

Instituto Tecnológico y de Estudios Superiores de Occidente.



ITESO

Universidad Jesuita
de Guadalajara

Avance de proyecto 2

Materia: PROGRAMACION DE DISPOSITIVOS MOVILES

Profesor: Francisco Javier Camacho Gil

Fecha: 2 de mayo de 2021

Autor(es): Armando Emmanuel Correa Amorelli

Introducción

El objetivo de esta Aplicación es el proporcionar una plataforma en la que se concentre tanto el progreso nutricional de un atleta de equipos representativos de ITESO, como tu trabajo de gimnasio. Con la finalidad de facilitar su consulta para el deportista como para el entrenador.

Desarrollo

Ventana de inicio de sesión:

The image shows a mobile application interface for 'EQUIPOS REPRESENTATIVOS ITESO'. At the top, there is a status bar with the name 'Marvel' and the time '13:00'. Below this is a dark blue header with the text 'EQUIPOS REPRESENTATIVOS ITESO' in white. The main content area has a white background. In the center, there is a circular logo with a black border. Inside the circle, the text 'MI EQUIPO, MIS VALORES' is written in a bold, sans-serif font. Below this, in a smaller font, are the words 'TRIUNFO', 'HONESTIDAD', and 'JUSTICIA' stacked vertically. Below the logo, there are two text input fields. The first is labeled 'Expediente' and the second is labeled 'Contraseña'. Both fields have a light gray border. Below the 'Contraseña' field is a blue button with the text 'LOG IN' in white. At the bottom of the screen, there is a dark blue navigation bar with three white icons: a back arrow, a circle, and a square.

Esta ventana cuenta con dos campos de entrada de texto y un botón para llevar a cabo la verificación.

El primer campo de texto es para ingresar el expediente del alumno/entrenador. Mientras que el segundo campo es para ingresar la contraseña del mismo.

Ventanas del deportista



Rutina del Día/Ventana principal

En esta ventana se le despliega al usuario los músculos a trabajar en ese día y una imagen alusiva a estos.

Después de esto, se muestra una lista con cada ejercicio, junto con la cantidad de series y repeticiones para cada uno. Al hacer tap en el ejercicio, se despliega una descripción del mismo.

Por último en la barra de la aplicación, se muestra un icono para desplegar un drawer, que funciona como menú de navegación, y un saludo personalizado para cada usuario.



Descripción de ejercicio

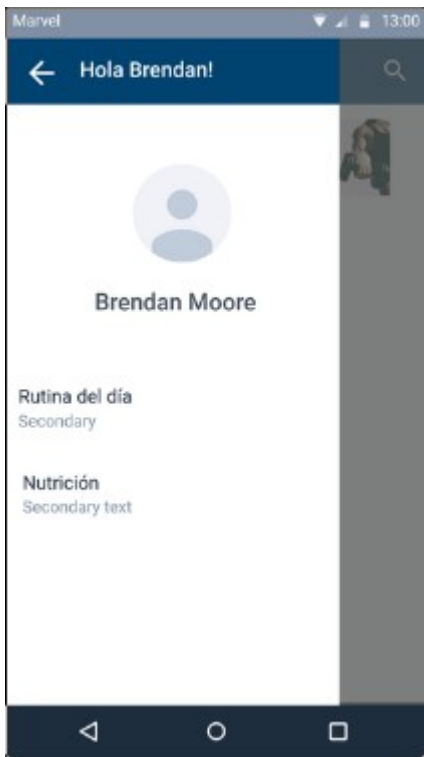
Esta es un slide card, que se despliega cuando se hace tap en algún ejercicio. En esta se muestra una imagen/video demostrativo, el nombre del ejercicio, la carga a realizar y por último una descripción del mismo.

Se puede desascer de esta al deslizarla hacia abajo.



Perfil del usuario

En esta ventana se le despliega el perfil del usuario, donde se muestra su altura, edad, deporte, posición y composición corporal.



Menú de navegación


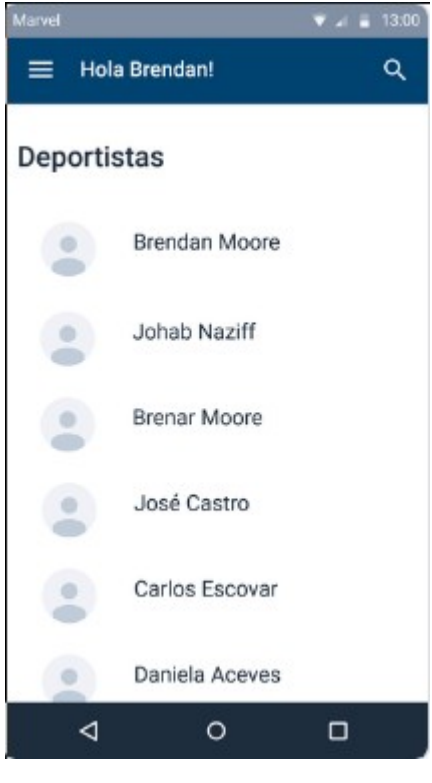
Este es un drawer que muestra la foto y nombre del usuario y le permite seleccionar entre la ventana de la rutina del día y su perfil nutricional.



Perfinl nutricional

En esta ventana se le despliega el perfil nutricional del usuario, donde se muestra su altura, peso, compocisión corporal y una grafica en la cual se mostrará la evolución del peso corporal del atleta

Ventanas del entrenador

	<p>Menú de navegación</p> <p>Este es un drawer que muestra la foto y nombre del usuario y le permite seleccionar entre la ventana de la rutina del día y la lista de deportistas bajo su cuidado.</p>
	<p>Rutina del Día/Ventana principal</p> <p>En esta ventana se le despliega al entrenador los deportistas bajo su cuidado. Al hacer tap en el atleta, se despliega una descripción del mismo donde el entrenador puede asignarle una rutina personal</p> <p>Por ultimo en la barra de la aplicación, se muestra un icono para desplegar un drawer, que funciona como menú de navegación, y una opción de búsqueda de deportista.</p>

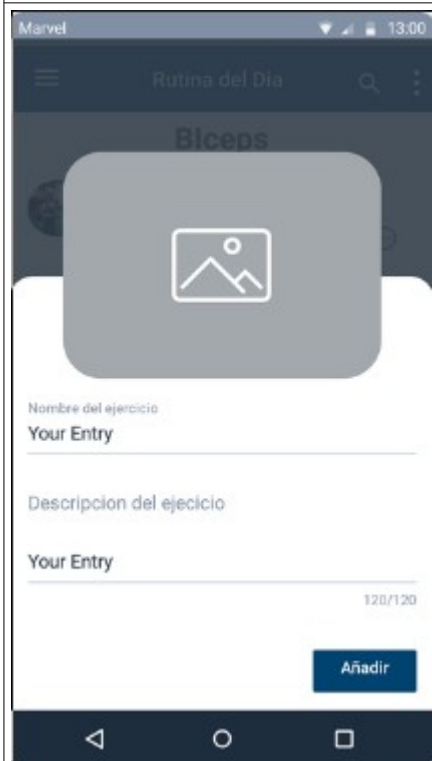


Nueva Rutina del Día

En esta ventana se le despliega al entrenador los ejercicios disponibles, en la aplicación, separados por musculos principales

Cada elemento de la lista tiene un contador de series y repeticiones, para así ajustar la carga de trabajo de la rutina

Por ultimo cuenta con un botón flotante para añadir un ejercicio especializado que no se encuentre dentro de las opciones presentadas.



Nuevo ejercicio

Esta es un slide card, que se despliega cuando se hace tap botón flotante de la ventana anterior.

En esta se solicita ingresar el nombre del ejercicio, la descripción de este y una imagen/video demostrativo.

Se puede desaser de esta al deslizarla hacia abajo.



Perfinl del deportista

En esta ventana se le despliega el perfil del deportista, donde se muestra su altura, edad, posición y compocisión corporal.

En esta ventana el entrenador solo puede cambiar la posición del jugador.

Interface final de usaurio

9:58

ITESO Universidad Jesuita de Guadalajara

Username

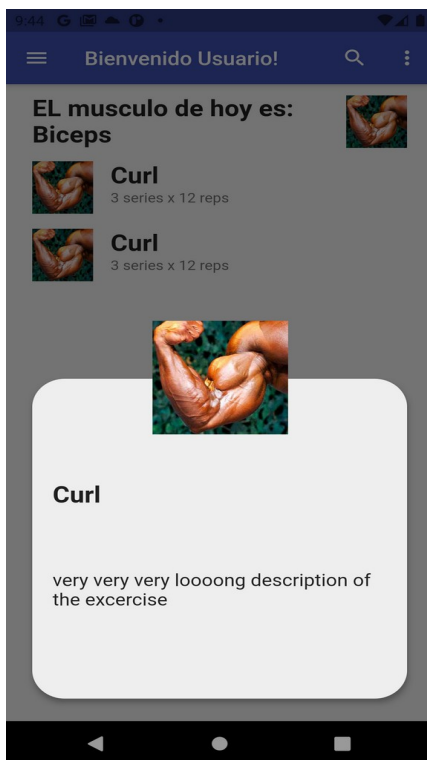
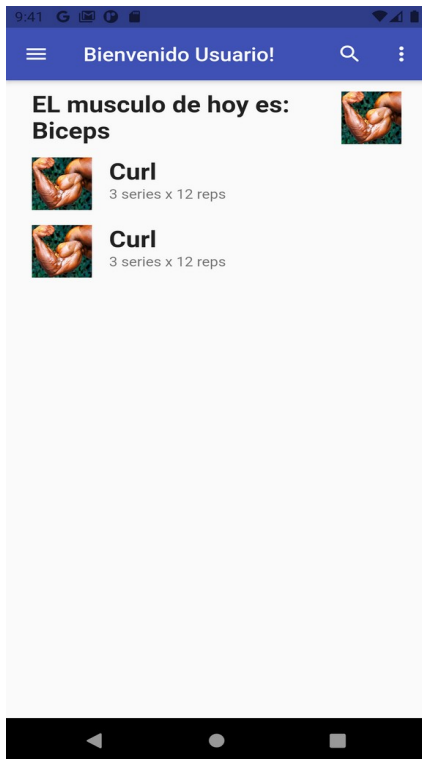
Password

INGRESA

Esta ventana cuenta con dos campos de entrada de texto y un botón para llevar a cabo la verificación.

El primer campo de texto es para ingresar el expediente del alumno/entrenador. Mientras que el segundo campo es para ingresar la contraseña del mismo.

Ventanas del deportista



Rutina del Día/Ventana principal

En esta ventana se le despliega al usuario los músculos a trabajar en ese día y una imagen alusiva a estos.

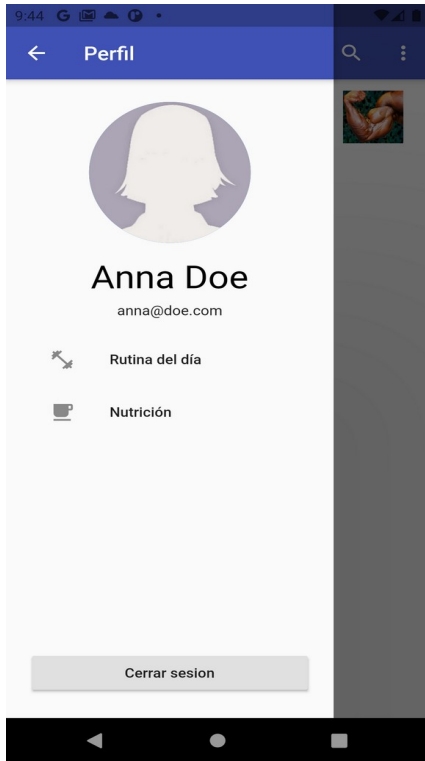
Después de esto, se muestra una lista con cada ejercicio, junto con la cantidad de series y repeticiones para cada uno. Al hacer tap en el ejercicio, se despliega una descripción del mismo.

Por último en la barra de la aplicación, se muestra un icono para desplegar un drawer, que funciona como menú de navegación, y un saludo personalizado para cada usuario.

Descripción de ejercicio

Esta es un slide card, que se despliega cuando se hace tap en algún ejercicio. En esta se muestra una imagen/video demostrativo, el nombre del ejercicio, la carga a realizar y por último una descripción del mismo.

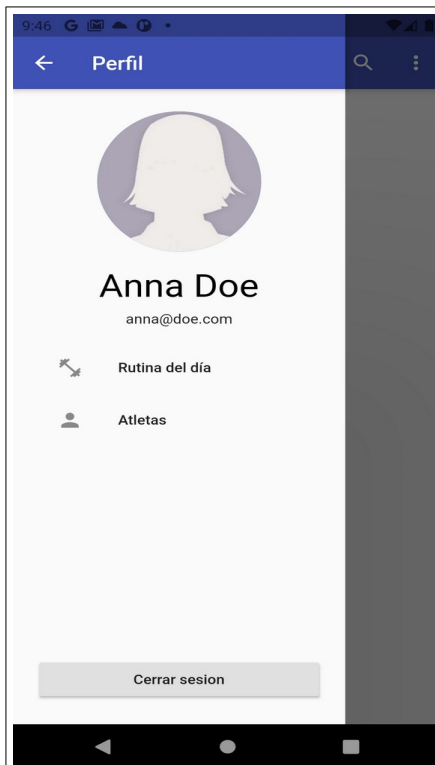
Se puede deshacer de esta al deslizarla hacia abajo o haciendo tap en la pantalla.



Menú de navegación

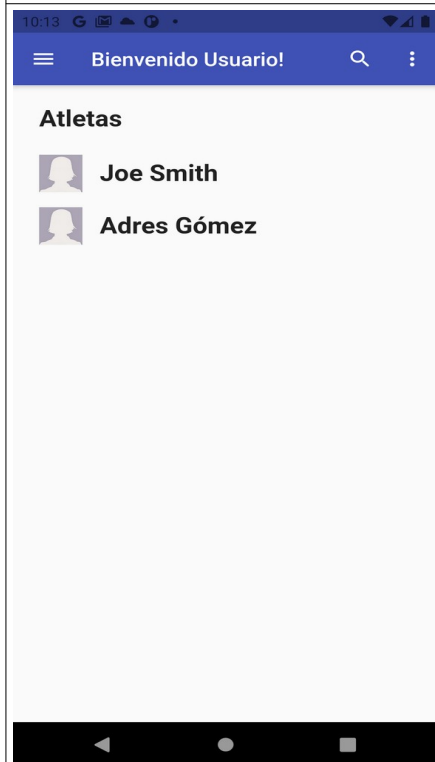
Este es un drawer que muestra la foto, nombre del usuario y correo electrónico. Le permite seleccionar entre la ventana de la rutina del día y su perfil nutricional o cerrar sesión

Ventanas del entrenador



Menú de navegación

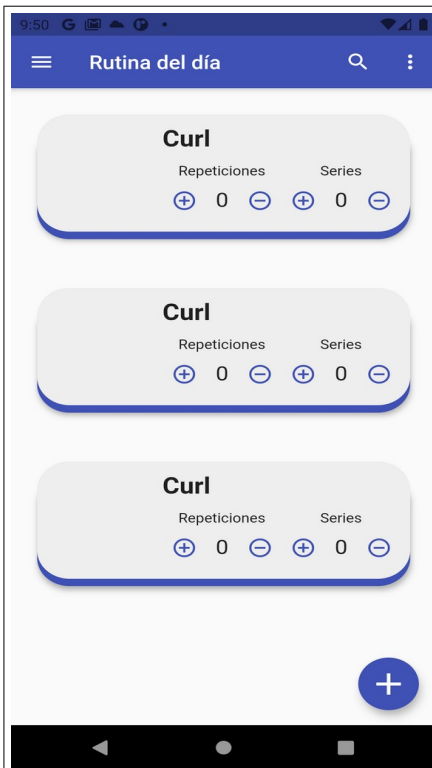
Este es un drawer que muestra la foto, nombre del usuario y su correo electrónico. Le permite seleccionar entre la ventana de la rutina del día y la lista de deportistas bajo su cuidado.



Lista de atletas/Ventana principal

En esta ventana se le despliega al entrenador los deportistas bajo su cuidado. Al hacer tap en el atleta, se despliega una descripción del mismo donde el entrenador puede asignarle una rutina personal.

Por último, en la barra de la aplicación, se muestra un icono para desplegar un drawer, que funciona como menú de navegación, y una opción de búsqueda de deportista.



Nueva Rutina del Día

En esta ventana se le despliega al entrenador los ejercicios disponibles, en la aplicación, separados por musculos principales

Cada elemento de la lista tiene un contador de series y repeticiones, para así ajustar la carga de trabajo de la rutina

Por ultimo cuenta con un botón flotante para añadir un ejercicio especializado que no se encuentre dentro de las opciones presentadas.



Nuevo ejercicio

Esta es un slide card, que se despliega cuando se hace tap botón flotante de la ventana anterior.

En esta se solicita ingresar el nombre del ejercicio, la descripción de este y una imagen/video demostrativo.

Se puede desascer de esta al deslizarla hacia abajo.

Contador

Se implementó un widget contador, el cuál cuenta con un elemento de texto, un contenedor donde se despliega el valor del contador y dos botones para aumentar o disminuir este contador, además de recibir dos funciones de callbac para realizar al pulzar cada boton.

La construccion de cada uno de sus elementos, están modelados como metodos privados de la clase, como se muestra a continuación:

```
Container _buildCountContainer() {  
  return Container(  
    child: Text(  
      '$_count',  
      textScaleFactor: 1.5,  
    ),  
  );  
}
```

```
IconButton _buildAddIconButton() {  
  return IconButton(  
    icon: Icon(Icons.add_circle_outline),  
    color: Colors.indigo,  
    onPressed: () {  
      setState(() {  
        _count++;  
      });  
      widget.onIncrement(_count);  
    },  
  );  
}
```

```
IconButton _buildRemoveIconButton() {  
  return IconButton(  
    icon: Icon(Icons.remove_circle_outline),  
    color: Colors.indigo,  
    onPressed: () {
```

```

        setState(() {
          if (_count >= 1) _count--;
        });
        widget.onDecrement(_count);
      },
    );
  }
}

```

Lo cual nos deja una estructura del widget más compacta y sencilla de entender.

```

Container(
  child: Column(
    children: [
      Text('${widget.title}'),
      Row(
        crossAxisAlignment: CrossAxisAlignment.center,
        children: [
          _buildAddIconButton(),
          SizedBox(
            width: 5,
          ),
          _buildCountContainer(),
          SizedBox(
            width: 5,
          ),
          _buildRemoveIconButton(),
        ],
      ),
    ],
  ),
);

```

Uso de la camara

Para la adición de nuevos ejercicios por parte del entrenador, se necesita el uso de la camara, por lo que se implementó una función dentro de la clase NewExercise, donde se utiliza un image picker para acceder a la camara y guardar la foto tomada.

```
Future _pickImage() async {
  final picker = ImagePicker();
  final PickedFile choosenImage = await picker.getImage(
    source: ImageSource.camera,
    maxHeight: 720,
    maxWidth: 720,
    imageQuality: 85,
  );
  setState(() {
    if (choosenImage != null) {
      _image = File(choosenImage.path);
    } else {
      print('No image selected');
    }
  });
}
```

Bloc para manejo de ventanas de entrenador.

Para el manejo de ventanas del entrenador se utilizó un bloc provider en el body del la ventana home del entrenador,este se encuentra dentro de un scaffold de la siguiente manera:

```
Scaffold(
  appBar: AppBar(
    title: Text('Bienvenido ${widget.coach.nombre}!'),
    actions: <Widget>[
      IconButton(icon: Icon(Icons.search_rounded), onPressed: ()
    {}),
      IconButton(icon: Icon(Icons.more_vert_rounded), onPressed:
    () {}),
    ],
  ),
```

```

drawer: CoachProfile(bloc: coachBloc),
body: BlocProvider(
  create: (context) {
    coachBloc = CoachBloc();
    coachBloc.add(ShowAthleteListEvent());
    return coachBloc;
  },
  child: BlocConsumer<CoachBloc, CoachState>(
    listener: (context, state) {},
    builder: (context, state) {
      if (state is NewRoutineState) {
        return CoachRoutine();
      } else if (state is AthleteProfile) {
        //TODO: show single athlete information card
      } else if (state is ShowAthletsState) {
        return CoachAthletes(
          listaAtletas: widget.coach.atletas,
        );
      }
    },
  )),
);

```

Esta implementación nos presenta un problema al utilizar el Drawer como un menú de navegación entre pantallas, ya que se le tiene que pasar el bloc como argumento, lo que provoca que no se ejecute de manera correcta las acciones. Para solucionar esto se implementará una barra de navegación en la parte inferior de la pantalla.

Consumo de API:

El consumo de la api, es el mismo consumo utilizado para el backend implementado con strapi. Actualmente este se encuentra de manera local, y se está trabajando en su implementación en un servidor.

El protocolo utilizado es HTTP, por lo que al no ser un protocolo de seguridad, se te tuvo que añadir la siguiente línea en el `AndroidManifest.xml`

```
android:usesCleartextTraffic="true"
```


Modelos Json:

Con la finalidad de utilizar la información proporcionada por la API, se crearon 3 modelos de datos basados en los Json retornados por esta. Estos son Atletas, Coach y deporte, los cuales se muestran a continuación:

```
const Atletas({
  this.id,
  this.nombre,
  this.apellido,
  this.pocision,
  this.idAtleta,
  this.entrenador,
  this.deporte,
  this.publishedAt,
  this.createdAt,
  this.updatedAt,
});

const Coach({
  this.id,
  this.idEntrenador,
  this.apellido,
  this.nombre,
  this.deporte,
  this.publishedAt,
  this.createdAt,
  this.updatedAt,
  this.atletas,
});

const Deporte({
  this.id,
  this.idDeporte,
  this.deporte,
  this.femenil,
  this.publishedAt,
  this.createdAt,
```

```
    this.updatedAt,  
  });
```

Autenticación:

la autenticación se realiza en la pantalla de login, donde se solicita a la api cualquier elemento de la colección de entrenador que contenga el nombre de usuario. Si no se encuentra en entrenador se realiza la misma solicitud en la colección de Atleta. De no encontrarse ahí, se mostrará un mensaje de error.

```
List<Coach> coachElement = await _getCoachList(coachQuery);  
List<Coach> athleteElement =  
    await _getCoachList(athleteQuery);  
if (coachElement != null) {  
    Navigator.push(  
        context,  
        MaterialPageRoute(  
            builder: (context) =>  
                CoachHome(coach: coachElement[0])));  
} else {}
```

Comentarios

El no tener el mismo historial académico que mis compañeros de la carrera de sistemas, se puede apreciar por mi lento avance en cuanto a entregables. Pero para estos últimos el avance fue considerable ya que se dedicó más tiempo en la coompección de base de datos relacionales y su modelado, además de cómo realizar consultas en las mismas.

TODO:

Add bloc logic for athletes

Add hive for coach routine creator

Add athlete extensive profile

[REPOSITORIO](#)