Universidad autónoma del estado de México

Centro Universitario UAEM Zumpango

Ingeniería en Computación

Programación Avanzada

Proyecto Ordinario

Alumnos:

* Joel David González Del Ángel
* Esteban Lira Alanís
* Guillermo López Rubio
* Juan Manuel Manjarrez Palacios
* Cristóbal Simón Ramírez Flores
* José Daniel Orio Islas

Profesor: Asdrúbal López Chau

09-Diciembre-2018

**Aplicación para Administradores**

Esta es una aplicación que permitirá al usuario (De preferencia un profesor o experto) dar de alta un curso para los alumnos de ingeniería, en la aplicación se es posible ver los cursos ya existentes si como sus detalles y se pueden editar o eliminar los cursos si es que se desea.

Para esta aplicación se utilizó la herramienta firebase para poder guardar los datos de los cursos en lo que es una base de datos no relacional en la nube, esta herramienta permite guardar y consultar los datos guardados.

**Aplicación para Asistentes**

En esta aplicación el usuario (un alumno o persona que dese asistir a los cursos dados de alta en la aplicación anterior) podrá visualizar los cursos que están disponibles ver sus detalles e inscribirse a los que más le agraden así como ver los cursos en los que está inscrito y darse de baja para no asistir (si así lo desea).

En esta aplicación se utilizó la misma herramienta de firebase para guardar y acceder a los datos de los usuarios.

**Sobre las dos aplicaciones**

Como ya se mencionó anteriormente en las dos aplicaciones se utilizó la herramienta de firebase y la navegación entre diferentes vistas para realizar sus diversas tareas.

**Compilación de las aplicaciones**

Para compilar las aplicaciones es necesario el uso de la terminal (símbolos del sistema), una vez en la terminal se accede a la dirección de la carpeta del proyecto por ejemplo suponiendo que la carpeta del proyecto se llama cursos y se encuentra en el escritorio se ejecutarían los siguientes comandos, *CD Escritorio*, *CD cursos*, una vez ubicados en la carpeta se ejecuta el comando *ionic serve* que empezará a ejecutar el proyecto en el navegador.

**Agregación de Plugins**

Para que la aplicación funcione es necesario la agregación de algunos plugins o complementos que permitirán acceder a algunas funciones que serán de mucha ayuda. Uno de estos es el que nos permite realizar la conexión entre la aplicación y la base de datos en firebase, también se necesitara otro para que nos permita leer los registros o mejor dicho mostrar los registros guardados.

Para la instalación de los plugins es necesario al igual que la compilación realizarlo desde la terminal y en la ruta del proyecto para la instalación del primer plugin se necesita introducir el siguiente comando *npm install firebase angularfire2 –sabe* este comando instala los componentes para realizar la conexión con firebase***.*** El siguiente plugin se instala con el comando *npm i rxjs@^6.0 rxjs-compat* este comando instala los componentes necesarios para facilitar la obtención y el listado de los registros

**Creación del APK**

Para la creación del APK es necesario tener el SDK del Android que tenga el teléfono o en otras palabras el SDK de la versión de Android para la que se lanzara la aplicación una vez con el SDK se prosigue a ejecutar el siguiente comando *ionic cordova build* en la terminal en el la ruta del proyecto como lo hemos estado haciendo damos enter y después escribimos Android con eso si se tiene el SDK correcto se abra creado el APK correctamente.

**Firma del APK**

La firma del APK es algo requerido para poder subir la aplicación a Google play para esto es necesario la creación de un Keystore que es un campo binario que contiene una o más claves privadas, este campo se genera desde la consola y como en los casos anteriores nos situamos en la carpeta de nuestro proyecto suponiendo que la aplicación se llama cursos se escribe el siguiente comando *keytool –genkey –v –keystore cursos.keystore –alias cursos –keyalg RSA –keysize 2048 –validity 10* en este comando, cursos.keystore es el archivo que se generara el alias es un nombre de identificación para la clave, con keysize se indica el tamaño de la clave (es preferible que sea 2048), y con validity se fija el periodo de valides de la clave en años.

Una vez ejecutado el comando se pedirán una serie de datos que hay que llenar después mostrara los datos que se han ingresado para confirmar se selecciona sí y con esto queda generado el archivo .keystore en la raíz de nuestro proyecto.

Estando el archivo .keystore generado se prosigue a generar un archivo en la raíz del proyecto llamado build.json donde se escribe el siguiente código

{

“android”:{

“debug”: {

“keystore”:

“Cursos.keystore”,

“storePassword”:”XXXXX”,

“alias”: “cursos”,

“password”: “XXXX”,

“keystoreType”: “”

},

“reléase”: {

“keystore”:

“Cursos.keystore”,

“storePassword”:”XXXXX”,

“alias”: “cursos”,

“password”: “XXXX”,

“keystoreType”: “”

}

}

}

En el campo storePassword se debe poner la contraseña del almacén de claves que se indica al generar el archivo .keystore. Y password es una contraseña cualquiera.se guarda el archivo y proseguimos a generar el APK como ya se indicó anteriormente.