

Appinya

Manual de Instal·lació Entorn

Daniel Horta Vidal

Grau d'Enginyeria Informàtica

05.666 - TFG-Desenv. aplic. disposit. mòbils (HTML5 o Windows Phone)

Carles Sànchez Rosa

Professor responsable de l'assignatura



Aquesta obra està subjecta a una llicència de
[Reconeixement-NoComercial 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc/3.0/es/)

INDEX

Guia de Instal·lació.....	5
Prerequisits.....	5
Instal·lar entorn.....	5
Instal·lar aplicació.....	8
Estructura del projecte.....	9
Entorn de Proves l'aplicació.....	10
Distribució.....	13

Guia de Instal·lació

Al ser un projecte Open Source, l'objectiu és realitzar un entorn de instal·lació fàcil executar i depurar l'aplicació. Per això definim aquest document com una guia d'instal·lació de l'entorn de l'aplicació. Explicuem com instal·lar l'entorn de desenvolupament i com crear, distribuir i provar.

Prerequisites

Per instal·lar es requereixen les següents aplicacions instal·lades en el vostre PC:

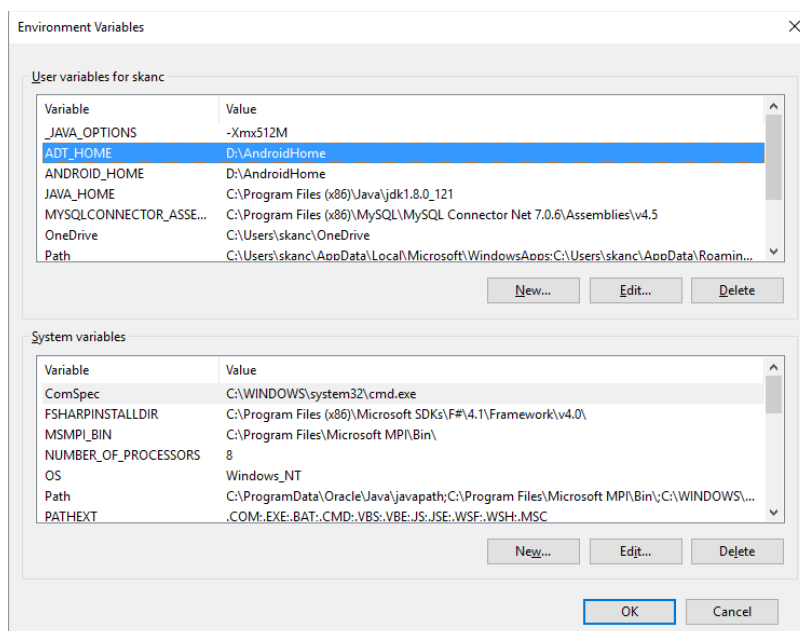
1. Visual Studio 2017
2. NodeJS 7.0
3. Sql Server Express 2017
4. Jdk 1.8
5. Gradle 3.5
6. Android Studio 2.31
7. Apache Ant

NOTA: Aquesta instal·lació es exclusiva per sistemes operatius Windows 10.

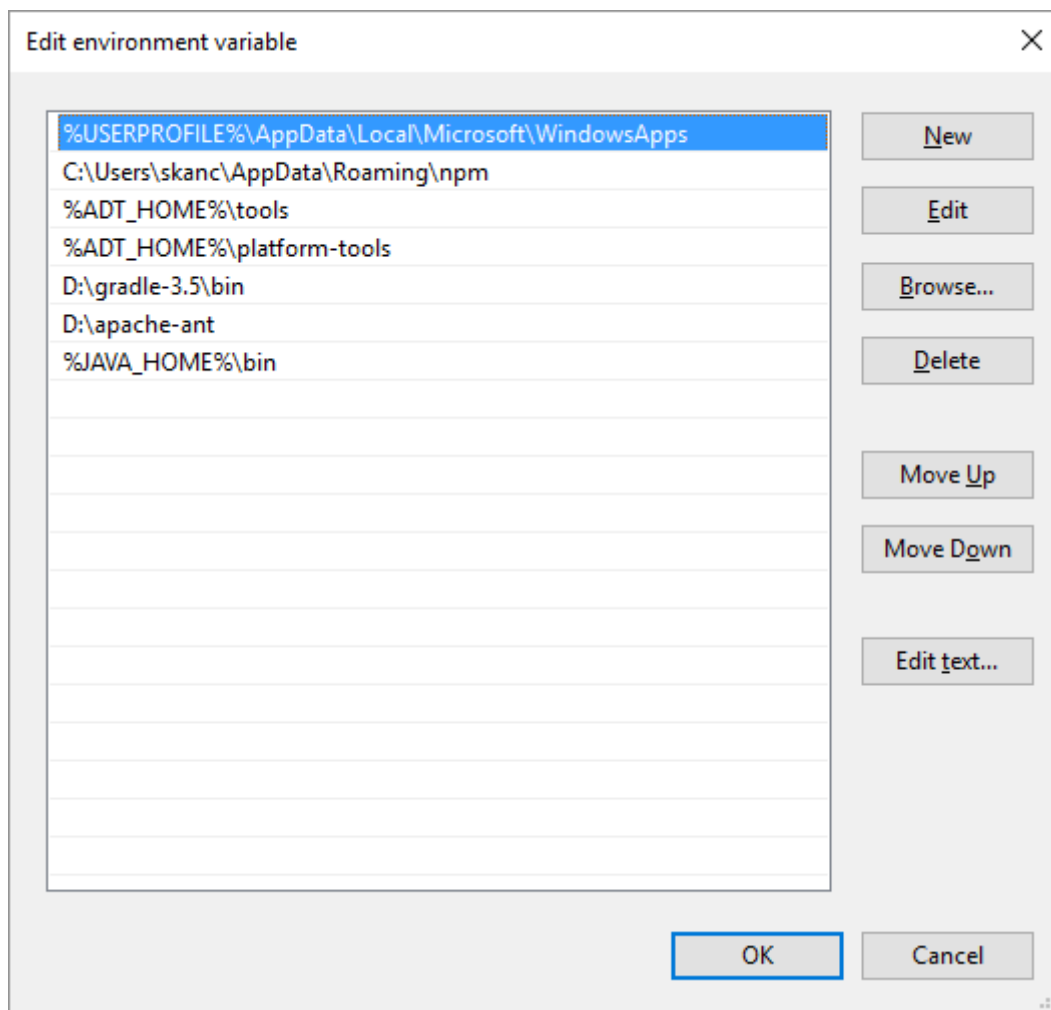
Instal·lar entorn

Seguir els següents passos :

- 1) Instal·lar les variables d'entorn de JAVA_HOME, ANDROID_HOME, _JAVA_OPTIONS i agregar al PATH



Variable PATH



2) Anar al Visual Studio , Tools> Extensions and updates.Afegir les següents extensions i executar aquesta sentència(NPM Task , TypeScript Editor , WebPack Task)

3)Anar a la línia de comandes i descarregar els següents paquets:

```
npm bower -g install
```

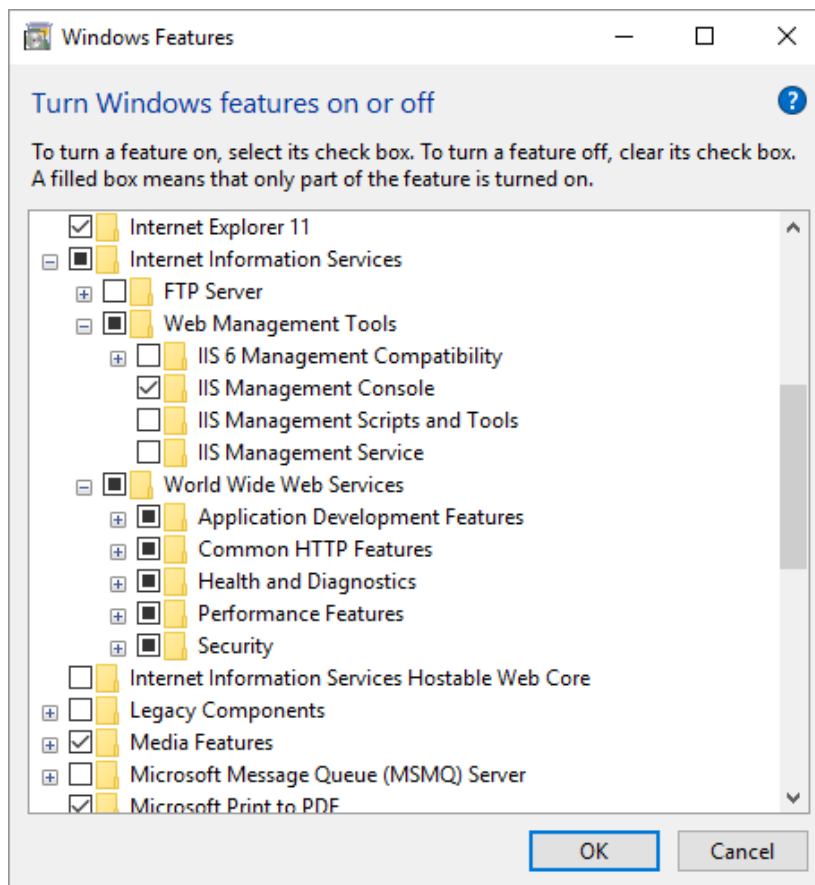
```
npm angular-cli -g install
```

```
npm apache -g install
```

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

```
npm install -g npm-cli
```

4) Anar a Control Panel\Programs\Programs and Features i seleccionar Turn Windows features on or off e instal·lar Internet Information Server On (Nota: Activar Development> ASP 4.5)



5) Anar a Visual Studio a Options>Project and Solutions> Web Package Management > External Tools i posicionar la variable (\$PATH) a la segona posició. (Nota: Reiniciar Visual Studio)

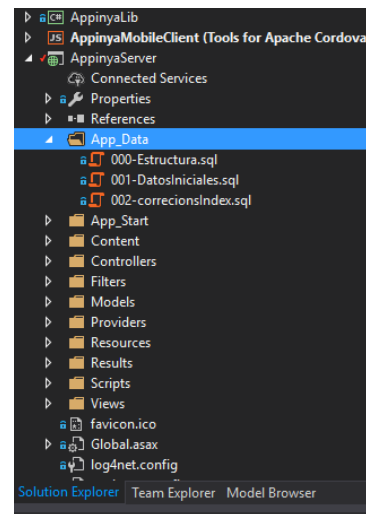
Instal·lar aplicació

1) Executar el Script d'inicialització de la Base de Dades projecte de la solució AppinyaServer\App_data*.sql

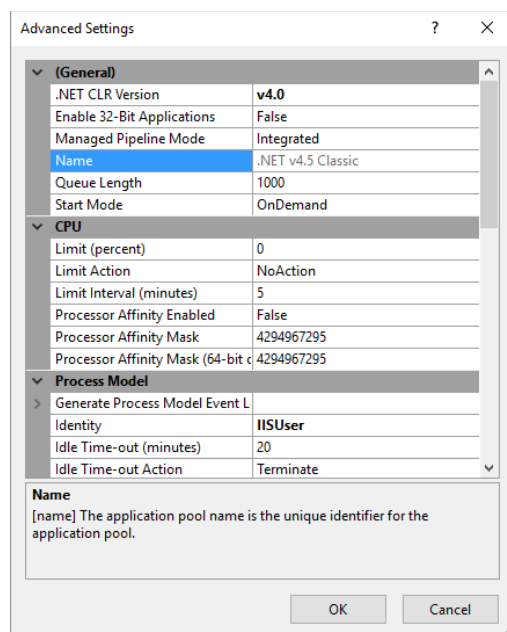
2) Anar a la Solució > Propiedades > Web , seleccionar Local IIS i prémer "Crear Directorio"(Nota: Run VS as administrator)

5) Anar al IIS , activar l' Application Pool de ASP 4.5 a la aplicació Appinya

6) Crear un usuari per accedir al SQLServer sense contrasenya. L'usuari IISUser i donar-li permisos per accedir a fitxers en Administrative Tools



al



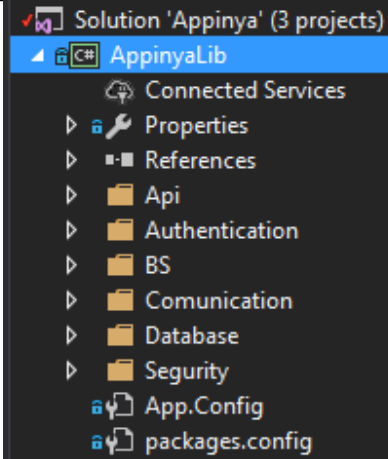
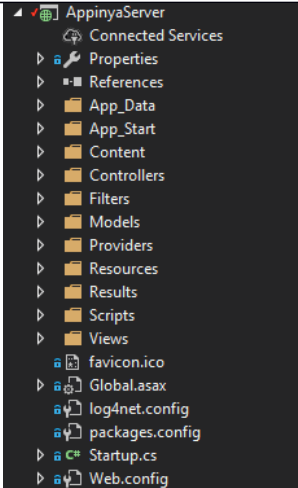
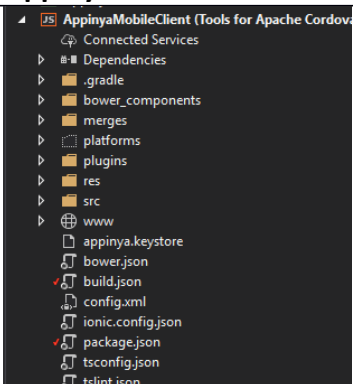
7) Assignar a l'usuari IISUser al ApplicationPool del IIS (reiniciar)

8) Anar al Sql Manager Studio i crear una credencial a Security > Users amb l'usuari IISUser a la base de dades dbappinya (Nota: Recordar donar permisos a l'usuari a la Base de Dades)

9) Restaura paquets Nodejs de la solució (Botó dret sobre el package.json)

Estructura del projecte

L'estructura del projecte, i. E. solució, és la següent. Hi ha 3 projectes que componen la solució:

AppinyaLib 	Llibreries DLL API > Entitats dels serveis REST i funcions utilitats Authentication> Classes associades a l'autenticació BS > Classes associades al negoci de l'aplicació Communication> Classes per enviar correus Database > Entitats de l'Entity Framework Segurity> Funcions de seguretat
AppinyaServer 	Aplicació Web Controllers > REST Services Filters>Filtres HTTP pe l'aplicació Models > Classes exclusives dels serveis REST Communication> Classes per enviar correus Resources > Plantilles HTML per enviar correus
AppinyaMobileClient 	Aplicació mòbil HTML5 Src> Codi font de l'aplicació client Res> Carpeta amb recursos per exportar a dispositius Plugins>Plugins Cordova utilitzats www> Resultat del desplegament del aplicació.

Entorn de Proves l'aplicació

Es defineixen 3 modes d'execució diferents per a poder executar l'aplicació i fer les diferents proves de la forma més àgil possible:

- **Mode Disseny.** Aquest mode es per dissenyar les vistes i vents de l'aplicació. S'han creat mocks dels serveis per poder oferir agilitat en els temps de reconstrucció i execució. Aquest mode es ideal pel disseny de pantalles, depuració i proves en local
- **Mode Depuració.** Aquest mode és per desenvolupar en un entorn similar a un dispositiu Android. No és tan àgil com el mode disseny però es molt útil per visualitzar els fluxos de l'aplicació amb una integració total amb la capa BackEnd i testejar la integració total del aplicació amb components i serveis.
- **Mode Simulador:** Aquestes proves són les que més s'apropen a un dispositiu real. Es realitzen sobre un emulador d'un dispositiu Android, i ofereix accés a totes les APIs d'Android.

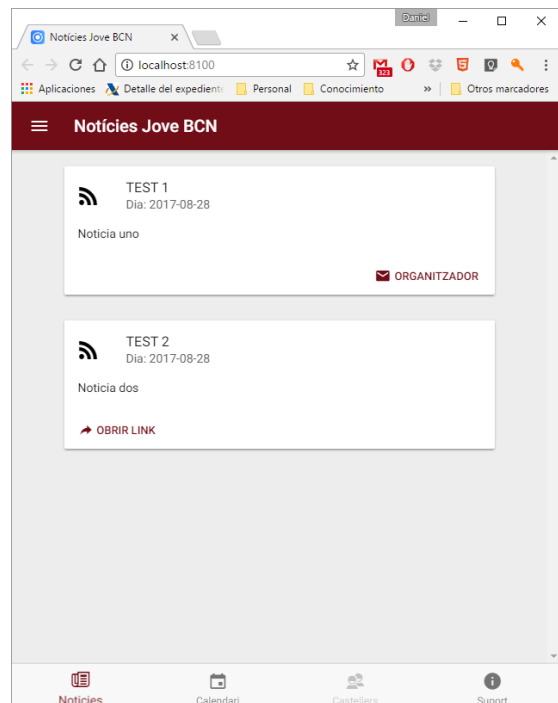
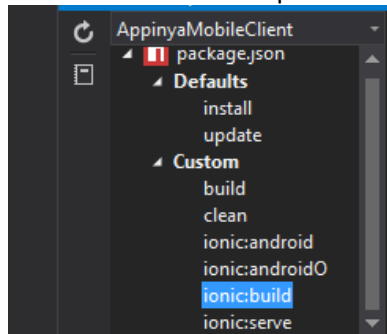
Mode Disseny.

Per activar el mode disseny s'han de realitzar les següents accions :

- 1) Anar a la classe RestService del projecte AppinyaMobileClient>src>providers. Activar la variable MockEnabled a true. Aquesta variable activa els serveis en mode Mock.

```
ppinyaMobileClient (tsconfig project) - RestService
1  import { Injectable } from '@angular/core';
2  import { Headers, RequestOptions } from '@angular/http';
3  import { Observable } from 'rxjs/Rx';
4
5  import { StoreData } from './storage.data';
6  import { AlertController } from 'ionic-angular';
7
8  /**
9   * Abstraccion del servicio para gestionar metodo comunmente llamados
10  */
11  @Injectable()
12  export class RestService {
13    rootPath: string = 'http://82.223.24.241'; // 'http://localhost/AppinyaServer';
14    mockEnabled: Boolean = false;
15    constructor(public store: StoreData, public alertCtrl: AlertController) {
16      if (this.mockEnabled) console.log("RestService - Modo Mock Activado");
17    }
18
19    /**
20     * Cabecera Standart de la peticion HTTP
21     */
```

- 2) Anar al Task Runner Explorer de Visual Studio, i prémer ionic:serve.



Avantatges d'aquest mode.

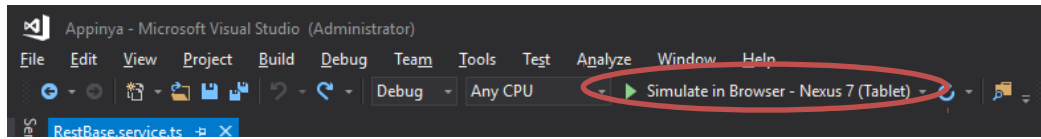
- Refresc automàtic dels canvis
- Execucions molt ràpides (<5 segons per execució)
- Es pot depurar codi

Inconvenients d'aquest mode

- No es pot connectar als serveis del BackEnd
- No té funcionalitats de dispositiu

Mode Depuració

El mode depuració utilitza un navegador Cordova que integra gran part de les funcionalitats d'un dispositiu Android.



Avantatges d'aquest mode.

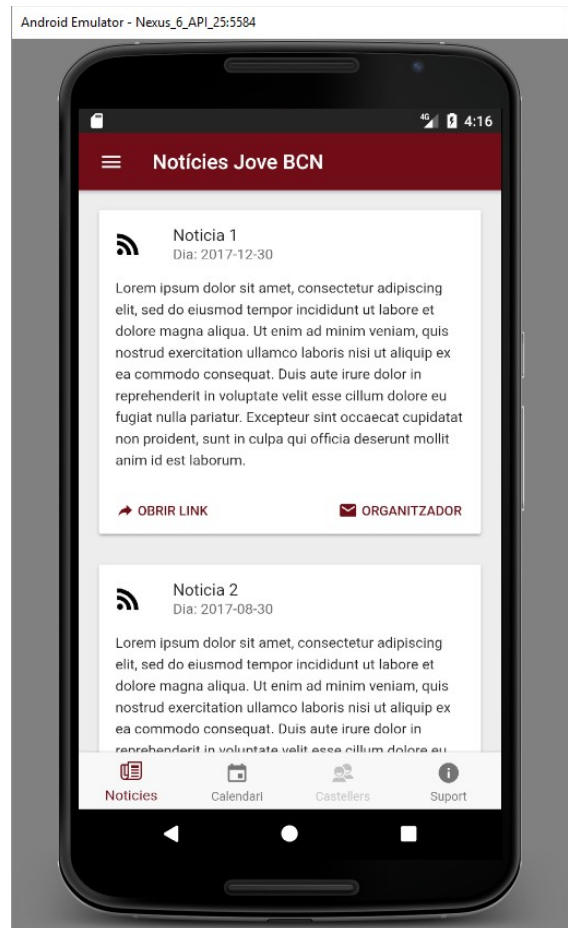
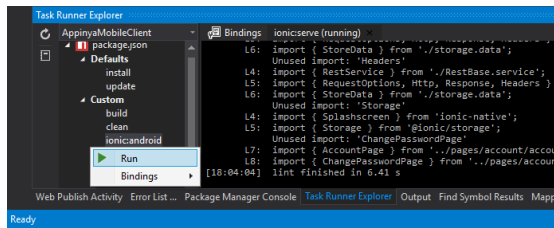
- Les execucions són ràpides (<45 segons per execució)
- Permet connectar-se al serveis del BackEnd
- Té un grup de funcions de simulació de dispositiu
- Es pot depurar codi

Inconvenients d'aquest mode

- No hi ha refresc automàtic dels canvis

Mode Simulador

El mode depuració utilitza un simulador Android. Aquesta és la fase final de proves, en la que el producte ha d'estar molt madur per ser executat. S'executa amb el Task Manager ionic:android



Avantatges d'aquest mode.

- Es pot connectar al serveis del Backend
- Accés complet a totes les funcionalitats del dispositiu

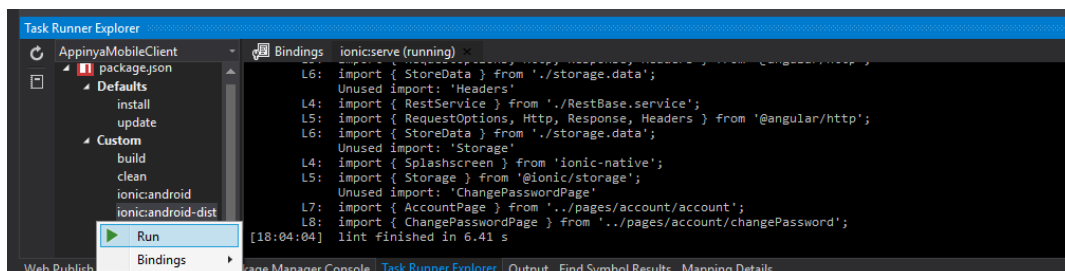
Inconvenients d'aquest mode

- Les execucions són lentes (+150 segons per execució)
- No hi ha refresc automàtic dels canvis
- No es pot depurar.

Distribució

Per la distribució del aplicació és important seguir les pautes pròpies de IONIC.

<http://ionicframework.com/docs/v1/guide/publishing.html>. Però s'han automatitzat les tasques de generació per ser més àgils. S'ha d'executar el Task Manager ionic:android-dist.



Nota: Mirar el build.json del projecte per configurar la automatització de la creació de la APK.