Configurant el projecte d’Ionic

El primer pas serà actualitzar la versió d’**Ionic** que tinguem instal·lada al nostre equip. Per fer això obrirem un terminal i iontroduirem el següent comandament:

$npm install @ionic/cli

Un cop hem actualitzar **Ionic** pasarem a crear un nou projecte. Això també ho farem des del terminal, on escriurem:

$ionic start ionic-storage blank --type=anglar

En uns minuts tindrem l’aplicació creada, el següent pas serà entrar a dins de la carpeta del projecte que acabem de crear. Per això introduirem el següent comandament al terminal:

$cd ionic-storage

Ara, ja dons del directori del nostre projecte, si tenim instal·lats **Visual Code**, i volem utilitzar-ho com a **IDE**, introduirem el següent comandament:

$code .

Això ens obtrirà de forma automàtica el nostre projecte al **Visual Code**, on podrem començar a introduir el nostre codi per crear l’aplicació.

# Instal·lant Native Storage a Ionic

Per a poder utilitzar l’emmagatzematge nadiu d’un dispositiu (**Native Storage**), l’hem d’afegir al nostre projecte. Per això al terminal introduirem el següent comandament;

Npm install @ionic /ionic-storage

En aquest moment ja hem instal·lat **Storage** al nostre projecte. Un cop importar hem d’indicar al projecte on tenim el nou mòdul. Per fer-ho obrirem l’arxiu **src/app/app.module.ts** i importarem el mòdul afegint la següent línia a sota del darrer **import**.

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

El següent pas serà incloure dins de l’array **imports** el mòdul. Així doncs afegirem el següent al vector.

imports: [..., IonicStorageModule.forRoot()],

# Creant un Servei

Per a poder utilitzar els mètodes d’emmagatzematge a qualsevol part de l’aplicació crearem un servei on inclourem els mètodes que podran ser cridats des de qualsevol part de l’aplicació

Per a crear un nou servei anirem al terminal i escriurem:

$ionic g service services/storage

Amb això hem creat un nou servei que utilitzarem per a emmagatzemar les dades.

El següent pas és obrir l’arxiu **src/app/services/storage.service.ts**, on crearem els nous mètodes.

El primer pas serà la importació del mètode, pel que escriurem la següent línia a sota del darrer **import**.

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

Un cop importat el mòdul ja podem crear el mètodes, però abans crearem una variable per a la classe.

private \_storage: Storage | null = null;

Amb la variable creads injectarem dins el constructor el mòdul que acabem d’importar i cridarem a un mètode que anomenarem **init( )**.

  constructor(private storage: Storage) {

    this.init();

  }

## Mètode init( )

El primer mètode que crearem és el mètode **init( )**, on es crearà la base de dades que utilitzarem per emmagatzemar les dades.

  async init(){

    const storage = await this.storage.create();

    this.\_storage = storage;

  }

Utilitzant l’emmagatzmatge a l’aplicació

Per a poder utiltizar l’emmagatzematge s’ha d’importar el mòdul al nostre component i injectar-ho al constructor. Per això obrirem l’arxiu **src/app/app.component.ts** i farem la importació.

import { NativeStorage } from '@ionic-native/native-storage';

Un cop importat, l’injectem al constructor.

constructor(nativeStorage: NativeStorageOriginal) {}

# Accedint als mètodes d’emmagatzematge nadiu

Ara que hem realitzar les importacions del mòdul, podem accedir de forma ràpida als mètodes per a crear, actuatlitzar, i esborrar dades al emmagatzematge local del nostre terminal.

## Emmagatzemant valors

Començarem emmagatzemant dades de forma local al dispositiiu. Per fer-ho crearem el mètode **setItem( )**. Aquest mètode l’inclourem dins de l’arxiu **app.component.ts** on hem importat el mòdul.

Aquest mètode l’insertarem sota del constructor i serà de la següent manera.