

Desarrollo de aplicaciones para móviles

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

Modulo 2: Desarrollo de aplicaciones

Storage



Presentación:

En esta unidad aprenderemos como utilizar las herramientas de persistencia propuestas por IONIC.



Objetivos:

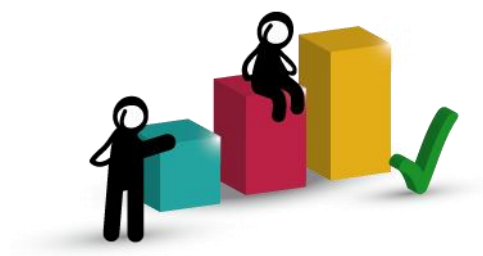
Que los participantes*:

- Aprendan herramientas básicas para la persistencia de datos.
- Apliquen herramientas de persistencia a ejemplos concretos.



Bloques temáticos*:

- Storage
- Como utilizar este modulo
- Ready
- Set
- Get
- Remove
- Clear
- Length
- forEach



Consignas para el aprendizaje colaborativo

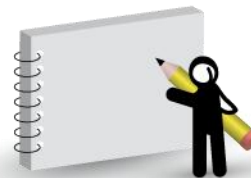
En esta Unidad los participantes se encontrarán con diferentes tipos de actividades que, en el marco de los fundamentos del MEC*, los referenciarán a tres comunidades de aprendizaje, que pondremos en funcionamiento en esta instancia de formación, a los efectos de aprovecharlas pedagógicamente:

- 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.



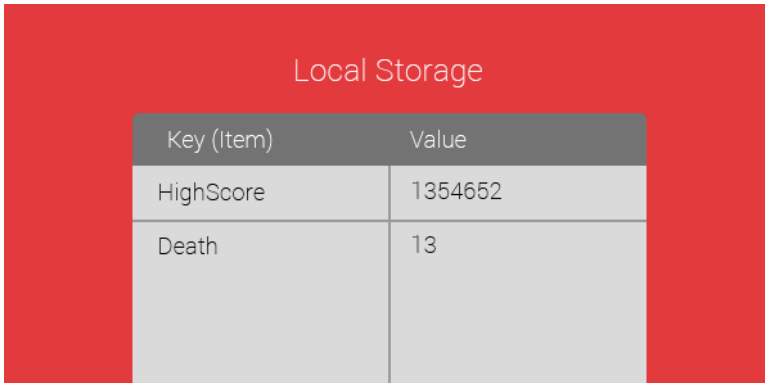
Tomen nota*

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.

Storage

IONIC 2 nos provee un modulo para el uso de almacenamiento local en nuestra aplicación. El mismo funciona a traves del par clave / valor:



Key (Item)	Value
HighScore	1354652
Death	13

Este modulo utiliza el mecanismo de almacenamiento más conveniente según la aplicación, abstrayéndonos de su compleja implementación a algo tan simple como el manejo del par clave / valor.

Cuando en la aplicación esté disponible se prioriza el uso de sql lite.



Como utilizar este modulo

1. Instalar el plugin de sql cordova lite

```
cordova plugin add cordova-sqlite-storage --save
```

2. Instalar el paquete de IONIC storage

```
npm install --save @ionic/storage
```

3. Agregar el modulo instalado en **src/app.module.ts**

```
import { IonicStorageModule } from '@ionic/storage';
```

```
imports: [
  BrowserModule,
  IonicModule.forRoot(MyApp),
  IonicStorageModule.forRoot()
],
```

```
providers: [
  StatusBar,
  SplashScreen,
  Storage,
  {provide: ErrorHandler, useClass: IonicErrorHandler}
]
```

4. Finalmente inyectar el modulo en la pagina que lo queramos utilizar, por ejemplo en login.ts

```
import { Storage } from '@ionic/storage';
```

```
constructor(public navCtrl: NavController, public navParams: NavParams, storage: Storage) {
  //this.local = new Storage(LocalStorage);
  storage.set("prueba", "prueba1");
  console.log(storage.get("prueba"));
}
```



Operaciones sobre el storage

Ready

Es una promesa (dentro del ready ejecutamos el código deseado para trabajar sobre el storage), y nos indica que el storage esta listo para utilizarse.

```
storage.ready().then(() => {  
  
  // set a key/value  
  storage.set('Nombre', 'Leandro');  
  
  // Or to get a key/value pair  
  storage.get('nombre').then((val) => {  
    console.log('Your age is', val);  
  })  
});
```

Dentro del then del ready ejecutamos el código para hacer un set y get.

Set

Setea el valor para la key dada como parámetro.

```
// set a key/value  
storage.set('Nombre', 'Leandro');
```

En este ejemplo establecerá el valor “Leandro” para la key “Nombre”

Get

Dado una key nos devolverá una promesa con la cual accedemos al valor de la misma.

```
// Or to get a key/value pair  
storage.get('nombre').then((val) => {  
  console.log('Tu nombre es', val);  
})
```

En este caso nos retorna la promesa con la cual accedemos al contenido de la key nombre.

Remove

Recibe la key a eliminar y retorna una promesa la cual nos informara acerca de la acción de eliminar.

```
storage.remove("nombre");
```

Clear

Elimina por completo todo el contenido almacenado en el storage.

Length

Retorna una promesa que nos devuelve la cantidad de keys almacenadas en el storage.

```
storage.length().then((val) => {  
  console.log('Hay tantas claves', val);  
});
```

Keys

Retorna una promesa que nos devuelve las keys almacenadas en el storage.

```
storage.keys().then((val) => {  
  console.log('Hay estas claves', val);  
});
```

forEach

Nos permite iterar sobre un valor del storage, debemos pasarle como parámetro el iteratorCallback

```
storage.forEach( (value, key) => {  
  console.log("Este es el valor", value)  
  console.log("Esta es la clave", key)  
})
```

Ejercicios

Continuando con nuestro ejercicio del colegio Manuel Belgrano, ahora nos piden que el mismo tenga una funcionalidad básica el registro. Por lo cual debemos hacer la pagina y formulario de registro y que dichos datos persistan en nuestro storage.

Luego debemos hacer una página de login, en la cual se compare el usuario ingresado en el cuadro de login con el usuario registrado, en caso de ser exitoso mostrar un alert por pantalla y setear en el storage la key **login=true**



Bibliografía utilizada y sugerida

<https://ionicframework.com/docs/storage/>

<https://forum.ionicframework.com/t/ionic-2-storage-get-all-stored-keys-foreach-method/71348/2>

Lo que vimos:

En esta unidad vimos los conceptos necesarios para poder persistir la información mediante el modulo storage de IONIC 2



Lo que viene:

En la próxima unidad veremos cómo compilar nuestra aplicación para Android.

