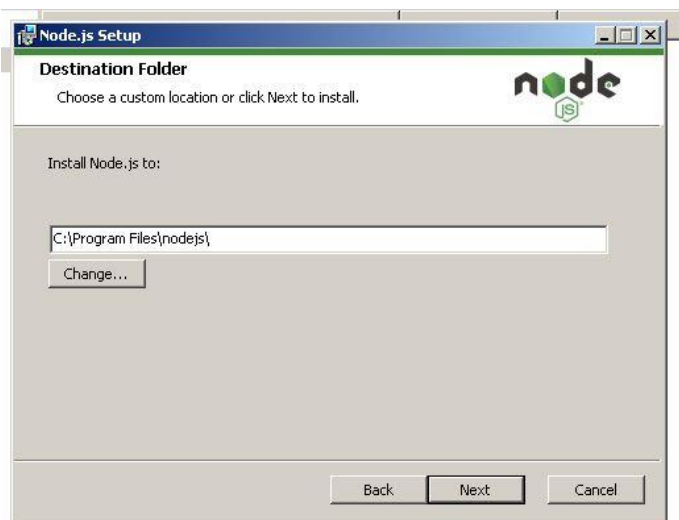
Uno de los motivos del éxito de Angular 1 fue el data binding, pero éste tenía un coste en tiempo de procesamiento en el navegador, que si bien no penalizaba el rendimiento en todas las aplicaciones sí era un problema en aquellas más complejas. El flujo de datos ahora está mucho más controlado y el desarrollador puede direccionarlo fácilmente, permitiendo optimizar las aplicaciones. El resultado es que en Angular 2 las aplicaciones pueden llegar a ser hasta 5 veces más rápidas.

Continuaremos aprendiendo mas sobre Angular 2 a lo largo del curso

Instalar Node JS

1. Ingresar a: <https://nodejs.org/en/>
2. Descargar la última versión de node Js
3. Ejecutar el archivo descargado y seguir los siguientes pasos:





Instalar editor de typescript

Necesitaremos un editor de código que nos coloree typescript para facilitarnos el trabajo. En realidad podríamos editar el código con cualquier editor de texto plano, pero personalmente recomiendo utilizar Visual Studio Code de Microsoft que es multiplataforma y podemos descargarlo desde el siguiente enlace: <https://code.visualstudio.com/>

¿Qué es typescript?

Lenguaje de programación libre y de código abierto desarrollado y mantenido por Microsoft. Es un superconjunto de JavaScript, que esencialmente añade tipado estático y objetos basados en clases.

En concreto TypeScript nos ofrece muchas de las utilidades que se necesitan en JavaScript para poder convertirlo en un lenguaje escalable, a la altura de las necesidades más exigentes. TypeScript nos ofrece muchas de las cosas que los desarrolladores de lenguajes más tradicionales vienen usando en su día a día.

Una de las diferencias fundamentales es que TypeScript es verdaderamente orientado a objetos, trayendo herramientas como la herencia, sobrecarga, etc. En resumen tiene cosas que suenan a lenguajes como Java, C++, C#, etc. Otro ejemplo clave es el del tipado estático. En el caso de TypeScript este tipado estático es opcional, pero obviamente su uso es muy recomendado y es una de las principales utilidades que nos van a facilitar mucho el trabajo y la depuración de los programas.

Instalar IONIC

Para instalar IONIC debemos haber instalado previamente node.js

1. Abrir la consola (ejecutar->cmd en Windows)
2. Ejecutar el siguiente código

```
$ npm install -g ionic cordova
```

Crear una aplicación con IONIC

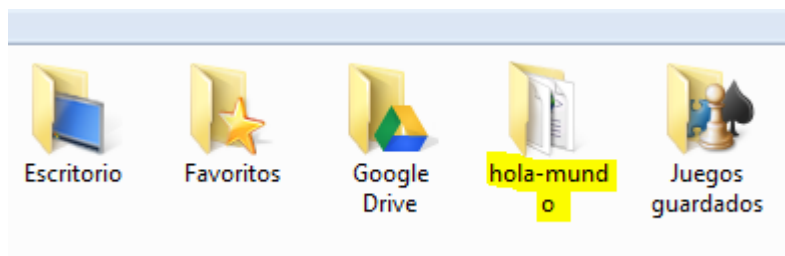
Una vez que tenemos instalado IONIC debemos:

1. Abrir la consola (ejecutar->cmd en Windows)
2. Ejecutar el siguiente comando (debemos ubicarnos en el directorio en el cual queremos que se cree la aplicación utilizando el comando cd):

```
ionic start hola-mundo blank --v2
```

- a. Ionic start: Es un comando propio de IONIC que nos indica que vamos a crear una nueva aplicación.
- b. Hola-mundo: Es el nombre que le vamos a dar a nuestra aplicación.
- c. Blank: indica que vamos a crear una aplicación desde "0"
 - i. También podemos utilizar:
 1. sidemenu: Crea una plantilla con menú lateral.
 2. tabs: Crea una plantilla con Tabs (Pestañas).
- d. -- v2: Indica que vamos a utilizar la versión 2 de IONIC.

3. Si vemos el directorio en el cual ejecutamos el comando, se habrá creada una carpeta con el nombre de la aplicación:



Ejecutar la aplicación en el navegador

Una vez creada nuestra aplicación, podemos ejecutar la misma en nuestro navegador (PC) utilizando IONIC SERVE

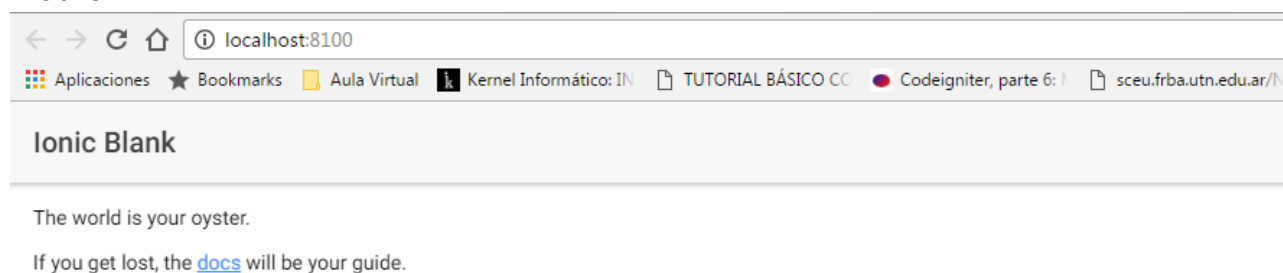
1. Abrir la consola (ejecutar->cmd en Windows)
2. Ubicarnos **dentro** del directorio en el cual se creó la aplicación.

```
C:\Users\gilca\hola-mundo>
C:\Users\gilca\hola-mundo>
```

3. Ejecutar

```
C:\Users\gilca\hola-mundo>ionic serve
```

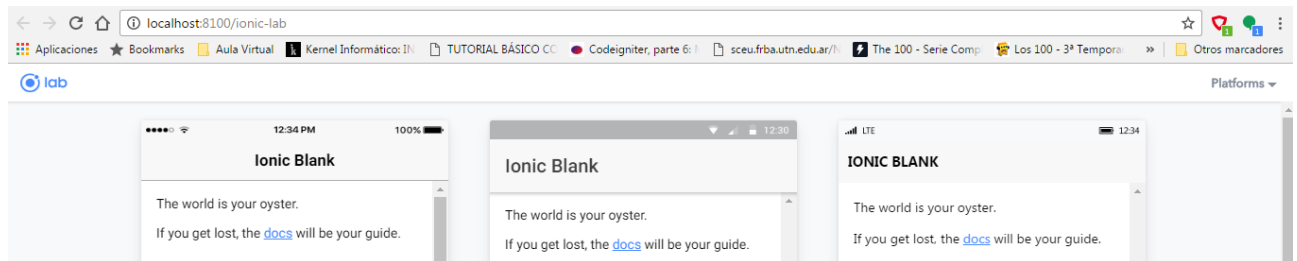
4. Se abrirá una nueva pestaña en el navegador con el contenido de nuestra app mobile.



a. Podemos ejecutar:

```
C:\Users\gilca\hola-mundo>ionic serve --l
```

De esta manera podremos ver cómo queda nuestra app para las distintas plataformas.



Ejercicio

Crear 2 aplicaciones mas utilizando **ionic start** y sus variantes sidemenu y tabs



UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

**Centro de
e-Learning**



Bibliografía utilizada y sugerida

[https://en.wikipedia.org/wiki/Ionic_\(mobile_app_framework\)](https://en.wikipedia.org/wiki/Ionic_(mobile_app_framework))

<http://ionicframework.com/getting-started/>

<https://nodejs.org/en/>

Centro de e-Learning SCEU UTN - BA.

Medrano 951 2do piso (1179) // Tel. +54 11 4867 7589 / Fax +54 11 4032 0148

www.sceu.frba.utn.edu.ar/e-learning

Lo que vimos:

En esta unidad hemos comprendido los pasos necesarios para instalar IONIC 2 framework.

Por otro lado hemos aprendido acerca de Angular 2 y typescript, y pudimos ejecutar nuestra primer aplicación.



Lo que viene:

Vamos a continuar trabajando sobre nuestra primera aplicación Hola Mundo, logrando comprender la estructura de directorios de IONIC 2 y los componentes más importantes del mismo.

