

MIRETOFITNESS

Descripción

Se trata de una App para realizar retos que tenga que ver con movimientos y cardio, ya que usa el sensor del podómetro para calcular tus movimientos y así realizar sus tareas.

Es capaz de calcular las kcal que quemas depende de tu peso, y depende de la intensidad del ejercicio, con una fórmula que se ideó en Estados Unidos que te daba una aproximación de cuantas calorías podrás quemar dependiendo del ejercicio, de tu peso y por cada 1000 pasos.

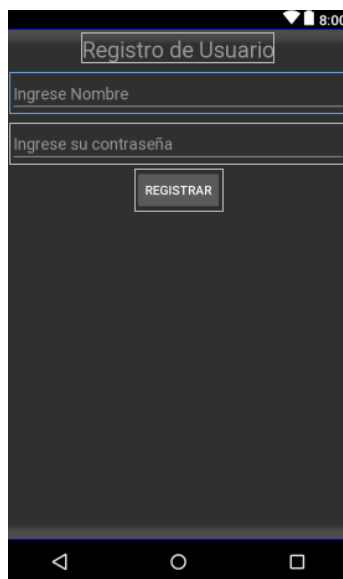
La app permite diseñar tus planes de entrenamiento y poder compartirlos con tus amigos, o realizar planes de entrenamiento preestablecidos.

Los planes constan de 5 ejercicios que tendrás que realizar teniendo el móvil encima.

Cuando llegues a la mitad de un ejercicio, éste te avisará y cambiará sus colores a un amarillo para así hacerte ver que has llegado al ecuador del ejercicio. Y una vez finalizado te avisará de ello y de que puedes pasar al siguiente ejercicio. (Recordemos que son 5).

Una vez llegado al final, te calculará una puntuación que guardará en tu perfil en una BD, y podrás ver diariamente tu evolución, y si completaste los retos de tus amigos o de la app.

Clases :



La clase principal es llamada **main_activity** y es una **activity de Login**.

En ella simplemente obtenemos los datos de los EditText, y los mandamos a un script en mi servidor, donde los compara y te obtiene tus datos al entrar en la siguiente activity.

También desde aquí da acceso a la **activity de registro** que simplemente obtiene los datos de el EditText y crea un nuevo Usuario en mi base de datos.

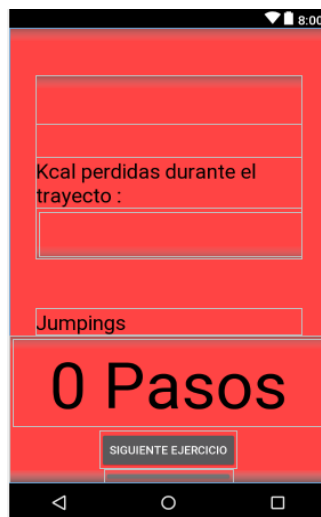
La siguiente clase es el **menú principal**, donde se encuentran todos los botones para acceder a los diferentes apartados de la App.

Aquí hace una carga de los datos como tus últimas puntuaciones y tus datos para usarlos en la app, justo al iniciarse la activity, usando Json como método de comunicación APP-SERVER



Además tendrás que ingresar tu peso antes de realizar cualquier rutina para que así se calcule más exactamente las kcal que pierdes al realizar las rutinas.

Te da también la información de tu id, por si quieres compartir tus rutinas con tus amigos y ponerles diferentes retos.



Una vez que seleccionas una rutina, pasamos a la **clase Entrenamiento** que es donde se implementa el sensor del podómetro y calcula los datos necesarios para obtener los puntos y mostrarte las kcal perdidas y cuándo llegas al final del ejercicio.

La siguiente clase es la de **resultado** que es sólo una pantalla donde van a salir los resultados obtenidos durante el ejercicio en unos TextView, y, volviendo al inicio, mandará a la base de datos esos resultados para guardártelos en tu perfil personal e ir viendo tu evolución.



Además de las activities, he creado una **clase Global** como en el resto de actividades que son variables globales que se comparten durante la ejecución de toda la aplicación y que son de donde obtienes los datos para sacar información y así no necesitas tanto paso de parámetros entre activities.

Y la clase rutina que consta de 5 nombres de actividades, 5 cantidad de pasos y 5 dificultades que son los atributos de los que consta una rutina, ya sea para cargar la de un amigo o para realizar la tuya.



En el **activity crear rutina**, te indica los pasos para crear una rutina y guardarla como tu rutina y poder compartirla o realizarla tu.

En los campos, ingresas los parámetros de tu rutina y al obtenerlos todos los lleva a una base de datos donde se guarda para poder tenerlos cuando tu desees o compartirlos.



En la **activity rutina amigo**, es donde podrás realizar las rutinas que están subidas a la BD, simplemente poniendo la id del creador y una vez obtenidos los datos de la base de datos podrás realizarla y guardar tus resultados en tu perfil.

Como en el resto de APP, hay clases especiales como **RegisterRequest** que son para comunicarme con el servidor donde tengo la base de datos y enviarle mediante POST los parámetros para obtener los datos.

- ☒ AñadirReto
- ☐ CrearRutina
- ☐ Entrenamiento
- ☐ EntrenamientoAmigo
- ☐ GlobalClass
- ☒ GuardarPuntos
- ☒ LoginRequest
- ☐ MainActivity
- ☐ MenuPrincipal
- ☒ ObtenerRutina
- ☐ Puntos
- ☐ Puntuaciones
- ☒ RegisterRequest
- ☐ Registrarse
- ☐ Reto
- ☐ RutinaAmigo
- ☐ Score

En todas mandas unos parámetros por post, y devuelve por Json varios parámetros para interpretarlos más tarde en el código Java.

```
public class LoginRequest extends StringRequest {
    private static final String LOGIN_REQUEST_URL="http://www.pgongorasoria.com/Notas/Login.php";
    private Map<String, String> params;

    public LoginRequest(String username, String password, Response.Listener<String> listener) {
        super(Method.POST, LOGIN_REQUEST_URL, listener, errorListener: null);
        params= new HashMap<String, String>();
        params.put("username", username);
        params.put("password", password);
    }

    @Override
    public Map<String, String> getParams() { return params; }
```

Una llamada a este método sería la siguiente :

```
loguear.setOnClickListener((v) → {

    final String user_name = usuario.getText().toString();
    final String password = pass.getText().toString();

    Response.Listener<String> responseListener = (response) → {
        try {
            JSONObject jsonResponse = new JSONObject(response);
            boolean success = jsonResponse.getBoolean( name: "success");
            if (success) {
                int id = Integer.parseInt(jsonResponse.getString( name: "ID"));
                String name = jsonResponse.getString( name: "NOMBRE"); //obtenemos el campo name
                String pass = jsonResponse.getString( name: "PASSWORD");

                Intent intent = new Intent( packageContext: MainActivity.this, MenuPrincipal.class);

                g.setIdUser(id);

                MainActivity.this.startActivity(intent); //realizamos el intent

            } else { // si no esta correcto lanzar un error
                AlertDialog.Builder builder = new AlertDialog.Builder( context: MainActivity.this);
                builder.setMessage(" Error en el Login")
                    .setNegativeButton( text: "Retry", listener: null)
                    .create().show();
            }
        } catch (JSONException e) {
            e.printStackTrace();
        }
    };

    LoginRequest loginRequest = new LoginRequest(user_name,password, responseListener );
    RequestQueue queue = Volley.newRequestQueue( context: MainActivity.this);
    queue.add(loginRequest);

});
```