Documentación TFG

Alejandro Moles Hurtado

SolySur_App

IES Francisco Ayala 2021/2022

Alejandro Moles Hurtado

Índice:

Alejandro Moles Hurtado

Introducción

Para desarrollar el TFG he requerido de hacer dos aplicaciones, una aplicación para administrar la base de datos (tanto externa como interna) y otra aplicación que se encarga de realizar los fichajes de los trabajadores que usan la aplicación. Las aplicaciones se basan en la empresa de mi padre, concretamente en la parte de fichar trabajadores. Mi padre me pidió una aplicación para su empresa de construcción la cual consistiese en una gestión de los fichajes que realiza un trabajador en el día a día. La base de datos local la he creado con Sqlite y la base de datos externa la he creado con firebase que es una base de datos NOSql la cual esta alojada en la nube.

Bases de Datos

La base de datos consiste en una simple estructura que tiene 5 tablas.

USERSTABLE → Esta tabla guarda los datos de los usuarios que se dan de alta en la aplicación, y tiene como campos:

- userDni: Esta es la clave primaria de la tabla, es tipo varchar e indica el DNI del usuarios
- userCategoria: Este campo es un booleano e indica si el usuario es administrador o empleado
- userEmail: Este campo es un varchar e indica el corro electrónico del usuario
- userName: Este campo es de tipo varchar e indica el nombre del usuario
- userTrabajo: Este campo es de tipo varchar e indica la ocupación del usuario
- userContrasenia: Este campo es de tipo varchar y guarda la contraseña encriptada del usuario

OBRASTABLE → Esta tabla guarda los datos de las obras que se dan de alta en la aplicación

- obraLatitud: Este campo es de tipo number y guarda la coordenada de latitud de la posición de la obra
- obraLongitud: Este campo es de tipo number y guarda la coordenada de longitud de la posición de la obra
- obraName: Esta es la clave primaria de la tabla e indica el nombre de la obra

MOTIVOSTABLE → Esta tabla guarda los motivos que tiene un usuario por no estar en la obra a la hora de realizar el fichaje

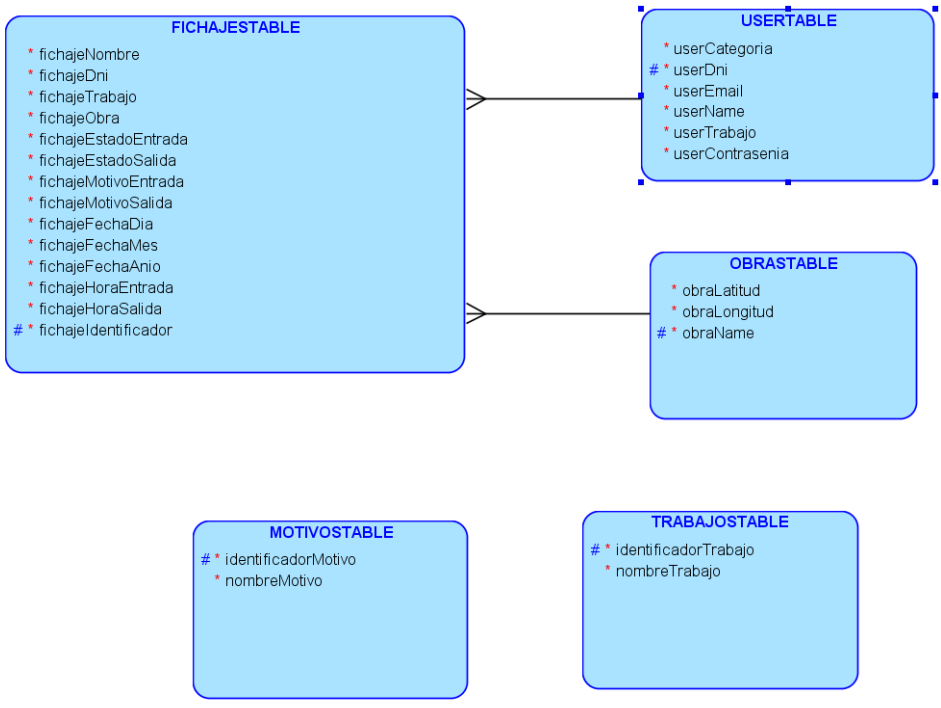
- identificadorMotivo: Esta es la clave primaria y se usa para identificar los diferentes motivos
- nombreMotivo: Este campo guarda el motivo por el cual no se esta en la obra

TRABAJOSTABLE → Esta tabla guarda los diferentes trabajos que se pueden asignar a un usuario

- identificadorTrabajo: Esta es la clave primaria y se usa para identificar los diferentes trabajos
- nombreTrabajo: Este campo guarda el trabajo que se puede asignar a los usuarios

FICHAJESTABLE → Esta tabla guarda los diferentes fichajes realizados por el trabajador

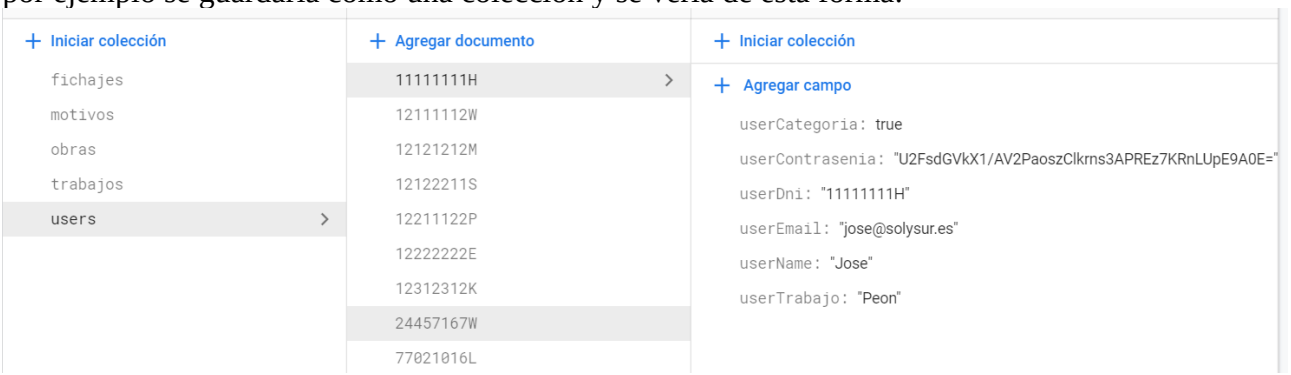
- fichajeNombre: Este campo guarda el nombre del usuario que ha realizado el fichaje
- fichajeDni : Este campo forma una clave foránea con la tabla de usuarios, y guarda el dni del usuario que ha realizado el fichaje
- fichajeTrabajo: Este campo guarda la ocupación del empleado que ha realizado el fichaje
- fichajeObra: Este campo guarda el nombre de la obra donde se ha realizado el fichaje, también forma una clave foránea
- fichajeEstadoEntrada:Este campo guarda si el trabajador estaba en la obra o no a la hora de fichar en la misma
- fichajeEstadoSalida: Este campo guarda si el trabajador estaba en la obra o no a la hora de fichar de salida en la misma
- fichajeMotivoEntrada:Este campo guarda el motivo por el cual el empleado no estaba en la obra a la hora de fichar
- fichajeMotivoSalida: Este campo guarda el motivo por el cual el empleado no estaba en la obra a la hora de fichar de salida
- fichajeFechaDia: Este campo guarda el día en el que se realizo el fichaje
- fichajeFechaMes: Este campo guarda el mes en el que se realizo el fichaje
- fichajefechaAnio: Este campo guarda el año en el que se realizo el fichaje
- fichajeHoraEntrada: Este campo guarda la hora de entrada de cuando se realizo el fichaje
- fichajeHoraSalida: Este campo guarda la hora de salida de cuando se realizo el fichaje
- fichajeIdentificador: Esta es la clave primaria de la tabla y esta compuesta por el nombre del trabajador, la fecha, y la hora de entrada



Alejandro Moles Hurtado

Firebase:

La base de datos de firebase en cambio funciona de una forma diferente, lo que en bases de datos relacionales conocemos como tablas, en firebase se llaman colecciones. Es decir la tabla usertable, por ejemplo se guardaría como una colección y se vería de esta forma:



En la columna de la izquierda se ven las colecciones que tengo, en mi caso 5 como las tablas que tengo en la base de datos. En la columna de el medio, aparecen los documentos, que serian parecidos a los registros que se guardan en la base de datos. Y en la tercera columna la de la derecha, se muestran los datos que tiene ese documento.

Para poder implementar firebase con mi aplicación primero tuve que crear un proyecto en firebase, para más tarde implementar en una variable de mi proyecto los datos para que se pueda conectar

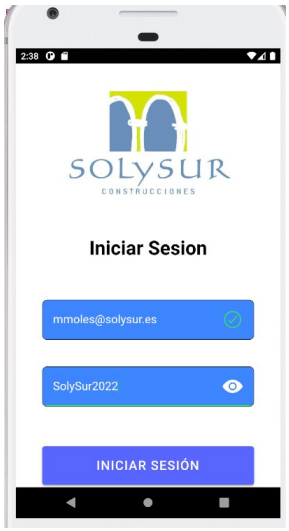
```
export const environment = {
  production : false,
   firebaseConfig : {
    apiKey: 'AIzaSyBaSwcB6tc7CSu_1JyN5w8WyYPCWSnOI2w',
    authDomain: 'solysur-ab2dd.firebaseapp.com',
    projectId: 'solysur-ab2dd',
    storageBucket: 'solysur-ab2dd.appspot.com',
    messagingSenderId: '125842925494',
    appId: '1:125842925494:web:aa4d7f6c14182f122ac317'
  }
};
```

En mi caso como se ve en la imagen de la colección, mi base de datos de firebase esta compuesta por 5 colecciones como las tablas base de datos local. Estas se llaman users (seria lo mismo a la usersTable, con los mismos datos y tipos de datos), obras (seria lo mismo a la obrasTable, con los mismos datos y tipos de datos), motivos (seria lo mismo a la motivosTable, con los mismos datos y tipos de datos), trabajos (seria lo mismo a la trabajosTable, con los mismos datos y tipos de datos), fichajes (seria lo mismo a la fichajesTable, con los mismos datos y tipos de datos)

Aplicación

Hablaré primero de la aplicación de la administración llamada solysur_administrador.
Al iniciar la app nos saldrá una pantalla de log-gin en la cual aparecerán dos inputs, uno para el email y otro para la contraseña. Gracias a un modulo de ionic llamado ReactiveForms he podido hacer que si por ejemplo no se ha escrito un email, o contraseña salgan diferentes notas debajo de los inputs indicando el error. Y no se pueda presionar el botón de login.
Para que los inputs estén correctos tienen que el email tener un formato de email y la contraseña debe tener mínimo 6 letras, con mayúsculas, minúsculas y un numero al menos.

Log-In:



Para iniciar en el login solo habrá un usuario al principio que sera con el correo **mmoles@solysur.es** y la contraseña **SolySur2022**

Solo estará este usuario a la hora de la entrega de este proyecto y solo se podrán añadir más usuarios dentro de la aplicación de administración al dar de alta a un usuario.

En el input de contraseña se hay un icono que al pulsarlo puedes ver la contraseña que has escrito.

La contraseña que se guardan en la base de datos no se pueden saber ya que están cifradas con un tipo de cifrado.

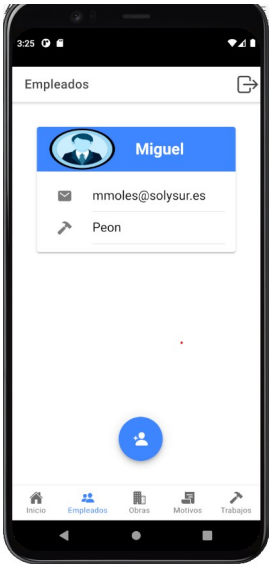
Inicio:

Una vez iniciada la sesión, si el correo y la contraseña es correcta la aplicación navegará hacia la otra ventana llamada tab1, la estructura de la aplicación funciona con tabs, es decir tendrán una barra horizontal en la parte de abajo que indicara en que parte de la aplicación estas.

En tab1 se mostrarán los datos del usuario con el que se ha logeado. La aplicación también tendrá en la barra superior derecha un icono de log-out el cual nos sacara de la sesión una vez pulsado.

Mientras no se pulse ese icono, no se nos sacara de la sesión incluso si nos salimos de la aplicación, al entrar nos logeara con email que nos logeamos anteriormente.

Empleados:

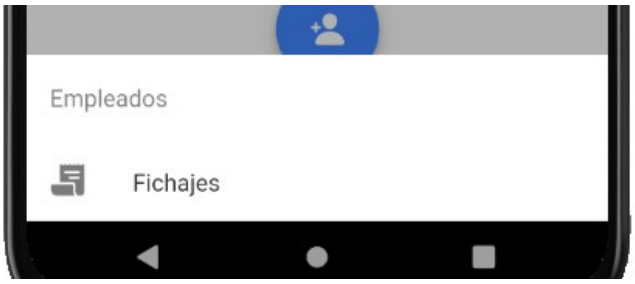


Si navegamos mediante el menú de abajo hacia la pestaña de empleados, nos aparecerá una ventana donde se nos muestra una lista de cards que contienen diferentes datos que nos interesa mostrar para los empleados que hay actualmente.

También habrá un botón que estará fijo en la parte de abajo que si se le pulsa nos enviará a otra ventana para que demos de alta a los trabajadores (añadir usuarios)

Los datos que se nos mostrarán en el card seran el correo del trabajador más la ocupación que tiene.

Si se le da click a la card, se nos mostrara un seleccionador donde tendremos la opción de ver los fichajes que tiene ese trabajador. Lo cual nos llevara a otra ventana de la aplicación.
Si el trabajador no ha hecho ningún fichaje nos lo dirá en un alert.
También aparecerá el icono de log-out.



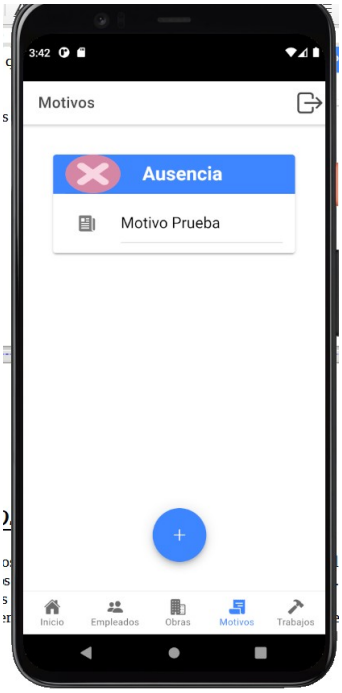
Obras:

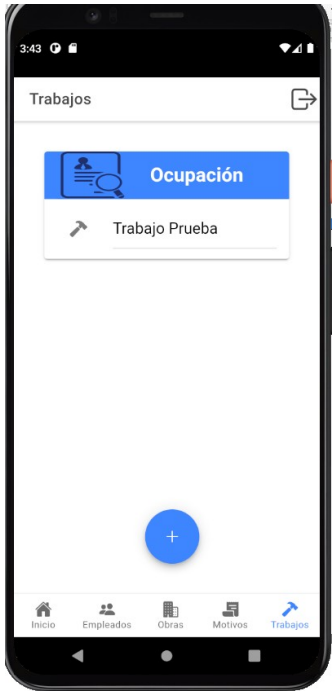
En el siguiente tab, nos encontramos el apartado de las obras, el cual será parecido al de los empleados ya que también nos mostrara diferentes cards con los datos de las obras, y un botón abajo del todo en el centro el cual nos servirá para que el administrador de de alta más obras.

En este caso si se le da al card de las obras no sucederá nada.

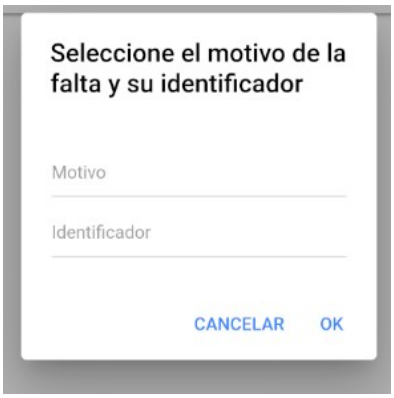
Motivos y Trabajos:

En las siguientes tabs nos muestra los diferentes motivos que un trabajador puede dar al indicar que no esta en la obra al fichar y trabajos que puedes seleccionarse al dar de alta un usuario. Se guardan en la base de datos para que sea más dinámico el proceso.
En esta ventana se muestran como en la anteriores una cards con los datos que se quieren mostrar.





En las dos pestañas se tendrá el mismo botón para añadir motivos o trabajos, el cual nos mostrara un input (en caso de añadir el motivo/trabajo) el que nos pedirá la información para añadirlo.



Si se hace click en el card, nos saldrá como en los empleados una opción para actualizarlos, en ese caso se nos volverá a aparecer un input pero solo para escribir el motivo o el trabajo no el identificador, ya que actualizaremos esos campos en la base de datos.

Añadir Empleados:

Si en la ventana de empleados se le da al boton para insertar trabajadores, se nos mostrara una ventana que estará compuesta por 6 campos que hay que “rellenar” para que el botón de Dar de Alta se active y se inserte un nuevo empleado.
Estos 6 campos están hechos con ReactiveForms y se requieren de diferentes condiciones para que estén correctos.

Nombre: Solo es el nombre, no puede haber ni espacios ni otros caracteres que no sean letras
DNI: Solo se pueden insertar números, se insertaran los primeros 8 dígitos numéricos e internamente se sacara la letra que le corresponde a ese DNI
Email: Debe tener un formato de corro electrónico.
Contraseña: Debe tener 6 caracteres, por lo menos 1 mayúscula, 1 minúscula y 1 numero
Puesto de trabajo: Debes de haber seleccionado alguno

El campo de administrador se seccionará si el usuario que se va a crear es administrador o no.

Una vez añadido el trabajador, se navegara hacia la ventana del inicio

Fichajes Empleado:

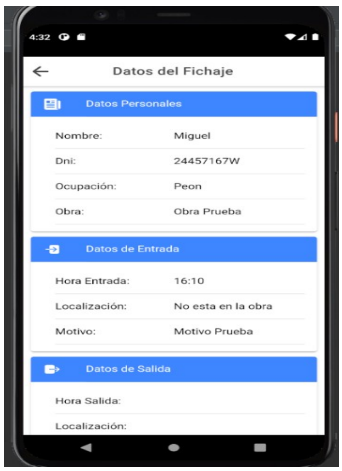
Una vez se ha ingresado a los fichajes de un empleado, a través de hacer click en fichajes, cuando sale la opcion al pulsar la card de empleados, se puede ver una lista simple de todos los fichajes que ha realizado ese empleado. Se nos mostrara con la forma de “Nombre + Fecha +Hora entrada ya. También aparecera un buscador arriba de la lista que servira para filtrar la lista de fichajes.

El buscador funciona de esta forma:

- Se tiene que poner las palabras exactas para que busque.
- Si se pone una palabra primero, luego se tiene que continuar con la que le precede, es decir si se filtra Miguel luego tienes que continuar con la Fecha, en este caso 10…. . No se pueden poner Miguel y luego la hora, ya que no aparecerá nada.
- Por así decirlo se tendrá que filtrar por lo que aparece en la lista por orden, aunque se puede filtrar por ejemplo, solo por la hora de entrada y te saldrían lo que la tienen, pero solo si unicamente pones en el filtrador la hora de entrada.

Si se le da click a cualquier fichaje de la lista nos llevara otra ventana la cual nos indicara los datos del fichaje realizado.

Datos Fichajes:



En esta ventana se muestran los datos del fichaje que previamente se ha seleccionado.
Nos lo muestra en modo de card y muestra los diferentes datos que interesan como la hora de entrada, salida, la obra, los datos del empleado etc. .

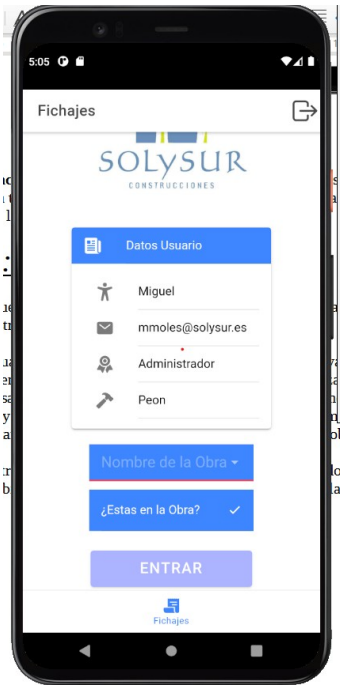
Añadir Obras:

A esta ventana se accederá si se ha pulsado el botón de añadir obras en la ventana que muestra esas mismas.
Lo que veremos en esta ventana sera un formulario el cual nos pedirá el nombre de la obra, el cual
puede tener espacios y numero, pero es requerido, también nos pedirá la latitud y longitud de localización de la obra, estos campos se podrán insertar, pero para la facilidad del usuario, hay un botón que saca la latitud y la longitud de la posición actual.

Alejandro Moles Hurtado

Para continuar explicare la aplicación de solysur_empleados_final la cual es la que se usa para realizar los fichajes. Esta aplicación tiene un login exactamente igual que la aplicación anterior, y tanto la base de datos externa como la local son la misma.

Realizar Fichajes:



Esta ventana cuenta con una card que nos muestra la información del usuario más relevante, tiene el mismo diseño que la card que muestro en la tab1 de la aplicación de administración.

Luego aparecerá un selector en el cual el trabajador seleccionara una obra(en la que vaya a fichar), después de eso seleccionara si esta en la obra o no.

Si selecciona que si, al darle a realizar fichaje, internamente tengo un método que sacara la posición del trabajador y calculara la distancia que hay desde ese punto hacia la obra.
Si hay menos de 150 metros entonces el trabajador se toma como si estuviese en la obra, si hay mas distancia saltara un mensaje diciendo que no esta en la obra

Si esta en la obra entonces nos mostrará un alert diciendo que el fichaje ha sido realizado y automáticamente la pantalla se cambiará a una en la que ya no aparezca el select, si no la obra en la que esta el fichaje todavía abierto. Y si se se le da a salir de la obra entonces nos actualizara el fichaje actualizándose la hora de salida etc. .

Si no se selecciona que estas en la obra es lo mismo pero se tendrá que seleccionar un motivo por el cual no se ha ido a la obra

Framework y Referencias

Esta aplicación esta hecha intentando solucionar un problema de tiempo en la empresa de mi padre, ya que no se tenia un total control sobre los fichajes de los trabajadores y al final se hacían los fichajes que los empleados decían de boca y a mano. Esto no era muy eficiente asi que con esta aplicación busco solucionar ese problema otorgando una mejor gestión de los fichajes y mayor rapidez a la hora de realizar los mismos.

Para realizar el proyecto he utilizado ionic que es un framework que esta basado en angular. Ionic utiliza diferentes lenguajes para sus componentes. Por ejemplo para lo que viene siendo las vistas de la aplicación, se crean con html y se da estilo con scss. Para lo que viene siendo la lógica del programa se realiza en type scrip, que es un lenguaje de programación de código abierto. Es un súper conjunto de JavaScript. Con ionic se pueden hacer tanto aplicaciones webs, como aplicaciones para android, como aplicaciones para ios.
En mi caso solo la he hecho para android ya que para hacerla para ios se necesita de un dispositivo Mac para compilar la aplicación para ios.

Para crear la aplicación he usado diferentes fuentes de información que me han ayudado a crearla:

- **Base de datos de firebase**: https://firebase.google.com/docs
- **Base de datos Local:** https://www.positronx.io/ionic-sqlite-database-crud-app-example-tutorial/
- **Libreria de math**: https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/Math
- **Imagenes de assets**: Diferentes imagenes de google
- **diferentes modulos de ionic:** https://ionicframework.com/docs